
Telelogic Tau

UML Coffee Machine Tutorial

本書は、Telelogic Tau バージョン 4.2、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

© Copyright IBM Corporation 1997, 2008.

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あり、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。

〒 106-8711
東京都港区六本木 3-2-12
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、製造元に連絡してください。

Intellectual Property Dept. for Rational Software
IBM Corporation
1 Rogers Street
Cambridge, Massachusetts 02142
U.S.A.

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。

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。

サンプル・コードの著作権

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のめかしたり、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。

IBM の商標

IBM、IBM ロゴ、ibm.com、および Tau は、International Business Machines Corporation 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。これらおよび他の IBM 商標に、この情報の最初に現れる個所で商標表示 (® または ™) が付されている場合、これらの表示は、この情報が公開された時点で、米国において、IBM が所有する登録商標またはコモン・ロー上の商標であることを示しています。このような商標は、その他の国においても登録商標またはコモン・ロー上の商標である可能性があります。IBM および関連の商標については、www.ibm.com/legal/copytrade.html をご覧ください。

他社の商標

Adobe、Adobe ロゴ、PostScript 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、Sun Microsystems, Inc. の米国およびその他の国における商標です。

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 2003、Windows XP、Windows Vista および / またはその他の Microsoft 製品は、Microsoft Corporation 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。

Pentium は、Intel Corporation の商標です。

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。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

サポートへのお問い合わせ

Telelogic 製品のサポートと情報は、Telelogic サポートサイトから IBM Rational Software Support に移行中です。この移行期間中は、サポートの連絡先がお客様によって異なります。

製品サポート

- 2008 年 11 月 1 日より前に Telelogic 製品を取引されたお客様は、Tau サポート ウェブサイトをアクセスしてください。製品情報の移行後に、IBM Rational Software Support site に自動で転送されます。
- 2008 年 11 月 1 日より前に Telelogic 製品のライセンスをお持ちではなかった新規のお客様は、[IBM Rational Software Support site](#) をアクセスしてください。

お客様サポートにお問い合わせいただく前に、問題を説明するために必要な情報をご用意ください。IBM ソフトウェアサポート担当員に問題を説明する際には、担当員が迅速に問題を解決できるように、問題の具体的な内容と必要な背景情報をすべて伝えてください。あらかじめ以下の情報をご用意ください。

- 問題発生時に使用していたソフトウェアとそのバージョン
- 問題に関連したログ、トレース、メッセージなど
- 問題を再現できるかどうか。再現できる場合はその手順
- 回避策があるかどうか。ある場合は、その回避策の内容

その他の情報

Rational ソフトウェア製品、ニュース、イベント、その他の情報については、[IBM Rational Software Web site](#) をご覧ください。

サポートへのお問い合わせ	iv		
はじめに	1		
このチュートリアル	の目的	1	
コーヒー販売機	2		
コーヒー販売機	の振る舞い	2	
モデルとダイアグラム	3		
概要	3		
エンティティ	の削除	3	
ダイアグラム	要素の作成	ツールバー	3
ユーザー インターフェイス	4		
概要	4		
ワークスペース	ウィンドウ	5	
デスクトップ	5		
ショートカット	バー	5	
出力ウィンドウ	5		
ショートカット	メニュー	5	
ワークスペース	6		
概要	6		
ワークスペース	の作成	6	
プロジェクト	7		
概要	7		
プロジェクト	の作成	7	
ユース ケース図	9		
概要	9		
ユース ケース図	の作成	9	
クラス図	11		
概要	11		
クラス図	の作成	11	
アクティブ クラス	とパッシブ クラス	12	
非形式クラス	12		
クラスの分割	13		
コンポーネント図	13		
コンポーネント図	の作成	14	
シグナル	15		
概要	15		
シグナル	の定義	15	
インターフェイス	の定義	16	
インターフェイス	の関係	18	
ポート	19		

シーケンス図 21	
概要 21	
シーケンス図の作成 21	
ユース ケースとサブジェクト フレーム 23	
参照付きシーケンス図 24	
モデル ナビゲータ 26	
状態機械図 28	
概要 28	
Hardware クラスの状態機械図 28	
状態 (ステート) 指向構文 28	
状態機械図複合ステート 29	
遷移指向構文 30	
Controller クラスの状態機械図 31	
合成構造図 34	
概要 34	
合成構造図の作成 34	
パート 34	
ポート 35	
コネクタ 35	
ポートとインターフェース 36	
関係 37	
概要 37	
関連 37	
複合 37	
チェック 39	
概要 39	
オートチェック 39	
チェック 39	
ビルド アーティファクト 40	
概要 40	
モデル ベリファイヤ (Model Verifier) のセットアップ 40	
ビルド アーティファクトの作成 40	
モデル ベリファイヤ (Model Verifier) 42	
概要 42	
コーヒー販売機の検証 42	
メッセージマトリックスの作成 42	
パラメータのウォッチ 44	
トレース 44	
ユース ケースの検証 45	

参照シーケンス図	47
反復と追加	48
目的	48
タイマー	48
構造データ	49
まとめ	53
モデル	53
エディタ	53
テスト	53
ワークフロー	53
今後の学習	54

はじめに

このチュートリアルの目的

このチュートリアルの目的は、読者が **Tau** と **UML** 言語に慣れ親しむことです。このチュートリアルは、**Tau** の操作経験はないが、**UML** とステート マシンの基本概念について知識がある読者を対象としています。

ここでは、コーヒー販売機のデザイン、つまり初期のモデリングから、最終的なデザインのシミュレーションまでを行います。

このチュートリアルでは、ワークスペースを起動する方法とプロジェクトを新しく作成する方法を手順を追って説明します。このプロジェクトを使用して **UML** 仕様を作成します。続いてモデル ベリファイヤ (**Model Verifier**) でデザインのシミュレーションを行える段階まで、順を追ってデザインを完成させていきます。このチュートリアルの説明に従えば、必要な手順をすべて実行できます。各作業手順の詳細については、**Tau** のオンライン ヘルプを参照してください。

本文の説明の余白部分に、ツールバー ボタンの絵が表示されていることがあります。これは、本文で参照されたボタンが簡単に見つかるようにするためのものです。通常、ボタンの絵は、そのボタンを最初に使用する箇所だけに表示されています。それぞれのボタンはさまざまなツールバーに属し、ツールバーの単位で表示 / 非表示を切り換えられます。そのため本文で参照されているボタンが現在のツールバーのセットに表示されていないことがあります。この場合は、まず、[ツール] メニューの [カスタマイズ] ダイアログからツールバーを選択して表示します。

ダイアグラムにシンボルを挿入する指示があるときには、必ずそのダイアグラム内でクリックしてダイアグラムをアクティブにします。最近開いたダイアグラムでも、直接ツールバーをクリックしてシンボルを挿入できない場合があります。まず、ダイアグラム内の任意の場所をクリックし、次に目的のシンボルをツールバーから選択します。

コーヒー販売機

コーヒー販売機の振る舞い

例題となるコーヒー販売機「CoffeeMachine」は、顧客「Customer」が必要な金額を投入すると1杯のコーヒーまたは1杯の紅茶用のお湯が出てくるものです。コーヒー販売機で処理できるのは、「5」と「10」の値の硬貨です。「5」は紅茶1杯の価格で、「10」はコーヒー1杯の価格です。振る舞いは次のようになります。

- 「10」の値の硬貨を投入され、「Coffee」ボタンが押されると、「Customer」は1杯のコーヒーを受けとります。
- 「10」の値の硬貨を投入され、「Tea」ボタンが押されると、「Customer」は1杯の紅茶とつり銭を受けとります。
- 「5」の値の硬貨を投入され、「Coffee」ボタンが押されると、「Customer」は硬貨を受けとります。
- 「5」の値の硬貨を投入され、「Tea」ボタンが押されると、「Customer」は1杯のお湯を受けとります。

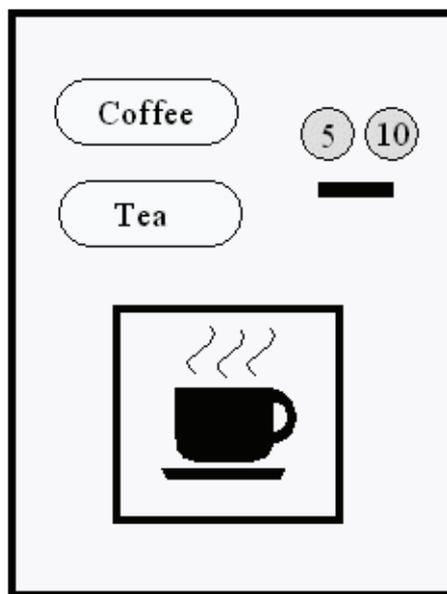


図 1: コーヒー販売機

モデルとダイアグラム

概要

ダイアグラムとは、要素の集まりと要素間の関係を示す UML モデルを視覚的に表現するものです。このチュートリアルでは、ツールの機能を使用して、モデルを直接操作する方法とダイアグラムを使用する方法の両方を行います。ユーザーはさまざまなダイアグラムを使用して作業します。各ダイアグラムはモデルの特定のビューを表示します。ダイアグラムにクラスなどの新しいエンティティが描かれると、そのエンティティはモデルの一部になります。異なるビューを表示するために、同じクラスを同じダイアグラムまたは異なるダイアグラムに描き直すこともあります。モデルに保存される情報は、エンティティで表現するすべての記述をまとめたものです。

エンティティの削除

Tau を使用した設計では、モデルとダイアグラムを区別することに注意が必要です。前の「概要」で述べたとおり、クラスなどのエンティティはダイアグラムに追加するとすぐにモデルに組み込まれます。しかし、あるダイアグラムからあるクラスを削除したからといって、そのクラスがモデルから削除されるわけではありません。なぜなら同じクラスが他のダイアグラムで使用されている可能性があるためです。ただし、エンティティをモデルから削除することはできます。これには、エンティティを右クリックすると表示される、状況に依存したショートカットメニューで、[削除]ではなく[モデルの削除]をクリックする必要があります。

ダイアグラム要素の作成ツールバー

[ダイアグラム要素の作成] ツールバーは、ダイアグラムを使用する場合にのみアクティブになります。ツールバーをアクティブにするには、ダイアグラム内をクリックします。[ダイアグラム要素の作成] ツールバーのボタンは、一般的な操作方法で利用できます。つまり、ツールバーのボタンをクリックし、次にダイアグラム内をクリックしてボタンで決められたエンティティを配置します。

ツールバーのクイックボタンは、状況に依存して変化することがあります。したがって、ボタン操作の結果は、現在のダイアグラム内で選択されているエンティティや、そのエンティティとボタンとの関係によって変わります。多くの場合、ショートカットメニュー（右クリックで表示される）を使用してモデルの動作から補助的な情報を得ることができます。たとえば、シーケンス図でメッセージを描く場合は、[メッセージライン] ボタンをクリックし、送信側ライフラインをクリックします。次に受信側ライフラインを右クリックすると、ショートカットメニューに、現在スコープ内にある全シグナルを含むドロップダウンボックスが表示されます。



ユーザー インターフェイス

概要

Tau ユーザー インターフェイスには、次の 4 つの領域があります。

1. ワークスペース ウィンドウ
2. デスクトップ
3. 出力ウィンドウ

このチュートリアルでは、上記以外の領域がアクティブになることもあります。たとえば、テスト時のウォッチ ウィンドウなどです。これらの領域は、ドラッグしてウィンドウ内で位置を変えたり、ユーザー インターフェイスのメイン ウィンドウの外側に移動したりできます。

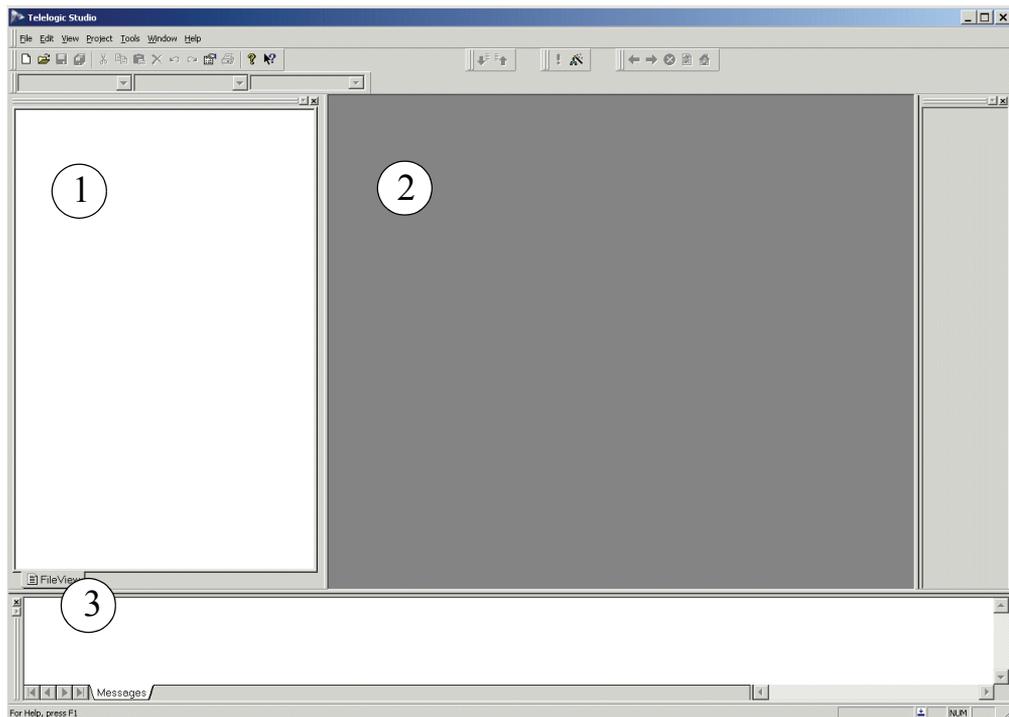


図 2: ユーザー インターフェイス

ワークスペース ウィンドウ

ワークスペース ウィンドウには、モデルに含まれているエンティティが表示されません。各種の情報を表示できるように、さまざまなビューが用意されています。[ファイル ビュー]は、ファイルとして表現されるすべての要素を表示します。[モデル ビュー]は、すべての UML 要素を表示します。[モデル ビュー]は、ダイアグラムを追加する場合や、モデル内を直接操作する場合に選択します。また、[インスタンス]と呼ばれるビューもあります。このビューは、モデルベリファイヤ (Model Verifier) を使用してシステムをシミュレートする場合にのみ有効になります。

