

第1部 RTミドルウェア OpenRTM-aist概要



RT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....)

産総研版RTミドルウェア

OpenRTM-aist

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

RTミドルウェアとは？



Joystick



Joystick
software



Robot Arm
Control software



Robot Arm

互換性のあるインターフェース同士は接続可能

RTミドルウェアとは？



Joystick



Joystick
software



Humanoid's Arm
Control software



Humanoid's Arm



Robot Arm

ロボットによって、インターフェースは色々
互換性が無ければつながらない

Robot Arm
Control software

RTミドルウェア

RTミドルウェアは別々に作られたソフトウェアモジュール同士を繋ぐための共通インターフェースを提供する



Joystick



Joystick software

Arm A Control software



Humanoid's Arm

compatible arm interfaces



Arm B Control software



Robot Arm

ソフトウェアの再利用性の向上
RTシステム構築が容易になる

重要なキーワードについて

- ミドルウェア
 - OSとアプリケーション層の中間に位置し、特定の用途に対して利便性、抽象化向上のために種々の機能を提供するソフトウェア
 - 例: RDBMS, ORB等. 定義は曖昧.
- 分散オブジェクト(ミドルウェア)
 - 分散環境におけるリモートのオブジェクトに対して透過的アクセスを提供する仕組み
 - 例: CORBA, Java RMI, DCOM等
- コンポーネント
 - 再利用可能なソフトウェアの断片(例えばモジュール)であり、内部の詳細機能にアクセスするための(シンタクス・セマンティクスともにきちんと定義された)インターフェースセットをもち、外部に対してはそのインターフェースを介してある種の機能を提供するモジュール.
- CBSD(Component Based Software Development)
 - ソフトウェア・システムを構築する際の基本構成要素をコンポーネントとして構成するソフトウェア開発手法

モジュール化のメリット

- 再利用性の向上
 - 同じコンポーネントをいろいろなシステムに使いまわせる
- 選択肢の多様化
 - 同じ機能を持つ複数のモジュールを試すことができる
- 柔軟性の向上
 - モジュール接続構成かえるだけで様々なシステムを構築できる
- 信頼性の向上
 - モジュール単位でテスト可能なため信頼性が向上する
- 堅牢性の向上
 - システムがモジュールで分割されているので、一つの問題が全体に波及しにくい

RTコンポーネント化のメリット

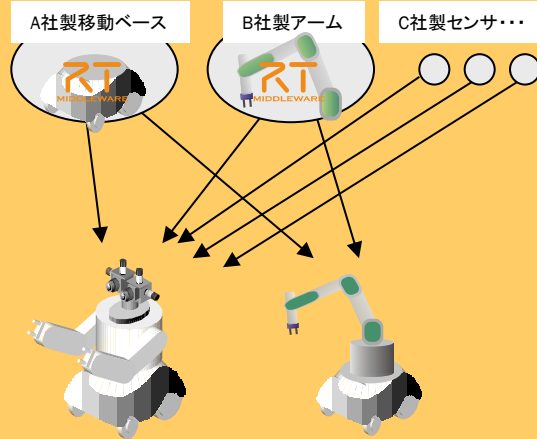
モジュール化のメリットに加えて

- ソフトウェアパターンを提供
 - ロボットに特有のソフトウェアパターンを提供することで、体系的なシステム構築が可能
- フレームワークの提供
 - フレームワークが提供されているので、コアのロジックに集中できる
- 分散ミドルウェア
 - ロボット体内LANやネットワークロボットなど、分散システムを容易に構築可能

RTミドルウェアの目指すところ

モジュール化による問題解決

コストの問題



モジュール化・再利用
ロボットの低コスト化

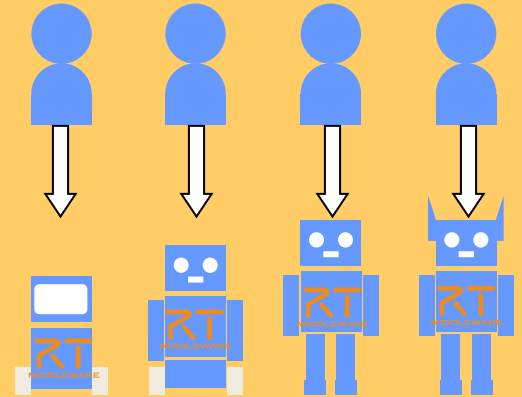
技術の問題



最新技術を利用可能

ニーズの問題

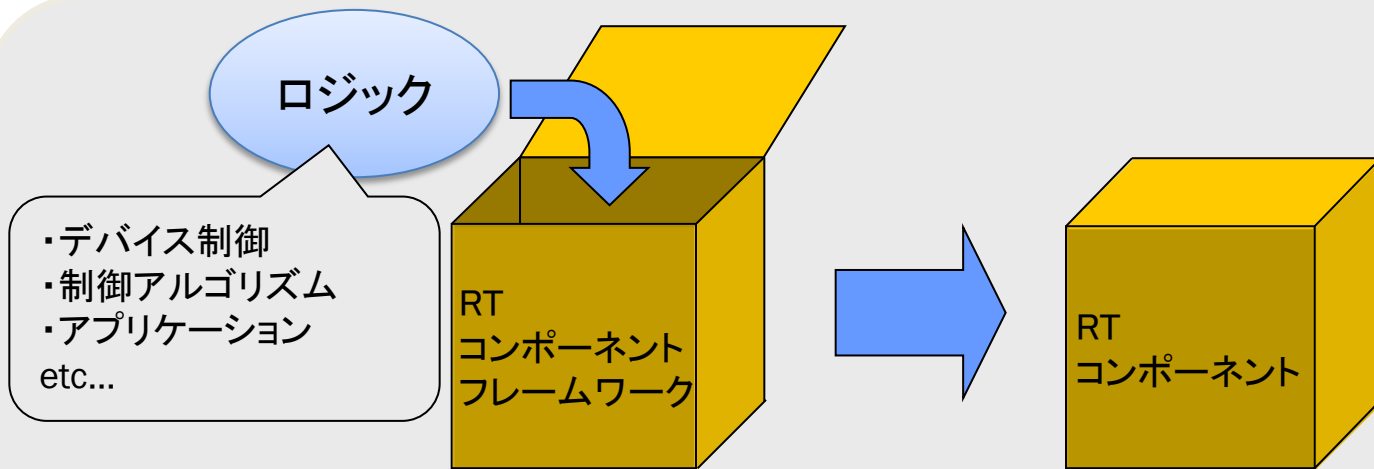
多様なユーザ



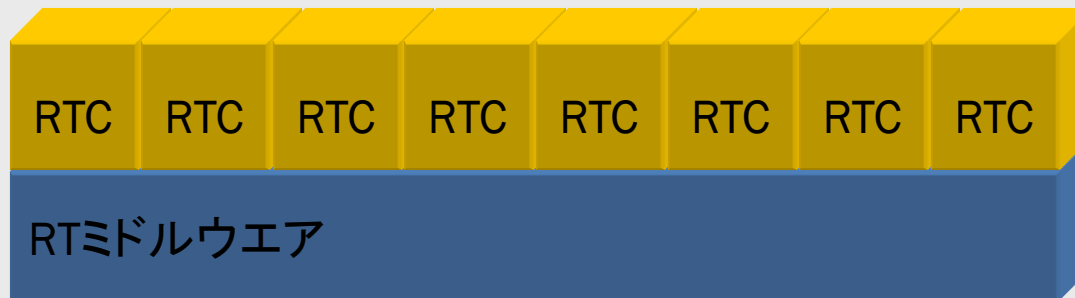
カスタマイズが容易に
多様なニーズに対応

ロボットシステムインテグレーションのイノベーション

RTミドルウェアとコンポーネント



ロジックを箱(フレームワーク)に入れたもの = RTコンポーネント (RTC)

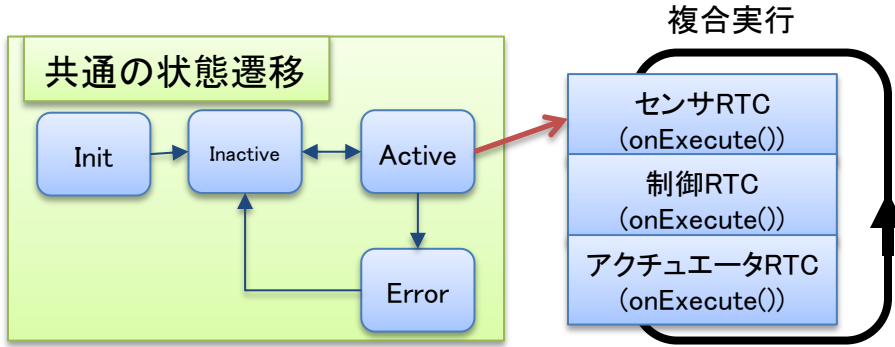


RTC ■ ● ☹️ # ③ (OSのようなもの) = RTミドルウェア (RTM)

※ RTCはネットワーク上に分散可能

RTコンポーネントの主な機能

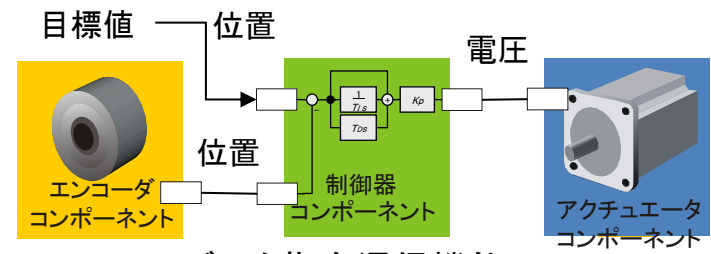
アクティビティ・実行コンテキスト



ライフサイクルの管理・コアロジックの実行

データポート

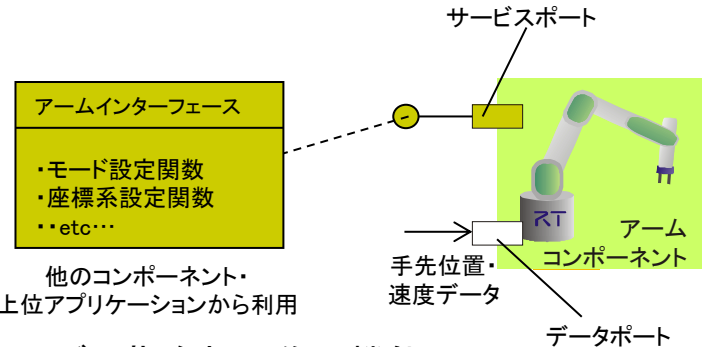
- ・ データ指向ポート
- ・ 連続的なデータの送受信
- ・ 同じデータ型のポート同士接続可能
- ・ 動的に接続・切断可能



データ指向通信機能

サービスポート

- ・ 任意に定義可能なインターフェースを持つポート
- ・ 内部の詳細な機能にアクセス



サービス指向相互作用機能

コンフィグレーション

- 内部パラメータを管理
- コンフィギュレーションセット
 - ・ セット名、名前: 値のリスト
 - ・ 複数のセットを保持
 - ・ セットを切替可能

複数のセットを動作時に切り替えて使用可能

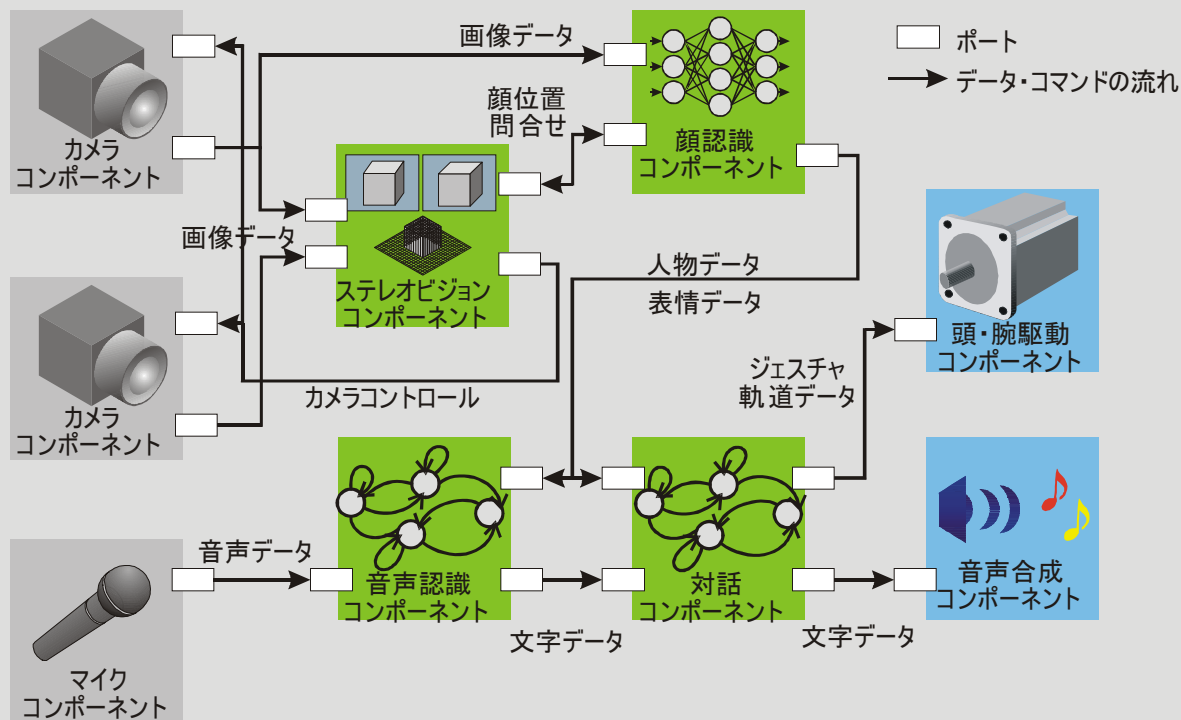
セット名	名前	値		
セット名	名前	値		



RTCの分割と連携



ロボット体内のコンポーネントによる構成例

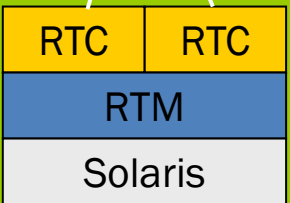
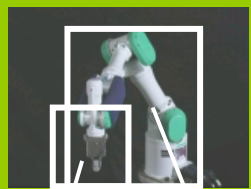


