



# RTミドルウェアの出口戦略と KEKさんとの共同研究

産業技術総合研究所  
知能システム研究部門

神徳徹雄

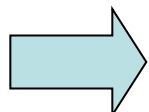
<http://staff.aist.go.jp/t.kotoku/>



# 技術背景

## 活発なロボット技術の研究開発

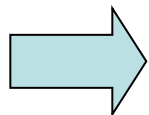
- 個々の技術課題の解決
- 事例紹介的で成果の共有が進んでいない



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)

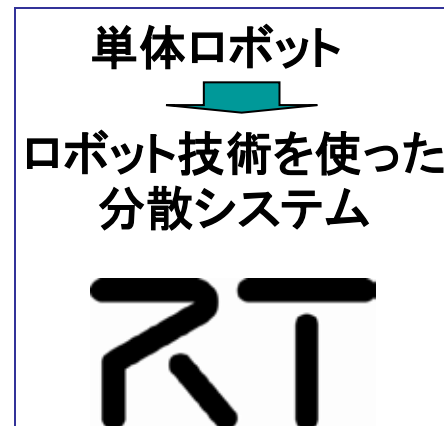
ロボットからRTへ、オーダーメイド型の  
 ロボット産業への変革を提言



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)

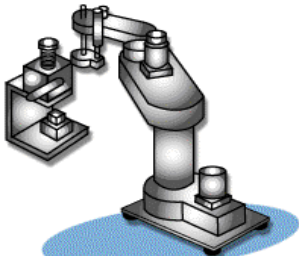
### ロボット産業の現状

#### 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

- ロボット開発の活性化
- 産業競争力の強化

ブレークスルー技術

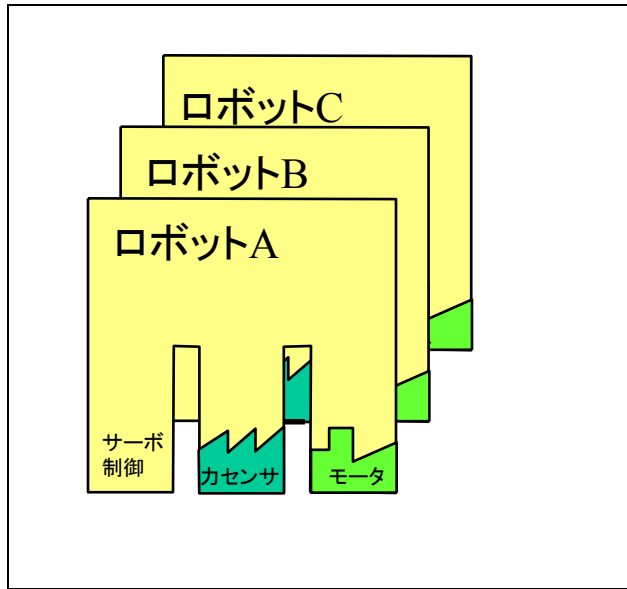
技術共有を促進する

ロボット共通基盤技術の確立

ロボット産業の産業構造変換

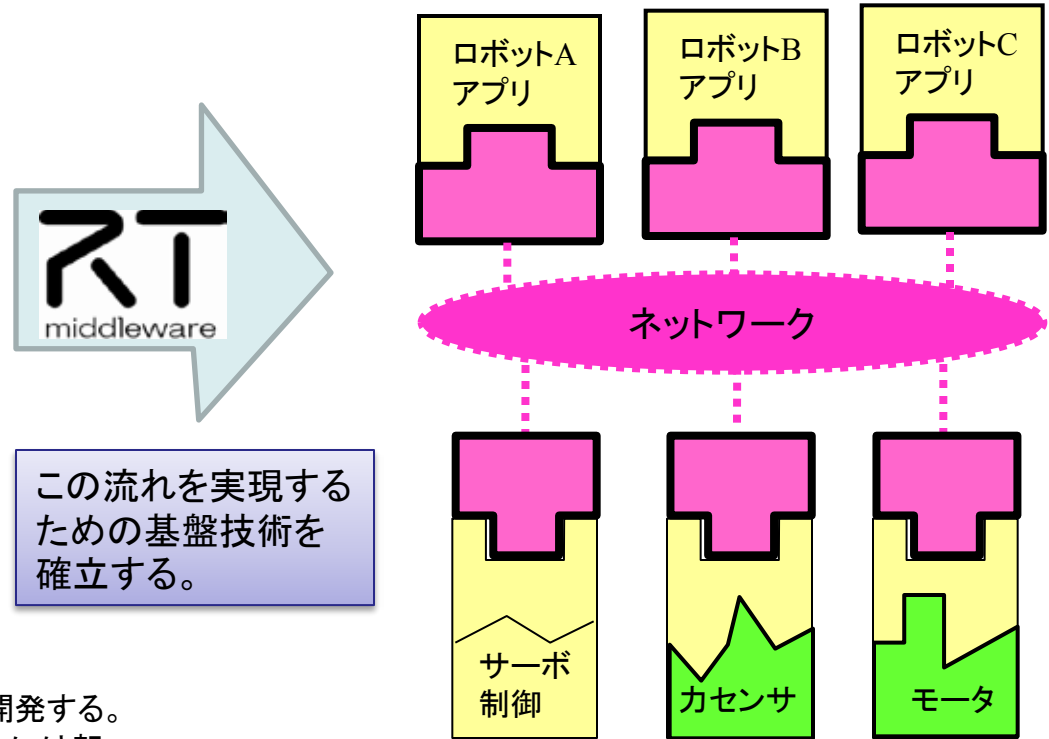
# RTミドルウェアプロジェクト(2002-2004)

## 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ことができる。
- ・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。

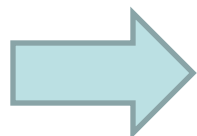
# なぜ標準化が必要か？

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



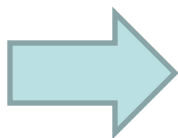
コンポーネントモデルの標準化

# 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)

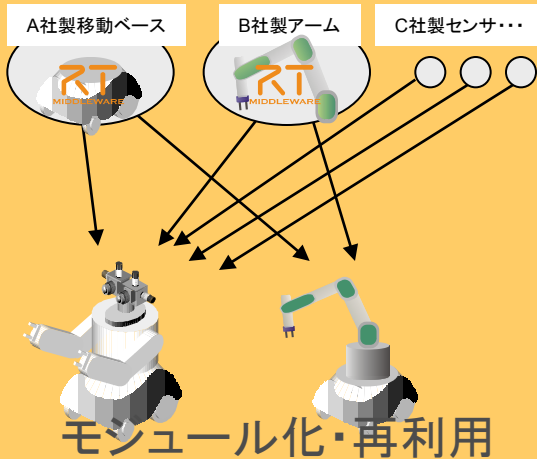


機能モデルごとの共通化・標準化  
(技術革新に合わせた拡張性をもたせて)

