



## SysML Simulation Feature Guide

---

*by SparxSystems Japan*

Enterprise Architect 日本語版

SysML パラメトリック図のシミュレーション 機能ガイド

(2023/06/06 最終更新)



## 目次

1	はじめに.....	3
2	利用条件.....	4
3	サンプルについて .....	6
3.1	OpenModelica を利用する場合 .....	6
3.2	SysPhS を利用する場合(Simulink を利用する場合).....	8
4	OpenModelica の場合のモデルの作成.....	11
5	SysPhS の場合のモデルの作成 .....	25
6	Simulink ファイルの生成についての補足.....	31

## 1 はじめに

このドキュメントでは、Enterprise Architect ユニファイド版以上で利用可能な、OpenModelica や MATLAB Simulink/Stateflow(このドキュメントでは、まとめて Simulink と表現します)と連携する SysML のシミュレーション機能について、最低限の構成を構築するための方法を説明します。

なお、バージョン 15.2 以降の Enterprise Architect を利用した SysML モデルのシミュレーションには、大きく分けると以下の 3 つの種類があります。

1. バージョン 15.1 以前から利用可能であった、Enterprise Architect 独自のルールに基づいて OpenModelica で実行可能なコードを生成し、シミュレーションを実行する
2. OMG が定義する仕様「SysML Extension for Physical Interaction and Signal Flow Simulation」(SysPhS) に基づいて作成された SysML モデルに対して、OpenModelica や Simulink でシグナルフローのシミュレーションを実行する
3. SysPhS に基づいて作成された SysML モデルに対して、OpenModelica や MATLAB/Simscape で物理シミュレーションを実行する

上記の内容のうち、このドキュメントでは 1 および 2 について説明します。3 については 2 と同様のモデルを作成しますが、このドキュメントでは触れません。SysPhS については、OMG の Web サイトをご覧ください。

このドキュメントは、Enterprise Architect 16.0 ビルド 1602 を利用して記述しています。過去のバージョン・ビルドについては利用できない機能や操作が異なる点がありますのでご注意ください。また、MATLAB/Simulink は、R2021b を利用しています。R2018 以降が必要です。

実際にこの機能を活用するためには、OpenModelica や Simulink に関する詳細な知識が必要です。例えば、OpenModelica が提供するライブラリの関数を利用する必要がある場面があったり、要素などの名前付けには OpenModelica で定義される予約語は利用できなかったりするなど、知識がない場合にはさまざまな問題が発生することが考えられます。OpenModelica そのものについては標準サポートの範囲外となり、サポート窓口では対応できない点につきましてあらかじめご了承ください。

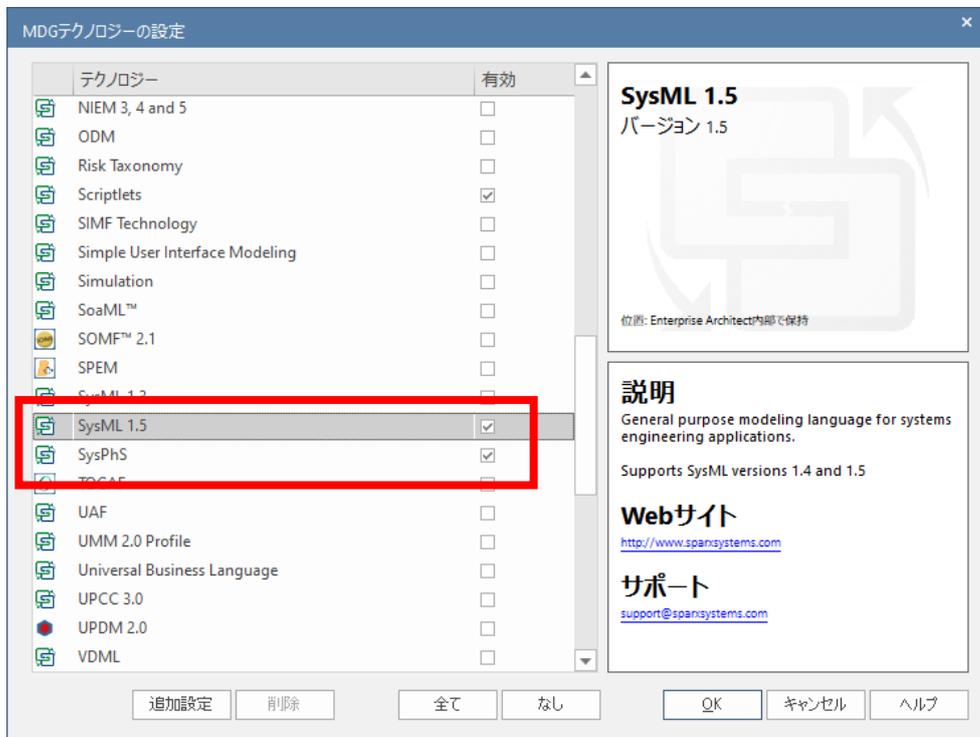
## 2 利用条件

OpenModelica や Simulink と連携する SysML のシミュレーション機能を利用するためには、以下のいずれかのエディションが必要です。Simulink との連携には、Enterprise Architect バージョン 15.2 以降が必要です。

- Enterprise Architect ユニファイド版
- Enterprise Architect アルティメット版

なお、このドキュメントでの説明用および動作確認用には、OpenModelica 1.14.1 を利用しています。OpenModelica は別途インストールしておいてください。

このドキュメントで紹介するサンプルを利用するためには、SysML 1.5 のみが有効である必要があります。Enterprise Architect を起動後、「アドイン・設定」リボン内の「MDGテクノロジー」パネルにある「設定」を実行して「MDGテクノロジーの設定」画面を表示し、「SysML 1.5」「SysPhS」にチェックが入り、「SysML 1.3」のチェックが外れていることを確認してください。変更した場合には、一度 Enterprise Architect を再起動して下さい。



また、パースペクティブを SysML に設定しておくともモデルの作成が効率的になります。

Enterprise Architect の画面右上にある「パースペクティブ」と書かれたボタンをクリックし、「システムズエンジニアリング」→「SysML」を選択してください。

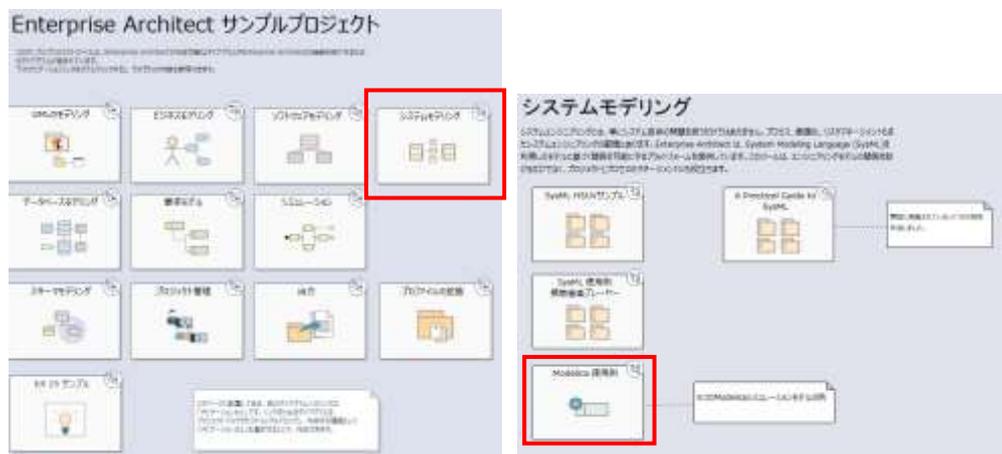
### 3 サンプルについて

OpenModelica や Simulink との連携につきましては、サンプルを利用して動作させることで、動作環境の確認と概要の把握が容易にできます。

#### 3.1 OpenModelica を利用する場合

初めて OpenModelica でのシミュレーションを利用する場合の手順は、以下の通りです。まずは、サンプルを利用してシミュレーションが動作する環境を構築してください。

1. OpenModelica をダウンロードし、インストールします。詳細についてはこのドキュメントの範囲外となります。
2. Enterprise Architect を起動し、「ホーム」リボン内の「ヘルプ」パネルにある「ヘルプ」ボタンを押し、「サンプルプロジェクトを開く」を実行します。
3. 表示される図から、「システムモデリング」→「Modelica 使用例」へとリンクをダブルクリックして移動します。

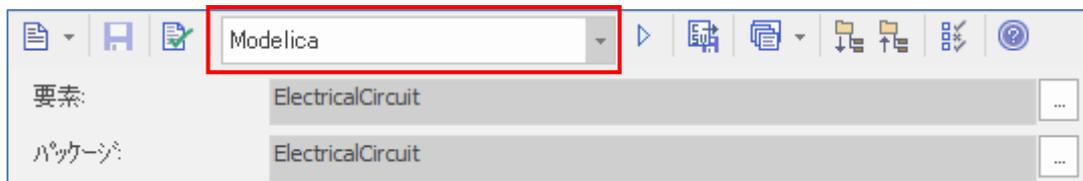


4. 「Modelica のサンプル」の図にある 6 組の SysMLSimConfiguration 要素とパッケージについては、以下のような動作になります。
  - SysMLSimConfiguration 要素をダブルクリックすると「SysML シミュレーションの設定」タブが開きます。
  - パッケージ要素をダブルクリックすると、メインとなる図が開きます。
  - パッケージ要素を右クリックし「検索」→「モデルブラウザ内の位置を表示」を実行すると、モデルブラウザ内のサンプルデータの位置がわかります。サンプルを構成する要素を把握する場合に便利です。

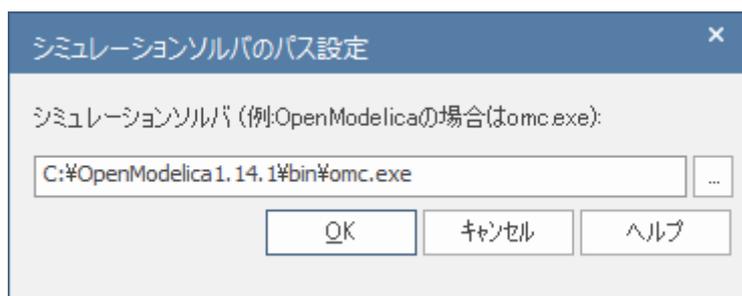
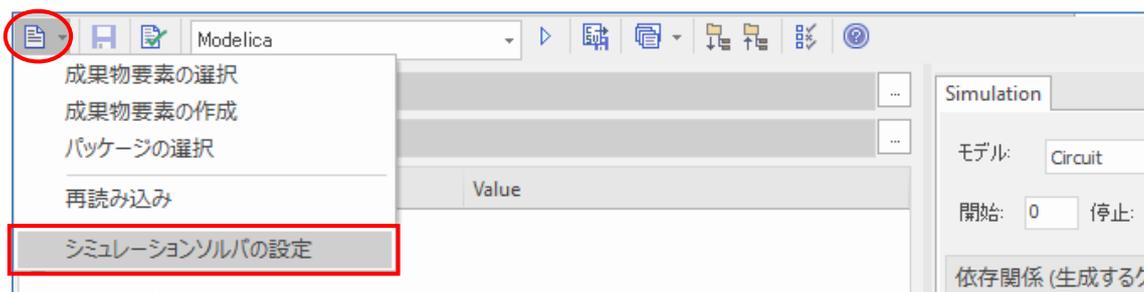
- 今回は、「ElectricalCircuit」の SysMLSimConfiguration 要素をダブルクリックしてください。



