

Sim4stamp 取扱い説明書

(作業中)

第 1 版

2007 年 8 月

by
Keiichi Tsumuta

ドキュメント履歴

目次

1.	はじめに	3
2.	動作環境	4
3.	シミュレーション手順	5
4.	Sim4stamp の起動	6
5.	Overture 環境の設定	7
6.	プロジェクト設定	8
7.	モデル生成	10
8.	VDM++モデル生成	14
9.	データ生成	16
10.	VDM++の実行	17
11.	シミュレーション実行結果表示	18
12.	Q&A	19

1. はじめに

Sim4stamp は、STAMP/STPA 解析を行うときの思考を補佐する簡易シミュレーションツールとして作成されたものです。

Overture を用いて、VDM++を使用してシミュレーションを作成し、実行することが出来ます。

VDM++は簡易シミュレーションを作成するには適切な手法ですが、データの設定や出力については貧弱な方法しかありません。Sim4stamp は、VDM++の I/O に関する機能を補うために、データの設定、出力、モデリングを、GUI を用いて行うことが出来るようにしたものです。ただし、機能的には STAMP/STPA 用に特化したものとなっています。

2. 動作環境

Sim4stamp の動作環境を以下に示します。

- (1) Overture VDM++モデリング統合ツール
- (2) Java 8 以降の Java ランタイム環境
- (3) Java と Overture が動作する 32Bit または 64Bit OS

Java の実行環境は、内部的な作りの関係上、Java 8 以降が必要になります。

コマンドプロンプトやコンソール上で「`java -version`」をタイプして「Enter」を打つと Java のバージョンが表示されますので、そのバージョンが「1.8.0」以上となっていることを確認します。

(例)

```
java -version
java version "1.8.0_141"
Java(TM) SE Runtime Environment (build 1.8.0_141-b15)
Java HotSpot(TM) 64-Bit Server VM (build 25.141-b15, mixed mode)
```

もし、このコマンドがエラーになる場合は、Java のランタイムがシステムにインストールされていないと思われますので、その場合は、以下の URL よりダウンロードし、Java をインストールしてください。

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Overture は以下の URL からダウンロードします。

<http://overturetool.org/download/>

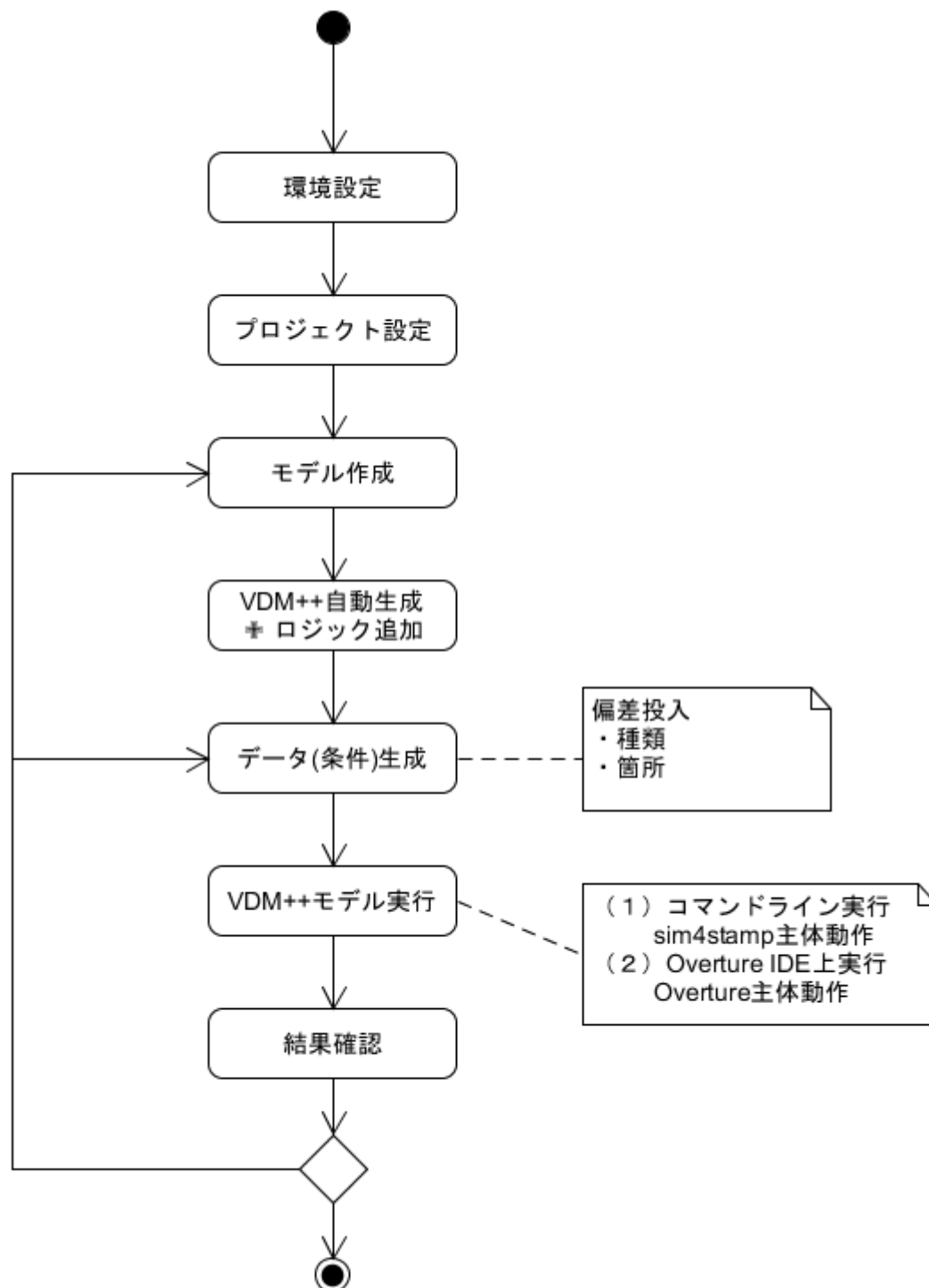
Overture では、デフォルトのエンコーディングは Windows では SJIS となっているので、Sim4stamp と組み合わせて用いるときは、マルチバイトコードが文字化けしてしまいます。そのため、以下の処置を行ってデフォルトのエンコーディングを UTF-8 に変更しておきます。

Overture のインストールフォルダの直下の「Overture.ini」をエディタで開き以下のコードを追加します。

```
-Dfile.encoding=utf-8
```

