Decision Model and Notation Feature Guide

by Sparx Systems Japan

Enterprise Architect 日本語版

DMN (Decision Model and Notation) 機能ガイド

目次

1 概要 3
2 DMN の概要 3
3 Enterprise Architect での DMN モデリング 6
3.1 ダイアグラムの作成 6
3.2 デシジョンの作成 7
3.3 ビジネス知識モデルの定義 10
3.4 要素間を接続する 13
4 シミュレーションの実施 15
4.1 単一のデシジョンに対するシミュレーション 15
4.2 複数のデシジョンに対するシミュレーション 17
4.2.1 アイテム定義の追加 18
4.2.2 入力データとの結びつけ 19
4.2.3 入力データとデシジョンの結びつけ 20
4.2.4 シミュレーション要素の作成と実行 21
5 ソースコードの生成 24
6 BPMN との連携 (BPSim) 26

Copyright (c) 2019-2024 Sparx Systems Japan

1　概要

Enterprise Architectでは、DMN (Decision Model and Notation) のモデリングやシミュレーションの機能を搭載しています。DMNとは、デシジョン(意思決定)を可視化するための記法で、主に業務上の判断条件やルールを明示し可視化するために利用できます。表記として記述しただけではその内容が妥当かどうか判断することは容易ではありませんので、シミュレーションを実行し内容が適切かどうかを確認できます。

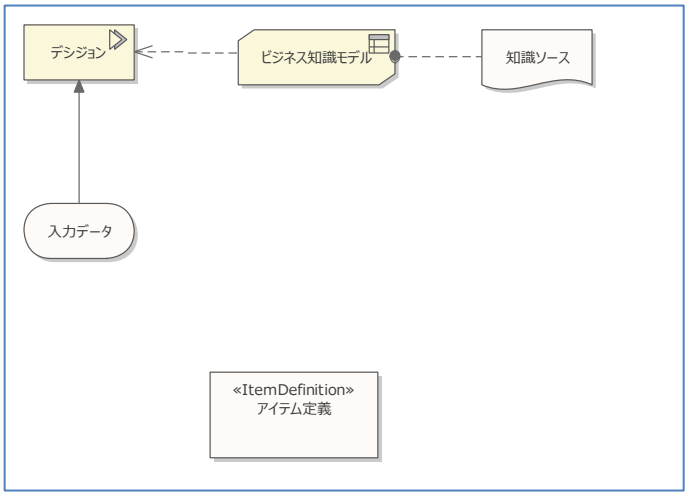
また、記述した内容を最終的に業務システムやアプリケーションとして実装する場合には、記述したルールをそのままソースコードとして生成し、実装を効率化できます。

本ドキュメントでは、このDMNに関連する機能について説明します。なお、このドキュメントで利用しているEnterprise Architectは、バージョン17.0です。DMNに関連する機能を利用するためには、ユニファイド版あるいはアルティメット版が必要です。
(DMNの作図のみであればコーポレート版も利用可能ですが、シミュレーションでの内容確認やソースコード生成などが利用できませんので、効果は限定的です。)

2　DMNの概要

DMN(Decision Model and Notation)は、UML などと同様に OMG (Object Management Group) によって定義された記法です。デシジョン(意思決定)を可視化するための利用できます。

DMN で定義された要素を利用したモデルの一例は次の通りです。なお、以下のそれぞれの用語は弊社独自の訳であり、一般的な翻訳とは異なるかもしれません。



「デシジョン」は、何らかの意思決定を示します。その意思決定をするためには、何らかのルールや根拠などがあるかと思います。例えば、「過去にキャンペーンに申し込んでいる人は、今回はキャンペーンの対象外にする」の

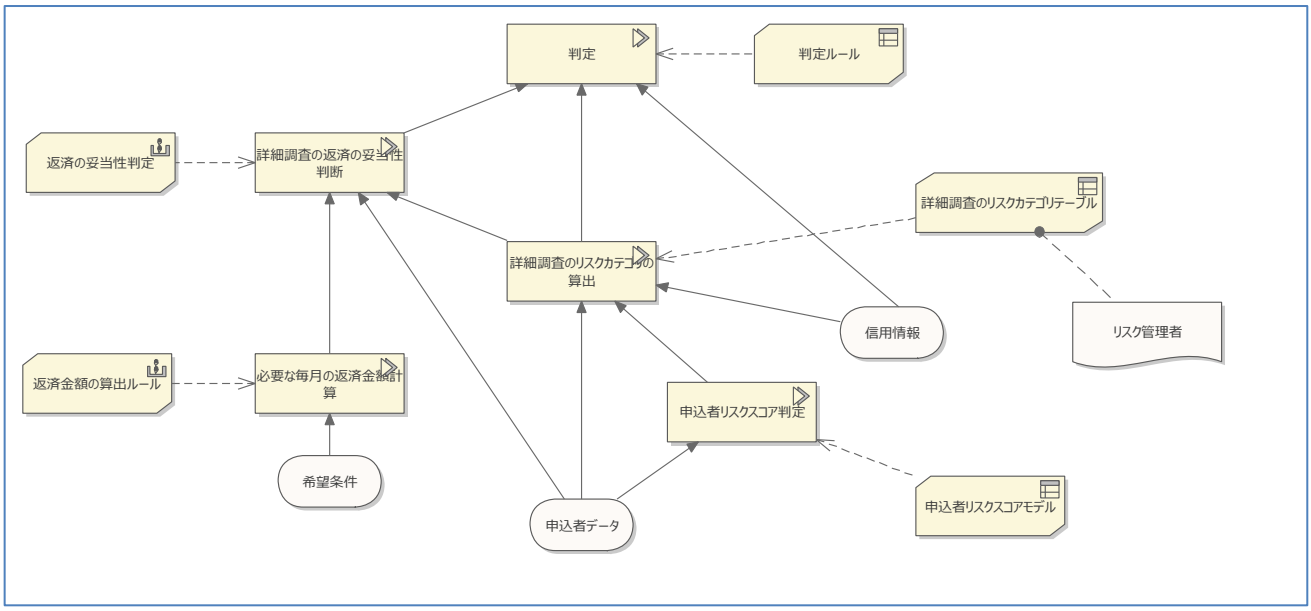
Copyright (c) 2019-2024 Sparx Systems Japan

ようなルールです。こうしたルールや根拠などが「ビジネス知識モデル」です。デシジョンとビジネス知識モデルは、N 対 N の関係になります。つまり、1 つのビジネス知識モデルが複数のデシジョンで参照・利用されたり、1 つのデシジョンで、複数のビジネス知識モデルで定義された内容を元に意思決定したりすることがあります。一般的には、ビジネス知識モデルは複数のデシジョンで共通に使われる「共通ライブラリ」のような位置づけに近いです。
(再利用されない局所的なルールであれば、デシジョン自体にルールを定義できます。)

「知識ソース」は、デシジョンやビジネス知識モデルと結びつく実際の情報源です。例えば、ビジネス知識モデルの内容が根拠としている法律文書や定義した組織、あるいはビジネス知識モデルの内容を把握している担当者などが該当します。

「入力データ」は、デシジョン要素に対して実際に適用する情報です。単一の値とは限らず、「年齢」「職業」「現在の借入金額」「収入」などの複数の情報の場合もあります。この入力データの「型」となるような情報が「アイテム定義」です。このアイテム定義は、下の画像のようなイメージで階層化されることや値の選択肢をもつことがあります。

これらの要素は複数利用され、階層的になることが多いです。下の図はその一例です。



Copyright (c) 2019-2024 Sparx Systems Japan

この図では、最上位のデシジョン「判定」に至るまでに、多くの他のデシジョンでの意思決定結果を反映しています。各デシジョンでは、その意思決定の根拠となる情報がビジネス知識モデルとして結びついています。

デシジョンやビジネス知識モデルでは、その意思決定のルールをいくつかの形式で定義できます。一般的に思い浮かぶのは、次のような「デシジョンテーブル」形式ではないでしょうか。

DMN 表現

詳細調査のリスクカテゴリテーブル　シミュレーションのためのパラメータ値

(既存顧客, 申込者リスクスコア, 信用スコア)

U	既存顧客	申込者リスクスコア	信用スコア	詳細調査のリスクカテゴリ
	true,false		-	低低, 低, 中, 高, 拒否
1	false	<120	<590	高
2	false	<120	[590..610]	中
3	false	<120	>610	低
4	false	[120..130]	<600	高
5	false	[120..130]	[600..625]	中
6	false	[120..130]	>625	低
7	false	>130	-	低低

このような表形式以外にも、数式や条件式で定義する場合もあります。以下はその例です。

DMN 表現

返済金額の算出ルール　シミュレーションのためのパラメータ値

(ローン種類, 利率, 期間, 希望額)

毎月の手数料	if(ローン種類 == "標準ローン")
	{
	毎月の手数料 = 20.00
	}
	else if(ローン種類 == "特別ローン")
	{
	毎月の手数料 = 25.00
	}
	else
	{
	毎月の手数料 = 0
	}

毎月の返済額	PMT(利率, 期間, 希望額)
毎月の返済額 + 毎月の手数料	

DMN を利用することで、このようなデシジョンやビジネス知識モデルを定義し、その利用関係・判断根拠や前提条件などの関係を可視化することができます。また、複数の要素に分割し関係を定義することで、意思決定に必要な情報・条件を明確に識別することができます。結果的に、意思決定に必要な情報の過不足を防ぐこともできま

Copyright (c) 2019-2024 Sparx Systems Japan

す。

3　Enterprise ArchitectでのDMNモデリング

それでは、実際に Enterprise Architect を利用して DMN のモデリングを行う方法を紹介します。

何か具体的な対象がある方が説明しやすいので、このドキュメントでは以下のようなルールをモデリングすることにします。

消費税について、通常は税率 10%だが、食品に限り、8%の税率が適用される。ただし、購入した店で食べる場合には、食品でも 10%の税率が適用される。また、食品と食品以外が組み合わされている商品の場合には、食品以外の原価が 33%を越える場合には食品ではないと見なし、常に 10%の税率が適用される。

3.1　ダイアグラムの作成

まず、Enterprise Architect を起動し、新しいプロジェクトを作成してください。基本的な操作方法はこのドキュメントでは説明を省略しますので、ドキュメント「ゼロから始める Enterprise Architect」などをご覧ください。
https://www.sparxsystems.jp/products/EA/ea_zero.htm

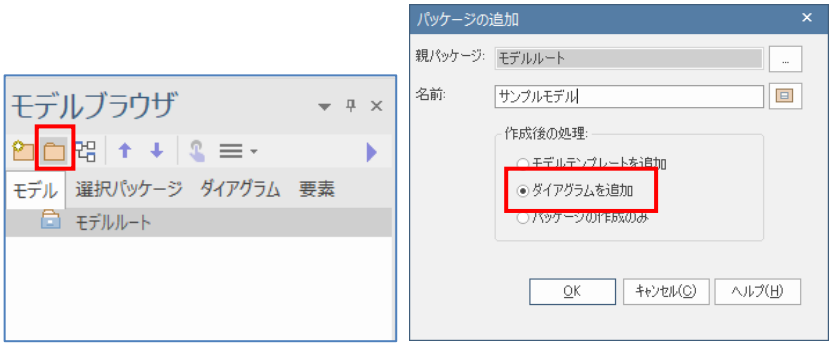
プロジェクトを開いたら、画面右上の「パースペクティブ」ボタンを押し、「要求」→「DMN (意思決定モデリング)」を選択することで、UML など今回利用しない表記方法を隠すことができます。

