Robotics Domain Task Force Final Agenda ver.1.0.5 **robotics/2012-12-01**

OMG Technical Meeting - **Burlingame, CA, USA** -- December 10-14, 2012

http://robotics.omg.org/

		TF/SIG				
		Host	**Joint (Invited**	**Agenda Item**	**Purpose**	**Room**
Sunday: Planning WG (pm)						
16:00	17:00			Robotics roadmap planning WG	Arrangement	Harbar B, Lobby Lv1
Monday: WG activities (pm)						
12:00	13:00			**LUNCH**		Atrium
13:00	18:00			Architecture Board Plenary		Harbor B, Lobby Lvl
13:00	17:00			Infrastructure WG (New Work Item Discussion) - Noriaki Ando(AIST) and Seung-woog Jung(ETRI)	discussion	Sandpebble E, Lobby Lvl
Tuesday: Robotics Information Day						
8:30	8:40	Robotics		**OMG Welcome**	greeting	Regency B, Lobby Lvl
8:40	9:20	Robotics		**Keynote : Intelligent RT Software Project -- National Project in Japan** - **Tomomasa Sato** (Professor, Univ. Tokyo, Japan)	presentation and discussion	
9:20	10:00	Robotics		**Keynote : Implementation of OPRoS to a human-friendly guide robot, FURO** **- Se-Kyeong Song**(CEO, Future Robot, Korea)	presentation and discussion	
				Morning Break (30min)		
10:30	11:00	Robotics		**Keynote: ISO Activity of Service Robot** - **Seongbin Moon** (ISO/TC184/SC2/WG8 Convenar, Sejong Univ.)	presentation and discussion	
11:00	11:45	Robotics		**Keynote: A Disruptive Community Approach to Industrial Robotics Software** **- Paul Evans** (Director, Manufacturing Systems Dept., Southwest Research	presentation and discussion	
12:00	13:00			**LUNCH**		Atrium
13:00	13:10	Robotics		Introduction to Robotic Technology Component (RTC-1.1) Specification (10min) - Geoffrey Biggs (RTC-1.1-RTF Chair)	presentation and discussion	Regency B, Lobby Lvl
13:10	13:20	Robotics		Introduction to Robotic Localization Service (RLS-1.1) Specification (10min) - Koji Kamei(JARA/ATR) and Shuichi Nishio (RLS-1.1-RTF Chair)	presentation and discussion	
13:20	13:30	Robotics		Introduction to Robotic Interaction Services Framework (RoIS-1.0) (10min) - Koji Kamei and Su-Young Chi (RoIS-FTF Co-Chairs)	presentation and discussion	
13:30	13:40	Robotics		Introduction to Dynamic Deployment and Configuration for RTC (DDC4RTC) (10min)	presentation and discussion	
13:40	14:10	Robotics		Using SysML in a RTC-based Robotics Application : a case study with a demo - Kenji Hiranabe(Change Vision) and Noriaki Ando (AIST)	presentation and discussion	
14:10	14:40	Robotics		Robotic Technology Component Interoperability Demo - Makoto Sekiya (Honda R&D Co., Ltd.)	presentation and discussion	
				Afternoon Break (20min)		
15:00	15:30	Robotics		Implementation of RoIS to robots in ETRI - Su-Young Chi (ETRI)	presentation and discussion	
15:30	16:00	Robotics		Component Management in OPRoS (Open Platform for Robotic Services) - Seung-Woog Jung (ETRI)	presentation and discussion	
16:00	16:30	Robotics		Cloud Networked Robotics and Acceleration Based Sensing - Miwako Doi (Toshiba)	presentation and discussion	
16:30	17:00	Robotics		Introduction to OpenEL (Open Embedded Library) for Robots - Kenichi Nakamura (JASA:Japan Embedded Systems Technology Association)	presentation and discussion	
17:00	17:40	Robotics		Discussion: Roadmap to the future activities in Robotics - All participants	discussion	
17:40				Adjourn Information Day meeting		
Wednesday: WG activitiy						
9:00	12:00			Robotics WG activity follow-up	discussion	Sandpebble E, Lobby Lvl
12:00	14:00			**LUNCH and OMG Plenary**		Pool Pavillion
14:00	16:00			Robotics WG activity follow-up	discussion	Sandpebble E, Lobby Lvl
16:00	16:10	Robotics		Robotics-DTF Plenary Opening Session (minitues approval, minutes taker)	presentation and discussion	Sandpebble E, Lobby Lvl
16:10	16:45	Robotics		WG Reports and Discussion (Service WG, Infrastructure WG, Models in Robotics WG)	presentation and discussion	
16:45	17:00	Robotics		Robotics-DTF Plenary Wrap-up Session (**DTF Co-Chair Election,** Roadmap and Next meeting Agenda)	Robotics plenary closing	
17:00				Adjourn joint plenary meeting		
Thursday: WG activitiy						
12:00	13:00			**LUNCH**		Atrium
13:00	18:00			Architecture Board Plenary		Harbor B, Lobby Lvl
Friday						
8:30	12:00			AB, DTC, PTC		Regency ABC, Lobby Lvl
12:00	13:00			**LUNCH**		Atrium
				Other Meetings of Interest		
Monday						
8:00	8:45	OMG		New Attendee Orientation		Sandpebble D, Lobby Lvl
9:00	12:00	OMG		Introduction to OMG Specifications Tutorial		Sandpebble D, Lobby Lvl
Tuesday						
7:30	9:00	OMG		Liaison ABSC		Regency C, Lobby Lvl
15:00	17:00	OMG		IPR Policy Transition Briefing		Bayside, Lobby Lvl
17:00	18:00	OMG		RTF-FTF Chair's Workshop		Boardroom 2, Atrium Lvl
Wednesday						
9:00	18:00	SysA		System Assurance PTF		Regency C, Lobby Lvl
18:00	20:00	OMG		OMG Ateendee Reception		Regency Foyer, Lobby Lvl
Thursday						
16:00	17:00	MARS		MARS Chairs Planning/Agenda Coordination		Regency A, Lobby Lvl

Please get the up-to-date version from http://staff.aist.go.jp/t.kotoku/omg/RoboticsAgenda.pdf

Minutes of the Robotics Domain Task Force Meeting
June 18-22, 2012
Cambridge, MA, USA
(robotics/2012-12-02)

Meeting Highlights

- The DDC4RTC submission was reviewed and adopted in MARS-PTF and AB.
- The DDC4RTC-FTF charter was adopted.
- The RoIS-FTF Final Report was reviewed in AB, but a vote by e-mail will be held on next Monday. (There is no time to review the revised report)

List of Generated Documents

mars/2012-05-02 Dynamic Deployment and Configuration for Robotic Technology Component (DDC4RTC) Specification Revised Submission (Noriaki Ando)
mars/2012-05-06 ComponentDataModel.xsd
mars/2012-05-07 ExecutionDataModel.xsd
mars/2012-06-08 DDC4RTC Presentation on Mon. (Noriaki Ando)
mars/2012-06-09 Errata
mars/2012-06-10 DDC4RTC with change-bar
mars/2012-06-11 DDC4RTC without change-bar
mars/2012-06-12 Updated XMI file
mars/2012-06-13 Updated IDL file
mars/2012-06-23 Updated Inventory file
mars/2012-06-26 DDC4RTC Presentation on Thu. (Noriaki Ando)

dtc/2012-05-28 RoIS FTF Report (Toshio Hori)
dtc/2012-06-25 Updated RoIS FTF Report
dtc/2012-06-26 Updated Inventory file
dtc/2012-06-27 Updated Beta 2 document without change bars
dtc/2012-06-28 Updated Beta 2 document with change bars
dtc/2012-06-29 Updated C++ PSM header file
dtc/2012-06-30 Updated CORBA PSM IDL file
dtc/2012-06-31 Updated XML PSM schema file
dtc/2012-06-32 Updated XMI file
dtc/2012-06-33 Errata

robotics/2012-06-01 Final Agenda (Tetsuo Kotoku)
robotics/2012-06-02 Reston Meeting Minutes [approved] (Geoffrey Biggs and Seung-woog Jung)
robotics/2012-06-03 Opening Presentation (Tetsuo Kotoku)
robotics/2012-06-04 Roadmap for Robotics Activities (Tetsuo Kotoku)
robotics/2012-06-05 Contact Report: Standardization of RTC-CANopen in CiA (Makoto Mizukawa)
robotics/2012-06-06 Robotic Functional Service WG Report (Toshio Hori)
robotics/2012-06-07 Infrastructure WG Progress Report (Noriaki Ando)
robotics/2012-06-08 Wrap-up Presentation (Tetsuo Kotoku)
robotics/2012-06-09 Next Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)
robotics/2012-06-10 DTC Report Presentation (Tetsuo Kotoku)
robotics/2012-06-11 DDC4RTC-FTF Charter (Noriaki Ando)
robotics/2012-06-12 Reston Meeting Minutes - DRAFT (Geoffrey Biggs and Seung-woog Jung)

Minutes

Tuesday, 19 June, 2012, William Dawes B, Lobby Lvl
Robotics DTF Plenary Meeting
AIST, ETRI, JARA, Shibaura-IT, Univ. of Tsukuba, (Quorum: 3)
10 attendees

13:00 - 13:10 Robotics-DTF Opening Session,
Chair: Tetsuo Kotoku (AIST)

- Minutes takers: Seung-Woog Jung (ETRI) and Geoffrey Biggs (AIST)
- Brief summary of Reston meeting
  - 6 participants
  - 1 special talk
  - 2 WG reports
  - DDC4RTC passed MARS but was rejected by the AB due to too many changes to the submission.
  - Deadline was extended one meeting.
- Reston Meeting minutes (robotics/2012-06-02) approved
  - AIST (motion), Shibaura-IT (second), Univ. of Tsukuba (white ballot)

13:10 - 13:25 Contact report: Standardization of RTC-CANopen in CiA,
Makoto Mizukawa (Shibaura-IT)

- RT-Component standardized by the OMG in 2008.
- CANopen is a widely-used technology.
- RTC-CANopen is an RT-Middleware for embedded systems.
- Advantages:
  - Existing CANopen devices in robotics can be used.
  - Many MPUs support CAN.
  - CANopen is already known to be quite reliable.
- 2008: Start negotiation with CiA (CAN in Automation)
- 2011.11 Working draft issued
  - CiA 318: Implementation guideline - Mapping of RTC to CANopen
  - CiA 460: Service robot controller profile
  - NMT master application and CANopen device proxies
- 2012.02 DSP (Draft Standard Proposal) issued

13:25 - 13:35 Contact report: ISO TC 184 WG meetings, Politecnico di Milano, Italy,
Su-Young Chi (ETRI)

- ISO TC 184/WG 8 (Service robots), July 11-12, 2012
- RoIS is one of the candidate of NWIP

13:35 - 13:45 Working group report: Robotic Functional Service WG,
Toshi Hori (AIST)

- A meeting was held in Seoul in May to work on the remaining issues for the RoIS framework.
  - 5 participants
  - 27 issues raised and discussed
    - 20 resolved
    - 3 deferred
    - 2 closed without changed
    - 2 merged with others
- Three polls were held in April and May.

- Sam (Sparx) was removed from the FTF due to his leaving Sparx and OMG activities.
- WG activities during this meeting
  - Attended the AB on Monday afternoon. Several problems were pointed out in the XML PSM. These have been resolved.
  - A revised beta document will be posted by Thursday.
  - The revised document will go to a vote at the AB on Thursday afternoon, or, if the revised document is not ready in time, by an email vote.
- FTF report deadline is 29 June, 2012.

13:45 - 13:50 Working group report: Infrastructure WG,
Noriaki Ando (AIST)

- Activities at the Reston meeting:
  - 64 issues from the AB comments. All were resolved.
  - Approved in MARS
  - Rejected by the AB: Too many modification from the 4-week document
- Revised specification submitted in May
  - 15 comments from the AB review and 4 from MARS (Remedy IT)
  - 14 AB comments and 3 MARS comments are resolved. 2 issues remain.
- Vote will be held in MARS on Thursday morning. If passed, then it will go to the vote in the AB on Thursday afternoon.

13:50 - 16:00 Robotics DTF Wrap-up Session,
Chair: Tetsuo Kotoku (AIST)

- Robotics-DTF Co-Chair selection: postpone one more meeting
- Roadmap discussion.
  - An exhibition and Information day will be held at the December meeting presenting products developed based on the OMG's robotics standards.
- No meeting will be held in September.

ATTENDEE (10 attendees):

  - Seung-Woog Jung (ETRI)
  - Su-Young Chi (ETRI)
  - Young-Jo Cho (ETRI)
  - Tetsuo Kotoku (AIST)
  - Noriaki Ando (AIST)
  - Geoffrey Biggs (AIST)
  - Toshio Hori (AIST)
  - Koji NAMEI (ATR)
  - Takashi Tsubouchi (Univ. of Tsukuba)
  - Makoto Mizukawa (Shibaura-IT)

Prepared and submitted by Geoffrey Biggs (AIST) and Seung-Woog Jung (ETRI)

Intelligent RT Software Project

～Next Generation Robot Projects sponsored by NEDO: Ministry of Economy, Trade and Industry～

Problems
to be solved

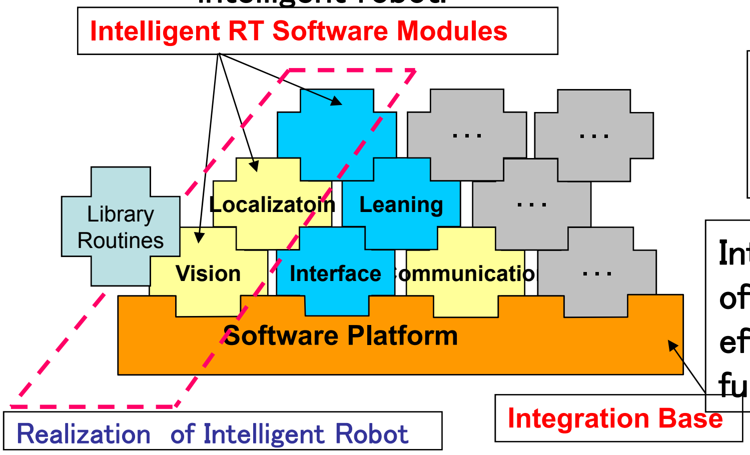
・The robot system tend to be developed from scratch.

・It takes time to reach high level robot performance.

Project Requirements

●Software platform is needed on which intelligent RT software modules can be integrated.

●Robust intelligent software modules are required to realize an intelligent robot.



Practical Robot

Robot + RT Services

+ Social mechanism

Intelligent RT software modules offer basis of realizing cost-effective robot with useful functions.

Re-usability of software modules：Project Key

・Modules should have common interface	Set common interface by establishing sub-WG
・Modules should have integrity	Repeated integration evaluation by performing demonstration
・Modules should be exchangeable	・Accepted research group focused on module utilization ・Combined research groups to stimulate mutual utilization of modules ・Made full use of determined common interface
・Modules should be easy to use	・Started open source software development ・Put stress on writing manuals and documents
・Modules should be useful	・Evaluated developed modules by realizing ann integrated robot ・Wrote documents and manuals
・Modules should be complete to realize intelligent robot	・Evaluates the completeness of software modules by demonstration during international exhibition

Research Targets

4 Targets consisting of 8 Research Items

Ⅰ. Software Platform

①－1 Development of intelligent robot software platform

①－2 Improvement of reusability of software modules

Ⅱ. Intelligent software modules for manipulation

② Manipulation inteligence（Industrial robot）

③ Manipulation intelligence（Social ・Life area）

Ⅲ. Intelligent software modules for navigation

④ Navigation Intelligence（Service robot）

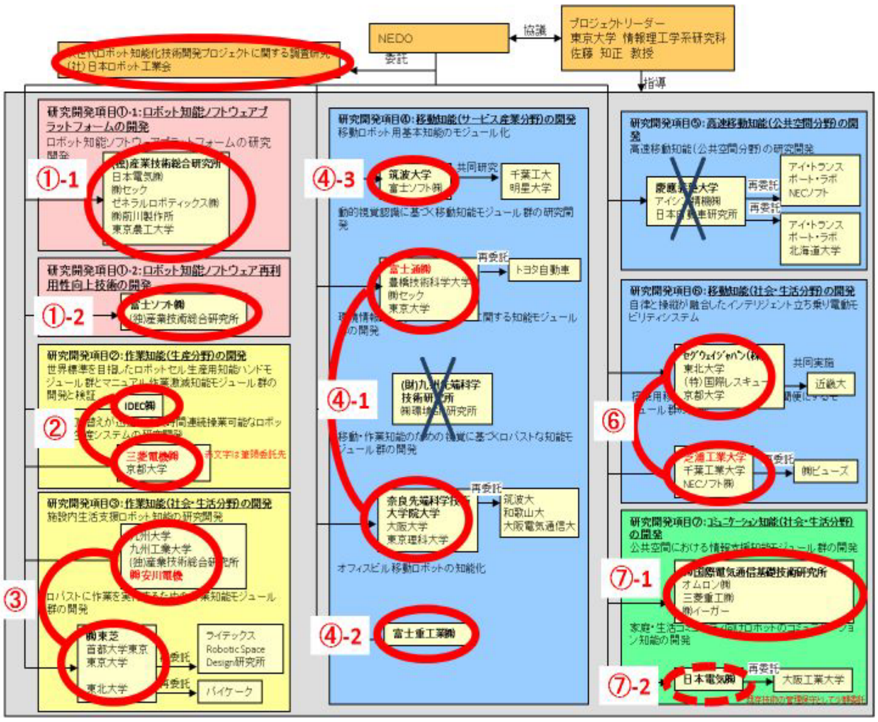
⑤ Rapid navigation intelligence（Public space）

⑥ Navigation Intelligence（Social ans Life area）

Ⅳ. Intelligent software for communication

⑦ Communication Intelligence（Social life area）

Project Formation
at the beginning and after midterm evaluation



再利用性の強化による実用化への対応

1. 他のコンソとのつながりが薄い（独立している）事業を終了した

2. 統合検証を行うため、開発内容が近いコンソを統合した

3. 一部モジュールをオープンソースで提供することとした

Research Coverage and Research Institutions

Basis

①Robot Software Platform (AIST,NEC,SEC,Mayekawa,GR,TUAT)

①-2 Intelligent RT Software Modules Verification (AIST,Fujisoft)

Manipulation

②Manipulation Intelligence（Industrial Field）(Mitsubishi Electric, Kyoto Univ.,IDEC)

③Manipulation Intelligence（Social･Life related Field）(Toshiba, Tokyo Metropolitan Univ., Univ.Tokyo,Tohoku Univ.,Rightechs,Robotics Space Design, PieCake, KYUTECH, Kyushu Univ., AIST,Yaskawa)

Mobility

④Mobility Intelligence（Service Industrial Field）(Fujitsu,TUT,SEC,Univ.Tokyo,Toyota,Univ.Tsukuba,Fujisoft,Chiba I.T.,Meisei Univ.,NAIST,Tokyo Univ. Science,Osaka Univ.,Wakayama Univ.,OECU, Fuji Heavy, ISIT, ENGIS)

⑤High-speed Mobility Intelligence（Public Facility Field）(Keio Univ., JARI、AISIN, itransport,NECsoft,Hokkaido Univ.)

⑥Mobility Intelligence（Social･Life related Field）(Segway Japan, Kyoto Univ.,IRSI,Tohoku Univ.,Kinki Univ.,Shibaura I.T.,Chiba I.T.,NEC Soft, PUES)

Communication

⑦Communication Intelligence
（Social･Life related Field）(NEC, Osaka I.T. , ATR,Omron, MHI, Eager)

Project Management

2007	■Rapid start of project Add new research team responsible for only module utilization
2008	■ Start sub-working group of manipulation, navigation and communication to realize common interface amang modules Execution of demonstration in early project stage
2009	■Started re-usable center to verify all modules ■Project steering at every Thursday from Akihabara Midterm-evaluation ■Started Working Team and rearrangement of research teams
2010	■Open source development by additional funding ■Development of dual eye&arm robot software
2011	■Promoted final evaluation of all developed modules ■RTM-Ros interoperability project ■Development of RTM safety ■Efforts to make the module in practice

Project Basic Output

Software Platform

～Overview of Intelligent RT Software Platform～

- Integrated development set for developing RT components, RT systems and application scenarios

OpenRT Platform

Specification

- ■Hardware
- ■RT Components
- ■RT Systems
- ■Scenario Descriptions
- ■Motion Pattern

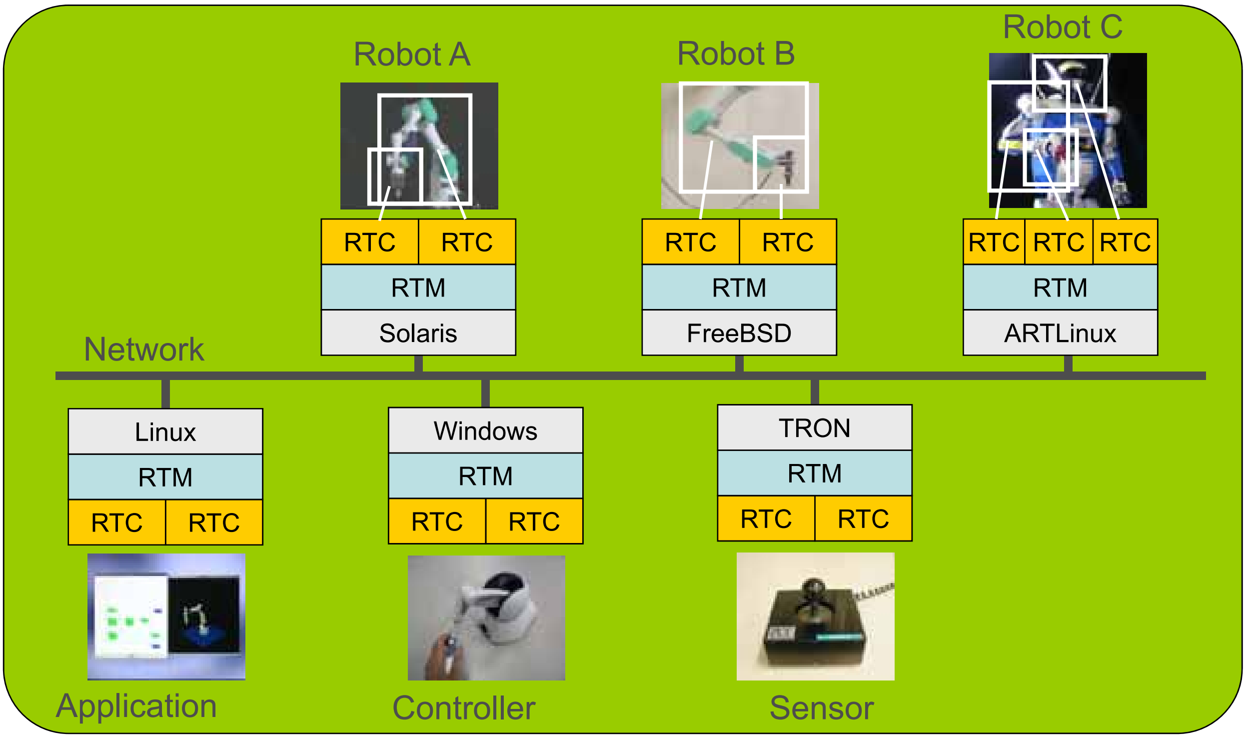
Middleware for modularization

- ■**OpenRTM-aist**

Tools

- For Components
  - RTC Builder
  - RTC Debugger
  - RT System Editor
- For System
  - Hardware Design Tool
  - RT Repository
  - Simulator (OpenHRP3)
- For application
  - Motion Planner
  - Scenario Editor
  - Real-Time verification tool

Robot System based on OpenRTM-aist-1.1



Robot A

RTC	RTC
RTM	
Solaris	

Robot B

RTC	RTC
RTM	
FreeBSD	

Robot C

RTC	RTC	RTC
RTM		
ARTLinux		

Network

Linux	
RTM	
RTC	RTC

Windows	
RTM	
RTC	RTC

TRON	
RTM	
RTC	RTC

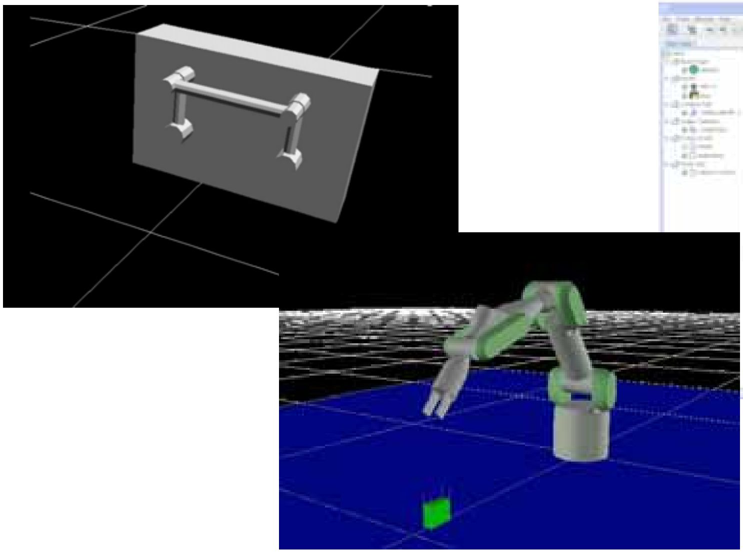
Application

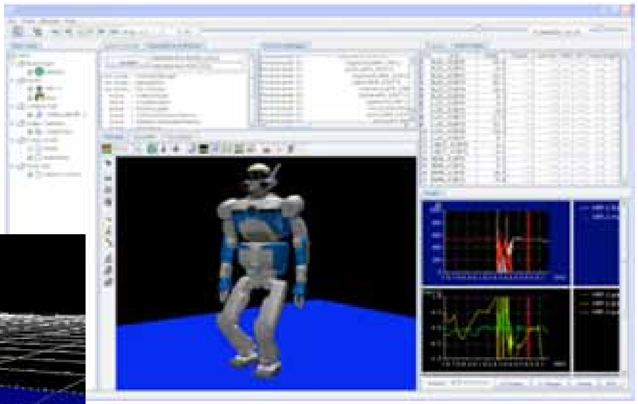
Controller

Sensor

Dynamics Simulator: OpenHRP3

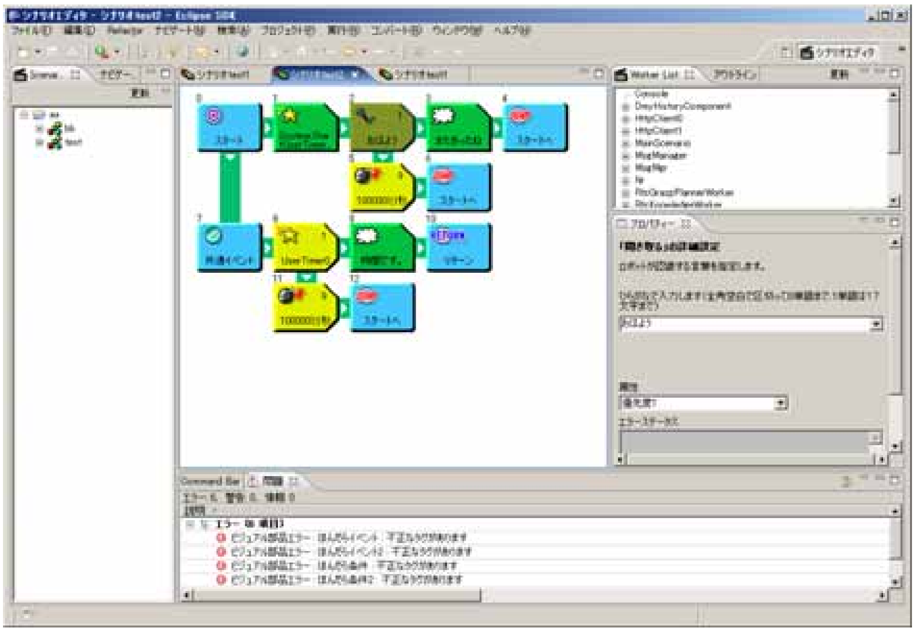
- Featherstone's O(N) algorithm
- Contact force simulation using LCP solver
- Sensor simulation :
  accelerometer, gyro, force/torque sensor, camera, range sensor





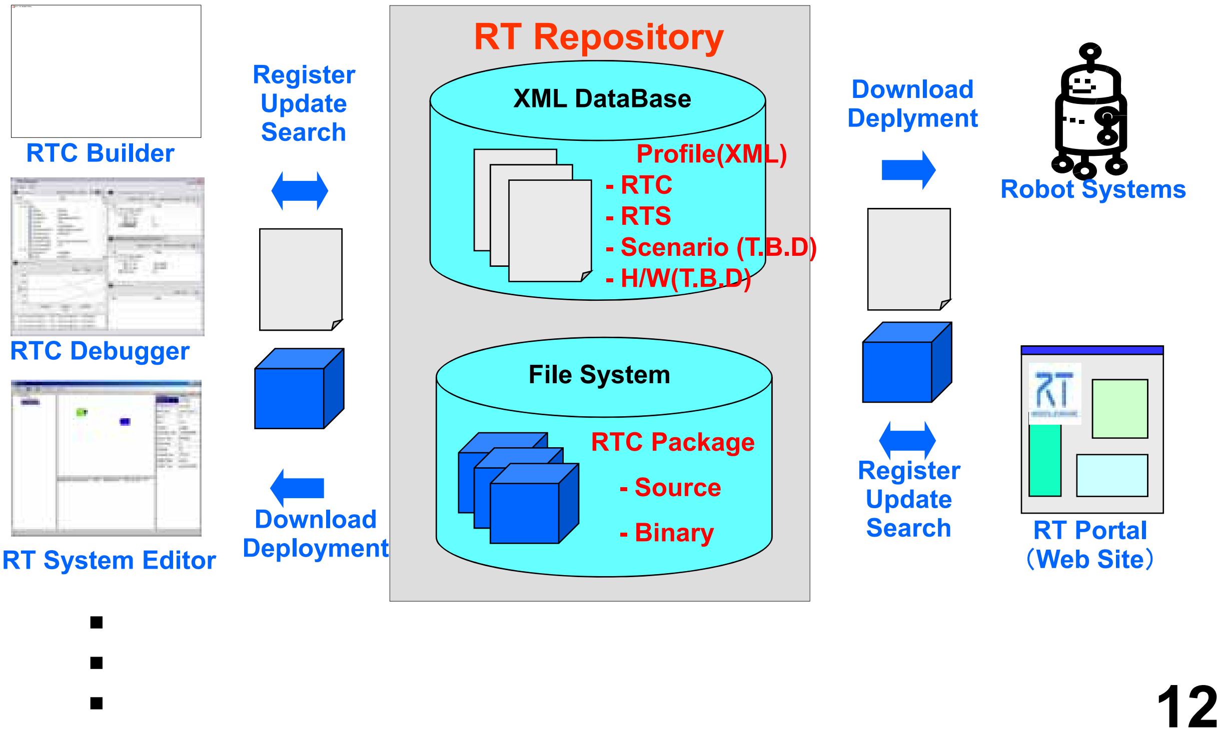
Scenario Editor

- Create and edit a sequence of motion patterns by using a script language or GUI
- Control event flow among RT Components



RT Repository

•Public/personal database for RTC/RTS etc.



