

ロボットミドルウェア

安藤慶昭

国立研究開発法人産業技術総合研究所
ロボットイノベーション研究センター
主任研究員



概要

- ロボット革命と「ロボット新戦略」
- ロボットミドルウェアとは？
- RTミドルウェア : OpenRTM-aist
- 様々なミドルウェア/プラットフォーム
- 終わりに

ロボット革命と「ロボット新戦略」

ロボット新戦略

- 日本経済再生本部決定文書
 - ロボット革命実現会議の取りまとめ

内容

- 第1部 総論
 - 現状分析
 - 革命実現のための方策(3つの柱)
- 第2部 アクションプラン
 - 分野横断事項
 - 協議会設置・次世代技術開発・標準化・人材育成等
 - 分野別事項
 - ものづくり・サービス・インフラ/災害対応/建設・農林水産/食品産業分野

ロボット新戦略

Japan's Robot Strategy

— ビジョン・戦略・アクションプラン —

日本経済再生本部
2015年2月10日

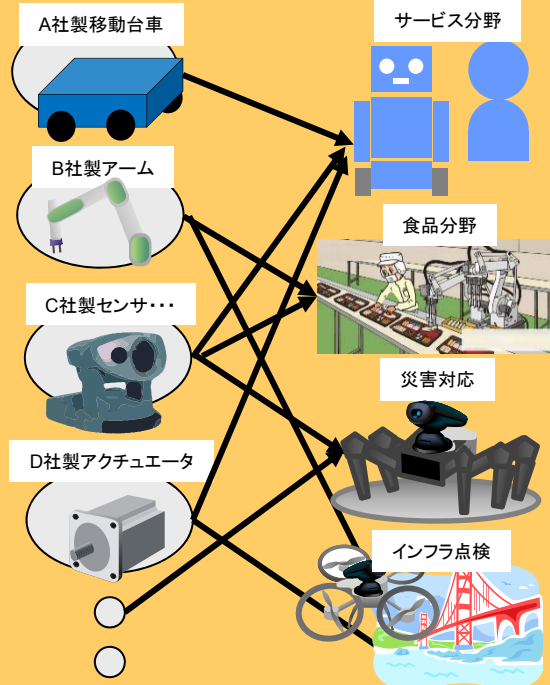
ロボット新戦略の重要性

関係各省庁も議論に参加し、内容の詳細に関与し合意したロボット戦略

→ 今後の施策に大きな影響

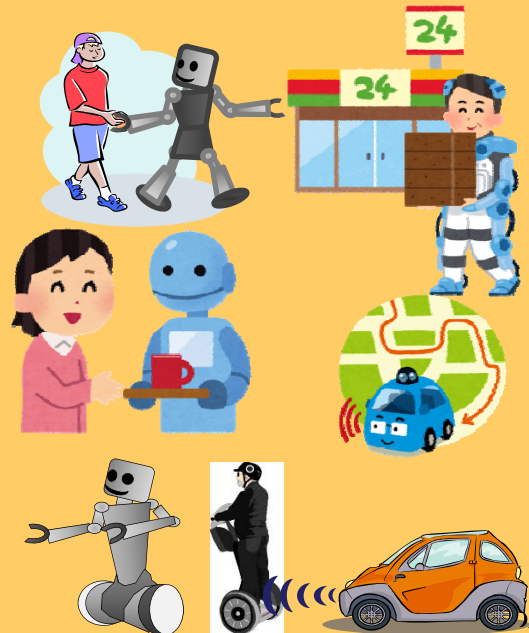
ロボット新戦略：3つの柱

ロボット創出力強化



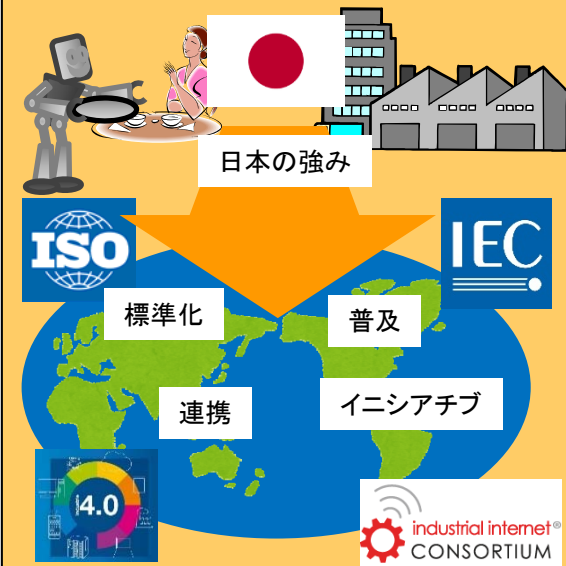
多様な分野の要請に柔軟に対応できるロボット創出力強化とイノベーションを誘発する場の創出

ロボットの活用・普及



サービス、介護等様々な分野でロボットの開発・導入を進める環境整備の実施

世界を見据えたロボット革命の展開・発展



Industry4.0やIndustrial Internet等世界的な潮流の中で、ものづくりを引き続きリードするロボット技術の世界展開

(ロボット創出力を) **強化**

(メーカーと地域を) **つなぐ**

(市場を) **広げる**

ロボット新戦略とロボットミドルウェア(1)

p.13

第1部 総論 第2章 ロボット革命実現のための方策

第1節 ロボット創出力ー日本のロボットを徹底して強化する

第3項 次世代に向けて

(2) 国際的な展開を見据えた規格化・標準化

我が国において開発されたロボットの国際的な展開を見据えると、予め国際標準や規格を構築し、それに準拠する形で実用化を目指すことが重要である。

その際、今後、複数のロボットが連携しシステムとして機能することが必要となってくることを踏まえ、個々のデバイスのほか、**ミドルウェア(ロボットOS)等のソフトウェア・インターフェース、通信等の機器間連携、そしてこれらの機能評価に関し、規格化・標準化に重点的に取り組む。**

ロボット新戦略とロボットミドルウェア(2)

p.23

第2部 アクションプラン—五カ年計画

第1章 分野横断的事項

第2節 次世代に向けた技術開発

(3) 開発すべき技術

④OS・ミドルウェア等

ロボットやロボットシステムを構成するためには、要素技術・部品・ロボットそのものをOS・ミドルウェア等の基盤的なソフトウェアで統合する必要がある。また、これにより互換性、開発の生産性も高めることが可能となる。そうした技術について、現状では、

- (中略)使い勝手が良く一定程度標準化されたOS・ミドルウェア・プログラミング言語等を、将来の要素技術の発展に対応させる必要がある。
- 異なるOSのロボット同士が対話する場合、あるいは、ロボットに新たなモジュールを搭載する場合など、ロボット及びモジュールのインターフェースを標準化する必要がある。

等の課題が存在している。こうした課題に対して、動作環境を模擬するシミュレータや、シミュレータと連携可能なOS・ミドルウェア、また、標準になり得る汎用的なOS・ミドルウェア等の研究開発が必要と考えられる。

ロボット新戦略とロボットミドルウェア(3)

pp.27-32

第2部 アクションプランー 五カ年計画

第1章 分野横断的事項

第3節 ロボット国際標準化への対応

第1項 ハードウェア／ソフトウェアのモジュール化を見据えた共通基盤 (ミドルウェア・ロボット OS)

(中略) ロボットを構成するハードウェア、ソフトウェアの機能要素をモジュール化・共通化することで、(中略) ロボットを低価格で構成することが可能となる。また、共通のソフトウェアプラットフォームを利用することで、(中略) ロボットインテグレータは対象毎の機能の実現に注力することができる。

(3) 2020 年に目指すべき姿

ロボットによる産業革命を日本主導で実現するためにも、(中略) ロボット制御の基幹部分であるロボット OS・ミドルウェアについても積極的な取組を進めていくことが重要である。(中略) 様々な国際標準を日本主導で策定することで、(中略) 日本のロボット産業が世界をリードすることが可能となる。

(中略) そのため、新たに設置される「ロボット革命イニシアティブ協議会」(第1項参照) が中心となって、必要な情報収集及び国際発信を行い、国際標準化の流れを主導していく。

ロボットミドルウェアとは？

ロボットミドルウェアとは？

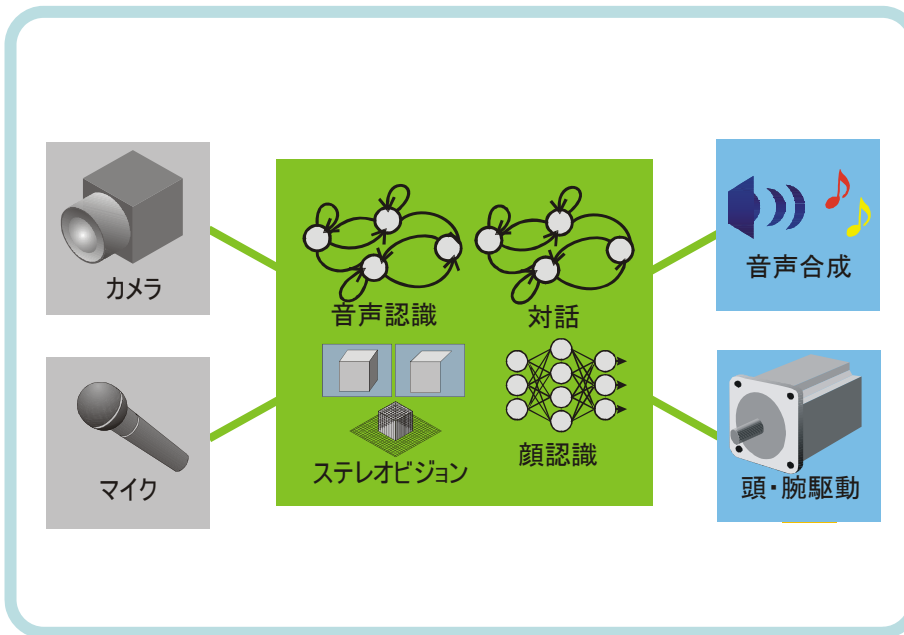
- ロボットシステム構築を効率化するための共通機能を提供する**基盤ソフトウェア**
 - 「ロボットOS」と呼ばれることもある
 - インターフェース・プロトコルの共通化、標準化
 - 例として
 - モジュール化・コンポーネント化フレームワークを提供
 - モジュール間の通信をサポート
 - パラメータの設定、配置、起動、モジュールの複合化（結合）機能を提供
 - 抽象化により、OSや言語間連携・相互運用を実現
- 2000年ごろから開発が活発化
 - 世界各国で様々なミドルウェアが開発・公開されている

