マイクロマシン / MEMS展
ROBOTECH 次世代ロボット製造技術展

RTミドルウェア講習会

RTミドルウェアインストールワークショップ

株式会社グローバルアシスト
坂本 武志

日時：2012年7月13日(金)　10:30～12:30
場所：東京ビッグサイト 東ホール 特設会場

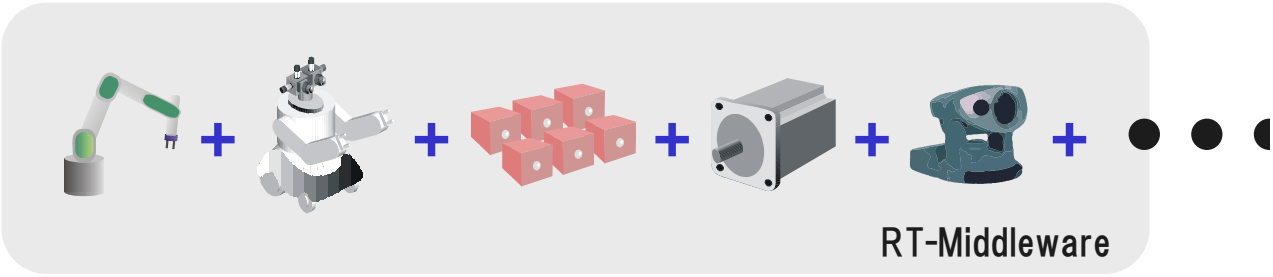
RT MIDDLEWARE

RTミドルウェアの概要

RT MIDDLEWARE

RTミドルウェアとは？

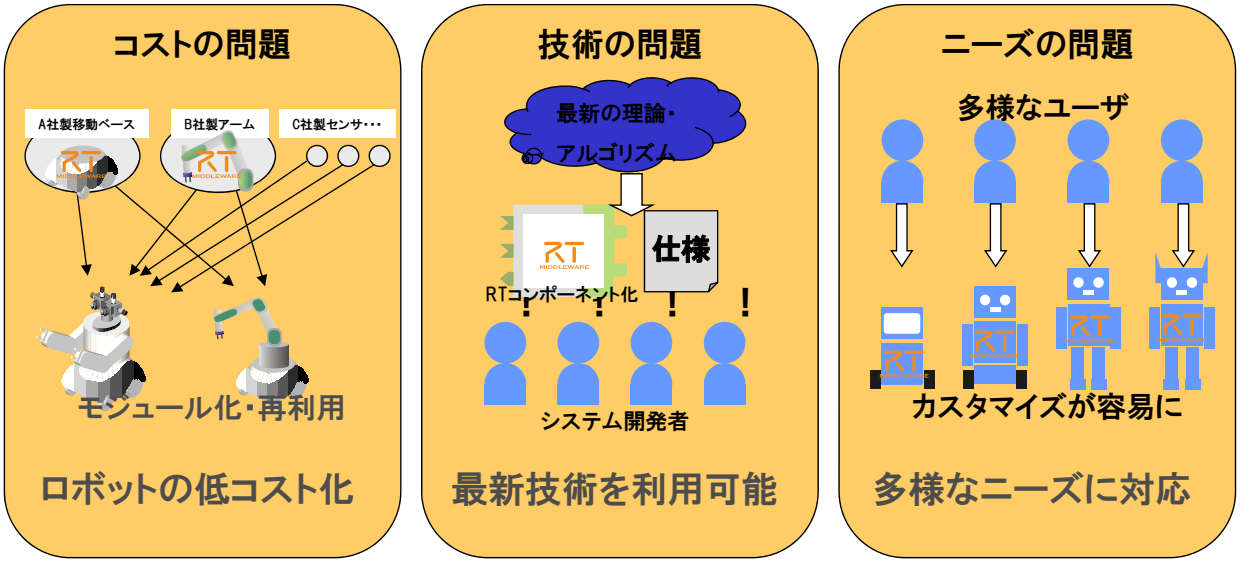
- RT = Robot Technology　　cf. IT
  - ≠Real-time
  - 単体のロボットだけでなく、さまざまなロボット技術に基づく機能要素をも含む（センサ、アクチュエータ, 制御スキーム、アルゴリズム、etc…）



- RT-Middleware（RTM）
  - RT要素のインテグレーションのためのミドルウエア
- RT-Component（RTC）
  - RT-Middlewareにおけるソフトウエアの基本単位

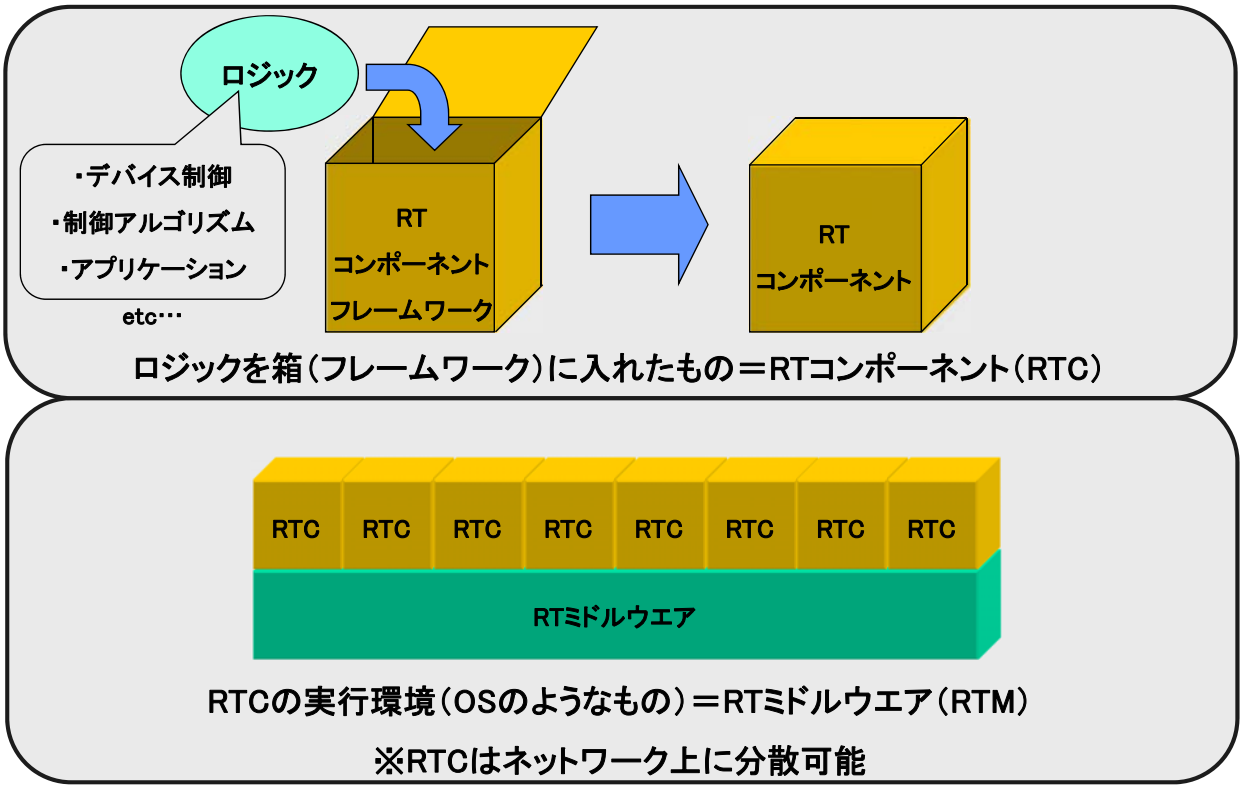
RTミドルウェアの目的

- モジュール化による問題解決



ロボットシステムインテグレーションによるイノベーション

RTミドルウェアとRTコンポーネント

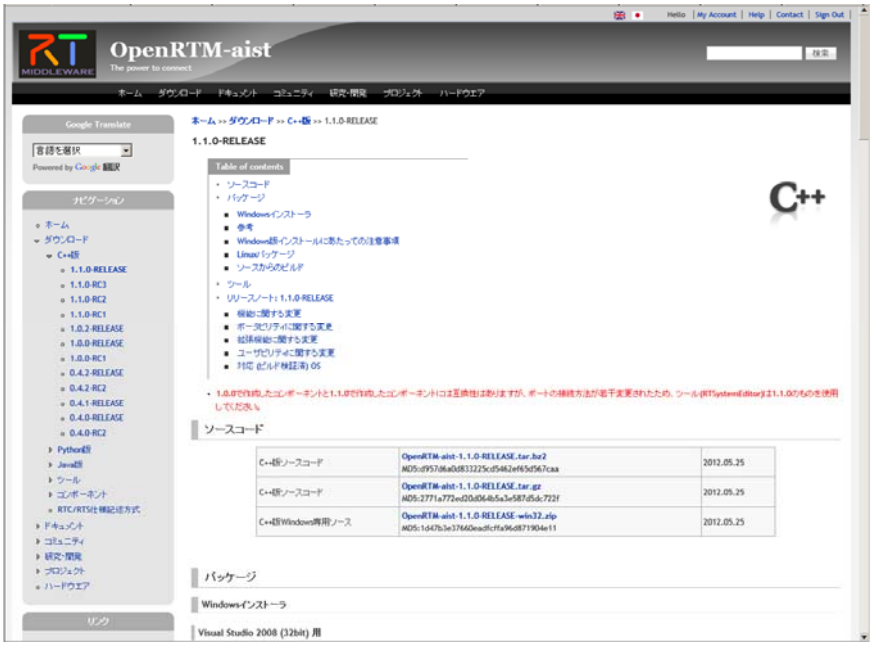


RTミドルウェアによる分散システム

RTミドルウェアのインストール

RTミドルウェアの取得先

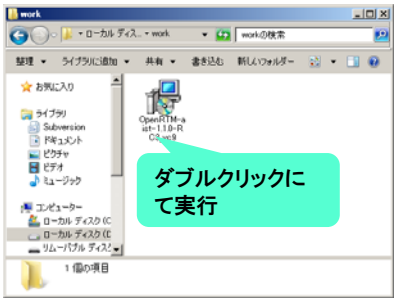
- OpenRTM-aist公式ページの「ダウンロード」ページ
  - http://www.openrtm.org/openrtm/ja/node/5012