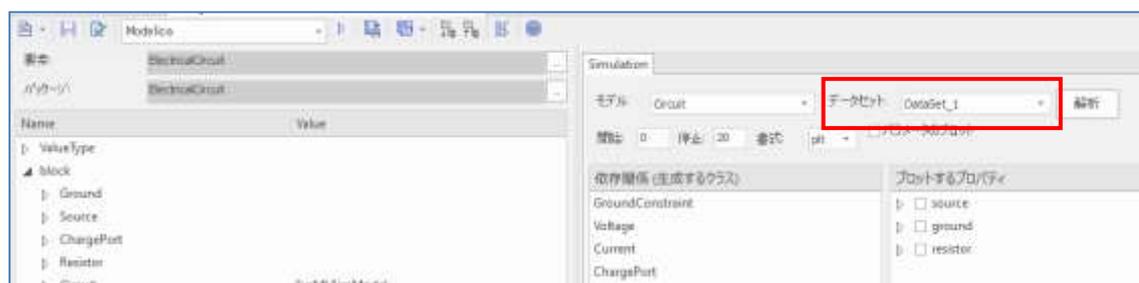
- シミュレーション画面を開いたら、OpenModelica の実行ファイルの指定が必要です。まず、「SysML シミュレーションの設定」タブ内のツールバーにあるシミュレーションの実行対象として、「Modelica」を選択してください。



その後、ツールバーの左端のボタンから「シミュレーションソルバの指定」を実行し、omc.exe ファイルの位置を指定してください。

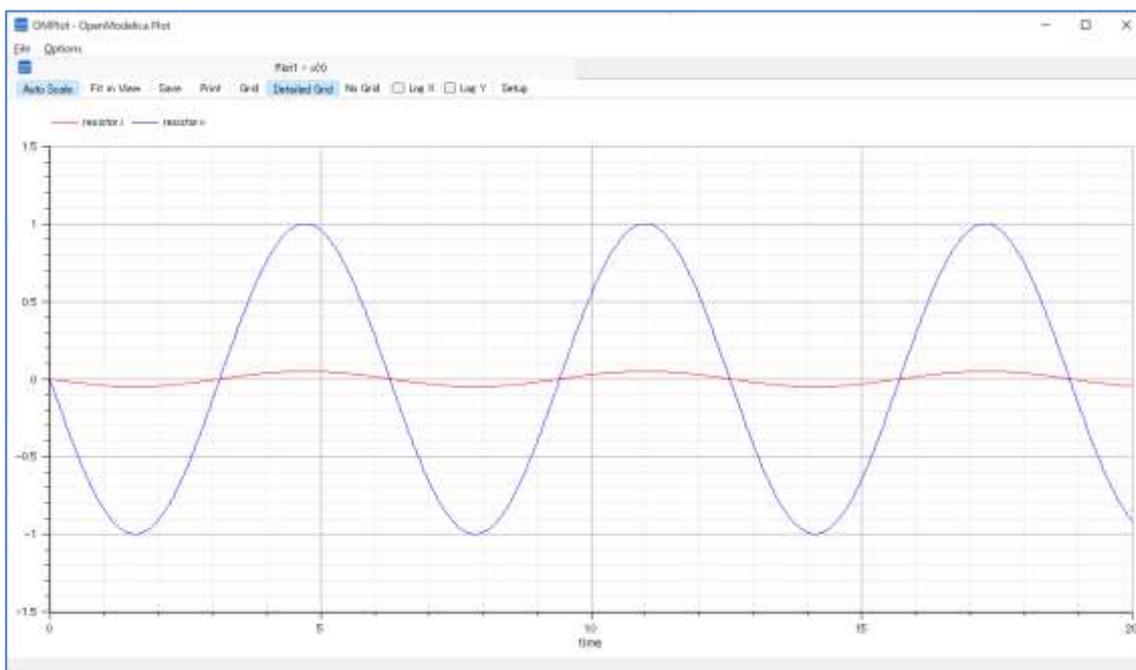


- タブ内、右側の「Simulation」タブにある「データセット」(シミュレーションで利用するパラメータの設定)として、「DataSet\_1」を指定してください。



8. タブ内、右側の「Simulation」タブにある「解析」ボタンを押してください。ここまでの設定が正しければ、自動的に OpenModelica 用のスクリプトを生成し、実行・結果表示まで進みます。

この結果、出力サブウィンドウが開いて OpenModelica 用のスクリプトが自動生成されて実行します。自動的にグラフが表示されれば、正しく動作しています。



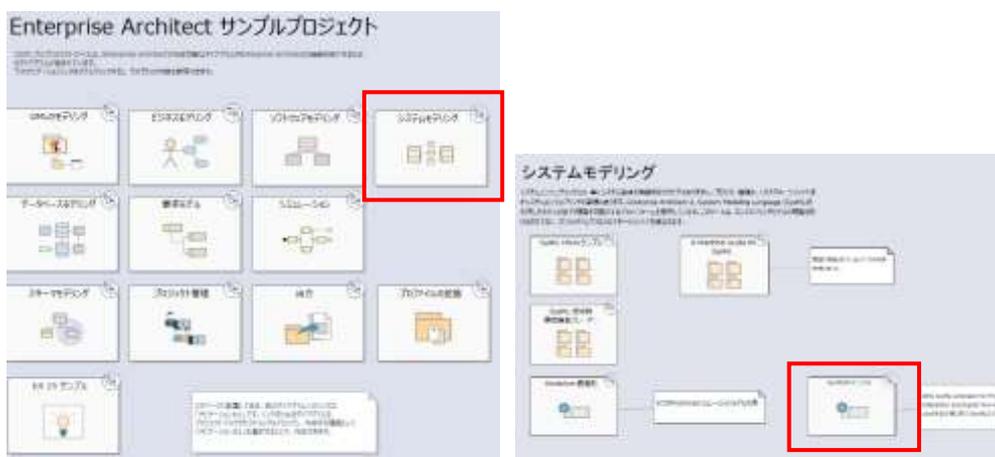
Enterprise Architect のサンプルプロジェクトには、これ以外にもいくつかのサンプルが含まれています。これらのサンプルは、実際にモデルを作成する際のリファレンスとしても役立ちます。

### 3.2 SysPhSを利用する場合(Simulinkを利用する場合)

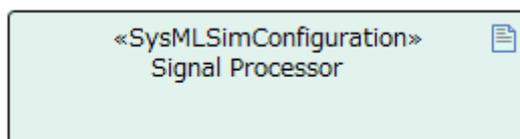
Simulink の場合など SysPhS も、基本的には上記の OpenModelica の場合と同様です。な

お、この方法は SysPhS 仕様に対応したモデルを作成し、シミュレーションを実行します。この 3.2 章で説明する方法は、OpenModelica を利用する場合でもシミュレーションを実行できます。以下の説明は、Simulink を利用する前提の説明です。

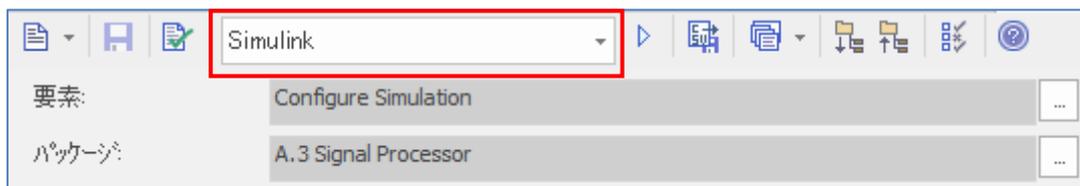
1. MATLAB および Simulink をインストールし、利用可能な状態とします。
2. Enterprise Architect を起動し、「ホーム」リボン内の「ヘルプ」パネルにある「ヘルプ」ボタンを押し、「サンプルプロジェクトを開く」を実行します。
3. 表示される図から、「システムモデリング」→「SysPhS のサンプル」へとリンクをダブルクリックして移動します。



4. 「SysPhS のサンプル」の図は、SysPhS の仕様書に記載されている「SignalProcessor」を Enterprise Architect で作成しています。
5. SysMLSimConfiguration 要素をダブルクリックして SysML シミュレーションの設定タブを開きます。



6. 「SysML シミュレーションの設定」タブ内のツールバーにあるシミュレーションの実行対象として、「Simulink」を選択してください。

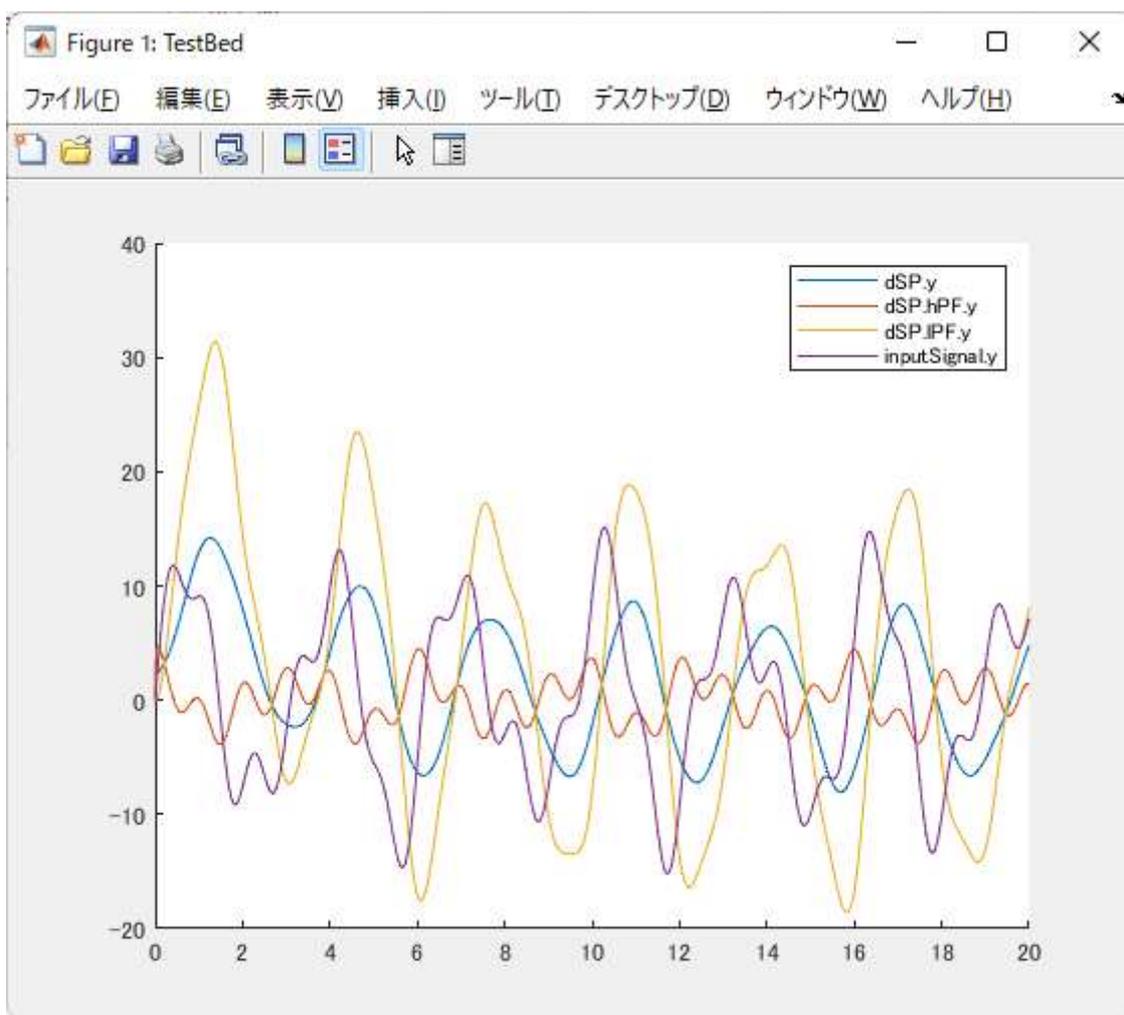


7. タブ内、右側の「Simulation」タブにある「解析」ボタンを押してください。ここまでの設定が正しければ、自動的に Simulink 用のスクリプトを生成し、実行・結果

