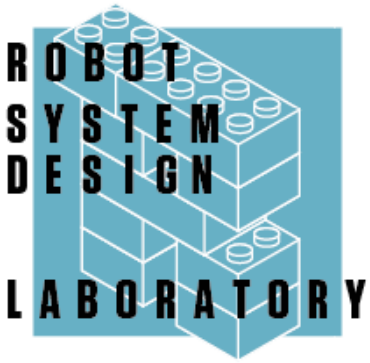


第1部　RTミドルウェア
OpenRTM-aist概要

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ミドルウエア

OpenRTM-aist

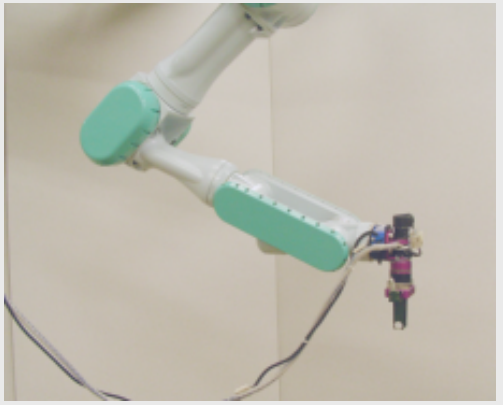
- RT-Middleware　(RTM)
  - RT要素のインテグレーションのためのミドルウエア
- RT-Component　(RTC)
  - RT-Middlewareにおけるソフトウエアの基本単位

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？

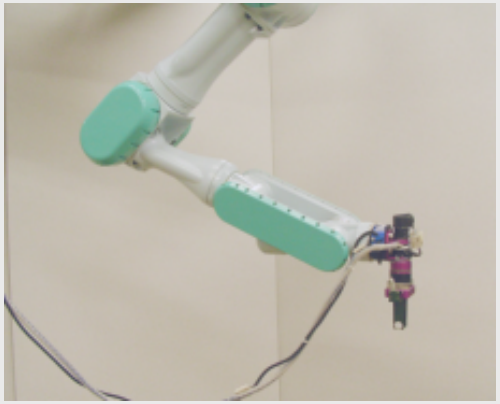
Joystick

Joystick
software

Humanoid's Arm
Control software

Humanoid's Arm

Robot Arm
Control software



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

RTミドルウェア

RTミドルウエアは別々に作られたソフトウエ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

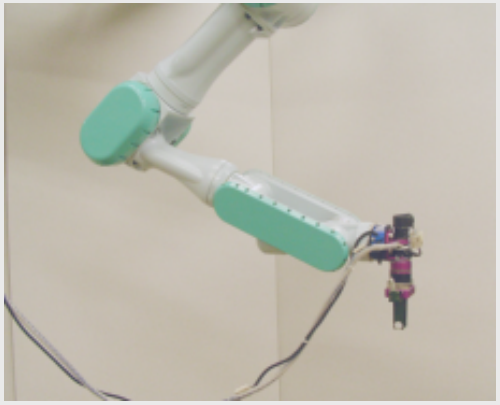
Joystick software

compatible arm interfaces

Arm A Control software

Humanoid's Arm

Arm B Control software



Robot Arm

ソフトウエ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

- ミドルウエア
  - 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エア
  - 例：RDBMS，ORB等．定義は曖昧．
- 分散オブジェクト（ミドルウエア）
  - 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エアの断片（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

モジュール化のメリット

モジュール化していくことでシステム開発に関わる様々な点でメリットを享受することができる

- 再利用性の向上
  - 同じコンポーネントを様々なシステムに応用可
- 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
- 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

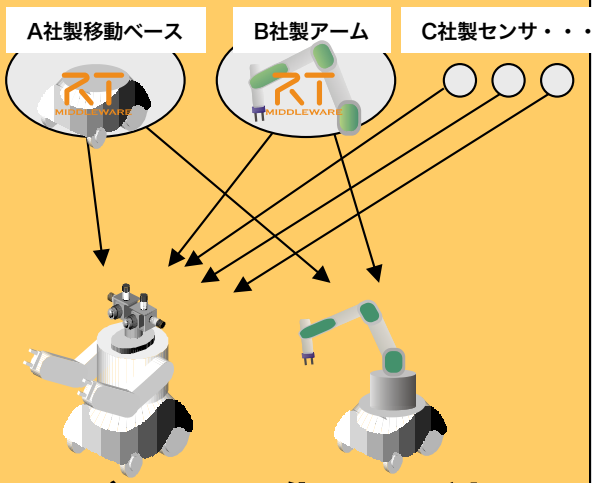
モジュール化のメリットに加えて

- ソフトウエアパターンを利用可能
  - ロボットに特有のソフトウエ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
- 分散ミドルウエア
  - 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
- ツールの利用
  - ロボットシステム開発を円滑化する様々なツールを利用可能.

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

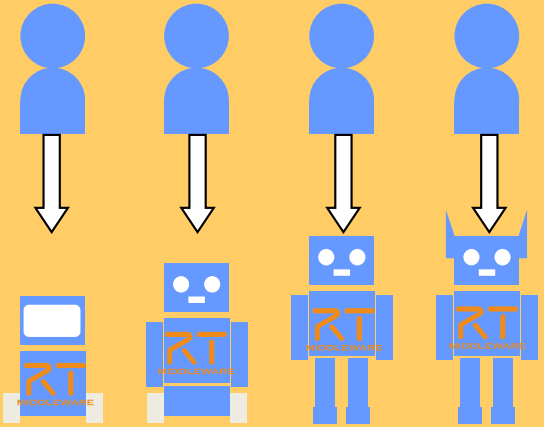
ロボットの低コスト化

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

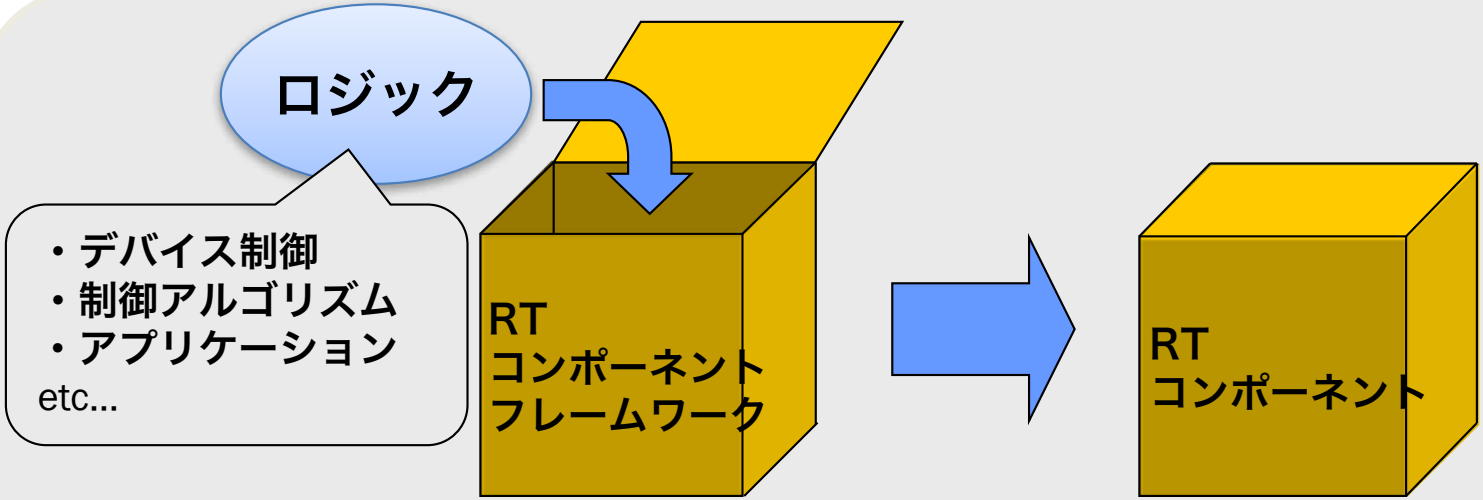


カスタマイズが容易に

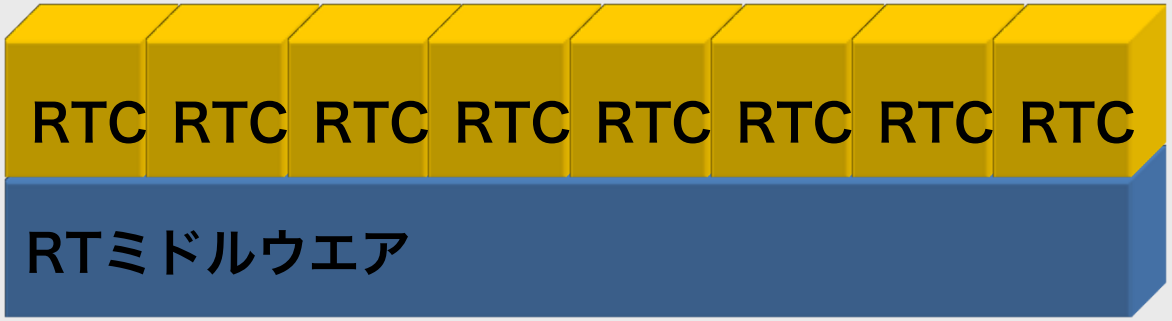
多様なニーズに対応

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



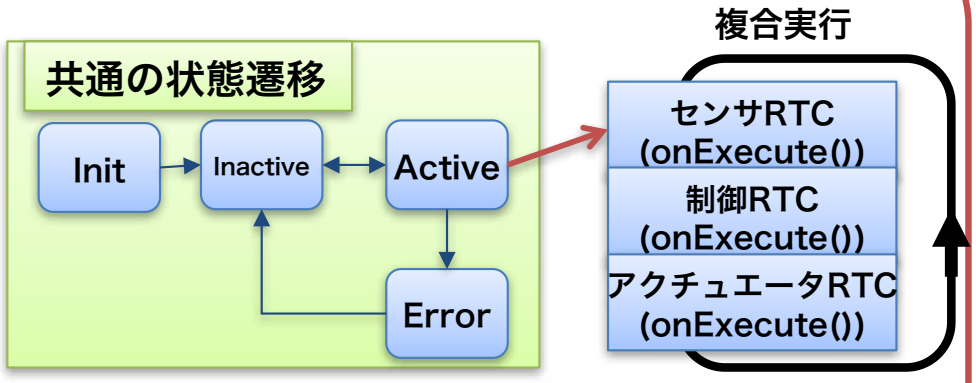
ロジックを箱（フレームワーク）に入れたもの＝RTコンポーネント（RTC）



RTCの実行環境（OSのようなもの）＝RTミドルウエア（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

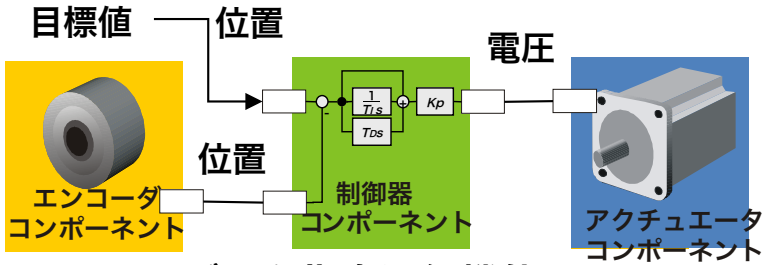
アクティビティ・実行コンテキスト



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

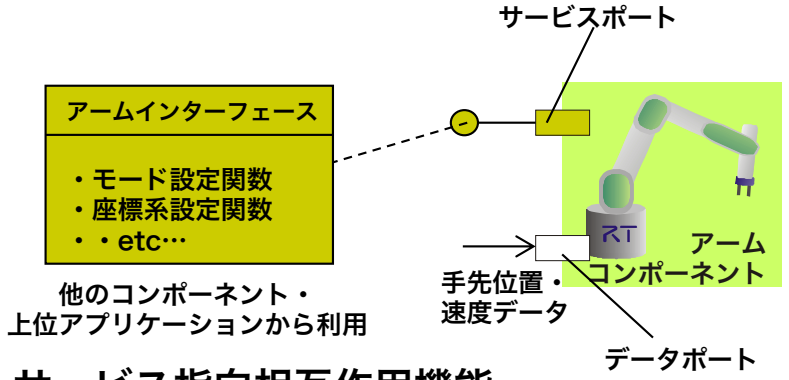
- データ指向ポート
- 連続的なデータの送受信
- 同じデータ型のポート同士接続可能
- 動的に接続・切断可能