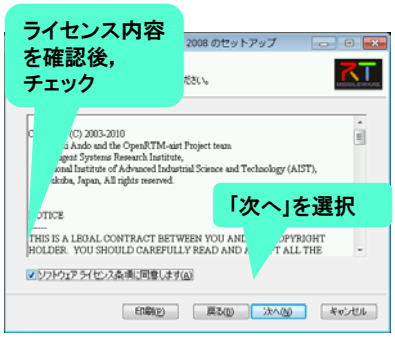


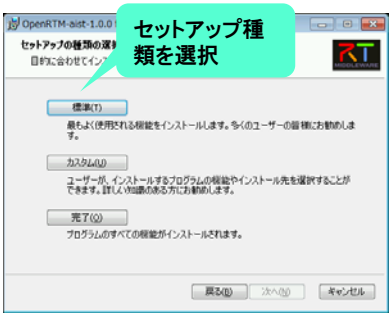
※使用環境に対応したパッケージをダウンロード

RTミドルウェアのインストール

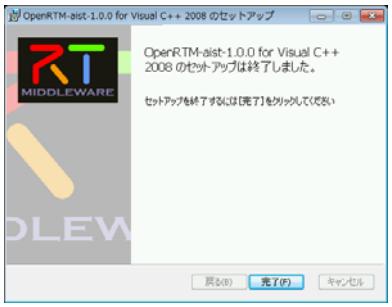
- ダウンロードしたインストーラを実行
  - OpenRTM-aist-1.1.0-RC3_vc9.msiなど



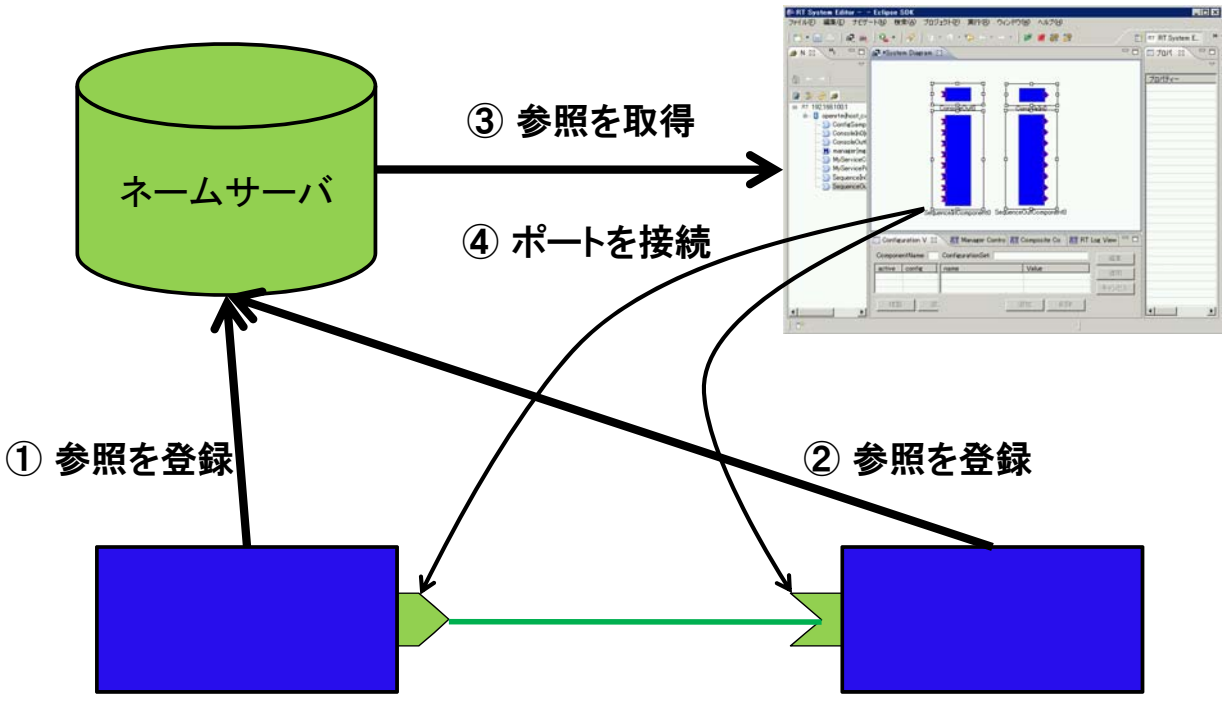




- 標準:全ての機能をインストール
- カスタム:インストールする機能を選択
- 完了:全ての機能をインストール（標準と同様）



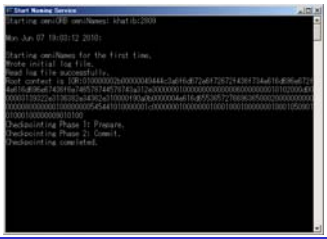
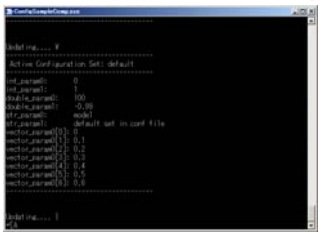
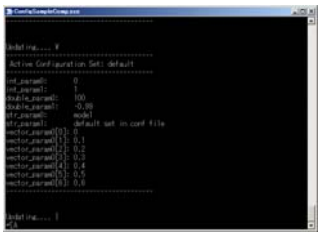
動作シーケンス



動作確認

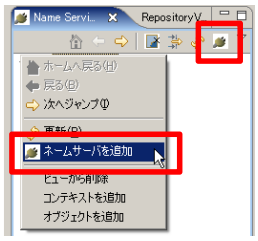
- Naming Serviceの起動
  - [スタート]メニューから
    [プログラム]→[OpenRTM-aist 1.1]→[C++]→[tools]→[Start Naming Service]
- CosoleInCompの起動
  - [スタート]メニューから起動
    [プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
    →[examples]→ [ConsoleInComp.exe]

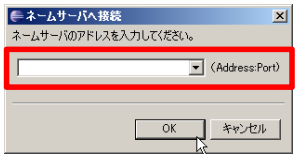
CosoleOutCompの起動
[スタート]メニューから起動
[プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
→[examples]→ [ConsoleOutComp.exe]

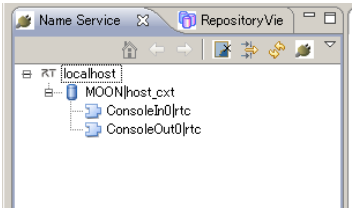
動作確認

- ツールの起動
  - [スタート]メニューから
    [プログラム]→[OpenRTM-aist 1.1]→[C++]→[tools]→[RTSystemEditor]
- ネームサービスへ接続

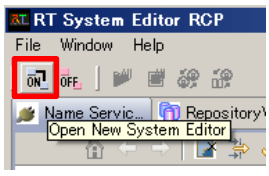


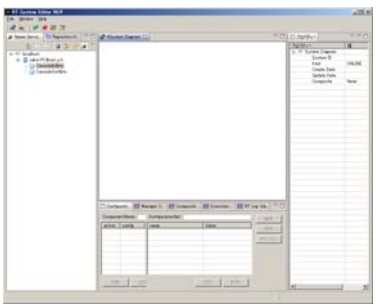


※対象ネームサーバのアドレス，ポートを指定
→ポート省略時のポート番号は
設定画面にて設定可能



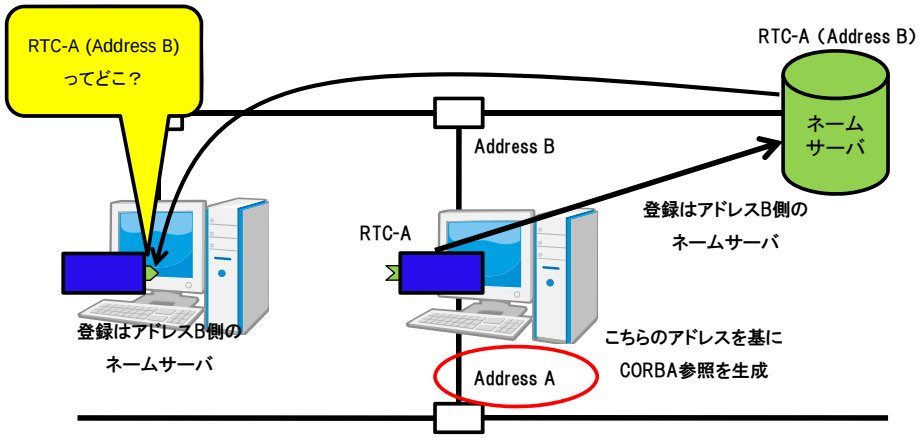
- システムエディタの起動





注意事項

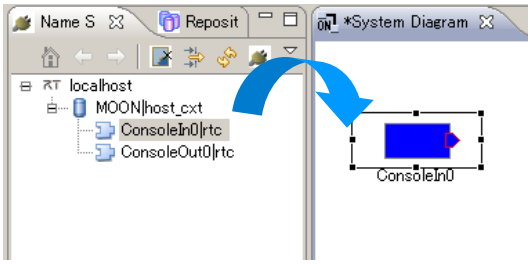
- ネットワークインターフェースが2つある場合



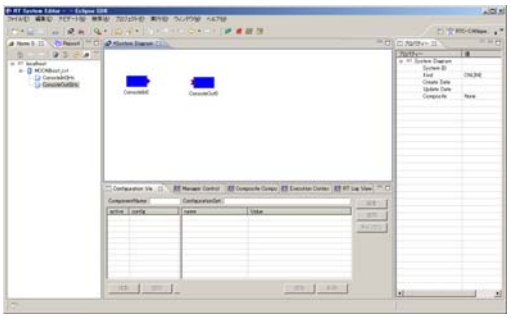
- RTC.confについて
  - RTC起動時の登録先NamingServiceや、登録情報などについて記述
  - 記述例：
    - corba.nameservers: localhost:9876
    - naming.formats: SimpleComponent/%n.rtc
    - corba.endpoints:192.168.0.12:

動作確認

- RTコンポーネントの配置

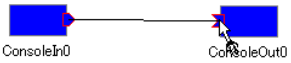


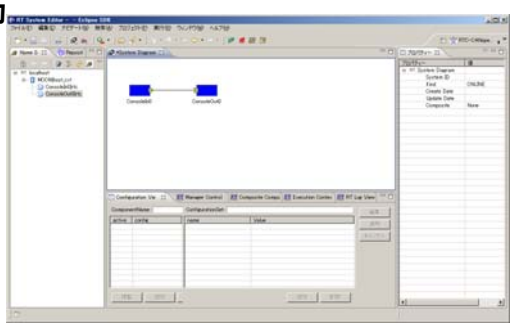
※ネームサービスビューから対象コンポーネントをドラッグアンドドロップ



- ポートの接続

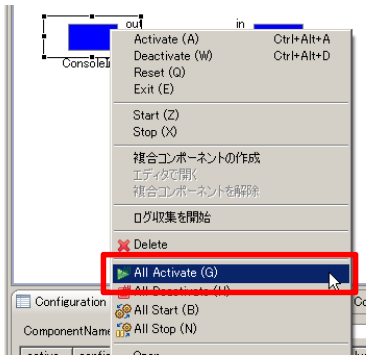
①接続元のポートから接続先の②接続プロファイルを入力ポートまでドラッグ

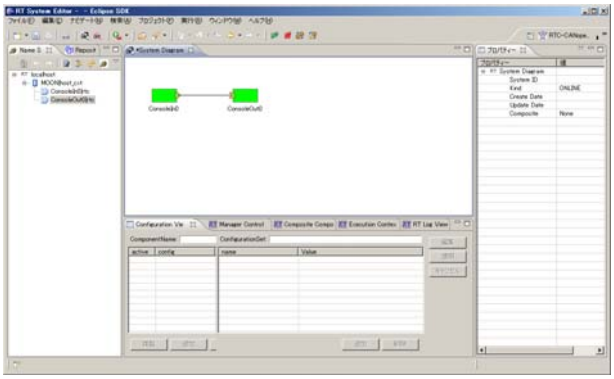




動作確認

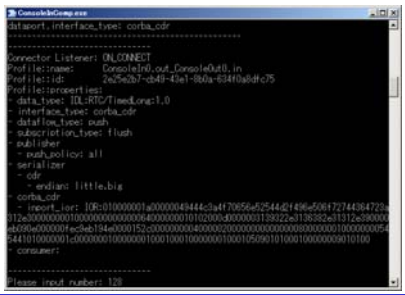
- コンポーネントの起動



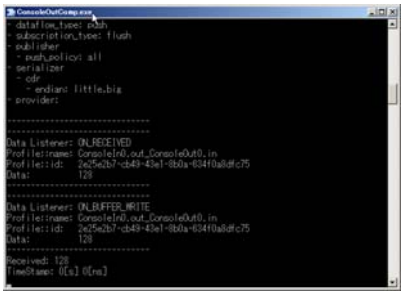


- 動作確認

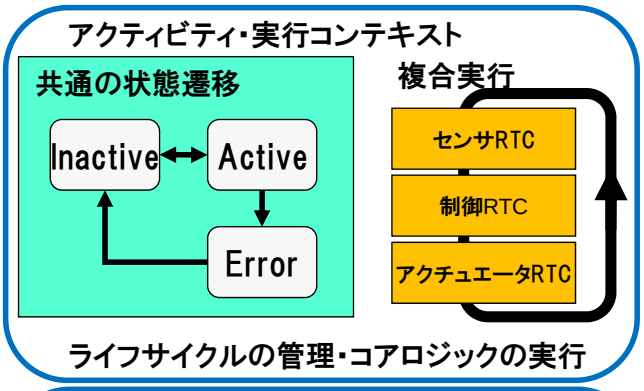
①ConsoleIn側で数字を入力

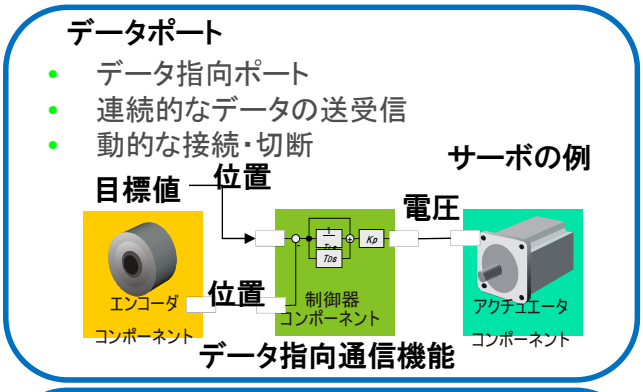


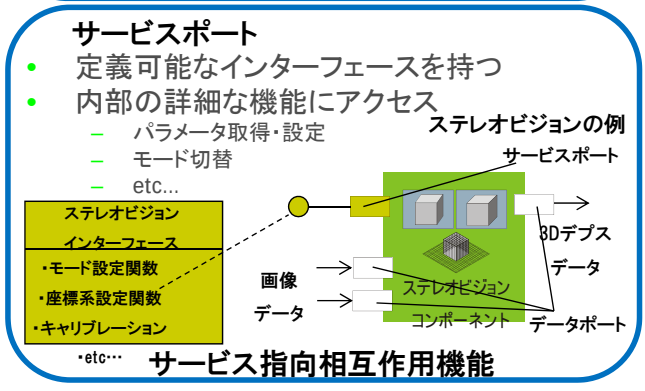
②ConsoleOut側が表示

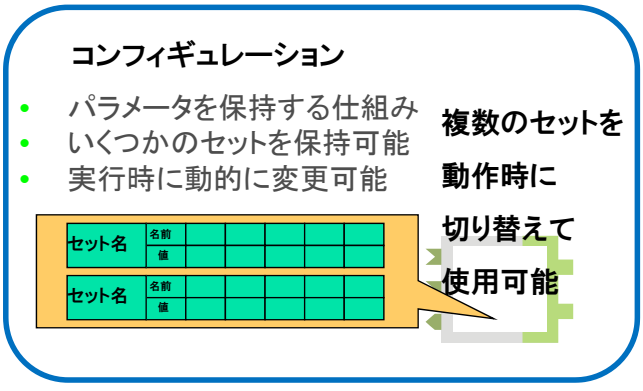


RTコンポーネントの主な機能









OpenRTM-aist

- コンポーネントフレームワーク + ミドルウエアライブラリ
- コンポーネントインターフェース:
  - OMG Robotic Technology Component Specification ver1.0 準拠
- OS
  - 公式:FreeBSD, Linux (Fedora, Debian, Ubuntu, Vine, Scientific), Windows
  - 非公式:Mac OS X, uITRON, T-Kernel, VxWorks
- 言語:
  - C++ (1.1.0), Python (1.0.0), Java (1.0.0)
  - .NET (implemented by SEC)
- CPU アーキテクチャ (動作実績):
  - i386, ARM9, PPC, SH4
  - PIC, dsPIC, H8 (RTC-Lite)
- ツール (Eclipse プラグイン)
  - テンプレートソースジェネレータ: rtc-template、RTCBuilder
  - システムインテグレーションツール: RTSystemEditor

実用化・事業化

DAQ-Middleware: KEK/J-PARC

KEK: High Energy Accelerator Research Organization

J-PARC: Japan Proton Accelerator Research Complex

HRP-4: Kawada/AIST

TAIZOU: General Robotics Inc.

HIRO: Kawada/GRX

HRP-4C: Kawada/AIST

OMG RTC ファミリ

Name	Vendor	Feature
OpenRTM-aist	AIST	C++, Python, Java
OpenRTM.NET	SEC	.NET(C#,VB,C++/CLI, F#, etc..)
miniRTC, microRTC	SEC	CAN・ZigBee等を利用した組込用RTC実装
Dependable RTM	SEC/AIST	機能安全認証 (IEC61508) capableなRTM実装
RTC CANOpen	SIT, CiA	CANOpenのためのCiA (Can in automation) におけるRTC標準
PALRO	富士ソフト	小型ヒューマノイドのためのC++ PSM 実装
OPRoS	ETRI	韓国国家プロジェクトでの実装
GostaiRTC	GOSTAI, THALES	ロボット言語上で動作するC++ PSM実装

同一標準仕様に基づく多様な実装により
- 実装(製品)の継続性を保証
- 実装間での相互利用がより容易に

RTミドルウェアの広がり

ダウンロード数

2012年2月現在

	2008年	2009年	2010年	2011年	2012年	合計
C++	4978	9136	12049	1851	253	28267
Python	728	1686	2387	566	55	5422
Java	643	1130	685	384	46	2888
Tool	3993	6306	3491	967	39	14796
All	10342	18258	18612	3768	393	51373

ユーザ数

タイプ	登録数
Webページユーザ	420 人
Webページアクセス	約 300 visit/day 約 1000 view/day
メーリングリスト	447 人
講習会	のべ 572 人
利用組織(Google Map)	46 組織

プロジェクト登録数

タイプ	登録数
RTコンポーネント群	603
RTミドルウエア	24
ツール	33
仕様・文書	4
ハードウエア	28

OMG RTC規格実装（11種類）

Name	Vendor	Feature
OpenRTM-aist	AIST	C++, Python, Java
OpenRTM.NET	SEC	.NET(C#,VB,C++/CLI, F#, etc..)
miniRTC, microRTC	SEC	CAN・ZigBee等を利用した組込用RTC実装
Dependable RTM	SEC/AIST	機能安全認証 (IEC61508) capableなRTM実装
RTC CANOpen	SIT, CiA	CANOpenのためのCiA (Can in automation) におけるRTC標準
PALRO	富士ソフト	小型ヒューマノイドのためのC++ PSM 実装
OPRoS	ETRI	韓国国家プロジェクトでの実装
GostaiRTC	GOSTAI, THALES	ロボット言語上で動作するC++ PSM実装

