

青梅商工会議所・産総研 RTミドルウェア講習会



RTミドルウェア概要

日時: 2011年7月25日(月) 10:30- 11:15
場所: 産業技術総合研究所 つくば第2事業所ネットワーク会議室
(独)産業技術総合研究所 知能システム研究部門
神徳徹雄、安藤慶昭



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

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)



概要

- RTミドルウェア概要
- RTコンポーネントの作り方
- 今後の展望
- 幾つかのテクニック

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)

2

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

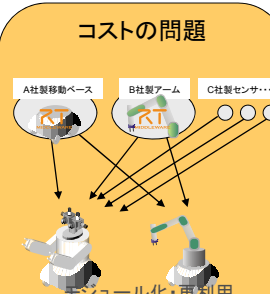
NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

4

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




A社製移動ベース B社製アーム C社製センサ...

モジュール化・再利用

ロボットの低コスト化

技術の問題



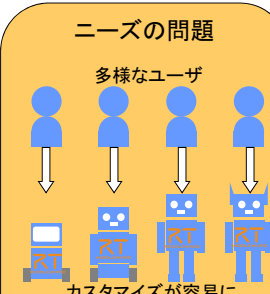
最新の理論・アルゴリズム

RTコンポーネント化 仕様

システム開発者

最新技術を利用可能

ニーズの問題



多様なユーザ

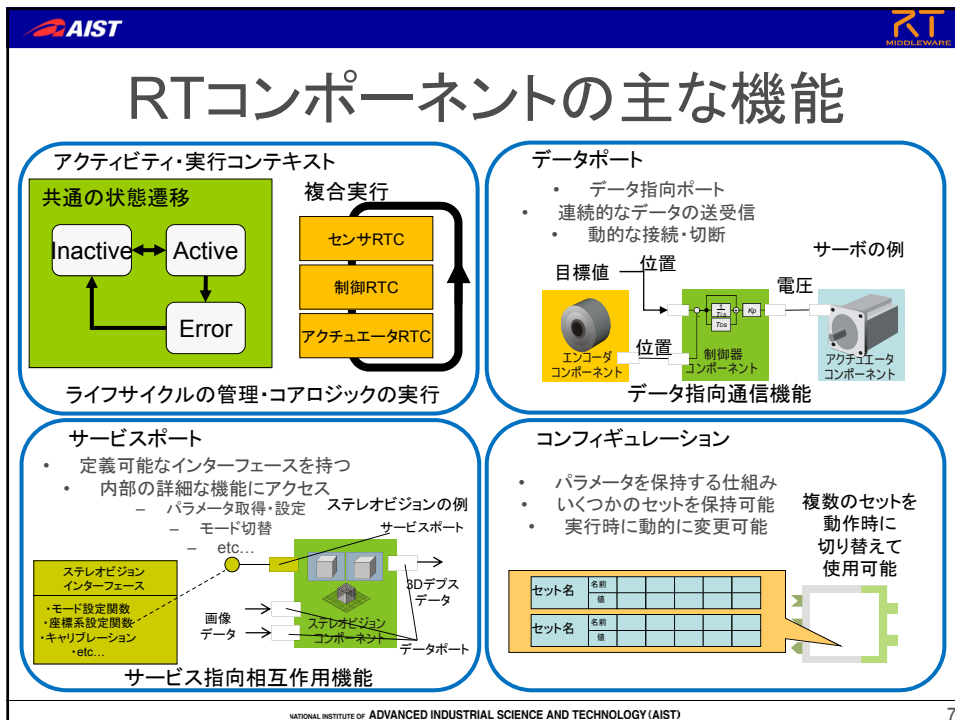
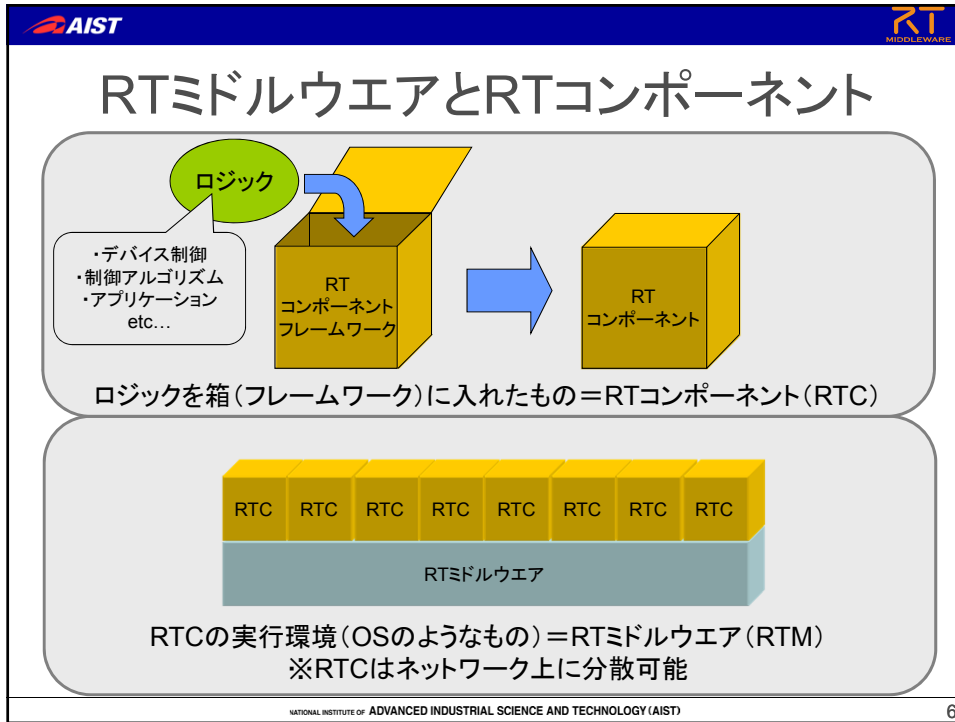
カスタマイズが容易に



