



RTミドルウェアの現状と 今後の展望について

産業技術総合研究所
知能システム研究部門

神徳徹雄

<http://staff.aist.go.jp/t.kotoku/>



NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

Outline

1. 背景（技術戦略）
2. RTミドルウェア概要
3. 関連プロジェクト紹介
4. 開発のヒント
5. まとめ



本日の狙い：
RTミドルウェア技術のコンセプトを理解し、
皆でロボット技術の共有と再利用を考えて
いただくきっかけとする。



NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

技術背景

活発なロボット技術の研究開発

- 個々の技術課題の解決
- 事例紹介的で成果の共有が進んでいない

➡ 非効率な研究開発

システム開発の効率化

- 各組織の独自のアーキテクチャによるモジュール化

➡ ノウハウ化、競争力

これから より複雑なRTシステムの開発 異なるベンダーのシステムが連携して動作

技術の蓄積の上に新しい技術を構築される仕組み
標準化されたシステムインテグレーション技術の確立

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....)

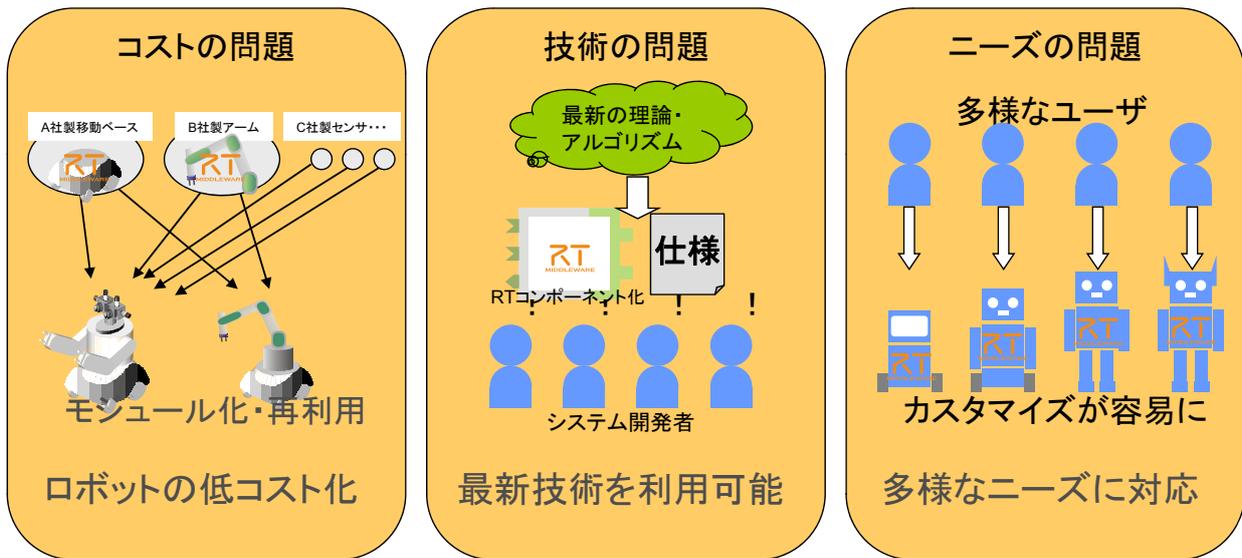
産総研版RTミドルウェア

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

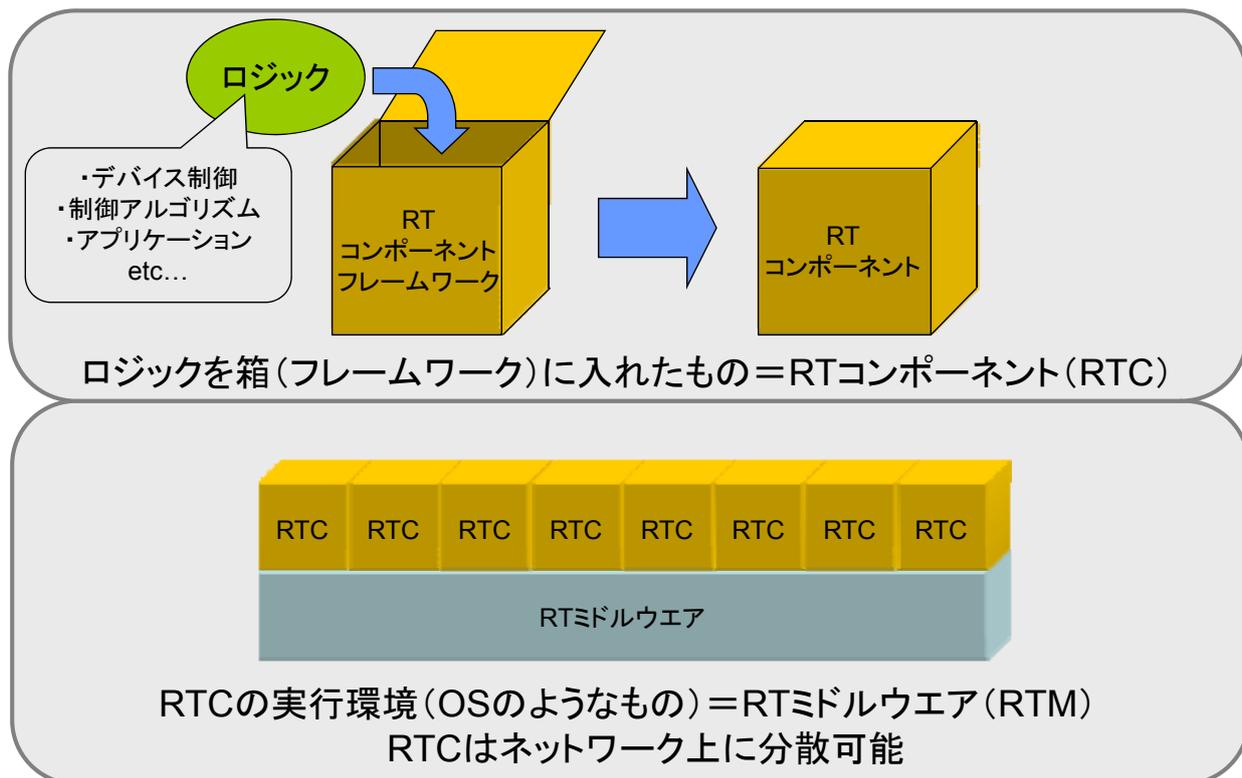
RTミドルウェアの目的

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

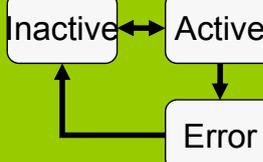
RTミドルウェアとRTコンポーネント



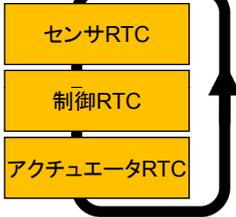
RTコンポーネントの主な機能

アクティビティ・実行コンテキスト

共通の状態遷移



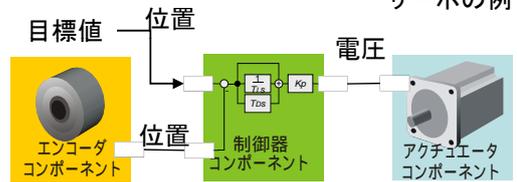
複合実行



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

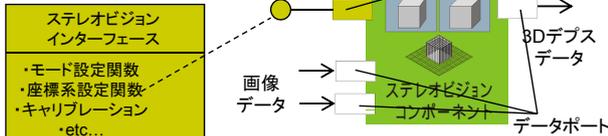
- データ指向ポート
- 連続的なデータの送受信
- 動的な接続・切断



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...

