

# ST-Bridge→Revit 変換を例とした Dynamo の検証

ソリューション本部 建設事業部 東日本システム開発部

小平 進

## 1. はじめに

本稿では、ビジュアルプログラミングである Dynamo for Revit<sup>※1</sup>(以下、Dynamo という)を利用して、建築構造解析のデータ交換フォーマット ST-Bridge<sup>※2</sup>データから Revit モデルへの変換を試みることで、Dynamo の技術及び利用可能領域の検証を行う。

ST-Bridge データを Revit モデルに変換する Revit プラグインは、「ST-Bridge Link 2017」としてすでにオートデスク株式会社より提供されているが、本稿では、Dynamo の標準ノードを利用して同様のプロセスを検証することとする。

なお、検証には「ST-Bridge Link 2017」に付属の ST-Bridge データ SampleS.stb(S 造)と SampleRC.stb(RC 造)、並びに Revit ファミリを利用する。

## 2. 準備

### 2.1 ST-Bridge データの整理

ST-Bridge データを Dynamo で扱いやすい Excel データに変換する。

- ① 「<」「>」行を削除、「=」を「スペース」に変換する。
- ② 「スペース」区切りで Excel に読込む。

### 2.2 ファミリ対応テーブル作成

「ST-Bridge Link 2017」の共通マッピングテーブル「ConvBase2017\_1.xls」を参照して、ファミリ名を抽出するテーブルを作成する。項目以下のとおり。

- ① 「分類コード」項目:ST-Bridge 分類コード
- ② 「タイプ名」項目:ST-Bridge の「name」値
- ③ 「分類名」項目:分類名(柱、大梁など)
- ④ 「ファミリ名」項目:分類図形のファミリ名

### 2.3 パラメータ対応テーブル作成

「ST-BRIDGE XML ファイル仕様書(ver.1.4)」(Building Smart Japan 構造分科会)の属性一覧を参照して、分類図形(ファミリ)のパラメータに設定するテーブルを作成する。項目以下のとおり。

- ① 「分類コード」項目:ST-Bridge 分類コード
- ② 「分類名」項目:分類名
- ③ 「属性名」項目:ST-Bridge の項目名
- ④ 「型」項目:値のデータ型
- ⑤ 「パラメータ名」項目:Revit のパラメータ名

<sup>※1</sup> Revit のプラグインとしてオートデスクから提供されている。Autodesk Maya などの他のソフトウェアのプラグインとしても使用可能。

<sup>※2</sup> 日本国内の建築構造設計分野でのデータ交換用

フォーマット。IFC の弱点である鉄筋詳細情報などを補完し、構造解析アプリケーションと BIM ツールとのデータ交換の制度を高めることを目的としている。

## 2.4 ファミリ作成

本稿では、「ST-Bridge Link 2017」のファミリーを利用する。また利用に際しては以下の注意が必要である。

- ① ST-Bridge の断面情報は、タイプ パラメータに登録する。
- ② 位置情報及び図形状状などの情報は、インスタンス パラメータに登録する。

## 3. 変換プログラム

### 3.1 Dynamo による変換プログラムの概要

本稿では、Dynamo プログラムを部位毎に作成した。また、本プログラムは、データの読み込み、データ整理、図形作成、情報登録などの小プログラムに分かれており(「図 3-1 フロー図」参照)、各分類のプログラムで共有している。

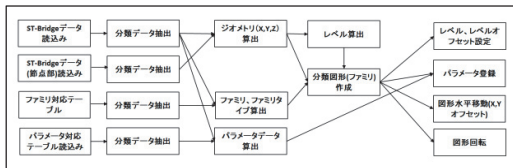


図 3-1 フロー図(全分類共通)

以下、構造柱(柱)を例として、Dynamo プログラムを説明する。

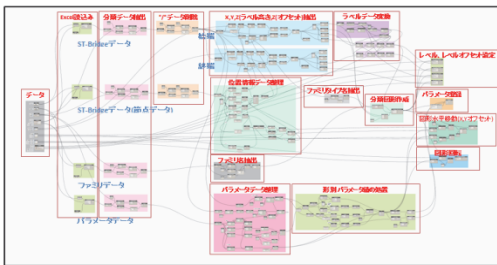


図 3-2 Dynamo プログラム(構造柱(柱))