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によるイノベーション

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

5



リアルタイム実行コンテキスト

非リアルタイムLinux

非リアルタイムスケジューラ

ExecutionContext

↔

リアルタイムLinux

リアルタイムスケジューラ

ExecutionContext

実行コンテキストを動的に差し替えることが可能

非リアルタイム Linux用 実行コンテキスト



同一のRTコンポーネントバイナリ

リアルタイムLinux用 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

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

8

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

カメラコンポーネント

カメラコンポーネント

マイクコンポーネント

画像データ

顔位置問合せ

顔認識コンポーネント

人物データ 表情データ

カメラコントロール

ステレオビジョンコンポーネント

音声データ

音声認識コンポーネント

文字データ

対話コンポーネント

文字データ

ジェスチャ軌道データ

頭・腕駆動コンポーネント

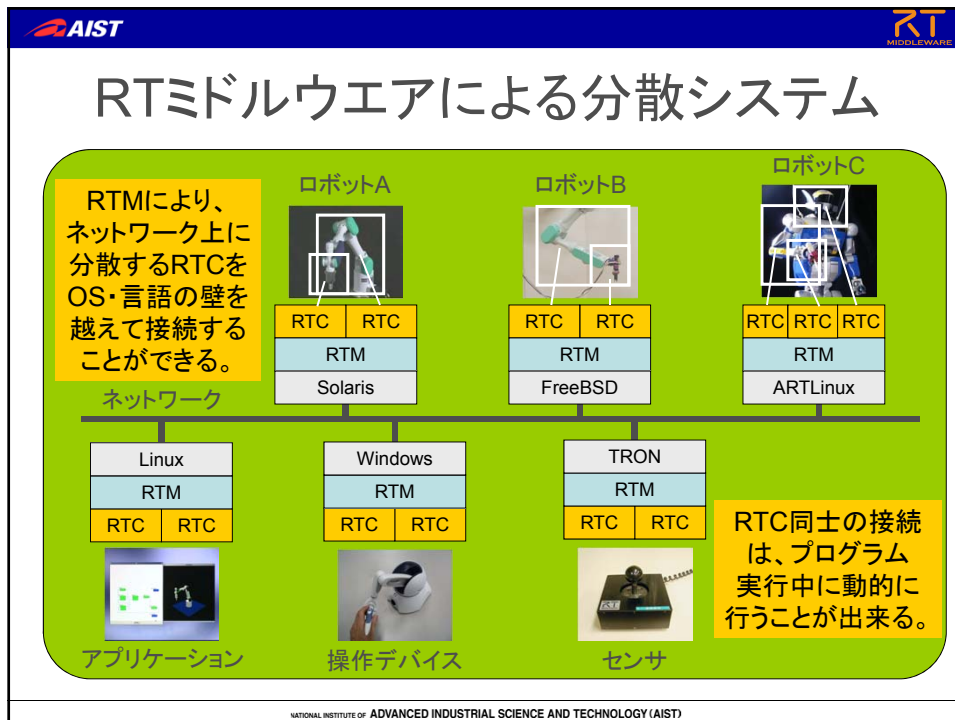
音声合成コンポーネント

ポート

データ・コマンドの流れ

(モジュール)情報の隠蔽と公開のルールが重要

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)



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

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

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)




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++ (1.1.0), 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 (株式会社テクノジックアートより発売中)