RT Repository

XML DataBase

Profile(XML)
- RTC
- RTS
- Scenario (T.B.D)
- H/W(T.B.D)

File System

RTC Package
- Source
- Binary

RTC Builder

RTC Debugger

RT System Editor

■
■
■

Register
Update
Search

Download
Deployment

Download
Deplyment

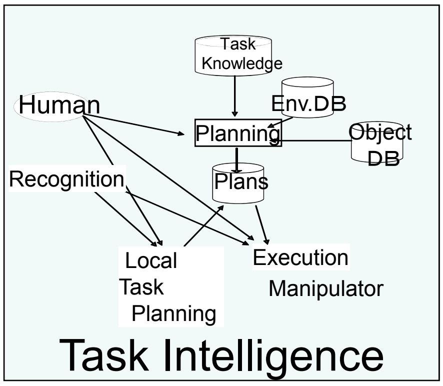
Robot Systems

Register
Update
Search

RT Portal
（Web Site）

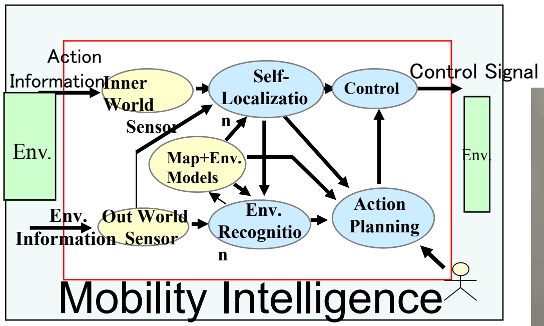
Intelligent RT Software Modules

Manipulation


Task Knowledge
Human
Env.DB
Planning
Object DB
Recognition
Plans
Local Task Planning
Execution
Manipulator
Task Intelligence

Intelligent RT Software Modules

Navigation


Action Information
Inner World Sensor
Self-Localization
Control
Control Signal
Env.
Map+Env. Models
Env. Information
Out World Sensor
Env. Recognition
Action Planning
Env.
Mobility Intelligence

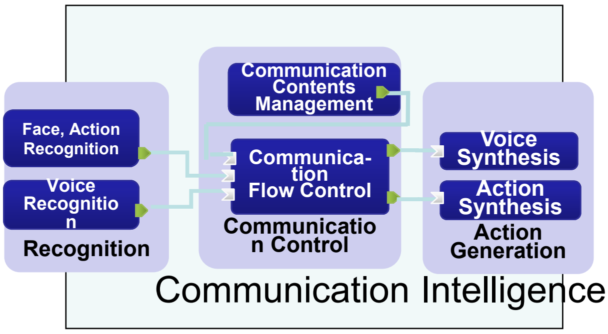
Prototype 1

Prototype 1
With ABS cover

Prototype 1
With Metal cover

Intelligent RT Software Modules

Communication



Face, Action Recognition

Voice Recognition

Recognition

Communication Contents Management

Communication Flow Control

Communication Control

Voice Synthesis

Action Synthesis

Action Generation

Communication Intelligence

Number of developed Modules

Number of Developed Modules（ Target：340）					
2007	2008	2009	2010	2011	Total
48	136	120	16	42	362

Intellectual Properties

	Patent （出願）	Research Presentation （Papers、Conference）		PR	
		International	Domestic	Mass Media	Exibition
2007～2009	50（0）	55	336	119	57
2011	13（7）	53	172	44	5
2012	5（0）	51	99	60	11
Total	6	159	582	223	73
	8(7)	766			

Towards Practical Utilization of RT modules ～Steps and Efforts～

1st Step

- Accumulation of practical RT modules

2nd Step

- To offer software platform and RT modules

3rd step

- Contribution to National Requirements

Towards Practical RT　Module Application

1st Step

- Accumulation of practical RT modules

1）Open　source　development of intelligent RT components for palletizing tasks by dual-eye&arm　robot

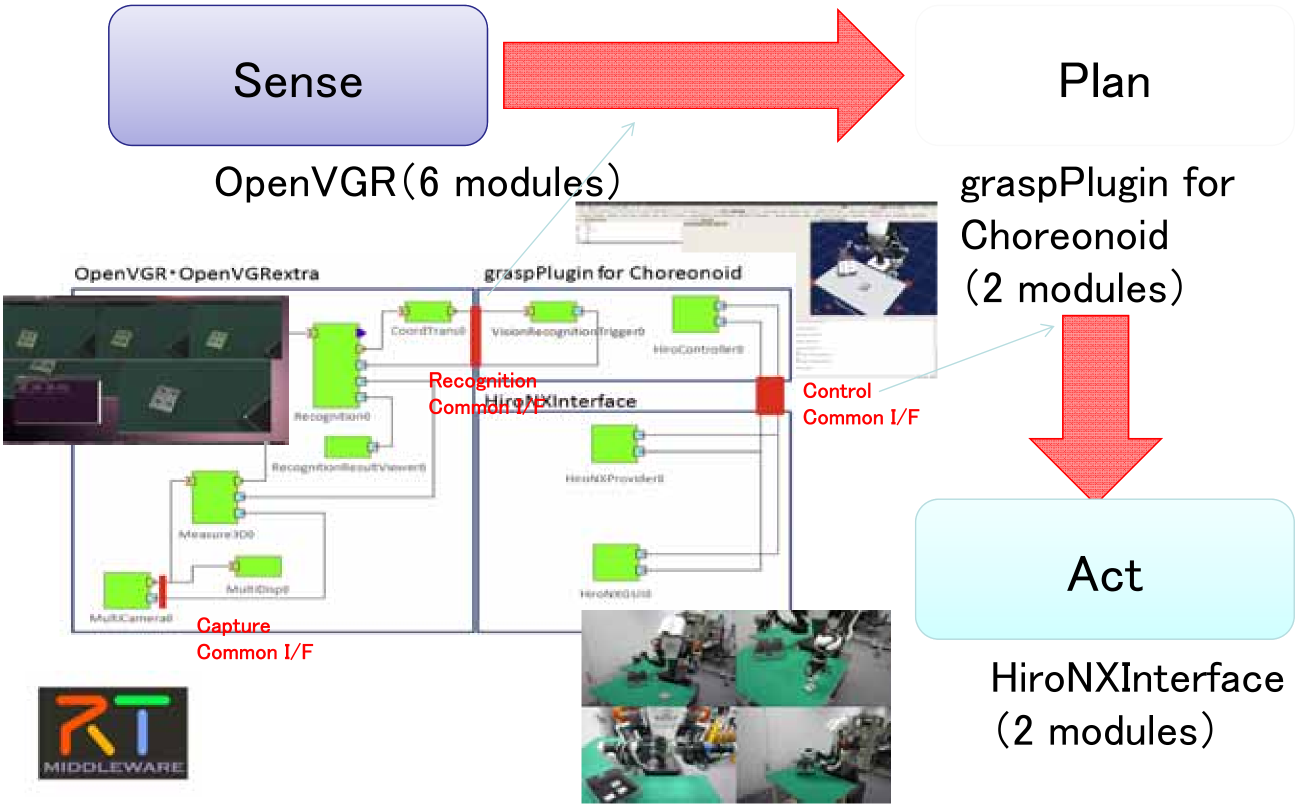
2）Open　source　development of　intelligent RT components for assort tasks by dual-eye&arm robot

---

1）Open Source Development of Intelligent RT Components for Palletizing tasks by Dual-eye&arm Robot

作業例：ピッキング、パレタイジング、搬送など

System



Sense

OpenVGR（6 modules）

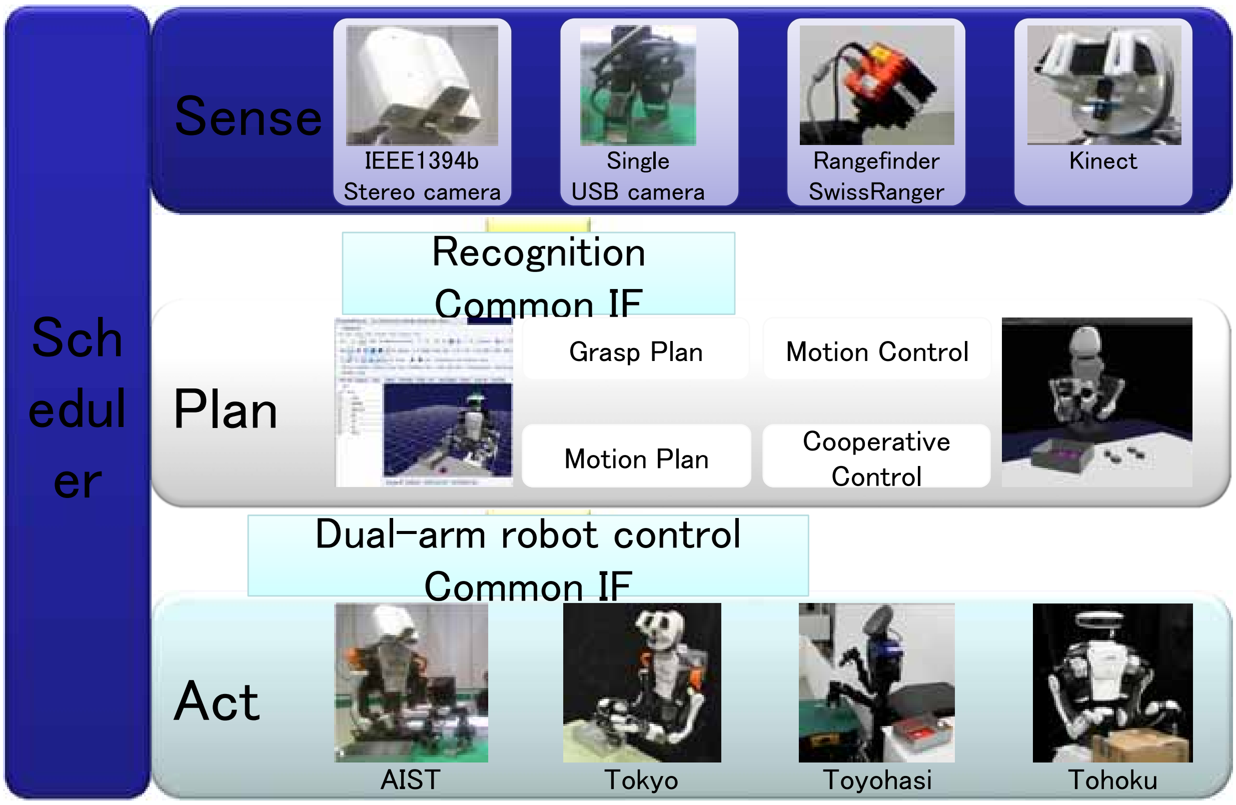
Plan

graspPlugin for Choreonoid
（2 modules）

Act

HiroNXInterface
（2 modules）

User Can Select the best Module



Scheduler

Sense

- IEEE1394b Stereo camera
- Single USB camera
- Rangefinder SwissRanger
- Kinect

Recognition Common IF

Plan

- Grasp Plan
- Motion Control
- Motion Plan
- Cooperative Control

Dual-arm robot control Common IF

Act

- AIST
- Tokyo
- Toyohasi
- Tohoku

Video of each system

AIST 頭部ステレオカメラを用いた双腕ロボット
によるマニピュレーション作業

東京大学 手先カメラを用いた双腕ロボットによる
マニピュレーション作業システム

TOYOHASHI 双腕ロボットとAGVの連携システム

双腕ロボットによる双腕協調
マニピュレーション作業

---

2)Open Source Development of Intelligent RT Components for Assort tasks by Dual-eye&arm Robot

Assort tasks for
service robot

NAIST

Osaka
University

Tokyo University
of Science

Tsukuba
University

AIST

Over-the-counter sales robot system

This project presents a dual-arm service robot system reusing open source RT components. The service robot system has four functions which are user detection, user interface, ordered object(Japanese confections) recognition and ordered object manipulation. Each function is developed by several groups as an open source RT module.

Scenario

Python sample scenario（IREX2011）

Each module controlled by the scenario

User detection

Camera image based modules
LRF data based modules

Object recognition

Template matching based modules
SIFT based modules
Background subtraction based modules
Bar code based modules

User interface

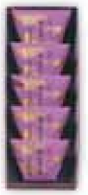
Web browser based modules
Voice input based modules

Robot Control

Dual-arm control modules

Target: Japanese confections


Turuyahatiman


Turuyahatiman


Tsuruya Yoshinobu


Fugetsudo


Turuyahatiman


GODIVA


Tsuruya Yoshinobu


MARKT


Morozoff


Taneya


Taneya


ANTÉNOR


PIERRE HERMÉ


Osaka Univ.


Cracker


Taneya


MARKT


MARKT


BOUL’MICH

Challenges

Easy system integration

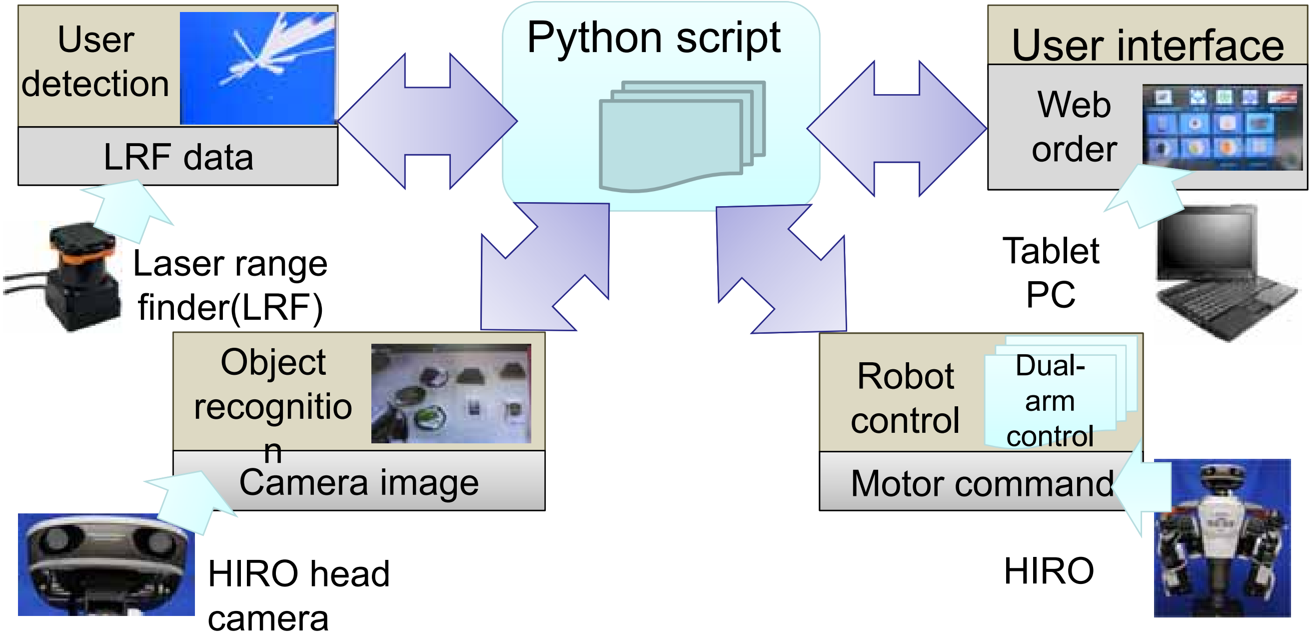
Standardized interface of RT modules

Easy replace of RT modules

Using open source RT modules

Demonstration at IREX 2011

Setup of RT modules



User detection

LRF data

Laser range finder(LRF)

Python script

User interface

Web order

Tablet PC

Object recognition

Camera image

HIRO head camera

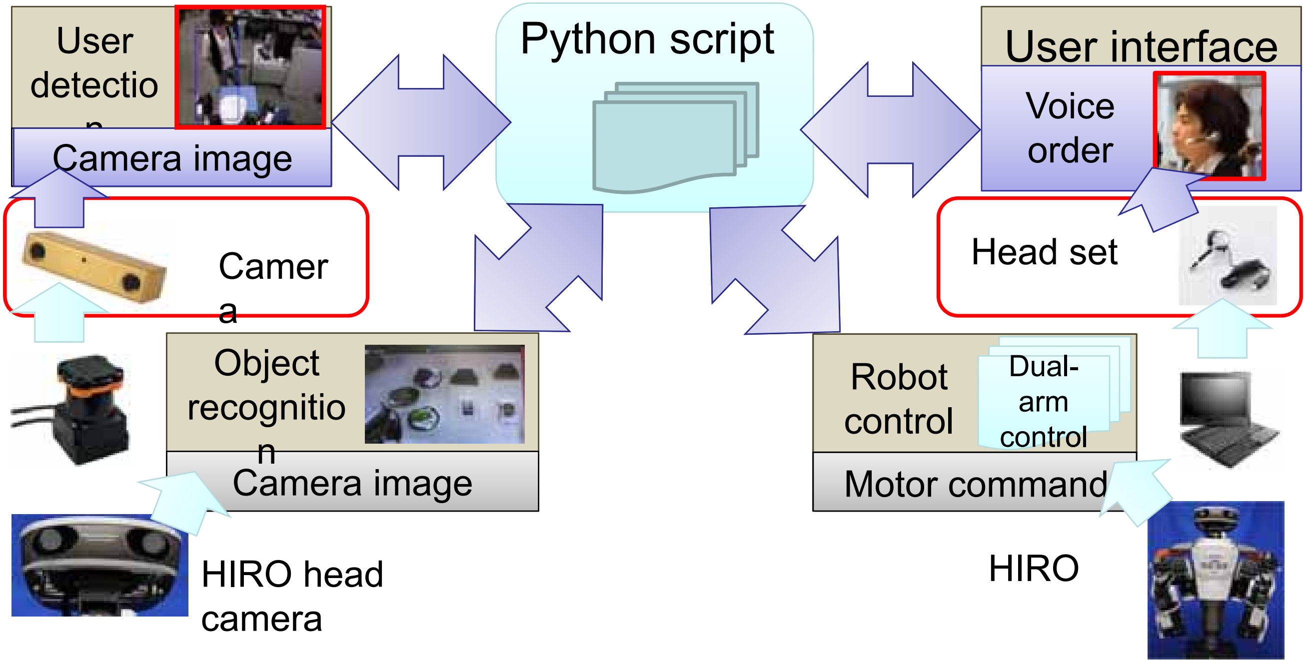
Robot control

Dual-arm control

Motor command

HIRO

Setup of RT modules: Changed hardware



User detection

Camera image

Camera

Python script

User interface

Voice order

Head set

Object recognition

Camera image

HIRO head camera

Robot control

Dual-arm control

Motor command

HIRO

System updating

Easy to replace to a new module which has new algorism

Object recognition

Template matching position/pose estimation

Appearance based position/pose estimation

Template matching

SIFT matching

Assort task by dual eye&arm robot（Tsukuba Univ.）

Towards Practical RT Module Application

2nd Step

- To offer software platform and RT modules

1）Robossa

2)Inter-operability between RTM and ROS

事業原簿 p.105

1）Intelligent Robot Software Suite

－ROBOSSA －

Intelligent Systems Institute,
AIST

— ROBOSSA —

(Open Source RT Components)

- Organize in three categories:

  Manipulation, Navigation, Communication

  Accumulation of basic software modules for intelligent robots

- Open source intelligent robot software modules

  Collection of enabling modules to select and combine freely

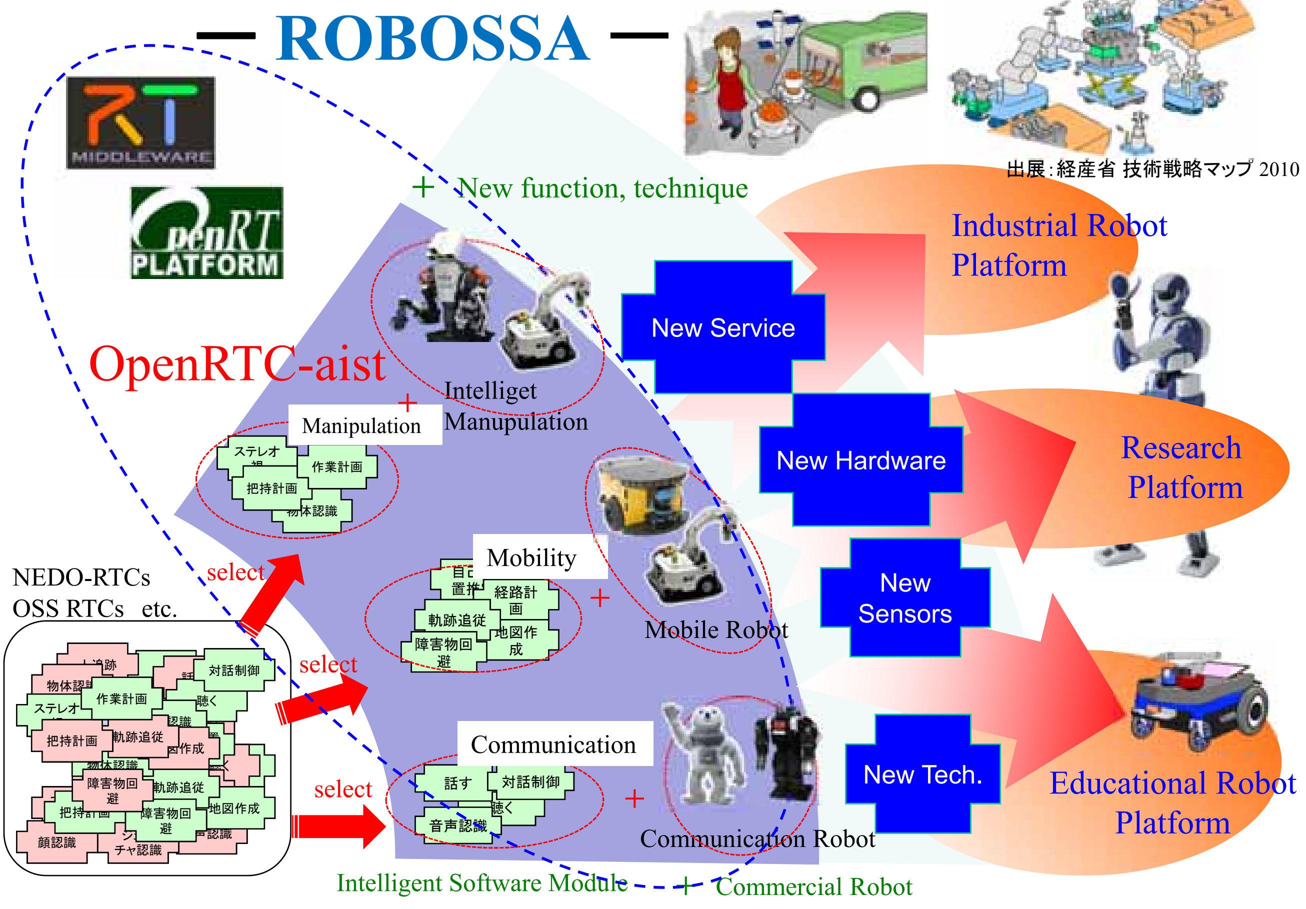
- Commercial robots are supported.

  Collection of modules easy to use on available reliable robot

---

Intelligent Robot Software Suite

— ROBOSSA —

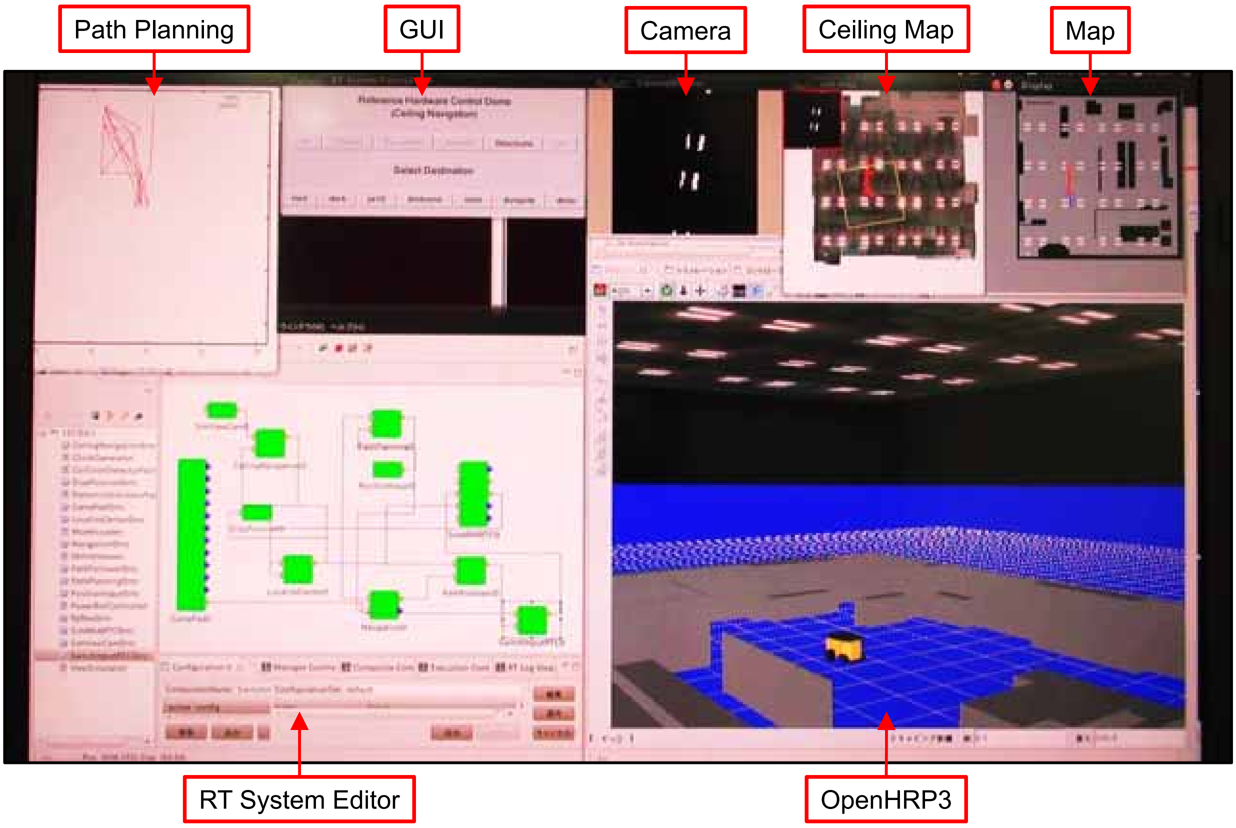


出展:経産省 技術戦略マップ 2010

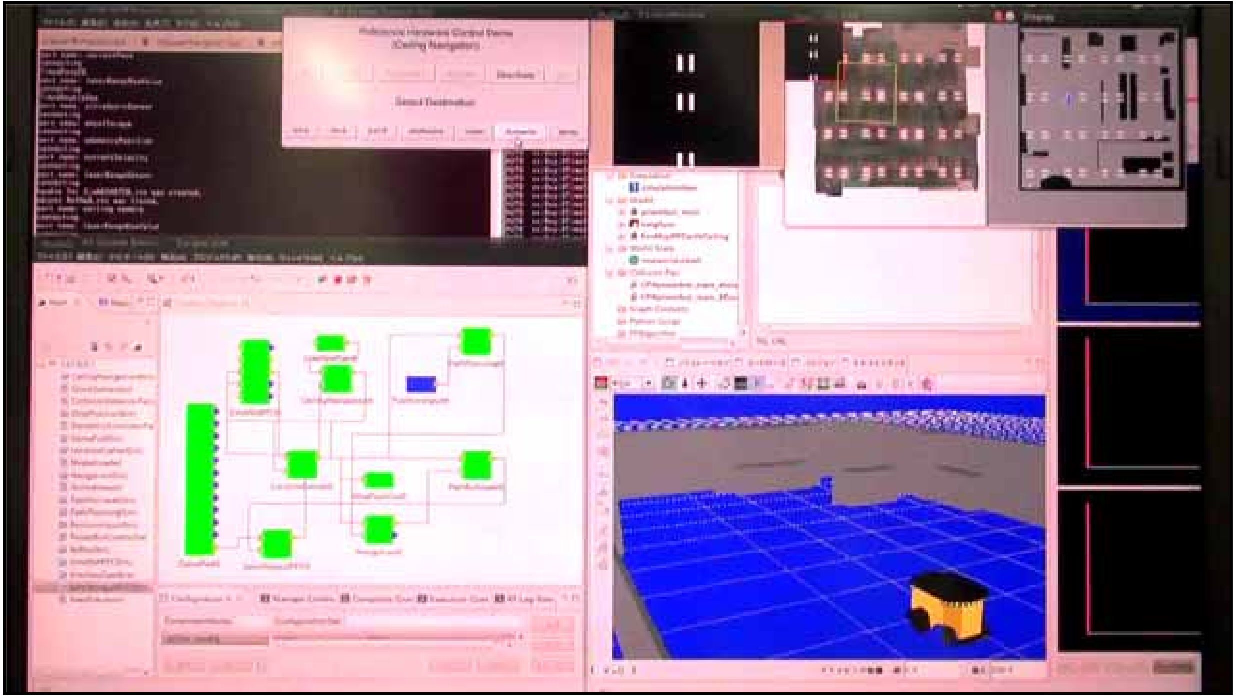
RT-Components for Mobile Robots (OpenNavigation)

- Base on the common interface of mobility intelligence
- Mobility functions for wheeled robot to follow the pathway
- Modules developed in the intelligent robot software project
  - Sensing（2 modules）
  - Self-localization（4 modules）
  - Mobility control（3 modules）
  - Path planning and Path tracking（4 modules）
  - User-Interface（2 modules）

Ceiling　Navigation



Simulation



2）OpenRTM-ROS interopelablity

University of Tokyo

OpenRTM and ROS: Comparison Overview

- OpenRTM is designed on RTM standards and focus on a quality guaranteed component development, specially for the enterprise users.
- ROS is designed for research community and focus on providing development environment.

	OpenRTM	ROS
Sponsers	MITI, MEXT, NEDO	WillowGarage
License	Open / Closed	Open License (BSD)
PI	AIST	Open Source Robotics Foundation
Design Principle	Component Strict framework for re-usability	Library Loose framework for development speed
Quality Control	OMG standard Reusability Center	None (voluntary based control)

OpenRTM and ROS: locations and number of modules

- ROS
  - 114 repositories (including 14 companies = 8%)
  - 150 software modules (number of ROS stacks, number of package is 3000)
  - http://www.ros.org/wiki/Metrics
- OpenRTM
  - 45 repositories (include 15 companies = 33 %)
  - 332 software modules

OpenRTM is widely used in enterprise users

ROS User Map

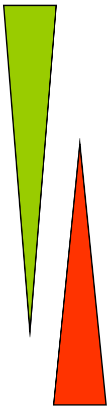
http://maps.google.com/maps/ms?ie=UTF&msa=0&msid=209668390659853657363.00049c608b78bc7779683

OpenRTM User Map

http://maps.google.com/maps/ms?ie=UTF&msa=0&msid=202046448223103795061.0004af11ddd066defcdfb

OpenRMT and ROS integration

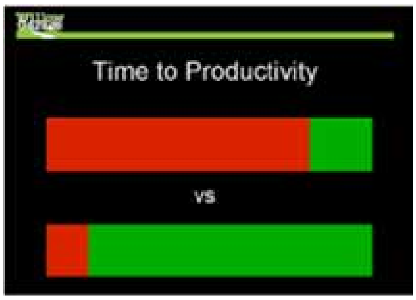
What is ROS exactly?
ROS = plumbing + tools + capabilities + ecosystem
B. Gerkey, Dec 06 ’11. answers.ros.org



- Application
- Modules
- Library
- Simulator
- Communication
- Device Drivers
- Tools

Research
Target of OpenRTM project

Tools
ROS provides extensive set in this layer



Red indicates time to build tools,and greer shows the research. Current PhD student spend most of their time to build tools. ROS is designed to provide efficient tools for researchers to concentrate on the “research”（ Steve Cousins speaking at Robo Development：http://www.willowgarage.com/blog/2008/11/17/steve-cousins-speaking-robo-development-tuesday)

→ Builing OpenRTM-ROS environment on ROS-tools

- Connecting OpenRTM and modles developed in all over the word.
- Efficient development and maintenance

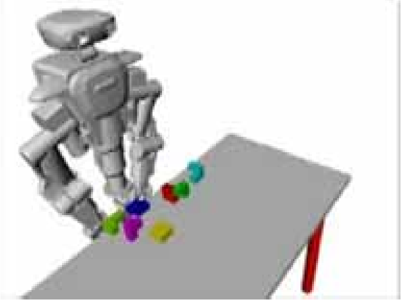
RTM-ROS Interoperability Project

(A) Interoperability platform hardware

1. OpenRTM Mobile base
   + ROS Navigation
   - Support common interface designed in navigation SW group
2. OpenRTM Mbile Base
   + ROS Navigatoin
   + OpenRTM manipulator
   + OpenRAVE Planning
   - Common interface is designed in manipulation group
   - Using joint angle interface of SequencePlayer


Mobaile robot beego

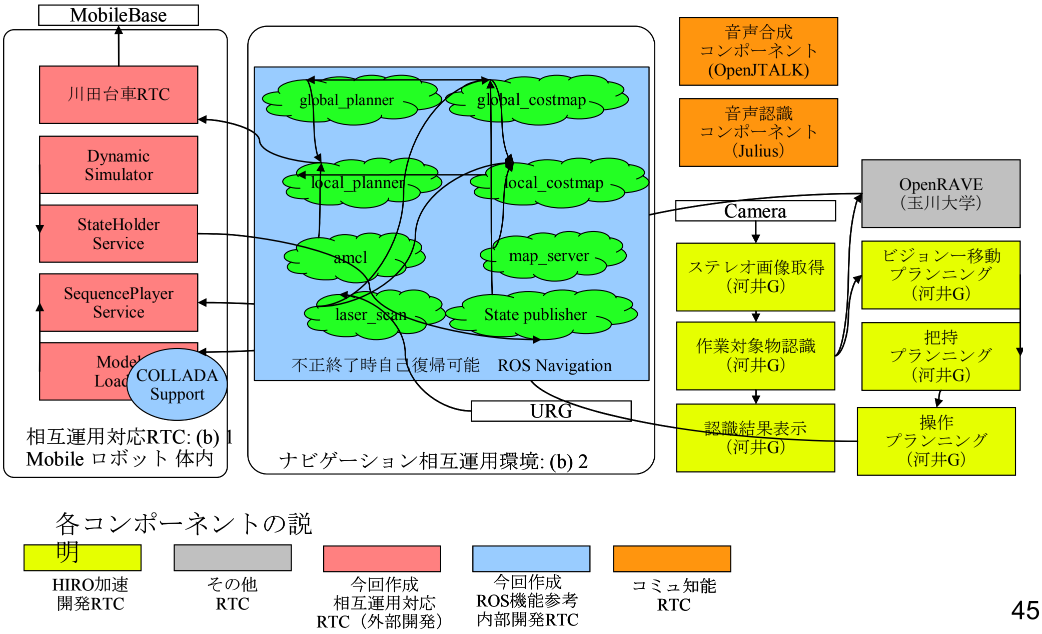

Yasukawa’s mobile unit FMK


3D block manipulation using OpenRAVE

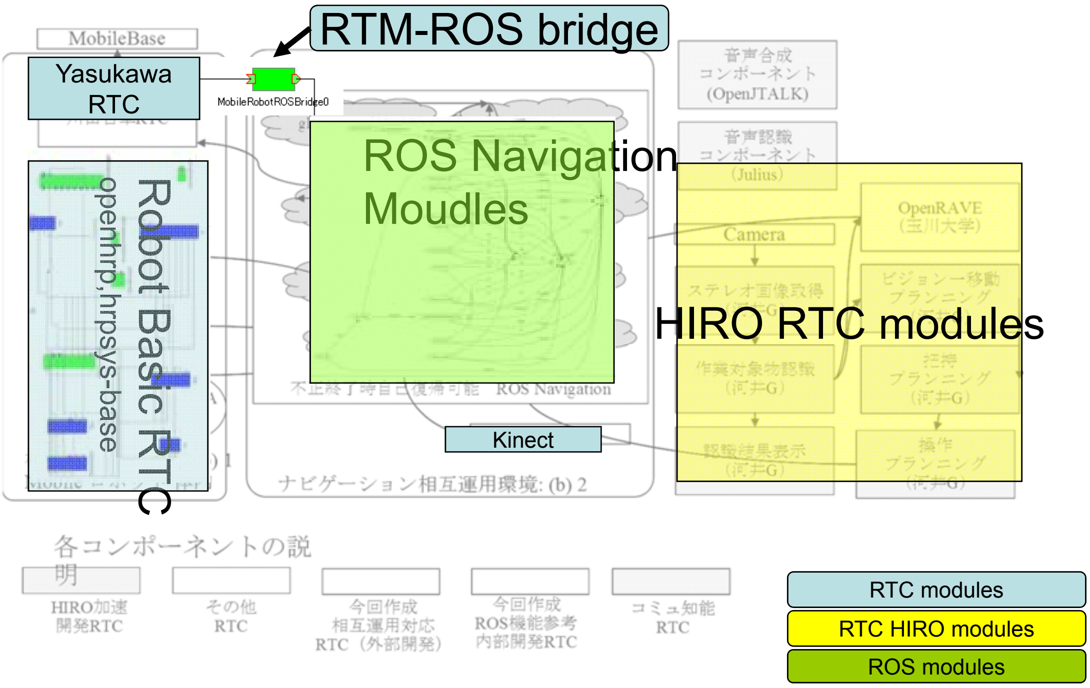

Mobile manipulation robot

RTM-ROS Interoperability Project

(B) Interoperability platform software design



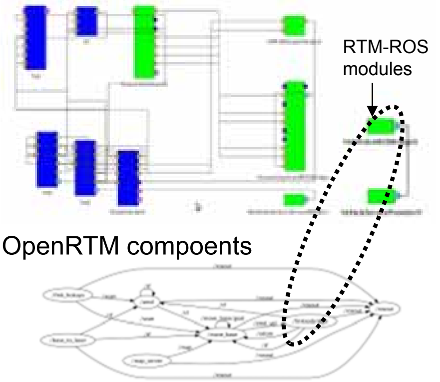
(B) Interoperability platform software design



