

**RTミドルウェアサマーキャンプ2011
特別講演2：「RTミドルウェアによる
システム事例の紹介とRTC開発のポイント」**

アジェンダ

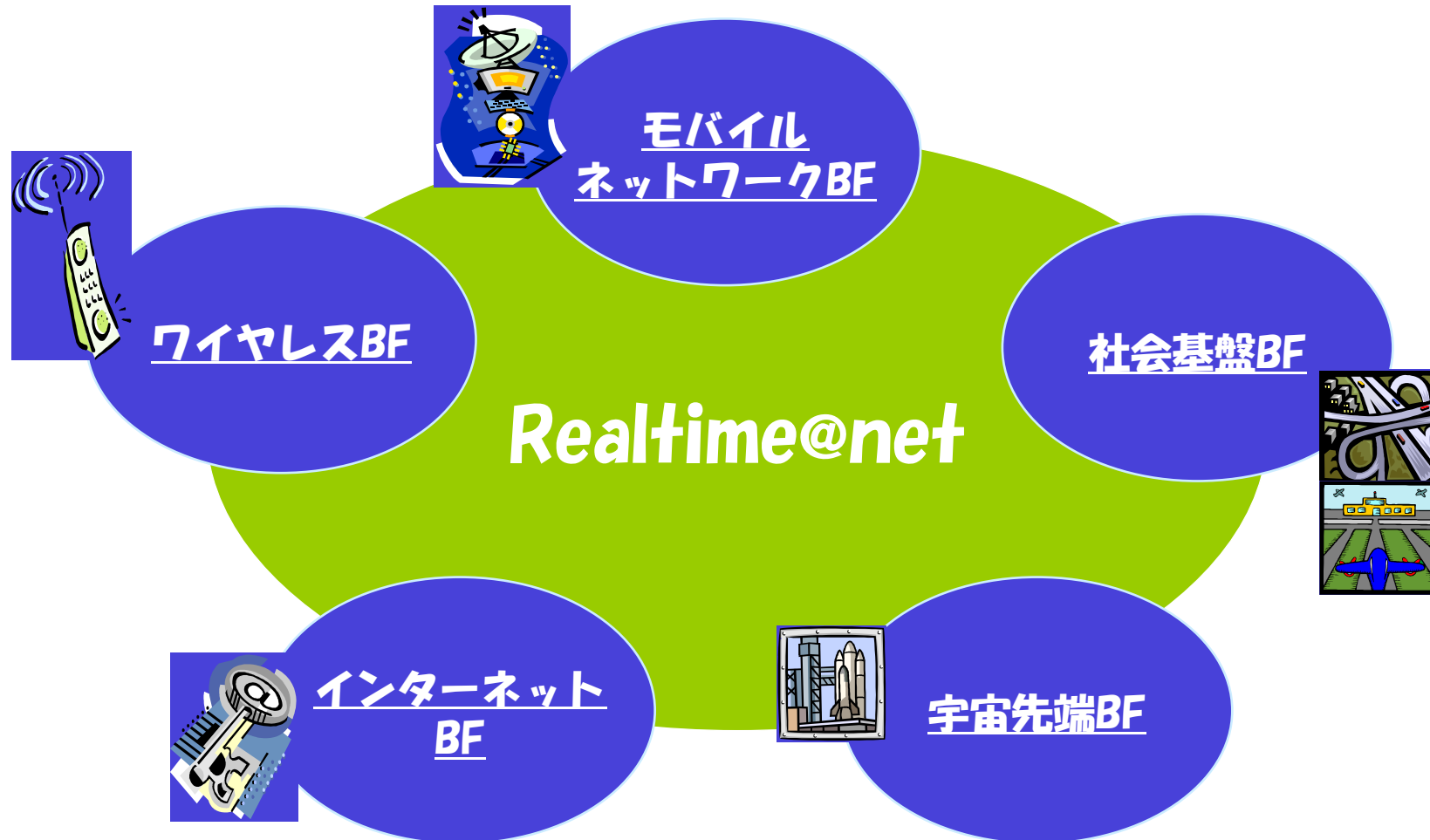
- 株式会社セックのご紹介
- RTミドルウェアのシステム事例
 - さまざまなRTミドルウェア
 - ロボットシステムにおけるRTC構成例
- RTC開発のポイント
 - RTCの粒度、I/Fの決め方など
 - RTCのテスト方法(RTCデバッグ紹介)

株式会社セックのご紹介

- 社名：株式会社セック（略称：SEC）
Systems Engineering Consultants Co.,LTD.
- 設立：1970年5月
- 資本金：4億7730万円（2011年6月28日現在）
- 従業員数：264名（2011年4月1日現在）
- 本社：東京都世田谷区
世田谷ビジネススクエア



株式会社セックのご紹介



ロボティクス分野への取り組み

- 2002年から研究開発に取り組んでいます



- 標準化活動
- 大学・研究機関・メーカーからの受託開発
- 国プロ研究委託
- 自社ビジネス

標準化活動

- RTミドルウェア国際標準化調査専門委員会
- 次世代ロボット知能化技術開発プロジェクト標準化委員会
- ロボットビジネス推進協議会RTミドルウェアWG
- RTシステムインテグレーション部会 (SICE)
- RSi (Robot Services initiative)
- OMG Robotics DTF
- 日本ロボット工業会正会員
- RooBO会員

RTミドルウェア

■ 特にRTミドルウェアに注力！



- ロボットラボラトリー主催
「RTミドルウェア研究会ビジネスモデル検討会」
- 福岡ビジネス創造センター主催
「ロボットビジネスセミナー～ロボット知能ソフトウェアプラットフォームの研究開発～」
- 名城大学 理工学部 電気電子工学科様
RTミドルウェア講習会
- NEC様
「RTミドルウェアの本質～技術、取り巻く環境、ビジネスの可能性～」
- 書籍
「はじめてのコンポーネントベース・ロボットアプリケーション開発～RTミドルウェア超入門～」
- Think IT 6月特集記事
「SEのためのRTシステム概論」 <http://thinkit.jp/article/950/1/>



国プロ研究委託

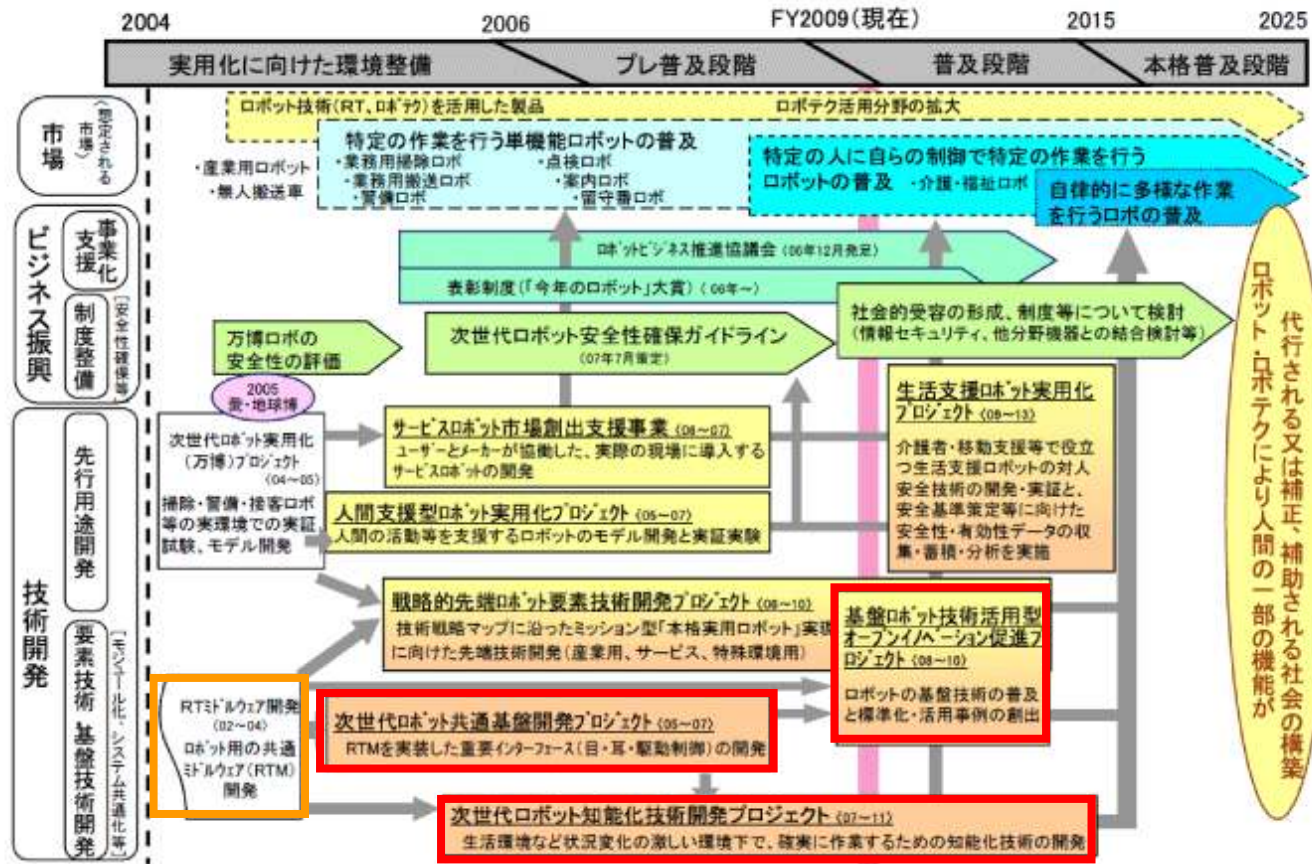


図 2：ロボット分野の導入シナリオ

「経済産業省 技術戦略マップ2009 システム・新製造 ロボット分野」より出典

セックが関わっている ロボット技術マップ

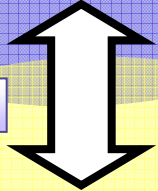


産業用ロボットの
監視・制御
アプリケーション



ORiNプロトコル

DeviceNetなどの
制御系ネットワーク



知能モジュール(知
能化PJ)を搭載した
RTミドルウェア対応
の各種ロボット



産業用
ロボット

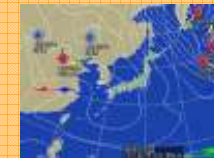
RSNPプロトコル

ネットワークを利用した
ロボットサービス
アプリケーション
(RSiやRoIS)

- ・スマートハウス/EMS
- ・見守りサービス
- ・気象/防災情報サービス
- ・セキュリティサービス

など

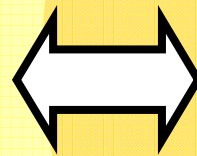
RSNPプロトコル



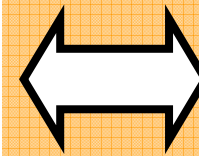
知能モジュール群
・移動知能
・作業知能
・コミュニケーション知能



サービスロボット

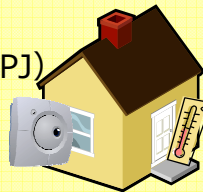


インターネット



ユーザインタフェース

RT住宅
(オープンイノベーション)



組み込み向けRTミドル
ウェアやRTミドル
ウェア対応センサ、
電力線通信(PLC)を
利用



アジェンダ

- 株式会社セックのご紹介
- RTミドルウェアのシステム事例
 - さまざまなRTミドルウェア
 - ロボットシステムにおけるRTC構成例
- RTC開発のポイント
 - RTCの粒度、I/Fの決め方など
 - RTCのテスト方法(RTCデバッグ紹介)

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 準拠
- OS
 - 公式: FreeBSD, Linux (Fedora, Debian, Ubuntu, Vine, Scientific), Windows
 - 非公式: Mac OS X, uTRON, T-Kernel, VxWorks
- 言語:
 - C++, Python, Java
 - .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)

多様な実装

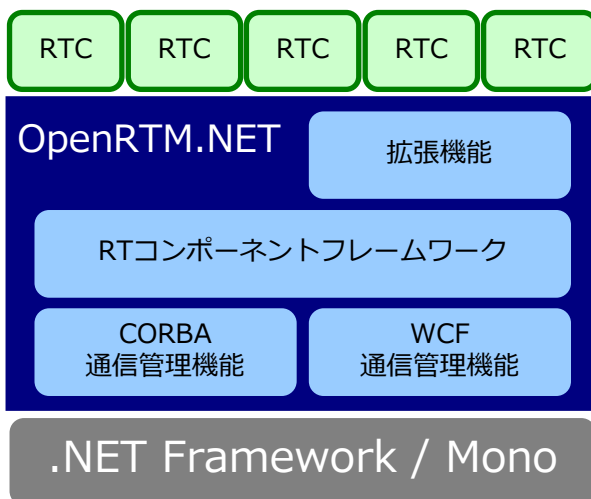
- OpenRTM-aist: 産総研実装
 - C++, Python, Java の3種類
- OpenRTM.NET: 株式会社SEC版実装
 - .NET版: VB, C#
- 韓国ETRI
 - OPRoS コンポーネント:一部準拠
- PALRO: 富士ソフト
 - 小型ヒューマノイド制御フレームワークがC++ソースレベルでOpenRTM互換
- GostaiRTC: 仏GOSTAI & Thales
 - OMG RTC Local PSM に準拠



OpenRTM.NETは、OMGの国際標準仕様に準拠したRTミドルウェア実装です。

OpenRTM.NETの特徴

- ✦ **高い生産性**：C#やVisual Basicを用いて、RTコンポーネントを効率的に開発することができます。カスタム属性の採用により、シンプルな記法でRTコンポーネントを実装できます。
- ✦ **導入が容易**：インストーラが用意されており、環境構築が容易です。通信層を隠蔽することにより、IDLなどのCORBAの知識が技術的な敷居が低いのも特徴です。
- ✦ **サービス指向**：CORBAやWCFなど複数の通信ミドルウェアに対応しています。インターネット上のサービスと連携したアプリケーションの開発も容易です。



3Dシミュレータ



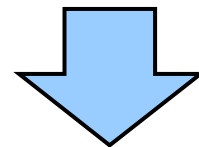
受付案内ロボット



OpenRTM.NETの仕様	
動作OS	Windows XP / Vista / 7 Linux (Ubuntu / openSUSEなど)
実行環境	.NET Framework 3.5 Mono 2.4
プログラミング言語	C# / Visual Basic / C++ / CLI
通信ミドルウェア	CORBA / WCF / Local
RTC仕様	OMG RTC Specification 1.0