デスクトップ

デスクトップは、ダイアグラムやドキュメントを開いたときに表示される領域です。この領域でモデルをさまざまなダイアグラムで表示しながら作業します。

ショートカット バー

オプションのショートカット バーを使用すると、頻繁に使用するツールバーを専用の領域に配置したり、独自に作成したスクリプトへのショートカットを作成したりできます。ショートカット バーはオプションです。

出力ウィンドウ

出力ウィンドウは、イベントのログ出力やエラーおよび警告の表示に使用されます。

ショートカット メニュー

エンティティを右クリックすると、ショートカット メニューが表示されます。このメニューには、多くの場合、操作状況に依存したコマンドが表示されます。

ワークスペース

概要

ワークスペースは、個人用の作業領域です。この領域で個々のプロジェクトの作業を行います。特定のプロジェクトに関連しないファイルを組み合わせることもできます。ワークスペースは複数定義できますが、同時に作業できるワークスペースは1つだけです。ワークスペースは他のユーザーと共有できません。ワークスペースに含まれている情報は、拡張子 *.ttw のテキスト ファイルに保存されます。

ワークスペースの作成

ここでは、例題であるコーヒー販売機の開発用のワークスペースを作成します。以下の操作を行います。

1. **Tau** を起動します。
2. [ファイル]メニューで[新規...]をクリックします。[ファイル]、[プロジェクト]、[ワークスペース]、[テンプレート]の4種類のタブを持つダイアログが表示されます。
3. [ワークスペース]タブをクリックします。
4. [ワークスペース名]フィールドに「Tutorial」という名前を入力し、[場所]フィールドにワークスペース ファイル (Tutorial.ttw) の保存場所を指定します。[OK]をクリックします。ワークスペース ウィンドウの[ファイルビュー]に、新しいワークスペースが表示されます。

次の手順はプロジェクトの追加です。

プロジェクト

概要

プロジェクトを使用してワークスペースの内容をグループ分けできます。たとえば、ワークスペース「Tutorial」にこのコーヒー販売機を含む複数の異なる例題を格納し、各例題に1つのプロジェクトを割り当てることができます。ダイアグラムとドキュメントはプロジェクト間で移動できます。プロジェクトは個人用ではなく、複数のユーザーで共有できます。プロジェクトの持つ情報は、拡張子*.tppが付くテキストファイルに保存されます。

プロジェクトの作成

ここでは、例題であるコーヒー販売機用のプロジェクトを作成します。以下の操作を行います。

1. [ファイル]メニューで[新規...]をクリックします。[プロジェクト]タブをクリックします。各種アプリケーション作成用の選択項目を持つダイアログが表示されます。
2. ダイアログから[UML(モデルベリファイ用)]を選択します。
3. [プロジェクト名]フィールドに「CMdesign」という名前を入力し、[場所]フィールドにプロジェクトファイルの保存場所を指定します。[現在のワークスペースに追加する]を選択し、[OK]をクリックします。
4. モデルのファイル名とファイルの場所がそれぞれ[新しいUMLモデルのファイル名]フィールドと[ファイルディレクトリ]フィールドに入力された、2番目のダイアログが表示されます。[1つのファイルと1つのパッケージのプロジェクト]が選択されていることを確認し、[次へ]をクリックします。
5. 3番目のダイアログが表示されます。プロジェクトを表すファイル名とモデルを表すファイル名がそれぞれ[プロジェクト名]フィールドと[ファイル名]フィールドに表示されます。[完了]をクリックします。作成したプロジェクトがワークスペースウィンドウに表示されます。

注：

ワークスペースウィンドウ内のアイコンの左側にあるプラス記号(+)は、そのアイコンが折りたたまれており、詳細を表示できることを示しています。このアイコンの構造を展開するには、プラス記号をクリックします。ワークスペースウィンドウ内の折りたたまれたアイコンを選択し、テンキーパッドの乗算キー(*)を押すと、下部構造全体を展開できます。下部構造を折りたたむには、ルートアイコンのマイナス記号(-)をクリックします。

- ツリー構造を展開します。[ファイル ビュー]に作成済みファイルが表示され、[モデル ビュー]に空のパッケージが表示されます。[モデル ビュー]には、UML の Tau 表記の内部構造に関する情報も表示されます。この情報は、[Library] および [Predefined] という名前のパッケージ内にあります。

ユース ケース図

概要

ユース ケース図は、ある1つのシステムのユース ケースとアクター間の関係を表現します。1つのユース ケース図内で、サブジェクト フレームを使用してユース ケースをグループ化できます。

ユース ケース図の作成

ここでは、例題であるコーヒー販売機のユース ケースのためのユース ケース図を作成します。コーヒー販売機のユース ケースについては説明済みです。2 ページの「[コーヒー販売機の振る舞い](#)」を参照してください。

1. ワークスペース ウィンドウに [モデル ビュー] が表示されていることを確認します。
2. 「CMdesign」 パッケージを選択します。右クリックしてショートカットメニューから [新規] を選択し、[コラボレーション] をクリックします。[モデル ビュー] に作成された コラボレーションに「UserDrinks」という名前を付けます。

注：

モデル エンティティの名前を変更するには、要素を選択してF2キーを押すか、または要素を選択して名前を1度クリックします。これにより名前の文字列を編集できるようになります。

モデル内のコラボレーションを使用して、ユース ケース図をカプセル化したり、後にシーケンス図で表現するユース ケースをカプセル化したりします。

3. コラボレーションを選択します。右クリックしてショートカットメニューから [新規] をポイントし、[ユースケース図] をクリックしてユースケース図を挿入します。

ユース ケース図のアイコンが [モデル ビュー] で表示されます。ユース ケース図がデスクトップに表示されます。ダイアグラム内をクリックしてアクティブにします。

使用可能なシンボルがツールバーで強調表示されます。シンボルの上にマウス ポインタを置くとツールチップによってシンボル名が表示されます。



4. ダイアグラムにシンボルを挿入するには、シンボルを表すクイックボタンをクリックし、次にデスクトップ上のダイアグラムをクリックします。この方法で**アクター**をダイアグラムに配置します。そのアクターに「**Customer**」クラスを与えます。構文に従ってクラス名の前にコロンが付きます。アクターには名前を付けることができますが、必須ではありません。

「**Customer**」クラスは、設計のこの段階ではバインドされません。これは2つの方法で確認できます。1つは、タイプ名の下にバインドエラーを示す赤い波線が表示されます(10ページの図3:「ユース ケースとアクター」参照)。もう1つは、出力ウィンドウ([オート チェック]タブ)にエラーメッセージが表示されます。



5. ユース ケース シンボルをクリックし、ダイアグラム内に配置します。ユース ケースに「**MakeCoffee**」という名前を付けます。このユース ケースを選択して、アクターへの関連ラインを作成します。関連ラインを作成するには、ユース ケース シンボルの3つのライン ハンドルのうちの左側のもの([関連ライン])を、ユース ケース シンボルからアクターまでドラッグします。10ページの図3:「ユース ケースとアクター」を参照してください。

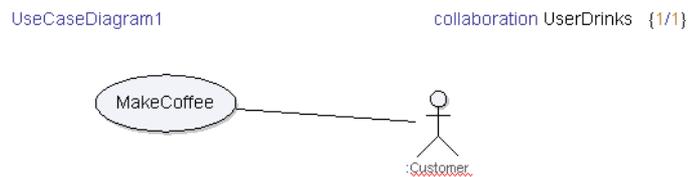


図 3: ユース ケースとアクター

6. [標準] ツールバーの [保存] ボタンを押して、作業内容を保存します。

このシステムで考えられるユース ケースの1つを含んだユース ケース図が完成しました。このダイアグラムには、シーケンス図の作成時にさらにユース ケースを追加します。

クラス図

概要

クラス図は、システムを構成するオブジェクトの種類とオブジェクト間の関係を表現します。また、クラスの属性と操作も示します。

クラス図の作成

ここでは、例題であるコーヒー販売機をモデリングするためにクラス図を使用します。

1. ワークスペース ウィンドウに [モデル ビュー] が表示されていることを確認します。
2. 「CMdesign」 パッケージを選択します。右クリックしてショートカットメニューで [新規] をポイントし、[クラス図] をクリックします。
3. クラス図のアイコンが [モデル ビュー] に表示されます。そして、ダイアグラムがデスクトップに表示されます。ダイアグラムに「DomainModel」という名前を付けます。
4. [クラス シンボル] クイックボタンを使用してクラスを 2 つ追加します。このボタンをクリックし、次にダイアグラム内をクリックしてクラスを配置します。クラスに「CoffeeMachine」と「Customer」という名前を付けます。この 2 つのクラスは、それぞれコーヒー販売機と顧客を表します。11 ページの図 4: 「CoffeeMachine クラスと Customer クラス」を参照してください。



図 4: CoffeeMachine クラスと Customer クラス

注：

「Customer」クラスが作成されると、[オートチェック]タブのアクター型に関するエラーが解決されます。

アクティブ クラスとパッシブ クラス

クラスは、アクティブまたはパッシブのどちらにでもすることができます。アクティブ クラスは振る舞いを含みますが、パッシブ クラスは定義 (属性) だけを含みます。パッシブ クラスの例はデータ型です。作成した「CoffeeMachine」クラスは振る舞いを含んでいますので、アクティブに設定する必要があります。以下の操作を行います。

1. 「CoffeeMachine」クラス シンボルを右クリックします。
2. 表示されるショートカット メニューで [プロパティ] をクリックします。プロパティ ダイアログが開き、「CoffeeMachine」クラスのプロパティを表示します。
[フィルタ] フィールドで「クラス」を選択します。
3. [アクティブ] チェック ボックスをオンにします。このダイアログは、開いたままにしておくことができます。

注 :

プロパティ ダイアログは、ALT+Enter キーを押して開くことができます。

アクティブ クラスは、ダイアグラム内に縦辺が二重線のシンボルで表示されます。

12 ページの図 5: 「アクティブ クラスと外部クラス」を参照してください。



図 5: アクティブ クラスと外部クラス

非形式クラス

「Customer」クラスは、コーヒー販売機のデザインには属さないエンティティです。顧客は外部に存在してシステムと相互作用します。したがって、このクラスには**非形式**というステレオタイプを与えます。クラスに非形式というステレオタイプを与えると、モデルのシミュレーションを行う際にこのクラスのためのコードは生成されません。以下の操作を行います。

4. 「Customer」クラスを右クリックします。表示されるショートカットメニューで [ステレオタイプ] をクリックします。

5. [TTDPredefineStereotypes:informal] チェックボックスを選択します。ステレオタイプはダイアログが閉じるとすぐに有効になります。
6. 「Customer」のクラスシンボルを右クリックして[アクティブ]を選択します。

クラスの分割

ここでは、コーヒー販売機の制御部分とハードウェア部分を表す2つの新しいクラスを追加して、モデルを改良します。「Controller」クラスにはシステムのロジックを組み込み、「Hardware」クラスはハードウェアの動作をシミュレートします。

これらのクラスは別の方法で追加します。モデルにクラスを直接追加した後、エディタのモデリング機能を使用します。以下の操作を行います。

1. ワークスペース ウィンドウの[モデル ビュー]で「CMdesign」パッケージを探して、右クリックします。ショートカットメニューで[新規]をポイントし、[クラス]をクリックします。[モデル ビュー]にクラスアイコンが表示されます。
2. クラスに「Controller」という名前を付けます。
3. [モデル ビュー]で「CMdesign」パッケージを選択します。
4. 「CMdesign」パッケージを右クリックします。ショートカットメニューで[新規]をポイントし、[クラス]をクリックします。[モデル ビュー]にクラスアイコンが表示されます。
5. クラスに「Hardware」という名前を付けます。
6. [モデル ビュー]で「Hardware」クラスを選択します。「Hardware」クラスを右クリックします。ショートカットメニューでプロパティを変更して、「Hardware」をアクティブクラスにします。「Controller」クラスにも同じ操作を繰り返します。

