

双腕ロボットの教示ソフトウェア マニュアル



株式会社 グローバルアシスト

目 次

1. 双腕ロボットの教示ソフトウェアのインストール, 初期設定	2
1.1 TEACHING PLUGIN のインストール	2
1.2 TEACHING PLUGIN の初期設定	4
1.2.1 使用するデータベースの作成	4
1.2.2 各種情報の設定	4
2. 各ビューの説明	5
2.1 タスクインスタンス情報の編集	5
2.1.1 タスクインスタンス情報のインポート/エクスポート	5
2.1.2 タスクインスタンス情報の検索	5
2.1.3 タスクインスタンス情報の削除	6
2.1.4 タスクインスタンス情報の保存	6
2.2 フロー情報の編集	7
2.2.1 フロー情報の新規作成/検索/削除/登録	7
2.2.2 フロー情報の編集	7
2.3 状態遷移情報の編集	8
2.3.1 状態遷移情報の組み立て	8
2.4 メタ情報の編集	11
2.4.1 モデル情報	11
2.4.2 コメント, 付属文書, 付属図	12
2.5 パラメータの設定	14
2.6 タスクの実行	15
2.6.1 タスク単体での実行	15
2.6.2 フローの実行	15
3. 付録	16
3.1 データベース定義	16
3.2 コマンド定義ファイル	エラー! ブックマークが定義されていません。
3.3 タスクモデル記述ファイル	18

1. 双腕ロボットの教示ソフトウェアのインストール, 初期設定

双腕ロボットの教示ソフトウェア(以下, Teaching Plugin)は, 多様な作業に利用できる可能性を持つ双腕ロボットのユーザが教示により作成したロボットの動作に関するデータを一旦保存し, 類似した作業に対して保存したデータを再利用するためのツールです。

1.1 Teaching Plugin のインストール

Teaching Plugin は, 国立研究開発法人 産業技術総合研究所が開発を行っている, オープンソースのロボット用統合 GUI ソフト「コレオノイド(Choreonoid)」のプラグインとして動作します。

Teaching Plugin をインストールするためには, コレオノイドをソースコードからビルド・インストールする必要があります。コレオノイドのソースファイルを展開した後, extplugin の配下に teachingPlugin をディレクトリごとコピーし, コレオノイド全体のビルド・インストールを行ってください。

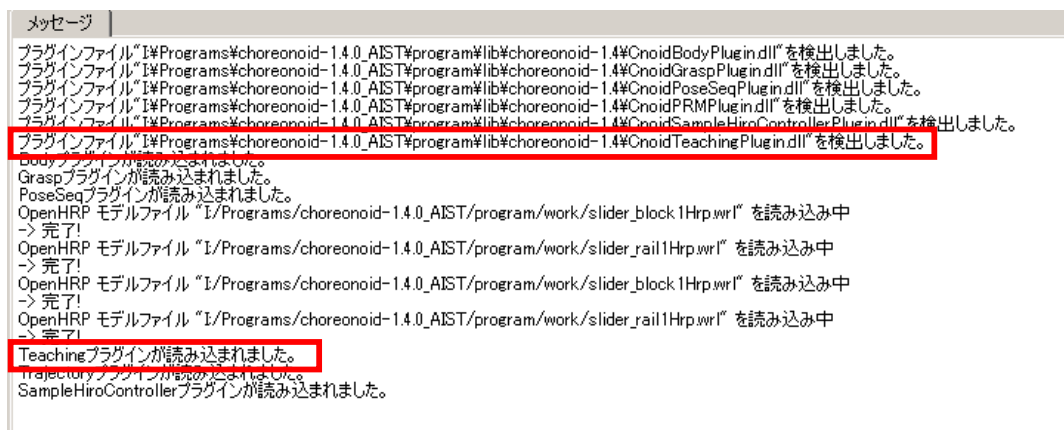
※コレオノイドをソースコードからインストールする手順については, 以下のサイトなどをご確認ください。

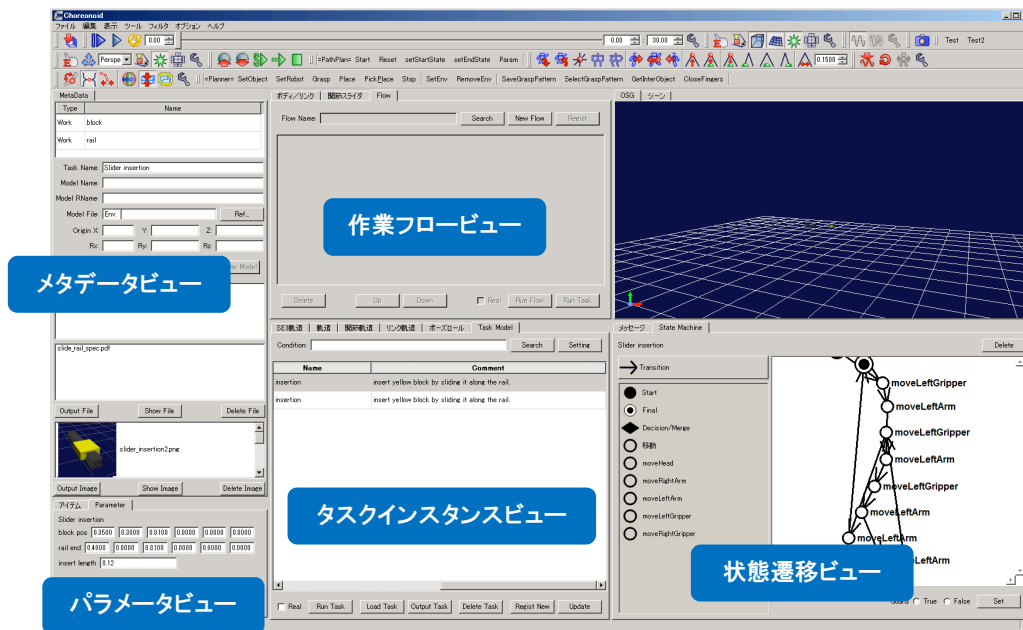
<http://choreonoid.org/ja/install/install.html>

※現状の Teaching Plugin は, Choreonoid バージョン 1.4 のプラグインとして動作します。バージョン 1.5 では動作しませんので, ご注意ください。

Teaching Plugin ではデータベースとして SQLite を使用しています。ご使用の OS に応じて, 対応する SQLite も併せてインストールしておいてください。

Teaching Plugin は正常にビルド・インストールされると, コレオノイドを起動時に, 自動的に読み込まれます。コレオノイドを起動した際に, メッセージ・ビューに以下のようなメッセージが表示されていれば, インストールは成功です。





