

RT コンポーネント操作マニュアル

**RT によるプレゼンテーション支援  
コンポーネント群**

2022 年 9 月 6 日版

芝浦工業大学

デザイン工学部デザイン工学科

佐々木 毅

## 更新履歴

- |                   |   |
|-------------------|---|
| 2012 年 11 月 19 日版 | 第 1 版。  |
| 2013 年 1 月 30 日版  | コメント表示のコマンドによるテキスト色、サイズ指定機能の追加に伴う該当箇所の加筆・修正。<br>1.1 節の参考文献にページ数を追加。 |
| 2022 年 9 月 6 日版   | 5 章の問合せ先の変更。  |

# 目次

<b>1</b>	<b>本コンポーネントの概要</b>	<b>1</b>
1.1	開発の背景	1
1.2	プレゼンテーションコンポーネントの機能と用途	1
1.3	開発環境	1
1.4	本稿で利用しているコンポーネント群	2
<b>2</b>	<b>コンポーネントの説明</b>	<b>2</b>
2.1	基本コンポーネント群	2
2.1.1	プレゼンテーションコンポーネント (CVPresentation)	3
2.1.2	キーボードコンポーネント (PresentationKeyDecoder)	5
2.1.3	マウスコンポーネント (PresentationMouseDecoder)	6
2.1.4	コメント入力コンポーネント (ConsoleInString)	6
2.2	機能拡張の紹介で用いるコンポーネント群	7
2.2.1	タイマーコンポーネント (PresentationCommentTimer)	7
2.2.2	レーザポインタインタフェースコンポーネント (PresentationLPInterface)	8
<b>3</b>	<b>CVPresentation の使用方法</b>	<b>10</b>
3.1	RTSystemEditor によるシステム構築の手順	10
3.2	CVPresentation の使用手順	10
3.2.1	スライド画像の準備	10
3.2.2	Configuration の設定	11
3.2.3	コンポーネントのアクティブ化	12
3.3	CVPresentation の機能	12
3.3.1	スライドのページ変更	12
3.3.2	スライド内への描画	13
3.3.3	コメント（テキスト）の表示	13
<b>4</b>	<b>CVPresentation の利用例</b>	<b>15</b>
4.1	基本コンポーネント群を用いた CVPresentation の機能の利用	15
4.1.1	キーボードの追加	15
4.1.2	マウスの追加	16
4.1.3	コメントの表示	17
4.2	様々なコンポーネントによる機能の拡張	17
4.2.1	プレゼンテーション時間の計測	17
4.2.2	レーザポインタを用いたプレゼンテーション	18
<b>5</b>	<b>お問い合わせ</b>	<b>21</b>

# 1 本コンポーネントの概要

## 1.1 開発の背景

自身のアイデアや自社製品の魅力、研究成果の意義などを伝える上でプレゼンテーションは欠かせない手段となっています。プレゼンテーションはスライドを用いて行うことが一般的となっていますが、スタイルが画一化されているがために、注目を集めるプレゼンテーションや印象に残るプレゼンテーションを行うことは容易ではありません。そこで、RT(Robot Technology)を利用することにより、より効果的・魅力的なプレゼンテーションを実現するためのプレゼンテーション支援コンポーネント群を開発することといたしました。

コンポーネントの設計指針等につきましては、

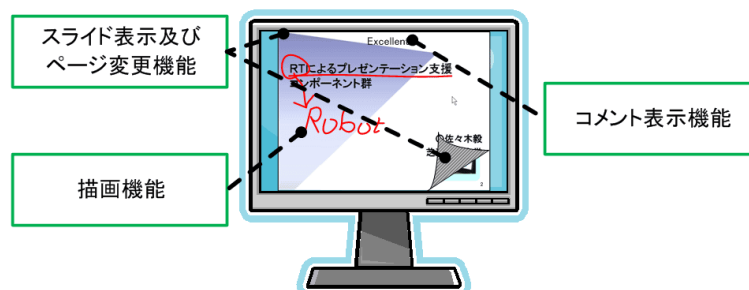
佐々木毅, “RT によるプレゼンテーション支援コンポーネント群”, 第 13 回計測自動制御学会システムインテグレーション部門講演会, pp.498–501, 2012.

に詳細がありますので、そちらもご参照いただけましたら幸いです。

## 1.2 プレゼンテーションコンポーネントの機能と用途

プレゼンテーションはスライドを使用するものが一般的であり、それが必須となっている場合もあることから、プレゼンテーションコンポーネントはスライドの表示を主な機能としていますが、様々な RT 要素を組み合わせることで機能を拡張することが可能です。プレゼンテーションコンポーネントには以下のような機能があります。

- スライド表示及びページ変更機能
- スライド内への描画機能
- コメント（テキスト）表示機能



これらの機能を用いて、例えば以下のように用いることができます。

- ジェスチャ認識や音声認識によるスライドの操作
- 手の位置を計測するセンサとの組み合わせによる手の動きに合わせたスライド内への描画
- ガイドロボットやプレゼンテーションロボットへの展開

## 1.3 開発環境

本コンポーネント群は Linux 及び Windows にて動作確認を行っています。開発環境は以下の通りです。

- Ubuntu Linux 12.04LTS / Windows 7 Service Pack 1 (ともに 32bit 版)
- RT ミドルウェア: OpenRTM-aist-1.1.0-RELEASE (C++版)

- コンパイラ: gcc 4.6.3-1 / Microsoft Visual C++ 2010 Express
- CORBA: omniORB 4.1.6-1 / 4.1.5
- Eclipse: Eclipse 3.4.2 + OpenRTM Eclipse tools 1.1.0-RC3
- CMake: CMake 2.8.7-0 / CMake 2.8.8
- Java 実行環境: openjdk-6-jre 6b24-1.11.4 / Oracle Java Version 6 Update 37

また、開発したコンポーネントのうち、プレゼンテーションコンポーネント (CVPresentation)、マウスコンポーネント (PresentationMouseDecoder)、レーザポインタインタフェースコンポーネント (PresentationLPInterface) については依存ライブラリとして OpenCV を利用しています。

- OpenCV 2.4.2

## 1.4 本稿で利用しているコンポーネント群

プレゼンテーションコンポーネントと基本的な操作を提供するものとして、以下のコンポーネント群を開発しました。

- プレゼンテーションコンポーネント (CVPresentation)  
プレゼンテーションコンポーネント。
- キーボードコンポーネント (PresentationKeyDecoder)  
キーボードによるスライドページの変更を可能にするコンポーネント。
- マウスコンポーネント (PresentationMouseDecoder)  
マウスによるスライドページの変更とスライド内への描画を可能にするコンポーネント。
- コメント入力コンポーネント (ConsoleInString)  
コンソールから文字列を入力することで、コメントの送信を可能にするコンポーネント。

また、プレゼンテーションコンポーネントに提供する機能の例を示すものとして、以下のコンポーネント群を開発しました。

- タイマーコンポーネント (PresentationCommentTimer)  
プレゼンテーションの時間を計測するためのコンポーネント。
- レーザポインタインタフェースコンポーネント (PresentationLPInterface)  
カメラコンポーネントと併用することで、レーザポインタによるスライドページの変更とスライド内への描画を可能にするコンポーネント。