Experiments in Interoperability platform

ROS Navigation module on OpenRTM mobile robot base

Mobile Manipulation Robot



ROS modules

Mobile Manipulation robot using OpenRTM Controller and ROS Interface

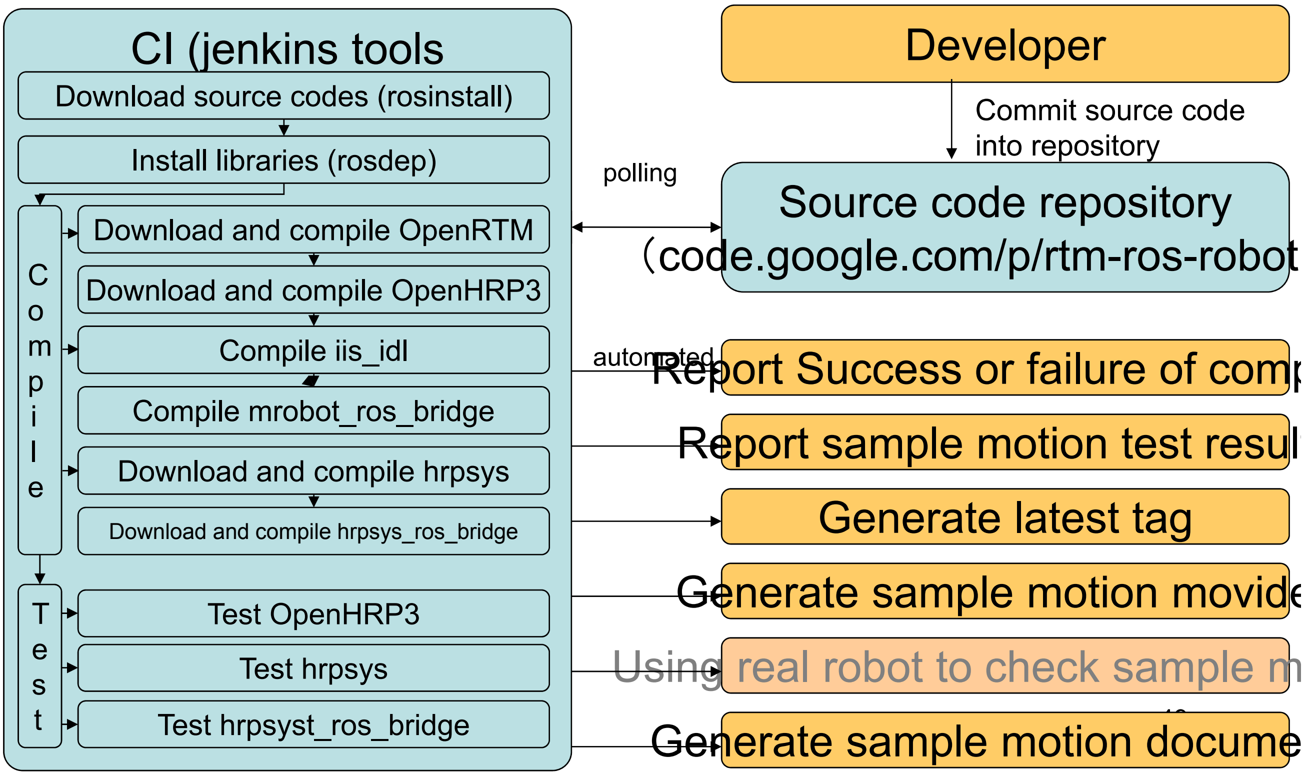
carry tray task experiment

RTM-ROS Interoperability Project

(C) Continues development of Intelligent RT Component

- Need framework that enable us to keep interoperability between OpenRTM and ROS even after the projects terminated
- Each software will continuously be improved respectably, interoperability tools need to adapt to such progresses
  - Automatic testing of intelligent components
  - Automatic tools to generate RTC component from ROS nodes

(X2) Automatic testing and documentation of RT components



CI (jenkins tools

Download source codes (rosinstall)

Install libraries (rosdep)

Compile

Download and compile OpenRTM

Download and compile OpenHRP3

Compile iis_idl

Compile mrobot_ros_bridge

Download and compile hrpsys

Download and compile hrpsys_ros_bridge

Test

Test OpenHRP3

Test hrpsys

Test hrpsyst_ros_bridge

Developer

Commit source code into repository

polling

Source code repository
（code.google.com/p/rtm-ros-robot

automated

Report Success or failure of comp

Report sample motion test resul

Generate latest tag

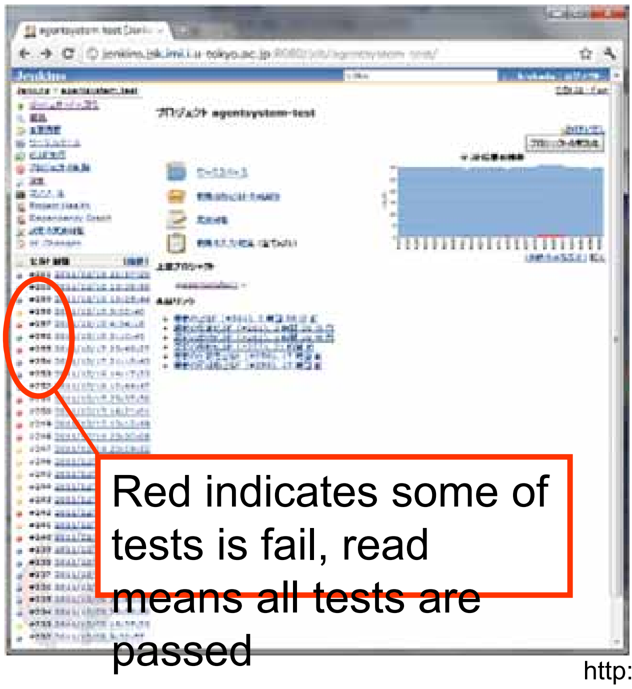
Generate sample motion movide

Using real robot to check sample m

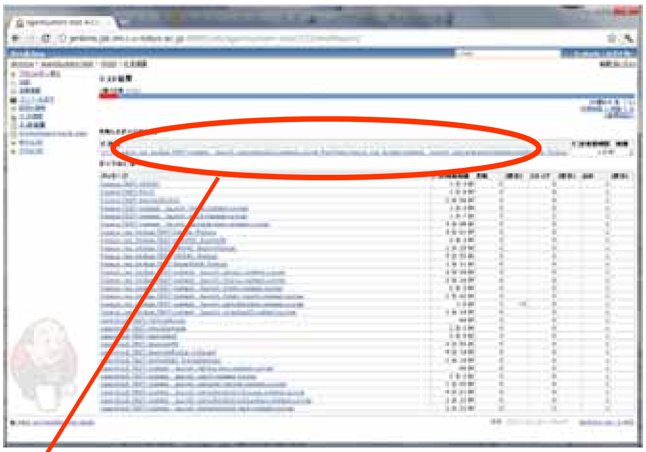
Generate sample motion docume

Test result of intelligent component(2)

- Report of sample testing code



Red indicates some of tests is fail, read means all tests are passed



Web interface also provides the link to the failed test

http://jenkins.jsk.imi.i.u-tokyo.ac.jp:8080/job/agentsystem-test/

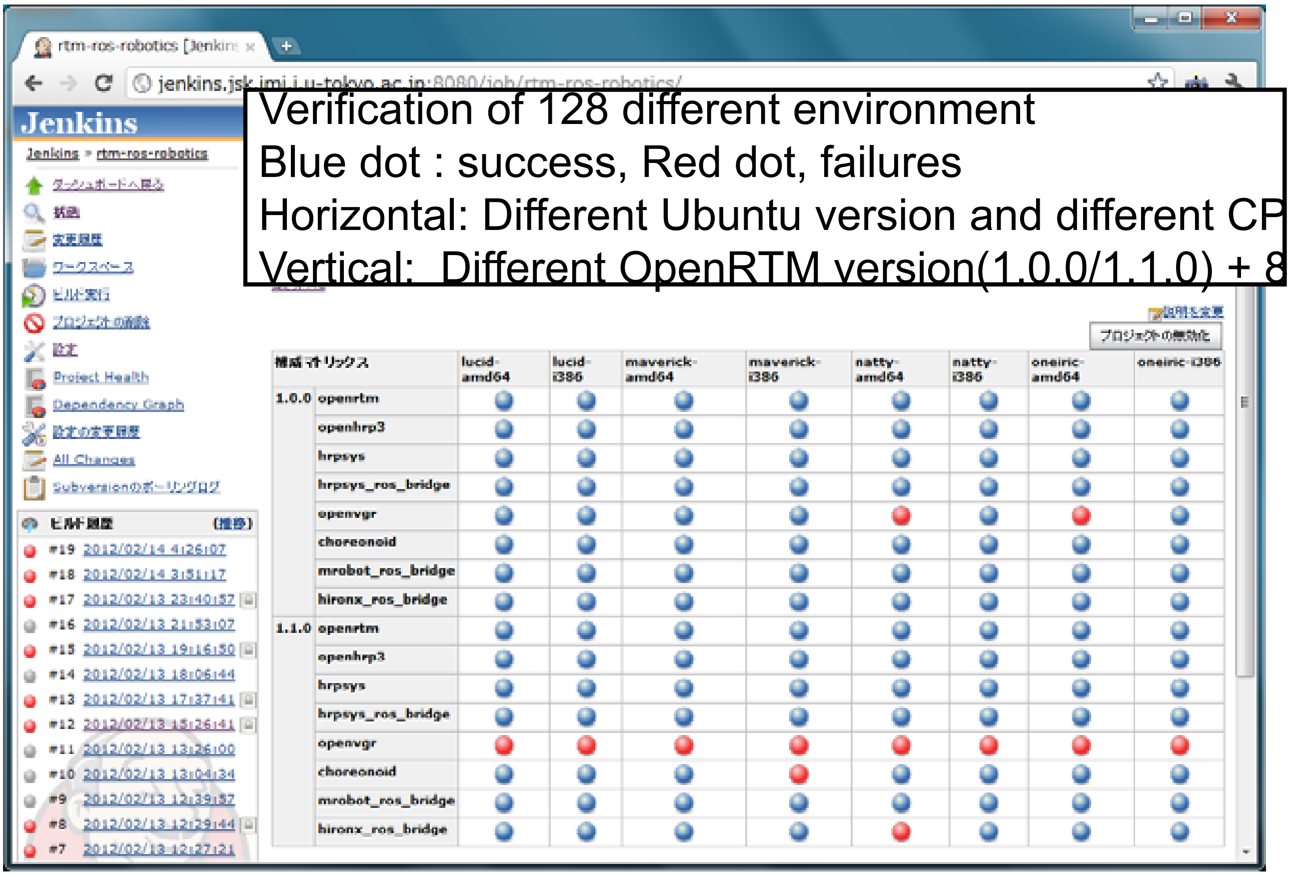
Verify on different version of CPU, OS Middleware

- Verify OS and middleware updates
- Combination of ordinal environment
  - CPU：32bit (i386), 64bit (amd64)
  - OS：Ubuntu 10.04, 10.10, 11.04, 11.10
  - OpenRTM: 1.0.0, 1.1.0
- Test each component for each of above 16 combinations
- Right figure shows verification for 8 components. As a total, we executes 128 verifications

horizontal：8 combinations
(CPU x OS)

vertical： 8 components for
two different
OpenRTM versions

http://jenkins.jsk.imi.i.u-tokyo.ac.jp:8080/job/rtm-ros-robotics/



Verification of 128 different environment
Blue dot : success, Red dot, failures
Horizontal: Different Ubuntu version and different CP
Vertical: Different OpenRTM version(1.0.0/1.1.0) + 8

ビルド履歴 (推移)

- 🔴 #19 2012/02/14 4:26:07
- 🔴 #18 2012/02/14 3:51:17
- 🔴 #17 2012/02/13 23:40:57
- ⚪ #16 2012/02/13 21:53:07
- 🔴 #15 2012/02/13 19:16:50
- ⚪ #14 2012/02/13 18:06:44
- 🔴 #13 2012/02/13 17:37:41
- 🔴 #12 2012/02/13 15:26:41
- ⚪ #11 2012/02/13 13:26:00
- ⚪ #10 2012/02/13 13:04:34
- ⚪ #9 2012/02/13 12:39:57
- 🔴 #8 2012/02/13 12:29:44
- 🔴 #7 2012/02/13 12:27:21

構成マトリックス		lucid-amd64	lucid-i386	maverick-amd64	maverick-i386	natty-amd64	natty-i386	oneiric-amd64	oneiric-i386
1.0.0	openrtm	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
	openhrp3	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
	hrpsys	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
	hrpsys_ros_bridge	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
	openvgr	🔵	🔵	🔵	🔵	🔴	🔵	🔴	🔵
	choreonoid	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
	mrobot_ros_bridge	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
	hironx_ros_bridge	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
1.1.0	openrtm	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
	openhrp3	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
	hrpsys	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
	hrpsys_ros_bridge	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
	openvgr	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴
	choreonoid	🔵	🔵	🔵	🔴	🔵	🔵	🔵	🔵
	mrobot_ros_bridge	🔵	🔵	🔵	🔵	🔵	🔵	🔵	🔵
	hironx_ros_bridge	🔵	🔵	🔵	🔵	🔴	🔵	🔵	🔵

Towards Practical RT Module Application

3rd step

Contribution to National Requirements

- **RTM Safety**

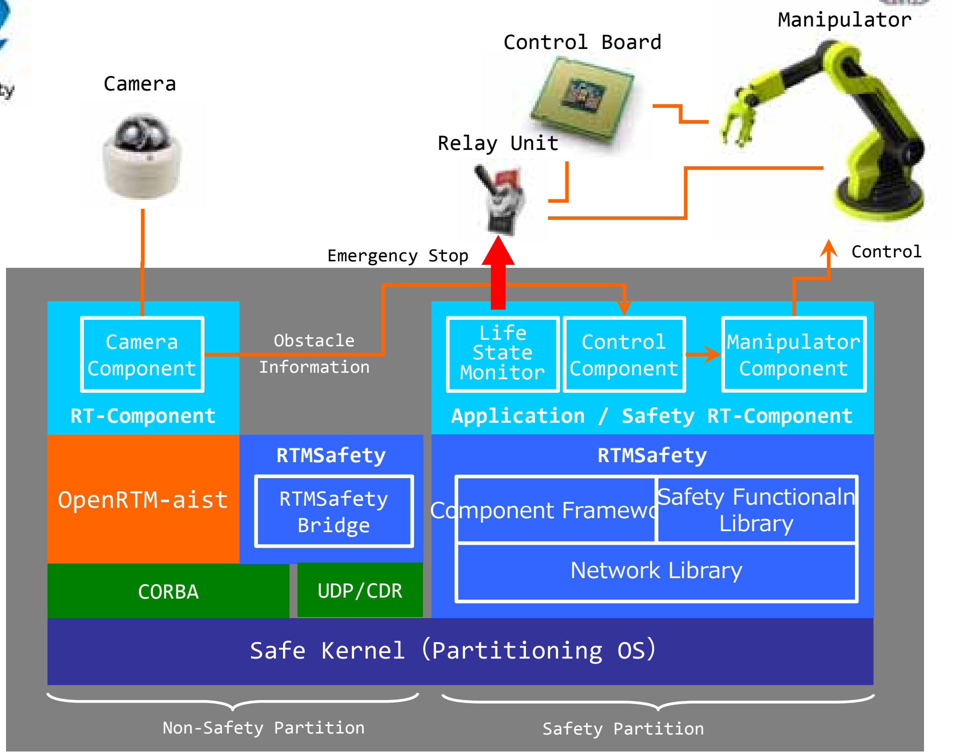
- **Disaster Robot equipped with RT modules**

RTM Safety

SEC cooperation

RTM obtained IEC61508 Functional Safety Standard

- **First in the word R middleware product equipped with Safety concept**
- **Obtained IEC61508 SIL3 Capable Certificate**
- Offer framework to adjust the load between Robot Component（RTC）and CPU load
- Equipped with the function of RTC monitor（Safety Function Library）
- Equipped with Light communication protocol following GIOP / CDR、cope with various types of network protocol（Network Protocol）
- Equipped with cooperation function with OpenRTM-aist（RTM Safety Bridge）



Nuclear Power Plant Robot equipped with RT modules

International Rescue Robot Center, Kyoto University, Tohoku UniversitySegway Japan）

※1：上記ロボットはすべてRTミドルウェア 上で動作
※2：NEDO「戦略的先端ロボット要素技術開発プロジェクト」で開発された成果の一部

- purpose：Verification of speck of Nuclear Power Plant robot, Disaster Robot
- Operator：　Kyoto Uni.、Int. Resque Center、Okayama Univ.、Nagoya Tech. Uni.
- Technical Advisor：Tohoku Uni.、Segway Japan、Tokyo Elec. Com. Univ.
- Cooperated by Tyugoku Electric Power Co.、NEC、Sick Co.
- Date：2012 Mar
- Place：　Shikoku Electric Power Com.Shimane Nuclear Power Plant
- Experiments：(1) Performance evaluation in real site
- (2) Sharing common experience with robot user
  (3) Evaluation of RT software modules
- **<u>Merits of utilization of RTM</u>：**
  - Exchangeability of hardware as well as software
    for exploratory realization of robot in such environment
    where target task cannot be clearly fixed in advanced.
    →Easy prototyping
  - Shortening of developing time ：Only 3～4 months
- Realized robot：“MATOI”（Kyoto Uni.）、“KOHGA3”（Kyoto Uni.）

Concluding Remarks

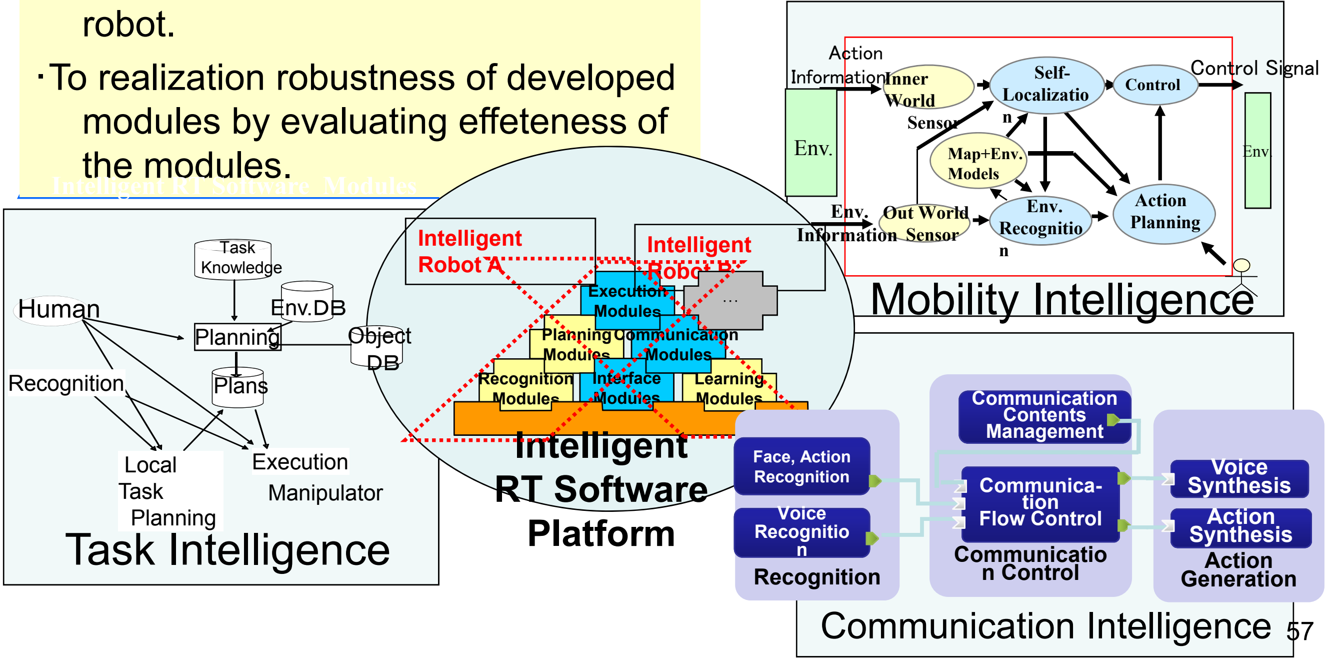
"Intelligent RT Software Project" Overview

Research Target

- To realize a software platform on which intelligent RT software can be integrated.
- To accumulate intelligent RT software modules to construct an intelligent robot.
- To realization robustness of developed modules by evaluating effeteness of the modules.

Period and Budget

Period
2007-2011 (5 year project)
Funding 67M$(Total)
19M$(2007), 15M$(2008),
13M$(2009),11M$(2010),10M$(2011)



Towards Practical Utilization of RT modules ～Steps and Efforts～

1st Step

・Accumulation of practical RT modules

2nd Step

・To offer software platform and RT modules

3rd step

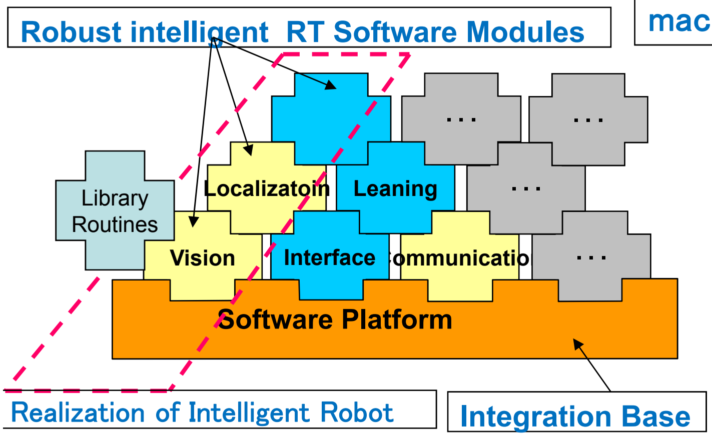
・Contribution to National Requirements

Intelligent RT Software Project

～Next Generation Robot Projects sponsored by NEDO: Ministry of Economy, Trade and Industry～

Project Targets

- ●Realization of “software platform” on which intelligent RT software modules can be integrated.
- ●Accumulation of “robust intelligent software modules” are required to realize an intelligent robot.



Robust intelligent RT Software Modules

Realization of Intelligent Robot

Integration Base

Practical Robot＝Individually functioning machine realized by social co-creation.

Robot ＋ RT Services
＋Social Implementation mechanism

Rapid prototyping and rapid feedback are essential for the robot to be implemented in the society. Therefore, intelligent RT software modules offer basis to realize cost-effective robot with useful functions.

OMG Robotics DTF 2012

Implementation of OPRoS to a human-friendly guide robot, FURO

Ph.D. Se-Kyong Song
CEO of FUTURE ROBOT

Dec.11, 2012

Contents

1. FUTURE ROBOT®
2. CEO Song
3. Service Robot
4. OPRoS™ OPRoS(Open Platform for Robotic Services)
5. FURO™ FURO (Future Robot)
6. Future

1. FUTURE ROBOT

FUTURE ROBOT is making a '*future robot*'.

Established Venture Company in August, 2009

Located in Pangyo Techno-Valley (Silicon Valley of Korea)

Robot Design, Manufacturing & Robot Solution Provider

Service Robot Global Leader in Future

FUTURE ROBOT is in the spotlight with '*FURO*'.

CeBit 2012 in German

Hong Kong Electronics Fair 2012

Global Leader of Smart Service Robot

www.futurerobot.com

FUTURE ROBOT®

FUTURE ROBOT is in the spotlight with '*FURO*'.

Roadshow in Singapore

KOMAF Fair 2012 in China

Global Leader of Smart Service Robot

www.futurerobot.com

FUTURE ROBOT®

FUTURE ROBOT is a service robot leader.

FUTURE ROBOT has appeared on Worldwide Media with '*FURO*'.

FUTURE ROBOT IN Discovery CHANNEL

2. CEO Song

I am making FUTURE ROBOT.

Ph.D. Se-Kyong Song

Graduated from KAIST, Specialized in Robotics & Control

2002~2008 Worked in Samsung Electronics

2008~2009 Worked in Philips Electronics

2009.8 Established FUTURE ROBOT

2012 Awarded several notable awards from the government

A 'First Mover' to open service robot global market

Future Robot Introduces New Humanoid Kiosk

The Korean company has sold 100 FURO-K robots to Brazil

Global robotics news sites

Future Robot gives 'life' to kiosks

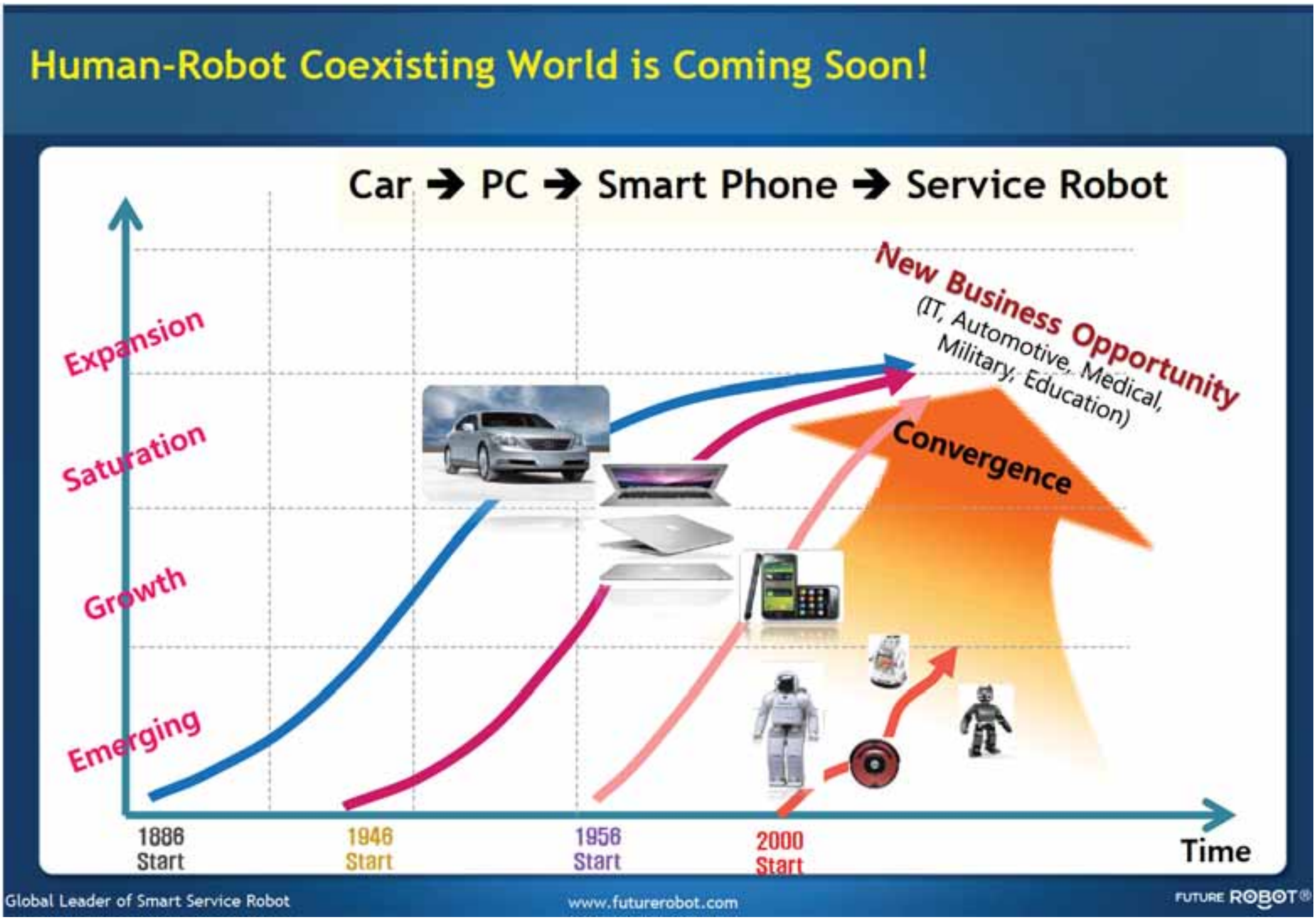
B2B Sites

IFA 2012 TechWatch

FURO on Media

---

3. Service Robot



Human-Robot Coexisting World is Coming Soon!

Car → PC → Smart Phone → Service Robot

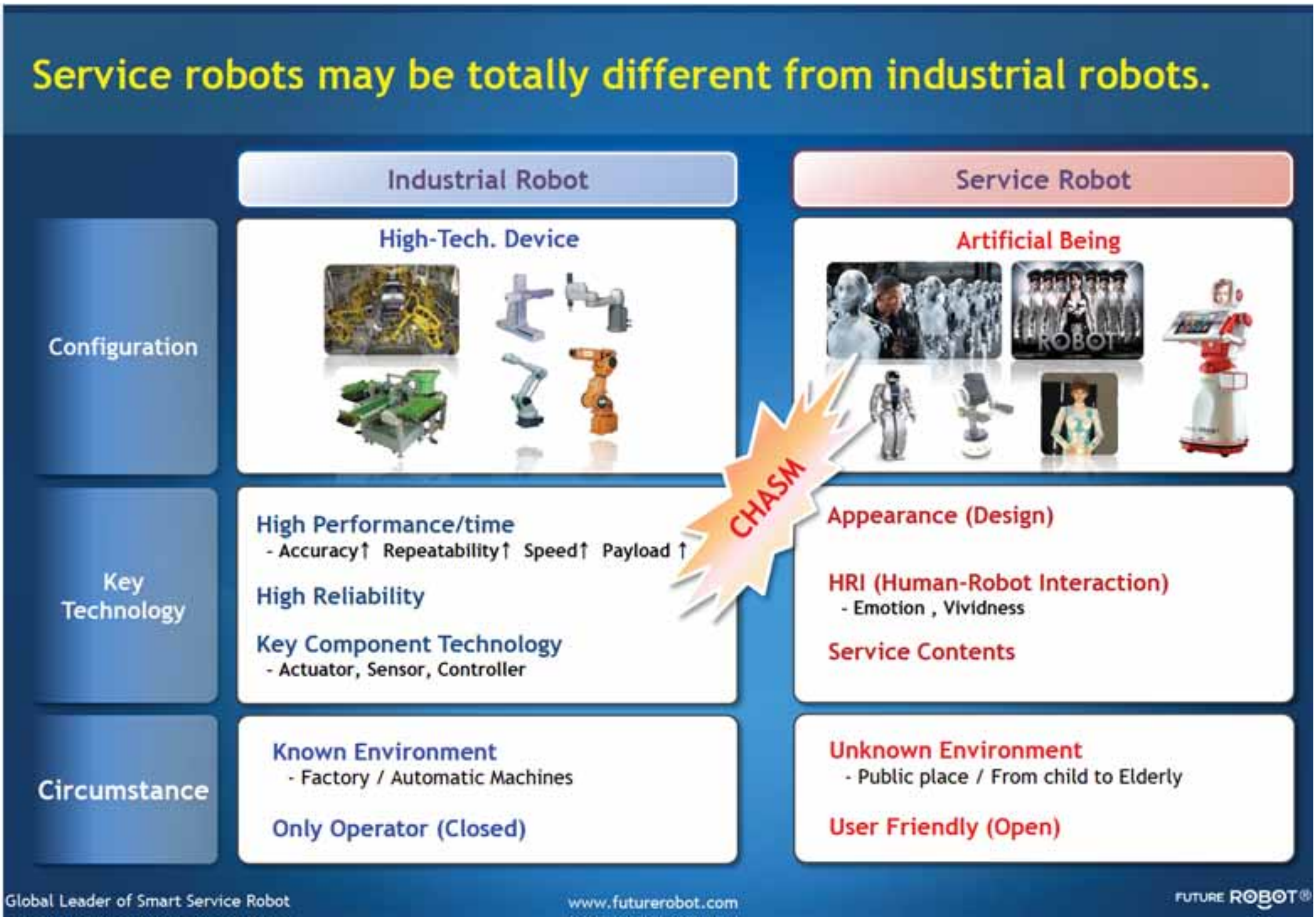
Expansion
Saturation
Growth
Emerging

New Business Opportunity
(IT, Automotive, Medical, Military, Education)

Convergence

1886 Start
1946 Start
1956 Start
2000 Start
Time

Global Leader of Smart Service Robot
www.futurerobot.com
FUTURE ROBOT®

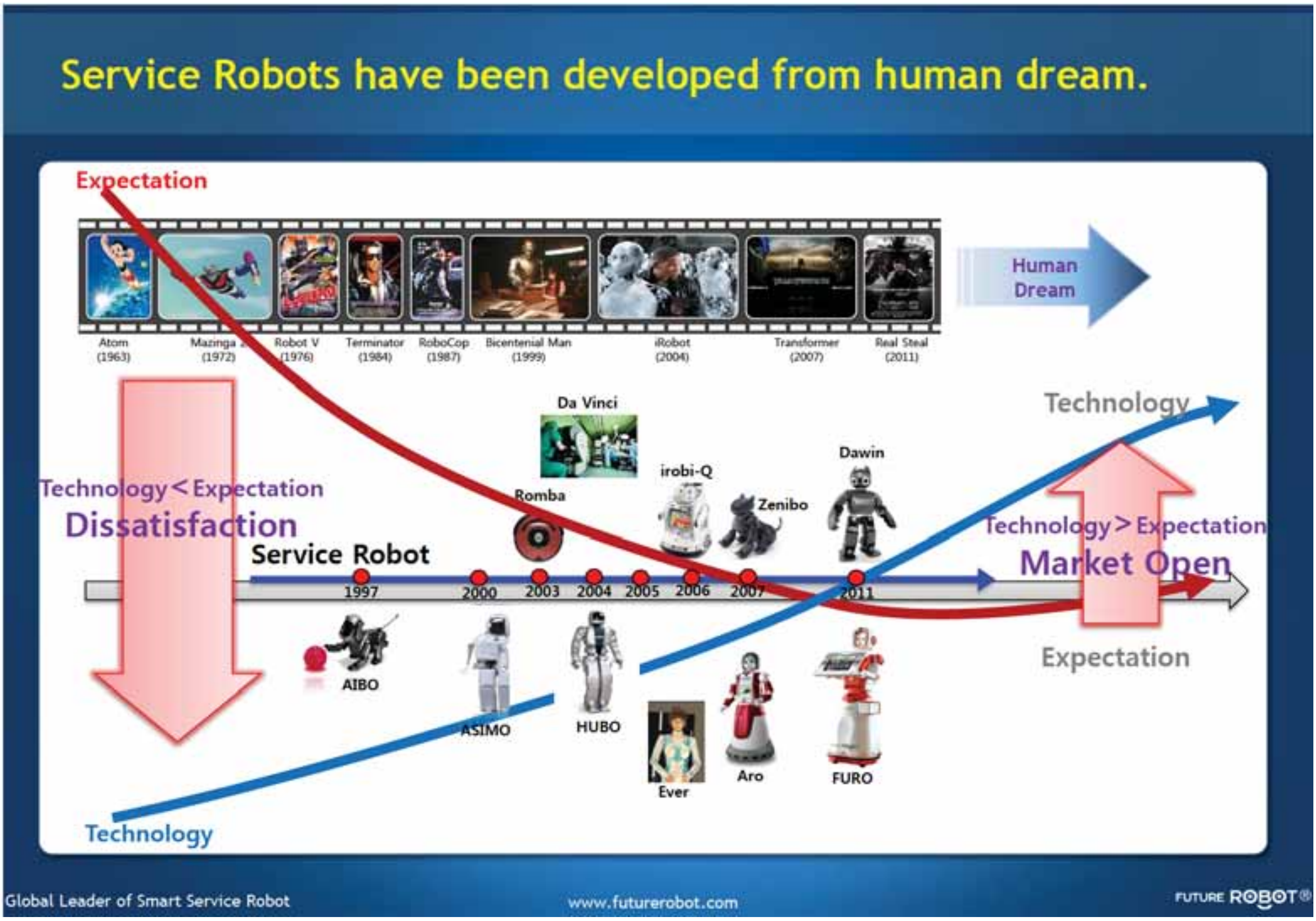


Service robots may be totally different from industrial robots.