3. シミュレーション手順

以下に、本ツールの使用手順を示します。



4. Sim4stamp の起動

Sim4stamp の起動は、以下に示す方法（複数あり）で行います。

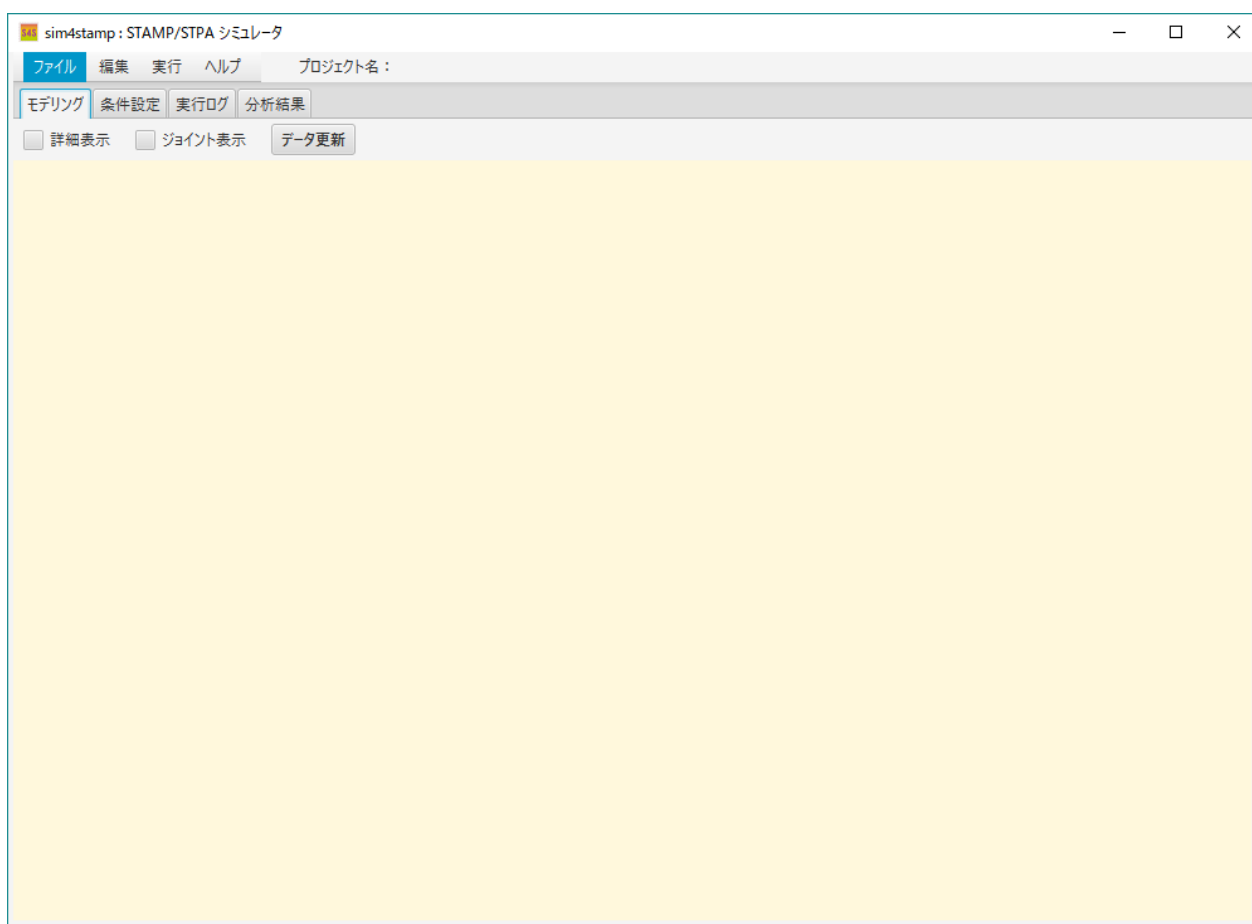
(1) 「sim4stamp-X.X.X.jar」をダブルクリックする。

(2) コマンドプロンプトより

```
java -jar sim4stamp-X.X.X.jar
```

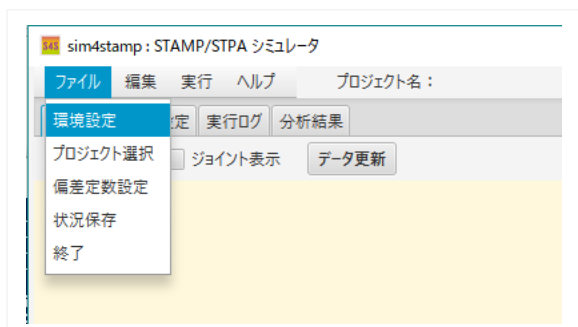
をタイプし、Enter キーを押します。

起動が成功すると以下に示す画面を表示します。

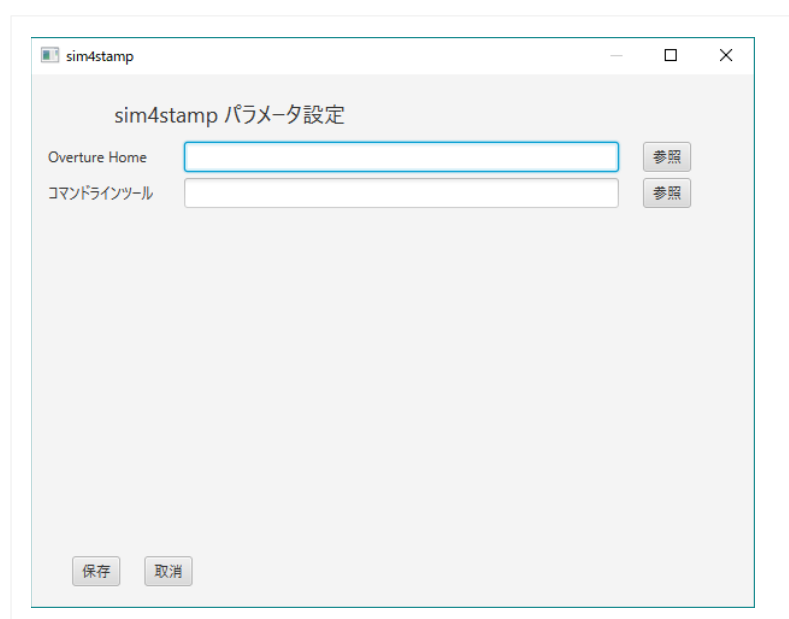


5. Overture 環境の設定

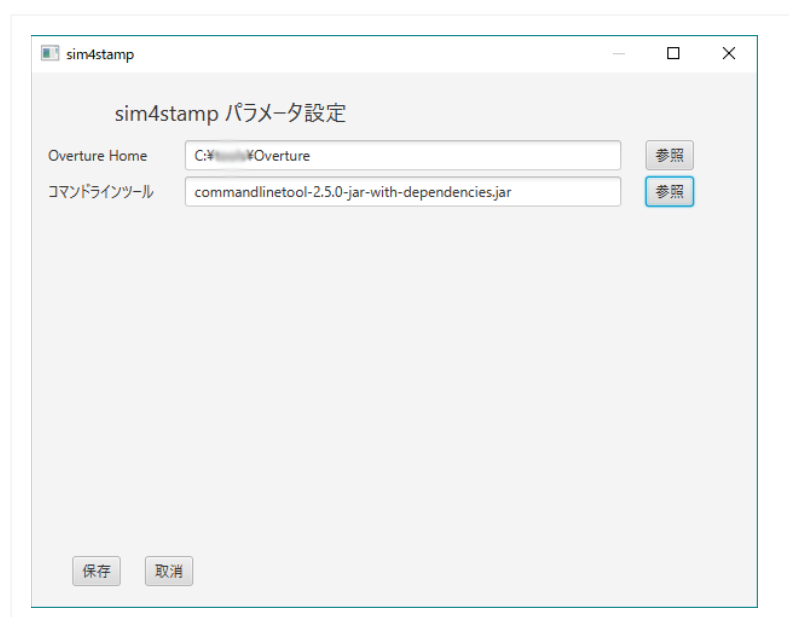
Sim4stamp より Overture を用いるために、Overture を登録します。



「ファイル」－「環境設定」を開きます。



「Overture Home」には Overture のインストールフォルダを指定します。



注)

コマンドラインツールは、Overture のホームフォルダ下の「commandline」にあります。

6. プロジェクト設定

VDM++のプロジェクトを **Overture** 側で作成し、そのプロジェクトを **Sm4stamp** に設定します。
「ファイル」－「プロジェクト選択」を開くと以下のダイアログを表示します。