ロボットソフトウェア開発の方向

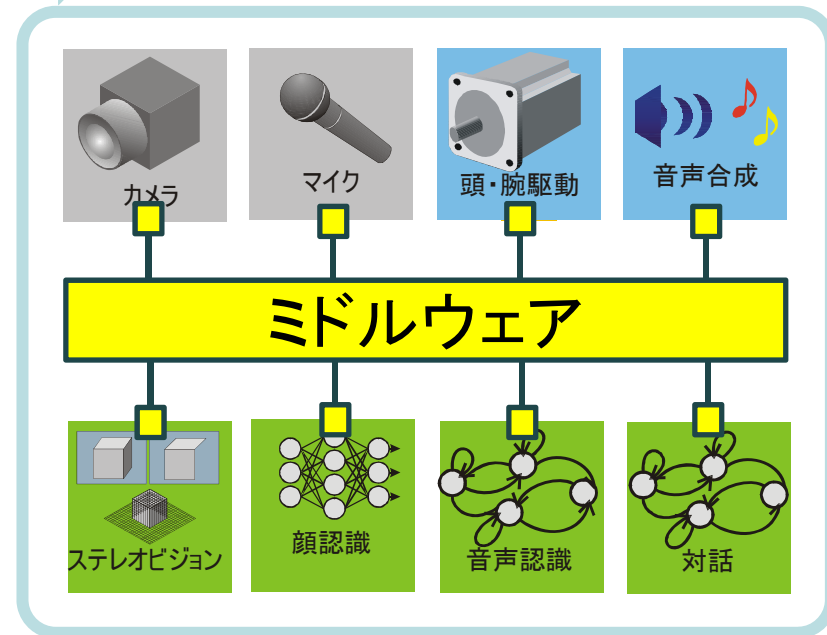
従来型開発



コンポーネント指向開発



- ✓ 様々な機能を融合的に設計
- ✓ 実行時の効率が高いが、柔軟性に欠ける
- ✓ システムが複雑化してくると開発が困難に



- ✓ 大規模複雑な機能の分割・統合
- ✓ 開発・保守効率化(機能の再利用等)
- ✓ システムの柔軟性向上

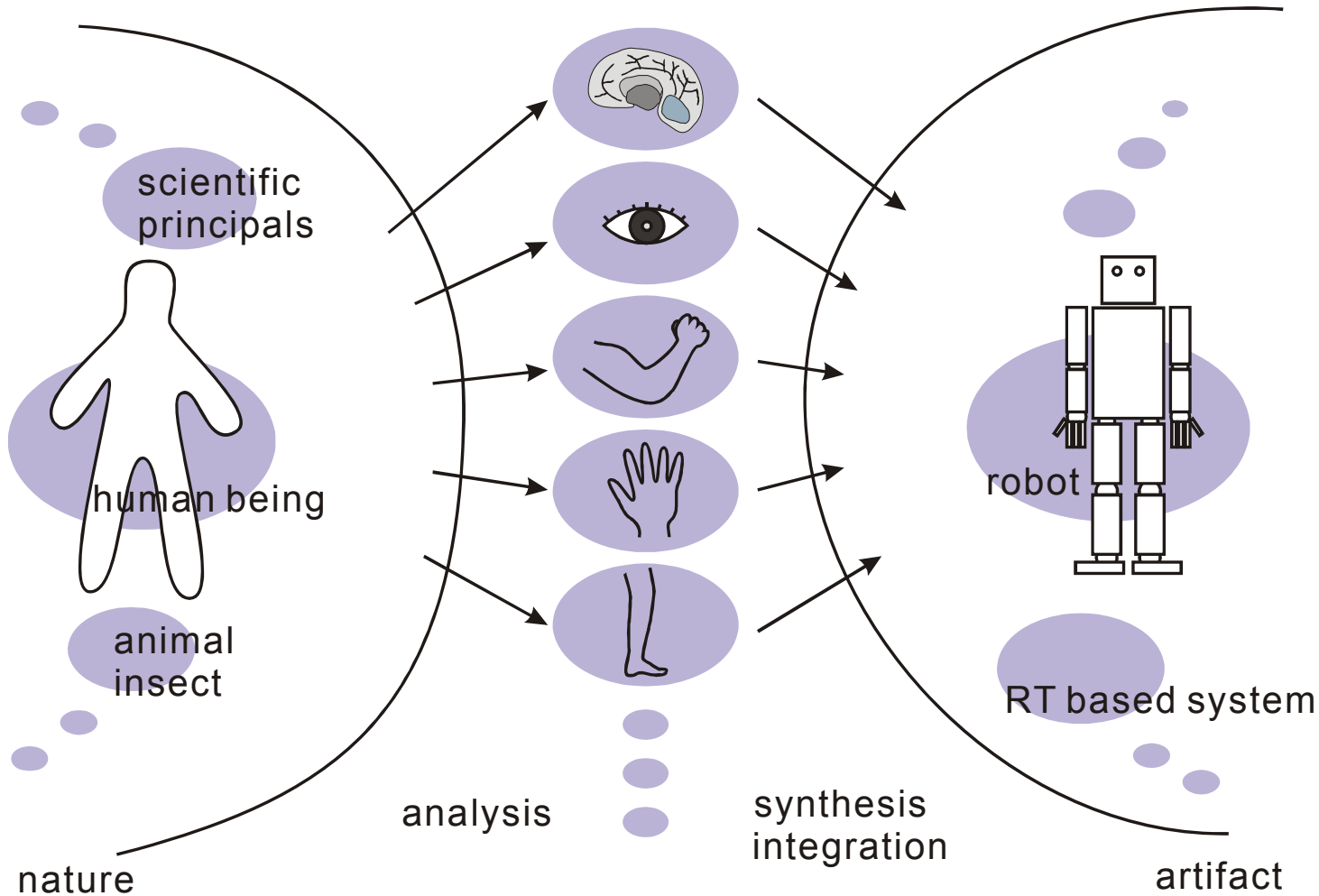
モジュール化のメリット

- 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か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

ミドルウェア、コンポーネント、etc...

- ミドルウェア
 - OSとアプリケーション層の間に位置し、特定の用途に対して利便性、抽象化向上のために種々の機能を提供するソフトウェア
 - 例: RDBMS、ORB等。定義は結構曖昧
- 分散オブジェクト(ミドルウェア)
 - 分散環境において、リモートのオブジェクトに対して透過的アクセスを提供する仕組み
 - 例: DDS、CORBA、Ice、Java RMI、DCOM等
- コンポーネント
 - 再利用可能なソフトウェアの断片(例えばモジュール)であり、内部の詳細機能にアクセスするための(シンタクス・セマンティクスともにきちんと定義された)インターフェースセットをもち、外部に対してはそのインターフェースを介してある種の機能を提供するモジュール。
- CBSD(Component Based Software Development)
 - ソフトウェア・システムを構築する際の基本構成要素をコンポーネントとして構成するソフトウェア開発手法

アナリシスからシンセシスへ



様々なミドルウェア/プラットフォーム

- OpenRTM-aist
- ROS
- OROCOS
- OPRoS
- ORCA
- Microsoft Robotic Studio
- Player/Stage/Gazebo
- ORiN
- RSNP
- UNR Platform
- OPEN-R
- Open Robot Controller architecture

RT-Middleware OpenRTM-aist



RTとは？

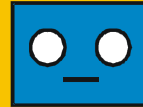
- RT = Robot Technology cf. IT
 - ≠Real-time
 - 単体のロボットだけでなく、さまざまなロボット技術に基づく機能要素をも含む (センサ、アクチュエータ, 制御スキーム、アルゴリズム、etc....)

産総研版RTミドルウェア

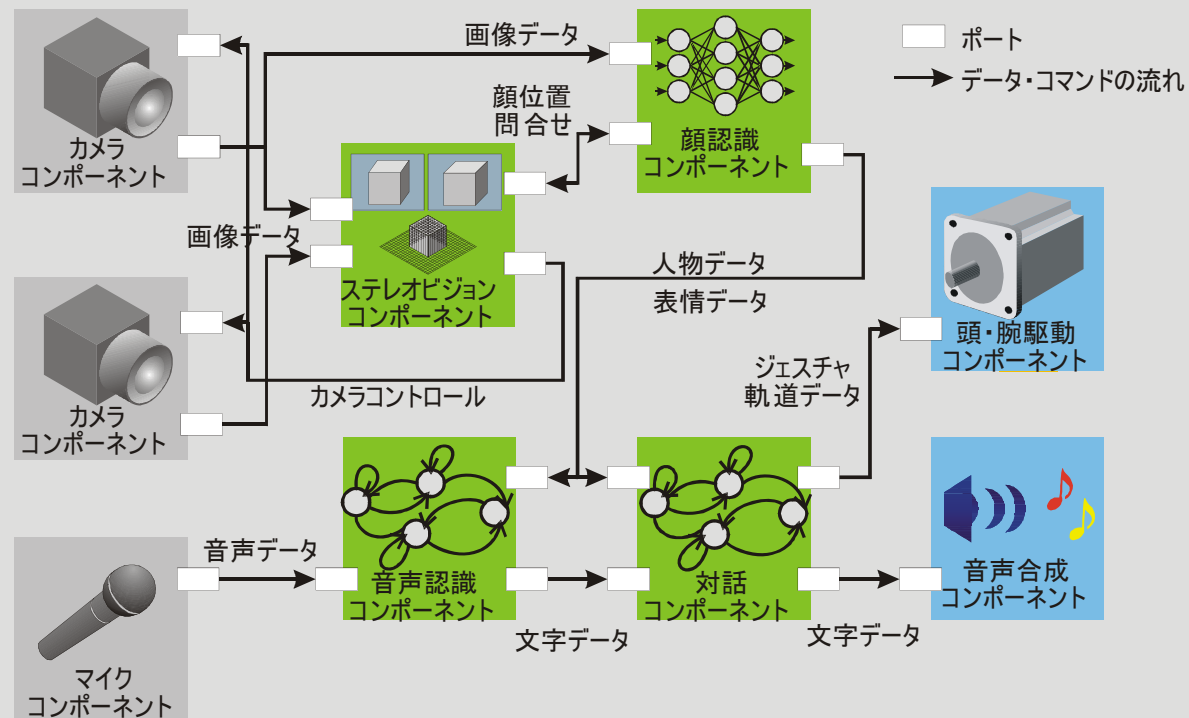
OpenRTM-aist

- 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

ロボットシステムの構造



ロボット体内のコンポーネントによる構成例



多くの機能要素(コンポーネント)から構成される

RTミドルウェアプロジェクト

NEDO 21世紀ロボットチャレンジプログラム (2002-2004年度)

「ロボット機能発現のために必要な要素技術開発」

- RT分野のアプリケーション全体に広く共通的に使われる機能およびRT要素の部品化(モジュール化)の研究開発
- 分散オブジェクト指向システムのミドルウェアであるCORBAをベースとして行う。
- RT要素の分類を行い、モジュール化の形態、必要な機能、課題、インタフェース仕様などを明確にする。

14年度成果報告書より

従来のシステムでは...



Joystick



Joystick
software



Robot Arm
Control software



Robot Arm

互換性のあるインターフェース同士は接続可能

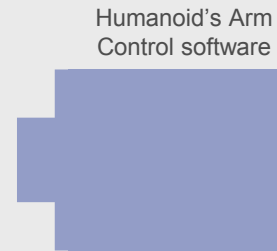
従来のシステムでは...



Joystick



Joystick
software



Humanoid's Arm
Control software



Humanoid's Arm



Robot Arm

Robot Arm
Control software

ロボットによって、インターフェースは色々
互換性が無ければつながらない

RTミドルウェアでは...