パースペクティブを選択すると、自動的に「ダイアグラム・モデルテンプレートの追加」ダイアログ(設

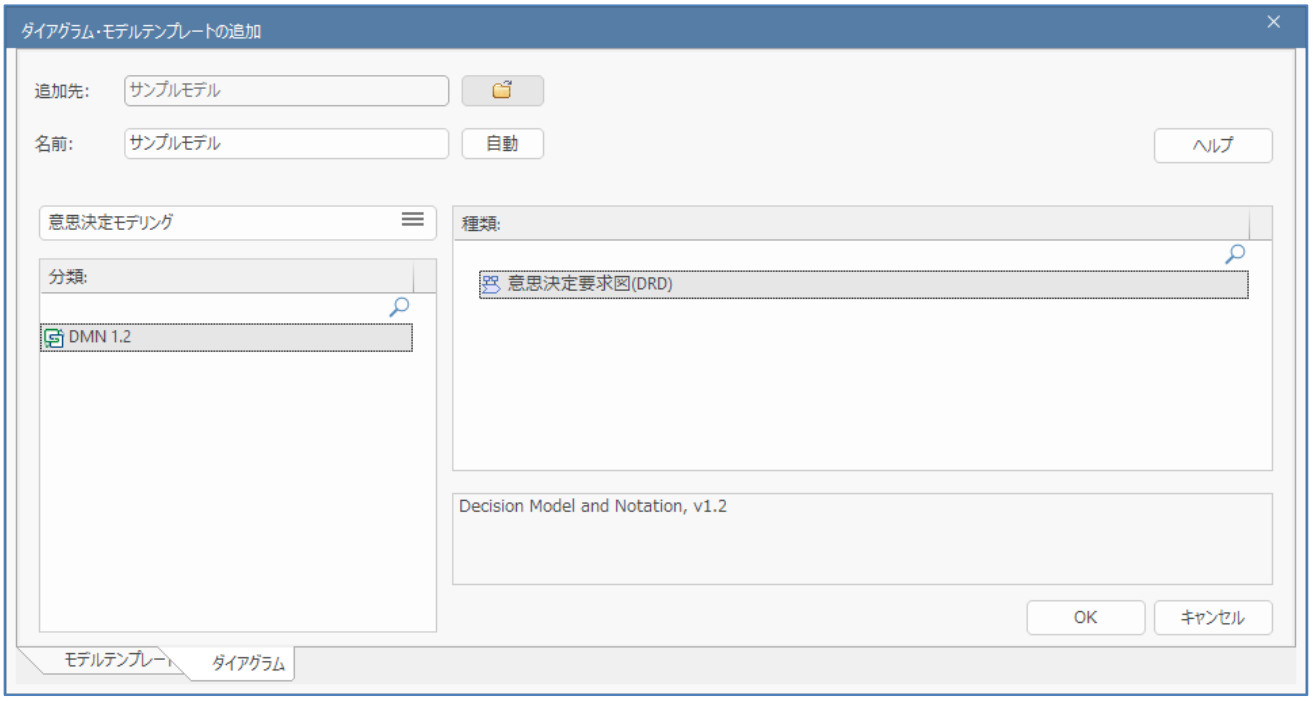
Copyright (c) 2019-2024 Sparx Systems Japan

定によってはタブ)が開きます。このテンプレートには、いくつかの DMN のサンプルモデルが含まれています。この説明では利用しませんので、ダイアログあるいはタブを閉じてください。

左側にある「モデルブラウザ」ウィンドウにあるツールバーから、パッケージの作成ボタンを押すことで、パッケージを追加できます。「パッケージの追加」画面が表示されたら、「名前」を入力し、「ダイアグラムの追加」を選択して OK ボタンを押してください。



続いて「ダイアグラム・モデルテンプレートの追加」画面が開きますので、唯一の選択肢である「DMN1.2」「意思決定要求図(DRD)」を選択して OK ボタンを押してください。これで空のダイアグラム(図)を作成できました。

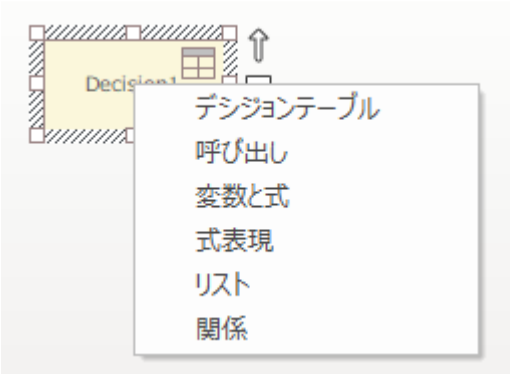


3.2 デシジョンの作成

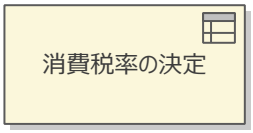
Copyright (c) 2019-2024 Sparx Systems Japan

次に、最終的な意思決定となるデシジョン要素を配置します。今回は、最上位のデシジョンを「消費税率の決定」とし、ルールを分解する形で定義することにします。

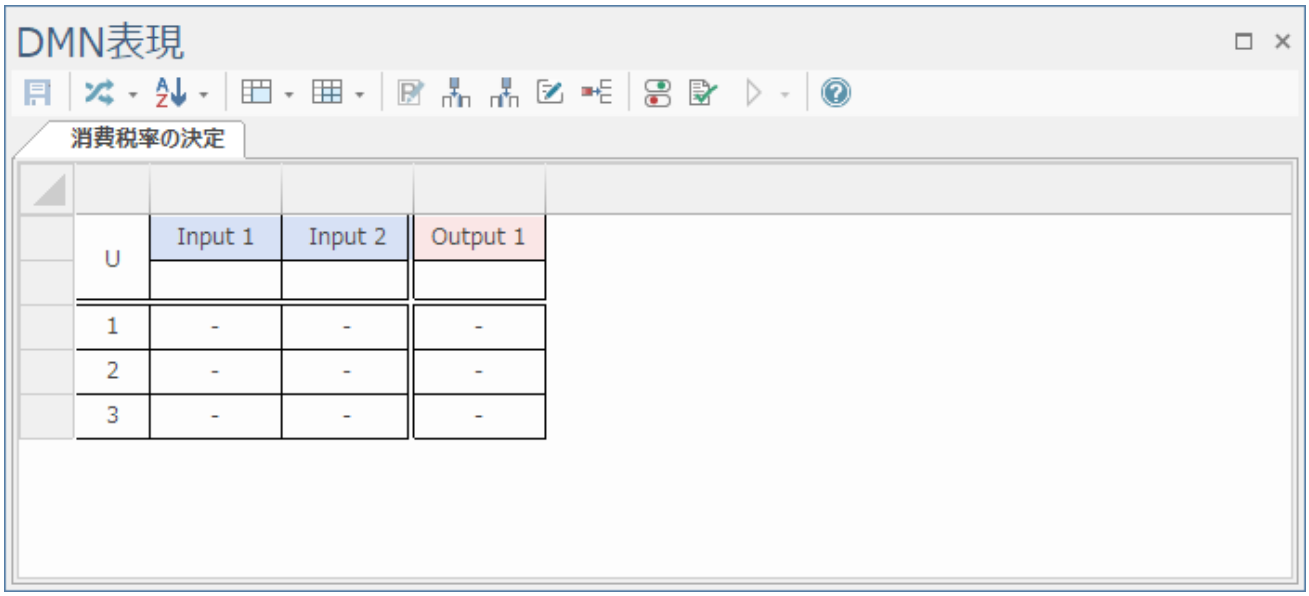
ツールボックスの「デシジョン」をドラッグ&ドロップでダイアグラムに配置します。ドロップした際に種類を指定するメニューが出ますので、今回は「デシジョンテーブル」を選択します。



名前は「Decision1」となりますので、Enterprise Architect の画面の右側にある「プロパティ」サブウィンドウを利用して名前を「消費税率の決定」とします。
(名前の変更は、ショートカットキー「F2」も利用できます。)