### 3.2 プログラム説明

#### (1) Excel データ取込部

Excel で整理したデータを読み込む。

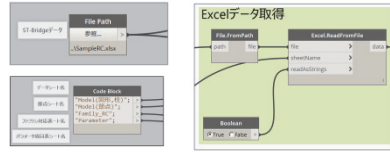


図 3-3 プログラム(Excel 読み込み)

対象の分類データを抽出する。

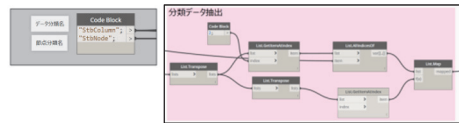


図 3-4 プログラム(分類データ抽出)

#### (2) ジオメトリ(X,Y,Z)算出

節点 ID で X,Y,Z を取得し、Z データを Z(レベル),Z(オフセット)に分ける。

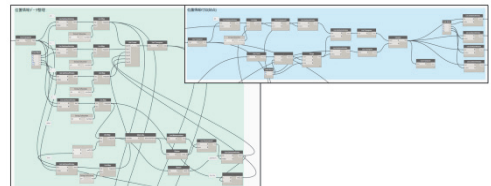


図 3-5 プログラム(ジオメトリ算出)

#### (3) レベル算出

Z(レベル)値→レベル名→レベルを算出する。

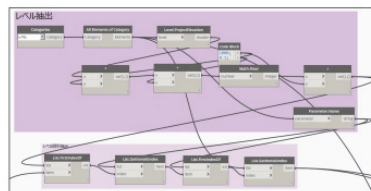


図 3-6 プログラム(レベル算出)

#### (4) ファミリ、ファミリータイプ算出

ST-Bridge データの「分類コード」+「name」で、ファミリー対応テーブルよりファミリー名を抽出する。

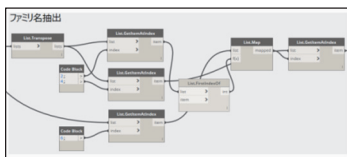


図 3-7 プログラム(ファミリー名算出)

ファミリタイプ名を ST-Bridge データの「name」より算出する。

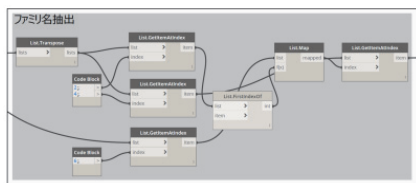


図 3-8 プログラム(ファミリタイプ名算出)

(5) 分類ファミリ作成

位置情報(X,Y)、ファミリ名、ファミリタイプ名、レベルで分類ファミリを作成する。

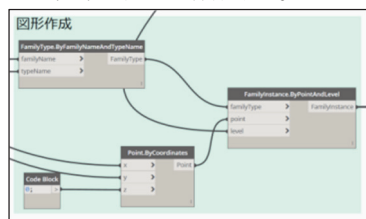


図 3-9 プログラム(分類ファミリ作成)

(6) パラメータデータ算出

パラメータテーブルより、分類図形に登録するパラメータを算出する。

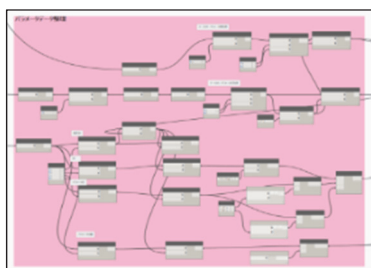


図 3-10 プログラム(パラメータデータ算出)

(7) パラメータデータ登録

分類図形に、パラメータデータを登録する。

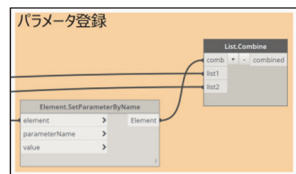


図 3-11 プログラム(パラメータデータ登録)

(8) 図形移動

「レベルのオフセット」パラメータに Z(オフセット)を登録して、図形を垂直に移動する。

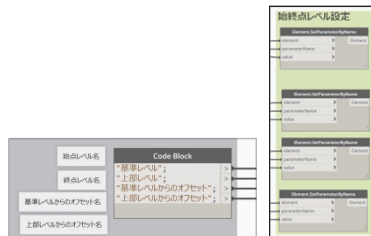


図 3-12 プログラム(レベルのオフセット)