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

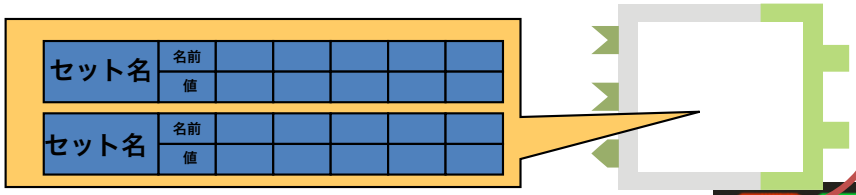


サービス指向相互作用機能

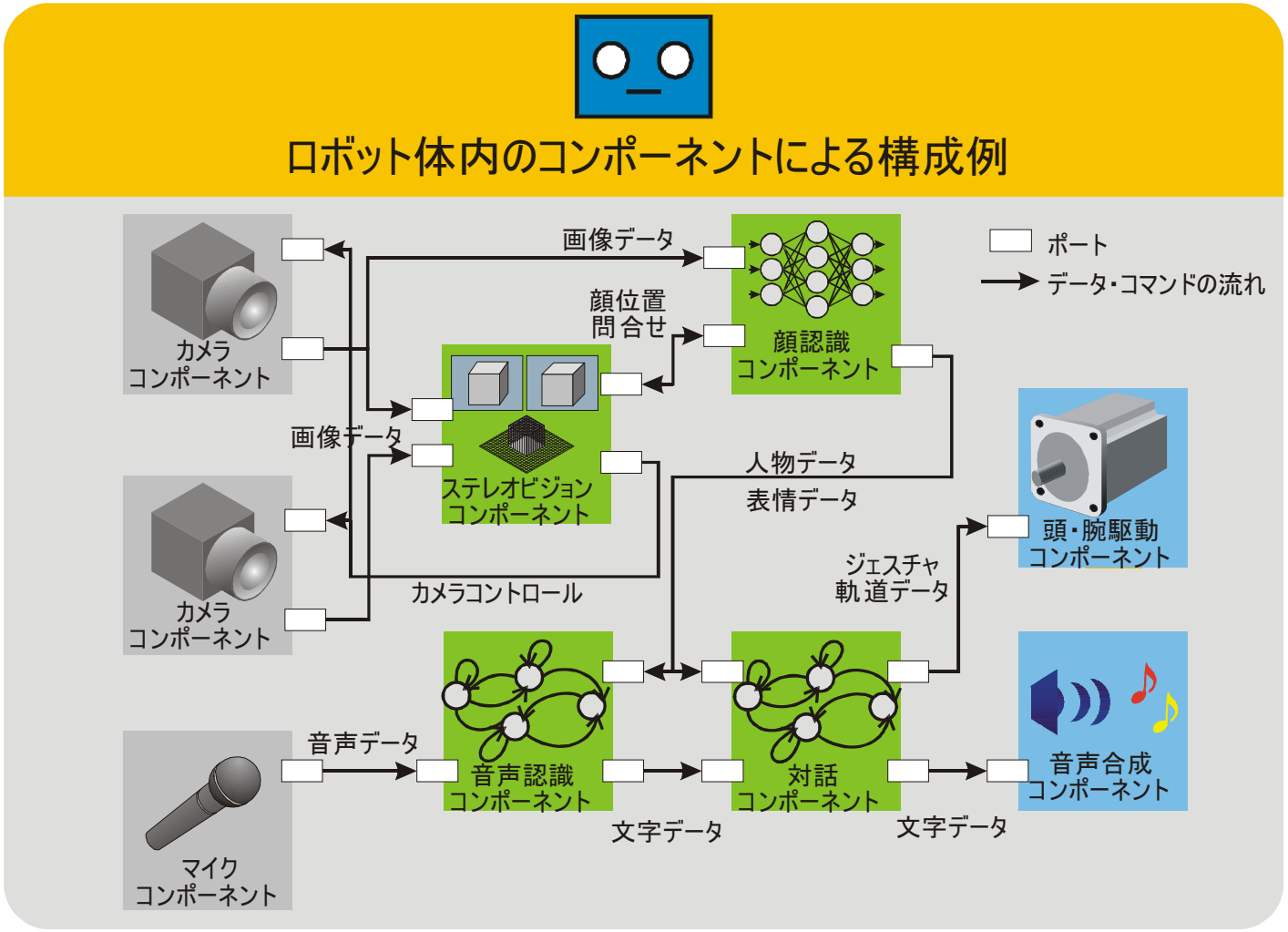
コンフィグレーション

- 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



RTCの分割と連携

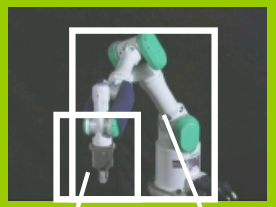


コンポーネントの中実の隠蔽と
他のコンポーネントをつなぐインタフェースの仕様が重要

RTMを用いたシステム構築例

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

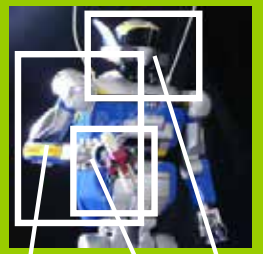


RTC | RTC
RTM
Solaris

ロボットB

RTC | RTC
RTM
FreeBSD

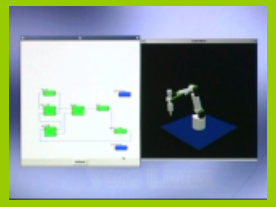
ロボットC



RTC | RTC | RTC
RTM
ARTLinux

ネットワーク

Linux
RTM
RTC | RTC



アプリケーション

Windows
RTM
RTC | RTC

操作デバイス

TRON
RTM
RTC | RTC

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出来る。

OSや幅広い開発言語を用いたRTコンポーネントの開発が可能

OpenRTM-aistとは？

- 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：Linux, FreeBSD, Windows, Mac OS X, QNX
  - 非公式: uITRON, T-Kernel, VxWorks
- 言語:
  - C++ (1.1.2), Python (1.1.2), Java (1.1.2)
  - .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

  ABにて承認、事実上の国際標準獲得

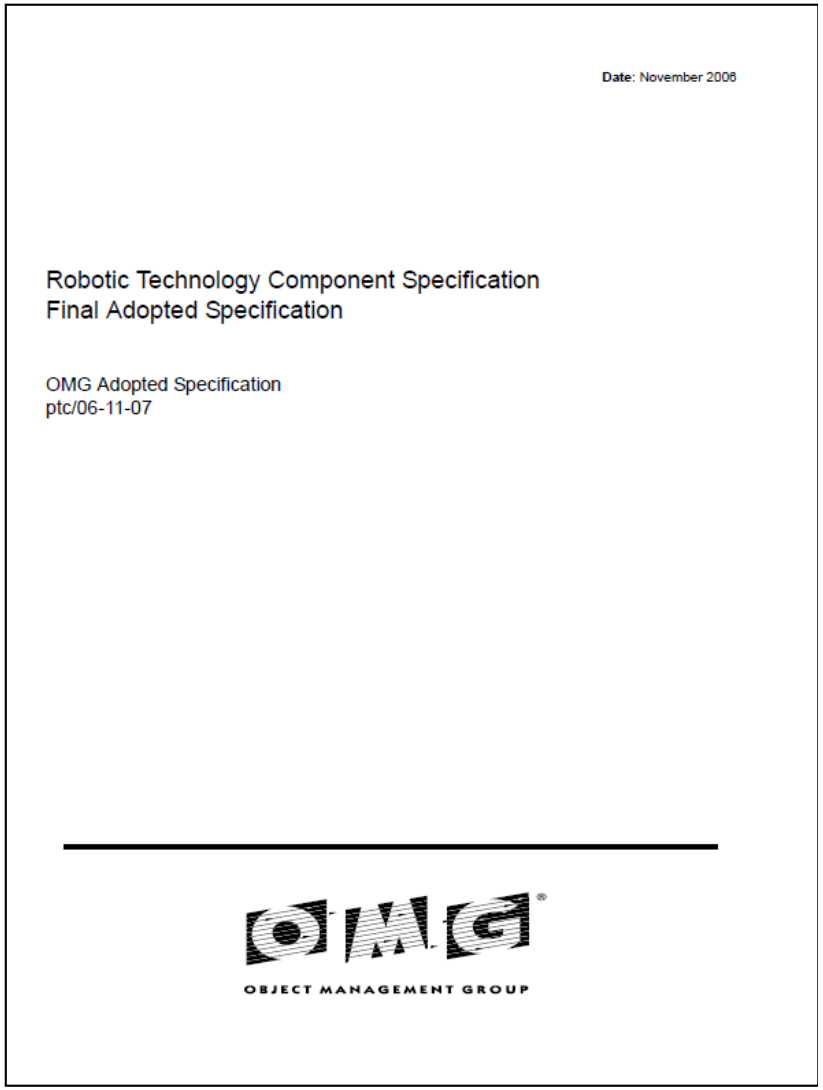
  FTFが組織され最終文書化開始
- 2007年8月

  FTFの最後の投票が終了
- 2007年9月

  AB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 2006

Robotic Technology Component Specification
Final Adopted Specification

OMG Adopted Specification
ptc/06-11-07

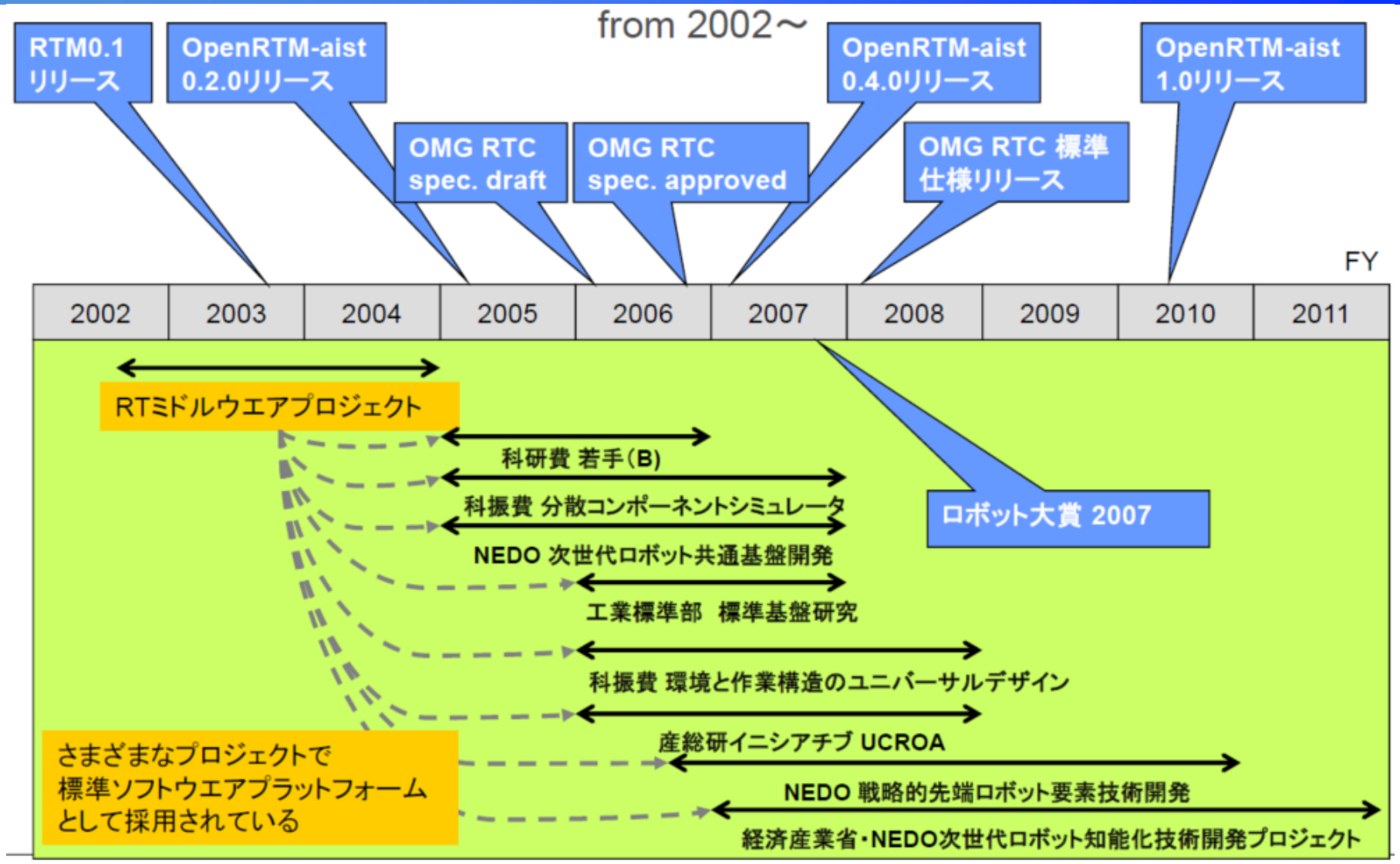
OBJECT MANAGEMENT GROUP

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ミドルウエア	◎
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)