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



compatible
arm interfaces



Arm B
Control software



Humanoid's Arm

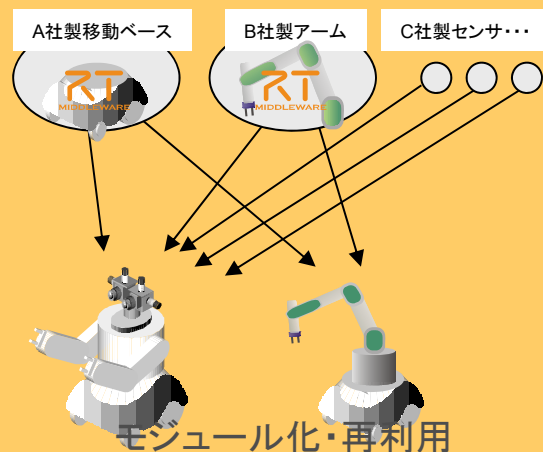


Robot Arm

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



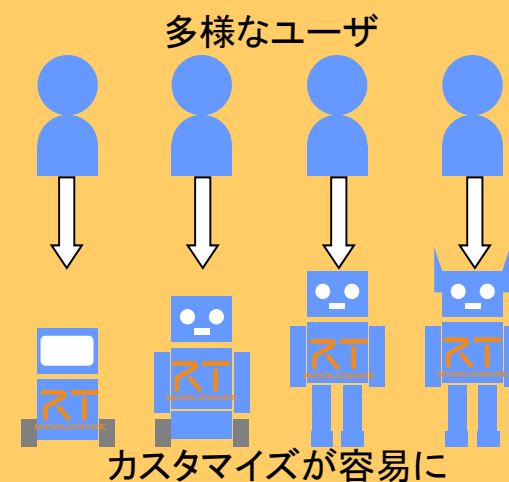
ロボットの低コスト化

技術の問題



最新技術を利用可能

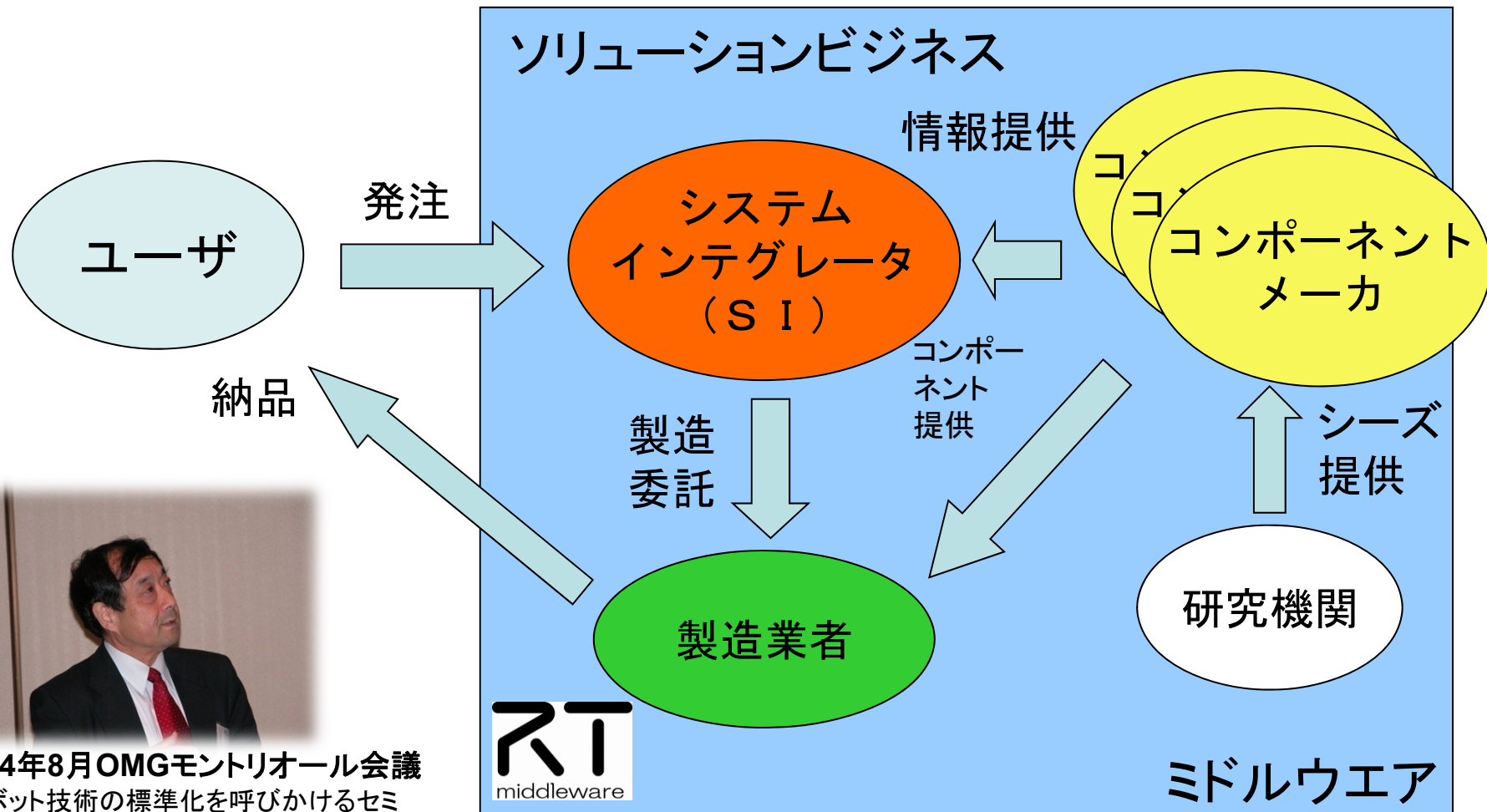
ニーズの問題



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

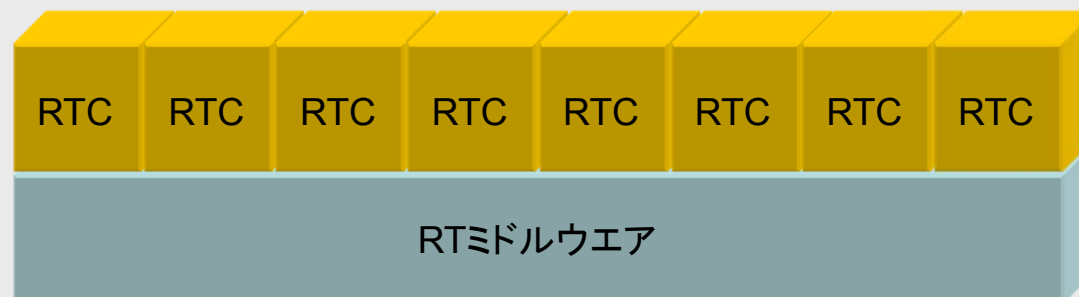
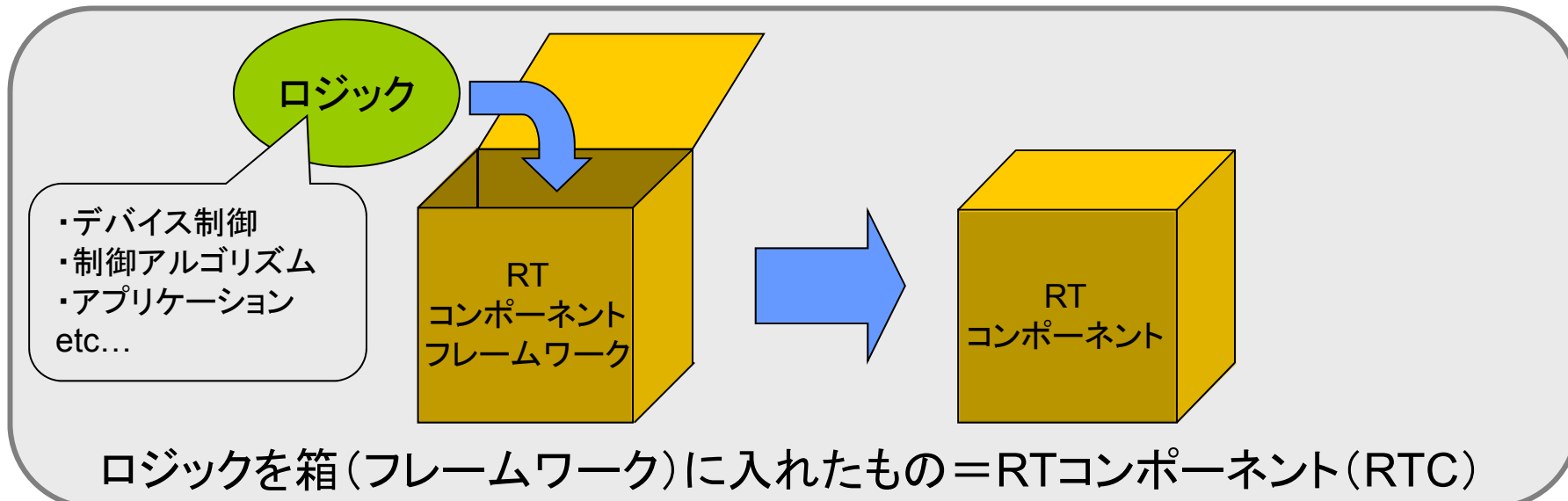
期待される未来のロボット産業のモデル



2004年8月OMGモントリオール会議
(ロボット技術の標準化を呼びかけるセミナーを企画し、基調講演する故谷江和雄 元
産総研知能システム研究部門長、
President of IEEE Robotics and Society)

ミドルウェア：SI、コンポーネントメーカー、製造業者間の
交流を促進する情報基盤

RTミドルウェアとRTコンポーネント



RTCの実行環境(OSのようなもの)=RTミドルウェア(RTM)
※RTCはネットワーク上に分散可能

RTコンポーネント化のメリット

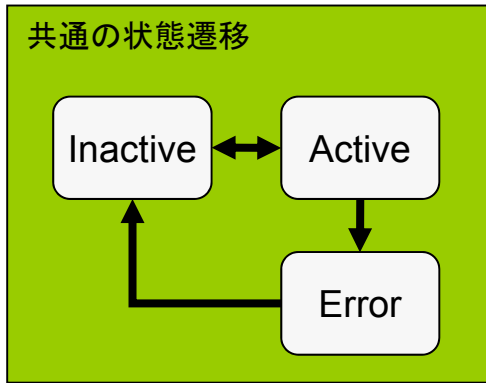
モジュール化のメリットに加えて

- 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

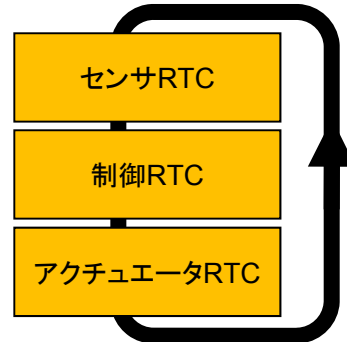
RTコンポーネントの主な機能

アクティビティ・実行コンテキスト

共通の状態遷移



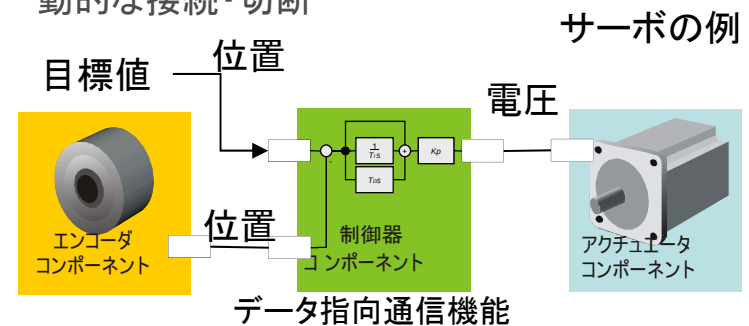
複合実行



ライフサイクルの管理・コアロジックの実行

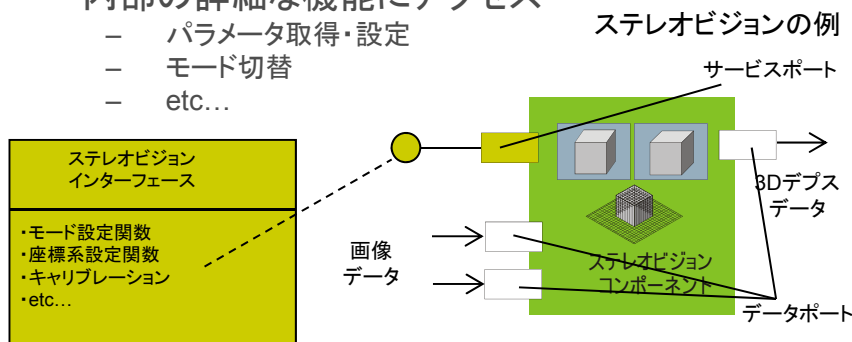
データポート

- 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



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...

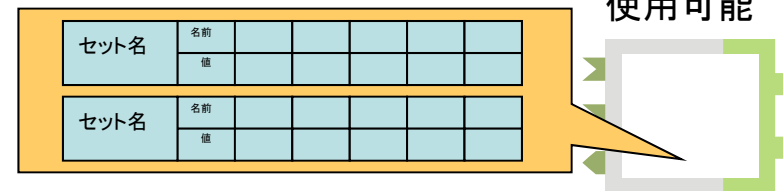


サービス指向相互作用機能

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

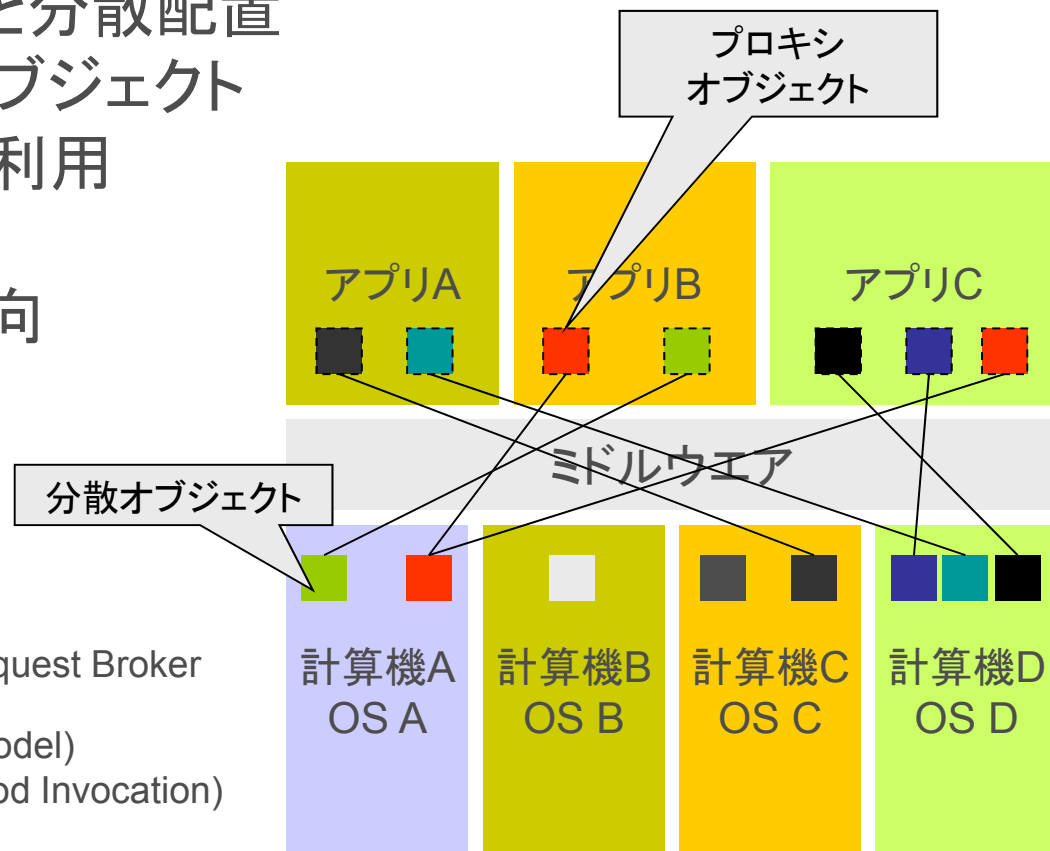


分散オブジェクトとは？

- システムの機能分割と分散配置
- ネットワーク透過なオブジェクト
- コンポーネント化と再利用

↓
オブジェクト指向
＋
ネットワーク

- 代表例
 - CORBA (Common Object Request Broker Architecture)
 - CCM (CORBA Component Model)
 - JavaRMI (Java Remote Method Invocation)
 - EJB (Enterprise Java Beans)
 - DCOM, HORB etc...



CORBAの例

本題にたどり着くまでが面倒

```
class MobileRobot_Impl  
: public virtual POA_MobileRobot,  
  public virtual  
    PortableServer::RefCountServant
```

```
robot->gotoPos(pos);
```

サーバスケルトン

クライアントスタブ

RTミドルウェアが
全部面倒みます！！

呼び出し

RTM、RTCとは？

ソフトウェアアーキテクチャの違い



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

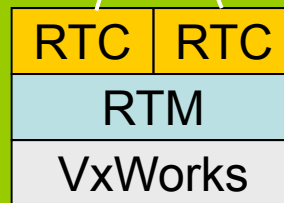
分散オブジェクトからRTCへ

- 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

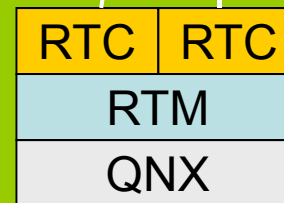
RTミドルウェアによる分散システム

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。

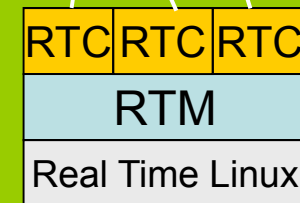
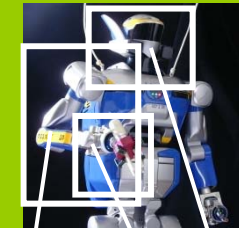
ロボットA



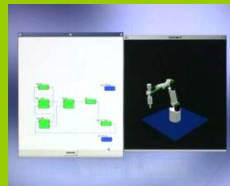
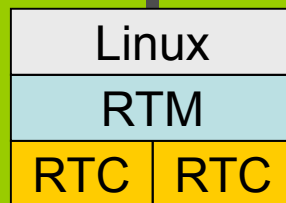
ロボットB