◎作業フロービュー

双腕ロボットの一連の作業フローを表示，組み立てするための画面です．組み立てたフローや，フロー構成するタスクインスタンスの実行を行うことも可能です．

◎タスクインスタンスビュー

双腕ロボットの作業単位(タスクインスタンス)を定義，検索するための画面です．タスクインスタンスは，作業フローを組み立てるための基本動作要素となります．

◎状態遷移ビュー

対象タスクを実行するための詳細手順を確認，定義するための画面です．各タスクを実行するための一連の動きを，コマンドや条件分岐などを使用して記述します．

◎メタデータビュー

タスクインスタンスに関連した各種メタデータを設定，表示する画面です．タスクインスタンスには，モデルファイルや，関連文書・図などを設定することができます．

◎パラメータビュー

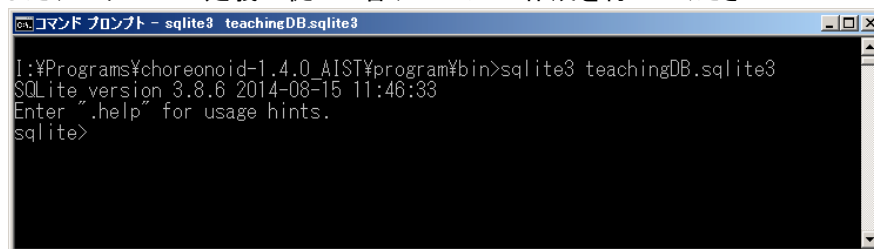
タスクインスタンスに関連したタスクパラメータの値を設定するための画面です．

1.2 Teaching Plugin の初期設定

Teaching Plugin を使用するためには、事前にいくつか設定を行う必要があります。

1.2.1. 使用するデータベースの作成

Teaching Plugin で使用するデータベースを事前に作成する必要があります。SQLite のコンソール画面を起動し、付録に示したデータベース定義に従って各テーブルの作成を行ってください。



```
コマンド プロンプト - sqlite3 teachingDB.sqlite3
I:\Programs\choreonoid-1.4.0_AIST\program\bin>sqlite3 teachingDB.sqlite3
SQLite version 3.8.6 2014-08-15 11:46:33
Enter ".help" for usage hints.
sqlite>
```

1.2.2. 各種情報の設定

【タスクインスタンスビュー】の「Setting」ボタンをクリックすると【Setting】画面が表示されます。こちらの画面で以下の設定を行ってください。

• Database:

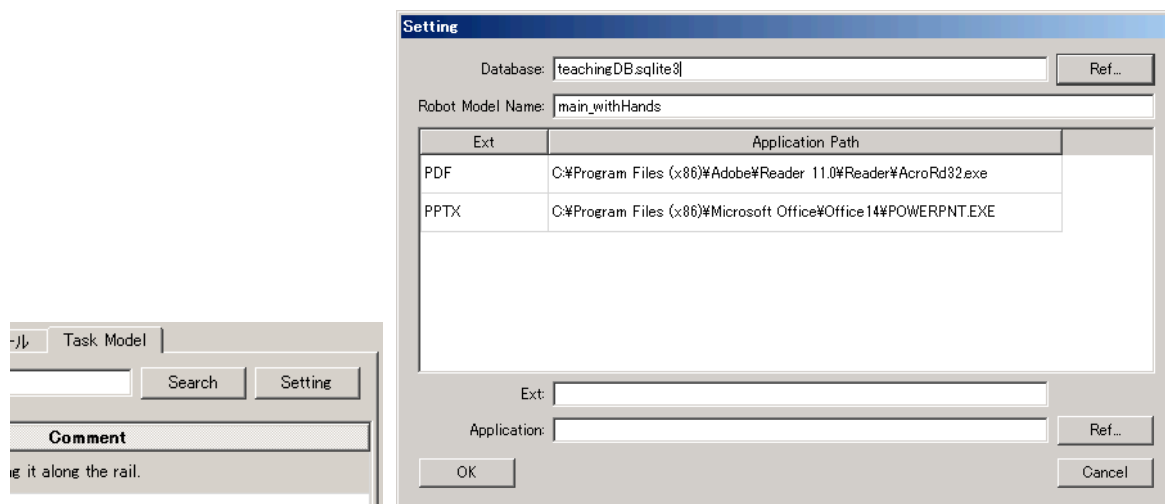
使用するデータベースファイルを指定してください。複数のデータベースを使用したい場合には、この画面で対象のデータベースを切り替えてください。

• Robot Model Name

対象プロジェクト内のロボットモデルの名称を設定してください。

• 関連アプリケーション

メタデータビュー内に登録した付属文書を表示するためのアプリケーションを登録します。対象ファイルの拡張子と対応するアプリケーションを設定してください。



※【Setting】画面で設定した内容は、Choreonoid の実行ファイル(choreonoid.exe)と同じディレクトリに存在する設定ファイル Setting.prm 内に保存されます。画面から設定ができない場合には、設定ファイルを直接編集してみてください。設定ファイルの書式については、付録を参照してください。

2. 各ビューの説明

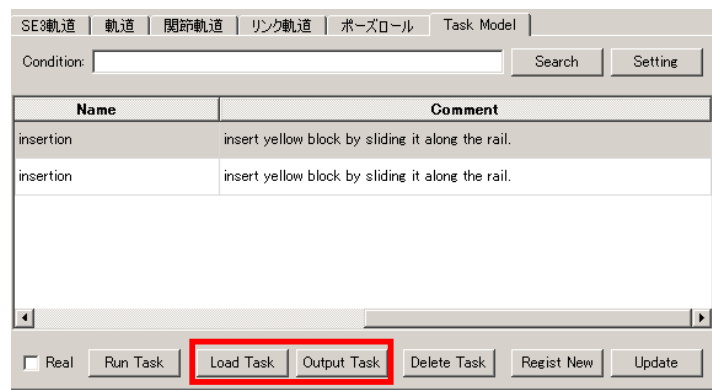
2.1 タスクインスタンス情報の編集

2.1.1. タスクインスタンス情報のインポート/エクスポート

タスクインスタンスは、タスクモデル記述ファイルの形式でインポート/エクスポートが可能です。タスクインスタンスを追加したい場合には、事前にタスクモデル記述ファイルを作成し、インポートを行ってください。(タスクモデル記述ファイルの詳細定義については、付録を参照してください)

タスクインスタンス情報をインポートする場合には、【タスクインスタンスビュー】の「Load Task」ボタンをクリックしてください。【ファイル選択】画面が表示されますので、インポート対象のファイルを選択してください。