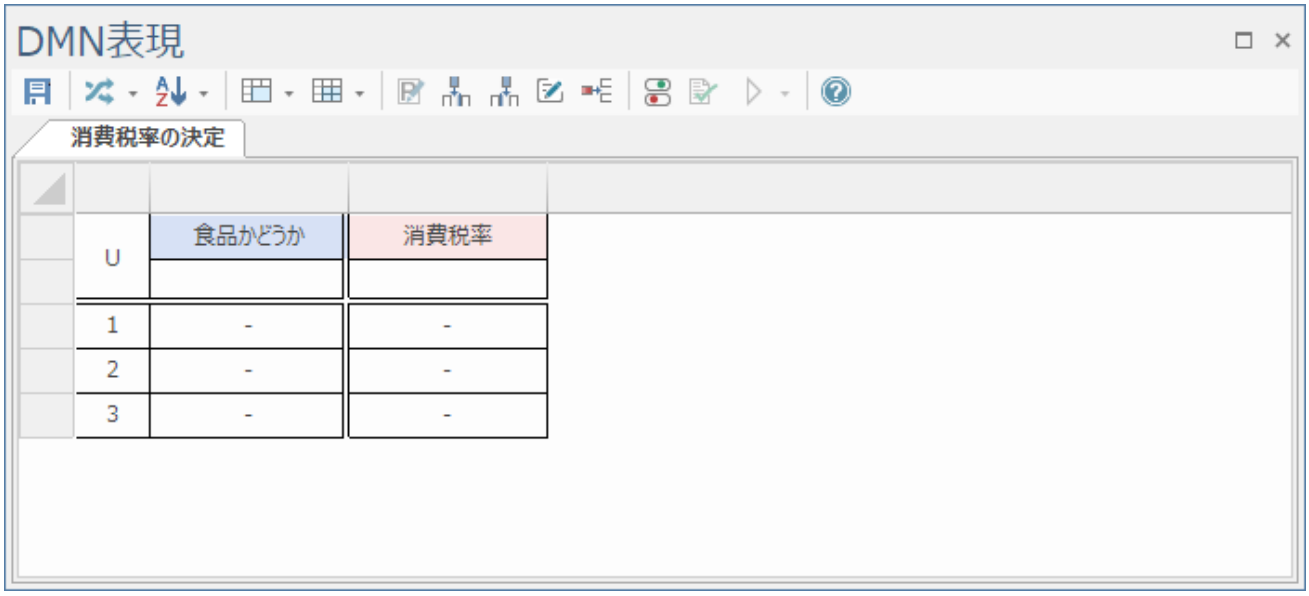


作成した要素をダブルクリックすると、「DMN 表現」ウィンドウが開きます。

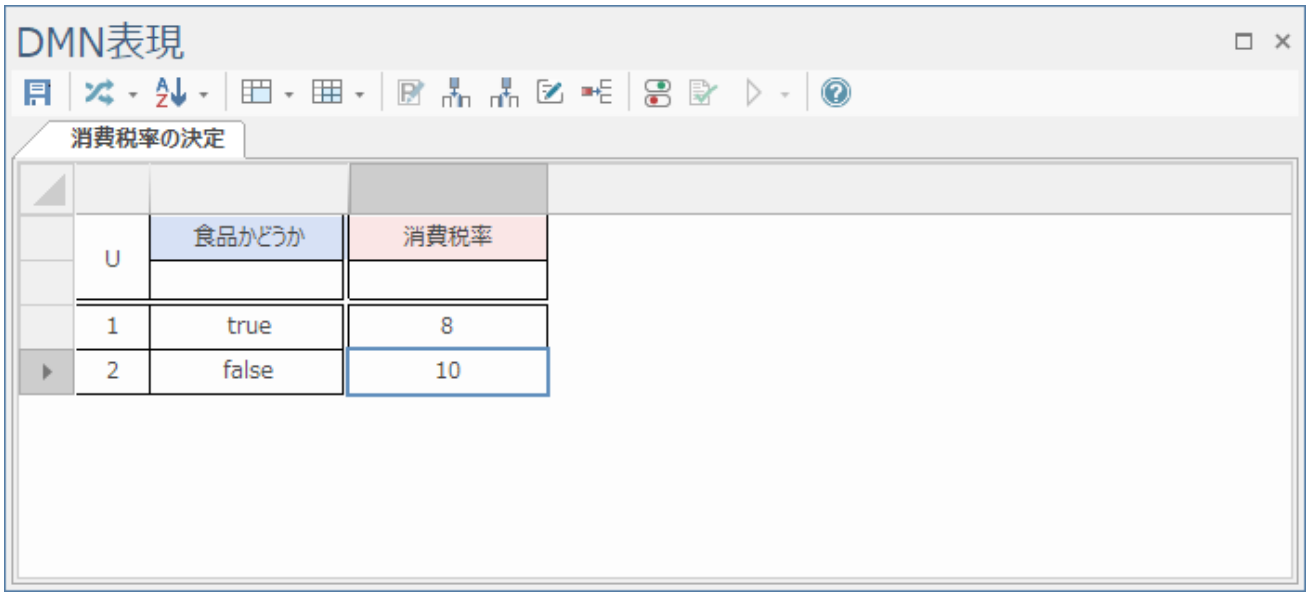


Copyright (c) 2019-2024 Sparx Systems Japan

「Input 1」や「Output 1」と記載されている灰色の見出し部分は、2 回クリックすると文字列を編集できます。それぞれ「食品かどうか」「消費税率」と入力します。入力値は 1 つしかないので、「Input2」の列は、右クリックして「入力列の削除」を実行してください。以下のようになります。



表のそれぞれのセルは、クリックすると内容を編集できます。以下のように内容を入力します。また、今回は 8%か 10%のどちらかになりますので、行は 2 つのみ必要ですので、一番下の行は、行を右クリックして「ルール行の削除」を実行してください。



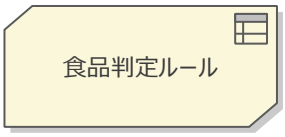
これで、消費税率のデシジョンを定義できました。なお、デシジョンにはこの例のように直接ルールを定義することもできますし、ビジネス知識モデルにルールを定義してそのルールを参照するようにすることもできます。汎用的なルールはビジネス知識モデルとして定義し、デシジョンから参照するようにすることをお勧めします。

Copyright (c) 2019-2024 Sparx Systems Japan

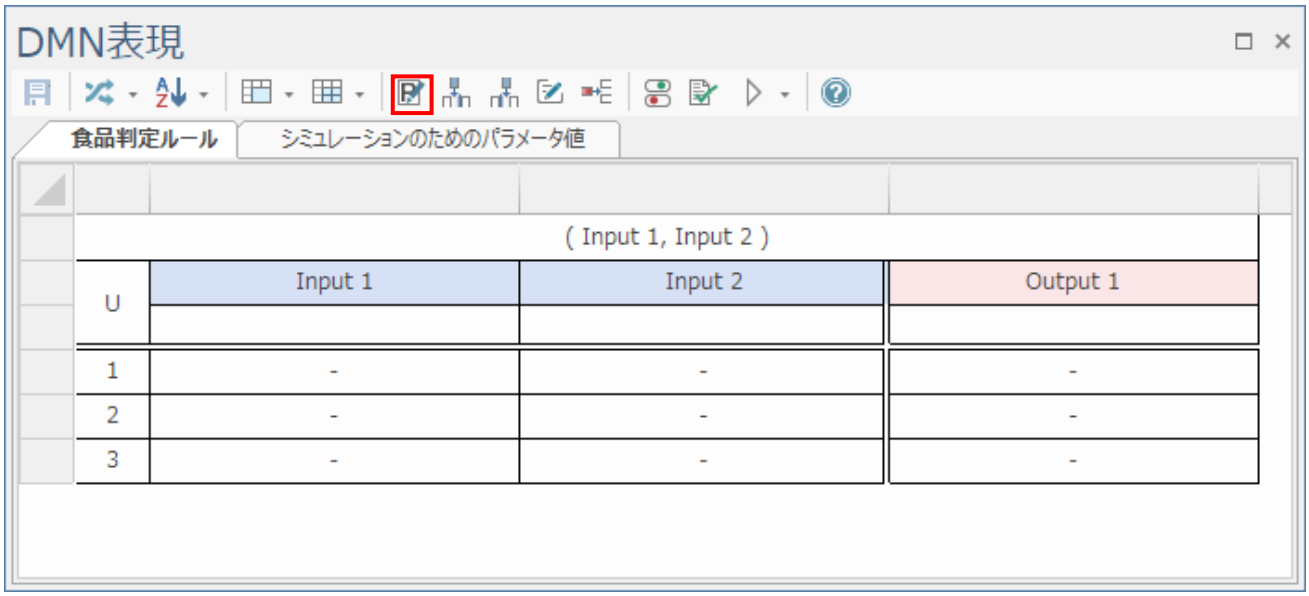
3.3 ビジネス知識モデルの定義

次に、上記のデシジョンで「食品かどうか」の部分を定義します。この定義ではビジネス知識モデルを利用し、デシジョンと結びつける方法を紹介します。

ツールボックスから「ビジネス知識モデル」要素をドラッグ&ドロップで作成してください。先ほどと同様にメニューが表示されますので、今回も「デシジョンテーブル」を選択します。要素の名前を「食品判定ルール」とします。



この食品判定ルール要素についてもデシジョンテーブルの内容を記載しますが、ビジネス知識モデルは再利用を想定していますので、入力値についての定義が必要です。DMN 表現ウィンドウにあるツールボックスの、パラメータの定義のためのボタンを押します。



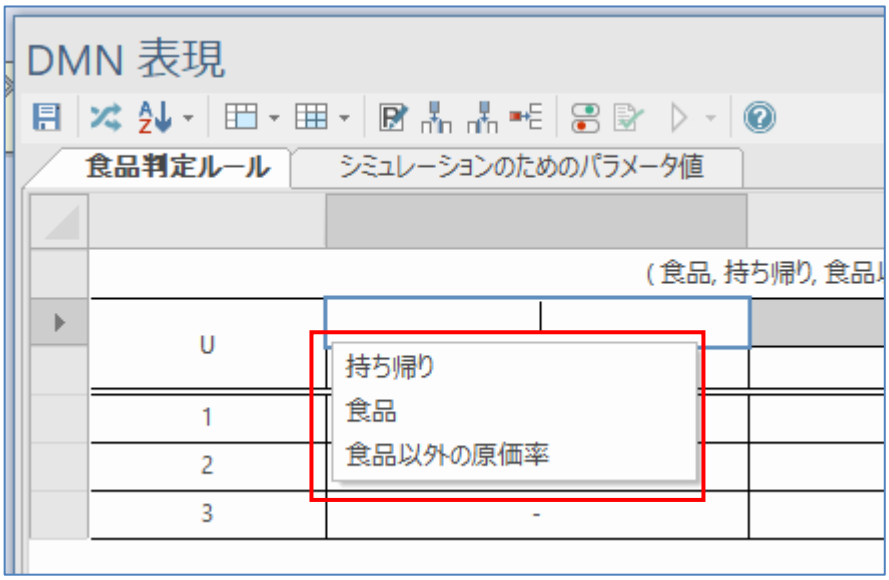
パラメータの編集画面では、既存の内容については名前の欄(Input 1 や Input 2)は 2 回クリックすることで編集できます。また、「パラメータの追加…」と薄く書かれているセルをクリックすることで、パラメータを追加することもできます。それぞれのパラメータには number や boolean 等の種類を指定する必要があります。

Copyright (c) 2019-2024 Sparx Systems Japan

今回は、以下のように設定しました。

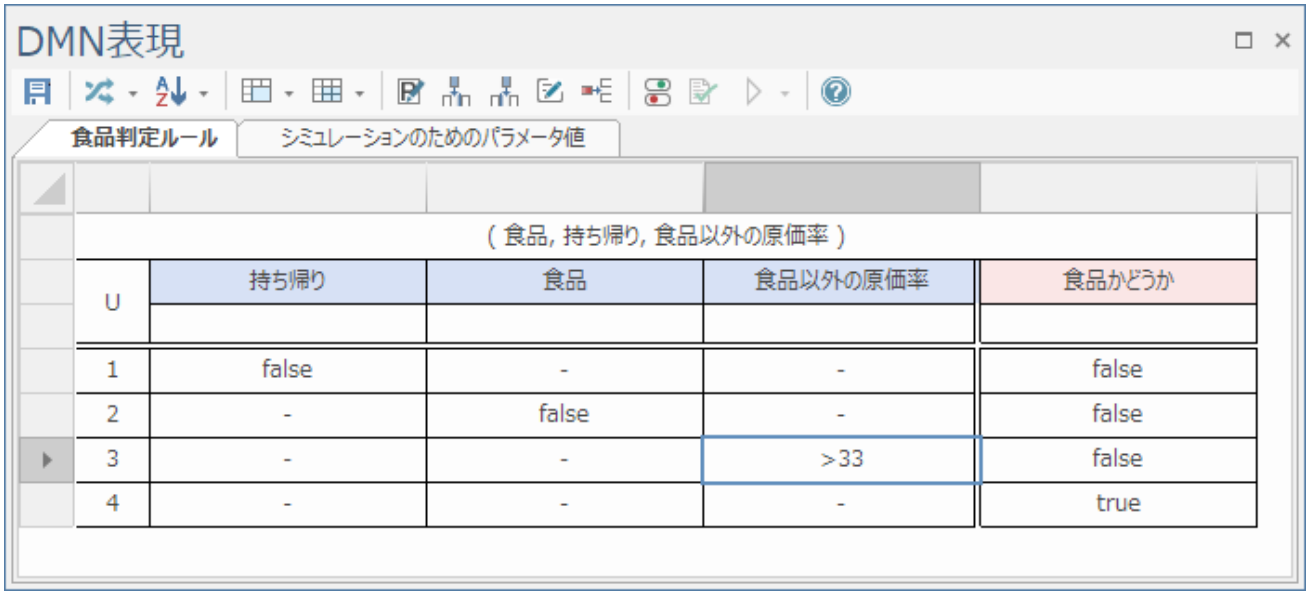
設定後に、デシジョンテーブルの列を調整する必要があります。(自動的に変更はされません)