注：

クラスをアクティブに設定するためにプロパティ ダイアログを表示する必要はありません。ショートカットメニューには、状況に依存した選択項目が表示されます。そのメニューのうちの1つの項目がクラスをアクティブにするかどうかを制御しています。

コンポーネント図

コンポーネント図はクラス図と密接な関係があり、クラス図のサブタイプと考えることができます。コンポーネント図は、識別可能で置き換え可能なシステムのコンポーネントを記述します。クラス図と似て、コンポーネント図にはクラスのポートとインターフェースが表示されます。

コンポーネント図の作成

コーヒー販売機の例をさらにモデリングしてゆくために、コンポーネント図を使用します。

1. ワークスペース ウィンドウに [モデルビュー] が表示されていることを確認します。
2. 「Controller」 クラスを右クリックします。ショートカットメニューで [プレゼンテーションの作成] をポイントします。表示されたダイアログの [新しいシンボル] タブで、表中の [New ComponentDiagram] 行をクリックします。
3. ダイアグラムのアイコンが [モデルビュー] に表示されて、デスクトップにダイアグラムが開きます。ダイアグラムに「ControlComponent」という名前を付けます。
4. 「Hardware」 クラスのアイコンを [モデルビュー] からデスクトップで開いているコンポーネント図「ControlComponent」にドラッグアンドドロップします。
5. [標準] ツールバーの [保存] ボタンを押して、作業内容を保存します。

ControlComponents



図 6: コーヒー販売機のコンポーネント

コンポーネント図は 14 ページの図 6: 「コーヒー販売機のコンポーネント」のようになるはずですが。

シグナル

概要

UML には、シグナルを定義するための専用のクラス シンボルがあります。このシンボルは定義済みステレオタイプの一つであり、<<signal>> ヘディングで示されます。UML モデル内で使用されるすべてのシグナルは定義されている必要があります。あるレベルで定義されたシグナルは、それよりも下位のレベルにあるすべてのエンティティから参照および使用できます。

シグナルの定義

ここでは、コーヒー販売機の振る舞いの実装に必要な、すべてのシグナルを定義します。クラス図の中にシグナルの定義を描いてゆきます。以下の操作を行います。

1. [モデルビュー] で「CMdesign」パッケージを選択します。右クリックしてショートカットメニューから [新規] をポイントし、[クラス図] を選択します。クラス図がワークスペース ウィンドウの [モデル ビュー] に表示されます。
2. そのダイアグラムに「Signals」という名前を付けます。ダイアグラムはデスクトップに表示されています。
3. ツールバーからシグナル シンボルを選択してダイアグラム内に配置します。ここでは、下表に示されている各シグナルを定義します (表中のシグナル 1 つにつきシグナル シンボル 1 つ)。



注：

シンボルを配置するとき Ctrl キーを押したままにすると、ツールバーの選択を保持するショートカットとして機能します。この方法で、ダイアグラムとツールバーの間を往復せずに同じ種類の複数のシンボルを配置できます。

「Customer」へ	「Customer」から	「Hardware」へ	「Hardware」から
CupOfCoffee	Coffee	FillWater	WaterOK
CupOfWater	Tea	FillCoffee	CoffeeOK
ReturnChange	Coin (Integer)	HeatWater	Warm

4. 「Coin」シグナルは整数型のシグナル データを 1 つ持ちます。このシグナル データは、投入された、硬貨の値を保持します。「Coin」シグナルを表すシンボルの中段の入力域に「Integer」型を追加します。16 ページの図 7: 「シグナルの定義」を参照してください。
5. [標準] ツールバーの [保存] ボタンを押して、作業内容を保存します。

注：

シグナル名や型などのエンティティの色付表示は、Tauが対応する定義を検出したことを示します。これは名前解決という機能であり、Tauを使った設計ではあらゆるレベルで動作します。IntegerはUMLの定義済み型の1つです。定義が存在しない場合は、名前に赤の下線が引かれます。

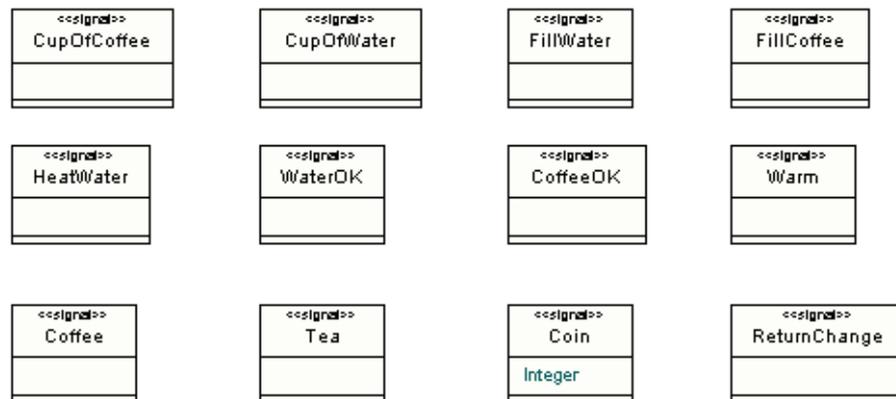


図 7: シグナルの定義

インターフェイスの定義

ここでは、環境と相互作用するために、シグナルを格納するインターフェイスを作成します。以下の操作を行います。

1. 「Signals」クラス図を開きます。
2. ツールバーから、インターフェイス シンボルを使用して以下の 2 つのインターフェイスを定義します。
 - 「FromUser」
 - 「ToUser」



シグナル

3. [モデル ビュー] 内でシグナルをインターフェイスにドラッグします。
 - 「Coin」シグナル、「Coffee」シグナル、および「Tea」シグナルは「FromUser」インターフェイスに移動します。
 - 「ReturnChange」シグナル、「CupOfCoffee」シグナル、および「CupOfWater」シグナルは「ToUser」インターフェイスに移動します。
4. [標準] ツールバーの [保存] ボタンを押して、作業内容を保存します。

注：

各シグナルはインターフェイスの機能となり、テキスト ラインはプレゼンテーション要素として処理されます。テキスト ラインは、テキストと同じように編集できるばかりでなく、カーソルをラインの起点に置き、**BackSpace** キーを押すと、ライン全体(プレゼンテーション要素)を削除できます。

インターフェイス シンボル内のシグナルを表示するには、インターフェイス シンボルを右クリックし、ショートカットメニューから[すべての操作の表示]をポイントします。シグナルパラメータの変更も可能です。

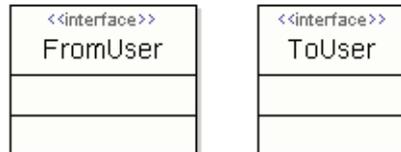


図 8: インターフェイス定義

インターフェイスの関係

インターフェイスを使用する準備が整いました。ここでは、2つのインターフェイスの間に関連 (Association) を作成し、「ToUser」から「Customer」に依存 (Dependency) を作成します。

1. 「ToUser」インターフェイス シンボルを選択し、「ToUser」から「FromUser」に向けて関連ラインのハンドルをドラッグします。この操作をすると、インターフェイスの一方が使用されている場合に常にもう一方も使用できるようになります。
2. 関連のナビの可能性を削除します。これには、関連ライン上のインターフェイスとの接点に近い部分を右クリックし、ショートカットメニューの [ターゲット] からサブメニュー内の [ナビ可能] の選択を解除します。さらに関連ライン上の相手のインターフェイスとの接点に近い部分を右クリックし、ショートカットメニューの [ソース] からサブメニュー内の [ナビ可能] の選択を解除します。
3. 「DomainModel」クラス図に移動します。これには、[ナビゲーション] ツールバーの [戻る] ボタンをクリックするか、[モデル ビュー] で目的のダイアグラムのアイコンをダブルクリックします。
4. 「ToUser」インターフェイスを [モデル ビュー] から「DomainModel」クラス図にドラッグします。「Customer」クラス シンボルを選択し、依存ラインのハンドルを「ToUser」インターフェイス シンボルまでドラッグします。これは、「Customer」が「ToUser」という依存を持っていることを示します。この例題では「Customer」クラスは実装されないため、生成されたコードに影響はありません。



ポート

あるクラスの1つのインスタンスから出入りするすべてのシグナルは、ポートを通過します。アクティブクラスのインスタンスであるポートは通信するエンティティごとに、独立したポートを使用できます。ここでは、モデル内のクラスにポートを追加して通信を可能にし、ポートを使用して外部との通信を開始します。以下の操作を行います。



1. 「DomainModel」ダイアグラムに移動します。「CoffeeMachine」クラスを選択します。SHIFT キーを押しながら、ツールバーで [ポート シンボル] を選択します。ポートが境界線上に表示されます。このポートはクラスを出入りする通信を表します。ポートに「P1」という名前を付けます。



2. 「P1」ポートが選択されていることを確認します。ツールバーのボタンをクリックして実現化インターフェイスを「P1」に追加します。そのインターフェイスに「FromUser」という名前を付けます。

3. 「P1」を選択したまま、ALT キーを押しながら Enter キーを押してプロパティダイアログを開きます。[フィルタ] フィールドで「Port」が選択されていることを確認します。インターフェイス名が [実現化 (先)] フィールドに挿入されていることに注意してください。次に [要求 (先)] フィールドにインターフェイス名の「ToUser」を追加します。



4. 「DomainModel」ダイアグラムに再度移動します。「P1」ポートを選択し、要求インターフェイスのツールバー ボタンをクリックします。インターフェイス シンボル上に名前が自動的に追加されることを確認します。

注 :

インターフェイス間には関連が存在するので、要求インターフェイスの追加は必須ではありません。これは可読性の向上を意図したものです。

5. 「Controller」クラス アイコンを [モデルビュー] からデスクトップ上に開いたクラス図「DomainModel」にドラッグアンドドロップします。
6. 「Hardware」クラス アイコンを [モデルビュー] からデスクトップ上に開いたクラス図「DomainModel」にドラッグアンドドロップします。
7. 「Controller」クラスを選択し、「P2」ポートを追加します。「P1」の場合と同じように、要求インターフェイスと実現化インターフェイスを「P2」ポートに追加します。シグナルが必ず同じ方向になることに注意してください。
8. [標準] ツールバーの [保存] ボタンを押して、作業内容を保存します。

シグナル

この段階で、ダイアグラムのインターフェイスとポートは、20 ページの図 9: 「インターフェイスとポート」のようになります。以前に描かれた図についてダイアグラムが再調整されていること、およびシンボルのサブセットのみが表示されていることに注意してください。

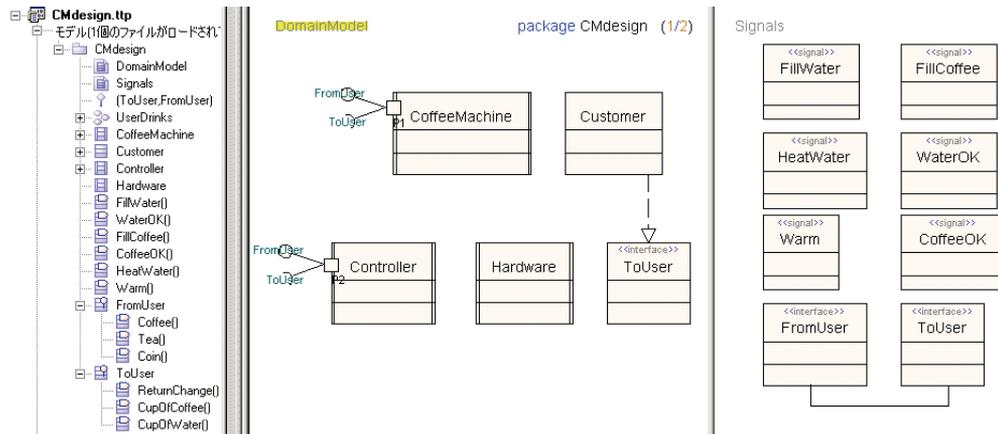


図 9: インターフェイスとポート

シーケンス図

概要

シーケンス図は、ユースケースの振る舞いを表現します。通常、シーケンス図は、システム内のアクティブクラスのインスタンスとインスタンス間のメッセージ(シグナルインスタンス)で構成されます。シーケンス図は、手動で記述するか、モデルベリファイヤ(Model Verifier)トレースのシミュレーションで生成します。

シーケンス図の作成

ここでは、「MakeCoffee」ユースケースのシーケンス図を作成します。ワークスペースウィンドウに[モデルビュー]が表示されていることを確認します。

1. [モデルビュー]内で「UserDrinks」コラボレーションを探します。
「MakeCoffee」ユースケースを右クリックします。ショートカットメニューで[新規]をポイントし、[シーケンス図]をクリックします。

注：

アクタータイプが「Customer」クラスにバインドされます。ユースケース図では、クラス名の下線は赤から緑色になります。これは、バックグラウンドのチェックが実行されていることを示す例です。モデルの変更の解析とプレゼンテーション要素の更新が常時行われて、モデルの一貫性を維持します。

2. シーケンス図が開き、そのアイコンが[モデルビュー]のコラボレーション内に表示されます。シーケンス図は相互作用(Interaction)に含まれています。相互作用にはユースケースを記述したダイアグラムを1つ以上入れることができます。
3. [モデルビュー]で「Customer」クラスを探して、シーケンス図までドラッグします。クラスがライフラインとして表示されます。
4. 「Controller」クラスをシーケンス図までドラッグします。
5. 「Hardware」クラスをシーケンス図までドラッグします。