(モジュール)情報の隠蔽と公開のルールが重要

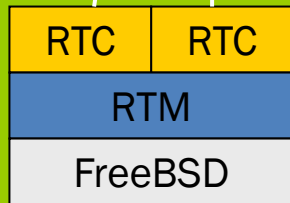
RTミドルウェアによるシステム構築例

RTMにより、ネットワーク上に分散するRTCをOS・言語の壁を越えて接続することができる。

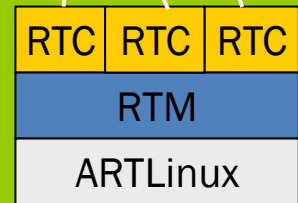
ロボットA



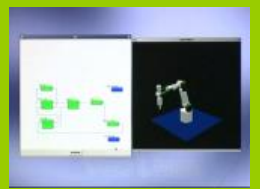
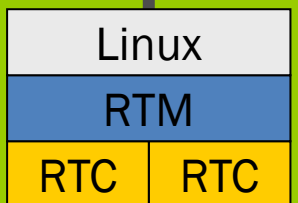
ロボットB



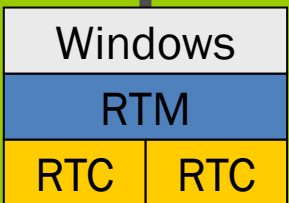
ロボットC



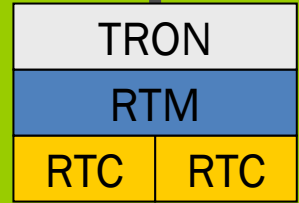
ネットワーク



アプリケーション



操作デバイス



センサ

RTC同士の接続は、プログラム実行中に動的に行うことができる。

OpenRTM-aistとは？

- コンポーネントフレームワーク+ ミドルウェアライブラリ
- 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: Linux, FreeBSD, Windows, Mac OS X, QNX
 - 非公式: uITRON, T-Kernel, VxWorks
- 言語:
 - C++ (1.1.1), Python (1.1.0), Java (1.0.0)
 - .NET (implemented by SEC)
- CPU アーキテクチャ(動作実績):
 - i386, ARM, PPC, SH4
 - PIC, dsPIC, SH2, 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

OMG RTCの標準化活動

- 2005年9月
RFP: Robot Technology Components (RTCs) 公開
- 2006年2月
Initial Response : PIM and PSM for RTComponent を執筆し提出.
提案者: AIST(日)、RTI(米)
- 2006年4月
両者の提案を統合した仕様を提案
- 2006年9月
ABIにて承認、事実上の国際標準獲得
FTFが組織され最終文書化開始
- 2007年8月
FTFの最後の投票が終了
- 2007年9月
ABIにてFTFの結果を報告
- 2008年4月
OMG RTC標準仕様公式リリース
- 2010年1月
OpenRTM-aist-1.0リリース

Date: November 2008

Robotic Technology Component Specification
Final Adopted Specification

OMG Adopted Specification
ptc/06-11-07



OMG RTCファミリ

名称	ベンダ	特徴	互換性
OpenRTM-aist	AIST	C++, Python, Java	---
OpenRTM.NET	SEC	.NET(C#,VB,C++/CLI, F#, etc..)	◎
RTM on Android	SEC	Android版RTミドルウェア	◎
H-RTM	本田R&D	OpenRTM-aist互換、FSM型コンポーネントをサポート	◎
RTC-Lite	AIST	PIC, dsPIC上の実装	○(ブリッジ)
miniRTC, microRTC	SEC	CAN・ZigBee等を利用した組込用RTC実装	○(ブリッジ)
RTMSafety	SEC/AIST	機能安全認証 (IEC61508) capableなRTM実装, 商用	○(ブリッジ)
RTC CANOpen	SIT, CiA	CANOpen-RTCマッピングを定めたCiA 標準	○(ブリッジ)
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ミドルウェアプロジェクト

- 名称: NEDO 21世紀ロボットチャレンジプログラム
 - 「ロボット機能発現のために必要な要素技術開発」
- 目的:
 - RT要素の部品化(モジュール化)の研究開発
 - 分散オブジェクト指向開発
 - RT要素の分類・モジュール化に必要な機能・インタフェース仕様の明確化
- 予算規模:
 - 65百万円
 - 全体267.3百万円

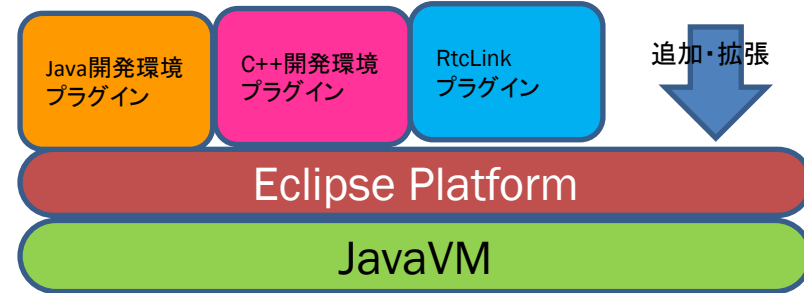


NEDO基盤PJ

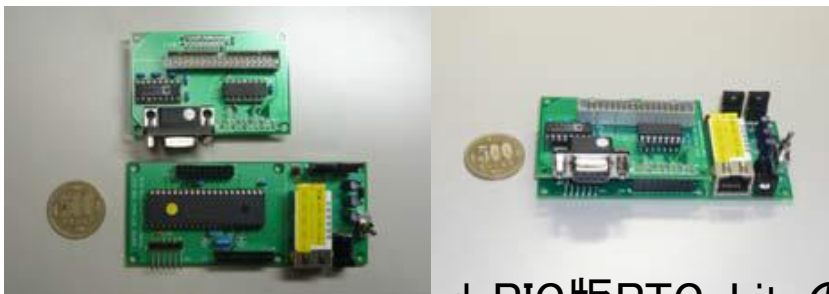
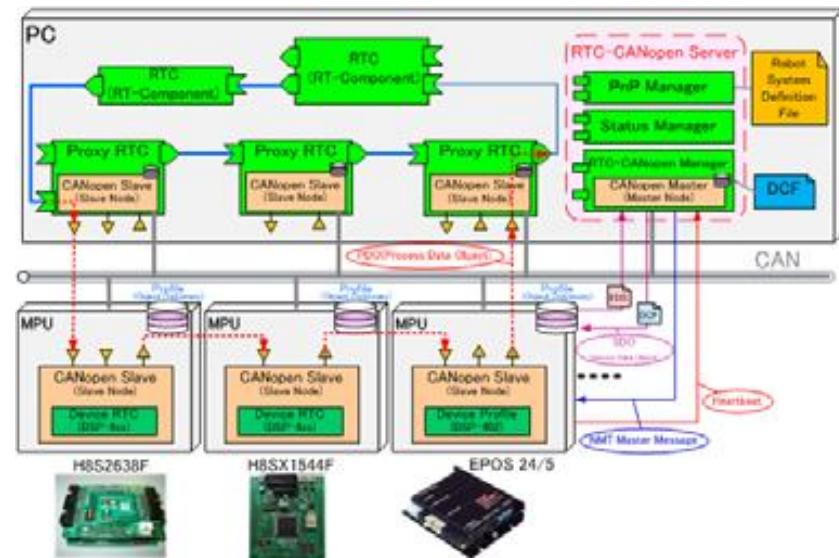
FY2003-FY2007

- 名称:「運動制御デバイスおよびモジュールの開発」
- 目的:
 - 運動制御デバイスの開発
 - デバイ스에搭載するRTCの開発
 - その他モーションコントロールに資するRTM/RTCの開発
- 予算規模:
 - 15百万円/年
 - 371百万円、全体1,259百万円

その他の
ロボット開発
ツール
プラグイン



ツールのEclipseプラグイン化



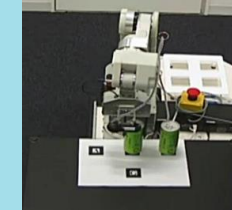
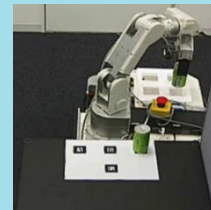
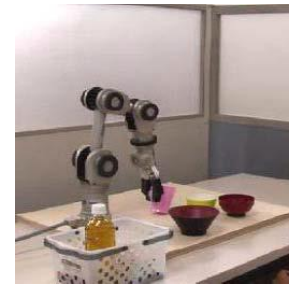
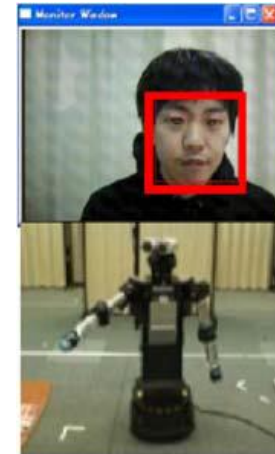
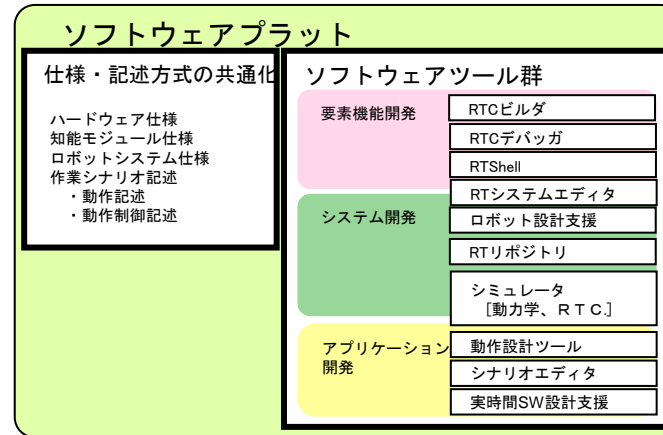
dsPIC版RTC-Liteの開発

RTC-CANの開発

知能化プロジェクト

FY2007-FY2012

- 名称:「次世代ロボット知能化技術開発プロジェクト」
- 目的
 - ソフトウェアプラットフォームの開発
 - 作業知能、移動知能、コミュニケーション知能に関するモジュールの開発
- 予算:
 - 400百万円
 - 全体7,000百万円
- 研究グループ
 - 15グループ



(提供: RTC再利用技術研究センター殿)

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	2009	2010	2011	2012	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
合計	10342	18258	18612	3768	393	51373

ユーザ数

タイプ	登録数
Webページユーザ	988
Webページアクセス	約300 visit/day 約1000 view/day
メーリングリスト	424 人
講習会	のべ約700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	321
RTミドルウェア	17
ツール	22
仕様・文書	5
ハードウェア	31

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..)
miniiRTC,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
H-RTM(仮称)	ホンダR&D	OpenRTM-aist互換、FSM型コンポーネントをサポート


RTコンポーネントの探し方

OpenRTM-aistの
ホームページで
利用可能なサービス

プロジェクトページ

ユーザが自分のRTC
などを登録可能

他のユーザの作った
RTCを検索可能



ハードウェア集

OpenRTMで利用可能なハードウェア集


ハードウェアに対応したRTCのリスト



知能化プロジェクトRTC集

NEDO知能化プロジェクトにおける成果物をまとめたページ

- ツール
- 作業知能モジュール
- 移動知能モジュール
- 対話知能モジュール
- 商用ライセンスモジュール



RTミドルウェア普及活動

RTミドルウェア講習会

- RTミドルウェア初心者を対象とした導入向けの講習会
- 年に数回開催
- 要望が合った場合、適宜開催



RTミドルウェアサマーキャンプ

- 8月上旬に開催(今年は8月1~5日)
- RTミドルウェアを用いたシステム開発を行う合宿
- グループに分かれ、目標とするシステムを開発
- 講師陣による密なサポート



RTミドルウェアコンテスト

2016年12月に
札幌で開催

- 12月に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の特別セッションとして開催
- RTミドルウェアを用いたシステム、開発支援のためのツールなどを対象としたコンテスト



ROSとRTMの比較

ROSの良い点

- UNIXユーザに受けるツール(特にCUI)が豊富
- 独自のパッケージ管理システムを持つ
- ノード(コンポーネント)の質、量ともに十分
 - WGが直接品質を管理しているノードが多数
- ユーザ数が多い
- メーリングリストなどの議論がオープンで活発
- 英語のドキュメントが豊富