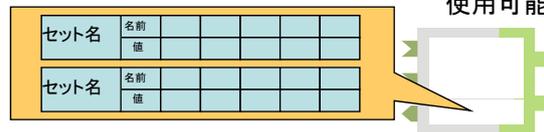


サービス指向相互作用機能

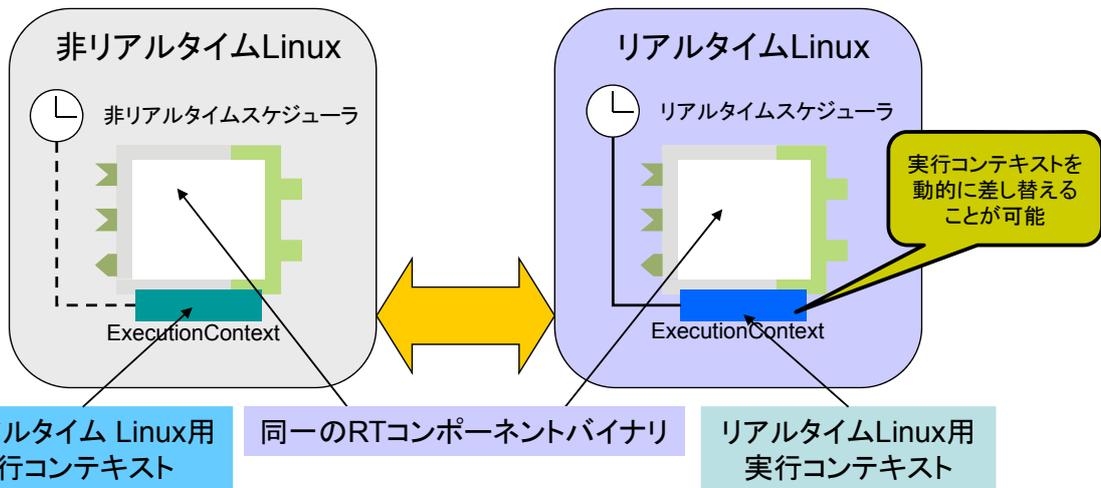
コンフィギュレーション

- 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

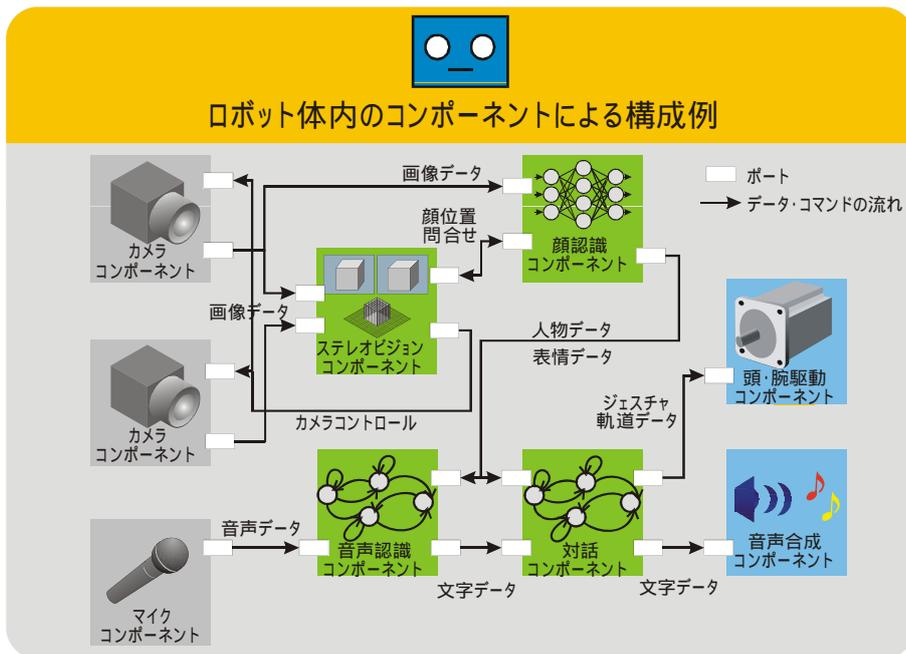


リアルタイム実行コンテキスト



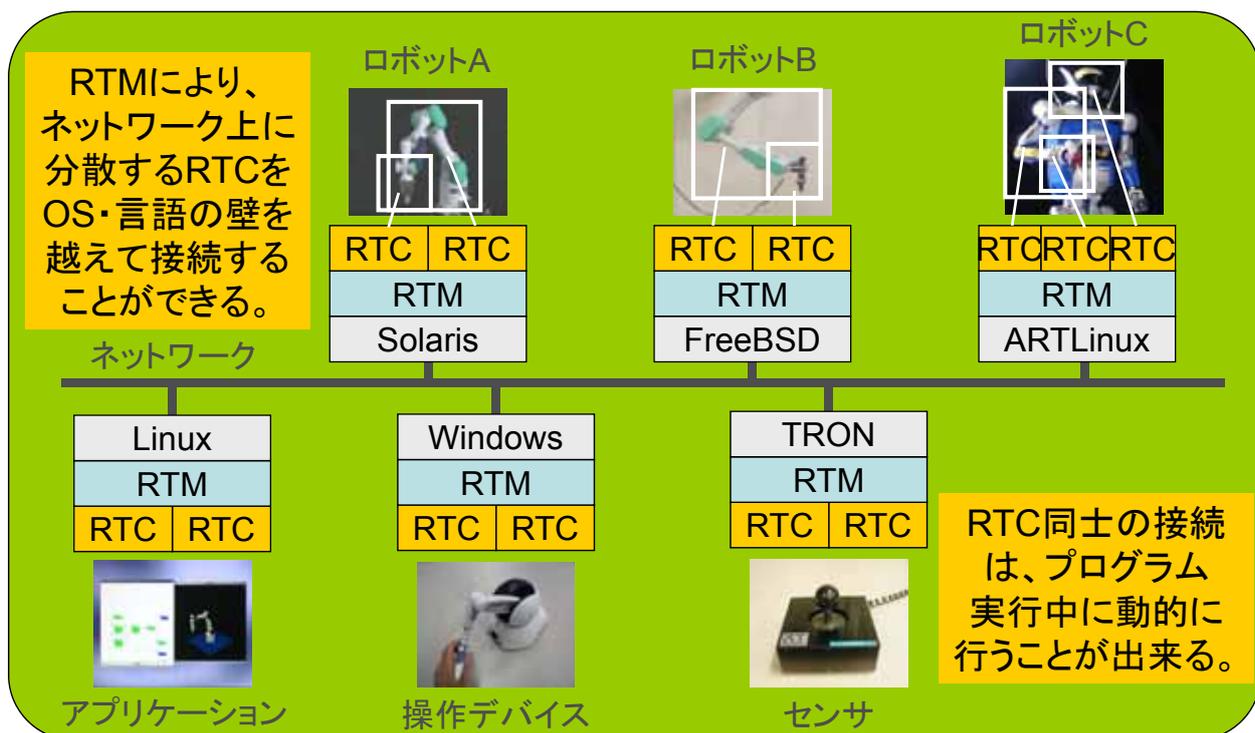
非リアルタイムLinux環境で作られたRTコンポーネントを再コンパイルせずにリアルタイムLinux上でリアルタイム実行可能

RTCの分割と連携



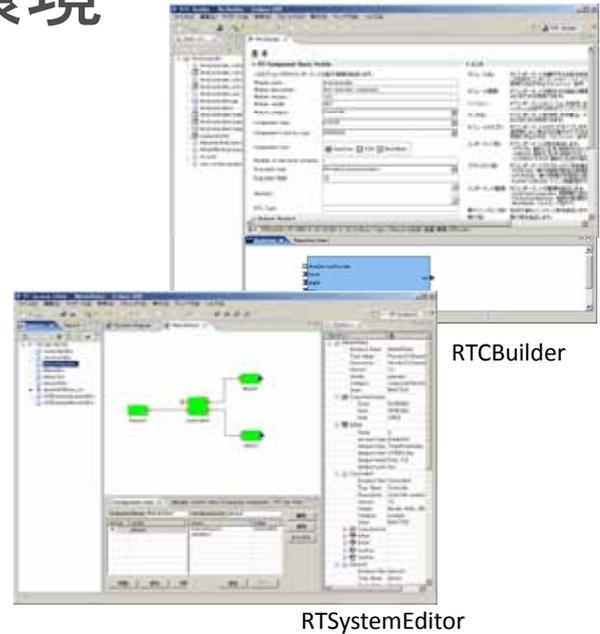
(モジュール)情報の隠蔽と公開のルールが重要

RTミドルウェアによる分散システム



開発環境

- RTCBuilder (GUI版)
- rtc-template (CUI)
 - RTコンポーネントのコードジェネレータ
 - GUI画面で必要事項を入力
 - C++, Python, Java, C#等のコードを自動生成
- RTSystemEditor
 - ネットワーク上のすべてのコンポーネントの操作が可能
 - コンポーネントのON/OFF、パラメータの変更、状態監視
 - コンポーネント間の接続



RTCBuilder

RTSystemEditor

RTC・RTM統合開発環境の整備
 RTC設計・実装・デバッグ、RTMによるインテグレーション・デバッグまでを一貫して行うことができる統合開発環境をEclipse上に構築

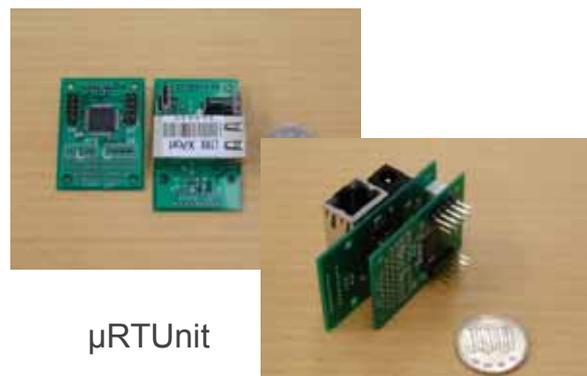
RTUnit(2003年～) /RTC-Lite(2005年～)

- ネットワーク型マイコン機器
 - small/ μ RTUnitをRTMで統合

RTUnit 仕様	
MPU	Microchip PIC16F877A
ROM	8kwords
RAM	368 bytes
EEPROM	256 bytes
クロック	max 20MHz
A/D	10bit × 8ch
DIO	24ch
シリアル	2ch
通信	LANTRONIX XPort
電源	DC 5V



small RTUnit

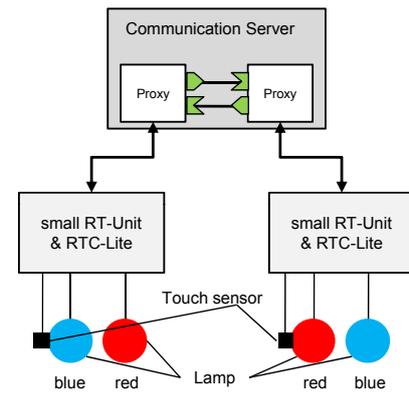


μ RTUnit

RTC-Liteの応用

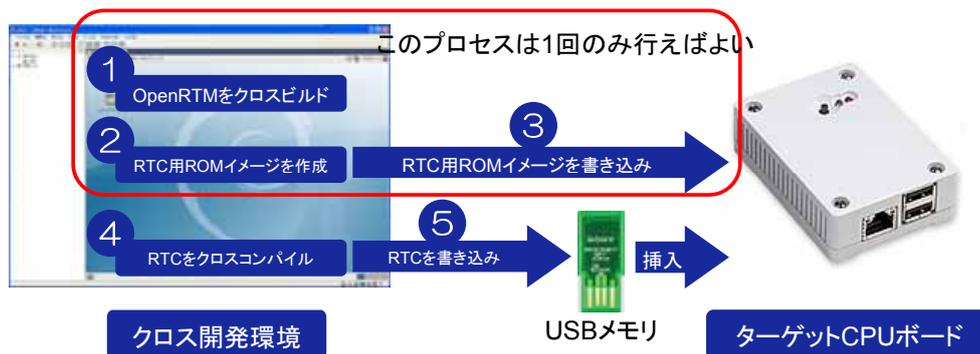


メディアアート作品への適用



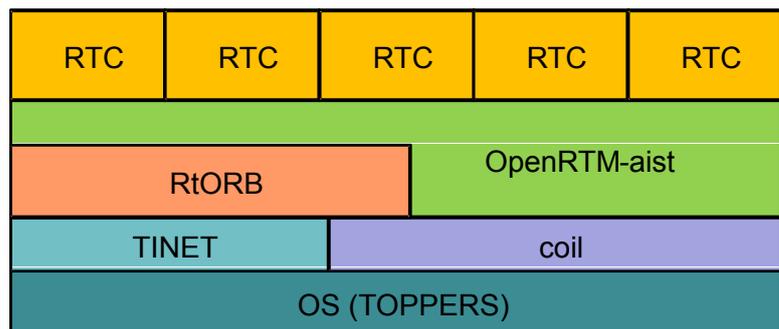
移動プラットフォームへの適用

組み込みLinux版OpenRTM



TOPPERS版OpenRTM

- TOPPERS
 - uITRON互換OS
 - オープンソース
 - 製品搭載実績多数
- RtORB
 - 産総研版CORBA
 - 軽量・高速
- coil
 - OS抽象化層



NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

OpenRTMの利点

- 共通コンポーネントフレームワークを提供
 - OMG標準
 - コールバックベースの枠組み、共通状態マシン、複合化に対応
 - 大部分のコード生成を自動化
- 多言語対応
 - C++, Java, Python, .NET (by SEC)
- 多様なOSへのネイティブ対応
 - FreeBSD, Linux, Mac OS X, Windows
 - 試験的: TOPPERS, T-Kernel, VxWorks
- ツールの提供
 - Eclipseベースのツール群 (RTCB, RTSE)
 - コマンドラインツール群 (rtchell)
- デュアルライセンス (EPLと個別ライセンス)
 - RTCにはライセンスが及ばない(RTCのバイナリ供給が可能に)
 - 商用化、事業化、組込み用途には個別ライセンスで対応

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

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

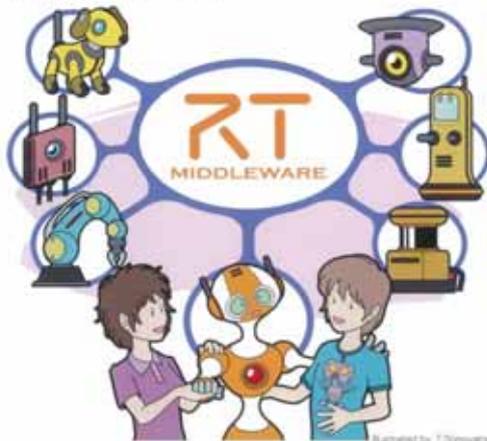


- UMLとRTミドルウェアによるモデルベースロボットシステム開発
- 水川 真, 大原 賢一, 坂本 武志 著

書籍



【特集】
使えるRTミドルウェア



RSJ 社団法人 日本ロボット学会

「使えるRTミドルウェア」特集号
日本ロボット学会論文誌 vol.28, no.5

再利用性の高いロボットシステムを
RTミドルウェアを用いてどう構築する
かという“方法論”に関する実践
的な特集号

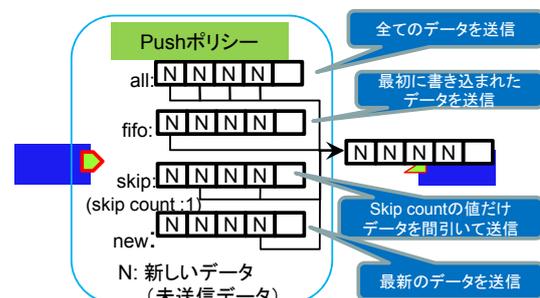
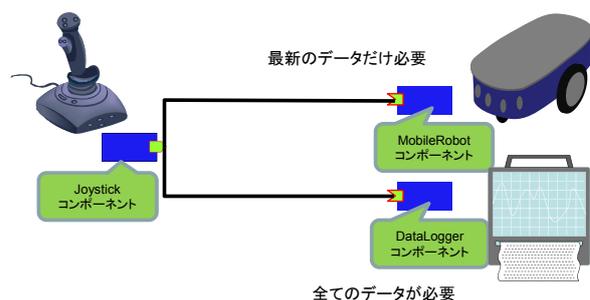
http://www.openrtm.org/OpenRTM-aist/html/JRSJ2010_28_5.html

OpenRTM-aist-1.0.0の新機能

- データポートの高機能化
- サービスポートの拡張
- RTCマネージャの提供
- OSサポートの強化
- Coilの導入
- ツールのユーザビリティの向上

データポートの高機能化

- データフロー制御機能
 - 多様なデータの利用の仕方に対応
- シリアライズ機構の導入
 - データポート内で直列化
 - 多様な通信チャネルに対応
 - Rawソケット、共有メモリ
- バッファリングポリシーの導入
 - バッファフル状態等の検出

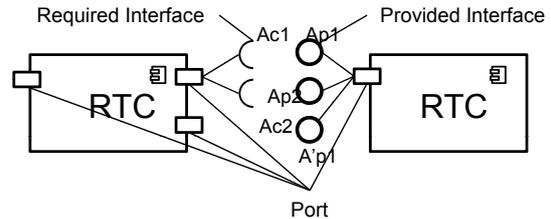


詳細は

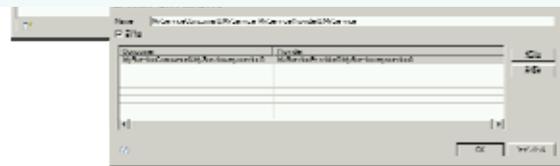
次世代ロボット智能化技術開発プロジェクト
2A1-G03: 9:30-11:00

サービスポートの拡張

- 多くのインターフェースが関連する複雑な接続
 - 実行時に接続インターフェースペアを指定可能
- RTSystemEditor
 - 接続関係を詳細に指定可能なダイアログを追加



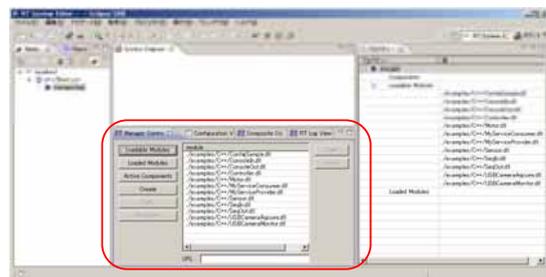
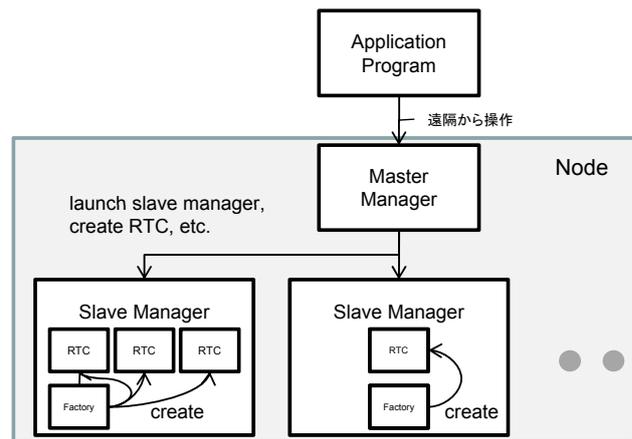
ダイアログで接続関係を指定



詳細は
次世代ロボット智能化技術開発プロジェクト
2A1-G01: 9:30-11:00

RTCマネージャの導入

- RTシステムライフサイクル管理のためのプログラム
- RTCを遠隔から起動、停止、削除可能
- ツールのサポート
 - RTSystemEditor
 - rtcshell



マネージャコントロールビューの追加

詳細は
次世代ロボット智能化技術開発プロジェクト
2A1-G02: 9:30-11:00

OSサポート

- Windowsインストーラ
 - omniORB同梱
 - RTSystemEditor (RCP版)を同梱
 - VC再配布ライブラリ同梱
- Mac OS Xサポート
 - 公式サポート
 - MacPortsのportsファイルを提供
 - RTSE等の全部入りパッケージを提供



今後の展望(1)

- OSサポートの追加
 - TOPPERSを公式サポート予定
- RTCマネージャ機能
 - すべてのRTCをシステム構成ファイルから自動起動
 - オフラインエディタの活用
 - 名前解決方法の追加:ブロードキャスト、URL
- RTコンポーネント機能
 - Raw TCPデータポート
 - 共有メモリ型データポート

今後の展望(2)

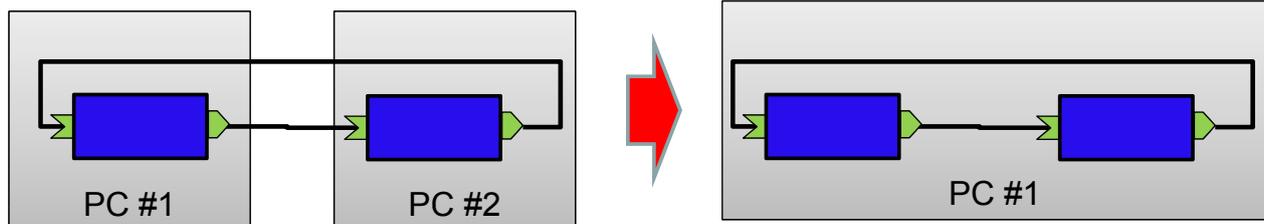
- 新Webページ
 - ドキュメント
 - サンプルの追加
 - ユーザプロジェクトページの追加
 - コンポーネント一覧
- リポジトリ
 - ツール自動更新機能の追加



テクニック(資源を考慮)

実時間(リアルタイム)処理

現状ではネットワークを介したループは無理。
時間にクリティカルな処理は必要なRTCを同じ計算機に乗せる。



RTミドルウェアは魔法ではない

テクニック(資源を考慮)

計算資源(計算機の負荷)