次の手順では、シーケンス図にメッセージを作成する3種類の方法を使用します。それぞれの方法では、特にモデルからシグナル名をバインドする方法に注目してください。これら3種類の方法は、すべて[ダイアグラム要素の作成]ツールバーの[メッセージライン]ボタンから始めます。



6. 「Customer」から「Controller」までメッセージラインを描きます。「Coin」シグナルをメッセージライン上にドラッグします。シグナル名がパラメータの型を含むパラメータリストと共に表示されます。この場合は、パラメータは整数パラメータが1つです。パラメータ型の情報を整数値10で置き換えます。
7. [ダイアグラム要素の作成] ツールバーで[メッセージライン] ボタンをクリックします。「Customer」をクリックし、次に受信イベントを「Controller」に指定する位置で右クリックします。ショートカットメニューで[既存のものを参照] をポイントし、候補から[Coffee] をクリックします。

注：

[既存のものを参照] は一連のシグナルを定義する実現化インターフェースやポートをもつクラス用に使用できます。

8. 「Controler」から「Hardware」に新しいメッセージラインを描きます。メッセージライン上に「FillWater」というシグナル名を入力します。

注：

CTRLキーを押しながらSpaceキーを押すと、名前を確定するショートカットコマンドが実行されます。名前を入力を開始すると場合、このショートカットコマンドによってスコープ内での入力可能な名前がエンティティに対して確定されるか、またはこのスコープで入力可能なエンティティのリストが表示されます。

9. 引き続き 23 ページの図 10: 「MakeCoffee ユース ケースのシーケンス図」のようにメッセージシーケンスを描きます。
10. [標準] ツールバーの[保存] ボタンを押して、作業内容を保存します。

シーケンス図

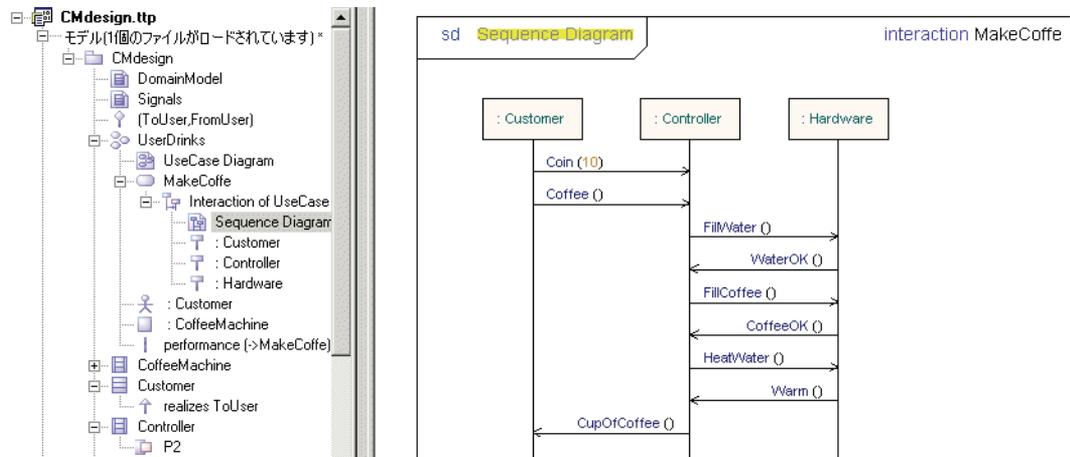


図 10: MakeCoffe ユース ケースのシーケンス図

ユース ケースとサブジェクト フレーム

1. 「UserDrinks」 コラボレーション内のユース ケース図を開きます。新規ユース ケースを作成し、「MakeTea」という名前を付けます。
2. 新規ユース ケースを作成し、「TeaMaking」という名前を付けます。
3. 「MakeTea」 から「TeaMaking」 まで依存ラインを引きます。「MakeTea」 からアクターまで関連ラインを引きます。

「MakeTea」は、主要なユース ケースの1つで、システムが1杯のお湯を作るシナリオを表現します。「TeaMaking」ユースケースは、カップを満たし、お湯を加熱することを表現する従属的なシナリオです。

4. ユース ケースの周辺にサブジェクト シンボルを作成します。[ダイアグラム要素の作成] ツールバーで板状のシンボルを選択し、ユースケース シンボルをサブジェクト シンボルで囲むようにドラッグします。フレームに名前を付けて、「CoffeeMachine」クラスに属するフレームであることを指定します。24 ページの図 11: 「サブジェクト フレーム内のユース ケース」を参照してください。
5. [標準] ツールバーの[保存] ボタンを押して、作業内容を保存します。

UseCase Diagram

collaboration UserDrinks {1/1}

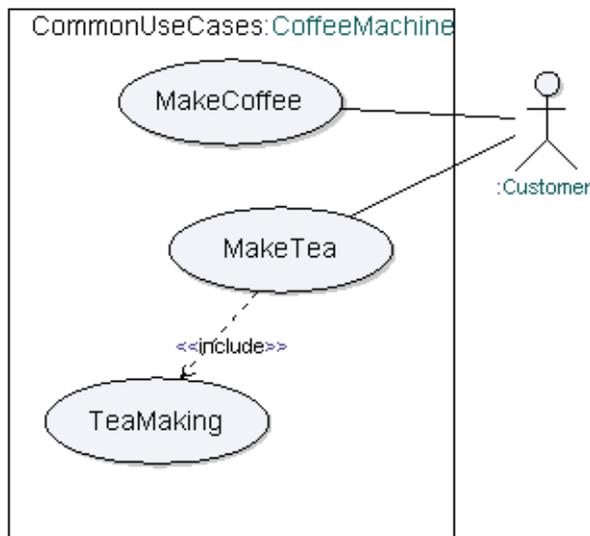


図 11: サブジェクト フレーム内のユース ケース

サブジェクト フレームは必須ではありませんが、複数のユース ケースをカプセル化してモデル内の特定のクラスに属させることができるため、共通のアクターから複数のシステムをアクセスするような複雑なユース ケース図を描くことができます。

参照付きシーケンス図

1. [モデル ビュー] 内で「UserDrinks」コラボレーションを探します。「MakeTea」ユース ケースを右クリックします。ショートカット メニューで [新規] をポイントし、[シーケンス図] をクリックします。
ダイアグラムが表示され、シーケンス図のアイコンが [モデル ビュー] のコラボレーション内に表示されます。
2. [モデル ビュー] で「Customer」クラスを見つけて、シーケンス図までドラッグします。クラスがライフラインとして表示されます。「CoffeeMachine」クラスをシーケンス図までドラッグします。
3. 「Customer」から「CoffeeMachine」までメッセージを描きます。メッセージ ラインに「Coin」というシグナル名と、パラメータの「5」を入力します。
「Customer」から「CoffeeMachine」まで新しいメッセージを描きます。メッセージ ラインに「Tea」というシグナル名を入力します。

シーケンス図



4. 参照シンボルをシグナルの下に配置します。参照シンボルに「TeaMaking」という名前を付けます。25 ページの図 12: 「参照付きシーケンス図」を参照してください。
5. [標準] ツールバーの [保存] ボタンを押して、作業内容を保存します。

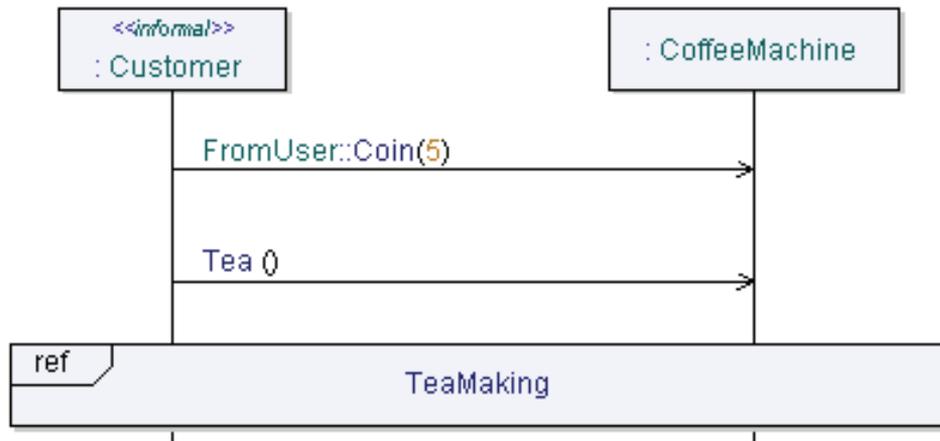


図 12: 参照付きシーケンス図

単純なシーケンスの共通セットから長く複雑なシーケンス図を作成するためのビルドブロックとして、参照シンボルを使用できます。

モデルナビゲータ

モデルが大きく複雑になるにつれ、モデルのナビゲーション作業に時間がかかるようになり、作業を習得するための時間も長くなります。モデルナビゲータは、要素に適用できるナビゲーションモードを動的に決定することによって、多くのナビゲーション作業の負担を軽減することを目的に設計されています。モデル要素のスコープやさまざまなダイアグラムでのビューを決定するような軽微なナビゲーションもあれば、特定の上位クラスから派生したクラスをすべて検索したり、特定の型の属性をすべて検索したりするような重要なナビゲーションもあります。

モデルナビゲータは、ダイアグラムを作成して、現在のモデル要素に配置する場合にも使用できます。

ここでは、モデルナビゲータを使用して、「FillWater」シグナルを起点にモデル内を移動していきます。このシグナルの定義から、スコープをたどってシグナルとの接続を維持しながら移動し、シグナルが使用されているダイアグラムを見つけます。

1. 「CMDesign」ルートパッケージを選択し、CTRL キーを押しながら F キーを押して [ダイアグラムと定義の検索] ダイアログを呼び出します。[名前] フィールドに「FillWater」と入力して、[検索] を押します。これにより、モデル内の「FillWater」シグナルが検索され、出力ウィンドウの [検索結果] タブにリストが表示されます。

注：

[ダイアグラムと定義の検索] ダイアログでは、使用されている場所は検索されず、定義されている場所が検索されます。あるエンティティが使用されている場所を見つけるには、モデルビューでエンティティを右クリックし、ショートカットメニューから [参照の一覧表示] をクリックします。

2. 出力ウィンドウの [検索結果] タブをクリックします。「FillWater」を右クリックし、ショートカットメニューから [検索] をポイントします (ダブルクリックによる検索も可能です)。
「Signals」ダイアグラムが開きます。
3. CTRL キーを押しながら、ダイアグラム内の「FillWater」シグナルのアイコン内にある名前「FillWater」をポイントしてクリックします。カーソルが手のアイコンに変わります。モデルビューでエンティティ (「FillWater」シグナル) が選択されていない場合は、ここで選択されます。

4. これは、**Ctrl** キーを押したままにしてダイアグラム内の他のシグナルをクリックすることで試すことができます。モデルビュー内の選択がどのように変化するかも確認してください。**Ctrl** キーを押したままにして複数のプレゼンテーションが含まれるもの(インターフェイスに属するシグナルなど)をポイントすると、モデルナビゲータが開きます。
5. モデルビュー内の「**FillWater**」シグナルを右クリックし、ショートカットメニューで[モデルナビゲータ]をポイントします。

これにより、出力ウィンドウ内に[ナビゲート]タブがアクティブになります。モデル内で「**FillWater**」シグナルを起点とする経路をたどります。

6. [参照]タブをクリックし、[ロール]項目の**InstanceOf**を選択します([場所]**MakeCoffee**)。これは、(この例において)シーケンス図のメッセージ上で見つかった情報の定義です。**InstanceOf**ノードは、シグナルが使用されている場所を示します。

注：

モデルナビゲータで新しい定義を選択するたびに、新しい定義が表示されます。戻る場合には、[最近]タブを使用するか、いずれかのタブを右クリックします。

7. [ショートカット]タブに移動し、[スコープ]を選択します。これは、シーケンス図を含むユースケースの相互作用ノードです。
8. 相互作用ノードには[ダイアグラム]タブがあります。このタブでは、ノードに定義したダイアグラムを確認できます。この例の場合は、ダイアグラムは1つだけであり、この項目をクリックするとデスクトップ領域に現在のダイアグラムとして表示されます。

モデルのバインドをたどることによる、システム内の簡単なナビゲートはこれで終わりです。

状態機械図

概要

状態機械図は、ステートマシンを表示し、アクティブクラスの振る舞いを表現します。ステートマシンは、1つ以上の可能性のあるステート (**possible state**) を持ち、ステートの変更はシグナルの受信によって起こります。UML では、ステートマシンの概念はデータ操作に対して拡張されています。つまり、シグナルデータとその他の変数は宣言や操作が可能です。表記法は、**遷移指向の構文**と**状態 (ステート) 指向構文**の2種類がサポートされています。どちらの構文もステートマシンの動作を記述するために使用できますが、大規模なデザインの概要を把握するには、状態指向構文の方が適しています。遷移指向の構文は詳細なデザインに適しています。