RTミドルウェアの目的

# モジュール化による問題解決

## コストの問題



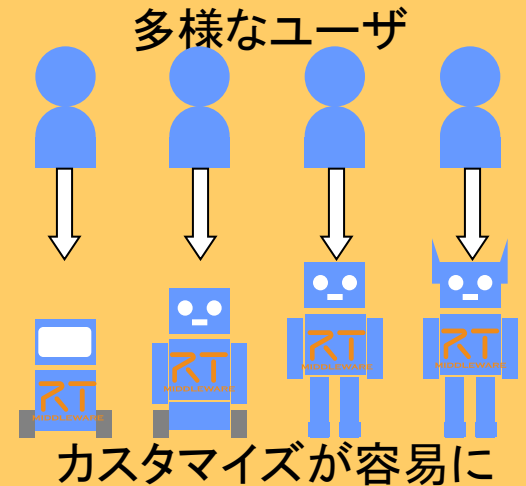
ロボットの低コスト化

## 技術の問題



最新技術を利用可能

## ニーズの問題

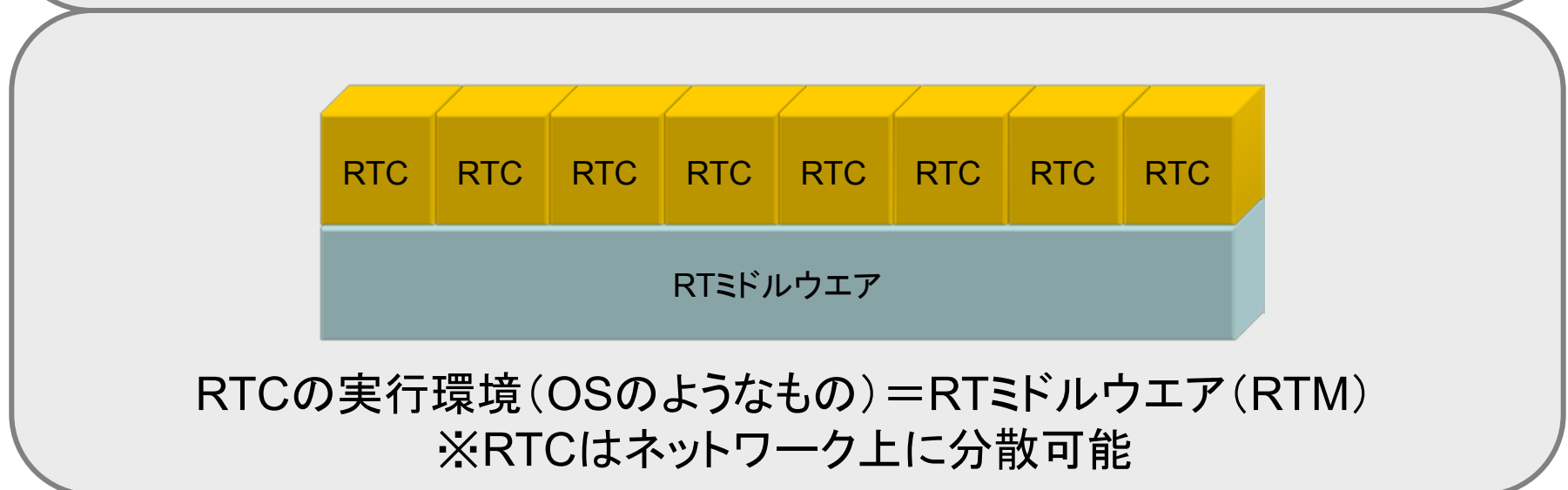
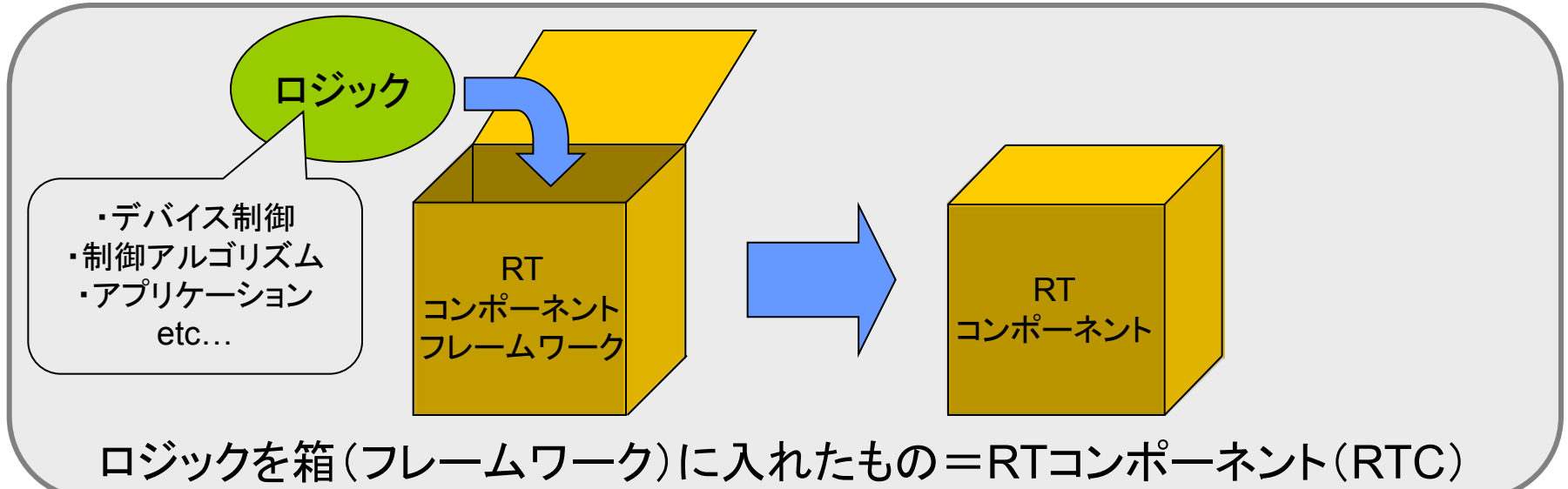