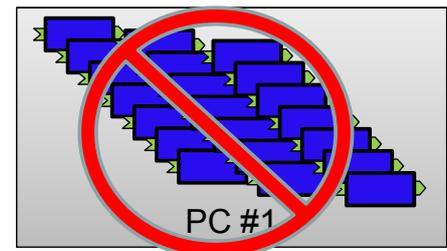
不必要に実行周期が早くないか？ 計算機
の能力を超えた数のRTCを載せていないか？

rtc.conf

```
exec_cxt.periodic.rate:10
実行コンテキストの周波数[Hz]を指定。
::有効範囲:(0, 1000000].
::デフォルト:1000.
```

- センサ出力が80msec毎なのに1msecでループを無駄に回していないか？
- OSのプロセス切り替え周期設定が10msecなのに10KHzで回そうとしていないか？

数多くの画像処理RTC



Pentium3-300MHz

RTミドルウェアは魔法ではない

テクニック(接続タイプ)

処理に合わせたデータフロー選択

- Flush (同期送信)
書き込みが終了するまで呼び出したコンポーネントの実行が待たされる。ネットワークを介した接続に使うと性能低下。
- New (非同期送信)
書き込みは異なるスレッドで実施。push ポリシー (送信バッファ制御) をサポート
- Periodic (周期送信)
事前に設定された周期の別スレッドとして送信を実行。Pushポリシー (送信バッファ制御) をサポート。

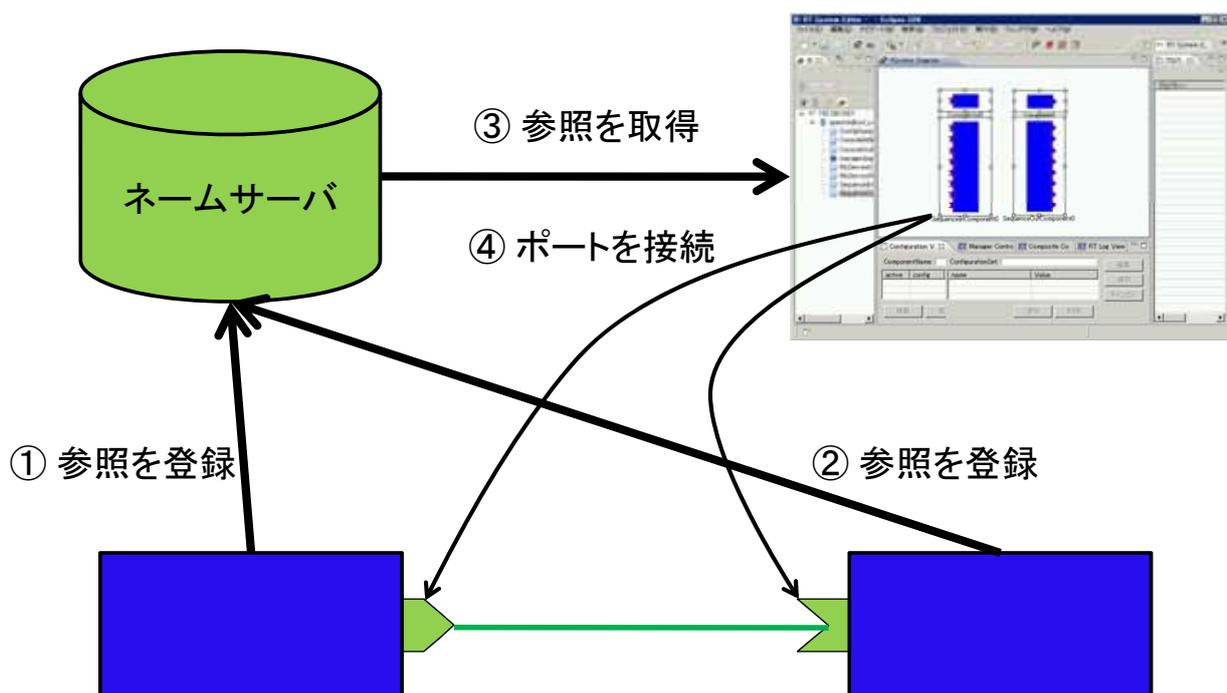
テクニック(接続タイプ)



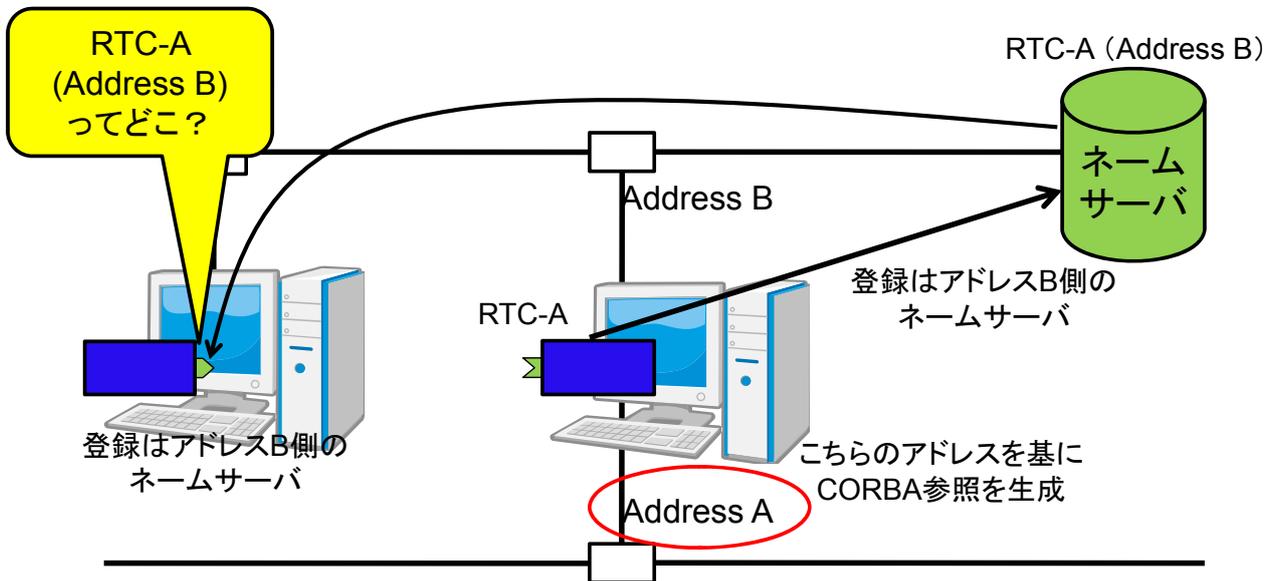
push ポリシー (送信バッファ制御)

- ALL: バッファに入っているすべてのデータを送信
- FIFO: バッファに入っている一番古いデータをひとつ送信
- SKIP: 特定の数だけ飛ばしてバッファのデータを送信
- NEW: 過去のデータを無視して最新データをひとつ送信

動作シーケンス



ネットワークインターフェースが 2つある場合の注意



NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

テクニック(通信路指定)

通信経路が複数ある場合

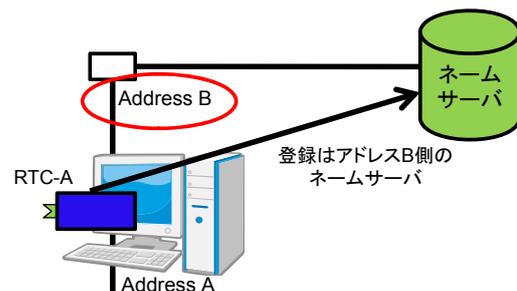
複数のネットワークインターフェースを持つ場合
(Ethernetの他に無線やVirtual Machineの通信経路がアクティブな場合)、ネームサーバに間違っただ情報が登録される場合がある。



rtc.conf

```
corba.endpoints: 192.168.0.12:
(最後のコロンを忘れずに!)
```

- 正しい経路のネットワークインターフェース情報を設定する



通信路を確認

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

テクニック(ユーザ名)

日本語のユーザ名(ログインアカウント)