Hardware クラスの状態機械図

状態 (ステート) 指向構文

状態指向構文を使用する場合は、以下の項目に注意してください。

- ステートを接続する場合は、1番目のステートを選択します。2つのハンドルが表示されるので、黒丸で表されているハンドルを選択します。2番目のステートまでラインを引き、2番目のシンボルをクリックします。
- 接続しているラインの形状を変えるには、そのラインをクリックします。正方形のハンドルが表示されます。ハンドルをドラッグして適当な形状にします。
- ステートからそのステート自身にラインを引く場合は、黒丸で表されているハンドルを選択し、ステートから離れるようにラインをドラッグします。クリックして方向を変え、ラインをステートまでドラッグして戻し、シンボルをクリックします。
- 次の表記が遷移ラインで使用されます。
`<input>/<transition statement>;`
この場合 `<transition statement>` は `^ <signal name>` などになります。パラメータが指定されていないシグナルの後ろは、空の括弧にします。1つの遷移には、複数のステートメントを記述でき、その場合はセミコロンで区切ります。

状態機械図複合ステート

UML のステートマシンでは、複合ステートを定義できます。これは、ステートマシンにサブステートを定義する方法の 1 つです。

複合ステートは、複数のアクションが 1 つの特定のステートに常に戻る傾向があるような場合に利用できます。「Hardware」のステートマシンにはこのような傾向があります。紅茶またはコーヒーを入れる時の「Controller」と「Hardware」間のシグナル交換はよく似ていますが、ユーザーがコーヒーの要求をしたときにはシグナル交換が 1 つ多くなっています。このシグナル交換は複合ステートにすることができます。

まず状態 (ステート) 指向構文を使用して、「Hardware」クラスの振る舞いをステートマシンに記述します。以下の操作を行います。

1. 状態機械図を「Hardware」クラスに追加します。右クリックしてショートカットメニューから [新規] をポイントし、[状態機械図] をクリックします。状態機械図に「HandlingWater」という名前を付けます。この状態機械図は、「state machine implementation」に含まれていて、「state machine implementaion」はデフォルトで「initialize」というステートマシンに含まれています。
2. 29 ページの図 13: 「HandlingWater 状態機械図」のように振る舞いを実装します。
3. [標準] ツールバーの [保存] ボタンを押して、作業内容を保存します。

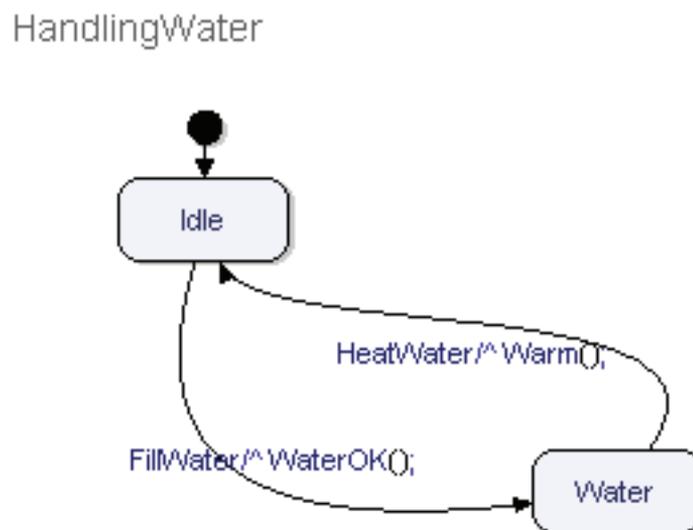


図 13: HandlingWater 状態機械図

注：

名前解決は、エンティティの色付表示で表現され、一致するエンティティが検出されると必ず表示されます。定義がモデルに組み込まれると、ナビゲーションが可能になります。定義をナビゲートするには、CTRLキーを押しながら名前をクリックします。

UMLでは大文字と小文字が区別されます。名前解決では、大文字と小文字の一致まで確認されます。

遷移指向構文

遷移指向構文を使用する場合は、以下の項目に注意してください。

- シンボルを追加するには、ツールバーで対応するクイックボタンをクリックして、デスクトップをクリックします。
- 2つのシンボルを接続するには、1番目のシンボルを選択します。2つのハンドルがシンボルの下に表示されます。正方形のハンドルを選択し、2番目のシンボルまでラインを引きます。シンボルをクリックします。
- 自動フロー：フロー内のシンボルを選択し、Shiftキーを押したままにします。次に、ツールバーのシンボルの1つをクリックします。新しいシンボルが自動的に選択されたシンボルに接続されます。

ここでは、遷移指向構文を使用して複合ステートの振る舞いを記述します。

1. 「Hardware」クラス内の「HandlingWater」状態機械図を開きます。
2. 「Water」ステートをダブルクリックします。モデルナビゲータが開きます。[ダイアグラムの作成]タブに移動し、[状態機械図]をポイントしてクリックします。新しいダイアグラムがデスクトップに表示されます。
3. 31 ページの図 14: 「複合ステート」のようにサブステート定義を描きます。ヒストリステートは「WaitFill」ステートへ戻ることを意味します。
4. [標準]ツールバーの[保存]ボタンを押して、作業内容を保存します。

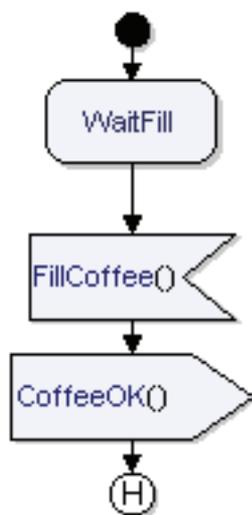


図 14: 複合ステート

Controller クラスの状態機械図

ここでは、遷移指向構文を使用して、ステートマシン内の「Controller」クラスの振る舞いを記述します。状態機械図を追加するには、以下の操作を行います。

1. ワークスペース ウィンドウの [モデル ビュー] で、「Controller」クラスを表すアイコンを選択し、「Controller」用の新しい状態機械図を作成します。
2. 状態機械図の名前を「NewOrder」に変更し、このダイアグラムが、新たにコーヒーまたは紅茶の注文を受けたときの振る舞いを表すこと示します。
3. 32 ページの図 15: 「NewOrder 状態機械図」のように振る舞いを実装します。

状態機械図

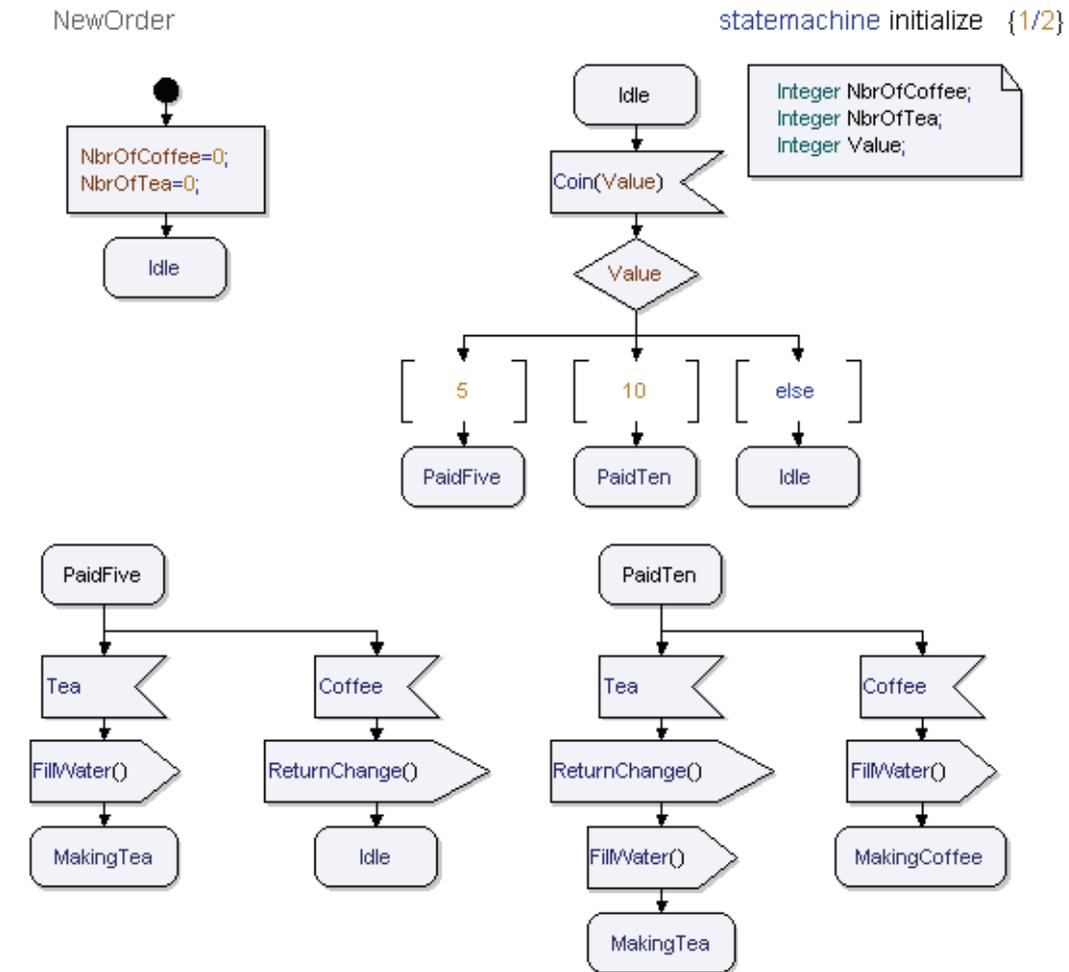
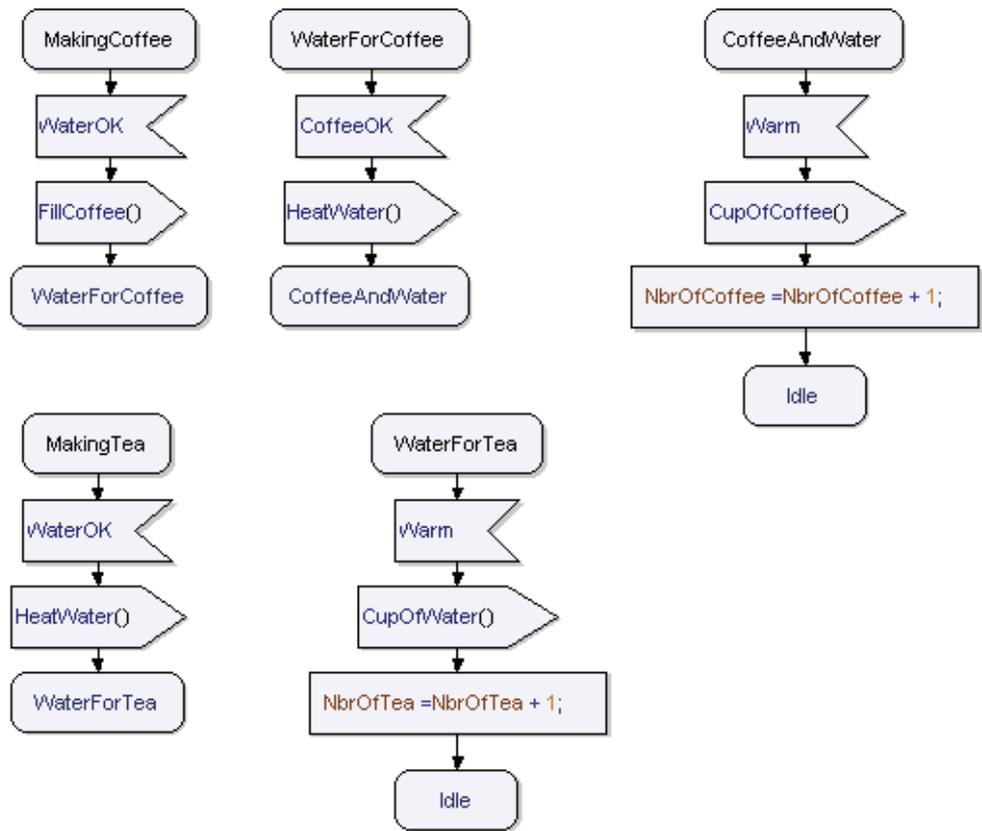


図 15: NewOrder 状態機械図

この状態機械図は、硬貨の受領、コーヒーまたは紅茶の注文の受け付け、およびハードウェア通信の開始を表現しています。何杯のコーヒーまたは紅茶を販売したかを数えるために2つの変数が使用されています。さらに、硬貨の値を保持するために別の変数が必要です。

4. 記述する領域を広げるために、残りの振る舞い用に新しい状態機械図を追加します。[モデルビュー]で「Controller」の下の「state machine implementation」アイコンを選択し、「Controller」に別の状態機械図を作成します。
5. 新しい状態機械図に「MakingBeverage」という名前を付けます。
6. 33 ページの図 16: 「MakingBeverage 状態機械図」に従って残りの振る舞いを実装します。
7. [標準] ツールバーの [保存] ボタンを押して、作業内容を保存します。

MakingBeverage



statemachine initialize {2/2}

図 16: MakingBeverage 状態機械図

これで状態機械図は完成です。

合成構造図

概要

次の手順は合成構造図の追加です。合成構造図は、アクティブクラスのインスタンスとインスタンス間の接続を表現します。このダイアグラムでは、モデルのオブジェクトをインスタンス化する方法と、一連のオブジェクトを組み立ててシステムを形成する方法を表現します。インスタンスは**パート**と呼ばれます。パートは、**ポート**を通して他のパートや環境と通信します。ポートは**インターフェイス**や**コネクタ**を通して接続します。