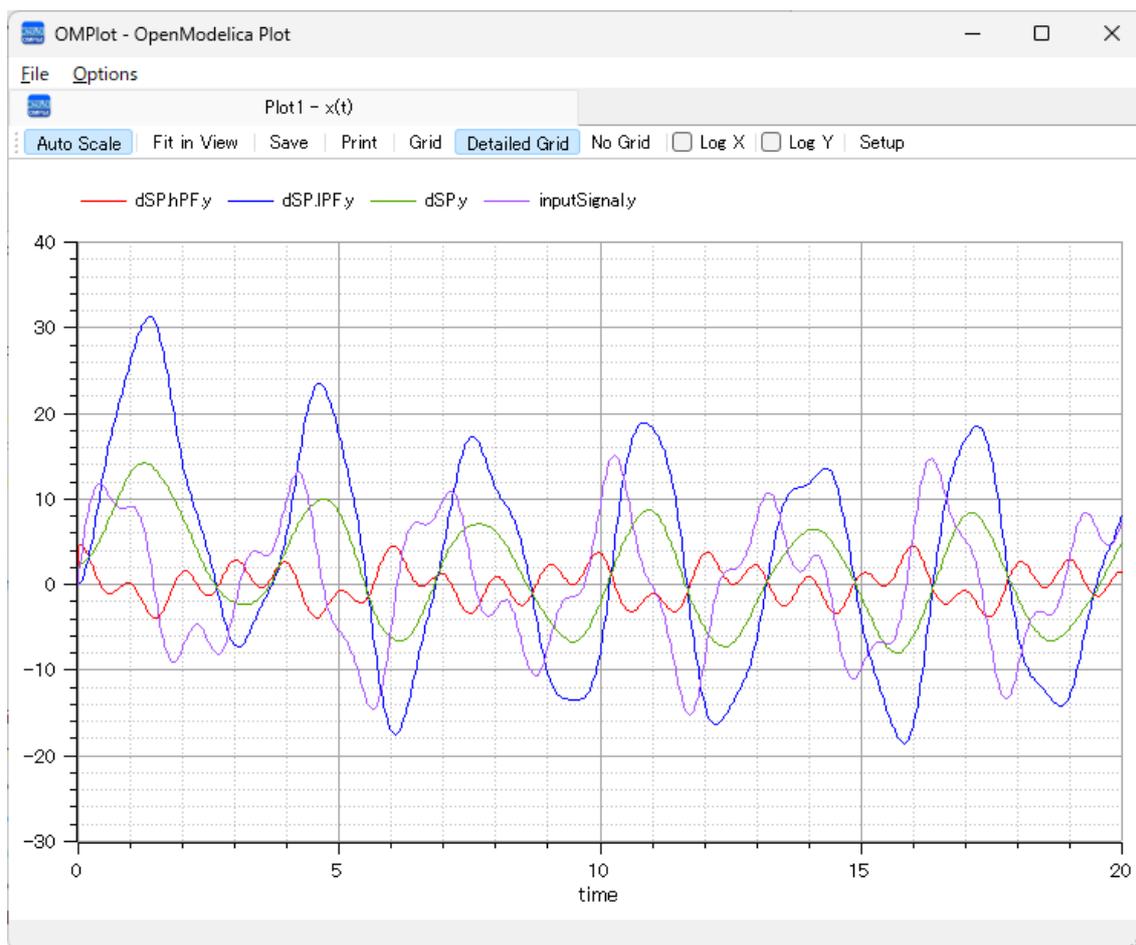
表示まで進みます。

この結果、出力サブウィンドウが開いて **Simulink** 用のファイルが自動生成されて実行します。自動的に以下のようなグラフが表示されれば、正しく動作しています。

(MATLAB および Simulink を Enterprise Architect から起動し実行するまで、マシンの性能により時間がかかります。動作完了までお待ちください。何か問題がある場合には、出力サブウィンドウに問題となる内容が出力されます。)



このモデルを、OpenModelica で実行した場合の結果は以下の通りです。同じモデルから異なるツールを使い、同じ結果になっていることが確認できるかと思います。



## 4 OpenModelicaの場合のモデルの作成

ここからは、新規のプロジェクトからシミュレーションできるまでのモデルを作成するまでの流れを具体的に紹介します。この章では、OpenModelica を利用し、シミュレーションを実行する場合について説明します。Enterprise Architect 自身の基本的な操作方法についてはこのドキュメントでは述べませんので、次の URL からダウンロードできる PDF ドキュメントをご覧ください。「Enterprise Architect 入門セミナー」の受講もお勧めします。

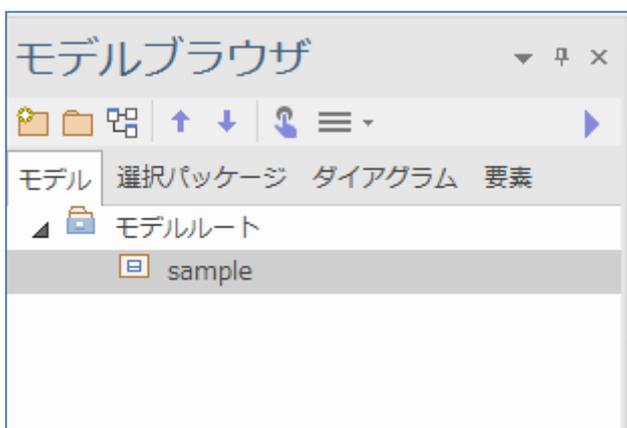
Enterprise Architect ドキュメント ライブラリ

[https://www.sparxsystems.jp/products/EA/ea\\_documents.htm](https://www.sparxsystems.jp/products/EA/ea_documents.htm)

SysPhS (Simulink)を利用する場合には、第5章をご覧ください。

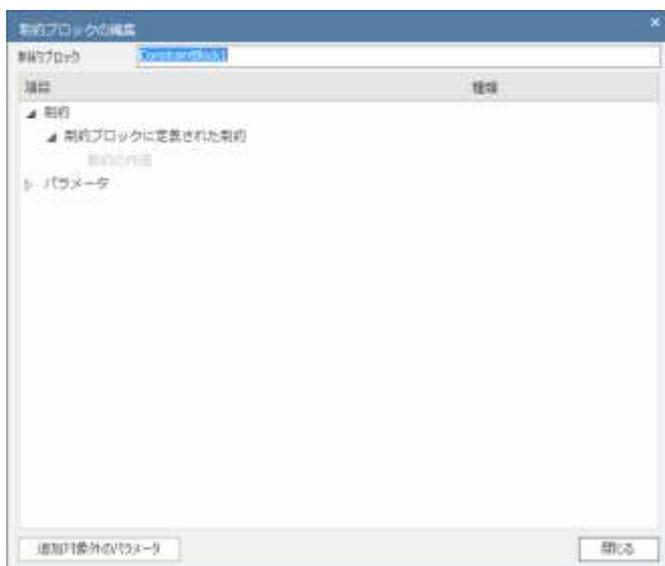
まず、Enterprise Architect の新しいプロジェクトファイルを作成してください。次の図の

ように、パッケージを1つ作成してください。



このパッケージ内に、SysML のブロック図を作成してください。そして、ブロック図に制約ブロック要素をツールボックスからドラッグ&ドロップし、図に配置します。自動的に、次のような画面が表示されます。

(もし表示されない場合には、図に配置した制約ブロック要素を右クリックして「制約ブロックの編集」を実行してください。)

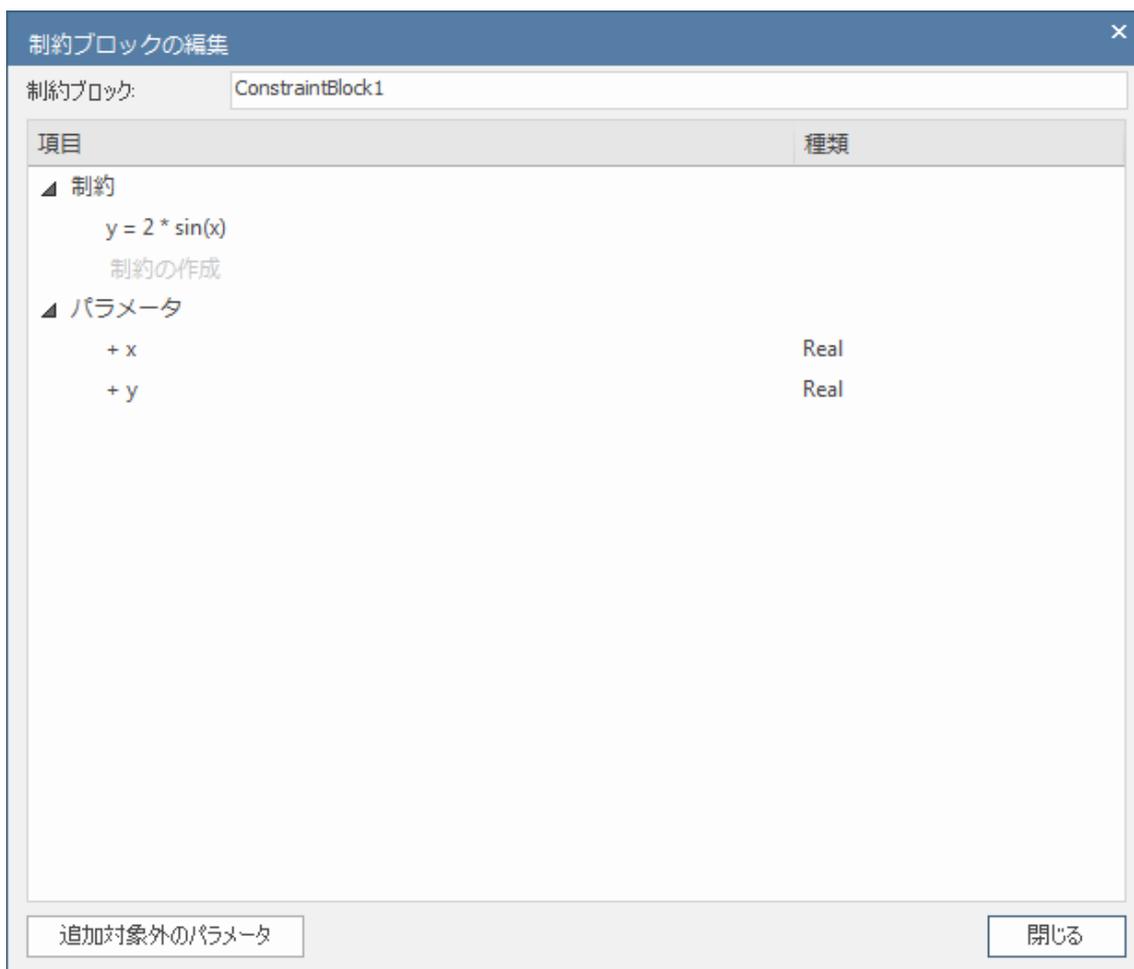


シミュレーションのためには、シミュレーション対象のブロック要素に、制約とプロパティが必要です。この画面を利用すると、必要となる制約ブロック要素を簡単に作成できます。

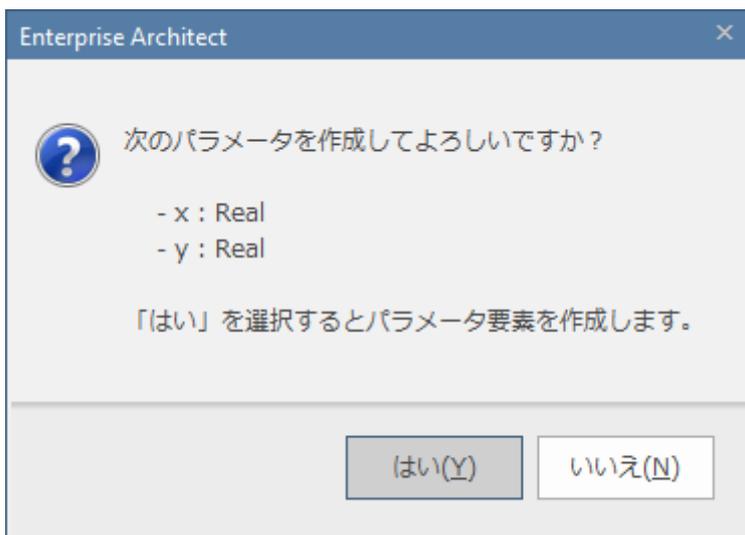
上記の画面内の「制約の作成」と薄く表示された部分をダブルクリックすることで、制約