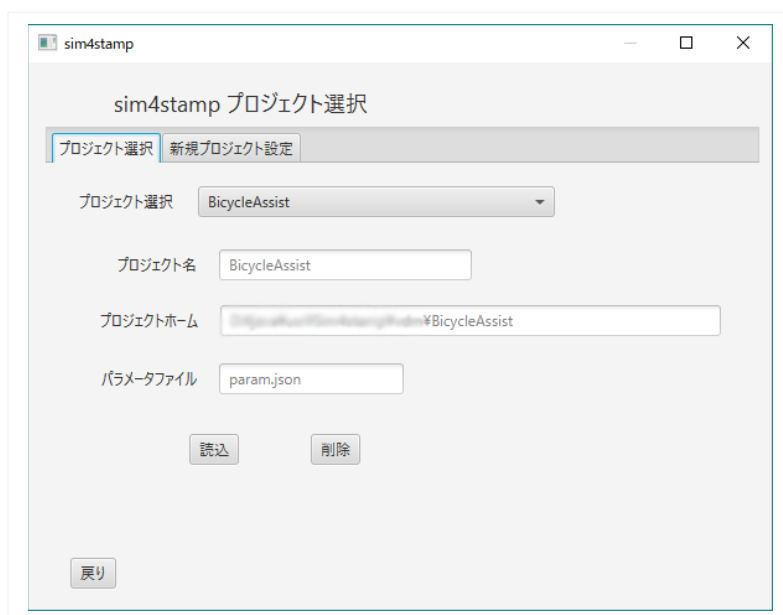
The screenshot shows the 'sim4stamp プロジェクト選択' dialog box with the 'プロジェクト選択' tab selected. The '新規プロジェクト設定' tab is also visible. The 'プロジェクト選択' dropdown menu is set to '未選択'. Below it are three text input fields: 'プロジェクト名', 'プロジェクトホーム', and 'パラメータファイル'. At the bottom are two buttons: '読込' and '削除'. A '戻り' button is located at the bottom left of the dialog.

The screenshot shows the 'sim4stamp プロジェクト選択' dialog box with the '新規プロジェクト設定' tab selected. The 'プロジェクト名' field contains 'BicycleAssist'. The 'プロジェクトホーム' field contains 'http://www.sim4stamp.jp/BicycleAssist' and has a '参照' button to its right. The 'パラメータファイル' field contains 'param.json'. At the bottom is a '登録' button. A '戻り' button is located at the bottom left of the dialog.

「新規プロジェクト設定」タブに移行し、**Overture** 側のプロジェクトの情報を設定します。

パラメータファイルは、**Sim4stamp** 側で追加するファイルですが、デフォルトのままとします。

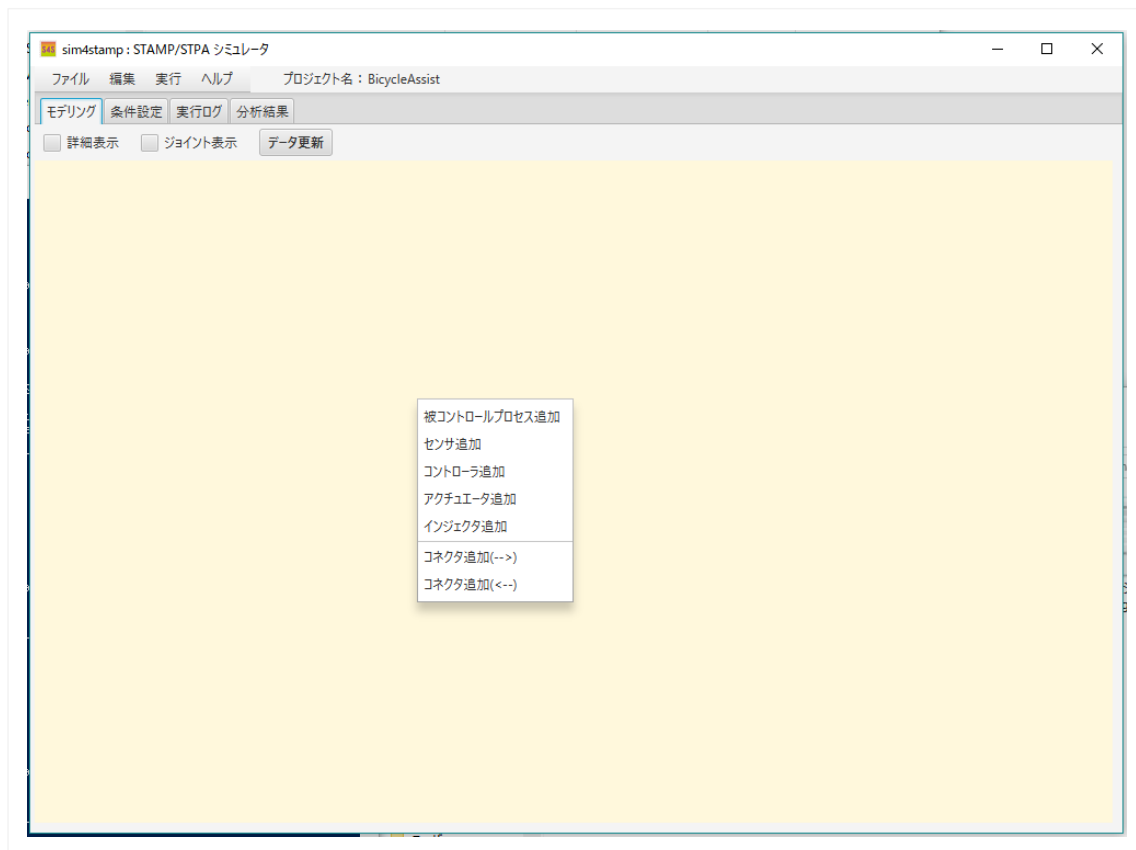
「登録」ボタンをクリックし、登録を行います。



登録後、「プロジェクト選択」タブに戻り、プロジェクト選択（プルダウンメニュー）を行って、「読み込」ボタンをクリックすることにより、**Sim4stamp** のプロジェクトが選択プロジェクトになります。