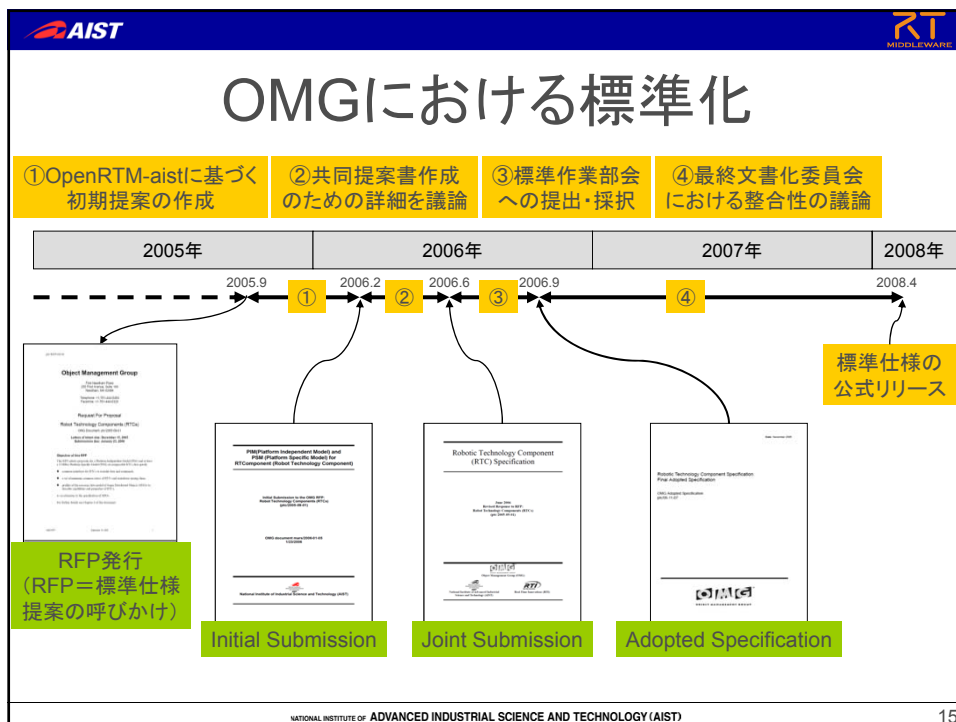
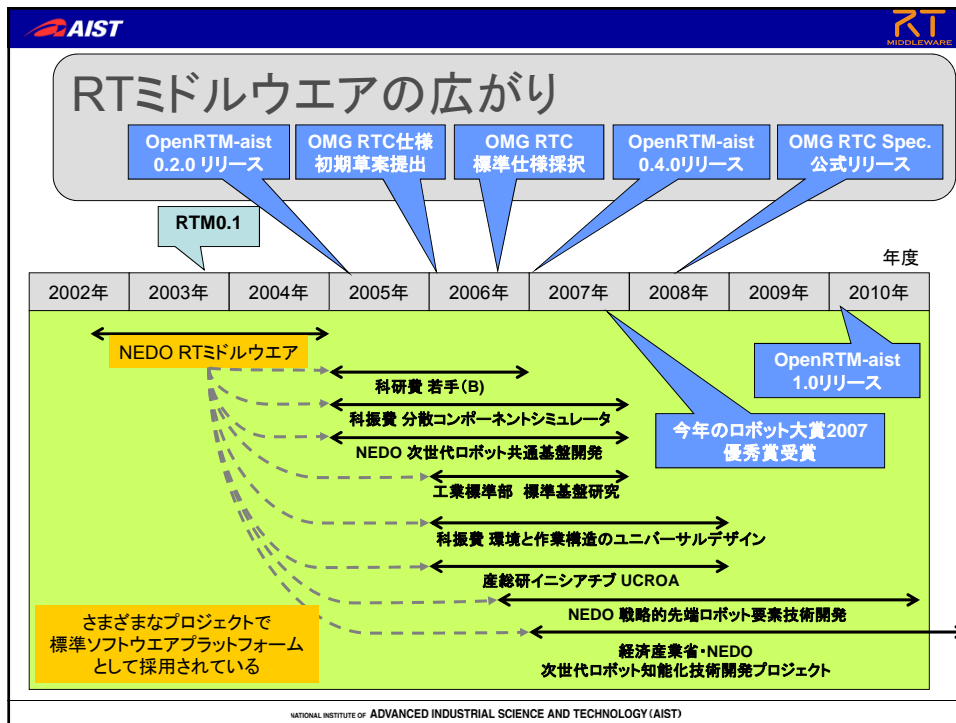
NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)
12





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)
13






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




National Institute of
Advanced Industrial Science
and Technology
AIST



株式会社セック
Systems Engineering Consultants Co., Ltd.



ETRI
한국전자통신연구원
Electronics and Telecommunications Research Institute







GOSTAI
Robotics for everyone
THALES


NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

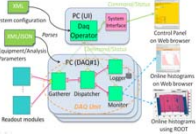
16






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














NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

17


書籍 (1)



*** 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

株式会社エック
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

- 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

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)
18

書籍 (2)



Journal of the Robotics Society of Japan
日本ロボット学会誌
June 2010
Vol.28 No.5

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)

