

# オフィスソフトを操作するための RTC群

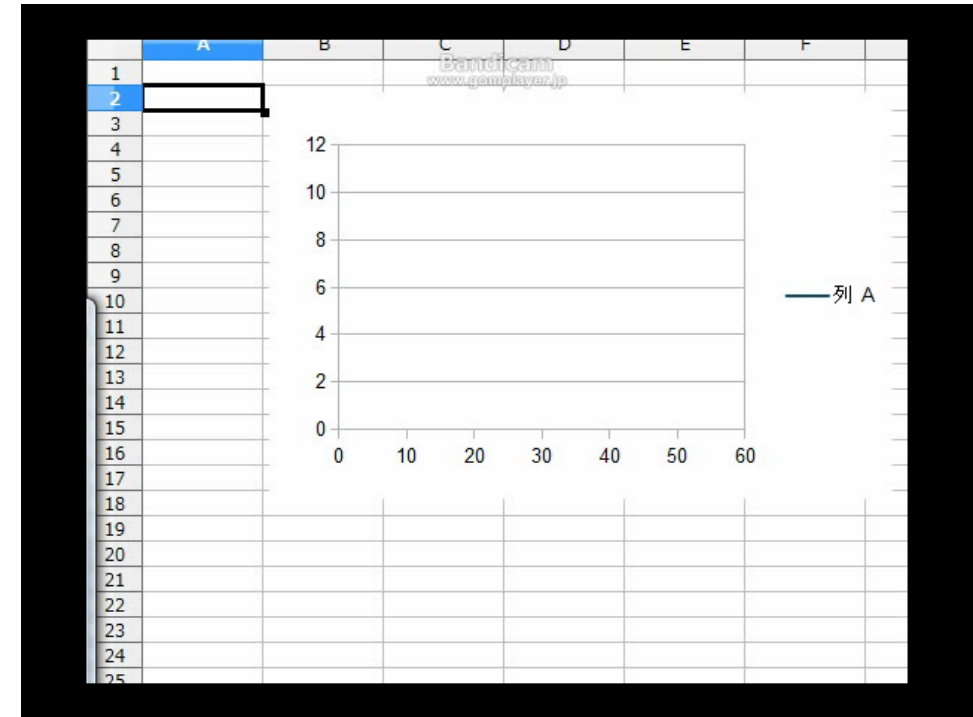
宮本 信彦

# 開発概要

- Microsoft Office、OpenOfficeの表計算機能、文書作成機能等をRTミドルウェアを用いたシステムで利用するためのRTC群
  - Excel、Calc(表計算機能)を操作するRTC
    - セルのデータをデータポートから入出力
  - Draw(図形描画)を操作するRTC
    - 図形的位置、姿勢をデータポートから入出力
  - Base(データベース)を操作するRTC
    - サービスポートを利用することでデータベースを操作
  - PowerPoint、Impress(プレゼンテーション)を操作するRTC
    - 上映中のスライドの番号の変更、線の描画
  - Word、Writer(文字表示)を操作するRTC
    - データポートで受信した文字列を文書に入力

# 開発概要

- InPortで受信したデータをセルに入力する例



- OpenOffice、Microsoft OfficeのドキュメントをRTCで操作する

# OpenOfficeを操作するためのRTC群

OpenOffice	Apache OpenOffice 3.4.1 Apache OpenOffice 4.1.1 OpenOffice.org 3.2 LibreOffice 3.5
OS	Windows Vista Windows 8.1 Ubuntu 10.04LTS Ubuntu 12.04LTS
言語	Python
RTミドルウェア	OpenRTM-aist-Python-1.1.0-RC1


- 5種類のコンポーネントを作成
  - Calc(表計算)、Draw(図形描画)、Base(データベース)、Impress(プレゼンテーション)、Writer(文書作成)

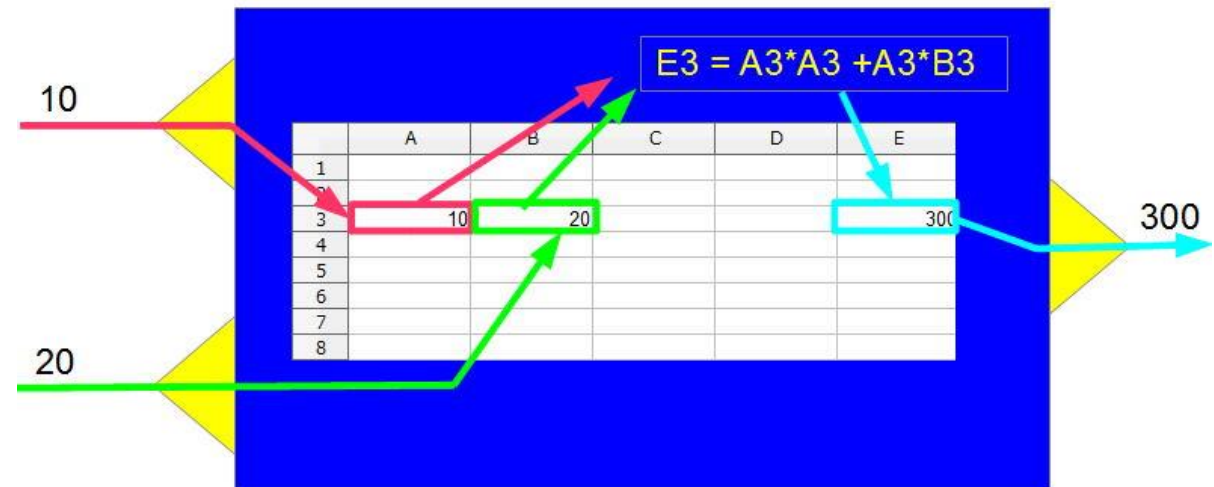
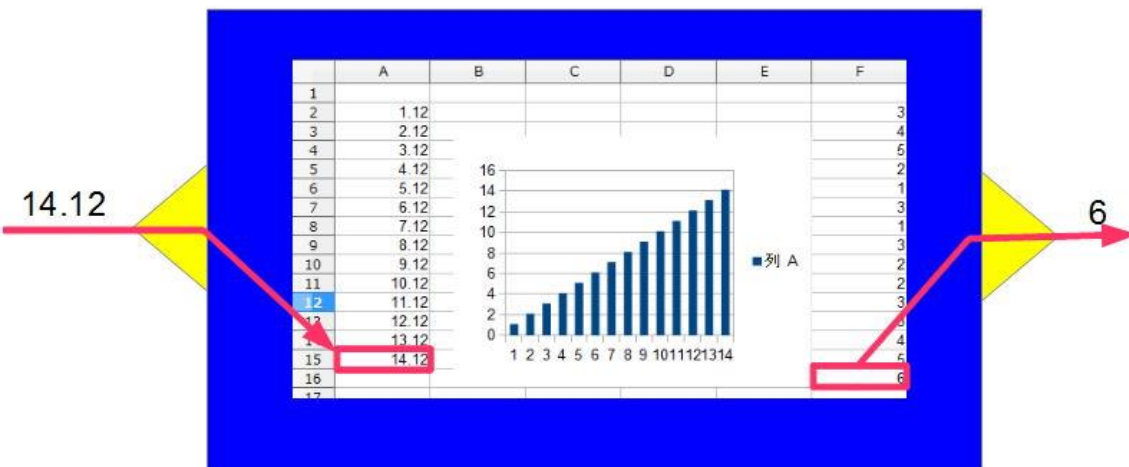
# Microsoft Officeを操作するためのRTC群

Microsoft Office	Word 2007 Word 2013 Excel 2007 Excel 2013 PowerPoint 2013
OS	Windows Vista Windows 8.1
言語	C++/CLI
RTミドルウェア	OpenRTM-aist-1.1.0-Release

- 4種類のコンポーネントを作成
  - Excel(表計算)、PowerPoint(プレゼンテーション)、Word(文書作成)、MODI(文字認識)