RTM on Android

- Android端末をRTシステムのUIとして利用
- Android端末でロボットの制御
- Android端末でセンサネットワークを構築

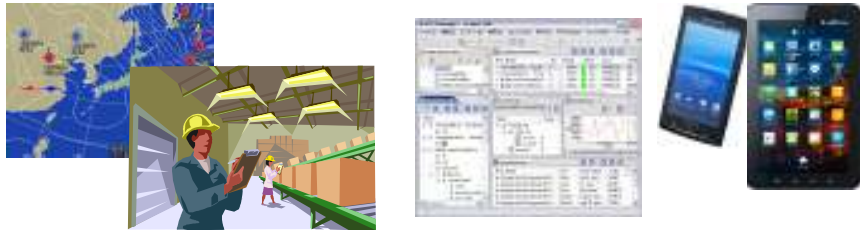



- Android プラットフォーム上で動作するRTミドルウェアを追加



RTミドルウェアラインナップ

エンタープライズ層

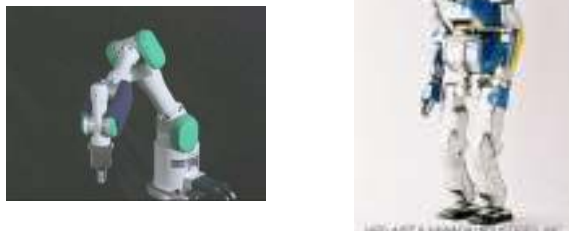


RTミドルウェア製品群 

OpenRTM.NET

RTM on Android

ロボット層



OpenRTM-aist

OpenRTM-aist for
VxWorks

高信頼
RTミドルウェア

エンベデッド/デバイス層



RTC-CANopen

miniRTCs-CAN

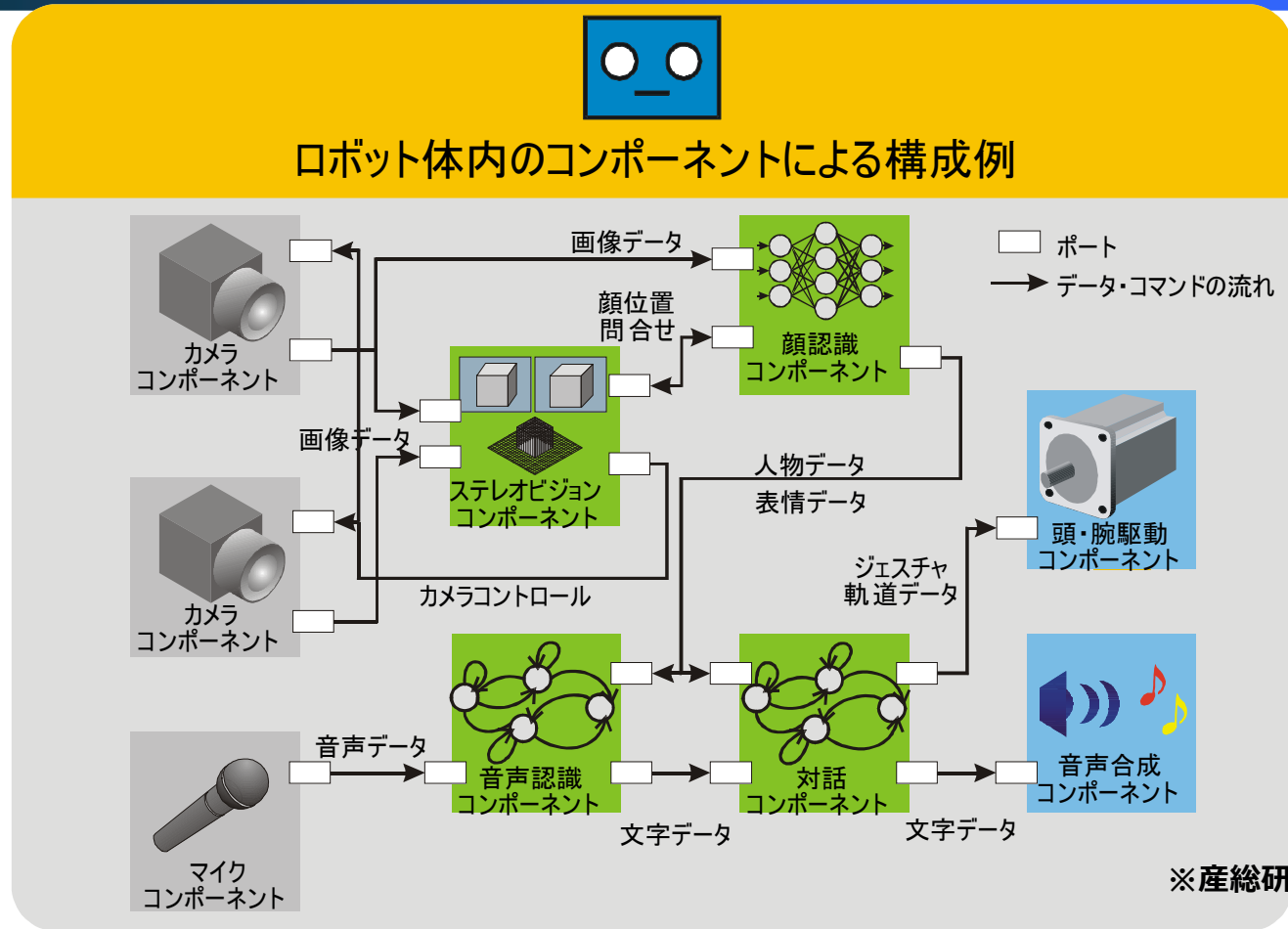
microRTCs-Zigbee

RTC-Lite

RTミドルウェアによるシステム構築例

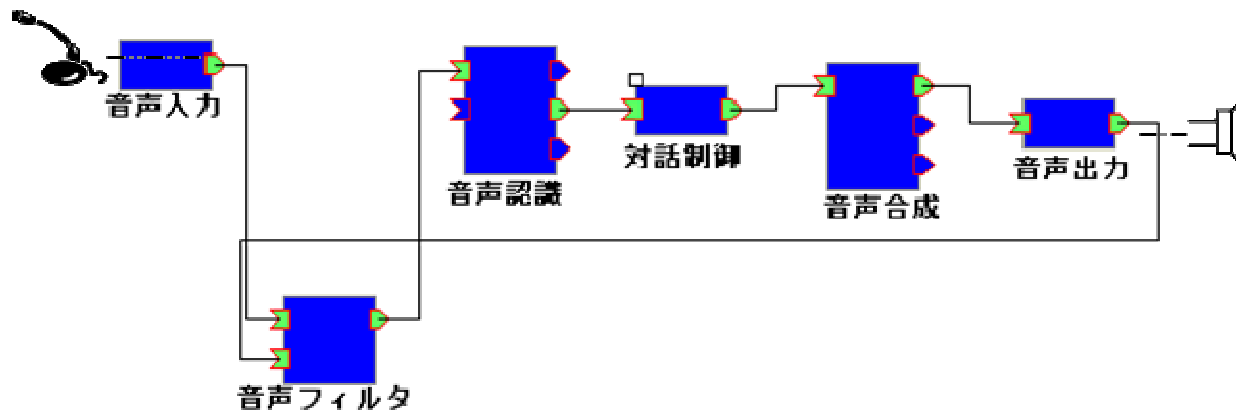
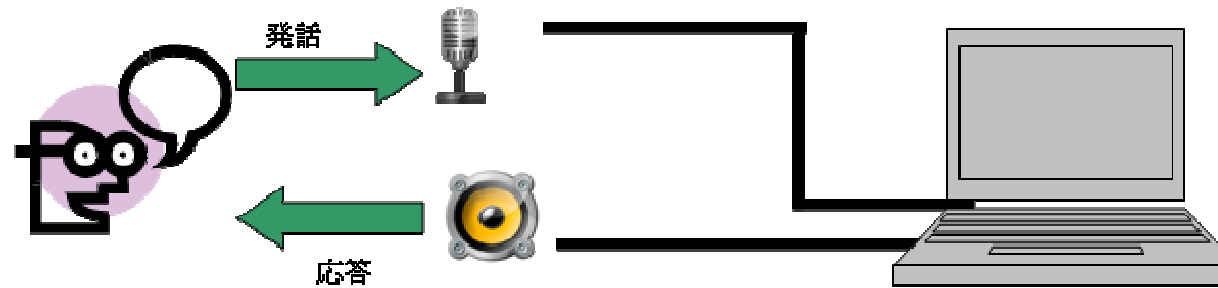


RTCの分割と連携



(モジュール)情報の隠蔽と公開のルールが重要

音声対話システムの構成 (OpenHRI)



ロボカップ@ホーム

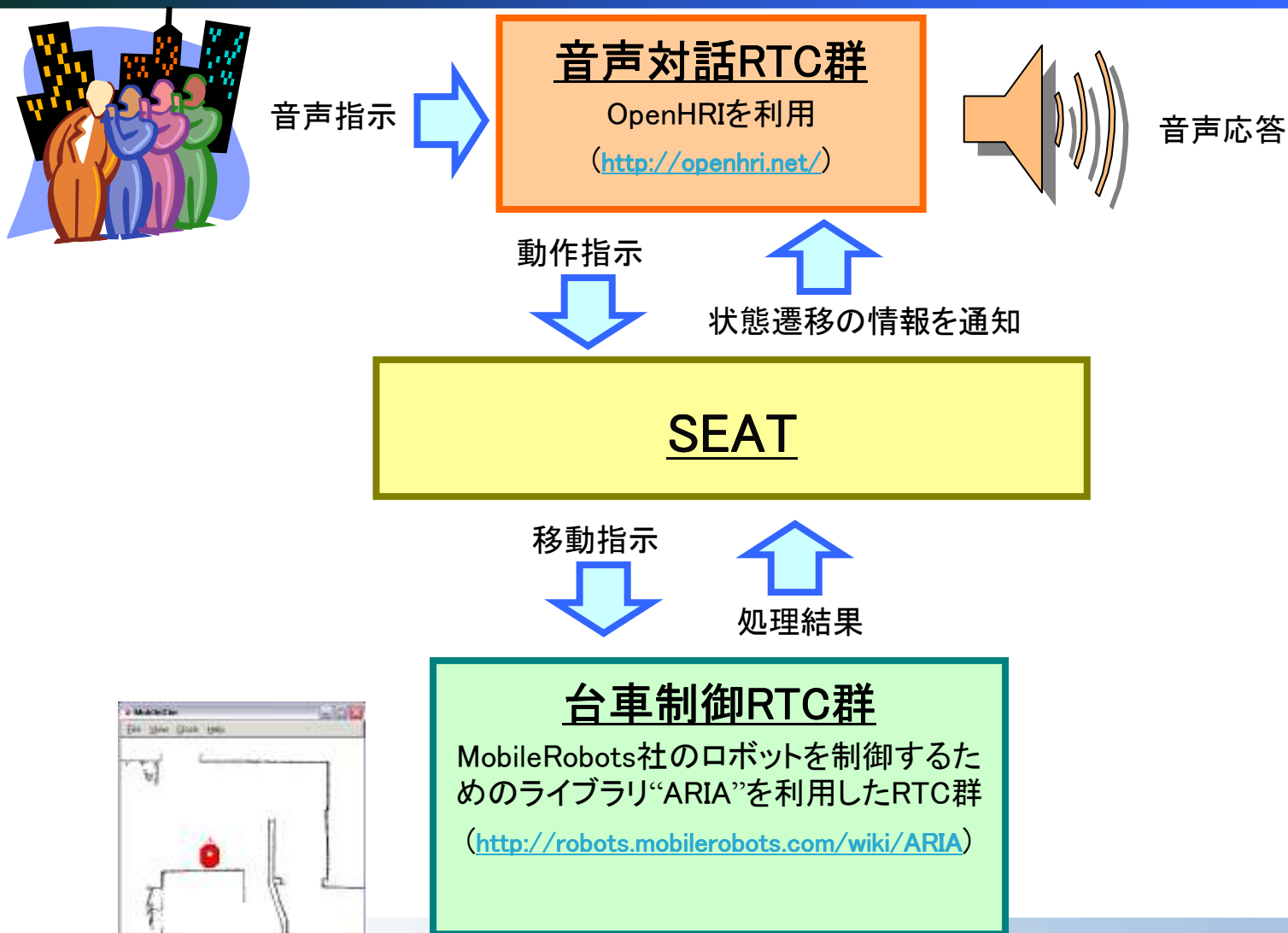
- ロボカップ@ホームのタスク例
 - Robot Inspection and Poster Session(RISP)
 - Follow Me
 - Go Get It!
 - Who Is Who
 - General Purpose Service Robot
 - Shopping Mall



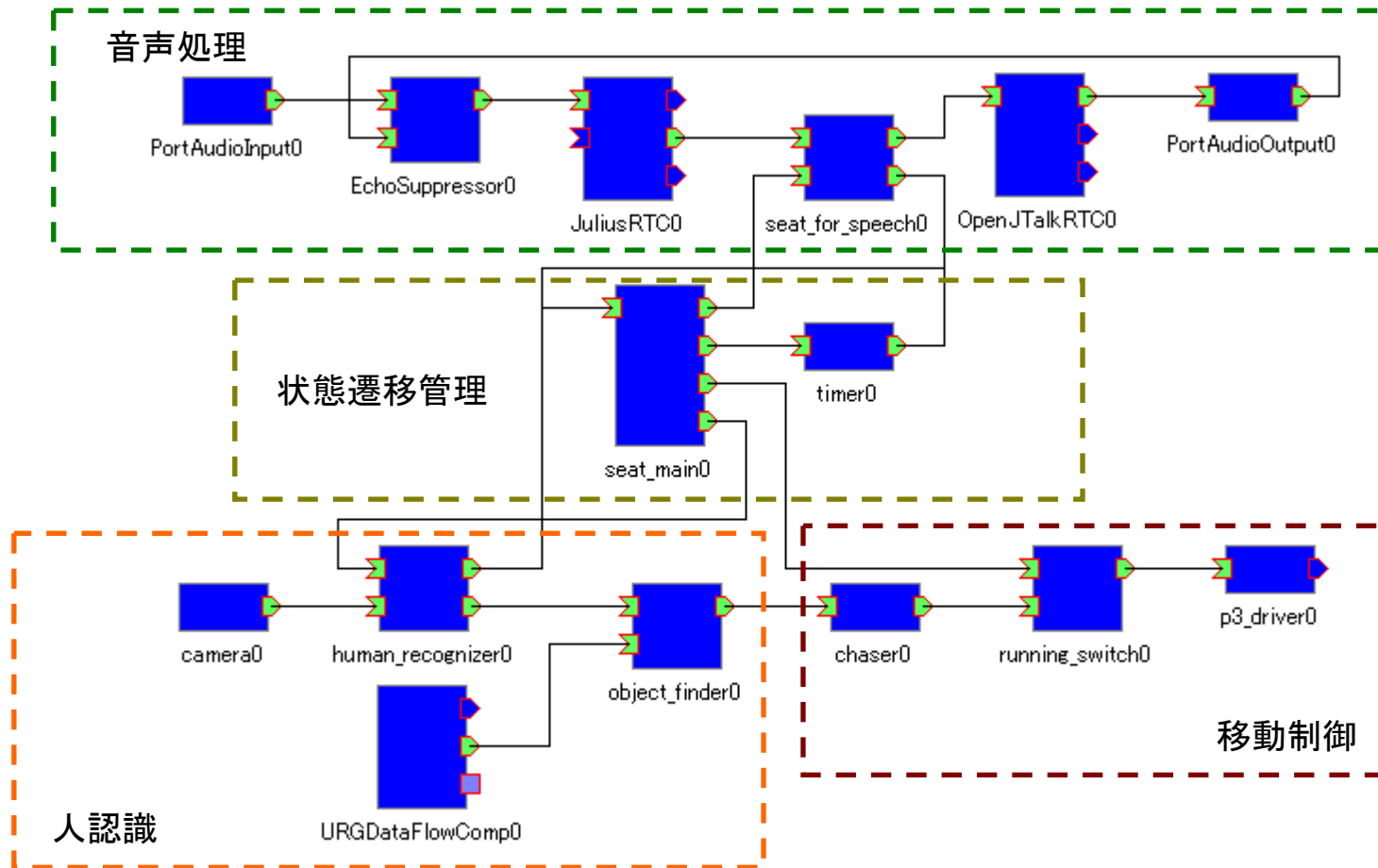
ロボカップ@ホーム

モジュール分類	モジュール名	モジュール補足	利用可能なライブラリ
自律移動	大域的経路計画	地図からの経路生成	知能化PJモジュール
	局所的経路計画	軌道追従、衝突回避	知能化PJモジュール
	障害物検知	LRF	知能化PJモジュール
	自己位置推定	オドメトリ、SLAM	知能化PJモジュール
	地図更新	SLAM	知能化PJモジュール
	人検知・追跡	カメラ	知能化PJモジュール
	モータ出力	動作指令	知能化PJモジュール
	センサ入力	LRF、(ステレオ)カメラ	—
	コミュニケーション	音声認識	日本語、英語
対話スクリプト		—	
音声合成		—	
音声出力		スピーカー	
顔認識・認証		見知らぬ人の顔と名前を覚える	オムロン OKIAO Vision OpenCV
ジェスチャー認識		身振り、手振りの認識	オムロン OKIAO Vision OpenCV
マニピュレーション	センサ入力	カメラ、マイクロフォンアレイ	音声認識モジュール
	順・逆運動学計算	—	OpenRAVE
	軌道生成	—	OpenRAVE
	ハンドアイマニピュレーション	—	OpenRAVE
	物体探索・移動・認識・把持	ペットボトル、ドア、冷蔵庫、缶	OpenRAVE
	パンチルト台		
	オブジェクトの発見・識別	—	OpenCV
	センサ入力	TOF、ステレオカメラ	—
コントローラ	マニピュレーション制御	—	
システム	タスク管理	シナリオによるタスク実行制御	シナリオエディタ フィロソフィー
	モジュール操作	RTシステムエディタの自動化	rtsshell rtcshell
	システム管理	動作状況管理、ログ管理	—

