

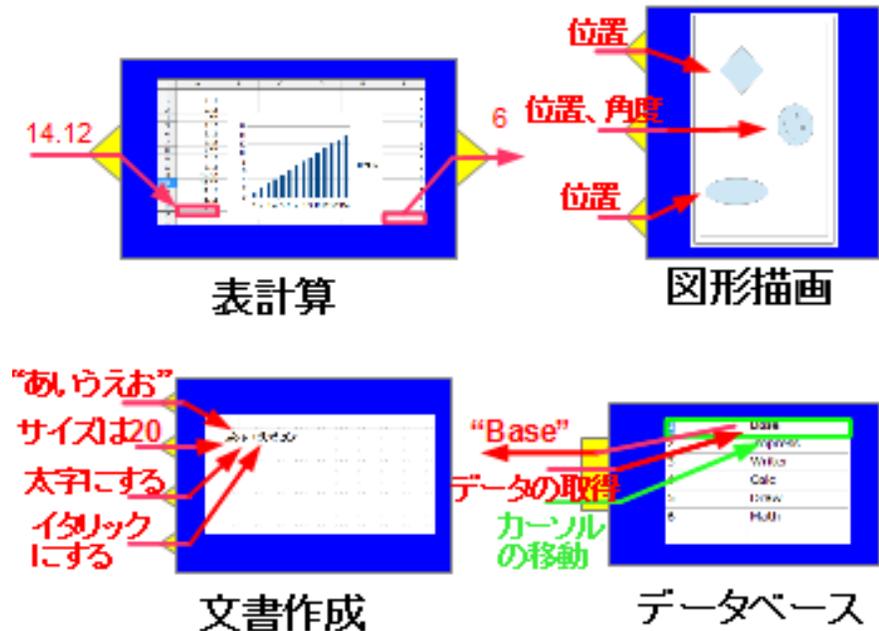
オフィスソフトを操作するための RTC群

宮本 信彦



開発概要

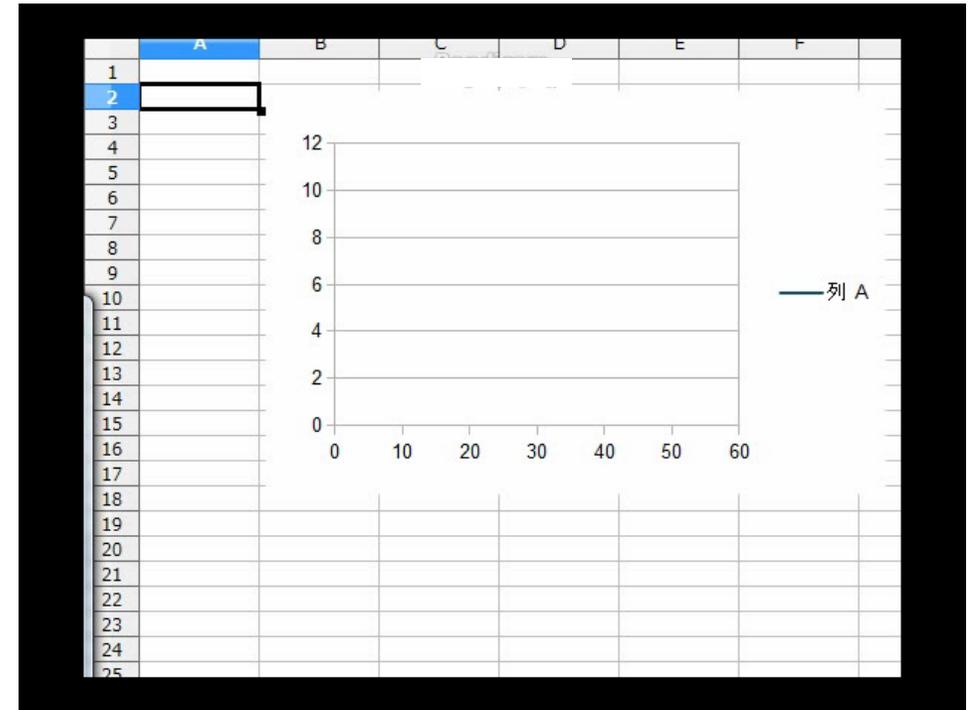
- Microsoft Office、OpenOfficeの表計算機能、文書作成機能等をRTミドルウェアを用いたシステムで利用するためのRTC群
 - Excel、Calc(表計算)を操作するRTC
 - セルのデータをデータポートから入出力
 - Draw(図形描画)を操作するRTC
 - 図形の位置、姿勢をデータポートから入出力
 - Base(データベース)を操作するRTC
 - サービスポートを利用することでデータベースを操作
 - PowerPoint、Impress(プレゼンテーション)を操作するRTC
 - 上映中のスライドの番号の変更、線の描画
 - Word、Writer(文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



- UbuntuにはLibreOfficeが標準でインストールされている
- 多くのWindowsを搭載しているPCでMicrosoft Officeがインストールされている
 - → 手軽に使えるRTCになるのではないか？

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で操作できれば面白いかもしれない
 - 用途は各RTCによって様々
 - 中には使いどころのわからないRTCもある

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)

目次

- Excel、Calc(表計算機能)を操作するRTC
- Draw(図形描画機能)を操作するRTC
- Base(データベース機能)を操作するRTC
- PowerPoint、Impree(プレゼンテーション機能)を操作するRTC
- システムの例
 - その他のRTCについては以下のサイトで説明してあります
 - http://www.openrtm.org/openrtm/ja/project/contest2014_6
 - 一応資料は作っているのでこの場で知りたい方は質問してください

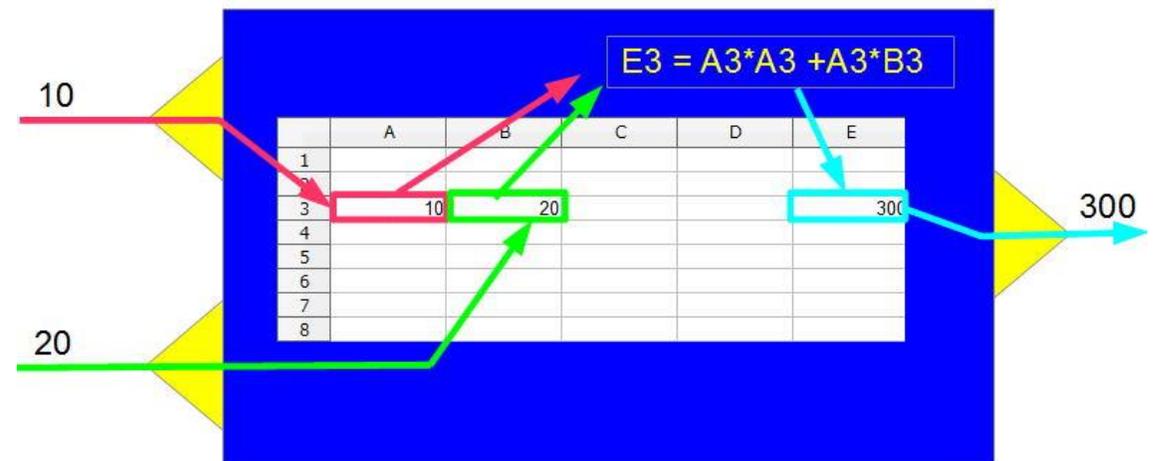
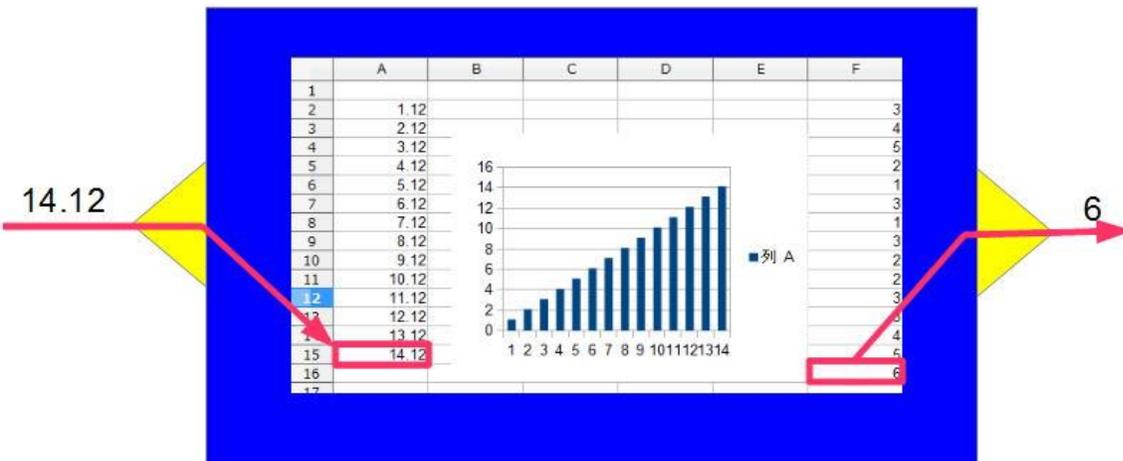
目次

- Excel、Calc(表計算機能)を操作するRTC
- Draw(図形描画機能)を操作するRTC
- Base(データベース機能)を操作するRTC
- PowerPoint、Impree(プレゼンテーション機能)を操作するRTC
- システムの例

Excel、Calc(表計算)を操作するためのRTC

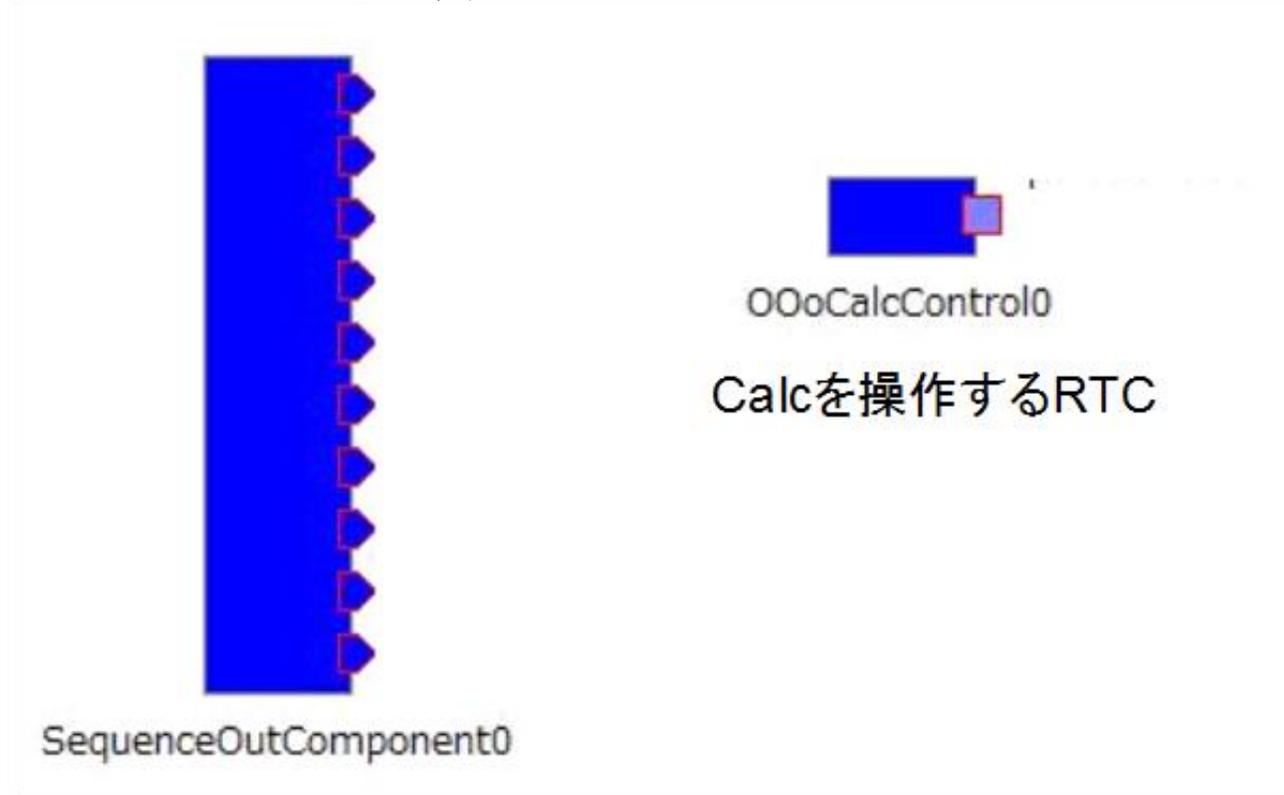
- InPortで受信したデータをセルに入力
 - データの保存、グラフで表示したい場合に使用
- セルのデータをOutPortから出力
- InPortで受信したデータを数式で計算後OutPortから出力
 - シミュレーション等で使用
- データポートは動的に追加可能
 - BasicDataType.idl、ExtendedDataTypes.idlで定義されたデータ型に対応
 - GUIによる操作、もしくはコンフィギュレーションパラメータで追加するデータポートの設定可