AIST RT MIDDLEWARE

RTコンポーネントプログラミング

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST) 21

AIST RT MIDDLEWARE

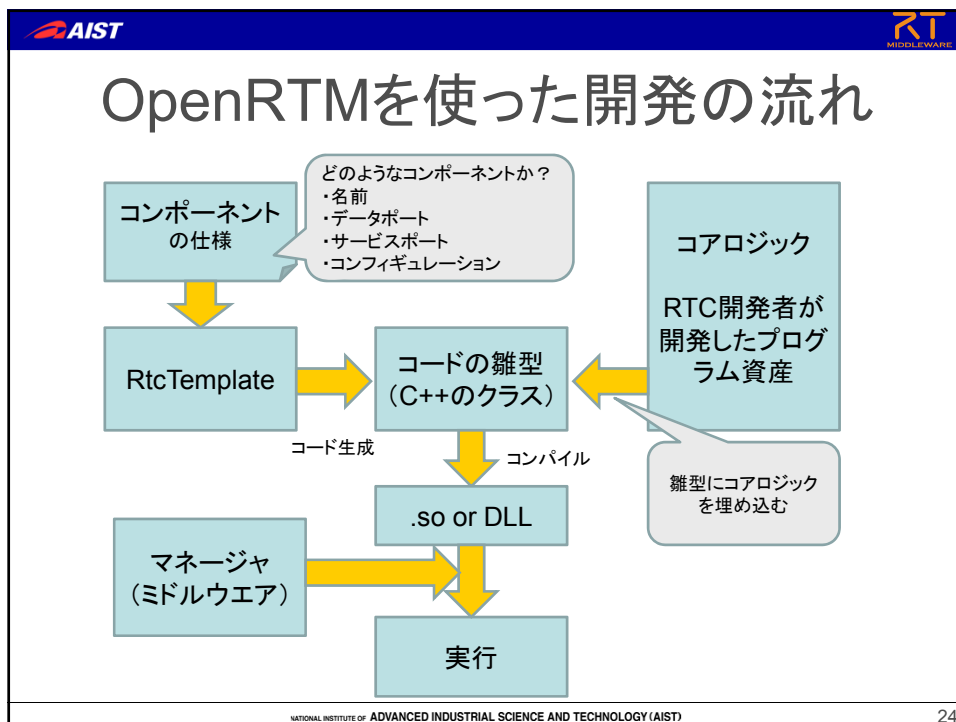
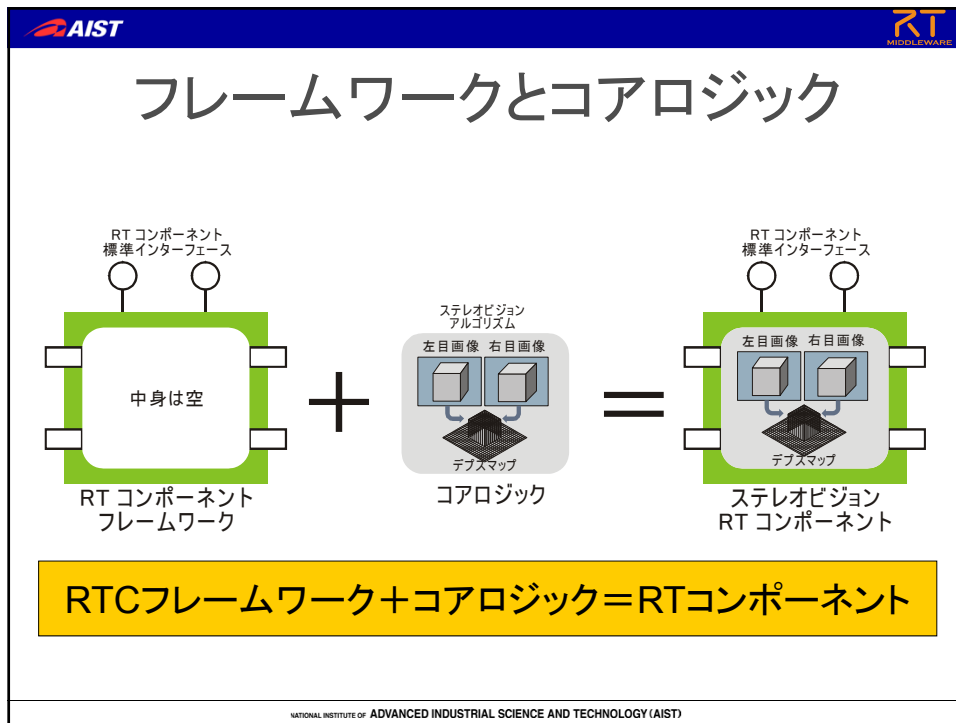
OpenRTMを使った開発の流れ



```
graph TD; A[OpenRTM 開発環境] --> C[RTコンポーネント]; B[ユーザが作ったライブラリ・クラス] --> C; C --> D[分散システムで利用]; C --> E[他のシステムで再利用];
```

この部分がきちんとできていることが非常に大事

OpenRTMを使えば簡単に分散オブジェクトになる。

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)