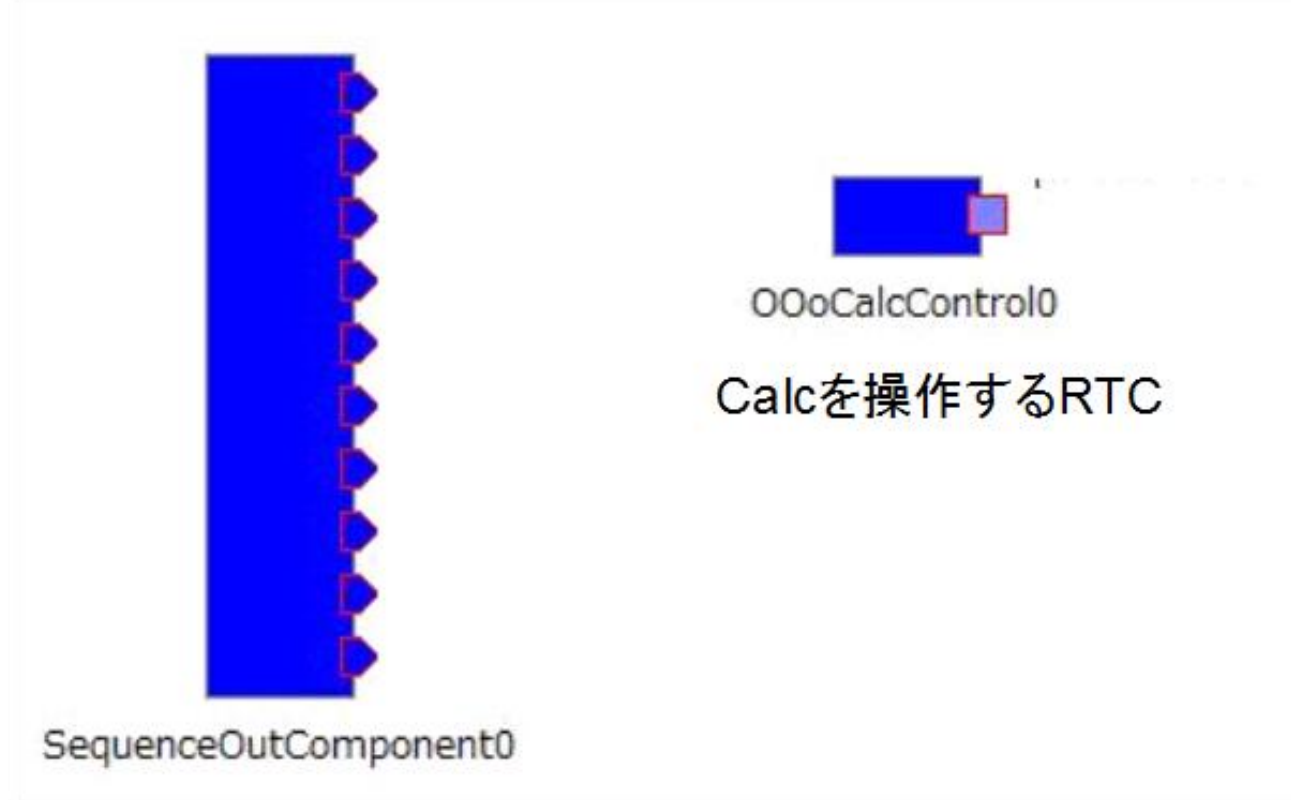
# Excel、Calc(表計算)を操作するためのRTC

- InPortで受信したデータをセルに入力  指定位置のセルを操作するサービス
- セルのデータをOutPortから出力
- InPortで受信したデータを数式で計算後OutPortから出力
- データポートは動的に追加可能
  - BasicDataType.idl、ExtendedDataTypes.idlで定義されたデータ型に対応
  - GUIによる操作、もしくはコンフィギュレーションパラメータで追加するデータポートの設定可能