 指定位置のセルを操作するサービス



Excel、Calc(表計算)を操作するためのRTC

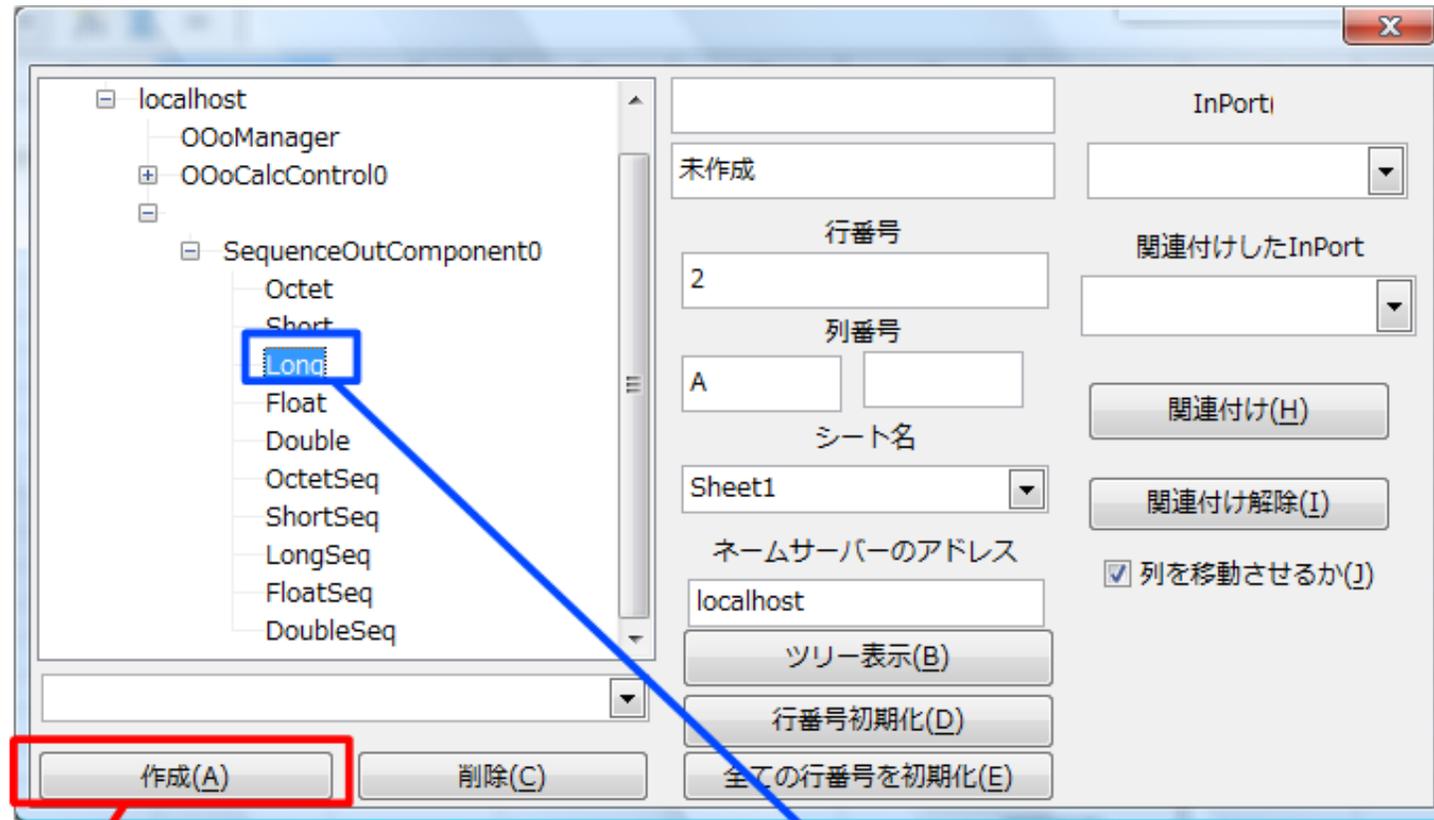
- データポートの追加の手順



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

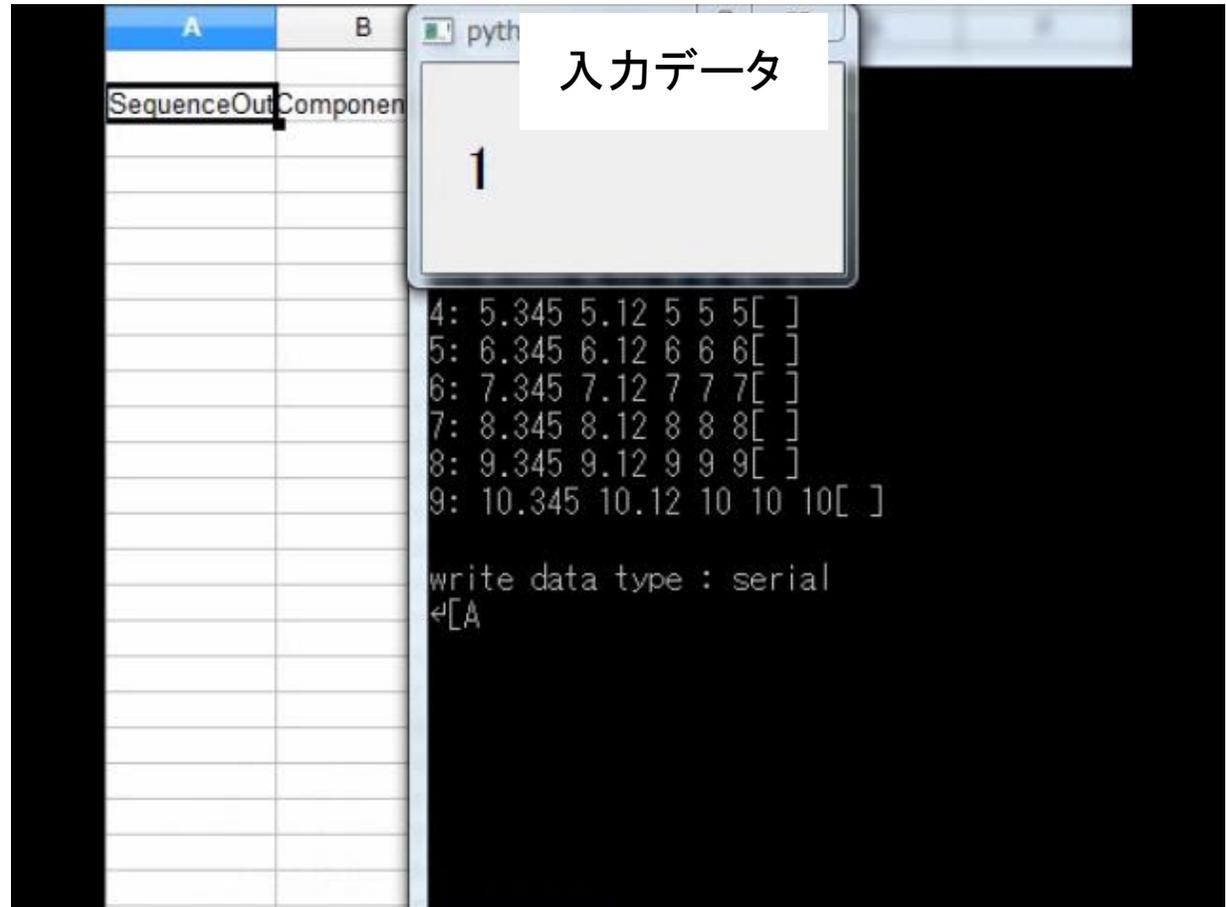
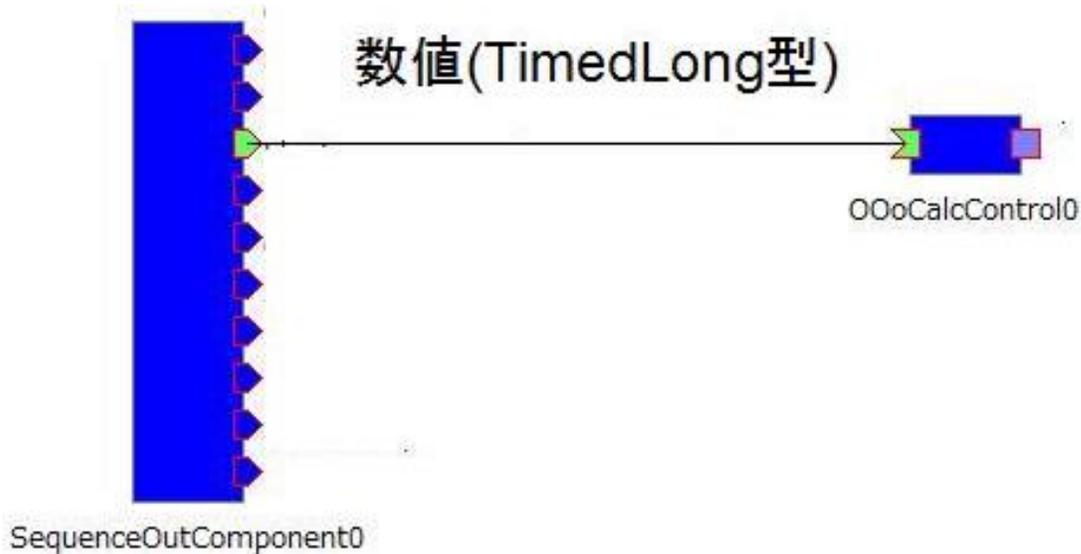
Excel、Calc(表計算)を操作するためのRTC