Follow Meのコンポーネント構成



Follow Meのコンポーネント構成



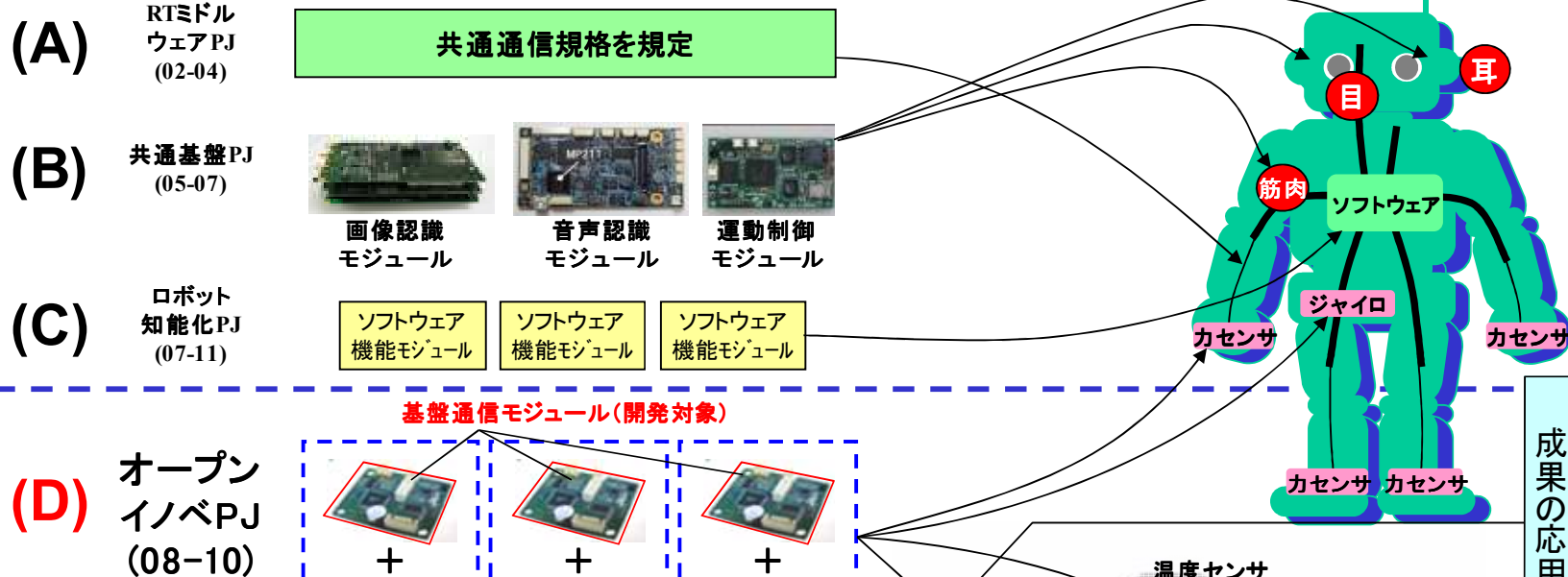
基盤ロボット技術活用型オープンイノベーション 促進プロジェクト

※NEDO作成資料を流用

対応PJ

ロボットの機能要素

ロボット用途



研究開発の内容

種々の既存部品に「**基盤通信モジュール**」を組み合わせ、専用処理ソフトを組み込んで種々の「**RT要素部品**」と呼ぶモジュールを開発する。これを組み合わせてRTシステムを構築・実証する。

研究開発の狙い

- ①市場拡大(生活空間や職場でのRT応用)
- ②新規参入の促進(接続・制御方法の標準化、オープン化)
- ③市場に存在する資産(部品)の有効活用(プラグ&プレイ機能の実現)



生活環境や職場でのRT応用による市場の拡大

基盤ロボット技術活用型オープンイノベーション 促進プロジェクト

インテリジェント空調
システム



セキュリティ
システム

パワーアシスト
ウインドウ



プラグアンド
プレイ



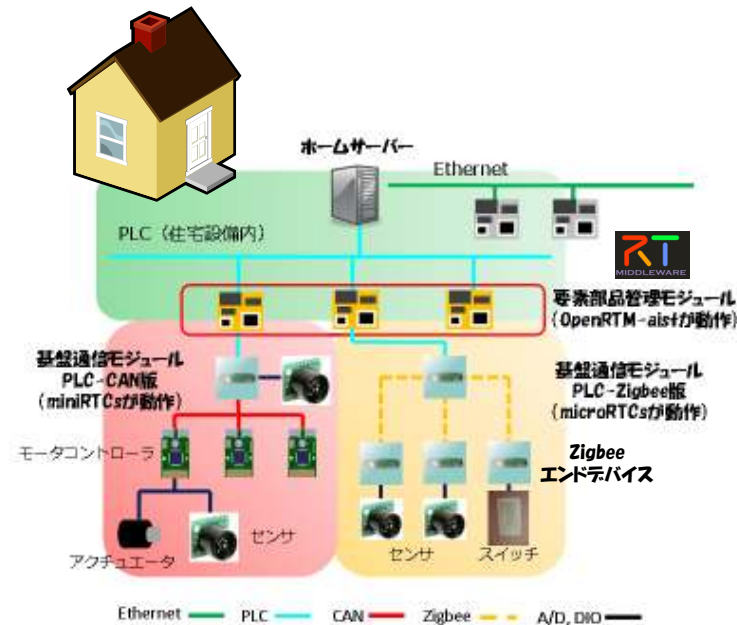
組み込み用軽量RTミドルウェア

- 窓やドア、家電、センサなどに組み込みモジュールと軽量RTミドルウェアを搭載し**RTモジュール化**



住宅内のネットワーク上に分散配置されたRTモジュールが、RTミドルウェア仕様で連携・協調することで、環境分散型の住宅RTシステムを実現

- ホームコントローラによる集中管理ではなく、**階層的な分散管理**のアーキテクチャを採用
- ネットワークプロトコルとして、窓開閉などのリアルタイム制御、通信の信頼性が必要なところは**CAN**、配置の自由度が必要なところは無線ネットワークの**Zigbee**、住宅内の基幹ネットワークには**PLC**を採用



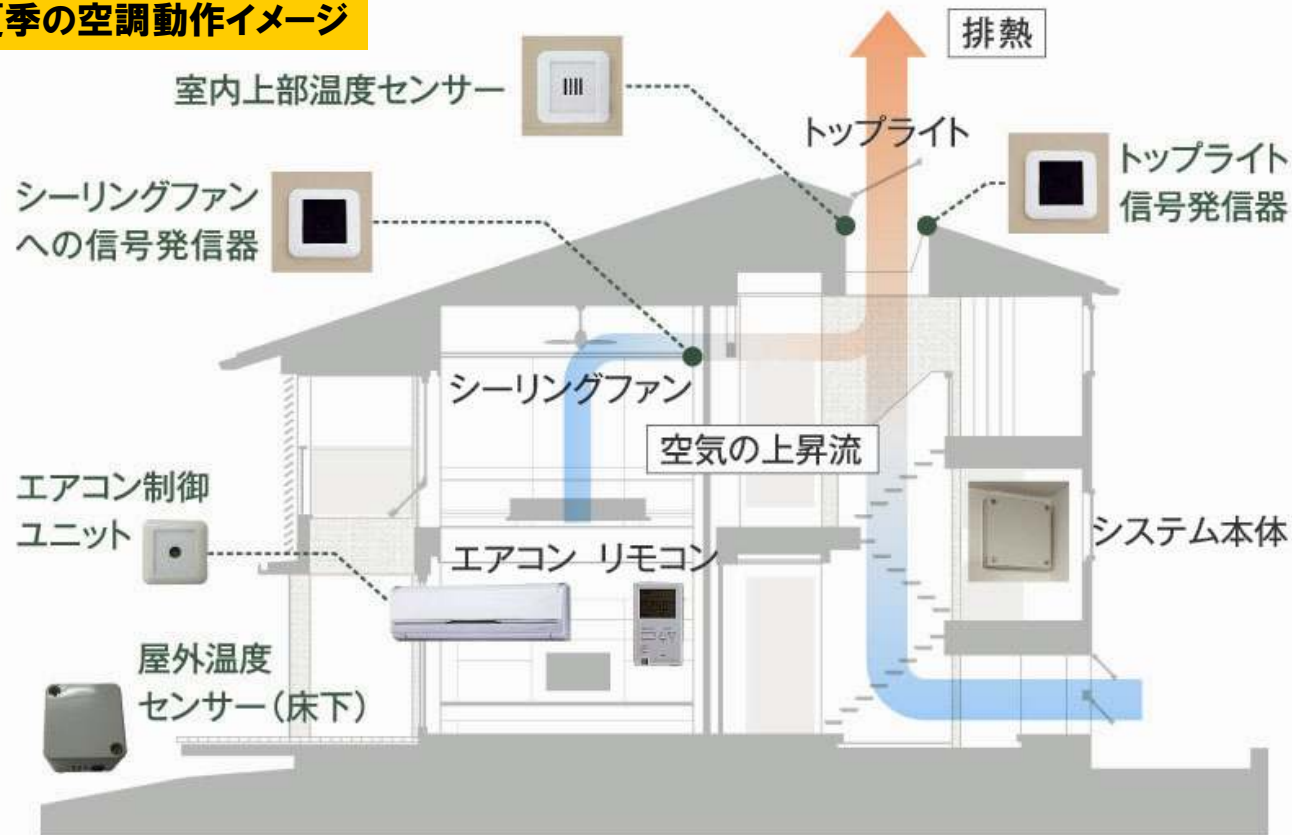
組み込みモジュールのスペック

要素部品管理モジュール		基盤通信モジュール		モータコントローラ (CAN)		Zigbeeエンドデバイス	
CPU	TI AM1808(375MHz)	CPU	SH7214/SH2(100MHz)	CPU	ARM7(60MHz)	CPU	TI CC2530(16MHz)
ROM	2GB	ROM	1MB	ROM	256KB	ROM	256KB
RAM	256MB	RAM	128KB	RAM	16KB	RAM	8KB
OS	Linux	OS	TOPPERS ASP	OS	なし	OS	なし
RTM	OpenRTM-aist	RTM	miniRTCs / microRTCs	RTM	miniRTCs	RTM	microRTCs

インテリジェント空調システム

～エアコン等の空調装置だけに頼らず、通風や排熱を適切にコントロールして
屋内環境を快適に保つ総合的な空調システム～

夏季の空調動作イメージ



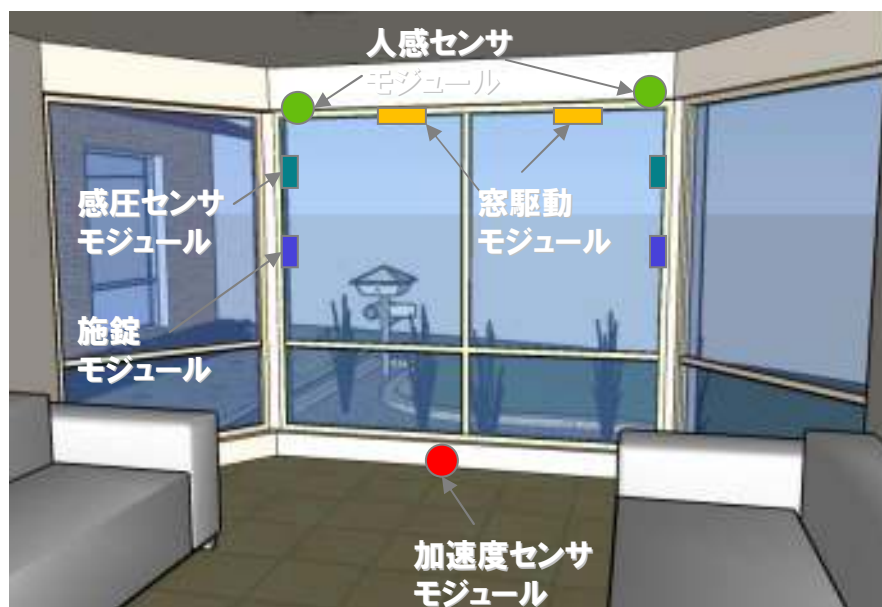
※ミサワホーム総研作成資料を流用

室温・外気温を温度センサーが感知し、トップライトやサッシの開閉やシーリングファン、エアコンを自動制御・運転します。

インテリジェントウィンドウシステム

～利便性、操作性、防犯性、快適性等を向上させる新しい住宅用窓システム～

- パワーアシスト機能 (大型化・重量化の進み高齢者や子供が開閉困難な窓の開閉のパワーアシスト)
- 一括戸締り機能 (玄関施錠で家全体の窓が閉じ施錠) ※侵入の手口の3割は鍵のかけ忘れ
- 空調システム等との連携 (夏季に外が涼しい時は窓を開けて風通しを良くした方が省エネで快適)



各種モジュールの設置イメージ

※ミサワホーム総研作成資料を流用

インテリジェントウィンドウ

※THK作成資料を流用

■ インテリジェントウィンドウ (THK)