# Excel、Calc(表計算)を操作するためのRTC

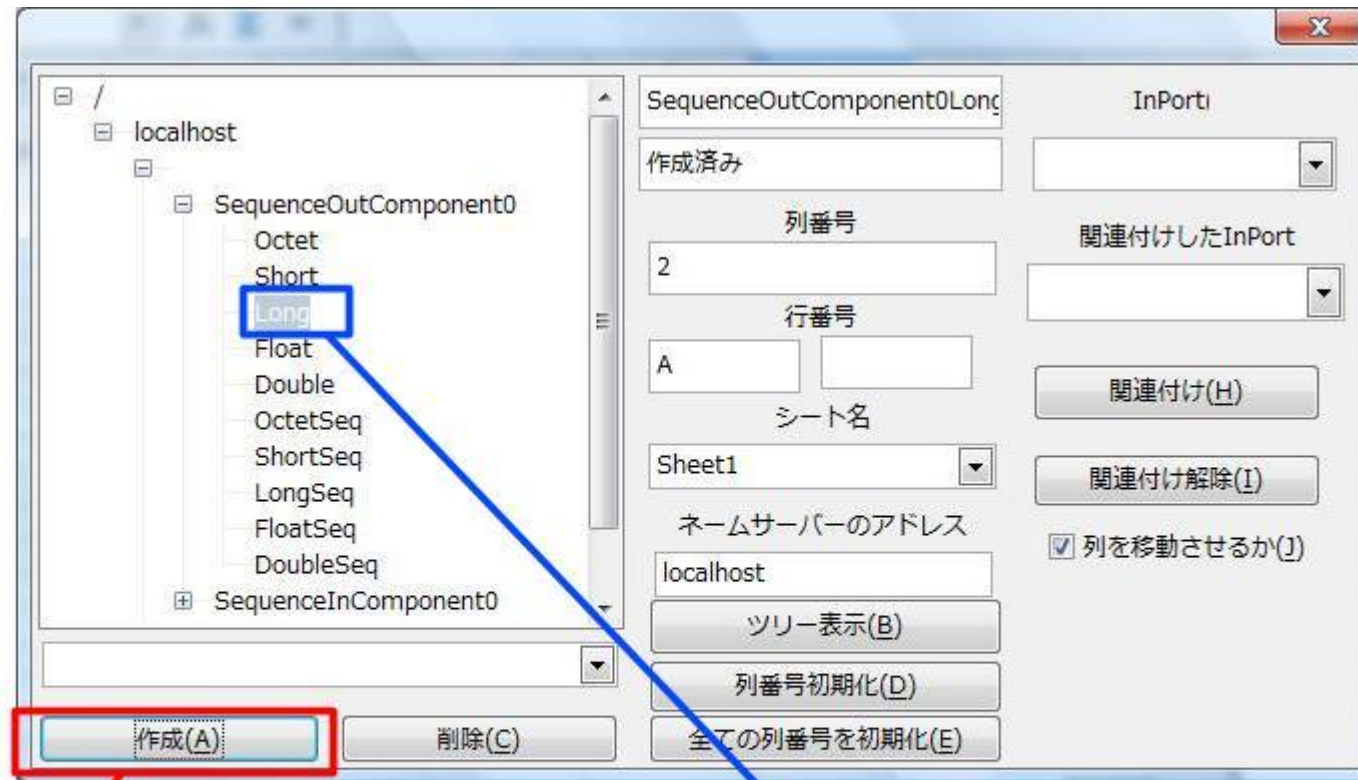
- データポートの追加の手順



テスト用RTC

# Excel、Calc(表計算)を操作するためのRTC

- GUIを操作することでセルのデータを入出力するデータポートを追加

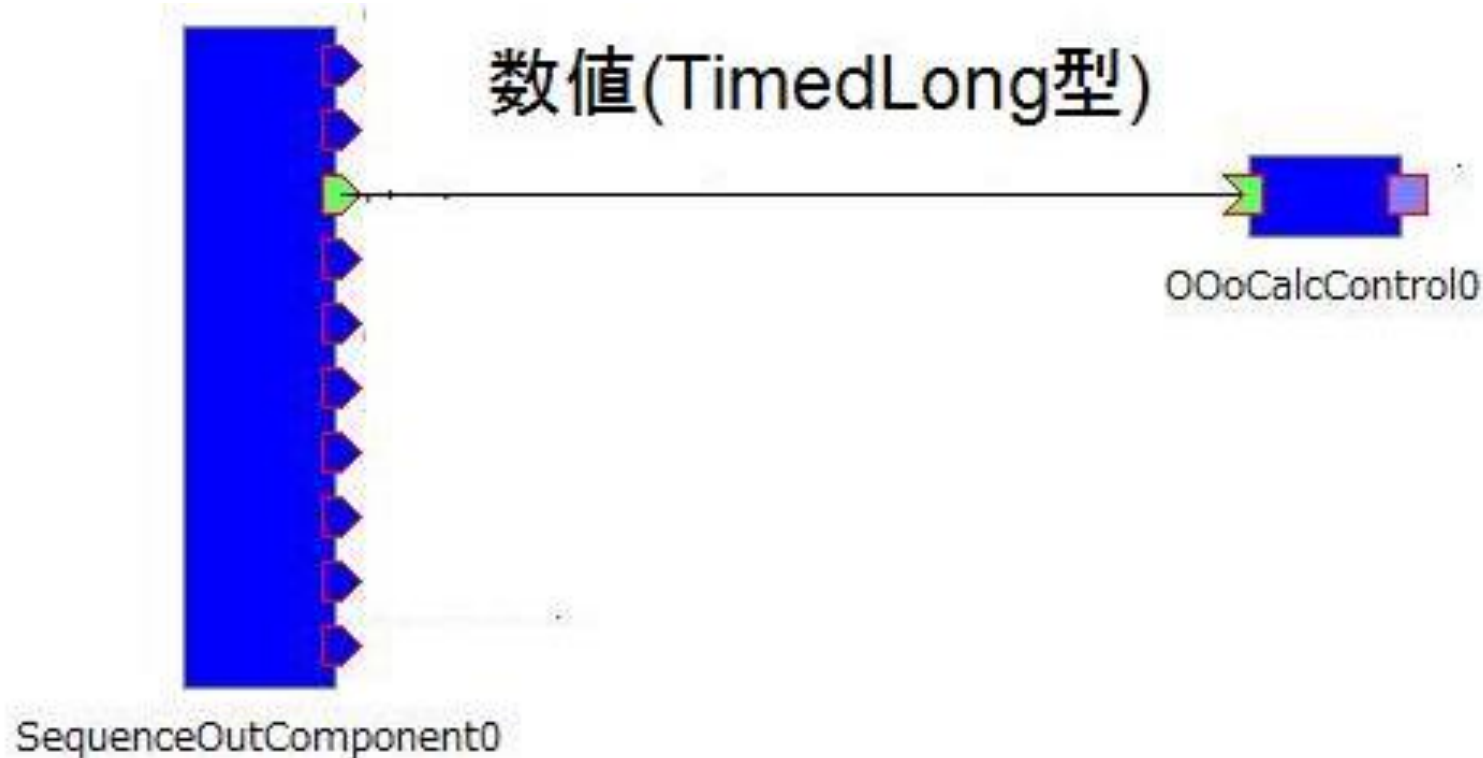


2: データポート作成ボタンを押す 1: ツリーからデータポートを選択



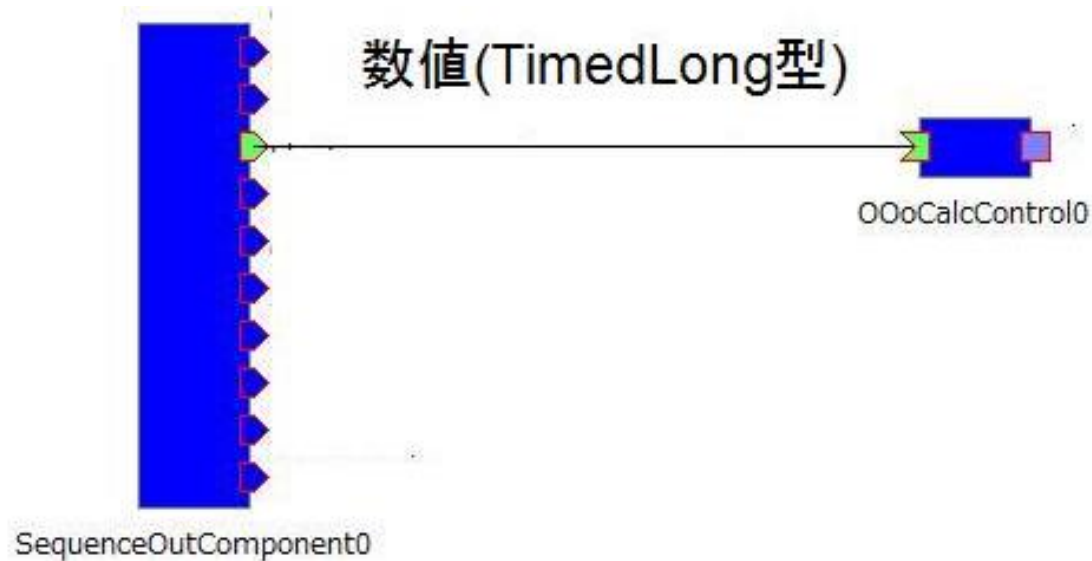
# Excel、Calc(表計算)を操作するためのRTC

- 以上の手順でデータポートの生成、接続ができる



# Excel、Calc(表計算)を操作するためのRTC

- InPortで受信したデータをセルに入力



	A	B
1		
2	1	
3	2	
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		

```
SeqOutComp.exe
5: 6.345 6.12 6 6 6 [ ]
6: 7.345 7.12 7 7 7 [ ]
7: 8.345 8.12 8 8 8 [ ]
8: 9.345 9.12 9 9 9 [ ]
9: 10.345 10.12 10 10 10 [ ]

write data type : serial
-: 2.345 2.12 2 2 2 [ ]

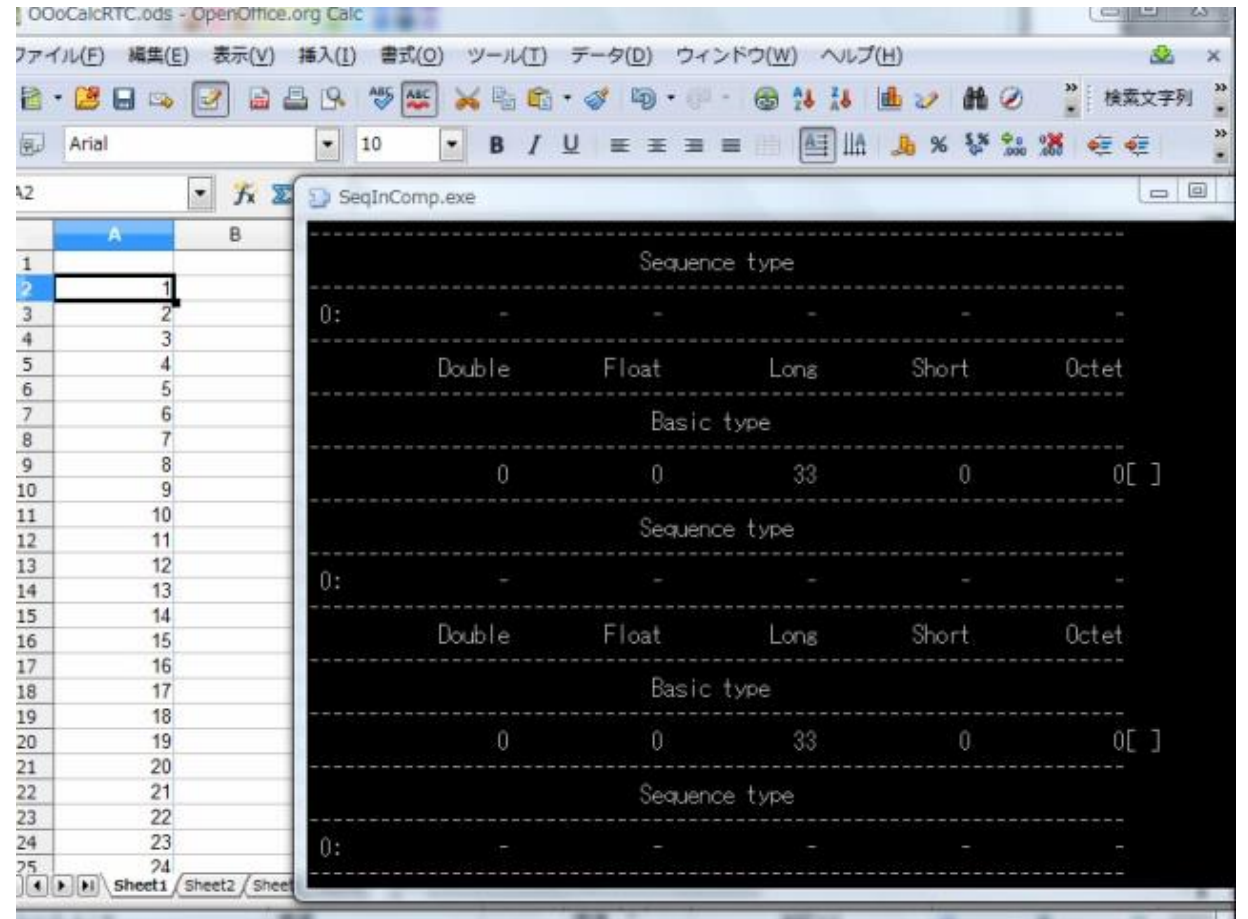
0: 11.345 11.12 11 11 11 [ ]
1: 12.345 12.12 12 12 12 [ ]
2: 13.345 13.12 13 13 13 [ ]
3: 14.345 14.12 14 14 14 [ ]
4: 15.345 15.12 15 15 15 [ ]
5: 16.345 16.12 16 16 16 [ ]
6: 17.345 17.12 17 17 17 [ ]
7: 18.345 18.12 18 18 18 [ ]
8: 19.345 19.12 19 19 19 [ ]
9: 20.345 20.12 20 20 20 [ ]

write data type : serial
-: 3.345 3.12 3 3 3 [ ]

0: 21.345 21.12 21 21 21 [ ]
```

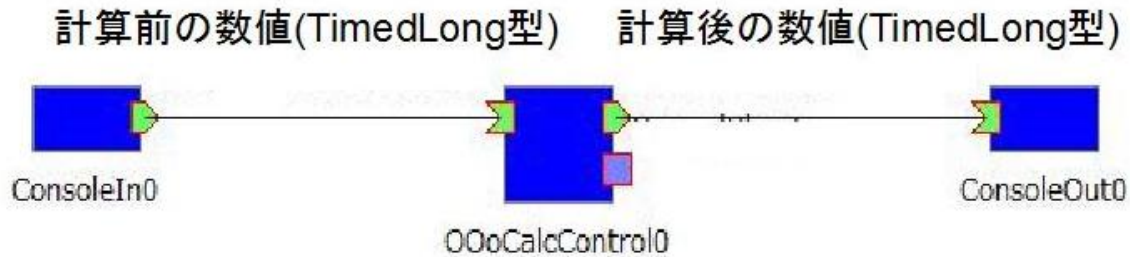
# Excel、Calc(表計算)を操作するためのRTC

- セルの値をOutPortから出力



# Excel、Calc(表計算)を操作するためのRTC

- InPortで受信したデータを数式で計算後OutPortから出力
  - 以下の例は入力した値を3倍にして出力する単純な計算式です



OpenOffice Calc window showing a spreadsheet with the following data:

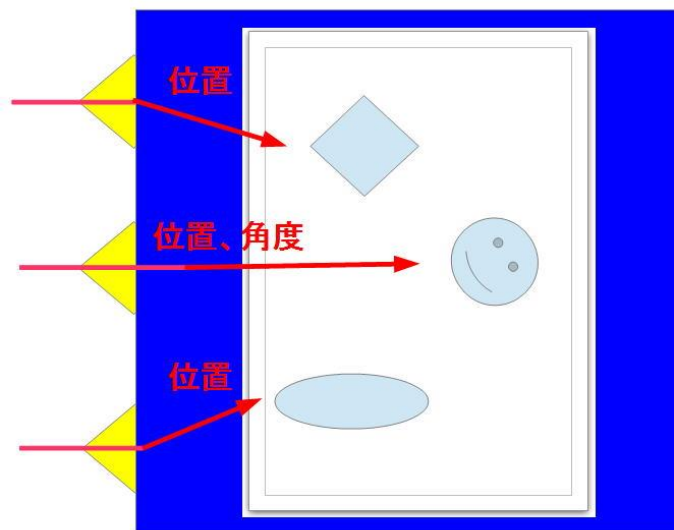
InPortで受信したデータ	計算後の数値
1	3
2	1
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	