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

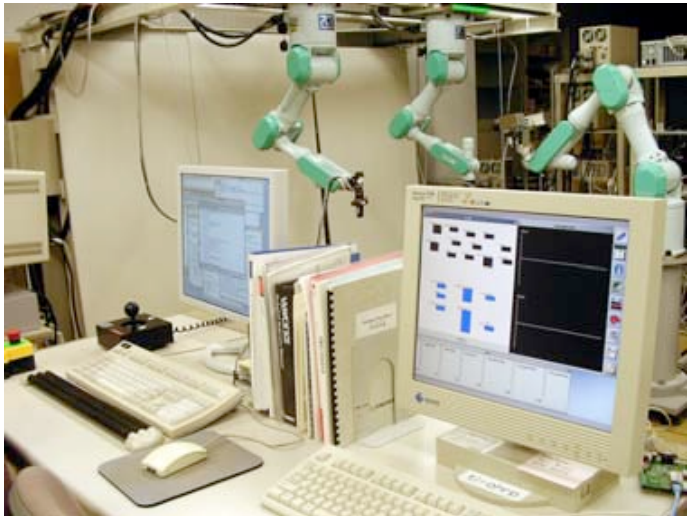
RTM関連プロジェクト



このほかにもRTミドルウェアを用いたプロジェクトが実施されており，資産の蓄積が進んでいる．

RTミドルウェアプロジェクト

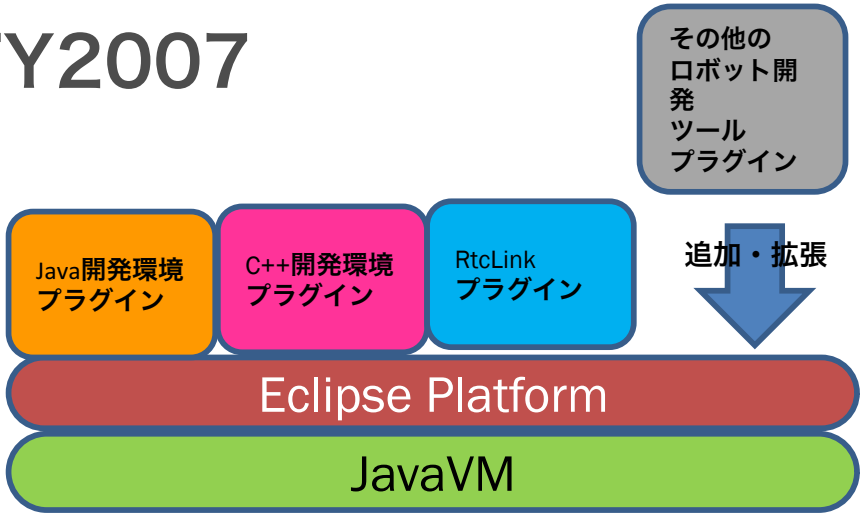
- 名称：NEDO 21世紀ロボットチャレンジプログラム
  - 「ロボット機能発現のために必要な要素技術開発」
- 目的：
  - RT要素の部品化（モジュール化）の研究開発
  - 分散オブジェクト指向開発
  - RT要素の分類・モジュール化に必要な機能・インタフェース仕様の明確化
- 予算規模：
  - 65百万円
  - 全体267.3百万円



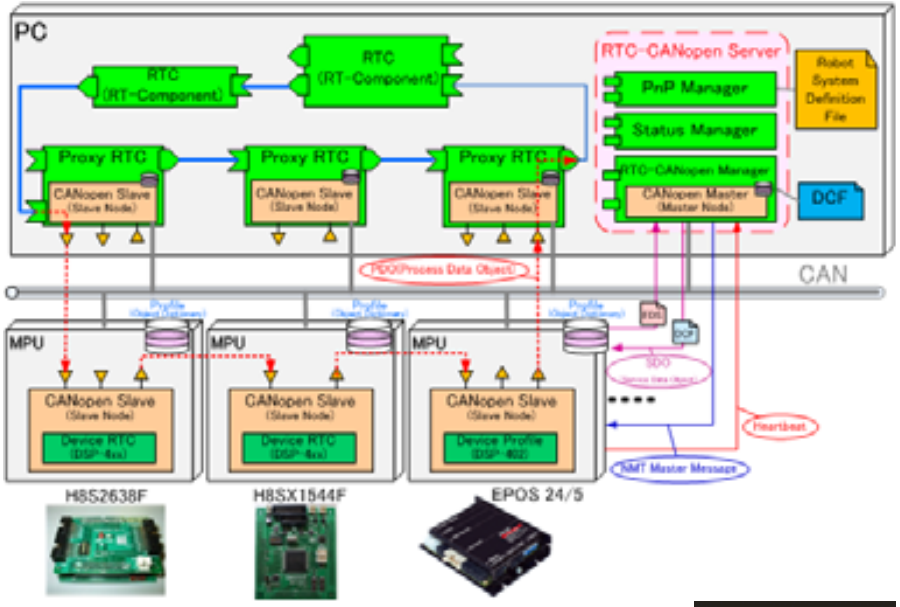
NEDO基盤PJ

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
  - デバイスに搭載する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



ツールのEclipseプラグイン化



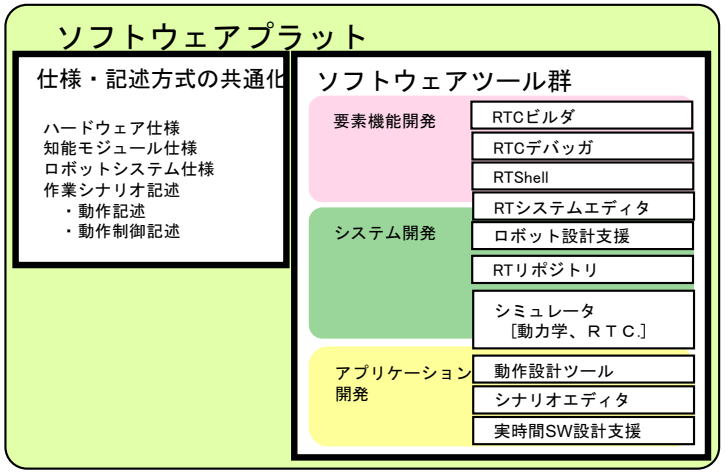
RTC-CANの開発

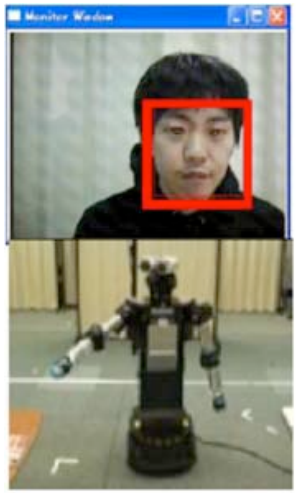
dsPIC版RTC-Liteの開発

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
  - ソフトウエ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

ユーザ数

1620名
（2017年7月2日現在）

プロジェクト登録数

タイプ	登録数
RTコンポーネント群	286
RTミドルウエア	21
ツール	20
仕様・文書	1

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

ユーザは国内中心．日本語の資料が多いのが特徴．

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

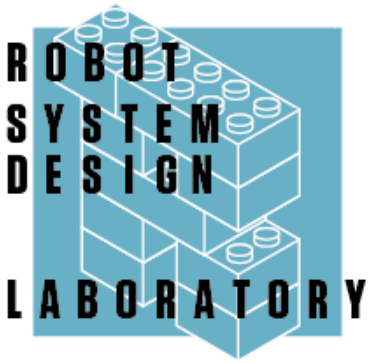
RTミドルウェアサマーキャンプ

- ８月上旬に開催(今年は7月31~4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

**2017年12月に
仙台で開催**

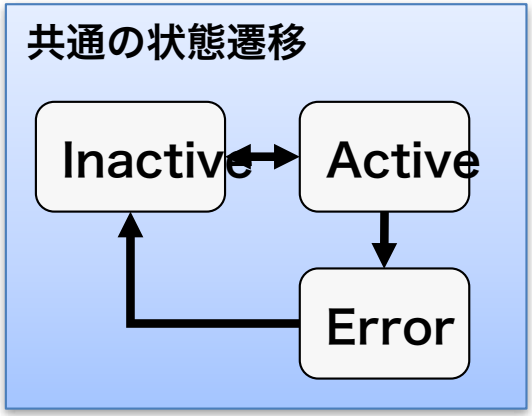
- 12月に計測自動制御学会システムインテグレーション部門講演会の特別セッションとして開催
- RTミドルウェアを用いたシステム，開発支援のためのツールなどを対象としたコンテスト

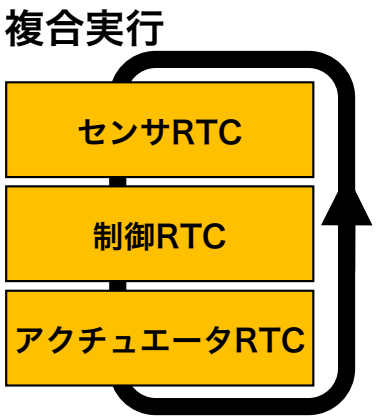


RTコンポーネントの基本機能

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

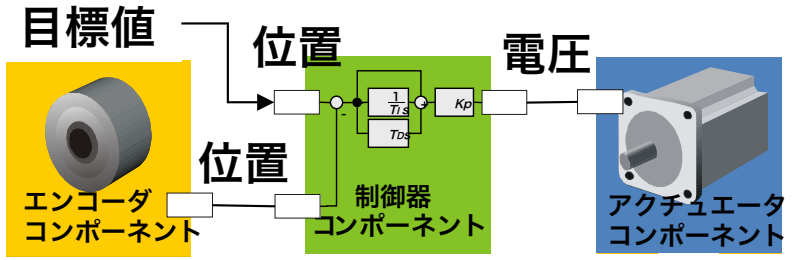




ライフサイクルの管理・コアロジックの実行

データポート

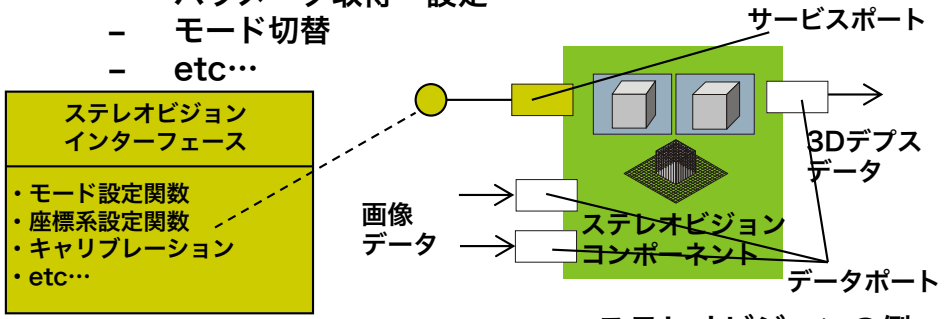
- データ指向ポート
- 連続的なデータの送受信
- 動的な接続・切断



データ指向通信機能

サービスポート

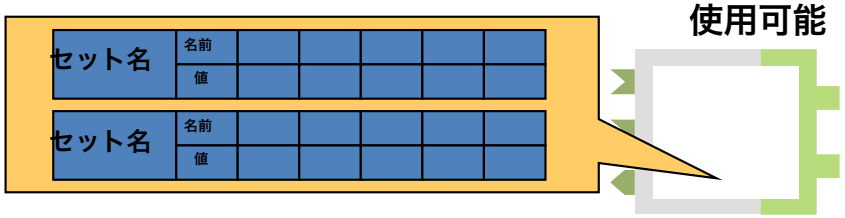
- 定義可能なインターフェースを持つ
- 内部の詳細な機能にアクセス
  - パラメータ取得・設定
  - モード切替
  - etc…