多様なニーズに対応

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



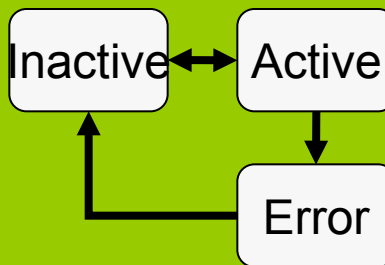
# RTミドルウェアとRTコンポーネント



# 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



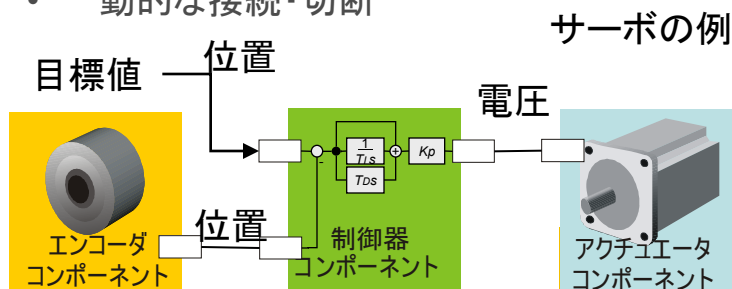
### 複合実行



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

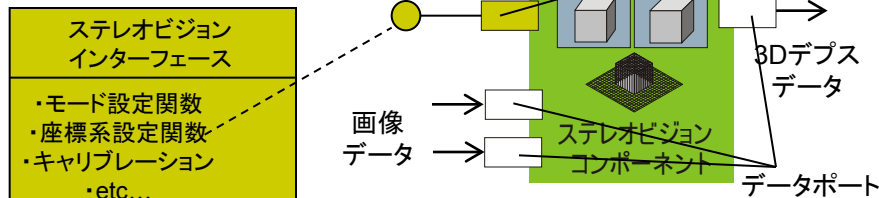
- データ指向ポート
- 連続的なデータの送受信
- 動的な接続・切断



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

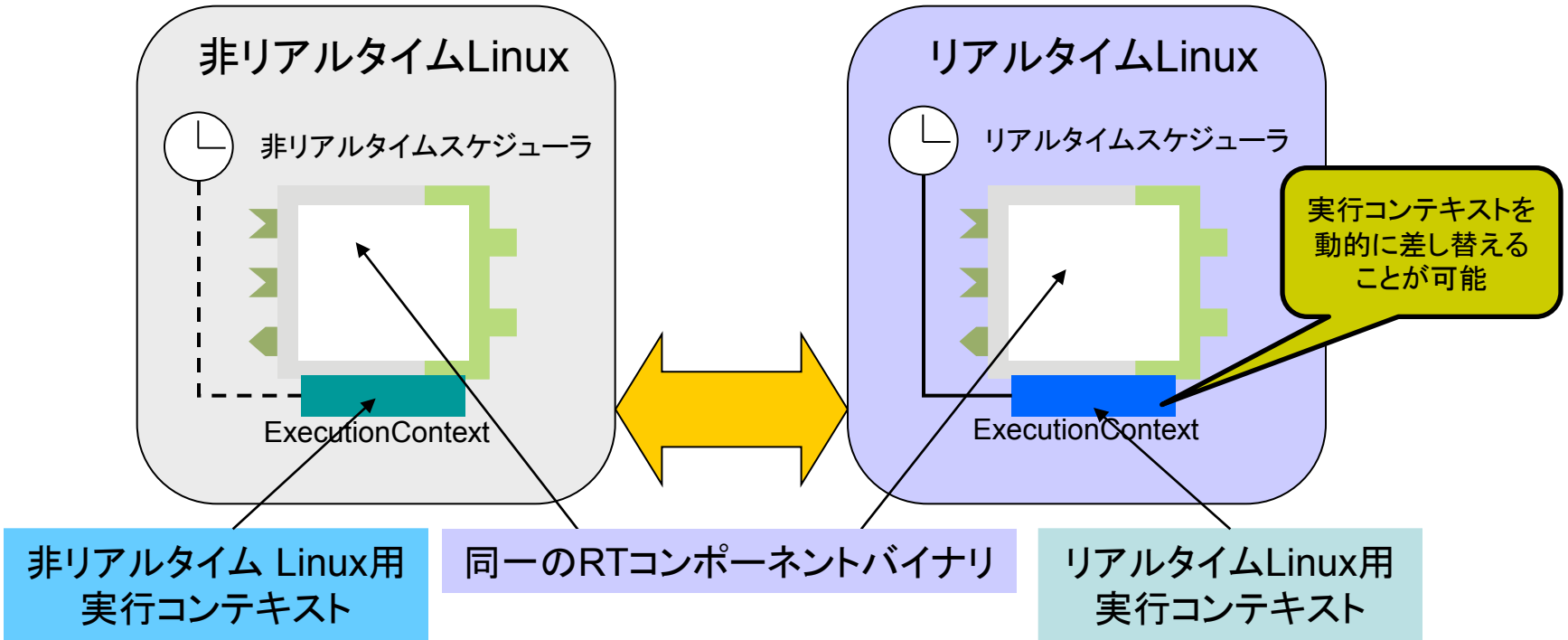
## コンフィギュレーション

- 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

セット名	名前				
	値				
セット名	名前				
	値				

# リアルタイム実行コンテキスト

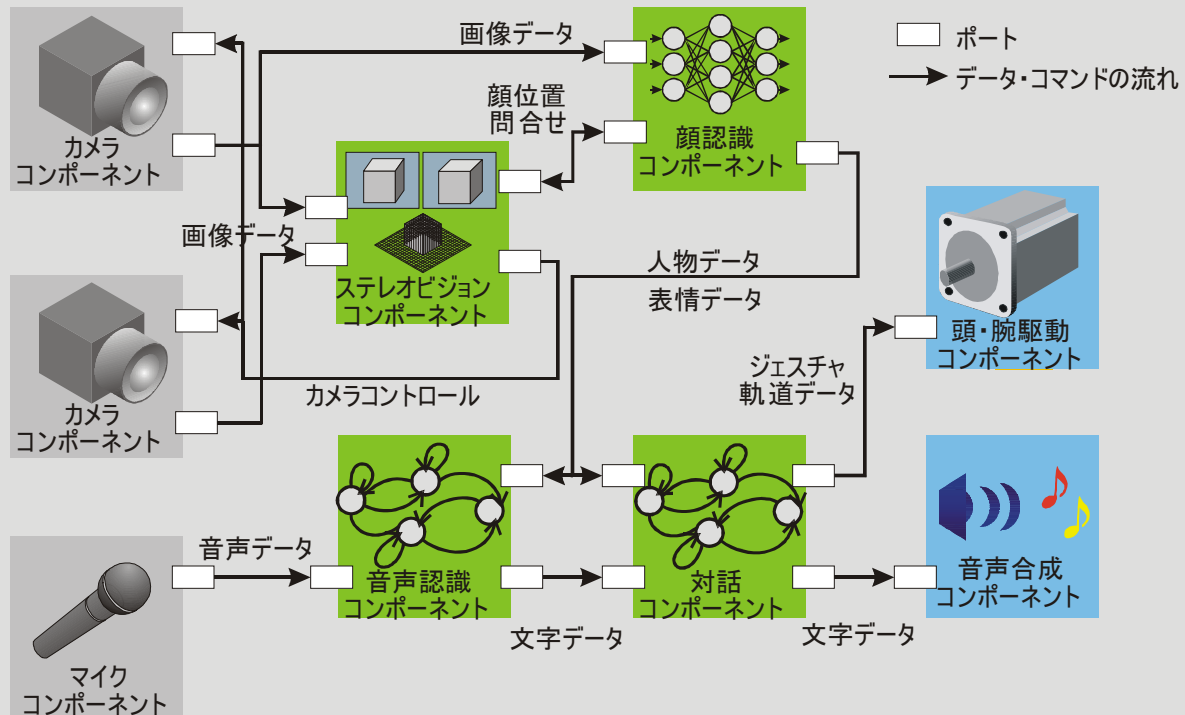


非リアルタイムLinux環境で作られたRTコンポーネントを再コンパイルせずにリアルタイムLinux上でリアルタイム実行可能

# RTCの分割と連携



ロボット体内のコンポーネントによる構成例



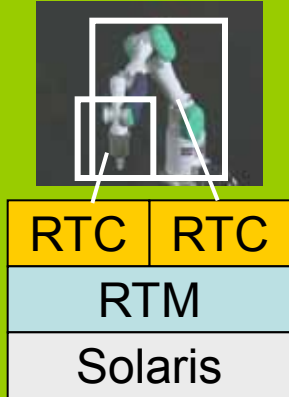
(モジュール)情報の隠蔽と公開のルールが重要