基本的には、先ほどのデシジョンの例と同じく、見出し部分をクリックして編集する必要があります。ただし、見出し部分の編集時に、Ctrl+スペースキー を押すことで、値を手入力することなく定義した内容を選択して挿入することができます。



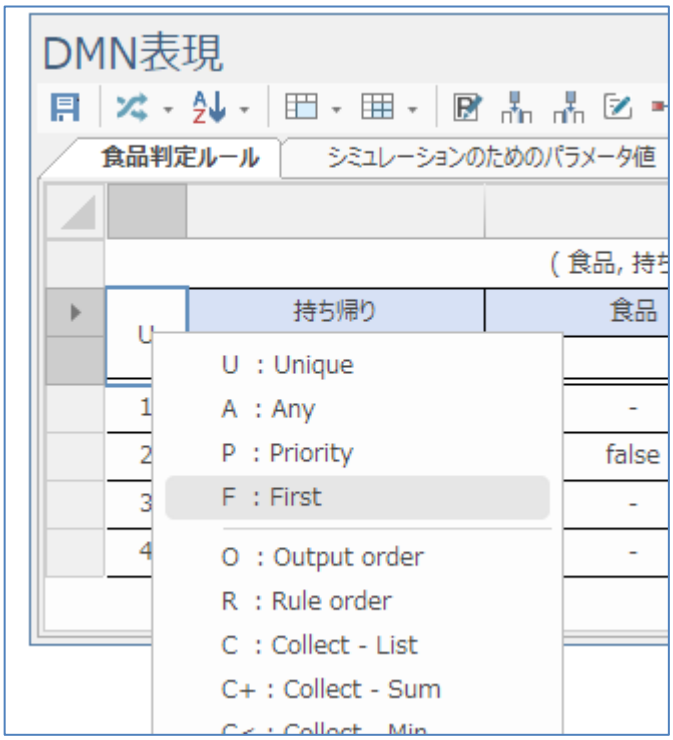
入力列を 1 つ増やし、それぞれの条件を入力します。入力例は以下の通りです。
(入力列を増やすには、ツールバーの ボタンを押します。)

Copyright (c) 2019-2024 Sparx Systems Japan



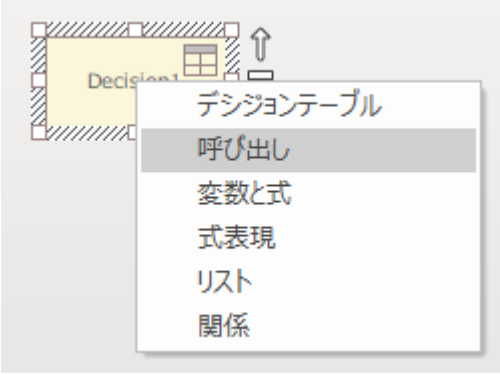
なお、今回の例では、各項目の「アイテム定義」を作成していませんので、true・false の値は手入力する必要があります。後述するアイテム定義要素を作成すると、それぞれのセルの入力時にも Ctrl+スペースキー のショートカットキーが使えるようになります。各セルの「-」(ハイフン 1 つ)は、その項目を判定の対象外とすることを示しています。

また、今回のルールは、上から順番に該当するかどうかをチェックし、最初に条件に該当するルールを採用するという方針で作成しています。このような方針についても指定できます。表の左上の「U」と記載されている箇所を右クリックすると、この方針を指定できますので、「F : First」を選択してください。



Copyright (c) 2019-2024 Sparx Systems Japan

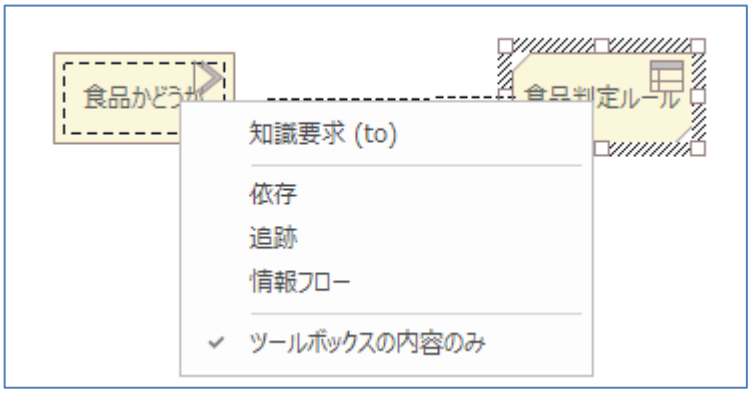
これでビジネス知識モデルの定義が完了です。このビジネス知識モデルを利用するデシジョン要素が必要ですので、デシジョン要素をツールボックスから作成します。ダイアグラムへの配置時に表示されるメニューでは、今回は「呼び出し」を選択してください。名前を「食品かどうか」に設定します。



3.4　要素間を接続する

最後に、今まで作成した要素を結びます。今まで作成した要素は、次の 3 つです。

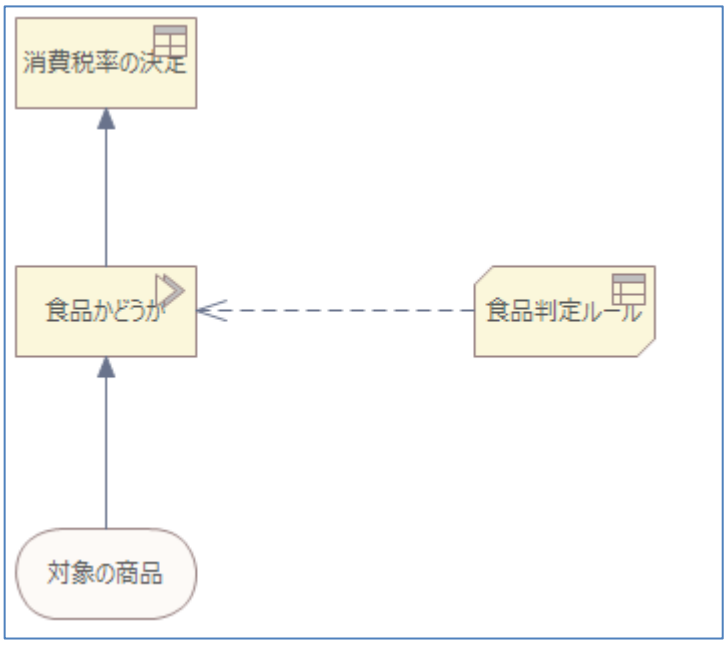
DMN では、要素間を結ぶ接続は「情報要求」「知識要求」「権限要求」の 3 つがあり、それぞれ作成可能な要素の種類が決まっています。Enterprise Architect の「クイックリンク」機能は、この DMN のルールにも対応していますので、以下のように適切な種類が選択肢に表示されます。



Copyright (c) 2019-2024 Sparx Systems Japan

また、既存の要素から、新規要素を作成できます。この場合には、接続の向きも適切に設定されます。ここでは、「食品かどうか」要素で利用する「入力データ」要素を作成し、名前を「対象の商品」とします。

このようにして、以下の図を作成しました。これで、対象の商品が指定された場合に、どのようにして消費税率を計算するかの意思決定の流れを可視化することができました。



この図によって、「対象の商品」について、「食品判定ルール」に従って(消費税率の判断基準による観点での)「食品かどうか」を判断し、その上で「消費税率の決定」をする、という意思決定プロセスが明示できました。

なお、今回は 1 つの「食品判定ルール」というビジネス知識モデルにルールを定義しましたが、内容に複雑度や他のデシジョンとの再利用性の関係によっては、複数のビジネス知識モデル要素にルールの定義を分散させる

Copyright (c) 2019-2024 Sparx Systems Japan

方がよい場合もあります。今回の内容は、ツールの基本的な機能の紹介であるため、実用性・拡張性などは考慮していません。
(逆をいえば、DMN を利用することで、どのようにビジネス知識モデルを定義しデシジョンと結びつけるか、という設計を可視化可能となり、再利用性や将来の拡張・修正を考慮した検討や議論を視覚的に行うことができる、ということになるかと思います。)

4　シミュレーションの実施

第 3 章で説明したように、DMN を利用して意思決定の流れやその根拠(ルール)を可視化できます。また、複数の入力データがある場合には、どのデシジョンでどの入力データを参照しているか、というような関係も可視化できます。

しかし、可視化するだけでは、定義した内容が正しいかどうかの判断は容易ではありません。そのため、実際にはこの章で説明するシミュレーション(試験)機能が重要になります。シミュレーションの機能はユニファイド版以上のエディションで利用できます。