登録済みのタスクインスタンス情報をタスクモデル記述ファイルにエクスポートする場合には、【タスクインスタンスビュー】でエクスポート対象のタスクモデルインスタンスを選択した状態で「Output Task」をクリックしてください。【ファイル選択】画面が表示されますので、エクスポートファイル名を入力してください。

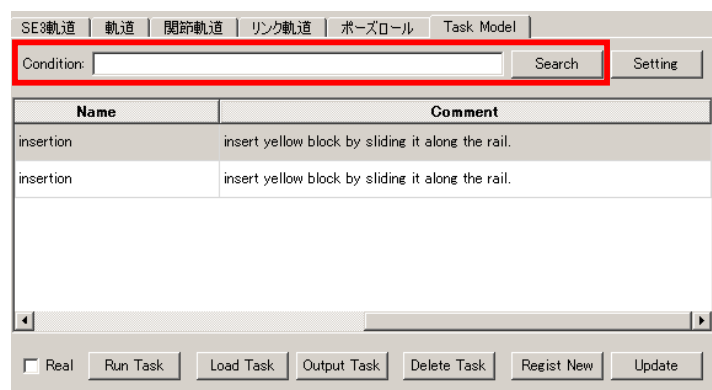


2.1.2. タスクインスタンス情報の検索

登録済みのタスクインスタンス情報を検索したい場合には、【タスクインスタンスビュー】上部の「Condition」欄に検索条件を入力し、「Search」ボタンをクリックしてください。検索条件に一致したタスクモデルインスタンスのみが一覧表示されます。

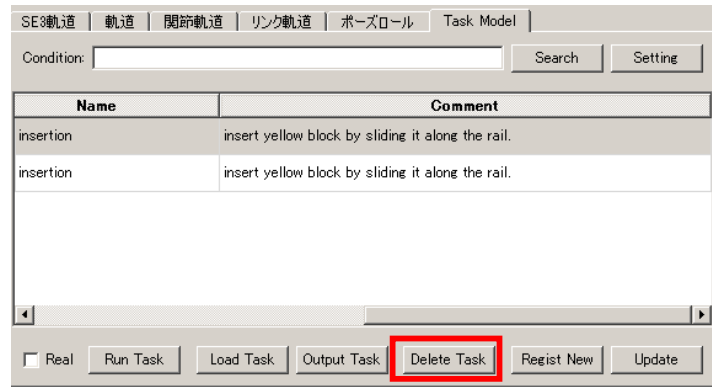
検索条件を複数設定したい場合には、各検索語をスペースで区切る(And 検索)か、各検索語の間に「||」を設定(Or 検索)してください。

検索対象は、タスクモデルインスタンスの名称とコメントの内容となります。



2.1.3. タスクインスタンス情報の削除

登録済みのタスクインスタンス情報を削除したい場合には、【タスクインスタンスビュー】の一覧画面から対象のタスクインスタンスを選択し、「Delete Task」ボタンをクリックしてください。

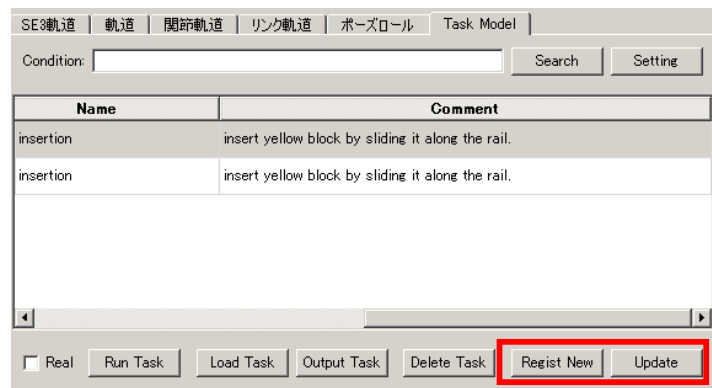


2.1.4. タスクインスタンス情報の保存

タスクインスタンス情報の内容をデータベースに保存したい場合には、【タスクインスタンスビュー】の「Regist New」ボタンまたは「Update」ボタンをクリックしてください。

「Update」ボタンをクリックした場合には、修正した内容を既存のデータに上書き保存します。

「Regist New」ボタンをクリックした場合には、修正した内容を新規データとして登録します。



2.2 フロー情報の編集

タスクインスタンスを複数使用して、一覧の動作をフローとして定義することが可能です。

2.2.1. フロー情報の新規作成/検索/削除/登録

新しいフローを作成したい場合には、【フロービュー】の「New Flow」ボタンをクリックしてください。【フロービュー】がアクティブとなります。

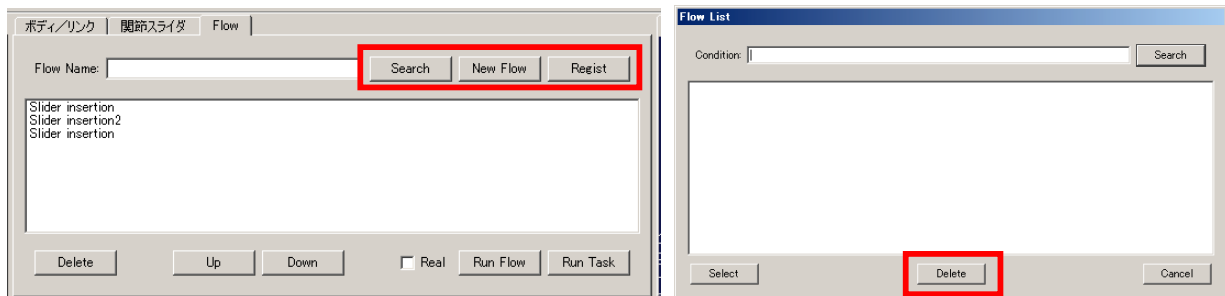
既存のフローの内容を修正したい場合には、【フロービュー】の「Search」ボタンをクリックしてください。【フロー一覧】画面が表示されますので、編集したいフローを選択し「Select」ボタンをクリックしてください。

【フロー一覧】画面では、対象フローの検索を行うことも可能です。上部の「Condition」欄に検索条件を入力し、「Search」ボタンをクリックしてください。検索条件に一致したフローのみが一覧表示されます。

検索条件を複数設定したい場合には、各検索語をスペースで区切る(And 検索)か、各検索語の間に「||」を設定(Or 検索)してください。なお、検索対象は、フロー名称となります。