それぞれのコンポーネントの詳細については2章を、使用方法については3章を参照してください。

## 2 コンポーネントの説明

### 2.1 基本コンポーネント群

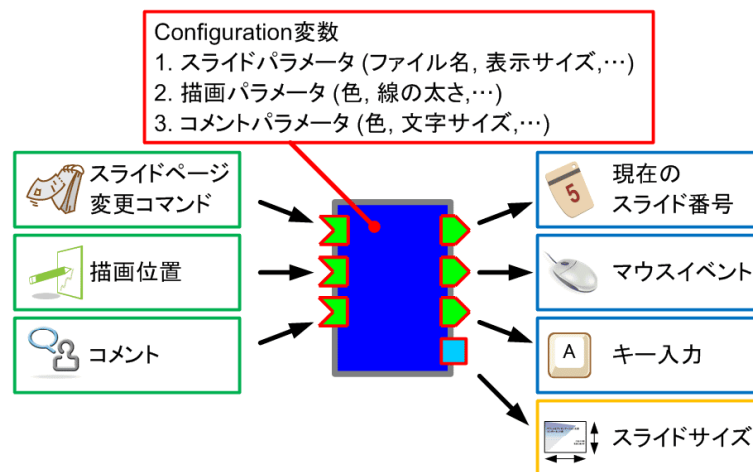
基本コンポーネント群は、中心となるプレゼンテーションコンポーネントと、キーボードやマウスといった従来のインタフェースによる基本的な操作を提供するコンポーネント群です。

### 2.1.1 プレゼンテーションコンポーネント (CVPresentation)

CVPresentation は、スライドを用いたプレゼンテーションへの RT 導入を支援するコンポーネントです。コンポーネントの概要を次ページの図に示します。

本コンポーネントはアクティブ化すると、Configuration で指定したスライド（画像ファイル）を画面上に表示します。各ページに相当する画像ファイルには通し番号を付ける形となっており、これをコマンドに応じて順に表示していくようになっています。また、スライド内に強調したい部分や補足事項をその場で書き加えることが可能な、ペンの機能を持っています。さらに、聴衆の側からも意見を発したり、発表者からの質問に答えたりすることが可能なように、スライドにコメントとしてテキストを表示することが可能です。

本コンポーネントはそれぞれ 3 つの InPort と OutPort を持っています。InPort の 1 つは、入力値に応じてスライドのページ送りをを行うためのポートです。他の InPort は、スライド内に描画する線の位置や、表示するコメントを指示するためのものです。なお、描画する線の色や太さ、コメントの色や大きさなどのパラメータは、Configuration から変更することが可能です。一方、OutPort からは、スライドのページが変更された場合のスライド番号、スライド上で操作したマウスイベント（左ボタンが押された等の情報とその位置）及びキーボード入力がそれぞれ出力されます。さらに、本コンポーネントはサービスポートとして、表示しているスライドのサイズを取得するためのインタフェースを備えています。



#### ● InPort

名称	型	説明
SlideNumberIn	TimedShort	表示するスライドの番号。Configuration を指定すれば、スライドの絶対番号 (例: test003.jpg の場合は 3) で指定することも、現在のスライドからの相対的な番号で指定することも可能。
Pen	TimedShortSeq	スライドに表示する線分・点の座標 $(x_1, y_1), (x_2, y_2), \dots$ 。Configuration で指定されたパラメータで、座標が 2 点以上の場合には指定された座標の間に線分を、座標が 1 点のみの場合にはその座標に点を描画する。
Comment	TimedString	スライドに表示するコメント（テキスト）。文字の大きさやサイズはコメントにコマンドを含めることで指定することが可能。指定がない場合は Configuration で指定されたパラメータで表示する。有効なのは半角英数字のみ。

- OutPort

名称	型	説明
SlideNumberOut	TimedShort	現在のスライド番号。スライド番号が変更された場合に出力される。
Key	TimedChar	スライドのウィンドウがアクティブのときにキーボードから入力された値。
MouseEvent	TimedShortSeq	スライド上で発生したマウスイベント (CV_EVENT_*: 左ボタンが押された、など)、その座標 ( $x,y$ )、フラグ (CV_EVENT_FLAG_*: 左ボタンが押されている、など)。

- Configuration 変数

名称	型	デフォルト値	説明
SlideFilePath	string	./	スライドが保存されているパス。アクティブ化時に読み込みを行う。アクティブ化後の変更は無効（反映されない）。
SlideFileName	string	test***.jpg	スライドの名前。通し番号の部分はアスタリスク (*) としておく。例えば、スライド画像のファイル名が “test1.jpg”, “test2.jpg”, ..., “test10.jpg”, ... の場合には “test*.jpg” を指定する。“test001.jpg”, “test002.jpg”, ..., “test010.jpg”, ... のように通し番号が最低 3 桁で表されている場合は “test***.jpg” を指定する。なお、スライドは OpenCV でサポートされている画像形式である必要がある。アクティブ化後の変更は無効（反映されない）。
SlideFileInitialNumber	int	1	スライド番号の初期値。例えば、最初のスライドが “test1.jpg” の場合には 1 を指定する。アクティブ化後の変更は無効（反映されない）。
SlideNumberInRelative	int	1	InPort の SlideNumberIn からスライド番号をスライドの絶対番号 (例: test3.jpg の場合は 3) で指定する場合は 0、現在のスライドからの相対的な番号で指定する場合は 1。
SlideSizeWidth	int	0	表示するスライドの幅。単位はピクセル。SlideSizeWidth と SlideSizeHeight がともに 0 の場合、元々の画像サイズでスライドを表示する。一方が正で他方が 0 の場合は、正の値のみを利用し、元の画像と同じアスペクト比になるようにスライドを表示する。アクティブ化後の変更は現在のスライドが変更されたときに反映される。
SlideSizeHeight	int	0	表示するスライドの高さ。単位はピクセル。他の説明は SlideSizeWidth を参照。
CommentColorRGB	vector<int>	0,0,0	コメントの文字の色 (RGB) のデフォルト値。
CommentSize	double	1.0	コメントの文字の大きさ (スケール) のデフォルト値。
CommentBaseSpeed	int	4	コメントの文字が流れる基本速度。
PenColorRGB	vector<int>	255,0,0	ペンの色 (RGB)。
PenThickness	int	4	ペンの太さ。

- サービスポート

SlideSizeInfo (Provided): スライドの画像の幅と高さを取得するためのインタフェース。

関数名	引数	戻り値の型	説明
getWidth	なし	short	スライド画像の幅を返す。単位はピクセル。
getHeight	なし	short	スライド画像の高さを返す。単位はピクセル。

### 2.1.2 キーボードコンポーネント (PresentationKeyDecoder)

PresentationKeyDecoder は、キーボードによるスライドページの変更を可能にするコンポーネントです。InPort の Key から文字を読み込み、Configuration で指定したキーコードに相当する文字が得られた場合に、プレゼンテーションコンポーネントのスライドページ変更コマンドに対応する数値を OutPort の SlideRelativeCommand から出力します。

- InPort

名称	型	説明
Key	TimedChar	入力文字。

- OutPort

名称	型	説明
SlideRelativeCommand	TimedShort	スライドのページを変更するためのコマンドに対応する数値。InPort の Key への入力文字が Configuration の Next のいずれかに一致するときは 1、Previous のいずれかに一致するときは -1、Reload のいずれかに一致するときは 0。同じキーコードが複数の Configuration で指定されていた場合、優先順位は Next, Previous, Reload の順となる。

- Configuration 変数

名称	型	デフォルト値	説明
Next	vector<int>	10,13,32,110	ここに書かれたキーコードに相当する文字が入力されたときにスライドを次のページへ進めるコマンド (= 1) を出力する。
Previous	vector<int>	8,112	ここに書かれたキーコードに相当する文字が入力されたときにスライドを前のページへ戻すコマンド (= -1) を出力する。
Reload	vector<int>	99,114	ここに書かれたキーコードに相当する文字が入力されたときにスライドの現在のページを再読み込みするコマンド (= 0) を出力する。

- サービスポート

なし



### 2.1.3 マウスコンポーネント (PresentationMouseDecoder)

PresentationMouseDecoder は、マウスによるスライドページの変更とスライド内への描画を可能にするコンポーネントです。InPort の MouseEvent から値を読み込み、プレゼンテーションコンポーネントのスライドページ変更コマンドとペン描画位置に対応する数値を OutPort の SlideRelativeCommand と DrawPositions からそれぞれ出力します。