Terminal window (ConsoleOutComp.exe) output:

```
- provider:
-----
Data Listener: ON_SEND
Profile::name: ConsoleIn0out
Profile::id: b0bbb2c0-ad58-4526-9ea7-38506990ca4b
Data: 2
-----
Data Listener: ON_RECEIVED
Profile::name: ConsoleOut0in
Profile::id: f69acede-788d-11e4-bf95-005056c00008
Data: 6
-----
Data Listener: ON_BUFFER_WRITE
Profile::name: ConsoleOut0in
Profile::id: f69acede-788d-11e4-bf95-005056c00008
Data: 6
-----
Received: 6
TimeStamp: 1417351226[ms] 174999000[ns]
```

# Draw(図形描画)を操作するためのRTC

- InPortへの入力により図形の位置を操作
  - 図形の位置をOutPortから出力する事も可能
- データポートは動的に追加可能
  - 14種類のデータ型に対応
  - GUIによる操作により追加するデータポートの設定可能



# Draw(図形描画)を操作するためのRTC

- データポートの追加の手順



SliderComp0

テスト用RTC

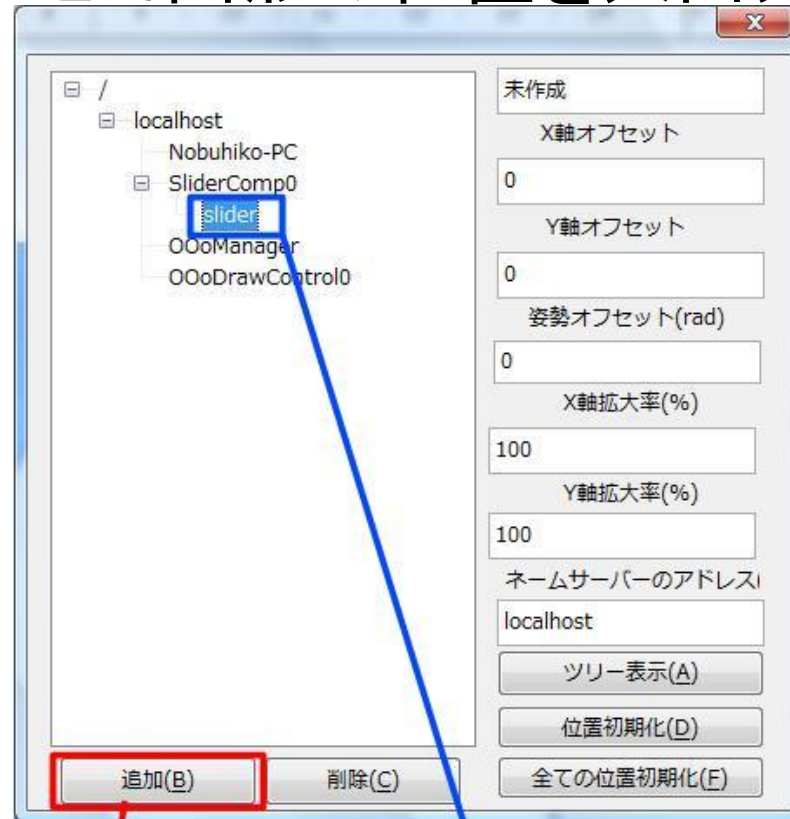


OOoDrawControl0

Drawを操作するRTC

# Draw(図形描画)を操作するためのRTC

- GUIを操作することで図形の入出力するデータポートを追加

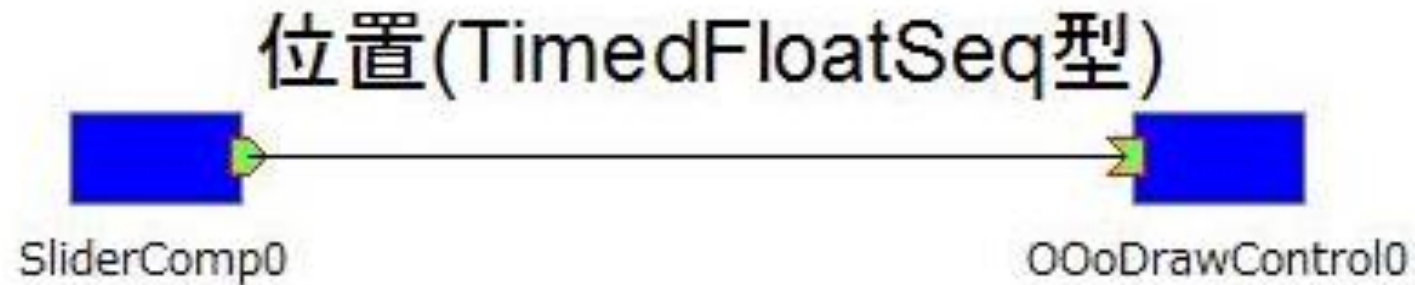


1: ツリーからデータポートを選択

2: データポート追加ボタンを押す

# Draw(図形描画)を操作するためのRTC

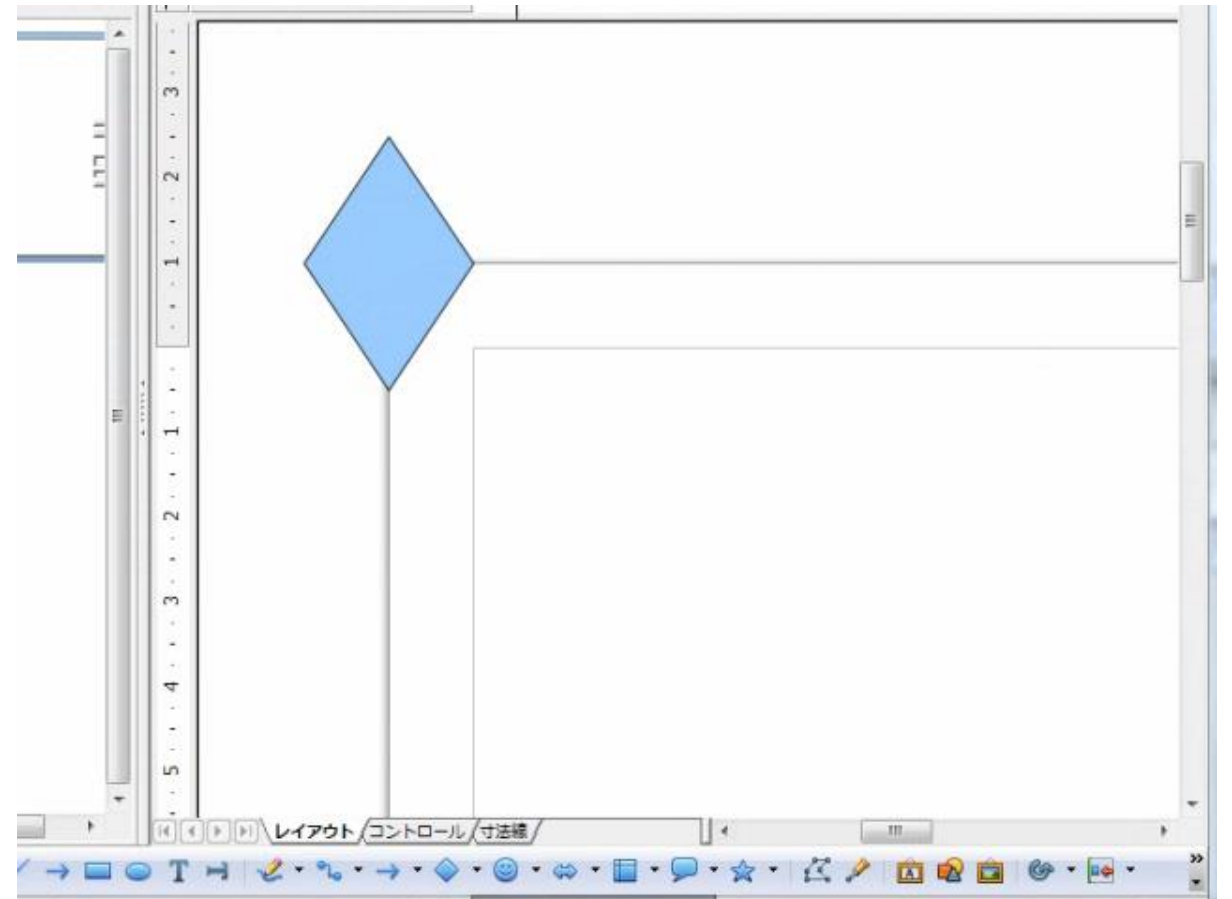
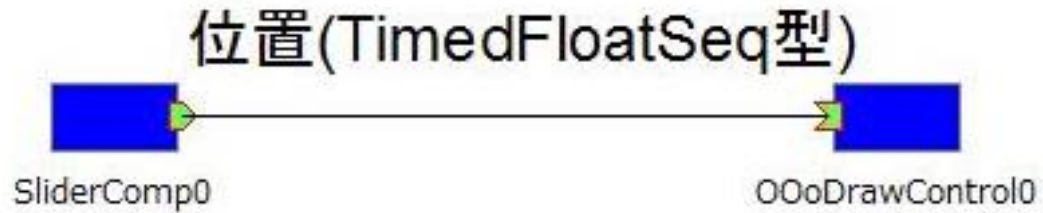
- 以上の手順でデータポートの生成、接続ができる