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

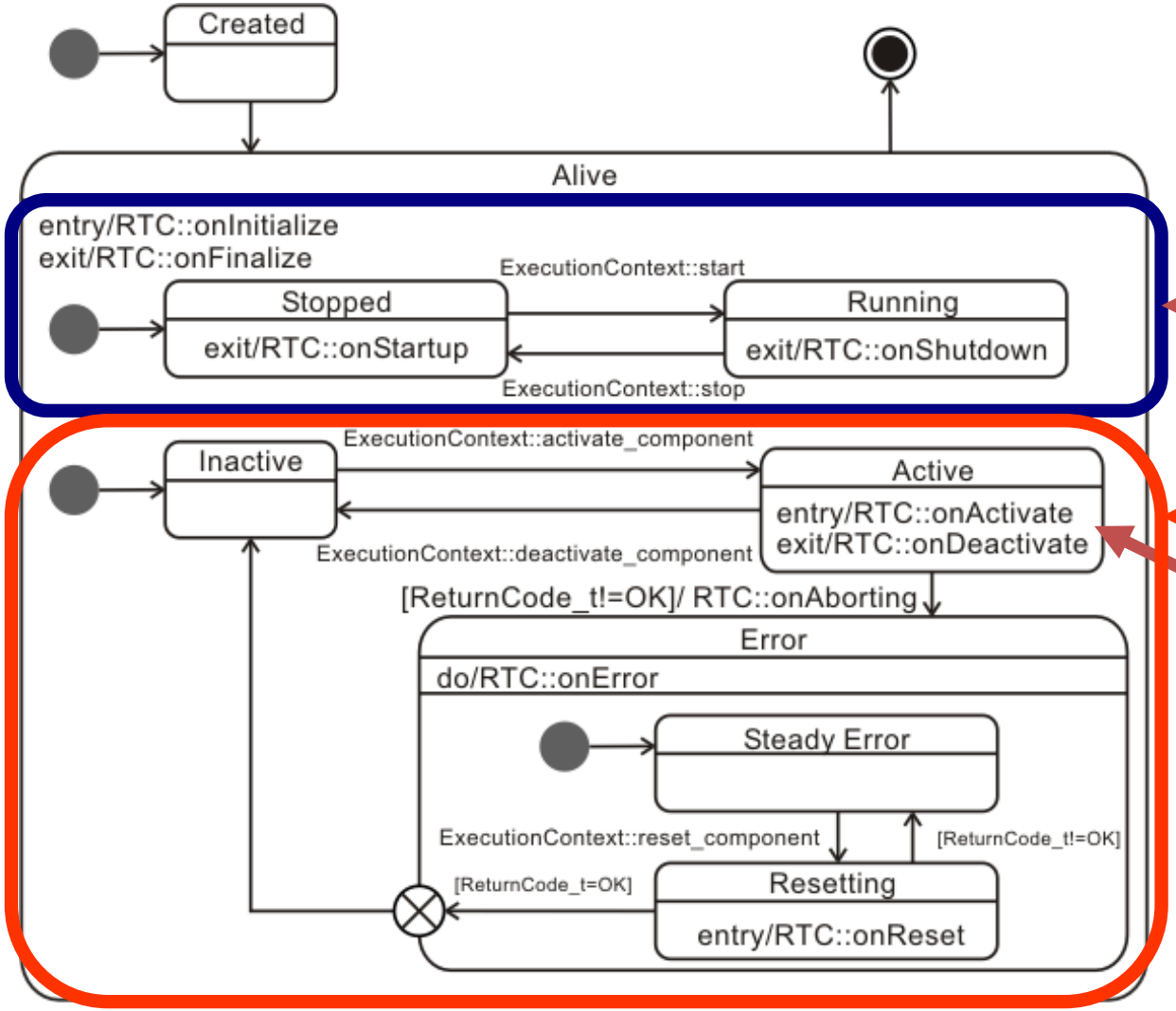


複数のセットを
動作時に
切り替えて
使用可能

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．



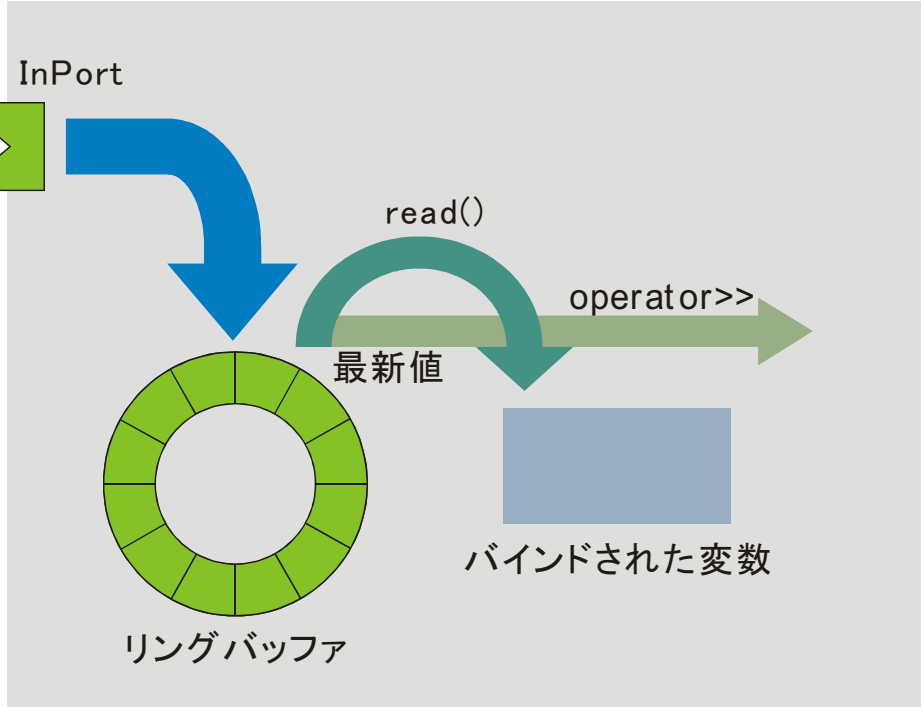
ユーザがあまり
意識しなくてよい部分

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

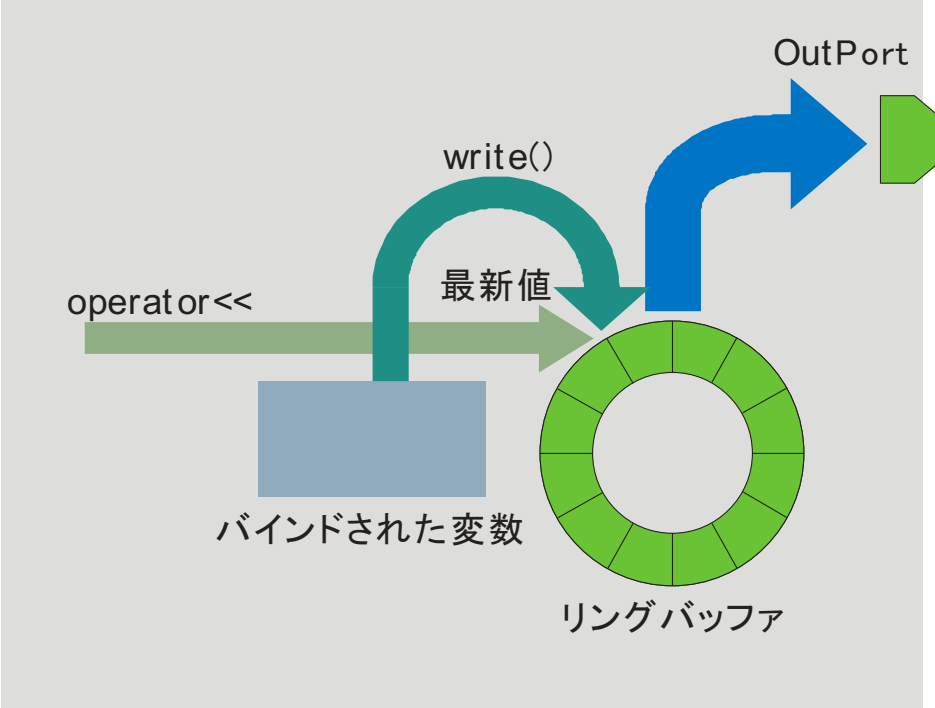
例

Sensor Data → Robot Component

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)

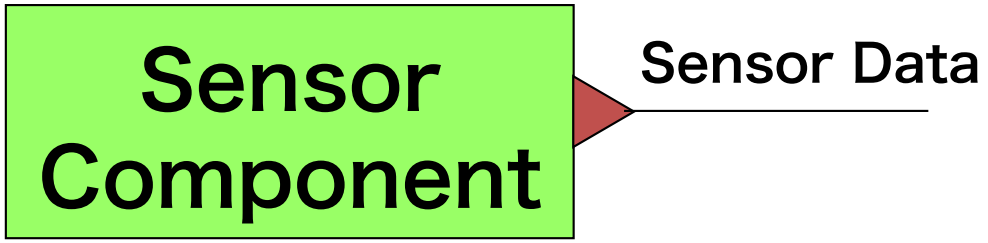
- OutPortのテンプレート第2引数：バッファ
  - ユーザ定義のバッファが利用可能
- OutPortのメソッド
  - write(): OutPort バッファへバインドされた変数の最新値として書き込む
  - >>：ある変数の内容を最新値としてリングバッファに書き込む



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

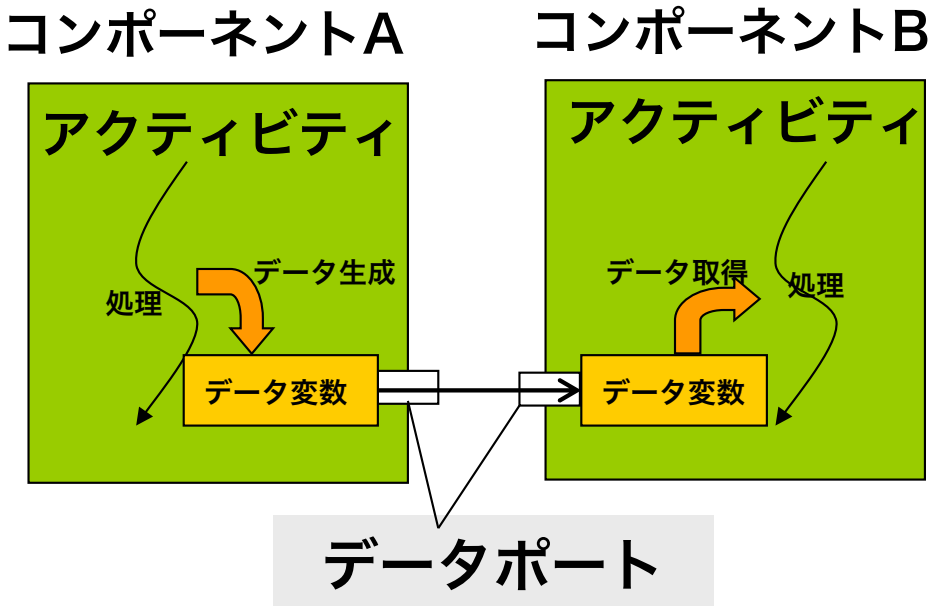
```
struct TimedShortSeq
 {
      Time tm;
      sequence<short> data;
 };
```