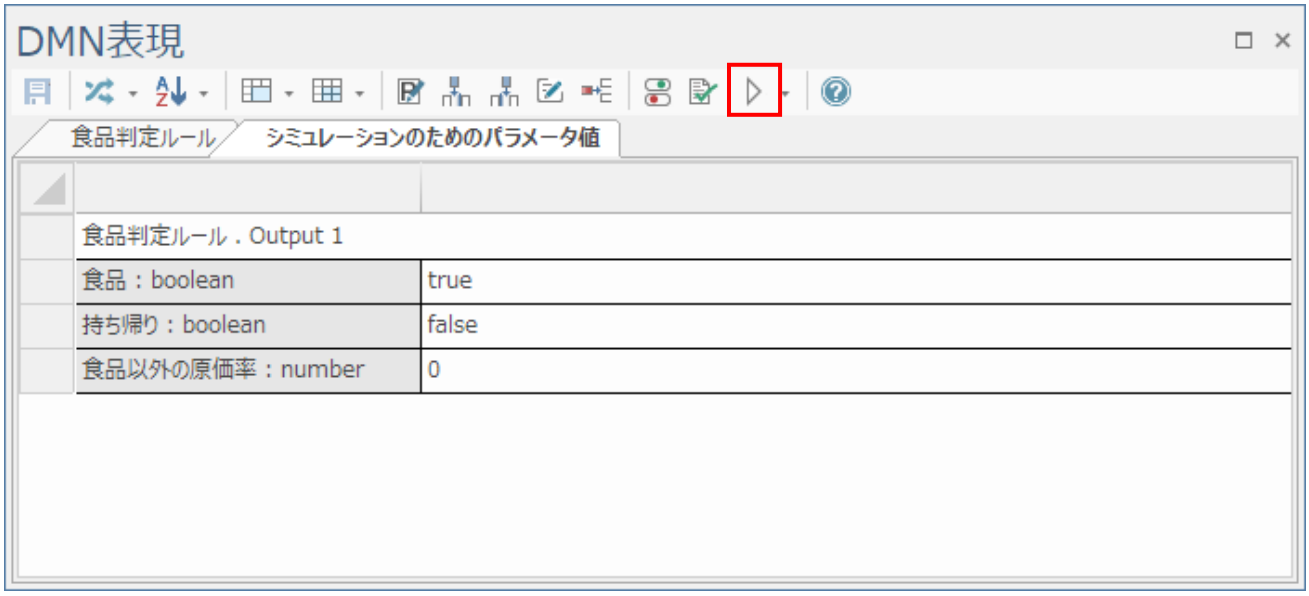
4.1　単一のデシジョンに対するシミュレーション

まずは、前章で作成した「食品判定ルール」の内容が正しいかどうかの確認のために、実際の値を指定してシミュレーションしてみます。

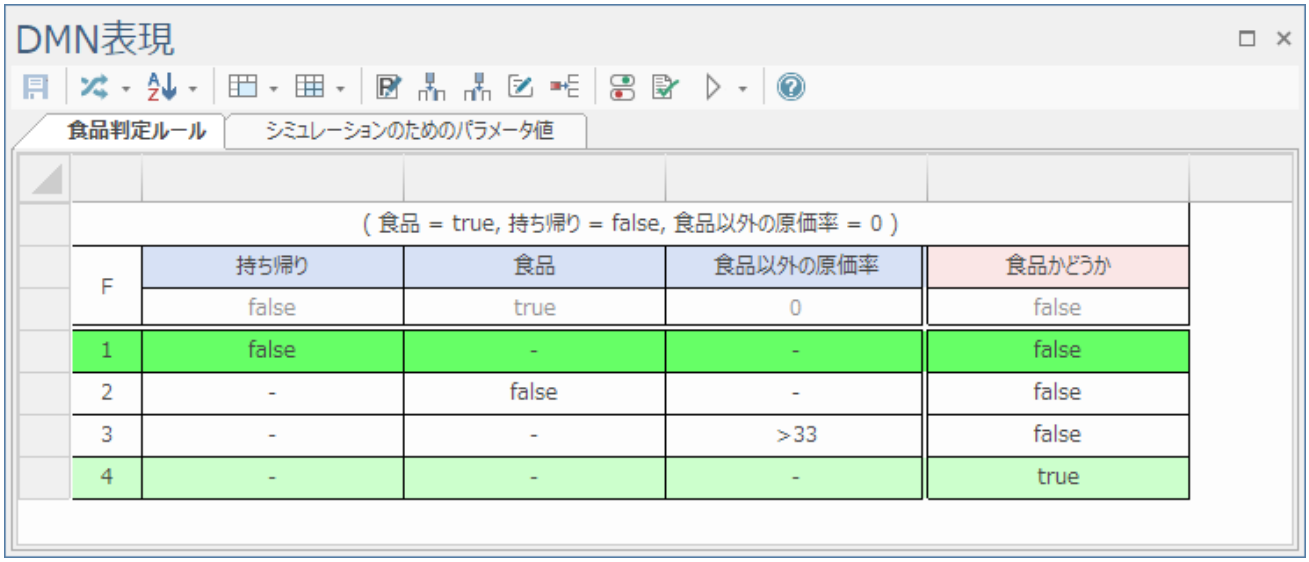
実行するためには、DMN 表現サブウィンドウの「シミュレーションのためのパラメータ値」タブを開き、それぞれのセルにシミュレーションのための値を入力します。

例えば、店内で食品を食べる場合には、以下のような値になります。
(この例では「食品以外の原価率」の項目は 0 としています)

Copyright (c) 2019-2024 Sparx Systems Japan



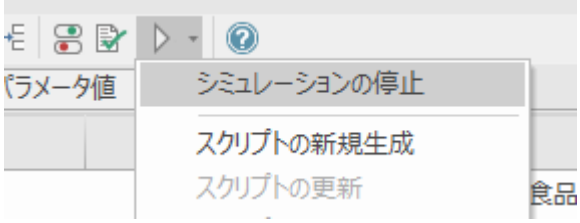
値を設定して保存後、上の画像の赤枠部分のシミュレーション実行ボタンを押すと、以下のように結果を確認できます。ユニファイド版あるいはアルティメット版ではない場合、保存しても赤枠のボタンを押すことはできず、シミュレーション機能は利用できません。



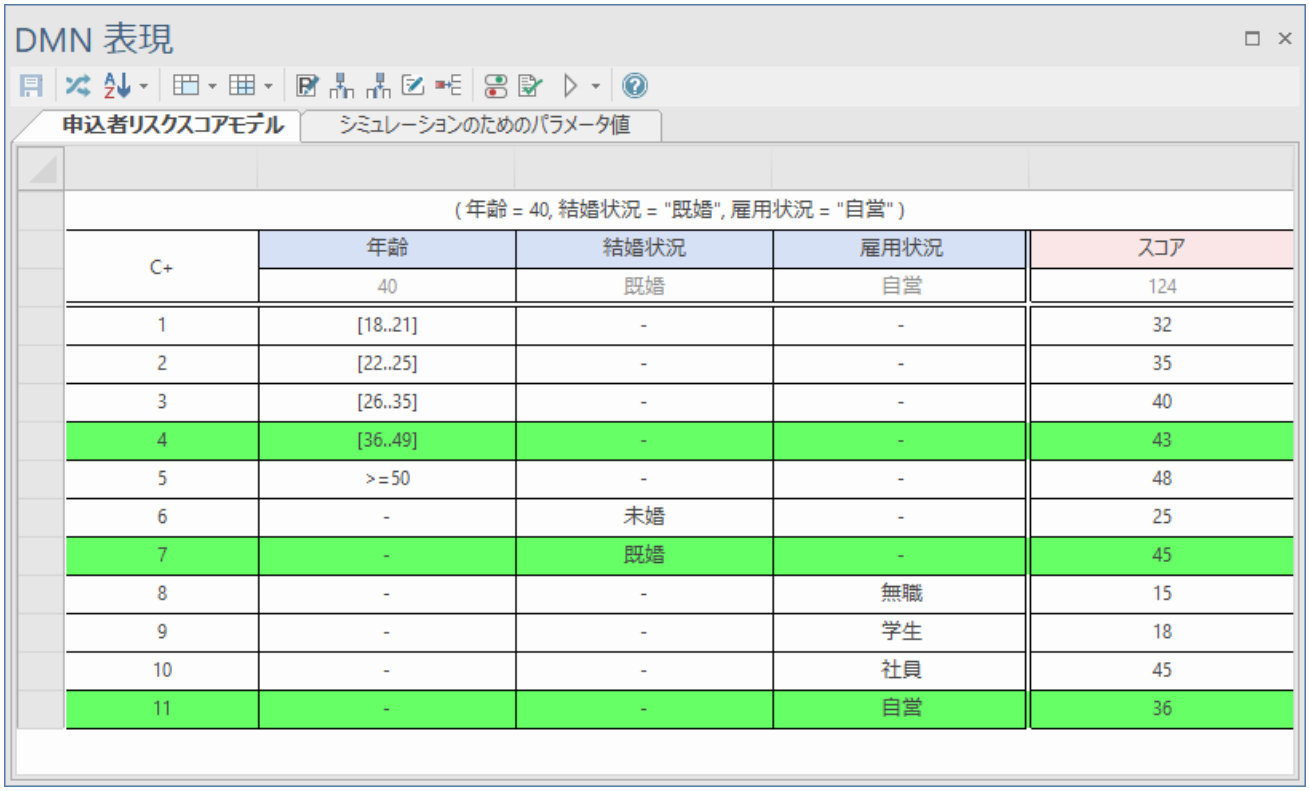
黄緑色の行が、「採用」となった行です。4 行目の薄い黄緑色の行は条件を満たしていますが、今回は「F」、つまり最初に見つかったルールが採用される条件ですので、4 行目は採用されません。

このようにして、さまざまな値を設定して、どのルールが採用されるのかを確認できます。シミュレーションを終了する場合には、シミュレーションの実行ボタンの右端にある黒三角の領域を押してメニューを表示し、「シミュレーションの停止」を選択してください。

Copyright (c) 2019-2024 Sparx Systems Japan



複雑な内容になるほど、シミュレーション機能の有用性は高まります。以下の例は、「C+」、つまり条件を満たす全てのルールの数値を足し合わせる場合の例です。



DMN 表現

申込者リスクスコアモデル　シミュレーションのためのパラメータ値

(年齢 = 40, 結婚状況 = "既婚", 雇用状況 = "自営")

C+	年齢	結婚状況	雇用状況	スコア
	40	既婚	自営	124
1	[18..21]	-	-	32
2	[22..25]	-	-	35
3	[26..35]	-	-	40
4	[36..49]	-	-	43
5	>=50	-	-	48
6	-	未婚	-	25
7	-	既婚	-	45
8	-	-	無職	15
9	-	-	学生	18
10	-	-	社員	45
11	-	-	自営	36

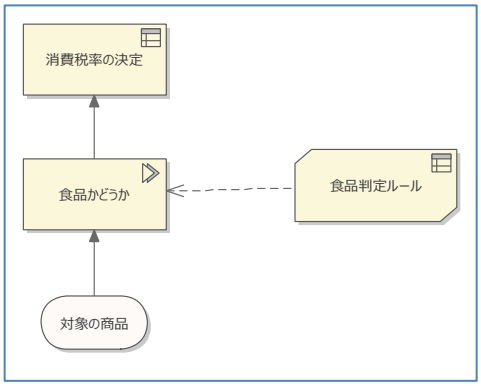
この画像の例のように、「C+」の場合には該当するルールに定義されている出力(「スコア」)の数値を全て足し合わせます。「スコア」のピンク色のセルの下に表示されている「124」が、この条件の場合に算出されるスコアの合計値となります。