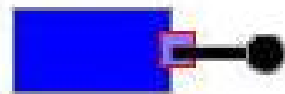
# Draw(図形描画)を操作するためのRTC

- InPortへの入力により図形の位置を操作



# Base(データベース)を操作するためのRTC

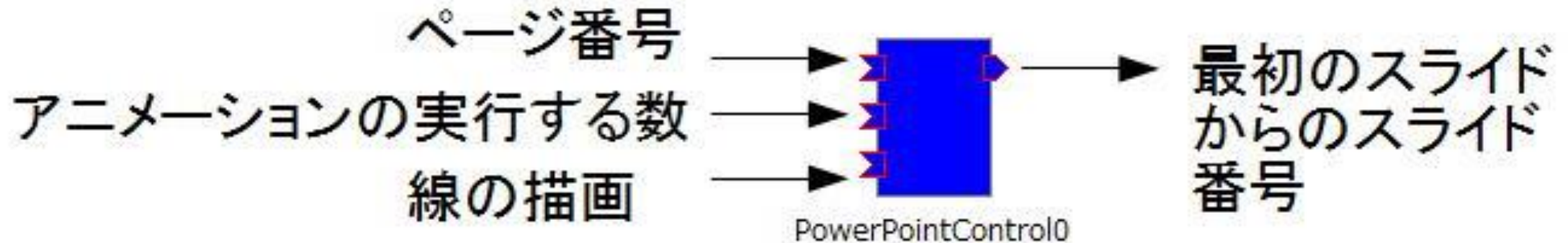
- 他のRTCはサービスポートを利用することでデータベースを操作できる
  - データベースへの接続
  - データベースへの問い合わせ(SQL文を入力)
  - データベースの追加、削除



データベースを操作するサービス

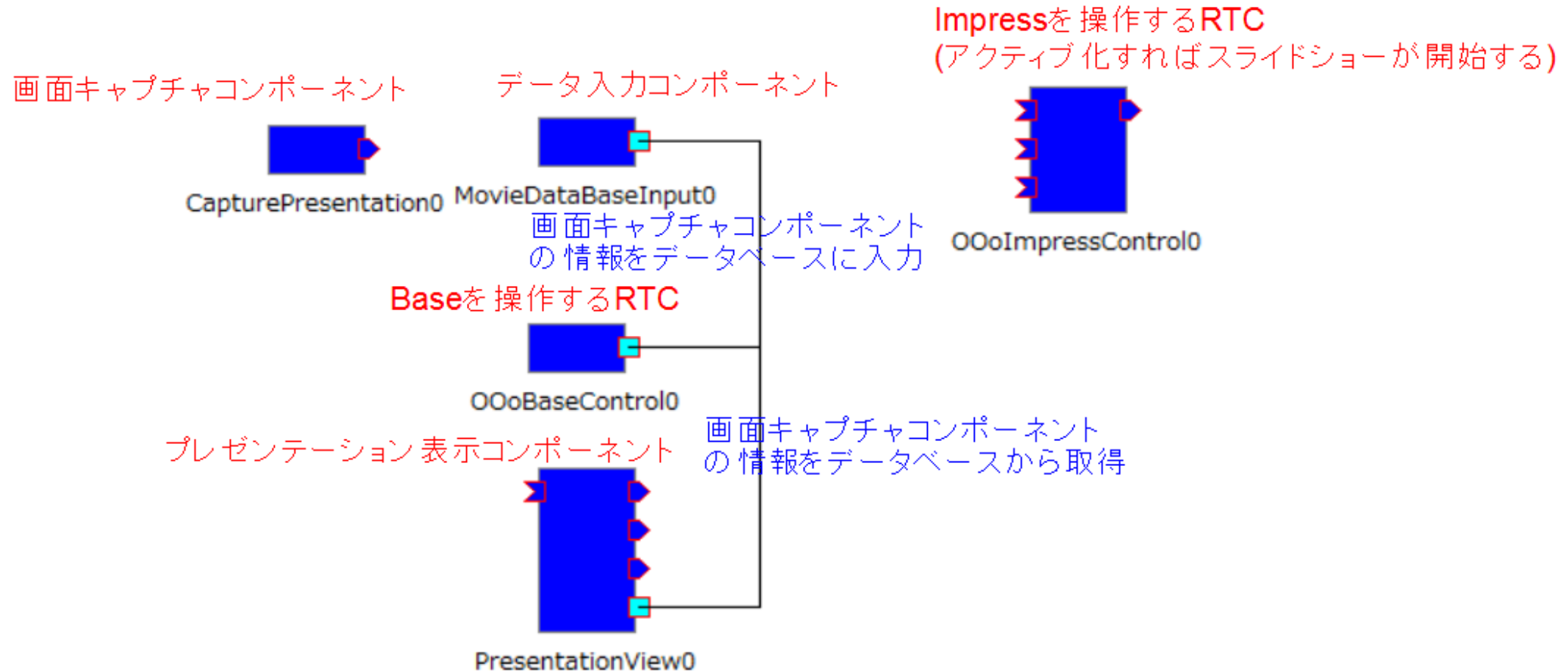
# PowerPoint、Impress(プレゼンテーション)を 操作するためのRTC

- 上映中のスライドの操作
  - スライド番号の変更
  - 設定したアニメーションの実行
  - 線の描画



# デモ1

- データベースにRTCのパスなどのデータを挿入
- そのデータを別のRTCで取得して、取得したRTCの情報を用いてデータポートを自動接続する



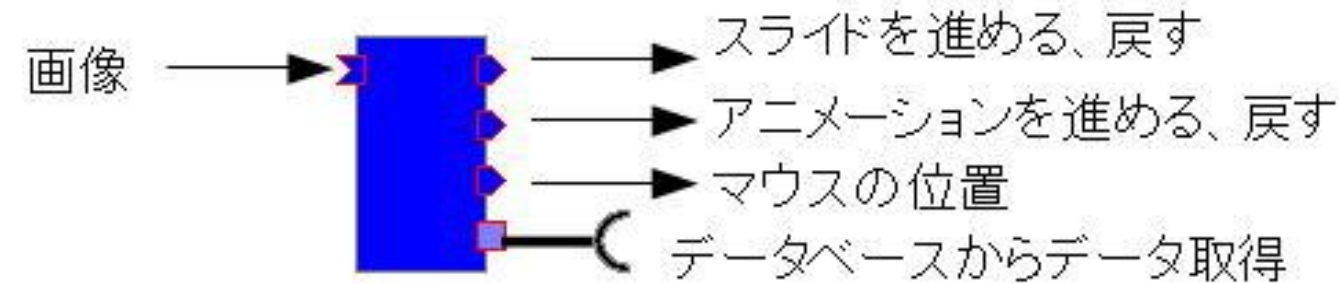
# 画面キャプチャコンポーネント

- 画面をキャプチャして画像データをOutPortから出力



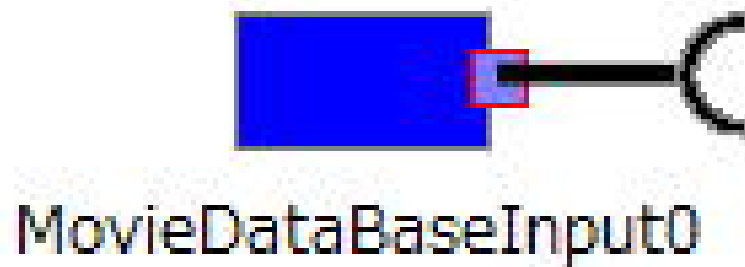
# プレゼンテーション表示コンポーネント

- InPortで受信した画像データの表示
- Baseを操作するRTCに接続することで、画面キャプチャコンポーネントが追加したデータを取得
- OutPortからの出力によりPowerPointを操作