を入力できます。今回はサンプルとして、入力値を **sin** 関数に入れ、得られる値の **2** 倍を返すというモデルを作成することにします。そこで、制約として「**y = 2 \* sin(x)**」と入力してください。入力してリターンキーを押すと、入力内容を解析してパラメータ **x,y** を抽出し、以下のような表示に変わります。

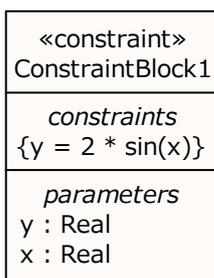
(sin 関数は OpenModelica が提供する関数です。)



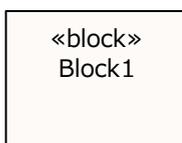
画面右下の「閉じる」ボタンを押すと、以下のようにプロパティを作成するかどうかの確認のメッセージが表示されますので、「はい」を選択してください。



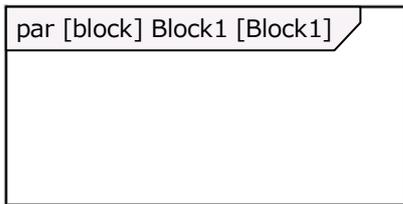
その結果、制約ブロック要素は以下ようになります。



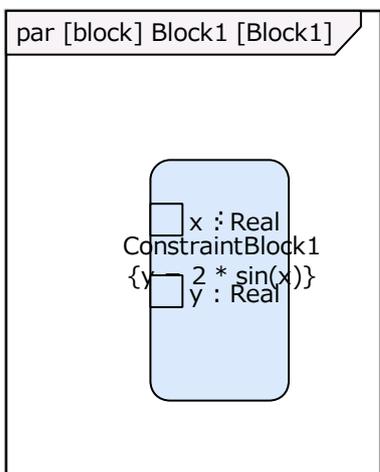
次に、この制約ブロック要素を利用するブロック要素を追加します。ツールボックスからブロック要素をドラッグ&ドロップで図に配置してください。



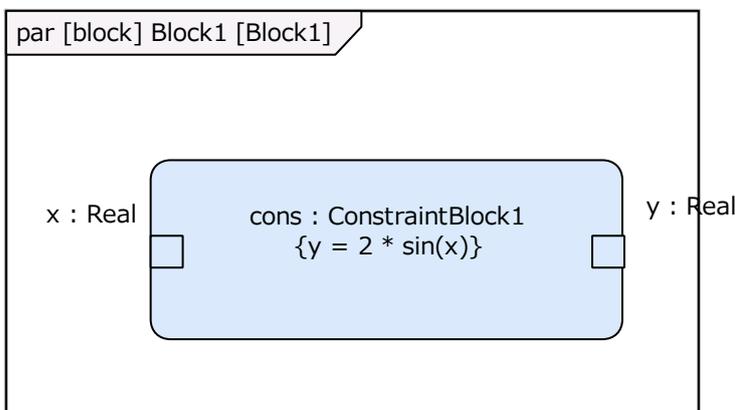
このブロック要素と制約ブロック要素を結びつけるために、ブロック要素を右クリックして「子ダイアグラムの追加」→「パラメトリック図」を選択し、子ダイアグラムを作成してください。



まず、先ほど作成した制約ブロック要素をこのパラメトリック図に配置します。モデルブラウザ内の「ConstraintBlock1」要素をドラッグし、図にドロップしてください。自動的に、制約プロパティ要素として配置されます。

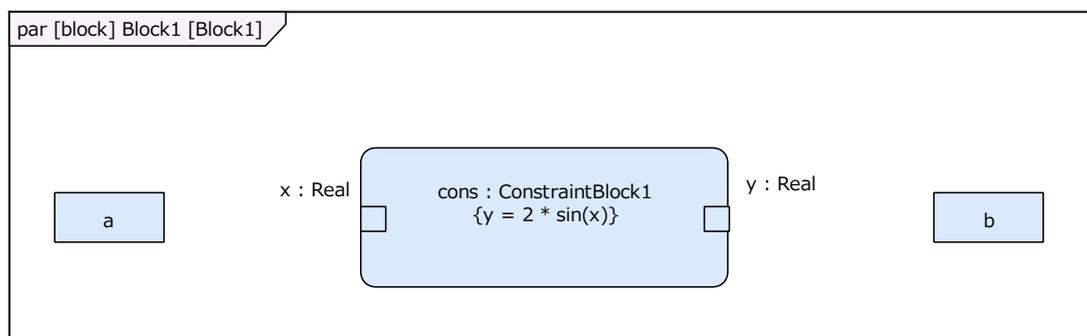


配置した直後は、上の例のように要素の配置が適切とは限りませんので、要素の大きさや位置を調整して下さい。また、制約プロパティ要素をダブルクリックするなどの方法で、名前を設定して下さい。下の図では、名前を「cons」としてあります。

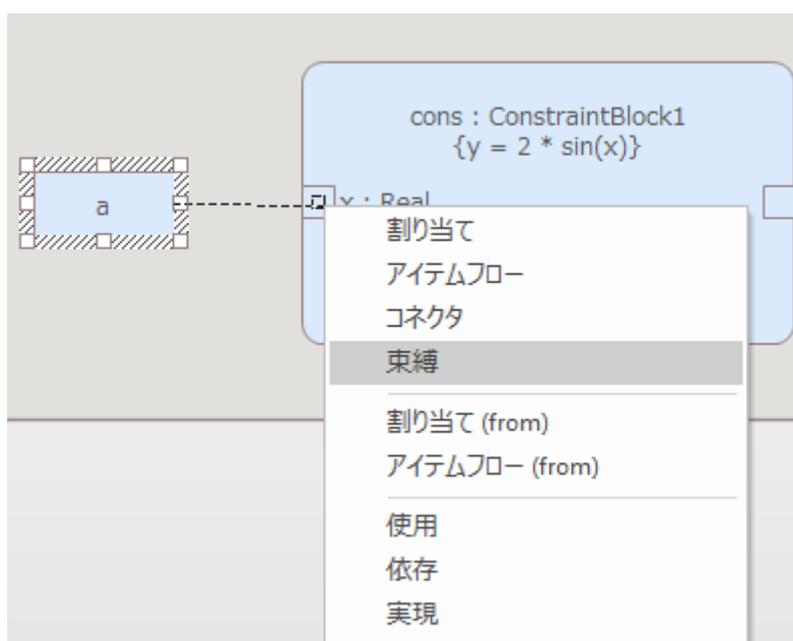


次に、この制約とブロック側のプロパティを結びつけます。しかし、まだブロック要素に

はプロパティを定義していません。そこで、ツールボックスからプロパティ要素を 2 つドラッグ&ドロップで配置してください。名前は、ここでは a と b としてあります。a が入力、b が出力という想定です。

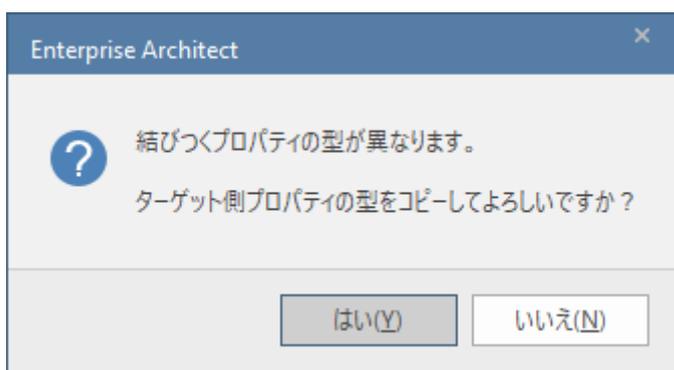


そして、制約ブロックで定義したプロパティと、今回追加したプロパティを「束縛」します。例えば、プロパティ a からクイックリンクで x と結びつけると、「束縛コネクタ」を作成することができます。

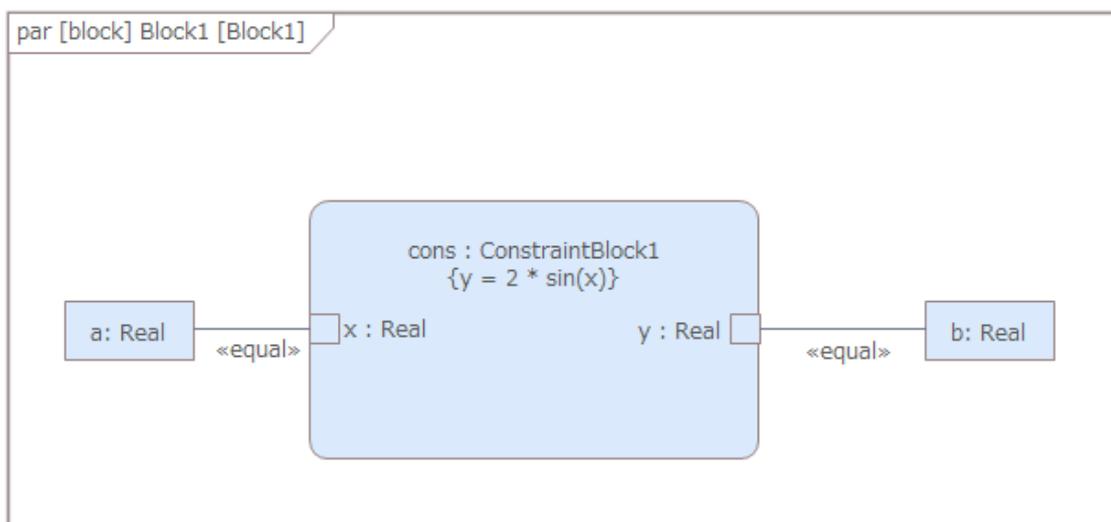


結びつけた場合に、以下のようなメッセージが表示された場合には、「はい」を選択することでプロパティの型が自動的に「Real」に設定されます。

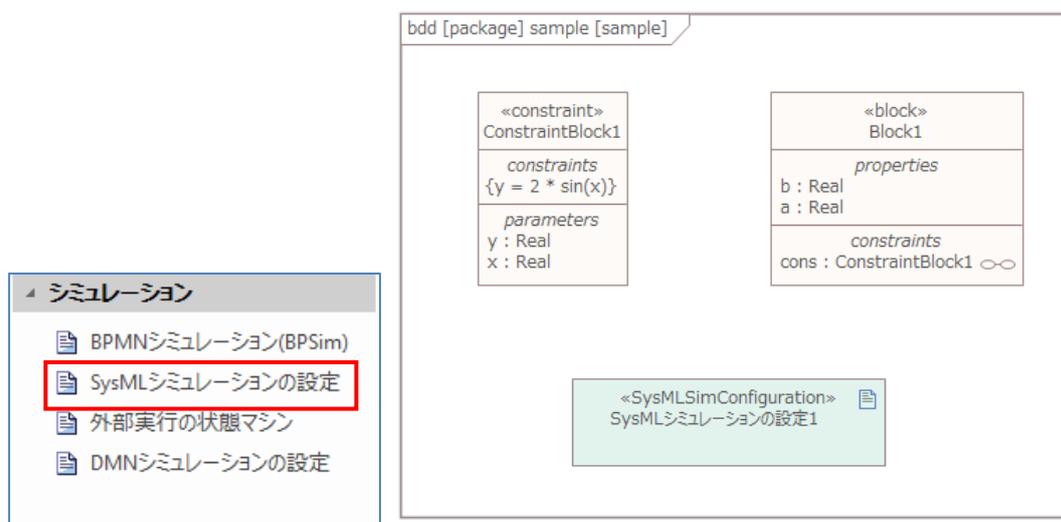
(型が定義済みの要素(今回は制約プロパティ)側から束縛する場合で相手に型指定がない場合は、メッセージを表示せずに自動的に型を一致させます。)



同様に、**b** と **y** も結びつけます。結びつける方向は、**b** から **y** でも、**y** から **b** でも構いません。



これでモデルの作成は完了です。次に、シミュレーションを実行するために必要な「SysMLシミュレーションの設定」要素(SysMLSimConfiguration)を作成します。この要素は、ツールボックスの「シミュレーション」グループ(バージョン 15.2 以前では「拡張要素」グループ)にあります。ドラッグ&ドロップで、ブロック定義図に配置してください。