こうして、定義したビジネス知識モデルについて、内容が正しいかどうかを実際の値を指定して検証することができます。

4.2　複数のデシジョンに対するシミュレーション

次に、第 3 章で作成した以下の内容を検証したいと思います。

Copyright (c) 2019-2024 Sparx Systems Japan

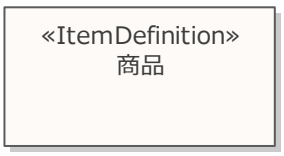


この例で、「対象の商品」について具体的な値を入力し、最終的に消費税率が 8%なのか 10%なのかを確認します。

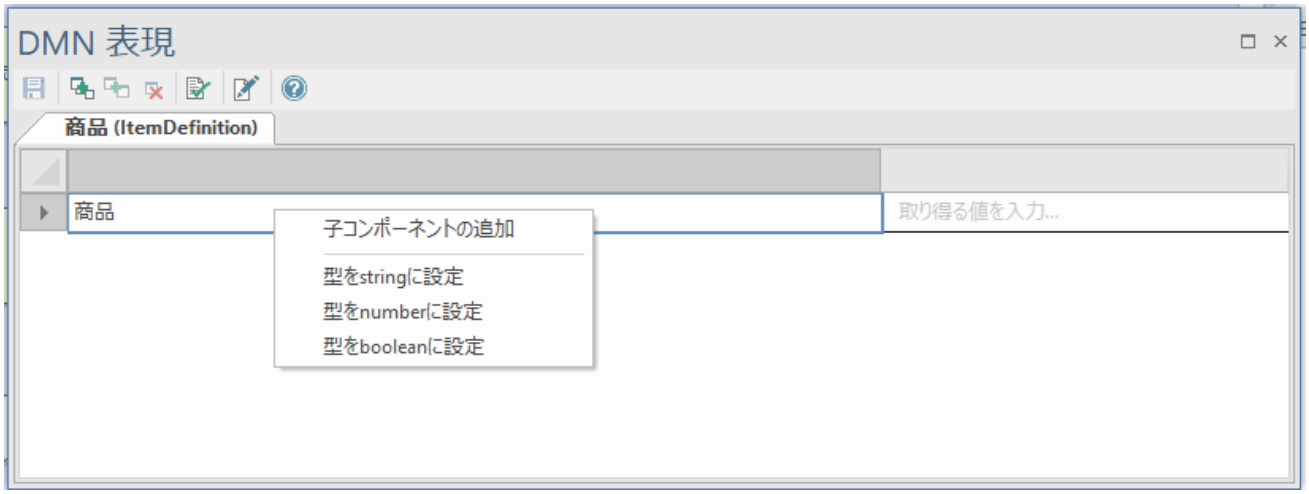
4.2.1 アイテム定義の追加

この「対象の商品」の入力データには、「食品かどうか」などのいくつかのパラメータがあります。こうした、複数の値で構成される入力データをシミュレーションする場合には、その値の構成を定義する必要があります。

まず、ツールボックスから「アイテム定義」要素を図にドロップして配置し、名前を「商品」としてください。



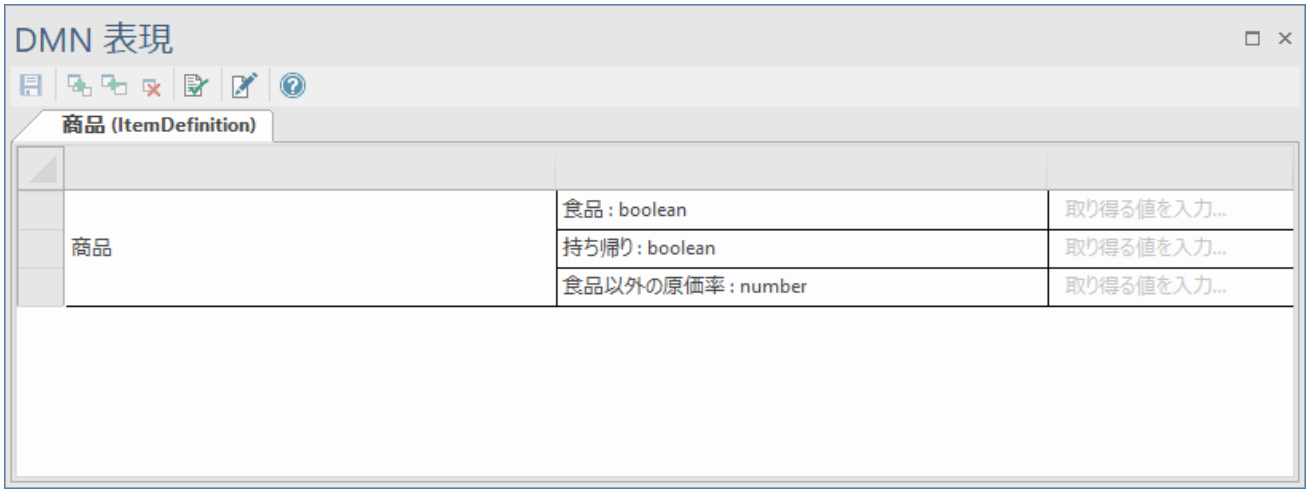
選択すると、DMN 表現ウィンドウにこの商品要素の定義が表示されます。右クリックして「子コンポーネントの追加」を選択してください。



Copyright (c) 2019-2024 Sparx Systems Japan

「New Component」の欄が増えますので、クリックして名前を編集し「食品」としてください。また、そのセルを右クリックして、「型を boolean に設定」を選び、真偽値(true/false)とします。

同様に右クリックして「兄弟コンポーネントの追加」を選択すると他の値を追加できますので、次の画像のように、値と型を指定します。

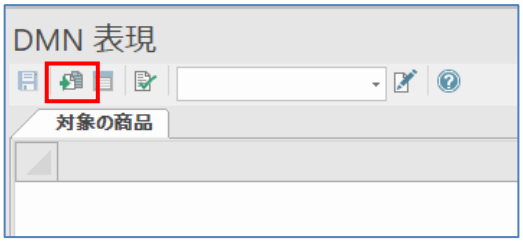


このようにして、商品がどのような値を持つのか、ということを階層的に定義しました。シミュレーション機能を利用する場合および次章のソースコード生成を利用する場合には、アイテム定義要素での型定義が必要です。

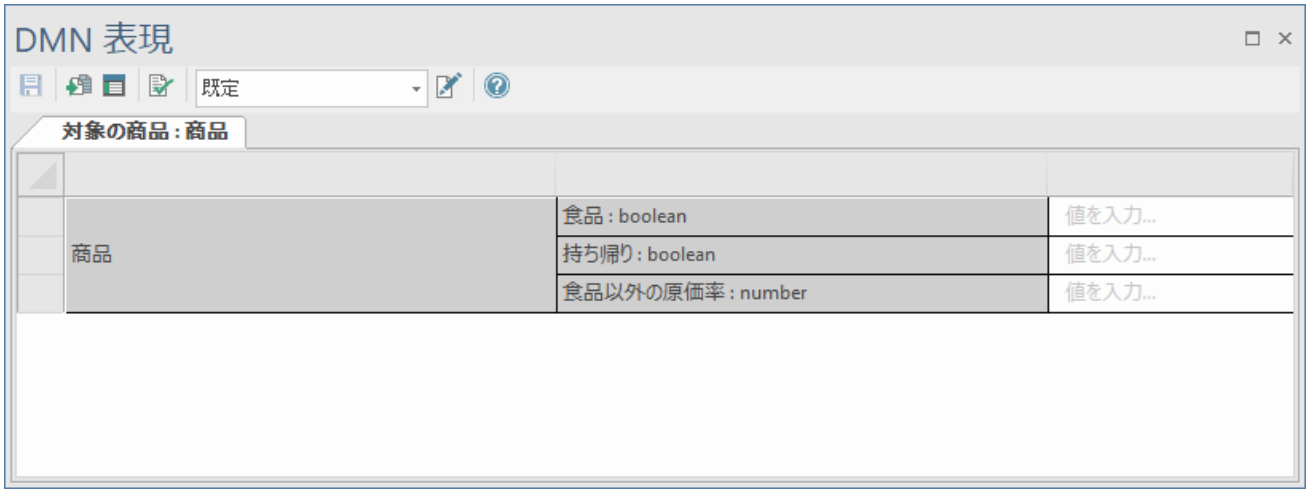
4.2.2 入力データとの結びつけ

このようにして、シミュレーションで利用する情報を定義したら、シミュレーションで利用する入力データ「対象の商品」と結びつけます。「対象の商品」要素をダイアグラム内で選択すると DMN 表現ウィンドウの内容が変わりますので、ツールバーにある結びつけのためのボタンを押し、今定義した「商品」のアイテム定義と結びつけます。

Copyright (c) 2019-2024 Sparx Systems Japan



すると、以下のように定義の内容が表示され、「値の入力」欄にシミュレーションの値を指定できるようになります。それぞれの「値の入力」欄では、Ctrl+スペースキー のショートカットキーで値を指定入力できます。



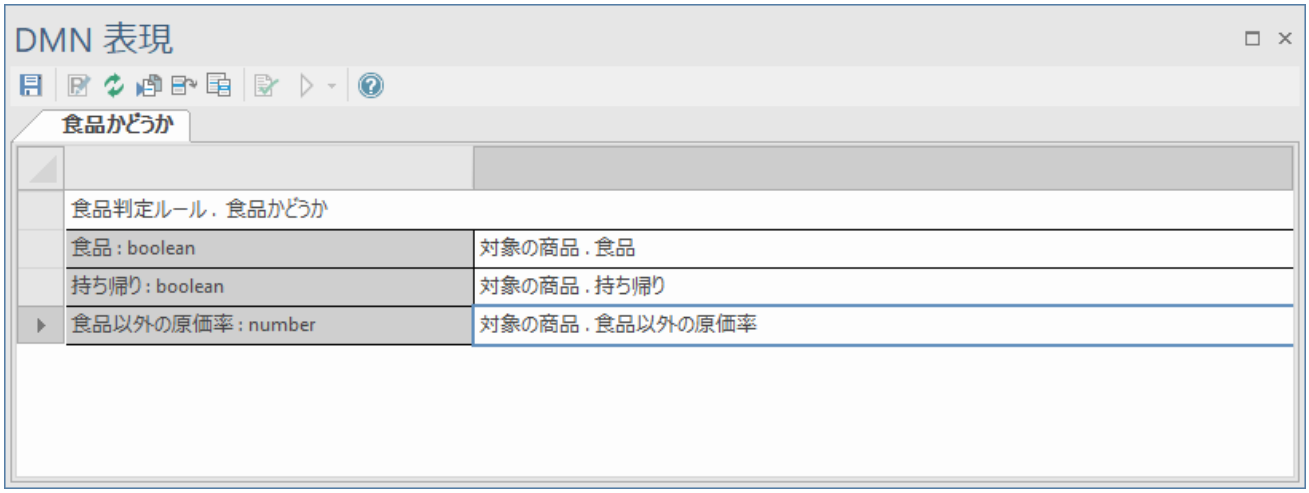
それぞれ、シミュレーションのための値を入力し保存してください。

4.2.3 入力データとデシジョンの結びつけ

入力データの型と値を指定したら、その入力データを参照するデシジョン「食品かどうか」に対して、入力データの結びつけを行う必要があります。「食品かどうか」要素をダブルクリックすると、DMN 表現ウィンドウに内容が表示されます。それぞれのセルをクリックして編集状態にした段階で、Ctrl+スペースキー のショートカットで、下の画像のようにシミュレーション実行時に参照する値を指定できます。