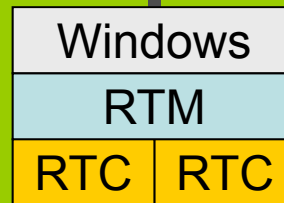
ロボットC



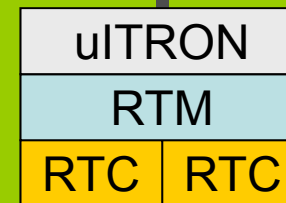
ネットワーク



アプリケーション



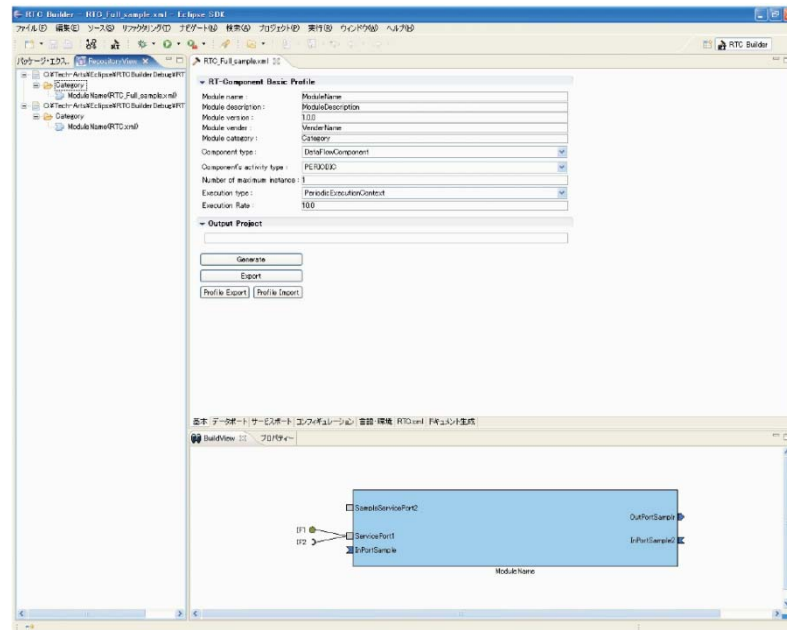
操作デバイス



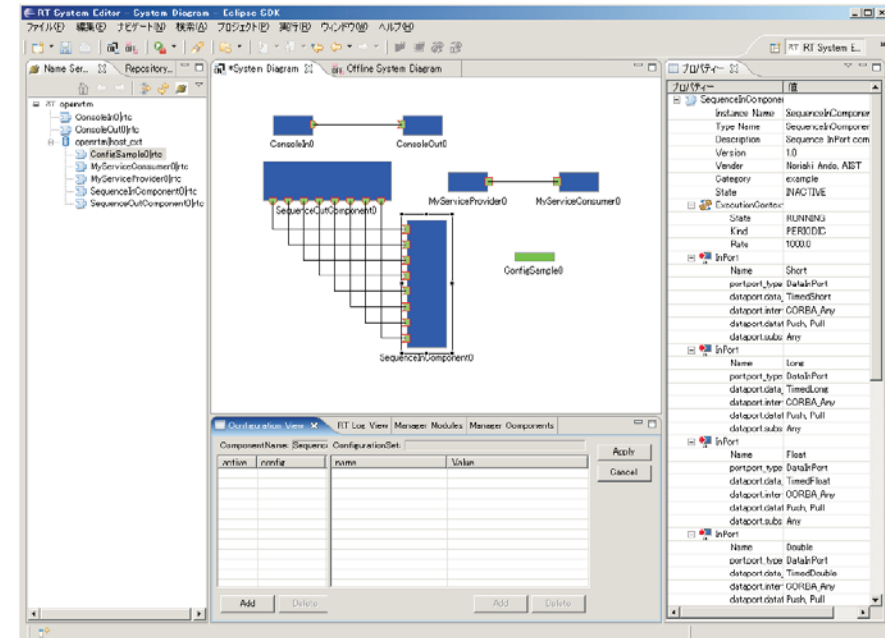
センサ

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。

ツールチェーン



RTCBuilder
RTコンポーネント設計・コード生成



RTSysytemEditor
RTCを組み合わせてシステムを設計

RTC・RTM統合開発環境の整備

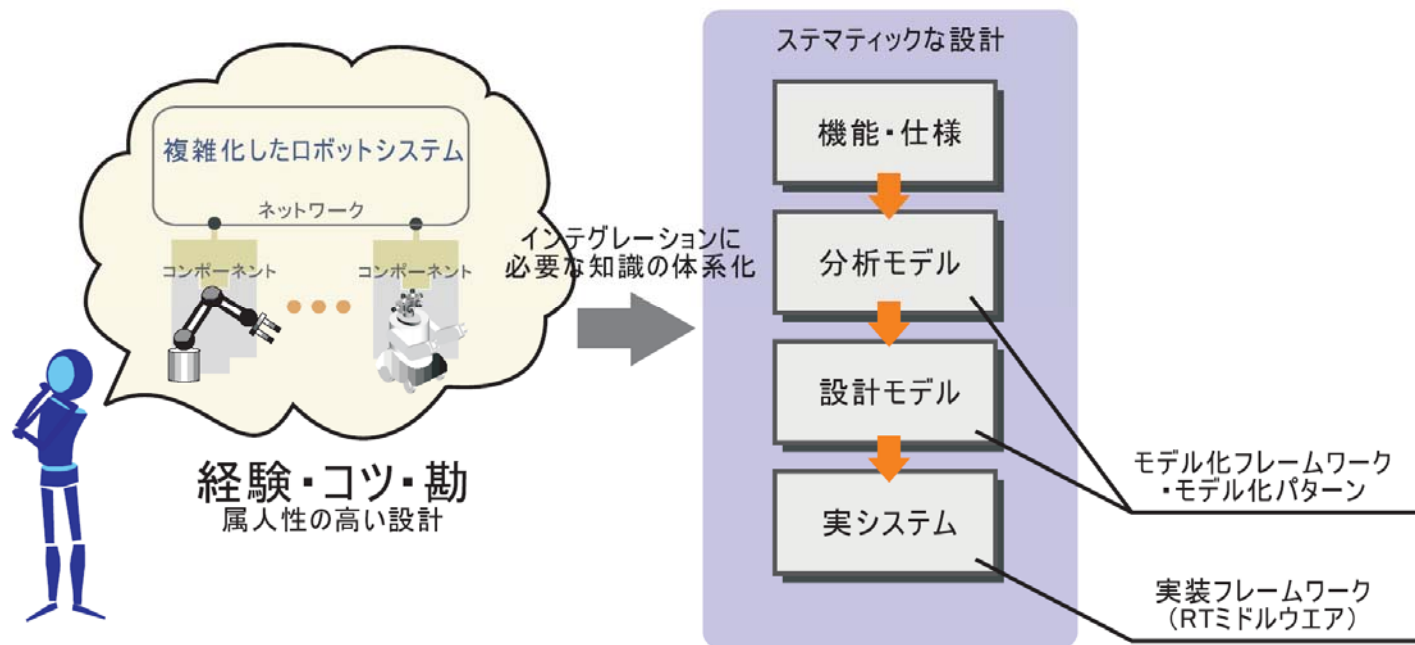
RTC設計・実装・デバッグ、RTMによるインテグレーション・デバッグまでを一貫して行うことができる統合開発環境をEclipse上に構築

体系的システム開発

開発者の経験やノウハウに依存したロボットシステム開発

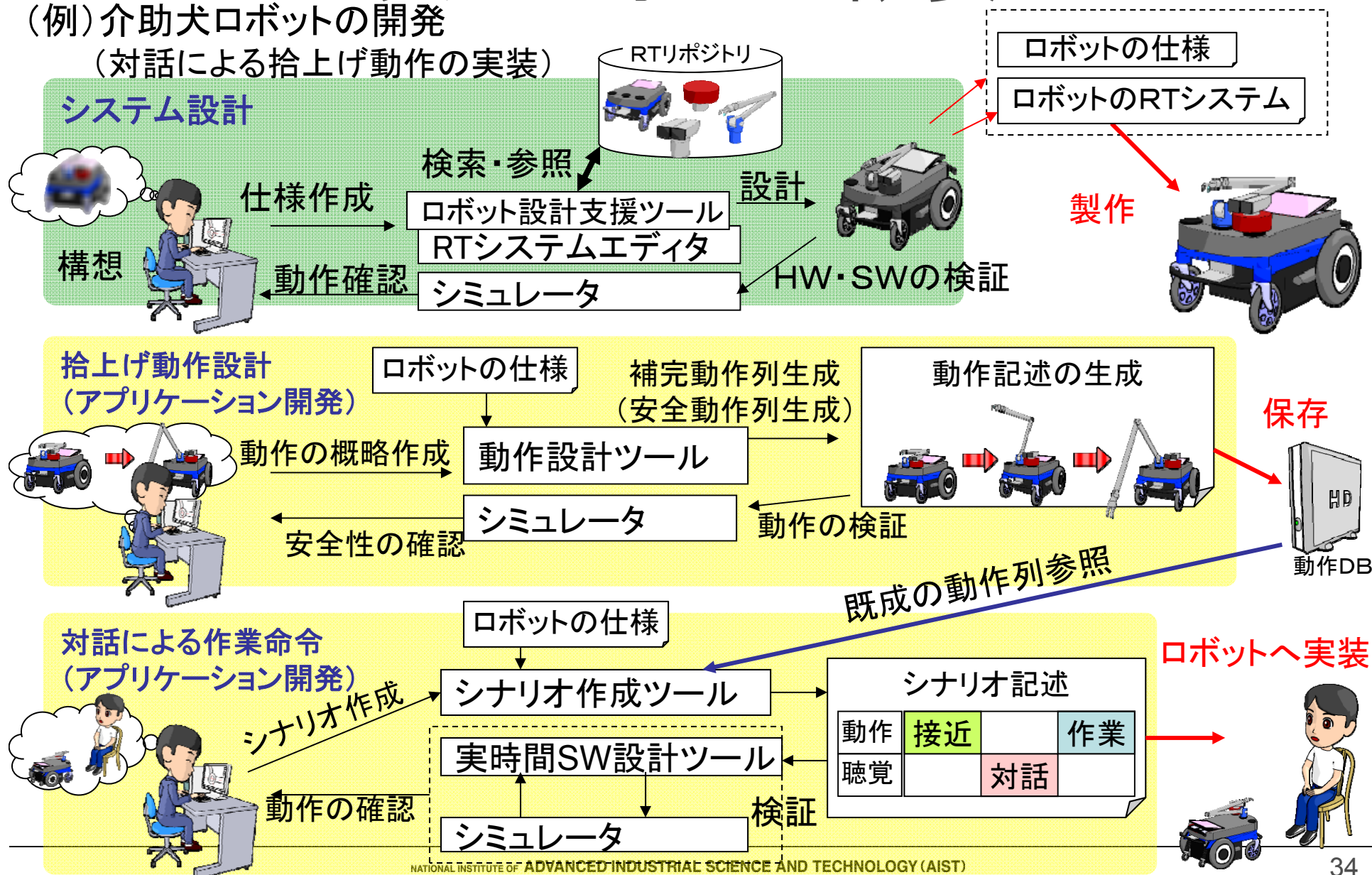


体系的システムデザイン: 分析、設計、実装の一連の流れ



プラットフォーム概要

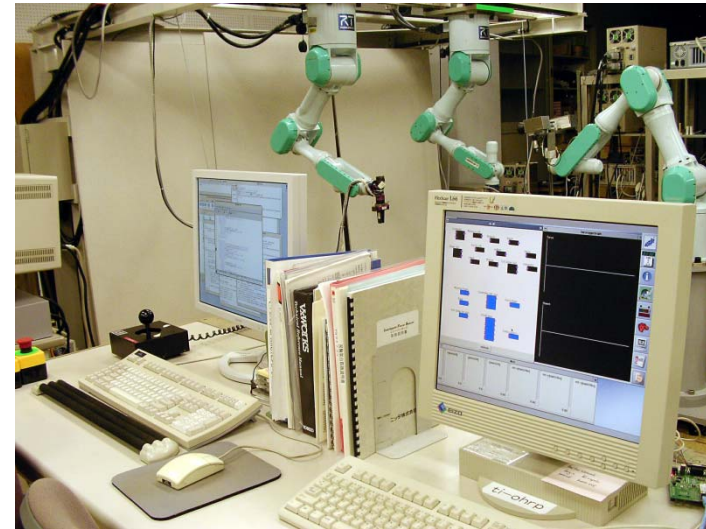
(例) 介助犬ロボットの開発
(対話による拾上げ動作の実装)



RTミドルウェアPJ

FY2002.12-FY2004

- 名称: NEDO 21世紀ロボットチャレンジプログラム
 - 「ロボット機能発現のために必要な要素技術開発」
- 目的:
 - RT要素の部品化(モジュール化)の研究開発
 - 分散オブジェクト指向開発
 - RT要素の分類・モジュール化に必要な機能・インタフェース仕様の明確化
- 予算規模:
 - 65百万円
 - 全体267.3百万円



RTミドルウェアの国際標準化

Date: September 2012



Robotic Technology Component (RTC)

Version 1.1

Normative reference: <http://www.omg.org/spec/RTC/1.1>
 Machine consumable files: <http://www.omg.org/spec/RTC/20111205/>
 Normative:
<http://www.omg.org/spec/RTC/20111205/rtc.xml>
<http://www.omg.org/spec/RTC/20111205/rtc.h>
<http://www.omg.org/spec/RTC/20111205/rtc.idl>
 Non-normative:
<http://www.omg.org/spec/RTC/20111205/rtc.eap>

標準化履歴

- 2005年9月
Request for Proposal 発行(標準化開始)
- 2006年9月
OMGで承認、事実上の国際標準獲得
- 2008年4月
OMG RTC標準仕様 ver.1.0公式リリース
- 2012年9月
ver. 1.1改定
- 2015年9月
FSM4RTC(FSM型RTCとデータポート標準) 採択

