



RTM概論

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ミドルウェアプロジェクト
3. 標準化活動
4. 関連プロジェクト紹介
5. まとめ



本日の狙い:

DAQミドルウェアのベースとなっているRTミドルウェアの概要を知り、DAQミドルウェアのコミュニティへの貢献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

技術戦略

21世紀におけるロボット社会創造のための技術戦略調査報告書、
(社)日本機械工業連合会、(社)日本ロボット工業会 (2001)

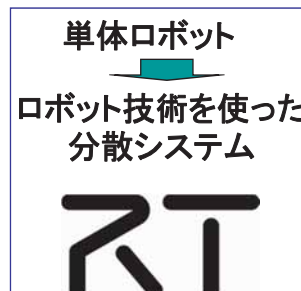
ロボットからRTへ、オーダメイド型の
ロボット産業への変革を提言



三浦宏文委員長



谷江和雄代表幹事



<http://www.jara.jp/publication/dl/rt.pdf>

RTミドルウェアプロジェクト

産業界の変革

(大量生産から多品種少量生産への構造変革)

ロボット産業の現状

大量生産向きロボット構築法

各社独自仕様によるハードウェアからソフトウェアまでの一括システム開発

- 設計に時間とコストがかかる
- 高い技術力のある少数メーカーのみ参入
- 研究成果の移転が困難

コスト競争の体質

21世紀のロボット産業

多品種少量生産向きロボット構築法

ロボット要素のモジュール化とその統合によるニーズ対応システム開発

- ロボット要素機器の新市場
- 各専門分野での分業体制
- インテグレート産業の創出

付加価値競争の体質

ブレークスルー技術

技術共有を促進する

ロボット共通基盤技術の確立

- ロボット開発の活性化
- 産業競争力の強化

ロボット産業の産業構造変換

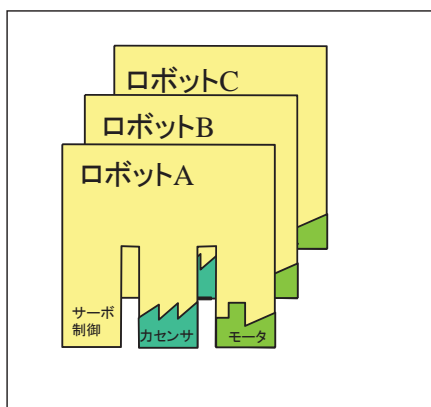


RT
middleware

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

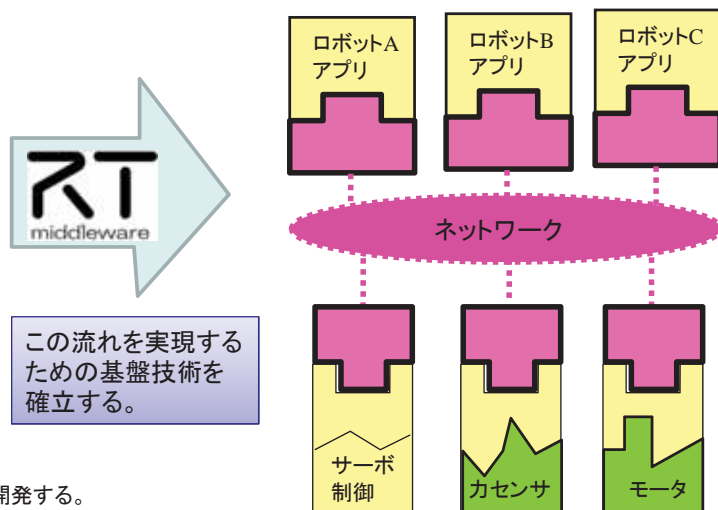
RTミドルウェアプロジェクト(2002-2004)

従来のロボット



- 各ロボットは必要なロボットの要素を全部一体として開発する。
- 各ロボットの要素の切れ目が明確でなく、他のロボットには転用ができない。
- 1つのロボットを作るのにコストがかかる。
- 新しいロボットの開発が難しい。

モジュール化に基づいたロボット



この流れを実現するための基盤技術を確立する。

- すでに供給されているモジュールを利用して新しいロボットを簡単に作るができる。
- 1つのロボットを作るためのコストが下がる。
- モジュールのみの供給ができる。
- ハードを持たないメーカーもロボット産業に参入できる。
- 新しいロボットが容易に開発できる。

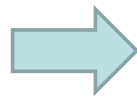
なぜ標準化が必要か？

ロボット(メカトロ)システムのソフトウェア開発コストの増大 (システムの複雑化に起因)

機械システムのCADのような
ロボット用ソフトウェア開発用のCADが必要

互いに繋がるだけでは駄目

- コンポーネントの定義
- コンポーネントのライフサイクルの操作・監視
- コンポーネント間の動的接続の実現



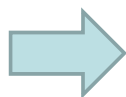
コンポーネントモデルの標準化

なぜ標準化が必要か？

- モジュール化ですべてが解決しない。
コンポーネントは器であり、その中身が問題

異なるインタフェースを持ちながら
よく似た機能を実現するコンポーネントの乱立

- 互換性がないとシステム設計者の選択肢がない (アダプタ設計のために余計な手間が必要)



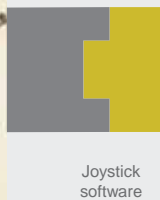
機能モデルごとの共通化・標準化
(技術革新に合わせた拡張性をもたせて)

RTミドルウェア

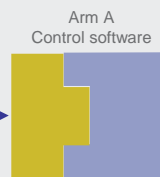
RTミドルウェアは別々に作られたソフトウェアモジュール同士を繋ぐための共通インターフェースを提供する