# データ入力コンポーネント

- コンフィギュレーションパラメータで設定した情報をデータベースに挿入したい時に利用



データベースに  
データ入力

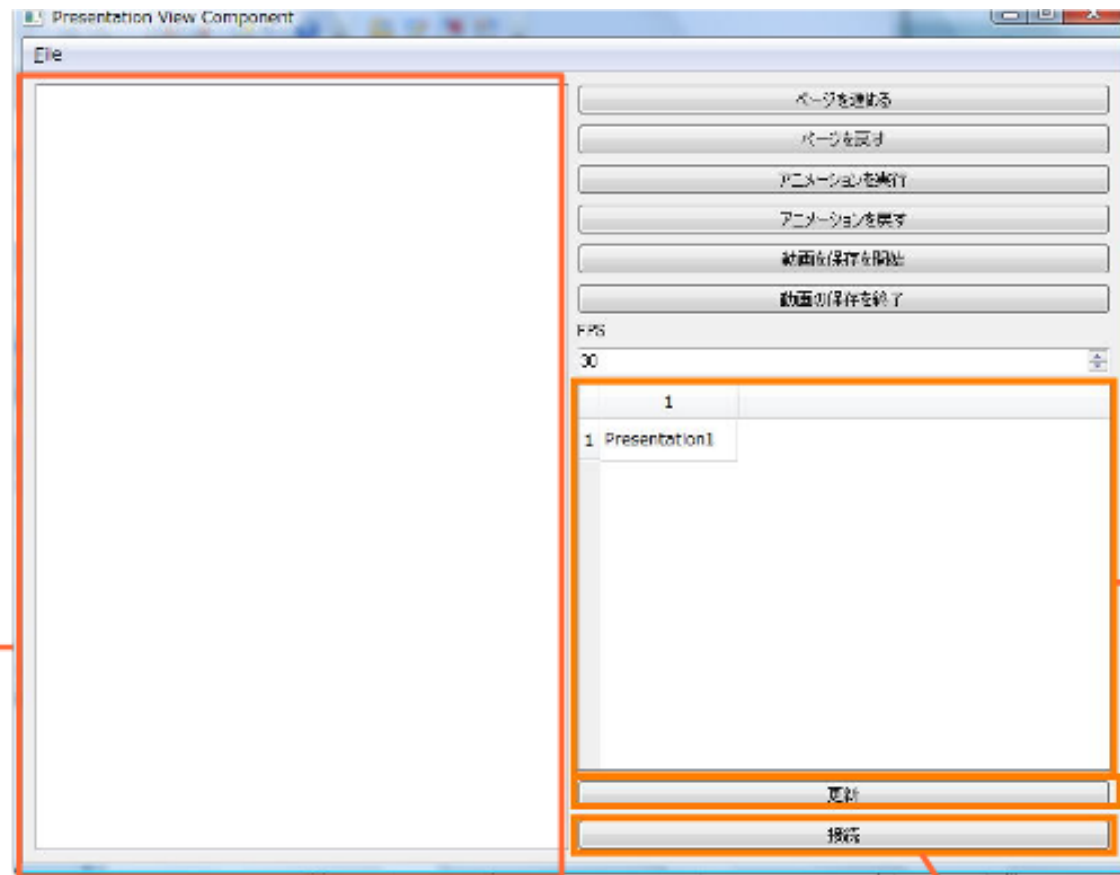
# デモ1

1. データ入力コンポーネントでデータベースにRTCのパス等のデータを挿入
  2. プレゼンテーション表示コンポーネントでデータベースからデータを取得
  3. 取得したRTCの情報からデータポートを自動接続
  4. スライドショーをキャプチャした画像がGUIに表示されるようになる
- 1はデータ入力コンポーネントをアクティブ化すればデータを挿入できる
  - 2・3・4はプレゼンテーション表示コンポーネントのGUIにより設定する



# デモ1

- プレゼンテーション表示コンポーネントのGUIを操作



4:この部分に受信した画像が表示される

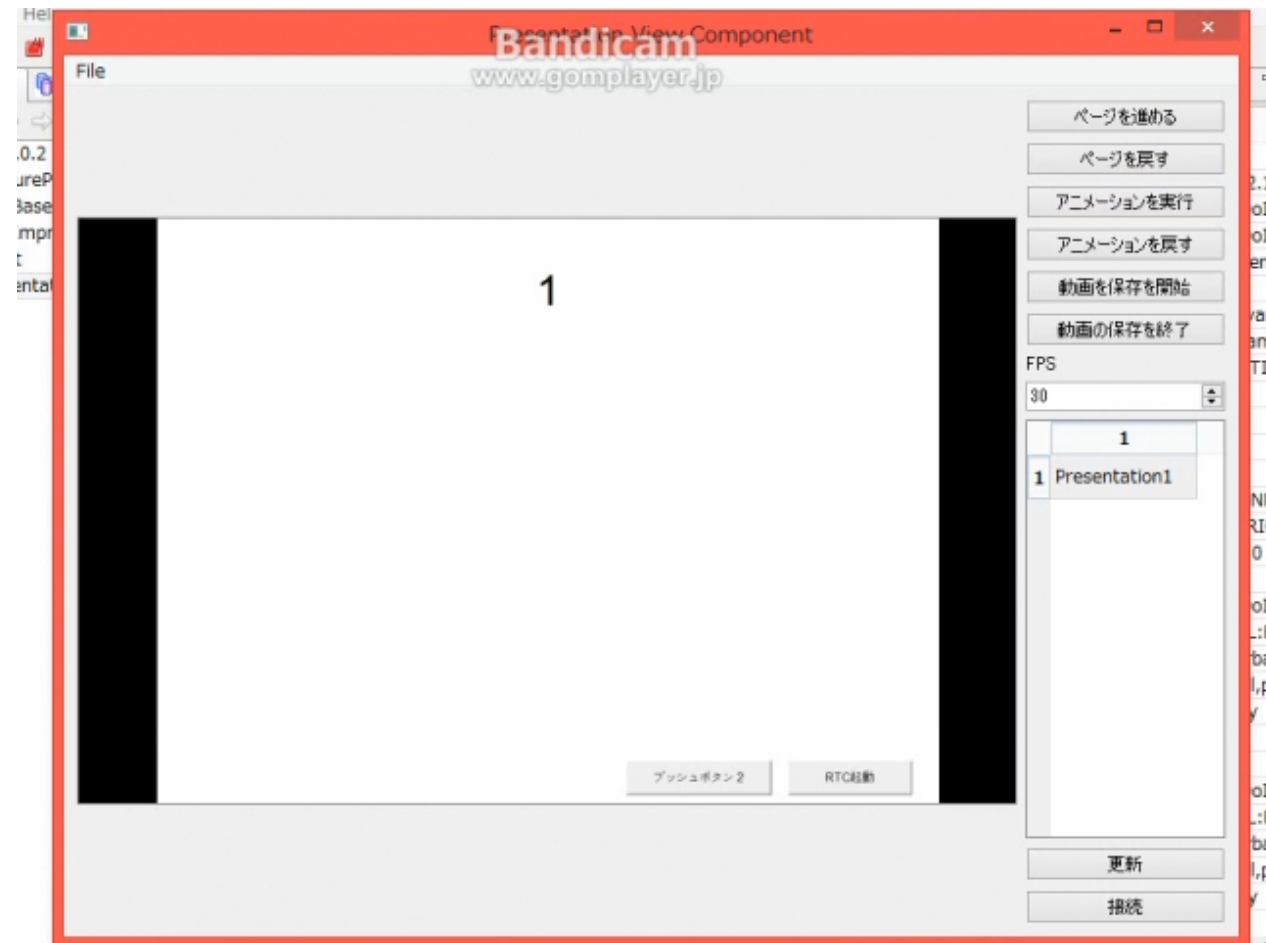
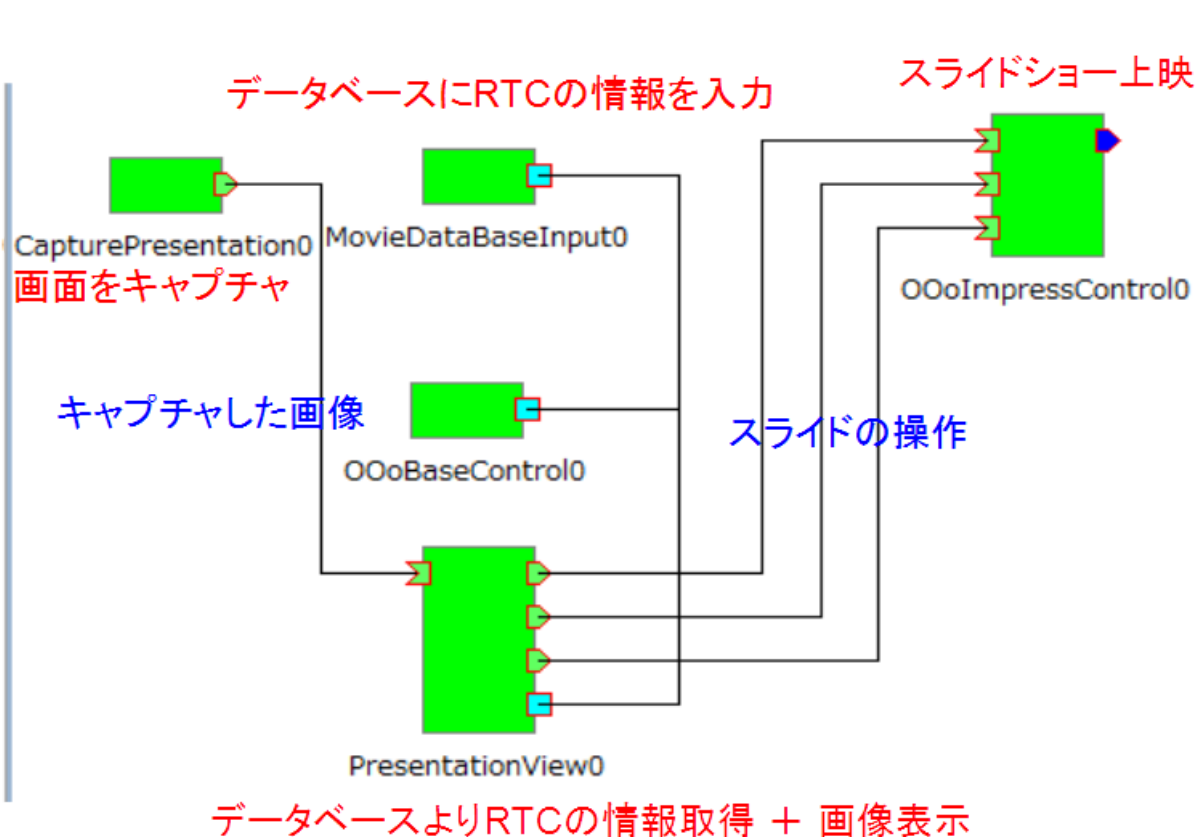
2: 表示したい画像データを送信しているRTCを選択