※上記のプロジェクト登録数の最新情報はOpenRTM-aistのホームページ(http://www.openrtm.org/)で確認できます

利用者のメリット・デメリット

- メリット
  - すぐに試せて，試したRTCをそのまま再利用が可能
  - フリーかつオープンソースであるため，RTミドルウエア自体をカスタマイズすることも可能(ライセンスに注意)
  - ネットワークを隠ぺいするので，分散システムが容易に開発できる
  - ユーザ向けのソフトウエア・インターフェースが決定できる
  - RTミドルウエア利用者には簡単に試してもらえる
  - ソフトウエアのドキュメントを簡潔にできる
- デメリット
  - ソフトウエアのオーバーヘッドは存在する
  - 初期導入時の時間的コスト
  - RTC開発自体のコスト
  - システムのチューニング作業は不可避

RTSystemEditorについて

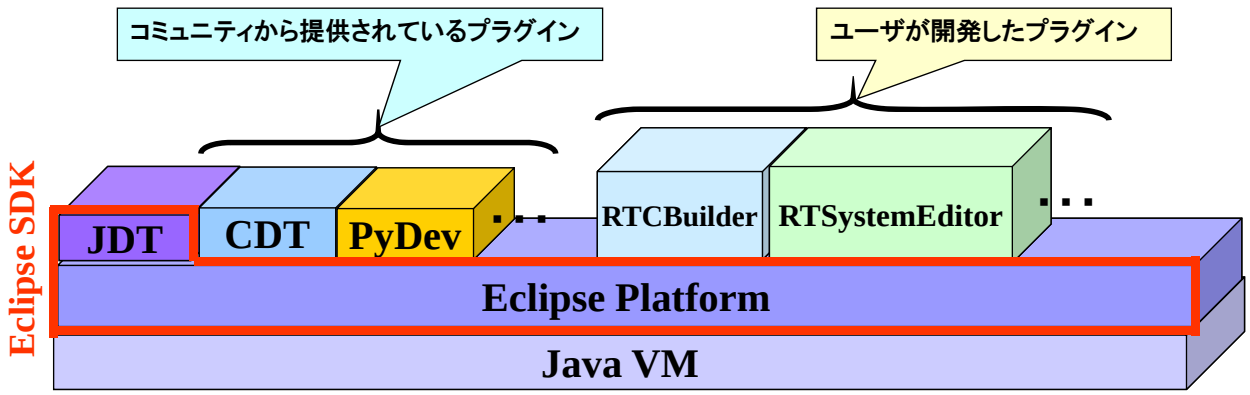
OpenRT Platform

- ロボット知能ソフトウェアプラットフォーム
  - http://www.openrtp.jp/wiki/
  - システム設計，シミュレーション，動作生成，シナリオ生成などをサポート
- OpenRT Platformツール群
  - コンポーネント開発，システム開発における各開発フェーズの作業支援
  - 開発プラットフォームにEclipseを採用
- 構成
  - RTCビルダ
  - RTCデバッガ
  - RTシステムエディタ
  - ロボット設計支援ツール
  - シミュレータ
  - 動作設計ツール
  - シナリオ作成ツール

など

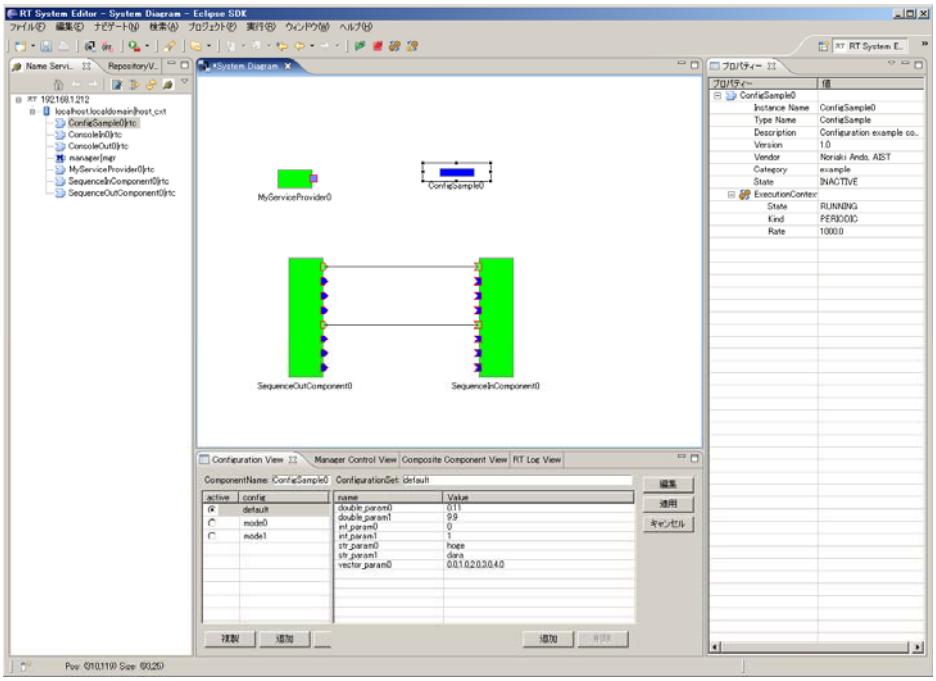
統合開発環境Eclipse

- オープンソース・コミュニティで開発されている統合開発環境
  - マルチプラットフォーム対応．WindowsやLinuxなど複数OS上で利用可能
  - 「Plug-in」形式を採用しており，新たなツールの追加，機能のカスタマイズが可能
  - RCP(Rich Client Platform)を利用することで，簡単に単独アプリ化が可能

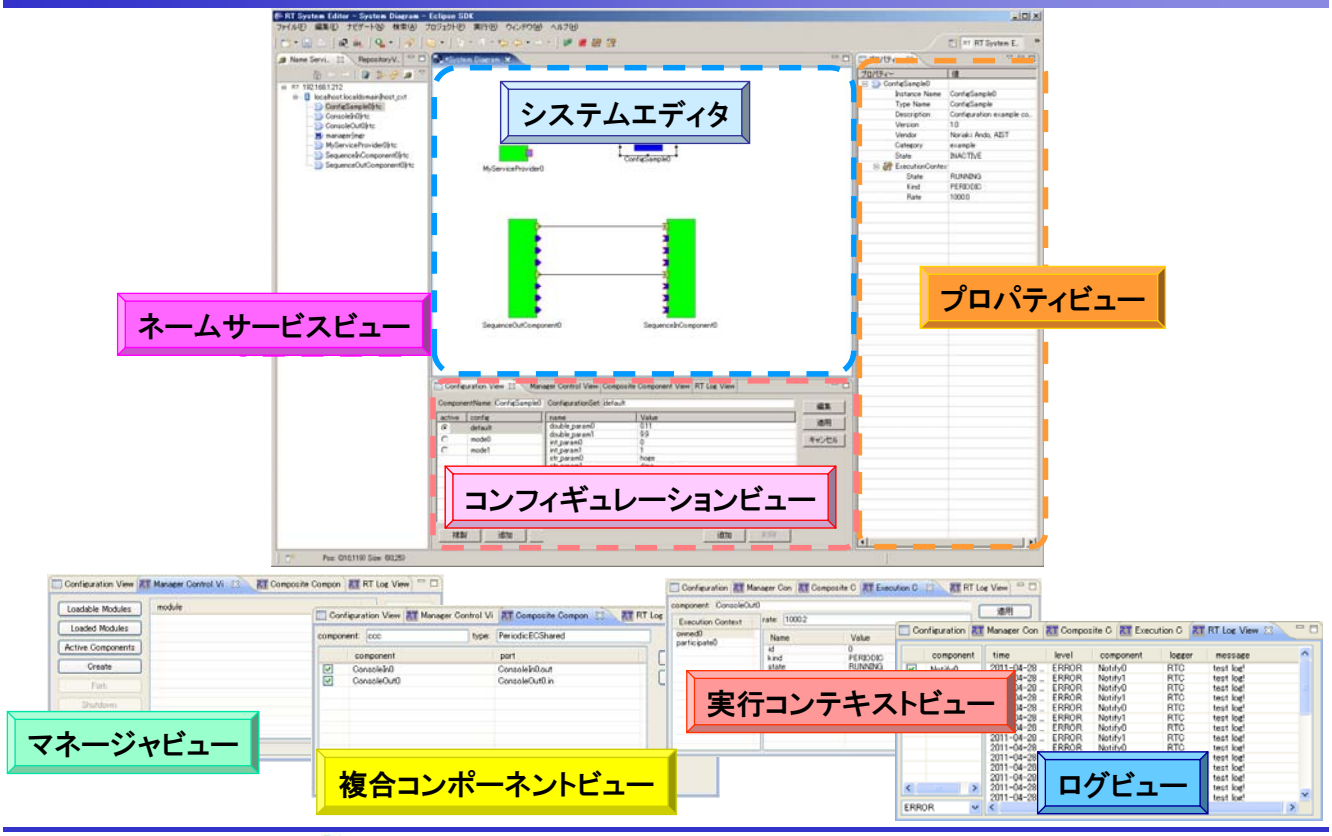


RTSystemEditor概要

- RTSystemEditorとは？
  - RTコンポーネントを組み合わせて，RTシステムを構築するためのツール



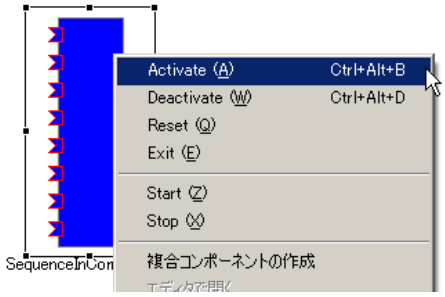
画面構成



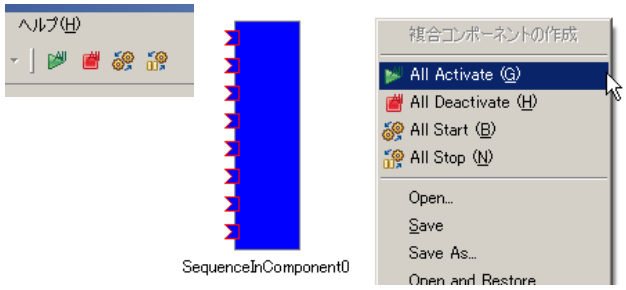
RTコンポーネントの動作

アクション名	説明
Activate	対象RTCを活性化する
Deactivate	対象RTCを非活性化する
Reset	対象RTCをエラー状態からリセットする
Exit	対象RTCの実行主体(ExecutionContext)を停止し，終了する
Start	実行主体(ExecutionContext)の動作を開始する
Stop	実行主体(ExecutionContext)の動作を停止する