Joystick



Joystick software



Arm A Control software



Humanoid's Arm



Arm B Control software



Robot Arm

compatible arm interfaces

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

RTミドルウェア普及によって

ユーザのニーズに応じたロボットシステムをインテグレータが容易に構築

RTスペースの導入



ユーザーとインテグレータがRTスペースの導入について打ち合わせ。



RTミドルウェアのRTスペース作成ツールを使用し、アプリケーションを作成。



作成したアプリケーションを、シミュレーションツールを使い動作確認。



ユーザー宅へのRT機器設置と、アプリケーションソフトのインストール。

RT機器の追加



最新のお掃除ロボットを購入。



お掃除ロボットの情報と、既設のRT機器と協調して動作させるためのアプリケーションソフトをインストール。

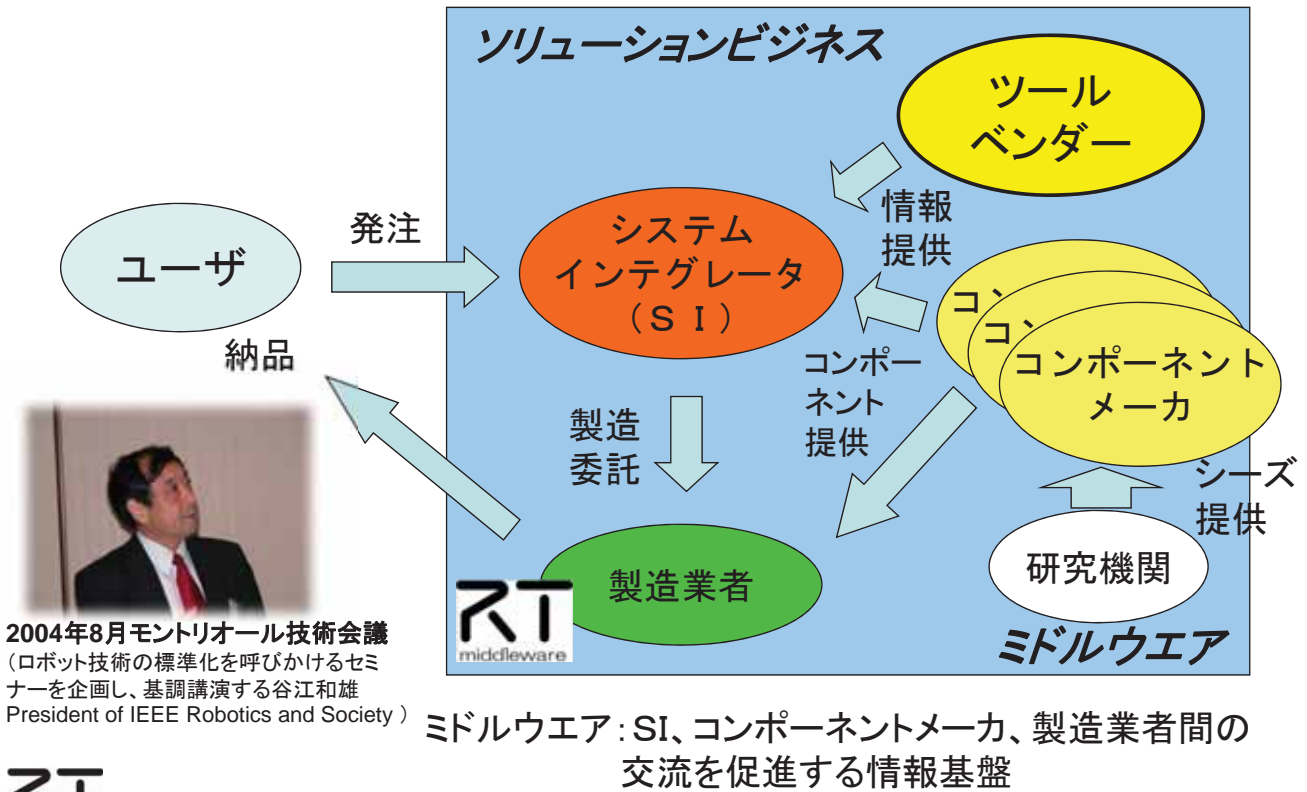


更新した機器情報とソフトを各RT機器へ送信。

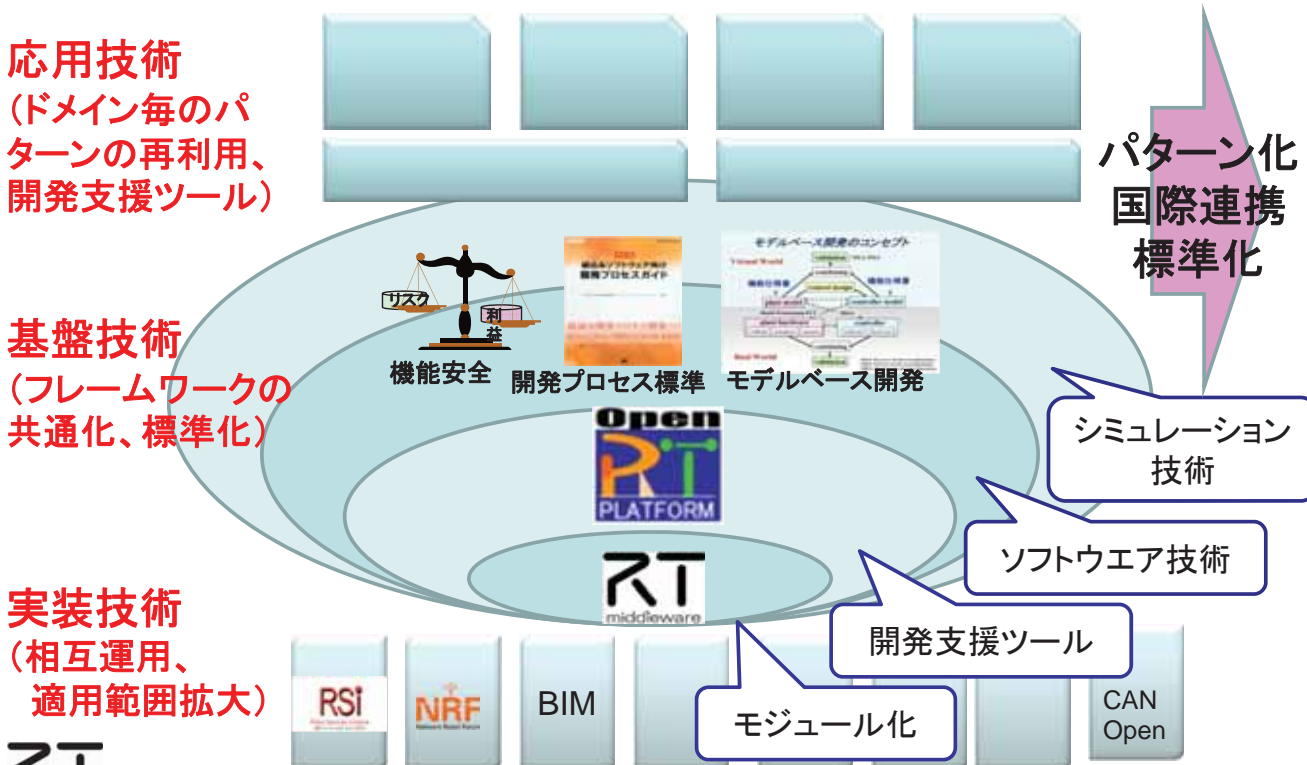


追加したお掃除ロボットと、既設のRT機器が連携して、新たな掃除機能を開始。

期待される未来のロボット産業のモデル

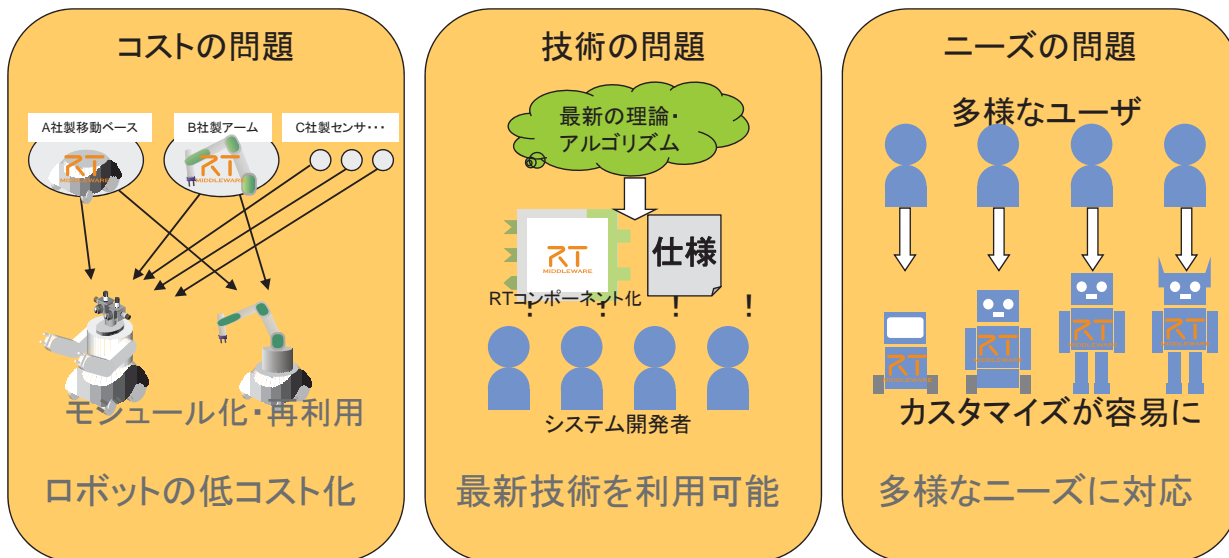


RTミドルウェアの目指すもの



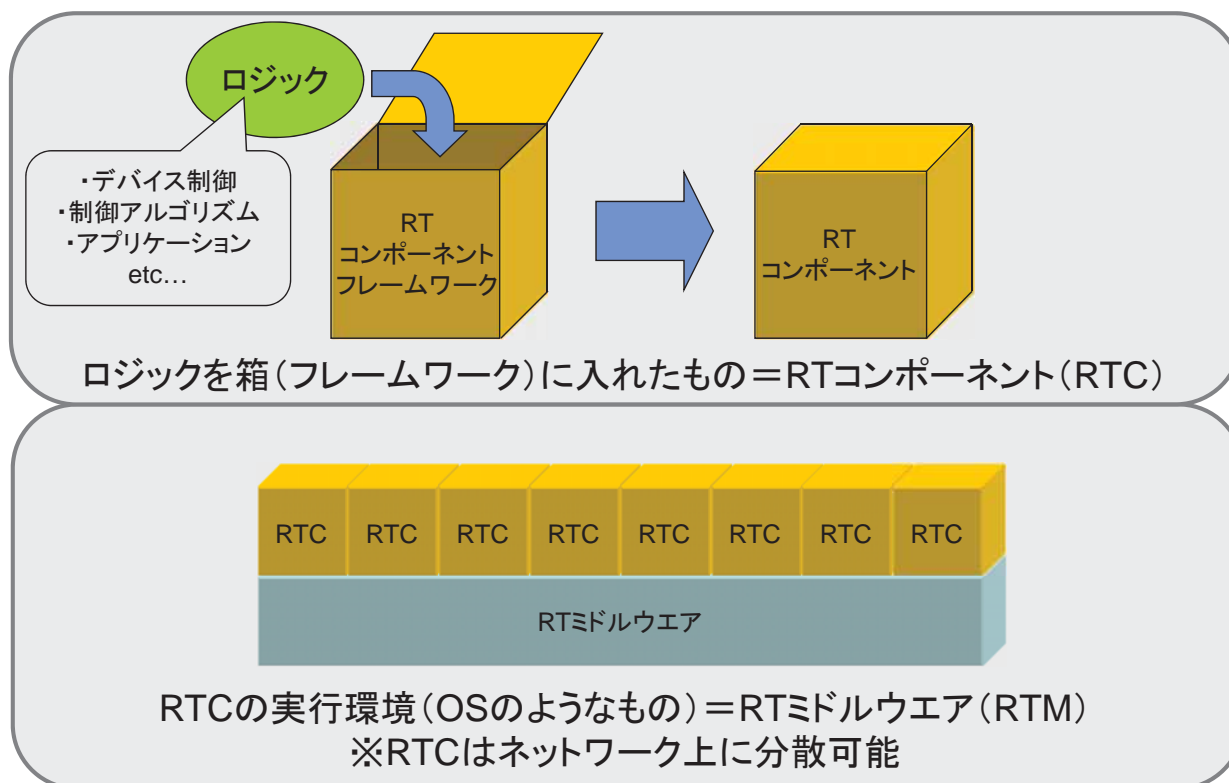
RTミドルウェアの目的

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

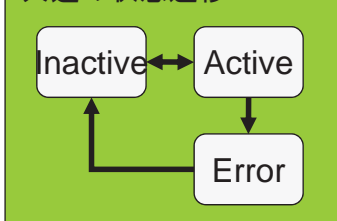
RTミドルウェアとRTコンポーネント



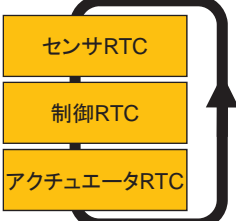
RTコンポーネントの主な機能

アクティビティ・実行コンテキスト

共通の状態遷移



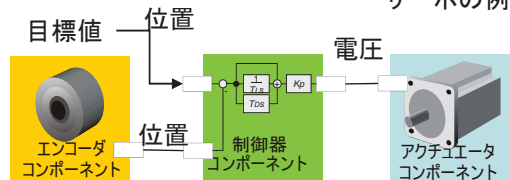
複合実行



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

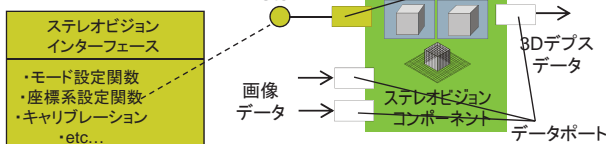
- データ指向ポート
- 連続的なデータの送受信
- 動的な接続・切断



データ指向通信機能

サービスポート

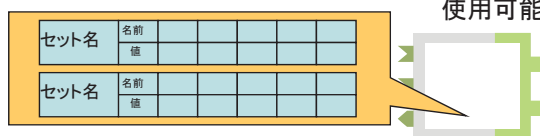
- 定義可能なインターフェースを持つ
- 内部の詳細な機能にアクセス
 - パラメータ取得・設定
 - モード切替
 - etc...



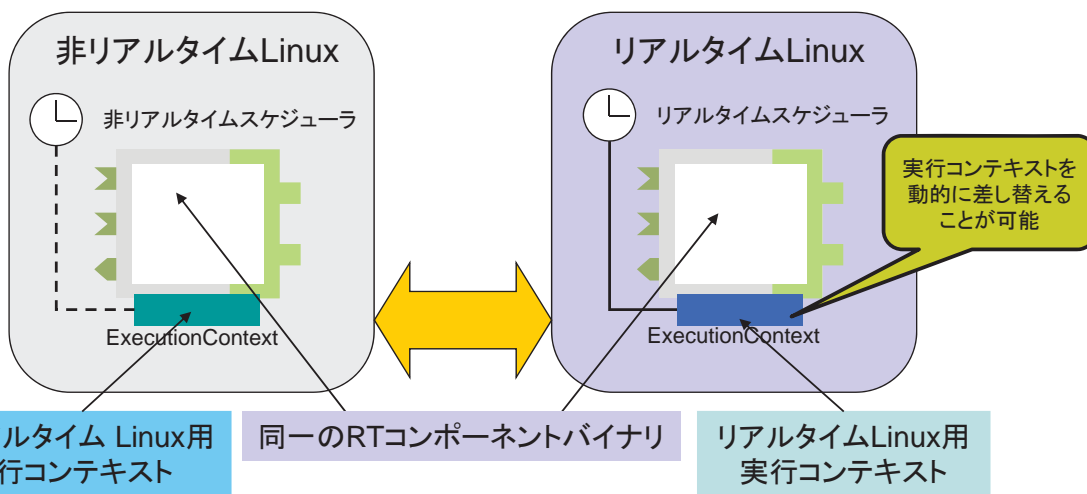
サービス指向相互作用機能

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

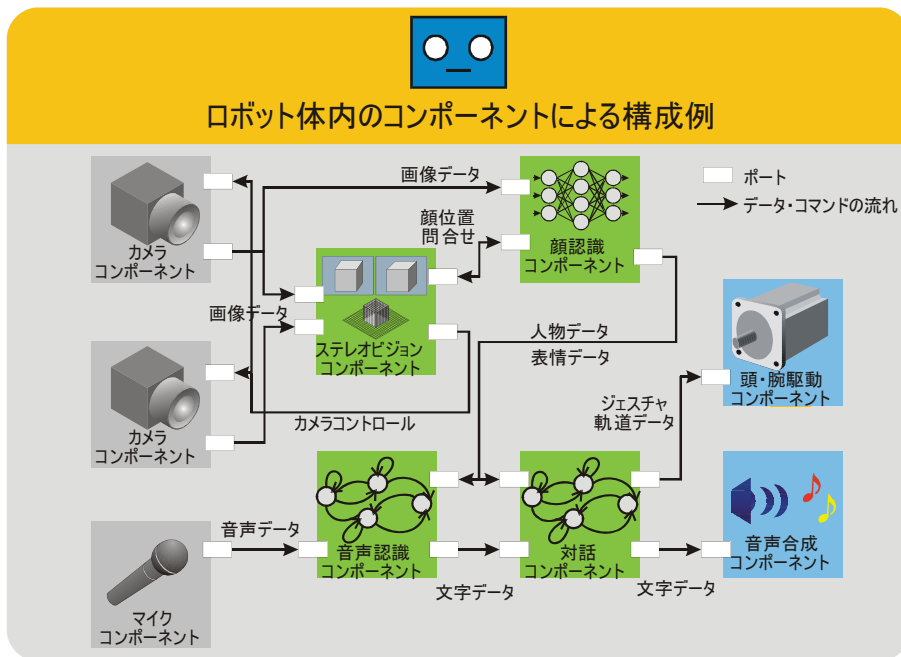


リアルタイム実行コンテキスト



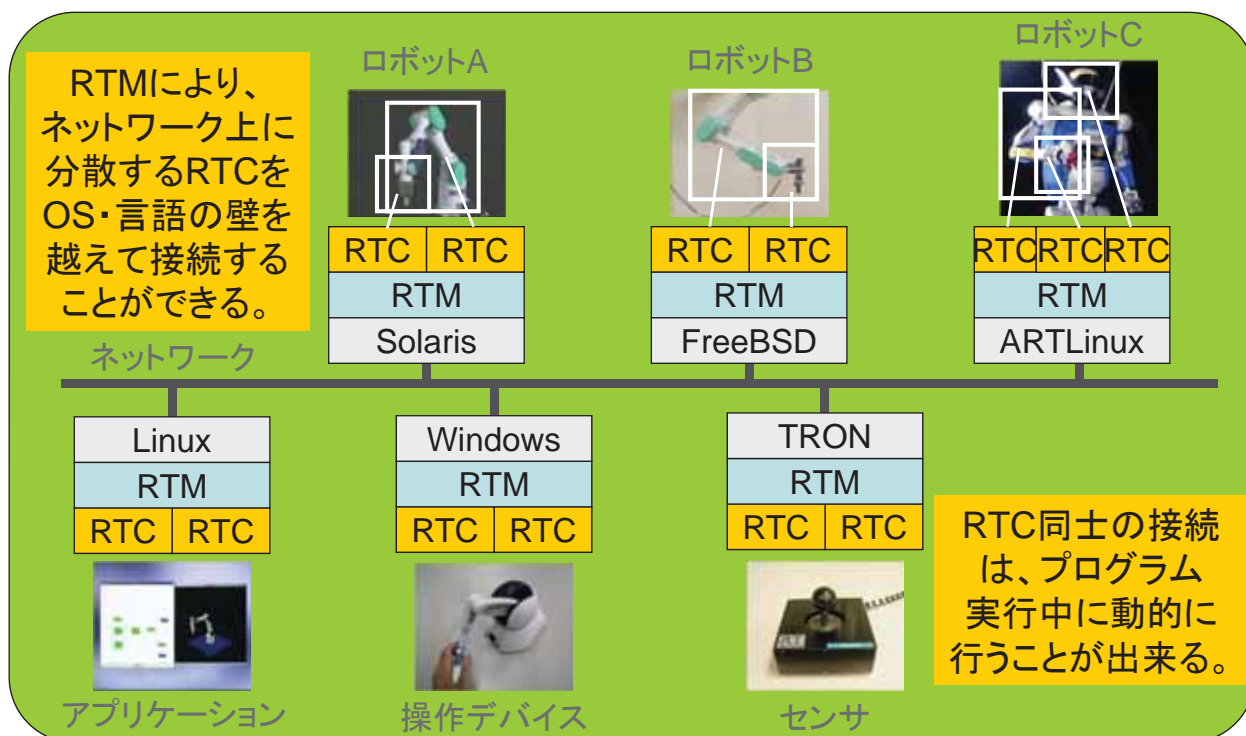
非リアルタイムLinux環境で作られたRTコンポーネントを再コンパイルせずにリアルタイムLinux上でリアルタイム実行可能