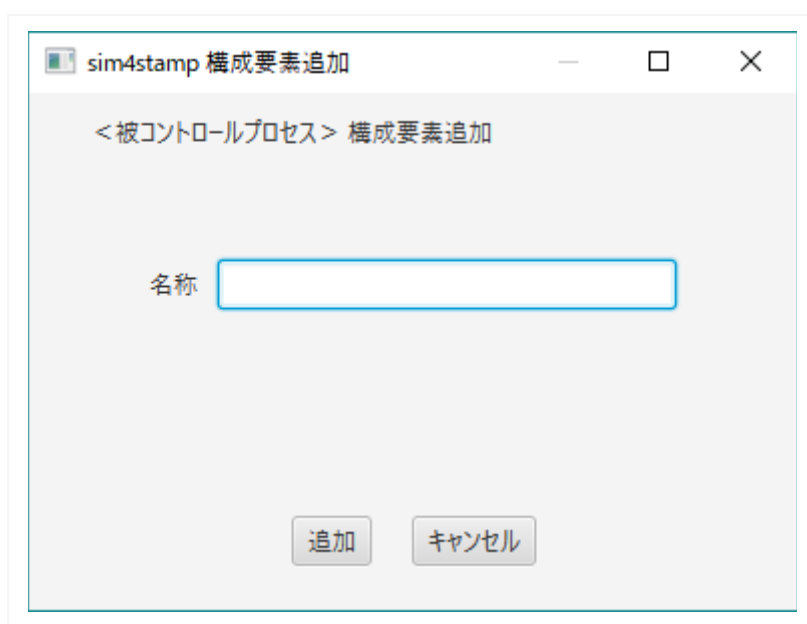
7. モデル生成

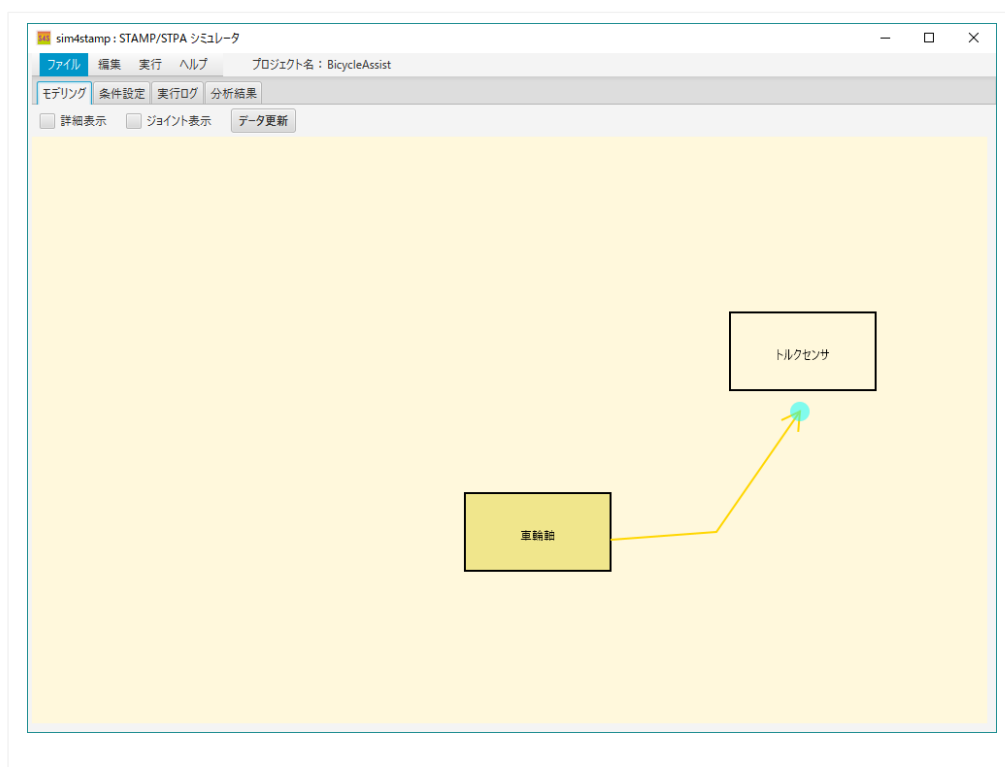
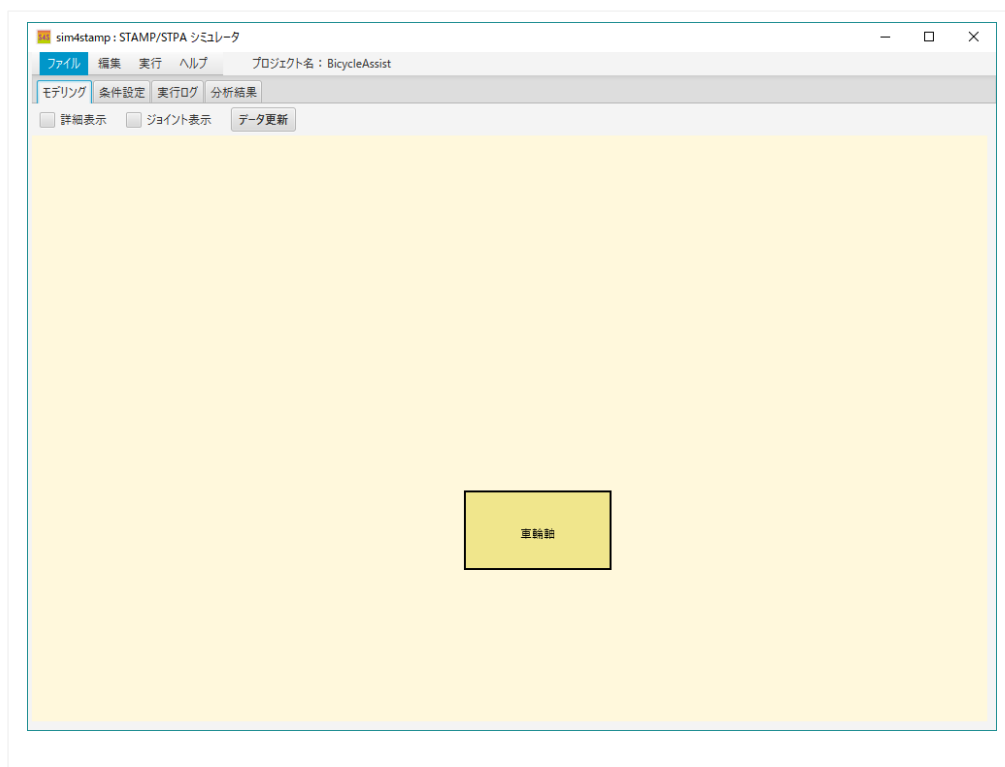
モデルの生成は、「モデリング」タブ内のモデルパネルで行います。