# RTミドルウェアによる分散システム

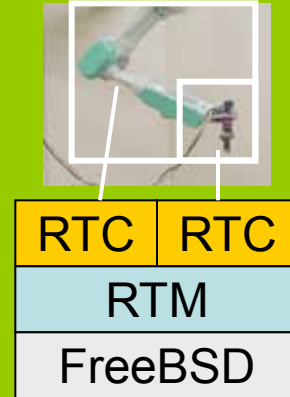
RTMにより、ネットワーク上に分散するRTCをOS・言語の壁を越えて接続することができる。

ネットワーク

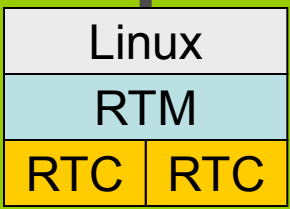
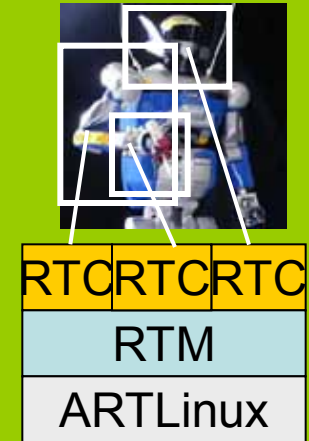
ロボットA



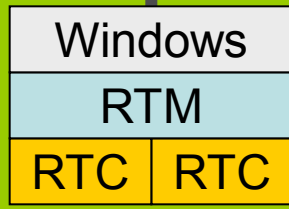
ロボットB



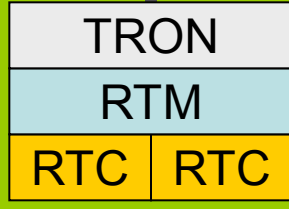
ロボットC



アプリケーション



操作デバイス

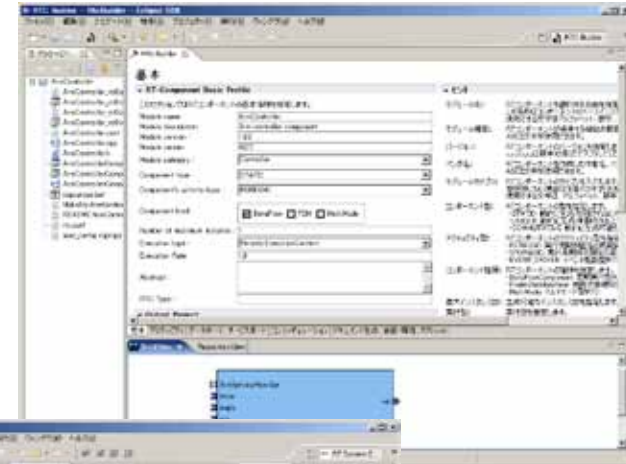


センサ

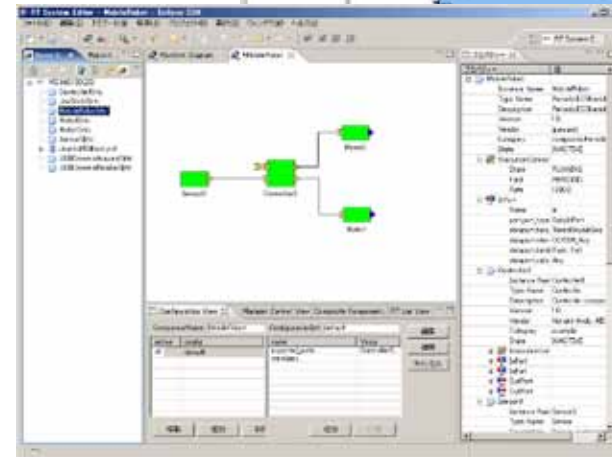
RTC同士の接続は、プログラム実行中に動的に行うことができる。

# 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

# 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, ec_id):
    
```

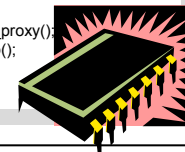


### RTC-Lite source for PIC C

```

#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, uLTRON, T-Kernel, VxWorks
- 言語:
  - C++ (1.0.0), 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