オフセット(X,Y)で図形を水平に移動する。

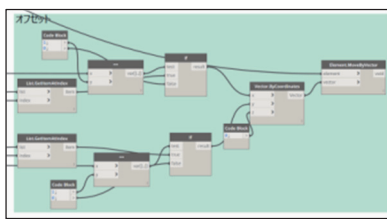


図 3-13 プログラム(X,Y のオフセット)

(9) 図形回転

ST-Bridge データの「rotate」を回転パラメータに設定する。

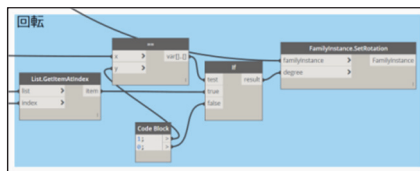


図 3-14 プログラム(図形回転)

4. 分類図形設定

分類により、位置情報、オフセット、回転などの設定が異なるので、下記に示す。

4.1 軸(レベル,通芯)

(1) レベル

項目	内容
分類コード	SttStory
位置情報	高さ[height]
レベル	ここでレベルを設定
オフセット(水平)	設定なし
オフセット(鉛直)	設定なし
回転	設定なし

## (2) 通芯

項目	内容
分類コード	StbX_Axis, StbY_Axis
位置情報	StbX_Axis : X=0からのX方向距離(distance) StbY_Axis : Y=0からのY方向距離(distance) ※ ST-Bridgeは、多入力だが、Dynamoでの過芯機能(標準機能)では、直線(2点入力)しかない
レベル	設定なし
オフセット(水平)	設定なし
オフセット(鉛直)	設定なし
回転	設定なし

## 4.2 構造基礎(フーチング,布基礎,杭基礎)

### (1) フーチング

項目	内容
分類コード	StbFooting
位置情報	1点入力:[dNode]
レベル	「節点情報」のZ値でレベル算出→「レベル」パラメータに設定
オフセット(水平)	X:[offset_X]→図形全体をX方向に移動移動 Y:[offset_Y]→図形全体をY方向に移動移動
オフセット(鉛直)	[level_bottom]→「レベル」からの高さオフセット「パラメータ」に設定
回転	[rotate]→図形全体を回転(軸:Z方向)

### (2) 布基礎

BT-Bridge サンプルデータに未記載のため、検証対象外とする。

### (3) 杭基礎

項目	内容
分類コード	StbPile
位置情報	1点入力:[dNode]
レベル	「節点情報」のZ値でレベル算出→「レベル」パラメータに設定
オフセット(水平)	X:[offset_X]→図形全体をX方向に移動 Y:[offset_Y]→図形全体をY方向に移動
オフセット(鉛直)	[level_top]→「レベル」からの高さオフセット「パラメータ」に設定
回転	回転なし

## 4.3 柱(柱,間柱)

### (1) 柱

項目	内容
分類コード	StbColumn
位置情報	1点入力:[dNode_bottom] ※ ST-Bridgeは多入力だが、Revitでは1点入力 ※ [dNode_top]の値は、「上部レベル」パラメータで使用
レベル	始端「節点情報」のZ値でレベル算出→「基準レベル」パラメータに設定 終端「節点情報」のZ値でレベル算出→「上部レベル」パラメータに設定
オフセット(水平)	X:[offset_X]→図形全体を移動 Y:[offset_Y]→図形全体を移動 始端X:[offset_bottom_X]→始端側をX方向に移動 始端Y:[offset_bottom_Y]→始端側をY方向に移動 終端X:[offset_top_X]→終端側をX方向に移動 終端Y:[offset_top_Y]→終端側をY方向に移動 ※ ST-Bridgeデータ列に始端X,Y、終端X,Yの記載がないため、未対応
オフセット(鉛直)	始端[Offset_bottom_Z]→「基準レベル」からのオフセット「パラメータ」に設定 終端[Offset_top_Z]→「上部レベル」からのオフセット「パラメータ」に設定 ※ ST-Bridgeデータ列に始端X,Y、終端X,Yの記載がないため、未対応
回転	[rotate]→図形全体を回転(軸:Z方向)