OMG国際標準

- 標準化組織で手続きに沿って策定
- 1組織では勝手に改変できない安心感
 - 多くの互換実装ができてつある
 - 競争と相互運用性が促進される

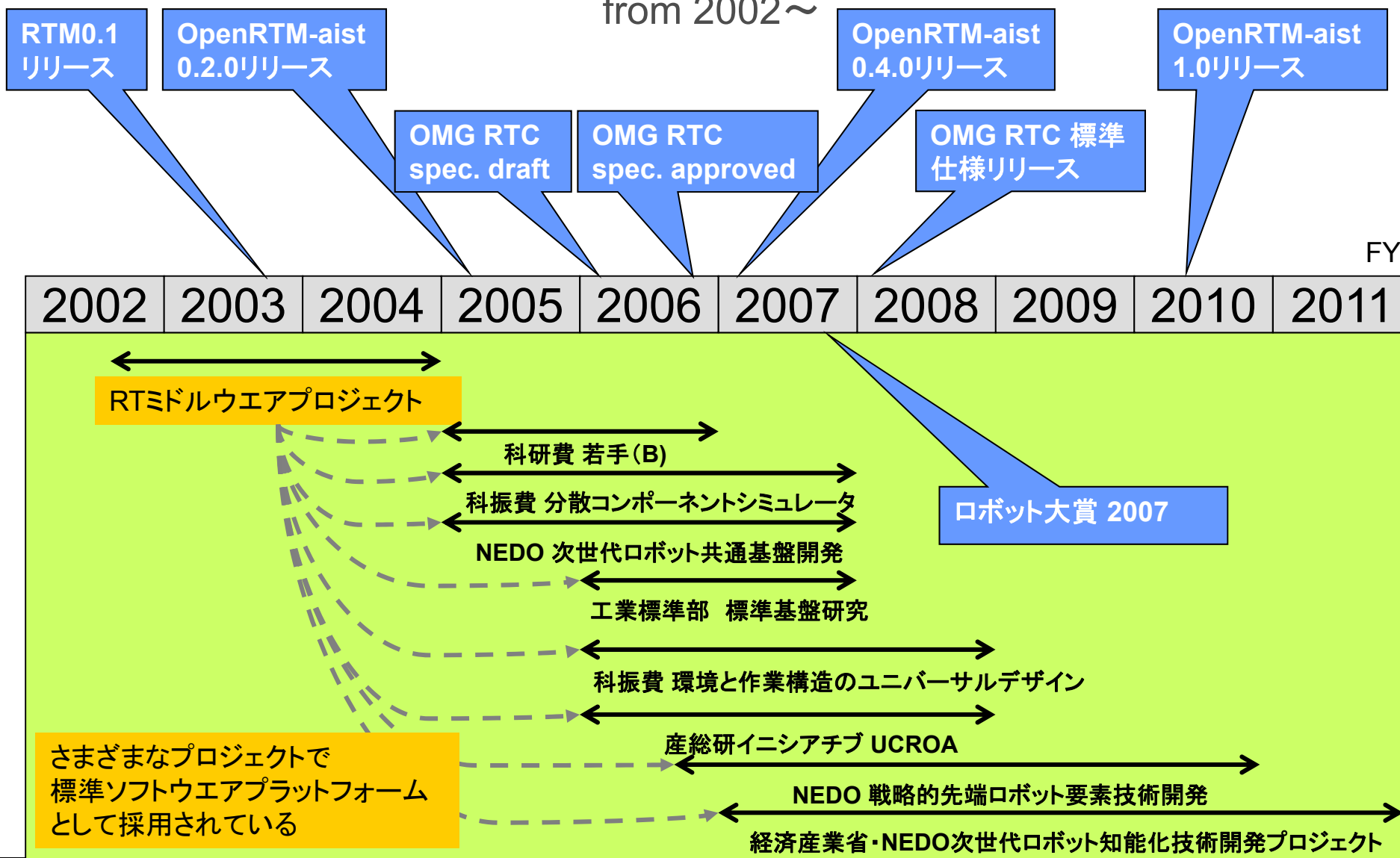
RTミドルウェア互換実装は10種類以上

名称	ベンダ	特徴	互換性
OpenRTM-aist	産総研	NEDO PJで開発。参照実装。	---
HRTM	ホンダ	アシモはHRTMへ移行中	◎
OpenRTM.NET	セック	.NET(C#,VB,C++/CLI, F#, etc..)	◎
RTM on Android	セック	Android版RTミドルウェア	◎
RTC-Lite	産総研	PIC, dsPIC上の実装	○
Mini/MicorRTC	SEC	NEDOオープンイノベーションPJで開発	○
RTMSafety	SEC/AIST	NEDO知能化PJで開発・機能安全認証取得	○
RTC CANOpen	SIT, CiA	CAN業界RTM標準	○
PALRO	富士ソフト	小型ヒューマノイドのためのC++ PSM 実装	×
OPRoS	ETRI	韓国国家プロジェクトでの実装	×
GostaiRTC	GOSTAI, THALES	ロボット言語上で動作するC++ PSM実装	×

特定のベンダが撤退しても
ユーザは使い続けることが可能

RT-Middleware関連プロジェクト

from 2002~



実用例・製品化例



HRPシリーズ: 川田工業、AIST

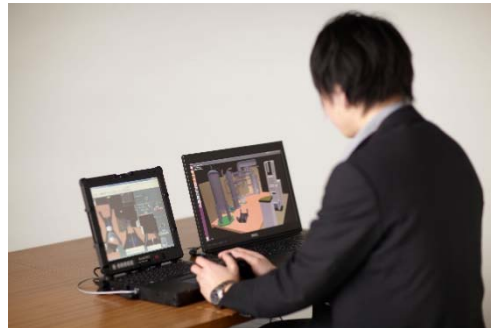


S-ONE: SCHAFT



DAQ-Middleware: KEK/J-PARC

KEK: High Energy Accelerator Research Organization
J-PARC: Japan Proton Accelerator Research Complex



災害対応ロボット操縦シミュレータ:
NEDO/千葉工大



HIRO, NEXTAGE open: Kawada Robotics



RAPUDA: Life Robotics



ビュートローバーRTC/RTC-BT(VSTONE)



OROCHI(アールティ)



新日本電工他: Mobile SEM

ROS(Robot Operating System)



<http://wiki.ros.org/>



Willow Garage

- Willow Garage
 - 2007年設立のロボットベンチャー(米、Menlo Park)
 - Scott Hassanが出資
 - googleの初期エンジンの作者の一人
 - Brian Gerkeyがソフトウェア部分のPL
 - Playerの作者の一人
 - ビジネスモデル
 - ソフト:ROS(無償)+ハード:PR2 を販売
 - その他は不明、資金がたくさんあるので急いで利益を上げる必要はないとのこと
 - まずPR2を10台無償で大学などに配布予定

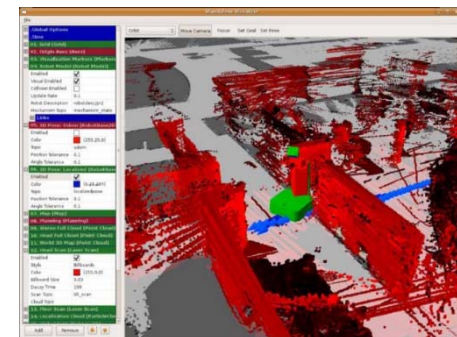
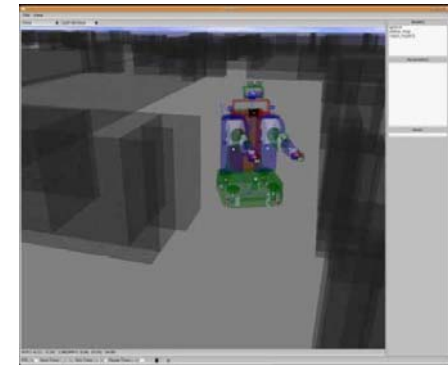


ROS (Robot Operating System)

- 目的: ロボット研究を効率的にする
- OSではない
 - Robot Operating System, Robot Open Source
- UNIX哲学
 - UNIX的なコンパクトなツール、ライブラリ群による効率的なインテグレーション環境を提供する。
 - モジュール性、柔軟性、再利用性、言語・OS非依存、オープンソース
- プラットフォームアーキテクチャ
 - 独自ミドルウェア(独自IDLからコード生成)
 - メッセージベース
 - pub/subモデル、RPCモデル

ROS(2)

- ライブラリ・ツール群が充実
 - シミュレータ: Gazeboベース
 - ナビゲーション: Player等
 - 座標変換・キャリブレーション
 - プランニング: TREX
 - マニピュレーション: OpenRAVE
 - ビジョン: OpenCV
 - 音声: ManyEars
- 外部のオープンソースライブラリを取り込みライブラリ群として整備
- 最近、これまでとは大きくアーキテクチャの異なるミドルウェア“ROS2”をリリース。



Player/Stage and Gazebo

<http://playerstage.sourceforge.net/>



Player/Stage and Gazebo

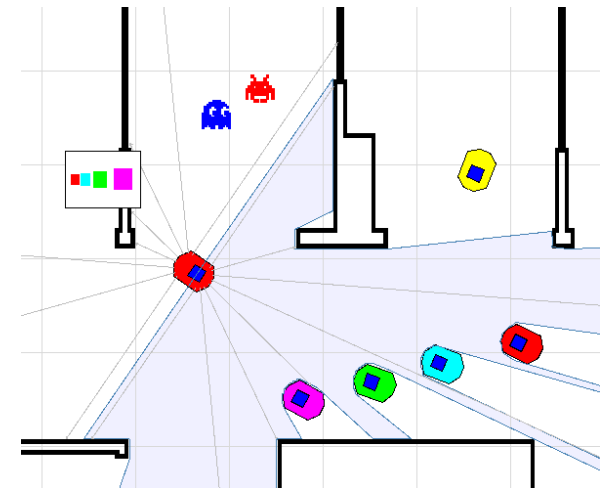
- USC Robotics Lab
- Player
 - ロボット用ネットワークサーバ
 - ネットワーク経由でセンサ・アクチュエータにアクセスするためのインターフェースを提供
 - Playerは多くのロボットハードウェアをサポート
 - ex. ActivMedia Pioneer 2 family



“Player” は移動ロボット制御のための一連の共通インターフェースを提供するプラットフォーム

Player/Stage and Gazebo

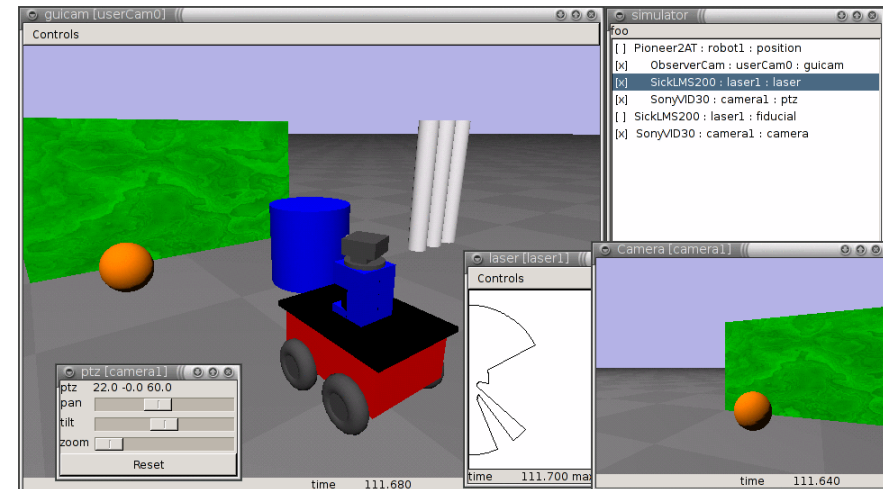
- Stage
 - 2次元のビットマップベースの環境において、移動ロボット、センサ、オブジェクト群をシミュレートする.
 - マルチエージェントシステムの研究のために設計されている.



“Stage” は2次元の移動ロボット
シミュレーション環境を提供する.

Player/Stage and Gazebo

- Gazebo
 - 屋外環境のマルチロボットシミュレーション環境を提供.
 - 3次元環境のロボット、センサ、オブジェクト群のシミュレーション環境を提供する.
 - 現在はDARPAの支援を受けてOSRF[†]において開発を継続中



[†]OSRF: OpenSource Robotics Foundation

3次元移動ロボットシミュレーション環境を提供.

Player/Stage and Gazebo

- パスプランニング
- 衝突回避
- センサフュージョン
- マルチエージェント制御
- etc...



ユーザが開発するPlayerクライアント.

C, C++, Tcl, LISP, Java, Python
をサポート.
しかし、OOPではない.

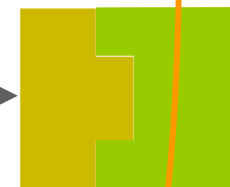
Robot server



Robot A



Robot B



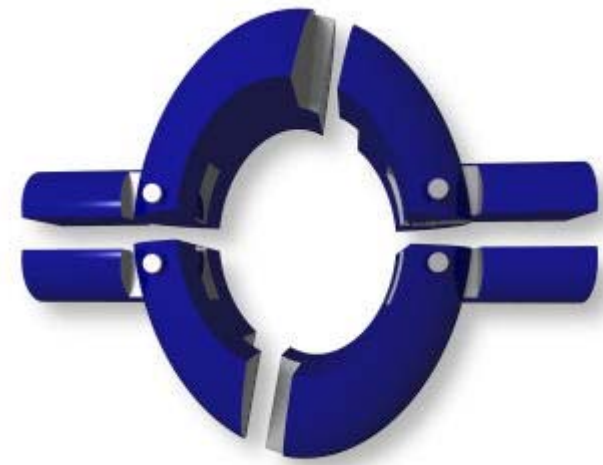
Robot C



Simulator

ロボット・センサに対する
共通インターフェースを提供.