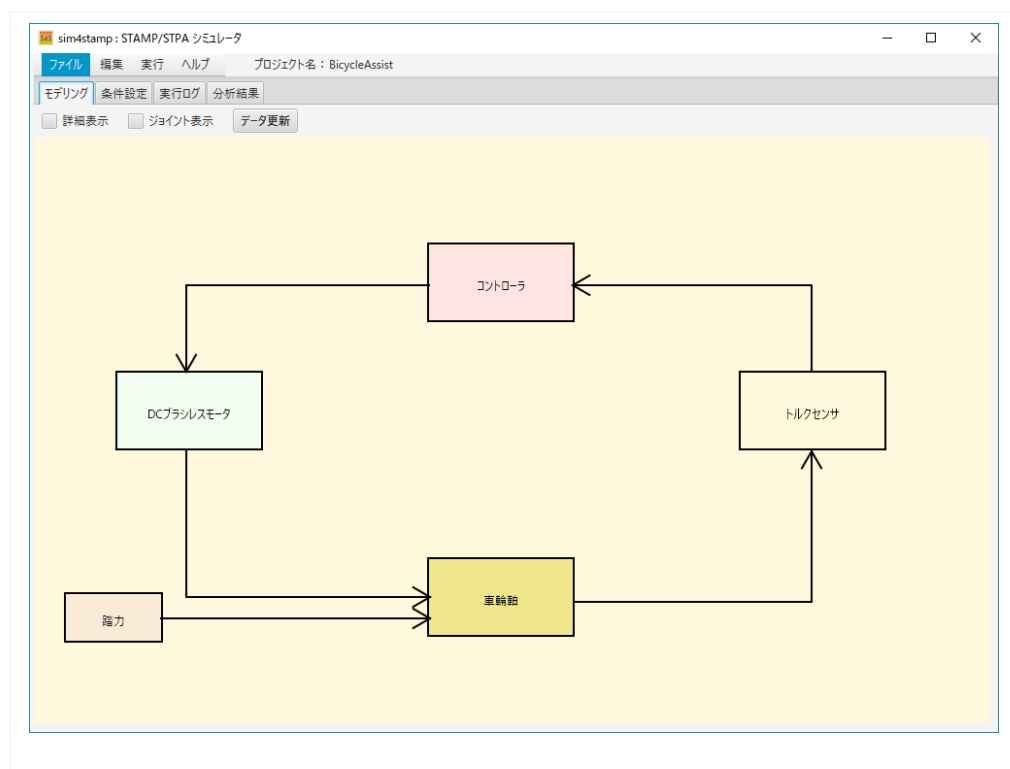
モデルパネル上で、右クリックを行うと追加する部品のリストが表示されます。

部品リストから目的の部品を選択することにより、追加部品のダイアログが表示され、必要事項を入力し、部品をモデルパネルに置きます。

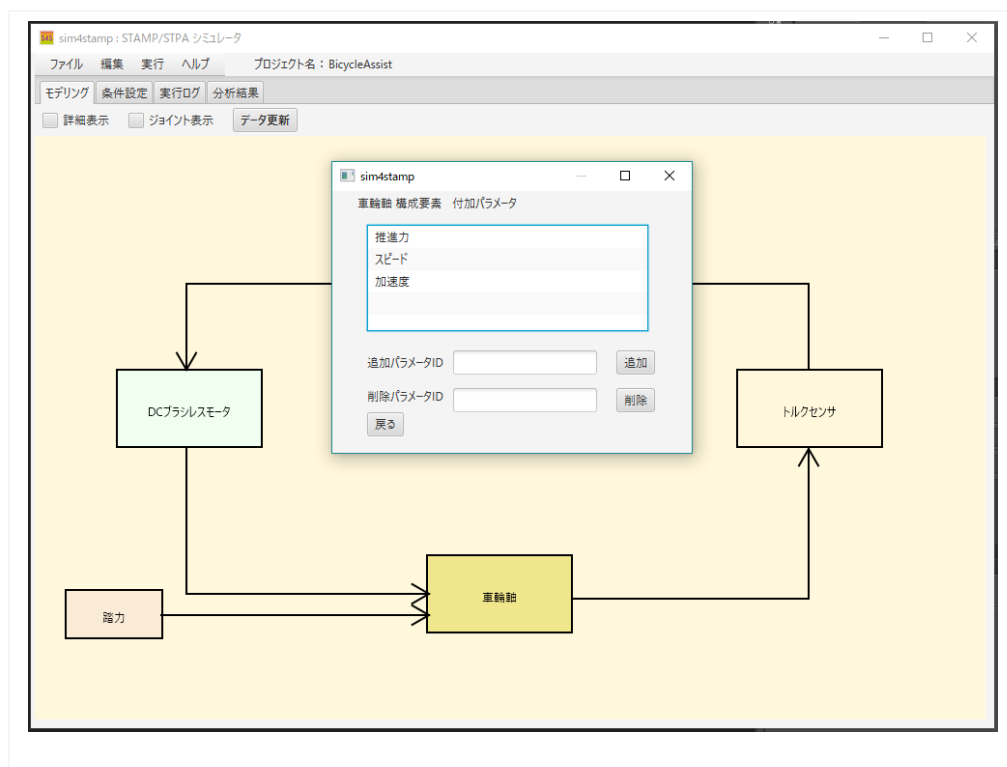




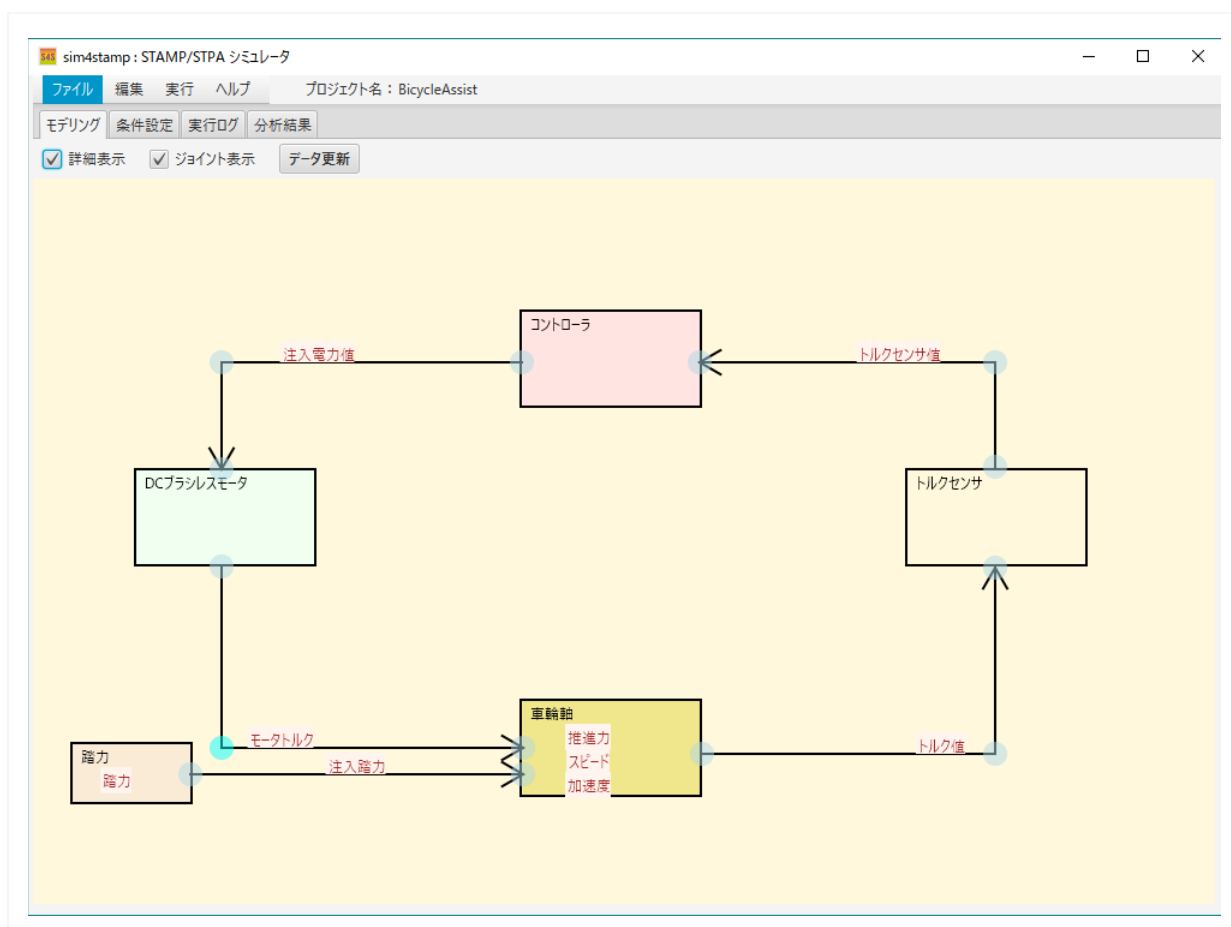
コネクタも設定し、部品要素間のデータの流れを設定します。コネクタは、両端が部品要素に接続されると色が黒に変化します。



部品要素とその接続関係の設定が完了後、パラメータを設定します。
 パラメータの設定は、部品要素であれば、部品要素の内部で右クリックを、コネクタであれば、折れ曲がる頂点上で右クリックします。