RTCの分割と連携



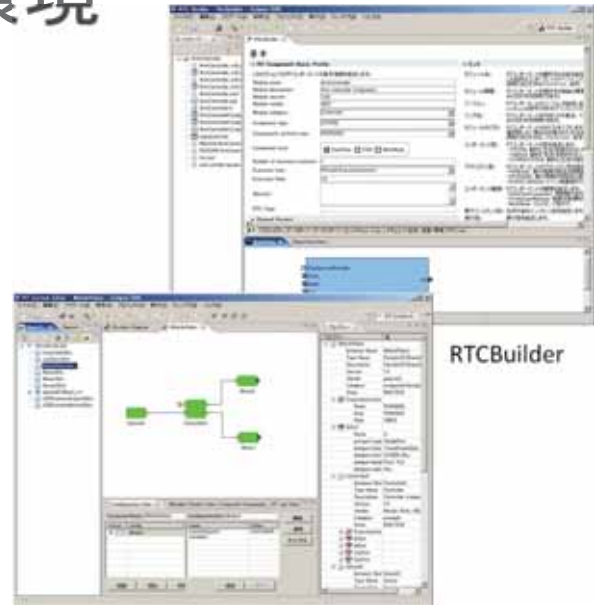
(モジュール)情報の隠蔽と公開のルールが重要

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 (コマンドライン版)
 - RTコンポーネントのテンプレート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 (GUI版)
- RTShell (コマンドライン版)
 - 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



RTSystemEditor

RTC・RTM統合開発環境の整備
 RTC設計・実装・デバッグ、RTMI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

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 ConsoleInImpl(mgr)
    {
    }
}
    
```

RTC source for Python

```

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

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):
    