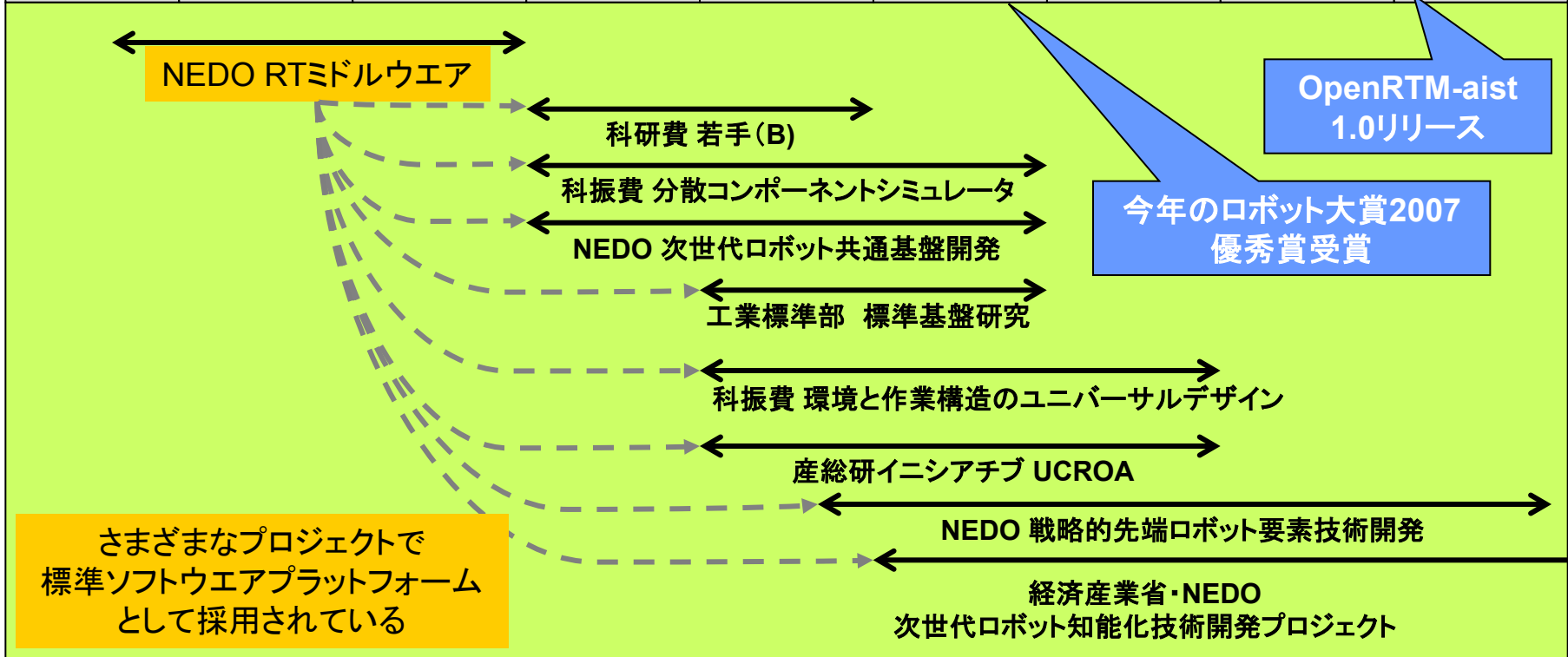
- 共通コンポーネントフレームワークを提供
  - OMG標準
  - コールバックベースの枠組み、共通状態マシン、複合化に対応
  - 大部分のコード生成を自動化
- 多言語対応
  - C++, Java, Python, .NET (by SEC)
- 多様なOSへのネイティブ対応
  - FreeBSD, Linux, Mac OS X, Windows
  - 試験的: TOPPERS, T-Kernel, VxWorks
- ツールの提供
  - Eclipseベースのツール群 (RTCB, RTSE)
  - コマンドラインツール群 (rtchell)
- デュアルライセンス (EPL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

OpenRTM-aist 0.2.0 リリース    OMG RTC仕様 初期草案提出    OMG RTC 標準仕様採択    OpenRTM-aist 0.4.0リリース    OMG RTC Spec. 公式リリース

RTM0.1

								年度
2002年	2003年	2004年	2005年	2006年	2007年	2008年	2009年	2010年



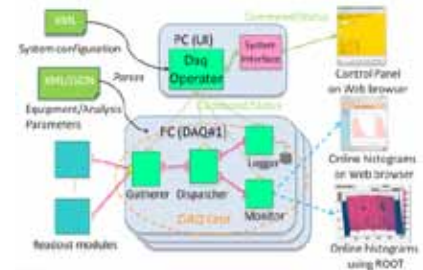
# 多様な実装

- 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
  - テクノロジックアートの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

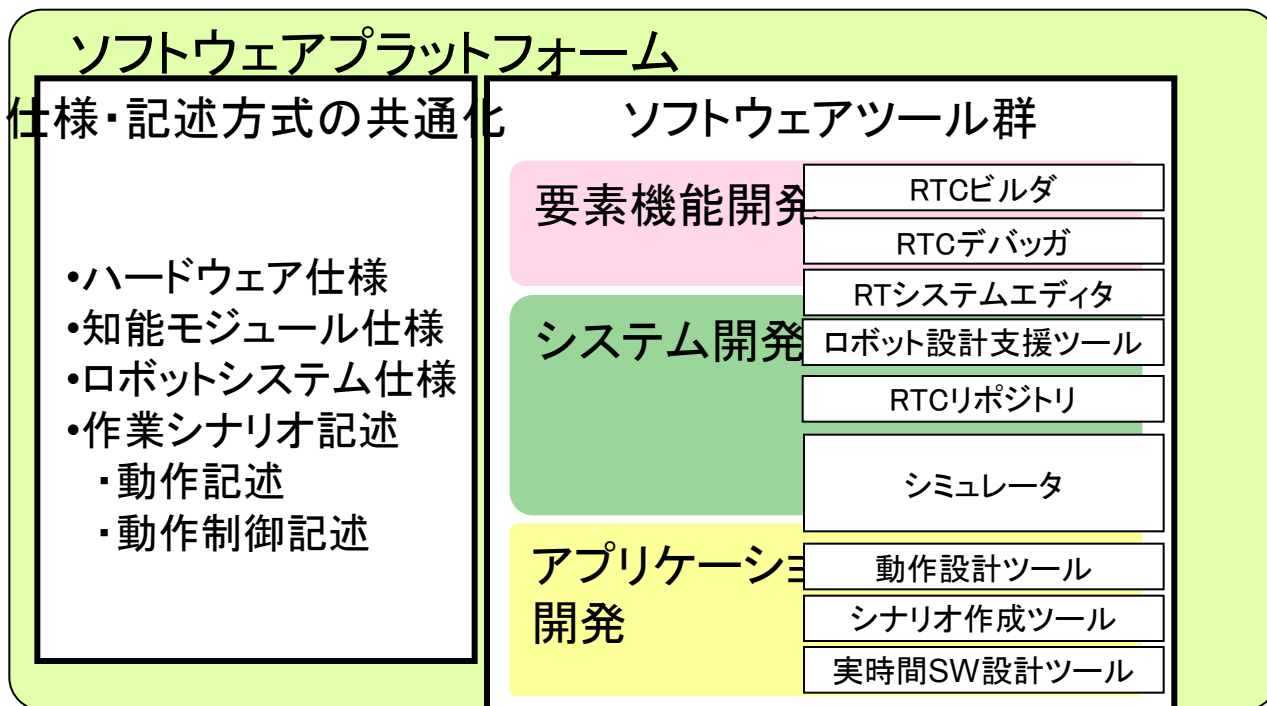
- UMLとRTミドルウェアによるモデルベースロボットシステム開発
- 水川 真、大原 賢一、坂本 武志 著