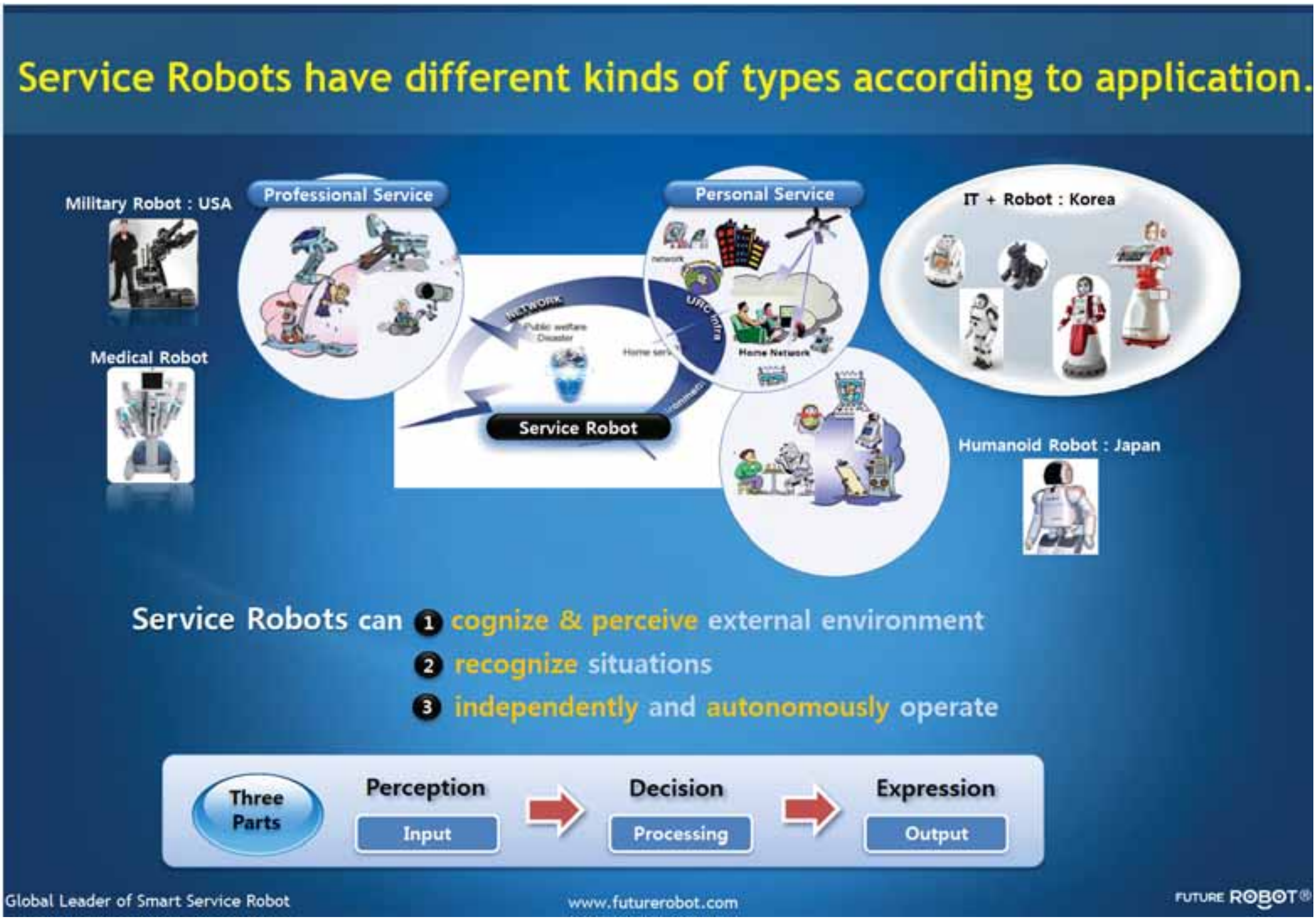
	Industrial Robot	Service Robot
Configuration	High-Tech. Device	Artificial Being
Key Technology	High Performance/time - Accuracy↑ Repeatability↑ Speed↑ Payload ↑ High Reliability Key Component Technology - Actuator, Sensor, Controller	Appearance (Design) HRI (Human-Robot Interaction) - Emotion , Vividness Service Contents
Circumstance	Known Environment - Factory / Automatic Machines Only Operator (Closed)	Unknown Environment - Public place / From child to Elderly User Friendly (Open)

CHASM

Global Leader of Smart Service Robot
www.futurerobot.com
FUTURE ROBOT®



Service Robots have been developed from human dream.



Service Robots have different kinds of types according to application.

Service Robots can ❶ cognize & perceive external environment
❷ recognize situations
❸ independently and autonomously operate

Three Parts

Perception	Decision	Expression
Input	Processing	Output

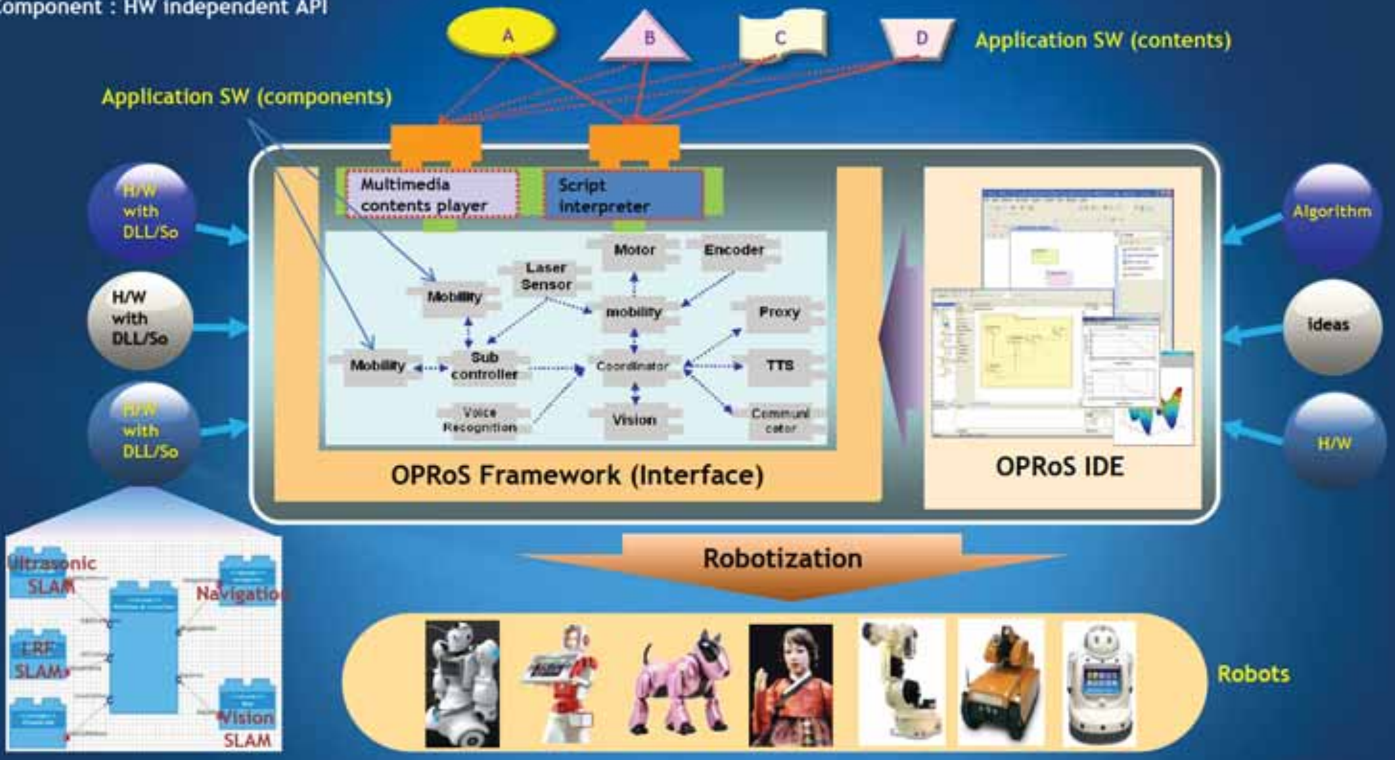
4. OPRoS

OPRoS(Open Platform for RObotic Services)

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

OPRoS is a Open Platform for RObotic Services.

Component : HW independent API



Application SW (contents)

Application SW (components)

Multimedia contents player

Script interpreter

OPRoS Framework (Interface)

OPRoS IDE

Robotization

Robots

OPRoS will provide new services as a integrated M/W.

Realtime scheduling service for the component performance (0.1ms)

Monitoring service for the components & core engine

High performance distributed communication middleware service

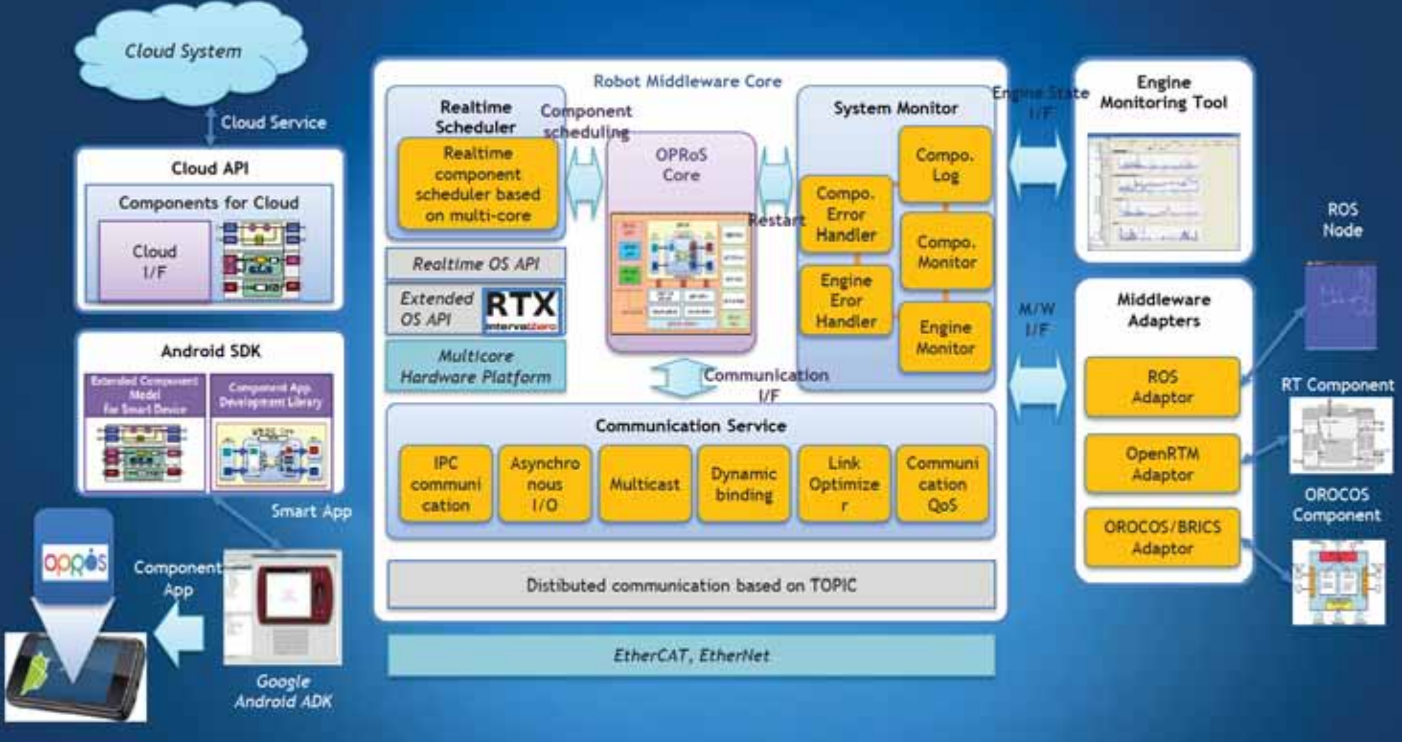
Cloud service components

Support other robot middlewares like ROS, OROCOS

Development toolkits for the Android app. components

ETRI 한국전자통신연구원 KNU 강원대학교 BONAVISION FUTURE ROBOT

OPRoS will be evolved as a global robot middleware.



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

5. FURO

FURO (Future Robot)

FURO is a smart service robot like smart phone.

Service Application

COEX_Shopping Mall

Seoul Nat'l Univ. Hospital

Severance Hospital

ETLAND_Tax Free Shop, Cinema

Service Application

HYUNDAI ASAN Museum

Gwancheon Nat'l Science Museum

Restaurant, SBS TV

Robot Event

Robotics(Technology) shall be intersected with Liberal Arts(Humanism).

Human

Robot

HRI(Human Robot Interaction)

FURO is made as a 'Convergence' product based on 'Liberal Art'.

Friendly Service

Information Device

Service + Information

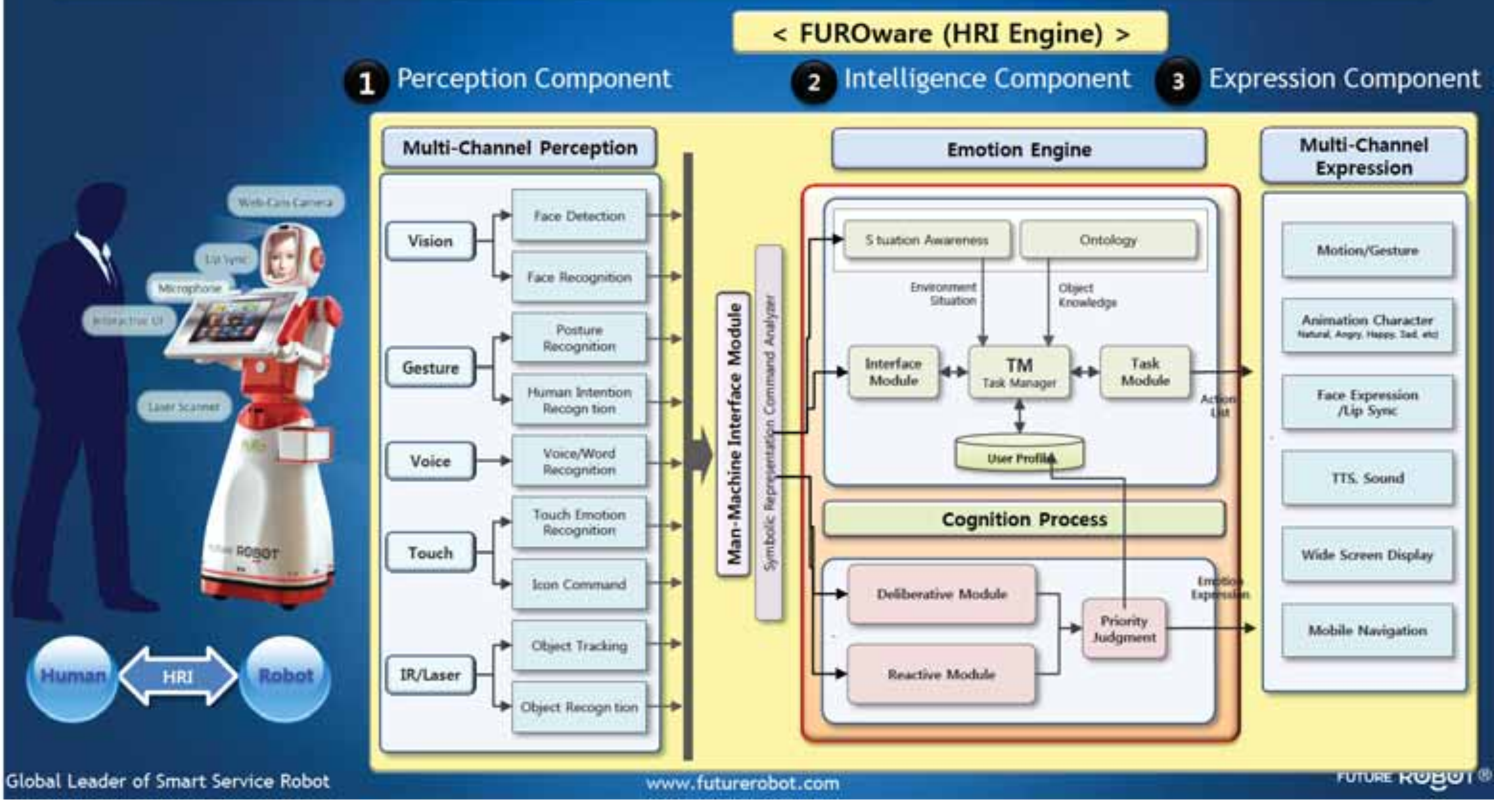
Emotional Avatar

Dynamic Interaction

Emotion + Dynamics

FURO has a Open Platform with OPRoS(Middleware) & FUROware(HRI Engine).

Expresses interactive emotions through multi-channel perception and AI

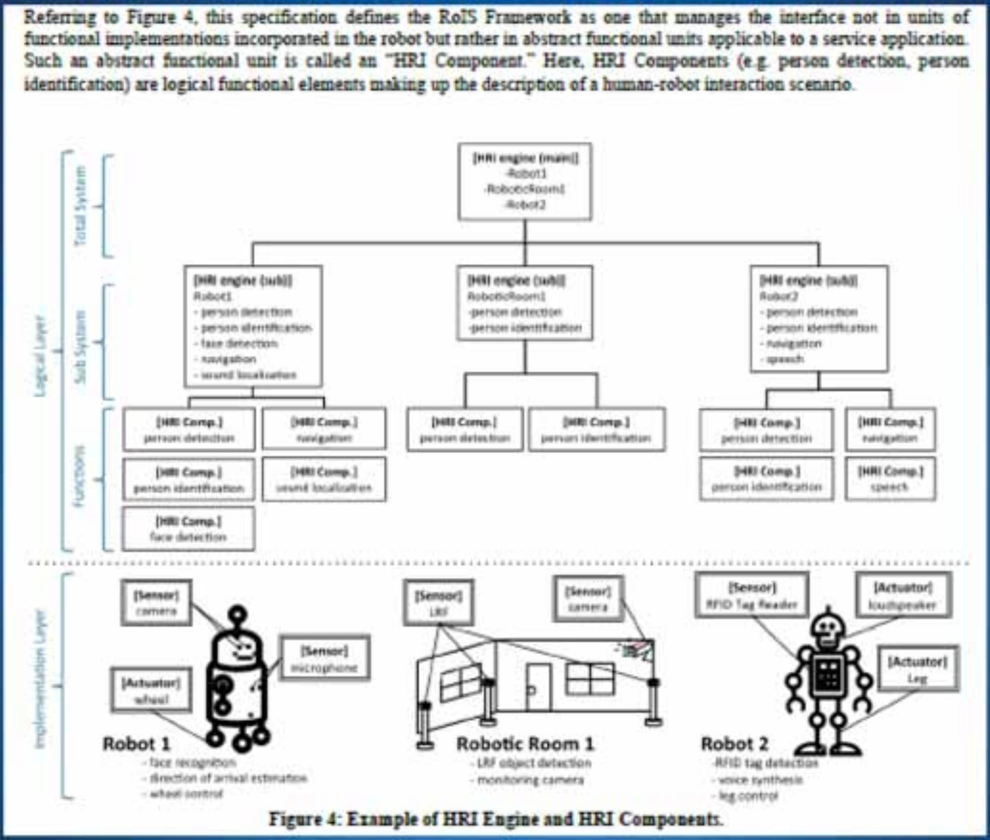


FUROware is Robotic solution made of OPRoS components.

Part	Solution (OPRoS Component)	Basic Module	Advanced Option
Perception & Cognition	FURO-Sense	Face Detection	Face/Sex/Age Recognition, Face Servoing
		Human Intention Detection	Human Intention Identification, Human Motion Servoing
		Keyword Recognition	Voice command
		Touch Sense	Emotion Expression according to Touch Pattern
Artificial Intelligence	FURO-Brain	Emotion Engine	Emotion Controller according to service scenario
		Dialogue Engine	Dialogue Engine according to service scenario and emotion status
		Service Robot Platform	Robot Personalization , Task Manager for multi-service events
Action & Expression	FURO-Expression	TTS (Text-to-Speech)	Natural TTS and Lip Sync
		Sound Effect	Emotional Sound Effect according to service scenario
		Language (Korean)	English, Japan, Chinese, France, German (30 countries)
		Character Emotion	Facial Avatar Emotion
		Emotional Gesture	Head, Monitor, Waist, Mobile, LED
Mobility	FURO-Mobile	Navigation	Map Building, Obstacle Avoidance, Object Chaser
		Charging Station	Automatic Recharging
Robot Management	FURO-Manager	Joystick	Voice Command
		Remote Monitoring	Teleoperation / Scheduling / Statistics
		Scenario Editor	User creative robot service editor
Others	FURO-Pay	Card & RFID Reader, Printing	Human Identification / Credit card reader / Printing service

RoIS S/W Framework is a interface with HRI Components *suitable for variety of robot service applications.*

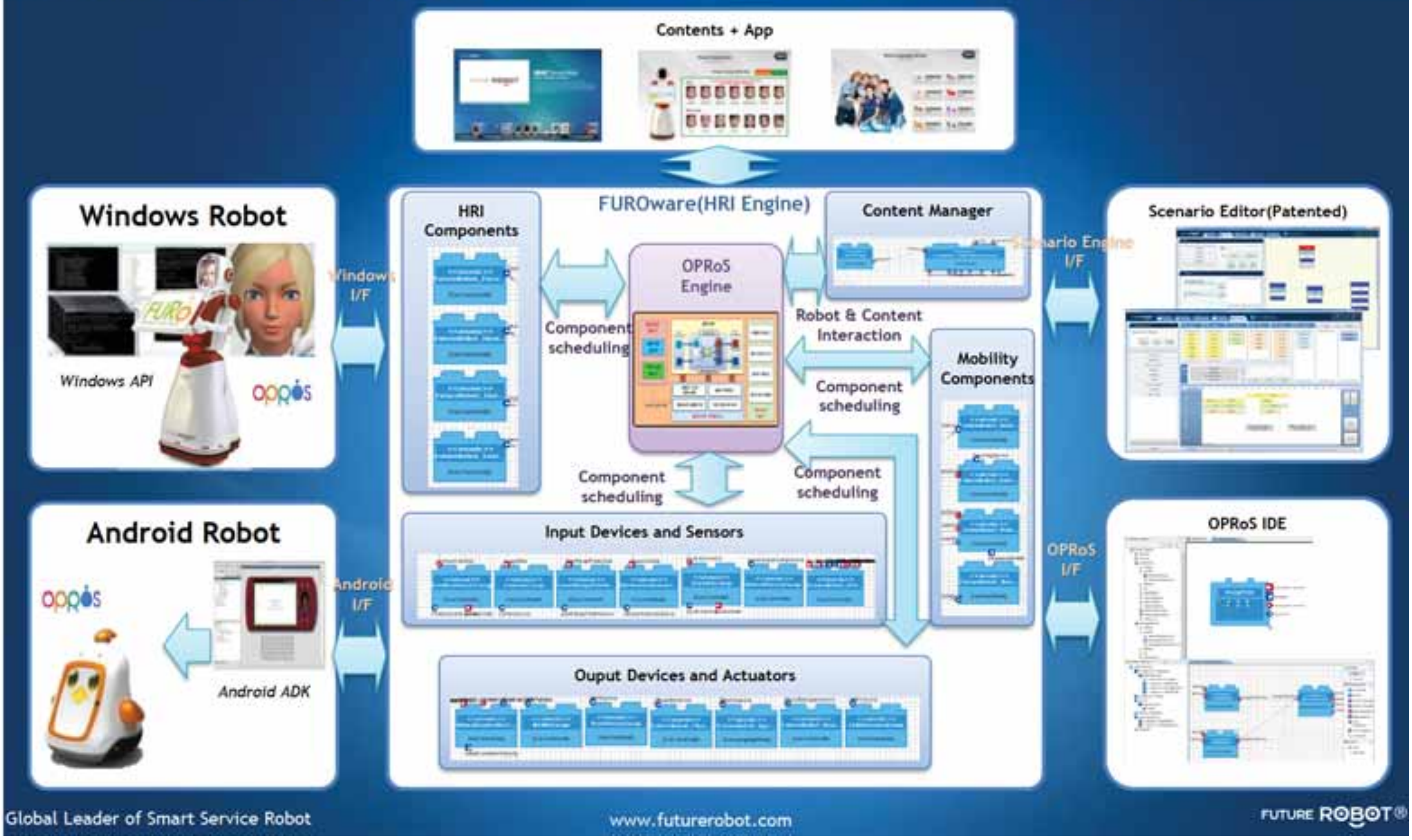
Ref : Robotic Interaction Service (RoIS) Framework



Referring to Figure 4, this specification defines the RoIS Framework as one that manages the interface not in units of functional implementations incorporated in the robot but rather in abstract functional units applicable to a service application. Such an abstract functional unit is called an "HRI Component." Here, HRI Components (e.g. person detection, person identification) are logical functional elements making up the description of a human-robot interaction scenario.

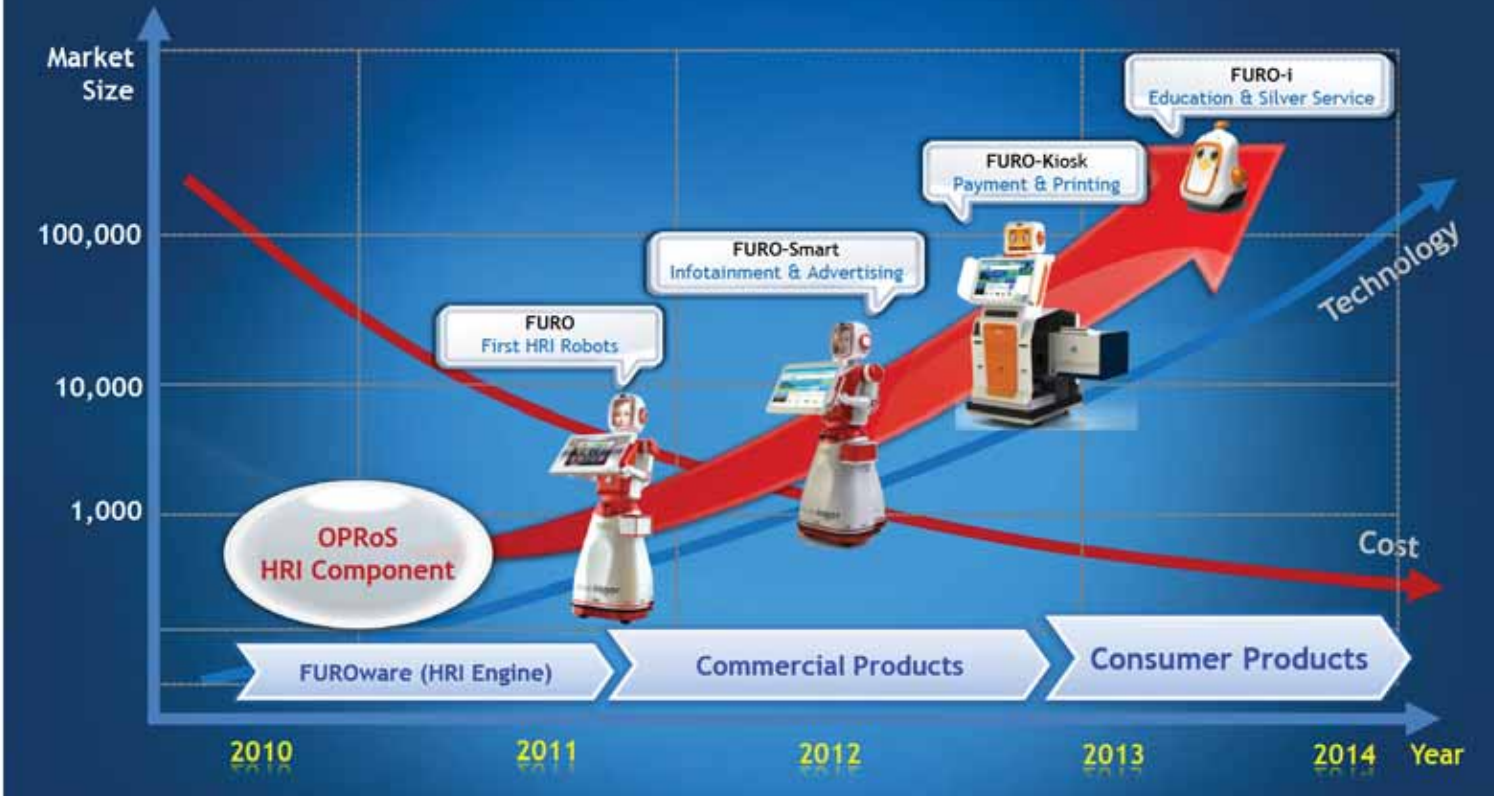
Figure 4: Example of HRI Engine and HRI Components.

FURO S/W Framework is a middleware with OPRoS + FUROware (HRI Engine) *suitable for variety of robot service applications.*



6. Future

FUTURE ROBOT will be more sustainable with '*FURO (Product)*' *and 'OPRoS (HRI Component)'.*



'FURO-i' will live together with humans.

FURO-i Services

Service 1 Heahth Care Service

- Checking health with U-Health Device
- Health treatment information / Emergency call

Service 2 Mental Care Service

- Pet Design / Avatar / Lip-sync / Interaction
- Delivering emotion with variety of facial expression
- Conversation Service - Caring for lonely

Service 3 Entertainment Service

- Family SNS Service
- Music, Movie, etc / Weather news

Service 4 Education Service

- Interesting smart education service with interaction between robot and kid

FURO-i System

OS : Android 3.x or Window 7

* Battery life - 3 hour(s) / Charging time - 1.5 hour(s)

Intelligent Avatar

Expression of various emotions / Lip sync expressions

Personal Robot : FURO-i

u-Health Care Service

FURO-i (Android Tap 10.1") *(Patented)*

Service Robots shall be contributed to Worldwide Risks as well as to Humankind Wellbeing.

Aging, Welling-being, Safety

Society

Energy

Economy

Solo, Manpower Shortage,
New Emerging Countries

Climatic Change, Declining Resources

* Ref. : Robot Future Strategy(2013~2022) / Korea Government / 2012/10.17

Global Leader of Smart Service Robot

www.futurerobot.com

FUTURE ROBOT®

FUTURE ROBOT make best effort to build 'better ecosystem'.

Let's build a 'cooperative ecosystem' for human and world.

Open mind ~ Open robot platform

Contact to us always be welcome !