Copyright (c) 2019-2024 Sparx Systems Japan

このようにして、入力データのそれぞれの値と、ビジネス知識モデルのそれぞれのパラメータをどのように結びつけるかを指定する必要があります。



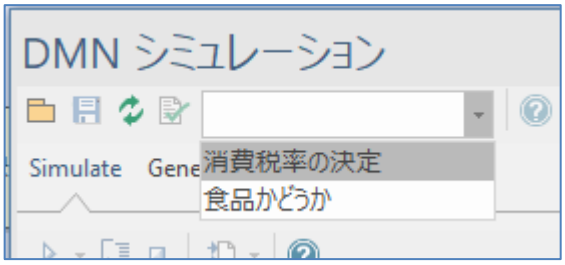
4.2.4 シミュレーション要素の作成と実行

複合的なシミュレーションを実行するためには、「シミュレーションの設定」要素が必要です。

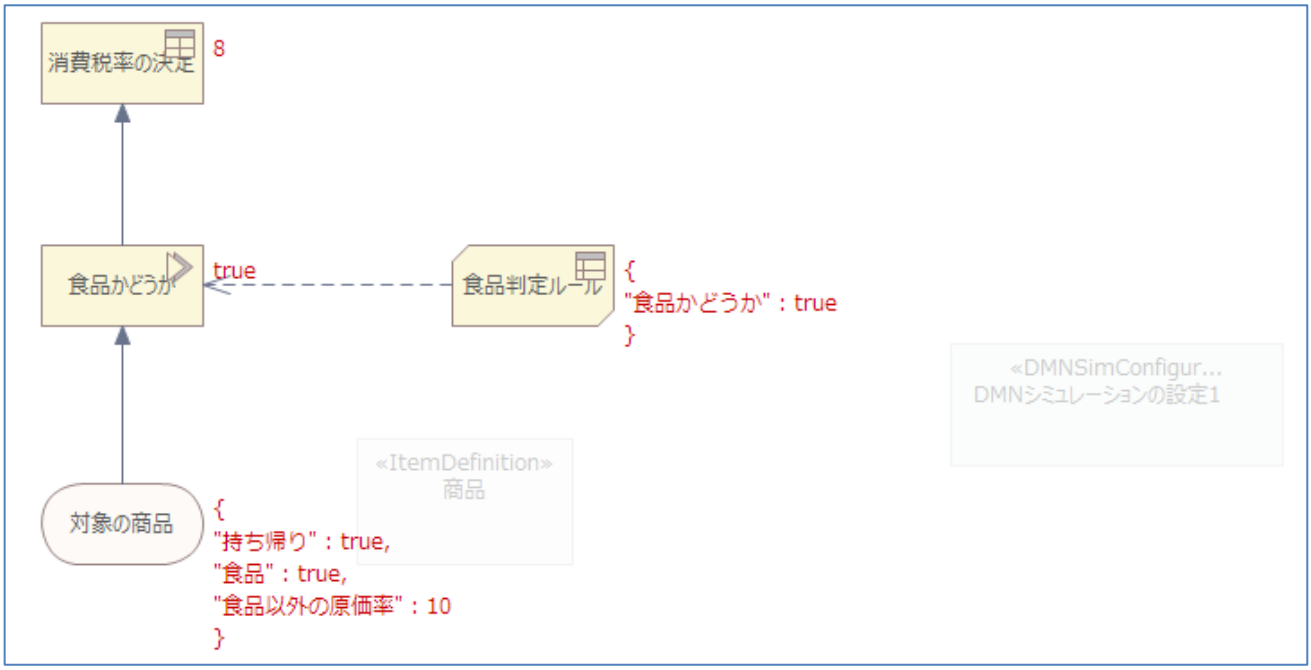
ツールボックス「DMN の要素と接続」グループから「シミュレーションの設定」をダイアグラムにドラッグ&ドロップして配置してください。(要素の名前はシミュレーションには直接関係しませんので、必要に応じて変更してください。)

Copyright (c) 2019-2024 Sparx Systems Japan

この要素をダブルクリックすると、「DMN シミュレーション」サブウィンドウが開きます。ツールバーにあるドロップダウンリストを開くと、デシジョン要素の名前が表示されますので、「消費税率の決定」を選択します。



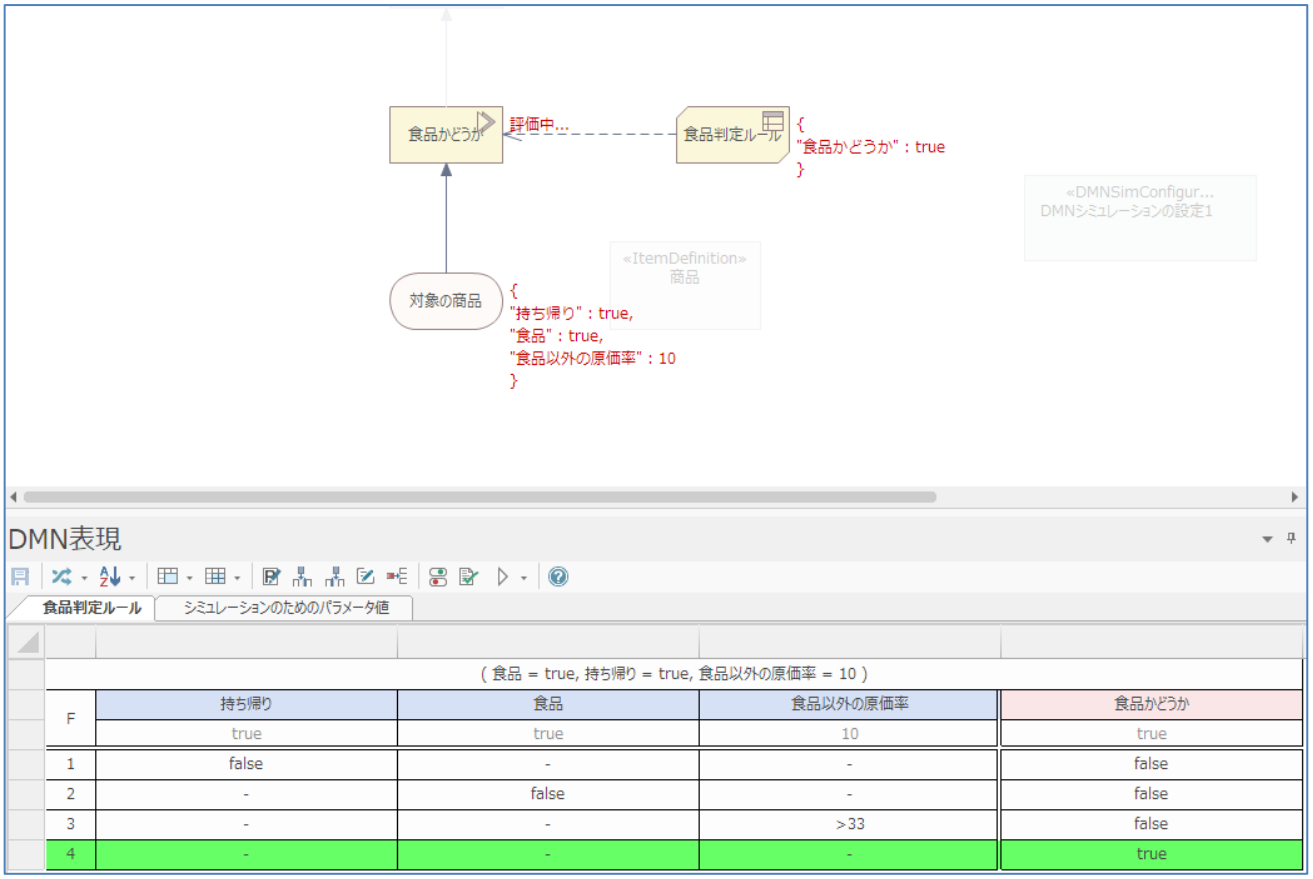
ここまでの設定が正しく完了していれば、シミュレーションの実行ボタン(Simulate タブ内のツールバーの左端の ボタン)を押すと、以下のように入力した値についての判断結果が表示されます。



上記のような結果が表示されない場合には、何らかの設定が不足しています。例えば、「対象の商品」要素の横に「null」と表示される場合、持ち帰りなどの値の設定が不足しています。

DMN シミュレーションウィンドウ内の ボタンで、いわゆるステップ実行を行うことができます。シミュレーション結果に至るまでの経過を 1 ステップずつ確認できます。
(確認中は、DMN 表現ウィンドウでも関係する内容が表示されます。)

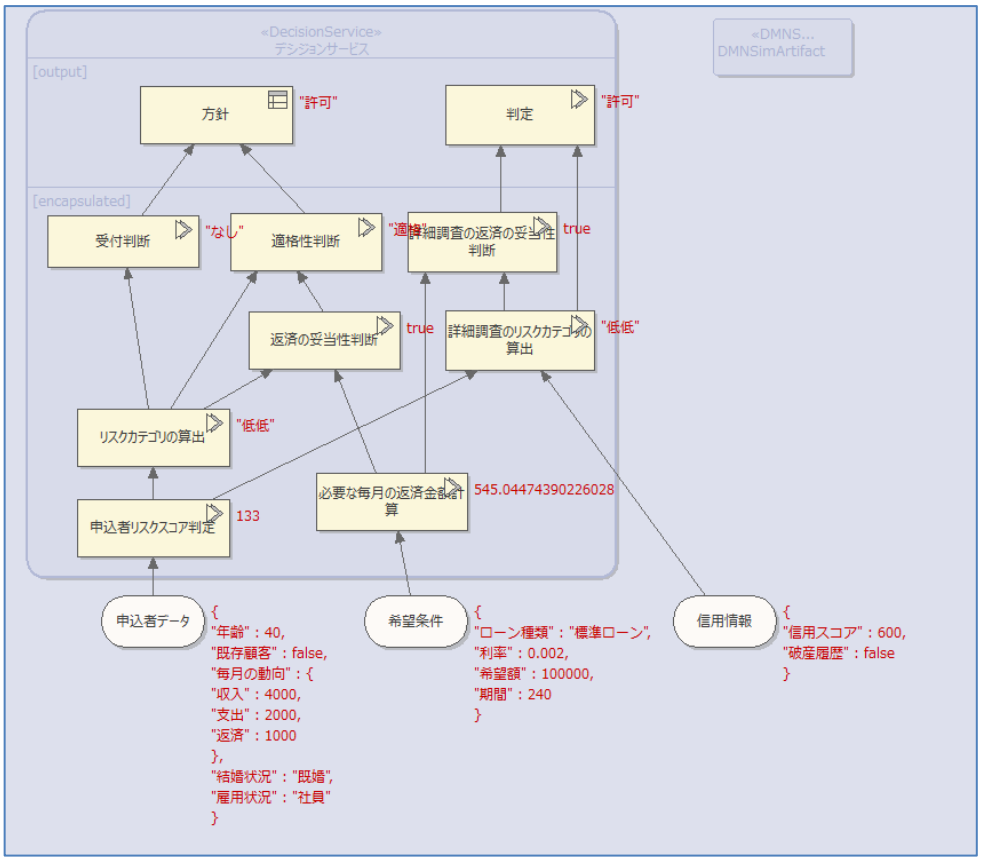
Copyright (c) 2019-2024 Sparx Systems Japan



このようにして、シミュレーション機能を利用すると、複雑なデシジョンの組み合わせで何らかの意思決定が行われる場合に、その経過も参照することができます。意図しない結果になった場合には、どうしてそうなったかを途中経過を確認し、問題点を見つけられます。