# 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



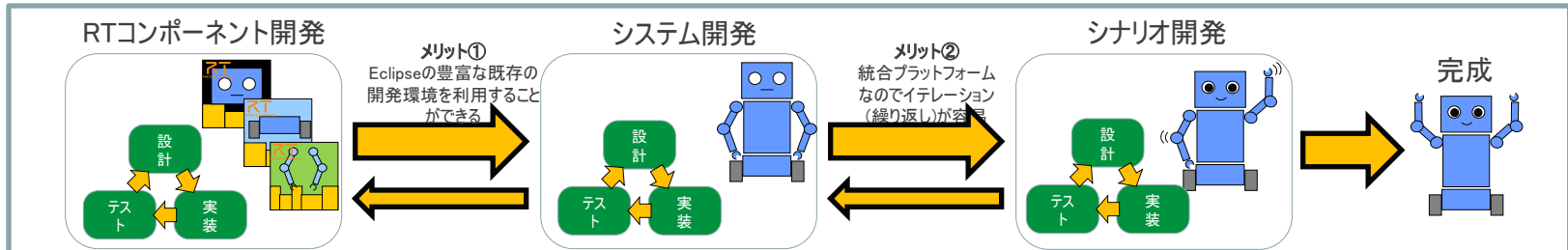
RTシステムエディタ



実時間ソフトウェア設計支援ツール



シナリオエディタ

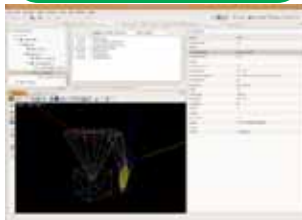


## 「OpenRTプラットフォーム」 RTミドルウェア + Eclipse + ツールチェーン

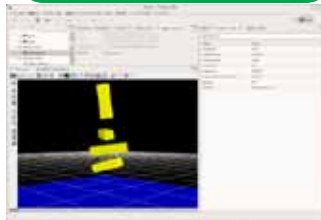
RTCデバッガ



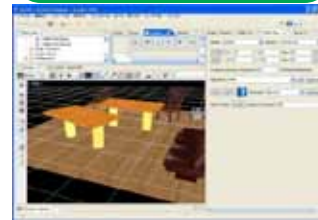
ハードウェア設計支援ツール



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。

# OpenRTM-aist-1.0.0の新機能

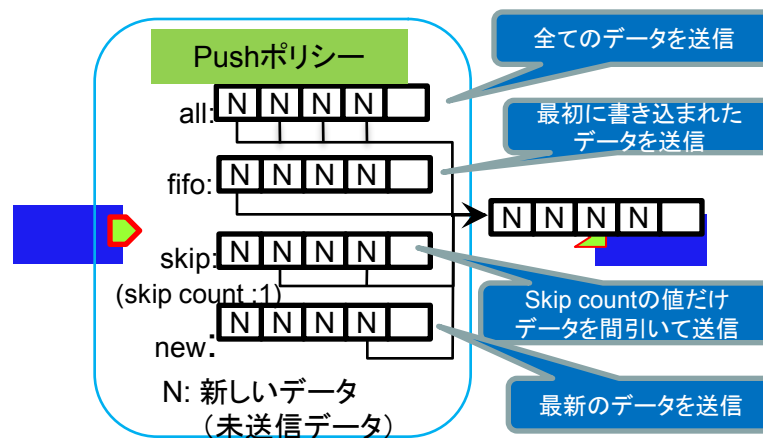
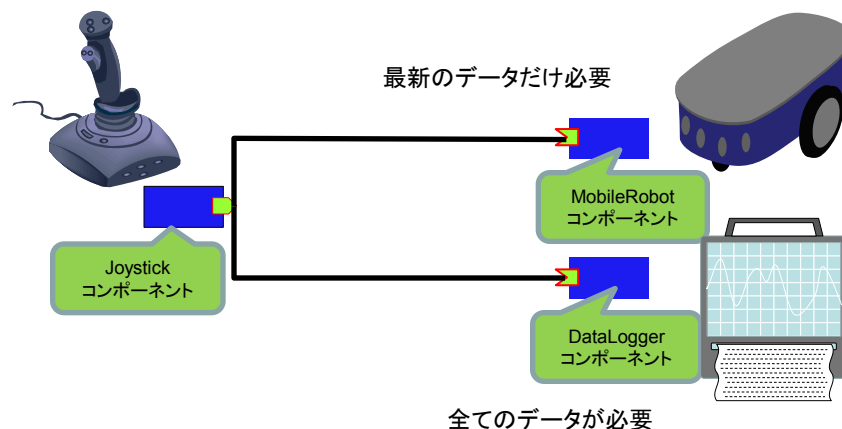


# OpenRTM-aist-1.0.0の新機能

- 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
- シリアルライズ機構の導入
  - 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