```

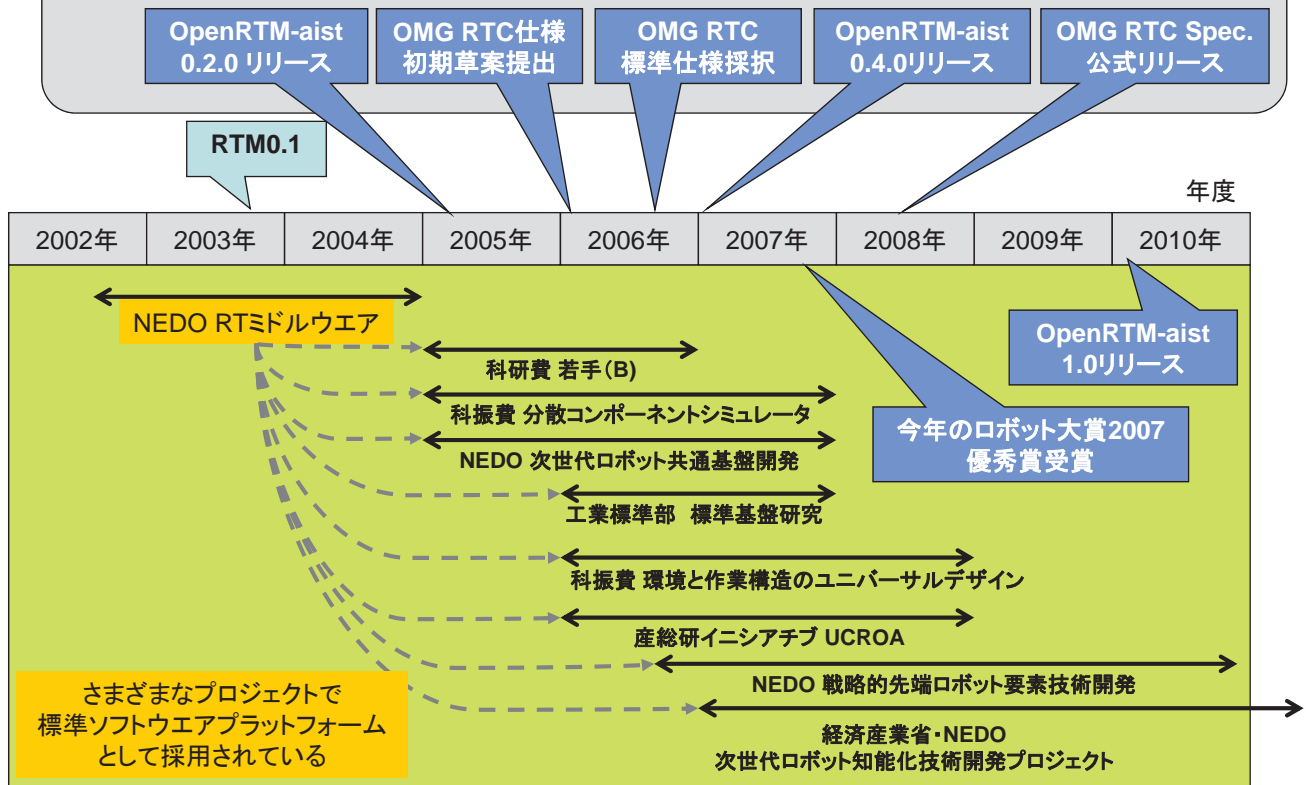
OpenRTM-aist

- 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: 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
 - その他
 - Pattern weaver for RT-Middleware (株式会社テクノロジックアートより発売中)

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

RTミドルウェアの広がり



NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

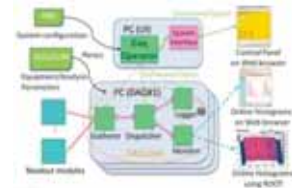
多様な実装

- OpenRTM-aist: 産総研実装
 - C++, Python, Java の3種類
- OpenRTM.NET: 株式会社SEC版実装
 - .NET版: VB, C#
- 韓国ETRI
 - OPRoS コンポーネント: 一部準拠
- PALRO: 富士ソフト
 - 小型ヒューマノイド制御フレームワークがC++ソースレベルでOpenRTM互換
- GostaiRTC: 仏GOSTAI & Thales
 - OMG RTC Local PSM に準拠



実用化・事業化

- Pattern Weaver for RTM
 - テクノジックアートのUMLツールの拡張
- J-PARC(大強度陽子加速器施設)DAQシステム
 - KEKのDAQミドルウェアはRTM上に構築
- HRP-2
 - GRXのヒューマノイドロボット制御プログラムがOpenRTMへ移行予定
- HIRO
 - GRXの双腕ロボットがQNX版OpenRTMへ移行予定
- たいぞう
 - GRX、産総研の体操ロボットがOpenRTMへ移行予定



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(2)



[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

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

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



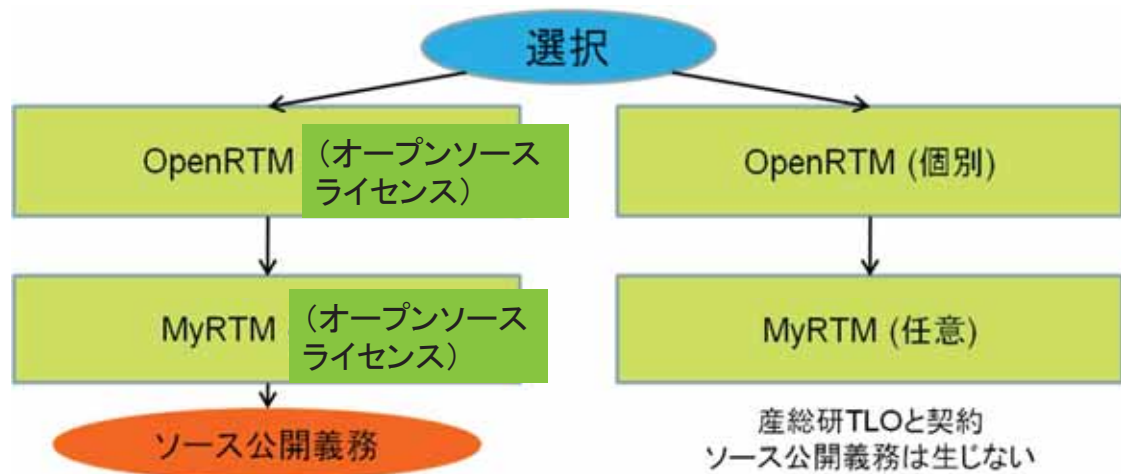
商用ライセンスを設定し、
販売することもできる。

動的リンクのため
ライセンス的には独立

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、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

今後の展望

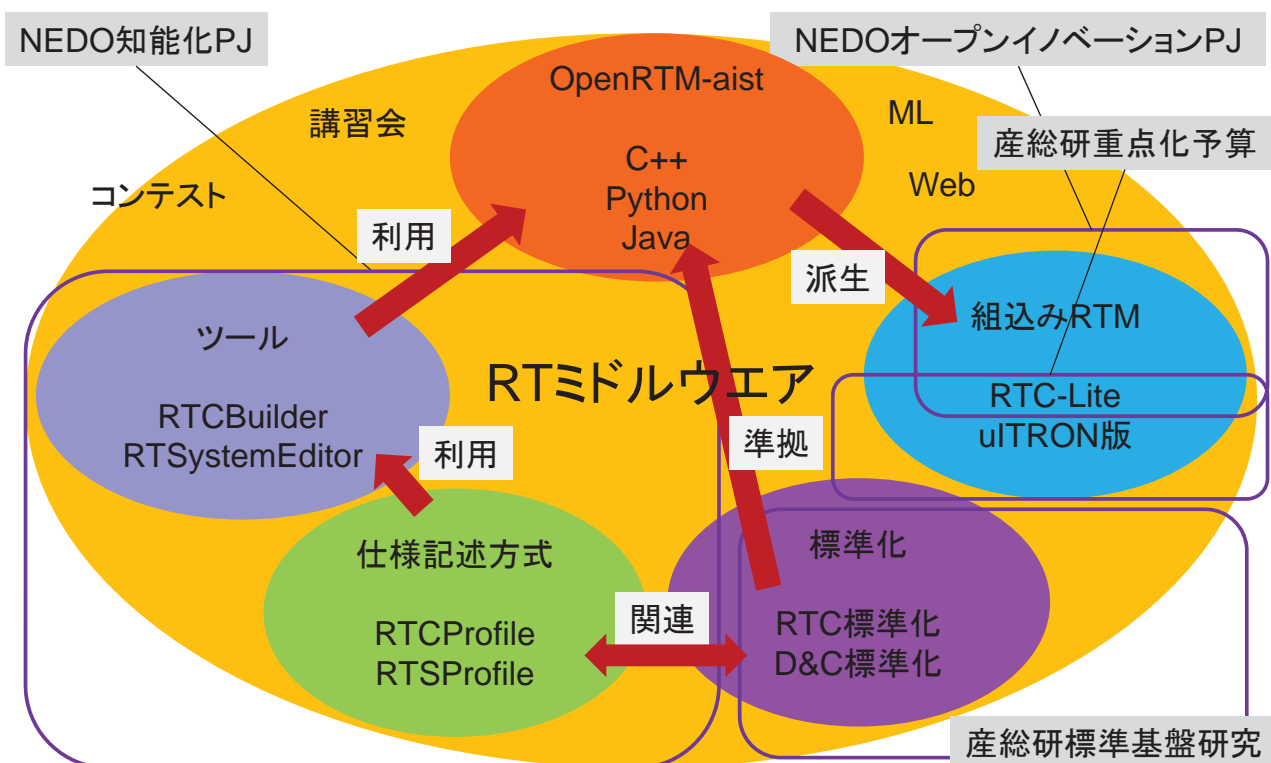
NEDO知能化プロジェクト

(次世代ロボット知能化技術開発プロジェクト)

- 今年度で終了
 - OpenRTM-aistの開発は今後も産総研・安藤が継続
- 成果を一般公開予定
 - 多数のRTコンポーネントやツール
 - 原則として自由に利用可能に
 - オープンソース公開、バイナリ公開など様々
 - 継続的メンテナンスを持続する仕組みを検討中
 - 著作編集権の委譲、メンテナの一般からの募集等

ユーザの皆さま役割がより重要になります。
ご協力よろしくお願ひします！！

OpenRTM開発体制(産総研)



新Webページ

ダウンロードサイトから、ユーザコミュニティページへ

- 新メーリングリスト
- フォーラムの設置
 - メーリングリストとの相互乗り入れ
- プロジェクト公開ページ
 - ユーザが自分のRTCやツールを自由に公開可能
- Wikiページ
 - ユーザが自由に編集できるページ
 - ちょっとしたノウハウ等の共有

一部は現在準備中ですので
もう少しお待ちください



標準化活動

OMG標準化活動のご紹介

標準化活動の開始

RTミドルウェアのコンセプト実現

ある程度の普及が不可欠(卵と鶏)

(RTコンポーネントの品揃え ⇄ 利用者の増加)

開発と並行して標準化活動(コンソーシアム標準)



2004年8月 Montreal会議
(次世代ロボットの標準化を呼びかける
フォーラムを開催したあとの打ち上げ)

OMG (Object Management Group)

Robotics Domain SIGの設立(2005.02)

- 水川真(芝浦工業大学)
- 神徳徹雄(産業技術総合研究所)

Robotics Domain TFへの移行(2005.12.)

日米韓の協力体制



2005年2月 Burlingame会議
(SIG設立祝いの巨大ステーキを楽しむ
小泉さん、小菅先生、水川先生、谷江さん)



NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

OMGの概要

Object Management Group



• 国際的ソフトウェア技術標準化コンソーシアム

- Distributed Object Middleware (CORBA)
- Object Model Language (UML)
- Model Driven Architecture (MDA)

• 各応用分野の標準化推進

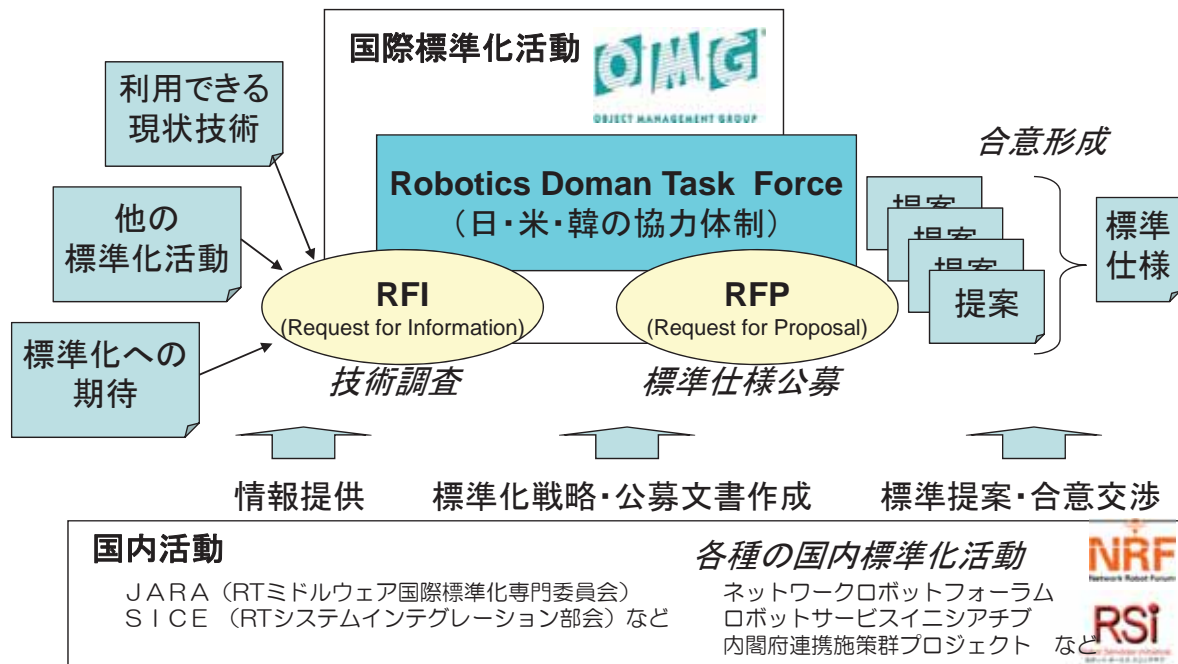
(企業情報システム, 防衛システム, ファイナンス, ヘルスケア, ライフサイエンス研究, 製造, ソフトウェアベース通信, 宇宙, **ロボット**)

-> **Domain Technology Committee**



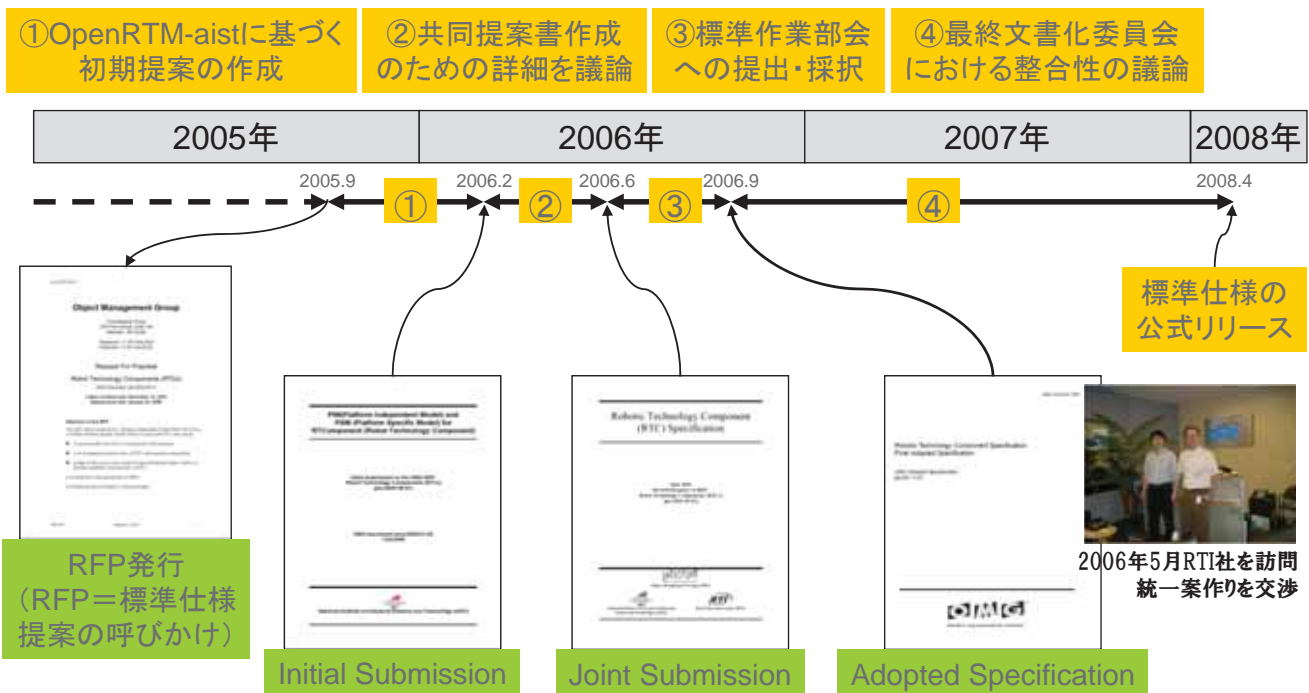
<http://www.omg.org/>

ロボット技術の国際標準化活動



NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

OMGにおける標準化



NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

コンポーネントモデルの統一提案

- 産総研提案
 - Real-Time Innovation提案
- それぞれのコンポーネントモデルの融合を図った統一提案



2006年9月Anaheim技術会議
(Revised Submissionが採択された後、RTI社のHung とRick と一緒に乾杯)



産業展開に向けた導入事例

Advanced Intelligence (株)アドイン研究所
学習・推論コンポーネントと応用システム

話者認識システム
学習

指示: 佐藤さんの声

複数の人に特定の言葉を数回言ってもらおう

話者認識システム
推論

結果: 田中さんの声 (合致度: 95%)

特定の言葉を言ってもらおうと話者が誰かを判定する

APPLIED VISION SYSTEMS (株)アプライド・ビジョン・システムズ
3次元視覚認識・追跡コンポーネント群

テクノロジック アート
RTコンポーネント設計ツール

Pattern Weaver for RT-Middleware

株式会社セック
SEC Systems Engineering Consultants Co., Ltd.
レーザレンジセンサコンポーネント



北陽電機社製
URG-04LX