■各コンポーネント単位での動作変更



■全コンポーネントの動作を一括変更



※ポップアップメニュー中でのキーバインドを追加
※単独RTCのActivate/Deactivateについては，グローバルはショートカットキー定義を追加

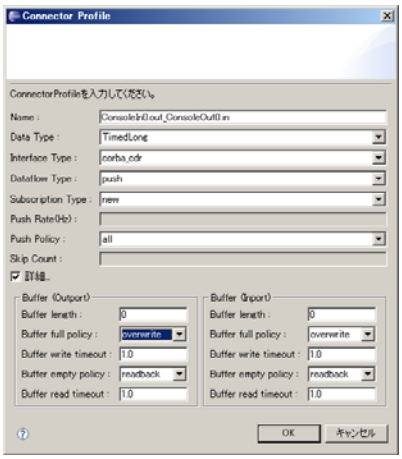
接続プロファイル(DataPort)について

項目	設定内容
Name	接続の名称
DataType	ポート間で送受信するデータの型. ex)TimedOctet,TimedShortなど
InterfaceType	データを送受信するポートの型. ex)corba_cdrなど
DataFlowType	データの送信方法. ex)push, pullなど
SubscriptionType	データ送信タイミング. 送信方法がPushの場合有効. New, Periodic, Flushから選択
Push Rate	データ送信周期(単位:Hz). SubscriptionTypeがPeriodicの場合のみ有効
Push Policy	データ送信ポリシー. SubscriptionTypeがNew, Periodicの場合のみ有効. all, fifo, skip, newから選択
Skip Count	送信データスキップ数. Push PolicyがSkipの場合のみ有効

- SubscriptionType
  - New ：バッファ内に新規データが格納されたタイミングで送信
  - Periodic ：一定周期で定期的にデータを送信
  - Flush ：バッファを介さず即座に同期的に送信
- Push Policy
  - all ：バッファ内のデータを一括送信
  - fifo ：バッファ内のデータをFIFOで1個ずつ送信
  - skip ：バッファ内のデータを間引いて送信
  - new ：バッファ内のデータの最新値を送信(古い値は捨てられる)

接続プロファイル(DataPort)について

項目	設定内容
Buffer length	バッファの大きさ
Buffer full policy	データ書き込み時に，バッファフルだった場合の処理. overwrite，do_nothing，blockから選択
Buffer write timeout	データ書き込み時に，タイムアウトイベントを発生させるまでの時間(単位:秒)
Buffer empty policy	データ読み出し時に，バッファが空だった場合の処理. readback，do_nothing，blockから選択
Buffer read timeout	データ読み出し時に，タイムアウトイベントを発生させるまでの時間(単位:秒)



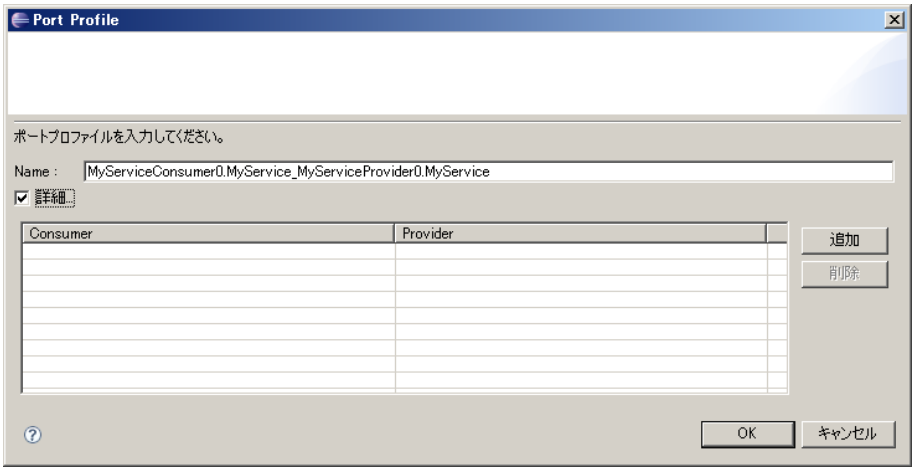
※OutPort側のバッファ，InPort側のバッファそれぞれに設定可能
※timeoutとして「0.0」を設定した場合は，タイムアウトしない

- Buffer Policy
  - overwrite ： 上書き
  - readback ： 最後の要素を再読み出し
  - block ： ブロック
  - do_nothing ： なにもしない

※Buffer Policy = Block+timeout時間の指定で，一定時間後読み出し/書き込み不可能な場合にタイムアウトを発生させる処理となる

接続プロファイル(ServicePort)について

項目	設定内容
Name	接続の名称
インターフェース情報	接続するインターフェースを設定. 接続対象のServicePortに複数のServiceInterfaceが定義されていた場合，どのインターフェースを実際に接続するかを指定

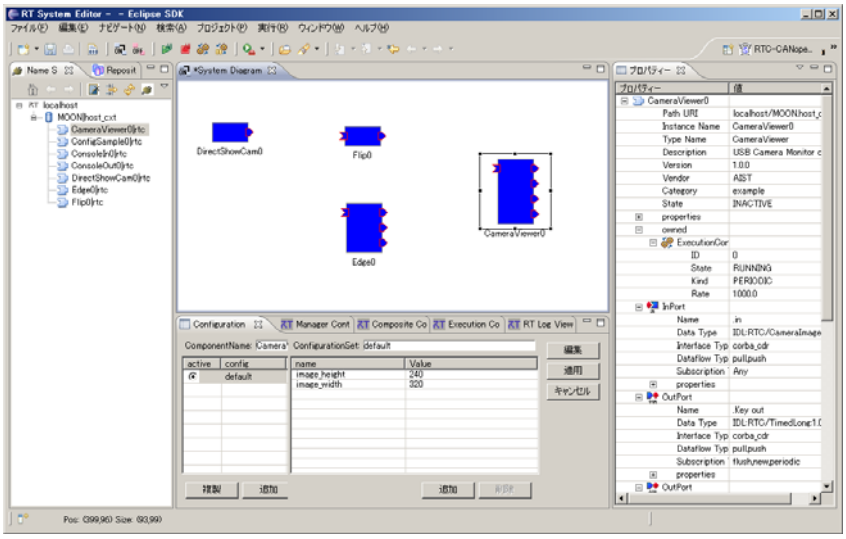


準備

- CameraViewerCompの起動
  - [スタート]メニューから起動
    [プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
    →[opencv-rtcs]→ [CameraViewerComp.exe]
- DirectShowCamCompの起動
  - [スタート]メニューから起動
    [プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
    →[opencv-rtcs]→ [DirectShowCamComp.exe]
- 画像処理用コンポーネントの起動
  - [スタート]メニューから起動
    [プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
    →[opencv-rtcs]→ [FlipComp.exe]
    [プログラム]→[OpenRTM-aist 1.1]→[C++]→[components]
    →[opencv-rtcs]→ [EdgeComp.exe]

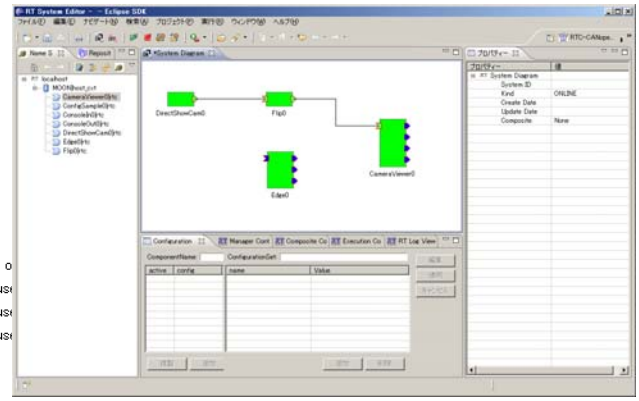
システムの構築

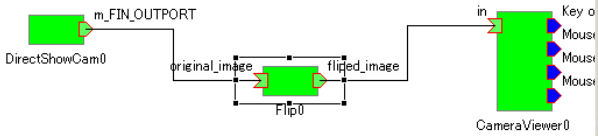
- 以下のコンポーネントをエディタ上に配置
  - DirectShowCam
  - Flip
  - Edge
  - CameraViewer

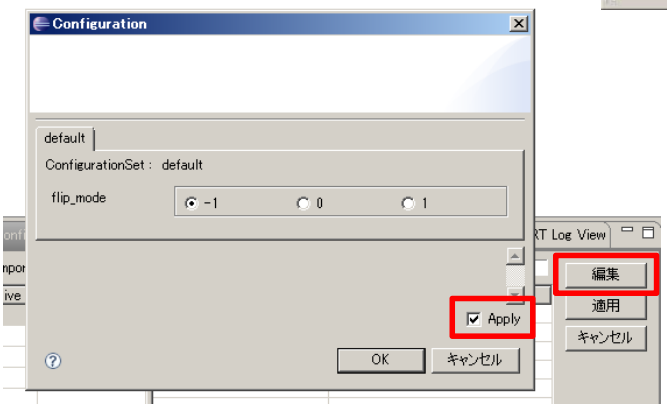


システム構成の変更

- Flip側との接続
  - DirectShowCam → Flip
    → CameraViewerと接続
    （接続プロファイルはデフォルト設定）
  - AllActivateを実行