以下の例のような複雑なモデルになればなるほど、シミュレーション機能の効果は高まります。

Copyright (c) 2019-2024 Sparx Systems Japan



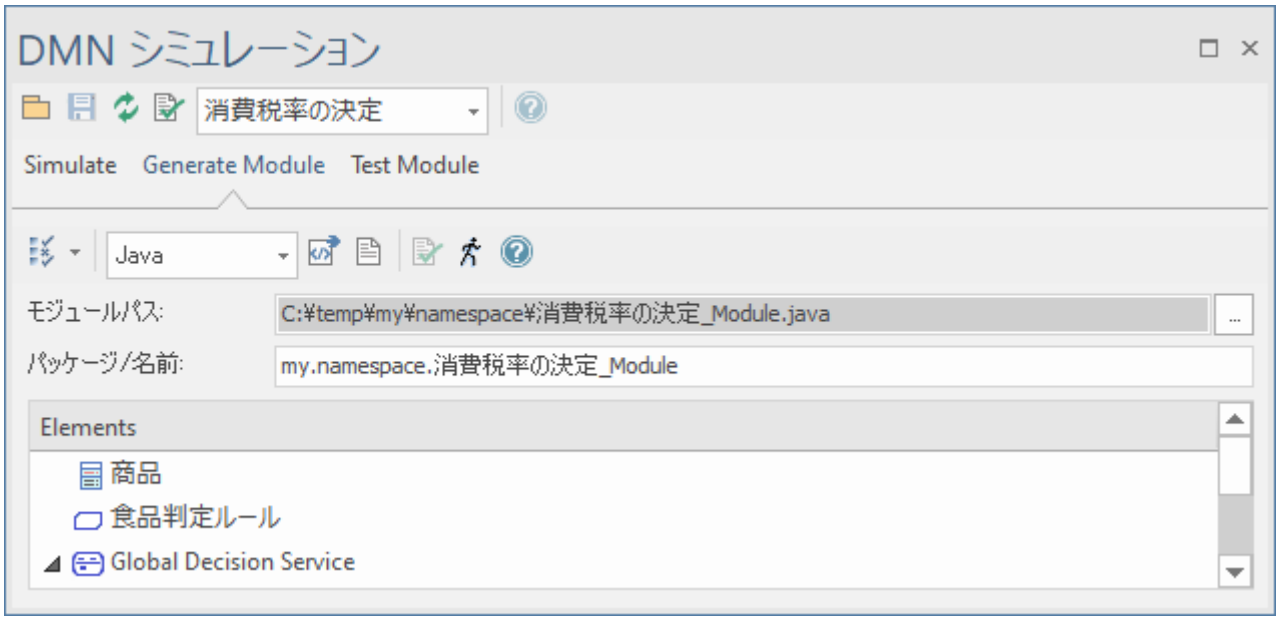
なお、このドキュメントでは触れませんが、入力データ要素には「データセット」として複数の値を保持し、切り替えて実行できます。今回の例では、例えば「食品以外の原価率」が異なる複数のデータセットを定義し、切り替えて結果を確認できます。DMN モデルは試行錯誤しながら内容を変えることが多いので、データセットを定義しておけば、モデルの変更後にシミュレーションする場合に、値をその都度入力する必要がなくなります。

5　ソースコードの生成

シミュレーション機能で定義した内容が適切であることが確認できたら、その内容からソースコードを生成できます。自動生成することにより、定義した内容をソースコードとして作成する手間が省けるほか、手作業による誤りの混入を防げます。

ソースコードを生成するためには、DMN シミュレーションウィンドウの Generate Module タブを利用します。

Copyright (c) 2019-2024 Sparx Systems Japan



上の画像のように「モジュールパス」と「パッケージ/名前」を指定し、生成ボタン を押します。次の画像のように、ソースコードが自動生成されます。

```
//Business Knowledge Model Definitions
//Function Defined In: "食品判定ルール", Request Output:"食品かどうか"
public static double 食品判定ルール_食品かどうか(boolean 食品, boolean 持ち帰り, double 食品以外の原価率)
{
    //DecisionTable HitPolicy: "FIRST"; Request Output: "食品かどうか"
    ArrayList<Double> 食品かどうか_collection = new ArrayList<>();
    //DecisionTable Rule Definitions Begin
    //Rule #: 1
    if((食品) == false)
    {
        double outputEntryValue = false;
        食品かどうか_collection.add(outputEntryValue);
    }
    //Rule #: 2
    if((持ち帰り) == false)
    {
        double outputEntryValue = false;
        食品かどうか_collection.add(outputEntryValue);
    }
    //Rule #: 3
    if((食品以外の原価率)>33)
    {
        double outputEntryValue = false;
```

なお、上の例のように、DMN モデルの内容がそのままソースコードとして生成されますので、要素名などに日本語を利用している場合には、そのまま日本語で出力されることになります。生成される内容は、Enterprise Architect の「コード生成テンプレート」の機能を利用してカスタマイズできます。

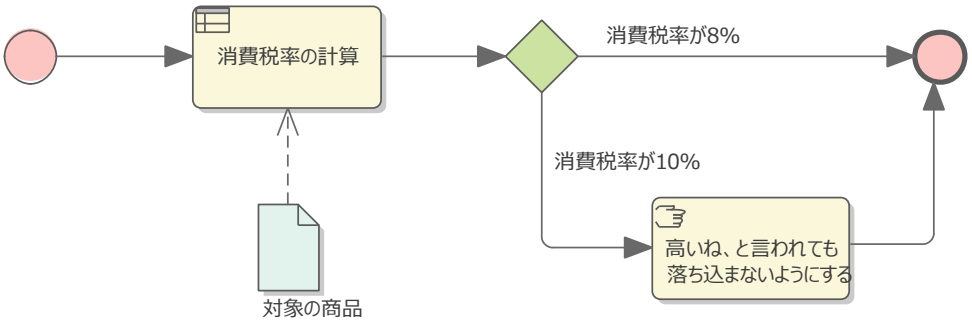
Copyright (c) 2019-2024 Sparx Systems Japan

6　BPMNとの連携（BPSim）

最後に、作成した DMN モデルのもう 1 つの活用法として、BPMN との連携について紹介します。

Enterprise Architect には、BPMN モデルのシミュレーション機能(BPSim)を搭載しています。この機能は、BPMN で表現される業務などの処理について、例えば「店にレジは何台必要か」「お客様の平均待ち時間はどのくらいになるか」「その場合の人件費はいくらになるか」などを検証するために利用できます。

この BPMN の処理(アクティビティ)において、DMN で定義した内容を参照できます。例えば、以下のような BPMN モデルで、アクティビティ「消費税率の計算」で、前の章までで作成したデシジョンを利用できます。

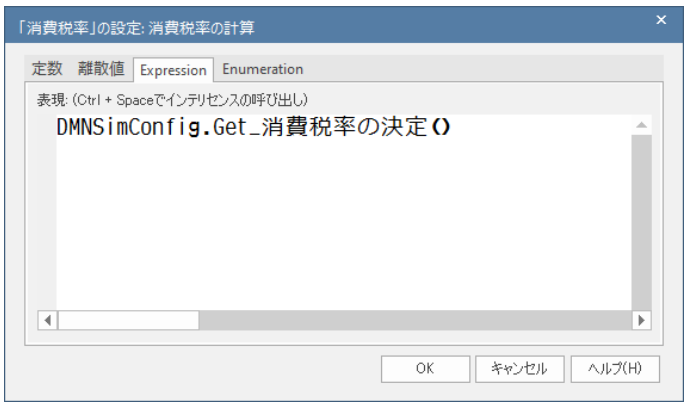


手順の概要は以下の通りです。詳細はこのドキュメントでは割愛します。

1. BPMN モデルを作成する
2. BPMN シミュレーション要素を作成し、DMN シミュレーション要素と依存の関係で結ぶ
   (これにより、他の設定で Ctrl+スペースキーのショートカットで、DMN で定義された内容を呼び出すことができるようになります。また、BPSim の実行時に、自動的に DMN で定義された内容からソースコードを生成し、ビルドします。)

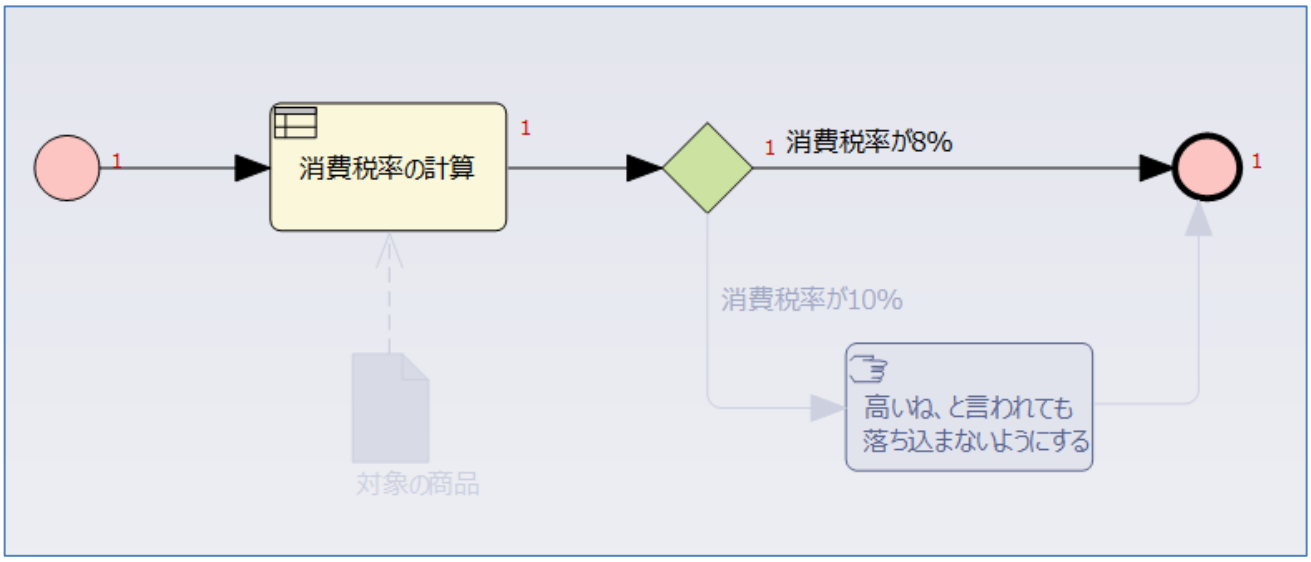
Copyright (c) 2019-2024 Sparx Systems Japan

3. 種類が「ビジネスルール」のアクティビティに対して、DMN モデルからコード生成する Java のメソッドを呼び出すようにプロパティを追加する



4. そのデシジョンで利用するデータオブジェクトに対して、DMN モデルのシミュレーションで利用した値をコピーする
5. その他、BPMN シミュレーションのための設定を行い、BPSim を実行する

これらの設定を行うことで、以下の例のように DMN モデルで定義された内容を利用し、BPMN シミュレーションを実行できます。



Copyright (c) 2019-2024 Sparx Systems Japan