- 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



Excel、Calc(表計算)を操作するためのRTC

- セルの値をOutPortから出力



出力データ

0

Basic type

0

Sequence type

0: - -

Double Float Lo

Basic type

0 0

Sequence type

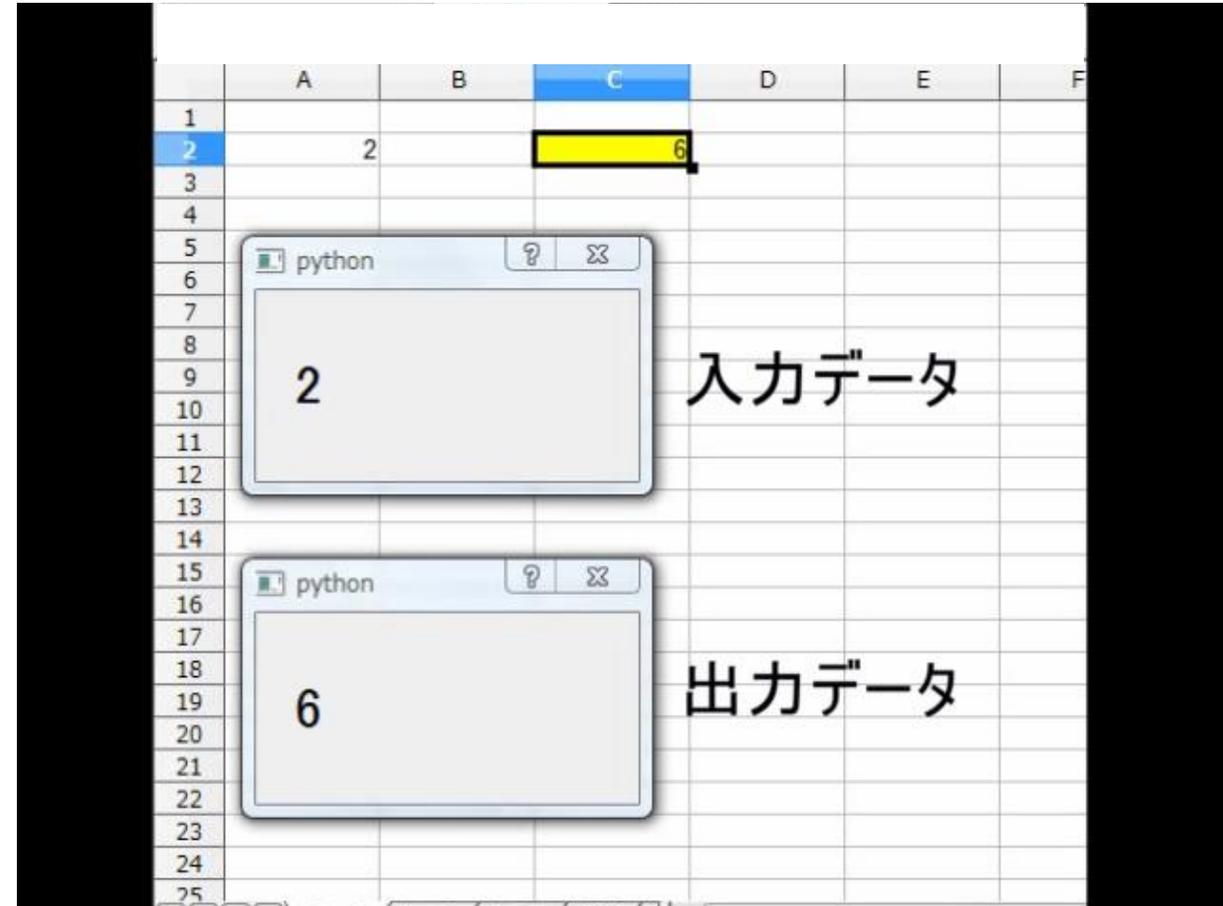
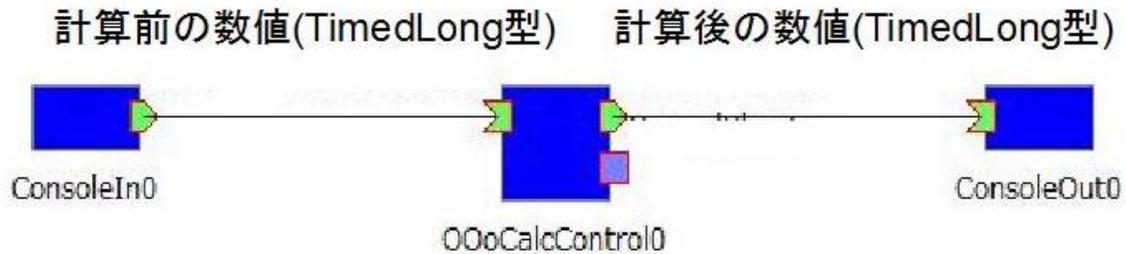
0: - -

Double Float Lo

The screenshot shows an Excel spreadsheet with columns A through F and rows 1 through 23. A window titled '出力データ' (Output Data) is overlaid on the spreadsheet, displaying the value '0'. The window has a close button in the top right corner. Below the window, there is a list of data types and their corresponding values: 'Basic type' with value '0', 'Sequence type' with value '0:', and 'Double Float Lo' with values '0' and '0'. The list is repeated twice.

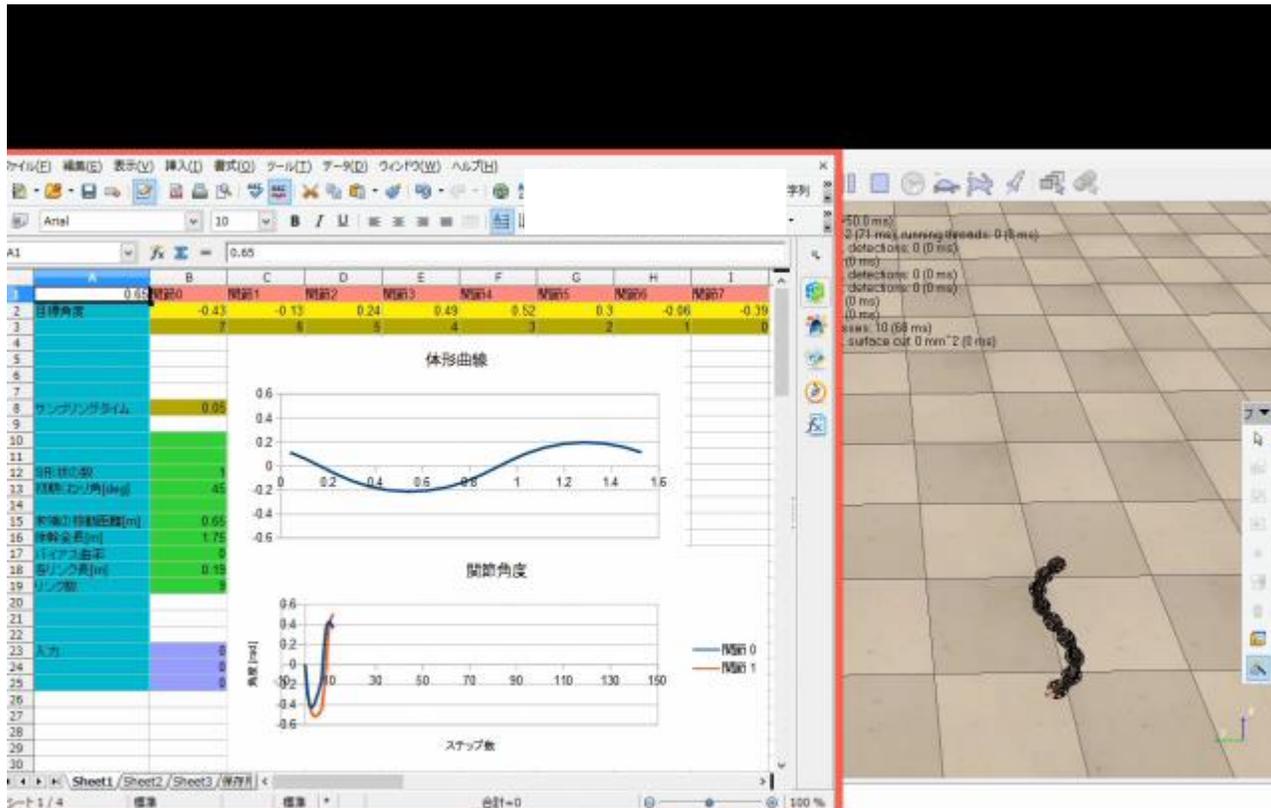
Excel、Calc(表計算)を操作するためのRTC

- InPortで受信したデータを数式で計算後OutPortから出力
 - OutPortと関連付けしたセルに数式を入力
 - 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

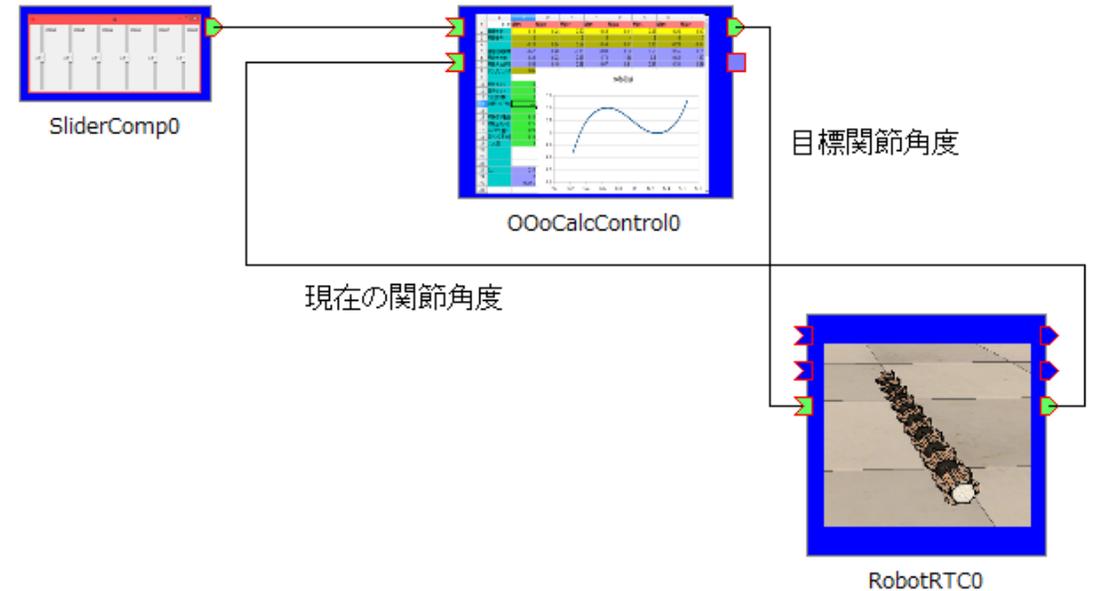


Excel、Calc(表計算)を操作するためのRTC

- シミュレーションへの応用



スライダーで一部のパラメータを変更



- 各パラメータを容易に変更可能

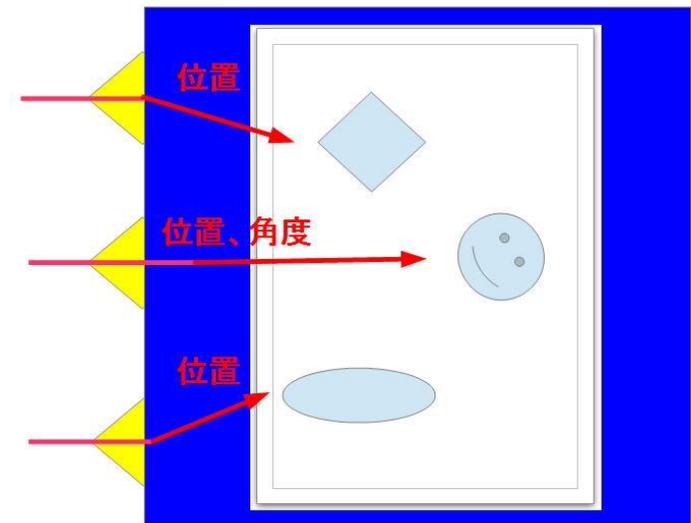
- V-REPとV-REPのOpenRTM-aist対応プラグインを使用

目次

- Excel、Calc(表計算機能)を操作するRTC
- Draw(図形描画機能)を操作するRTC
- Base(データベース機能)を操作するRTC
- PowerPoint、Impree(プレゼンテーション機能)を操作するRTC
- システムの例