コード例

- 生成されたクラスのメンバー関数に必要な処理を記述
- 主要な関数
 - onExecute (周期実行)
- 処理
 - InPortから読む
 - OutPortへ書く
 - サービスを呼ぶ
 - コンフィギュレーションを読む

```

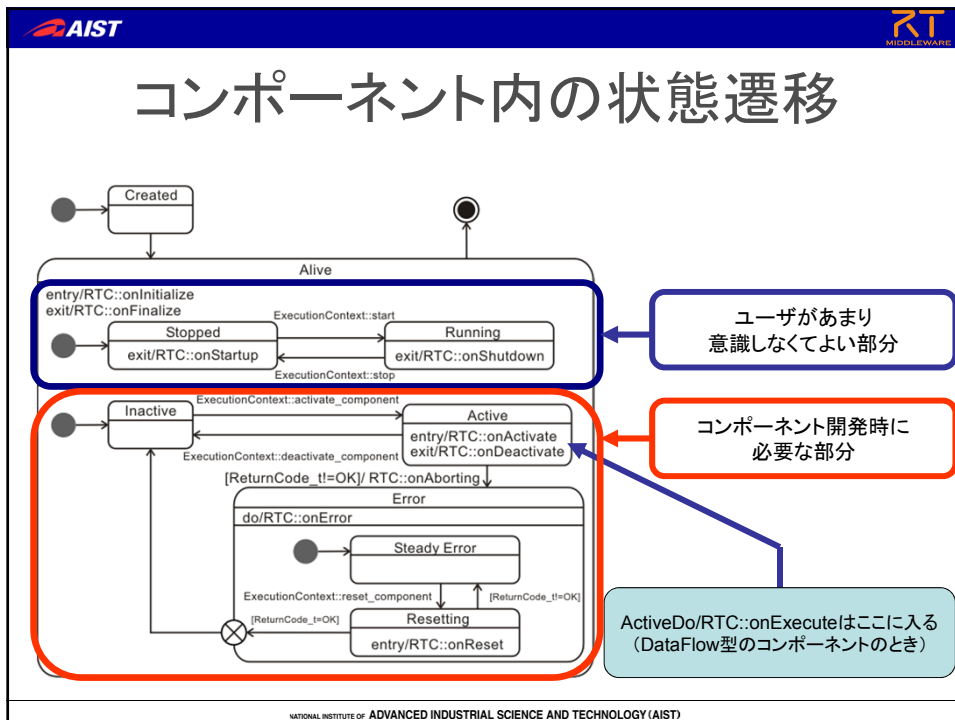
class MyComponent
: public DataflowComponentBase
{
public:
// 初期化時に実行したい処理
virtual ReturnCode_t onInitialize()
{
if (mylogic.init())
return RTC::RTC_OK;
return RTC::RTC_ERROR;
}



// 周期的に実行したい処理
virtual ReturnCode_t onExecute(RTC::UniqueId ec_id)
{
if (mylogic.do_something())
return RTC::RTC_OK;
return RTC::RTC_ERROR;
}

private:
MyLogic mylogic;
// ポート等の宣言
// :
};
            
```

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

25





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とき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とき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

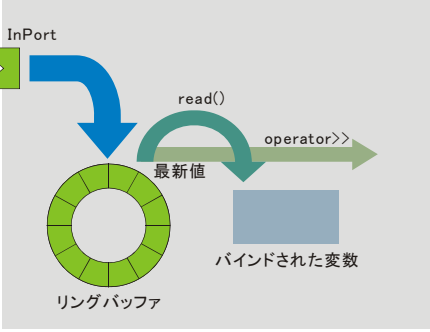
とりあえずはこの5つの関数を押さえておけばOK

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)





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

➡

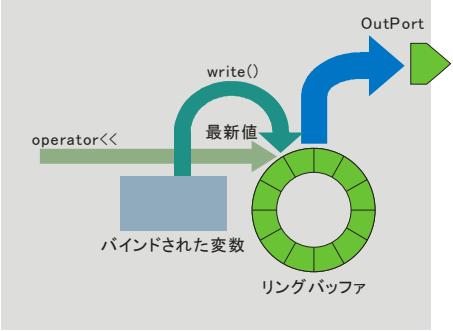
Robot Component

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)