### (2) 間柱

項目	内容
分類コード	StbGirder
位置情報	2点入力:[dNode_start],[dNode_end]
レベル	始端「節点情報」のZ値でレベル算出→算出レベルで作図
オフセット(水平)	[offset]→「Yオフセット」値「パラメータ」に設定 ※ 始端から終端方向+90° 方向に移動 始端X:[offset_start_X]→始端側をX方向に移動 始端Y:[offset_start_Y]→始端側をY方向に移動 終端X:[offset_end_X]→終端側をX方向に移動 終端Y:[offset_end_Y]→終端側をY方向に移動
オフセット(鉛直)	[level]→「Zオフセット」値「パラメータ」に設定(全体をZ方向に移動) 始端[Offset_start_Z] + 始端「節点情報」のオフセット 値→「始端レベル オフセット」パラメータに設定(始端Z方向に移動) 終端[Offset_end_Z] + 終端「節点情報」のオフセット 値→「終端レベル オフセット」パラメータに設定(終端Z方向に移動)
回転	[rotate]→「断面回転」パラメータに設定(軸:始端から終端方向)

## 4.4 梁(大梁,小梁,ブレース)

### (1) 大梁

項目	内容
分類コード	StbGirder
位置情報	2点入力:[dNode_start],[dNode_end]
レベル	始端「節点情報」のZ値でレベル算出→算出レベルで作図
オフセット(水平)	[offset]→「Yオフセット」値「パラメータ」に設定 ※ 始端から終端方向+90° 方向に移動 始端X:[offset_start_X]→始端側をX方向に移動 始端Y:[offset_start_Y]→始端側をY方向に移動 終端X:[offset_end_X]→終端側をX方向に移動 終端Y:[offset_end_Y]→終端側をY方向に移動
オフセット(鉛直)	[level]→「Zオフセット」値「パラメータ」に設定(全体をZ方向に移動) 始端[Offset_start_Z] + 始端「節点情報」のオフセット 値→「始端レベル オフセット」パラメータに設定(始端Z方向に移動) 終端[Offset_end_Z] + 終端「節点情報」のオフセット 値→「終端レベル オフセット」パラメータに設定(終端Z方向に移動)
回転	[rotate]→「断面回転」パラメータに設定(軸:始端から終端方向)

### (2) 小梁

項目	内容
分類コード	StbColumn
位置情報	1点入力:[dNode_bottom] ※ ST-Bridgeは多入力だが、Revitでは1点入力 ※ [dNode_top]の値は、「上部レベル」パラメータで使用
レベル	始端「節点情報」のZ値でレベル算出→「基準レベル」パラメータに設定 終端「節点情報」のZ値でレベル算出→「上部レベル」パラメータに設定
オフセット(水平)	X:[offset_X]→図形全体を移動 Y:[offset_Y]→図形全体を移動 始端X:[offset_bottom_X]→始端側をX方向に移動 始端Y:[offset_bottom_Y]→始端側をY方向に移動 終端X:[offset_top_X]→終端側をX方向に移動 終端Y:[offset_top_Y]→終端側をY方向に移動 ※ ST-Bridgeデータ列に始端X,Y、終端X,Yの記載がないため、未対応
オフセット(鉛直)	始端[Offset_bottom_Z]→「基準レベル」からのオフセット「パラメータ」に設定 終端[Offset_top_Z]→「上部レベル」からのオフセット「パラメータ」に設定 ※ ST-Bridgeデータ列に始端X,Y、終端X,Yの記載がないため、未対応
回転	[rotate]→図形全体を回転(軸:Z方向)

### (3) ブレース

項目	内容
分類コード	StbBrace
位置情報	2点入力:[dNode_start],[dNode_end]
レベル	始端「節点情報」のZ値でレベル算出→算出レベルで作図
オフセット(水平)	[offset]→「Yオフセット」値「パラメータ」に設定 ※ 始端から終端方向+90° 方向に移動 始端X:[offset_start_X]→始端側をX方向に移動 始端Y:[offset_start_Y]→始端側をY方向に移動 終端X:[offset_end_X]→終端側をX方向に移動 終端Y:[offset_end_Y]→終端側をY方向に移動
オフセット(鉛直)	[level]→「Zオフセット」値「パラメータ」に設定(全体をZ方向に移動) 始端[Offset_start_Z] + 始端「節点情報」のオフセット 値→「始端レベル オフセット」パラメータに設定(始端Z方向に移動) 終端[Offset_end_Z] + 終端「節点情報」のオフセット 値→「終端レベル オフセット」パラメータに設定(終端Z方向に移動)
回転	[rotate]→「断面回転」パラメータに設定(軸:始端から終端方向)