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
 - ロボットの位置を表示したい場合などに使用
 - 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

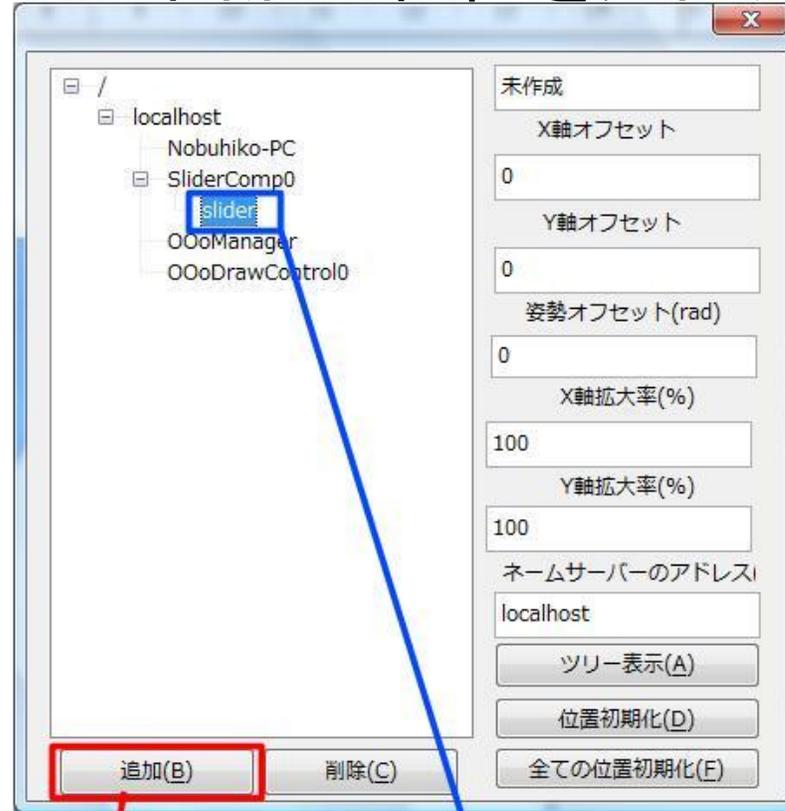


OOoDrawControl0

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

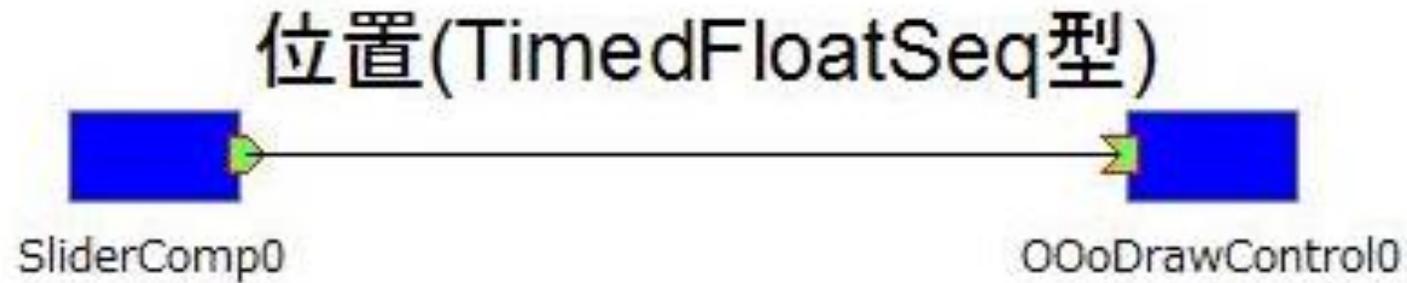


1: ツリーからデータポートを選択

2: データポート追加ボタンを押す

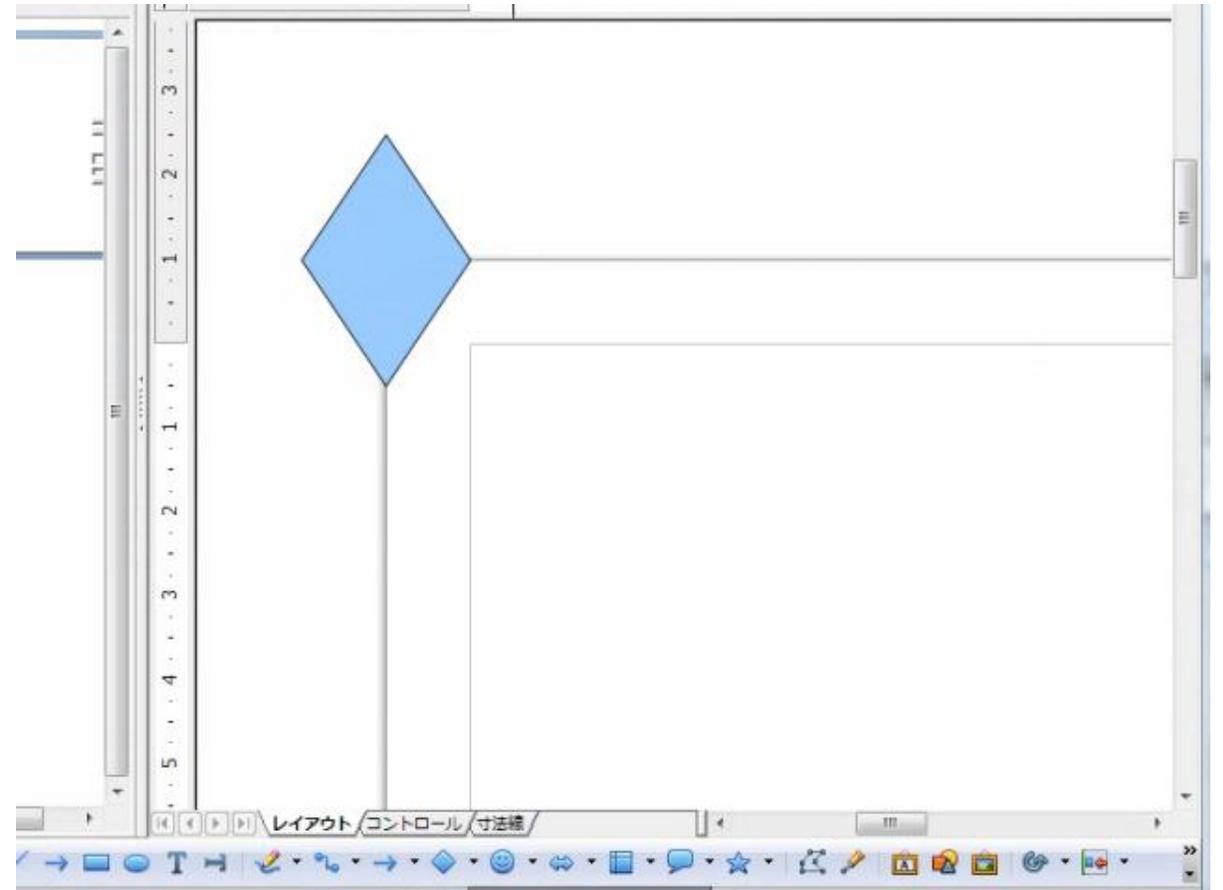
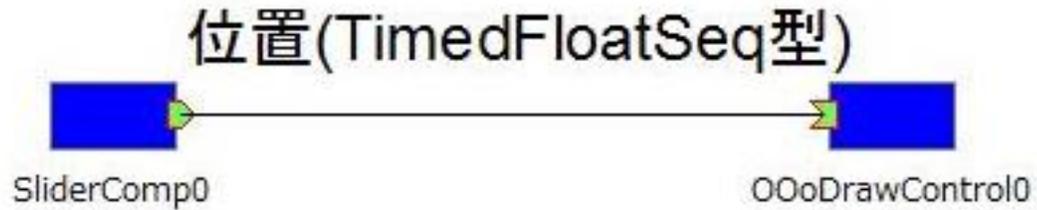
Draw(図形描画)を操作するためのRTC

- 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

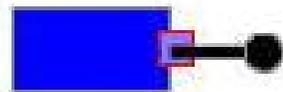


目次

- Excel、Calc(表計算機能)を操作するRTC
- Draw(図形描画機能)を操作するRTC
- Base(データベース機能)を操作するRTC
- PowerPoint、Impree(プレゼンテーション機能)を操作するRTC
- システムの例

Base(データベース)を操作するためのRTC

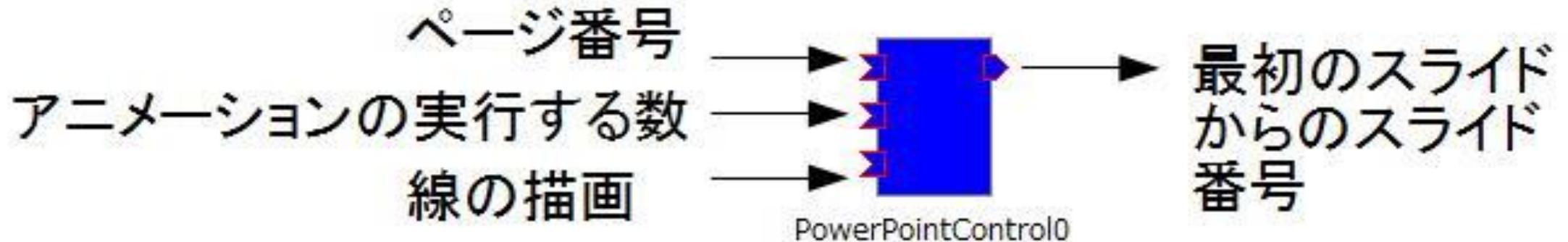
- 他のRTCはサービスポートを利用することでデータベースを操作できる
 - データベースへの接続
 - データベースへの問い合わせ(SQL文を入力)
 - データベースの追加、削除



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



システムの例

- Base、Impress(PowerPoint)のRTCを使用したシステム
- スライドショーをGUIにより操作する
- 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