作成した SysML シミュレーションの設定要素をダブルクリックし、「SysML シミュレーションの設定」タブを開いてください。最初に、対象のパッケージを指定する必要がありますので、次の画像の赤枠部分にある参照(...)ボタンを押し、図や要素を含むパッケージを指定してください。



パッケージを指定すると、パッケージ内のシミュレーションに関する要素が階層的に表示されます。下の画像は、すべての内容を展開した状態です。

(束縛の接続 (BindingConnector)の内容は、向きは問いません。a と cons.x、b と cons.y が逆方向でも問題ありません。)

Name	Value
▲ block	
▲ Block1	
▲ Part	
a : Real	
b : Real	
▲ constraintProperty	
cons : ConstraintBlock1	
▲ BindingConnector	
a --> cons.x	
cons.y --> b	
▲ constraintBlock	
▲ ConstraintBlock1	
▲ Part	
x : Real	
y : Real	
▲ Constraint	
y = 2 * sin(x)	

次に、定義したブロックのどの要素をシミュレーション対象にするかを具体的に指定する必要があります。「Block1」要素の「Value」欄のセルをクリックすると以下のように選択肢が表示されますので「SysMLSimModel」を選択してください。選択すると、右側の領域の「依存関係」の一覧に、「Block1」要素と、その要素の内部で利用している「ConstraintBlock1」要素の両方が表示されます。また、「プロットするプロパティ」の欄には、対象として指定した「Block1」要素が持つプロパティ要素の名前が表示されます。

Name	Value
▲ block	
▲ Block1	
▲ Part	
a : Real	SysMLSimClass
b : Real	SysMLSimModel
▲ constraintProperty	
cons : ConstraintBlock1	

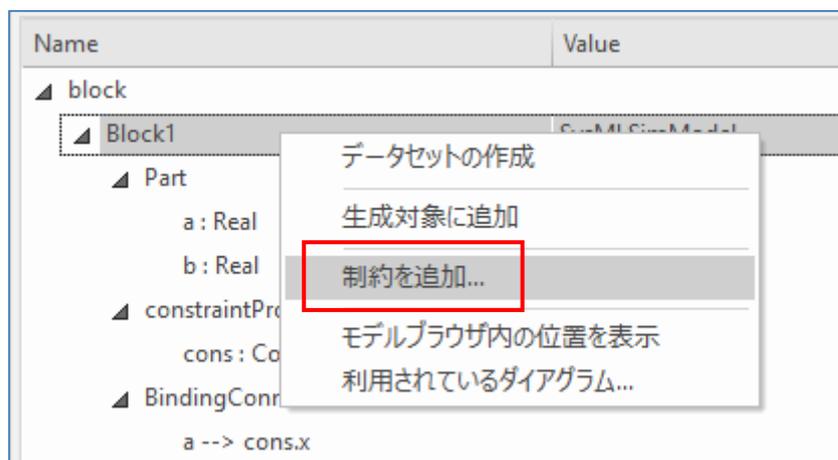


今回は、**a** を入力とし、**b** を出力とするので、「プロットするプロパティ」の一覧の **b** にチェックを入れてください。

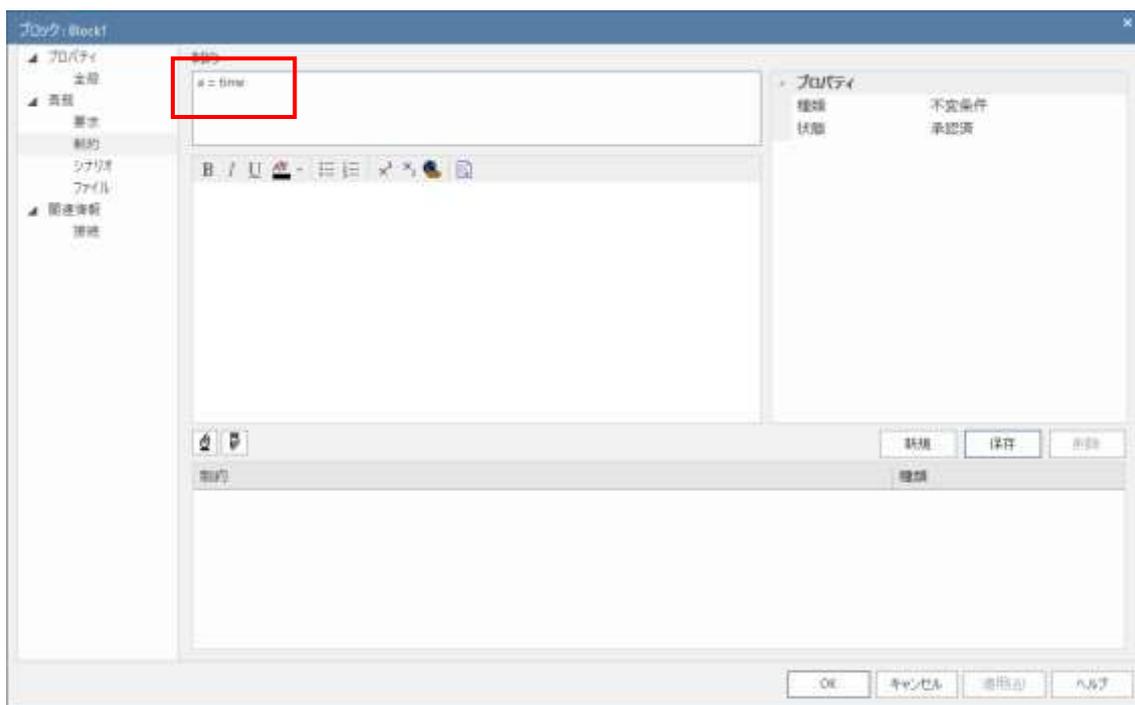
ここまでではほぼ準備が完了ですが、もう 1 つ制約が必要です。この時点では、**a** の値の入力をどうするのが決まっていませんので、「解析」ボタンを押して実行しても **a** の値が決まらないというエラーとなります。

(エラーの出力は **omc.exe** が出力する内容そのままです。シミュレーションがうまく実行できない場合のヒントを得るためには、**omc.exe** に関する情報を探してください。)

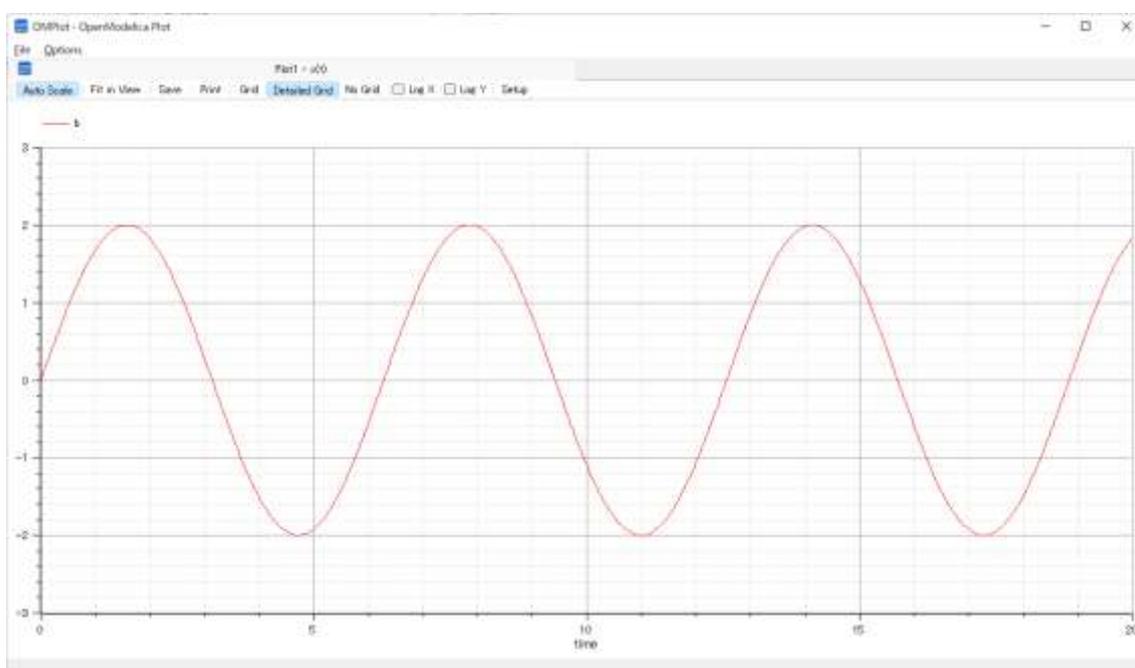
今回は、この情報をブロック **Block1** の制約として定義します。一覧内の **Block1** を右クリックして「制約の追加」を選択してください。



要素のプロパティ画面が開き、「制約」グループが選択された状態となります。今回は「**a = time**」と入力してください。なお、「**time**」はシミュレーション時の時間を示す **OpenModelica** のキーワードです。



「保存」ボタンを押してから「OK」ボタンを押して画面を閉じてください。その後、SysML シミュレーションタブの「解析」ボタンを押すことで、シミュレーションの実行結果が表示されます。



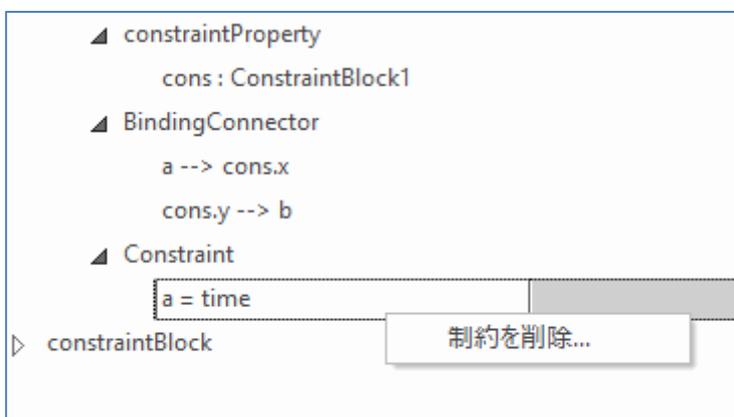
今回は、入力値の  $\sin$  関数の値を単に 2 倍にするだけですので、グラフとしては面白みがないかもしれませんが。しかし、モデルの作成からシミュレーションの実行までの基本的な流

これは把握できるかと思います。

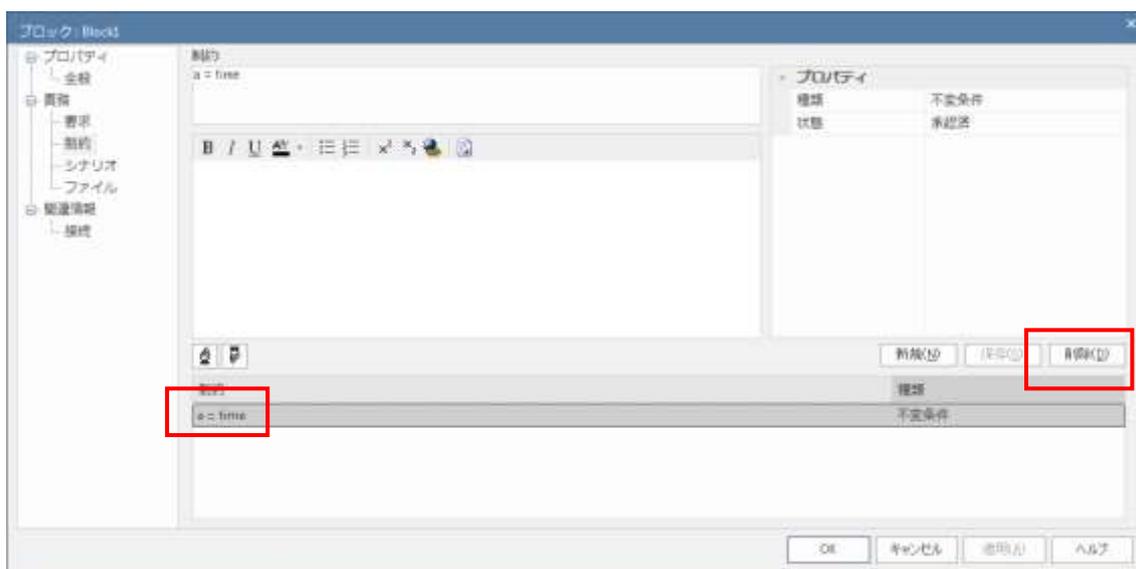
このドキュメントでは、もう1つ「データセット」の使い方についても紹介します。

実際のシミュレーションでは、パラメータについていくつかのパターンを設定し、その結果がどのように変わるのかを比較検討するようなことがあるのではないかと思います。そうした場合のパターンに対応するものが「データセット」です。