- InPort

名称	型	説明
MouseEvent	TimedShortSeq	マウスイベント (CV_EVENT_*: 左ボタンが押された、など)、その座標 ( $x,y$ )、フラグ (CV_EVENT_FLAG_*: 左ボタンが押されている、など)。

- OutPort

名称	型	説明
SlideRelativeCommand	TimedShort	スライドのページを変更するためのコマンドに対応する数値。
DrawPositions	TimedShortSeq	ペンで描画する位置に対応する数値。

- Configuration 変数

名称	型	デフォルト値	説明
SlideChangeClick	int	1	この値が 1 の場合、左シングルクリックに相当するマウスイベントが入力されたときにスライドを次のページへ進めるコマンド (=1) を出力し、右シングルクリックに相当するマウスイベントが入力されたときにスライドを前のページへ戻すコマンド (=−1) を出力する。この値が 2 の場合、シングルクリックではなくダブルクリックの場合に上記と同様の出力となる。

- サービスポート

なし

### 2.1.4 コメント入力コンポーネント (ConsoleInString)

ConsoleInString は、OpenRTM-aist にサンプルとして取り上げられている ConsoleIn コンポーネントの OutPort の変数の型を TimedString にしたものです。コンソールから入力された文字列を OutPort の String から出力します。

- InPort

なし

- OutPort

名称	型	説明
String	TimedString	コンソールから入力された文字列。

- Configuration 変数

なし

- サービスポート

なし

## 2.2 機能拡張の紹介で用いるコンポーネント群

### 2.2.1 タイマーコンポーネント (PresentationCommentTimer)

PresentationCommentTimer は、プレゼンテーションの時間を計測するためのコンポーネントです。アクティブ化後、InPort の MeasurementTrigger にデータが入力されると時間計測を開始し、以後 InPort の MeasurementTrigger にデータが入力される度に Configuration の OutputType で指定された形式で計測結果を OutPort の MeasuredTimeString から出力します。

- InPort

名称	型	説明
MeasurementTrigger	TimedShort	アクティブ化後に初めて何かデータが入力された場合、計測を開始する。また、何かデータが入力される度に Configuration の OutputType で指定された形式で計測結果を OutPort の SplitTimeString から出力する。

- OutPort

名称	型	説明
MeasuredTimeString	TimedString	<p>Configuration の OutputType で指定される出力形式に応じて以下のように計測時間を文字列にしたもの。(lap_time...現在のスライドの表示時間、split_time...計測開始からの累計時間、remaining_time...プレゼンテーションの残り時間、presentation_time...Configuration の PresentationTime で設定したプレゼンテーションの時間)</p> <ul style="list-style-type: none"><li>• lap の場合: lap_time</li><li>• split の場合: split_time</li></ul> <p>(Configuration の PresentationTime に正の値が指定されている場合は split_time/presentation_time)</p> <ul style="list-style-type: none"><li>• split+lap の場合: split_time (lap_time)</li></ul> <p>(Configuration の PresentationTime に正の値が指定されている場合は split_time/presentation_time (lap_time))</p> <ul style="list-style-type: none"><li>• remaining の場合: remaining_time</li><li>• remaining+lap の場合: remaining_time (lap_time)</li></ul>

- Configuration 変数

名称	型	デフォルト値	説明
OutputType	string	split	OutPort の MeasuredTimeString からの結果の出力形式。lap の場合はラップタイム（前の出力からの経過時間）、split の場合はスプリットタイム（計測開始からの累計時間）、split+lap の場合はスプリットタイムとラップタイム、remaining の場合は残り時間（Configuration の PresentationTime から計測開始からの累計時間を引いたもの）、remaining+lap の場合は残り時間とラップタイム。
PresentationTime	double	0.0	プレゼンテーションの時間。Configuration の OutputType で指定される出力形式が remaining または remaining+lap の場合には、この時間から計測開始からの累計時間を引いたものが出力される。また、この値が正で、出力形式が split または split+lap の場合、累積時間に加え、この時間も出力される。

- サービスポート

なし

### 2.2.2 レーザポインタインタフェースコンポーネント (PresentationLPInterface)

PresentationLPInterface は、カメラコンポーネントと併用することで、レーザポインタによるスライドページの変更とスライド内への描画を可能にするコンポーネントです。ただし、カメラはスライドを表示しているスクリーンやディスプレイを  $z = c$ （一定）の平面とし、表示されているスライドの左上を  $(0, 0, c)$ 、右下を  $(slide\_width-1, slide\_height-1, c)$ （ $slide\_width$  は表示されているスライドの幅（ピクセル）、 $slide\_height$  は表示されているスライドの高さ（ピクセル））としてキャリブレーションされているとします。InPort の Image から読み込んだ画像から、輝度が閾値（Configuration の DetectLuminanceDiff）以上大きくなった点を検出候補点とし、最も明るく変化した点をレーザポインタで指されたスライド上の位置として認識します。さらに、その位置に応じてプレゼンテーションコンポーネントのスライドページ変更コマンドとペン描画位置に対応する数値を OutPort の SlideRelativeCommand と DrawPositions からそれぞれ出力します。なお、カメラ画像の入力には NEDO 次世代ロボット知能化技術開発プロジェクトによる共通カメラインタフェース (<http://www-arailab.sys.es.osaka-u.ac.jp/CameraIF/index.php?Common%20Camera%20Interface>) を用いています。

- InPort

名称	型	説明
Image	TimedCameraImage	入力画像及び座標変換のためのカメラパラメータ。
BgUpdateTrigger	TimedShort	何かデータが入力された場合、背景画像を更新する。

- OutPort

名称	型	説明
SlideRelativeCommand	TimedShort	スライドのページを変更するためのコマンドに対応する数値。
DrawPositions	TimedShortSeq	ペンで描画する位置に対応する数値。

- Configuration 変数

名称	型	デフォルト値	説明
DetectLuminanceDiff	int	30	入力画像のある点の輝度が背景画像のその点の輝度よりもこの値以上大きい場合、検出の候補点となる。
DetectBgUpdate	int	5	閾値処理で得られた候補点の数がこの値以上の場合、背景は大きく変化してしまっているものと考え、背景を更新する。ただし、この値が負の場合は、何点検出されても背景を更新しない。
DetectUpdateTime	double	0.5	最新の検出から次の検出までの時間がこの値以下の場合、2つの点は連続したものとして考える。単位は秒。(毎ステップ確実に検出できるとは限らないため、プレゼンテーションコンポーネントで線を描画した場合に途切れ途切れになることを避ける等の目的で使用。)
ScreenZ	double	0.0	スクリーン平面の $z$ 座標。スクリーンを $z = (\text{一定})$ の平面と考え、単眼カメラで位置を出すために使用。
SlideChangeArea	string	top	スライドのページを変更するためのコマンドを生成する領域を定義する。この値が <b>top</b> の場合は画像の上端、 <b>left</b> の場合は画像の左端、 <b>right</b> の場合は画像の右端、 <b>bottom</b> の場合は画像の下端が基準となる。
SlideChangeAreaSize	int	50	スライドのページを変更するためのコマンドを生成する領域を定義する。単位はピクセル。Configuration の SlideChangeArea で指定した基準からこの値で指定された範囲までの領域 (例えば、SlideChangeArea が <b>top</b> 、SlideChangeAreaSize が 10 なら画像の上端から 10 画素の範囲、つまり $10 \times$ 画像幅の矩形領域) を範囲とする。
SlideChangeCount	int	5	Configuration の SlideChangeArea と SlideChangeAreaSize で指定された領域のある部分で SlideChangeCount 回以上連続して点が検出された場合に、スライドを変更するコマンドを生成する。命令の種類は、領域の左側 1/3 (SlideChangeArea が <b>top</b> または <b>bottom</b> のとき) または上側 1/3 (left または right のとき) で検出された場合は前のスライドへ戻すコマンド ( $= -1$ )、中央 1/3 で検出された場合はスライドの現在のページを再読み込みするコマンド ( $= 0$ )、右側 1/3 (SlideChangeArea が <b>top</b> または <b>bottom</b> のとき) または下側 1/3 (left または right のとき) で検出された場合はスライドを次のページへ進めるコマンド ( $= 1$ ) とする。
SlideChangeWait	double	1.0	このコンポーネントによってスライドが変更されたとき、この時間だけ何もせずに待つ。単位は秒。(スライドの変更が連続して起こったり、スライド変更直後の不要な描画を避けるため。)