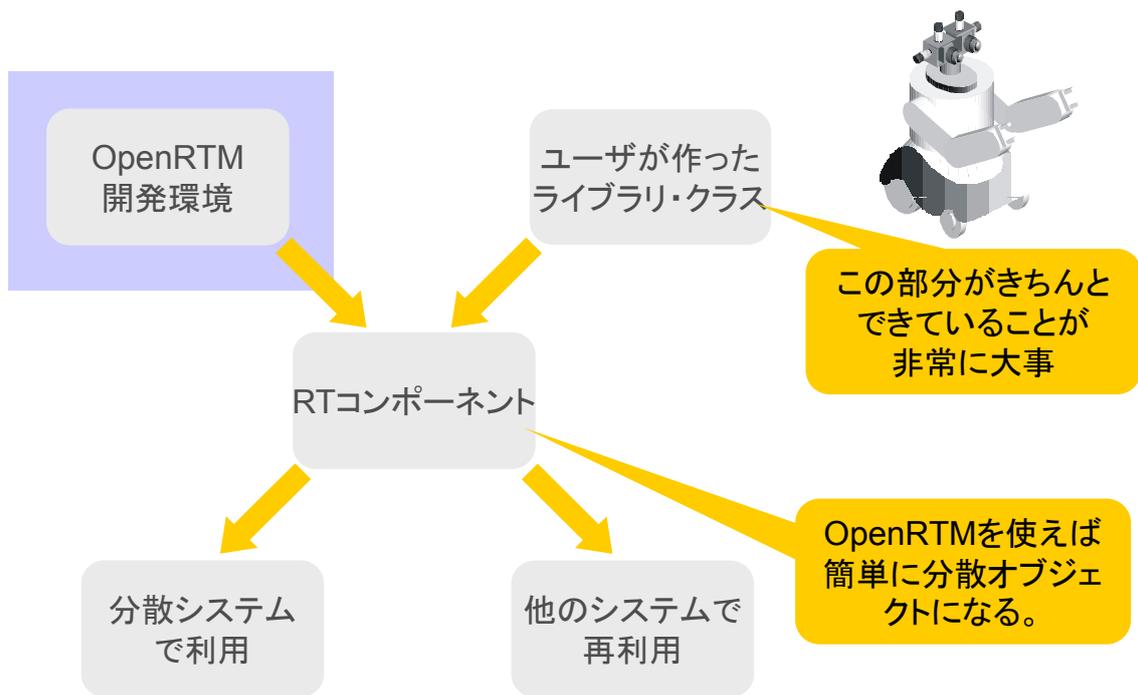


現在のところ、日本語のユーザ名を使うと通信に利用している分散ミドルウェア(OmniORB)が正常に動作しない。

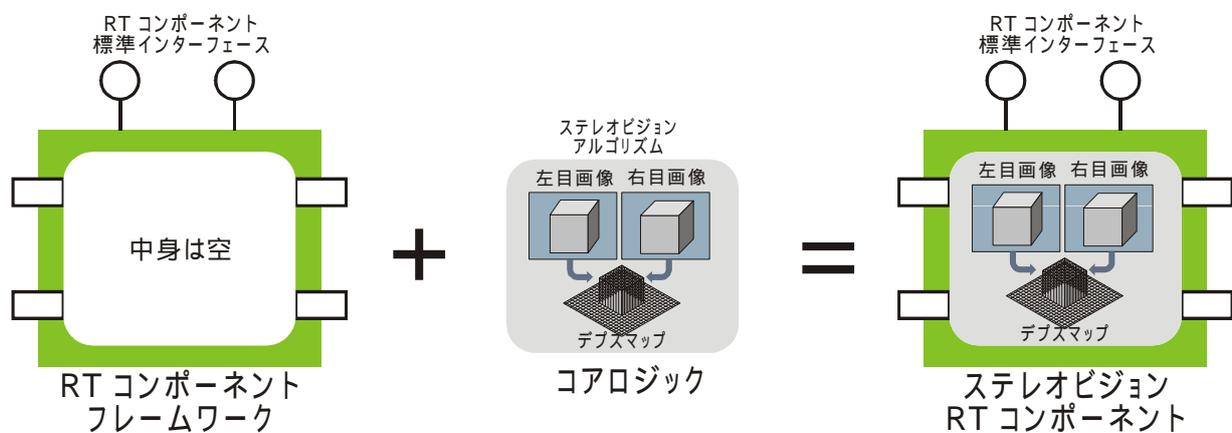
ユーザ名に日本語を使わない

RTコンポーネントプログラミング

OpenRTMを使った開発の流れ



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

モデルに基づくコード生成

コンポーネント仕様

```

name: MyComp
category: temp.sensor device
description: temp. sensor RTC
comp_type: STATIC
act_type: PERIODIC
InPorts: mode:TimedBool
OutPorts: temp: TimedDouble
    
```

同一のRTC仕様からは
言語が異なっても、
同じ(コンポーネントモデルの)RTCが生成される

Template code generator

C++
backend

Java
backend

Python
backend

RTC-Lite
backend

RTC source for C++

```

class MyComp
: public DataflowComponent {
public:
    virtual onExecute(ec_id);
private:
    TimedBool m_mode;
    TimedDouble m_temp;
};
    
```

RTC source for Java

```

import RTC.DataFlowComponent;
public class MyCompImpl
extends DataFlowComponent
{
    public ConsoleImpl(mgr)
    {
    }
};
    
```

RTC source for Python

```

#!/usr/bin/env python
import RTC
class MyComp(
    DataFlowComponent):
    def __init__(self, manager):
        :
    def onExecute(self,
        :
        :
    ):
        :
    ):
    
```

RTC-Lite source for PIC C

```

#include <16f877a.h>
#include "rtc_base.c"

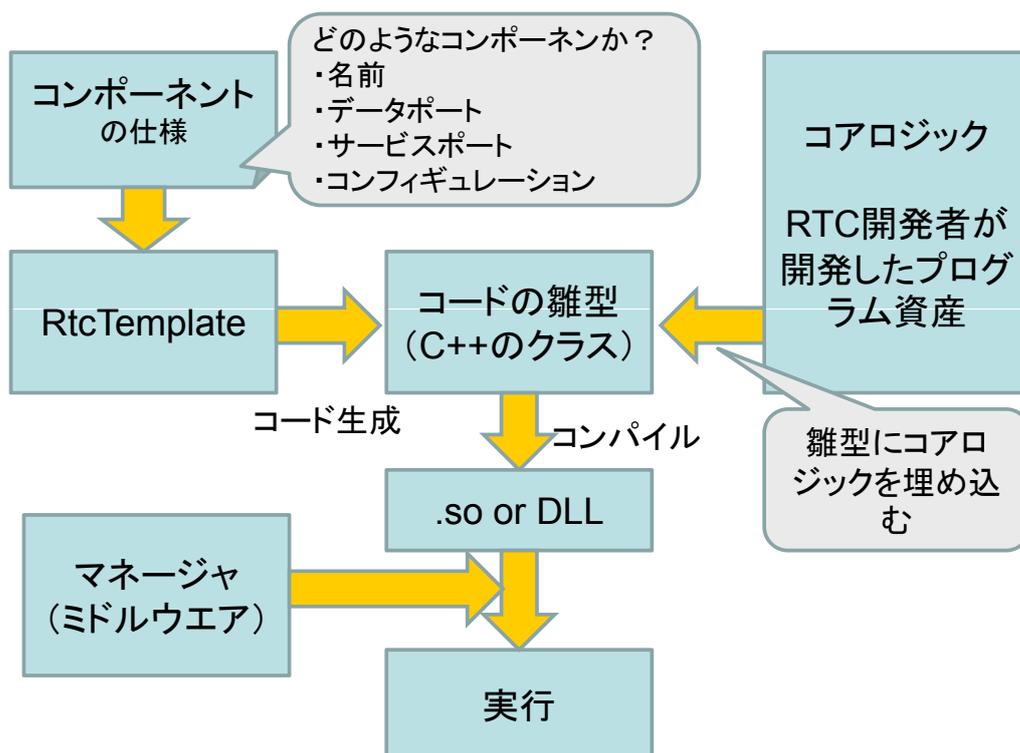
int main(void)
{
    rtc_connect_proxy();
    rtc_mainloop();
    return 0;
}
    
```

RTC-Lite proxy code

```

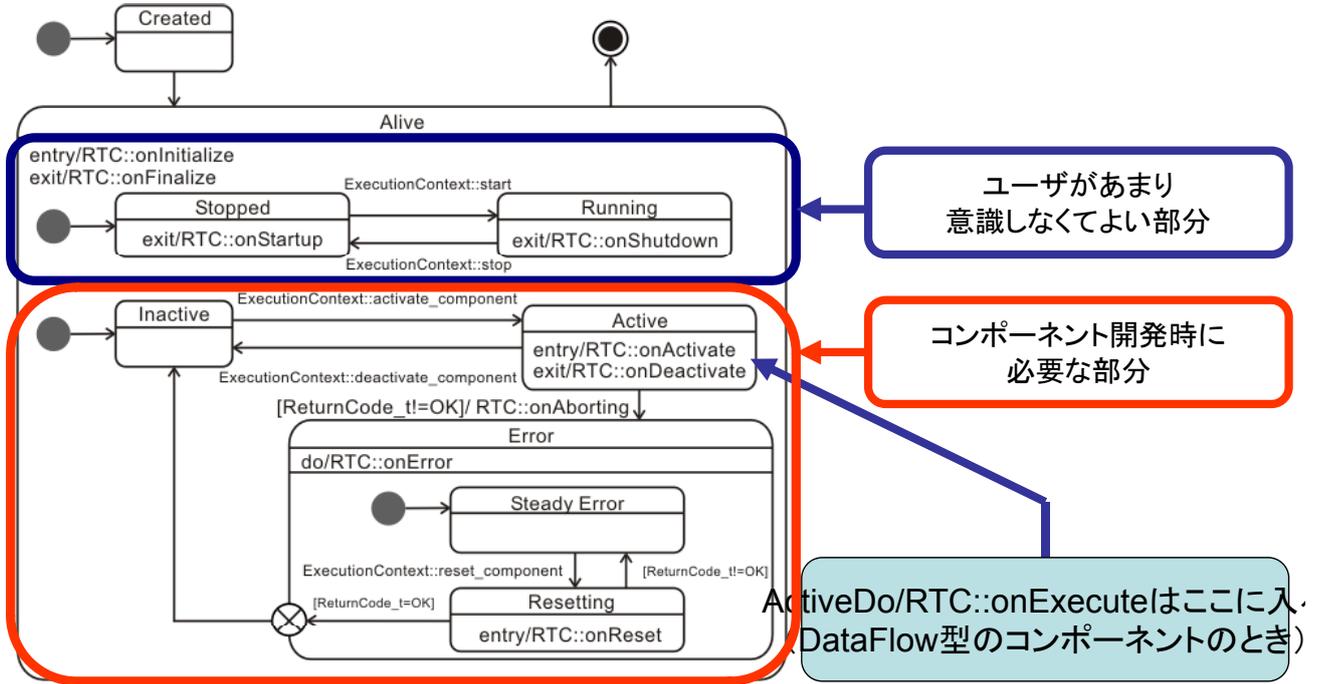
#!/usr/bin/env python
import RTC
class Proxy(
    DataFlowComponent):
    def __init__(self, manager):
        :
    def onExecute(self, ec_id):
        :
    ):
    
```

OpenRTMを使った開発の流れ



コンポーネント内の状態遷移

0.2.0の状態遷移から変更



コールバック関数

RTCの作成 = コールバック関数に処理を埋め込む

コールバック関数	処理
onInitialize	初期化処理
onActivated	アクティブ化される時1度だけ呼ばれる
onExecute	アクティブ状態時に周期的に呼ばれる
onDeactivated	非アクティブ化される時1度だけ呼ばれる
onAborting	ERROR状態に入る前に1度だけ呼ばれる
onReset	resetされる時に1度だけ呼ばれる
onError	ERROR状態のときに周期的に呼ばれる
onFinalize	終了時に1度だけ呼ばれる
onStateUpdate	onExecuteの後毎回呼ばれる
onRateChanged	ExecutionContextのrateが変更されたとき1度だけ呼ばれる
onStartup	ExecutionContextが実行を開始するとき1度だけ呼ばれる
onShutdown	ExecutionContextが実行を停止するとき1度だけ呼ばれる

とりあえずはこの5つの関数を押さえておけばOK

テクニック(初期化処理)



- onInitialize:
コンポーネントを使うのに必須となる資源の確保(チェック)など、後から再度初期化することの必要のない処理
- onActivated:
コンポーネントを立ち上げてから、パラメタを変更して初期化する必要がある処理

ひとことで初期化と言っても...