- 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

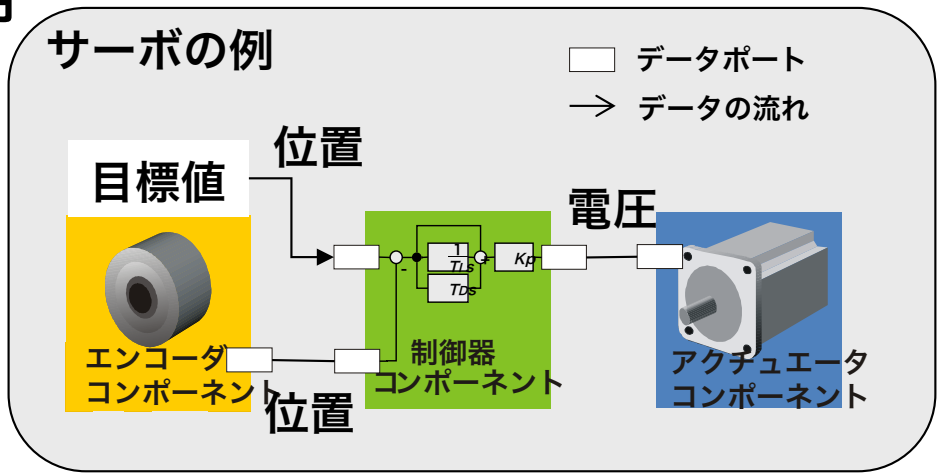
- データを入れるときにはあらかじめ長さをセットしなければならない。
- CORBAのシーケンス型そのもの
- 今後変更される可能性あり

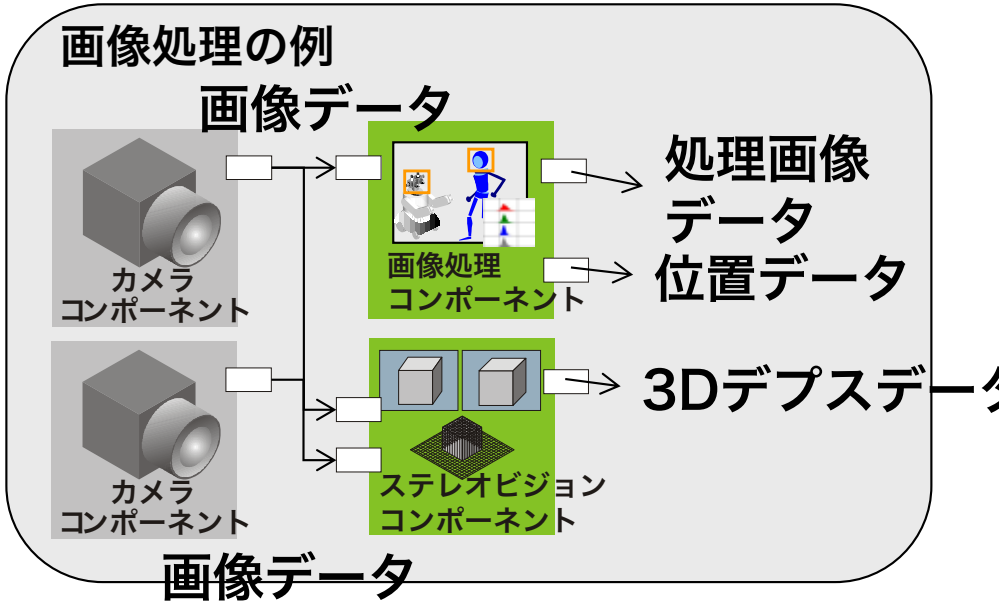
データポートの利用イメージ

- 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



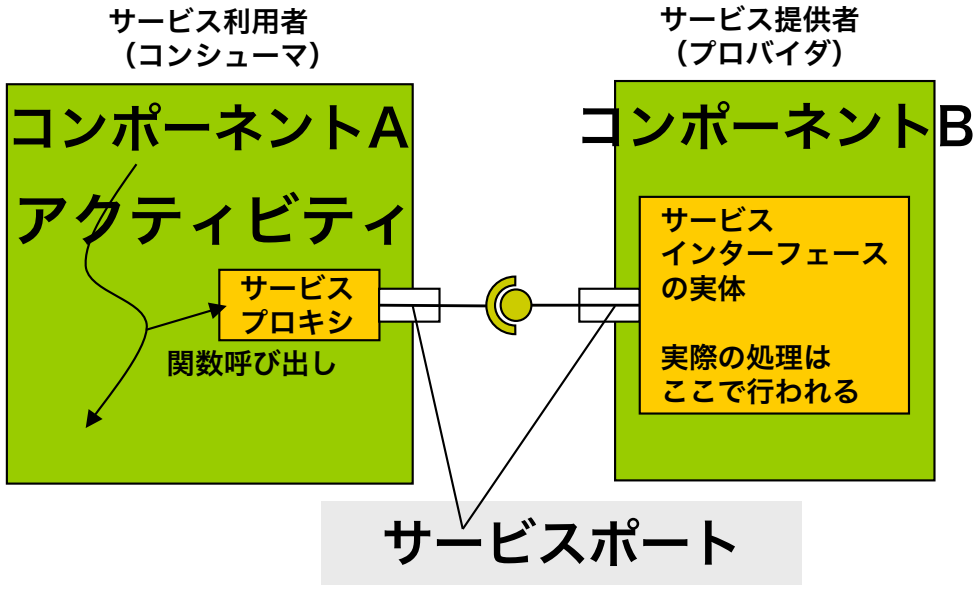
データが自動的に伝送される

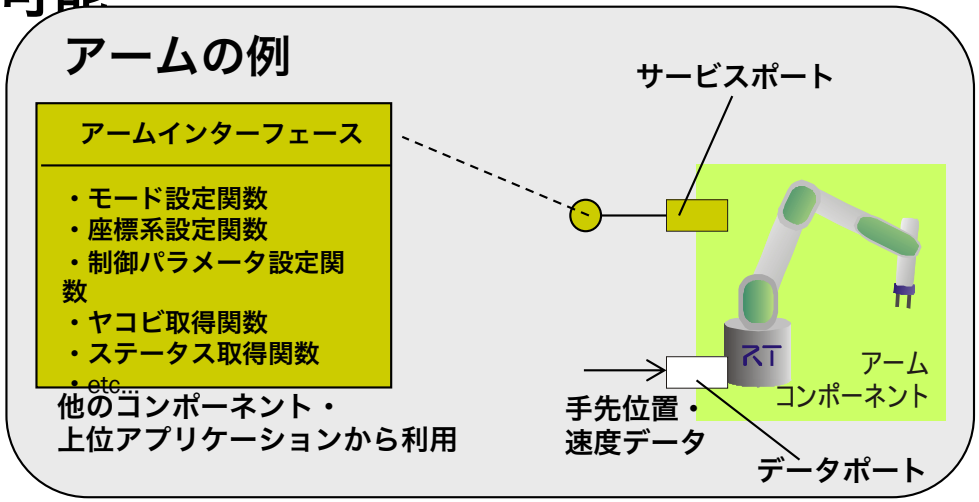


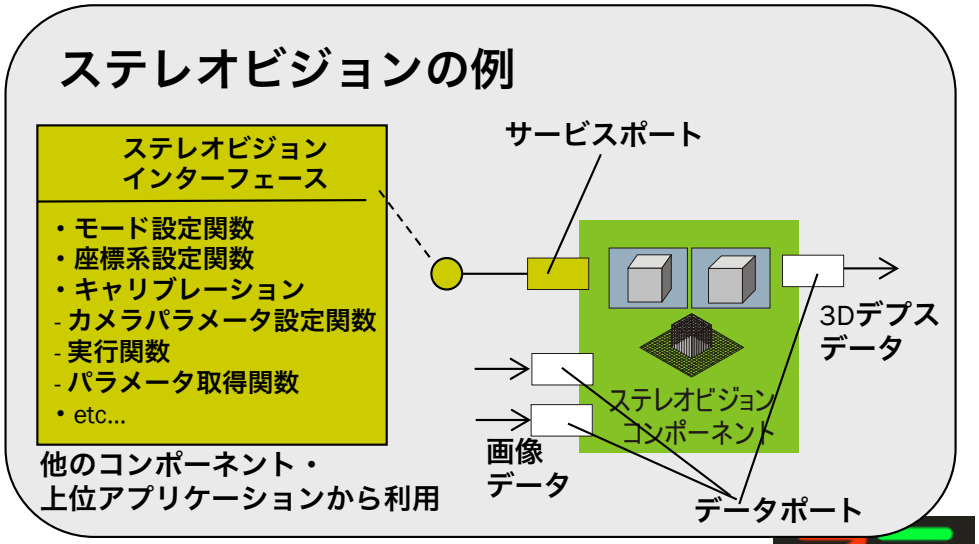


サービスポート

- 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...



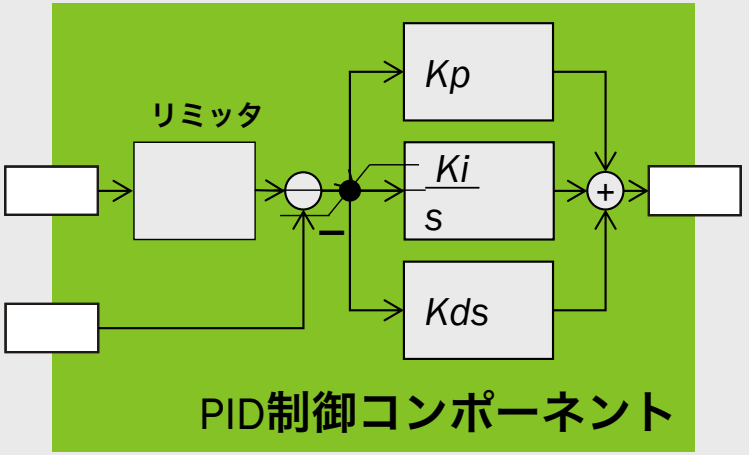




コンフィグレーション

- 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

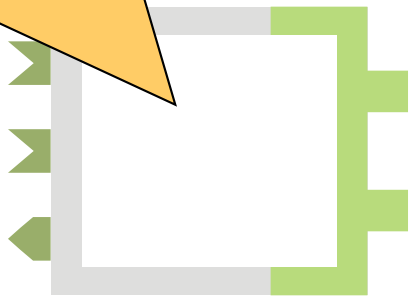
modeC	名前	Kp	Ki	Kd	Inmax	Inmin
	値	0.3	0.1	0.31	1.0	-1.0

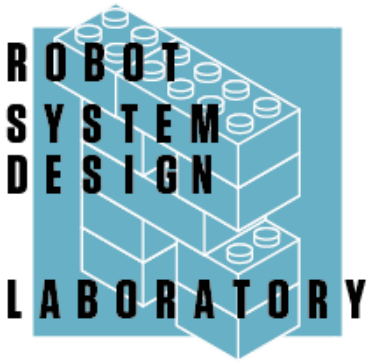
制御対象やモードに応じて複数のPIDゲインおよび入力リミッタ値を切り替えて使用することができる。
動作中の切り替えも可能。