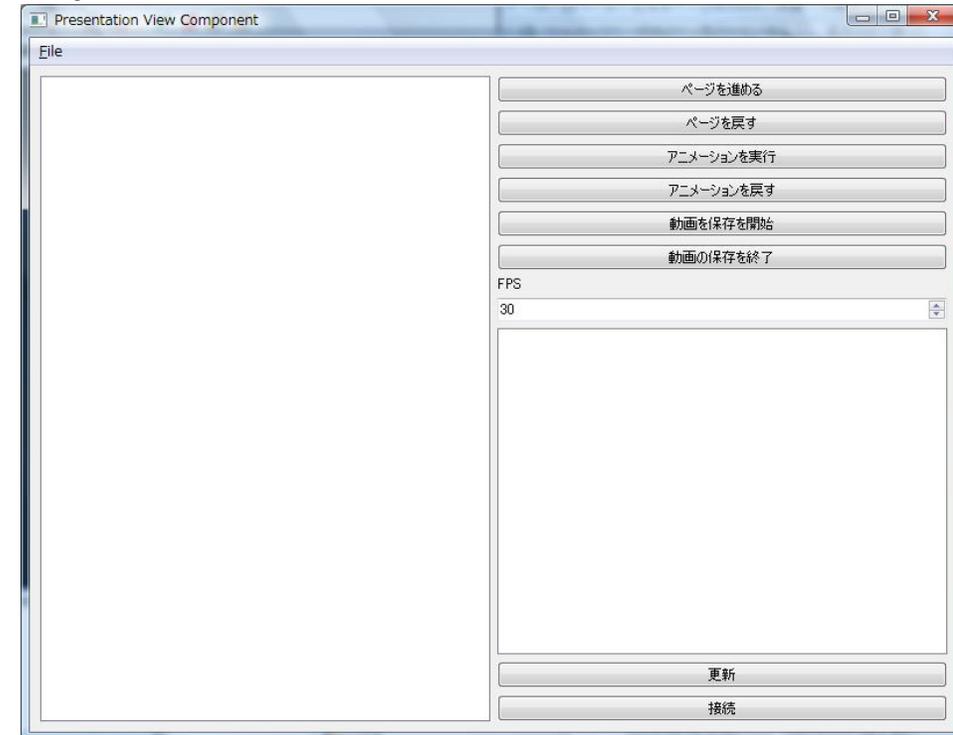
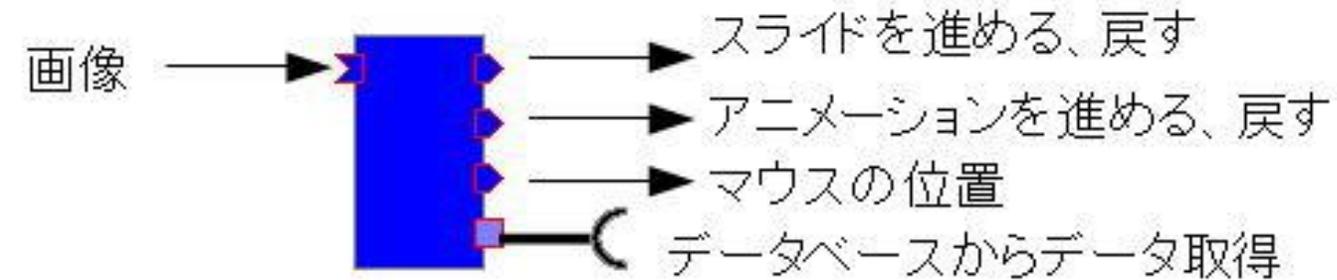
- 画面をキャプチャして画像データをOutPortから出力



- 上映中のスライドをキャプチャする

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、データ入力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Impress(PowerPoint)を操作



データ入力コンポーネント

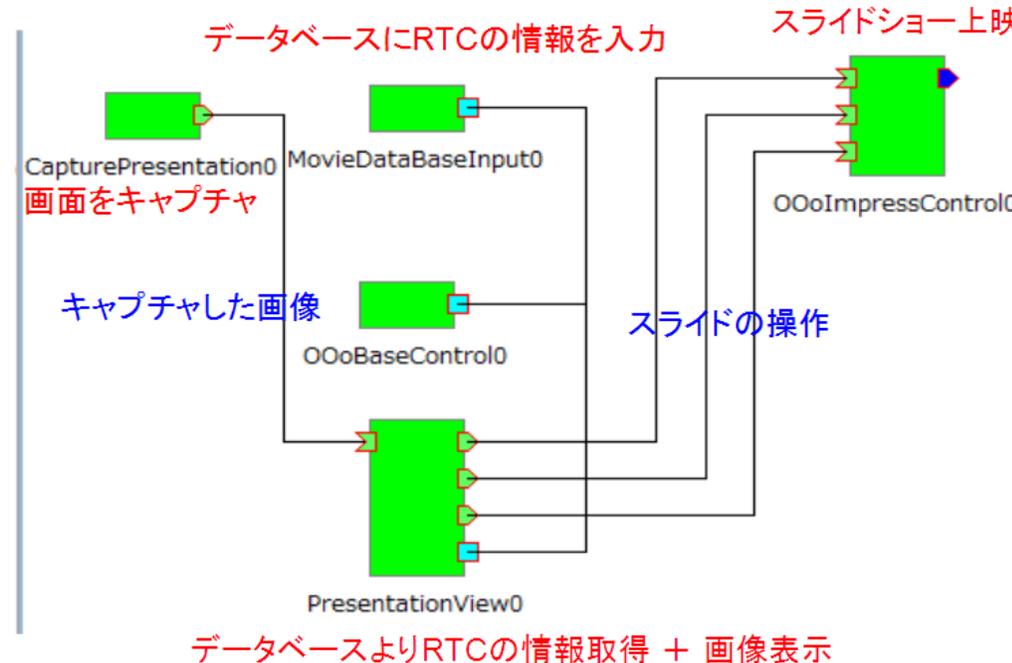
- コンフィギュレーションパラメータで設定した情報をデータベースに挿入したい時に利用



- 上映中のスライドショーの名前、画面キャプチャコンポーネント、Impress(PowerPoint)を操作するRTCのパス等をデータベースに入力

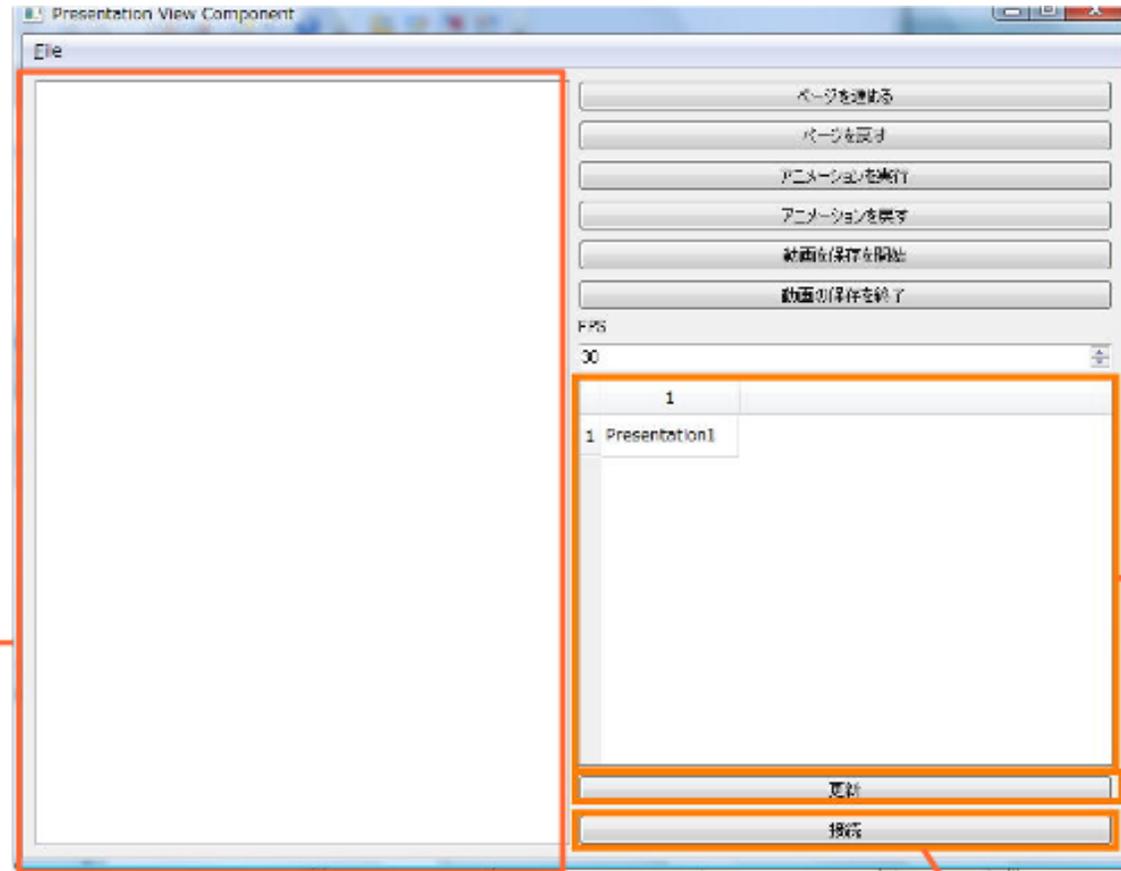
システムの例

- 全体的な流れ
 - データ入力コンポーネントがBaseのRTCのサービスポートを利用してデータベースにデータを挿入
 - Impress(PowerPoint)のRTCをアクティブにすることでスライドショーを開始
 - 画面キャプチャコンポーネントでスライドショーをキャプチャ
 - プレゼンテーション表示コンポーネントのGUIでの操作によりデータベースからデータを取得、データポートを自動接続してスライドショーの操作ができるようになる