登録済みのフローを削除したい場合には、【フロー一覧】画面で削除対象のフローを選択し、「Delete」ボタンをクリックしてください。

作成したフロー情報をデータベースに登録したい場合には「Regist」ボタンをクリックしてください。

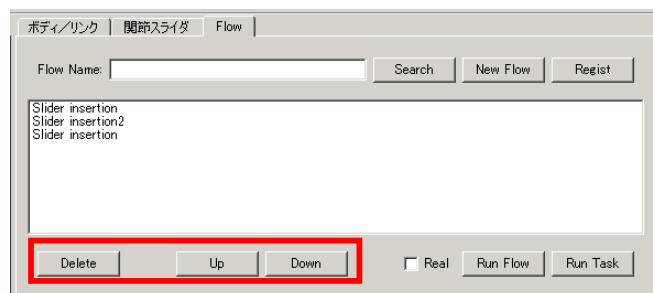


2.2.2. フロー情報の編集

フロー情報に新しいタスクインスタンスを追加したい場合には、対象となるタスクインスタンスを一覧からフロー画面にドラッグアンドドロップしてください。対象データが追加されます。

【フロービュー】に設定したタスクインスタンス情報の順番を入れ替えたい場合には、対象データを選択した状態で「Up」ボタン/「Down」ボタンをクリックしてください。それぞれ、選択したデータが上下方向のデータを入れ替わります。

また、追加したタスクインスタンス情報を削除したい場合には、対象データを一覧で選択した状態で「Delete」ボタンをクリックしてください。



2.3 状態遷移情報の編集

各タスクインスタンスには、処理を実行するための詳細な手順を状態遷移として定義することができます。

2.3.1. 状態遷移情報の組み立て

新しい状態を追加する場合には、左側の要素一覧から対象状態を状態編集エディタ内にドラッグアンドドロップします。

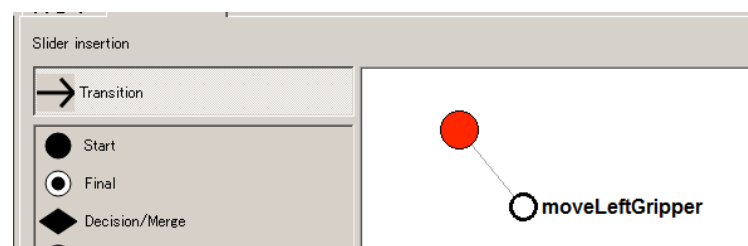
※要素一覧で利用できる状態(コマンド)は、ロボットコントローラから取得しています。状態(コマンド)を変更したい場合には、ロボットコントローラ内の定義を修正してください。



状態編集エリア内で、追加した状態の位置を移動したい場合には、対象要素を選択し、ドラッグすることで移動できます。なお、この際、「要素一覧」上部の「Transition」が OFF になっている事を確認してください。（現在選択中の要素は赤で表示されます）

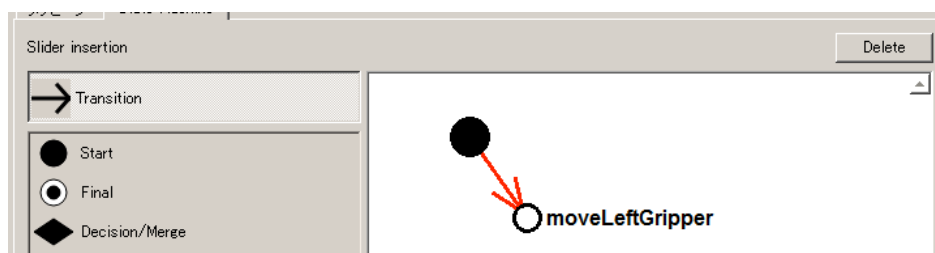


状態間に実行順序である遷移を定義したい場合には、「要素一覧」上部にある「Transition」のボタンを ON にした状態で、状態間をドラッグアンドドロップしてください。