合成構造図の作成

ここでは、合成構造図においてクラスをインスタンス化し、顧客「**Customer**」、コントローラ「**Controller**」、およびハードウェア「**Hardware**」間の接続を記述します。合成構造図を追加するには、以下の操作を行います。

1. [モデル ビュー] で、「**CoffeeMachine**」クラスを表すアイコンを選択します。
2. 「**CoffeeMachine**」クラス上で右クリックして、[新規] を選択し、[合成構造図] をクリックします。
3. [モデル ビュー] に合成構造図が表示されます。ダイアグラムに「**Communication**」という名前を付けます。

パート

パートとは、アクティブクラスをインスタンス化したものです。パートを追加するには、以下の操作を行います。



1. 「**Communication**」ダイアグラムがデスクトップに表示されていることを確認します。ツールバーでパートを表すシンボルを選択し、ダイアグラムに配置します。
2. シンボルの内側をクリックしてテキスト領域を有効にします。インスタンスに「**Ctrl:Controller**」という名前を付けます。この場合、「**Ctrl**」はパート名で、**Controller** はインスタンス化されたクラスの名前です。
3. **Hardware** クラスのインスタンスを表すパートを追加します。パートに「**Hw:Hardware**」という名前を付けます。

注：

アクティブクラスを合成構造図にドラッグアンドドロップすることもできます。

ポート

ここでは、顧客を表すポートから始めてさらにいくつかのポートをモデルに追加し、通信を可能にします。以下の操作を行います。



1. 「Ctrl」パートを選択します。SHIFT キーを押しながら、ツールバーで [ポートシンボル] を選択します。ポートがパートの境界線上に表示されます。ポートに「P3」という名前を付けます。このポートは、「Hw」パートとの接続を表します。
2. 「Hw」パートを選択します。ポートを 1 つ追加し、「P4」という名前を付けます。このポートは、「Ctrl」パートとの接続を表します。
3. ポート「P3」とポート「P4」のプロパティダイアログを開き、要求シグナルと実現化シグナルを追加します。なお、ポートのシグナルは接続ラインの方向と一致させる必要があります。

「Ctrl」では次のシグナルを受信できます。

- 「CoffeeOK」、「WaterOK」、「Warm」

「Hw」では次のシグナルを受信できます。

- 「HeatWater」、「FillWater」、「FillCoffee」

コネクタ

コネクタはシグナルの経路であり、双方向と単方向のいずれかに設定できます。コネクタによって、合成構造図内のポートを接続します。ここでは、モデルにコネクタを追加します。以下の操作を行います。

1. ポート「P3」とポート「P4」の間にコネクタを追加します。これを行うには、いずれかのポートを選択し、ハンドルをもう一方のポートまでドラッグします。3つのテキストフィールドが表示されます。コネクタに「CtrlbiHw」という名前を付けます。
2. コネクタライン上で右クリックし、ショートカットメニューから [すべてのシグナルを表示] を選択します。コネクタごとにポートの実現化プロパティと要求プロパティに対応するシグナルを入力します。

合成構造図は [36 ページの図 17: 「合成構造図」](#) のようになるはずです。

3. [標準] ツールバーの [保存] ボタンを押して、作業内容を保存します。

合成構造図

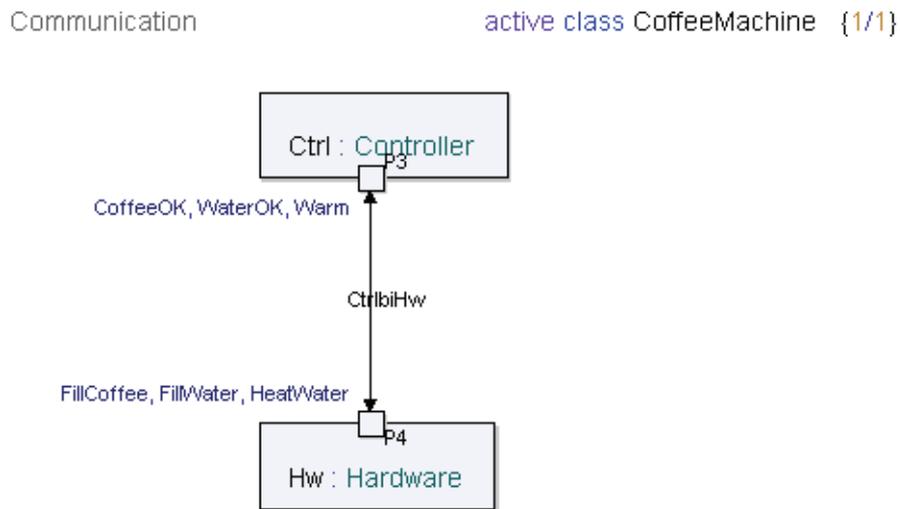


図 17: 合成構造図

これで、合成構造図は完成です。ダイアグラムの編集によって、モデルへポートとパートが追加されました。

ポートとインターフェース

コンポーネント図内のクラスにポートを追加してゆきます。

1. コンポーネント図「ControlComponent」を開きます。

ControlComponents

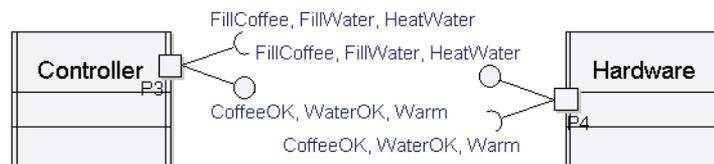


図 18: : ポートとインターフェースを含むコンポーネント図

2. ポート「P3」を「Controller」にドラッグアンドドロップします。
3. ポート「P4」を「Hardware」にドラッグアンドドロップします。

インターフェースとシグナルのリストは 36 ページの図 18: 「: ポートとインターフェースを含むコンポーネント図」のようになります。

関係

概要

次の手順ではクラス間に関係を追加します。モデル内のクラス間の関係は、クラス図で表現するのが最も適しています。

関連

関連とは、オブジェクト間の関係を表現したものです。各関連には2つのロールがあり、ロールは関連の方向によって表現されます。モデル内の「ToUser」インターフェイスと「FromUser」インターフェイス間の関連は、作成済みです。

複合

「Controller」クラスと「Hardware」クラスは常に「CoffeeMachine」クラスと共に存続します。これは、これらのクラスと「CoffeeMachine」クラスの関係は**複合**であることを意味します。

1. 「DomainModel」ダイアグラムで「CoffeeMachine」クラスを選択します。3つのハンドルがシンボルの下に表示されます。
2. 正方形 ([関連 / 集約 / 複合ラインの作成]) で表されたハンドルを選択し、ラインを「Controller」クラスまで引いてクラスシンボルを右クリックします。[既存のものを参照]を選択します。ドロップダウンボックスにCtrlパートが表示されるはずですが。

注：

ラインを「CoffeeMachine」との接続に近い位置で右クリックすると、ショートカットメニューで複合を関連または集約に変更できます。

3. 「CoffeeMachine」クラスと「Hardware」クラスの間にも同じ方法で**複合**を追加します。
4. [標準] ツールバーの [保存] ボタンを押して、作業内容を保存します。

関係

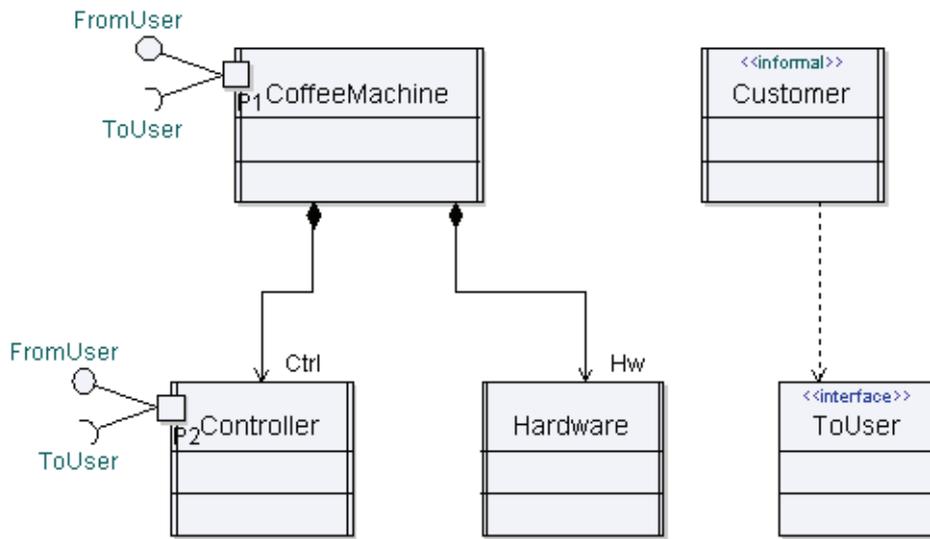


図 19: ドメインモデルでの複合

クラス図は 38 ページの図 19: 「ドメインモデルでの複合」のようになるはずです。

チェック

概要

Tauには、できるだけ迅速にエラーを見つけることができるように、組み込みのチェック機能があります。サポートされるチェックは、**オートチェック**と**チェック**の2種類です。

オートチェック

オートチェックは常時動作しており、作業中のスコープをチェックしています。スコープはデスクトップで開かれているダイアグラムで決まります。見つかったエラーはすべて出力ウィンドウに表示されます。このチェックは常に動作しているため、エラーが修正されるとエラーメッセージはすぐに消えます。

チェック

チェックは、[すべてをチェック]または[選択部分のチェック]のいずれかのクイックボタンをクリックすると開始されます。[すべてをチェック]は、モデル全体をチェックします。[選択部分のチェック]は、[モデルビュー]内で選択したエンティティとそのサブツリーだけをチェックします。見つかったエラーはすべて出力ウィンドウに表示されます。エラーは、新しいチェックが開始されるまで、リストに残りません。

モデルをチェックするには、以下の操作を行います。



1. [すべてをチェック]クイックボタンをクリックします。
2. 出力ウィンドウにエラーメッセージが表示された場合は、内容を確認して訂正します。
 - いくつかのエラーメッセージの種類では、ダブルクリックするとデスクトップに問題の領域を表示します。
 - エラーメッセージを右クリックし、表示されたショートカットメニューで[検索]をポイントする方法もあります。

修正を確認するには、[すべてをチェック]クイックボタンを再度クリックします。

ビルドアーティファクト

概要

ビルドアーティファクトは、モデルまたはモデルのサブセットを表現します。ビルドアーティファクトは UML モデルの要素であり、ビルドプロセス専用のプロパティを持ちます。ビルドアーティファクトを使用して作業すると、UML モデルから実行可能なプログラムを生成するために1つのソースコードカーネルを構成する多くのファイルが使用されます。カーネルは、アプリケーションやデバッグ用にさまざまなものが存在します。

モデルベリファイヤ (Model Verifier) のセットアップ

ビルドアーティファクトを使用するには、モデルベリファイヤ用のコードジェネレータアドインの適切なセットアップが必要です。通常、このセットアップはプロジェクトファイルの作成時に設定されますが、[ツール]メニューから[カスタマイズ]を選択してセットアップ内容を変更できます。[カスタマイズ]ダイアログを開いたら、[アドイン]タブでモジュール [Model Verifier] がアクティブであることを確認します。

ビルドアーティファクトの作成

ここでは、モデルを表すビルドアーティファクトを作成します。ビルドアーティファクトは後で実行されます。

1. 「CMdesign」パッケージを右クリックし、ショートカットメニューから [Model Verifier] をポイントし、[新しいアーティファクト] をクリックします。ビルドアーティファクトのアイコンが [モデルビュー] に表示されます。ビルドアーティファクトに「artifact_cm」という名前を付けます。
2. ビルドアーティファクトのルートを設定します。これによりビルドアーティファクトに取り込むモデル内の部分を決定できます。ビルドアーティファクトは、ルートとして設定したエンティティとそのサブツリーを表します。[モデルビュー] の artifact_cm を右クリックします。ショートカットメニューから [ビルドルートの選択] を選択します。ルートを「CoffeeMachine」クラスに設定します。[OK] をクリックします。

3. [モデルビュー]でビルドアーティファクトアイコンを右クリックし、ショートカットメニューから[ビルド(Model Verifier)]をポイントして[ビルド]をクリックします。この操作を初めて行う場合は、ビルドタイプの設定を求めるプロンプトが表示されます。リストから[Model Verifier]を選択します。

重要：

モデルベリファイヤ(Model Verifier)を使用してビルドを行うには、サポートされているCコンパイラをインストールしておく必要があります。GNU Cコンパイラを使用している場合は、アーティファクトを調整する必要があります。これには、「artifact_cm」を右クリックして[プロパティ]をポイントし、フィルタで表示されるメニューから[Model Verifier]を選択します。そして、[Target Kind]の値を「Win32-gcc」に設定します。

4. ビルドの結果が出力ウィンドウに表示されます。ビルドが完了するとステータスメッセージが表示され、最後にエラーメッセージと警告メッセージの件数が表示されます。

モデルベリファイヤ (Model Verifier)

概要

モデルを検証することで、デザインの動作をテストできます。受信されたシグナルに対する反応を評価できます。利用可能なトレースには、テキストトレース、グラフィカルUMLトレース、およびシーケンス図トレースの3種があります。