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					

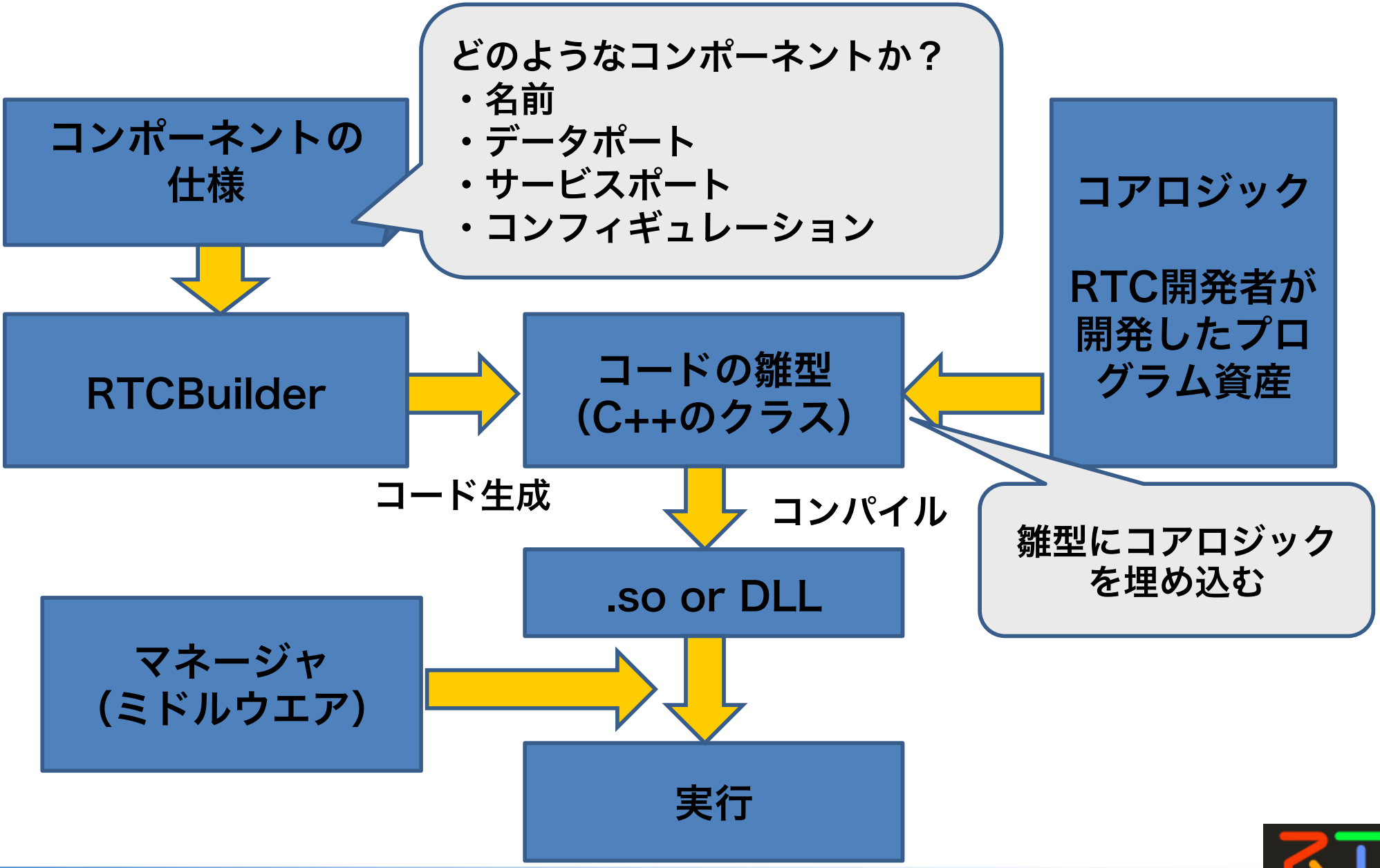
セット名	名前					
	値					



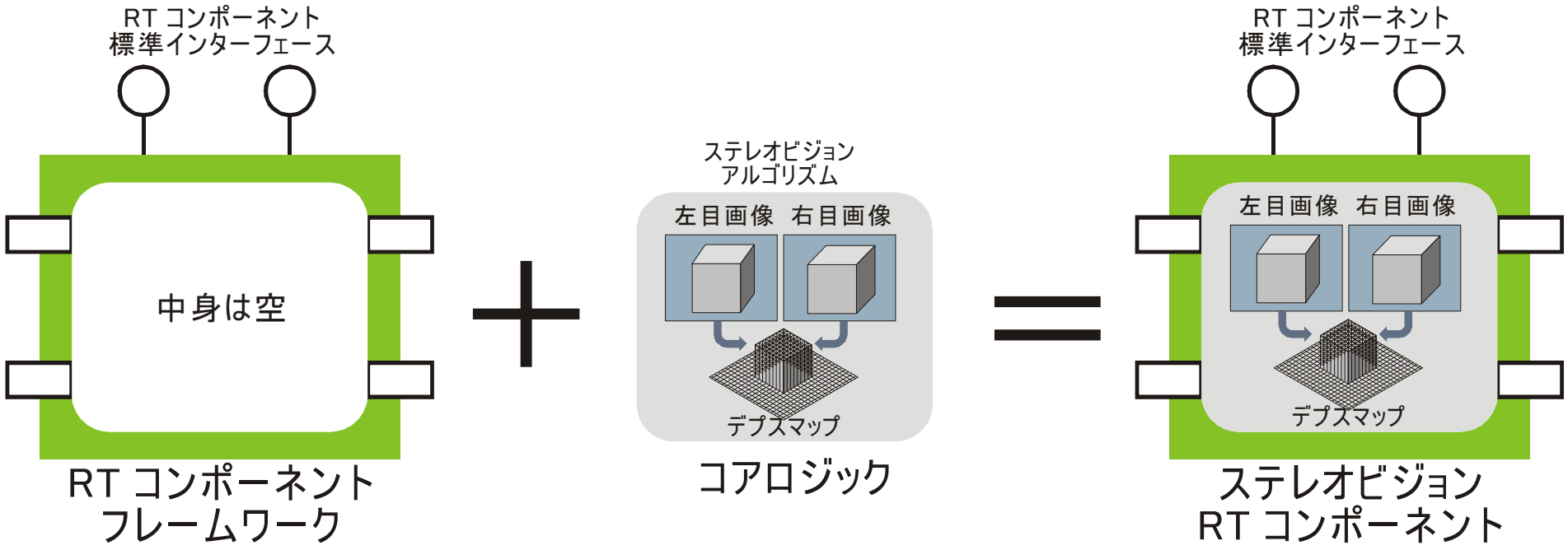


RTコンポーネントの開発・運用

RTC開発の流れ


どのようなコンポーネントか？
・名前
・データポート
・サービスポート
・コンフィギュレーション
コンポーネントの仕様
RTCBuilder
コード生成
コードの雛型
(C++のクラス)
コアロジック
RTC開発者が開発したプログラム資産
コンパイル
雛型にコアロジックを埋め込む
.so or DLL
マネージャ
(ミドルウエア)
実行

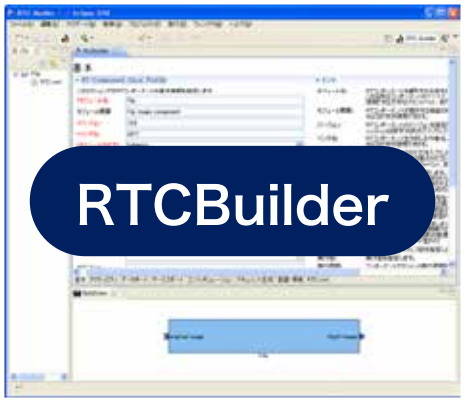
フレームワークとコアロジック



RTCフレームワーク＋コアロジック
＝RTコンポーネント

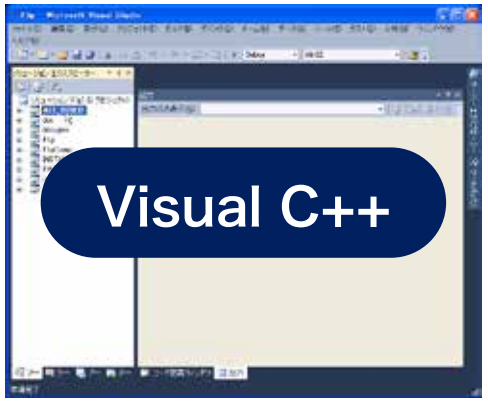
コンポーネントの作成・運用

Windowsの場合の開発イメージ



コンポーネントの
仕様の入力

VCのプロジェクト
ファイルの生成



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

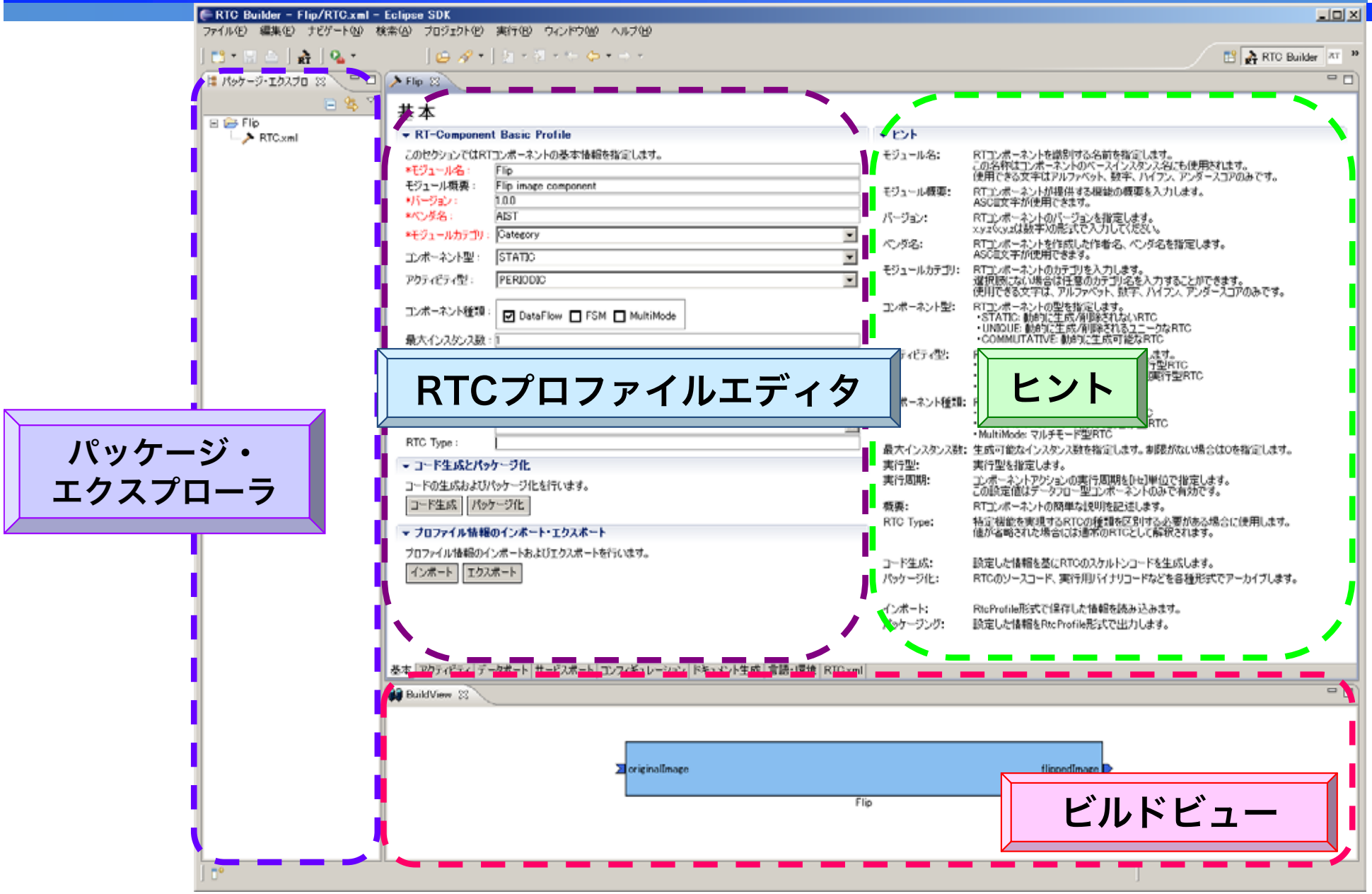
コンポーネント設計ツール

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



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た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！
・・・