ROSの問題点

- Ubuntu以外の対応が今一つ
 - Windows対応はいろいろな人が試したがいまだに公式には含まれない
- コアライブラリの仕様が固まっていない
 - 他の言語で実装する際の妨げ
 - サードパーティー実装が出にくい
 - 品質を保証しづらい
- コンポーネントモデルがない

ROSとRTMの比較

OpenRTMの良い点

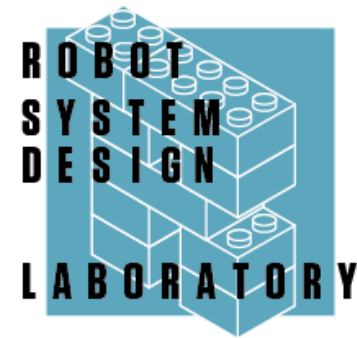
- 対応OS・言語の種類が多い
 - Windows, UNIXなど
 - GUIツールが豊富
- **仕様が標準化されている**
 - OMGに参加することで変更可
 - サードパーティー実装が容易
 - すでに10程度の実装あり
- **コンポーネントモデルが明確**
 - オブジェクト指向、UML・SysMLとの相性が良い
 - モデルベース開発
- **IEC61508機能安全認証取得**
 - RTMSafety

OpenRTMの問題点

- パッケージ管理システムがない
- CUIツールが少ない
 - UNIXユーザ受けしない
- ユーザ数が少ない
- 英語のドキュメントが少ない
- 開発速度が遅い
 - 現在は開発者一人
- コンポーネント数が少ない



再利用性を高めたコンポーネント創出のために、日本ロボット工業会のRTミドルウェアWGでロボット用の代表的な機能のインタフェースの標準化活動を2013年度から開始

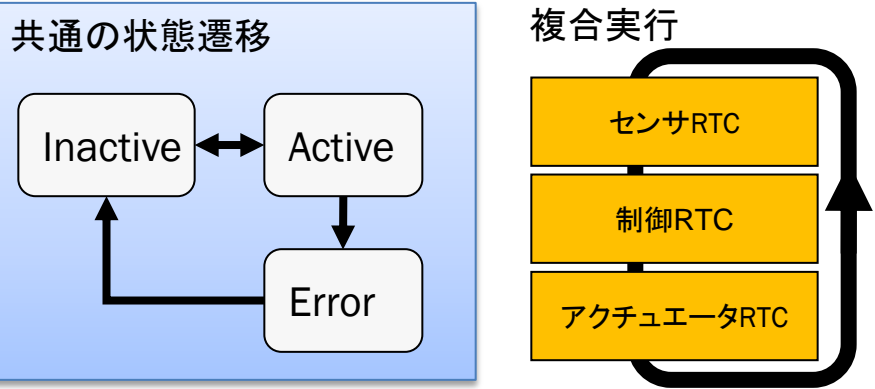


RTコンポーネントの基本機能



RTC開発で利用できる4つの要素

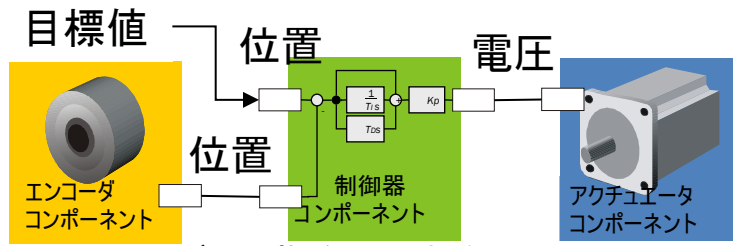
アクティビティ・実行コンテキスト



ライフサイクルの管理・コアロジックの実行

データポート

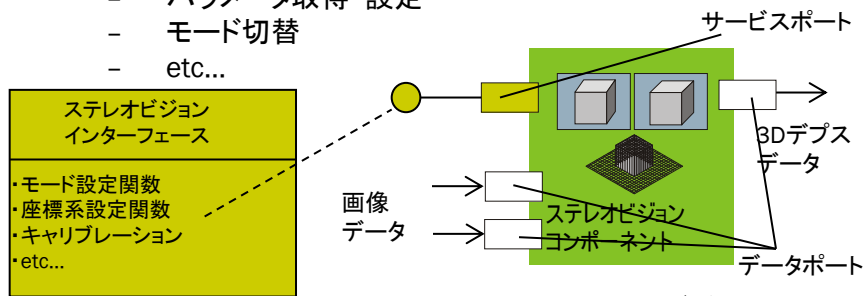
- データ指向ポート
- 連続的なデータの送受信
- 動的な接続・切断



データ指向通信機能

サービスポート

- 定義可能なインターフェースを持つ
- 内部の詳細な機能にアクセス
 - パラメータ取得・設定
 - モード切替
 - etc...



サービス指向相互作用機能 ステレオビジョンの例

コンフィギュレーション

- パラメータを保持する仕組み
- いくつかのセットを保持可能
- 実行時に動的に変更可能

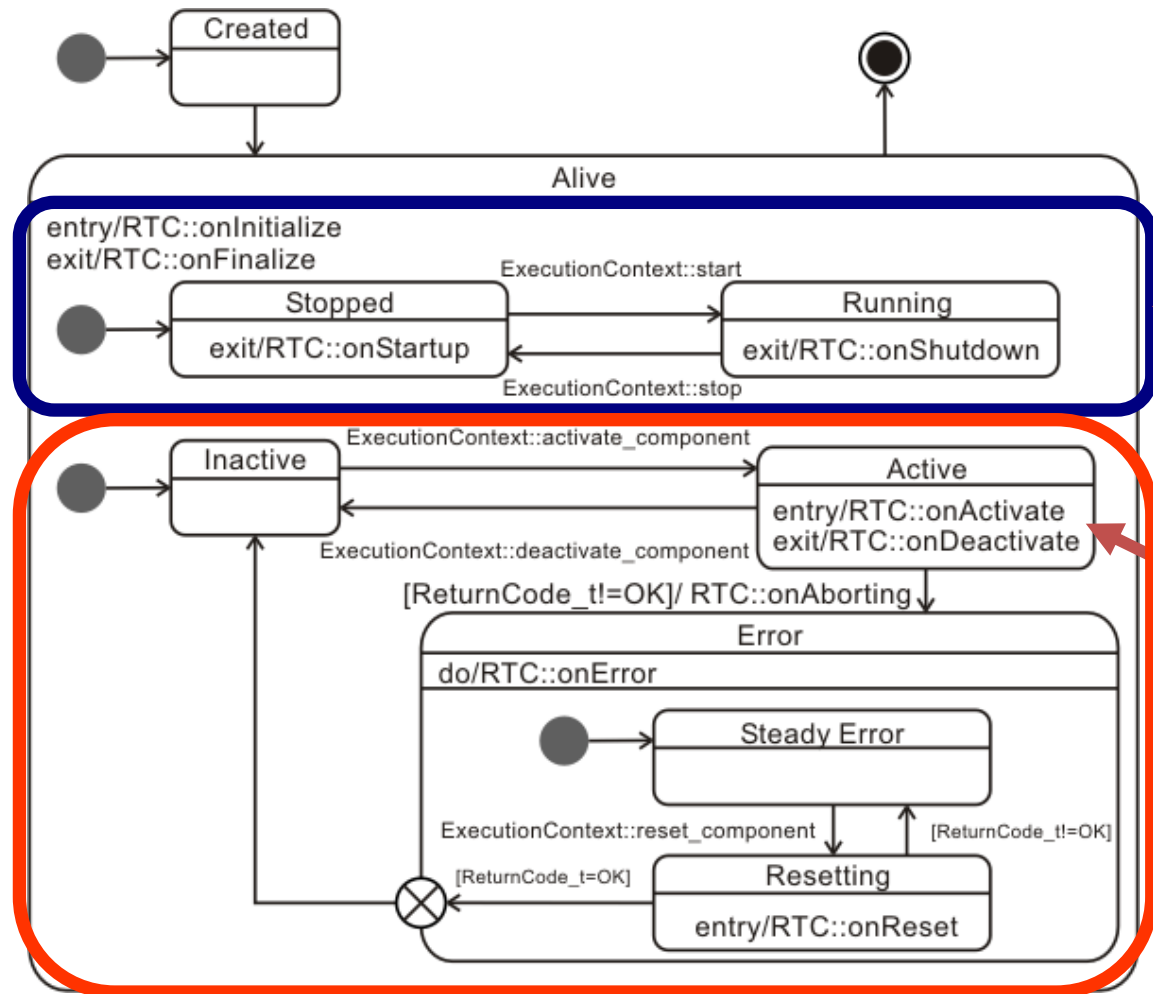
セット名	名前	値			
セット名	名前	値			

複数のセットを動作時に切り替えて使用可能

これら4つの要素を利用して、所望の機能を有するRTCの開発を行う。

アクティビティ

コンポーネント内の状態遷移をつかさどる機能
 コンポーネント開発時には、常にこの状態遷移を意識しながら開発をする必要がある。



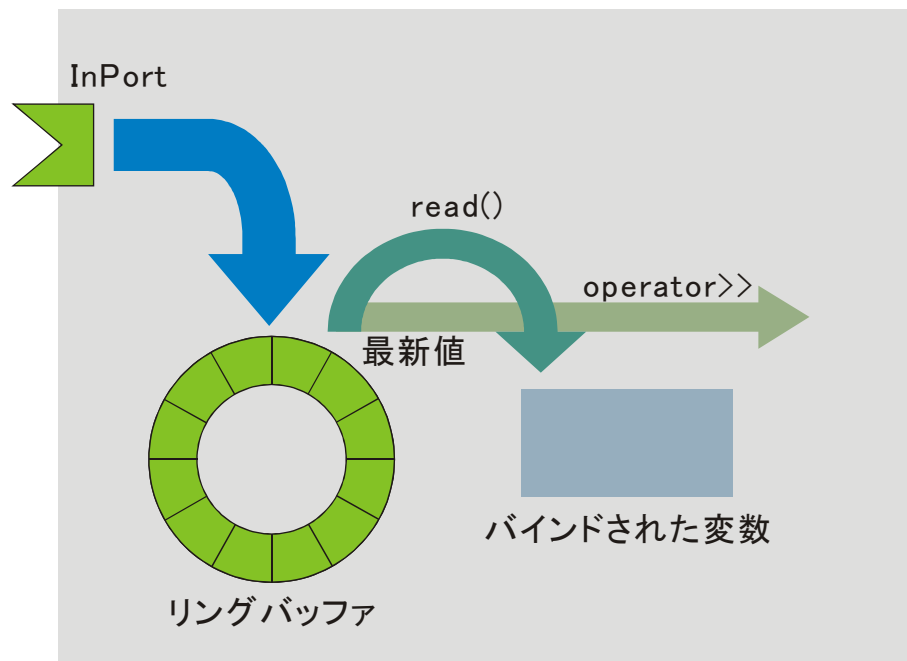
ユーザがあまり意識なくてよい部分

コンポーネント開発時に必要な部分

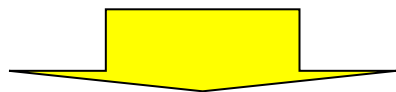
ActiveDo/RTC::onExecuteはここに入る
 (DataFlow型のコンポーネントのとき)

データポート(InPort)

- InPortのテンプレート第2引数: バッファ
 - ユーザ定義のバッファが利用可能
- InPortのメソッド
 - read(): InPort バッファからバインドされた変数へ最新値を読み込む
 - >> : ある変数へ最新値を読み込む

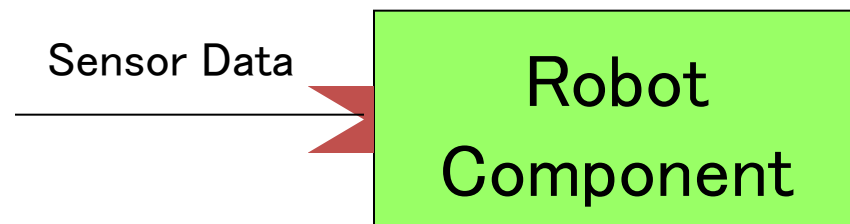


基本的にOutPortと対になる



データポートの型を
同じにする必要あり

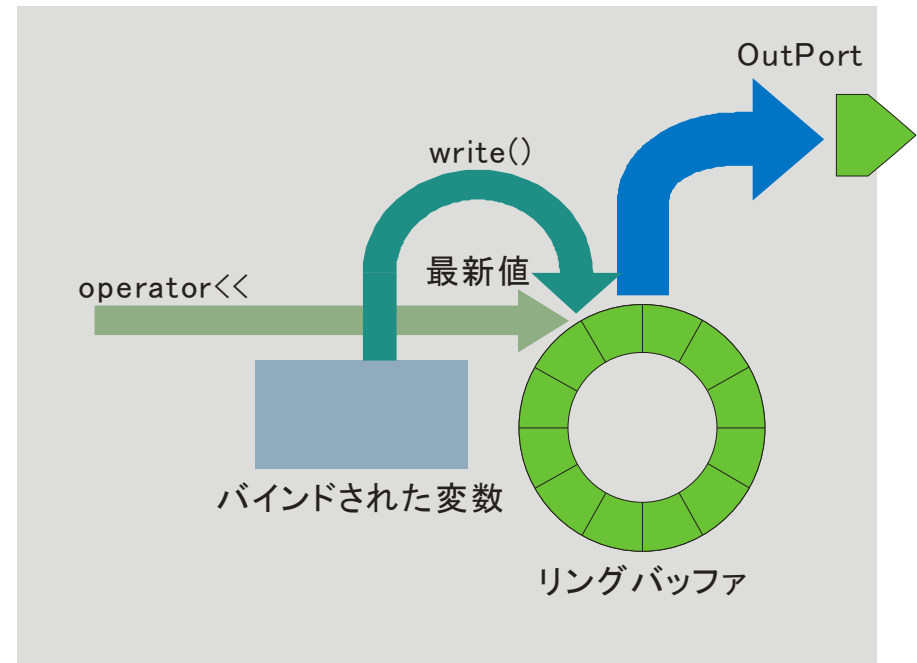
例



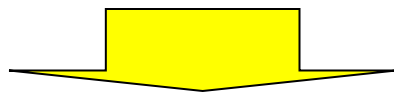
逐次ロボットの手先位置を変更しながら動作させる必要がある場合などに利用

データポート(OutPort)