Rtc.confについて

RT Component起動時の登録先NamingServiceや、登録情報などについて記述するファイル

記述例:

corba.nameservers: localhost:9876

naming.formats: SimpleComponent/%n.rtc

(詳細な記述方法は etc/rtc.conf.sample を参照)

以下のようにすると、コンポーネント起動時に読み込まれる

```
./ConsoleInComp -f rtc.conf
```

ネーミングサービス設定

corba.nameservers	host_name:port_numberで指定、デフォルトポートは2809(omniORBのデフォルト)、複数指定可能
naming.formats	%h.host_cxt/%n.rtc →host.host_cxt/MyComp.rtc 複数指定可能、0.2.0互換にしたければ、 %h.host_cxt/%M.mgr_cxt/%c.cat_cxt/%m.mod_cxt/%n.rtc
naming.update.enable	“YES” or “NO”: ネーミングサービスへの登録の自動アップデート。コンポーネント起動後にネームサービスが起動したときに、再度名前を登録する。
naming.update.interval	アップデートの周期[s]。デフォルトは10秒。
timer.enable	“YES” or “NO”: マネージャタイマ有効・無効。 naming.updateを使用するには有効でなければならない
timer.tick	タイマの分解能[s]。デフォルトは100ms。



必須の項目



必須でないOption設定

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

その他

corba.endpoints	IP_Addr:Port で指定 :NICが複数あるとき、ORBをどちらでlistenさせるかを指定。Portを指定しない場合でも”:”が必要。 例 “corba.endpoints: 192.168.0.12:” NICが2つある場合必ず指定。 (指定しなくても偶然正常に動作することもあるが、 <small>念のため</small> のため。)
corba.args	CORBAに対する引数。詳細はomniORBのマニュアル参照。
[カテゴリ名]. [コンポーネント名]. config_file または [カテゴリ名]. [インスタンス名]. config_file	コンポーネントの設定ファイル <ul style="list-style-type: none"> •カテゴリ名 : manipulator, •コンポーネント名 : myarm, •インスタンス名 myarm0,1,2,... の場合 manipulator.myarm.config_file: arm.conf manipulator.myarm0.config.file: arm0.conf のように指定可能



必須の項目



必須でないOption設定

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

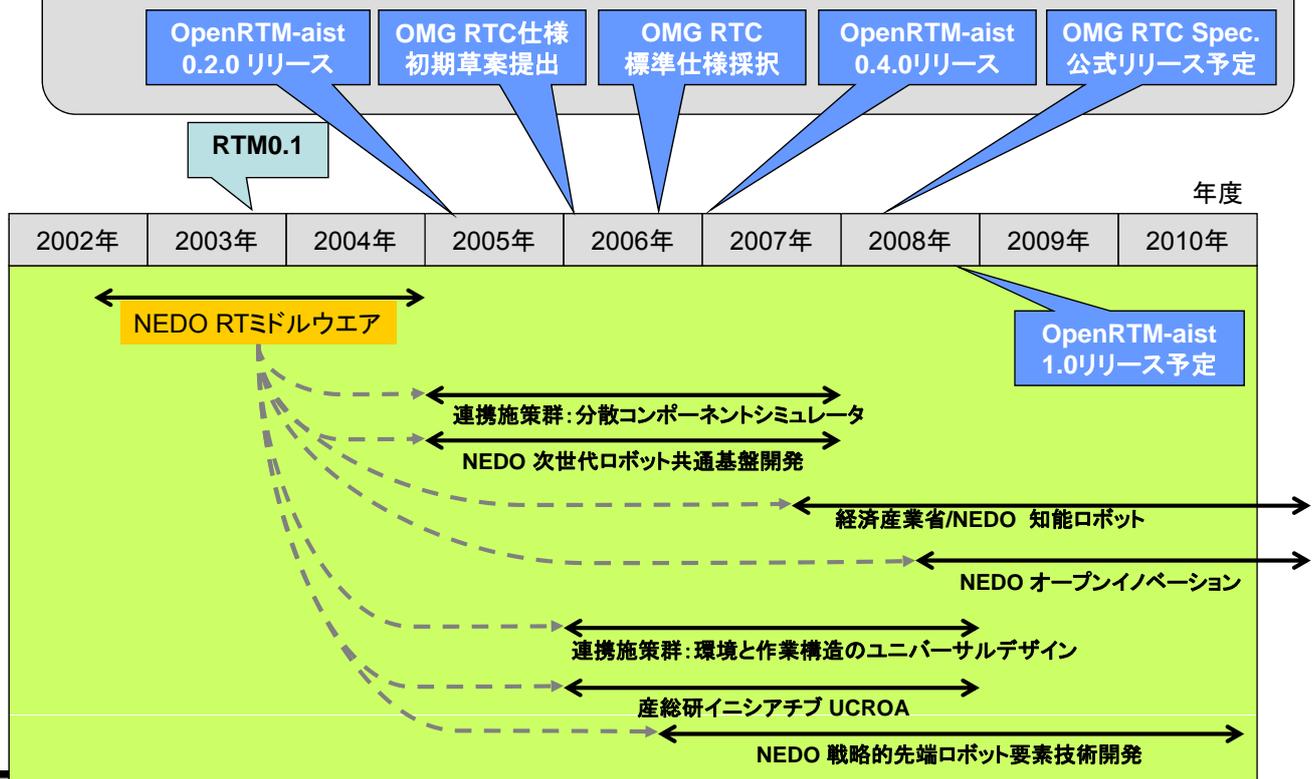
RTミドルウェアのまとめ

- RTミドルウェア
 - ロボットシステム用ソフトウェアプラットフォーム
- RTコンポーネント
 - ソフトウェア部品
- 利点
 - 構成を自由にかつ動的に決めることができる
 - 部品の再利用性の向上
 - システム構築が容易になる



システム開発例

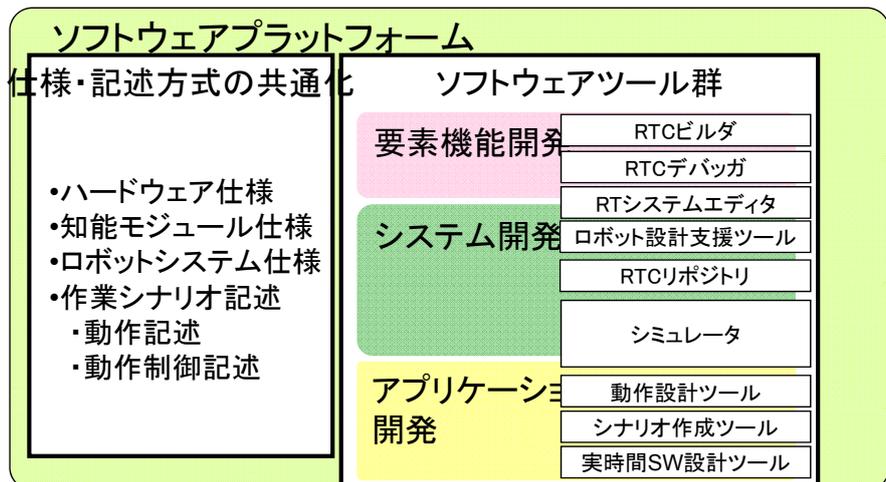
RTミドルウェアの広がり



次世代ロボット知能化技術開発プロジェクト

- 平成19年度(19億円)～23年度(5年間)
- 開発が計画されている知能モジュール群
 - 作業知能(三菱電機、東芝、安川電機、産総研等)
 - 移動知能(富士重工、富士通、アイシン、東大等)
 - コミュニケーション知能(NEC、ATR等)

• RTミドルウェアを基盤としたロボットソフトウェア開発のための統合プラットフォームを開発
 • 共通のプラットフォーム上で、さまざまな知能モジュールを開発する

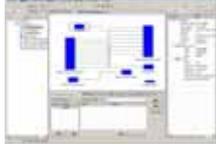


NEDO ROBOT PROJECT

OpenRTプラットフォームは、RTシステム開発プロセスをEclipse上で一貫して行うためのツールチェーンを提供します



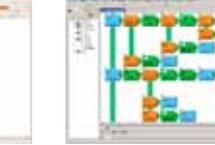
RTCビルダ



RTシステムエディタ



実時間ソフトウェア設計支援ツール

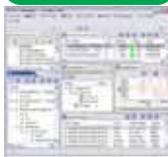


シナリオエディタ

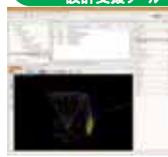
RTコンポーネント開発 システム開発 シナリオ開発 完成

「OpenRTプラットフォーム」
RTミドルウェア + Eclipse + ツールチェーン

RTCデバッグ



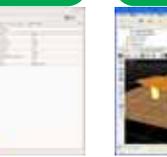
ハードウェア設計支援ツール



動力学シミュレータ



移動動作設計ツール



動作パターン設計ツール



※ 一部のツールは、OpenRTM-aist、OpenHRP3のWebページにて配布中です。
 ※ 動作パターン設計ツールはEclipseプラグインではありません。