4 Human Ecosystem

3 National Ecosystem

2 Industrial Ecosystem

1 Technological Ecosystem

Global Leader of Smart Service Robot

www.futurerobot.com

FUTURE ROBOT®

Thank You

International Standardization Activities for Robots

2012. 12. 11

Seungbin Moon
Convener of ISO TC 184/SC 2/WG 8(Service robots)
Dept of Computer Engineering
Sejong University, Seoul, Korea

International Organizations For Standardization on Robots

◆ **ISO : International Organization for Standardization**
- **Membership:157 Nations**
- **www.iso.org**

◆ **IEC : International Electrotechnical Commission**
- **Membership: 68 Nations**
- **www.iec.ch**

◆ **OMG : Object Management Group**
- **robotics.omg.org**

◆ **ASTM :American Society for Testing and Materials**
- **http://www.astm.org/**

◆ **IEEE : Institute of Electrical-Electronic Engineers**
- **www.ieee.org**

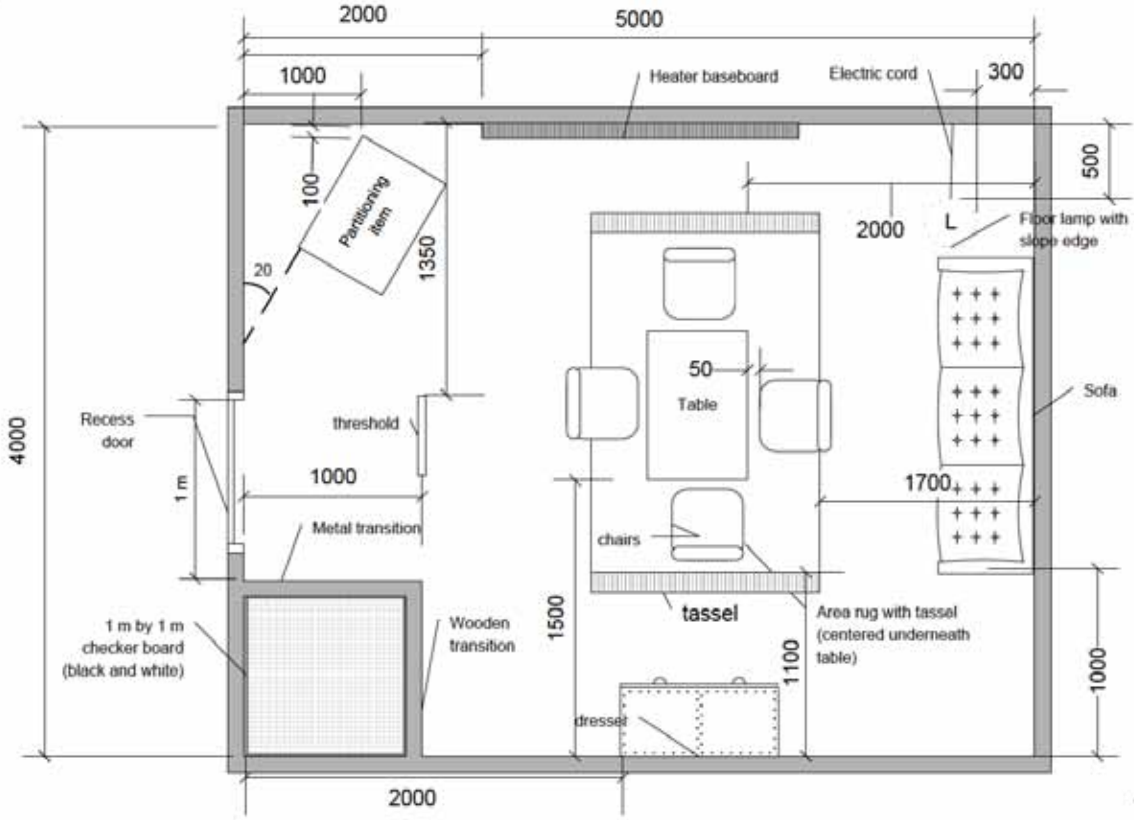
IEC Activities

- IEC SC 59F (Floor treatment appliances) established WG 5 (Methods of measuring the performance of household cleaning robots)
  - IEC 60312-3 Ed. 1.0/NP 2009: Methods of measuring the performance of household cleaning robots
- IEC TC 61(Safety of household and similar electrical appliances) developed a standard which covers safety aspect on vacuum cleaning robot
  - IEC 60335-2-2 Ed6.0:2009 Household and similar electrical appliances - Safety - Part 2-2: Particular requirements for vacuum cleaners and water-suction cleaning appliances
- IEC TC 116(Safety of hand-held motor-operated electric tools) is developing a standard on safety requirements related to lawn mowing robots
  - IEC 60335-2-107 Ed. 1.0/CDV stage: Household and similar electrical appliances - Safety - Part 2-107: Particular requirements for robotic battery powered electrical lawnmowers

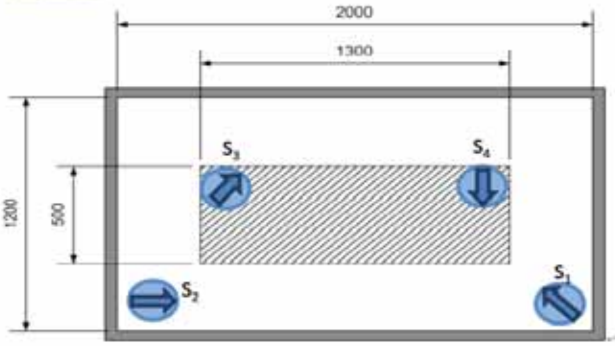
IEC SC 59F/WG 5

- TC 59 (performance of household and similar electrical appliances)
  - SC 59F(Floor treatment appliances)
    - WG 5 (Methods of measuring the performance of household cleaning robots)
    - Convenor: Sungsoo Rhim, Kyung Hee Univ, Korea
    - Started from May, 2009
    - Members: Electolux, iRobot, Philips, Kaercher, Samsung, LG, Yujin Robot, Dyson, TEK, etc.
    - Issues
      - Mobility - coverage rate in a standardized area
      - Dust removal - Dust collection in a standardized area
      - Noise, Battery power, ….

IEC SC 59F/WG 5 – Navigation



Dust removal - Box test



OMG Activities

- **Robotics DTF(Domain Task Force)**
  - To foster the integration of robotics system from modular components through the adoption of OMG standards.
  - Co-chairmen: Tetsuo Kotoku(AIST, Japan), Young-Jo Cho(ETRI, Korea), and Laurent Rioux(Thales)
  - robotics.omg.org
- **Technical Working Groups**
  - Infrastructure WG
    - Robotic Technology Component (RTC) Specification
    - Software modularization and integration at middleware level
  - Robotics Functional Service WG
    - Robotic Localization Service(RLS)
    - robot interaction service (RoIS)
  - Robotic Devices and Data Profile WG
    - Programmer API: Typical device abstract interface and hierarchies
    - Hardware-level resources: Define resource profiles

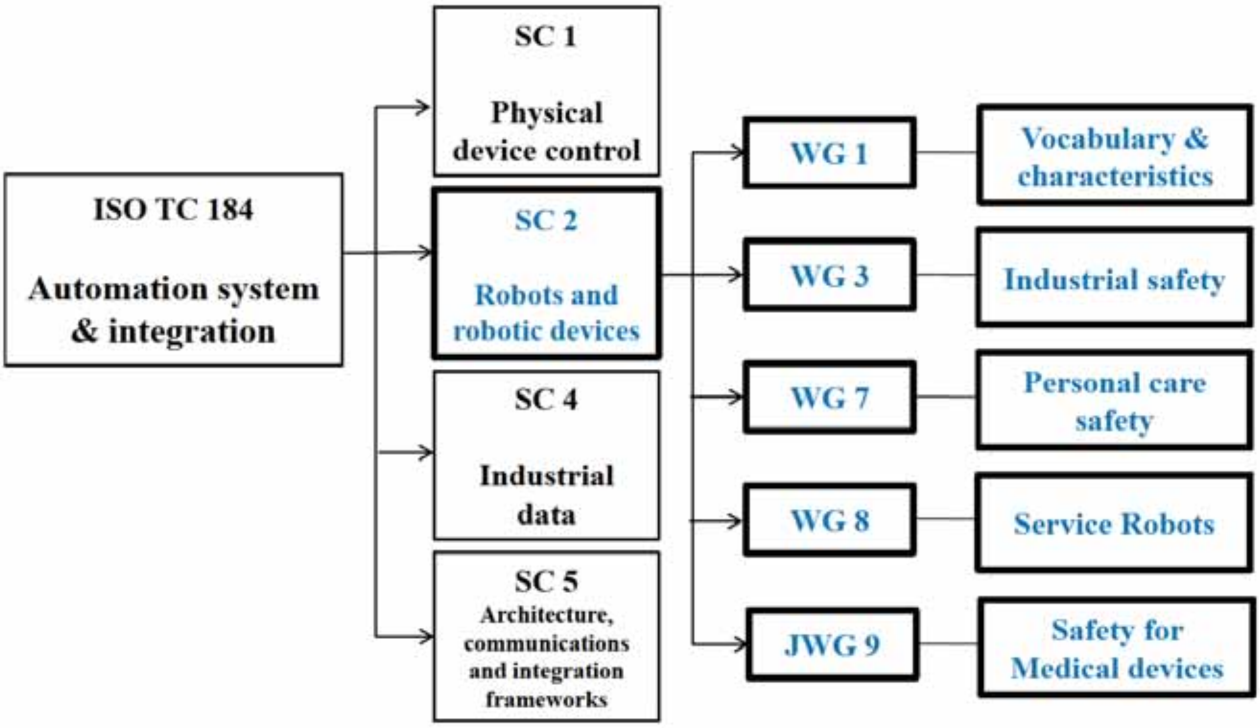
ASTM Activities

- **E54(Homeland Security Applications)**
  - Subcommittee E54.08(Operational Equipment)
  - http://www.astm.org/COMMIT/SUBCOMMIT/E5408.htm
  - E2521-07a: Standard Terminology for Urban Search and Rescue Robotic Operations
  - E2566-08 : Standard Test Method for Determining Visual Acuity and Field of View of On-Board Video Systems for Teleoperation of Robots for Urban Search and Rescue Application.
  - E2592-07: Standard Practice for Evaluating Cache Packaged Weight and Volume of Robots for Urban Search and Rescue
  - Under development for navigation and communication performances, etc.
- **F38 (Unmanned aircraft systems)**
- **F41 (Unmanned maritime vehicles)**

IEEE RAS(Robotics & Automation Society)

- **IEEE RAS formed standard committee**
  - **It is chaired by Raj Madhavan, NIST, USA.**
- **It recently formed 2 Working Groups in Sep, 2011.**
  - **Robot Map Data Representation for Navigation**
    - **Chair: Wonpil Yu, ETRI, Korea**
  - **Ontologies for Robotics and Automation**
    - **Chair: Craig Schlenoff , NIST, USA**

Organization of ISO TC 184/SC 2



ISO TC 184

Automation system & integration

SC 1

Physical device control

SC 2

Robots and robotic devices

SC 4

Industrial data

SC 5

Architecture, communications and integration frameworks

WG 1 — Vocabulary & characteristics

WG 3 — Industrial safety

WG 7 — Personal care safety

WG 8 — Service Robots

JWG 9 — Safety for Medical devices

ISO TC 184/SC 2

- **Background - created in 1983**
  - Chairman: Tomas Angelhag of ABB, Sweden
  - Secretary: Mattias Lafvas of SIS, Sweden
  - 19 P-member countries and 6 O-member countries
  - Published 10 International Standards, 1 Technical Reports
  - 2 are under revision, and 2 new documents are being developed
  - Fields: safety, performance criteria, I/F for mechanics & software
  - www.isotc184sc2.org

- **ISO TC184/SC2 Scope Changes**
  - 1983: "Robots for Manufacturing Environment"
  - 2003: "Robots for Industrial Environment"
  - 2006: "Robots and Robotic Devices"

- **Scope**
  - Standardization in the field of automatically controlled, reprogrammable, manipulating robots and robotic devices, programmable in more than one axis which may be either fixed in place or mobile.
    - Excluded: Toys, military applications.

TC184/SC2 Membership

Participating countries: 19

- Secretariat: Sweden (SIS)
- Bulgaria (BDS)
- Canada (SCC)
- China (SAC)
- Czech Republic (CSNI)
- Denmark (DS)
- France (AFNOR)
- Germany (DIN)
- Hungary (MSZT)
- Italy (UNI)
- Japan (JISC)
- Korea, Republic of (KATS)
- Portugal (IPQ)
- Romania (ASRO)
- Russian Federation (GOST R)
- Spain (AENOR)
- Switzerland (SNV)
- USA (ANSI)
- United Kingdom (BSI)

Observing countries: 6

- Finland (SFS)
- Netherlands (NEN)
- Norway (SN)
- Poland (PKN)
- Serbia (ISS)
- Slovakia (SUTN)

TC184/SC2: International Standards

1. ISO 8373:2012 Robots and robotic devices -Vocabulary
2. ISO 9283:1998 Manipulating industrial robots -Performance criteria and related test methods
3. ISO 9409-1:2004 Manipulating industrial robots -Mechanical interfaces -- Part 1: Plates
4. ISO 9409-2:2002 Manipulating industrial robots -Mechanical interfaces -- Part 2: Shafts
5. ISO 9787:1999 Manipulating industrial robots -Coordinate systems and motion nomenclatures
6. ISO 9946:1999 Manipulating industrial robots -Presentation of characteristics
7. ISO 10218-1:2011 Robots and robotic devices – Safety requirements for industrial robots– Part 1: Robots
8. ISO 10218-2:2011 Robots and robotic devices – Safety requirements for industrial robots– Part 2: Robot systems and integration
9. ISO 11593:1996 Manipulating industrial robots -Automatic end effector exchange systems -- Vocabulary/presentation of characteristics
10. ISO 14539:2000 Manipulating industrial robots -Object handling with grasp-type grippers -- Vocabulary and presentation of characteristics
11. ISO/TR 13309:1995 Manipulating industrial robots -- Informative guide on test equipment and metrology methods of operation for robot performance evaluation in accordance with ISO 9283

WG 1: Vocabulary and characteristics

- Structure
  - Convenor: Soon-Geul Lee, Kyung Hee Univ, Korea
  - Formed in June, 2007 at Washington DC Plenary Meeting.
- Revision ISO 8373:1994 (Vocabulary)
  - Feb. 2010: NP is approved
  - 2012: IS(International Standard) is published
- Revision ISO 9787:1999(Coordinate system)
  - NWIP: 2011
  - Revision to include mobile platform coordinate system
- Developing vocabulary for mobile robots
  - New standards on mobile robots(wheeled robots, legged robots)
- ISO concept database
  - http://cdb.iso.org

ISO 8373: Vocabulary – New terms

- **ROBOT-OLD Definition:**
  - **robot/manipulating industrial robot: automatically controlled, reprogrammable , multipurpose manipulator programmable in three or more axes (4.3) which may be either fixed in place or mobile for use in industrial automation applications**
    - **NOTE The robot includes**
    - **- the manipulator (including actuators);**
    - **- the control system (hardware and software).**
- **ROBOT-NEW Definition**
  - **robot: actuated mechanism programmable in two or more axes with a degree of autonomy, moving within its environment, to perform intended tasks**
    - **NOTE 1 A robot includes the control system (2.7) and interface of the control system.**
    - **NOTE 2 The classification of robot into industrial robot (2.9) or service robot (2.10) is done according to its intended application.**

WG 1: Vocabulary – New terms

- **Autonomy**
- **ability to perform intended tasks based on current state and sensing, without human intervention**

- **Robotic device**
- **actuated mechanism fulfilling the characteristics of an industrial robot (2.9) or a service robot (2.10), but lacking either the number of programmable axes (4.3) or the degree of autonomy (2.2)**
- **EXAMPLES Power assist device; teleoperated device; two-axis industrial manipulator (2.1)**

WG 1: Vocabulary – New terms

- **Service robot**
- robot (2.6) that performs useful tasks for humans or equipment excluding industrial automation applications
  - NOTE 1 Industrial automation applications include, but are not limited to, manufacturing, inspection, packaging, and assembly.
  - NOTE 2 While articulated robots (3.15.5) used in production lines are industrial robots (2.9), similar articulated robots used for serving food are servicerobots (2.10).

- **Personal service robot/Service robot for personal use**
- service robot used for a non-commercial task, usually by lay person

- **Professional service robot/Service robot for professional use**
- service robot used for a commercial task, usually by properly trained operator

WG 1: Vocabulary – New terms

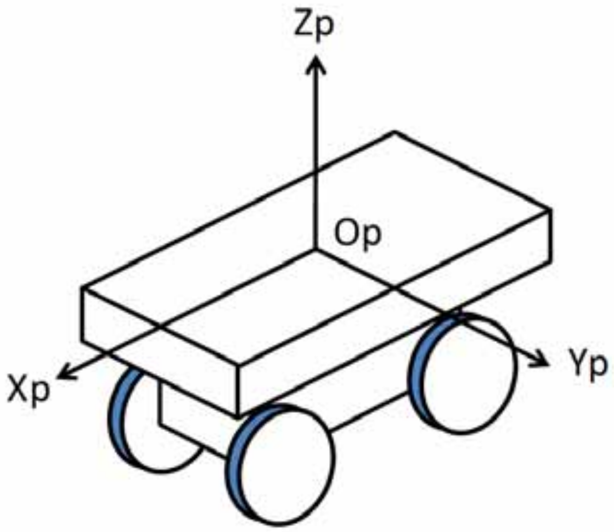
- **Human-robot interaction/HRI**
- tactile or other information exchanges between human and robot

- **User interface**
- means for information exchange between human and robot

- **Collaboration**
- work done by robot(s) and human(s) together to fulfil a task

- **Robot cooperation**
- interaction between multiple robots to ensure that their motions are effective together for the task

ISO 9787: Coordinate systems



- Figure 6 — Example of mobile platform coordinate system

ISO TC184/SC2/WG3 – Industrial Safety

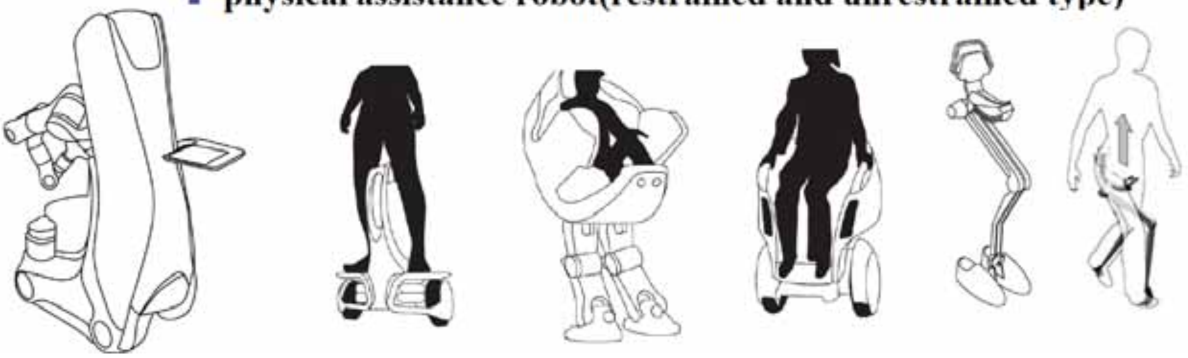
- **Convenor: Jeff Fryman, Robotic Industries Association, USA**
- **It has been active more than 10 years.**
- **Must satisfy ISO 10218 to export robotic products**
  - **USA: ANSI/RIA 15.06, EU: CEN**

- **10218 has been split into two parts.**
  - **Part 1(Robot) – Revised in 2006 and 2011**
    - **Cableless Teach Pendants**
    - **Simultaneous Motion**
    - **Collaborative Operation(Hand guiding, Speed and position monitoring, Power and force limiting)**

  - **Part 2(Robot System and Integration) – Published in 2011**
    - **Safety issues at system integration - Location, power, lighting, grounding, and end-effectors**
- **New Proposal: ISO/TS 15066 *Robots and robotic devices - Safety requirements - Industrial collaborative workspace***

WG 7: Personal Care Safety

- Structure
  - Convenor: Gurvinder Virk, CLAWAR Ltd, UK
  - Started from Oct. 2006 for Safety of Personal Care Robots
  - Consists of more than 30 experts representing USA, UK, Germany, France, Italy, Japan, Korea, IEEE.
- Current Status
  - ISO 13482: Safety requirements- Nonmedical personal care robot
  - Types
    - mobile servant robot
    - person carrier robot
    - physical assistance robot(restrained and unrestrained type)



WG 7: Personal Care Safety - 2

- International Standard by 2013
  - NWIP(New Work Item Proposal) approved in Feb. 2009
  - CD(Committee Draft), approved in Feb. 2010
  - DIS(Draft International Standard), approved in Feb. 2012
  - FDIS(Final Draft International Standard), to be approved in 2013
- Major Sections
  - Hazard identification and risk assessment
  - Safety requirement and protective measures
    - E-stop, Maximum speed, Force control, Human detection, Operational modes, Physical HRI
  - Verification and validation

JWG 9 Safety for Medical devices using robotic technology

- Scope of work
  - The range of medical applications being considered cover invasive and non-invasive procedures such as surgery, rehabilitation therapy, imaging and other robots and robotic devices for medical diagnosis and treatment.
  - In collaboration with
    - IEC/TC 62 (Electrical equipment in medical practice)
  - Will be published as IEC 60601-1-x Collateral standards.
  - Formed in June 2011 and experts from US, UK, Germany, France, Japan, Korea, China, Italy.

WG 8: Service Robots

- Structure
  - Convenor: Seungbin Moon, Sejong University, Korea
  - Started from Oct. 2006.
  - Task is not only to explore the need for standardization but also to develop standards for service robots.
  - Consists of more than 30 experts representing USA, UK, Germany, France, Italy, Japan, China, Sweden, Korea, Hungary, IEEE, IEC, IFR, OMG.

- Study Groups
  - SG 2: Performance => NWIP is proposed.
  - SG 4: Modularity => Study Group is formed.

- Any new issues could be discussed here

Service Robot – Performance Criteria

- Robots and robotic devices — Performance criteria and related test methods for service robot
  - Part 1: Wheeled mobile servant robot
  - Part 2: Wheeled person carrier robot
  - Part 3: Restrained wearable robots
  - Part 4: Legged robots

- Part 1. Wheeled mobile servant robot
  - Speed on different surfaces
  - Prevention of falling down
  - Mobility over the slope
  - Mobility over the sill
  - Stopping time
  - Obstacle avoidance
  - Obstacle detection
  - Relative distance/speed b/w human and ro

WG 8/Study Group on Modularity

- Structure
  - Internal study group is formed in June 2011.
  - Expanded in Oct. 2012
  - First meeting in Seoul, Oct. 2012.
  - Second meeting in San Francisco, Jan. 2013
  - Third meeting in Bristol, June 2013
  - Fourth meeting in Beijing, Oct. 2013

SG Discussion in Seoul

- What is Modularity?
  - Interoperability
  - Interface
  - Interchageability
  - Robot component

- Scope
  - Software interface
    - Architecture, Software component model, Componet API
    - Communication protocol, OS support
  - Hardware interface
    - connectors, sensor interface, communication physical layer
    - Manipulator link module
  - Mechanical interface
    - Motor shaft

Needs on Modularization

- Industrial robots
  - Market is fully developed already and end-users are system integrators.
  - Software : Major manufacturers do not want open up their software.
  - Hardware interface: Major manufacturers do not have much interest.
  - Mechanical interface: Some(mechanical interface) are standardized already.

- Personal care robots
  - Similar to consumer electronics industry, where end-users are individuals.
  - Software : Needed
  - Hardware interface : Needed
  - Mechanical interface : Needed

- Medical robots
  - It should consider regulation or safety issues.
  - Software :
  - Hardware interface: Surgery tool interface could be helpful.
  - Mechanical interface :

Urgency and Feasibility

- **What will be most urgent issues?**
  - **Software :**
  - **Hardware interface:**
  - **Mechanical interface:**

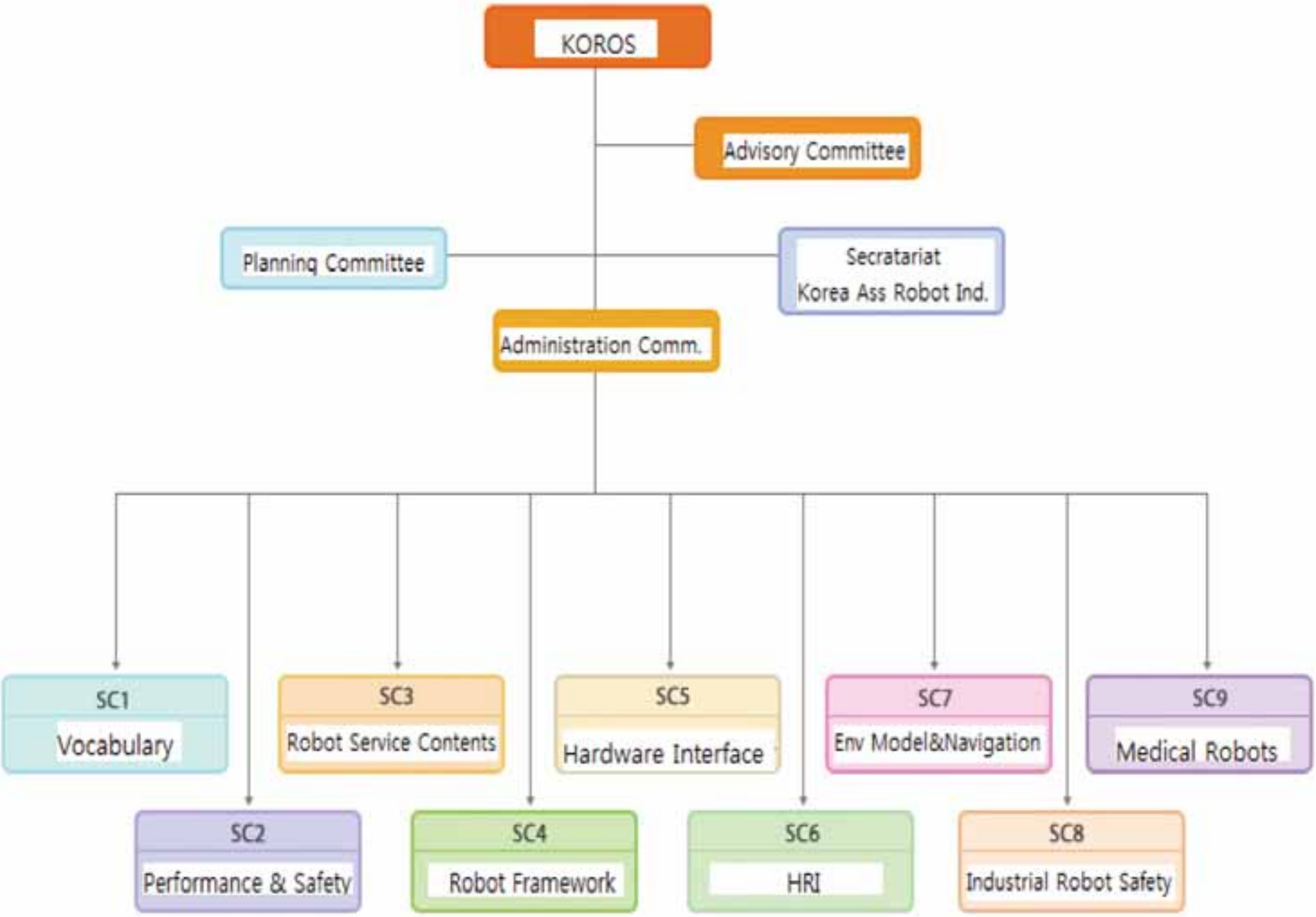
- **Will they be feasible?**
  - **OMG, OPROS, and China agree to discuss more on issue of merging into one software interface standard. More inputs from other countries will be sought.**
  - **Other groups, including ROS and OROCOS, will be invited to join the group.**
  - **Hardware interface and mechanical interface were discussed and will be further discussed in the future meetings.**

Korean Activities

- **KOROS (Korean Robot Standard) Forum**
  - **Formed in 2005: www.koros.or.kr**
  - **Maintains 68 standards currently**
  - **9 SCs**

- **KS (Korea Industrial Standard)**
  - **21 industrial robot standards**
  - **15 standards for service robots, including vocabularies, safety, performance.**

- **R-mark Certification**
  - **Cleaning robot certification initiated in 2009.**
  - **Education tool robot certification under preparation now.**



KOROS

Advisory Committee

Planning Committee

Secratariat
Korea Ass Robot Ind.

Administration Comm.

SC1
Vocabulary

SC2
Performance & Safety

SC3
Robot Service Contents

SC4
Robot Framework

SC5
Hardware Interface

SC6
HRI

SC7
Env Model&Navigation

SC8
Industrial Robot Safety

SC9
Medical Robots

Thank you !

TECHNICAL MEETING / Robotics Information Day

ROS-Industrial™

A Disruptive Community Approach to Industrial Robotics Software

Paul Evans

Southwest Research Institute® (SwRI®)

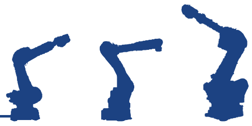
11 December 2012

Biography

- Paul Evans
- Director of Research and Development for the Manufacturing Systems Department at SwRI
- Focused on solving real-world challenges through applied research and development
- Specialized in advanced industrial robotics and automation programs
- Graduated with a MSME from Iowa State University and a Professional Engineer
- paul.evans@swri.org

Agenda

- Overview of ROS
- Overview of ROS-Industrial
- Applications for ROS-Industrial
- ROS-Industrial Community Approach



ROS Overview

ROS Motivation

Research Robotics Challenges

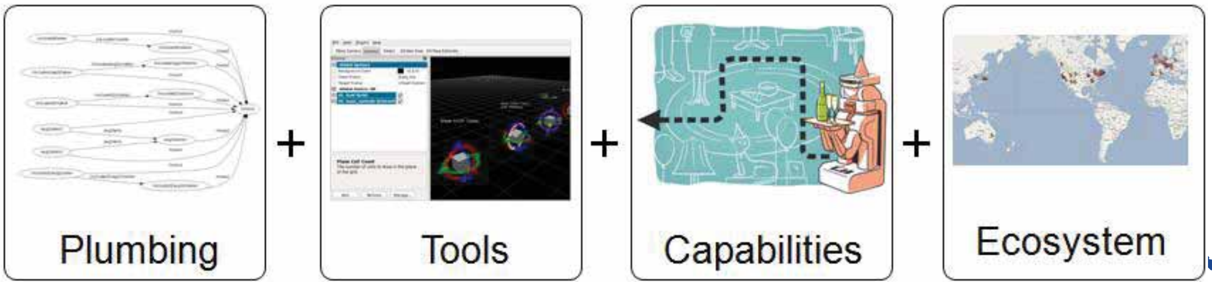
- Reinvention of the Wheel
- Little Commonality
- Short Lifespan
- Difficult to Compare Results

ROS Solves These

ROS: Robot Operating System

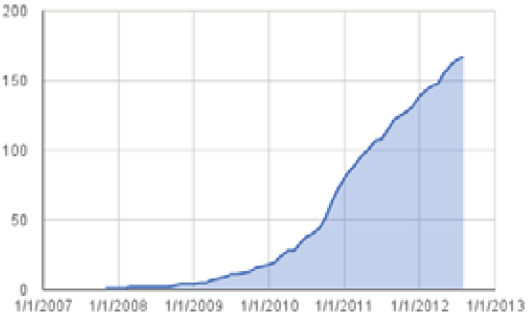
- Open source (BSD)
- Created by Willow Garage
- Maintained by Open Source Robotics Foundation (OSRF)

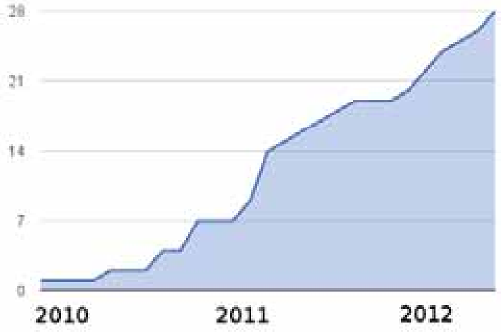
http://ros.org/wiki/Industrial



Statistics

- ROS Core statistics by: https://www.ohloh.net/p/ROSorg
- 11,146 commits
- 43 contributors
- 148,163 lines of code
- Long source history maintained by a large development team with stable year-over-year commits
- 38 years of effort (COCOMO model)
- Estimated cost $2,063,327





What Can ROS Do?

ROS 5 Year Video: http://youtu.be/zV48Pq0muEk

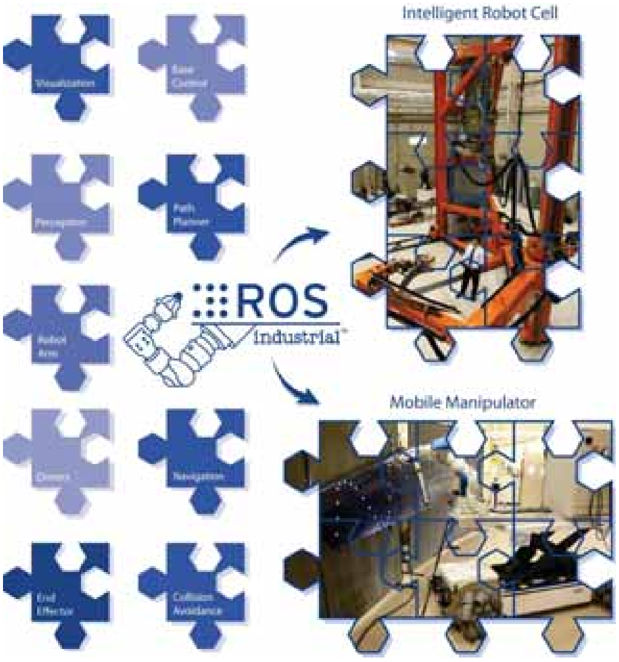
ROS-Industrial

ROS-Industrial Motivation

- Motivated by desire to solve industries toughest challenges using industrial robotics and automation
- Driven by application needs (i.e. real-world and challenging industrial needs)
    - – Fixtureless automation
    - – Dynamic pick and place
    - – Flexible automation (many small & diverse part runs)
    - – Sensor driven automation
- Reduction in integration cost by standardizing interfaces and enabling reuse

What is ROS-Industrial

- Open-Source (BSD) software distribution – extension of ROS
- Advanced development tools
- New and additional capabilities
- Software portability and flexibility for COTS hardware
- Technology compatibility and ease of integration
- Transition of basic research to applications
- A community of developers



Hardware Drivers Examples

- Robots
  - Motoman
  - Adept
  - Universal
  - ABB
  - Fanuc, Kuka (Coming soon)
- Peripherals
  - Robotiq
  - EtherCAT (Beckhoff Modules)
  - Serial
  - Ethernet

App: Automated Painting

- Automated spray paint processes
  - Reduce emissions (regulation)
  - Reduce exposure (personnel)
  - Reduce cost (materials)
  - Increase quality (consistency)
- Challenges
  - Unconstrained location
  - “Random” part order
  - Real time processing
  - Moving parts

Solution: Automated Painting

- 3D Sensing (ROS/OpenNI)
- 3D Processing (ROS/PCL)
- Process based path planning (SwRI)
- Robot IK solvers (ROS/MoveIt!)
- Robot workcell visualization (ROS/Rviz)
- Distributed system (ROS/Core)
- Data acquisition/playback (ROS/bag)

App: Robotic Sorting

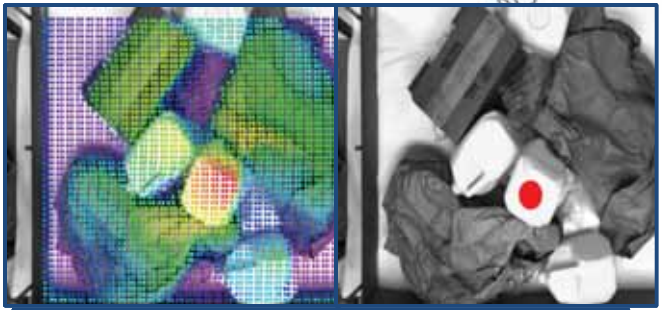
- Random product sorting application
  - Value in waste streams
  - Labor intensive, worker fatigue
  - Increased sorting rate/quality
- Challenges
  - Waste stream variety
  - High speed
  - Close quarters

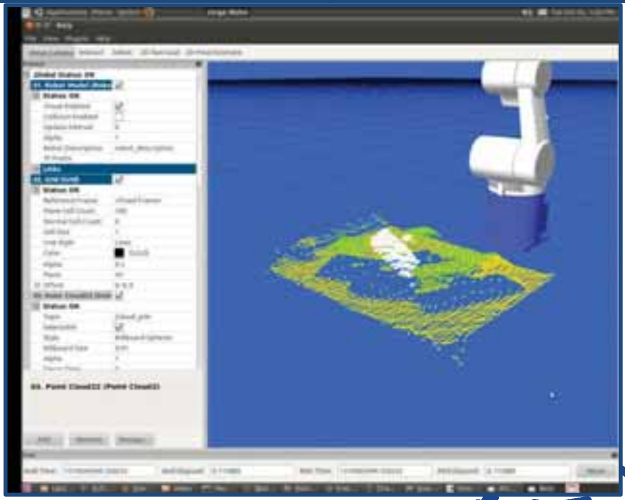
http://www.smh.com.au/news/national/drastic-plastic-bag-ban-looms/2007/03/10/1173478729172.html

---

Solution: Robotic Sorting

- 3D sensing (ROS/OpenCV, PCL)
- 3D processing (ROS/PCL)
- Pick selection (SwRI)
- Robot IK solvers (ROS/MoveIt!)
- Collision checkers (ROS/MoveIt!)
- Robot workcell visualization (ROS/Rviz)





More Capabilities

Leveraging ROS

Pick & Place Demonstration: http://youtu.be/_WG-45cZSUQ

Visualization

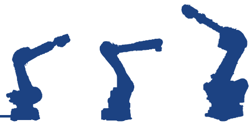
Visualization and Path Planning: http://youtu.be/qd76wAywZos

Platform Independence

Adept Robot Demonstration: http://youtu.be/awdTgpyOmxE

Future of ROS-Industrial

- Installed systems
- Process based path planners
- More hardware support
- Physics based simulation
- Incorporate external libraries
- Code analysis and statistics
- More tutorials and documentation
- Certified releases



ROS-Industrial Community Approach

Community

- Openness encourages participation and collaboration
- Many small, yet organized efforts result in more capable software
- Non-traditional approach for the industrial space

Partial View of the Community

SwRI

Willow Garage

YASKAWA
MOTOMAN ROBOTICS

SSS MANUFACTURING

Carnegie Mellon University

BOEING

ROBOTIQ

Open Source Robotics Foundation

CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

adept

CASE
CASE WESTERN RESERVE UNIVERSITY

hiDOF

ipi
industrial perception, inc.