- OutPortのテンプレート第2引数: バッファ
 - ユーザ定義のバッファが利用可能
- OutPortのメソッド
 - write(): OutPort バッファへバインドされた変数の最新値として書き込む
 - >> : ある変数の内容を最新値としてリングバッファに書き込む

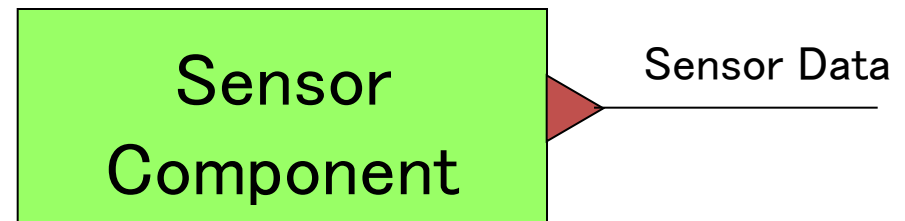


基本的にInPortと対になる



データポートの型を
同じにする必要あり

例



データポートの変数の型

データだけでなく、時間も含んだデータフォーマットを採用。
単一データからシーケンスデータまで利用可能

```
struct TimedShort
{
    Time tm;
    short data;
};
```

- 基本型

- tm: 時刻
- data: データそのもの

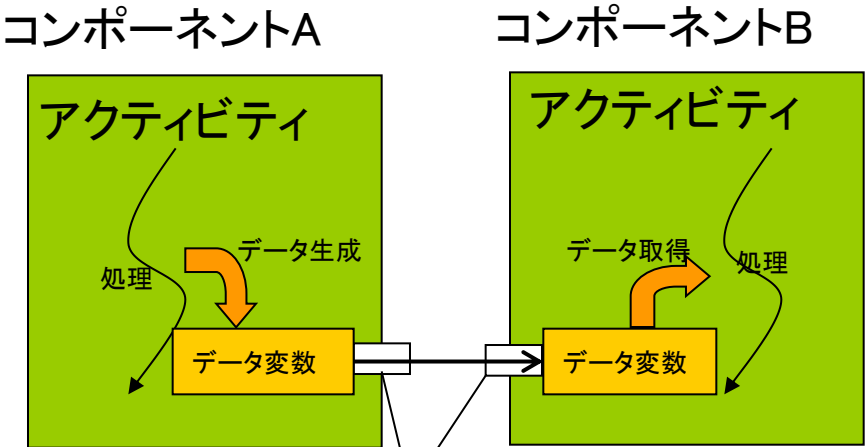
```
struct TimedShortSeq
{
    Time tm;
    sequence<short> data;
};
```

- シーケンス型

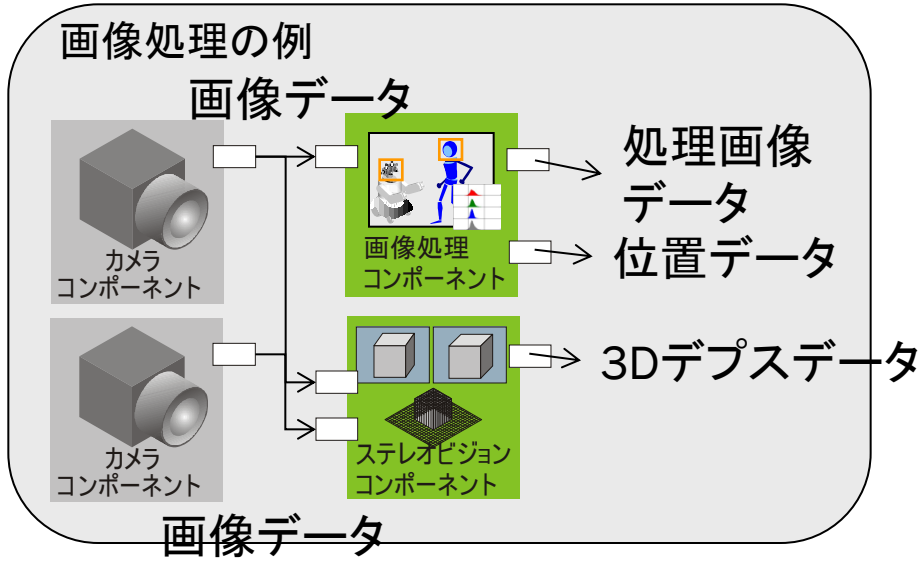
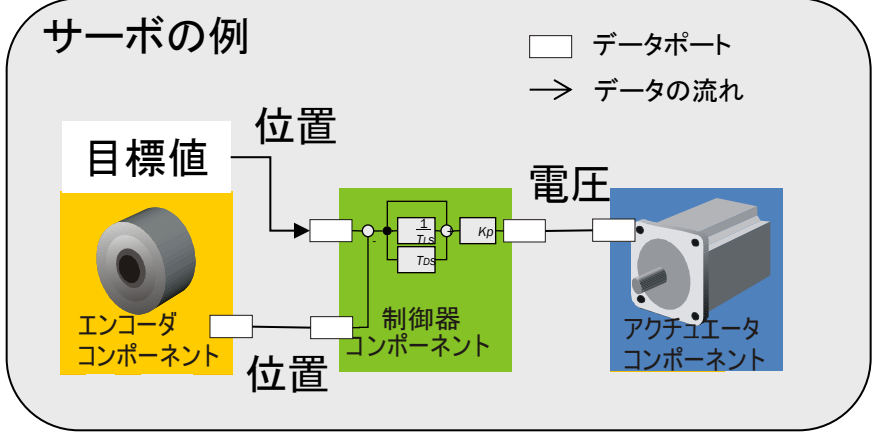
- data[i]: 添え字によるアクセス
- data.length(i): 長さiを確保
- data.length(): 長さを取得
- データを入れるときにはあらかじめ長さをセットしなければならない。
- CORBAのシーケンス型そのもの
- 今後変更される可能性あり

データポートの利用イメージ

- 主にロボットの下位レベル処理に利用
- 同じデータ型のポート同士接続可能
- 動的に接続・切断可能

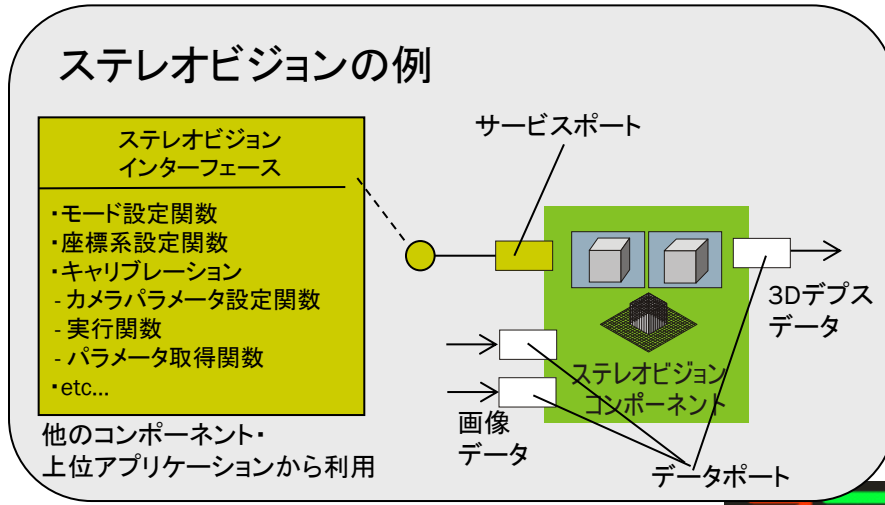
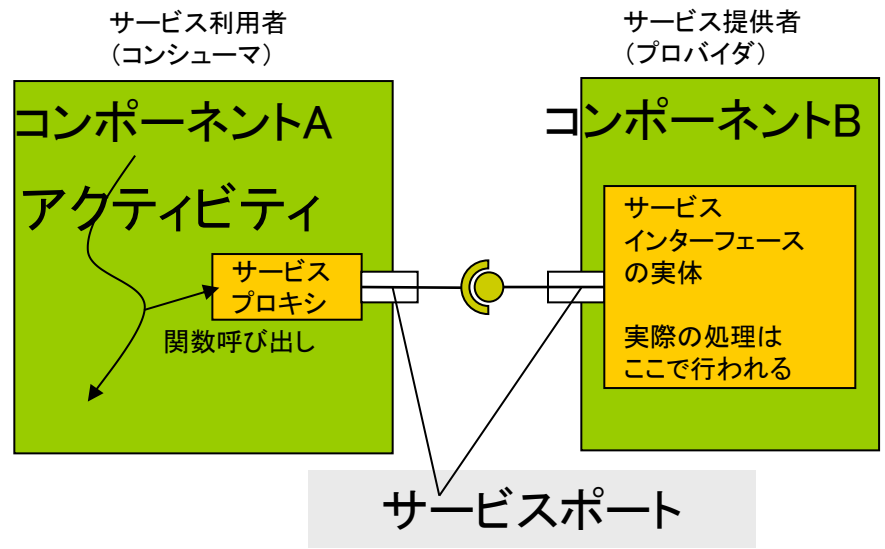
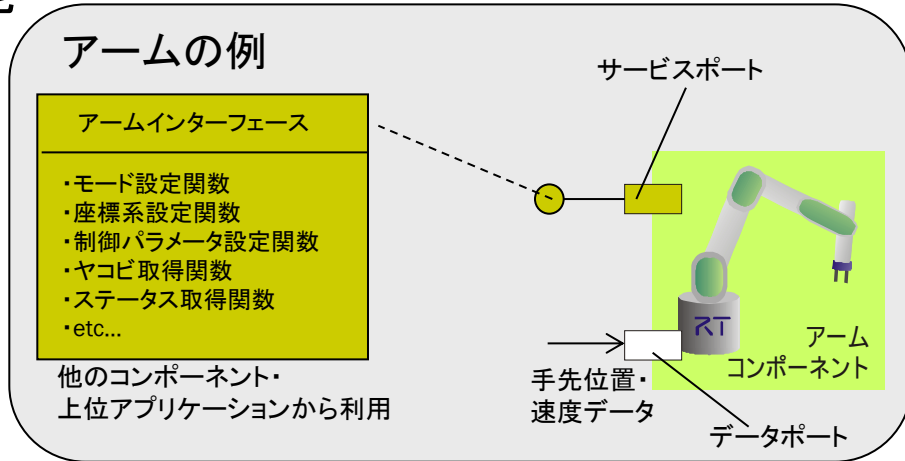


データポート
データが自動的に伝送される



サービスポート

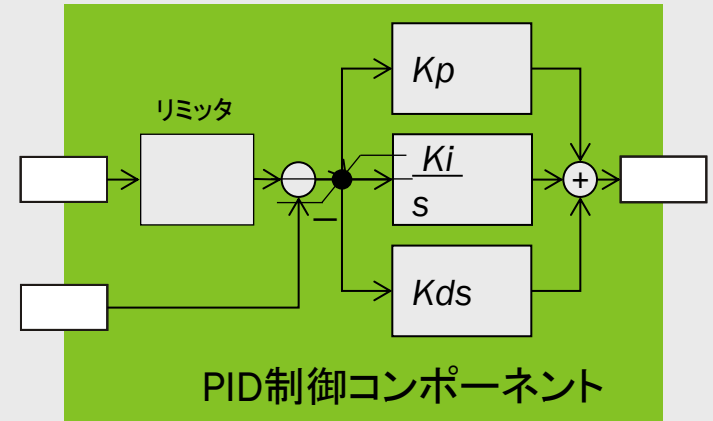
- 任意に定義可能なインターフェースを持つポート
 - コマンド・関数を自由に追加
 - 他のコンポーネントからアクセス可能
 - (本当は標準化したい)
- 内部の詳細な機能にアクセス
 - パラメータ取得・設定
 - モード切替
 - 処理の依頼と結果取得
 - etc...