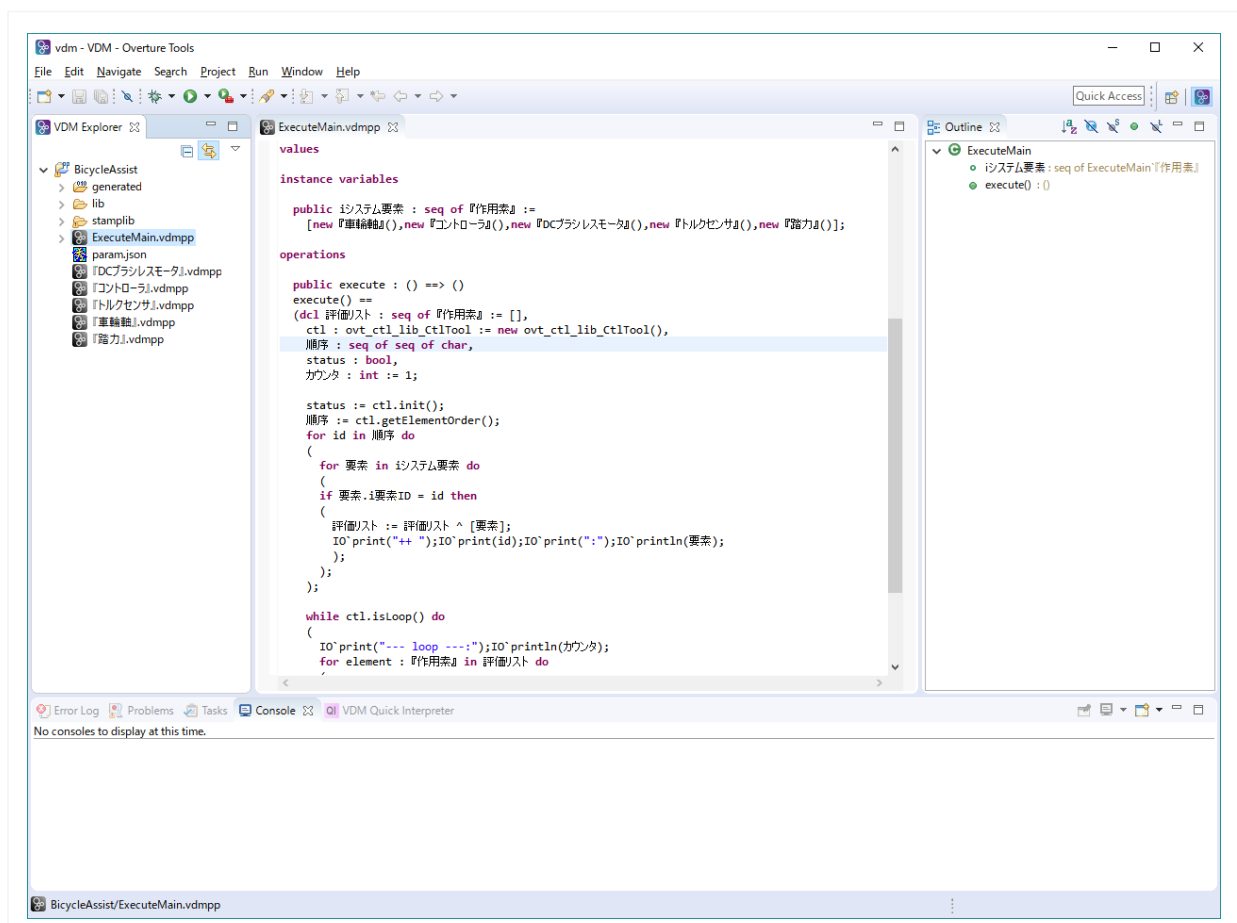
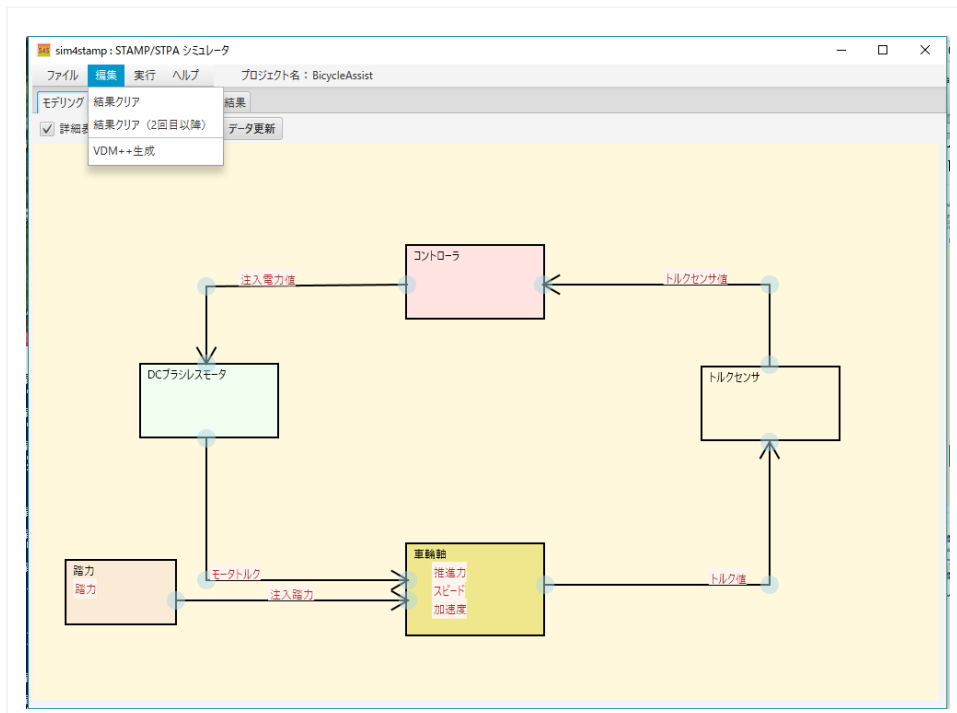
「詳細表示」をチェックすると設定されているパラメータが図上に表示されます。