1: 更新ボタンを押すとデータベースに入力された画面キャプチャコンポーネントの情報を表示

3: 選択したRTCのデータポートと接続

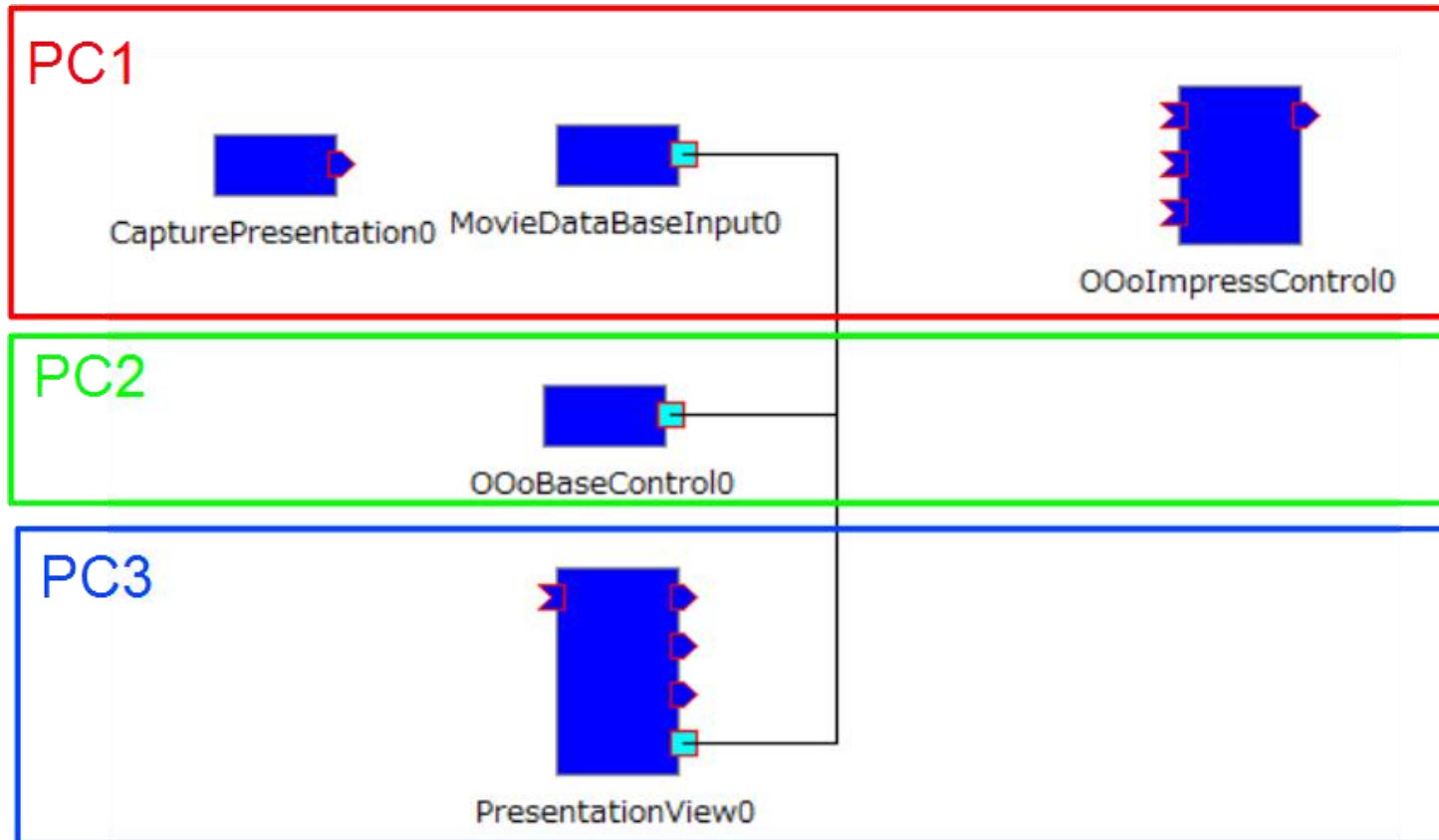
# デモ1

- データポートを自動的に接続する
  - プレゼンテーション表示コンポーネントのGUIにスライドショーの画像が表示



# デモ1

- 3台のPCでRTCを起動する



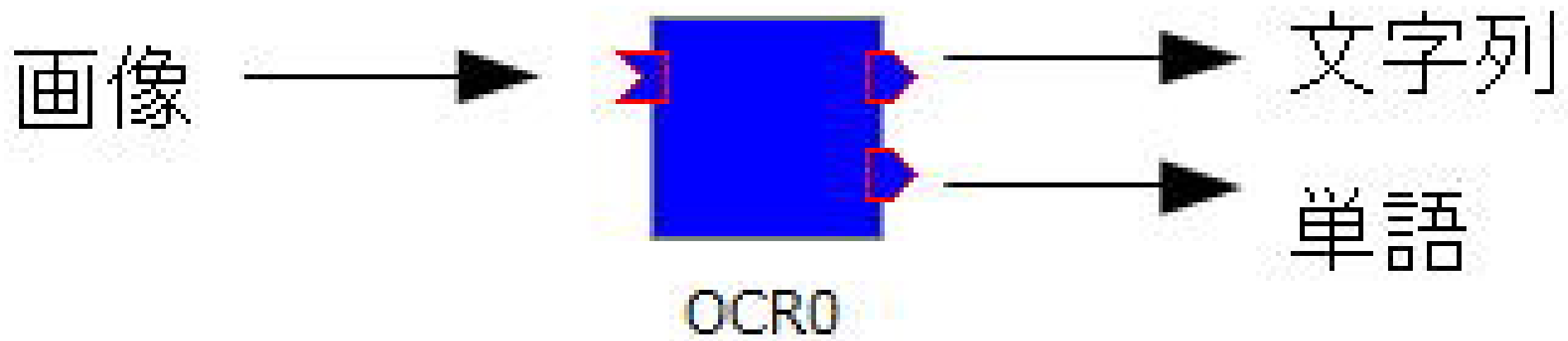
- Baseを操作するRTCのパスさえ知っていればデータベースに入力されているRTCの情報から自動的に接続できる

# 今後の課題

- デモシステムが不足
- Microsoft Officeを操作するRTC群をPythonで開発することで、OpenOfficeを操作するRTC群のコードを共有する
  - Excel、Word、PowerPointを操作するRTCのPython版を配布しているが、バグがある
- Microsoft Officeを操作するRTC群の動作速度が遅い
  - RTCを起動したプロセスからExcel等のプロセスへの通信に時間がかかるため