## 4. 5 床(スラブ)

### (1) スラブ

項目	内容
分類コード	StbSlab
位置情報	多点入力:[StbNodeid]...→4点入力:[d1][d2][d3][d4] ※ 本検証では、全てのデータが4点入力のため、4点入力として作成:[d1][d2][d3][d4]
レベル	端点[d1]より「節点情報」のZ値よりレベル算出→算出しレベルで作図
オフセット(水平)	端点[X:[offset_X_1]→端点1をX方向に移動 端点[Y:[offset_Y_1]→端点1をY方向に移動 端点2[X:[offset_X_2]→端点2をX方向に移動 端点[Y:[offset_Y_2]→端点2をY方向に移動 端点3[X:[offset_X_3]→端点3をX方向に移動 端点[Y:[offset_Y_3]→端点3をY方向に移動 端点4[X:[offset_X_4]→端点4をX方向に移動 端点[Y:[offset_Y_4]→端点4をY方向に移動 ※ 本検証では、全てのデータが4点入力のため、4点入力として作成: [offset_X_1][offset_X_2][offset_X_3][offset_X_4][offset_Y_1][offset_Y_2][offset_Y_3][offset_Y_4]
オフセット(鉛直)	[level]→レベルからの高さオフセットパラメータに設定全体をZ方向に移動 端点[d1]より「節点情報」のオフセット 値→端点1をZ方向に移動 端点[d2]より「節点情報」のオフセット 値→端点2をZ方向に移動 端点[d3]より「節点情報」のオフセット 値→端点3をZ方向に移動 端点[d4]より「節点情報」のオフセット 値→端点4をZ方向に移動
回転	回転なし

## 4. 6 壁(壁,パラペット)

### (1) 壁

項目	内容
分類コード	StbWall
位置情報	多点入力:[StbNodeid]... → 2点入力:[d1][d2] ※ 本検証では、全てのデータが4点入力のため、4点入力として作成:[d1][d2][d3][d4] ※ ST-Bridgeの入力は、多点のポリゴンデータで作成だが、Revitでは、多点のライン図形+高さ(又はレベル)で作図であるため、本検証では、端点[d1]と端点[d2]で作成
レベル	端点[d1]より「節点情報」のZ値でレベル算出→「レベル」として作図 端点[d2]より「節点情報」のZ値→端点3値-端点1値の高さ値として作図
オフセット(水平)	[offset]→Xオフセット 値とYオフセット 値を算出して移動(始端から終端方向+90° 方向に移動)
オフセット(鉛直)	端点[d1]より「節点情報」のオフセット 値→始端をZ方向に移動 端点[d2]より「節点情報」のオフセット 値→終端をZ方向に移動
回転	回転なし

### (2) パラペット

BT-Bridge サンプルデータに未記載のため、検証対象外とする。

## 5. まとめ

### 5. 1 「ST-Bridge Link 2017」と「Dynamo」の変換結果比較

作成した図形を比較したところ、「5. 2 問題点」を除いて変換結果は、ほぼ同等であったことから、BIMのデータ交換ツールの作成において、十分な性能を持っているものとする。

### (1) S 造図形(SampleS.stb)

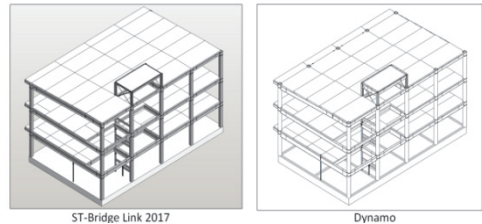


図 5-1 成果品比較(S 造図形)

### (2) RC 造図形(SampleRC.stb)

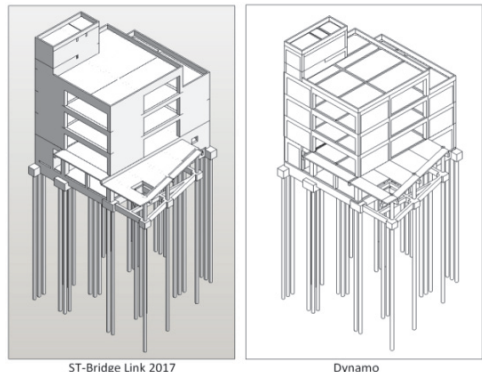


図 5-2 成果品比較(RC 造図形)