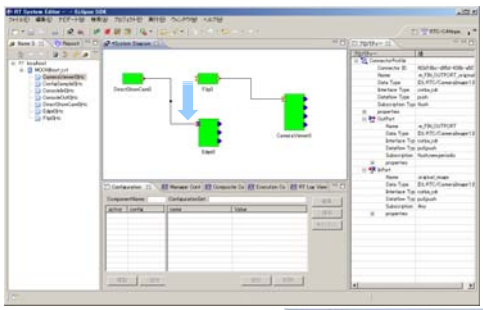


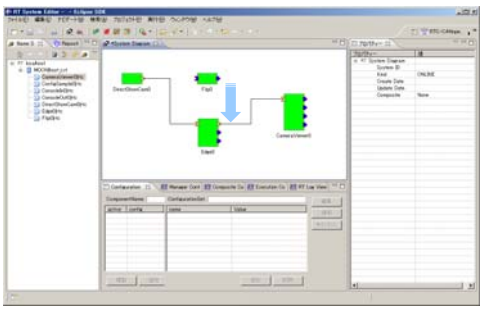


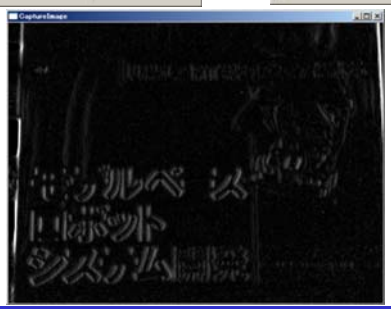
- ConfigurationViewの「編集」
- 表示されたダイアログ内で「flip_mode」の値を変更
- 「Apply」のチェックボックス

システム構成の変更

- Edge側への差し替え
  - Flipに繋がっている接続線を選択
  - Flip側のPort部分に表示されているハンドルをEdge側のPortに繋ぎ替え
    - 接続プロファイルはデフォルト設定のまま







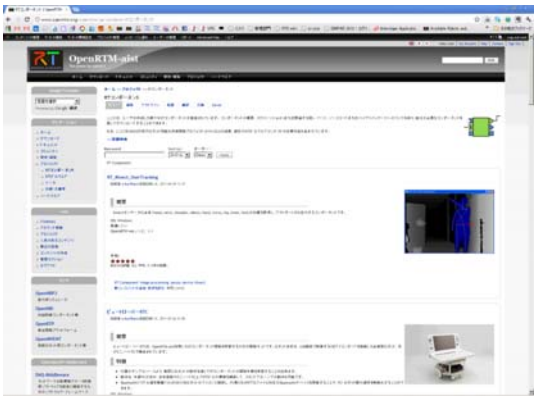
既存コンポーネントの再利用

既存コンポーネントの再利用

- プロジェクトとは
  - ユーザが作成した様々なコンポーネントやツールの公開場所
  - ユーザ登録すれば、誰でも自分の成果物の紹介ページを作成可能
  - 他のユーザに自分のコンポーネント等を紹介することができる

- プロジェクトのカテゴリ
  - RTコンポーネント: 1つのコンポーネントまたは複数のコンポーネント群などが登録されています。
  - RTミドルウエア：OpenRTM-aistや他のミドルウエア、ミドルウエア拡張モジュール等が登録されています。
  - ツール：各種ツール(RTSystemEditorやrtshellを含む)ツールはこのカテゴリになります。
  - 関連ドキュメント：関連ドキュメントとは、各種インターフェースの仕様書やマニュアル等を含みます。

プロジェクトページ



タイプ	登録数
RTコンポーネント群	287
RTミドルウエア	14
ツール	19
仕様・文書	4
ハードウエア	28

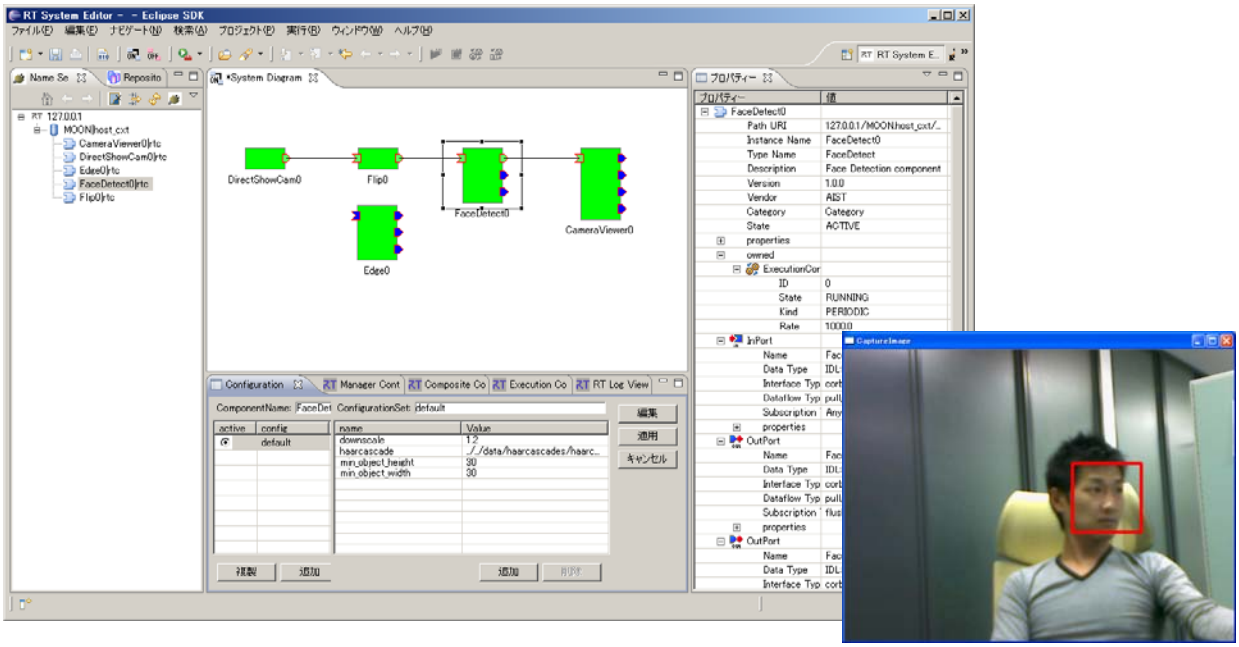
既存コンポーネントの再利用

- プロジェクトから対象コンポーネントを取得
  - 「顔検出コンポーネント」
    http://www.openrtm.org/openrtm/ja/project/facedetect
    対象コンポーネントをダウンロード

既存コンポーネントの再利用

- ダウンロードしたファイル(FaceDetect.zip)を解凍
- 解凍したディレクトリ内の以下のファイルを実行し，システムエディタ上に配置

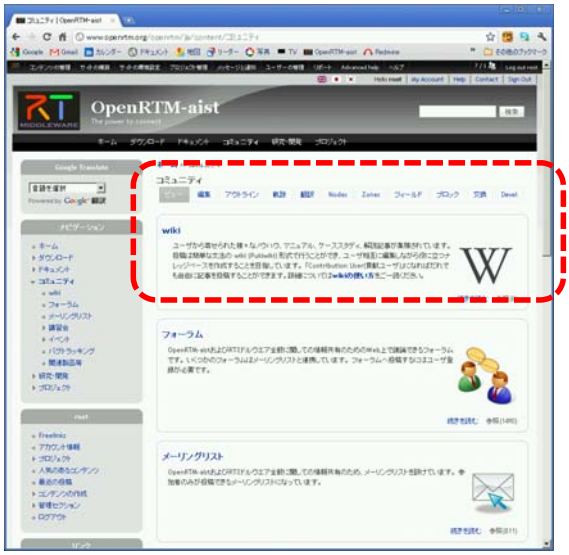
  $(FaceDetect_Root)/build/Release/FaceDetectComp.exe



その他

情報源

- Web
  - http://openrtm.org
- RTMコンテスト
  - http://www.openrtm.org/rt/RTMcontest
- Facebook
  - http://www.facebook.com/openrtm
- Wikiページ
  - ユーザによるページ作成が行える
  - 　　システム
  - Wiki（Pukiwiki）文法で簡単に作成可能
  - 4つのカテゴリ
    - ノウハウ
    - ケーススタディー
    - マニュアル
    - 解説



関連書籍

- はじめてのコンポーネント指向ロボットアプリケーション開発 ~RTミドルウェア超入門~
- 長瀬 雅之、中本 啓之、池添 明宏 著

- UMLとRTミドルウェアによるモデルベースロボットシステム開発
- 水川 真, 大原 賢一, 坂本 武志 著

- 第3章：ソフトウエア技術
  - 3.1 概論（安藤慶昭）
  - 3.2 並列処理（山崎信行）
  - 3.3 実時間処理（加賀美聡）
  - 3.4 プログラミング言語(松井俊浩)
  - 3.5 分散処理技術(成田雅彦)
  - 3.6 ロボット用ミドルウェア（安藤慶昭）
  - 3.7 ロボット開発プラットフォーム(金広文男)
  - 3.8 標準化(水川真)

講習会情報

- RTミドルウェアサマーキャンプ2012
  - 日時：2012年7月30日(月)～8月3日(金)
  - 場所：産業技術総合研究所 つくばセンター中央第二 ネットワーク会議室
  - 参加費：無料(ただし，宿泊費や食事代は参加者負担．産総研の宿泊施設を安価で提供できる予定です)
  - 学部4年生，大学院生や企業の若手研究者などに対して，実習形式の講習会を集中的に行い，RTミドルウェアを用いたロボット開発の機会を提供する。
  - http://openrtm.org/openrtm/ja/node/5048