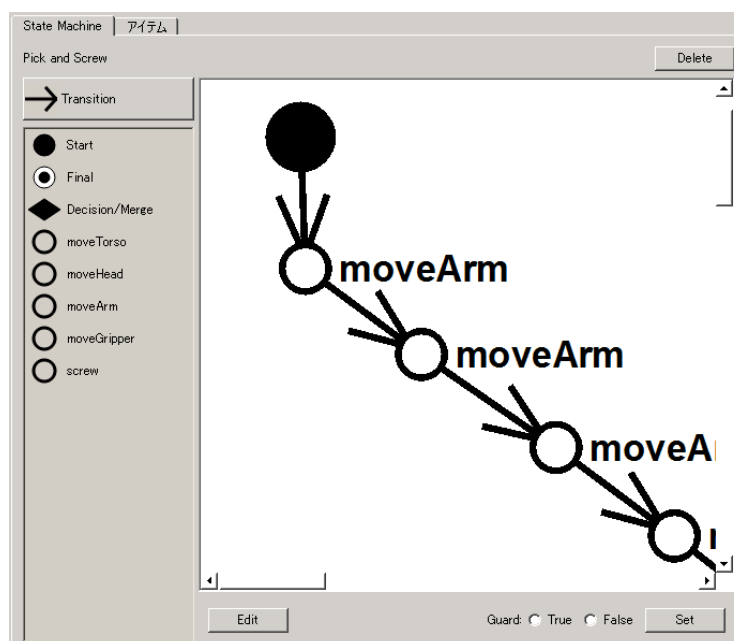


状態編集エディタ内に配置した状態や遷移を削除したい場合には、対象要素を選択した状態で、上部の「Delete」ボタンをクリックしてください。

赤色で表示されている現在選択中の要素、遷移が削除されます。

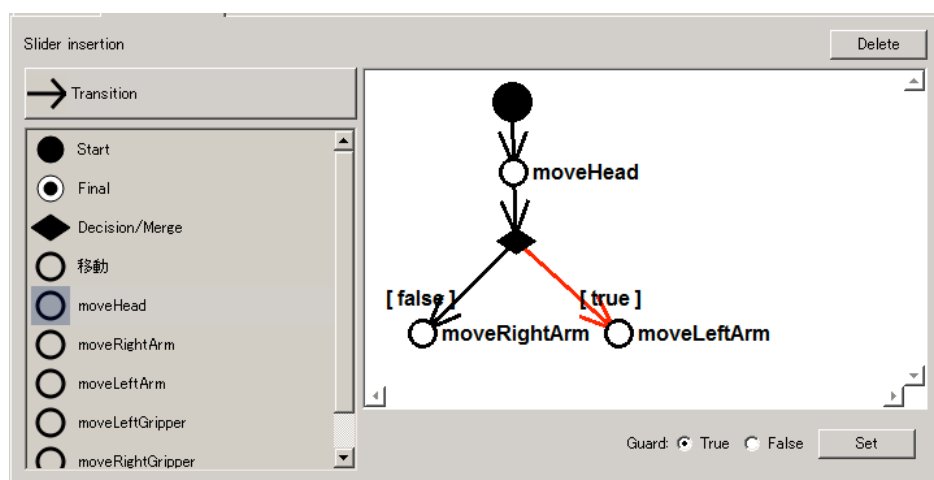


状態編集エディタ内でマウススクロールすることで、状態遷移図の表示を拡大/縮小することができます。



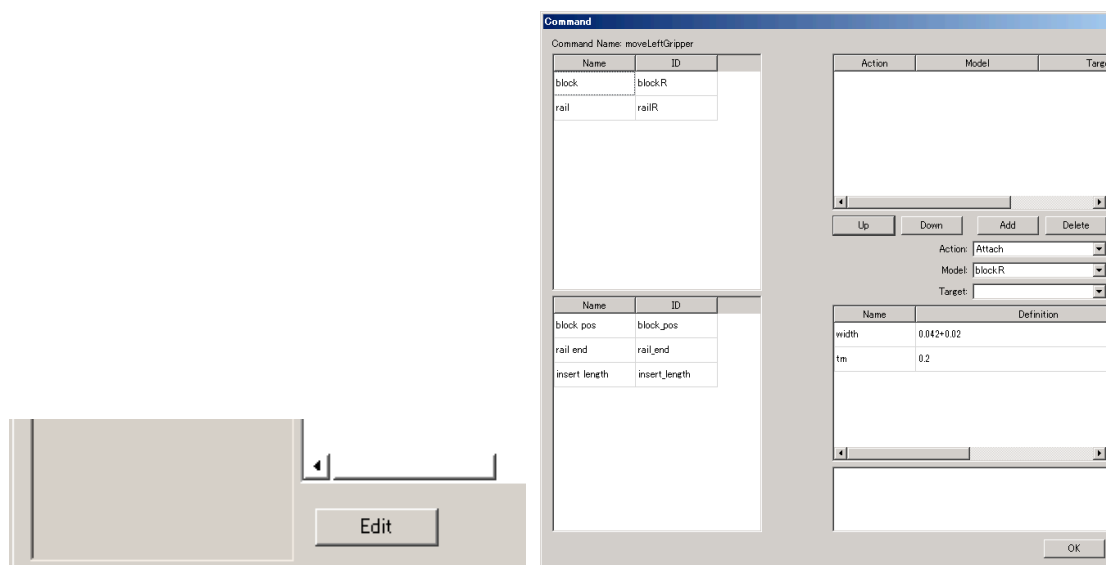
条件分岐(DecisionNode)の後の遷移には、遷移を行うための条件を設定することができます。対象となる遷移を選択した状態で、画面下のラジオボタンで True/False を選択後、「Set」ボタンをクリックしてください。遷移条件が設定され、状態編集エディタ内の表示が変わります。

※条件分岐に使用できるのは、直前に実行したコマンドの実行結果(True/False)のみです。



編集した状態遷移情報をデータベースに登録する場合には、対象タスクの保存(「Update」 or 「Regist New」)を行ってください。

コマンドに紐付けられている状態では、コマンドに受け渡し引数を算出するための計算式と、コマンド実行後に実行可能なモデルアクションを指定することができます。編集を行いたい状態を選択した状態で、「Edit」ボタンを選択してください。



モデルアクションとは、コマンド実行後にプロジェクト内に存在するモデルのツリー構造を変更したい場合に使用するアクションです。例えば、グリッパの開動作を行った際、対象物のモデルをグリッパのモデル配下に接続し、グリッパと同時に移動させたい場合などに使用します。

モデルアクションは、【コマンド定義】画面右上の一覧表を使用して定義することができます。新たなモデルアクションを追加したい場合には「Add」ボタンをクリックし、「Action」コンボボックスで、モデルを接続(Attach)するのか、切断(Detach)するのを選択してください。そして、「Model」コンボボックスで対象モデルを選択するとともに、「Target」コンボボックスで接続/切断元のモデルを選択してください。なお、「Model」および「Target」では各モデルの ID を指定します。モデルの表示名と ID の対応関係は画面左上の一覧表に表示されていますので、モデルを選択する際の参考としてください。

コマンド実行時の引数には、タスクパラメータや各種関数を使用した演算を定義することができます。画面右下の一覧表に、状態に紐付けられたコマンドの引数が一覧表示されています。数式を定義したい引数を選択すると、画面下部の編集エリアに、現在定義されている演算が表示されますので、編集してください。

コマンド引数の演算に使用できる演算子は以下の通りです。

○四則演算(+,-,*,/): スカラ値やベクトルの四則演算を定義します。使用可能な演算は以下の通りです。

<スカラ値> + <スカラ値>, <ベクトル> + <ベクトル>
 <スカラ値> - <スカラ値>, <ベクトル> - <ベクトル>
 <スカラ値> * <スカラ値>, <スカラ値> * <ベクトル>, <ベクトル> * <スカラ値>
 <スカラ値> / <スカラ値>, <ベクトル> / <スカラ値>

○関数: 3次元位置/姿勢情報の抜き出しや、回転行列の算出などを行います。

xyz: 6次元ベクトルから、3次元位置ベクトルを抜き出します。

rpy: 6次元ベクトルから、3次元回転ベクトルを抜き出します。

rotFromRpy: 3次元回転ベクトルから、3×3回転行列を算出します。

rpyFromRot: 3×3回転行列から、3次元回転ベクトルを算出します。

○即値: スカラ値やベクトルを直接記述します。

スカラ値: 直接数値を記述します。

ベクトル値: []で囲んでベクトル要素を記述します。

○タスクパラメータ: タスクパラメータとして定義した値を使用します。記述する際には、タスクパラメータ ID を指定してください。各タスクパラメータの名称と ID の対応関係は、画面左下の一覧表に表示されています。

タスクパラメータをベクトル要素や上記関数の引数として直接使用することも可能です。

2.4 メタ情報の編集

【メタデータビュー】では、各タスクインスタンスに関連した情報を設定することができます。また、タスクインスタンスの名称を変更することもできます。タスクインスタンスの名称を変更したい場合は、「Task Name」の内容を変更してください。