## 5. 2 問題点

### (1) 片持ち梁

Revit での片持ち梁の作成は根元→先端の順番が必要だが、ST-Bridge には順番の規定はない。変換の際には、片持ち梁などデータの始末端点の位置確認が必要である。

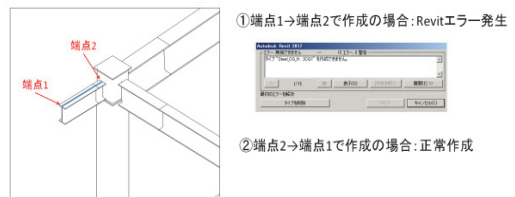


図 5-3 片持ち梁の問題点

### (2) 壁の作図方法

ST-Bridge データの壁の入力端点は、ポリゴン(X,Y,Z)での入力になるが、Revit の入力端点は、

レベル上のライン(X,Y)での入力になる。そのため、Revit への変換には、端点の加工が必要になる。

※ 「Wall.ByFace」機能でポリゴン入力での作成は可能だが、正規の壁図形で作成されないため、Revit での変更などの更新ができない。

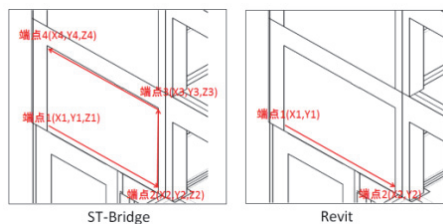


図 5-4 壁の作図方法の問題点

### (3) 床の傾き

Revit では勾配床を、「勾配矢印」機能などで作成するが、Dynamo の標準ノードでは勾配床の設定がないため、勾配のある床を作成できない。

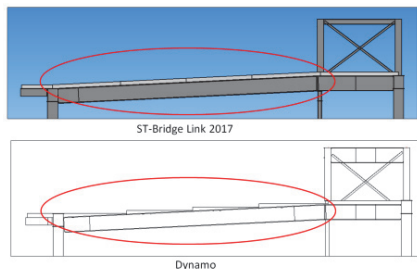


図 5-5 床の傾きの問題点

### (4) 開口部

Dynamo の標準ノードには、開口部の設定がないため、床、壁などの開口部の作成ができない。

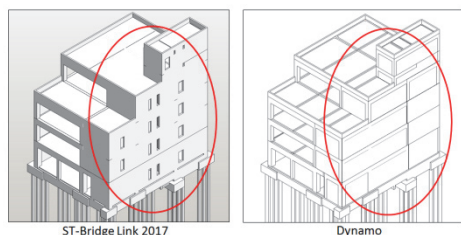


図 5-6 開口部の問題点

### (5) 折線通芯

Dynamo の標準ノードには、折線通芯の設定がないため、折線通芯の作成ができない。

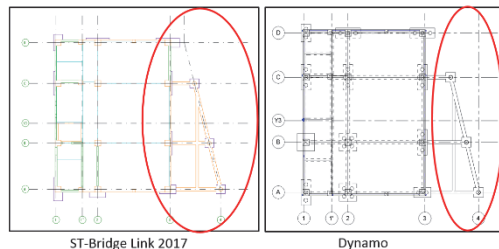


図 5-7 折線通芯の問題点

## 5.3 今後の課題

本稿では、Dynamo の標準ノード(関数)の利用で限定したが、独自のノード(カスタムノード)の開発も可能であるため現在の Dynamo の弱点を補う手段は構築可能と考える。

また、カスタムノードを提供している海外のサイトを見つけることはできるが、日本国内において皆無であることから Dynamo の普及には、海外サイトの情報収集、整理などの対応も重要になると考える。

今後当社としてもカスタムノードの可能性を模索することで、Dynamo の利便性向上に寄与できると思われる。

### <使用資料>

- 1) ST-BRIDGE XML ファイル仕様書(ver.1.4) (Building Smart Japan 構造分科会) ST\_BRIDGE\_XML 仕様説明書(ver.1.4)\_160601.pdf

### <使用ソフト>

- 1) Autodesk Revit 2017 オートデスク(株)
- 2) Dynamo 3.1.3 for Revit オートデスク(株)
- 3) ST-Bridge Link 2017 オートデスク(株)