①小型通信ドライバーモジュール



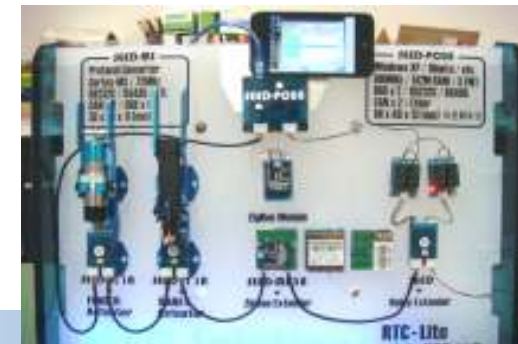
⑤インテリジェントウィンドウシステム



③小型ニアアクチュエータ



②RTC対応パラメータエディタ



④RT要素部品のラインナップ

センサモジュール

※大阪大学作成資料を流用

■ センサモジュール（大阪大学）



■ RTミドルウェア（miniRTCs）対応機器



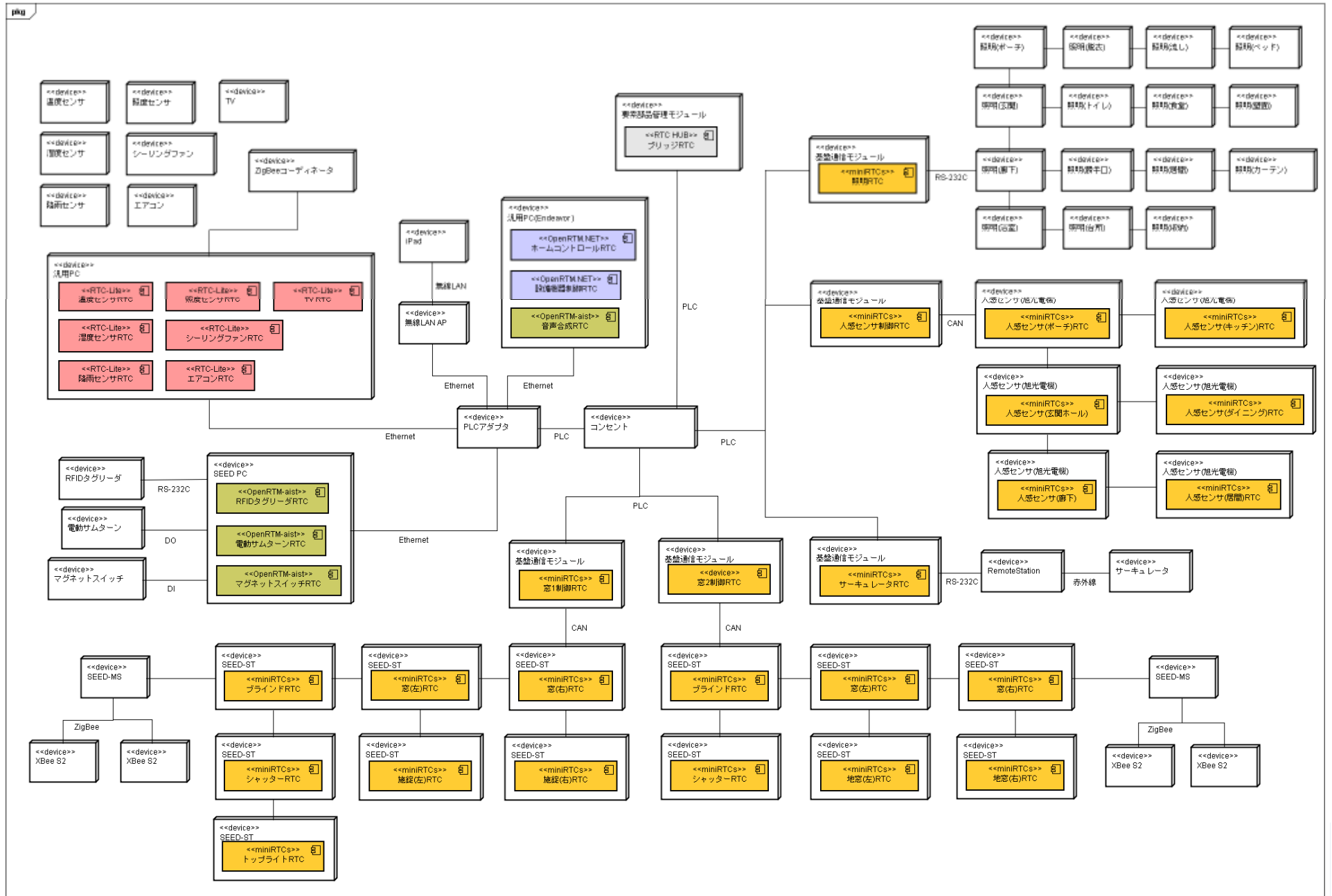
近赤外線反射方式
動線検出型人感センサー
ViS-M4000 iSearch
(旭光電機)



モータコントローラー
(オカテック/滝田技研)



オープンイノベPJの実証システムの構成

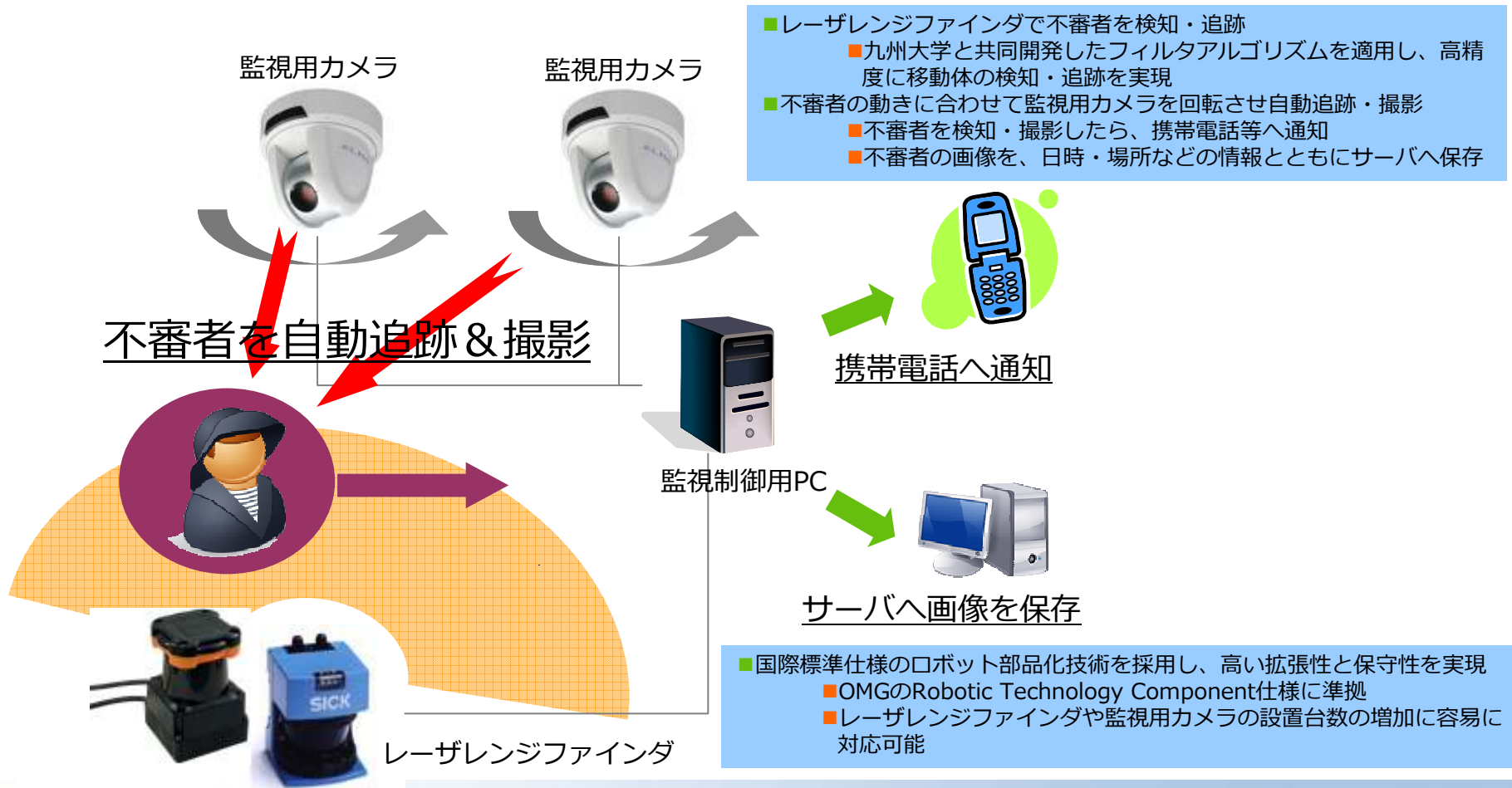


アジェンダ

- 株式会社セックのご紹介
- RTミドルウェアのシステム事例
 - さまざまなRTミドルウェア
 - ロボットシステムにおけるRTC構成例
- **RTC開発のポイント**
 - **RTCの粒度、I/Fの決め方など**
 - **RTCのテスト方法(RTCデバッグ紹介)**

ロボット防犯システム

レーザレンジファインダを使って不審者を検知し、カメラで自動追跡・撮影するロボット技術を活用した**次世代の防犯システム**です。



ロボット防犯システム



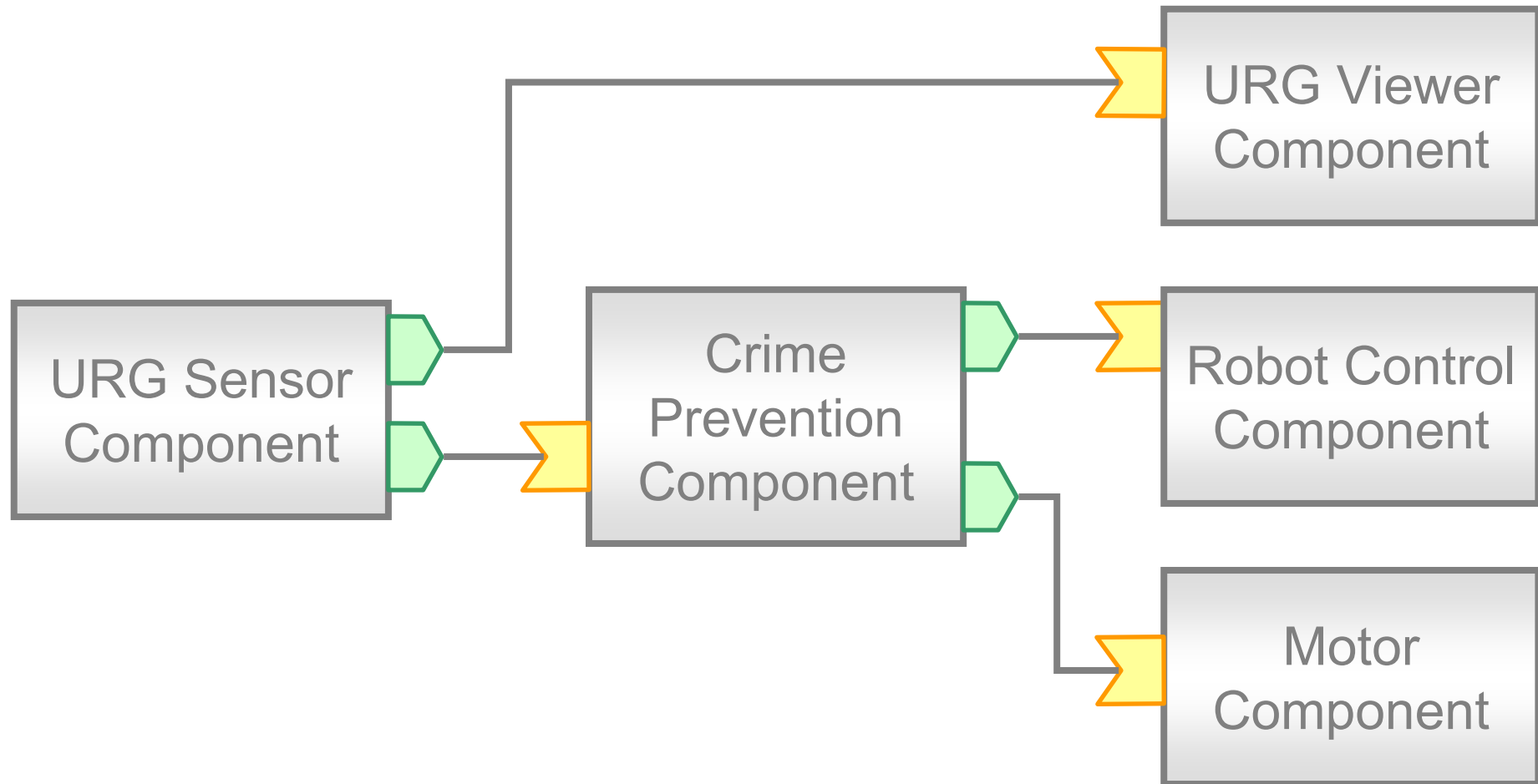
ギャラリーに設置されたロボット
警備システム

(株)セックが開発中の「人物追跡多目的ロボットシステム」を FBCC のギャラリー内に設置し、長期運用の実証実験を始めました。レーザーセンサを用いて人を探知し、位置を追跡して記録します。人の追跡の核となるアルゴリズムは、九州大学工学部倉爪教授の技術が採用されています。

この実証実験については、(株)セックの担当者が執筆した RT ミドルウェアの入門書「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～」(出版社:毎日コミュニケーションズ)でも紹介されています。

お問い合わせ:
福岡ビジネス創造センター運営委員会事務局
Tel: 092-672-7001 E-mail: info@fbcc.jp

ロボット防犯システムソフトウェア構成



コンポーネント指向

- ロボット防犯システムのソフトウェアは、コンポーネントを組み合わせて作られている



- ソフトウェアコンポーネントが備えるべき5つの基準
 - 複数使用（複数回/複数個所）
 - コンテキストに依存しない
 - 他のコンポーネントと接続可能
 - カプセル化（外部インタフェースのみ公開）
 - 独立して配布・バージョン管理可能

コンポーネント指向の利点

- コンポーネント化されていると. . .



- 再利用可能
- ロジック入れ替え可能
- 独立して開発、管理が可能



- コンポーネントの所在を問わない
→ネットワーク分散が可能

ロボット防犯システムでのメリット

- ロボット防犯システムの場合、コンポーネント化されていると. . .