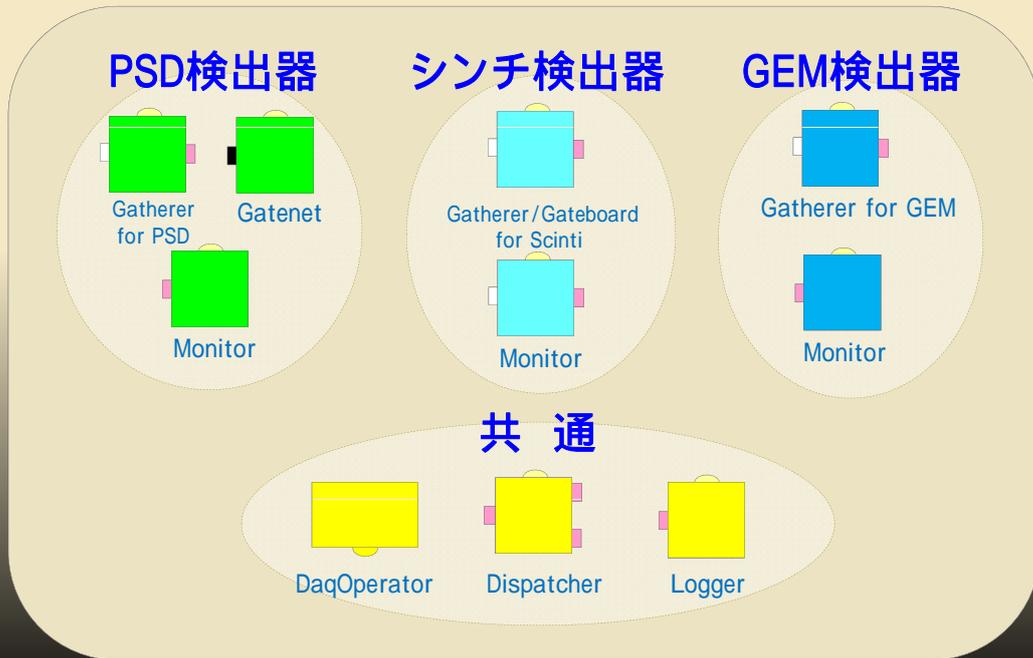
使用実績

J-PARC 物質・生命科学実験施設



使用実績

J-PARC MLF用のコンポーネント



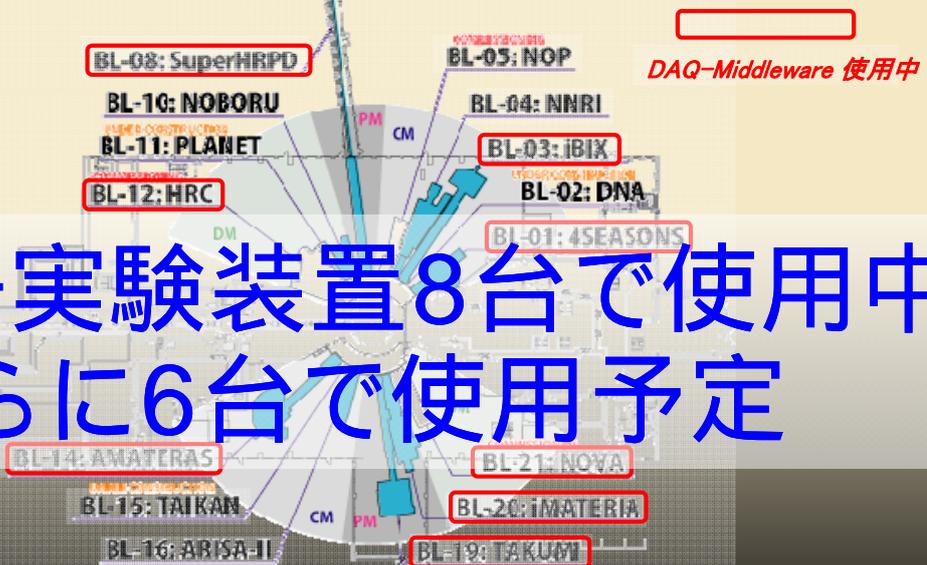
DAQ-Middleware



使用実績

中性子実験装置の現状

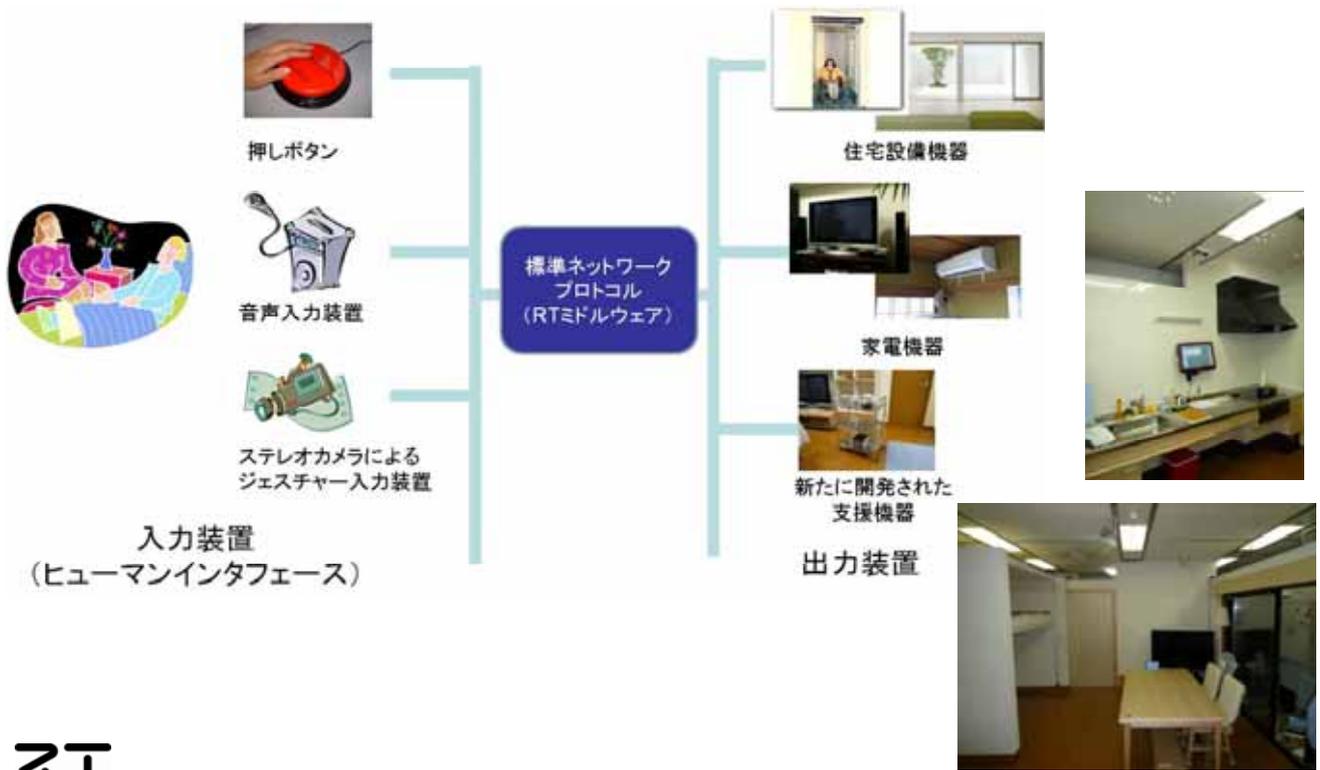
中性子実験装置8台で使用中
さらに6台で使用予定



DAQ-Middleware



障がい者が自立して住みやすい住環境モデル

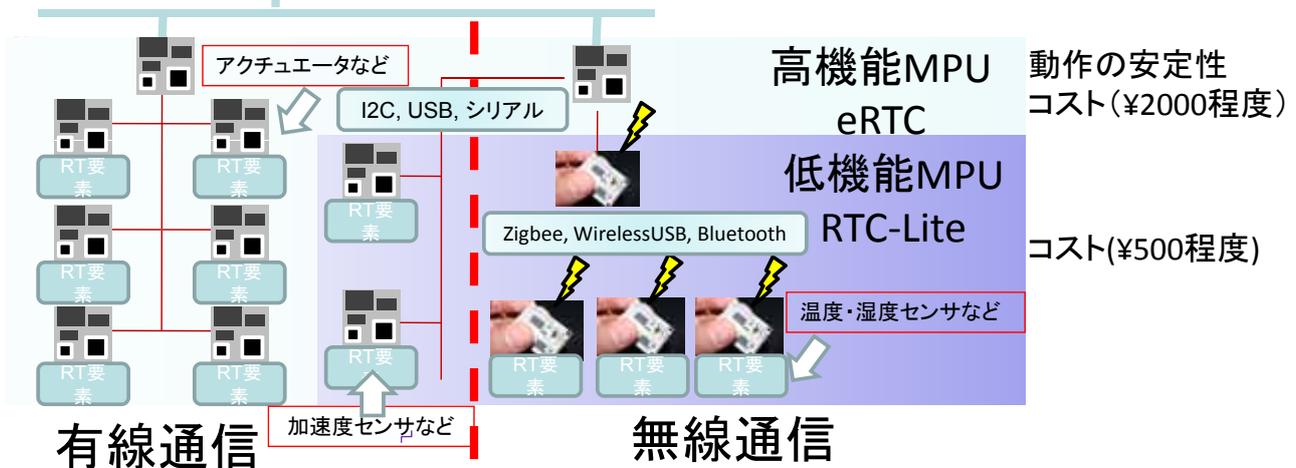


オープンイノベーションプロジェクト



ホームサーバ

居住環境のRT化



躯体 損壊度判定



パワーアシスト・ウィンドウ



省エネ

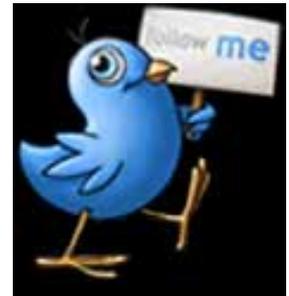


セキュリティ

OpenHRI

Opensource software components for Human Robot Interaction

- 音声合成・対話制御など、ロボットのコミュニケーション機能の実現に必要な各要素を実現するコンポーネント群です
- ライセンスについて OpenHRIのライセンスは、Eclipse Public License (EPL)