コンフィギュレーション

- コンフィギュレーション
 - パラメータを管理
 - コンフィギュレーションセット
 - セット名、名前:値のリスト
 - 複数のセットを保持
 - セットを切替可能

PIDコントローラの例



modeA	名前	Kp	Ki	Kd	Inmax	Inmin
	値	0.6	0.01	0.4	5.0	-5.0

modeB	名前	Kp	Ki	Kd	Inmax	Inmin
	値	0.8	0.0	0.01	10.0	-10.0

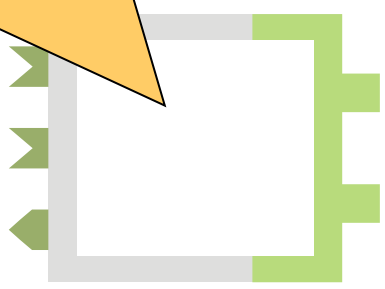
modeC	名前	Kp	Ki	Kd	Inmax	Inmin
	値	0.3	0.1	0.31	1.0	-1.0

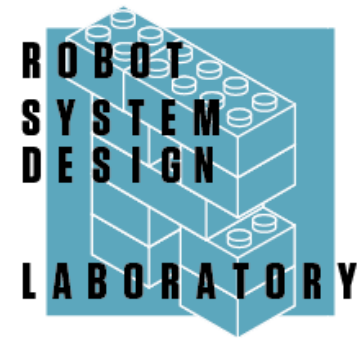
制御対象やモードに応じて複数のPIDゲインおよび入力リミッタ値を切り替えて使用することができる。
動作中の切り替えも可能。

複数のセットを動作時に切り替えて使用可能

セット名	名前	値				
	値					

セット名	名前	値				
	値					

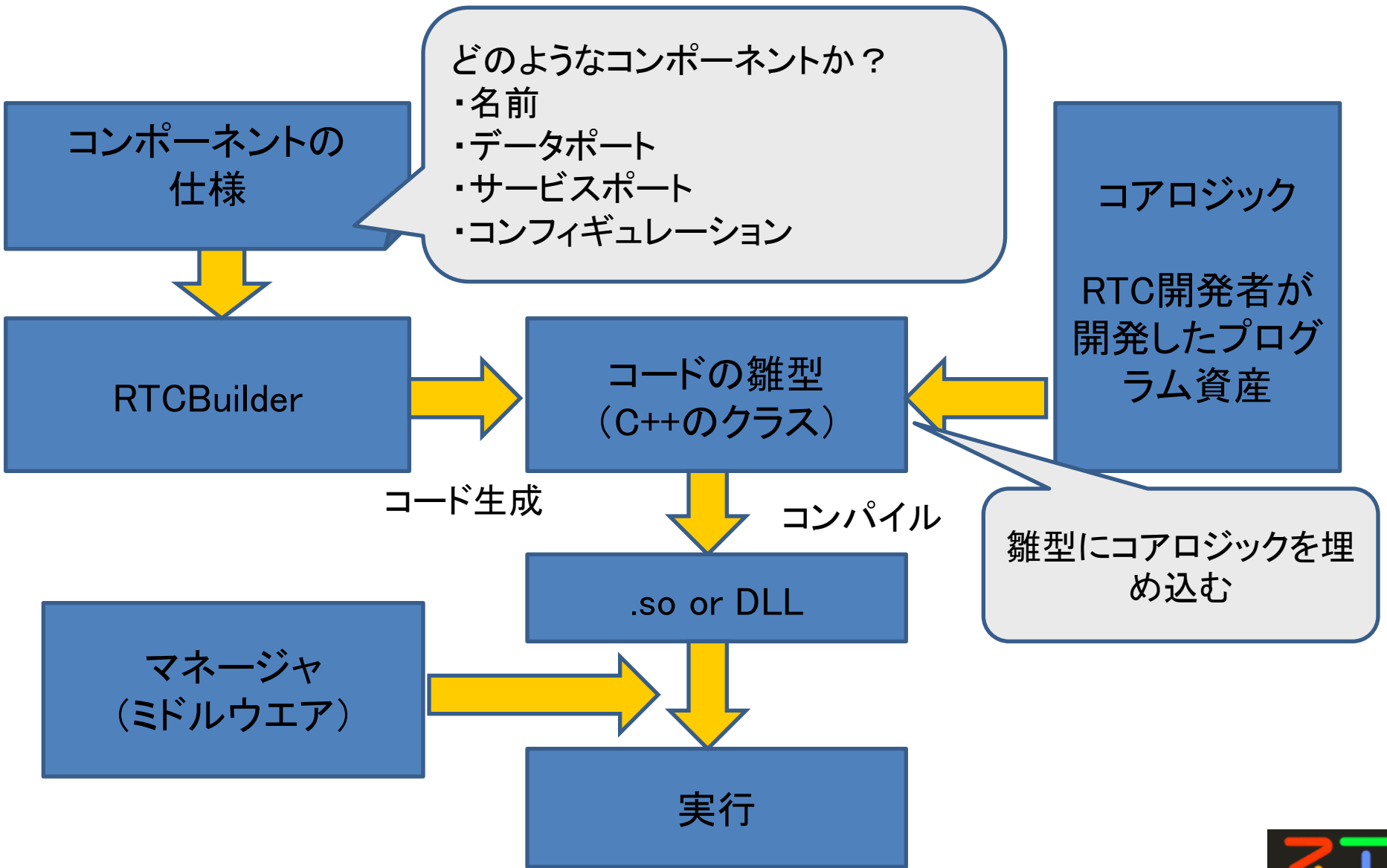




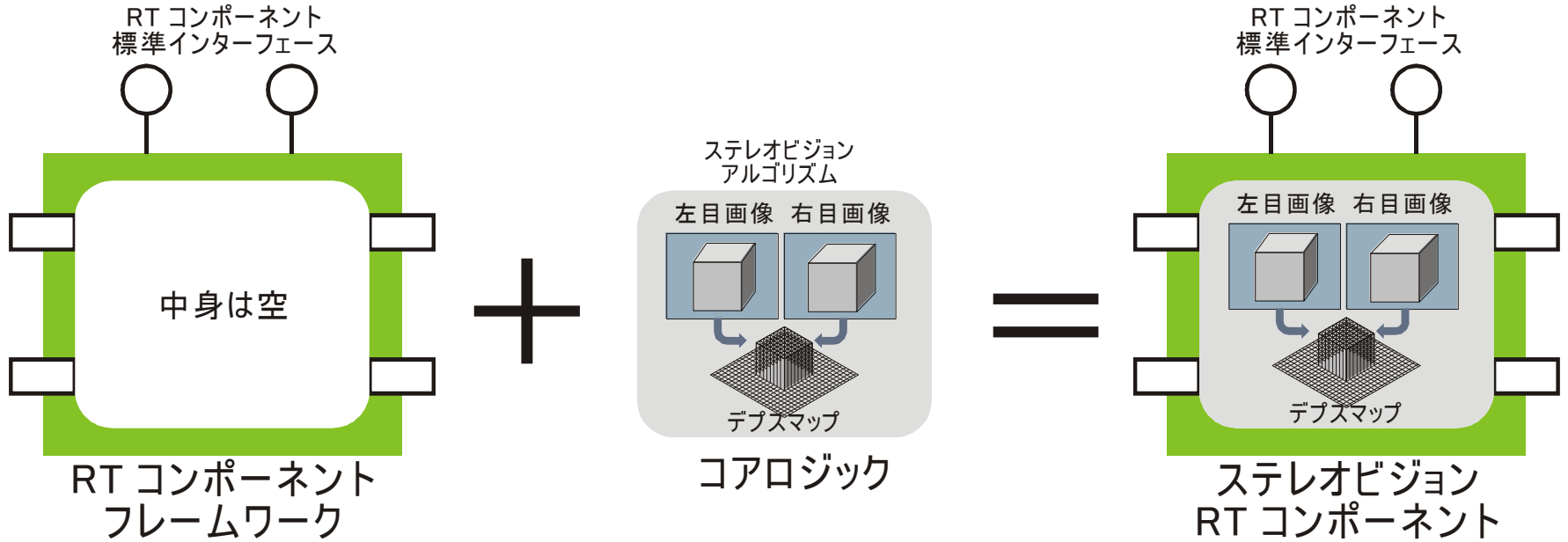
RTコンポーネントの開発・運用



RTC開発の流れ



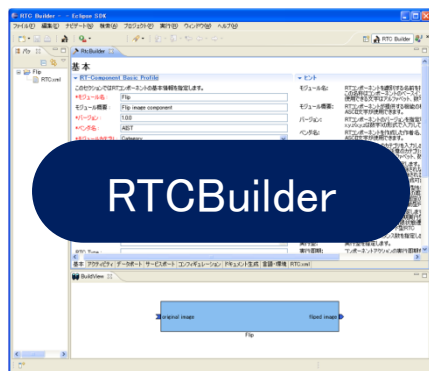
フレームワークとコアロジック



RTCフレームワーク+コアロジック
=RTコンポーネント

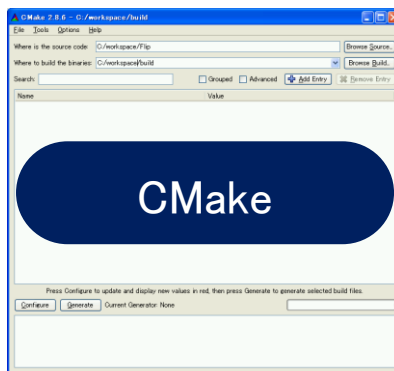
コンポーネントの作成・運用

Windowsの場合の開発イメージ



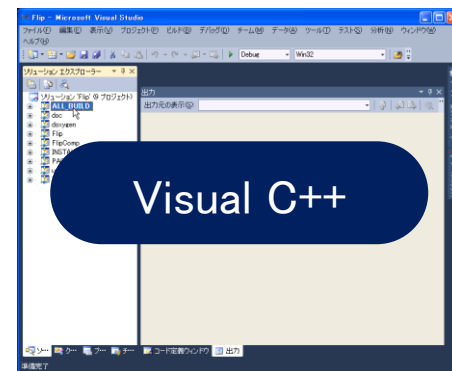
RTCBuilder

コンポーネントの仕様の入力



CMake

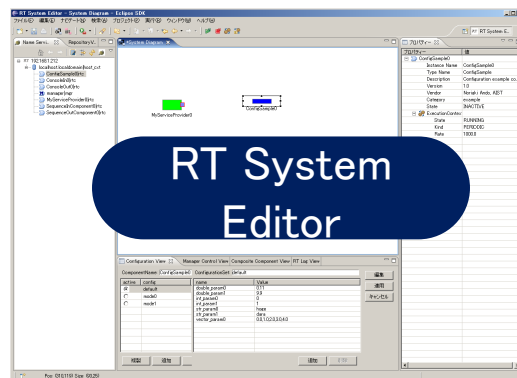
VCのプロジェクトファイルの生成



Visual C++

実装およびVCでコンパイル実行ファイルの生成

テンプレートコードの生成



RT System Editor

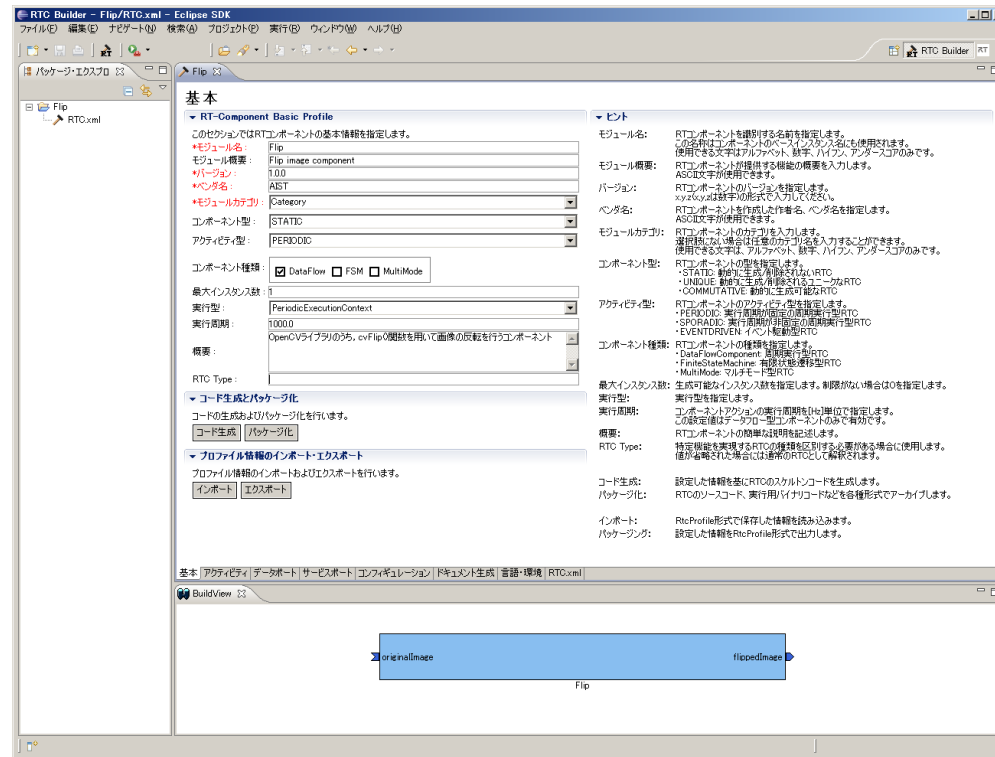
RTコンポーネント同士を接続し、RTシステムの構築

コンポーネント設計ツール

RTC Builder

- コンポーネントのプロファイル情報を入力し、ソースコード等の雛形を生成するツール
- 開発言語用プラグインを追加することにより、各言語向けRTCの雛形を生成することが可能