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

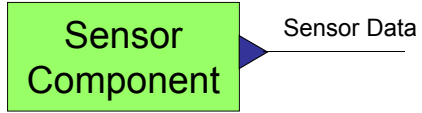


基本的にInPortと対になる





データポートの型を同じにする必要あり

例



NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

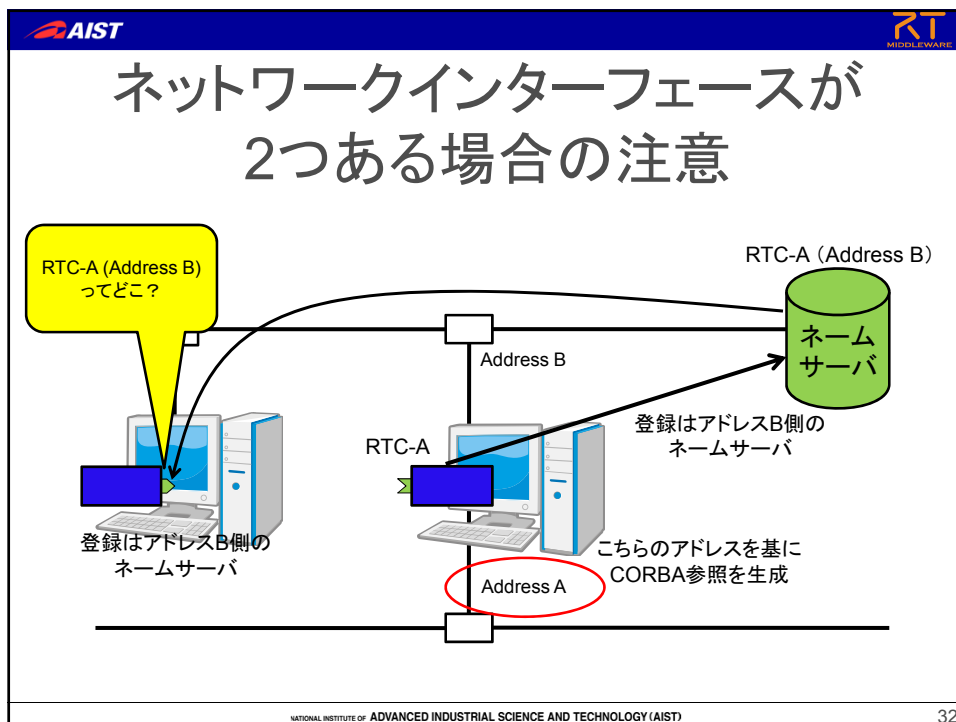
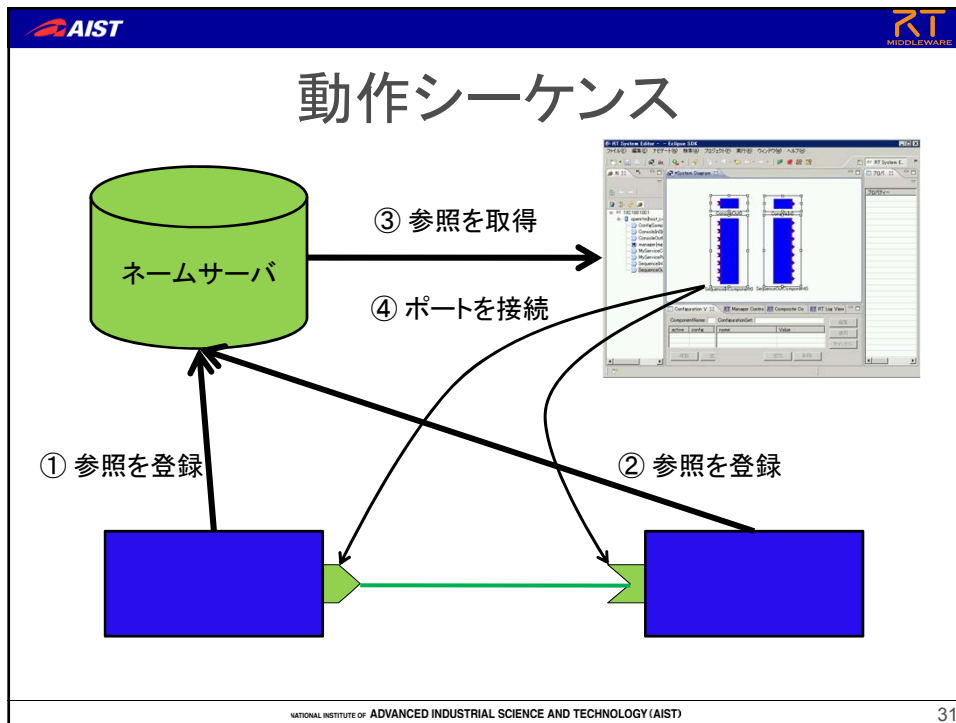
データ変数

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

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

- 基本型
 - tm: 時刻
 - data: データそのもの
- 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

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)






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

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

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)






ログ設定

logger.enable	“YES” or “NO”: ログ出力を有効・無効
logger.file_name	ログファイル名。 %h: ホスト名, %M: マネージャ名, %p: プロセスID 使用可
logger.date_format	日付フォーマット。strftime(3)の表記法に準拠。 デフォルト: %b %d %H:%M:%S → Apr 24 01:02:04
logger.log_level	ログレベル: SILENT, ERROR, WARN, NORMAL, INFO, DEBUG, TRACE, VERBOSE, PARANOID SILENT: 何も出力しない PARANOID: 全て出力する ※以前はRTC内で使えましたが、現在はまだ使えません。

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:” 使いたいNICに割り当てられているIPアドレス 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念のため。)
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 •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)

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

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)
45

Java版、Python版、ツール

Java、Python版

- 8月くらいまでに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

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)
48

RT
MIDDLEWARE

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 (GPL)

RTC (BSD)

RTC (MIT)

RTC (独自)

OpenRTM (オープンソースライセンス)

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

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

RT
MIDDLEWARE

ライセンス (OpenRTM-aistの改編、再配布)

選択

OpenRTM (オープンソースライセンス)

MyRTM (オープンソースライセンス)

ソース公開義務

OpenRTM (個別)

MyRTM (任意)