<http://openhri.net/>

RoboMec2009 講習会 [旭川]



- 約30名の参加
- RSiさんと合同
(午前中)

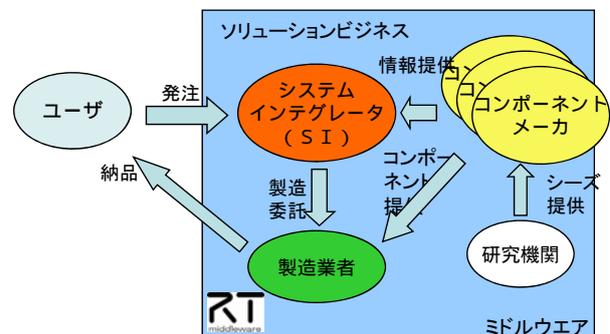
RoboMec2009 OS [旭川]

- 知能化プロジェクト (22件)
- RTシステムとオープン化 (26件)



RTミドルウェアコンテスト(趣旨)

ロボット技術の共有と蓄積を目的として、ソフトウェアのモジュール化の推進

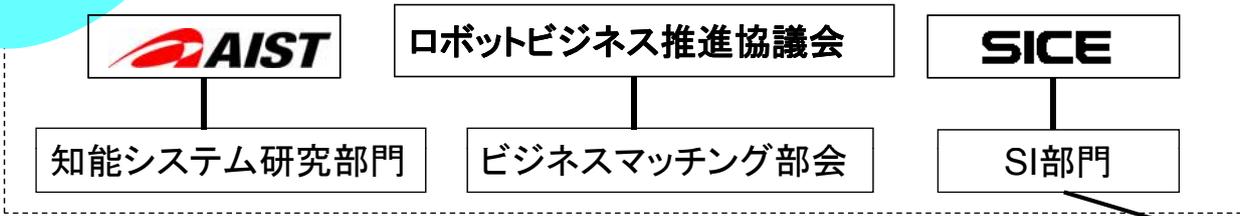


- 有益なコンポーネントやツールの充実を狙う
- これからのロボットソフトウェア開発者に不可欠なRTミドルウェアに精通する技術者の育成を期待

協賛(スポンサー)募集中!

コンテスト実施体制

共同開催



・ホームページ
・RTM技術サポート

・スポンサー募集
・事務局

・技術審査
・SI2009(会場、表彰式)

RTシステムインテグレーション部会

・企画

7月: 実行委員会設立
共催依頼
7月末: 要領発表
8月: エントリー開始
9月: エントリー締め
随時: 講習会開催
12月: SI2010でプレゼンテーションと表彰式

コンテスト実行委員会
ボランティア+JARA事務局

表彰委員会
SICEからの指名
スポンサーからの指名

個人スポンサーが直接投票

- ・SICE賞 (最優秀賞)
SICE SI部門長
- ・協賛企業賞(奨励賞)

10万円

2万円

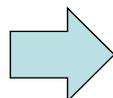
1万円

協賛個人賞(奨励賞)

特徴: 総取り可能な表彰

- ・ 応募者数に対して奨励賞の数が比較的多い
- ・ 全員に平等に奨励賞が贈られるのではない

奨励賞という投票システムの導入

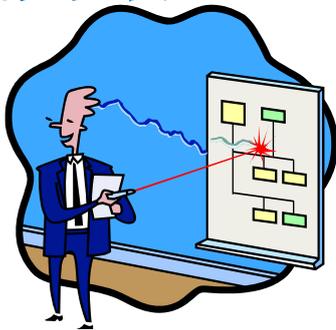


群を抜いた素晴らしい開発成果においては奨励賞の総取りも可能

参加者のインセンティブの向上

特徴: スポンサーニーズの吸収

研究発表



- シーズ提供
- プロトタイプ紹介

奨励賞



2009

- ニーズ開示
- 開発課題(審査基準)

双方向コミュニケーションの場を提供

特徴: 利用者の参加

ホームページ

提供者

- ソースコード
- マニュアル
- ノウハウ

コメント

利用者

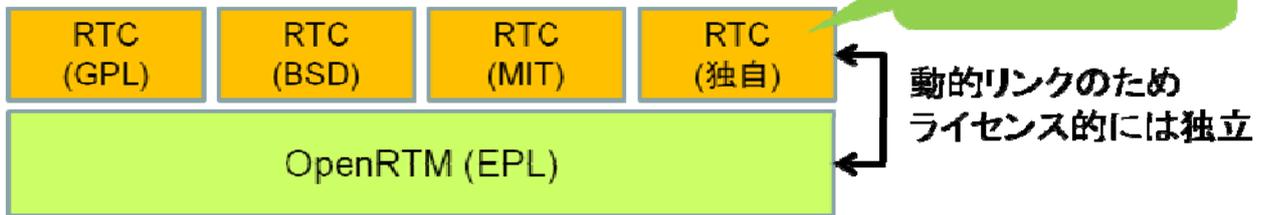
- 問い合わせ
- バグ報告
- 機能追加要望
- 感想



質の向上を目指したコミュニケーション

ライセンス (RTコンポーネント開発・配布)

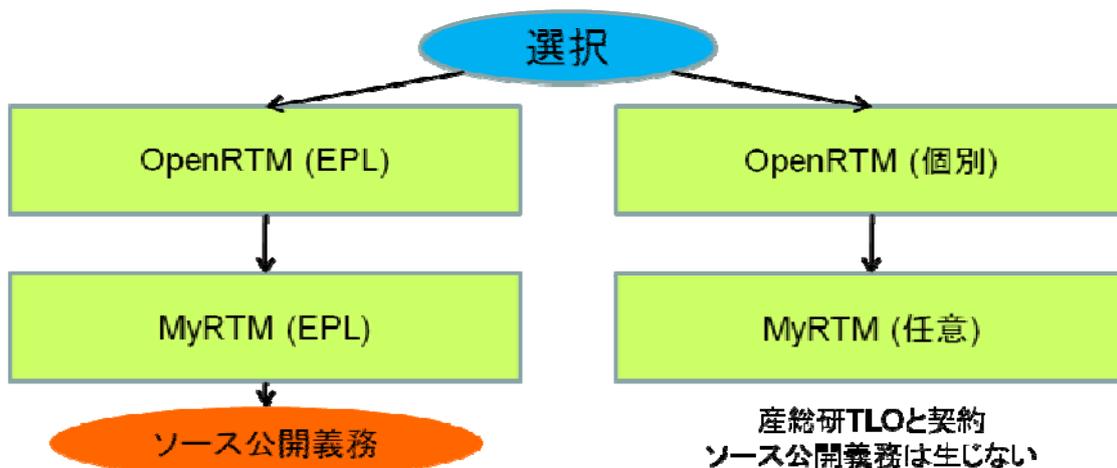
ソースコードをオープンにするかクローズにするかを自由に選ぶことができる



RTCには任意のライセンスを適用可能

OpenRTM-aist のライセンスは、個々のRTコンポーネントには及ばない。したがって、RTコンポーネントの作成者は、自由なライセンスで配布・販売することができる

ライセンス (OpenRTM-aistの改編、再配布)



2つのライセンスのうち一つを選択

OpenRTM-aist は、EPLに基づくオープンソースライセンスと、個別契約に基づくライセンス付与のどちらかを選択する、デュアルライセンス

テクニック(分からないこと)



- ホームページから情報を探す
- **検索エンジンで情報を探す**
(OpenRTMの問題でないことも多い)
- 近くのOpenRTMユーザに聞いてみる
- ユーザメーリングリストで、**自分の状況をきちんと説明して**質問する。皆から貰った情報を結果とともに**整理して報告**する。

[ホーム](#) » [コミュニティ](#) » [メーリングリスト](#)

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

これからのロボット開発

ソフトウェアのモジュール化が現実になることで...

- 既存のモジュールを組み合わせて設計
- 既存のシステム設計をテンプレートとして活用
- システムのカスタマイズが容易
- 開発したシステムのメンテナンス性も高まる
- ロボットを作ることよりも、ロボット技術を利用したサービス開発に研究開発資源を集中
- 研究成果は論文だけでなく、モジュール化して提供する。(技術移転も容易であり、技術の比較検討も容易)

これからのロボット開発

ソフトウェアのモジュール化が現実になることで...

ロボットを作ることよりも、**ロボット技術を利用した異分野融合によるサービス開発**に研究開発資源を集中

- 生活支援・介護のシステム化
- 農林水産分野のシステム化
- 実験系研究のシステム化
- 医療分野のシステム化
- 交通・物流分野のシステム化
- セキュリティ・防衛分野のシステム化 など

社会の中の諸課題をRT技術を導入して
システム化して効率を高めることで解決を目指す

参考

- [OpenRTM-aistのホームページ](http://www.openrtm.org/)
<http://www.openrtm.org/>
- [OpenRT Platform フィシャルサイト](http://www.openrtp.jp/wiki/)
<http://www.openrtp.jp/wiki/>
- 「使えるRTミドルウェア」特集号
日本ロボット学会論文誌 vol.28, no.5
- [Robotics -DSIGのホームページ](http://robotics.omg.org/)
<http://robotics.omg.org/>

コンポーネント開発例

- [RT ミドルウェアコンテスト](http://www.openrtm.org/rt/rtmcontest.html) (産総研)
<http://www.openrtm.org/rt/rtmcontest.html>
- [NEDO知能化モジュール集2009](http://www.nedo.go.jp/kankobutsu/pamphlets/02kikai/chinou.pdf) (NEDO)
<http://www.nedo.go.jp/kankobutsu/pamphlets/02kikai/chinou.pdf>
- [RTミドルウェア技術カタログ2009](http://www.openrtm.org/rt/OpenLab2009Catalog.pdf) (産総研)
<http://www.openrtm.org/rt/OpenLab2009Catalog.pdf>