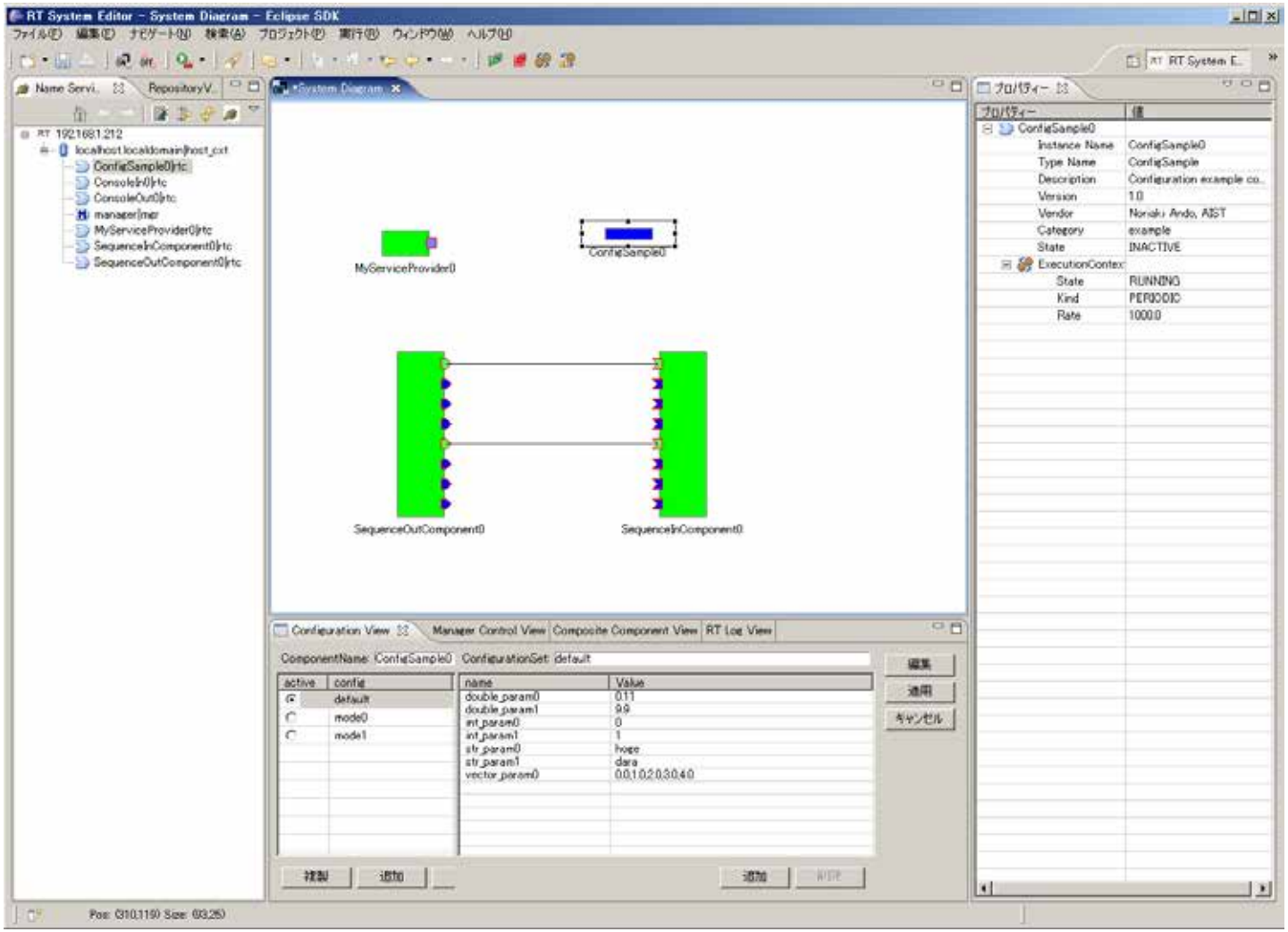
といった問題も解消しやすくなる．

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

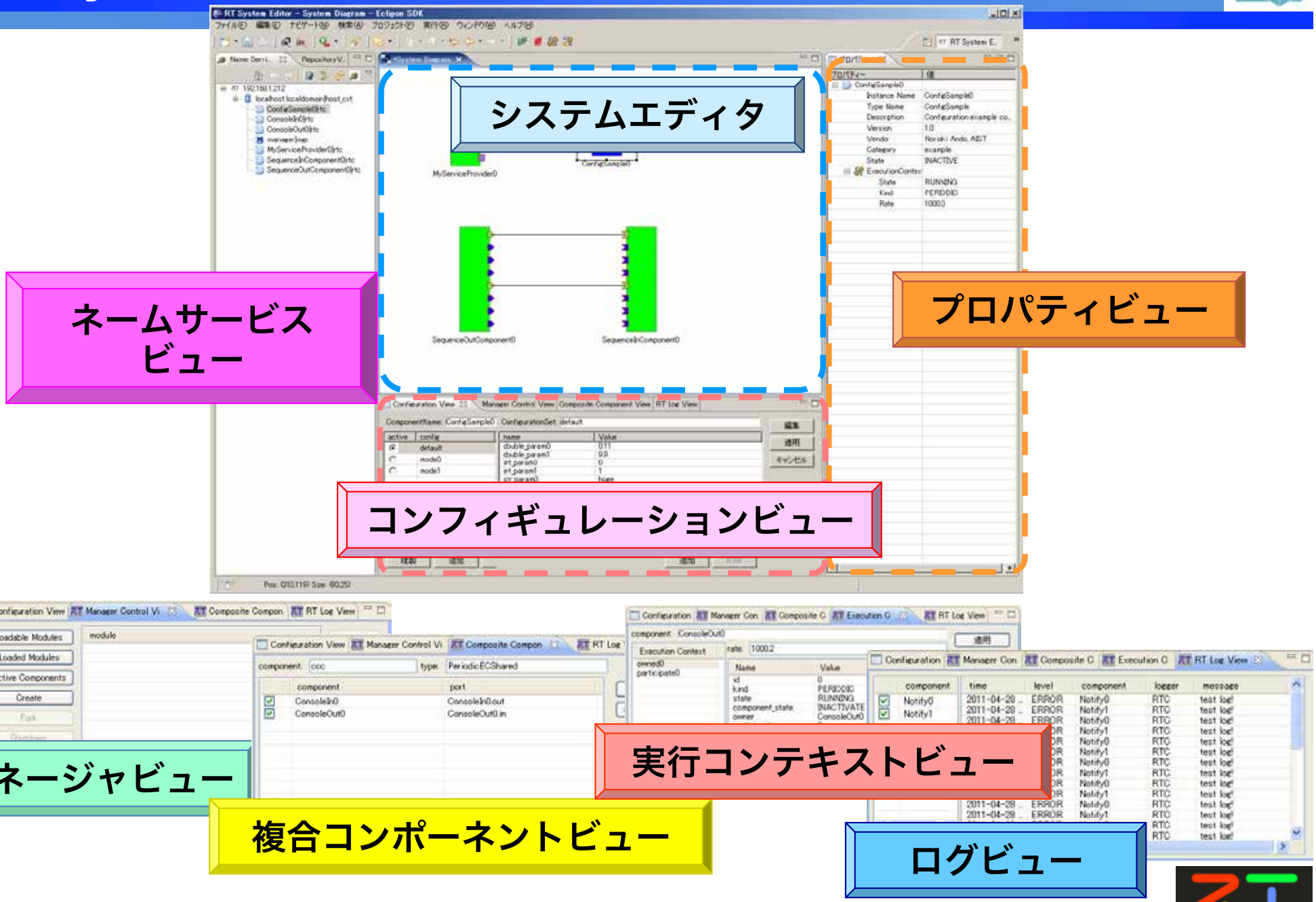
システム構築支援ツール

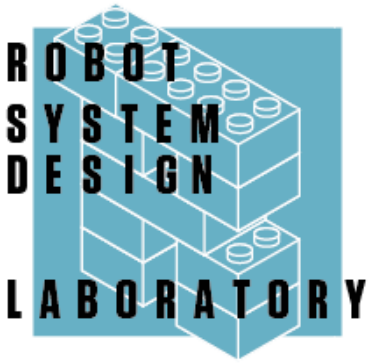
RT System Editor

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



RT System Editorの概観





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.

RTCベースのシステム構築手順例

1. 構築したいシステムの決定
2. コンポーネントを探す.
3. 要求と完全一致の場合，そのRTコンポーネントの要求ハードウェアをそろえる.

OpenRTC-aist(http://openrtc.org/)など,公開されているシステムパッケージに該当するハードウェア環境をそろえる

作業知能のパッケージ

移動知能のパッケージ

コミュニケーションのパッケージ

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を見つけることが出来る.

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

ロボット
アーム

カメラ
(手先)

カメラ

移動台車

音声入力用
マイク

マイク

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

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

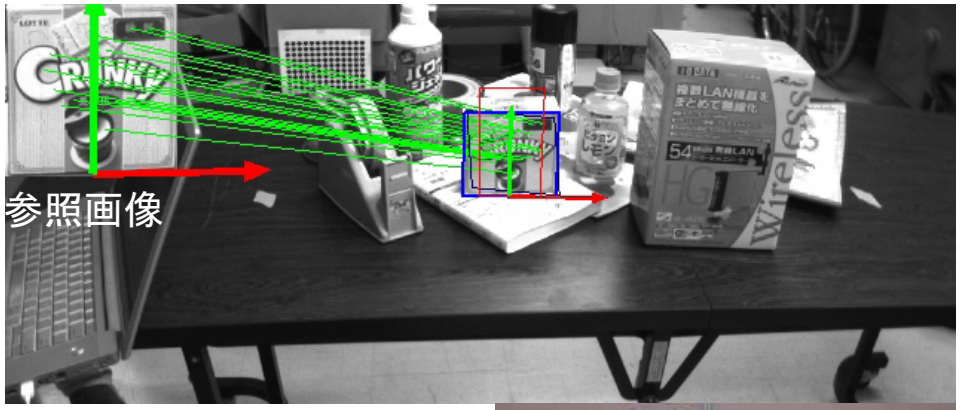
把持物体情報を獲得するRTC

アピア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:

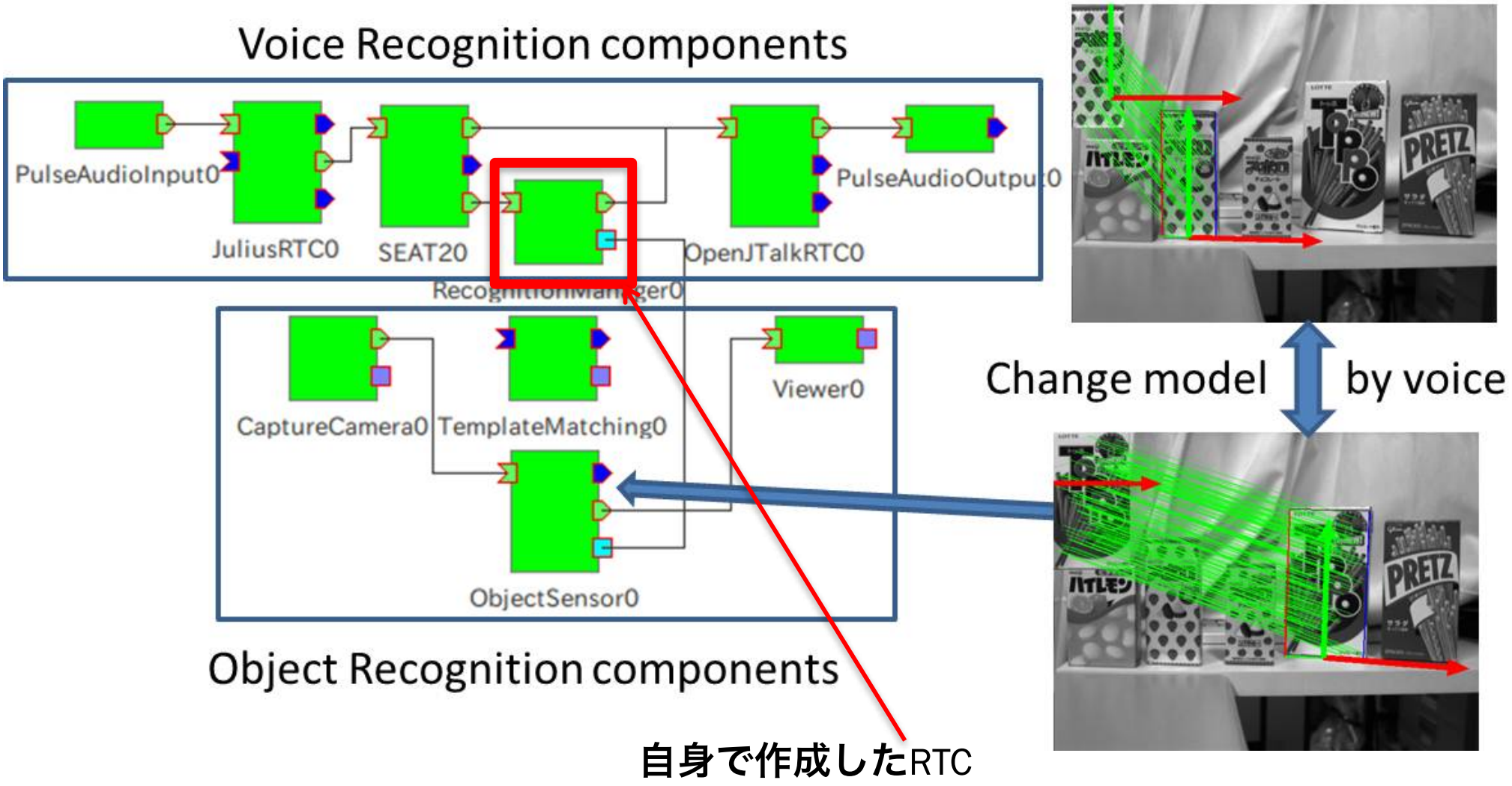
- ◆1枚の参照画像のみで、位置姿勢を推定できる
- ◆SIFT特徴によるロバストな推定
- ◆GPUによる高速化
- ◆ステレオカメラにも対応



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I/Fに対応したものを利用

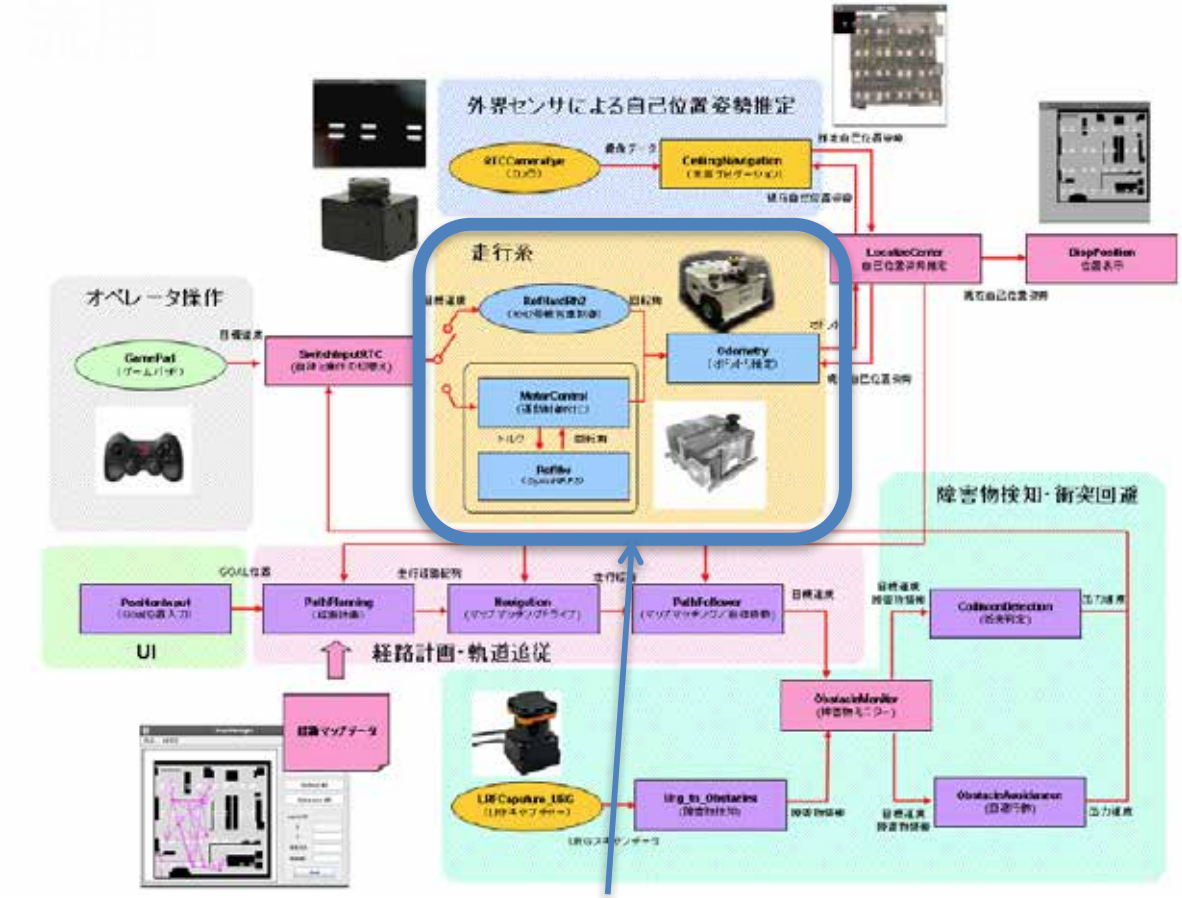
音声入力による物体探索システム



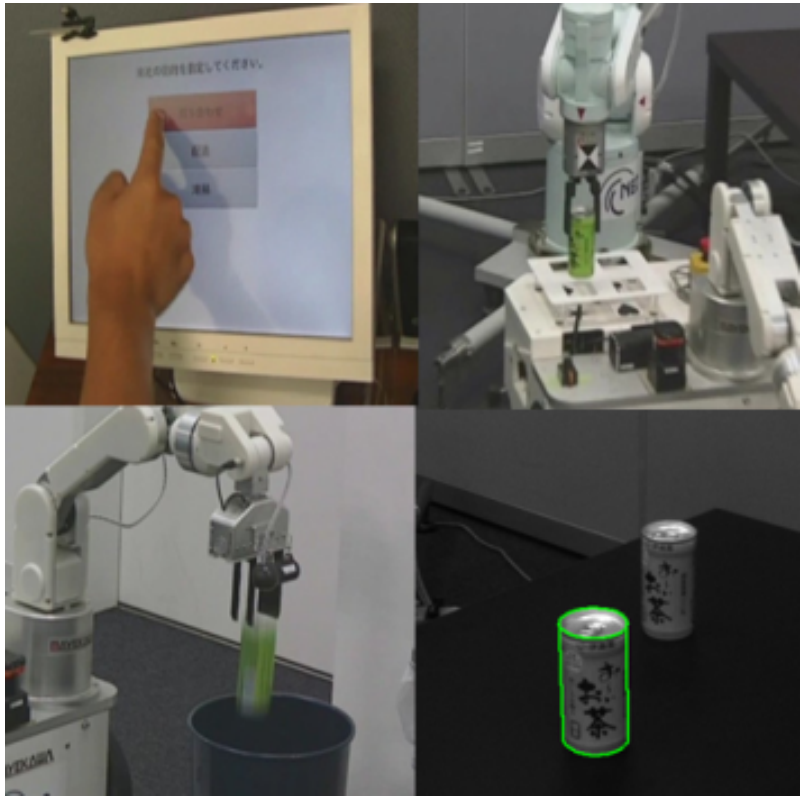
音声認識結果から，物体認識用のモデルデータに変換するコンポーネントのみ作成することで実現可能

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



走行系部分のみ利用

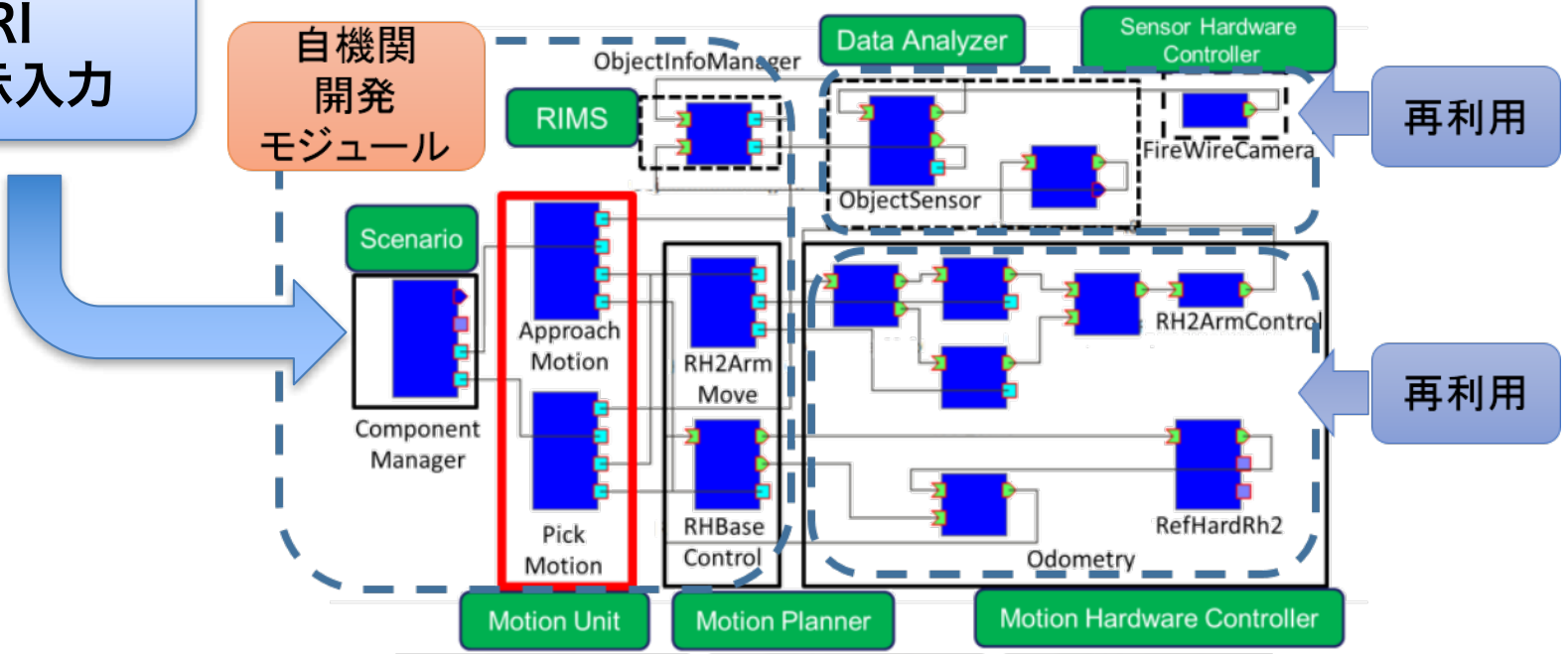


アーム制御部のみ利用

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

OpenHRI
を用いた指示入力



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

まとめ

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