コーヒー販売機の検証

ここでは、コーヒー販売機の振る舞いをテストします。モデルベリファイヤのセッションを開始するには、以下の操作を行います。

1. [モデルビュー]で、**artifact_cm** ビルドアーティファクトを右クリックします。ショートカットメニューから [ビルド (Model Verifier)] を選択し、[実行] をポイントします。新しい [インスタンス] タブがワークスペース ウィンドウで開きます。
2. 「CMdesign.ttp」を選択しながらテンキーパッドの乗算キー (*) を押して、[インスタンス] タブの情報を展開します。Ctrl、Hw、および環境 (env) に属する多数のパラメータが表示されます。

メッセージマトリックスの作成

デバッグを単純化するために、シグナルのリスト (メッセージマトリックス) を作成できます。これを行うには、[表示] メニューから [モデルベリファイヤ ウィンドウ] を開き、[メッセージ] を選択します。

すべてのユースケースをテストするには、以下に示す、環境からの4種類のシグナルが必要です。

- 値「10」を持つ「Coin」シグナル
このシグナルは、値が「10」である硬貨の投入を表します。
- 値「5」を持つ「Coin」シグナル
このシグナルは、値が「5」である硬貨の投入を表します。
- 「Coffee」シグナル
このシグナルは「Coffee」ボタンを押すことを表します。
- 「Tea」シグナル
このシグナルは「Tea」ボタンを押すことを表します。

モデルベリファイヤ (Model Verifier)

値「10」を持つ「Coin」シグナルをメッセージマトリックスに追加するには、以下の操作を行います。

1. メッセージマトリックスの左上にある「...」を含むフィールドをダブルクリックします。1行目がアクティブになります。
2. [送信側]列で「unspecified」をクリックします。ドロップダウンボックスが表示されます。「env[1]」をクリックします。
3. [シグナル]列で「unspecified」をクリックします。ドロップダウンボックスが表示されます。「::CMdesign::Coin」をクリックします。
4. [チャンネル]列は現在の値の「unspecified」のままにします。
5. [受信側]列で「unspecified」をクリックします。ドロップダウンボックスが表示されます。「CoffeeMachine.Ctrl[1]」をクリックします。
6. [パラメータ]列で「unspecified」をクリックします。表示されたフィールドに、「10」の値を入力して、キーボードのEnterキーを押します。シグナルがメッセージマトリックスに組み込まれます。
7. 43ページの図20:「メッセージマトリックス」に従って、その他のシグナルを追加します。

	送信側	シグナル	チャンネル	受信側	パラメータ
1	env[1]	::CMdesign::Coin	unspecified	CoffeeMachine.Ctrl[1]	10
2	env[1]	::CMdesign::Coin	unspecified	CoffeeMachine.Ctrl[1]	5
3	env[1]	::CMdesign::Coffee	unspecified	CoffeeMachine.Ctrl[1]	unspecified
4	env[1]	::CMdesign::Tea	unspecified	CoffeeMachine.Ctrl[1]	unspecified
...					

図 20: メッセージマトリックス

注:

メッセージマトリックス内の特定のラインで、シグナルを有効にして送信するには、「...」を含む1番目のフィールドをダブルクリックします。

パラメータのウォッチ

「env」、「Ctrl」、および「Hw」の現在のステータが表示されます。スタートアップ時には、すべてのステータスが「start」に設定され、スタートシンボルが参照されます。これは、ステータスマシンが最初のステータスに到達していないことを意味します。デバッグ時にその他のパラメータのどれをウォッチすべきかを選択することもできます。

1. インスタンスビューで、以下のパラメータを右クリックします。表示されたショートカットメニューで [ウォッチ] をポイントします。

Ctrl:

- NbrOfCoffee
- NbrOfTea
- Queue

Hw:

- Queue

env:

- Queue

トレース

デバッグのトレースには、さまざまな方法を使用できます。デフォルトでは、テキストトレースが起動します。ユーザーインターフェイスの右下隅にテキストトレースが表示されます。デフォルトでは、実行トラッキングも行われます。つまり、現在実行されている UML シンボルがデスクトップの状態機械図に強調表示されます。オプションで、シーケンス図トレースも使用できます。デバッグを開始するには、以下の操作を行います。



1. 最初のユースケースでは、シーケンス図トレースのみを使用します。[次のステートメントを表示] クイックボタンをクリックして、追跡の実行をオフにします。



2. [シーケンス図でトレース] クイックボタンをクリックしてトレースを開始します。デスクトップにシーケンス図が表示されます。シーケンス図は環境、Ctrl、および Hw を示すインスタンスを含みます。



3. [次の遷移] クイックボタンを2度クリックします。Ctrl と Hw の両方が Idle ステータスになります。

ユースケースの検証

ここでは、コーヒーマシンに硬貨を投入します。デバッグする最初のユースケースは、値「10」の硬貨が投入され、ユーザーがコーヒーを選択するケースです。

「CupOfCoffee」(1杯のコーヒー)が返されるはずですが、

1. メッセージマトリックスで、値「10」の「Coin」シグナルを表す1行目の左端のフィールドをダブルクリックします。または右クリックして[送信]をポイントします。シーケンス図トレースに、「Coin」シグナルの送信が表示されます。「Coin」シグナルは、「Ctrl」の入力待ち行列 (Queue) に配置されます。これは、「Ctrl」の「Queue」パラメータで確認できます。
2. [次の遷移]クイックボタンをクリックします。「Coin」シグナルが「Ctrl」によって消費されます。「Ctrl」の待ち行列は、現在空になっているはずですが、「Ctrl」のステートが「PaidTen」に変更されていることに注意してください。
3. メッセージマトリックスの3番目のシグナルを送信して、コーヒーを選択します。シーケンス図にシグナルの送信が表示されます。
4. [次の遷移]クイックボタンをクリックします。「Coffee」シグナルが消費され、「FillWater」シグナルが「Ctrl」から「Hw」に送信されます。
5. [次の遷移]クイックボタンを必要な回数クリックして、ユースケースが完了したことを示す「CupOfCoffee」シグナルを表示します。「Ctrl」の「NbrOfCoffee」パラメータが1に増えたことに注意してください。

「Ctrl」と「Hw」の両方は、タスクを完了し、「Idle」ステートに戻り、新しい注文を処理できるようになります。



6. シーケンス図のトレースに慣れたので、[次のステートメントを表示]クイックボタンをクリックして実行トラッキングを有効にします。実行されたUMLシンボルが状態機械図で強調表示されます。



7. 後でこのテストを再実行できるように、テストを保存します。[リプレイモード]ボタンをクリックします。これにより、モデルベリファイヤ (Model Verifier) をリプレイモードに設定し、セッションの最初からステップ実行でテストできます。



8. [再実行]ボタンを押します。次に、[次の遷移]を繰り返し押し、同じテストが再度実行されることを確認します。

モデルベリファイヤ (Model Verifier)

テストが完了すると、[モデルベリファイヤ]ツールバーのほとんどのボタンが灰色表示になります。使用したトレースが変更されたことでテストの完了を確認します。リプレイテストでは、他のシグナルをシステムに送信するなどして、遷移の間に割り込むことができます。リプレイモードを無効にしない限り、割り込みの位置から再度実行が試みられます。

9. ここで、このテストのシナリオを保存します。保存するには、[ファイル]メニューで[シナリオの保存]を選択します。このテストを「Coffee10.ttdscn」として保存します。システムに変更を加えた場合には、次にモデルベリファイヤを実行するときはこのシナリオを再実行できます。

注：

Ctrl+Tabキーを押すと、デスクトップに表示されるダイアグラムを変更できます。

10. 同じ方法で、2ページの「コーヒー販売機の振る舞い」で説明した残りのユースケースを実行します。各シナリオの前でシーケンス図トレースを開始し、各シナリオの最後でシーケンス図トレースを停止します。これにより、シナリオごとに新しいユースケースが作成されます。トレースから生成されたユースケースは、[DebugTrace]と呼ばれる独立したパッケージに格納されます。

注：

ステートマシンの開始状態は、モデルベリファイヤセッションを完全に再スタートした時のみトレース内に表示されます。

ヒント：

「DebugTrace」パッケージは、デフォルトでは現在のモデルファイルに保存されません。ダイアグラムを保存しておきたい場合は、新しいファイルに保存するか、現在の.u2ファイルに移動する必要があります。プロジェクトを右クリックして表示されるメニューから [モデルビューフィルタ]、[ファイルの表示] と選択してファイルセクションを表示し、保存したいパッケージなどをファイルセクションの.u2ファイルにドラッグアンドドロップします。



11. すべてのシナリオが完了したら、モデルベリファイヤ (Model Verifier) を停止します。[モデルベリファイヤの停止]クイックボタンを押します。

参照シーケンス図

ここでは、モデルベリファイヤセッションにシーケンス図トレースを使用して、「TeaMaking」参照シーケンス図を作成します。

1. トレース用のシーケンス図から飲み物の選択が「Tea」になっているシーケンス図を探します。メッセージライン上(またはダイアグラムの表題部分)でダブルクリックします。これにより、[モデルビュー]の[DebugTrace]パッケージが検出されて開かれ、さらにユースケースが検索されます
2. このユースケースの相互作用ノードを「TeaMaking」ユースケースまでドラッグします。
3. 47 ページの図 21: 「TeaMaking シーケンス図」のようにシーケンス図を編集します。該当するクラスを「CMdesign」パッケージの[モデルビュー]からライフラインのヘディングまでドラッグします。「Coin」シグナルと「Tea」シグナルを削除します。これらのシグナルは「MakeTea」シナリオで指定されています(25 ページの図 12: 「参照付きシーケンス図」参照)。すべてのステートとシグナルがモデルにバインドされているかチェックします。
4. [標準] ツールバーの[保存] ボタンを押して、作業内容を保存します。

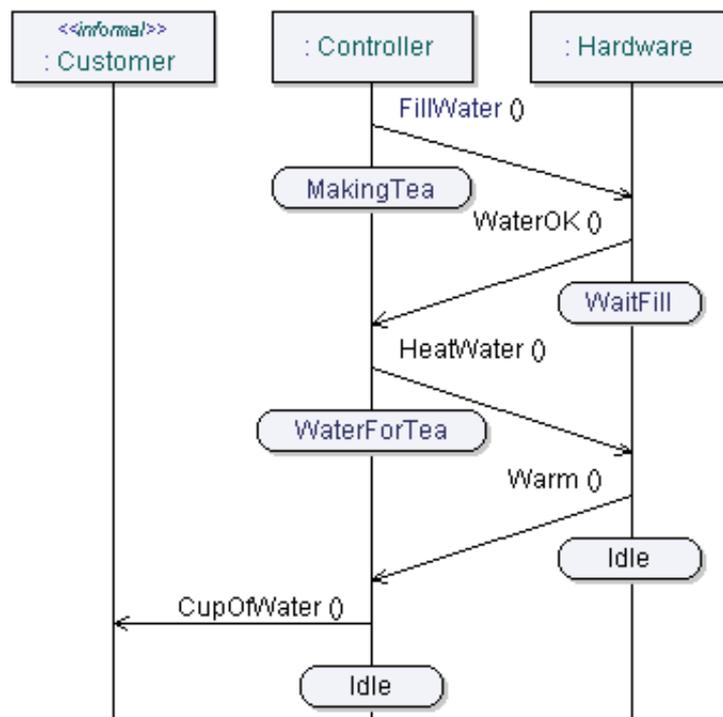


図 21: TeaMaking シーケンス図

反復と追加

目的

この項では、UML の他の機能を紹介しながら、モデルへの追加を行います。すでにこのチュートリアルで説明した内容の繰り返しも一部あります。この説明によって、**Tau** でサポートされている反復作業プロセスの概略を理解できます。

この項では、一部の操作説明は短く、以前の説明ほど詳しくありません。以前に説明した作業と類似する部分は特に省略されています。

タイマー

コーヒー販売機「CoffeeMachine」にタイマーを導入します。タイマーは 10 時間単位でタイムアウトにする必要があります。また、お湯の加熱処理と連動するように設定します。

1. [モデル ビュー] で「Hardware」クラスに移動します。「Heater」というタイマーをクラスに追加します。「Hardware」クラスを右クリックし、ショートカットメニューから [新規] をポイントして、[タイマー] をクリックします。
2. 「Hardware」の状態機械図に移動します。「Warm」シグナル送信のためのトリガーとなるように、タイマーのシグナルを受信する必要があります。これには、「Water」ステートから「Idle」ステートへの遷移で「HeatWater」シグナルを「Heater」シグナルに置き換えます。