産総研TLOと契約
ソース公開義務は生じない

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

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)



今後の展望

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST) 51

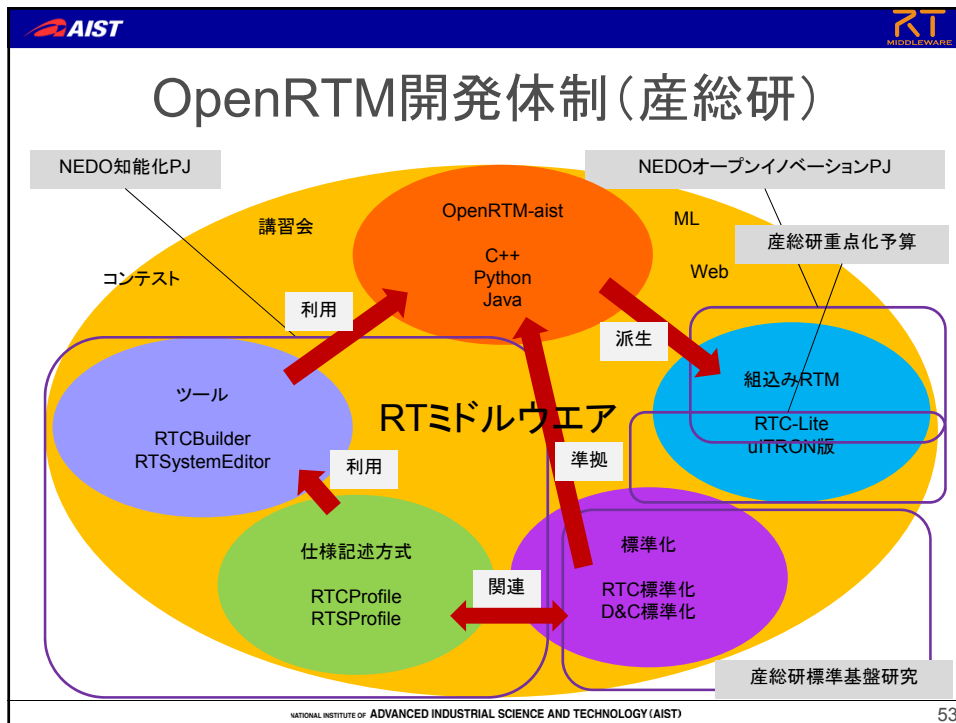


NEDO知能化プロジェクト (次世代ロボット知能化技術開発プロジェクト)



- 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！！**

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST) 52



- ## 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
- 54





まとめ

- ロボット用ミドルウェア : OpenRTM
 - ロボットに適した共通フレームワークの提供
 - OMG国際標準
 - 多様な実装、多様な言語、OSに対応
- RTコンポーネントの作り方
- 今後の展望
 - 次世代ロボット知能化PJ、Webページ

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)



55




初心者のための テクニック

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

56



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)

テクニック(日本語)



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。

プログラムの再利用

外国で開発されたプログラムやUNIX用のプログラムの再利用をするときには、「日本語の利用」や「スペースを含むフォルダ名」が想定されていない。

ユーザ名、フォルダ名に日本語を使わない

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

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




PC #1
Pentium3-300MHz

RTミドルウェアは魔法ではない

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

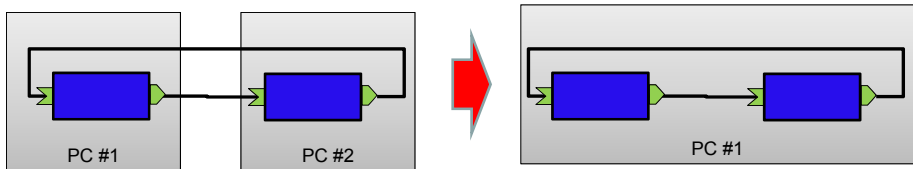



テクニック(資源を考慮)





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

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)



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、パラメー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...

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)





テクニック(接続タイプ)



Flash(同期送信に注意)

同期送信ではデータが受信側まで届いて送信が完了するまで書き込み側が待たされる。送信側で時間にクリティカルな処理は駄目。



実時間処理

➔

New
(非同期送信)
を選択する

RTミドルウェアは魔法ではない

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

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。


送信バッファ制御

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

テクニック(通信路指定)

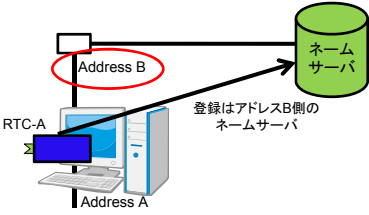


通信経路が複数ある場合
 複数のネットワークインタフェースを持つ場合(Ethernetの他に無線やVirtual Machineの通信経路がアクティブな場合)、ネームサーバに間違っただ情報が登録される場合がある。

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)