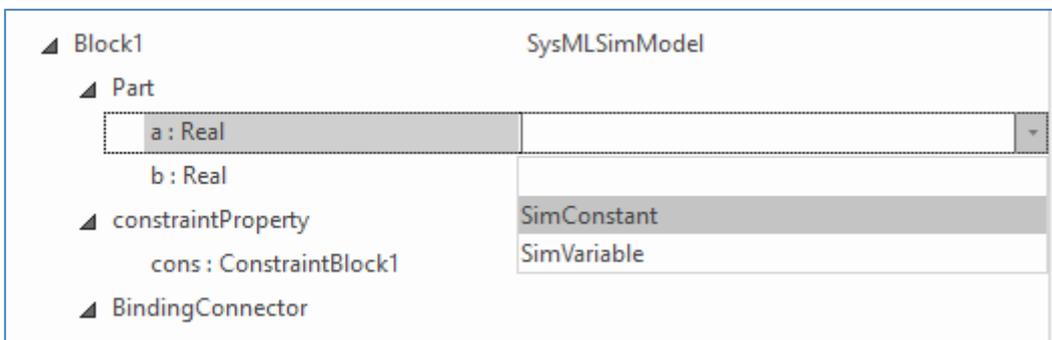
先ほどのサンプルモデルを利用し、データセットの基本的な使い方を紹介します。まず、最後に作成した「a = time」の制約を削除します。一覧で「a = time」の制約を右クリックし、「制約を削除」を実行してください。



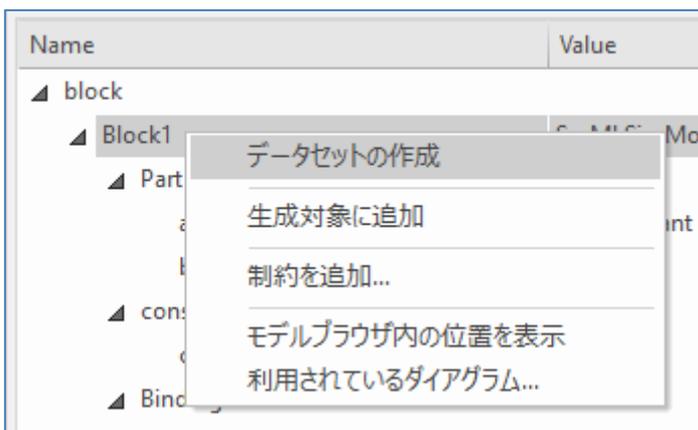
現在は、この一覧画面からの直接削除には対応していないため、先ほど作成に利用したプロパティ画面が開きます。右上の「a = time」を選択し「削除」ボタンを押してから画面を閉じてください。



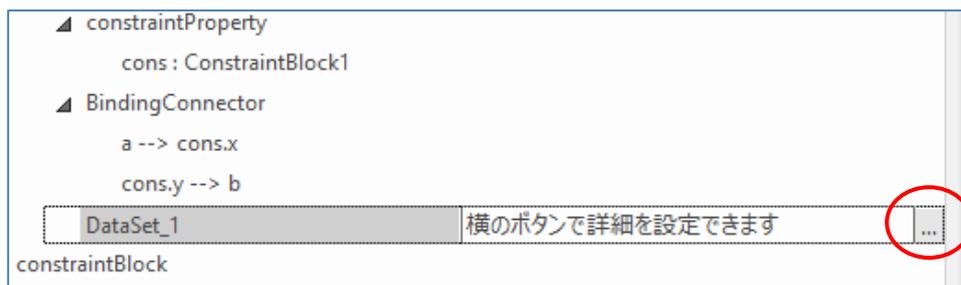
次に、プロパティ a を固定値であると指定します。指定する方法は、一覧内で a の横のセルをクリックし、表示されるドロップダウンリストで「SimConstant」を選択して下さい。



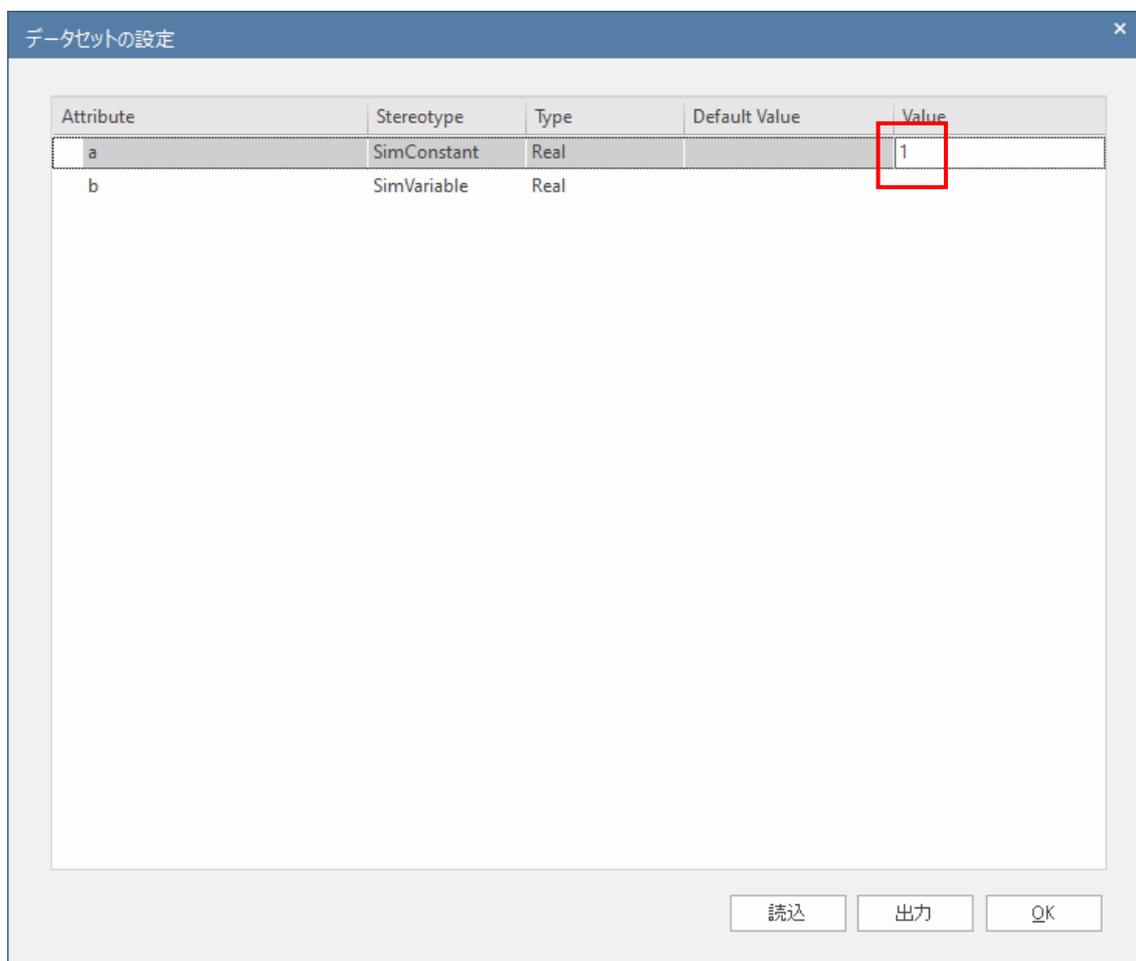
このようにして定数値として指定したプロパティについて、「データセット」で具体的な値を指定できます。データセットを追加するには、そのセットを保持するブロック(今回は Block1)を一覧内で右クリックし、「データセットの作成」を選択して下さい。



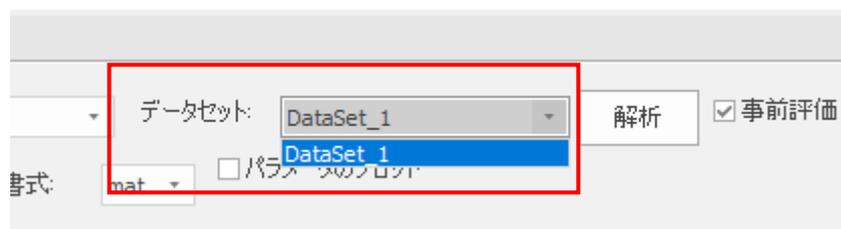
一覧内にデータセット「DataSet\_1」が追加されます。「横のボタンで詳細を設定できます」と書かれたセルをクリックすると参照(...)ボタンが表示されますので、そのボタンを押して下さい。



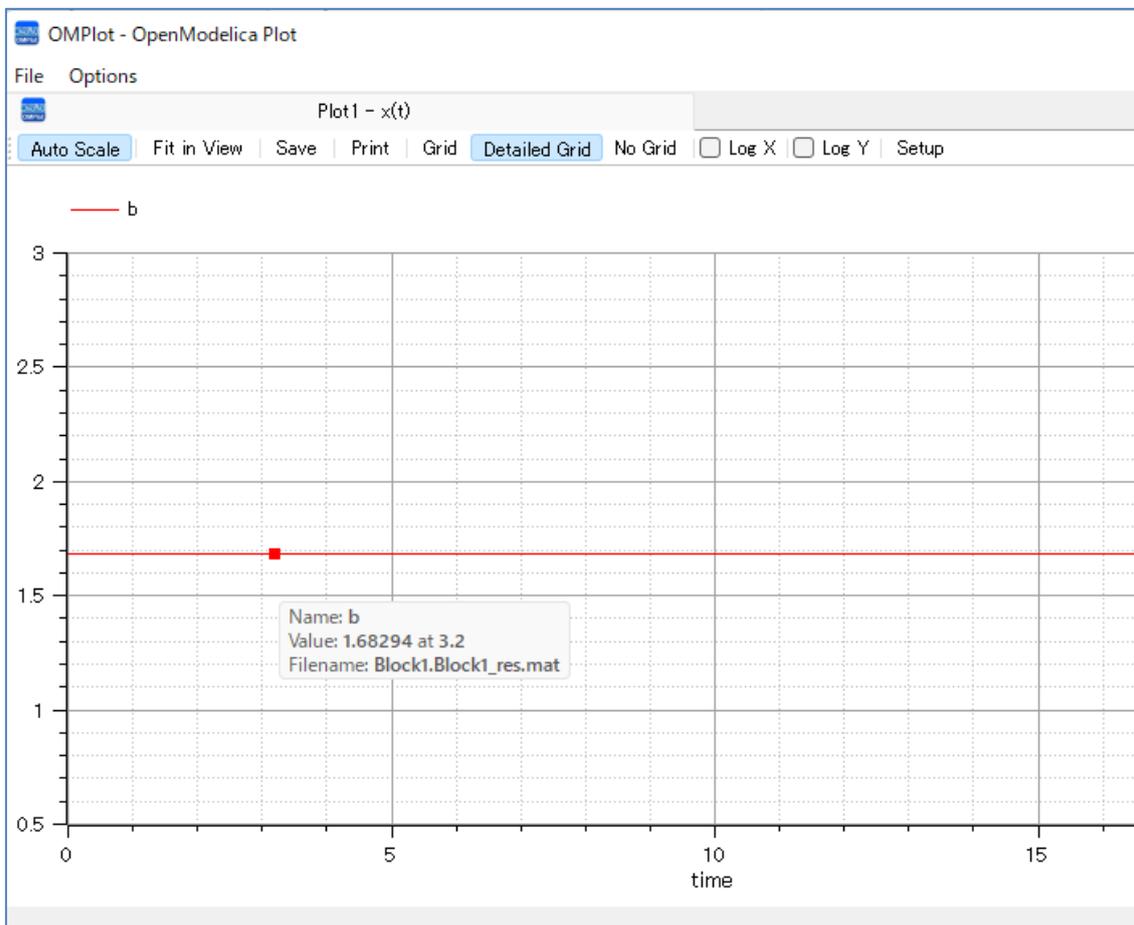
「データセットの設定」画面が表示されますので、先ほど定数値に指定した変数の Value の欄に、数値を入力します。今回は「1」と入力しました。