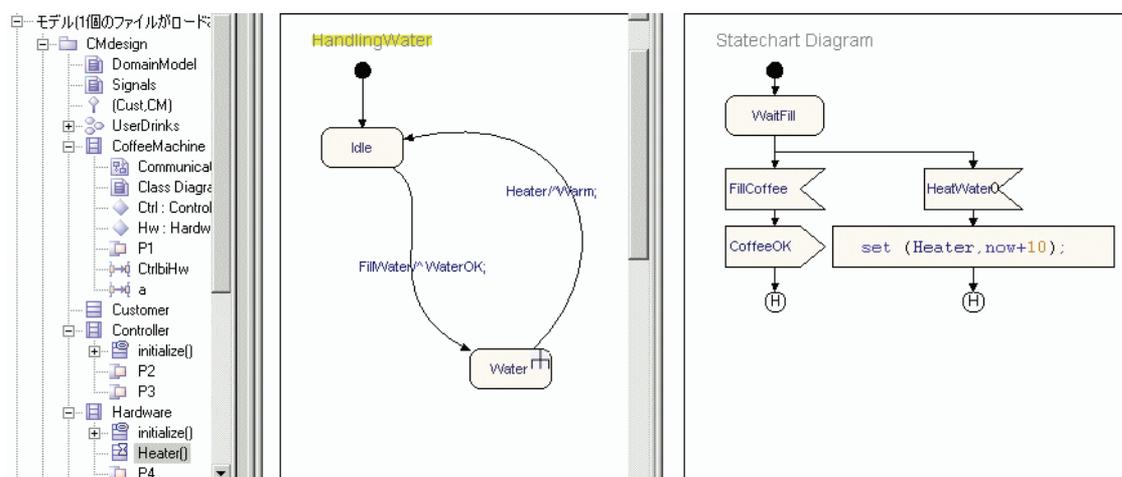


図 22: タイマーの宣言と使用

3. 「Water」複合ステートの状態機械図に移動します。「FillCoffee」の受信と平行処理される「HeatWater」シグナルを追加します。この遷移には、「Heater」タイマーの「set」ステートメントを、”set(Heater(),now+10;”と記述する必要があります。
48 ページの図 22: 「タイマーの宣言と使用」を参照してください。
4. [標準] ツールバーの [保存] ボタンを押して、作業内容を保存します。
5. モデルベリファイヤ (Model Verifier) をビルドおよび実行して、タイマーが期待どおりに動作するかテストします。シーケンス図でシナリオをトレースします。タイマーはシーケンス図に表示されます。このテストの実行には保存していたシナリオ「Coffee10.ttdscn」を使用します。テストがタイマーイベントに到達すると、システムが変更されているため続行できなくなります。テストのリプレイを停止し、手動でシナリオの残り部分を実行します。
6. すべてのシナリオが完了したら、モデルベリファイヤ (Model Verifier) を停止します。[モデルベリファイヤの停止] クイックボタンを押します。



構造データ

コーヒー販売機に、一連の構造データを表すクラスを導入します。このデータモデルの内容は、コーヒーに入れるミルクと砂糖です。シグナル送受信と内部の振る舞いの一部に影響するため、このクラスを追加するには複数の手順を実行する必要があります。さらに、モデルベリファイヤを実行する場合にシナリオがより複雑になります。

1. 「Signals」クラス図を開きます。ダイアグラム内にクラスを追加し、そのクラスに「Additives」という名前を付けます。
2. Boolean 型の「Milk」と Integer 型の「Sugar」という 2 つの属性をクラスに追加します。属性をパブリック (public) で宣言します。

ヒント :

可視性を「public」にするには、インターフェース(またはクラス)シンボルの属性名の前に「+」マークを入力してください。

注 :

次に、Additives型のパラメータを「Coffee」シグナルと「CupOfCoffee」シグナルに追加します。インターフェイス内でシグナルのパラメータを編集できるようにするには、まずインターフェイスステレオタイプシンボルからこのシグナルが可視になるようにする必要があります。

反復と追加

3. [モデル ビュー]で「Coffee」シグナルを選択し、クラス図内の「FromUser」インターフェイス シンボルまでドラッグします。パラメータを入力し、パートとして宣言します。
4. [モデル ビュー]で「CupOfCoffee」シグナルを選択し、クラス図内の「ToUser」インターフェイス シンボルまでドラッグします。パラメータを入力し、パートとして宣言します。50 ページの図 23: 「Additives クラス」を参照してください。
5. 「Controller」クラスの「NewOrder」状態機械図を開きます。Additives 型の「Add」という変数を宣言します。その変数をパートとして宣言します。51 ページの図 24: 「Additives を追加した Controller のステートマシン」を参照してください。

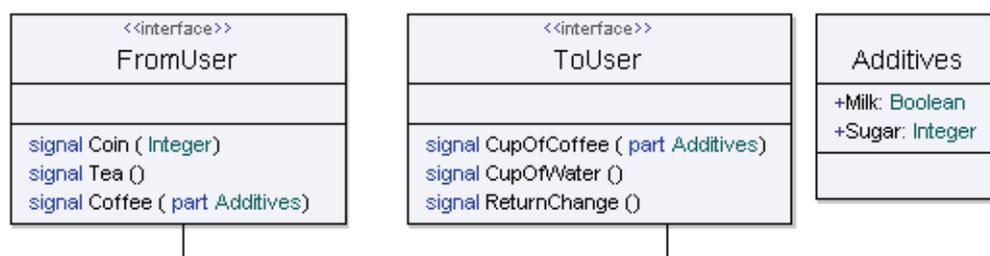


図 23: Additives クラス

注：

パラメータまたは変数をパートとして宣言すると、コード生成時の処理に影響するため、モデルベリファイヤ(Model Verifier)インターフェイスを通る値の送信方法に影響します。デフォルトでは、クラスは構造データモデルへのポインタとして表現され、パートの変数は値による参照が可能になります。

6. Additives 型の「NoAdd」という変数を宣言します。この変数はパートとして宣言しないでください。
7. 「Add」変数を使用して、「Coffee」シグナルを受信したときに必ず「Milk」（ミルク）と「Sugar」（砂糖）の要求を受信します。51 ページの図 24: 「Additives を追加した Controller のステートマシン」では、生成されたコードでのデータの処理方法をコードによって表現しています。
8. 「CupOfCoffee」を環境へ送信する際に、「Add」変数の値をパラメータとして使用します。
9. [標準] ツールバーの [保存] ボタンを押して、作業内容を保存します。

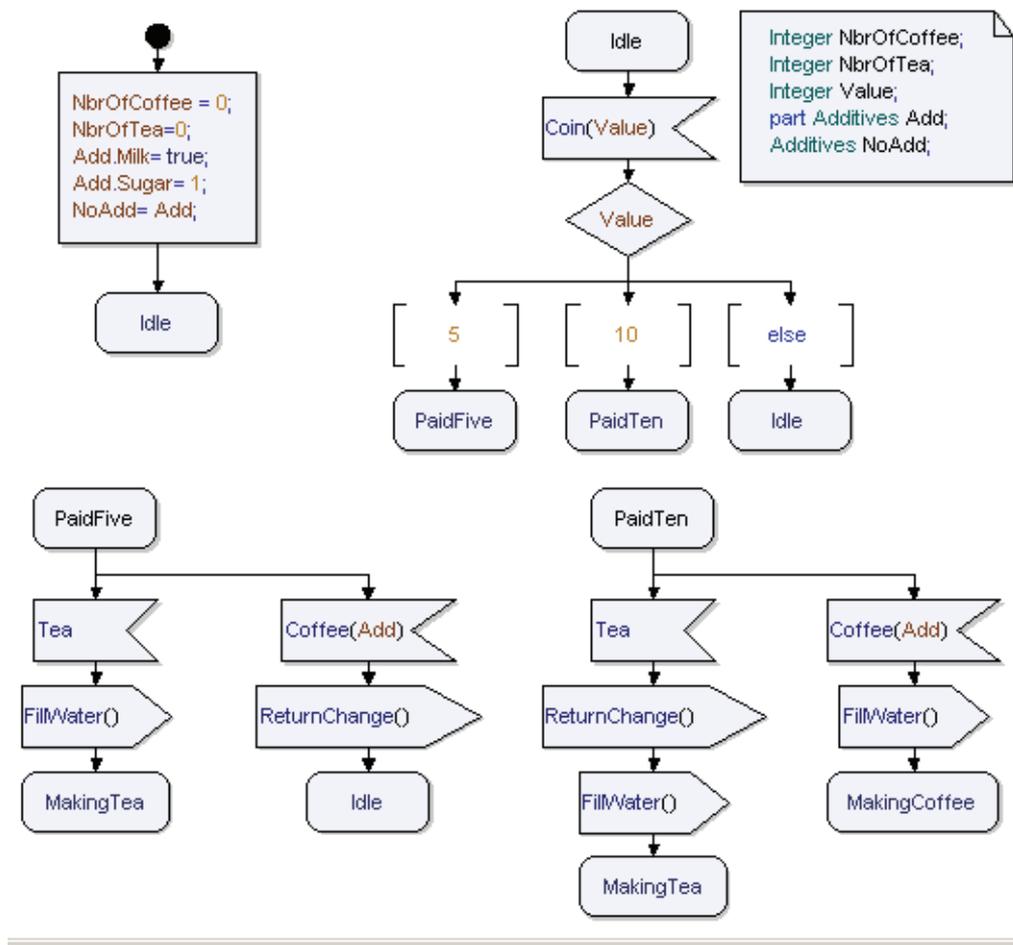


図 24: Additives を追加した Controller のステートマシン

10. モデル ベリファイヤ (Model Verifier) をビルドおよび実行して、新しいシグナルパラメータが期待どおり動作するかテストします。モデル ベリファイヤのウォッチ属性を利用します。たとえば、「Add」と「NoAdd」がどのような値になっているかを確認してください。
11. また、シーケンス図を更新して新しいパラメータを有効にする必要があります。52 ページの図 25: 「Additives 型のパラメータを使用したシーケンス図」を参照してください。パラメータの前の型修飾子を確認してください。[メッセージ] ウィンドウの [パラメータ] フィールドで、すべてのクラスに型修飾子を指定する必要があります (Additives (.Milk=true, Sugar=3.))。ウォッチ ウィンドウからショートカットメニューの [ディープコピー] を使用してコピーし、[パラメータ] フィールドで Ctrl キーを押しながら V キーを押して貼り付け、パラメータ付のクラスをコピー アンドペーストすることも可能です。

反復と追加

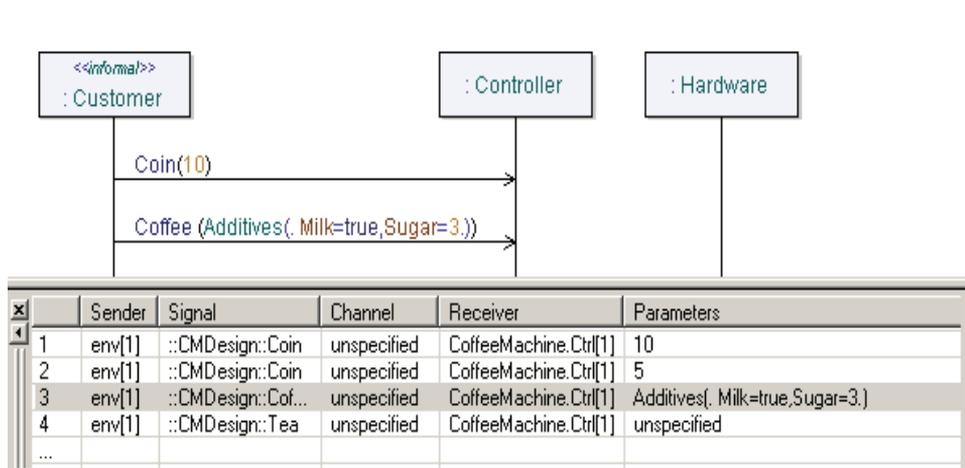


図 25: Additives 型のパラメータを使用したシーケンス図

まとめ

モデル

UML におけるダイアグラムとは作成したモデルを視覚的に表現したものです。そして、定義の作成とその定義と要素との結び付けを常に行ってユーザを支援します。支援ツールとして用意されているモデルナビゲータを使用すると、選択したモデルの結び付き (モデルバインディング) をたどって確認できます。

エディタ

このチュートリアルを通して、**Tau** のエディタにおける作業に親しんでいただけたと思います。作成した例は、機能は単純ですが UML で使用する可能性のあるシンボルの大部分を含んでいます。このチュートリアルでは、同じような構造を記述するために別の方法を示す場合もありました。特定の状況でどの方法を使用するかは問題の性質によって異なり、個人的な好みに左右されることもあります。重要なのは、土台となるモデルが作業の効率と正確性を改善するために役立つことを理解することです。

テスト

このチュートリアルモデルで説明したテスト作業では、通常時のシナリオが正しく動作することの確認だけに限定されています。つまりシステムが期待どおり動作することの確認だけです。

ワークフロー

このチュートリアルのワークフローは、より大きな規模のソフトウェアプロジェクトで実施され得るワークフローに倣うことを意図しています。このチュートリアルでは、手法を強調するのではなく、ツールのデモンストレーションを中心に説明しています。ただし、ここで説明されている作業の順序は、より大きな規模のソフトウェアプロジェクトの設計フェーズにも適合できます。今回の例題で忘れてはならないことは、分析はすでに完了されているということ、および新規にすべてのモデルを編集する場合でも設計はすべて前もって決定されているということです。ワークフローの手法の詳細については、**Tau** のオンラインヘルプの「ワークフローの説明」の章を参照してください。

今後の学習

チュートリアルはこれで終了です。UML と Tau を利用して、実際の作業を行う準備が整いました。モデルを使用した作業の詳細については、オンライン ヘルプの「モデルの操作」の章を参照することをお勧めします。