OROCOS



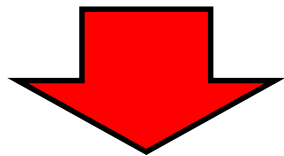
<http://www.oroocos.org/>

OROCOS

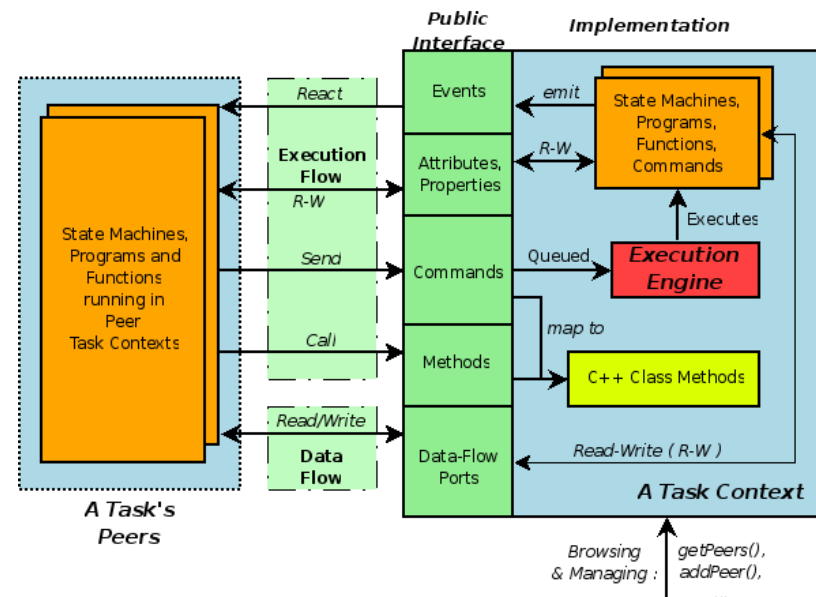
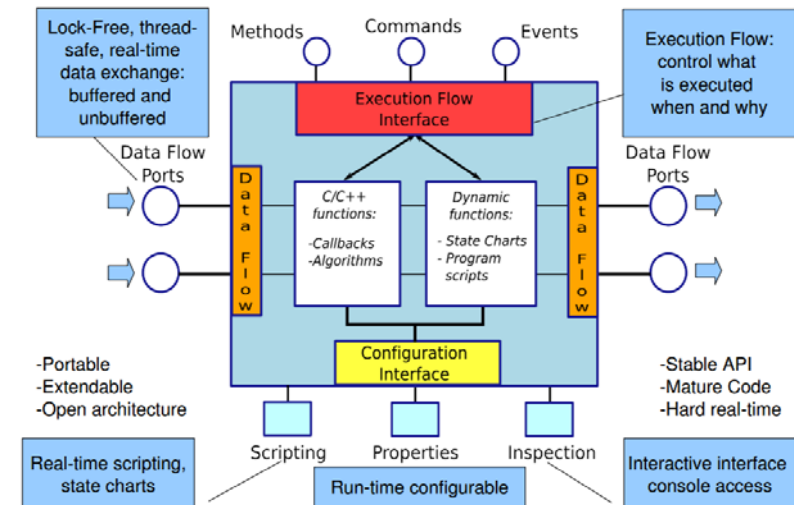
- EUプロジェクトで開発(2001-2003)
 - K.U.Leuven(ベルギー)
 - LAAS Toulouse(フランス)→ORCAへ
 - KTH Stockholm(スウェーデン)
- ハードリアルタイムのソフトウェアフレームワーク
- ロボットの制御に必要なライブラリ集(運動学、リアルタイム制御、etc...)
- 最近はコンポーネントベース開発のフレームワークも提供
- ツールによるモデルベース開発

OROCOS

- コンポーネントモデル
 - データフローポート
 - 種々のサービスインターフェース
 - コンフィギュレーション機能
 - コールバックベースのロジック実行フレームワーク



ほぼRTCと同じ(マネ?)



OPRoS

(The Open Platform for Robotic Services)

<http://www.opros.or.kr/>

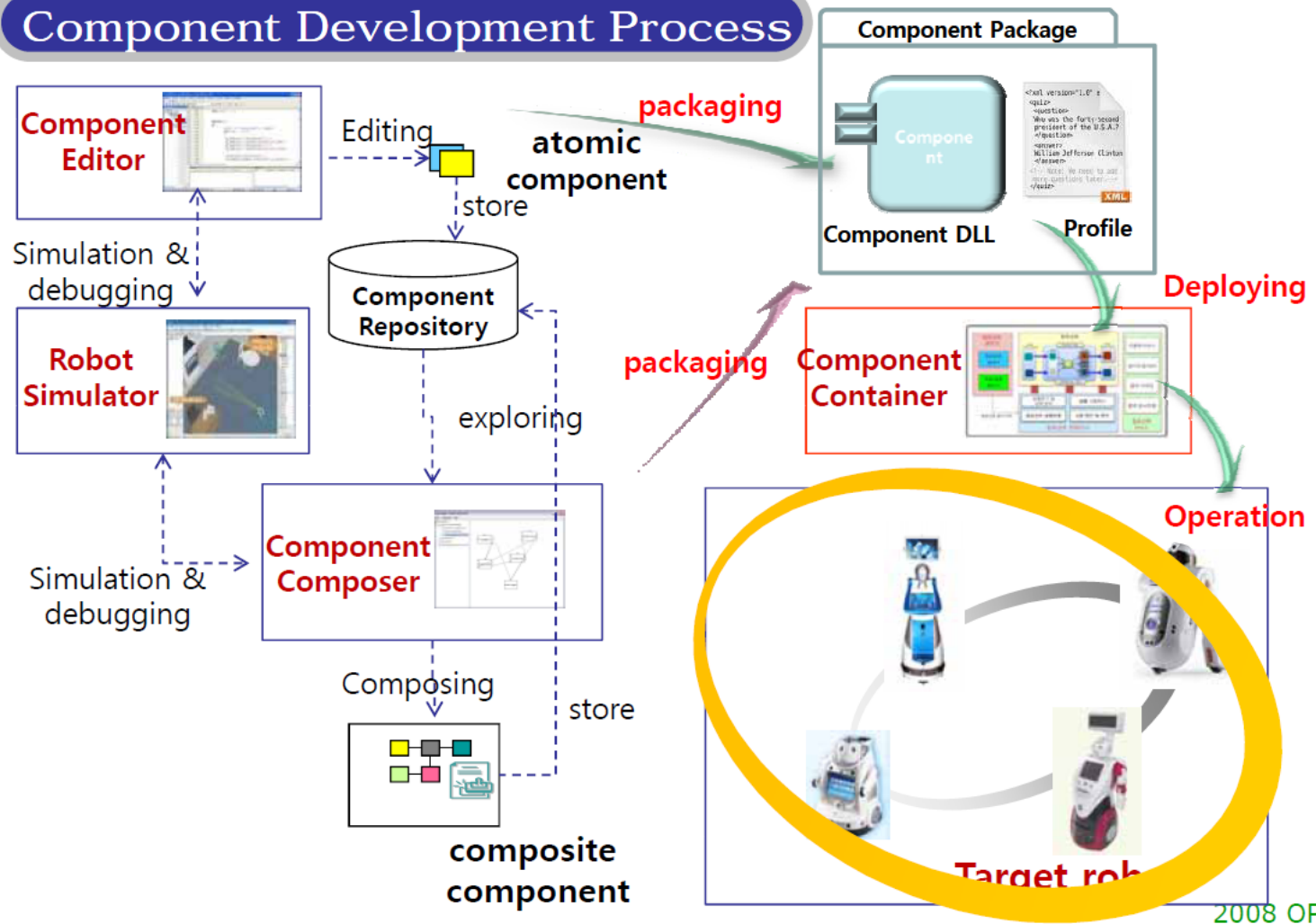


OPRoS

- 韓国の国プロジェクトで開発されたロボット用プラットフォーム
 - ETRI(Electronics and Telecommunication Research Institute)
 - KIST, Kanwong Univ., etc...
- OMG RTC(ほぼ)準拠
- 通信ミドルウェアは独自(URC_{PJ}(Ubiquitous Robot Companion)で開発したもの)
- ツールチェーンなども提供

OPRoS

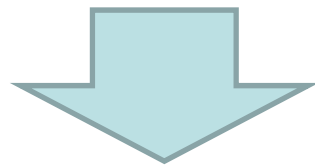
Component Development Process



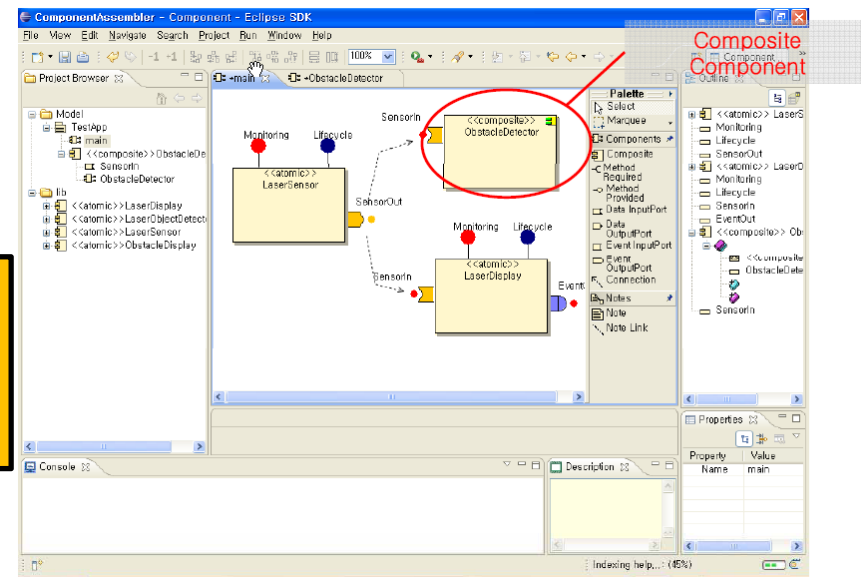
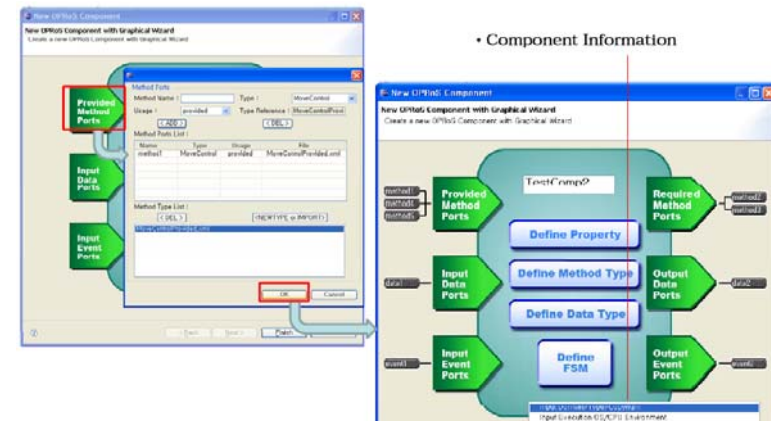
© ETRI, OMG Infra. WG, OPRoS Component Tools

OPRoS

- コンポーネント開発
 - Component Editor
- システム構築
 - Component Composer



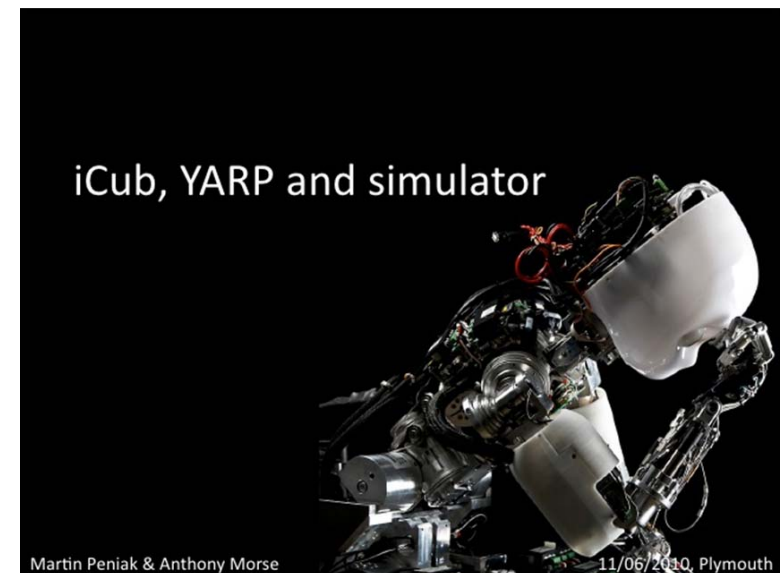
OpenRTM-aistに
よく似たツールチェーン



© ETRI, OMG Infra. WG, OPRoS Component Tools

YARP

<http://eris.liralab.it/yarp/>



YARP

(Yet Another Robot Platform)

- IIT (Istituto Italiano di Tecnologia) で開発された iCub のためのソフトウェアプラットフォーム
 - iCub: EU プロジェクト RobotCub,
 - 53DOF の赤ちゃんの様なヒューマノイド
- YARP



YARP

- コンポーネントフレームワークは無し
 - Mainから書き始める
 - 原則1プロセス1モジュール
- 多様な伝送方式のデータポートを提供
 - TCP, UDP, multicast
 - Carrier: 様々なマーシャリング、プロトコルを利用可能
- 簡単なRPCもある
- 独自のマーシャリング方式
- ノード間の利用にはネームサービスを利用
- CUIツール: yarp
 - 接続制御、モジュール制御

Microsoft Robotics Studio

<http://www.microsoft.com/robotics/>

The logo for Microsoft Robotics Developer Studio 4, featuring the text "Microsoft Robotics Developer Studio 4" in white on an orange rectangular background.