2.4.1. モデル情報

各タスクインスタンスで使用する環境、ワーク、エンドエフェクタのモデル情報を設定することができます。ここで設定したモデルは、【作業フロービュー】や【タスクインスタンスビュー】で対象タスクインスタンスを選択した際に、3D 表示画面に表示されます。なお、「モデル種類」に「エンドエフェクタ」を選択した場合、【設定】画面で設定した「Robot Model Name」のモデルの子要素として追加されます。

Type	Name
Work	block
Work	rail

Task Name:

Model Name:

Model ID:

Model File:

Origin X: Y: Z:

Rx: Ry: Rz:

各モデルには、以下の情報を設定してください。

- ・モデル名「Model Name」
任意の名称を設定可能です。
- ・モデル ID「Model ID」
【アイテム】ビューに表示されるモデル名を設定します。各種パラメータの演算などで使用します。
- ・モデルファイル「Model File」
各モデルの幾何情報などを記述したモデルファイル名です。「Ref」ボタンを使用することで、使用するモデルファイルを入れ替えることができます。また、左側のコンボボックスで、モデル種類(環境、ワーク、エンドエフェクタ)を設定することができます。
※設定できるモデルファイルは、OpenHRP 形式のモデルファイルのみです。
- ・原点情報「Origin」
対象モデルの原点位置・姿勢を設定します。回転要素の単位は degree です。この値を変更すると、自動的に 3D 表示画面の表示が更新されます。また、【アイテム】ビューなどを使用して、3D 表示画面中のモデルの位置を変更した場合、自動的にこの原点位置も更新されます。

新たなモデル要素を追加したい場合には、「Add Model」ボタンを押下してください。新しいモデル要素が一覧画面に追加されます。

また、既存のモデルを削除したい場合には、対象モデルを一覧画面で選択した後、「Delete Model」ボタンを押下してください。

2.4.2. コメント、付属文書、付属図

対象タスクインスタンスに対するコメントは、「コメント設定エリア」に記述してください。

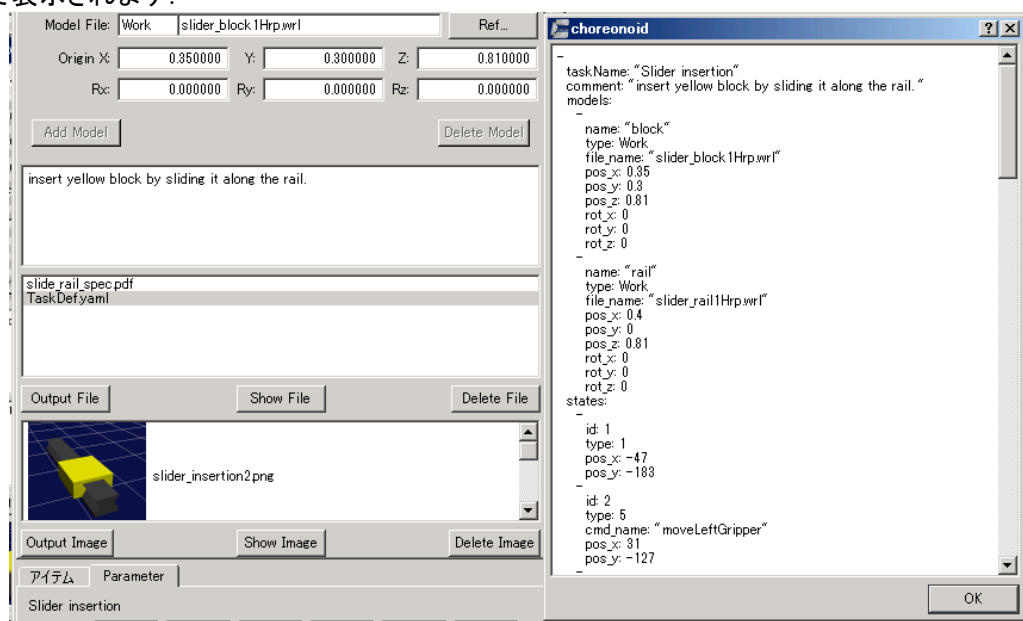
付属ドキュメントや、関連図を設定したい場合には、「ドキュメントエリア」もしくは「画像エリア」に対象要素をドラッグアンドドロップしてください。

「ドキュメントエリア」に関連文書をドロップすると、ファイル名が一覧表に追加されます。同様に「画像エリア」に関連図面をドロップすると、ファイル名とサムネイル表示が追加されます(「画像エリア」に追加可能な画像形式は、JPG 形式と PNG 形式のみです)。



設定した関連文書の中身を確認したい場合には、対象データを選択して「Show File」ボタンをクリックするか、ドキュメントエリアで対象データをダブルクリックしてください。対象文書の中身を別ウィンドウで表示します。

※各文書は、それぞれの拡張子に応じて、設定画面で指定したアプリケーションを利用して表示を行います。対象文書の拡張子に対応するアプリケーションが定義されていない場合は、テキスト形式のデータとして表示されます。

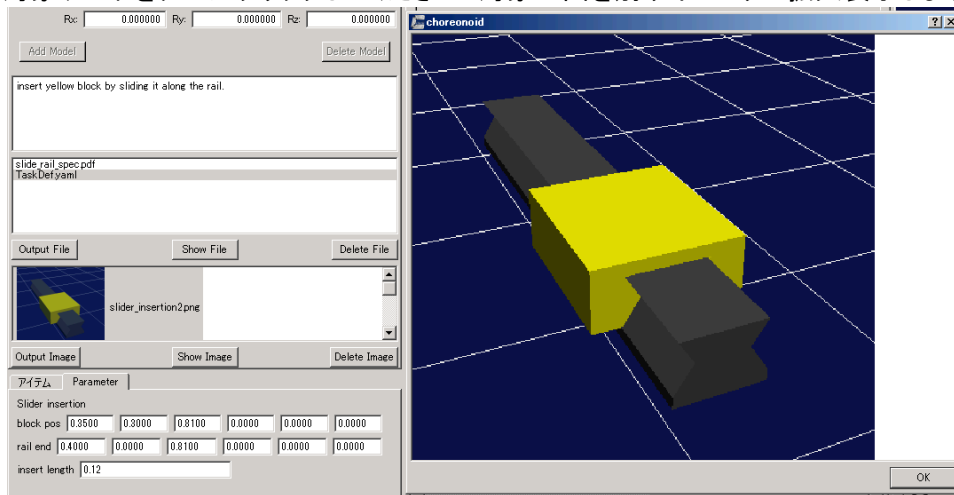


追加した関連文書を削除したい場合には、対象データを選択して「Delete File」ボタンをクリックしてください。対象データが削除されます。

関連文書を再度ファイルとして出力したい場合には、対象データを選択して「Output File」ボタンをクリック

してください。ファイルの保存先を指定すると、関連文書をファイル保存します。

設定した関連図を確認したい場合には、対象データを選択して「Show Image」ボタンをクリックするか、画像エリアで対象データをダブルクリックしてください。対象の図を別ウインドウで拡大表示します。