- C++
- Java
- Python

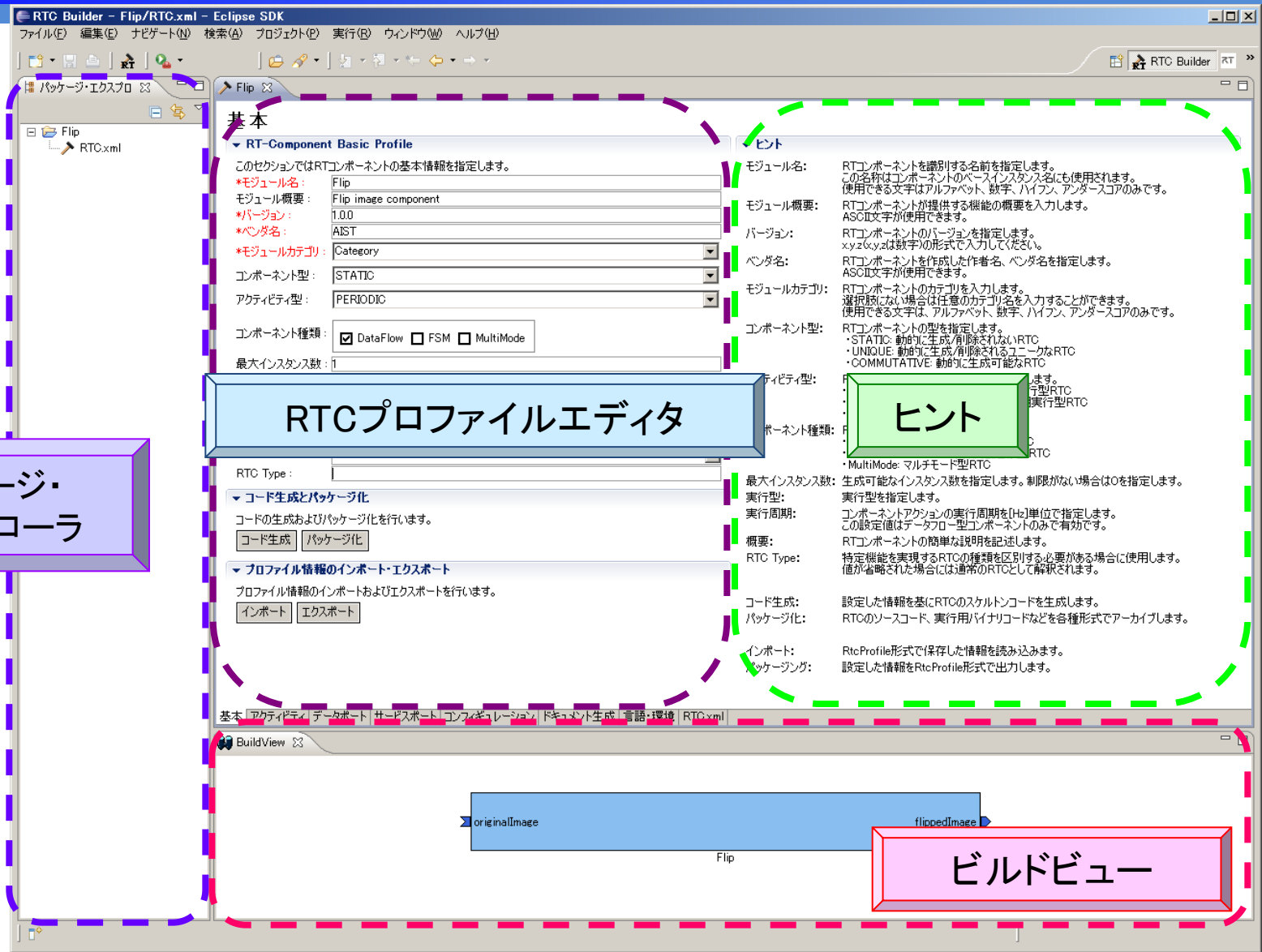


※C++用コード生成機能は

RtcBuilder本体に含まれています。

※その他の言語用コード生成機能は追加プラグインとして提供されています

RTC Builderの概観



The screenshot shows the RTC Builder Eclipse SDK interface. The main window is titled "RTC Builder - Flip/RTC.xml - Eclipse SDK". The interface is divided into several panes:

- Left Pane (Package Explorer):** Shows the project structure with "Flip" and "RTC.xml".
- Center Pane (Basic Profile):** Contains the "RT-Component Basic Profile" configuration. Fields include:
 - *モジュール名: Flip
 - モジュール概要: Flip image component
 - *バージョン: 1.0.0
 - *ベンダ名: AIST
 - *モジュールカテゴリ: Category
 - コンポーネント型: STATIC
 - アクティビティ型: PERIODIC
 - コンポーネント種類: DataFlow FSM MultiMode
 - 最大インスタンス数: 1
- Right Pane (Hints):** Contains a "ヒント" (Hints) section with detailed instructions for various fields like "モジュール名", "バージョン", "コンポーネント型", etc.
- Bottom Pane (Build View):** Shows a diagram of the component with two data flows: "originalImage" and "flippedImage".

パッケージ・
エクスプローラ

RTCプロファイルエディタ

ヒント

ビルドビュー

CMake

- コンパイラに依存しない, ビルド支援のためのツール.
- 異なる動作環境においても自動で環境に合わせてプロジェクトを生成可能

ダウンロードしてきたコンポーネントが動かない！
依存パッケージがよくわからない！
...

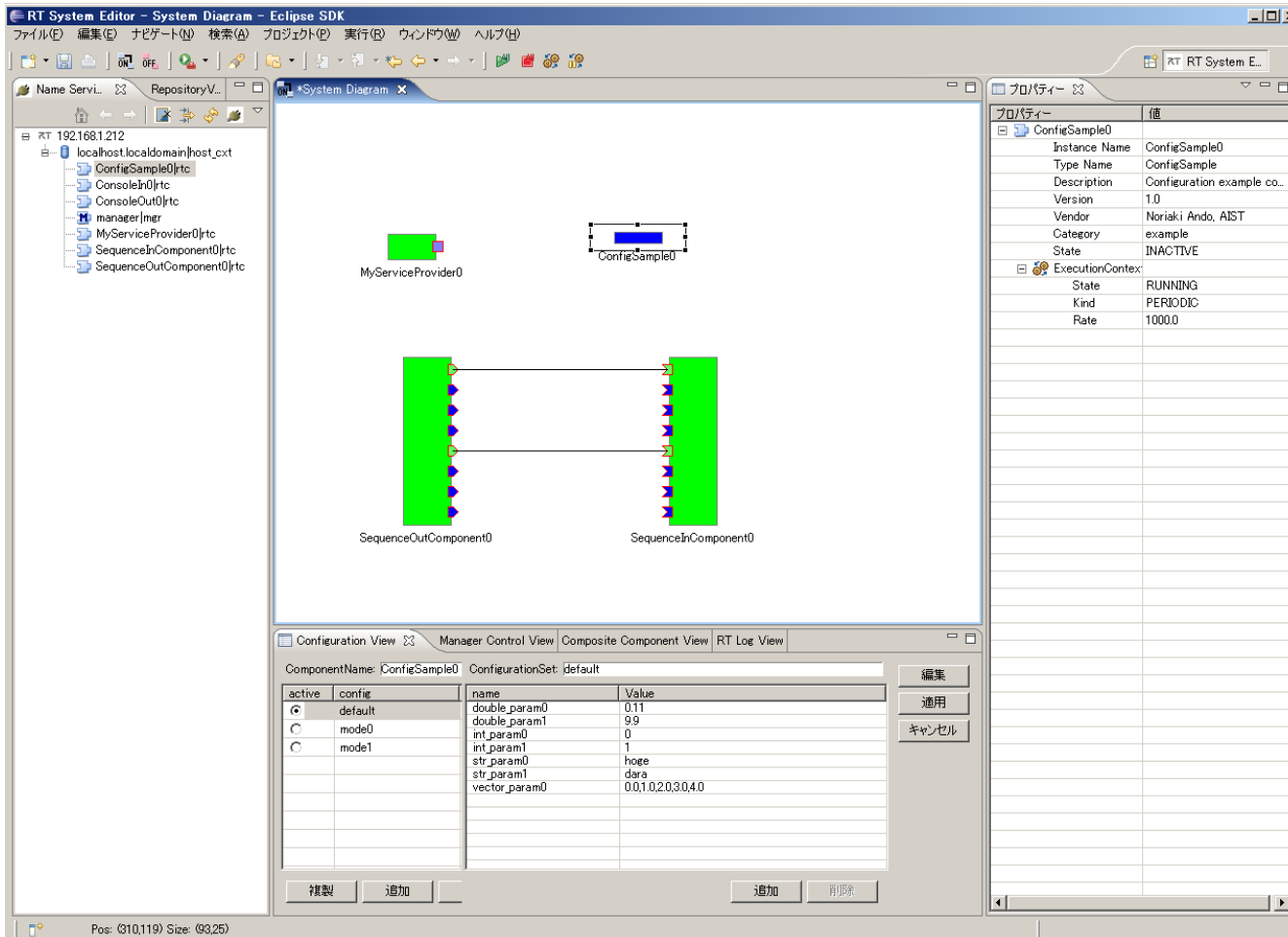
といった問題も解消しやすくなる.

OpenRTMのようなマルチプラットフォームで
動作するミドルウェアとは相性のよいツール

システム構築支援ツール

RT System Editor

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

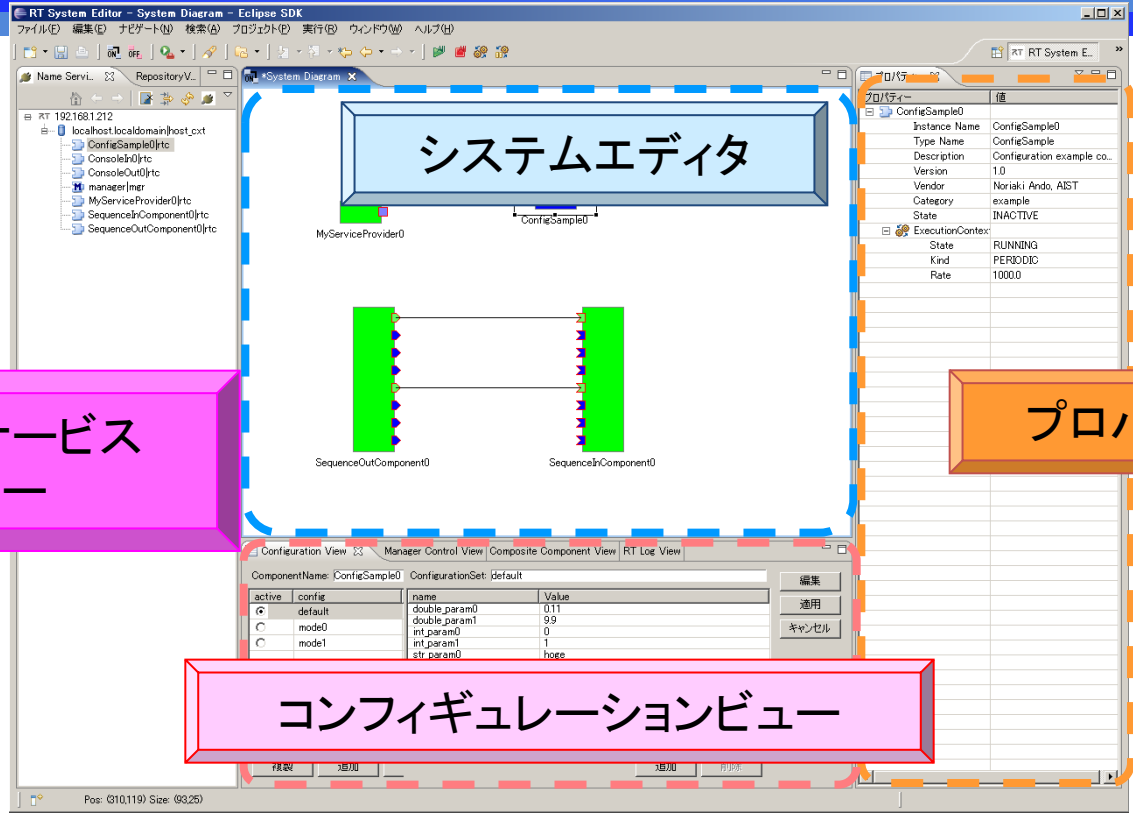


The screenshot displays the RT System Editor interface with the following components:

- System Diagram:** Shows a central configuration component (ConfigSample0) connected to two sequence components (SequenceOutComponent0 and SequenceInComponent0) via data flow lines.
- Component List (Left):** Lists available components such as ConfigSample0, ConsoleIn, ConsoleOut, manager, MyServiceProvider, SequenceInComponent, and SequenceOutComponent.
- Configuration View (Bottom):** A table showing the configuration for the selected component (ConfigSample0).