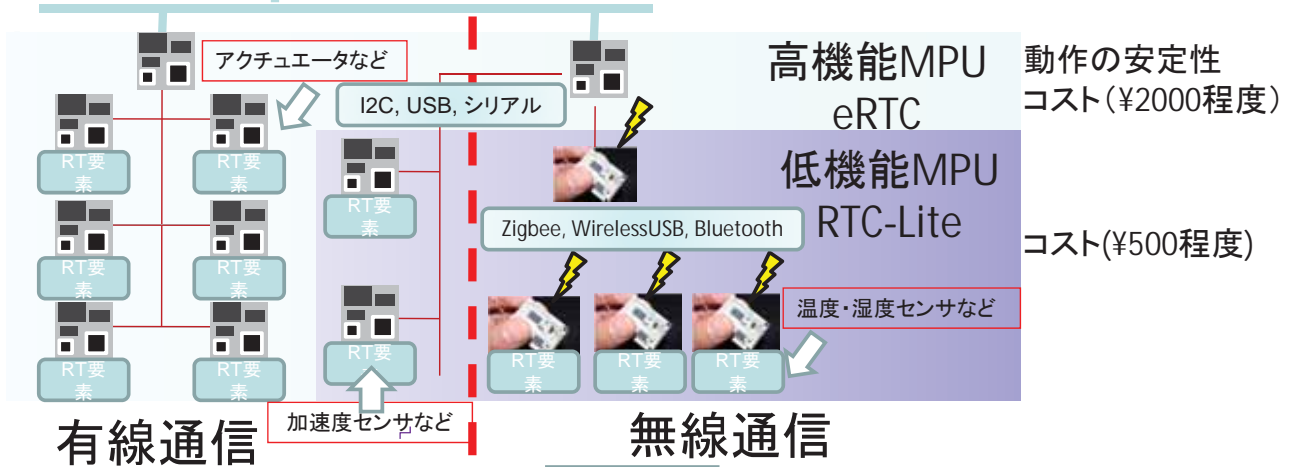
SICK製
LMS200

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



有線通信

無線通信



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

RT
middleware

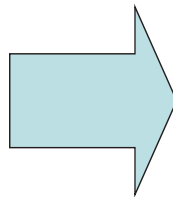
NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

環境構造化

産業ロボットは工場内の環境がすでにロボット用に構築されているため、実用性が高い。実用性を向上するためには、環境にもある程度の構造化が必要。



産業ロボットの環境はロボットに合わせて構築されている。

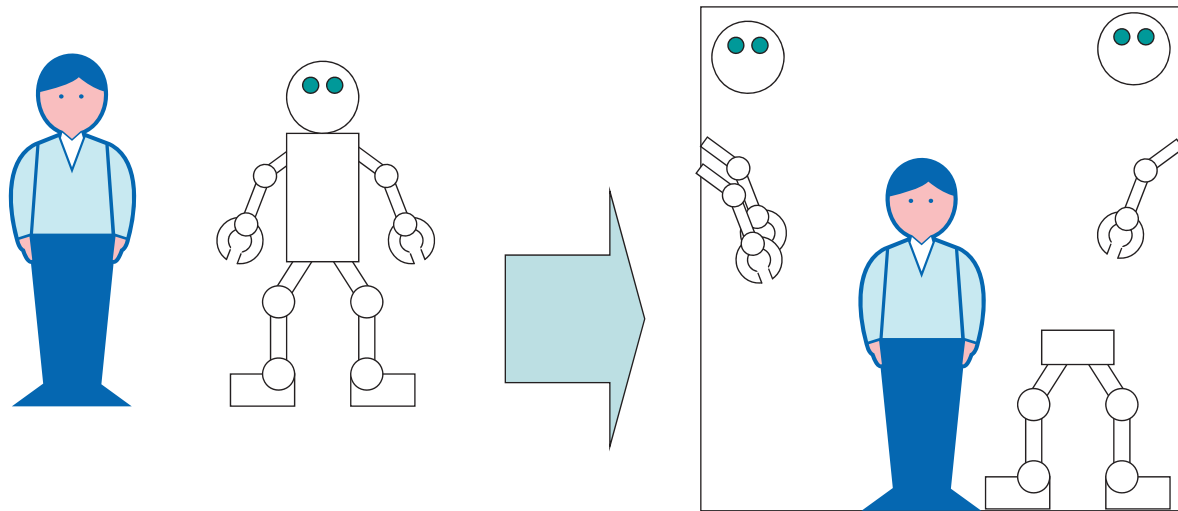


住環境にロボットが導入される際はロボットが作業しやすい環境を構築する必要がある。

RT
middleware

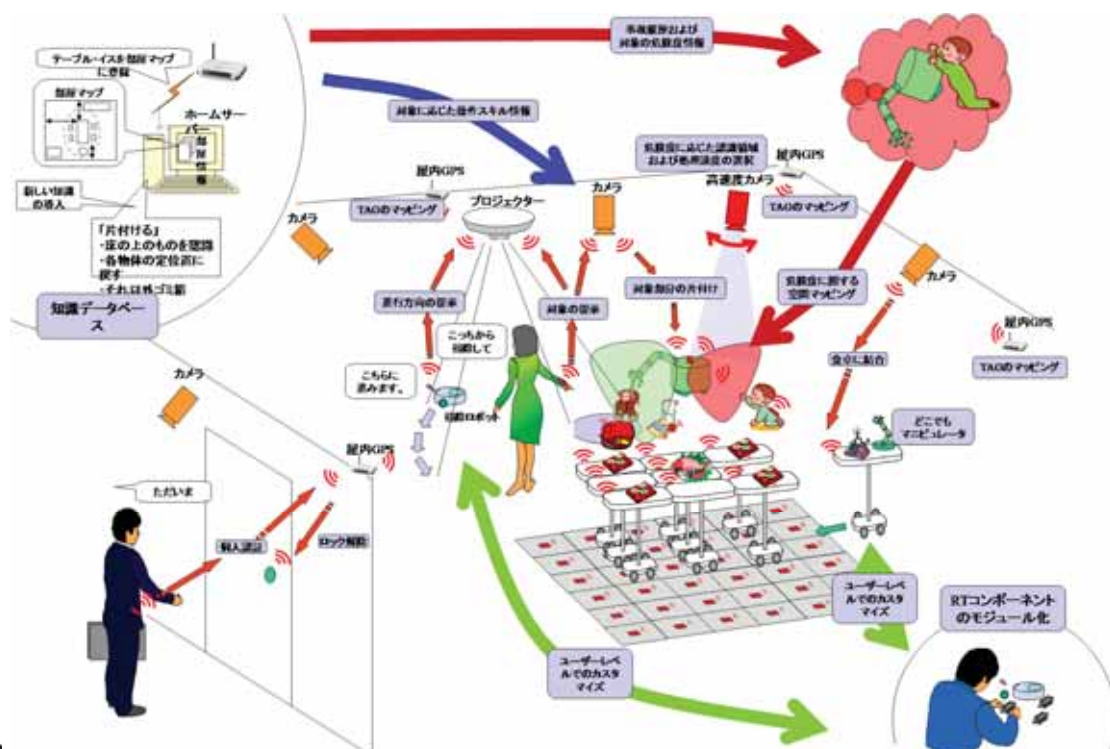
NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

環境のロボット化

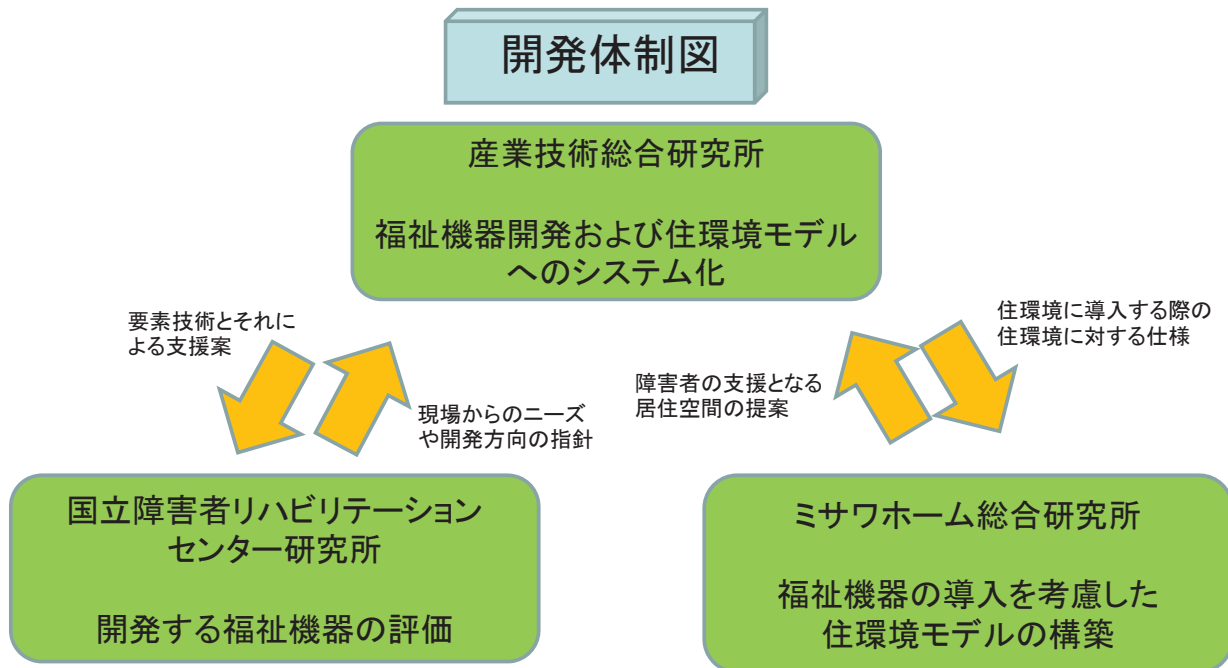


各RTパーツ(ビジョン、センサ、マニピュレータ、知能等)を環境に分散することで環境をロボット化し、人にサービスをすることが可能。
 加えて、サービスに応じて、適宜RTパーツを追加することが可能であり、多様なサービスを提供できる可能性が高い。

ユビキタスロボティクス概念図



厚労省障害者自立支援機器等研究開発プロジェクト 「障害者が自立して住みやすい住環境モデル」

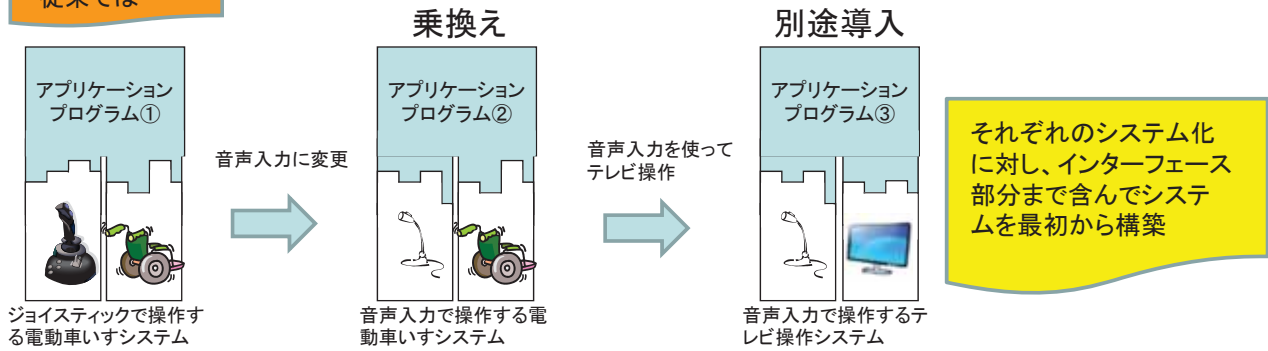


福祉機器のシステム化に際しての課題

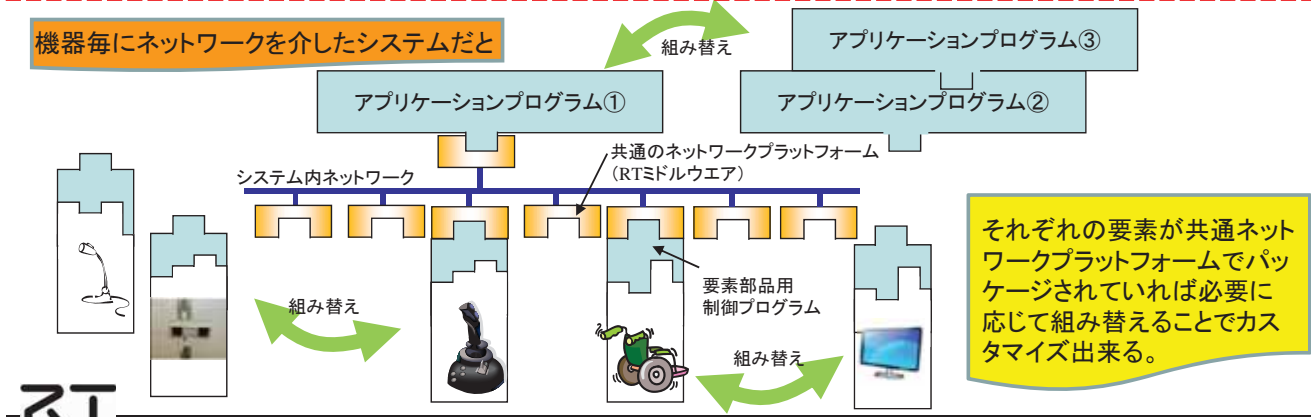
- 福祉機器の多くは入力側、出力側が一体型であり、システムとして固定化していてカスタマイズが困難
- 操作方法を個別に覚える必要があり、操作ツールや機器間の連携動作が困難
- 障害の状態は千差万別、加えて進行性の場合、同じシステムで継続的に利用することが困難

ネットワークを介した機器の連携操作

従来では



機器毎にネットワークを介したシステムだと



ステレオビジョンによるジェスチャー認識技術 ～ 非接触非拘束ジェスチャーインターフェース ～

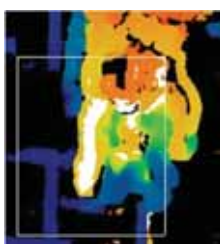
入力装置:ステレオカメラ



上肢機能の障害のためリモコン操作が困難なユーザの対象部位に柔軟に対応

実環境で高精度に動作可能

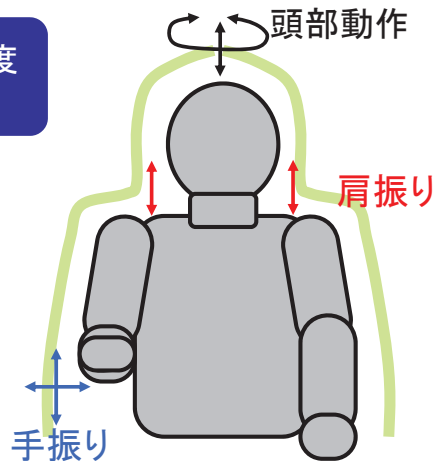
形状情報をリアルタイムで取得手の3次元的な動きだけを抽出



距離情報 (距離が近いほど赤色)



画像情報



ユーザの加齢による身体状況変化にもソフト的に対応

不明瞭な音声でも認識可能な音声入力装置

多様な発音への対応技術

障害に伴う言い方の揺れに対応

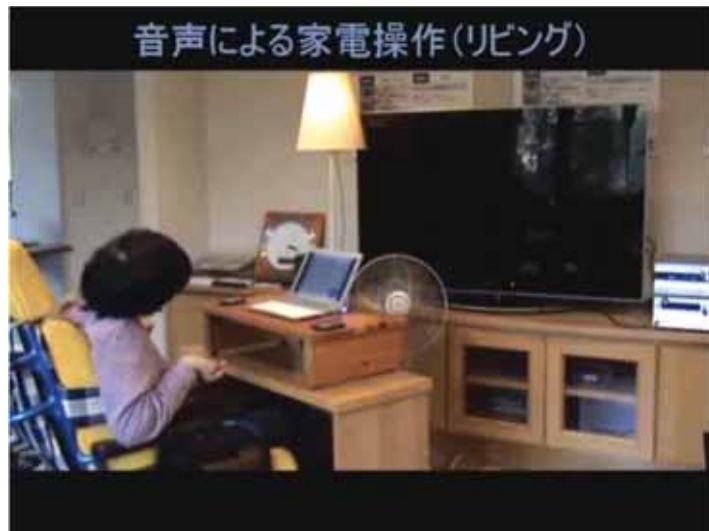
声の特徴が違ってもしっかり認識

多様な雑音への対応技術

マイクへの接触音や咳など
不要な音の影響を除去

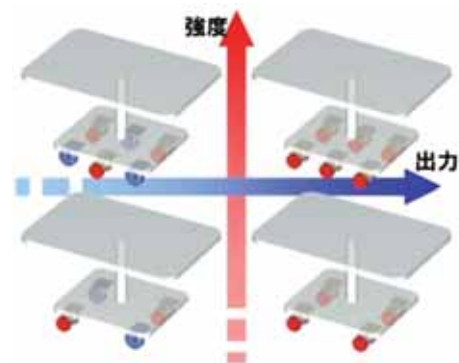
多様な状況への対応技術

部屋のどこに居ても音声認識を利用可能



アクティブキャスター

モーター2個、制御用マイコンが組み込まれ、自身の取り付け位置に応じて自律的に動作制御するキャスター型移動機構



アクティブキャスターの強度と出力の関係
(赤がアクティブキャスター、ブルーがフリーキャスター)

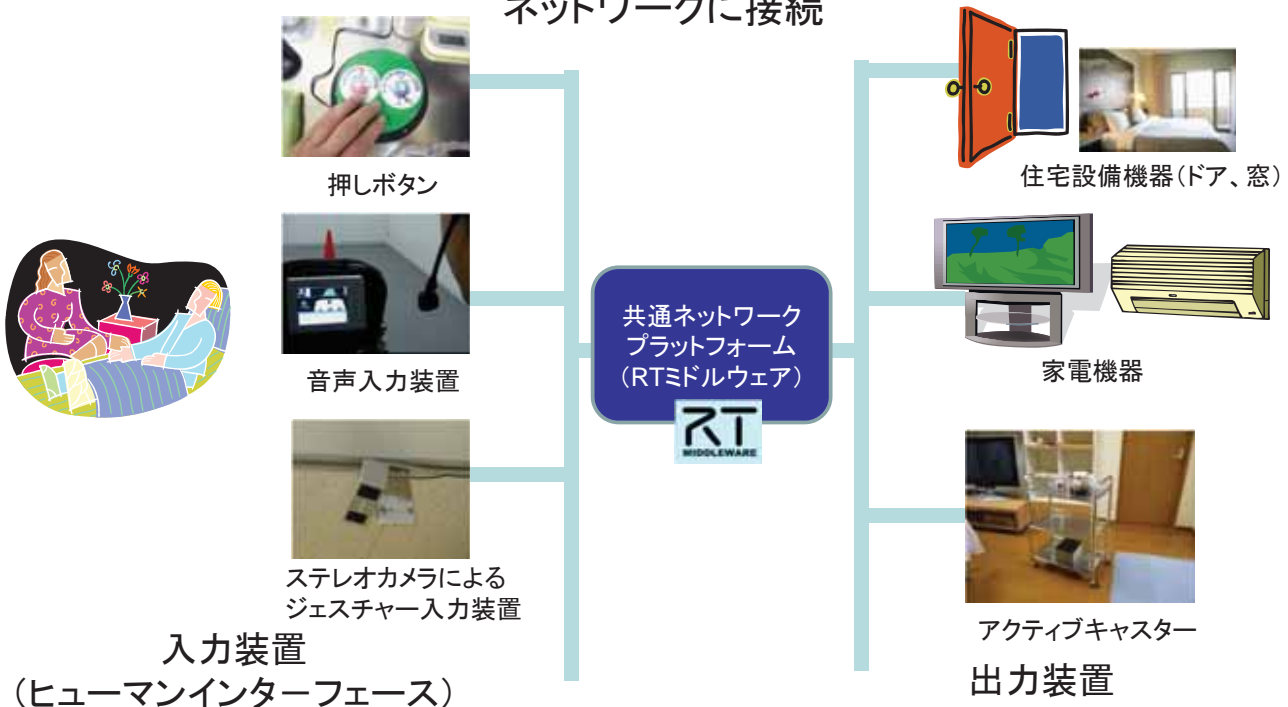
移動物の重量、必要なトルクに応じてフリーキャスター、
アクティブキャスターを自由に使い分けできる。

アクティブキャスターの取り付け



ユーザーレベルで移動したい物に取り付け可能
取り付け位置はアクティブキャスター自身が自動認識

各機能を分類し、共通ネットワークプラットフォームを介して
ネットワークに接続



ヒューマンインターフェースと住宅設備をつなげることで
障害者支援につながる住環境を提示

肢体不自由者を支援する入力機器および出力機器

入力機器

出力機器



ステレオカメラによる
ジェスチャー認識装置



音声入力装置

標準ネットワーク
プロトコル
(RTミドルウェア)



赤外線リモコン



アクティブキャスター

車椅子移動に伴う家具の移動

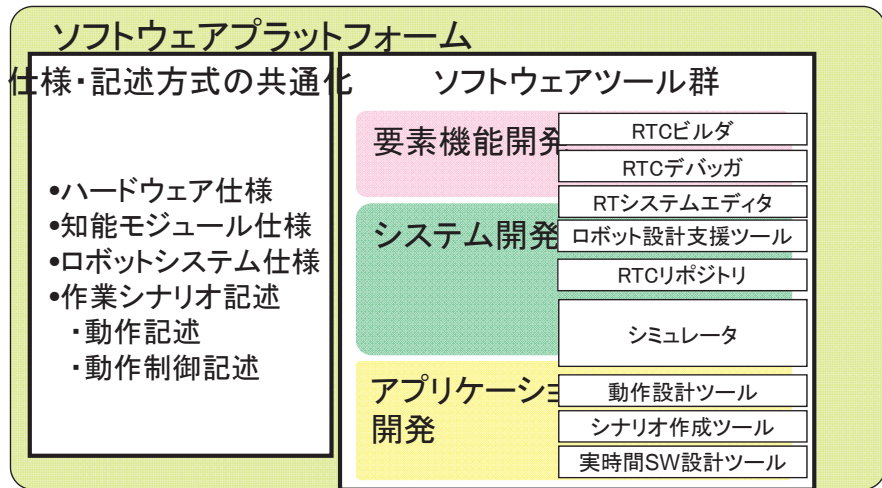


次世代ロボット知能化技術開発プロジェクト

- 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

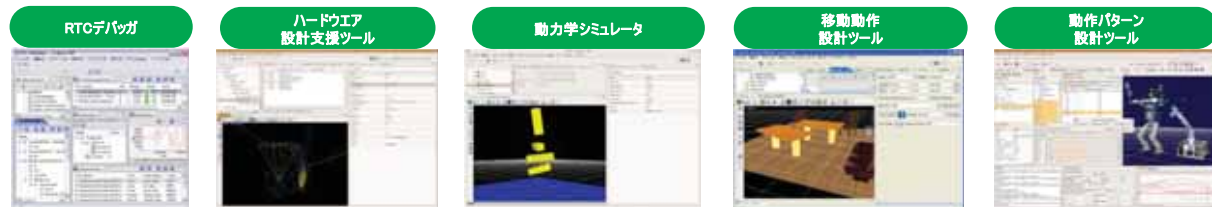
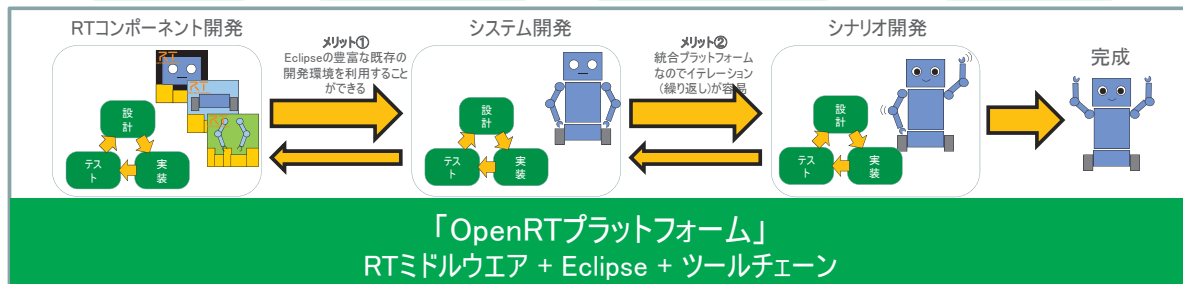
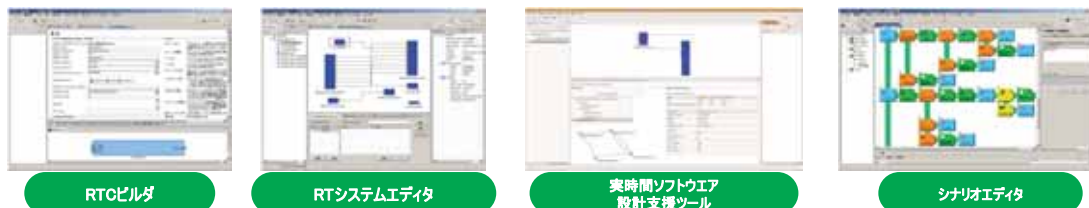


NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)



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



※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。



NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

知能化PJにおけるRTCの開発

- 作業知能
 - 視覚、マニピュレーションを使い、対象物の把持、組立を行う機能
- 移動知能
 - 車輪型ロボットが様々な環境下で安全に移動するための機能
- コミュニケーション知能
 - 人間と対話をベースに、ロボットへタスク教示等を行う機能

オープンソース版視覚(OpenVGR)モジュール群

(独)産業技術総合研究所

概要:

ステレオカメラを利用し、指定された作業対象物の検出結果(3次元位置・姿勢)を共通形式で出力するモジュール群です。

特徴:

◆ステレオカメラ画像出力、認識結果は共通のインターフェース仕様に準拠していますので、他のモジュールと入れ替えが可能です。

インターフェース:

作業対象認識モジュール:

ポート名	入出力信号等の意味
トリガ入力 (サービスポート)	認識を行う物体のID
結果出力 (データポート)	カメラID, 物体ID, 認識候補No, 座標系No, 認識確度, エラーNo, 位置・姿勢行列

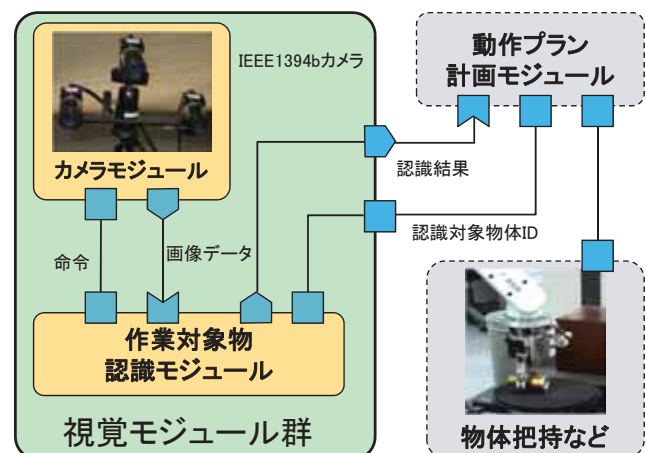
(OpenRTM-aist-1.0.0[C++], Ubuntu 10.04)

ライセンス(公開条件):

ソースファイルをEPLで公開。

産業技術総合研究所が著作権を保持します。

詳細については別途お問い合わせください。



連絡先:

独立行政法人 産業技術総合研究所
 知能システム研究部門 タスクビジョン研究グループ
 email: openvgr-contact@m.aist.go.jp

第1版 2011.06.24

RTC接続例



Capture共通IF

画像出力(データポート)