- USBカメラコンポーネントを別システムに移植
- 人検知ロジックの入れ替え
- SICK社製LRFコンポーネントの開発・置き換え

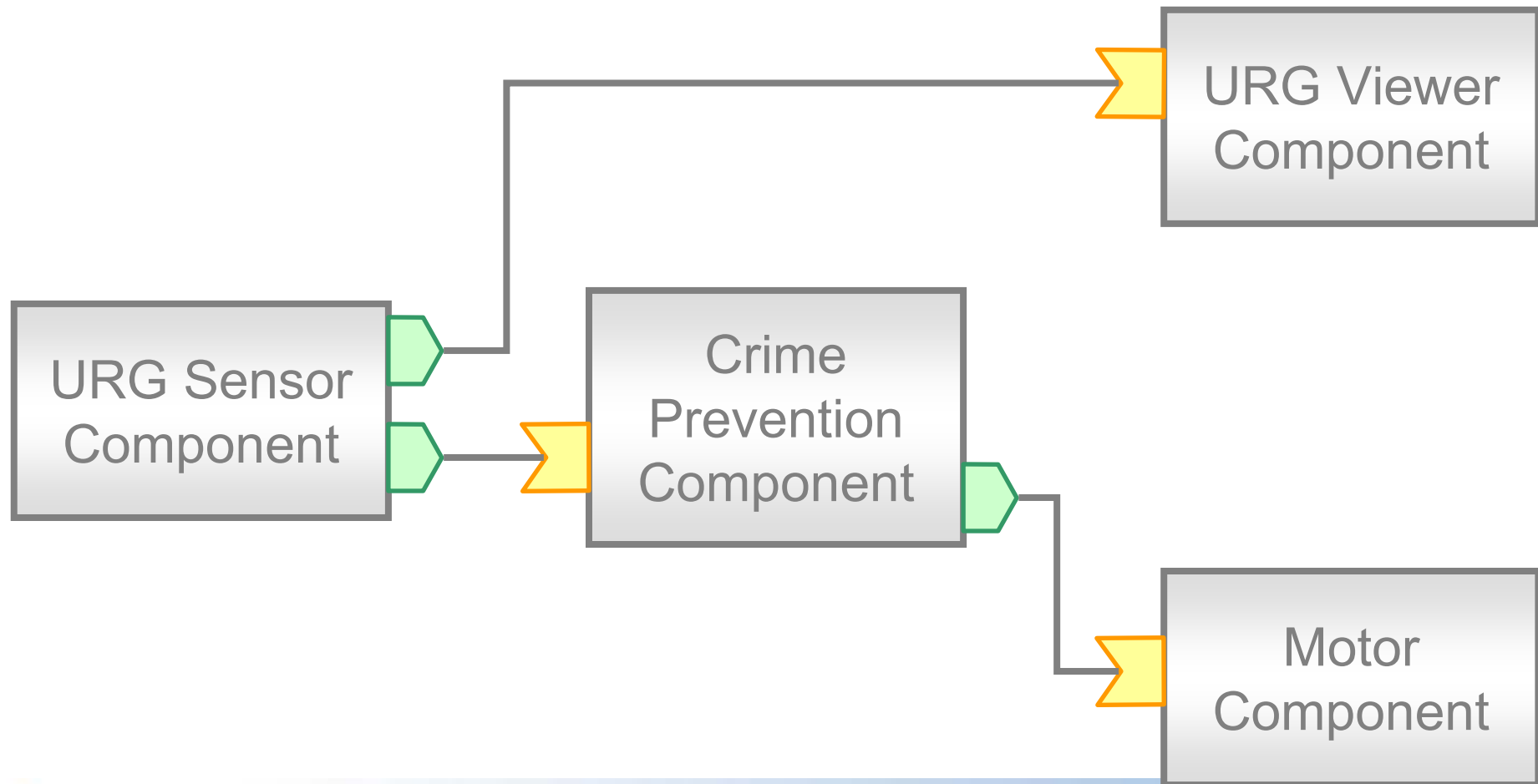


- 複数センサーの一元処理、計算機の負荷分散

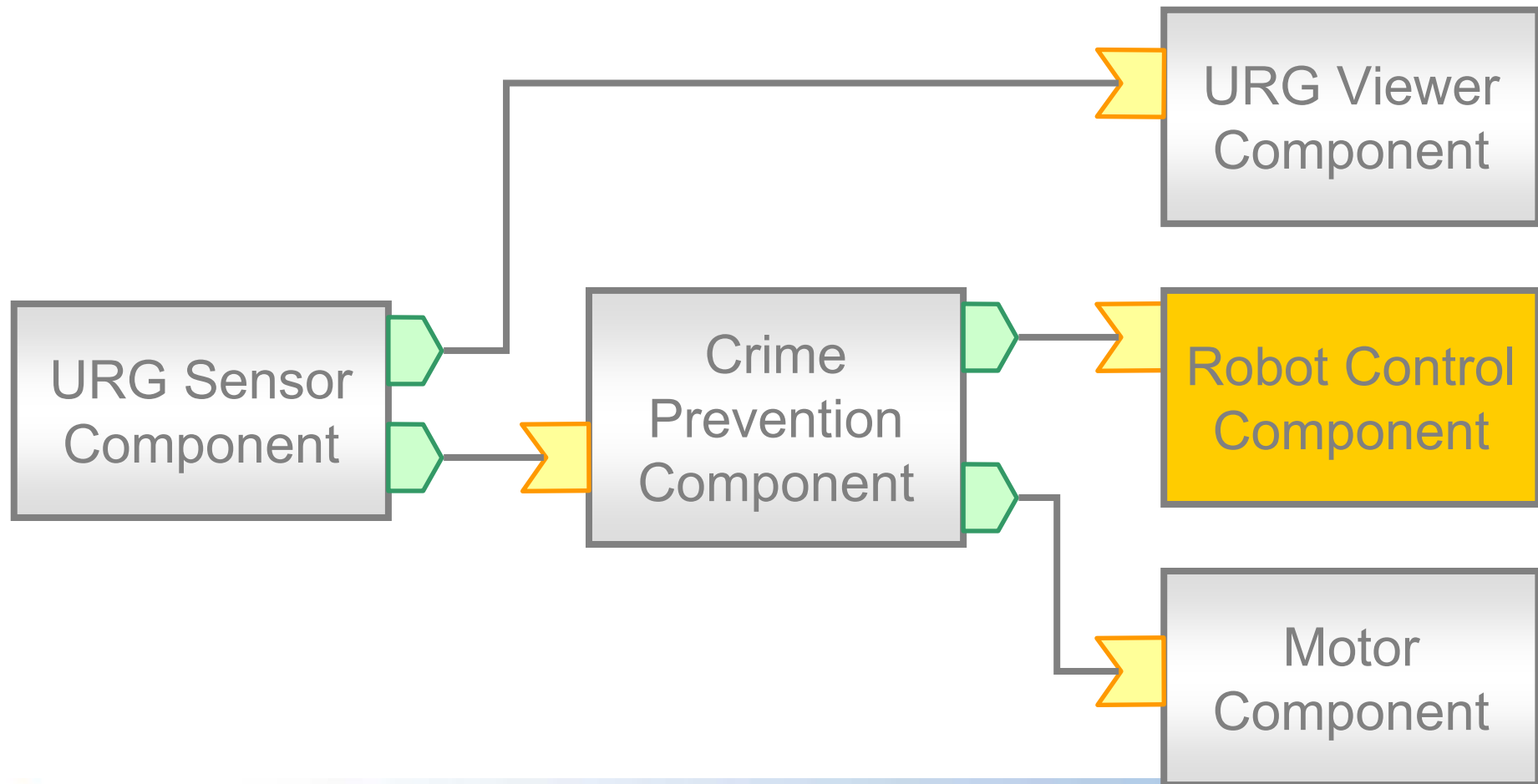
実際には. . .

- コンポーネントの積み上げで設計していたため、ロボット防犯システムは成長する

ソフトウェア構成（その1）



ソフトウェア構成（その2）

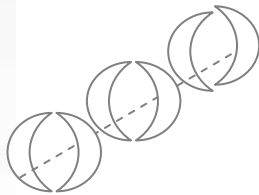


ロボット防犯システム (デモ構成)

PC for Monitoring	Pentium M(Intel) CPU 1.60GHz, 512 MB RAM OS : Windows XP
Scanning Laser Range Finder	URG-04LX(北陽電機)
USB Camera	Qcam Orbit(Logicool)



Bluetooth



Scanning Laser Range Finder

ロボット防犯システム



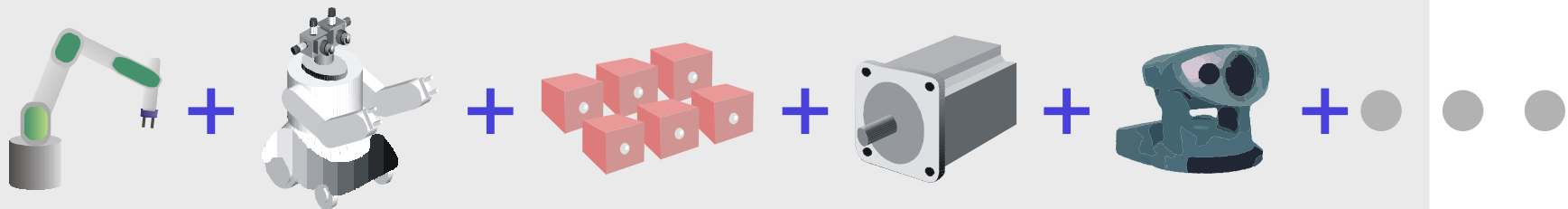
インフォメーションロボットシステム

- インフォメーションロボットシステムは、人間などの移動体を捕捉し高精度に追跡を行うことができるロボット
- 頭部に搭載した液晶ディスプレイを使って、ユーザーに様々な情報を伝えることが可能
- RTミドルウェアを採用し、高い拡張性と保守容易性を備える
- 防犯システムをはじめ、自動受付システムや店舗内の動線計測システムなど、様々な分野に応用することが可能



RTミドルウェアについて

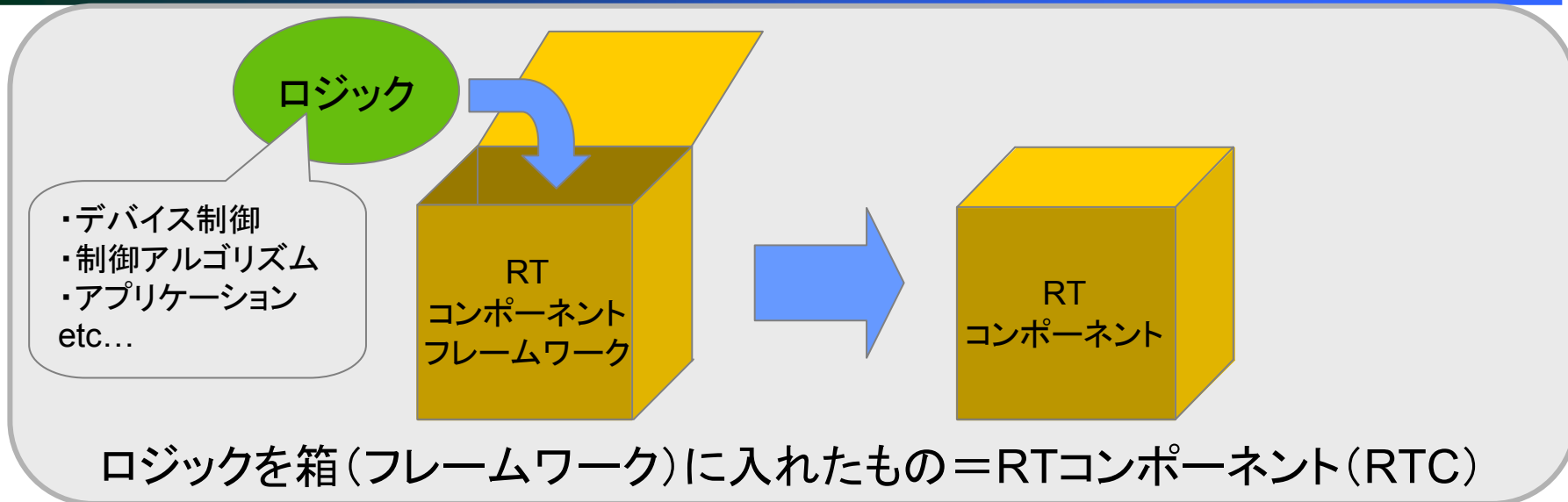
- RTミドルウェアは、ロボットを構成する要素（アクチュエータやセンサなど）やロボットを制御するソフトウェアのコンポーネント化技術
- 2008年4月に国際標準化団体OMG（Object Management Group）にて標準化
- 産総研がオープンソースのミドルウェアOpenRTM-aistを開発、提供



コンポーネントを組み合わせるだけでロボットを容易に開発

※産総研作成資料を流用

RTミドルウェアとRTコンポーネント

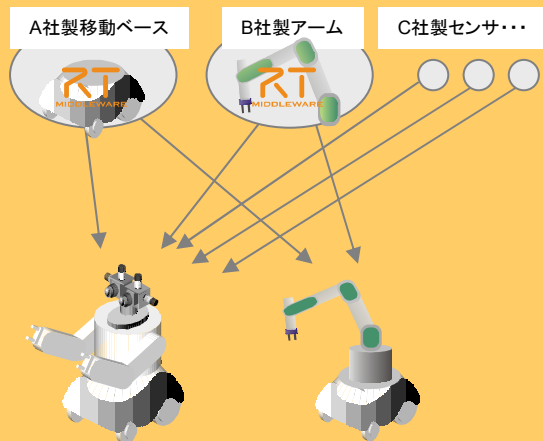


モジュール化によるアプローチ

- 仕様の明確化
- 最新技術を容易に利用可能
- 誰でもロボットが作れる

※産総研作成資料を流用

コストの問題



モジュール化・再利用

ロボットの低コスト化

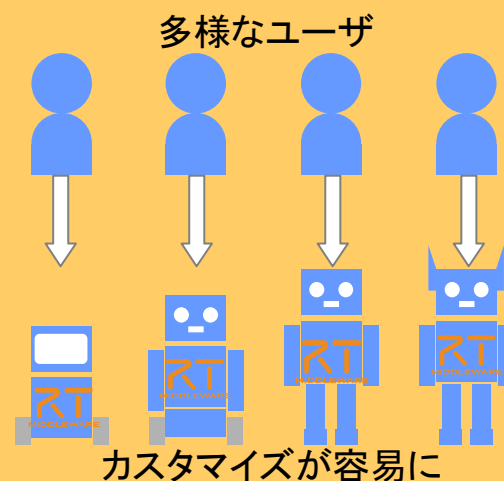
技術の問題



システム開発者

最新技術を利用可能

ニーズの問題



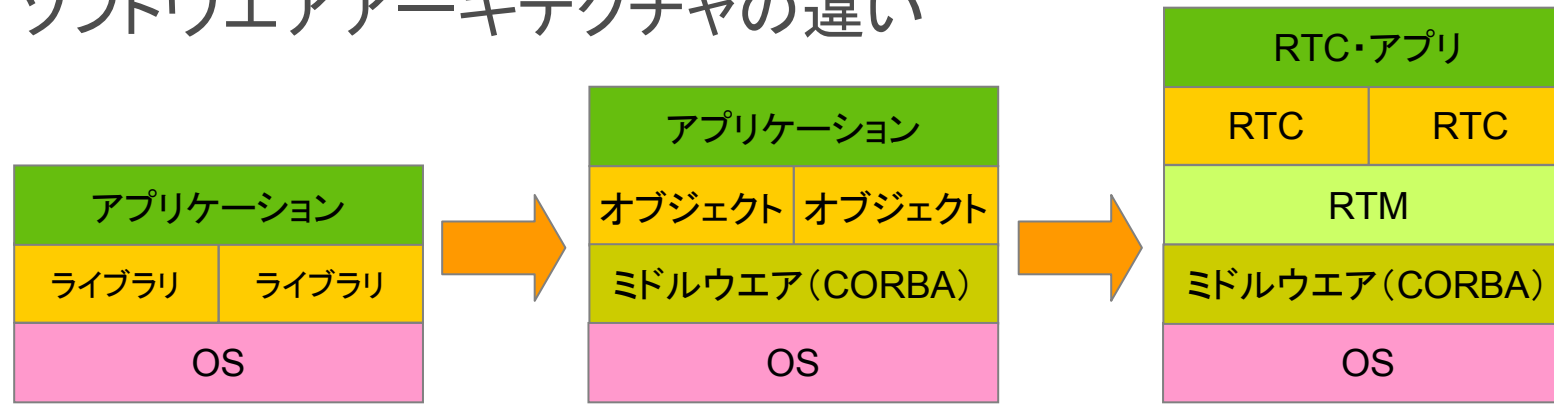
カスタマイズが容易に

多様なニーズに対応

RTコンポーネントとは？

ソフトウェアアーキテクチャの違い

※産総研作成資料を流用



従来ソフトウェアから分散オブジェクトへ

- オブジェクト指向開発
- 言語・OSの壁を越えて利用できる
 - インターフェースをIDLで定義
 - 各言語へ自動変換
 - OS、アーキテクチャの違いを吸収
- ネットワーク透過に利用できる
 - 分散システムを容易に構築可能

分散オブジェクトからRTCへ

- インターフェースがきちんと決まっている
 - IDLで定義された標準インターフェース
 - 呼び出しに対する振る舞いが決まっている(OMG RTC 標準仕様)
 - 同じ部品として扱える
- コンポーネントのメタ情報を取得することができる
 - 動的な接続や構成の変更ができる
- ロボットシステムに特有な機能を提供
 - 後述