Microsoft Robotics Studio

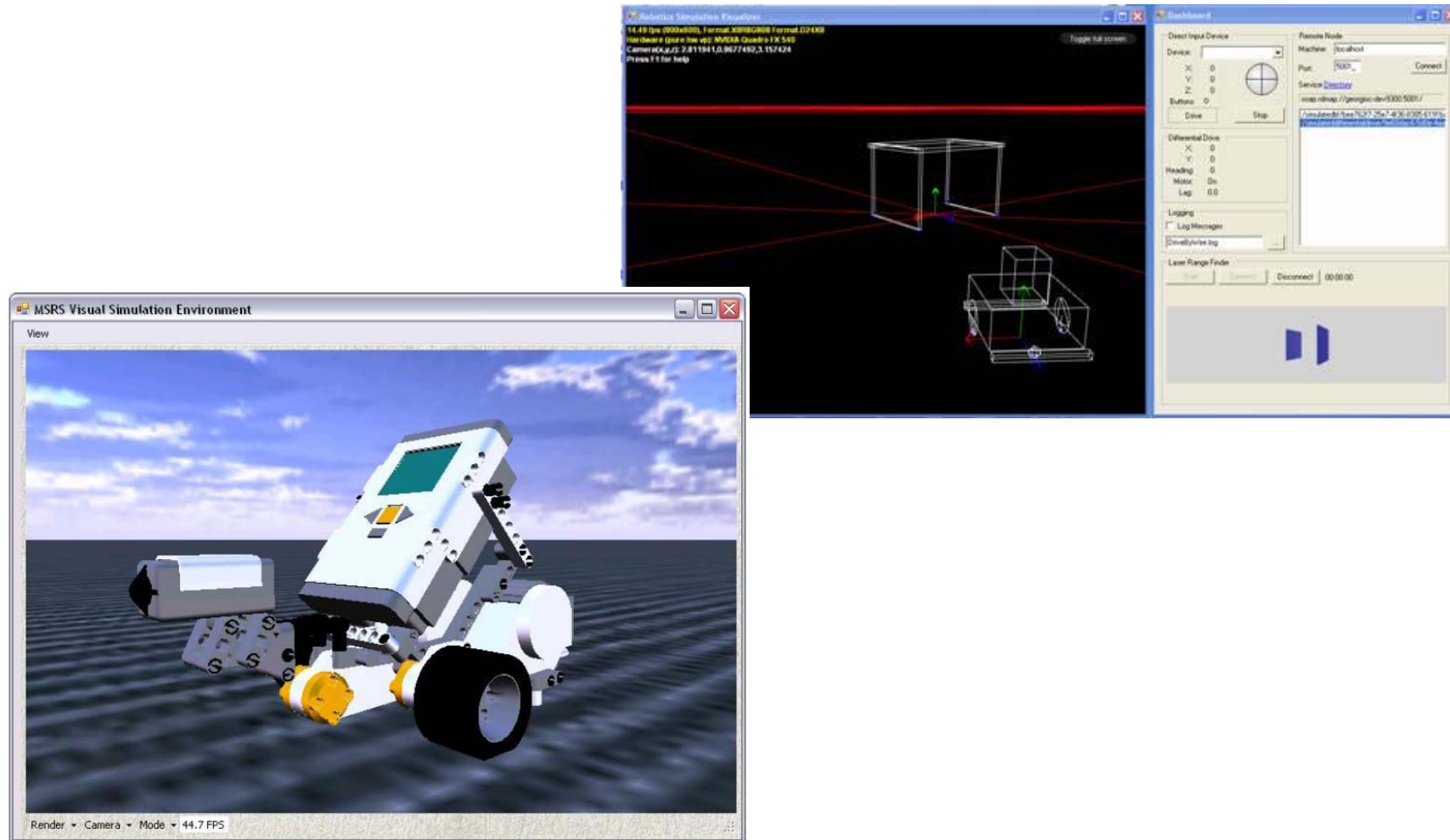
- 高レベルのビヘイビアやハードウェアのサービス志向の抽象化を提供
- 再利用性
- シミュレータエンジンを提供
 - AGEIA PhysXも使える
- ブラウザインターフェース
- Windowsのみで利用可能
 - ただし、ある種のハードウェアに対応
 - LEGO NXT, i-Robot, ActivMedia等

MSRSアーキテクチャ

DSSP: Decentralized Software Services Protocol

- DSSP: SOAPベースのプロトコルであり軽量なサービスモデルを提供する。
- DSSPはステート志向のメッセージオペレーションをサポート。
- アプリケーションはサービスの集合体として定義され、HTTPの拡張として提供される。
- サービスとは: アプリケーションのライフタイム中であれば、生成、操作、モニタリング、削除可能な軽量なエンティティ。
- サービス: ユニークなID、状態、振る舞い、コンテキストを持つ実体

MSRS シミュレータ



OPEN-R



OPEN-R



- 開発元: SONY
- AIBO, SDR-[3,4]X, QURIO等の制御アーキテクチャ
- 目的
 - スケーラビリティ
 - ポータビリティ
 - インターオペラビリティ
 - 異なるタイプのロボットに対して同じインターフェースを提供
 - スタイル・フレキシビリティ
 - 車輪型、4足歩行型、2足歩行型...



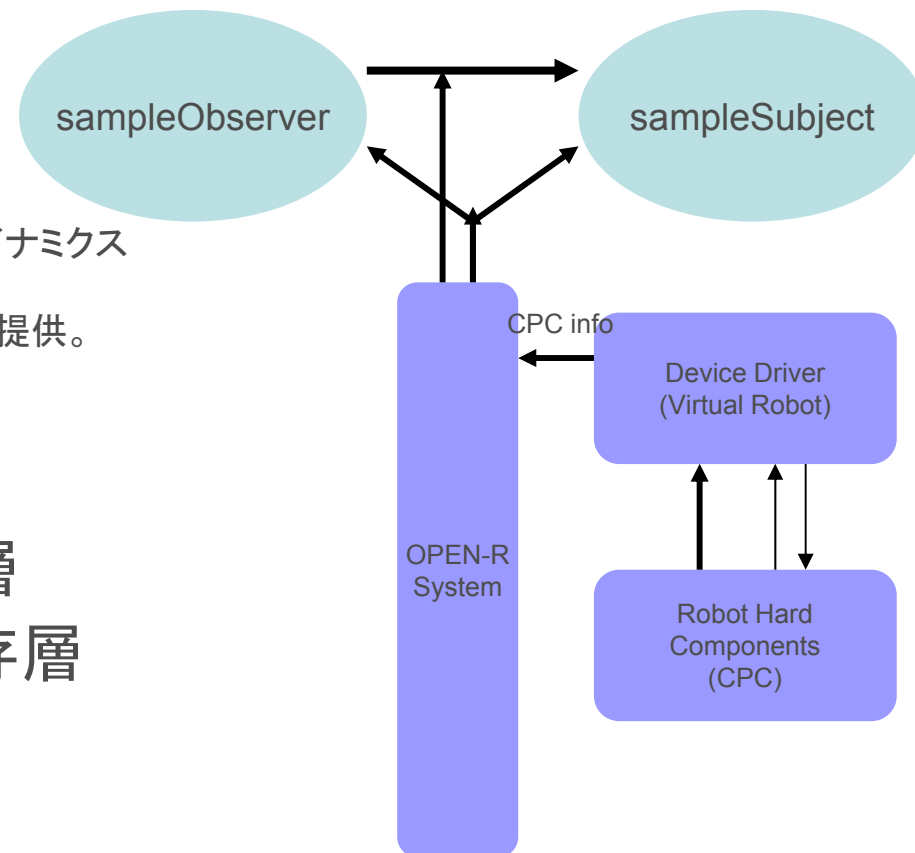
OPEN-Rアーキテクチャ

ハードウェアコンポーネント

- CPC (Configurable Physical Component)
 - 各コンポーネントは自身の機能、位置、ダイナミクスパラメータなどの情報を内蔵している。
 - ハードウェア自体がリフレクティブな機能を提供。

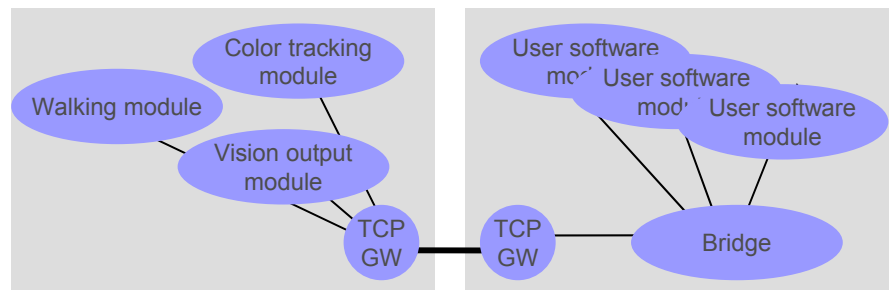
ソフトウェアコンポーネント

- コンフィギュレーション依存層
- コンフィギュレーション非依存層
- コンポーネント指向



OPEN-Rの特徴

- リモート処理
 - コードのポータビリティ
 - 組込システムとPC間
 - ネットワークを介したメッセージパッシング
 - ソフトウェアの再利用
- コンカレント開発環境
 - PCと組込み
- スケーラビリティ
- モジュール非依存性の確保
 - メッセージパッシング
- モジュラリティ
 - ハード&ソフト



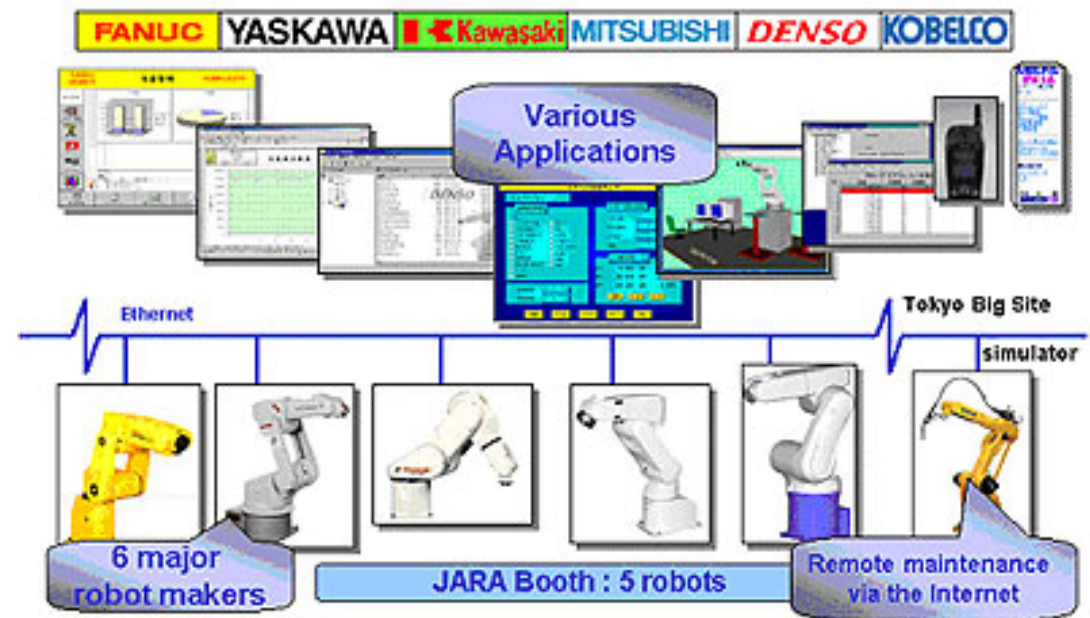
ORiN

<http://www.orin.jp/>



ORiN

- ORiN (Open Robot Interface for the Network)
 - 日本のFAロボット標準
 - FAロボットコントローラを抽象化
 - マルチベンダロボットシステムを容易に実現可能
 - メンバー:
 - ORiN consortium (FANUC, YASUKAWA, Kawasaki, MITSUBISHI, DENSO, KOBELCO)
 - <http://www.orin.jp/>

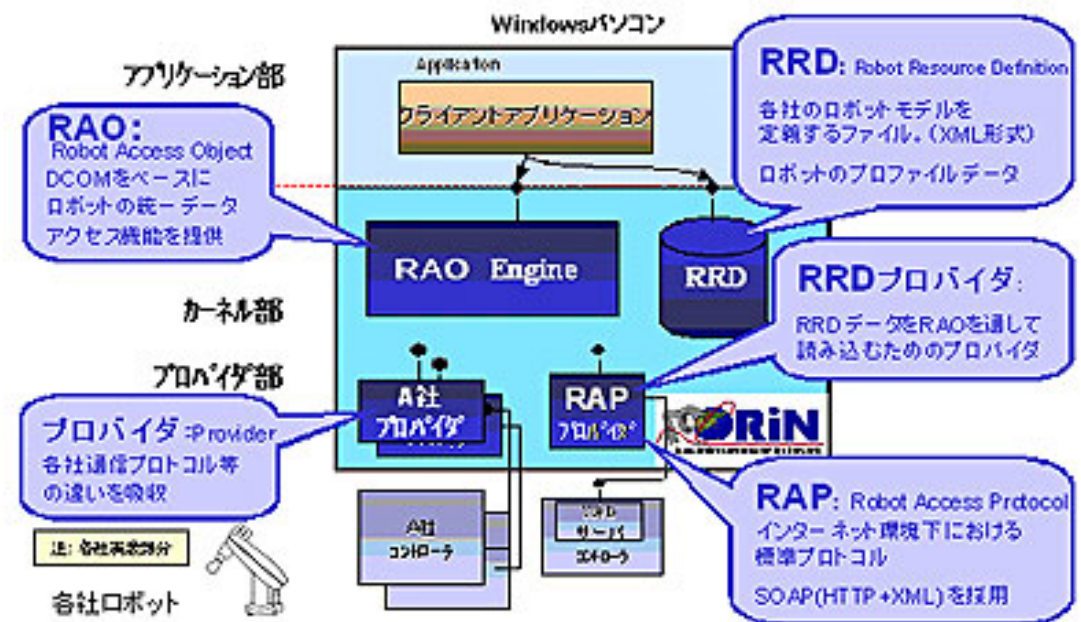


ORiN

- 設計ポリシー
 - 緩やかな標準化
 - 様々なタイプのロボット仕様を包含可能
 - 拡張性
 - ベンダ特有のオプションを定義可能
 - ネットワークプロトコルのモジュール化
 - 既存のロボットに適用可能
 - 実装と仕様の分離
 - OOP
- 実装ポリシー
 - デファクト標準
 - PC&Windows
 - Distributed object model (DCOM)
 - ネットワーク透過
 - 言語非依存
 - XML
 - ベンダ独自の仕様を記述するための標準フレームワーク
 - インターネット技術
 - HTTP, XML, SOAP

ORiNアーキテクチャ

- RAO (Robot Access Object)
 - ロボットコントローラに対する統一されたデータアクセシビリティを提供
- RRD (Robot Resource Definition)
 - ロボットプロフィールデータ
- RAP (Robot Access Protocol)
 - インターネットを介したアクセシビリティを提供



RSNP

(Robot Service Network Protocol)

<http://robotservices.org/>

<http://www.robotservices.org/wiki/jp/>



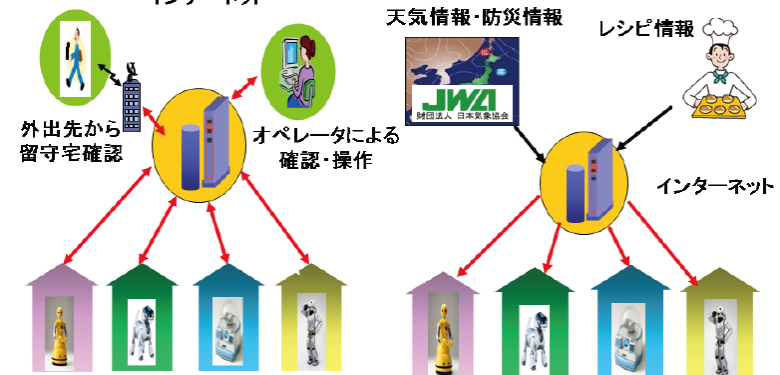
RSi (Robot Service Initiative)