なお、コネクタに設定できるパラメータは1つのみです。複数設定する必要がある場合は、複数コネクタを設定してください。

8. VDM++モデル生成

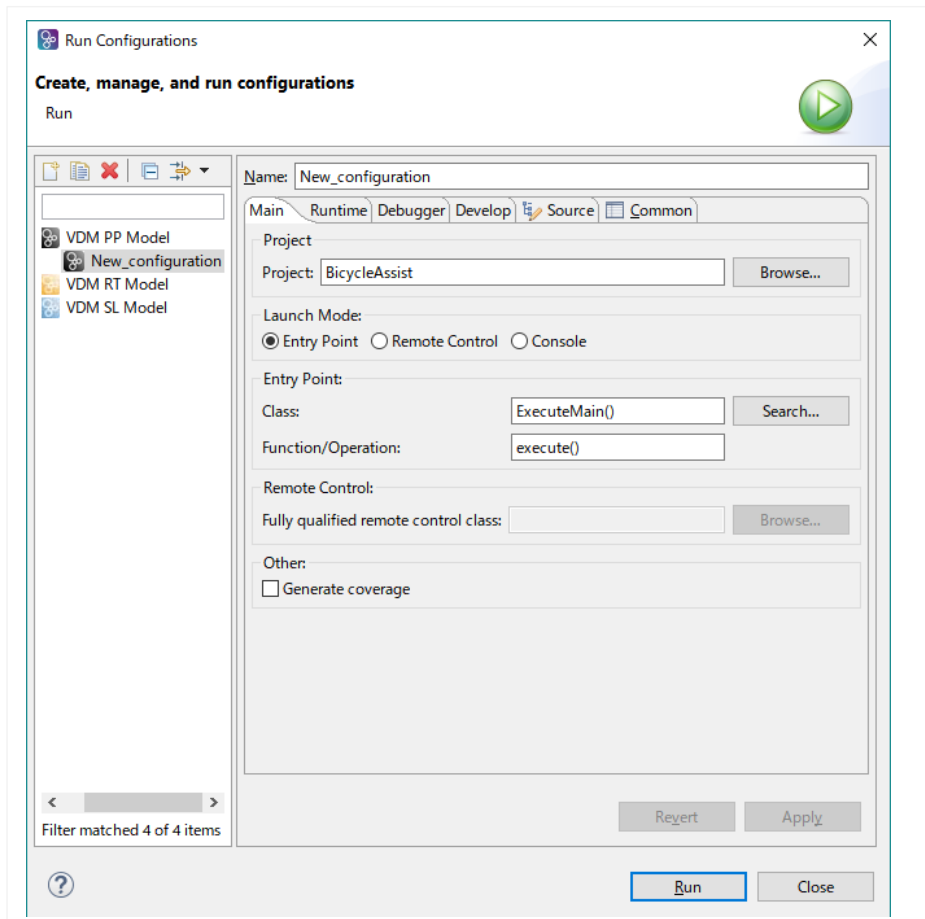
「データ更新」ボタンクリック後に、「編集」－「VDM++生成」を行うことにより、VDM++のテンプレートコードが自動生成されます。



生成されたテンプレートコードを元に、Overture 側で VDM++ のロジックを追加します。

なお、Sim4stamp 側との通信ライブラリも同時にプロジェクトフォルダにインストールされます。

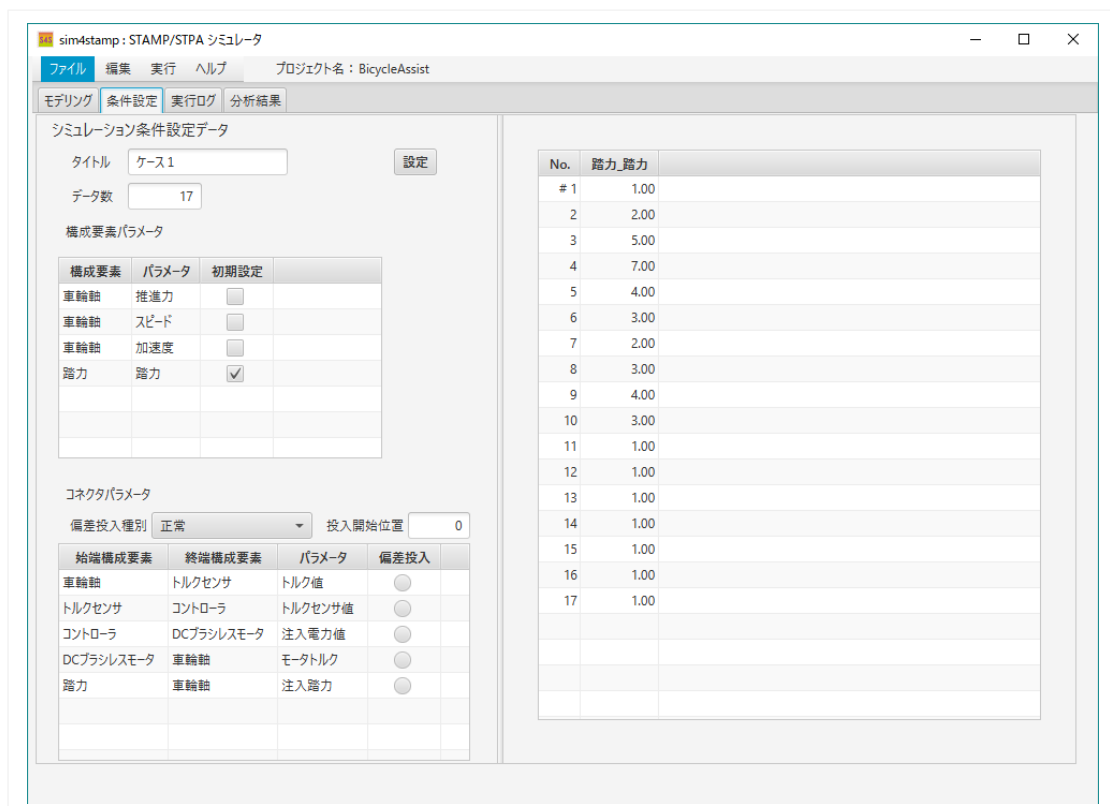
作成された VDM++ コードを実行する場合、Overture から実行するには、以下に示すように Entry Point を設定します。



なお、VDM++ の実行は Sim4stamp からコマンドラインツールを起動して行う方法でも可能ですが、VDM++ のロジックを作成する場合は、Overture 側で Debug 機能を用いて作業を行う方が効率的です。

9. データ生成

シミュレーション条件の設定は、「条件設定」タブで行います。



初期設定データ数はシミュレーションの各ステップで設定するデータ数です。デフォルトでは 10 個となっています。

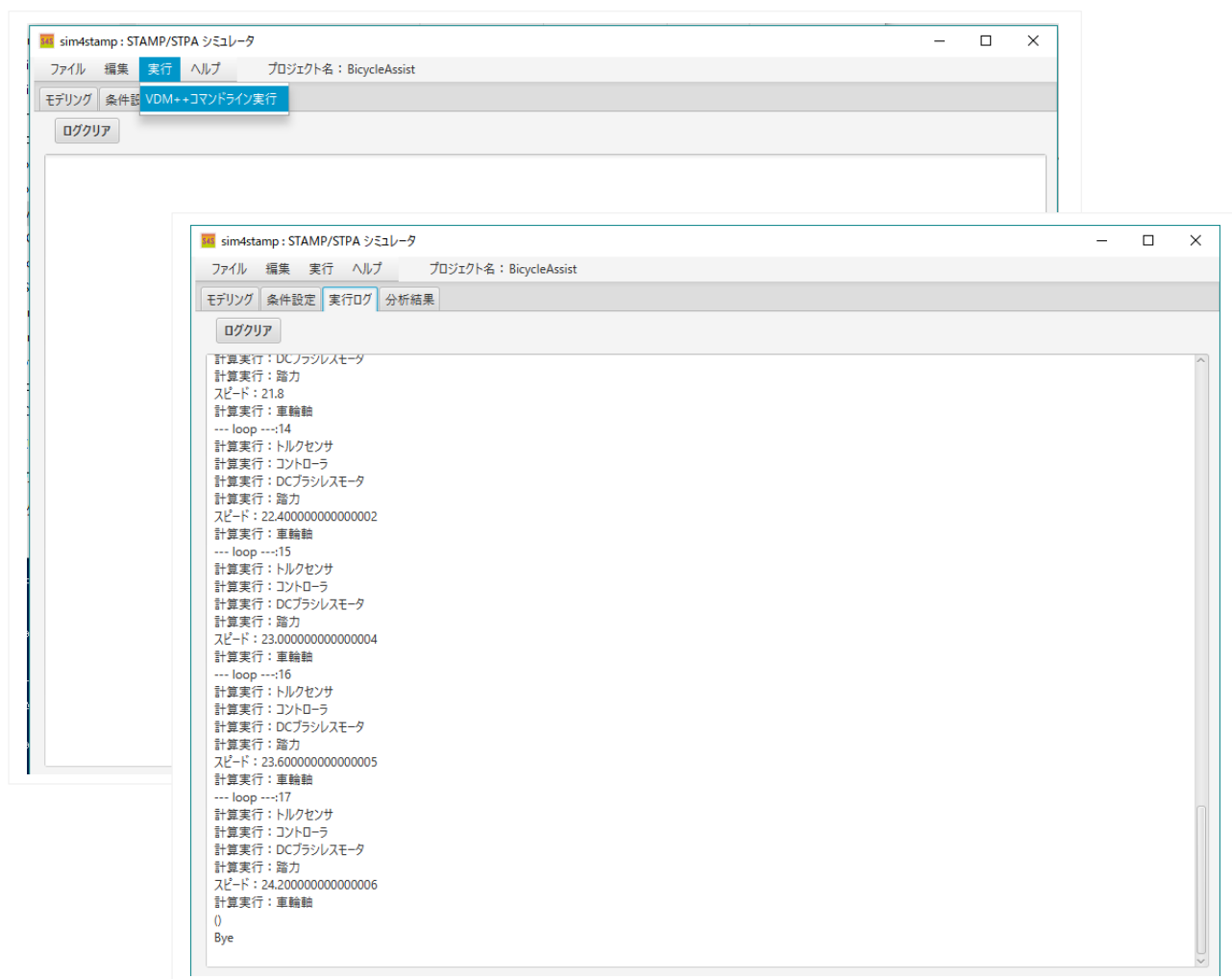
偏差投入はコネクタで行います。そのため、偏差投入を行うコネクタを一つのみ指定できるようになっています（ラジオボタン）。例えば「Providing More」を指定した場合、コネクタの始端値を終端値に渡す場合、定数倍されて渡されます。なお、この定数は「ファイル」－「偏差定数設定」で設定パネルを開き、変更が可能です。