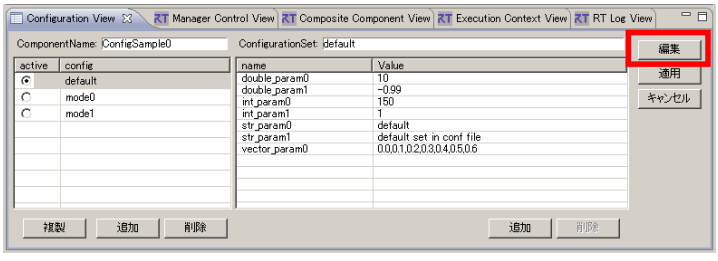
RTミドルウェアコンテスト2012

- RTミドルウェアを利用した技術・コンポーネントに関するコンテスト
  - 日時：2012年12月18日(火)(予定)
  - 場所：福岡国際会議場
    - 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(SI2012)の併催行事として開催予定
  - 表彰(2011年度)
    - 最優秀賞(副賞10万円)
    - 団体協賛(副賞2万円)×11件
    - 個人協賛(副賞1万円)×7件
  - 応募点数(2011年度)：14件

RTSystemEditor補足説明

コンフィギュレーションビュー

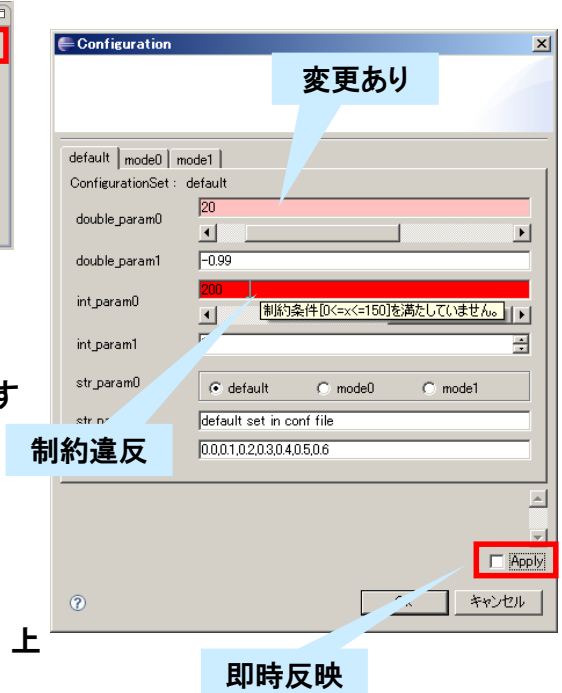
- RTコンポーネントのコンフィギュレーション情報の確認/編集



※「編集」ボタンにより，各種コントロールを用いた一括編集が可能

※「Apply」チェックボックスがONの場合，設定値を変更すると即座にコンポーネントに反映
→テキストボックスからフォーカス外れる，
ラジオボタンを選択する，
スライドバーを操作する，
スピナを変更する，などのタイミング

※コンフィギュレーション情報を複数保持している場合，上部のタブで編集対象を切り替え



コンフィギュレーション情報の設定方法

- rtc.conf内

  [カテゴリ名].[コンポーネント名].config_file: [コンフィギュレーションファイル名]

  ※例) example.ConfigSample.config_file: configsample.conf

- コンフィギュレーションファイル内
  - コンフィギュレーション情報

    conf.[コンフィグセット名].[コンフィグパラメータ名]:[デフォルト値]

    ※例) conf.mode0.int_param0: 123
  - Widget情報

    conf.__widget__.[コンフィグパラメータ名]:[Widget名]

    ※例) conf.__widget__.str_param0: radio
  - 制約情報

    conf.__constraints__.[コンフィグパラメータ名]:[制約情報]

    ※例) conf.__constraints__.str_param0: (bar,foo,foo,dara)

    conf.__[コンフィグセット名].[コンフィグパラメータ名]:[制約情報]

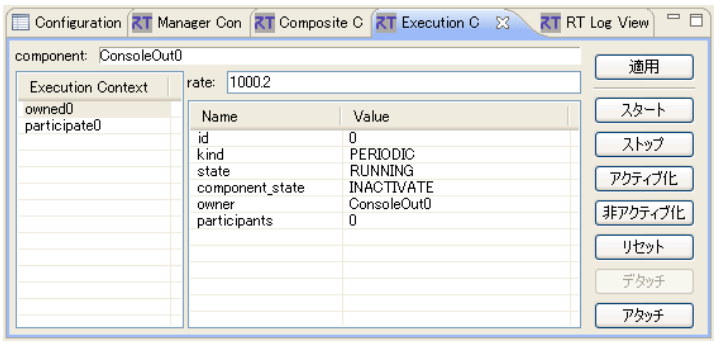
    ※例) conf.__mode1.str_param0: (bar2,foo2,dara2)

RTCの利用者が設定するのではなく，RTC開発者，RTC管理者が設定することを想定．

RTCBuilderを使用することで設定可能

実行コンテキストビュー

- RTコンポーネントが属する実行コンテキスト(EC)を一覧表示

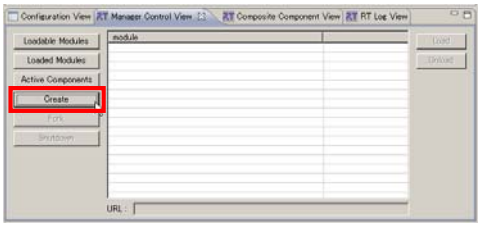


属性名	説明
id	ECのID．オンラインの場合には，context_handleを表示
kind	ECの種別(PERIODIC/EVENT_DRIVEN/OTHER)
state	ECの状態(RUNNING/STOPPING)
component state	対象RTCの状態(ACTIVE/INACTIVE/ERROR)
owner	対象ECを所有しているオーナーRTCのインスタンス名
participants	対象ECに参加中のRTCの数

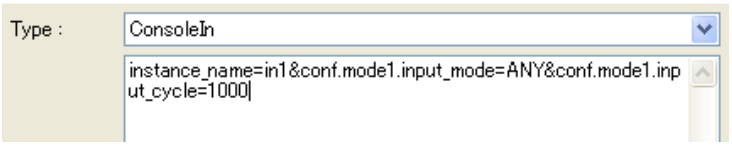
※対象ECの実行周期の変更，EC自身の動作開始/終了，新規RTCへのアタッチ，アタッチ済みRTCのデタッチも可能

マネージャビュー

RTコンポーネントの新規インスタンスの生成

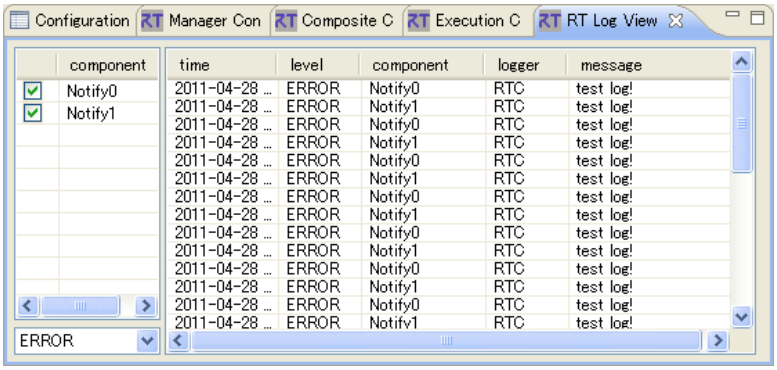


- コンフィギュレーション指定パラメータ
  - conf. [ConfigSet名]. [Configパラメータ名]=[設定値]の形式にてConfigurationSetの値も設定可能



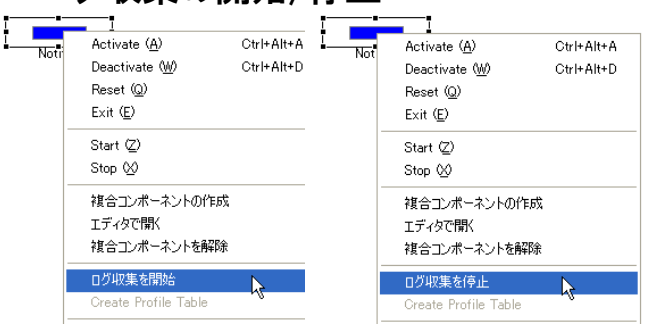
ログビュー

選択したRTCから収集したログ情報を一覧表示

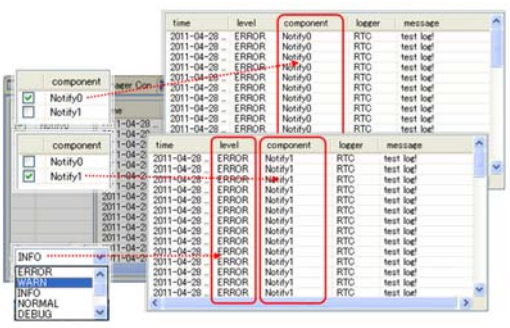


※近日機能追加予定

- ログ収集の開始/停止



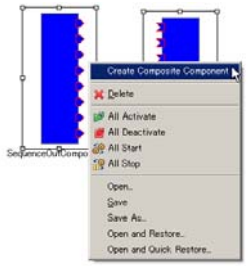
- ログ情報のフィルタリング



複合コンポーネント

- 複数のRTCをまとめて，1つのRTCとして扱うための仕組み
- 複合コンポーネントの作成方法

①複数RTCを選択している状態で右クリック

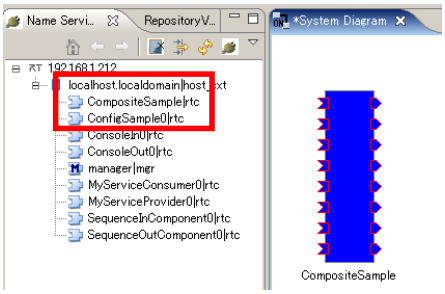


②複合コンポーネントのプロパティを設定

項目	設定内容
Manager	複合コンポーネントを制御するマネージャを選択
Name	複合コンポーネントのインスタンス名を入力
Type	複合コンポーネントの型を選択
Path	複合コンポーネントのパスを入力
Port	外部に公開するポートを選択