- RSi (Robot Service Initiative)が主導となって定めるロボット用インターネットサービスのプロトコル
 - RSi:2004年発足
 - 富士通、三菱重工、東芝、安川電機、日本気象協会等が加盟
- ロボットをプラットフォームとしたインターネットと連携した新たなビジネス創出を目指す



例1 遠隔操作によるサービス
(見守りサービス等)
インターネット

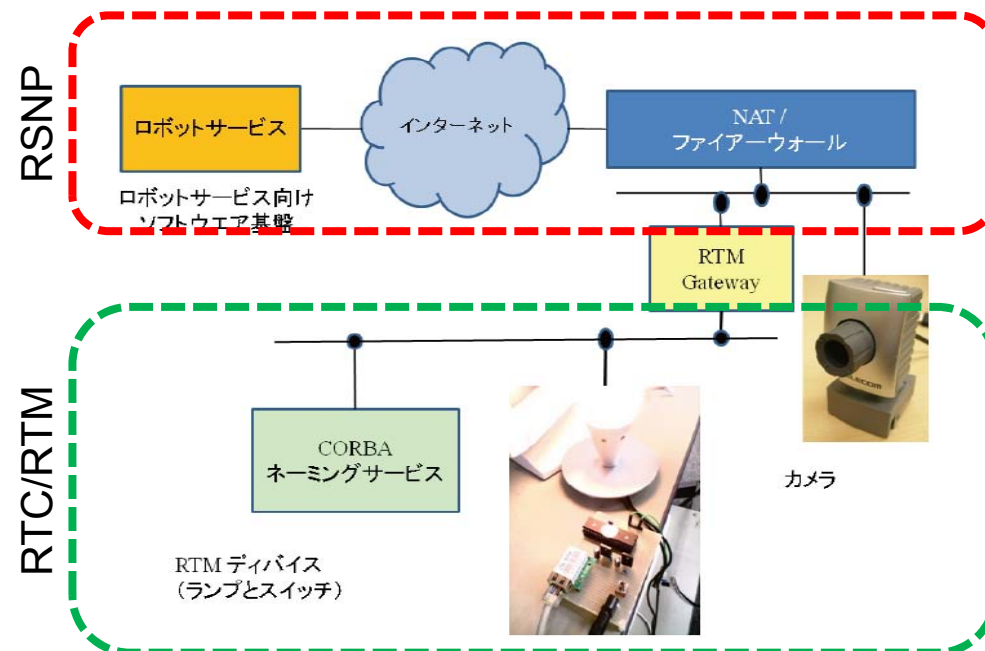
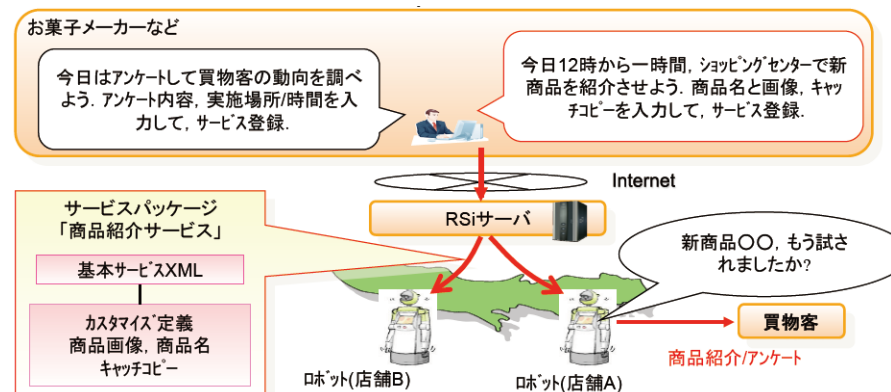
例2 情報提供サービス



© 成田雅彦, 産業技術大学院大学
ROBOMECH2013 RTM講習会資料より

RSNP

- RSiサーバ等で提供するサービスを各地のロボットを介して提供
 - 天気予報、見守り、ロボットマップ、各種ロボット制御
- SOAPを利用
- 疑似Push機能を利用しFirewall越し通信を実現
- 主としてロボットとインターネット(クラウド)との連携に利用
 - cf. RTC



© 成田雅彦, 産業技術大学院大学 ROBOMECH2013 RTM講習会資料より

UNR Platform

(Ubiquitous Network Robot Platform)

<http://www.irc.atr.jp/std/UNR-Platform.html>

ATR Advanced
elecommunications
Research Institute International

UNR Platform

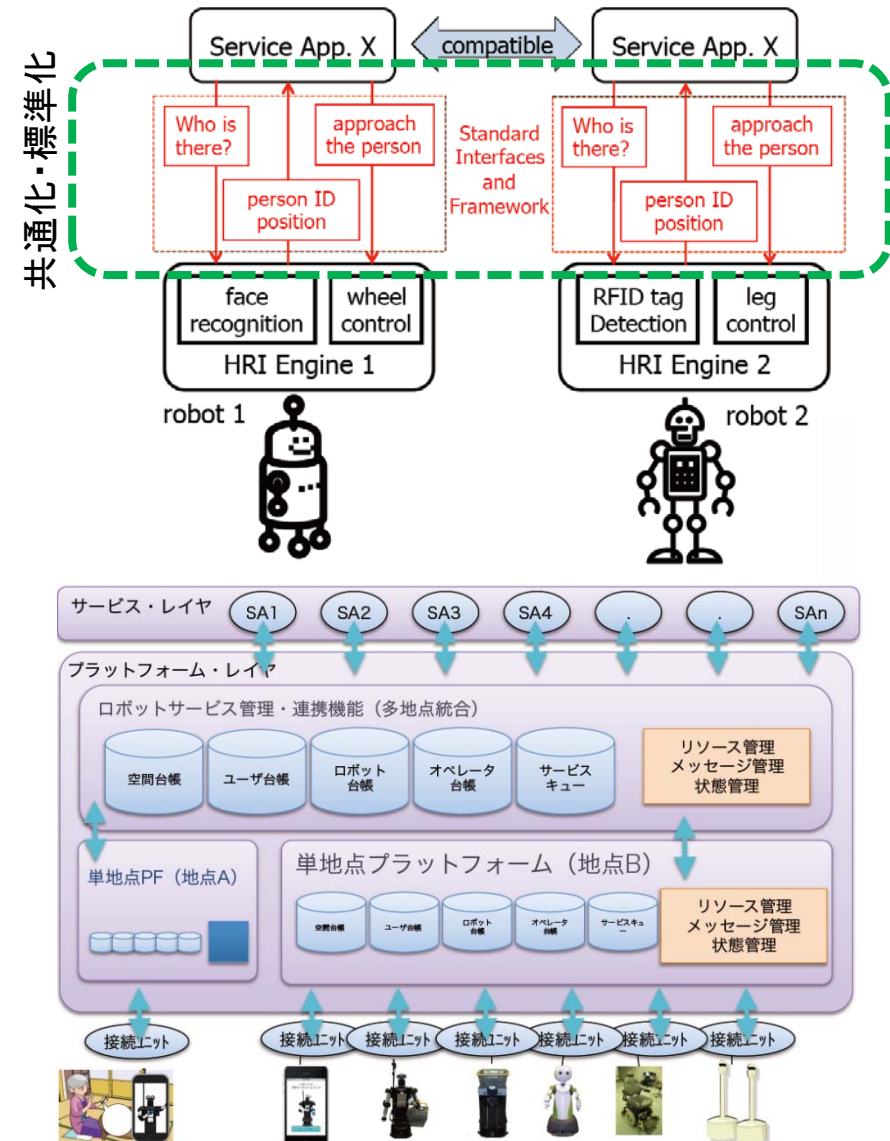
- 商業施設・病院・家などのさまざまな場所における人々の活動を支援
- ロボット、スマートフォンアプリ、環境センサがネットワークを介して連携し、多地点で人々にサービスを提供することを目指す
- ARTにより開発、配布



© 亀井剛次 ATR, CNR研究会2013

UNR Platform

- 一部はOMG RoIS (Robot Interaction Service)標準に準拠
- 個々のロボット仕様を気にせずアプリケーションを記述可能
- アプリと下位コンポーネントのデータのやり取りを仲介

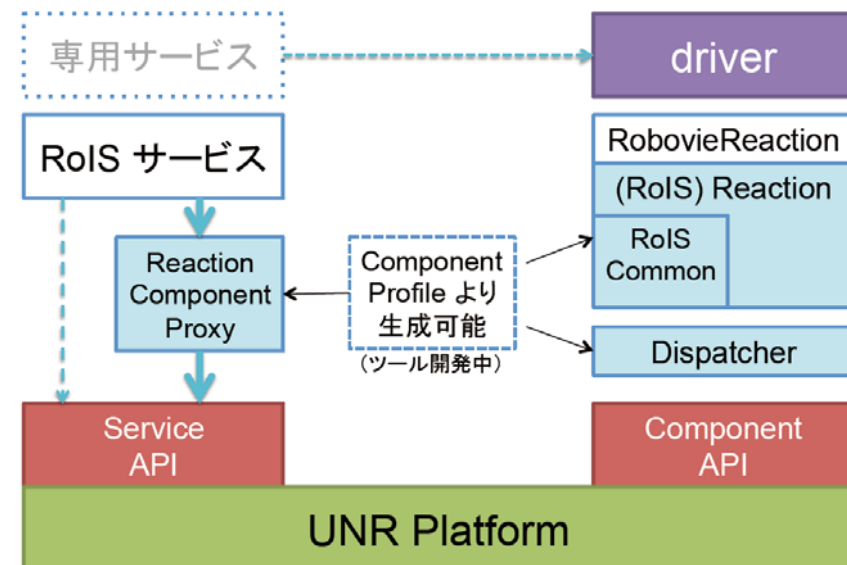


UNR Platform

- RoIS: ロボット対話サービスに必要な機能を標準化
- UNRプラットフォームは各種RoISサービスをクライアントの要求に応じて仲介
- ロボット側はRTMやROSなど何を利用してよい

RoIS機能コンポーネント群

1) システム情報	9) 音声認識
2) 人検出	10) ジェスチャ認識
3) 人位置検出	11) 音声合成
4) 個人同定	12) 応答動作
5) 顔検出	13) ナビゲーション
6) 顔位置検出	14) 追従
7) 音検出	15) 移動
8) 音源位置検出	



Comparison of open frameworks

	Target	Open spec/source	Real-time	Language	OS	modularity	communication
OpenRTM-aist	Universal	○/○ OMG RTC	○	C++, C, Python, Java, .NET, Android	UNIX, Mac OS X, Windows, uTRON, QNX, VxWorks	○ CBSD	CORBA
ROS	Universal	△/○	○	C++, Python, Java, LISP, Matlab	Linux, (OS X, Windows)	× Free style	original protocol
OROCOS	Universal	△/○	○	C++, (scripting: Lua)	Linux, Windows, Etc..	○ CBSD	Ice, CORBA
OPRoS	Universal	○/○ OMG RTC	○	C++	Windows, Linux	○ CBSD	Original protocol
YARP	Humanoid/ Universal	△/○	×	C++, Java, Python, Lua, Matlab	Linux, Windows, Mac OS X	△ Free style	Original protocol
ORCA/ORCA2	Universal	△/○	○	C++, Python	RTLinux, Other	○ CBSD	CORBA, CURD, Ice
MSRS	Universal	△/△	×	.NET (C++, C#, VB, etc.)	Windows	△ SOA	DSS·SOAP

Comparison of open frameworks

	Target	Open spec/source	Real-time	Language	OS	modularity	communication
OpenRTM-aist	Universal	○/○ OMG RTC	○	C++, C, Python, Java, .NET, Android	UNIX, Mac OS X, Windows, uTRON, QNX, VxWorks	○ CBSD	CORBA
ORiN	FA robots	○/× ISO ORiN	×	C++	Windows	△ OOP	DCOM, CORBA
RSNP	Internet Service	○/× RSi RSNP	×	Java	Java VM	△ OOP	SOAP
UNR Platform	Internet Service	○/○ OMG RoIS/RLS	×	Java	Java VM	△ OOP	SOAP
PlayerStage	Mobile robot	△/○	○	C, C++, Tcl, LISP, Java, and Python	Linux, Etc..	△ PO	original protocol
OPEN-R	AIBO, SDR3X	○/×	×	C++	Linux	△ OOP	original protocol
Open Robot Controller Architecture	Universal	△/×	×	Java, Python	Linux, Etc..	△ OOP	HORB

終わりに

- ロボット新戦略とロボットOS・ミドルウェア
- RTミドルウェアの目的、アーキテクチャ、応用
- その他のロボットOS/ミドルウェアの動向

詳しくは...

検索



本日の資料はopenrtm.orgに掲載します。

<http://openrtm.org/openrtm/node/5934>

レポート課題(1)

1. ミドルウェアを利用したサンプルプログラムを示せ
 - a. ロボットミドルウェアを一つ選び、データの送信を行う手順・方法を調べ説明せよ。結果として、コメントを付したソースコード(完全である必要はないが、データ送信に必要な最低限の部分を示すこと)を添付せよ。
 - b. 同様に、データの受信を行う手順・方法を調べ説明せよ。結果として、コメントを付したソースコード(完全である必要はないが、データ送信に必要な最低限の部分を示すこと)を添付せよ。

コードに付記されたコメントを重視します。

レポート課題(2)

2. サービスロボットを設計せよ

- a. どのようなサービス・機能を提供するロボットか仕様を定めよ。
- b. そのロボットが有するハードウェア(センサ、アクチュエータ等)を仮定せよ。
- c. そのロボットの持つ機能のうち少なくとも一つを実現するために必要なソフトウェアコンポーネントを考え、データ・コマンドの流れとともに図示せよ。

3. 授業の感想

仕様とコンポーネントの整合性に注意。