```

Img.idl
module Img {
    /* vector and matrix type */
    typedef double Vec3[3];
    typedef double Mat44[4][4];

    /* image */
    enum ColorFormat
    {
        CF_UNKNOWN, CF_GRAY, CF_RGB
    };

    struct ImageData
    {
        long width;
        long height;
        ...
    }
}
    
```

Sense共通IF

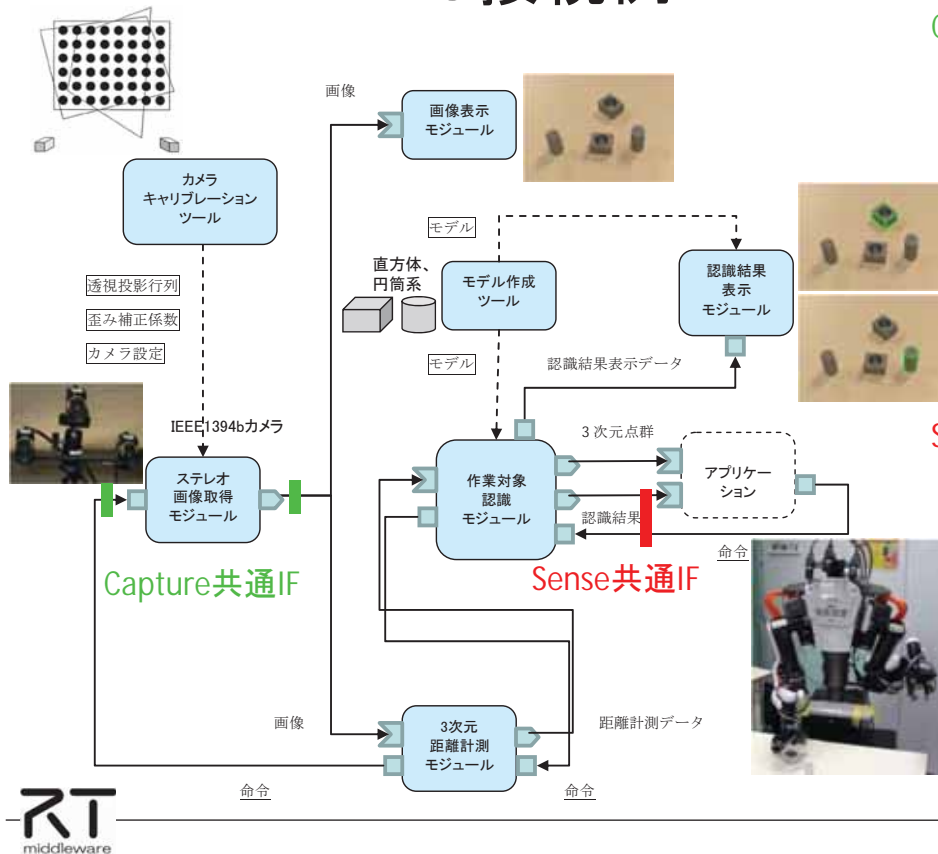
トリガ入力(サービスポート)

```

認識行う物体のID

結果出力(データポート)

【型】 TimedDoubleSeq
【要素数】 20 × n
【配列内容】
カメラID, 物体ID, 認識候補No, 座標系No,
認識精度, エラーNo, 予備1, 予備2,
r000, r001, r002, t0x,
r010, r011, r012, t0y,
r020, r021, r022, t0z,
...
    