モジュール化のメリット

■ 再利用性の向上

- 同じコンポーネントをいろいろなシステムに使いまわせる

■ 選択肢の多様化

- 同じ機能を持つ複数のモジュールを試すことができる

■ 柔軟性の向上

- モジュール接続構成を変えるだけで様々なシステムを構築できる

■ 信頼性の向上

- モジュール単位でテスト可能なため信頼性が向上する

■ 堅牢性の向上

- システムがモジュールで分割されているので、一つの問題が全体に波及しにくい

RTコンポーネント化のメリット

- モジュール化のメリットに加えて
- ソフトウェアパターンを提供
 - ロボットに特有のソフトウェアパターンを提供することで、体系的なシステム構築が可能
- フレームワークの提供
 - フレームワークが提供されているので、コアのロジックに集中できる
- 分散ミドルウェア
 - ロボット体内LANやネットワークロボットなど、分散システムを容易に構築可能

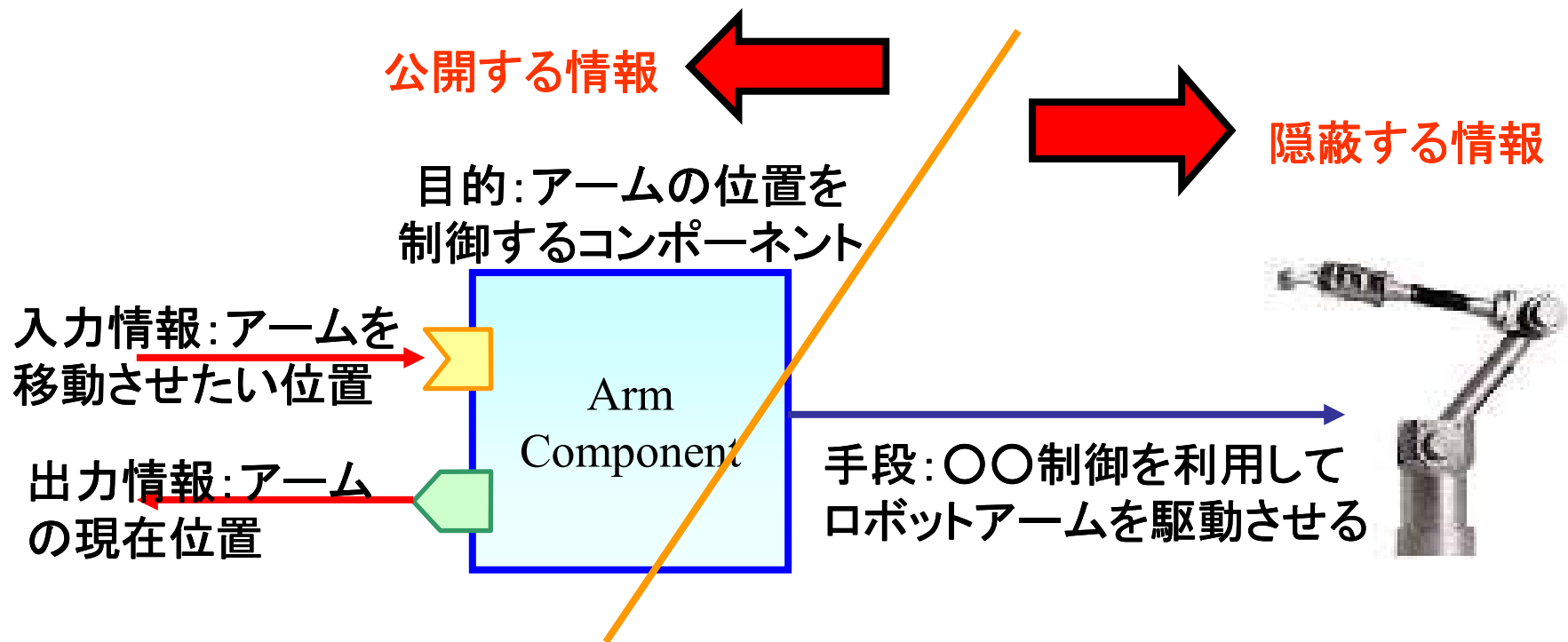
コンポーネント指向設計

■ コンポーネント指向設計とは

- ロボットシステムを構築するために必要なコンポーネントを抽出する
- コンポーネントの粒度を考える
- コンポーネントが外部に提供する情報を考える
- コンポーネント間でやり取りされるデータを考える