active	config	name	Value
<input checked="" type="radio"/>	default	double_param0	0.11
<input type="radio"/>	mode0	double_param1	9.9
<input type="radio"/>	mode1	int_param0	0
		int_param1	1
		str_param0	hoge
		str_param1	dara
		vector_param0	0.01,0.20,0.40

RT System Editorの概観

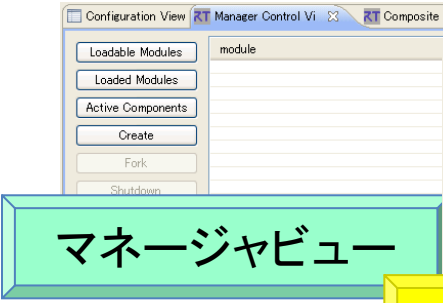


システムエディタ

ネームサービス
ビュー

プロパティビュー

コンフィギュレーションビュー

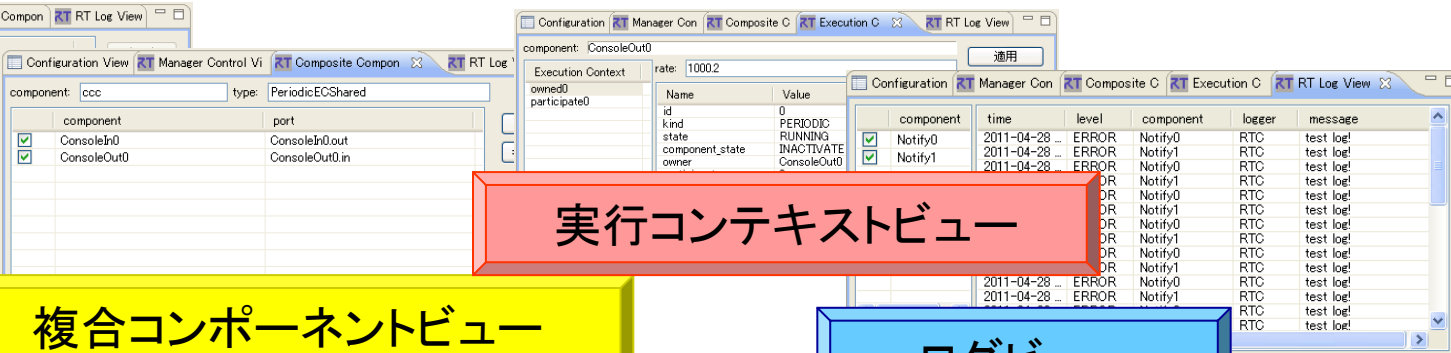


マネージャビュー

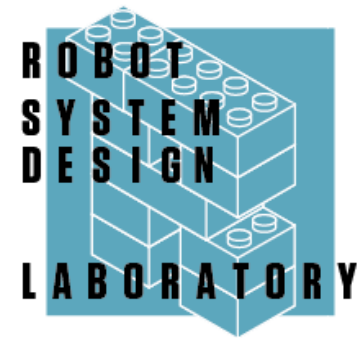
複合コンポーネントビュー

実行コンテキストビュー

ログビュー



component	time	level	component	logger	message
Notify0	2011-04-28	ERROR	Notify0	RTC	test log!
Notify1	2011-04-28	ERROR	Notify1	RTC	test log!
Notify0	2011-04-28	ERROR	Notify0	RTC	test log!
Notify1	2011-04-28	ERROR	Notify1	RTC	test log!
Notify0	2011-04-28	ERROR	Notify0	RTC	test log!
Notify1	2011-04-28	ERROR	Notify1	RTC	test log!
Notify0	2011-04-28	ERROR	Notify0	RTC	test log!
Notify1	2011-04-28	ERROR	Notify1	RTC	test log!



RTコンポーネントの再利用



RTC再利用のメリット・デメリット

メリット

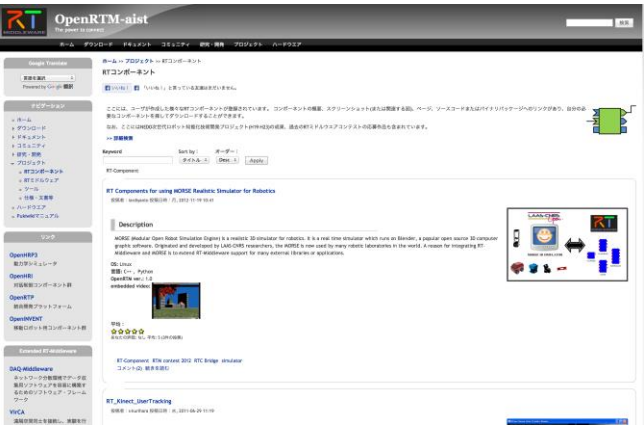
- ソフトウェアの開発にかかる工数削減
 - 本質的な部分に注力出来る.
- 最新の技術が利用可能

デメリット

- 良くも悪くもブラックボックス
- すべてがすんなり動くわけではない
 - 多少動かすことに手間がかかることも.

RTコンポーネントの探し方

- OpenRTM-aistのサイトで検索する。
 - NEDO知能化プロジェクトの成果やRTミドルウェアコンテスト作品が検索可能
- Google等で検索をかける。
 - RTミドルウェアユーザの公開しているRTCを見つけることができる。



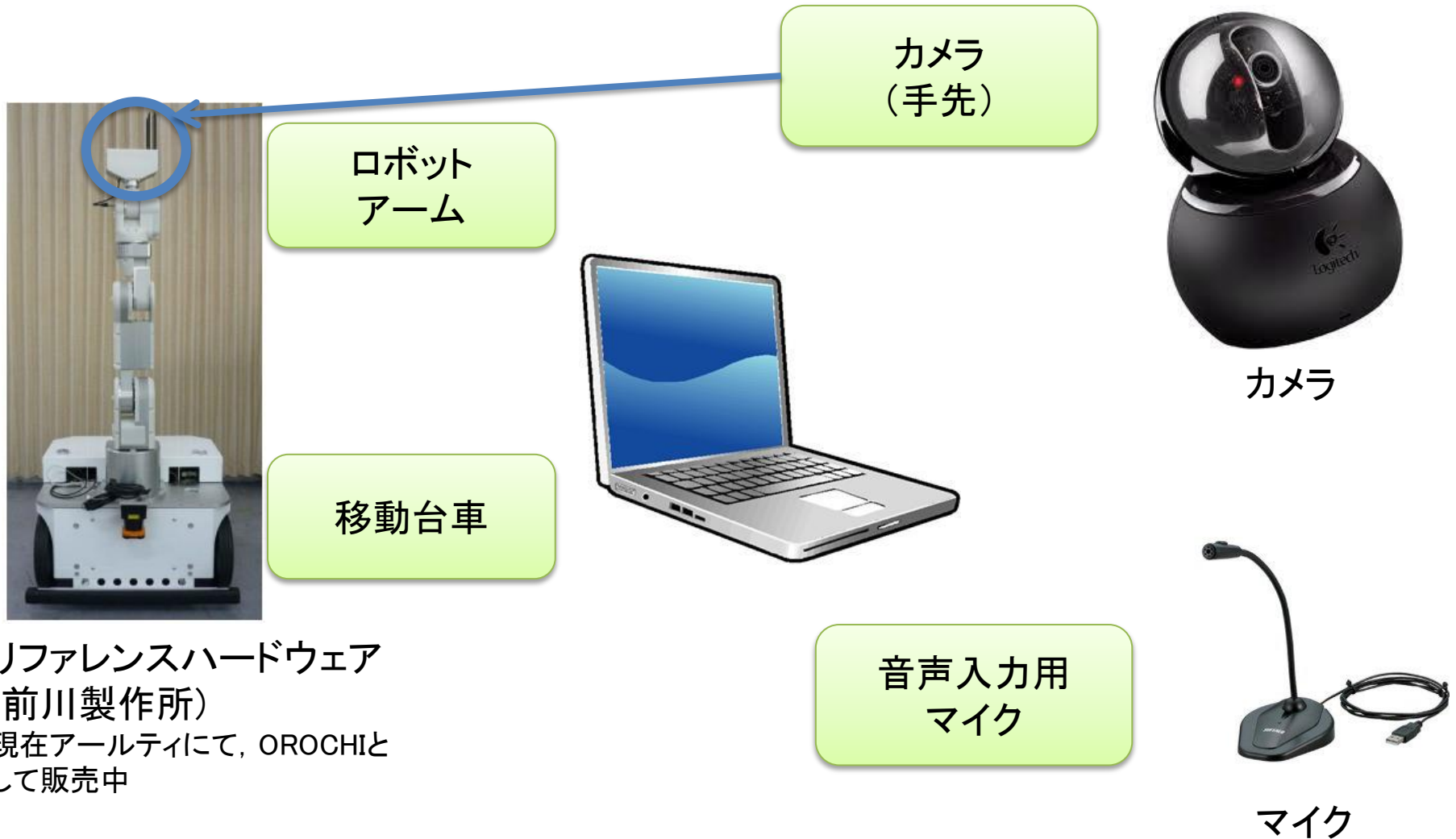
OpenRTM-aistのHP

研究室で公開しているHP

ユーザHP



構築したシステムの例



リファレンスハードウェア
 (前川製作所)
 現在アールティにて、OROCHIと
 して販売中

音声入力可能なRTC群

- OpenHRI

- コミュニケーション用途のRTコンポーネント群



OpenHRI
Opensource software components for Human Robot Interaction

HOME チュートリアル マニュアル 各コンポーネントの説明 ダウンロード サポート

Type your search here... GO

Featured Posts

ライセンスについて
OpenHRIのライセンスは、Eclipse Public License (EPL)です。 <http://www.eclipse.org/legal/epl-v10.html> EPLは、コードの改変や再配布、営利利用を許可するオープンソースライセンスですが、場合によっては改変内容をライセンス元に開示する必要があります。 OpenHRIが内部で用いているJuliusやOpen...

Readmore

オーディオコンポーネントをアップデートしました

Posted on : 23-02-2012 | By : [Yosuke Matsusaka](#) | In : Uncategorized

オーディオコンポーネントをアップデートしました (バージョン2.01)。

変更点は以下の通りです：

- ・VADコンポーネント (背景雑音のオンライン学習機能を備える音声検出・雑音抑制コンポーネント)の追加
- ・WAVファイル再生コンポーネントの追加
- ・ゲートコンポーネントの追加
- ・その他のバグフィックス

アップデートを適用するには、Ubuntuの場合、端末上で

```
Code block
% sudo apt-get update
% sudo apt-get upgrade
```

と入力してください。

Windowsの場合、

<http://openhri.net/getinstaller.php>

上記のURLからダウンロード出来るアップデートを起動してください。

目次

- チュートリアル
 - Step1 インストール
 - Ubuntu版
 - Windows版
 - Step2 RTSystemEditorの使い方と音声入出力コンポーネントのテスト
 - Step3 音声合成コンポーネントのテスト
 - Step4 対話システムを作る
 - Step5 対話システムを外部に接続しよう
- アプリケーション例
 - マルチリンガル対話システム
 - 旗上げゲーム
 - 音響エコー除去
- マニュアル
- 各コンポーネントの説明

特徴

- ・マニュアルが整備されている。
- ・チュートリアルがわかりやすい。
- ・WindowsやLinuxで利用可能。

利用しやすいコミュニケーション系RTコンポーネントの代表格

応用事例

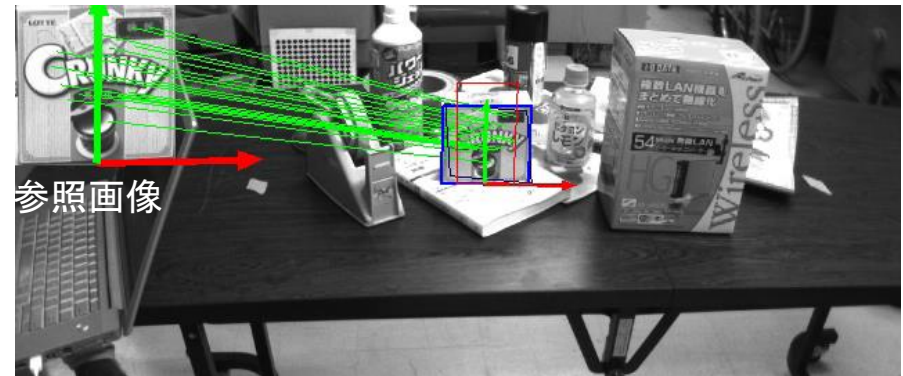
- 物体探索システム
- 小型ヒト型ロボットの音声による制御
- モバイルマニピュレータの制御