追加した図を削除したい場合には、対象データを選択して「Delete Image」ボタンをクリックしてください。対象データが削除されます。

図を再度ファイルとして出力したい場合には、対象データを選択して「Output Image」ボタンをクリックしてください。ファイルの保存先を指定すると、関連文書をファイル保存します。

なお、関連文書、図ともに、上記の操作で削除を行っただけではデータベースからの削除は行われません。データベースからも対象データを削除したい場合には、データベースの保存処理(「Update」or「Regist New」)を行ってください。

2.5 パラメータの設定

【パラメータ】ビューでは、各タスクインスタンスに設定されたタスクパラメータの値を入力することができます。ここで設定された各パラメータ値は、タスク実行時のコマンド引数の計算などに使用されます。

なお、パラメータタイプが「1:モデル連動型パラメータ」のパラメータの場合は、パラメータ値を変更すると、対応するモデルの原点位置・姿勢が変更され、3D 表示画面での表示位置が更新されます。同様に、3D 表示画面で対象モデルの原点位置・姿勢を変更した場合には、対象パラメータの値が自動で更新されます。

タスクパラメータの追加/編集/削除を行いたい場合は、画面右上の「Edit」ボタンを押下してください。【タスクパラメータ】画面が表示されます。

【タスクパラメータ】画面では、右上の一覧表に、定義されているタスクパラメータが一覧表示されます。タスクパラメータの定義を変更したい場合には、対象のタスクパラメータを選択してください。画面右下に選択したタスクパラメータの詳細情報が表示されます。

○Name: タスクパラメータの名称を定義します。

○Id: タスクパラメータの内部 Id を定義します。指定した値がコマンド引数演算時などに使用されます。

○Type: タスクパラメータの種類を選択します。

Normal: 通常のパラメータ。任意の値を設定することができます。

Model: モデル要素連動型パラメータ。モデルの原点位置・姿勢の値となります。

○Model Name: タスクパラメータの種類が Model の場合に指定する要素です。どのモデル要素の原点位置と連動するかを指定します。

○Num: タスクパラメータの種類が Normal の場合に指定する要素タスクパラメータの数を設定します。ここで設定された数に従って、パラメータビューに表示されるテキストボックスの数が増減します。

○Unit: 現在は使用していません。

○Elem Type: 現在は使用していません。

Name	ID	Name	Type	Model	Unit	Num
SGCB40-40-5	cover	ネジ供給位置:xyz[m]RPY[deg]	Model	screw		
SGRB40-40-12	block	組付け対象物位置:xyz[m]RPY[deg]	Model	cover		
SCB4-10	screw	ネジ穴オフセット:xyz[m]RPY[deg]	Normal			6
		ネジ締めトルク[Nm]	Normal			1
		ネジ長[m]	Normal			1

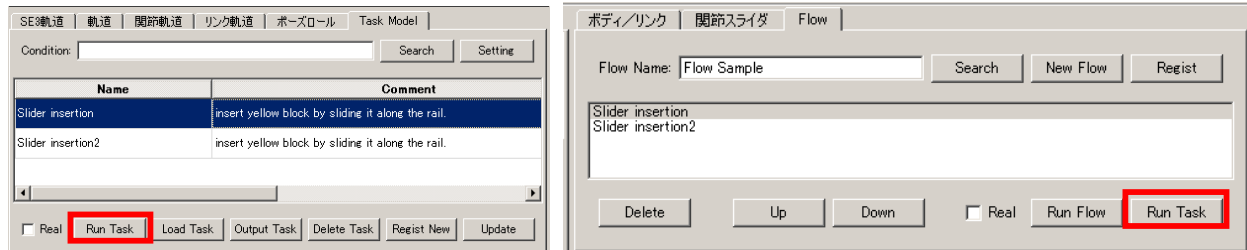
2.6 タスクの実行

各ビューで設定した内容を使用して、タスクインスタンスやフローを実行することができます。

2.6.1. タスク単体での実行

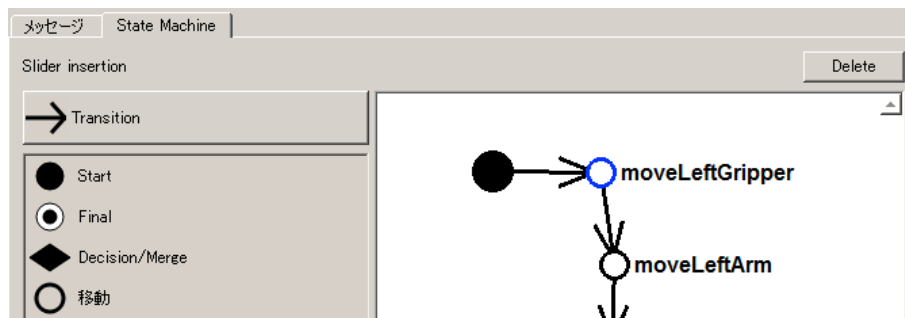
各タスクインスタンス単体で実行を行いたい場合には、【タスクインスタンスビュー】で対象タスクを選択後に「Run Task」ボタンをクリックするか、【作業フロービュー】でフロー内のタスクを選択し「Run Task」ボタンをクリックしてください。

※シミュレータではなく、実ロボットを動作させる場合には「Real」のチェックボックスを ON にして、タスクを実行してください。



各タスクインスタンスに設定された内容を使用して、定義された状態遷移の処理を実行していきます。なお、タスク実行中は、現在実行している状態(コマンド)が【状態遷移ビュー】内で青で表示されるとともに、3D 表示画面の表示も切り替わります。

※タスクを実行するためには、状態遷移図で定義された各コマンドを実行するためのコントローラが必要となります。

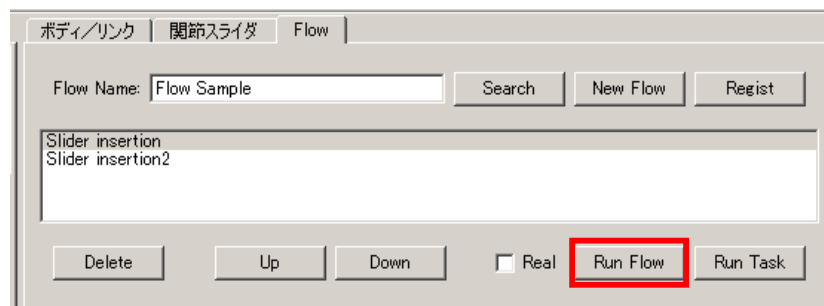


2.6.2. フローの実行

一連の流れを定義したフローを実行したい場合には、【作業フロービュー】で「Run Flow」ボタンを選択してください。現在選択中のフローが実行されます。

※フロー実行時には、フロー内に定義された各タスクインスタンスを順々に実行していきます。この際、タスクインスタンスが切り替わるタイミングで、3D 表示ビューないのモデル表示も各データに設定されているモデルに自動的に切り替わります。