システムの例

- プレゼンテーション表示コンポーネントのGUIを操作

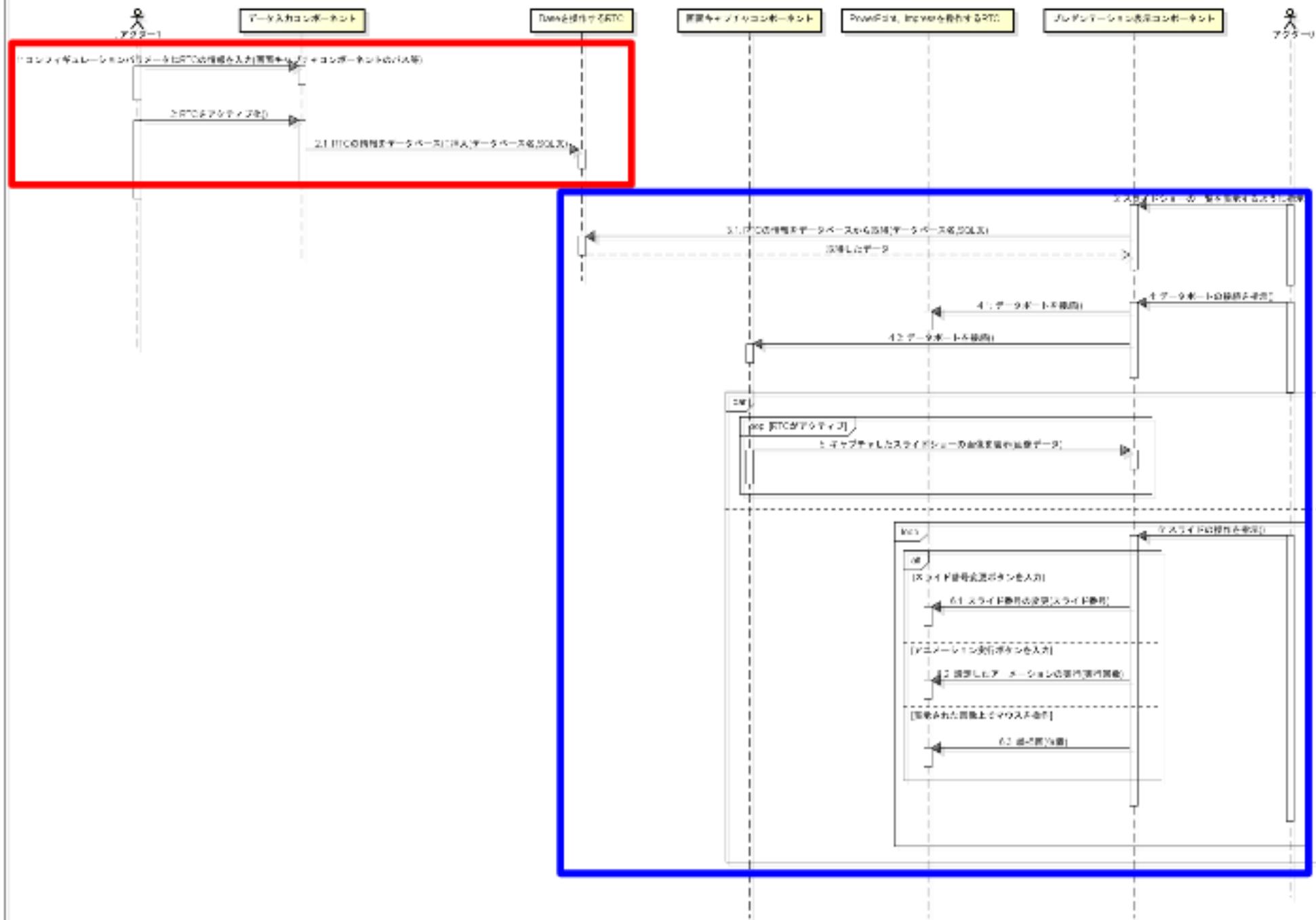


4: この部分に受信した画像が表示される

2: 表示したいスライドショーの名前を選択

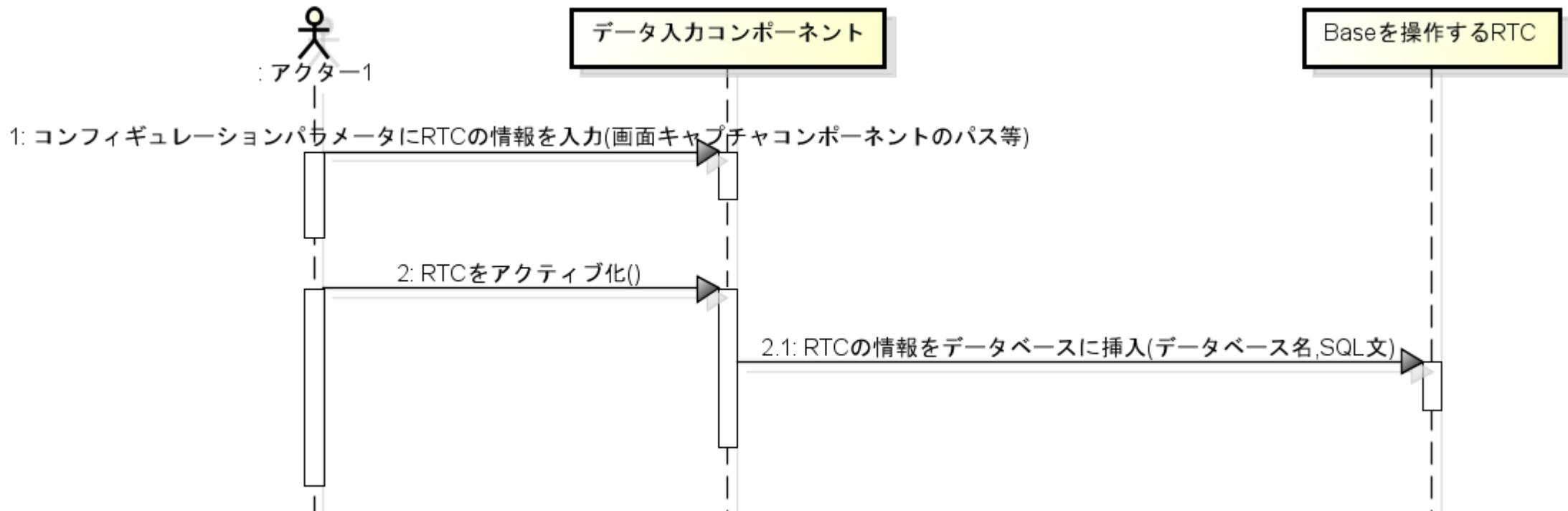
1: 更新ボタンを押すとデータベースに入力されたスライドショーの一覧を表示

3: 選択したRTCのデータポートと接続



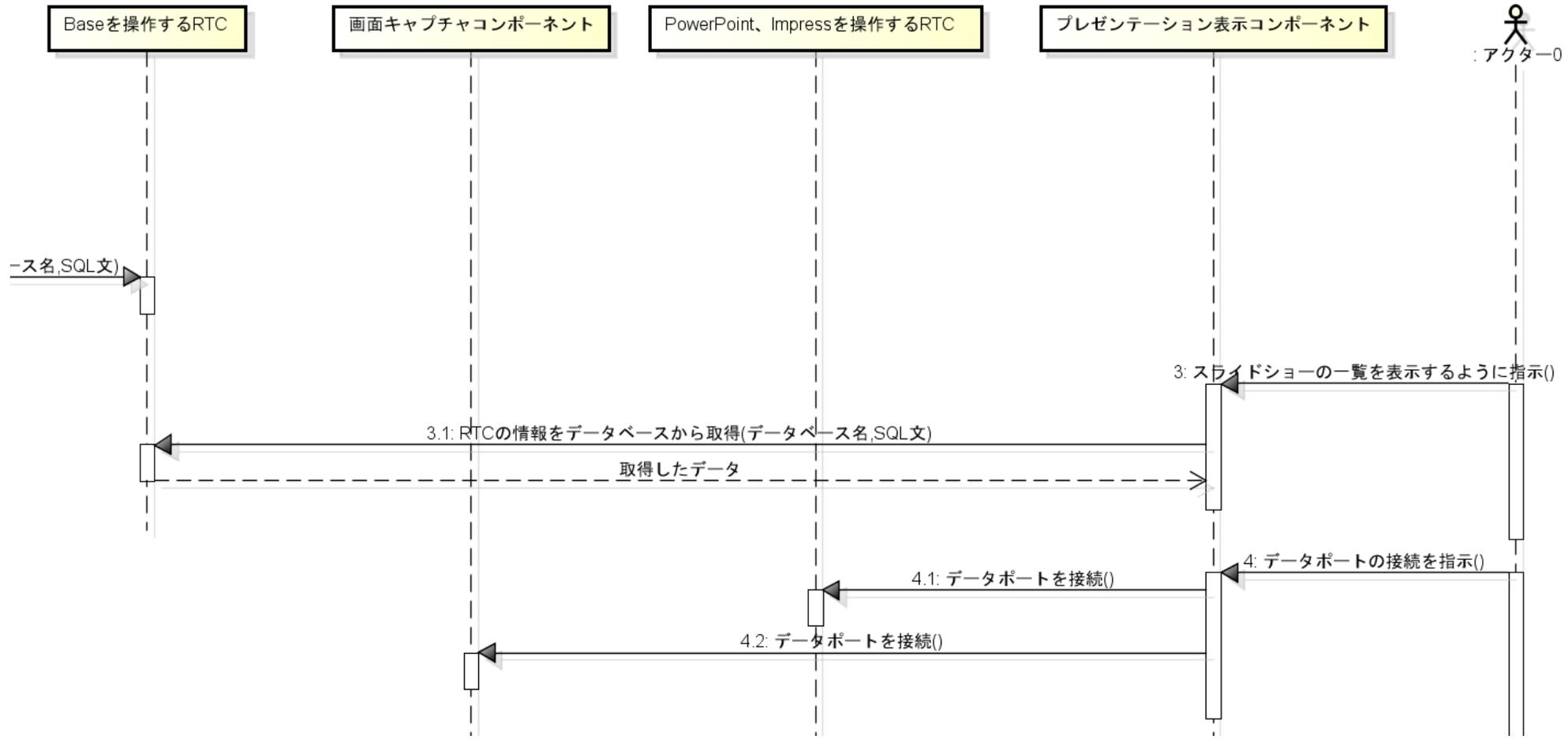
システムの例

- コンフィギュレーションパラメータの設定、RTCのアクティブ化
 - データベースに上映するスライドショーの名前、画面キャプチャコンポーネント、Impress(PowerPoint)を操作するRTCの情報を入力