※生成対象複合コンポーネント外部と接続されているPortは強制的に公開されます

③複合コンポーネントを生成

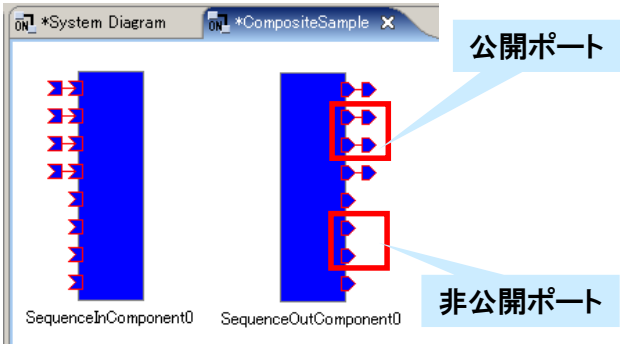


複合コンポーネント

- 複合コンポーネントのタイプについて

タイプ名	説明
PeriodicECShared	実行主体であるExecutionContextのみを共有． 各子コンポーネントはそれぞれの状態を持つ
PeriodicStateShared	実行主体であるExecutionContextと状態を共有
Grouping	便宜的にツール上のみでグループ化

- 複合コンポーネントエディタ
  - 複合コンポーネントをダブルクリックすることで表示



※エディタ内に別RTCをDnDすることで，子コンポーネントの追加が可能
→追加したRTCのポートは全て非公開に設定

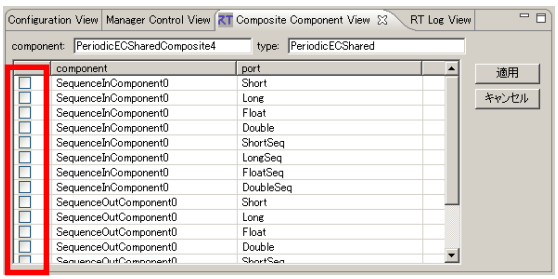
※エディタ内のRTCを削除することで，子コンポーネントの削除が可能
→削除されたRTCは，親エディタに表示

複合コンポーネント

■ 公開ポートの設定

- 複合コンポーネントビュー

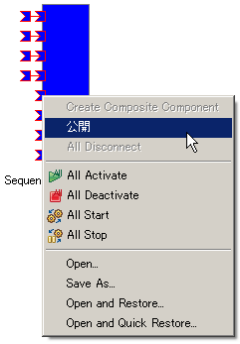
ポート公開情報



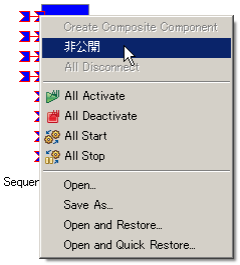
※ポート公開情報を変更し，「適用」をクリック

- 複合コンポーネントエディタ

※非公開ポートを「公開」



※公開ポートを「非公開」



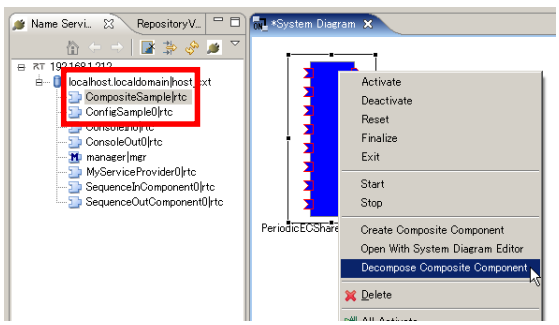
外部コンポーネントと接続されているポートを「非公開」に設定することはできません

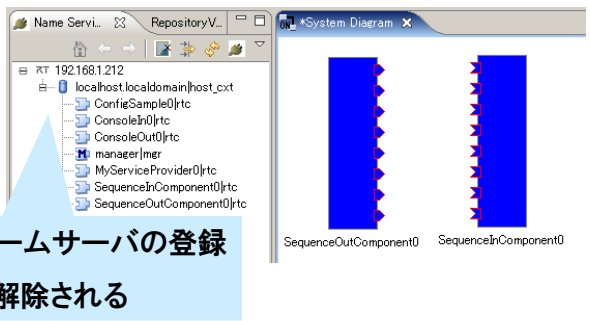
複合コンポーネント

■ 複合コンポーネントの解除

①複合RTCを右クリックし，複合コンポーネントの解除を選択

②複合コンポーネントが分解され，内部のRTCが表示



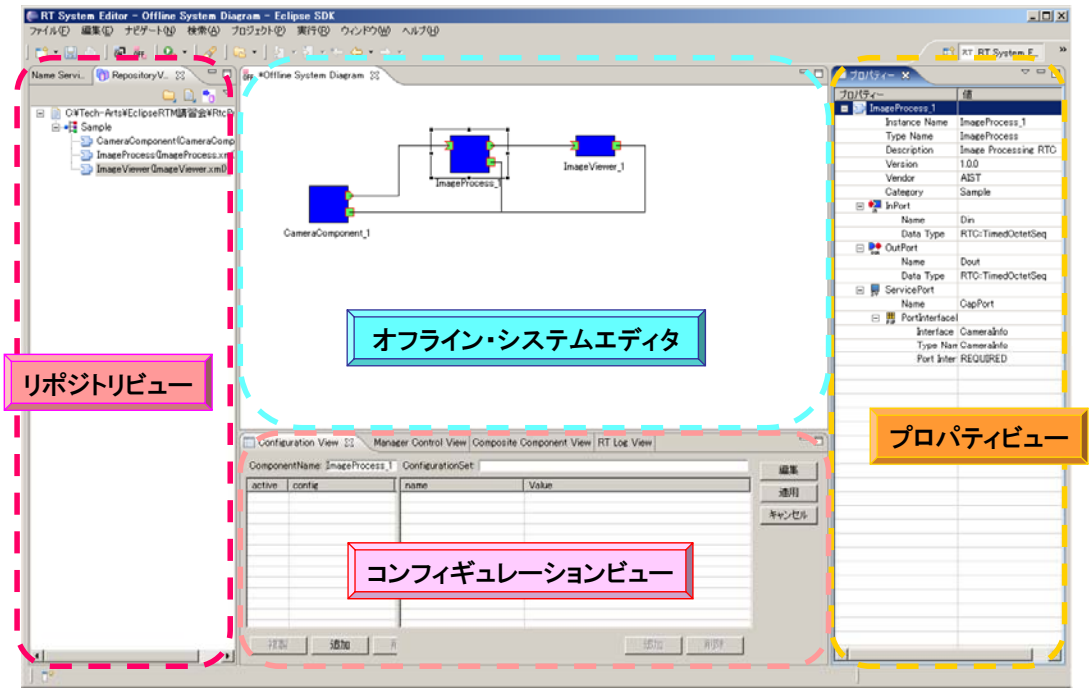


ネームサーバの登録も解除される

※エディタ上で，(Deleteキーなどで)単純に削除した場合は，エディタから表示が消えるのみ
複合コンポーネントは解除されない

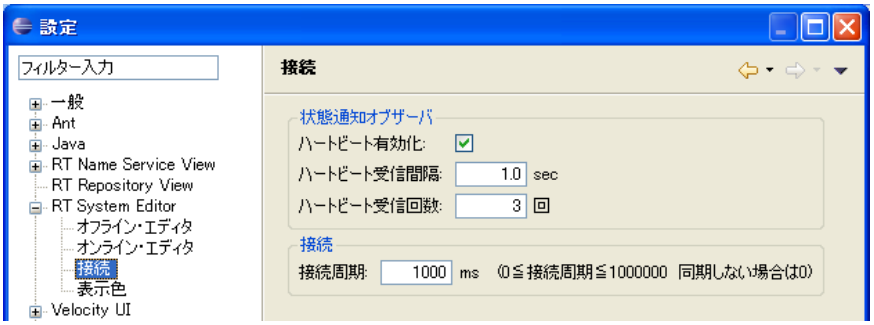
オフラインエディタ

- RTコンポーネントの仕様を用いてRTシステムを構築
  - 実際のRTコンポーネントが動作している必要はない



設定画面

- 接続ー状態通知オブザーバ
  - RTCの生存確認用オブザーバに関する設定
    - RTSE側から生存確認を行うのではなく，RTC側から通知(ハートビート)を行う形
    - OpenRTM-aist-1.1以降で対応



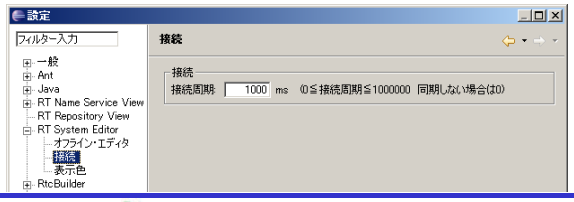
- ハートビート有効化：ハートビートによる生存確認機能の有効化
- ハートビート受信間隔：ハートビートの受信間隔．この間隔以内にRTC側からハートビートが送られてこないと生存確認失敗と判断
- ハートビート受信回数：この回数を超えて生存確認に失敗した場合，対象RTCに異常が発生したと判断

設定画面

- 「RT Name Service View」ー「接続」【接続周期】
  - ネームサービスビューが，ネームサーバに情報を問い合わせる周期

- 「RT Name Service View」ー「同期」【タイムアウト待ち時間】
  - ネームサービスビューが，リモートオブジェクトのレスポンスを待つ時間

- 「RT System Editor」ー「接続」【接続周期】
  - システムエディタが，ネームサーバに情報を問い合わせる周期



【接続周期】をゼロに設定すると
ネームサーバとの同期を行わない

マイクロマシン/MEMS展
ROBOTECH 次世代ロボット製造技術展

RTミドルウェア講習会

RTミドルウェアインストールワークショップ

日時：2012年7月13日(金) 10:30～12:30
場所：東京ビッグサイト 東ホール 特設会場