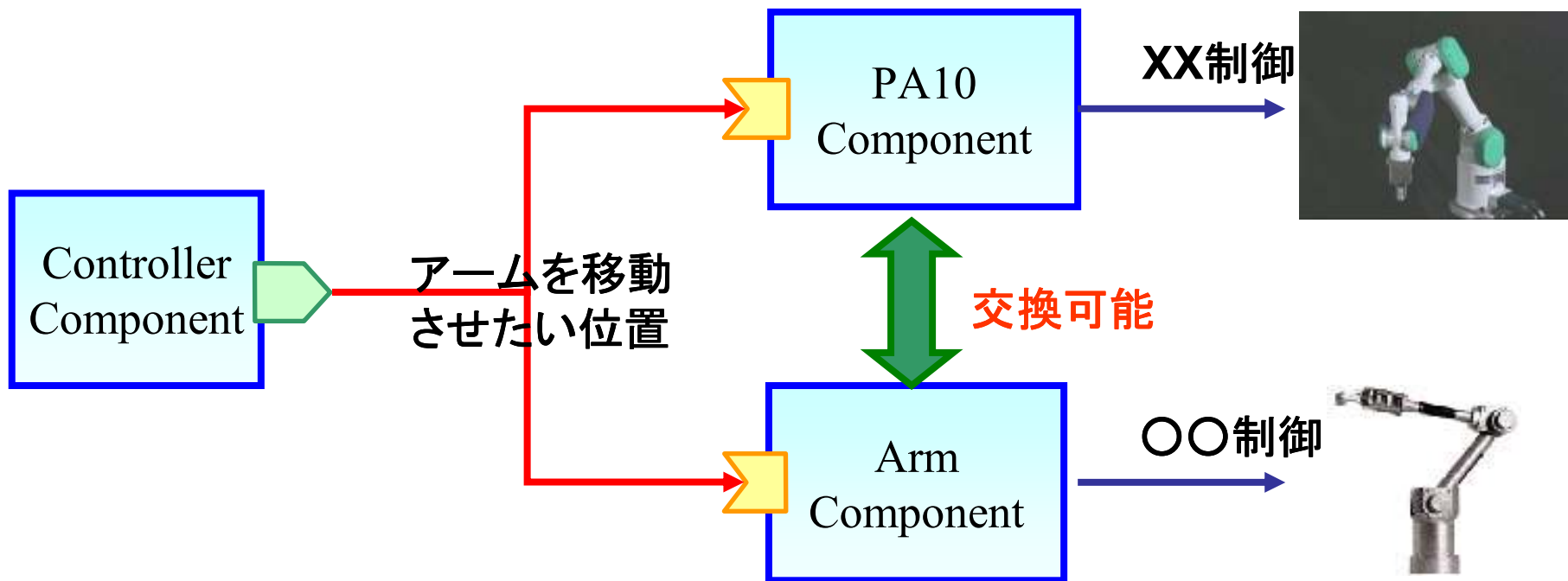
コンポーネント指向設計のコツ

- 何をするか(目的：What)を公開し、
どのようにするか(手段：How)は隠蔽する。



なぜ手段の隠蔽をおこなうのか？

- 目的の等しいコンポーネントを簡単に交換することが可能になる！



コンポーネントの粒度

■ 粒度とは

- ロボットシステムをコンポーネントに分割する際の単位

■ 小さな粒度のコンポーネント

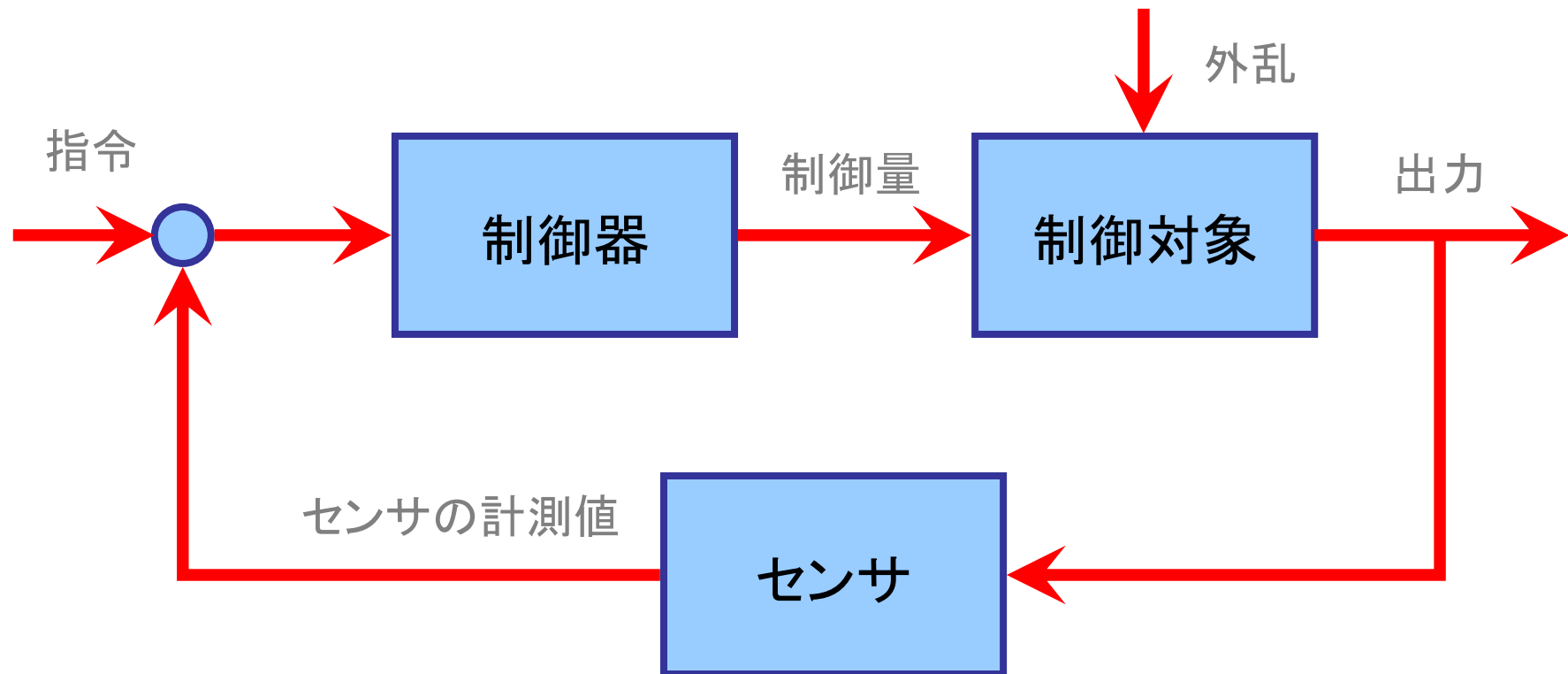
- モータ速度制御コンポーネント
- 光センサコンポーネント

■ 大きな粒度のコンポーネント

- お手伝いロボットコンポーネント

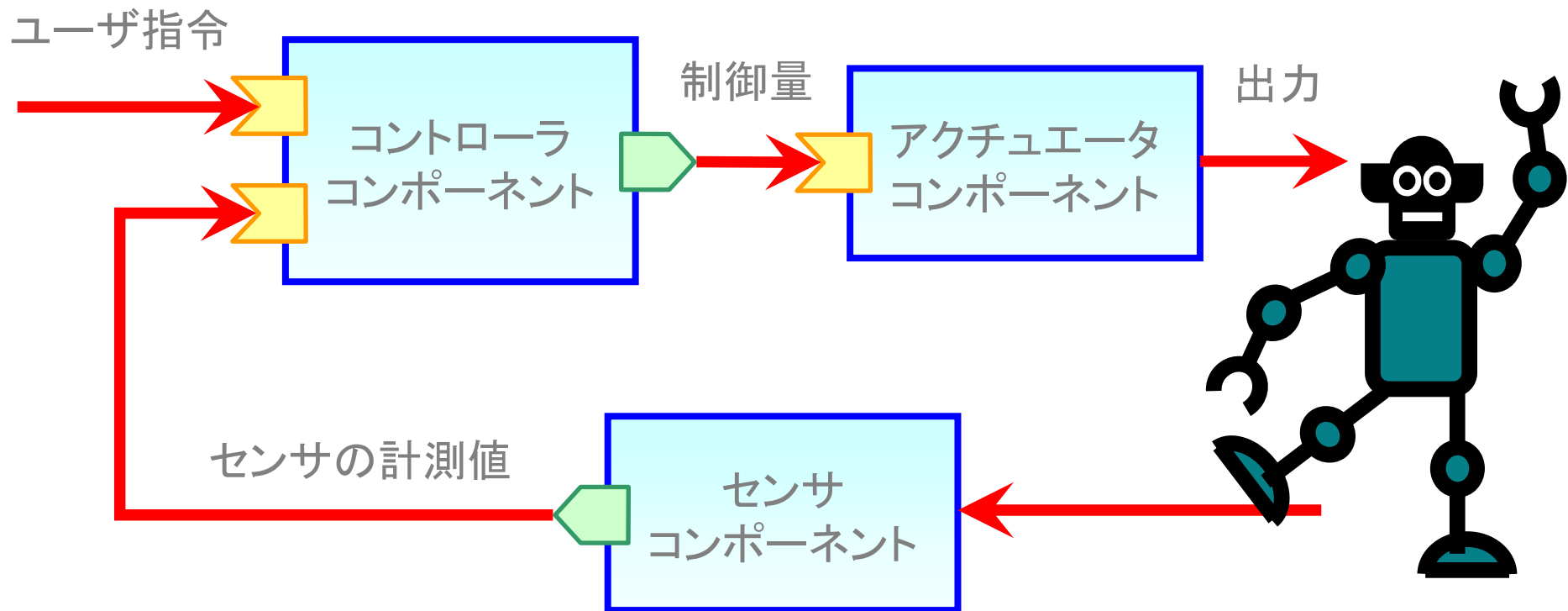
ロボット制御の粒度

- ロボットの制御は、ブロック線図で表現されることが多いですが・・・



コンポーネントによる制御

- コンポーネントをブロック線図のブロックと同様の単位で考えることもできます。



コンポーネントのカテゴリ分け

■ アクチュエータコンポーネント

- アクチュエータを動作させるコンポーネント。
- 例：アームコンポーネント、移動ロボットコンポーネント

■ センサコンポーネント

- センサの計測データを取得するコンポーネント。
- 例：超音波センサコンポーネント、カメラコンポーネント、カセンサコンポーネント

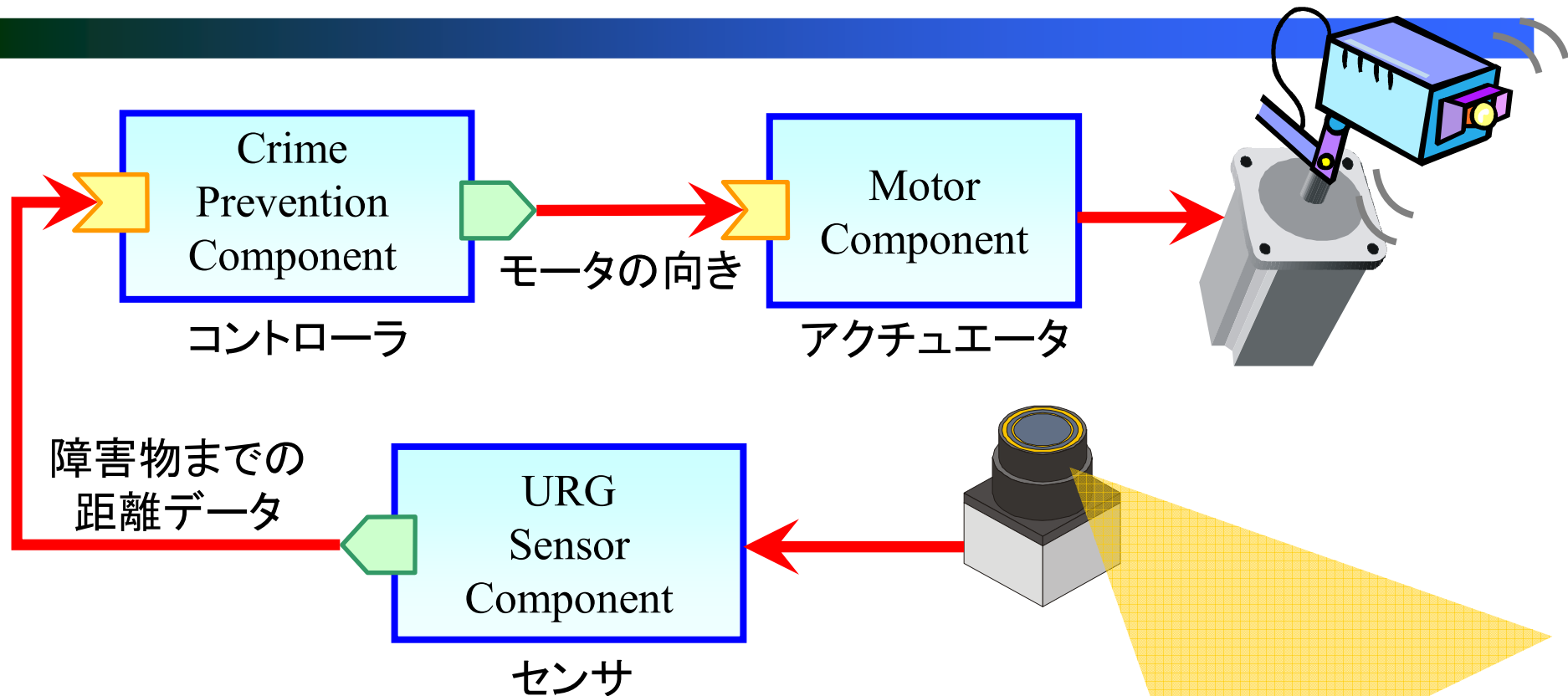
■ コントローラコンポーネント

- センサ入力やユーザの指令に応じて、ロボットの行動を決定するコンポーネント。
- 例：人物追跡コンポーネント、顔認識コンポーネント

入出力ポートの設計

- コンポーネント間でやり取りする必要のあるデータについて検討する。
 - コンポーネントの目的達成のために必要な情報
→ 入力ポート
 - コンポーネントが動作した結果、得られた情報
→ 出力ポート
- 手段に依存しないデータ形式が望ましい。
 - 悪い例： ○○制御のためのパラメータ
 - 良い例： アームの目標指先位置
x,y,z[mm]

ロボット防犯システムの場合



- アクチュエータ、センサ、コントローラのカテゴリに分類することが可能。
- 入出力データの形式は汎用的

コンポーネントのI/Fガイドライン

- コンポーネント単独ではロボットシステムは構築できないため、コンポーネントがつながることが大事



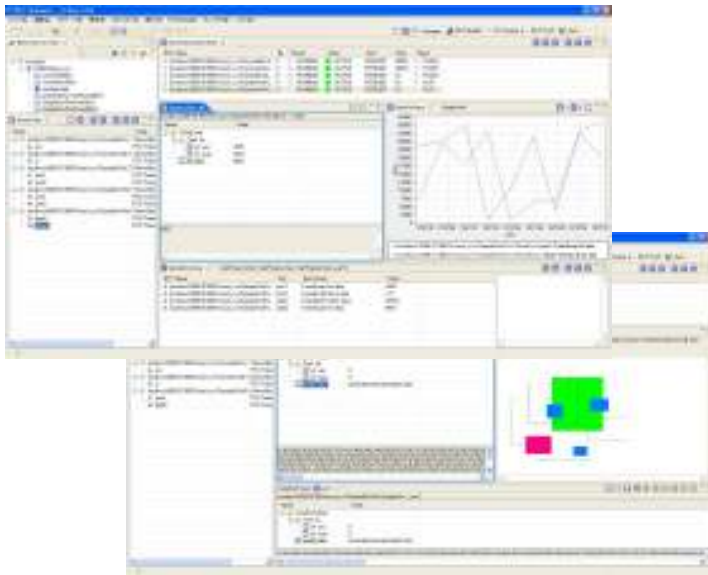
- コンポーネントのI/Fを共通化することが重要



- OpenRTM-aistで定義されているBasicDataTypeやExtendedDataTypesなど標準化されたデータを使うようにしよう！

RTコンポーネントデバッガ

RTコンポーネントデバッガは、RTコンポーネントの動作を検証するためのテストツールです。RTコンポーネントデバッガを使うと、RTコンポーネントのテストを効率的におこなうことができます。



RTコンポーネントデバッガの機能

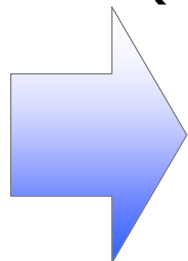
- RTコンポーネントの状態制御
 - データポートへのデータ入出力
 - サービスポートの呼び出し
 - データの視覚化
 - データの記録/再生
- etc...

RTコンポーネントデバッガの動作環境

OS	Javaの動作する各種OS
Java	JDK 6 Update 10以降
Eclipse	Eclipse 3.2以降
RTC仕様	OMG RTC Specification 1.0
RTミドルウェア	OpenRTM-aist-0.4.2/1.0.0 OpenRTM.NET-0.4/1.0/1.1

RTCデバugga

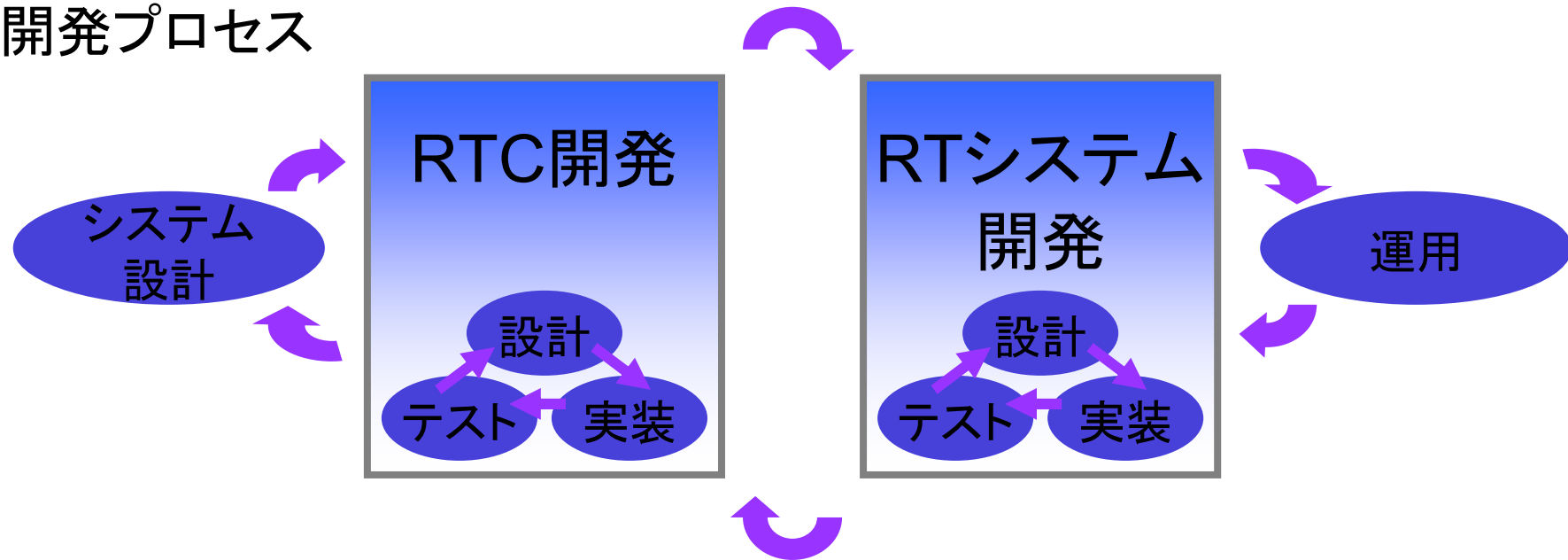
- NEDOの委託事業である「次世代ロボット
知能化技術開発プロジェクト」にて、
OpenRT Platformが開発されている
 - OpenRT Platform : RTミドルウェアを基盤と
したソフトウェアプラットフォーム
 - モジュール化の単位 : RTコンポーネント
(RTC)



RTCデバugga : OpenRT Platformに
おけるRTCの検証ツール

OpenRT Platformの開発プロセス

開発プロセス



主なツール

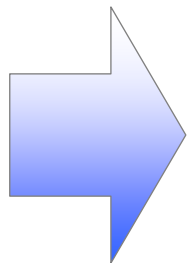
RTCビルダ
(RtcTemplate)
RTCデバッガ

RTシステムエディタ
(RtcLink)
RTCシミュレータ

RTC単体を作成 ⇒ RTシステムに組込

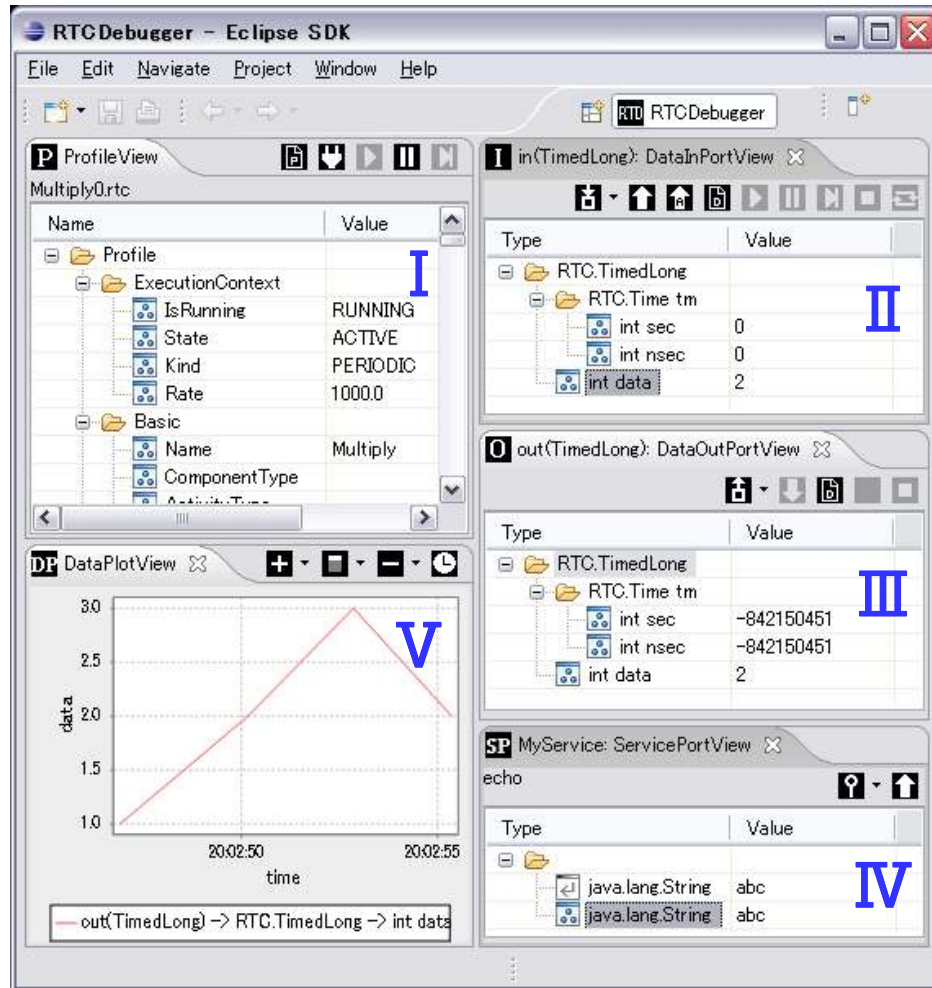
RTCデバッガの開発経緯

- 高品質なシステム構築のためには？
 - システムに組み込む前に、RTC単体で十分なテストやデバッグが必要
- 現状：各々の開発者が、テスト用のドライバやスタブを作成
 - RTCの開発効率は良い状況とはいえない



RTC検証の汎用的なツールとして、
RTCデバッガを開発

RTCデバッガ オーバービュー



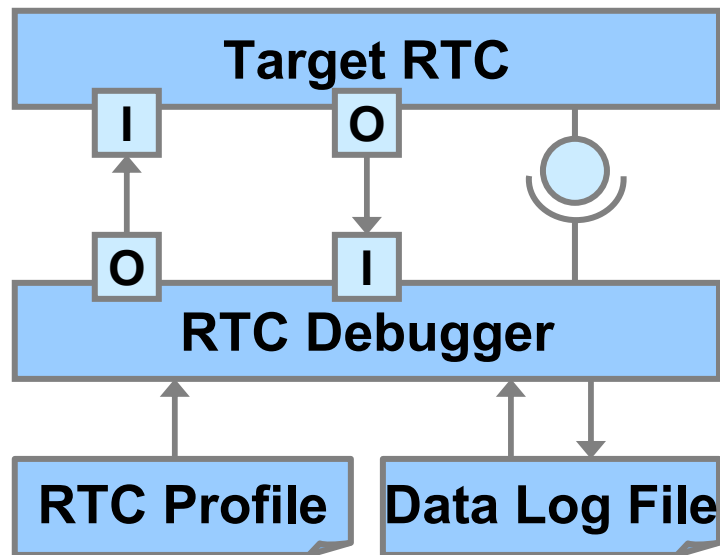
■ Eclipse Plug-inとして開発

- I プロファイルビュー
- II データ入力ポートビュー
- III データ出力ポートビュー
- IV サービスポートビュー
- V データプロットビュー

RTCデバッグ機能一覧

- 実行コンテキスト制御機能
- データ入力ポート検証機能
- データ出力ポート検証機能
- サービスポート検証機能
- データ記録/再生機能
- データプロット機能