Fraunhofer

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

Ways to Participate

- Independently Contribute/Participate:
    - Define interface standards
    - Develop software
    - Documentation
- OEMs – develop interfaces to your equipment
- Integrators - Use it for projects and customers
- Join the ROS-Industrial Consortium
- There are a number of other ways as well…

ROS-Industrial Consortium

- Accelerate Code Development
    - Advanced Capabilities
    - Code Quality Standards/Enforcement
    - Testing, Reliability, Robustness
    - Training
    - Maintenance
- Build Community
    - Attract User-Generated Content
    - Maintain Open-Source Repository, Wiki, Roadmap
    - Ensure Code Reusability

How Will it Work?

- Membership fees first cover operational expenses
- Funds, over and above the operating expenses, will be appropriated toward research objectives.
- Focused technical projects will be formed and funded by full members
- Open source software:
  - All software developed under general funds
  - Project software at the discretion of the funding group

Conclusions

- ROS has proven to be disruptive to robotics research
- ROS architecture, capabilities, tools, and open source approach rival commercial options
- ROS-Industrial brings the power of ROS to the industrial robotics and automation market
- Support for ROS-Industrial is growing
- The ROS-Industrial Consortium will foster the continued development and maintain focus on industry needs

Questions?

Main site:	rosindustrial.org
Software site:	code.google.com/p/swri-ros-pkg/
Docs site:	ros.org/wiki/Industrial
Consortium site:	ric.swri.org

Introduction to Robotic Technology Component (RTC-1.1) Specification

Geoffrey Biggs
Intelligent Systems Research Institute
National Institute of Advanced Industrial Science and Technology
Japan

The Robot Technology Component standard

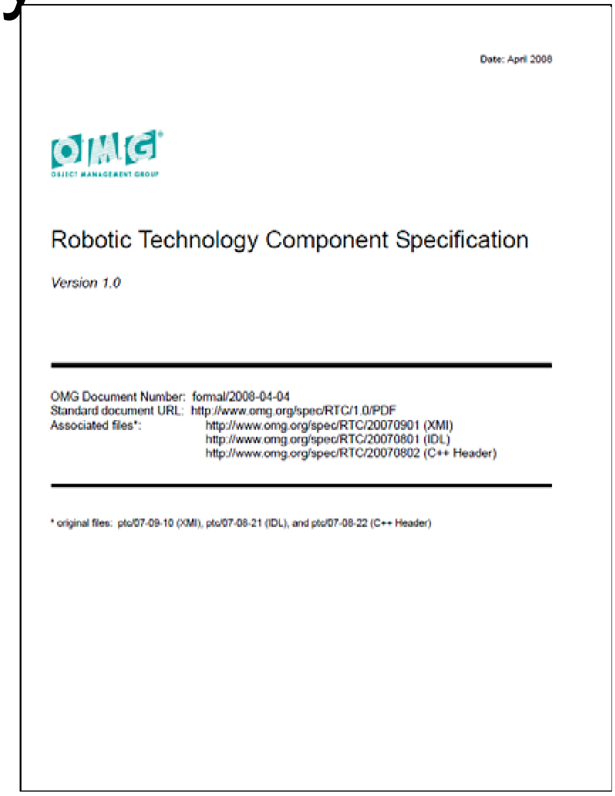
- OMG specification for component-based robot software.
- Aims for greater compatibility and reusability amongst vendors of robot software.
  - Not just the software itself but also the tools.
- An open specification that anyone can implement.

Benefits of the OMG RTC standard

- Manage the lifecycle of all components in a uniform way.
- Provides the data structures necessary to specify components and the related objects for a complete system.
- Separates functional specification and execution control.
  - Supports a variety of execution patterns.
- Static and dynamic component networks.

OMG RTC standardisation history

- September, 2005
  RFP：Robot Technology Components (RTCs) publication。
- February, 2006
  Initial Response : PIM and PSM for RTComponent submissions from AIST (Japan)、RTI (America)
- April, 2006
  Merged submission
- September, 2009
  Accepted by the architecture board. Finalisation begins.
- August, 2007
  Finalisation completed
- September, 2007
  Finalisation result passed by the AB.
- April, 2008
  OMG RTC standard 1.0 published
- September, 2012
  OMG RTC standard 1.1 published



Date: April 2008

OMG OBJECT MANAGEMENT GROUP

Robotic Technology Component Specification

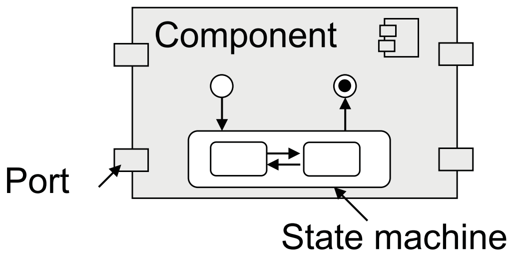
Version 1.0

OMG Document Number: formal/2008-04-04
Standard document URL: http://www.omg.org/spec/RTC/1.0/PDF
Associated files*: http://www.omg.org/spec/RTC/20070901 (XMI)
http://www.omg.org/spec/RTC/20070801 (IDL)
http://www.omg.org/spec/RTC/20070802 (C++ Header)

* original files: ptc/07-09-10 (XMI), ptc/07-08-21 (IDL), and ptc/07-08-22 (C++ Header)

Package 1: Lightweight RTC

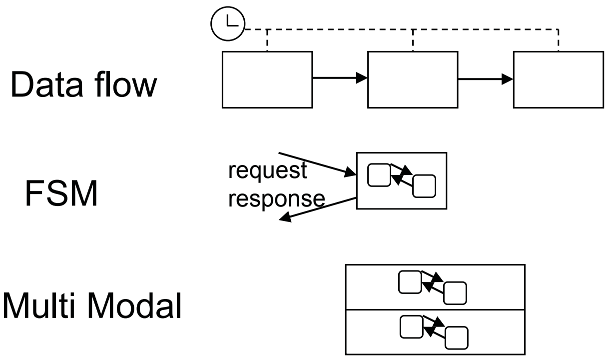
- Lightweight RTC
  - Stereotypes for components, ports, connectors, etc.
  - Component lifecycle
  - Execution contexts
  - Does not include the introspection functionality
  - For static component networks



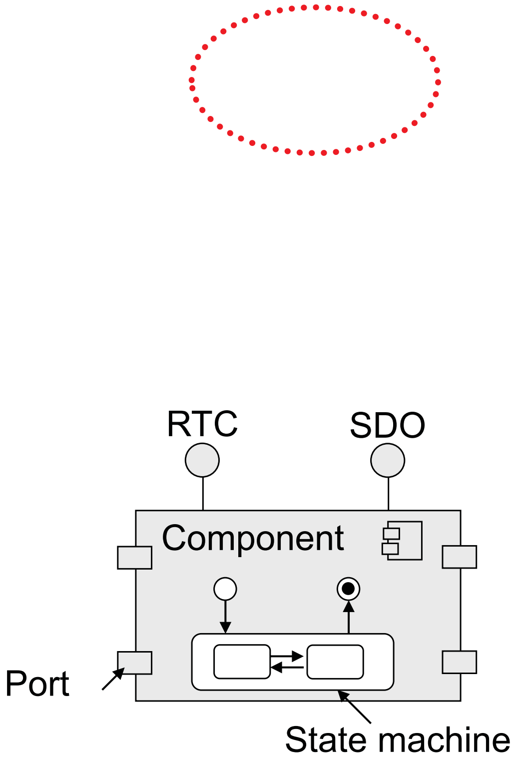
Package 2: Execution Semantics

- Execution Semantics
  - Provides execution patterns commonly used in robotics

1. Synchronous execution processing data (data-flow type)
2. Stimulus-response or event-driven (FSM type)
3. Mixture of execution methods (multi-modal type)



Package 3: Introspection



- Introspection
  - Interfaces for acquiring component information
  - Based on the OMG Super Distributed Object standard
  - Dynamic component networks

OMG RTC family

Name	Vendor	Feature
OpenRTM-aist	AIST	C++, Python, Java
OpenRTM.NET	SEC	.NET (C#,VB,C++/CLI, F#, etc..)
miniRTC, microRTC	SEC	CAN・ZigBee
Dependable RTM	SEC/AIST	Functional safety (IEC61508) capable and certified
RTC CANOpen	SIT, CiA	CANOpen CiA (Can in automation) based RTC standard
PALRO	Fuji Soft	C++ implementation for small humanoids
OPRoS	ETRI	Korean national project
GostaiRTC	GOSTAI, THALES	C++ implementation for the URBI robot language
Honda RTM	Honda	C++ implementation

RTC RTF 1.1 overview

- Published September, 2012
- Fixed 8 issues and deferred 9
- Resolved changes were correcting the specification
  - Diagram fixes
  - Grammatical corrections
- Minor comments received from AB review
  - Missing #pragmas in IDL, XMI bugs

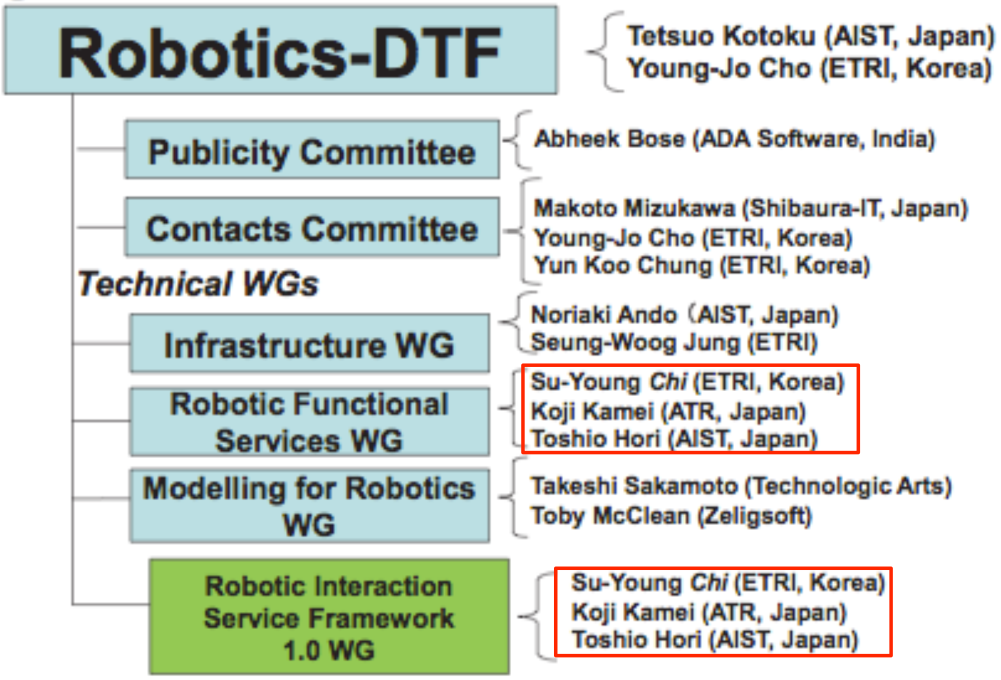
Part 1: Introduction to RLS 1.1 Spec.
Part 2: Introduction to RoIS 1.0 Spec.
Part 3: Introduction of UNR Platform

an implementation of RoIS and RLS Spec. in Japan

Presentation: Koji Kamei (JARA / ATR)
Co-Chair of Robotic Functional Services WG

OMG Robotics Information Day, Burlingame, Dec. 11, 2012

Organization (from March. 20th, 2012)



Robotics-DTF
- Tetsuo Kotoku (AIST, Japan)
- Young-Jo Cho (ETRI, Korea)

Publicity Committee
- Abheek Bose (ADA Software, India)

Contacts Committee
- Makoto Mizukawa (Shibaura-IT, Japan)
- Young-Jo Cho (ETRI, Korea)
- Yun Koo Chung (ETRI, Korea)

Technical WGs

Infrastructure WG
- Noriaki Ando（AIST, Japan)
- Seung-Woog Jung (ETRI)

Robotic Functional Services WG
- Su-Young *Chi* (ETRI, Korea)
- Koji Kamei (ATR, Japan)
- Toshio Hori (AIST, Japan)

Modelling for Robotics WG
- Takeshi Sakamoto (Technologic Arts)
- Toby McClean (Zeligsoft)

Robotic Interaction Service Framework 1.0 WG
- Su-Young *Chi* (ETRI, Korea)
- Koji Kamei (ATR, Japan)
- Toshio Hori (AIST, Japan)

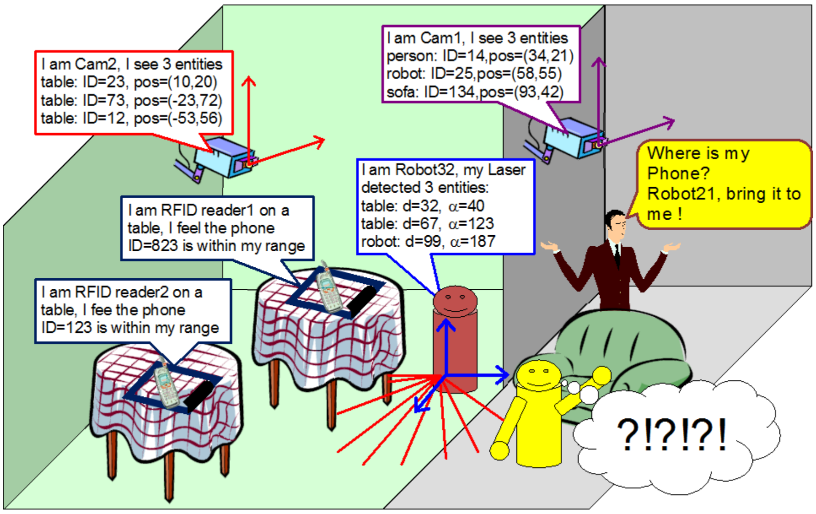
Introduction to Robotic Localization Service (RLS-1.1) Specification

Shuichi Nishio (JARA / ATR)
Chair of Robot Localization Service (RLS-1.1) RTF

Koji Kamei (JARA / ATR)

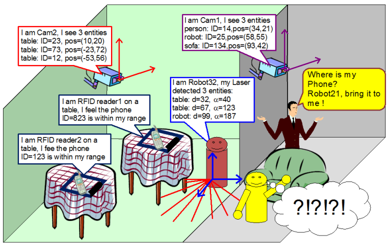
RLS: Robotic Localization Service

Defines a localization service for providing robotic services:
ex. navigation, manipulation and human-robot interaction.



Challenging issues in the example

- Some sensors only provide partial location information.
  - The camera sensor can only provide 2D information, and RF tag reader can only provide proximity information.
- Sensor outputs are not always correct.
  - They might measure two or more entities as a single object or even miss it.
- Matching observations between different sensors require efforts.
  - The identity association problem happens every time multiple sensors are used: the wall camera and the laser range scanner install in the robot provide different aspects of objects.



RLS Features

- A new framework for robotic localization (“RoLo” used as prefix)
  - Representation of location information specific to robotic usage.
  - Based on the widespread GIS standards. (ISO/TC211)
- Architecture package (RoLo architecture)
  - Data architecture for representing structures and accompanying operations for representing information necessary for robotic usage.
  - Coordinate system / coordinate reference system definitions for pose or identity information.
  - Structures for representing error estimation.
- Data Format package (RoLo format)
  - Data formats for formatting and exchanging resulting localization data.
- Interface package (RoLo service)
  - Service interface for treating resulting localization data.
  - A basis for dynamically negotiating module functionality information.

RLS: History

2007/06: RFP for Robot Localization Service (RLS) issued. (Brussels)
2007/12: initial submissions from ETRI and JARA. (Burlingame)
2008/06: revised submission. FTF chartered. (Ottawa)
2009/06: final report from FTF approved. (Costa Rica)
2010/02: RLS 1.0 published. (formal/2010-02-03)
2011/06: final report from RLS1.1-RTF approved. (Salt Lake City)
2012/08: RLS 1.1 published. (formal/2012-08-01)

Liaison with ISO/TC211 will be considered as a part of RLS 2.0 work.

submitted by	Electronics and Telecommunications Research Institute (ETRI) Japan Robot Association (JARA) Samsung Electronics Co., Ltd.
supported by	Hitachi, Ltd. / National Institute of Advanced Industrial Science and Technology / New Energy and Industrial Technology Development Organization / Shibaura Institute of Technology / Technologic Arts Incorporated / University of Tsukuba

---

Changes in RLS version 1.1

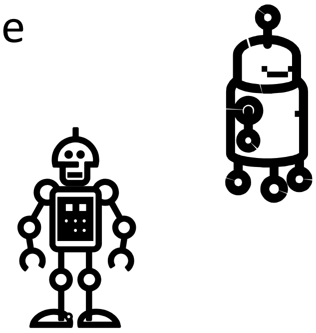
- Bug fixes
- Pose representation extended
  - representing pose information in combination with position information.
  - partially revised in this version and will be totally revised in the next version (in coordination with ISO/TC211).
- Identity (ID) Information extended
  - two coordinate reference systems and accompanying coordinate systems are defined for identity systems that are represented in numerical values and symbolic values
- Usage examples extended

Introduction to Robotic Interaction Service (RoIS-1.0) Specification

Su-Young Chi (ETRI), Young-Jo Cho (ETRI)
Toshio Hori (AIST), Koji Kamei (JARA/ATR)

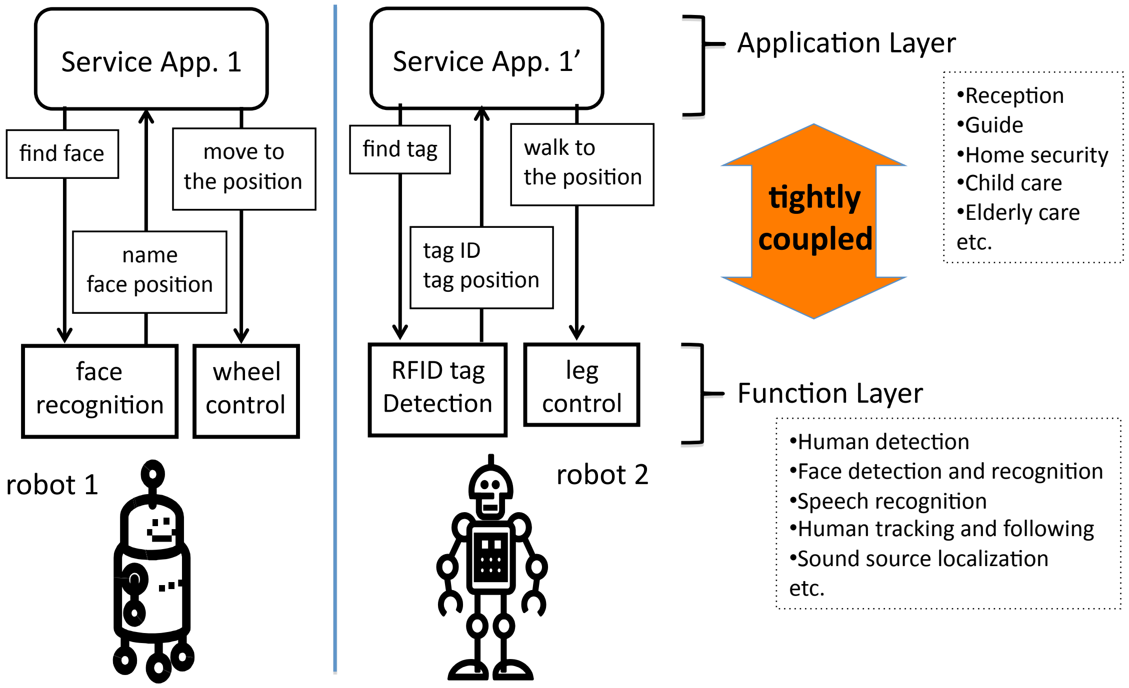
Service Robots in Our Living Environment

- Possible robotic services:
  - Reception service at the entrance
  - Guide service in museums
  - Home security
  - Childcare/Nursing robots
  - Elderly care robots, etc.



⇨ The service robot market will grow dramatically...

Existing Robotic Service Applications



Service App. 1

find face

move to the position

name
face position

face recognition

wheel control

robot 1

Service App. 1’

find tag

walk to the position

tag ID
tag position

RFID tag Detection

leg control

robot 2

Application Layer

- Reception
- Guide
- Home security
- Child care
- Elderly care

etc.

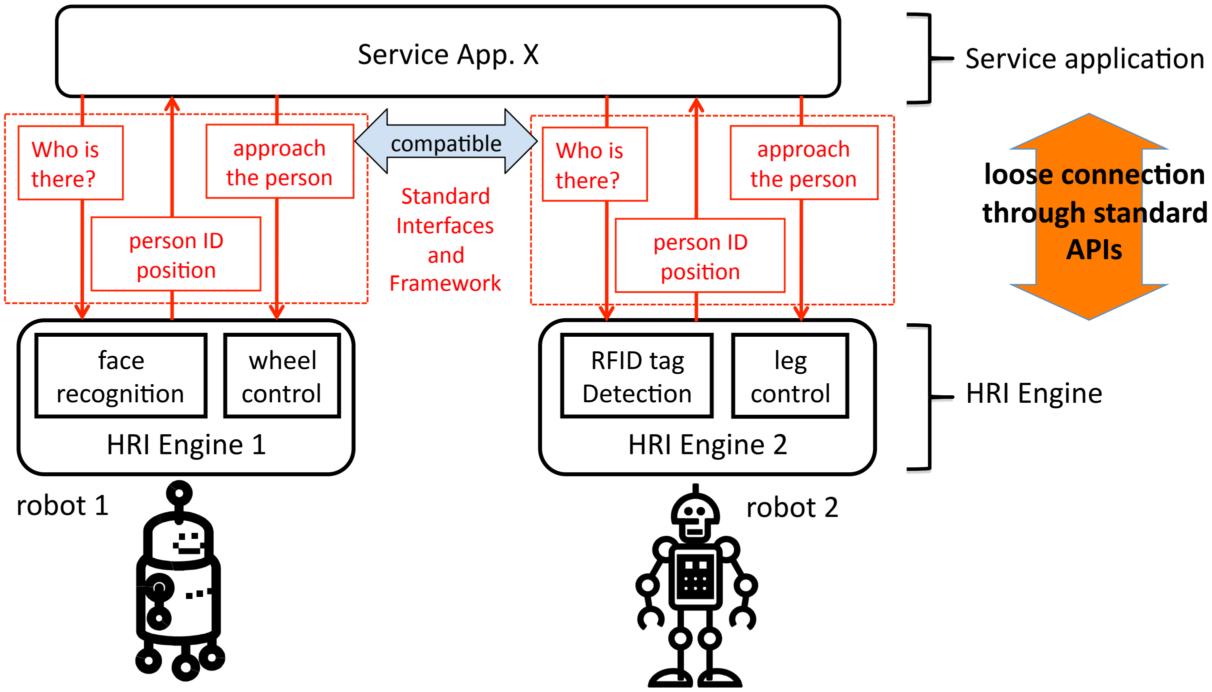
tightly coupled

Function Layer

- Human detection
- Face detection and recognition
- Speech recognition
- Human tracking and following
- Sound source localization

etc.

Possible Solution of Software Reuse



Service App. X

Service application

Who is there?

approach the person

person ID
position

compatible

Standard Interfaces and Framework

Who is there?

approach the person

person ID
position

loose connection through standard APIs

face recognition

wheel control

HRI Engine 1

RFID tag Detection

leg control

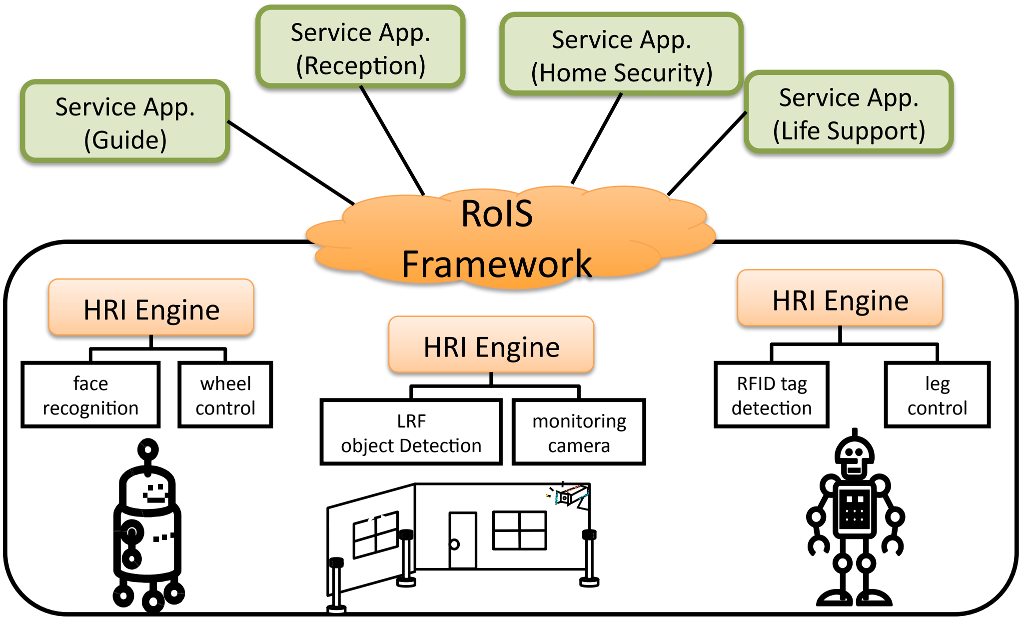
HRI Engine 2

HRI Engine

robot 1

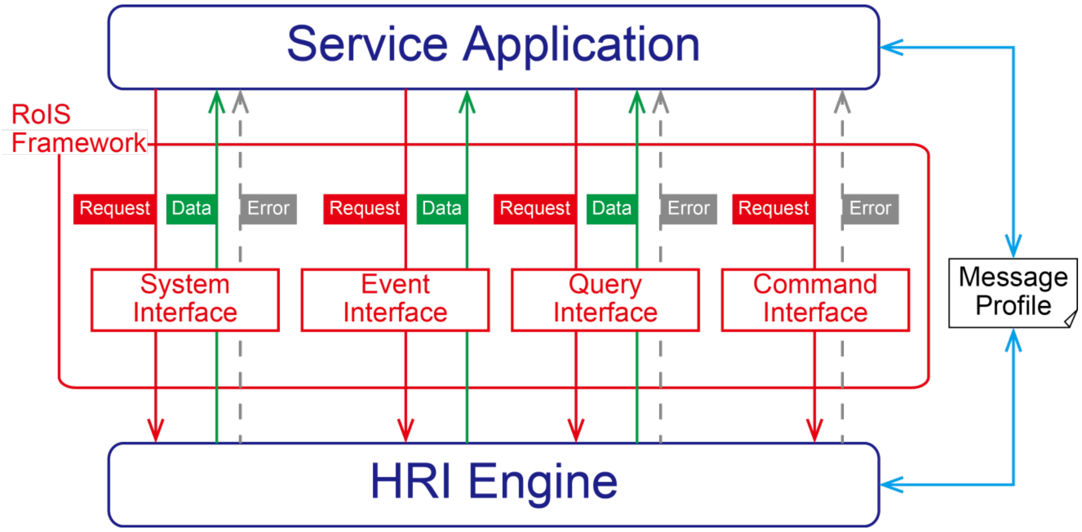
robot 2

Schematic Picture of RoIS Framework



RoIS Framework defines messaging protocols between services and HRI Engines and profile description of functional components including 15 basic components.

Interfaces of RoIS Framework and its message flows



Interfaces defined in RoIS Framework

- **System Interface**
  - Manages the connection status between the Service Application and HRI Engine.
- **Event Interface**
  - Enables the Service Application to receive notifications on changes in HRI-Engine status.
- **Query Interface**
  - Enables the Service Application to query the HRI Engine on information it holds.
- **Command Interface**
  - Enables the Service Application to send commands to the HRI Engine.

HRI functions for Robotic Services

- Robotic services require Human-Robot Interaction functions
  - **To understand the surroundings**, such as
    - Person detection, Person identification,
    - Sound detection and
    - Speech recognition,

Sensor-related functions

  - **To provide services**, such as
    - Speech synthesis,
    - Reaction,
    - Navigation and Follow.

Actuator-related functions

⇨ RoIS provides 15 Basic HRI Components.

15 Basic HRI Components

Sensor-related Components

名称	機能
System Information	Acquire the system status
Person Detection	Detect number of person (s)
Person Localization	Detect position of people
Person Identification	Identify person(s)
Face Detection	Detect number of face(s)
Face Localization	Detect position of face(s)
Sound Detection	Detect number of sound sources
Sound Localization	Localize sound source(s)
Speech Recognition	Recognize spoken language
Gesture Recognition	Recognize person's gesture

Actuator-related Components

名称	機能
Speech Synthesis	Generates robot speech
Reaction	Performs specified reaction
Navigation	Moves to specified target location
Follow	Follows a specified target object
Move	Moves to specified distance or curve

Robot developers (i.e. HRI Engine providers) can provide HRI functions not listed here as User-defined HRI Components (with their profiles).