- サービスポート

SlideSizeInfo (Required): スライドの画像の幅と高さを取得するためのインタフェース。

提供される関数については 2.1.1 節プレゼンテーションコンポーネント (CVPresentation) のサービスポートを参照。

## 3 CVPresentation の使用方法

### 3.1 RTSystemEditor によるシステム構築の手順

RTSystemEditor を用いたシステム構築は通常、以下の手順で行われます。

RTSystemEditor を用いたシステム構築の手順

1. ネームサーバの起動
2. RTSystemEditor の起動
3. ネームサーバへの接続
4. コンポーネントの起動
5. システムの構築と実行

これらの手順はどのようなシステムでも共通ですので、操作方法は OpenRTM-aist のホームページ (<http://www.openrtm.org/>) を参照してください。以下の節では手順 5 に関して、CVPresentation の使用手順を説明します。

### 3.2 CVPresentation の使用手順

CVPresentation の使用手順をまとめると以下のようになります。

CVPresentation の使用手順

0. (コンポーネント実行前の準備) 表示させるスライド画像を作成し、通し番号を付けて保存しておく
1. Configuration を設定する
  - (a) SlideFilePath, SlideFileName, SlideFileInitialNumber でスライド画像が保存されている場所とスライド画像のファイル名を指定する
  - (b) SlideSizeWidth, SlideSizeHeight で画像の表示サイズを指定する (必要な場合のみ)
  - (c) 必要があれば他の Configuration についても設定する
2. コンポーネントをアクティブ化する
3. InPort, OutPort, ServicePort を他のコンポーネントに接続し、システムを拡張していく

手順 1(b), 1(c) の内容は手順 2 のアクティブ化後に設定・変更することも可能です。また、手順 3 の他のコンポーネントの接続についても、手順 1 や手順 2 の前に行っても構いません。

#### 3.2.1 スライド画像の準備

スライドは OpenCV で読み込み可能な画像形式 (JPEG, PNG, BMP など) で作成する必要があります\*。画像の表示サイズはプレゼンテーションコンポーネントで指定することが可能ですが、作成した画像を大きく拡大する場合などは画質が悪くなってしまうので注意が必要となります。また、表示するときにアスペクト比 (縦横比)

\*既存のプレゼンテーションソフトでスライドを作成する場合、例えば以下のようにスライド画像を作成することができます。

Microsoft PowerPoint を用いる場合: スライド作成後「名前を付けて保存」を選択し、ファイルの種類を JPEG や PNG 等の画像形式を選択して保存。

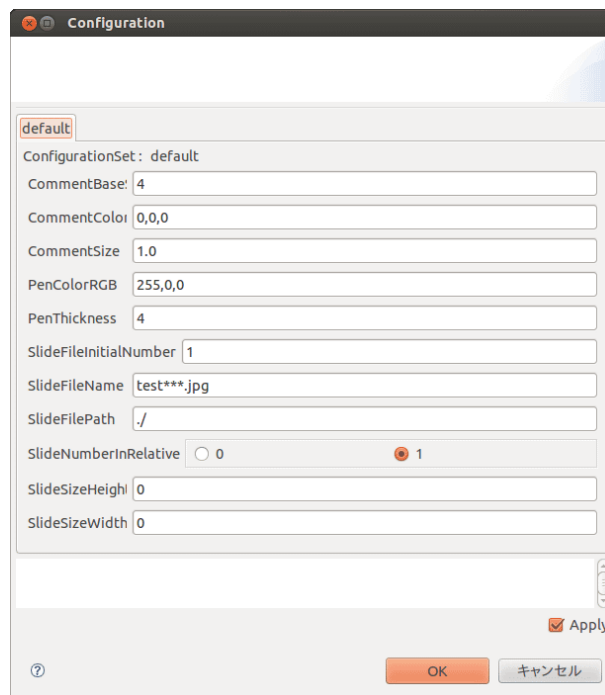
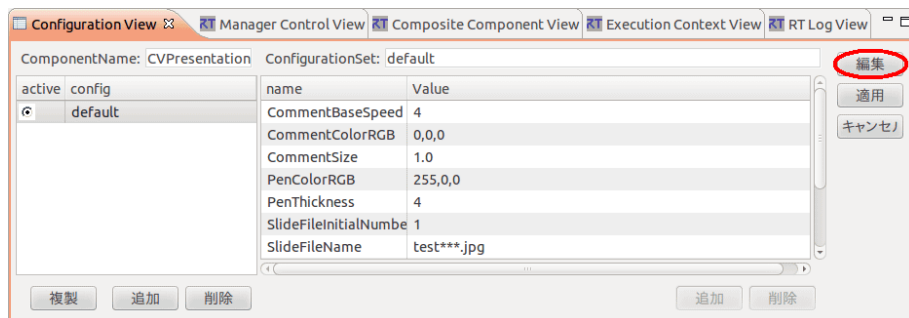
LibreOffice Impress を用いる場合: 1 ページだけ保存する場合は「エクスポート」を選択し、ファイルの種類を JPEG や PNG 等の画像形式にして保存。複数のスライド画像をまとめて保存する場合は、「エクスポート」を選択し HTML として保存する (HTML 内の画像ファイルとしてスライドの各ページが出力される) か、一度「PDF としてエクスポート」してから別のソフトを用いて画像に変換する。

を変更すると画像がつぶれてしまうため、表示したいアスペクト比でスライド画像を作成しておくことを推奨します。

画像を作成したら、表示したい順に通し番号をつけて保存しておきます。通し番号は、例えば test1.jpg, test2.jpg, ..., test10.jpg, ... のように付けます。test001.jpg, test002.jpg, ..., test010.jpg, ... のように桁数を固定しても、test1\_new.jpg, test2\_new.jpg, ... や 1test.jpg, 2test.jpg, ... のようにファイル名の途中や先頭につけても構いません。ただし、通し番号の固定の桁数が途中で変わるような付け方 (例: test001.jpg, test02.jpg, ...) や通し番号の位置が変わるような付け方 (例: test001.jpg, 002test.jpg, ...) はできません。

### 3.2.2 Configuration の設定

RTSystemEditor 上で CVPresentation コンポーネントを選択し、Configuration View の“Edit Value”（日本語版の場合は「編集」）をクリックすると、Configuration 変数の設定画面が表示されます<sup>†</sup>。



#### (a) スライド画像の指定 (SlideFilePath, SlideFileName, SlideFileInitialNumber)

まず、SlideFilePath にスライド画像が保存されているフォルダを指定します。絶対パスで指定することも、相対パスで指定することも可能です。次に、SlideFileName にスライド画像のファイル名を指定します。このとき、通し番号の部分はアスタリスク (\*) とします。例えば、スライド画像が test1.jpg, test2.jpg, ..., test10.jpg, ... の場

<sup>†</sup>このような設定画面が表示されない場合、RTSystemEditor が最新版でない可能性があります。OpenRTM-aist のホームページ (<http://www.openrtm.org/>) をご確認ください。