あとは、作成したデータセットを選択後、「解析」ボタンを押してシミュレーションを実行します。



今回もモデルが単純なので結果のグラフはつまらないものですが、指定したデータセットについて制約を満たす場合、つまり  $2 * \sin(1)$  の値(今回は 1.68294)を知ることができます。



このようなデータセットを複数作成し、切り替えて検証することができます。

ここまでの、SysML パラメトリック図のシミュレーション機能を利用する基本となります。サンプルモデルには、さまざまな例が含まれていますので、実際のモデルを作成する際に参考になるかと思います。

## 5 SysPhSの場合のモデルの作成

SysPhS (Simulink あるいは OpenModelica)に対応したモデルを作成する場合には、以下の点に注意する必要があります。

- ブロックが保持するポートおよびポート間のコネクタ(接続)は、すべて向きが一意に決まるように定義する必要があります。コネクタにアイテムフローを定義する

場合、方向が両端のポートの方向と一致してはなりません。ポートは **in** か **out** のみとなります。

(Simscape を利用する場合で物理的なやりとりの場合は、ポートは **inout** のみとなります。)

- 定数値となるプロパティには **PhSConstant** のステレオタイプを、変数となるプロパティには **PhSVariable** のステレオタイプを付加してください。
- 各ブロックは、内部ブロック図・パラメトリック図・ステートマシン図(状態マシン要素)のいずれか 1 つのみを保持できます。
- 制約を式で定義する場合、左辺は 1 つの変数のみとなります。例えば、「 $x = y + z$ 」の形式は利用できますが、「 $y + z = x$ 」の形式は利用できません。また、左辺の変数が出力(結果)を示します。
- 最上位(シミュレーションの実行対象)となるブロック要素を 1 つ作成し、そのブロック要素に内部ブロック図あるいはパラメトリック図を追加してください。

(SysML シミュレーションの設定タブでは、この最上位の要素を「SysMLSimModel」として設定します。)

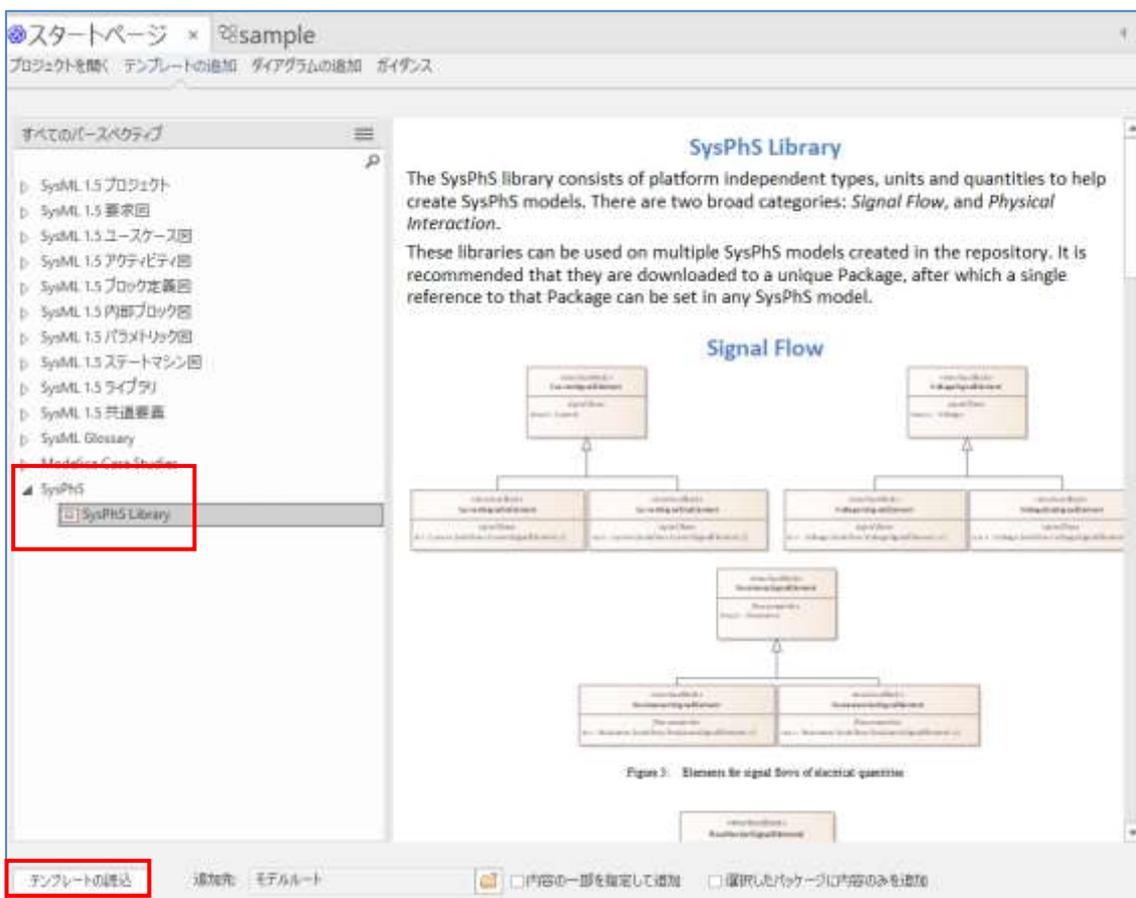
基本的な作業の流れは以下の通りです。

1. ブロック定義図に「最上位」となるブロック要素を作成します。
2. そのブロック要素から利用する他のブロック要素や制約ブロック要素を作成します。それぞれの要素について、入出力となるポートやプロパティを追加します。また、内部ブロック図あるいはパラメトリック図を作成し、その要素を配置します。(制約ブロックの場合で、式によって制約を示す場合にはパラメトリック図は不要です。)
3. 「最上位」となるブロック要素の内部ブロック図に、上記の要素を配置します。
4. 「SysML シミュレーションの設定」要素(SysMLSimConfiguration)を作成し、「最上位」となるブロック要素をシミュレーションの実行対象に設定し、シミュレーションを実行します。

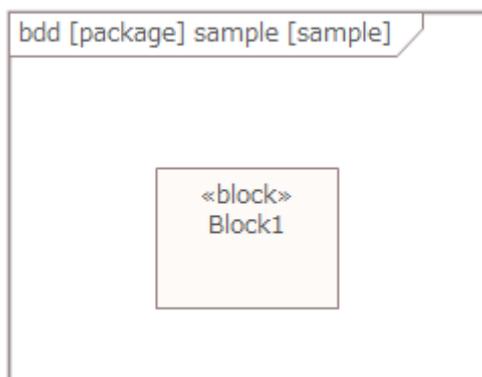
以下、作成例となります。

まず、モデルルート直下に **SysPhS** ライブラリを読み込みます。モデルブラウザ内でモデルルートを選択した状態で、モデルテンプレートの追加画面から、「**SysPhS Library**」を読み込みます。

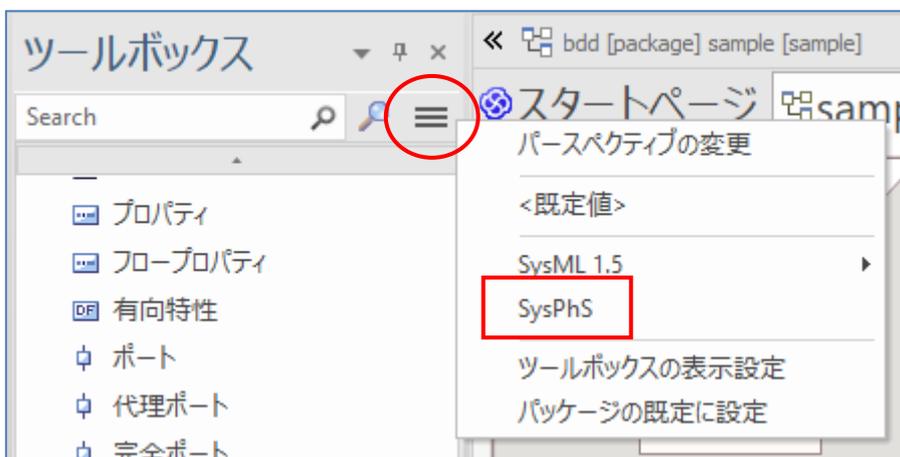
(バージョン 15.2 では「**SysPhS Elements for signal flow**」のテンプレートを読み込みます。画面構成も下の画像と多少異なります。)