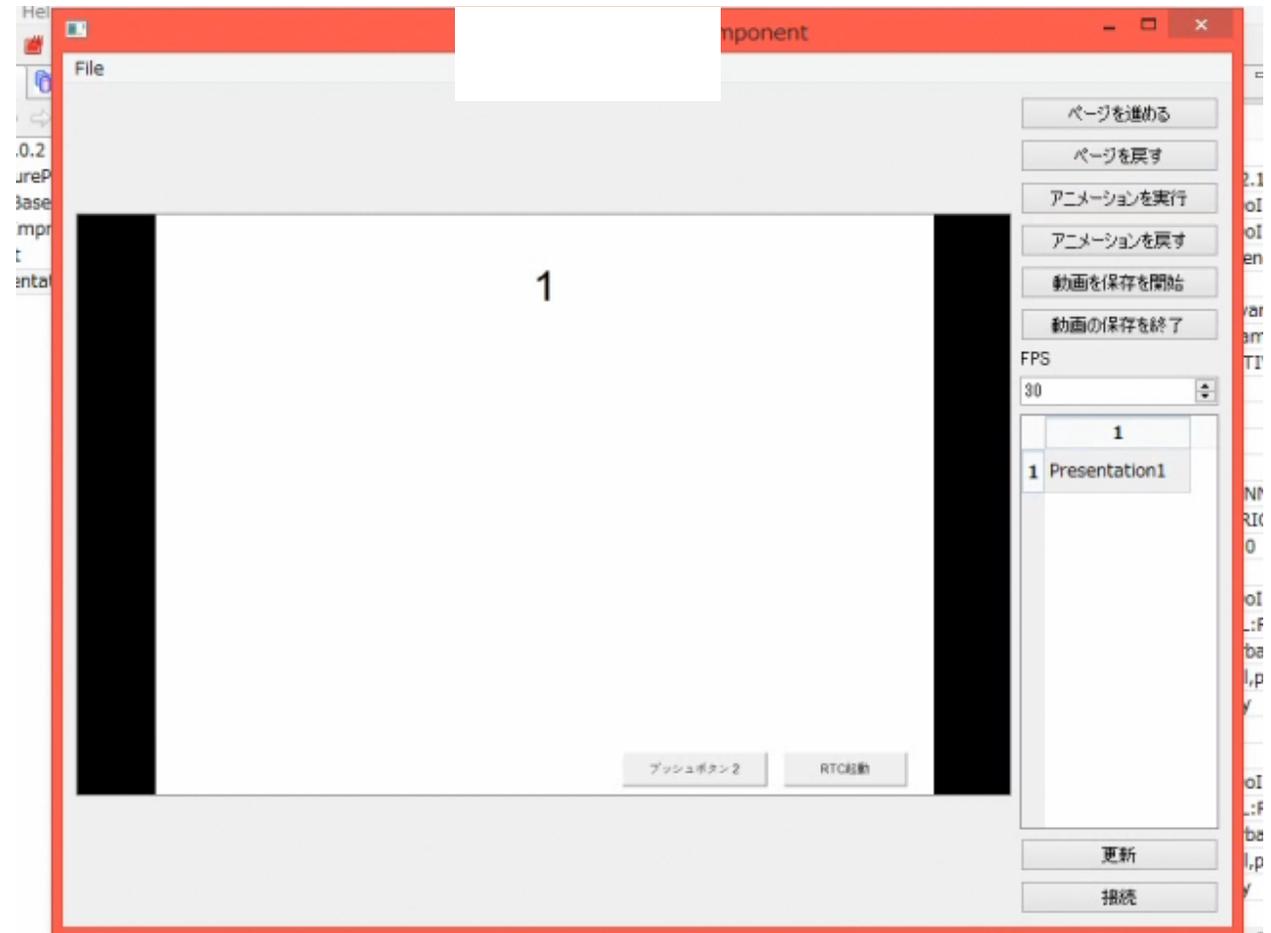
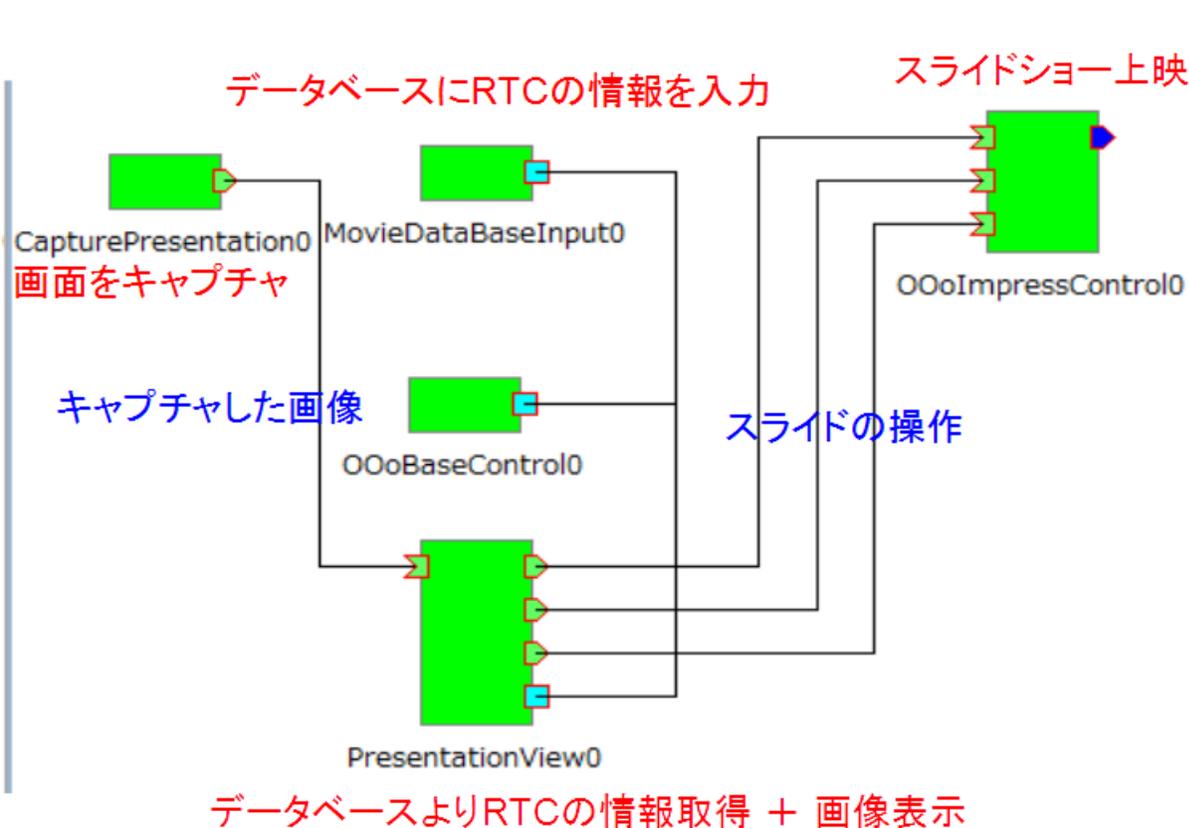
システムの例

- プレゼンテーション表示コンポーネントのGUIを操作
 - データポートの接続



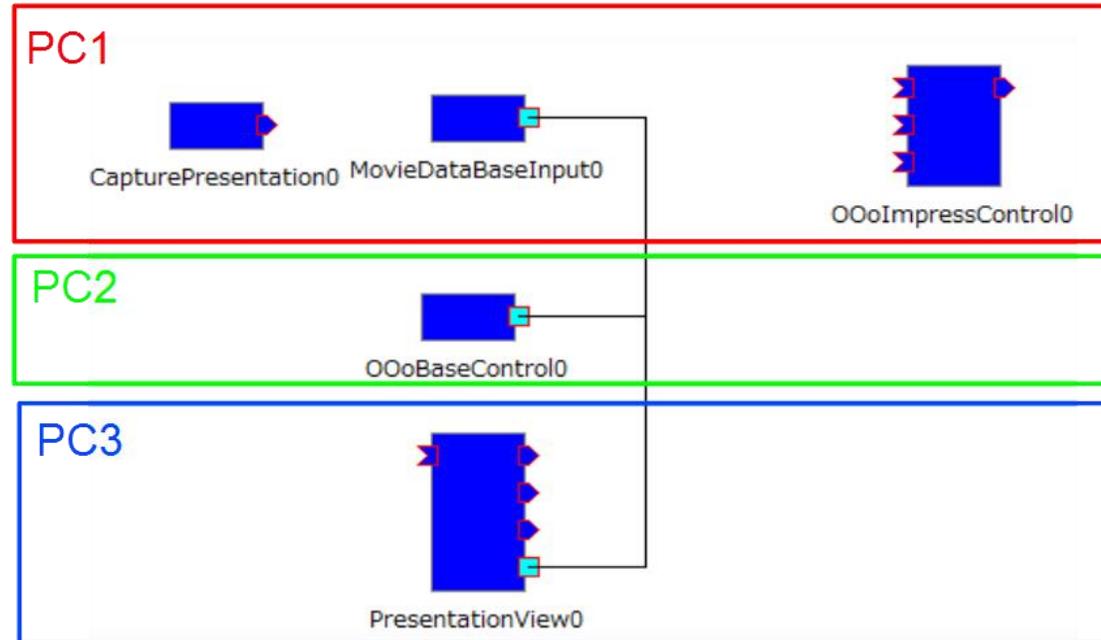
システムの例

- 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
 - ボタン押下によるスライド番号の変更、マウス操作による線描画



システムの例

- 3台のマシんでRTCを起動する
 - PC1はスライドショーを上映しているPC
 - PC3はプレゼンテーションを聞いている人のPC



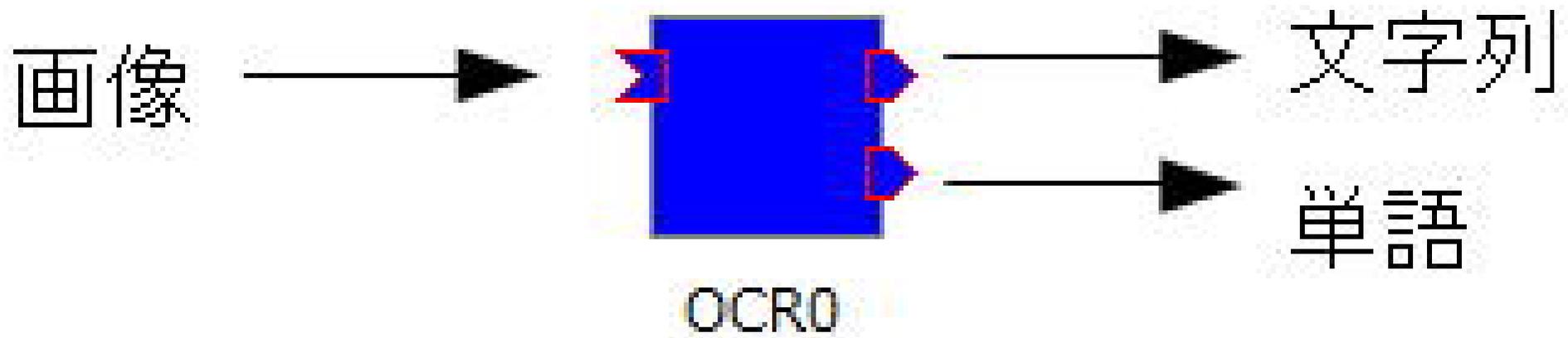
- PC3からはBaseを操作するRTCのパスさえ知っていればデータベースに入力されているRTCの情報からGUIでの操作により簡単に接続できる
 - 質疑応答などで利用