合には test\*.jpg を指定します。スライド画像が test001.jpg, test002.jpg, ..., test010.jpg, ... のように通し番号が3桁で表されている場合は test\*\*\*.jpg を指定します。最後に、通し番号の最初の値を指定します。最初のスライド画像が test1.jpg や test001.jpg の場合には 1 を設定します。通し番号が test0.jpg のように 1 以外の値から始まっている場合は、その値を設定してください。

設定を確定するには、設定画面右下の “Apply” にチェックが入っていることを確認し、OK をクリックします。

#### (b) 画像の表示サイズの指定 (SlideSizeWidth, SlideSizeHeight)

SlideSizeWidth で幅、SlideSizeHeight で高さを指定することで、表示する画像のサイズを設定することができます。SlideSizeWidth と SlideSizeHeight がともに 0 の場合、保存されている画像の元々のサイズで表示を行います。また、一方の値が正で、もう一方の値が 0 の場合は、正の値のみを利用し、元の画像と同じアスペクト比になるようにスライドを表示します。例えば、保存されている画像のサイズが  $800 \times 600$  で SlideSizeWidth に 400、SlideSizeHeight に 0 を設定した場合、幅が半分になっているため、高さもそれに合わせて半分 (300) になります。同様に、SlideSizeWidth に 0、SlideSizeHeight に 300 を設定した場合、高さが半分になっているため、幅もそれに合わせて半分 (400) になります。

設定を確定するには、設定画面右下の “Apply” にチェックが入っていることを確認し、OK をクリックします。

#### (c) 他の Configuration の設定

他のパラメータについても用途に応じて設定が必要となる場合があります。内容については、パラメータに関連する機能ごとに 3.3 節で説明します。

### 3.2.3 コンポーネントのアクティブ化

Configuration の設定が終了したら、RTSystemEditor 上で CVPresentation コンポーネントを右クリックし、Activate を選択してアクティブ化します。スライド画像のウィンドウが表示されれば完了です。スライド画像の指定に誤りがある場合は、“Cannot find initial slide” というメッセージがコンソールに表示され、コンポーネントがエラー状態になります。そのときはファイルの保存場所、ファイル名や拡張子、通し番号の初期値に誤りがないかを確認してください。誤りを修正したら、コンポーネントを右クリックして Reset を選択し、エラー状態を解除してからもう一度アクティブ化を行ってください。

## 3.3 CVPresentation の機能

本節では、InPort を利用した CVPresentation の機能の使い方について紹介します。

### 3.3.1 スライドのページ変更

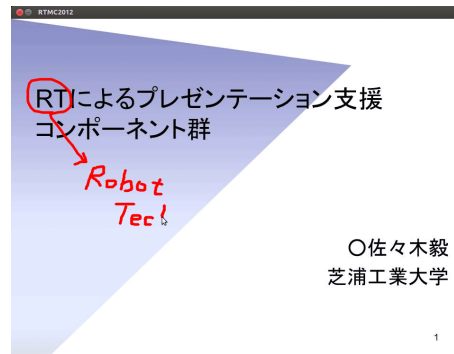
スライドのページ変更は InPort の SlideNumberIn に値を入力することで行えます。ページ番号の指定方法は現在のスライドからの相対値で指定する方法と、絶対番号で指定する方法があり、Configuration の SlideNumberInRelative で変更することができます。

相対値で指定する場合は SlideNumberInRelative を 1 に設定します。このとき、例えば現在のスライドの通し番号が 3 であれば、4 を入力すると通し番号が  $7 (= 3 + 4)$  のページが、-2 を入力すると通し番号が  $1 (= 3 + (-2))$  のページが表示されます。また、0 を入力すると、現在のスライドページの再読み込み (3.3.2 節で紹介する描画機能による描画内容の消去) を行うことができます。

絶対値で指定する場合は SlideNumberInRelative を 0 に設定します。このときは、現在のスライドの通し番号の値にかかわらず、例えば 4 を入力すると通し番号が 4 のページになります。また、現在のスライドの通し番号と同じ値が入力された場合、現在のスライドページの再読み込み (3.3.2 節で紹介する描画機能による描画内容の消去) が行われます。

### 3.3.2 スライド内への描画

InPort の Pen に座標を入力することによって、スライド内に描画を行うことができます。座標はスライドの左上が (0,0)、右下が (slide\_width, slide\_height) (slide\_width は表示されているスライドの幅 (ピクセル)、slide\_height は表示されているスライドの高さ (ピクセル)) です。入力された座標が 1 点のみ ((x,y)) の場合はその座標に点を、座標が 2 点以上 ((x<sub>1</sub>,y<sub>1</sub>), (x<sub>2</sub>,y<sub>2</sub>), ...) の場合は指定された座標の間に線分を描画します。描画パラメータは Configuration の PenColorRGB, PenThickness で行えます。PenColorRGB には描画する点や線分の色の RGB 値をカンマ区切りで指定します。例えば、赤であれば 255,0,0、青であれば 0,0,255、白であれば 255,255,255 を設定します。PenThickness には点の大きさや線分の太さを指定します。値は正の整数で、値が大きくなるほど大きく/太くなります。



### 3.3.3 コメント (テキスト) の表示

InPort の Comment に文字列を入力することによって、スライドにコメント (テキスト) を表示することができます。対応している文字は半角英数字です。コメントはスライドの上部に表示され、右から左へ移動していきます。

入力文字列に文字の色、大きさを指定するためのコマンドを含めると、その書式で表示を行うことができます。(指定がない場合は Configuration で指定したデフォルト値で表示します。) コマンドを指定する場合は、入力する文字列の先頭を “[” (半角スペースと [) とし、コマンドの最後に “]” をつけ、続けて表示したい文字列を入力します。

\_[コマンド] 表示したい文字列

ただし、\_は半角スペースを表すものとします。コマンド部分には、文字の色を指定するためのコマンド、文字の大きさを指定するコマンド、もしくはその両方をカンマ区切りで並べたもののいずれかを設定します。

色を指定するためのコマンドは、“RGB(Rの成分,Gの成分,Bの成分)”です。例えば、赤であれば RGB(255,0,0)、青であれば RGB(0,0,255)、白であれば RGB(255,255,255) となります。また、いくつかの色については予め色名が定義されており、この名前を用いて色を指定することも可能です。利用可能な色名は下記のとおりです。

コマンド	等価なコマンド	コマンド	等価なコマンド
black	RGB(0,0,0)	aqua	RGB(0,255,255)
gray	RGB(128,128,128)	lime	RGB(0,255,0)
silver	RGB(192,192,192)	yellow	RGB(255,255,0)
white	RGB(255,255,255)	fuchsia	RGB(255,0,255)
blue	RGB(0,0,255)	olive	RGB(128,128,0)
navy	RGB(0,0,128)	purple	RGB(128,0,128)
teal	RGB(0,128,128)	maroon	RGB(128,0,0)
green	RGB(0,128,0)	red	RGB(255,0,0)



大きさを指定するためのコマンドは、“SIZE(大きさの値)”です。値は正の実数で、値が大きくなるほど大きくなります。また、色名と同様、予め定義されたサイズ名があり、これでサイズを指定することも可能です。利用可能なサイズ名は下記のとおりです。

コマンド	意味
big	Configuration の CommentSize の 2 倍のスケールで表示
small	Configuration の CommentSize の 0.5 倍のスケールで表示

SIZE(大きさの値) として指定した場合は絶対的な大きさの指定になるのに対し、これらのコマンドでは Configuration で指定したデフォルト値に対する相対的な値になる点に注意してください。

コマンド指定の例を以下に示します。ただし、`_`は半角スペースを表すものとします。

- ABCDE という文字列を赤色、大きさ 2.0 で表示

`_ [red, SIZE(2.0)] ABCDE`

(red の部分は RGB(255,0,0) としても、2.0 の部分は 2 としても同じです。)