RoIS: History

2010/06: RFP for Robotic Interaction Service (RoIS) Framework issued. (Long Beach)
2010/12: initial submissions from ETRI and JARA. (Burlingame)
2011/06: revised submission approved. FTF chartered. (Salt Lake City)
2011/08: RoIS 1.0 Beta1 published. (dtc/2011-08-06)
2012/06: final report from FTF approved. (Cambridge)
2012/09: RoIS 1.0 Beta2 published. (dtc/2012-06-27)

RoIS 1.0 Specification will be published soon.

submitted by

Electronics and Telecommunications Research Institute (ETRI)
Japan Robot Association (JARA)

supported by

Advanced Telecommunications Research Institute International / Future Robot Co., Ltd. / Hitachi, Ltd. / Korean Robot Association / National Institute of Advanced Industrial Science and Technology / New Energy and Industrial Technology Development Organization / Shibaura Institute of Technology / Technologic Arts Incorporated / University of Tokyo / University of Tsukuba

UNR Platform

a software platform for robotic services that implements RoIS and RLS Specifications

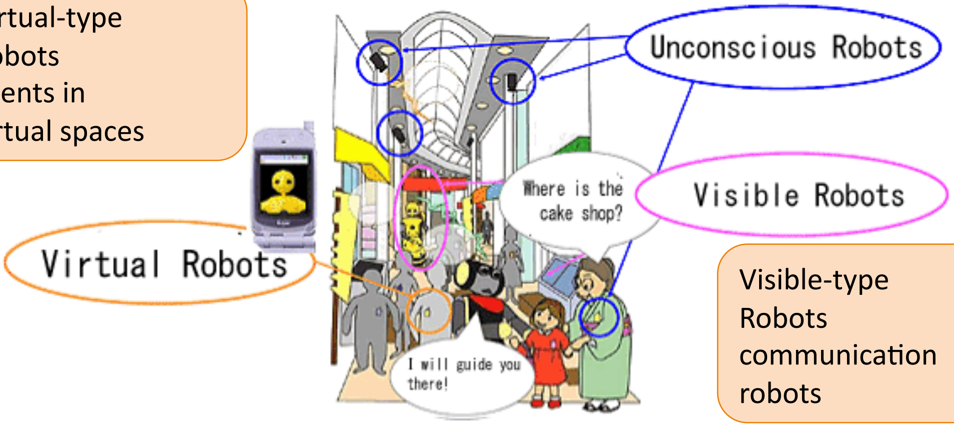
Koji Kamei (ATR)

Shuichi Nishio (ATR)

Networked Robot System (2004 - 2009)

Unconscious-type Robots
sensors embedded in environments

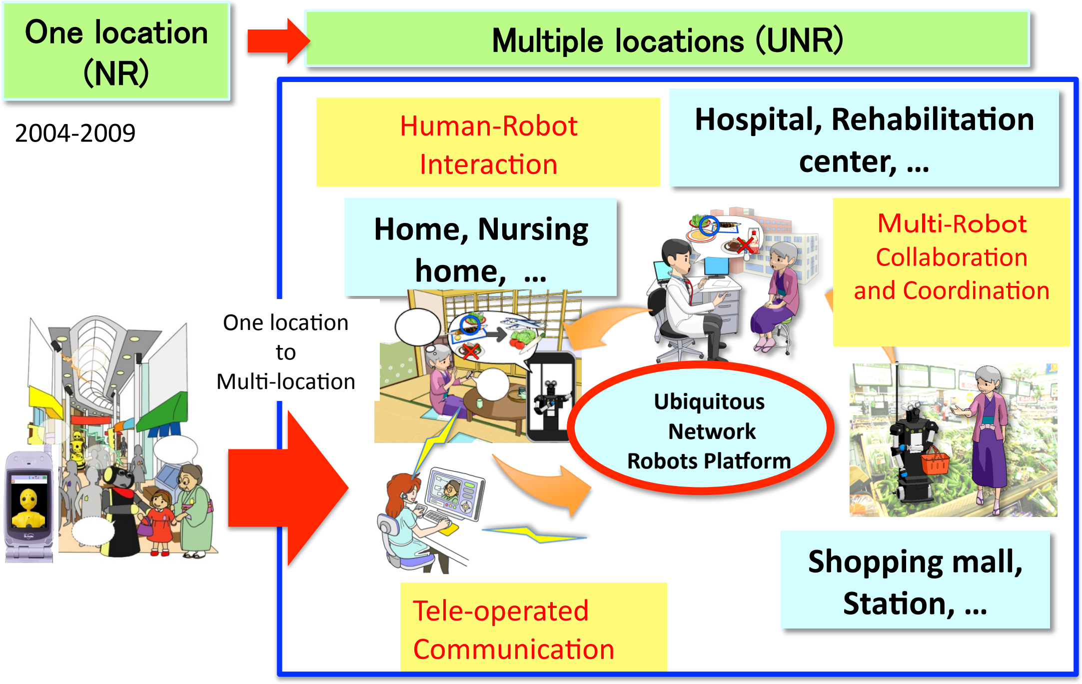
Virtual-type
Robots
agents in
virtual spaces



Visible-type
Robots
communication
robots

Coordination of three different kinds of robots
to support activities of people in real-world environments

Ubiquitous Networked Robots for Life Support (2009 – 2013)



One location (NR)

2004-2009

Multiple locations (UNR)

One location to Multi-location

Human-Robot Interaction

Hospital, Rehabilitation center, …

Home, Nursing home, …

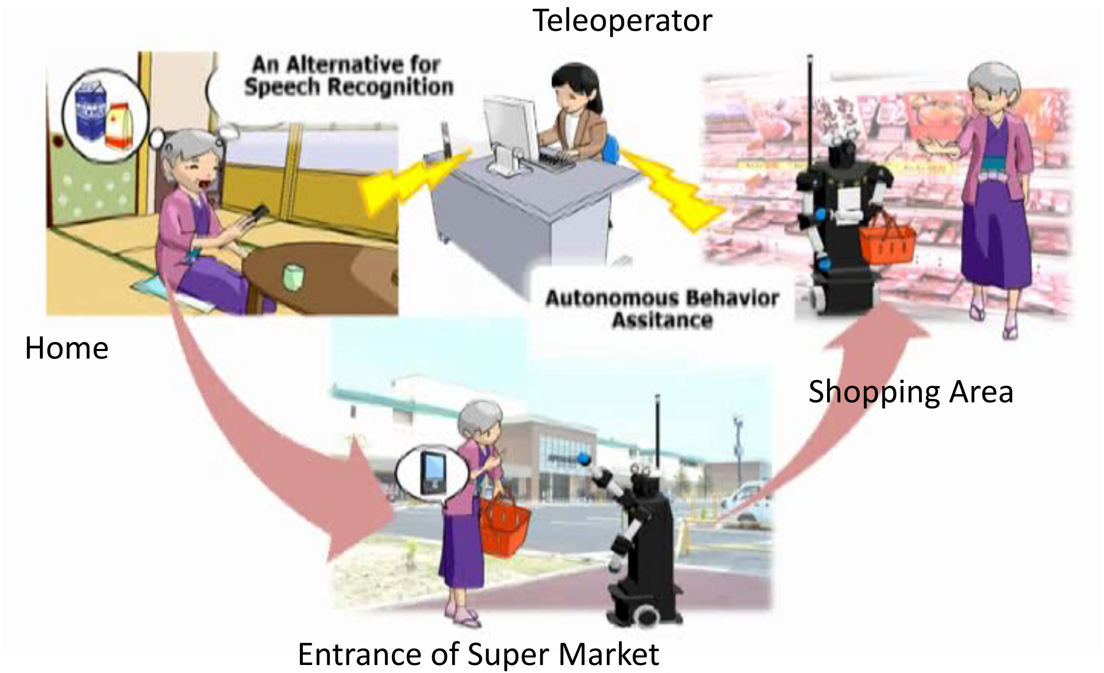
Multi-Robot Collaboration and Coordination

Ubiquitous Network Robots Platform

Tele-operated Communication

Shopping mall, Station, …

Shopping Support for Elderly Customer (2009)



Teleoperator

Home

Shopping Area

Entrance of Super Market

Shopping Guidance with UNR over Three Areas

Area 1: Home

Shopping list with a smart phone

Advanced ICT will be available

Area 2: Entrance of Super Market

Say hello with networked robots

Fully equipped environment in sensor network or WLAN environment

Area 3: Shopping Area

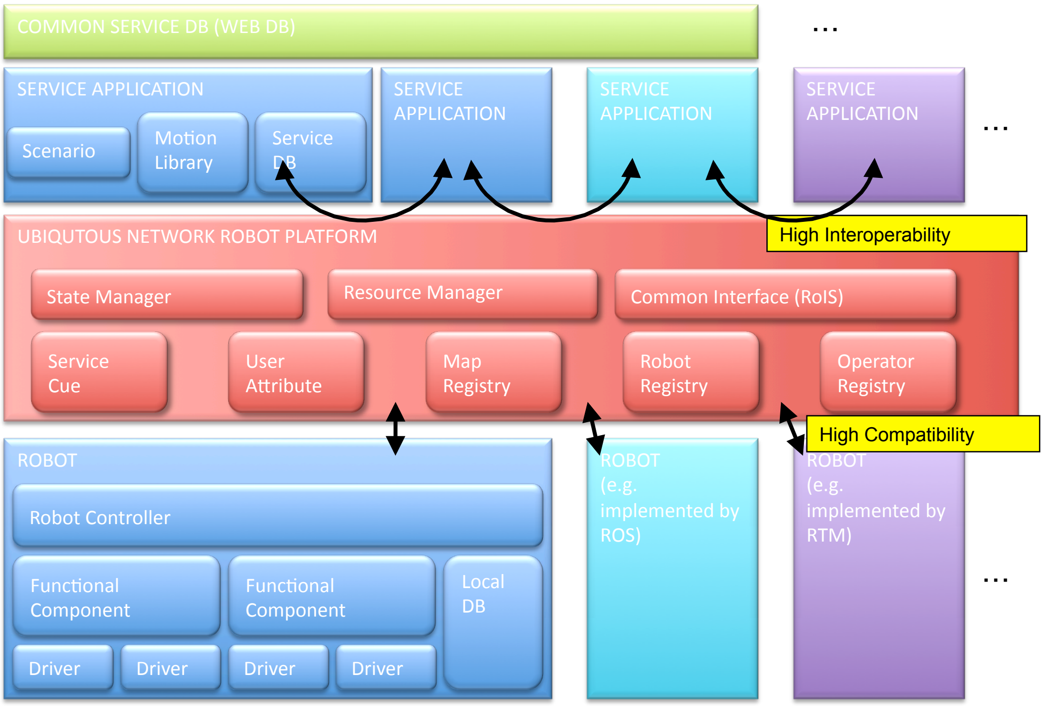
Combination of Autonomy and Tele-operation

Communication with robots create emotional happiness

UNR Platform

- separates the developments of robots and robotic service applications.
- connects various service applications to appropriate robots in a variety of places.
- permits service applications to share their own data between each other.

UNR Platform Architecture



COMMON SERVICE DB (WEB DB)

…

SERVICE APPLICATION

Scenario

Motion Library

Service DB

SERVICE APPLICATION

SERVICE APPLICATION

SERVICE APPLICATION

…

High Interoperability

UBIQUTOUS NETWORK ROBOT PLATFORM

State Manager

Resource Manager

Common Interface (RoIS)

Service Cue

User Attribute

Map Registry

Robot Registry

Operator Registry

High Compatibility

ROBOT

Robot Controller

Functional Component

Functional Component

Local DB

Driver

Driver

Driver

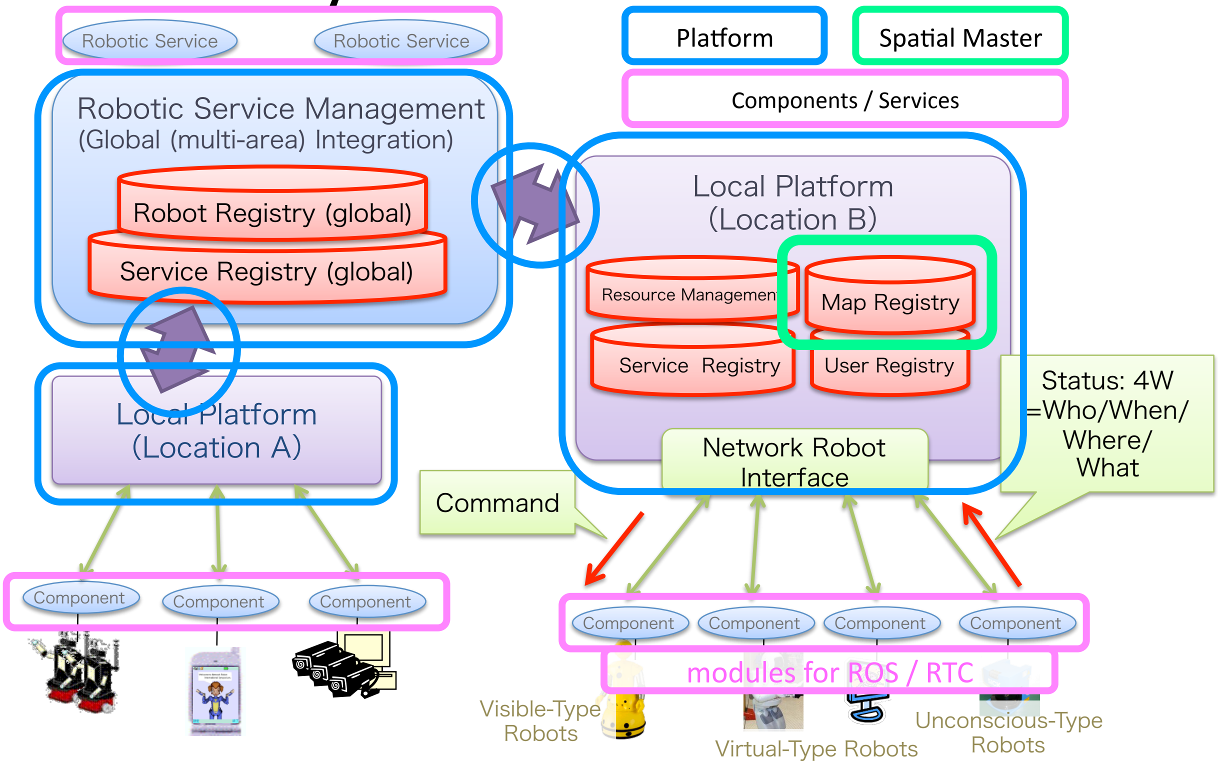
Driver

ROBOT (e.g. implemented by ROS)

ROBOT (e.g. implemented by RTM)

…

Two-Layer Platform Architecture



Robotic Service

Robotic Service

Platform

Spatial Master

Components / Services

Robotic Service Management
(Global (multi-area) Integration)

Robot Registry (global)

Service Registry (global)

Local Platform
(Location B)

Resource Managemen

Map Registry

Service Registry

User Registry

Network Robot Interface

Local Platform
(Location A)

Command

Status: 4W
=Who/When/
Where/
What

Component

Component

Component

Component

Component

Component

Component

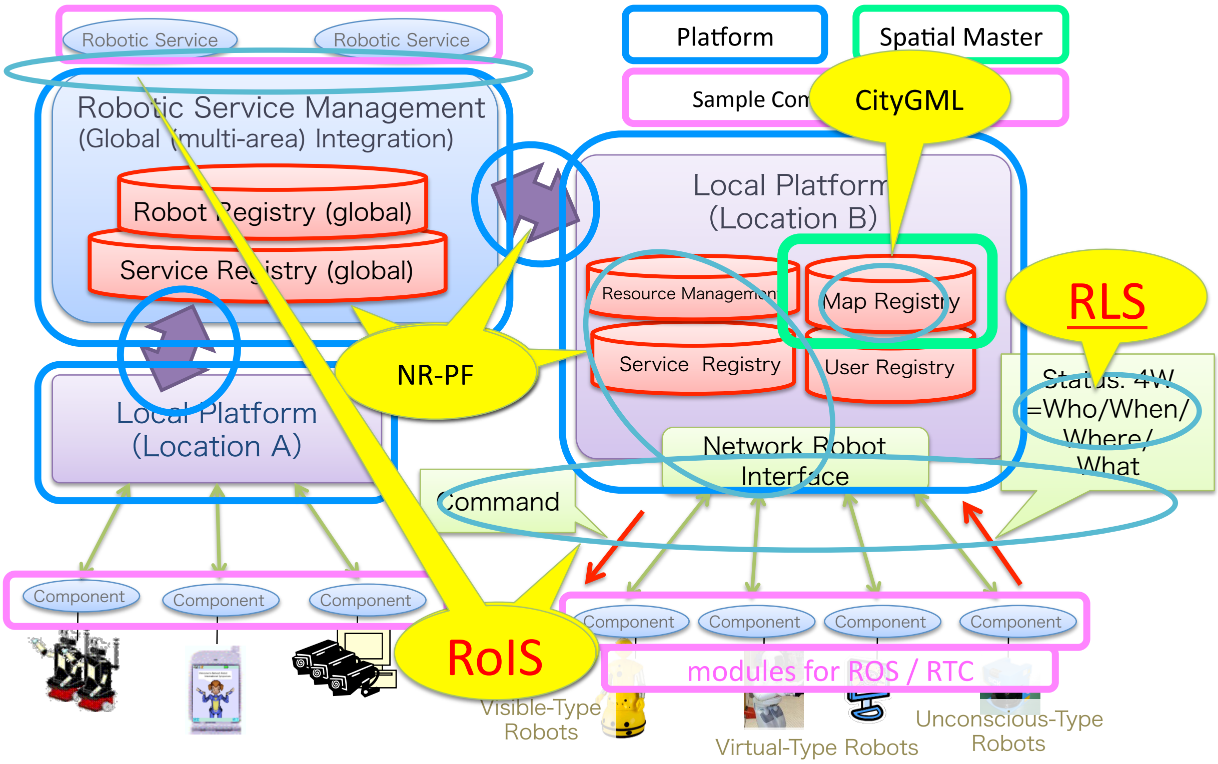
modules for ROS / RTC

Visible-Type Robots

Virtual-Type Robots

Unconscious-Type Robots

Standardization



Robotic Service
Robotic Service

Platform
Spatial Master
Sample Com
CityGML

Robotic Service Management
(Global (multi-area) Integration)
Robot Registry (global)
Service Registry (global)

Local Platform
(Location B)
Resource Management
Map Registry
Service Registry
User Registry
Network Robot Interface

RLS
Status: 4W
=Who/When/
Where/
What

NR-PF

Local Platform
(Location A)

Command

Component
Component
Component

RoIS

Component
Component
Component
Component
modules for ROS / RTC

Visible-Type Robots
Virtual-Type Robots
Unconscious-Type Robots

OGC CityGML

- OGC: Open Geospatial Consortium
- CityGML: City Geography Markup Language
  - An open data model and XML-based format for the storage and exchange of virtual 3D city models
  - CityGML 1.0 (2008/08) → CityGML 2.0 (2012/04)

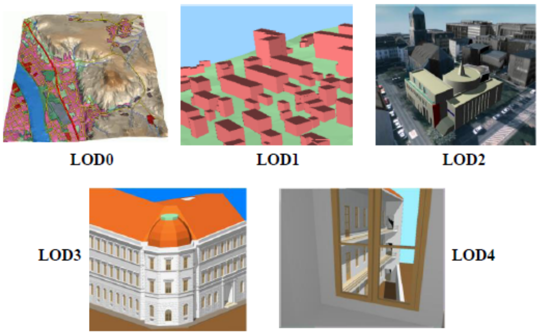
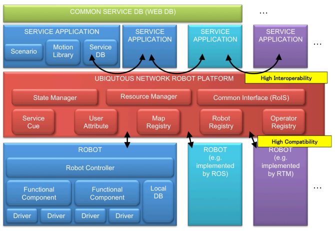


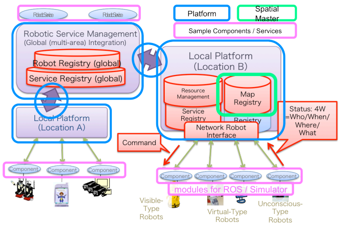
Fig. 2: The five levels of detail (LOD) defined by CityGML (source: IGG Uni Bonn)

from http://portal.opengeospatial.org/files/?artifact_id=47842

ITU-T SG16/Q25 F. USN-NRP

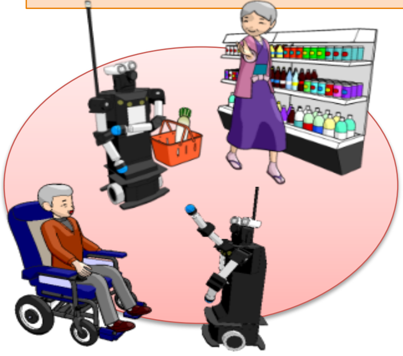
- Functional requirements for Network Robot Platform
- ITU-T SG16 Q25(USN: Ubiquitous Sensor Network)
- Originally proposed as NRP (Network Robot Platform) then expanded to multi-location UNR architecture.
- Expected to reach “concent” in Jan. 2013.

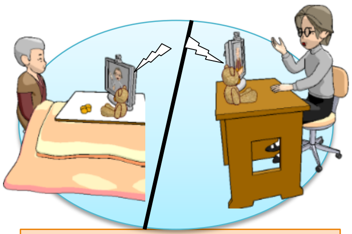

COMMON SERVICE DB (WEB DB)
…
SERVICE APPLICATION
Scenario
Motion Library
Service DB
SERVICE APPLICATION
SERVICE APPLICATION
SERVICE APPLICATION
…
UBIQUTOUS NETWORK ROBOT PLATFORM
High Interoperability
State Manager
Resource Manager
Common Interface (RoIS)
Service Cue
User Attribute
Map Registry
Robot Registry
Operator Registry
High Compatibility
ROBOT
Robot Controller
Functional Component
Functional Component
Local DB
Driver
Driver
Driver
Driver
ROBOT (e.g. implemented by ROS)
ROBOT (e.g. implemented by RTM)
…


Platform
Spatial Master
Sample Components / Services
Robotic Service Management
(Global (multi-area) Integration)
Robot Registry (global)
Service Registry (global)
Local Platform (Location A)
Local Platform (Location B)
Resource Management
Map Registry
Service Registry
Registry
Network Robot Interface
Command
Status: 4W =Who/When/ Where/ What
Component
Component
Component
Component
Component
Component
Component
modules for ROS / Simulator
Visible-Type Robots
Virtual-Type Robots
Unconscious-Type Robots

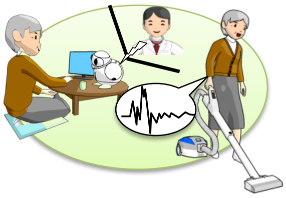
Robotic Services

- Shop guidance
- Shopping support
- Touring support

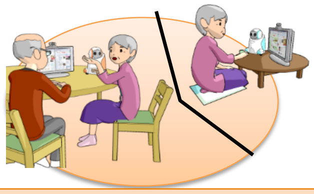




Active hearing

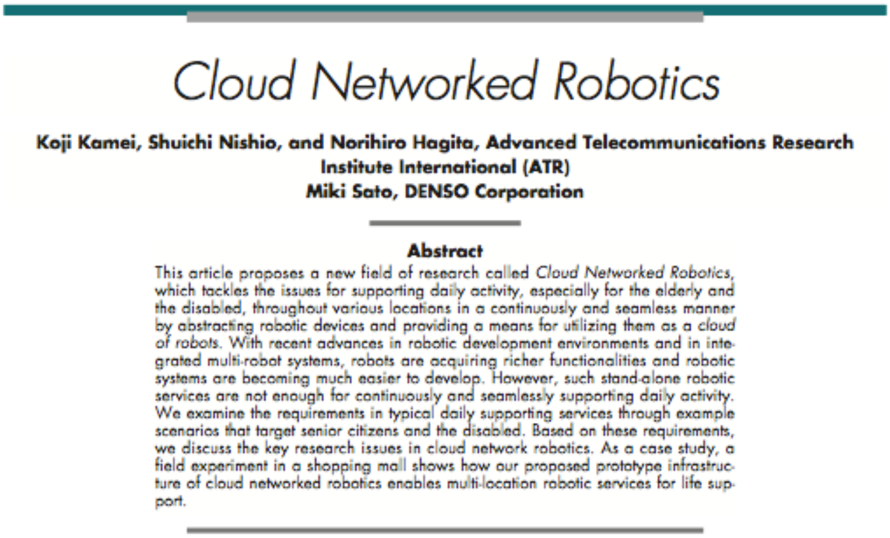


Health care



Community building support

Cloud Networked Robotics



Cloud Networked Robotics

Koji Kamei, Shuichi Nishio, and Norihiro Hagita, Advanced Telecommunications Research Institute International (ATR)
Miki Sato, DENSO Corporation

Abstract

This article proposes a new field of research called *Cloud Networked Robotics*, which tackles the issues for supporting daily activity, especially for the elderly and the disabled, throughout various locations in a continuously and seamless manner by abstracting robotic devices and providing a means for utilizing them as a *cloud of robots*. With recent advances in robotic development environments and in integrated multi-robot systems, robots are acquiring richer functionalities and robotic systems are becoming much easier to develop. However, such stand-alone robotic services are not enough for continuously and seamlessly supporting daily activity. We examine the requirements in typical daily supporting services through example scenarios that target senior citizens and the disabled. Based on these requirements, we discuss the key research issues in cloud network robotics. As a case study, a field experiment in a shopping mall shows how our proposed prototype infrastructure of cloud networked robotics enables multi-location robotic services for life support.

IEEE Network Magazine, vol. 26, no. 3, pp. 28-34, May 2012
Special Issue on Machine and Robotic Networking
http://dx.doi.org/10.1109/MNET.2012.6201213

UNR Platform Alpha Release

- URL
  - http://www.irc.atr.jp/std/UNR-Platform.html
- Alpha Release includes…
  - Platform System
  - Spatial Master Database System
  - Sample Programs
    - Sample Component and Service
    - Sample Scenario for Component Allocation
  - Documents
    - User Guide (How to setup and execute sample programs)
    - Programming Guide (How to use API libraries)
    - Technical Documents (Class Diagram, Sequence Diagram)

Open Source

Dynamic Deployment and Configuration Standard for Robotic Technology Component:

DDC4RTC

Noriaki Ando, AIST (DDC4RTC FTF Co-chair)
Seungwoog Jung, ETRI (DDC4RTC FTF Co-chair)

Overview

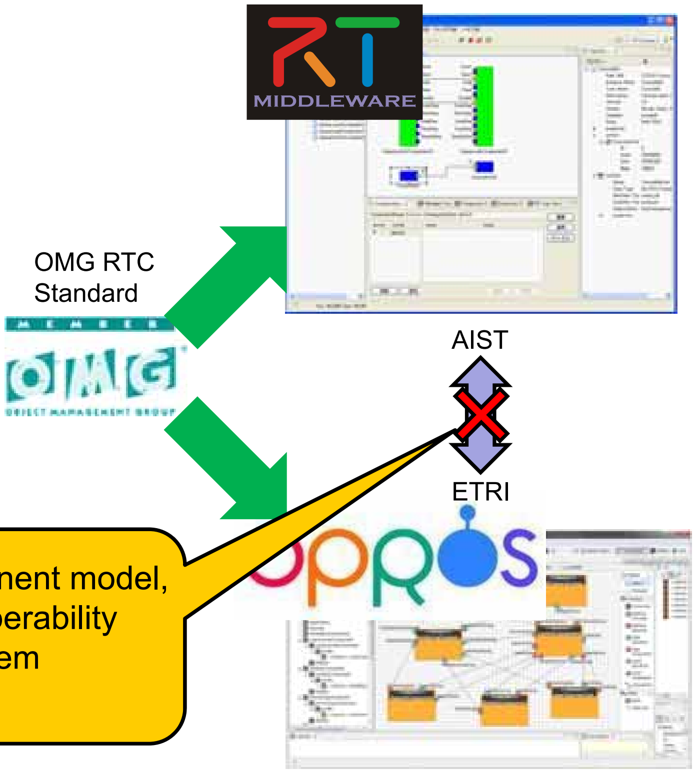
- RTC specification and its implementation
- Motivation
- DDC4RTC specification
  - RTC Specific features
  - ApplicationSupervisor
- Conclusion

OMG RTC Family

Name	Vendor	Feature
OpenRTM-aist	AIST	C++, Python, Java
OpenRTM.NET	SEC	.NET(C#,VB,C++/CLI, F#, etc..)
miniRTC, microRTC	SEC	RTC implementation for CAN･ZigBee based systems
RTMSafety	SEC, AIST	Functional safety standard (IEC61508) capable RTM implementation
RTC CANOpen	SIT, CiA	Standard for RTC mapping to CANOpen by CiA (Can in automation) and implementation by SIT
PALRO	Fuji Soft	C++ PSM implementation for small humanoid robot
OPRoS	ETRI	Developed by Korean national project
GostaiRTC	GOSTAI, THALES	C++ PSM implementation on URBI
Honda R&D RTM	Honda R&D	C++, Python. FSM Component.

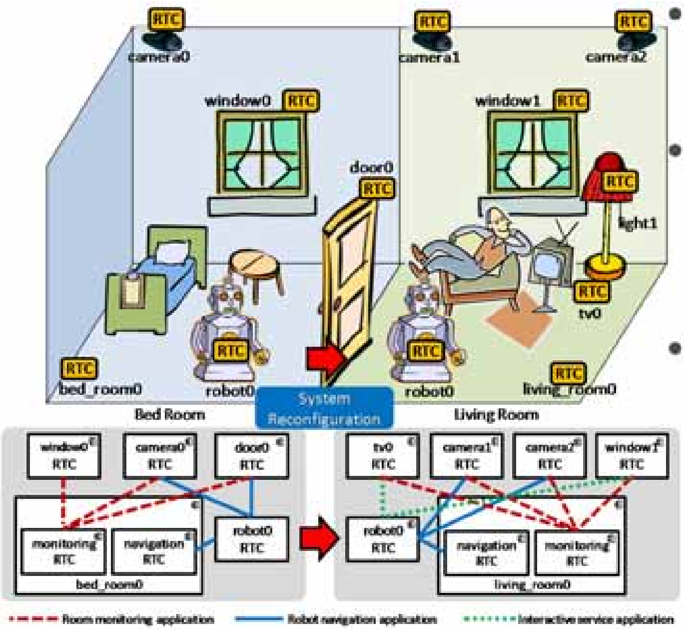
Background

- Component model standard and implementations
  - OpenRTM-aist and OPRoS
- No deployment standard for RTC



Same component model, but no interoperability between system description

Motivation



- Many RTCs are distributed spatially
- Systems would be constructed as RTCs aggregation
- System structure should be changed according to the environmental changes in run-time

---

DDC4RTC Specification

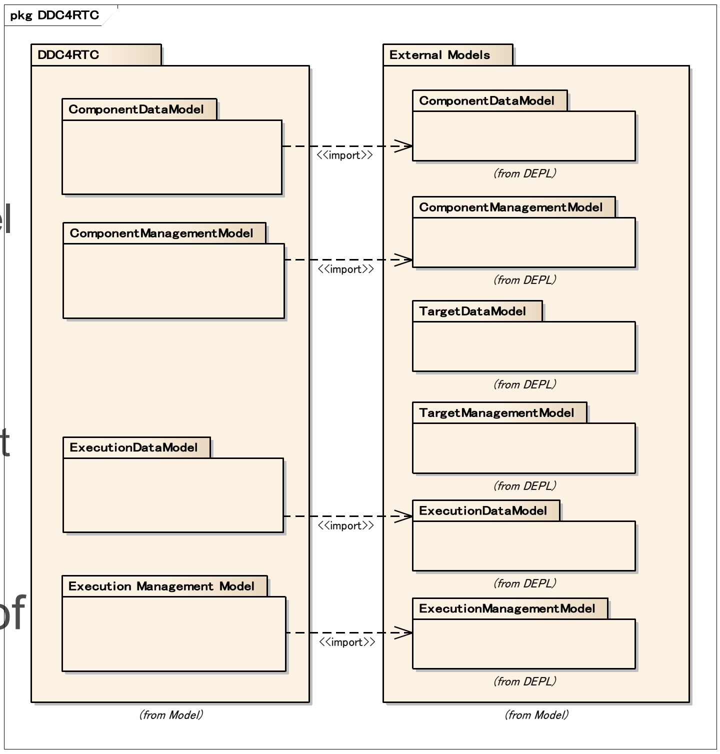
- RFP: Minneapolis meeting, Jun. 2010
  - mars/10-06-16 (Deployment and Dynamic Configuration (DDC) of Robotic Technology Components (DDC4RTC) RFP
- Submitters: ETRI, AIST
- Initial Submissions: Santa Clara Meeting Dec. 2010
- Approved by AB and TC: Jun. 2012

DEPL RTC + SupervisorFSM = DDC4RTC

- DEPL: Deployment and Configuration of Component-based Distributed Applications Specification
- RTC: Robotic Technology Component specification

DDC4RTC Packages

- Consists of four packages
  - Component Data Model
  - Component Management Model
  - Execution Data Model
  - Execution Management Model
- Each package inherits same name package of DEPL specification.

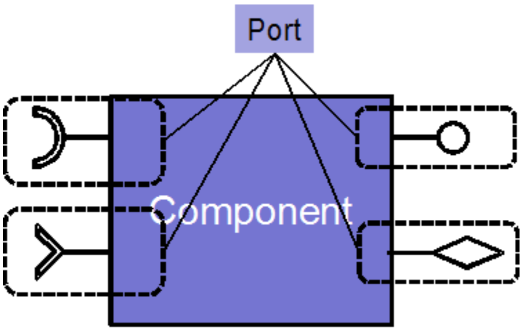


Component Data Model

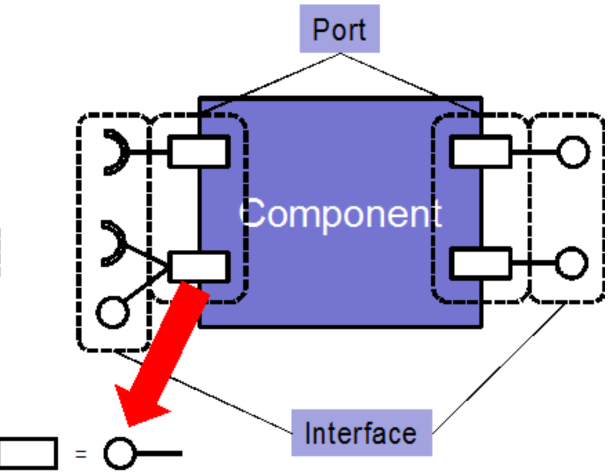
Port in DEPL and Port in RTC

Port models in DEPL and RTC are different

Port and Component in DEPL

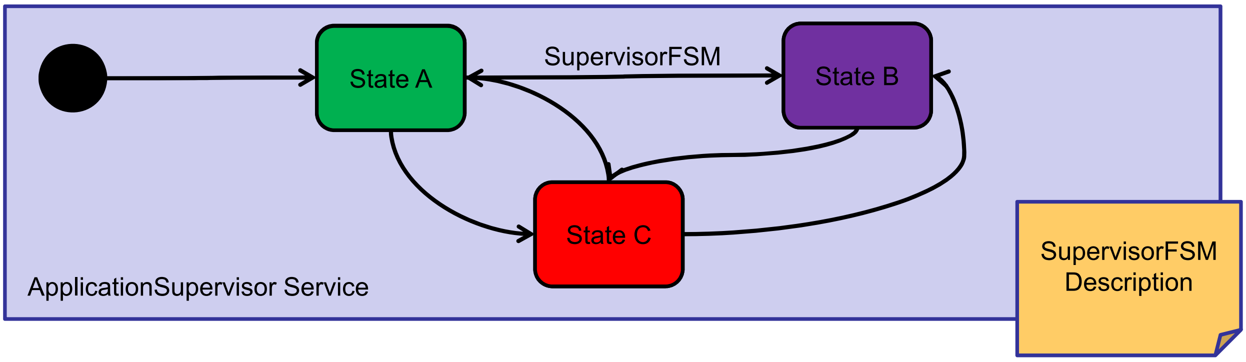


Port and Component in RTC



Port itself is a service (RTC::PortService)

Application Supervisor



SupervisorFSM

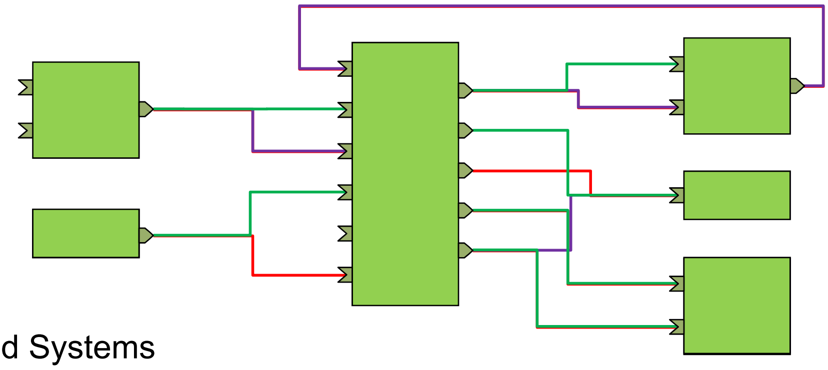
State A

State B

State C

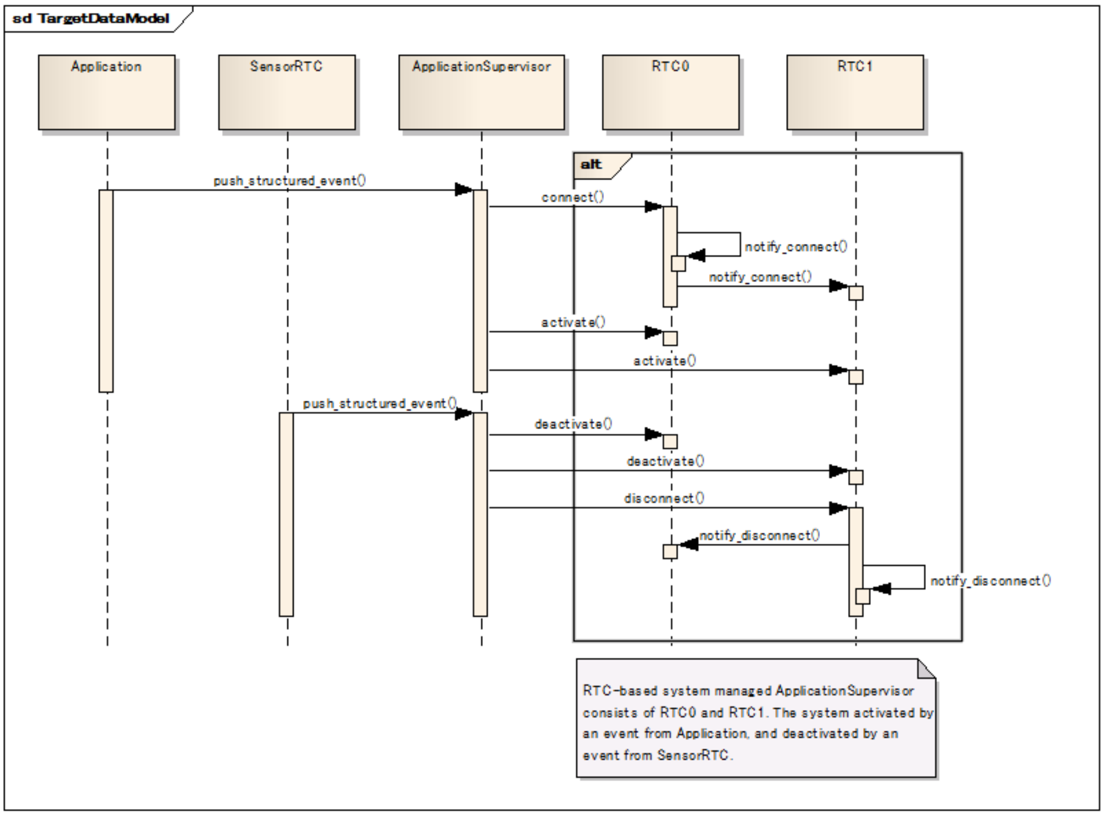
ApplicationSupervisor Service

SupervisorFSM Description



RTC based Systems

Behavior of ApplicationSupervisor and RTCs.