RTCデバッグ機能構成



■ RTC Profile

- RTCビルダにより生成される、RTC仕様記述ファイル

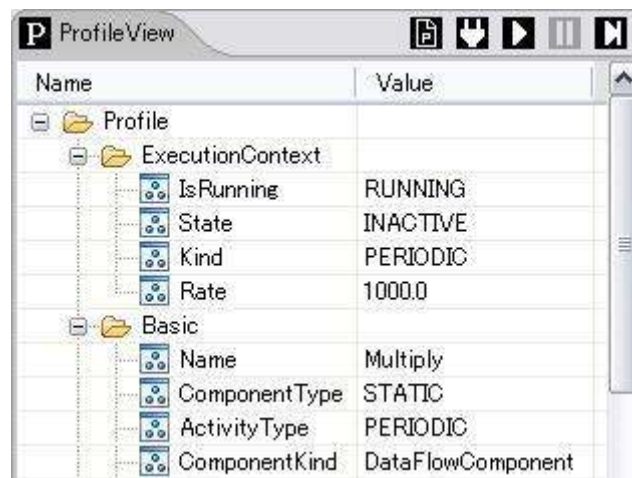
■ Data Log File

- RTCデバッガが受信したデータを蓄積したファイル（※RTCのデータ出力ポートから受信）

実行コンテキスト制御機能

■ ExecutionContextの制御

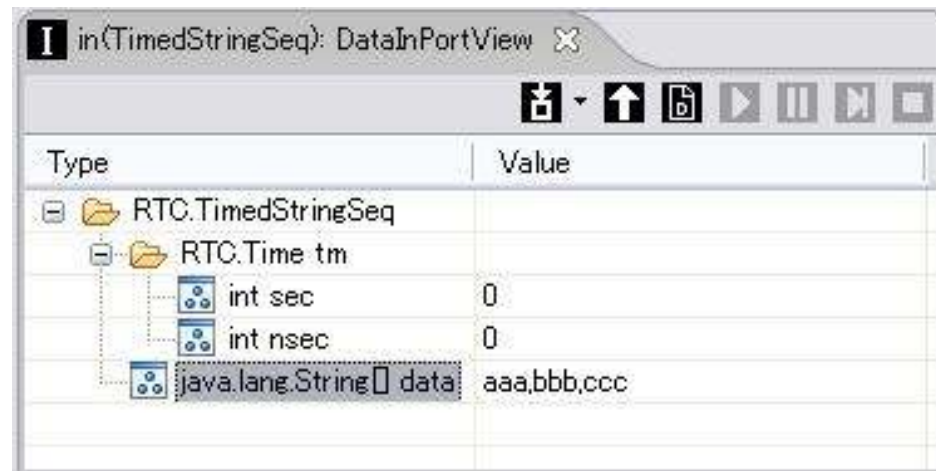
- 一時停止、ステップ実行(OnExecute1回実行)



プロファイルビュー

データ入力ポート検証機能

- 画面で入力したデータを送信
- MIN/MAX/インターバルを指定しての自動送信

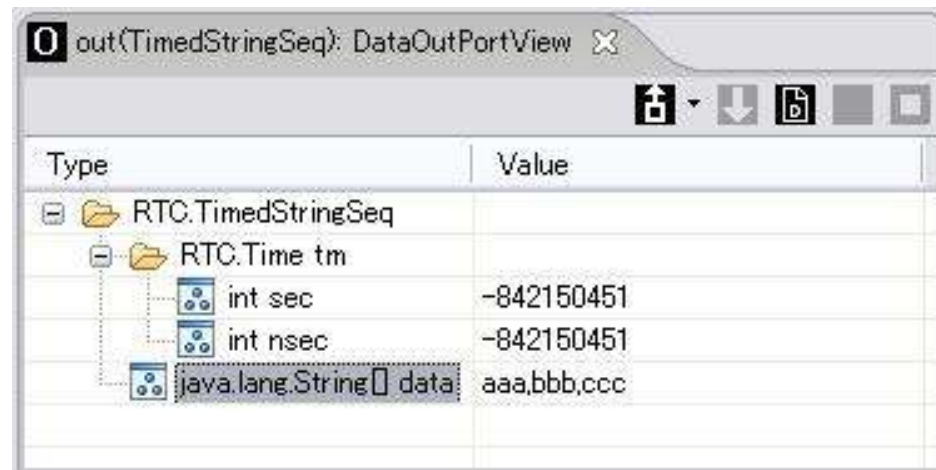


データ入力ポートビュー

TimedStringSeq型データの送信

データ出力ポート検証機能

- RTCデバッガで受信したデータをリアルタイムに表示



The screenshot shows a debugger window titled "out(TimedStringSeq): DataOutPortView". It displays a tree view of the data structure and a table of its values.

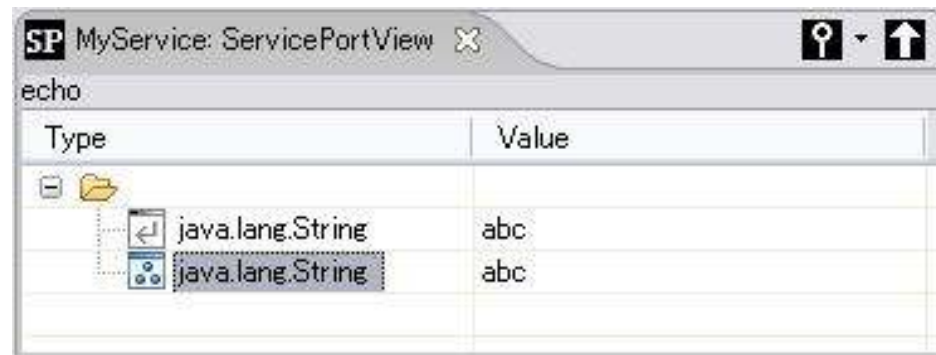
Type	Value
RTC.TimedStringSeq	
RTC.Time tm	
int sec	-842150451
int nsec	-842150451
java.lang.String[] data	aaa,bbb,ccc

データ出力ポートビュー

TimedStringSeq型データの受信

サービスポート検証機能

- サービスポート(Provider)のインタフェースを、画面で引数を入力して呼び出し
- 戻り値を表示



サービスポートビュー

string echo(in string msg)の呼出

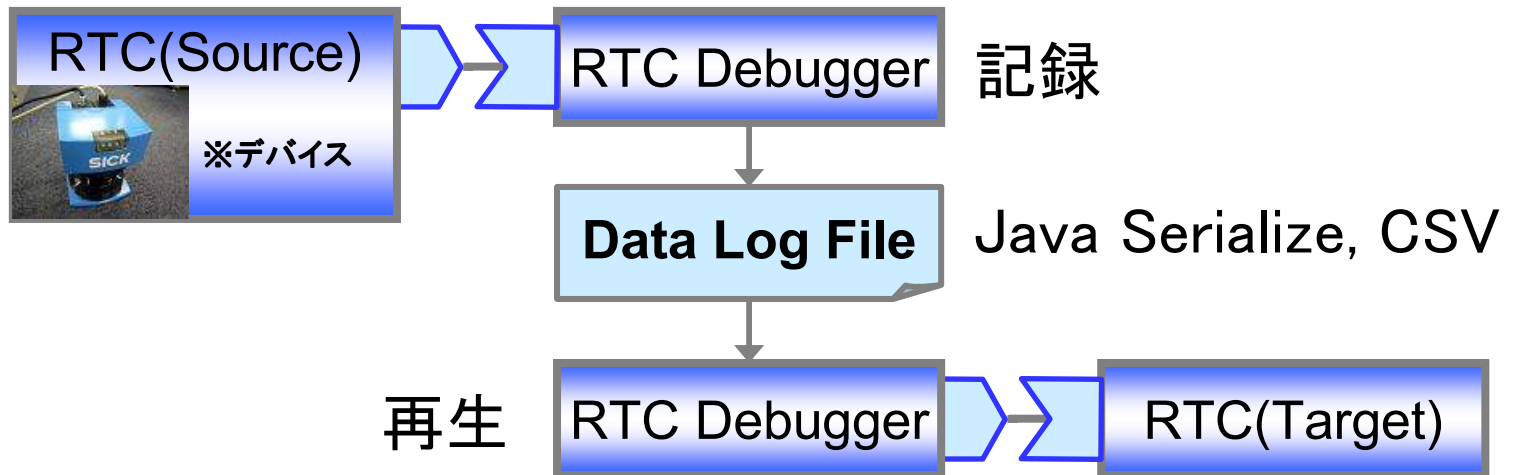
データ記録/再生機能

通常



一般的に、再現は容易ではない

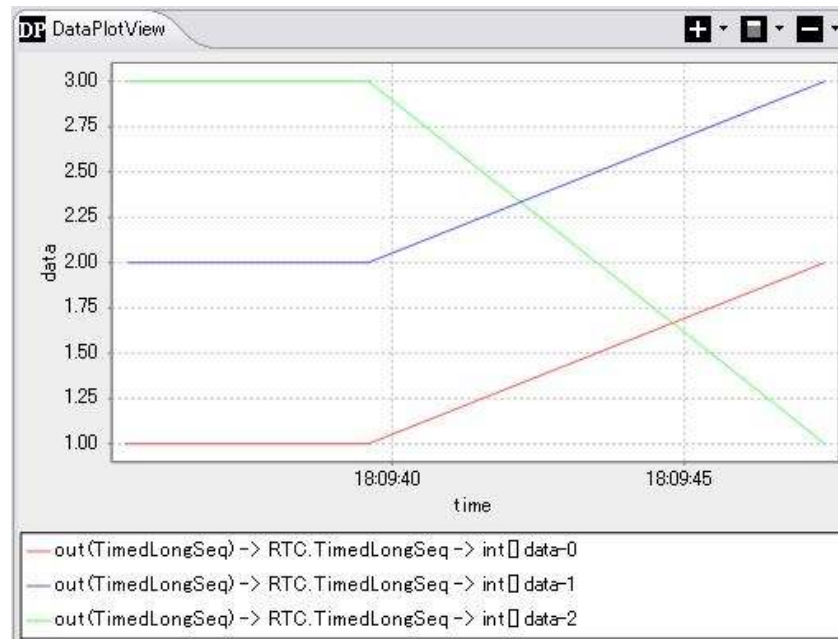
RTCデバッガを利用



再現性の確保、ハード使用面の省力化

データプロット機能

- 縦軸(データ) : Auto or 固定
- 横軸(時間) : ユーザ指定(最新 X ミリ秒間)



データプロットビュー

データプロット機能

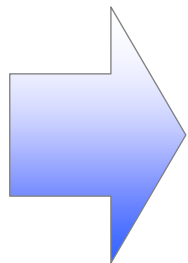
The screenshot displays the Eclipse SDK interface for the RTC Debugger. The main window is titled "DataPlotView" and shows a polar plot. The plot has a circular grid with radial lines at 45-degree intervals (0, 45, 90, 135, 180, 225, 270, 315 degrees). A red line represents the data being plotted, showing a complex, multi-lobed pattern. The plot is titled "PolarPlotView" and is part of the "Sample_testPolarPlotView20.rtc" project.

Other visible windows include:

- ExecutionContextView:** A table listing running RTCs with columns for Name, No, Thread, State, Kind, Rate, and Player.
- PropertyView:** A table showing properties for the selected RTC, including "PolarPlotTestType2" with sub-properties "char[] data1", "int[] data2", and "short[] data3".
- AttachView:** A tree view showing the system architecture with components like "ConsoleIn0|rtc", "ConsoleOut0|rtc", "manager|mer", "MyServiceConsumer0|rtc", "MyServiceProvider0|rtc", and "Sample_testPolarPlotView20|rtc".
- DataWatchView:** A table for monitoring data from various RTCs.

RTCの開発効率と品質の向上

- 適用したRTC単体のSPEC
 - 力覚センサのRTC
 - ソースコードのステップ数：2KStep
 - データ出力ポート数：2個
 - サービスポート[Provider]数：2個—関数：5個
- RTC単体のテスト工数が40%減少



ドライバ/スタブの作成が不要

データの可視化によるスムーズな動作確認

レーザレンジファインダコンポーネント(URG)

株式会社セック

概要 :

北陽電機株式会社製のレーザ測域センサ
URG-04LX(以降、URGセンサ)/UTM-30LX
(以降、TOP-URGセンサ)向けRTコンポーネント

特徴 :

- ◆センサ通信プロトコルがSCIP2に準拠しているセンサであれば利用可能
- ◆距離データのリアルタイム出力
- ◆センサステータスの取得、パラメタ設定が可能
- ◆インタフェースを共通化したOpenHRP3向けシミュレータRTコンポーネントの利用が可能
- ◆利用マニュアル完備

インタフェース :

ポート名	結果またはパラメタ
結果出力 (データポート)	距離データを出力(mm) 距離データを出力(m)
結果取得、パラメタ設定 (サービスポート)	距離データ、センサステータスを取得 各種パラメタを設定

(OpenRTM-aist-1.0.0-RELEASE、Windows / Linux)
(OpenRTM.NET-1.x、Windows)



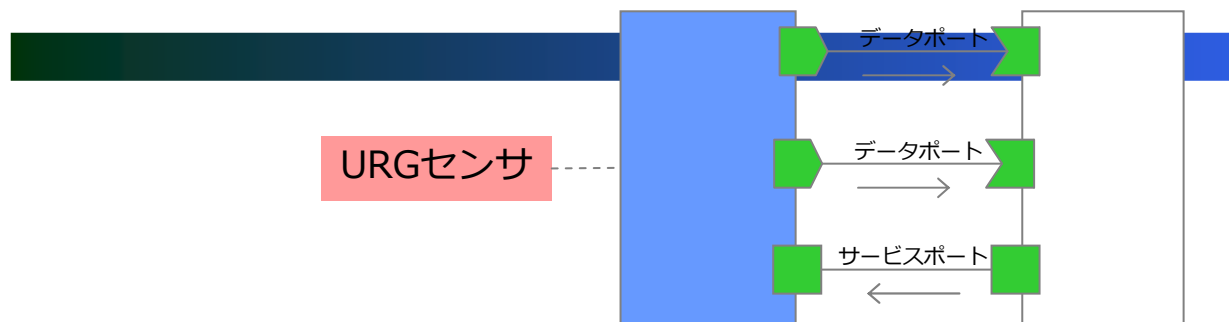
ライセンス(公開条件) :

非営利・非商用での利用に限り、バイナリファイルにて無償公開しています。商用利用やソースコードの提供に関しましては、別途ご相談ください。

version : 1.2
Date : 2011.05.16

詳細インターフェース：

URGセンサ コンポーネント コントローラ コンポーネントなど



URGセンサ

結果取得、パラメタ設定 (サービスポート)

- ・ URGセンササービス
 - センサリセット
 - 最新距離データの取得
 - センサステータス情報の取得
 - 各種パラメタの一括設定
 - 計測開始終了位置の取得
 - 計測開始終了位置の設定
 - スキャン間隔の取得
 - スキャン間隔の設定
 - 距離データまとめ数の取得
 - 距離データまとめ数の設定

URGセンサ仕様

インタフェース	RS-232C、USB2.0
有効計測エリア角	240[deg](センサ前方)
計測角度分解能	0.35[deg]
最大出力距離データ数	768[個]
モータ回転速度	600[rpm]
計測可能距離	5600[mm]

TOP-URGセンサ仕様

インタフェース	RS-232C、USB2.0
有効計測エリア角	270[deg](センサ前方)
計測角度分解能	0.25[deg]
最大出力距離データ数	1080[個]
モータ回転速度	2400[rpm]
計測可能距離	60000[mm]

- ・ センサステータス情報
 - タイムスタンプ
 - 計測開始/終了位置[deg]
 - 各計測データの間隔[deg]
 - 計測最小距離[mm]
 - 計測最大距離[mm]
 - 計測データ数
 - スキャン間引き数
 - まとめるステップ数
 - スキャン周期[ms]
 - センサ型式情報
 - モータ回転速度[rpm]
 - 通信速度[bps]
 - 計測モード
 - センサ状態
 - センサのバージョン情報
 - 計測範囲の単位
 - 計測値の単位

コンフィギュレーション

- ・ 通信速度[bps]
- ・ センサ接続時に認識されたデバイス名
- ・ スキャン間引き数
- ・ 計測モード
- ・ 計測開始/終了位置[deg]
- ・ まとめる方向
- ・ 高感度モード
- ・ モータ速度減速率

結果出力(データポート)

- ・ 距離データ[mm]
 - タイムスタンプ
 - 計測開始位置[deg]
 - 計測終了位置[deg]
 - スキャン間引き数
 - まとめるステップ数
 - 計測値[mm]
 - 各計測データの間隔[deg]
 - センサ状態
- ・ 距離データ[m]
 - タイムスタンプ
 - 計測開始位置[rad]
 - 計測終了位置[rad]
 - 計測値[m]
 - 各計測データの間隔[rad]



Realtime@net

セックはリアルタイム技術専門のソフトウェア会社です