- ABCDE という文字列を赤色、大きさは Configuration で指定したデフォルト値の 2 倍で表示

`_ [red, big] ABCDE`

(red の部分は RGB(255,0,0) としても同じですが、big はデフォルト値の 2 倍という意味なので、1 つ前の例のように SIZE(2.0) として絶対指定した場合とは異なります。)

- ABCDE という文字列を RGB=(127,255,0)、大きさは Configuration で指定したデフォルト値で表示

`_ [RGB(127,255,0)] ABCDE`

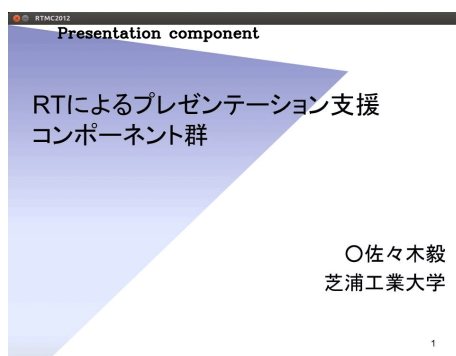
- ABCDE という文字列を色は Configuration で指定したデフォルト値で、大きさは Configuration で指定したデフォルト値の 0.5 倍で表示

`_ [small] ABCDE`

- ABCDE という文字列を色・大きさともに Configuration で指定したデフォルト値で表示

`ABCDE`

その他の表示パラメータや書式コマンドがない場合の文字の色、大きさは Configuration の CommentColorRGB, CommentSize, CommentBaseSpeed で行えます。CommentColorRGB には文字の色の RGB 値をカンマ区切りで指定します。例えば、赤であれば 255,0,0、青であれば 0,0,255、白であれば 255,255,255 を設定します。CommentSize には文字の大きさを指定します。値は正の実数で、値が大きくなるほど大きくなります。CommentBaseSpeed にはコメントの移動速度（右から左へ流れる速度）を指定します。値は正の整数で、値が大きくなるほど速くなります。



## 4 CVPresentation の利用例

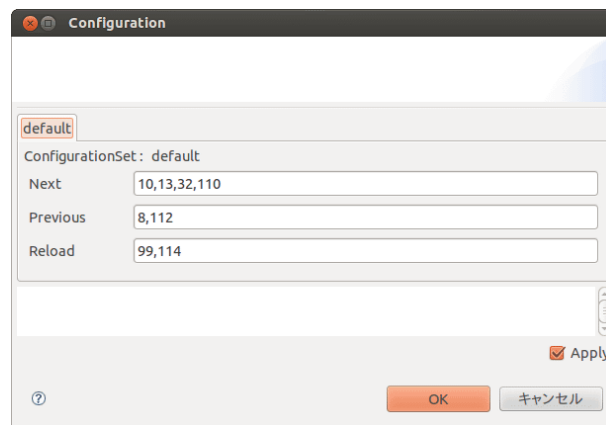
ここでは、提供しているコンポーネント群を用いた CVPresentation の機能の利用例を紹介します。各コンポーネントごとに説明をしていきますが、2 つ以上のコンポーネントを同時に利用することも可能です。

### 4.1 基本コンポーネント群を用いた CVPresentation の機能の利用

#### 4.1.1 キーボードの追加

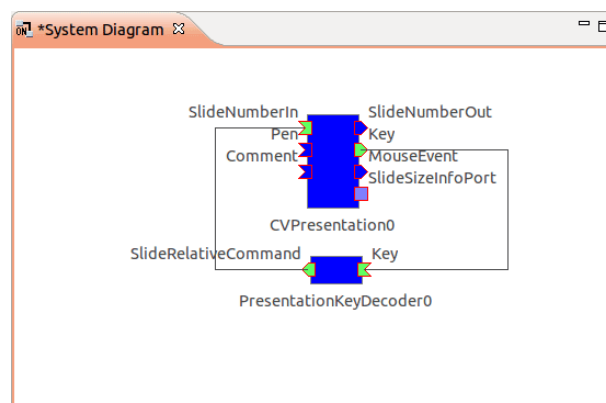
キーボードコンポーネント (PresentationKeyDecoder) を接続することにより、キーボードでスライドのページを変更できるようになります。

キーの設定は PresentationKeyDecoder の Configuration で行うことができます。Next にはスライドを次のページへ進めるためのキーのキーコードを指定します。(デフォルトでは LF, CR, Space, 'n' が設定されています。) Previous にはスライドを前のページへ戻すためのキーのキーコードを指定します。(デフォルトでは Back Space, 'p' が設定されています。) Reload にはスライドの現在のページを再読み込みするためのキーのキーコードを指定します。(デフォルトでは 'c', 'r' が設定されています。)



システムエディタ上で CVPresentation と PresentationKeyDecoder のポートを以下の通りに接続します。なお、PresentationKeyDecoder はスライドの変更指令を相対値で行うため、CVPresentation の Configuration の SlideNumberInRelative を 1 に設定しておく必要があります。

CVPresentation		PresentationKeyDecoder
SlideNumberIn (InPort)	–	SlideRelativeCommand (OutPort)
Key (OutPort)	–	Key (InPort)

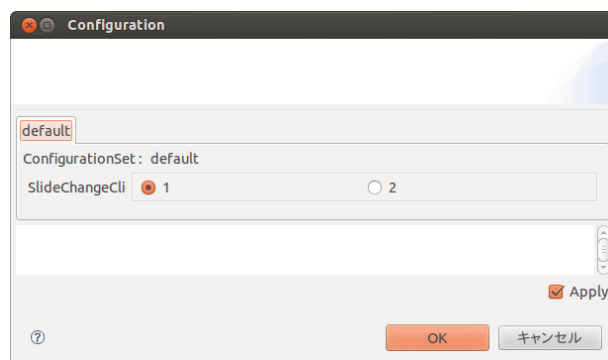


コンポーネントをアクティブ化すれば、指定したキーでスライドページの変更を行うことができます。

#### 4.1.2 マウスの追加

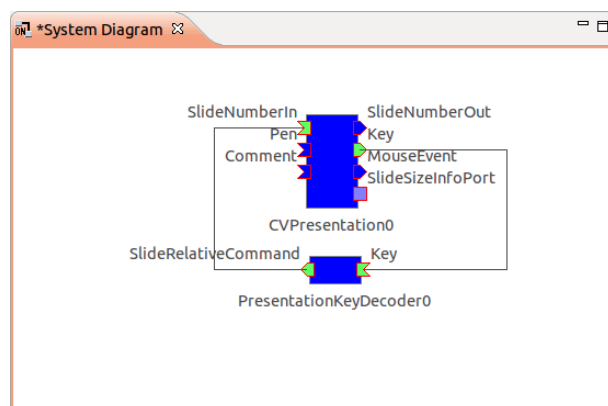
マウスコンポーネント (PresentationMouseDecoder) を接続することにより、マウスでスライドのページを変更したり、スライド内へ描画を行ったりできるようになります。

設定は PresentationMouseDecoder の Configuration で行うことができます。SlideChangeClick にはスライドページ変更コマンドに対応するクリック回数を指定します。1 にするとシングルクリックで、2 にするとダブルクリック<sup>‡</sup>でスライドのページ変更を行うことができますようになります。



システムエディタ上で CVPresentation と PresentationMouseDecoder のポートを以下の通りに接続します。SlideNumberIn (InPort) – SlideRelativeCommand (OutPort) の接続はスライドのページの変更を可能にするため、Pen (InPort) – DrawPositions (OutPort) の接続はスライド内への描画を可能にするためのものです。どちらかの機能のみを利用する場合は一方しか接続しなくても構いません。なお、ページ変更機能を利用する場合、PresentationMouseDecoder はスライドの変更指令を相対値で行うため、CVPresentation の Configuration の SlideNumberInRelative を 1 に設定しておく必要があります。