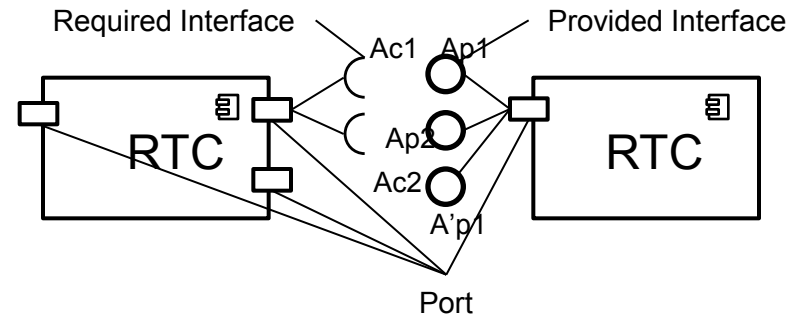


詳細は

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

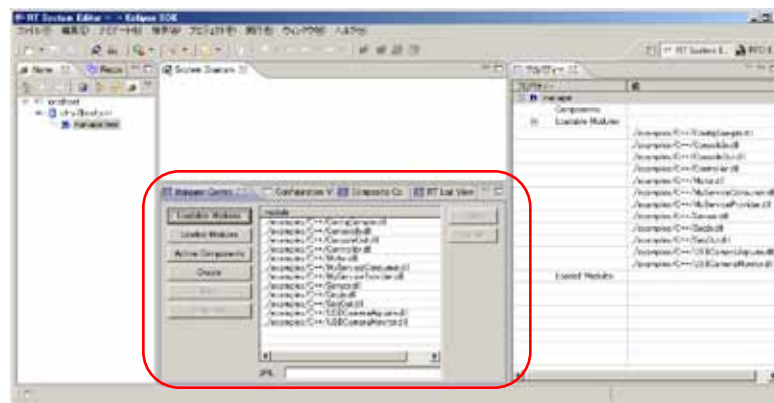
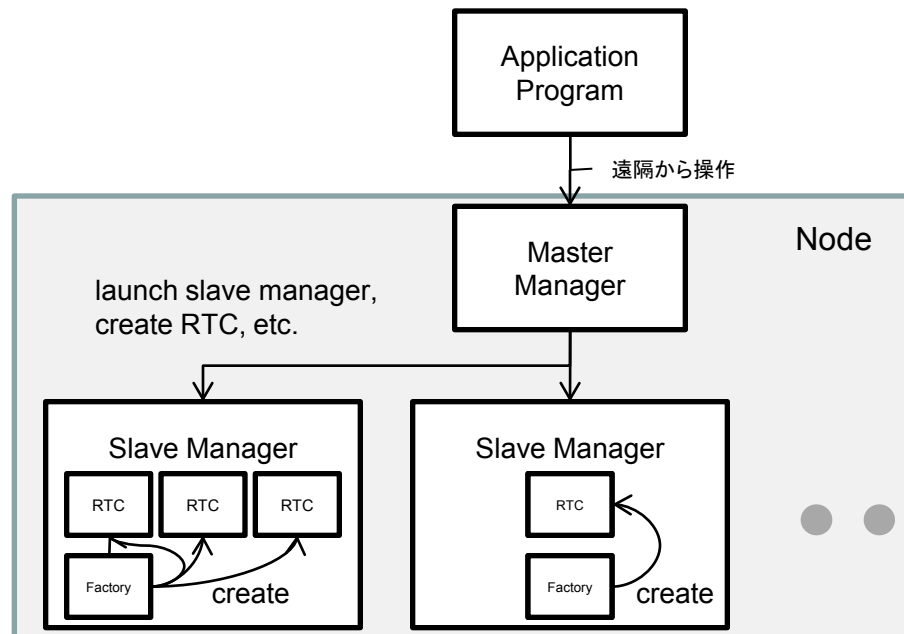


詳細は

次世代ロボット知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



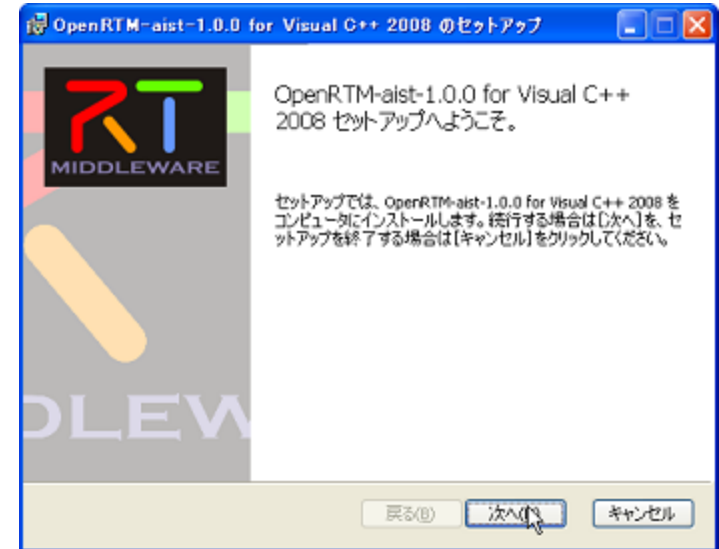
マネージャコントロールビューの追加

詳細は

次世代ロボット知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



# その他

- **coil** (Common Operating System Infrastructure Layer)
  - ACEと同等の機能を実装 (OpenRTMに必要なものののみ)
  - uITRON (TOPPERS) に対応 (0.4 + TOPPERS で動作を確認)
  - OS毎にソースを分割
  - 対応プラットフォーム (Linux、Windows、uITRON、ACE)
- ツールユーザビリティの向上
  - RTCBuilder
    - ヒントの追加、GUIの見直し
  - RTSystemEditor
    - ショートカットの追加
    - GUIの見直し