把持物体情報を獲得するRTC

アピランスベース物体認識RTC

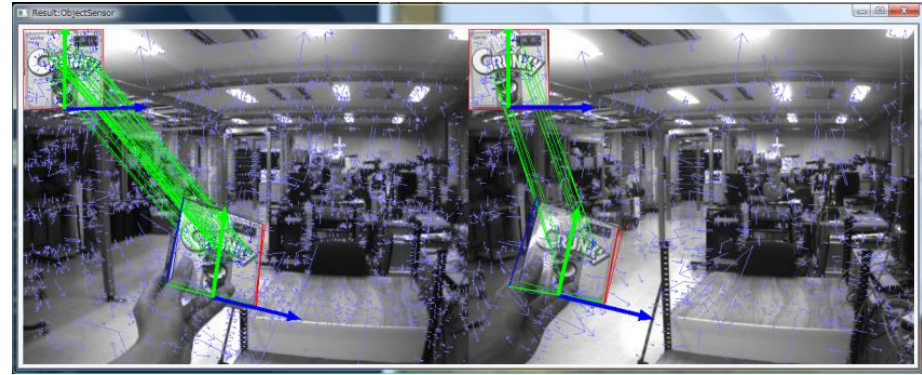
概要:
 参照画像から得られるSIFT特徴量を用いた物体検出にGPUを用いることで、CPUのみを用いた手法と比較して、高速な目標物体の位置及び姿勢を提供する機能を実現する。ステレオカメラ画像と単眼カメラ画像の両方に対応。

- 特徴:**
- ◆1枚の参照画像のみで、位置姿勢を推定できる
 - ◆SIFT特徴によるロバストな推定
 - ◆GPUによる高速化
 - ◆ステレオカメラにも対応



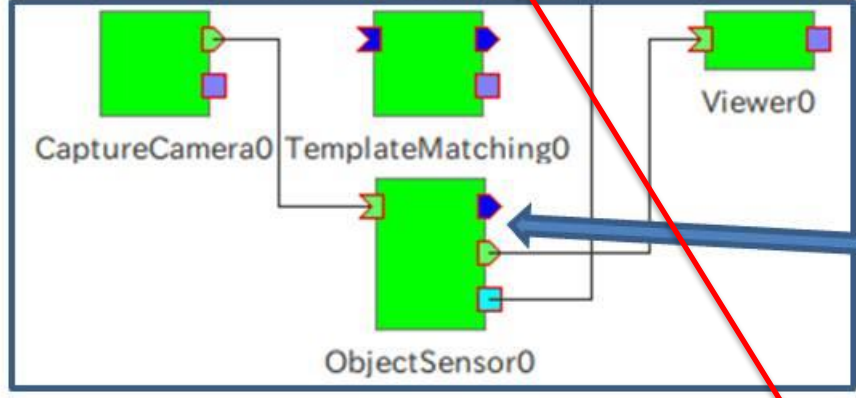
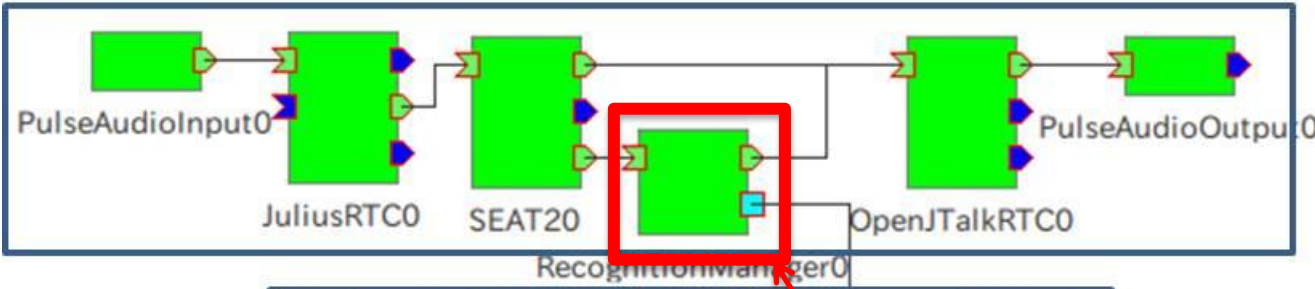
**モデルデータを用いて、
 物体の位置・姿勢を推定**

共通カメラ/Fに対応したものを利用

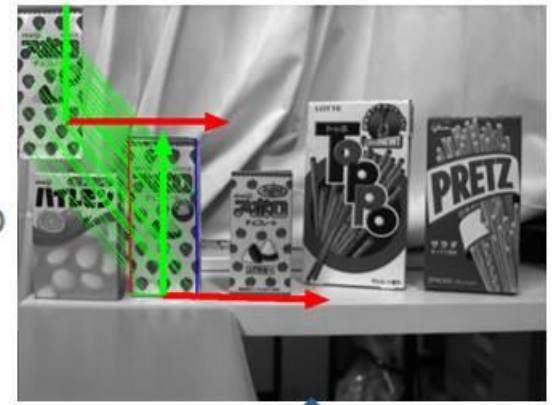


音声入力による物体探索システム

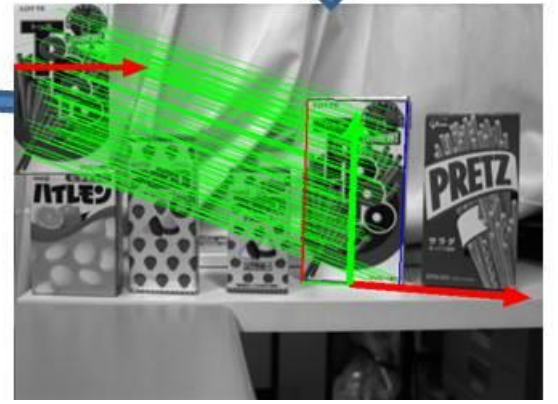
Voice Recognition components



Object Recognition components



Change model \updownarrow by voice

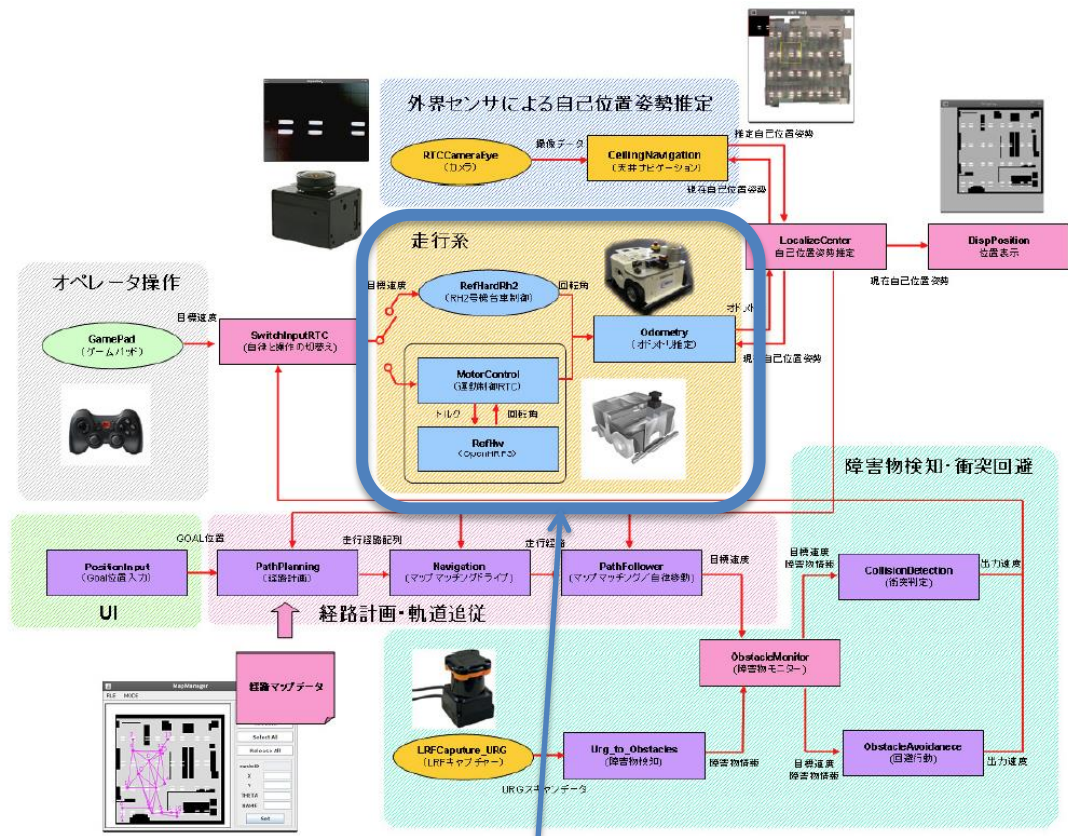


自身で作成したRTC

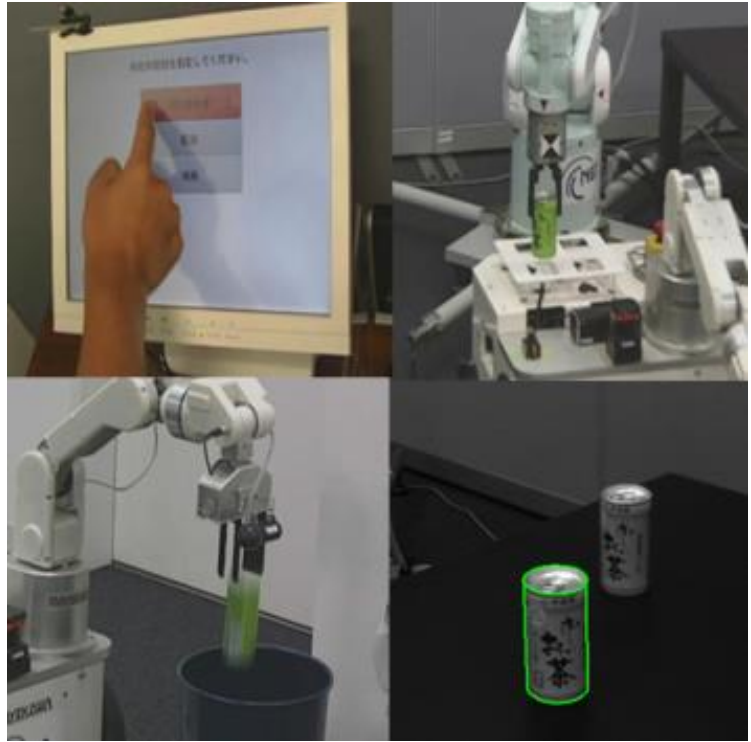
音声認識結果から、物体認識用のモデルデータに変換するコンポーネントのみ作成することで実現可能

ロボット制御系のRTC群

OpenRTC-aistで公開されている来訪者受付システムから流用



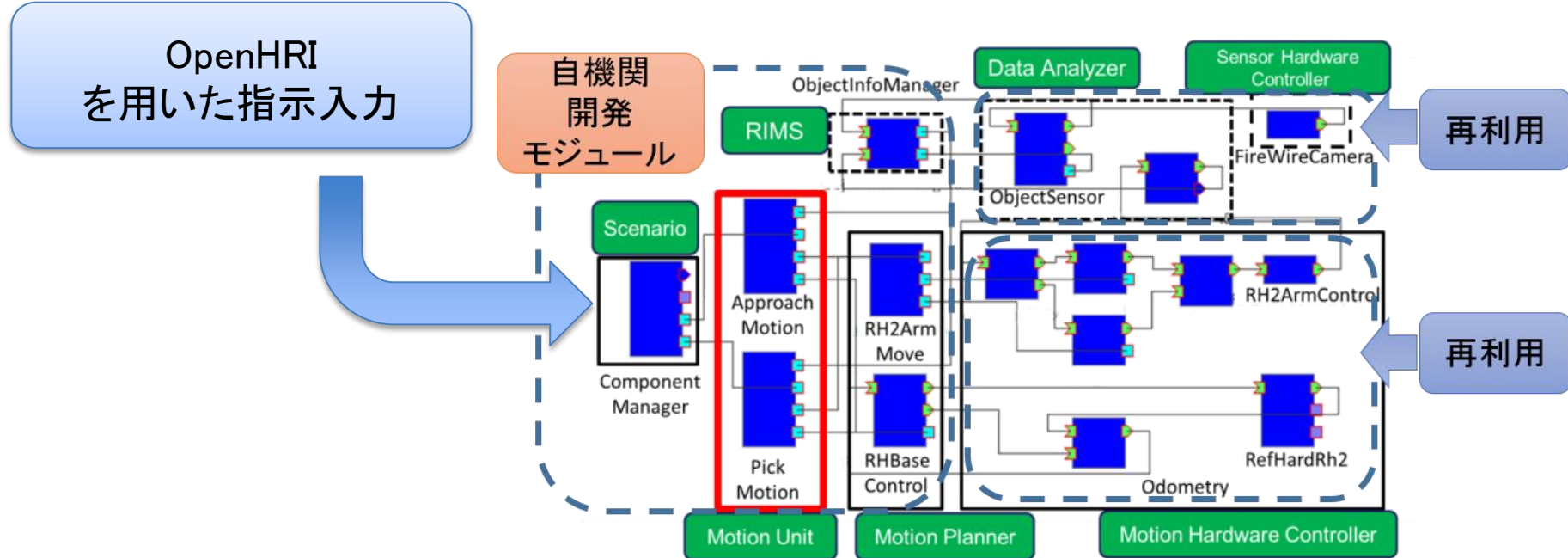
走行系部分のみ利用



アーム制御部のみ利用

ハードウェアに関連するRTCを用いることで、工数の大幅削減

用意したRTCの統合



リファレンスハードウェアでのPick-and-Place

ロボットの制御，画像処理，インタフェースを再利用システム統合部のみに注力可能

構築したシステム



第一部では，以下の説明を行った。

- OpenRTM-aistの概要
- RTコンポーネントの基本機能
- RTコンポーネントの開発・運用
- RTコンポーネントの再利用例