CVPresentation		PresentationKeyDecoder
SlideNumberIn (InPort)	–	SlideRelativeCommand (OutPort)
Pen (InPort)	–	DrawPositions (OutPort)
MouseEvent (OutPort)	–	MouseEvent (InPort)



コンポーネントをアクティブ化すれば、指定したクリックでスライドページの変更を行うことができます。左ボタンのクリック/ダブルクリックでスライドを次のページへ進めることができ、右ボタンのクリック/ダブルク

<sup>‡</sup>Windows 版 OpenCV では、version 2.4.2 現在、ダブルクリックの検出に対応していません。そのため Windows で動作させる場合はシングルクリックを指定する必要があります。

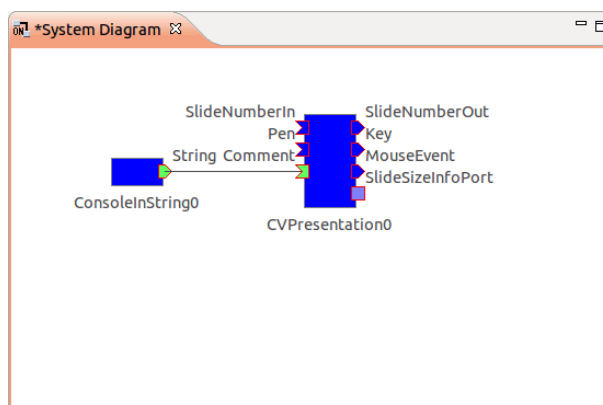
リックでスライドを前のページへ戻すことが可能です。また、マウスの左ボタンを押しながら移動させると、スライド内に線の描画を行うことができます。

### 4.1.3 コメントの表示

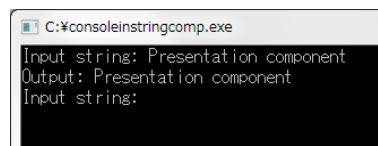
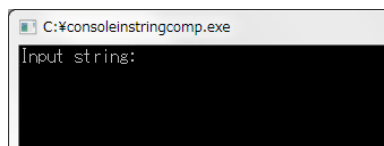
コメント入力コンポーネント (ConsoleInString) を接続することにより、コンソールから入力した文字列をコメントとして表示できるようになります。

システムエディタ上で CVPresentation と ConsoleInString のポートを以下の通りに接続します。

CVPresentation		ConsoleInString
Comment (InPort)	–	String (OutPort)



コンポーネントをアクティブ化すると、ConsoleInString のコンソールに Input String: と表示されます。表示させたい文字列を入力して Enter で決定すると、その文字列がスライド上に表示されます。

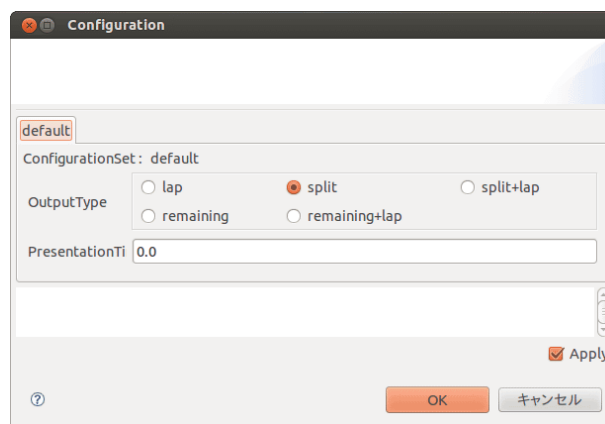


## 4.2 様々なコンポーネントによる機能の拡張

### 4.2.1 プレゼンテーション時間の計測

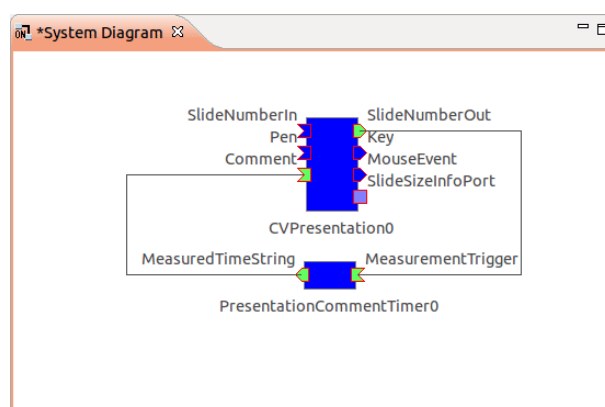
タイマーコンポーネント (PresentationCommentTimer) を接続することにより、プレゼンテーションの時間を計測し、それをコメントとして表示できるようになります。

時間の表示はスライドのページが切り替わったり、再読み込みを行ったりするたびに行われます。設定は PresentationCommentTimer の Configuration で行うことができます。OutputType には時間の表示形式を指定します。lap にすると、各スライドごとの表示時間（前に時間を表示した時から現在までの時間）を表示します。split にすると、計測開始からの累計時間を表示します。split+lap にすると、計測開始からの累計時間と各スライドごとの表示時間の両方を表示します。remaining にすると、プレゼンテーションの残り時間を表示します。remaining+lap にすると、プレゼンテーションの残り時間と各スライドごとの表示時間の両方を表示します。ただし、プレゼンテーションの残り時間の表示を行う際には、Configuration の PresentationTime にプレゼンテーション時間を設定する必要があります。PresentationTime には、プレゼンテーションの時間を分で指定します（例えば 5 分 30 秒の場合は 5.5 とします）。この値は remaining または remaining+lap で表示する場合に必要となります。また、この値を設定しておくと、split または split+lap で表示する場合にプレゼンテーション時間が表示されます。



システムエディタ上で CVPresentation と PresentationCommentTimer のポートを以下の通りに接続します。

CVPresentation		PresentationCommentTimer
Comment (InPort)	–	MeasuredTimeString (OutPort)
SlideNumberOut (OutPort)	–	MeasurementTrigger (InPort)

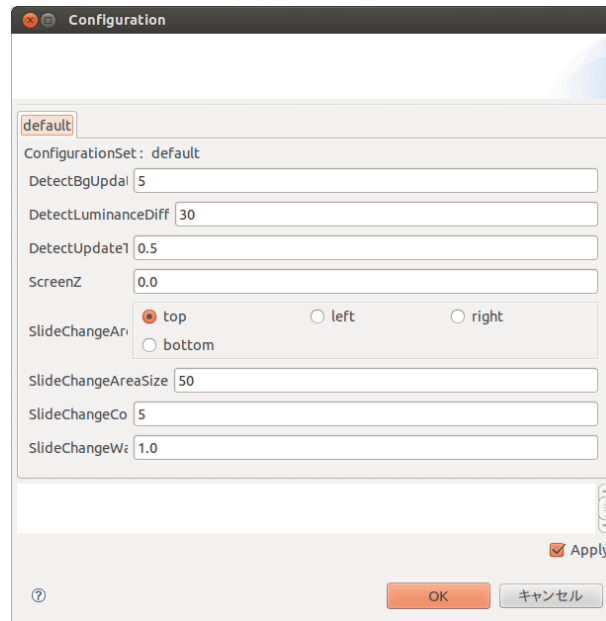


コンポーネントをアクティブ化すれば、プレゼンテーション時間の計測を行えます。PresentationCommentTimer がアクティブされた後に初めて InPort の MeasurementTrigger にデータを入力すると時間の計測が開始されます。ですので、例えば 4.1.1 節で述べたキーボードコンポーネント (PresentationKeyDecoder) を接続するなどし、スライドページの再読み込みをさせることで時間計測を開始するとよいでしょう。