今後の課題

- 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

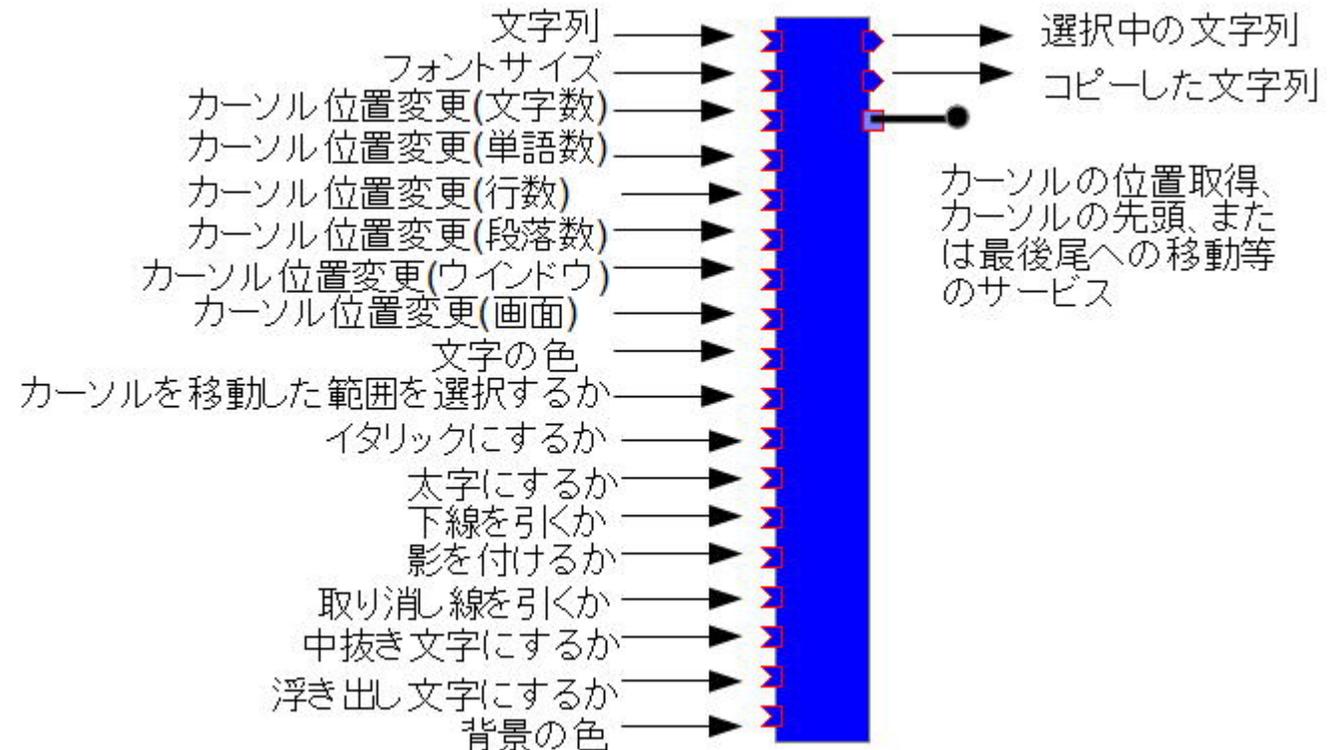
MODI(文字認識)のRTC

- 入力された画像から認識した文字列を出力



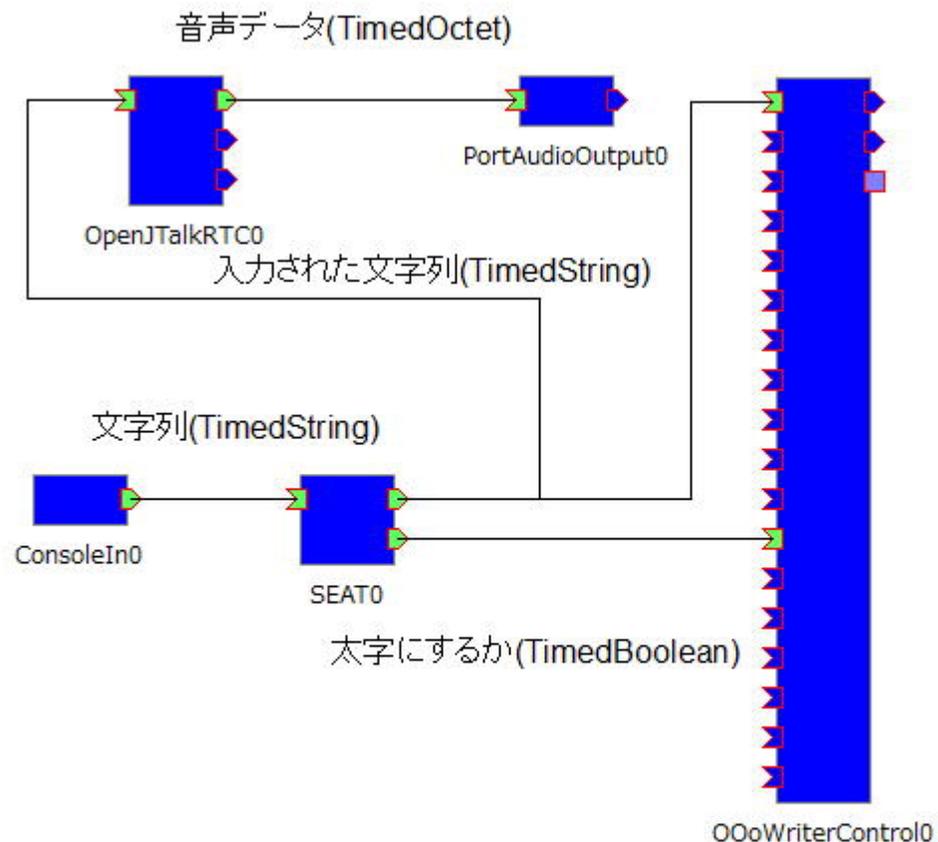
Word、Writer(文字表示)を操作するための RTC

- 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



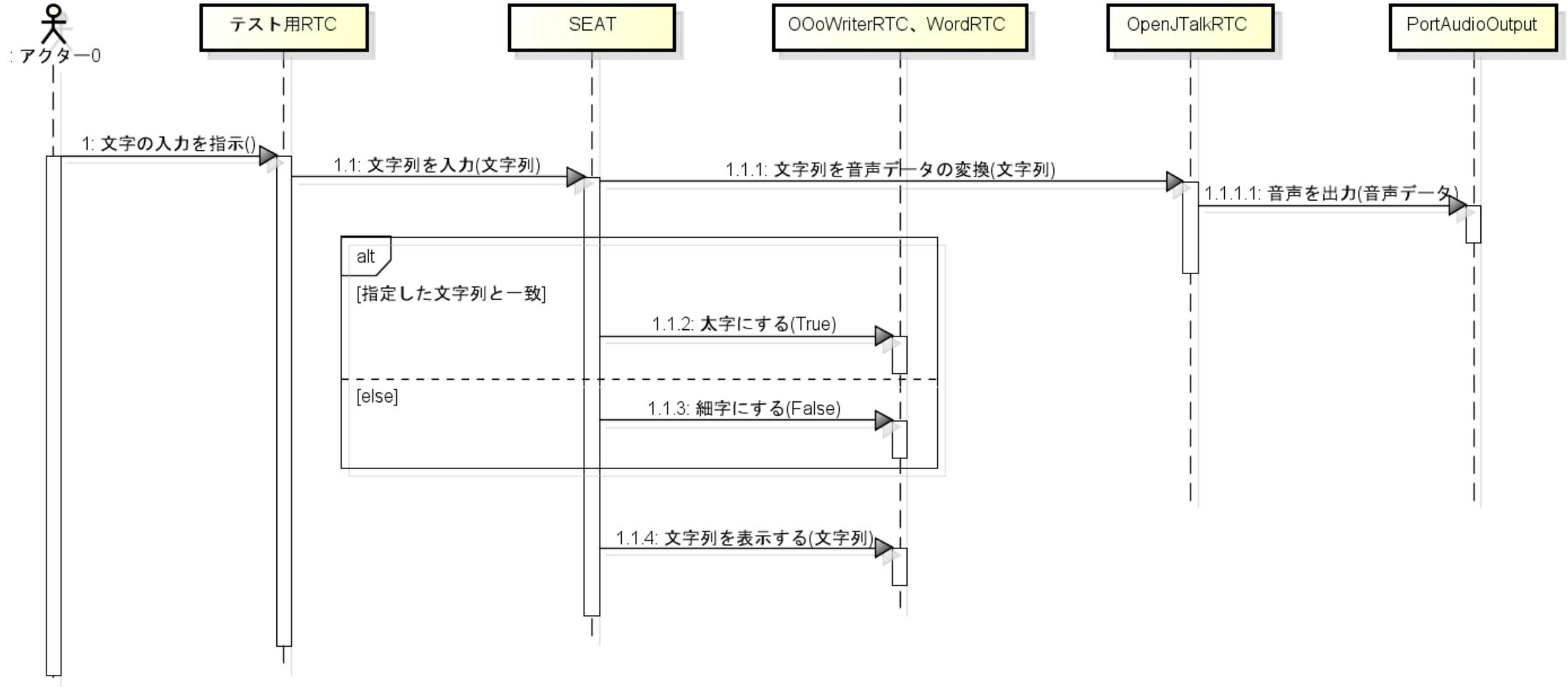
デモ2

- 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



デモ2

sd シーケンス図0



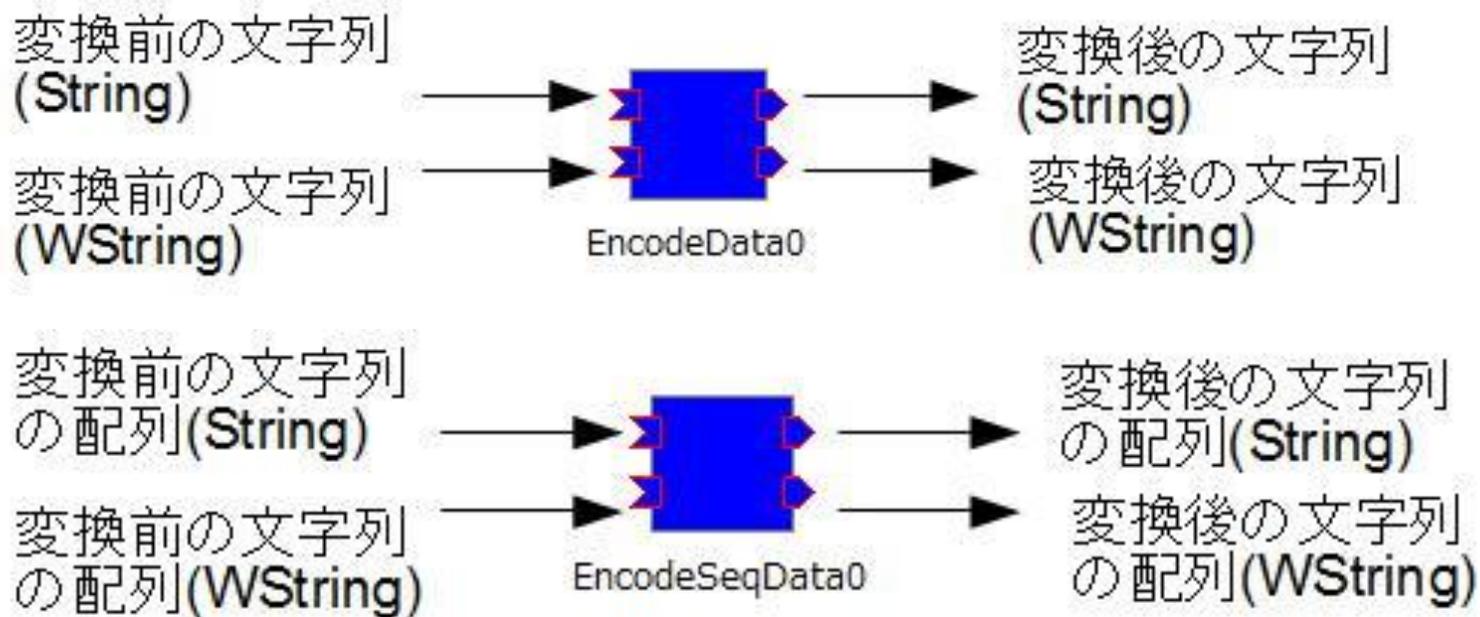
画像変換コンポーネント

- 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(Component Object Model)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]

サーペノイド曲線

- 体軸座標 s に沿って曲率が正弦波状に変化する曲線

$$\kappa(s_p) = -\frac{2K_n\pi\alpha_0}{L} \sin\left(\frac{2K_n\pi}{L}s_p\right) + K_l$$

- 曲線をリンク長で離散化した時の各関節角度

$$\theta_i(s) = -2\alpha_0 \sin\left(\frac{K_n\pi}{n}\right) \sin\left(\frac{2K_n\pi}{L}s + \frac{2K_n\pi}{n}i\right) + K_l l$$

$\theta_i(s)$: 関節角度

K_n : S-形状の数

α_0 : 初期くねり角

s_p : 体形曲線の曲線長

s : 体形曲線に沿う蛇型ロボット末端の移動距離

L : 体幹全長

K_l : バイアス曲率

l : 各リンク長

i : 関節番号