```

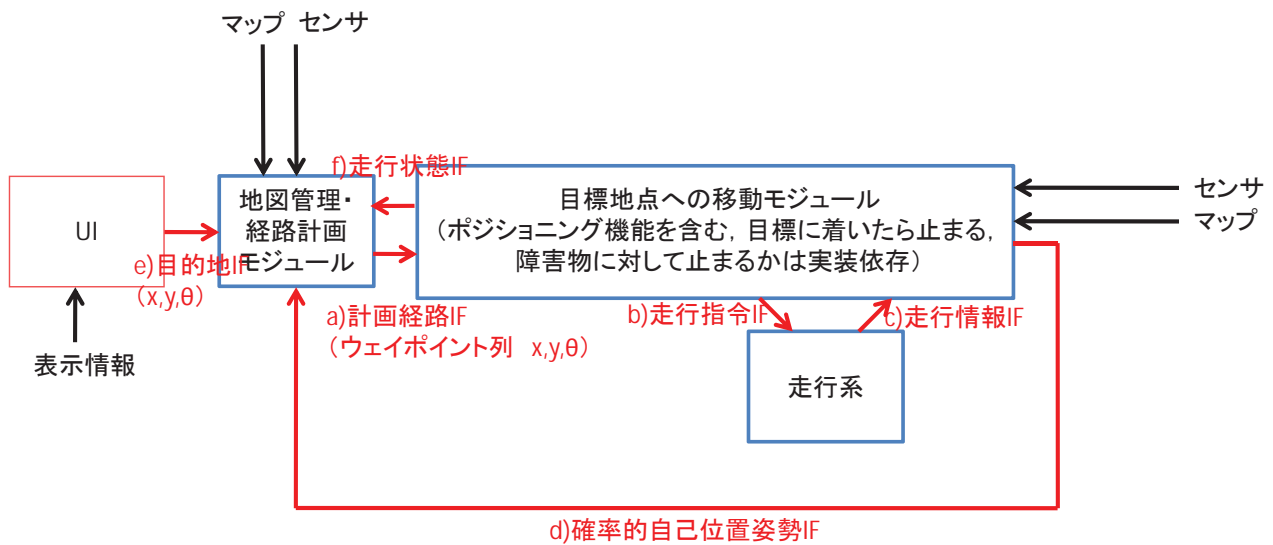


OSS版移動智能モジュール

- 地図のある室内の既知の場所から指定された任意の場所に自律移動する機能
 - オペレータ操作(セグウェイ)
 - 経路計画、軌道追従(東北大)
 - 自己位置姿勢推定(奈良先端大、再利用センター)
 - 障害物検知、衝突回避(東北大、奈良先端大、再利用センター)
 - リファレンスハード2用走行系(産総研、再利用センター)

システム構成

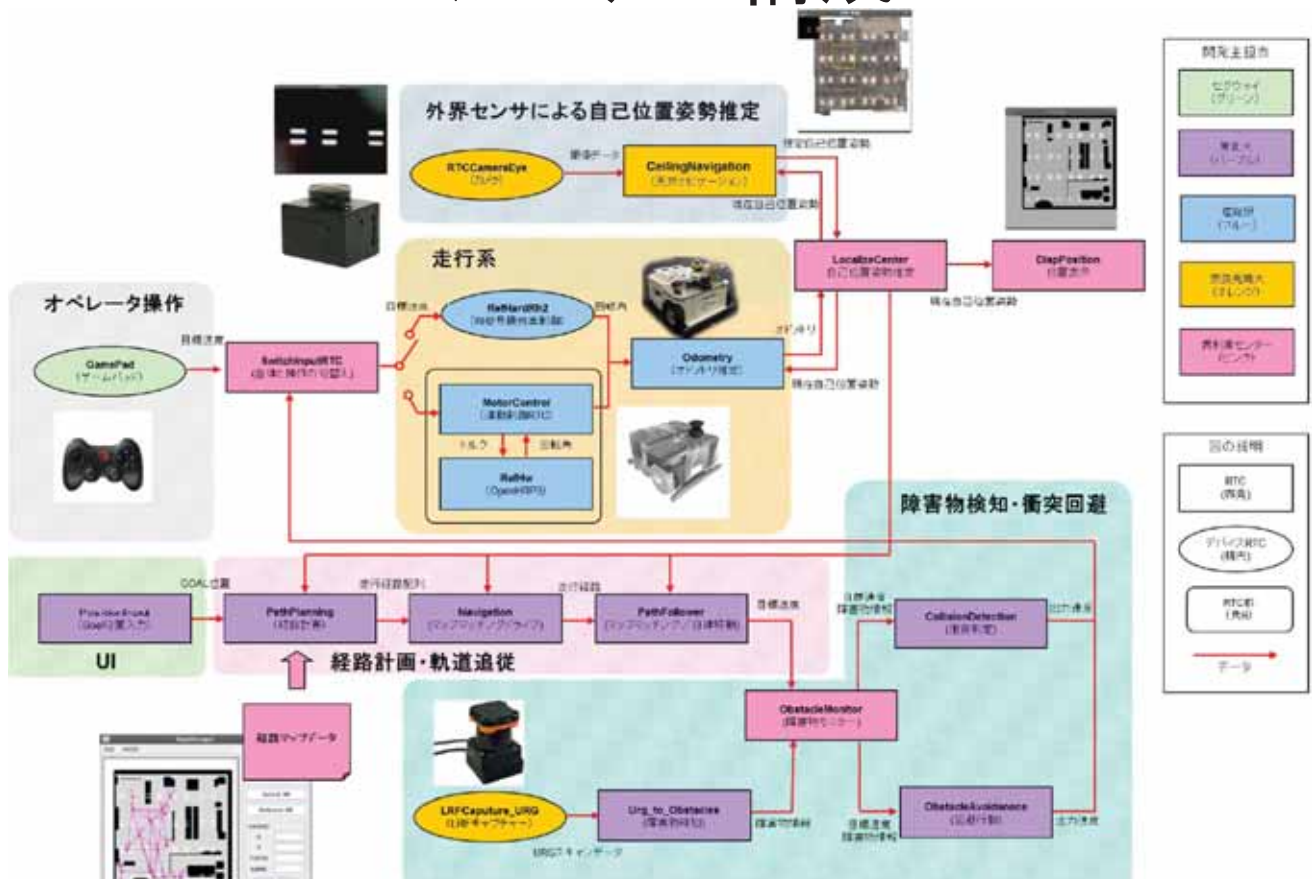
タスク: 地図のある室内の既知の場所から指定された任意の場所に自律移動



※赤い部分のIFを揃える

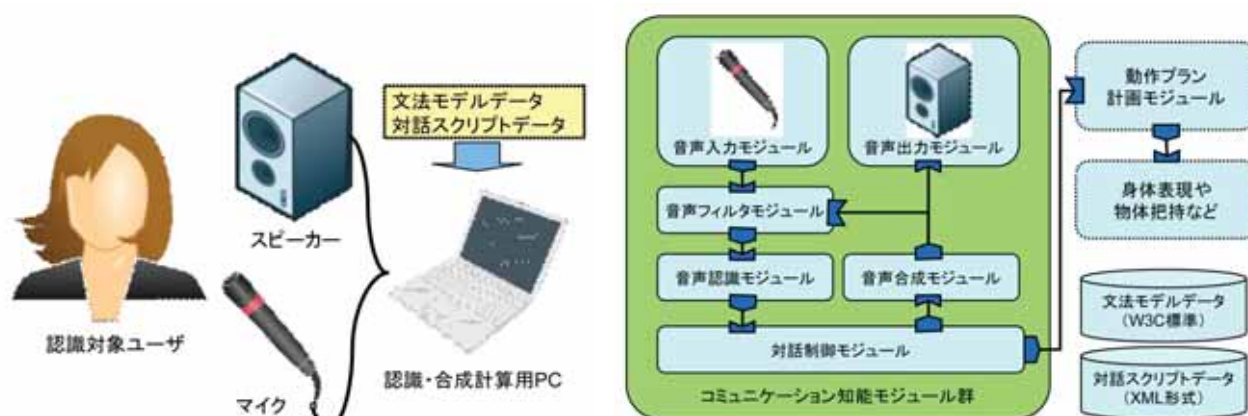
コンポーネントの共通構成及び共通IF型を10/4/9合意した

システム構成



OSS版コミュニケーション知能

- 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
(音声入出力・音声認識・音声合成・対話制御など)



※IFは全て各社共通形式(どのモジュールでも差し替え可能です)

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/>





2011年11月9日～12日
東京ビッグサイトにて

知能化プロジェクトの
成果デモ予定

DAQ-Middleware

実験データ収集システムを容易に構築

概要:

- ◆RTコンポーネント※を拡張したDAQ(データ収集)コンポーネントによりネットワークベースの実験データ収集システムを容易に構築できる

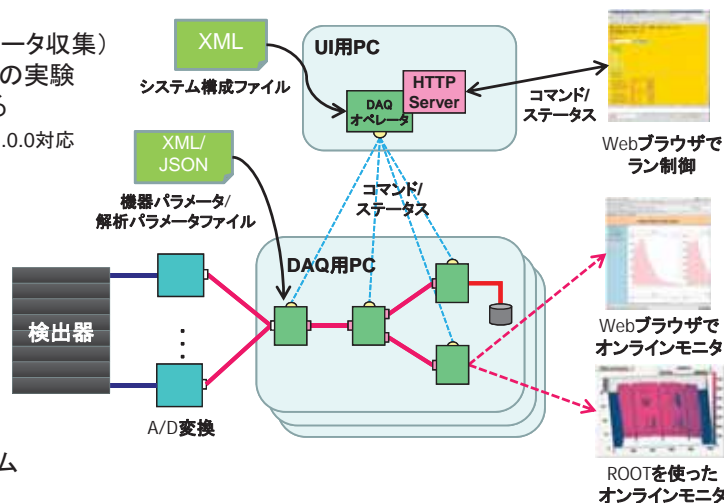
※OpenRTM-aist-1.0.0対応

特徴:

- ◆RTコンポーネントにデータ収集機能を追加したDAQコンポーネントを開発
- ◆DAQオペレータコンポーネントによりDAQコンポーネントを制御
- ◆XMLによるDAQシステム構成
- ◆Webブラウザによるラン制御・モニタ

実績:

大強度陽子加速器施設(J-PARC)の物質・生命科学施設のデータ収集システムに採用され現在稼働中



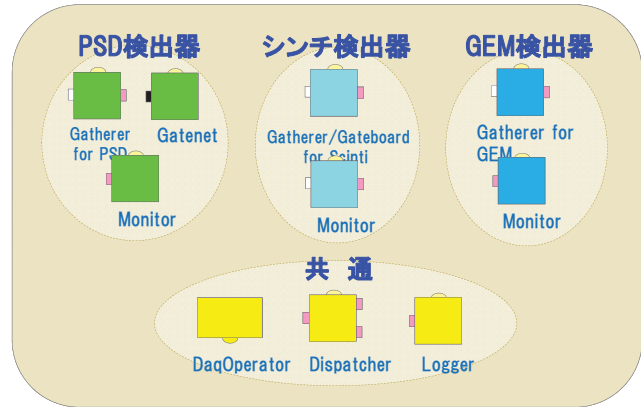
DAQ-MiddlewareはOpen Source Consortium of Instrumentation(Open-It)で開発を行っています。DAQ-Middleware関連の共同研究・開発を募集しています。

連絡先: 高エネルギー加速器研究機構(KEK).
仲吉一男 <kazuonakayoshi@kek.jp>
URL: <http://daqmw.kek.jp/>
<http://onlgw.kek.jp/OSC/>

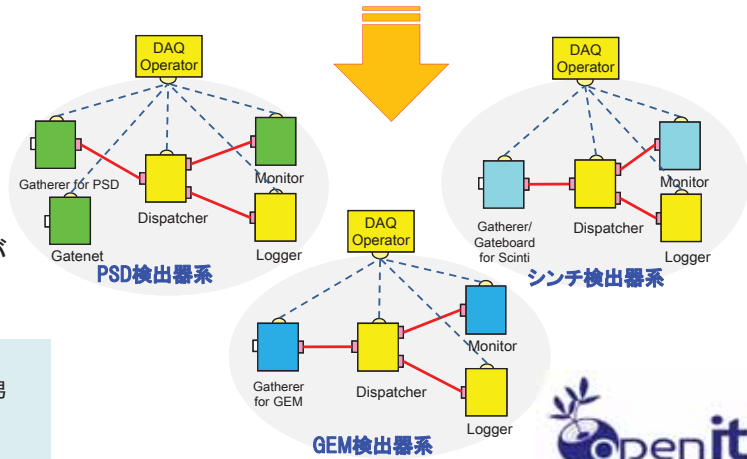




資料提供 JAEA/KEK J-PARCセンター



DAQ-Middlewareは茨城県東海村にある大強度陽子加速器施設(J-PARC)の物質・生命科学実験施設のデータ収集システムに採用され、現在8台の実験装置で稼働中。さらに6台の実験装置で導入が予定されている。また検出器のテストシステムへの適用が行われている。



DAQ-Middleware開発チーム
 KEK 千代浩司、安 芳次、井上栄二、仲吉一男
 AIST 神徳徹雄、安藤慶昭
 広工大 長坂康史、大阪大 味村周平



詳しい情報は...

- OpenRTM-aist公式Webページ
 - Googleで”OpenRTM”で検索
- ダウンロード
 - ソースコード、インストーラ等ダウンロード可能
 - C++版、Python版、Java版(予定)
 - Windows用インストーラ
 - Linux各種パッケージ
- メーリングリスト/フォーラム
 - ユーザコミュニティの情報交換
- 講習会
 - 不定期に開催
 - 実習形式
- マニュアル
 - 随時更新中



<http://www.openrtm.org/>



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

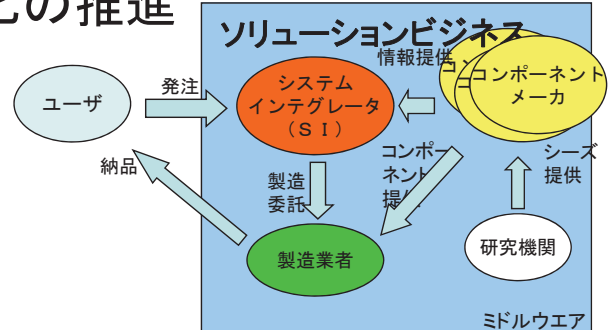
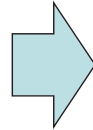
- 生活支援・介護のシステム化
- 農林水産分野のシステム化
- 実験系研究のシステム化
- 医療分野のシステム化
- 交通・物流分野のシステム化
- セキュリティ・防衛分野のシステム化 など

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

RTミドルウェアコンテスト(趣旨)

協賛募集中

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

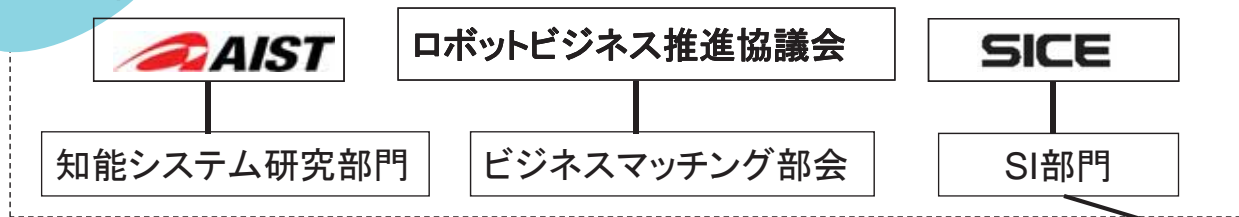


- 有益なコンポーネントやツールの充実を狙う
- これからのロボットソフトウェア開発者に不可欠なRTミドルウェアに精通する技術者の育成を期待

<http://www.openrtm.org/rt/RTMcontest/>

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: エントリー切
随時: 講習会開催
12月: SI2009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

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

個人スポンサーが直接投票

協賛個人賞(奨励賞)

10万円

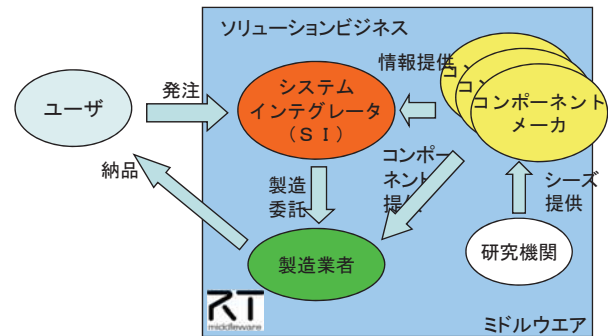
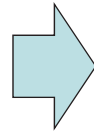
2万円

1万円

まとめ

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

RT
middleware



- 多品種少量生産の製品の開発コストを下げて市場を創成。(技術の再利用、相互接続性)
- 論文で終わっていた、ロボット技術を集積して、高度化を促進する。(技術の共有)

RT
middleware

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

参考

- [OMG のホームページ](http://www.omg.org/)
http://www.omg.org/
- [Robotics -DSIGの ホームページ](http://robotics.omg.org/)
http://robotics.omg.org/
- [OpenRTM-aistのホームページ](http://www.openrtm.org/)
http://www.openrtm.org/
- 「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
- [OMGテクニカルミーティングガイド](http://staff.aist.go.jp/t.kotoku/omg/guidance.html)
http://staff.aist.go.jp/t.kotoku/omg/guidance.html

RT
middleware

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

コンポーネント開発例

- [ホームページのプロジェクトページ](http://www.openrtm.org/openrtm/ja/content/プロジェクト-0) (産総研)
<http://www.openrtm.org/openrtm/ja/content/プロジェクト-0>
- [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2010](http://www.openrtm.org/rt/RTMcatalog2010_v2.pdf) (産総研)
http://www.openrtm.org/rt/RTMcatalog2010_v2.pdf
- [RTミドルウェア技術カタログ2009](http://www.openrtm.org/rt/RTMcatalog2009.pdf) (産総研)
<http://www.openrtm.org/rt/RTMcatalog2009.pdf>

RTミドルウェアによる開発解説

OpenRTM YouTube チャンネル

<http://www.youtube.com/user/OpenRTM>

(rtshell入門など)

東京大学のRTM-ROS総合運用プロジェクト

<http://code.google.com/p/rtm-ros-robotics/>

– ChoreonoidとOpenHRIについて(原功)

<http://rtm-ros-robotics.googlecode.com/svn/wiki/hara110706.pdf>

– ヒューマノイドロボットHRP-4におけるOpenRTM-aist
応用事例(金広文男)

<http://rtm-ros-robotics.googlecode.com/svn/wiki/20110713Kanehiro.pdf>

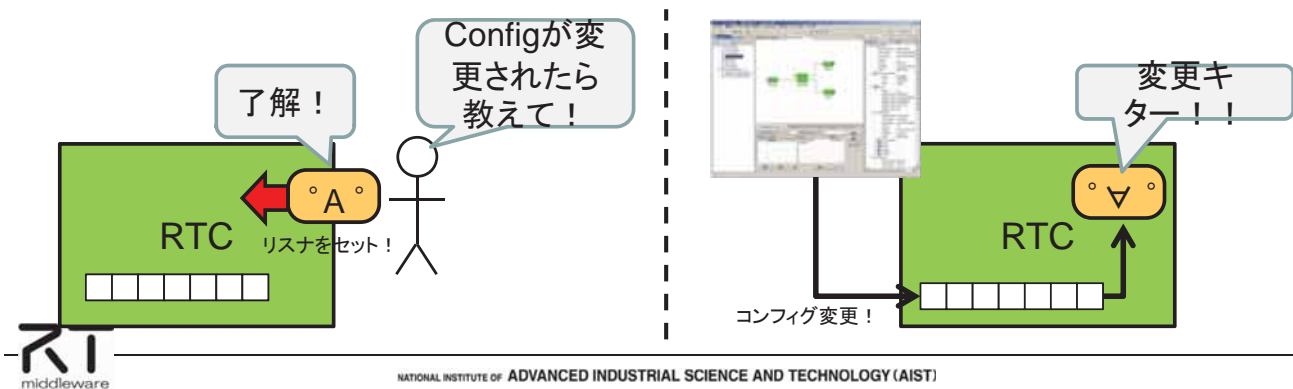
OpenRTM-aist-1.1.0の新機能

OpenRTM-aist-1.1.0の新機能

- コールバック機能強化
- SDOサービスフレームワークの導入
- 内部APIの追加
- リアルタイムECの提供
- 雑多なバグフィックス

コールバック機能

- コンポーネント内で発生する様々なイベントに対してあるアクションを行う機能
 - 例: コンフィグパラメータが変更されたら、画面の表示を更新する
 - 例: ポートが接続されたら、実際に計算を行ってデータを出力、等
- 開発者が予めセットしておいたリスナオブジェクトの特定の関数がイベント発生時に呼ばれる。



イベントタイプ(1)

タイプ	概要	ヘッダ
コンポーネント本体に関連するイベント		
PreComponentActionListenerType	ComponentAction (onInitialize()等)の実行直前に発生するイベント	ComponentActionListener.h
PostComponentActionListenerType	ComponentAction (onInitialize()等)の実行直後に発生するイベント	ComponentActionListener.h
PortActionListenerType	ポートの追加、削除イベント	ComponentActionListener.h
ExecutionContextActionListenerType	実行コンテキストのアタッチ、でタッチなどのイベント	ComponentActionListener.h
コンフィギュレーションパラメータに関連するイベント		
ConfigurationParamListenerType	コンフィグパラメータの更新操作時のイベント	ConfigurationListener.h
ConfigurationSetListenerType	コンフィグパラメータセットの操作時のイベント	ConfigurationListener.h
ConfigurationSetNameListenerType	コンフィグパラメータセットの操作時にイベント	ConfigurationListener.h

イベントごとに、イベントタイプやリスナクラスの型が異なる

イベントタイプ(2)

タイプ	概要	ヘッダ
ポート内部の振る舞いに関連するイベント		
InPort read系 (旧式)	InPortに対してreadを行う際に発生するイベント	PortCallback.h
OutPort write系 (旧式)	OutPortに対してwriteを行う際に発生するイベント	PortCallback.h
PortConnectListenerType	ポートの接続時の各種処理に関するイベント (notify_(dis)connect(), unsubscribeinterfaces()等)	PortConnectListener.h
PortConnectRetListenerType	ポートの接続時の各種処理に関するイベント (connect_next(), subscribeinterfaces(), 接続切断完了通知)	PortConnectListener.h
ConnectorDataListenerType	データポートのコネクタ内のイベント (bufferフル, send/receivedなどの完了やエラー通知)	ConnectorListener.h
ConnectorListenerType	データポートのコネクタ内のイベント (buffer空, 接続・切断などの完了やエラー通知)	ConnectorListener.h

どのようなイベントがあるかは、クラスリファレンスかイベントごとのヘッダファイル内のドキュメントを参照

使い方

1. 利用するイベントを決める
 - 例: Configurationの更新
2. イベントタイプを調べる
 - 例: ConfigurationParamListenerTypeのON_UPDATE_CONFIG_PARAMを利用
3. リスナの基底を継承してファンクタを実装
 - 例: ConfigurationParamListenerを継承してMyConfigUpdateParamを実装
4. onInitialize()などでリスナを登録
 - 例: addConfigurationParamListener()
5. イベント発生時にリスナがコールされる

コンポーネントのヘッダファイル