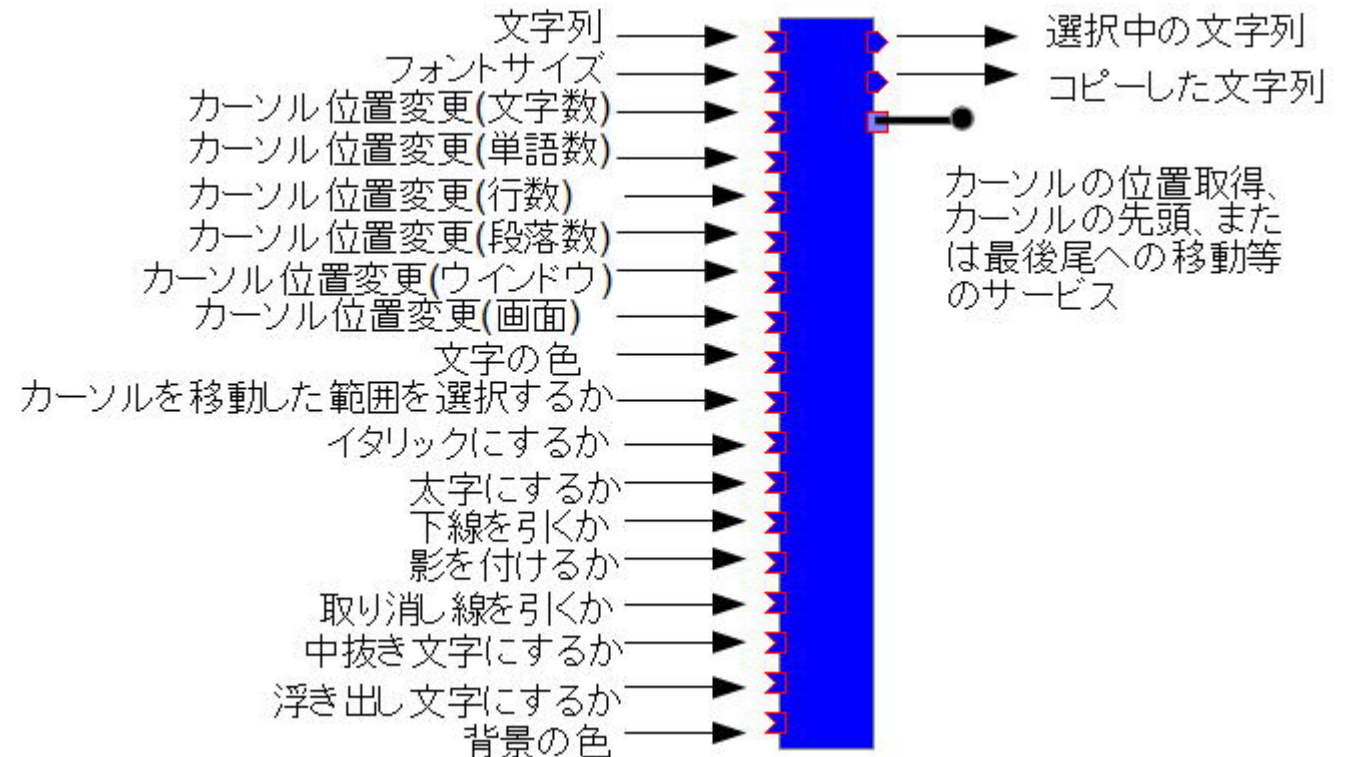
# MODI(文字認識)のRTC

- 入力された画像から認識した文字列を出力



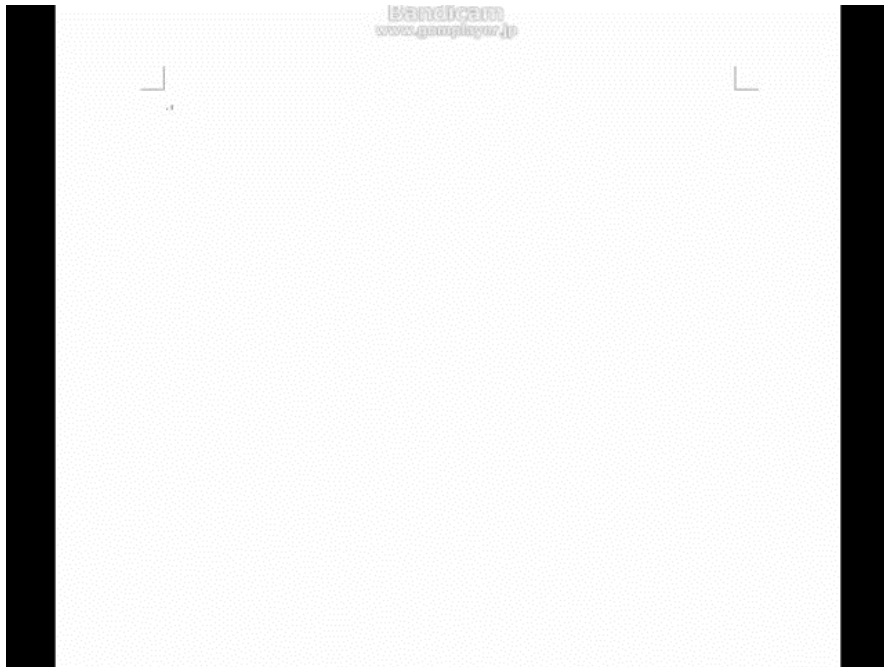
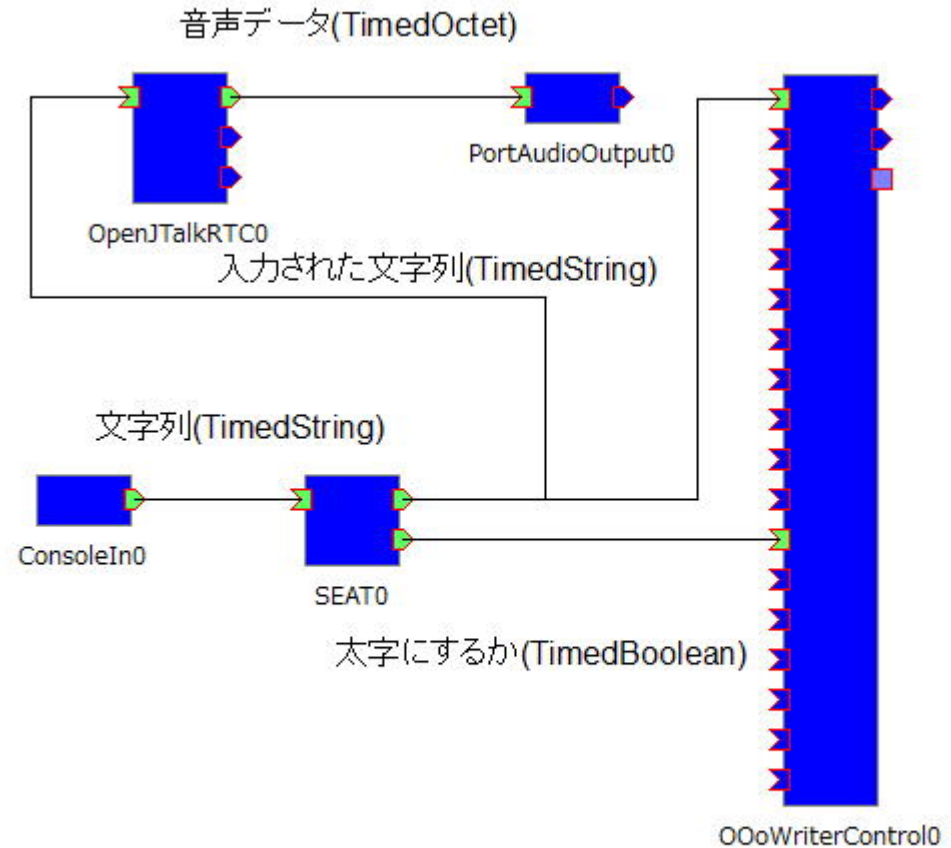
# Word、Writer(文字表示)を操作するための RTC

- InPortから入力された文字列の表示
  - フォントの変更
  - カーソル位置の変更



# デモ2

- OpenHRIのRTCを利用
- 入力された文字列を発音すると同時にWord、Writerの文書に表示
  - 指定した文字列は太字で表示



# 画像変換コンポーネント

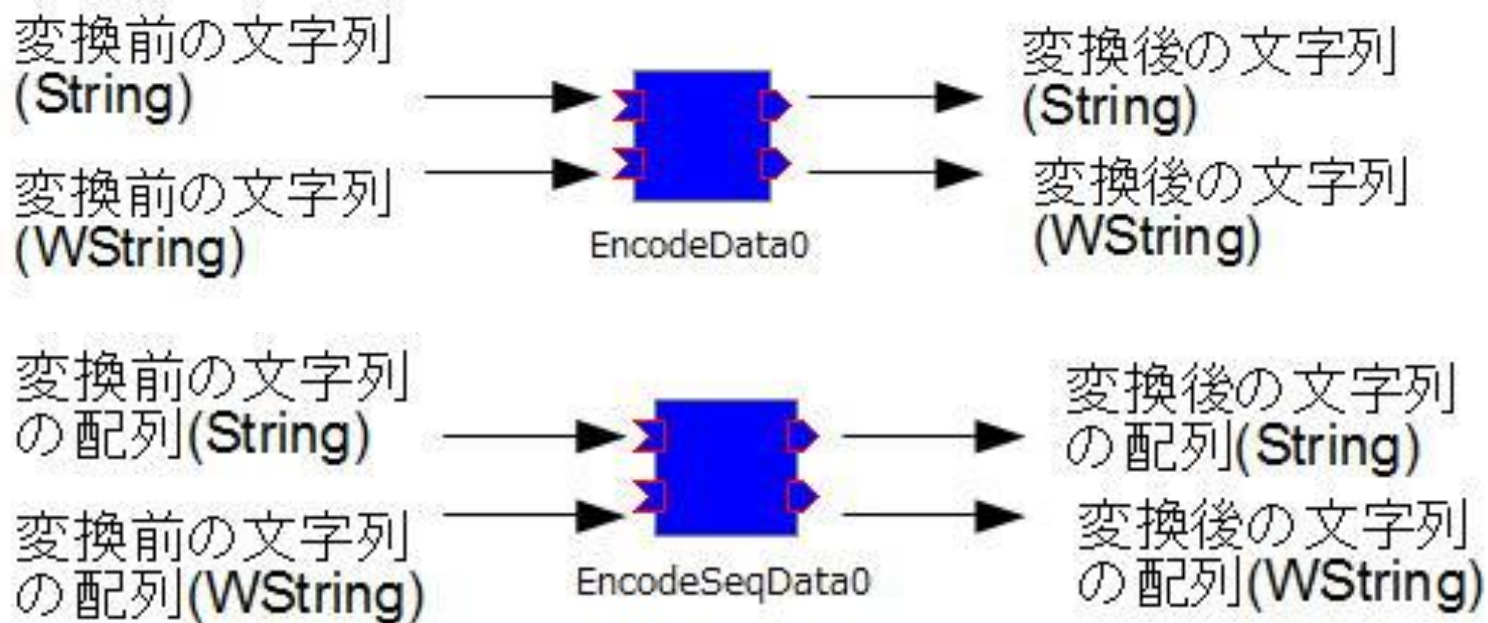
- 画像の圧縮、復元を行う





# 文字コード変換コンポーネント

- InPortから入力された文字列の文字コードを変換してOutPortから出力



# OpenOfficeを操作するためのRTC群の実装方法

- UNO(Universal Network Object)にPythonからOpenOfficeを操作する
  - C++、Java、Python等様々なプログラミング言語からOpenOfficeを動作可能
  - Scripting Frameworkで動くモードとプロセス間通信を行うモード
    - プロセス間通信を行うモードは動作が遅いのでScripting Frameworkで動くモードを選択
      - soffice.binでPythonファイルが実行される

Writerに“Hello World”と20回書き込むだけの簡単なプログラムで文字の書き込みにかかった時間の平均

Scripting Frameworkで動くモード	0.001189[s]
プロセス間通信を行うモード	0.006236[s]

# Microsoft Officeを操作するためのRTC群の実装方法

- COMによりOfficeを操作
  - Python版はWin32Comライブラリを使用

Writerに“Hello World”と20回書き込むだけの簡単なプログラムで文字の書き込みにかかった時間の平均

Wordを操作するRTC	0.003498[s]
Writerを操作するRTC	0.001189[s]