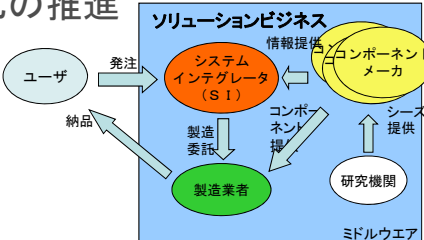
RTミドルウェアコンテスト(趣旨)

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/>

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)

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

これからのロボット開発

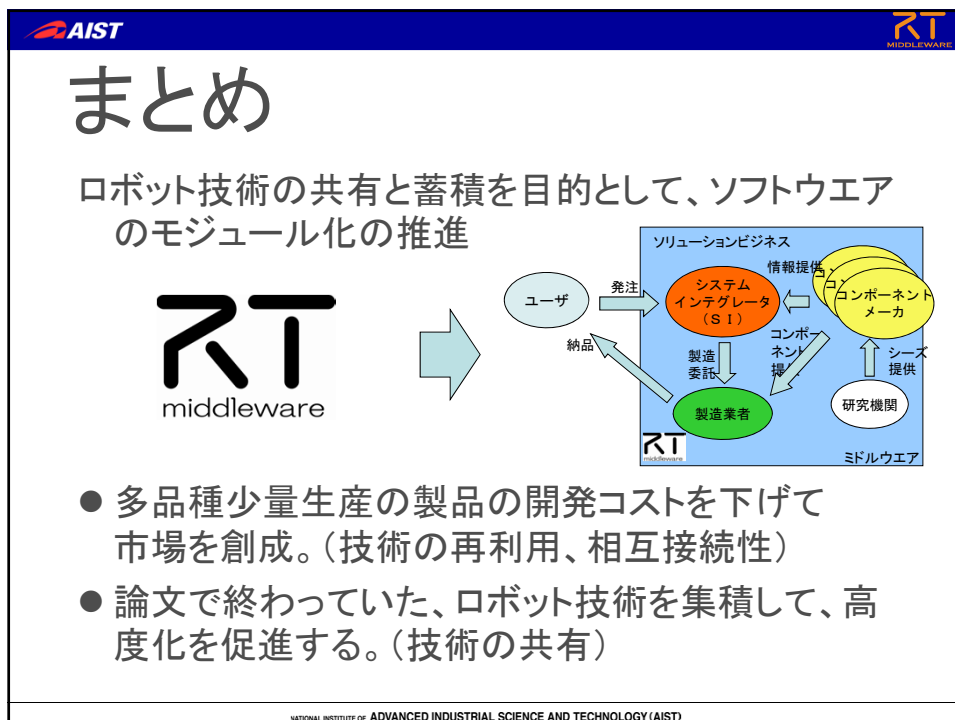
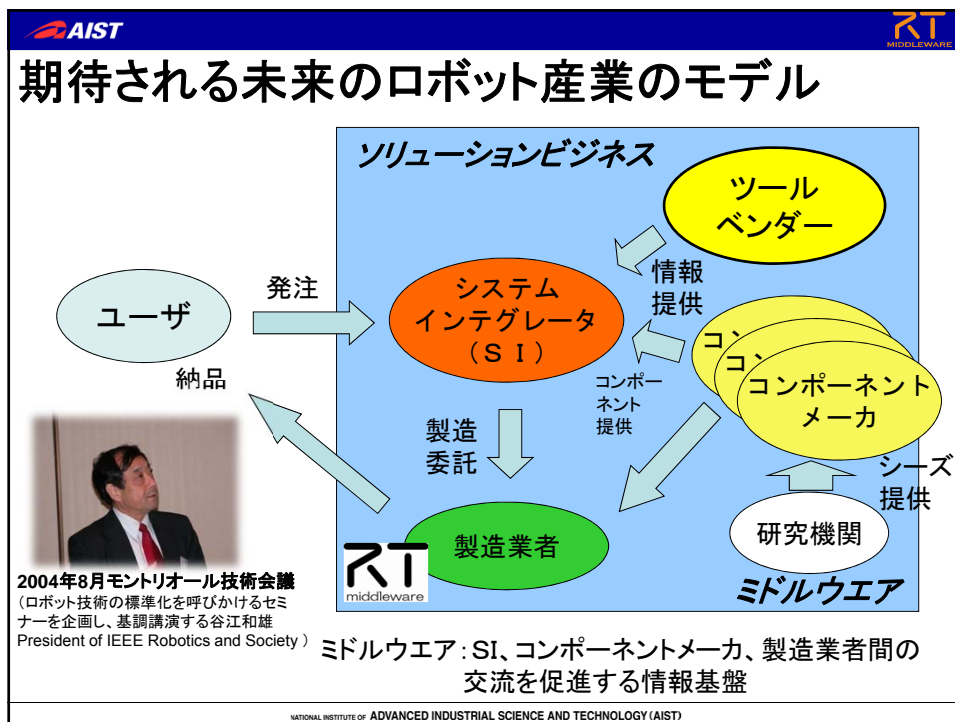
ソフトウェアのモジュール化が現実になることで...

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

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 (産総研)
(近日公開予定) <http://www.openrtm.org/>
- [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>

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)



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>

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)80