```
#include <rtm/ConfigurationListener.h>

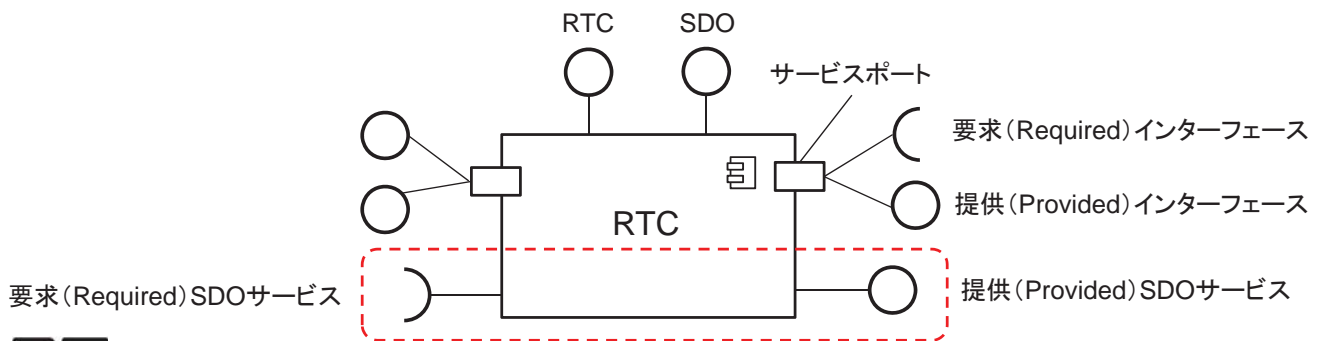
class MyConfigUpdateParam
: public ConfigurationParamListener
{
    virtual void operator()(const char* config_set_name,
                           const char* config_param_name)
    {
        std::cout << config_set_name << “の”
                  << config_param_name
                  << “が更新されました” << std::endl;
    }
}
```

コンポーネントの実装ファイル

```
RTC::ReturnCode_t ConsoleIn::onInitialize()
{
    // 中略
    addConfigurationParamListener(
        ON_UPDATE_CONFIG_PARAM,
        new MyConfigUpdateParam ());
}
```

SDOサービス

- SDO(Super distributed object)
 - RTCのベースとなっているOMG標準
- SDOサービス:ポートに属さないサービス
 - コンポーネント一般に関係するサービスを提供、要求する
 - 例:ロギング、デバッグ、パフォーマンス計測等
 - 特定のRTCの特定の機能(例:ナビゲーションアルゴリズム)に関するサービスはサービスポートで提供・要求する



新たなAPI

EC(Execution Context:実行コンテキスト)に関する操作

関数	意味
ExecutionContext_ptr getExecutionContext(RTC::Uniqued ec_id);	現在のECを取得
double getExecutionRate(RTC::Uniqued ec_id);	現在のECの実行周期を取得
ReturnCode_t setExecutionRate(RTC::Uniqued ec_id, double rate);	現在のECの実行周期をセット
bool isOwnExecutionContext(RTC::Uniqued ec_id);	現在のECが自身のECかどうか
ReturnCode_t deactivate(RTC::Uniqued ec_id);	現在のECでactive化する
ReturnCode_t activate(RTC::Uniqued ec_id);	現在のECで非active化する
ReturnCode_t reset(RTC::Uniqued ec_id);	現在のECでresetする

これらの関数は原則RTObject::onXXX() 関数内でのみ実行可能(詳細はリファレンスマニュアルを参照のこと)

リアルタイムECの提供

- 2つのリアルタイム実行コンテキスト(EC)
- ArtLinuxEC
 - ARTLinux用の実行コンテキスト
 - 1ms(orそれ以上)の精度でリアルタイム実行が可能
 - Ubuntu用のkernel debパッケージが利用可能なので、インストールし、rtc.confで利用するECをArtLinuxECに指定すれば利用可能
- PreemptEC
 - LinuxのPreemption Patchedなkernelのリアルタイム機能を利用したEC
 - 1ms程度の精度でリアルタイム実行が可能
 - Ubuntu等では標準でrt-kernelとして提供されている

雑多な機能追加

- deb,rpmパッケージ作成機能
 - configure, cd packages, make でパッケージ作成
- データポートのPortProfile内データ型をIFRに変更
- RtORB(産総研CORBA)正式対応、Cygwin対応
- クラスリファレンスの拡充
- Version.txtの導入
- ロギング時のタイムスタンプを μ sまで表示
 - See rtc.conf.sample

雑多なバグフィックス

- 標準ECのsleep時間計算方法の修正
 - 20ms以上の精度が必要な場合は
PreemptiveECかArtLinuxECを利用してください
- ある条件でコンパイルしたrtm-namingが動作しない問題の修正
- Rtcdでゾンビプロセスが残る問題の修正

Java版、Python版、ツール

Java、Python版

- 8月末を目処に1.1.0を順次リリース
- 追加機能についてはC++と同じ
 - APIのシグニチャなどもC++と極力同じに
- リアルタイム機能は提供されない

ツール

- Eclipse更新サイトを準備
- オンラインアップデートが可能に