次に、パッケージを新規に作成し、ブロック定義図を追加します。図内に、最上位となるブロック要素を配置します。

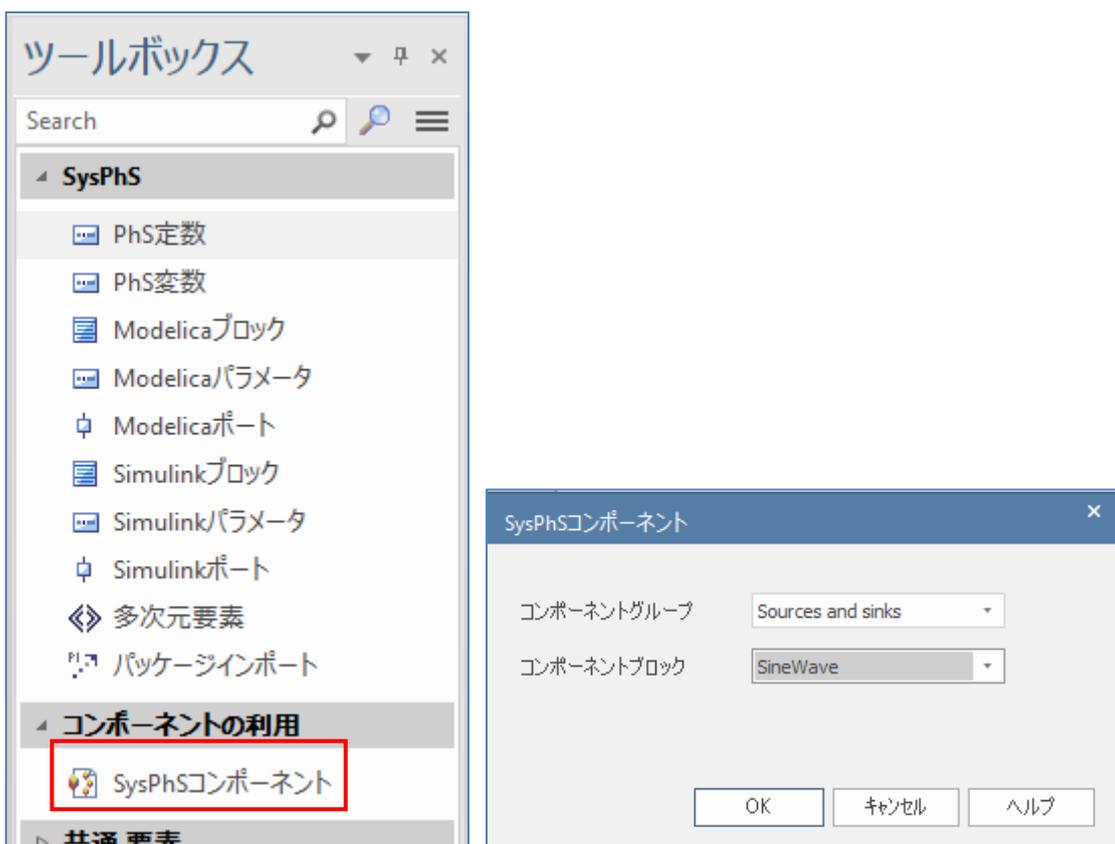


次に、ツールボックスを「SysPhS」に切り替えます。

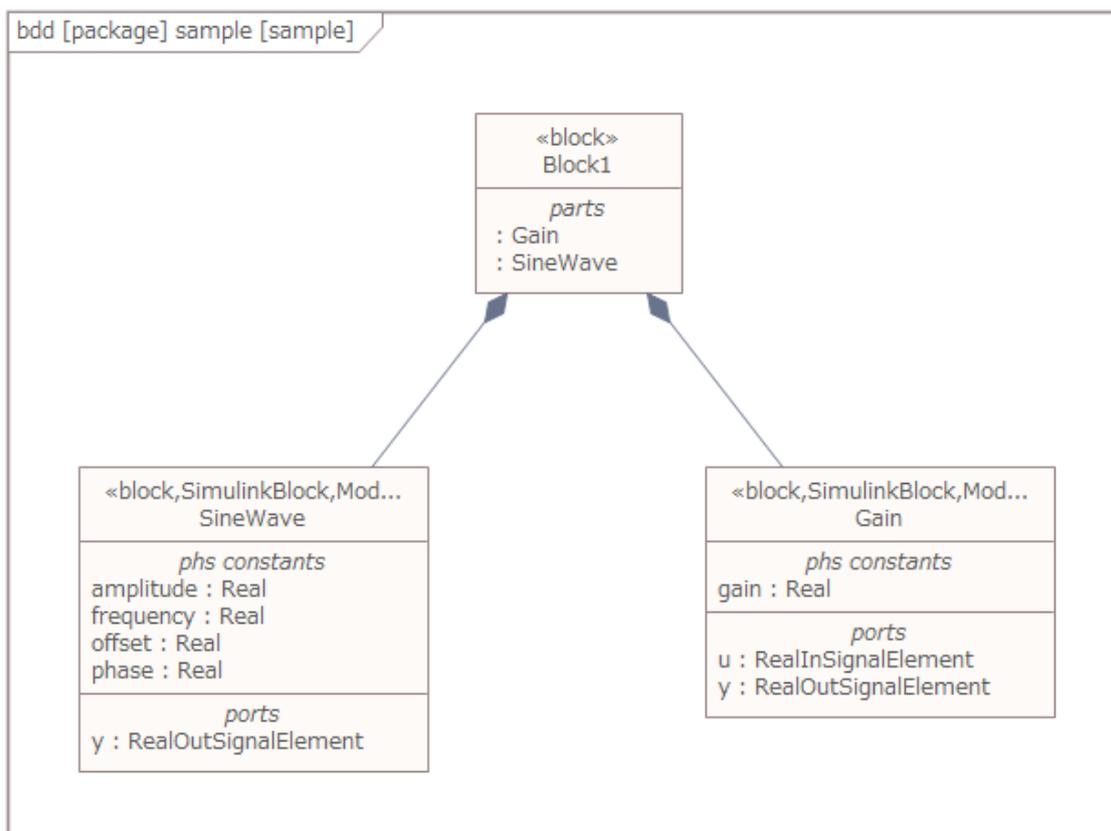


ツールボックスの「SysPhS コンポーネント」をブロック定義図内にドロップし、表示される画面で以下の要素を作成します。

- Source and Sinks から SineWave
- Mathematical Component から Gain

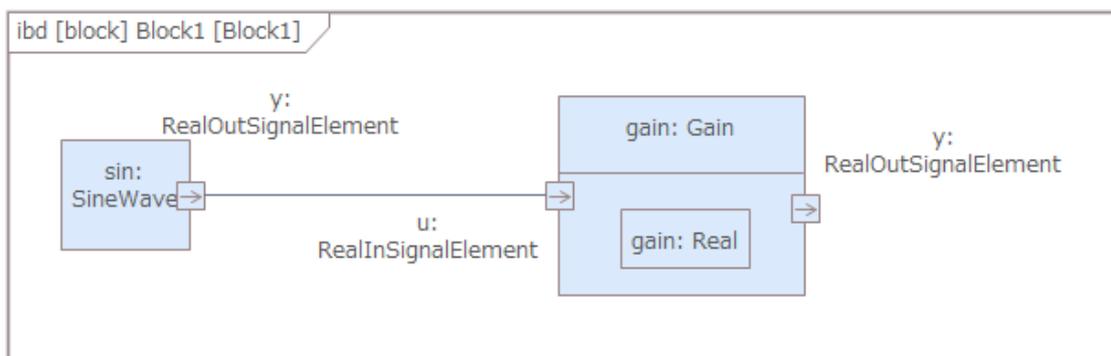


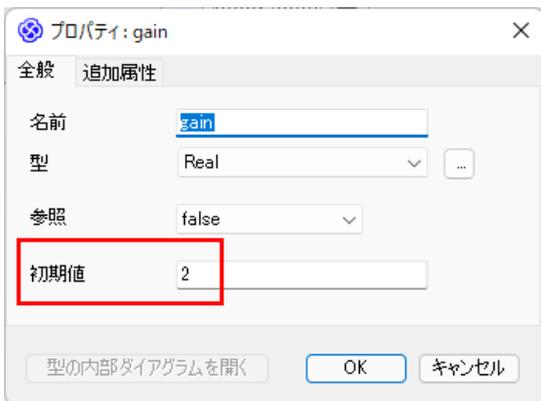
作成した要素は、コンポジションの関係で最上位となるブロックと結びます。



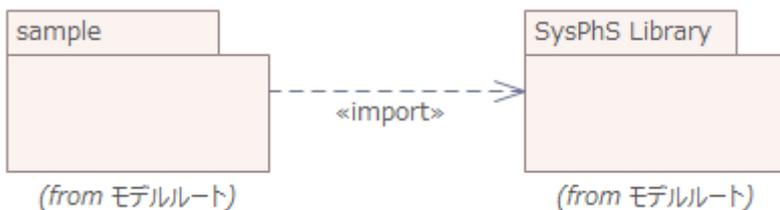
(ツールボックスが ShsPhS のままになっていると、クイックリンクでコンポジションを作成できません。ツールボックスを元に戻してください。)

次に、最上位となるブロック要素に内部ブロック図を作成し、プロパティのポート間をコネクタで接続します。また、Gain 要素が持つプロパティに、初期値 2 を設定します。

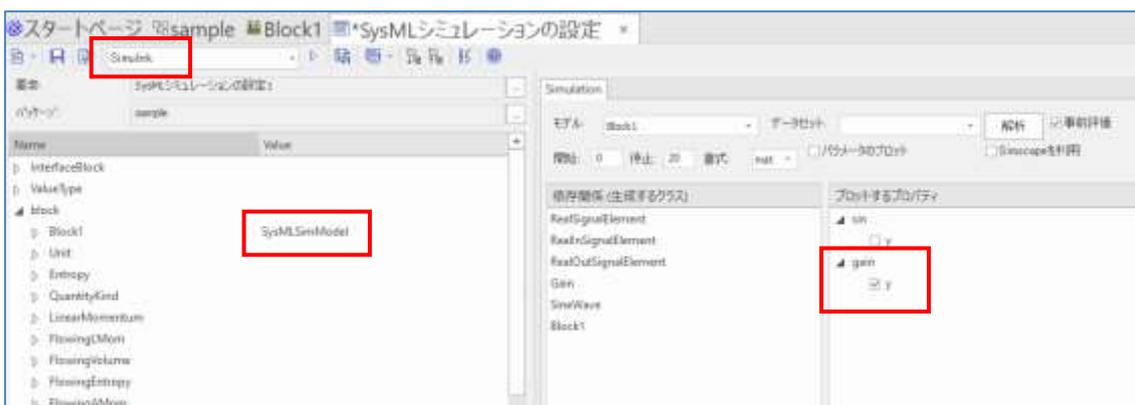




さらに、SysPhS ライブラリを利用するために、パッケージ図を作成してインポートの関係でモデルのパッケージとライブラリのパッケージ間を接続する必要があります。

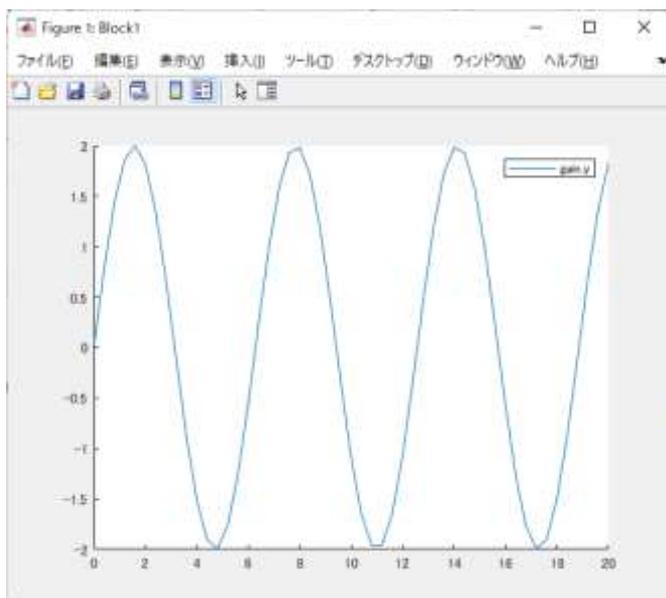


最後に、ブロック定義図に SysML シミュレーション要素を配置し、ダブルクリックして SysML シミュレーションの設定タブを開きます。下記赤枠部分の設定を行い、解析ボタンを押すことで、シミュレーションを実行できます。以下の例は、Simulink を利用する場合の例です。OpenModelica を利用する場合には、利用するツールの選択肢を「Simulink」から「OpenModelica」に変更してください。



(設定方法の詳細は、第 4 章の OpenModelica の説明をご覧ください。)

モデルで定義したとおり、Sin 関数の 2 倍の値が出力されていることが確認できます。

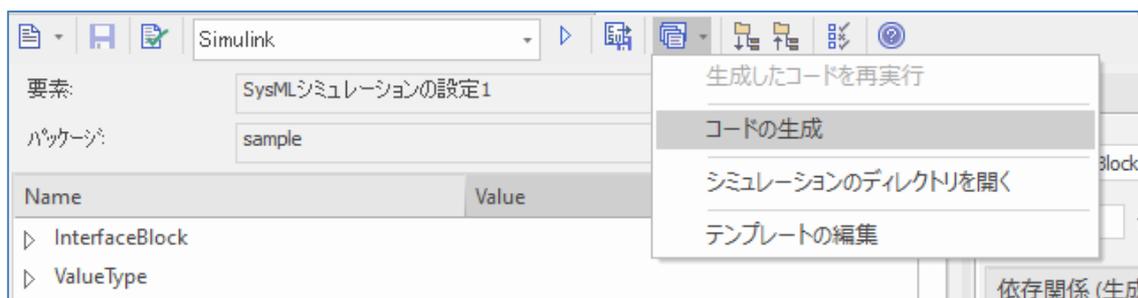


## 6 Simulinkファイルの生成についての補足

このドキュメントで紹介した内容は、SysML のモデルからシミュレーションを実行するまでの概要と手順になります。このシミュレーションの実行エンジンとして、OpenModelica や Simulink を利用できます。

この機能を応用することで、SysML モデルから Simulink のファイルを生成することができます。基本的にはシミュレーションの仕組みを使いますので、ある程度はこのドキュメントで説明した構成に基づいた SysML モデルを作成することが必要となります。ただし、シミュレーションを実行する必要はありませんので、シミュレーションを実施するための全ての条件を満たす必要はありません。

コードを生成する場合には、SysML シミュレーションの設定タブ内のツールバーから「コードの生成」を実行します。



実行すると、ファイルを保存する位置を選択する画面が表示されますので、出力先を指定してください。シミュレーションの実行用である「Solve.m」ファイルと、モデルの内容に関する m ファイルや slx ファイルが出力されます。

なお、現在は出力後に SysML モデルを変更した場合に、既存の slx ファイルなどとの同期(マージ)はできません。生成したファイルを Simulink 側で編集する場合にはご注意ください。