また、偏差投入の開始位置を設定することが変更可能になっています。投入開始位置を設定し、「設定」ボタンをクリックして設定します。なお、偏差投入位置は右側の「No.」欄の番号に「#」がついている行で表しています。

偏差投入の設定位置の用途は、電動自転車の例で言うと、自転車が走りはじめ、低速になったときに、上り坂に差し掛かるケースで、坂に差し掛かる位置を偏差投入したい場合等に使用します。

10. VDM++の実行

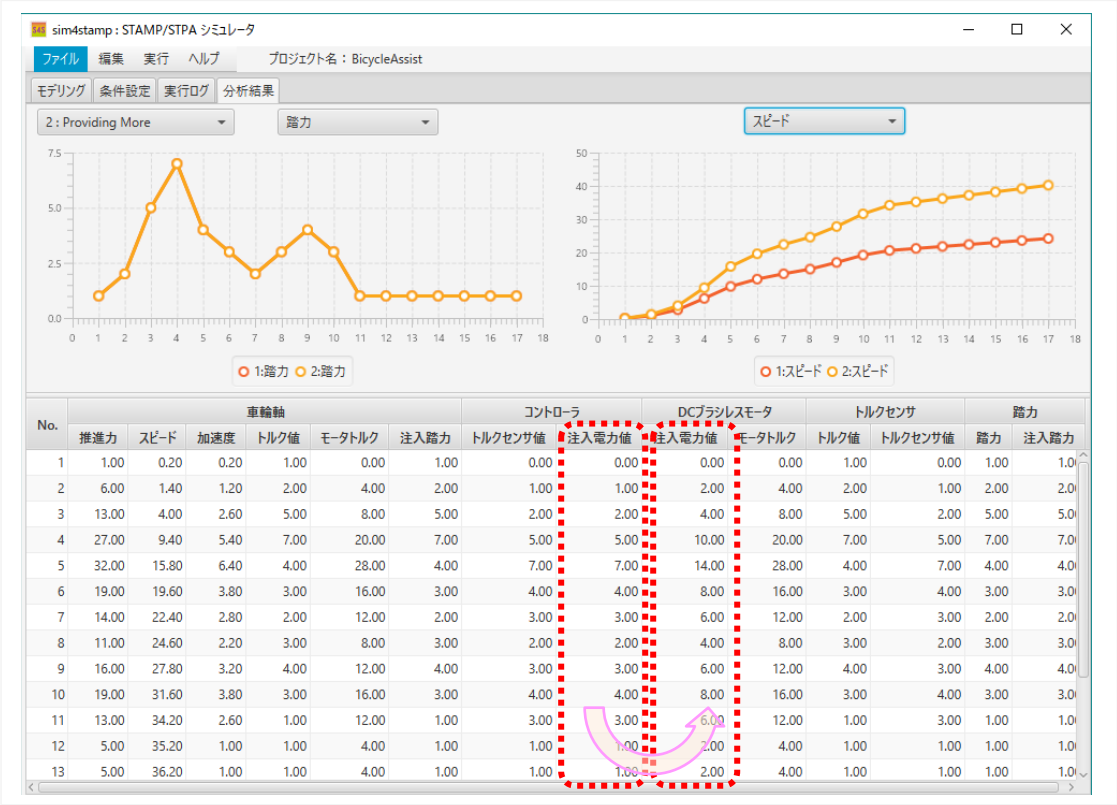
Sim4stamp 側からの VDM++の実行は、「実行」－「VDM++コマンドライン実行」をクリックします。



実行中には VDM++側で設定しているプリント文等の実行ログを表示します。

11. シミュレーション実行結果表示

シミュレーション結果の確認は、「分析結果」タブを開きます。

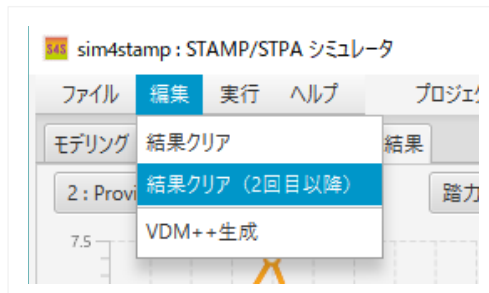


シミュレーション実行する毎に、グラフが追加されます。リセットは「編集」－「グラフ初期化」をクリック

クします。こうすることにより、偏差投入の種類と場所の効果を比較することが出来ます。

偏差投入は、コネクタで行われるので、偏差投入した要素間を接続するコネクタのデータは一致しない場所があります（上記の例ではコントローラの注入電力が DC ブラシレスモータ側では 2 倍になって伝達されています）。

なお、画面下側の数値編の表のデータは、画面左上のプルダウンでシミュレーションの任意の世代の値に変更することが出来ます。



グラフ及び結果の表のデータは、シミュレーションの世代を保存していて重ねまたは切り替え表示可能になっていますが、2 世代以降のみをクリアすることが出来ます（上図）。第一世代を偏差投入なしのデータとし、2 世代以降を偏差投入世代にし、繰り返しシミュレーション結果を表示し、比較することが可能になっています。

12. Q&A

<Q> Overture 側から実行するのと Sim4stamp 側で実行するのと違いはありますか？

<A> どちらで実行しても同じです。ただし、VDM++でログ等を出している場合は、Overture で実行すると Overture 側に、Sim4stamp で実行すると Sim4stamp の実行ログ画面にログが出ます。

<Q> 実行を一時的に中断し、再開するようなことは可能ですか？

<A> Overture 側のデバック機能を用いると可能になります。ただし、その場合は、Overture 側から実行する必要があります。