※シミュレータではなく、実ロボットを動作させる場合には「Real」のチェックボックスを ON にして、タスクを実行してください。




```
CREATE TABLE T_FIGURE (
  figure_id    integer primary key,
  task_inst_id integer not null,
  name         text not null,
  data         blob
);
```

```
CREATE TABLE T_FILE (
  file_id      integer primary key,
  task_inst_id integer not null,
  name         text not null,
  file_data    blob
);
```

```
CREATE TABLE T_TASK_PARAMETER (
  task_param_id integer primary key,
  task_id        integer not null,
  type           integer not null,
  model_name     text,
  elem_num       integer not null,
  elem_types     text,
  name           text,
  rname          text,
  unit           text
);
```

```
CREATE TABLE T_TASK_INST_PARAMETER (
  parameter_id integer primary key,
  task_inst_id integer not null,
  task_param_id integer not null,
  value         text
);
```

```
CREATE TABLE T_STATE (
  state_id      integer primary key,
  task_inst_id integer not null,
  type          integer not null,
  cmd_id        integer,
  cmd_name      text,
  pos_x         real not null,
  pos_y         real not null
);
```

```
CREATE TABLE T_STATE_ACTION (
  state_action_id integer primary key,
  state_id         integer not null,
  seq             integer not null,
  action          text not null,
  model           text not null,
  target          text
);
```

```
CREATE TABLE T_ARGUMENT (
  arg_id    integer primary key,
  state_id  integer not null,
  seq       integer not null,
  name      text,
  value     text
);
```

```
CREATE TABLE T_TRANSITION (
  trans_id      integer primary key,
  task_inst_id  integer not null,
  source_id     integer not null,
  target_id     integer not null,
  condition     text
);
```

3.2 タスクモデル記述ファイル

対象システムで使用するタスクモデルインスタンスを YAML 形式で記述したファイルです。

○タスク定義

- taskName: タスク名称
- comment: タスクに対するコメント

○モデル定義(models)

- name: モデル名
- type: モデル種類.
 - Env: 環境, E.E.: エンドエフェクタ, Work: ワーク
- file_name: モデルファイル名 (VRML 形式)
- pos_x, pos_y, pos_z, rot_x, rot_y, rot_z: モデル原点位置・姿勢

○タスクパラメータ定義(parameters)

- name: パラメータ名
- Type: パラメータタイプ
 - 0: 通常パラメータ, 1: モデル連動パラメータ
- model_name: 連動対象モデル名. Type=1 の場合は必須項目
- units: パラメータ単位
- elem_num: パラメータ数
- elem_type: 各パラメータの型情報(カンマ区切り): 0: double
- values: パラメータ値(カンマ区切り)

○状態定義(state)

- id: 状態を識別するための仮 Id. transition 定義のために使用
- type: 状態の種類
 - 1: 開始状態, 2: 終了状態, 3: デンジョン/マージ状態,
 - 4: フォーク/ジョイン状態, 5: コマンド
- cmd_name: コマンド名
- pos_x, pos_y: 状態表示位置
- モデルハンドリング情報(model_actions)
 - action: モデルの操作方法(attach/detach)
 - model: ハンドリング対象モデル名
 - target: attach/detach を行う親モデル
- コマンド実行時引数情報(arguments)
 - seq: 引数の順番
 - name: 引数名
 - value: 引数値. スカラー, ベクトルの即値, 四則演算などが設定可能

○遷移定義(transitions)

- source_id: 遷移元状態 Id
- target_id: 遷移先状態 Id
- gurad: ガード条件

○付属ファイル(files)

- name: ファイル名

○画像ファイル(images)

- name: ファイル名

```
-
  taskName: "Slider insertion"
  comment: "insert yellow block by sliding it
along the rail."
  models:
    -
      name: "block"
      rname: "blockR"
      type: Work
      file_name: "slider_block1Hrp.wrl"
      pos_x: 0.35
      pos_y: 0.3
      pos_z: 0.81
      rot_x: 0
      rot_y: 0
      rot_z: 0
  states:
    -
      id: 1
      type: 1
      pos_x: -47
      pos_y: -183
    -
      id: 2
      type: 5
      cmd_name: "moveLeftGripper"
      pos_x: 31
      pos_y: -127
      arguments:
        -
          seq: 1
          name: test01
          value: "[1.0, 2.0, 3.0] / 2 / 5"
        -
          seq: 2
          name: test02
          value: "(3.0 + 4.0) / 2.0 - 3.0"
  transactions:
    -
      source_id: 1
      target_id: 2
  parameters:
    -
      name: "block pos"
      rname: "block_pos"
      type: 1
      units: ""
      elem_num: 6
      elem_type: "0,0,0,0,0,0"
      model_name: "block"
      values: "0.35, 0.3, 0.81, 0, 0, 0"
    -
      name: "insert length"
      rname: "insert_length"
      type: 0
      units: ""
      elem_num: 1
      elem_type: "0"
      values: "0.12"
  files:
    -
      name: "slide_rail_spec.pdf"
  images:
    -
      name: "slider_insertion2.png"
    -
      name: "slider_insertion1.png"
```

3.3 設定ファイル

Teaching Plugin で使用する各種設定を記述したファイルです。Choreonoid の実行ファイル (choreonoid.exe)と同じディレクトリに存在します。

```
database : teachingDB.sqlite3
robotModelName : main_withHands
application : PDF|C:\Program Files (x86)\Adobe\Reader 11.0\Reader\AcroRd32.exe
application : PPTX|C:\Program Files (x86)\Microsoft Office\Office14\POWERPNT.EXE
```

○database

Teaching Plugin で使用するデータベースファイルを指定します。

○robotModelName

対象プロジェクト内で使用しているロボットモデルの名称を指定します。

○application

メタデータビュー内に登録した付属文書を表示するためのアプリケーションを登録します。対象ファイルの拡張子と対応するアプリケーションを「|」区切りで指定します。application 行は複数行記述することが可能です。