# 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





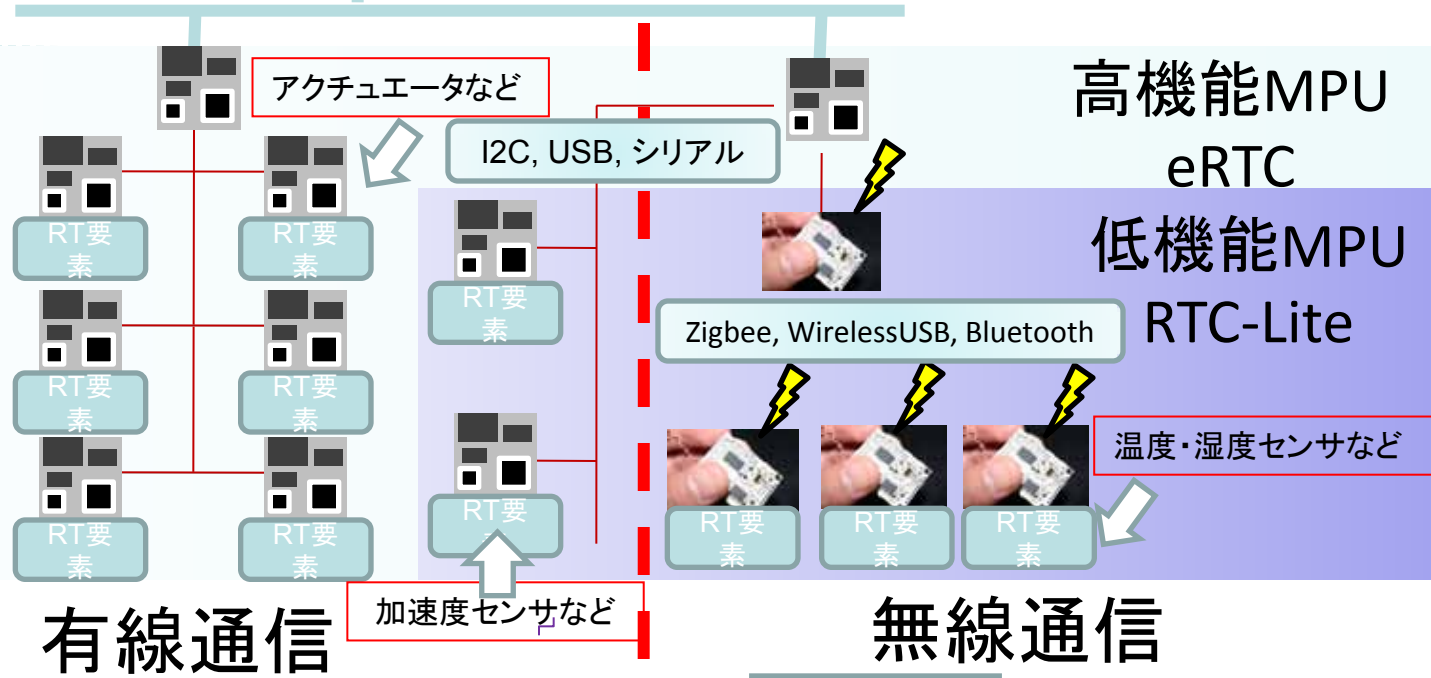
# RTミドルウェアのまとめ

- 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

# 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



高機能MPU  
eRTC  
動作の安定性  
コスト(¥2000程度)

低機能MPU  
RTC-Lite  
コスト(¥500程度)

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

# 詳しい情報は...

- OpenRTM-aist公式Webページ
  - Googleで”OpenRTM”で検索
  - <http://www.is.aist.go.jp/rt/OpenRTM-aist/>
- 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各種ディストリ用パッケージ
- メーリングリスト
  - 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



# 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ミドルウェアの目指すもの

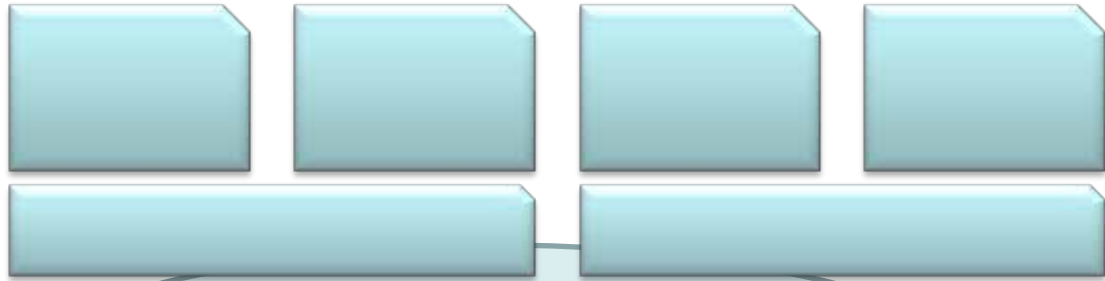
パターン化  
国際連携  
標準化

シミュレーション  
技術

ソフトウェア技術

開発支援ツール

モジュール化



応用技術  
(ドメイン毎のパ  
ターンの再利用、  
開発支援ツール)

基盤技術  
(フレームワークの  
共通化、標準化)

実装技術  
(相互運用、  
適用範囲拡大)



機能安全



開発プロセス標準



モデルベース開発



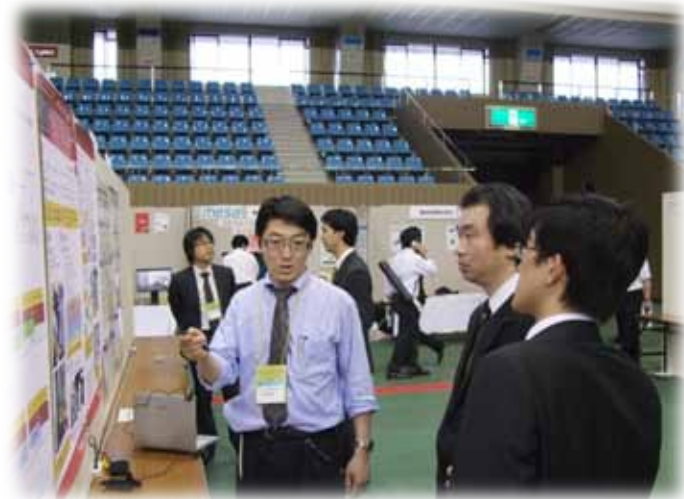
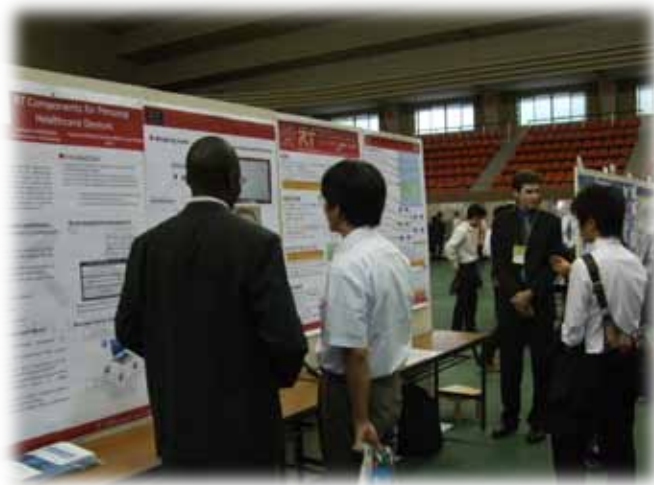
# RoboMec2009 講習会 [旭川]



- 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)





# コンソーシアムへの期待

- 高エネルギードメインの発展  
(経験やリソースの共有、人材・企業育成)
- 技術フィードバック  
(必要となる機能、開発手法、ツール)

共同研究

次世代高エネルギー加速器データ収集・制御システム  
に関する研究

# 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/>
- [RTミドルウェアコンテストのホームページ](http://www.is.aist.go.jp/rt/rtmcontest.html)  
<http://www.is.aist.go.jp/rt/rtmcontest.html>
- [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>