#### 4.2.2 レーザポインタを用いたプレゼンテーション

レーザポインタインタフェースコンポーネント (PresentationLPIInterface) とカメラコンポーネントを接続することにより、レーザ光で指示した位置に線を描画したり、指定した領域内を一定の間隔で指示することでスライドの変更を行ったりできるようになります。本コンポーネントでは、カメラで取得した画像からレーザポインタの光を検出し、その位置をカメラ座標系からスクリーン座標系に変換します。そして、その結果に応じてスライドのページ変更や描画の指示を行います。そのため、カメラがスライドを表示する画面を視野にとらえていることと、そのカメラがキャリブレーションされていることが必要です。カメラコンポーネントとの接続については、NEDO 次世代ロボット知能化技術開発プロジェクトによる共通カメラインタフェース (<http://www-arailab.sys.es.osaka-u.ac.jp/CameraIF/index.php?Common%20Camera%20Interface>) を利用しており、キャリブレーションのカメラパラメータもこれに従っています。なお、外部パラメータ (カメラの位置・姿勢) については、スライドを表示しているスクリーンやディスプレイを  $z = c$  (一定) の平面とし、表示されているスライド

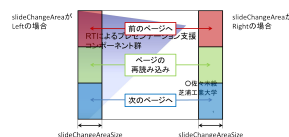
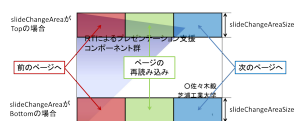
の左上を  $(0, 0, c)$ 、右下を  $(slide\_width-1, slide\_height-1, c)$  ( $slide\_width$  は表示されているスライドの幅 (ピクセル)、 $slide\_height$  は表示されているスライドの高さ (ピクセル)) としてキャリブレーションを行ってください。  
設定は PresentationLPInterface の Configuration で行うことができます。



ScreenZ にはキャリブレーションで用いたスクリーン平面の  $z$  座標 (=  $c$ ) の値を設定します。

検出のパラメータは DetectLuminanceDiff, DetectBgUpdate, DetectUpdateTime で設定します。基本的にはデフォルトのままで構いませんが、検出が困難な際には DetectLuminanceDiff を調整する必要があります。DetectLuminanceDiff には、レーザ光の検出の閾値を設定します。入力画像のある点の輝度が背景画像のその点の輝度よりもこの値以上大きい場合、検出の候補点となります。そのため、室内が明るいなどレーザ光の検出が難しい場合にはこの値を小さく、レーザ光以外の場所も検出してしまう場合にはこの値を大きくすることになります。最終的には候補点の中で輝度が最も大きく変化した点を検出点とします。DetectBgUpdate には、背景の更新条件を設定します。本来レーザ光は 1 点のみのはずですが、照明条件などによっては、複数の候補点が検出されてしまう場合があります。そこで、候補点の数がこの値以上検出されてしまう場合、背景は大きく変化してしまっているものと考え、背景更新処理を行います。この値が負の場合は、何点検出されても背景を更新しません。DetectUpdateTime には、あるフレームと別のフレームで検出された 2 点が連続したものとして考えるための時間を秒で設定します。最新の検出から次の検出までの時間がこの値以下の場合、2 つの点は連続したものとして考えます。これはカメラの毎フレームで確実にレーザ光を検出できるとは限らないため、プレゼンテーションコンポーネントで線を描画した場合に線が途切れ途切れになることを避ける目的で使います。

特定の領域にレーザ光を当てるとスライドのページを変更することができるようになります。SlideChangeArea, SlideChangeAreaSize には、この領域を指定します。下図のように、SlideChangeArea が top の場合は画像の上端、left の場合は画像の左端、right の場合は画像の右端、bottom の場合は画像の下端がその領域になり、SlideChangeAreaSize はその領域の大きさ (ピクセル) を指定します。また、対応するコマンドも図に示しています。例えば、SlideChangeArea が top、SlideChangeAreaSize が 10 の場合には、スライドの左上の上から 10 画素以内の範囲をポイントし続けるとスライドを前のページへ戻すことができます。このスライド変更までに必要な「ポイントし続ける」長さを SlideChangeCount で指定します。領域のある部分 (例えば、前のページへ戻すコマンドに対応する領域) でこの回数以上連続してレーザ光が検出された場合に、スライドを変更するコマンドを生成します。SlideChangeWait にはスライド変更直後にコマンドを受け付けられない時間を秒で指定します。これはスライドが変更されてから人間がそれを認識して変更領域を指すのを止めるまでに時間がかかるため、スライドの変更が連続して起こったり、スライド変更直後の不要な描画を避けるためです。

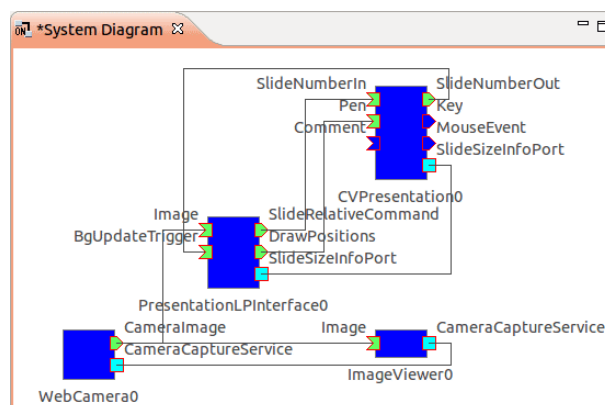


システムエディタ上で CVPresentation と PresentationLPInterface のポートを以下の通りに接続します。なお、PresentationLPInterface はスライドの変更指令を相対値で行うため、CVPresentation の Configuration の SlideNumberInRelative を 1 に設定しておく必要があります。

CVPresentation		PresentationLPInterface
SlideNumberIn (InPort)	–	SlideRelativeCommand (OutPort)
Pen (InPort)	–	DrawPositions (OutPort)
SlideNumberOut (OutPort)	–	BgUpdateTrigger (InPort)
SlideSizeInfo (Service Port)	–	SlideSizeInfo (Service Port)

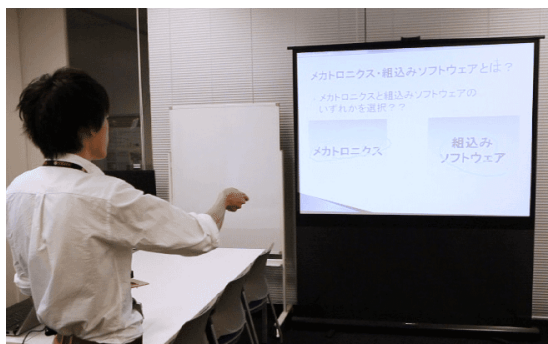
また、カメラ画像を PresentationLPInterface の Image (InPort) に入力する必要があります。なお、このデータにはキャリブレーションパラメータの情報も含まれています。下図では、これに加え、カメラ画像の表示やフレーム取得の指示を出すビューワコンポーネントも接続しています。

カメラコンポーネント		PresentationLPInterface
カメラ画像 (OutPort)	–	Image (InPort)



コンポーネントをアクティブ化すれば、レーザポインタによる指示位置の検出が行えるようになります。





検出精度については以下のような項目が影響します。検出が困難な場合は各項目について確認してみてください。

項目	検出しやすい		検出しにくい	備考
カメラの性能	高い	－	低い	数千円程度のウェブカメラでは検出は困難。上位機種であればウェブカメラでも可能。
レーザ光	明るい	－	暗い	
スライドの色	黒（暗い）	－	白（明るい）	プロジェクタの照度にも影響。
室内照明	暗い	－	明るい	

## 5 お問い合わせ

本コンポーネントにつきましては、まだ改善の余地があるものと考えております。ご要望、バグ報告、マニュアルの記述の不備等に関しましては、芝浦工業大学デザイン工学部デザイン工学科の佐々木までご連絡ください。

【問合せ先】

〒 135-8548

東京都江東区豊洲 3-7-5

芝浦工業大学 本部棟 6 階 06K10 佐々木研究室

Tel:03-5859-8834

sasaki-t at ieee. org