RTC-based system managed ApplicationSupervisor consists of RTC0 and RTC1. The system activated by an event from Application, and deactivated by an event from SensorRTC.

Conclusion

- A dynamic deployment and configuration standard: DDC4RTC was introduced.
  - Now finalization phase in OMG
  - FTF report will be submitted next June?, and the specification will be in public 2014.

- It is based on DEPL and RTC specifications in OMG.
- SupervisorFSM and ApplicationSupervisor is added for dynamic systems
- By reusing existing standard specification, most of parts could be shared and extension parts could be minimized.

Using SysML in a RTC-based Robotics Application : a case study with a demo

Kenji Hiranabe(Change Vision, Inc)
Noriaki Ando (AIST)

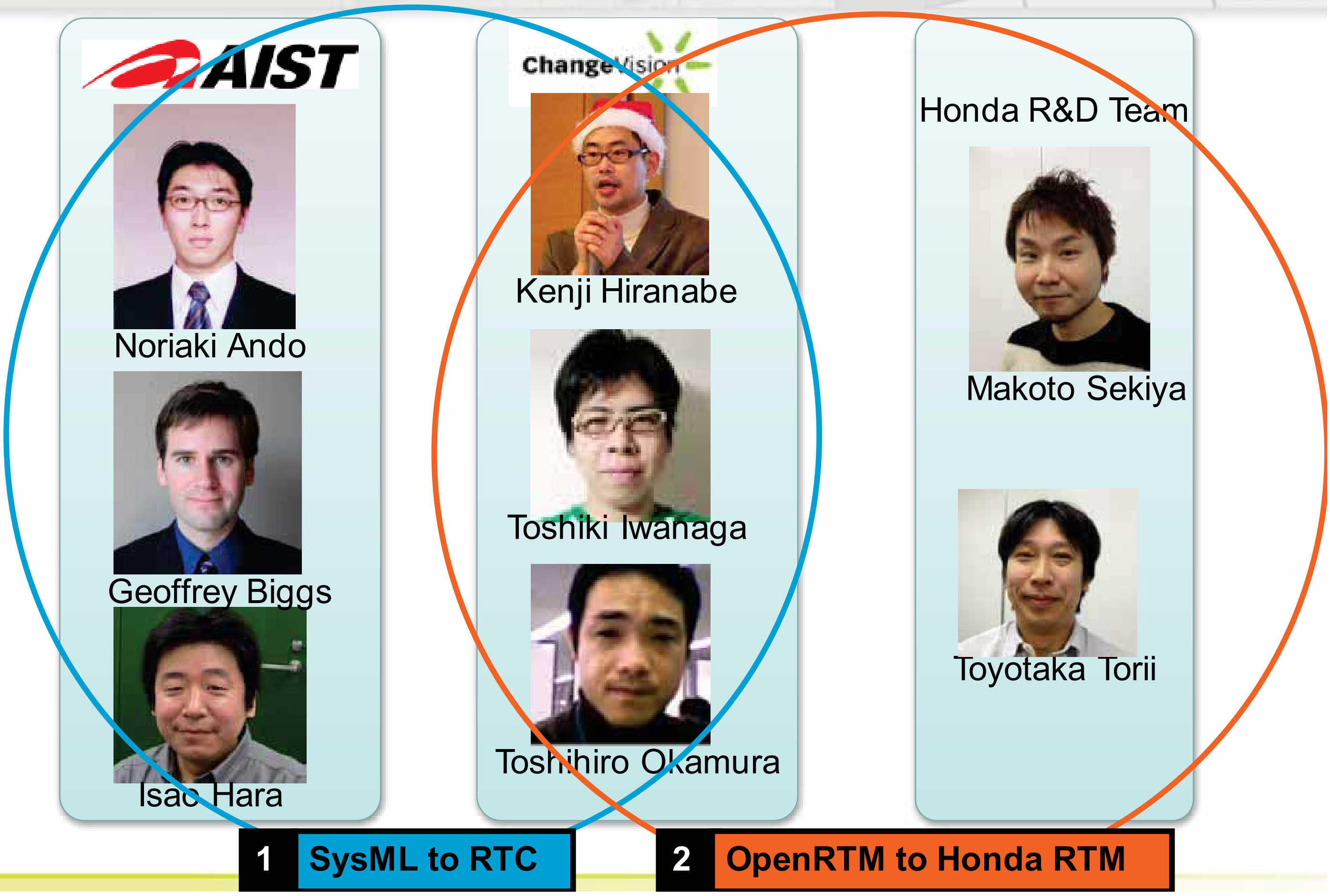
Agenda

- Introduction
- Background and Goals
- Problem
- Analysis and Design via Demo
- Conclusion
- Future Ideas

Who am I ?

- Kenji Hiranabe, Change Vision, Inc.(maker of Astah)
- Astah is a UML editor popular in Japan
  - http://astah.net/
- Astah/SysML
  - Newly developed
  - Focused on “Usability” and “Web collaboration”
- RTC plug-in
  - Plug-in for Astah/SysML to generate RTC.xml to OpenRTM

Project members



AIST

Noriaki Ando

Geoffrey Biggs

Isac Hara

ChangeVision

Kenji Hiranabe

Toshiki Iwanaga

Toshihiro Okamura

Honda R&D Team

Makoto Sekiya

Toyotaka Torii

1 SysML to RTC

2 OpenRTM to Honda RTM

Background and Goals

- Evaluate how SysML can help design a component(RTC)-based robotic application using a simple problem.

1 SysML to RTC

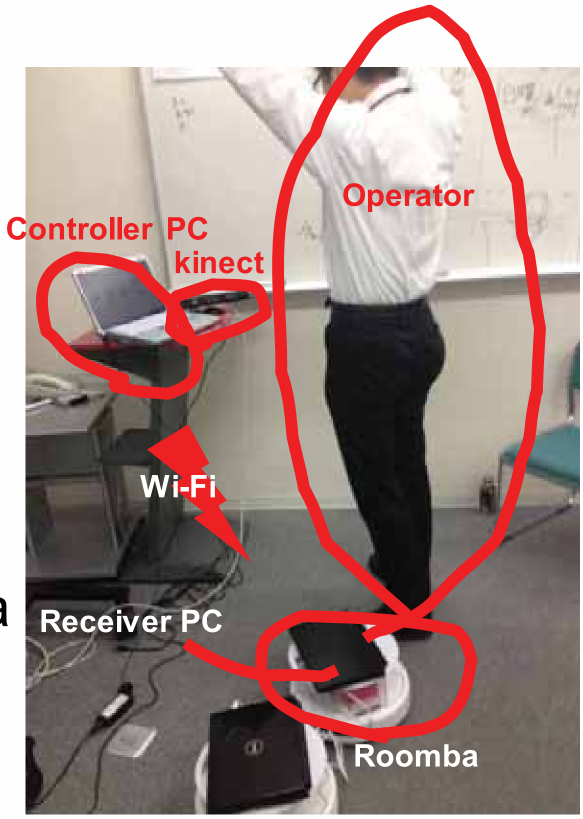
- Try a demonstration test to verify that one common model can work and interoperate on multiple RTM implementations.
  - OpenRTM-aist
  - Honda R&D RTM

2 OpenRTM to Honda RTM

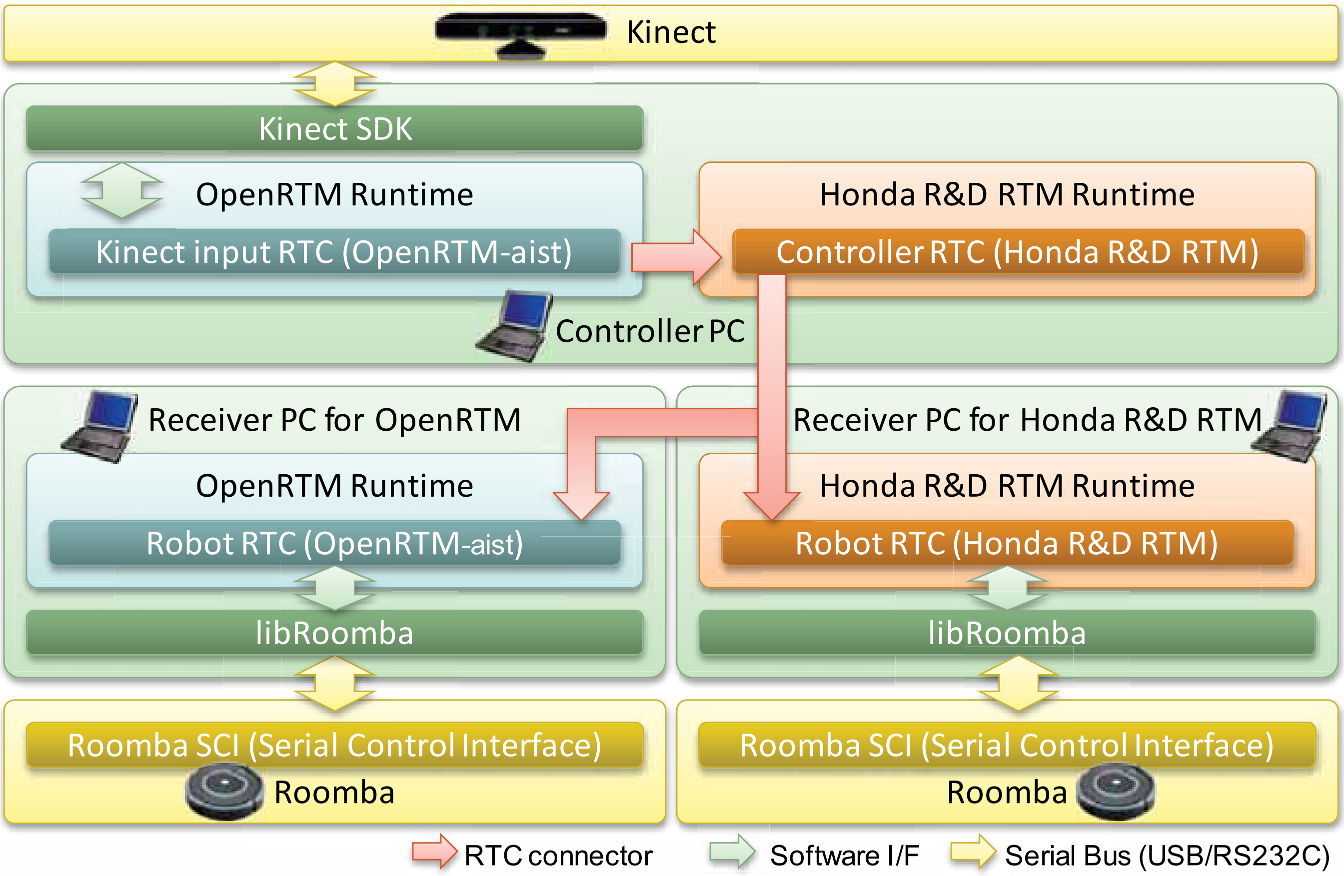
Problem Description

●Demonstrate the movements (Spiral and Back-and-Forth) by controlling multiple autonomous robots from externally. Operator can switch between the autonomous mode and demonstration mode.

●Hardware architecture is already known, we use Roomba with PC that can control it using Wi-Fi and use Kinect to switch the mode.



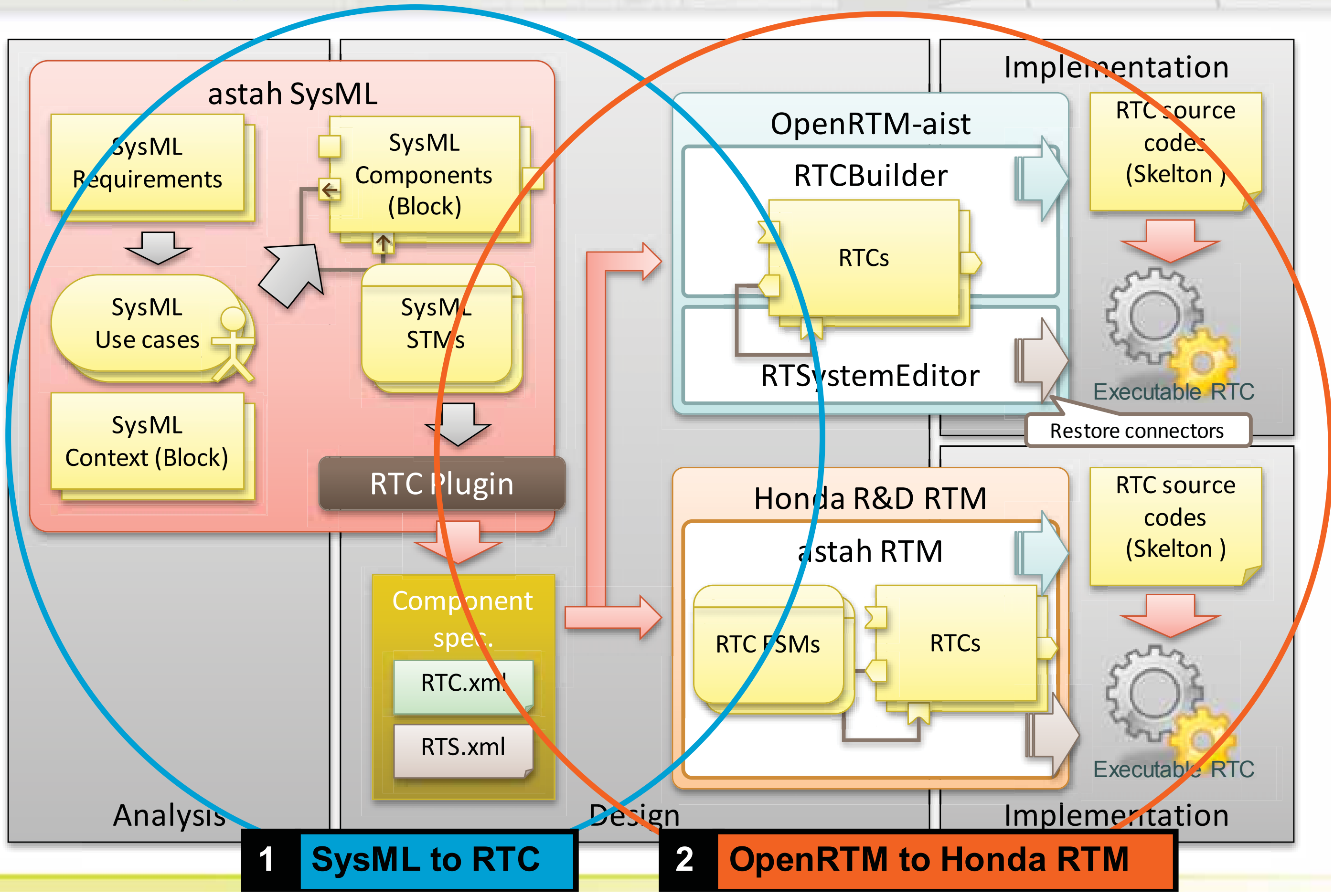
System architecture



OMG RTC Family

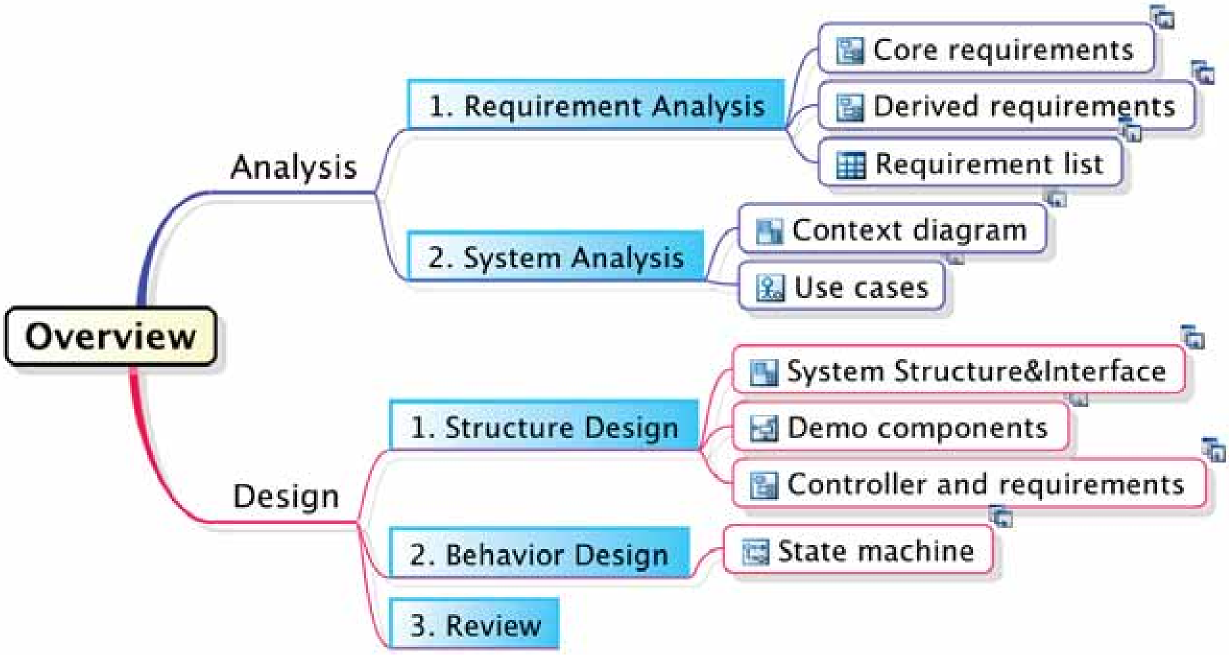
Name	Vendor	Feature
OpenRTM-aist	AIST	C++, Python, Java
OpenRTM.NET	SEC	.NET(C#,VB,C++/CLI, F#, etc..)
miniRTC, microRTC	SEC	RTC implementation for CAN･ZigBee based systems
Dependable RTM	SEC/AIST	Functional safety standard (IEC61508) capable RTM implementation
RTC CANOpen	SIT, CiA	Standard for RTC mapping to CANOpen by CiA (Can in automation) and implementation by SIT
PALRO	Fuji Soft	C++ PSM implementation for small humanoid robot
OPRoS	ETRI	Developed by Korean national project
GostaiRTC	GOSTAI, THALES	C++ PSM implementation on URBI
Honda R&D RTM	Honda R&D	C++, Python. FSM Component.

Process overview



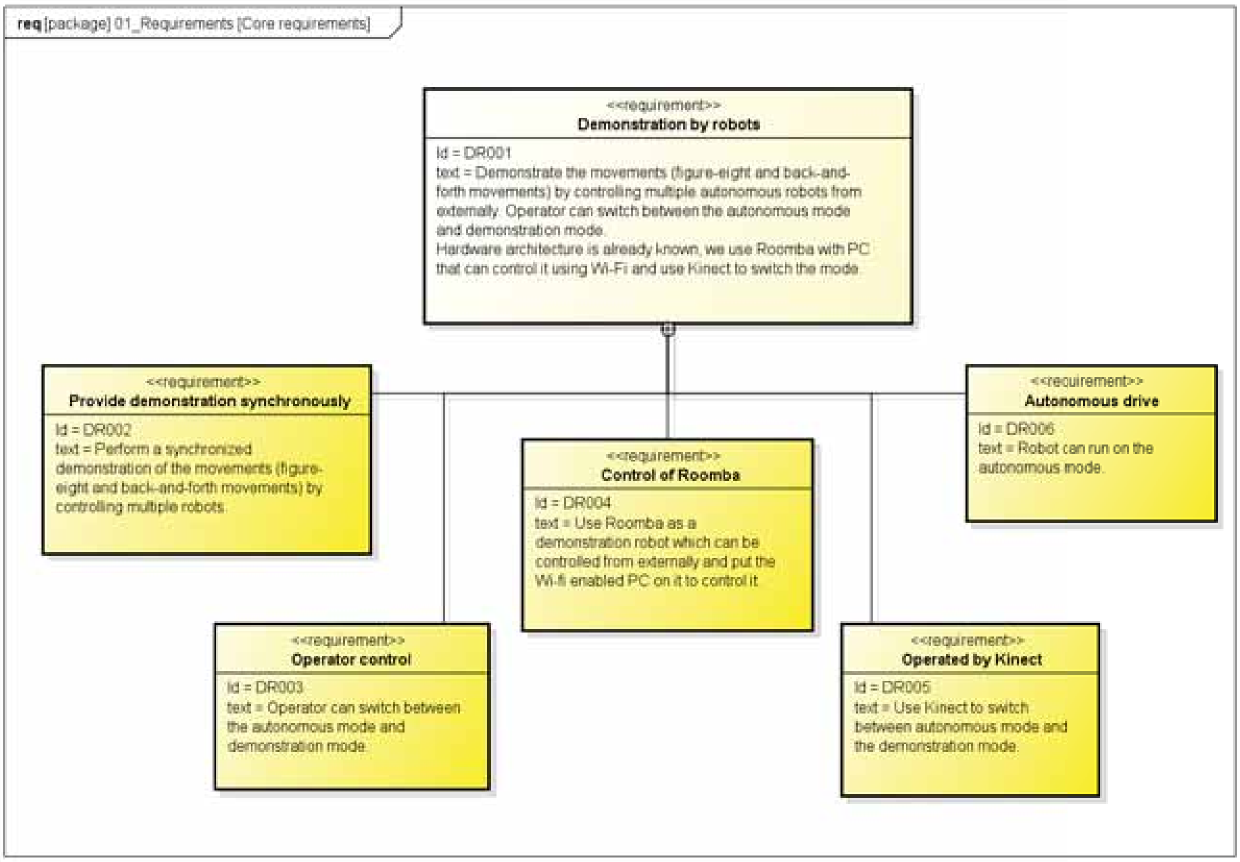
Analysis and Design Diagrams in Astah / SysML

Overview



---

req [Core requirements]



req [package] 01_Requirements [Core requirements]

<<requirement>>
Demonstration by robots
Id = DR001
text = Demonstrate the movements (figure-eight and back-and-forth movements) by controlling multiple autonomous robots from externally. Operator can switch between the autonomous mode and demonstration mode.
Hardware architecture is already known, we use Roomba with PC that can control it using Wi-Fi and use Kinect to switch the mode.

<<requirement>>
Provide demonstration synchronously
Id = DR002
text = Perform a synchronized demonstration of the movements (figure-eight and back-and-forth movements) by controlling multiple robots.

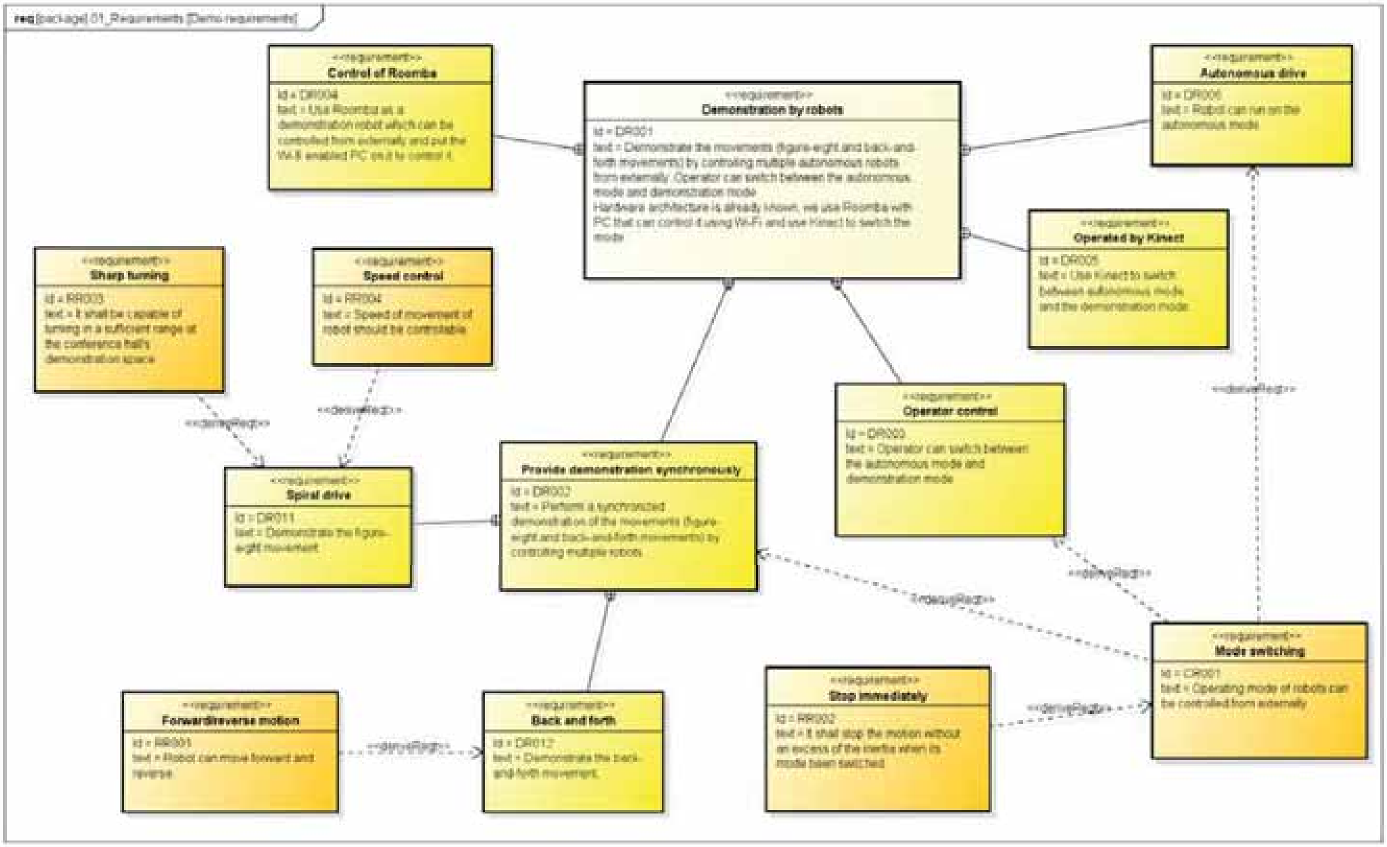
<<requirement>>
Control of Roomba
Id = DR004
text = Use Roomba as a demonstration robot which can be controlled from externally and put the Wi-fi enabled PC on it to control it.

<<requirement>>
Autonomous drive
Id = DR006
text = Robot can run on the autonomous mode.

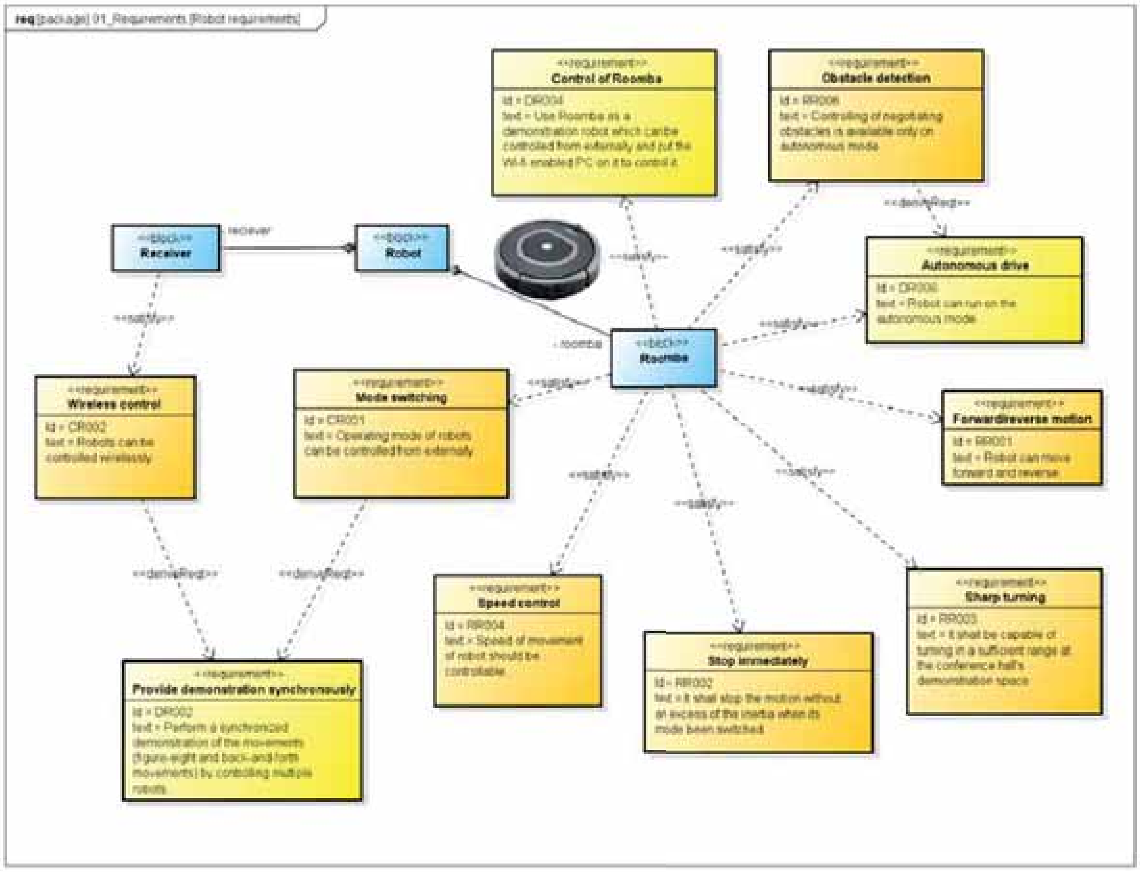
<<requirement>>
Operator control
Id = DR003
text = Operator can switch between the autonomous mode and demonstration mode.

<<requirement>>
Operated by Kinect
Id = DR005
text = Use Kinect to switch between autonomous mode and the demonstration mode.

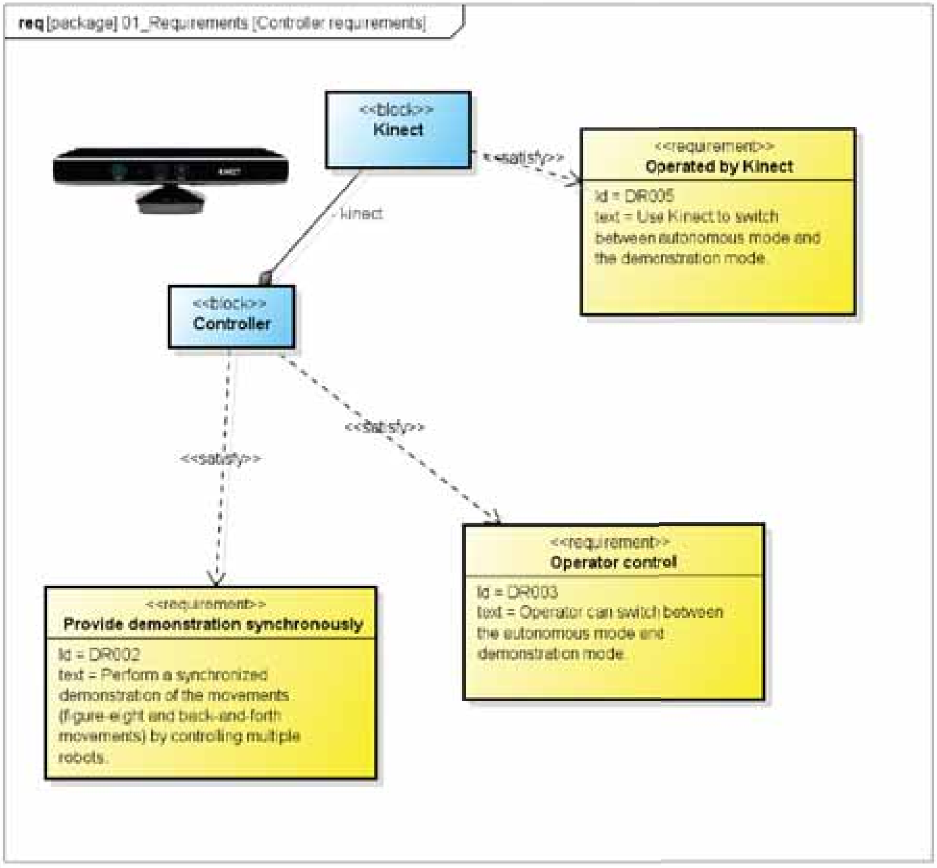
req [Derived Requirements]



req [Robot requirements]



req [Controller requirements]

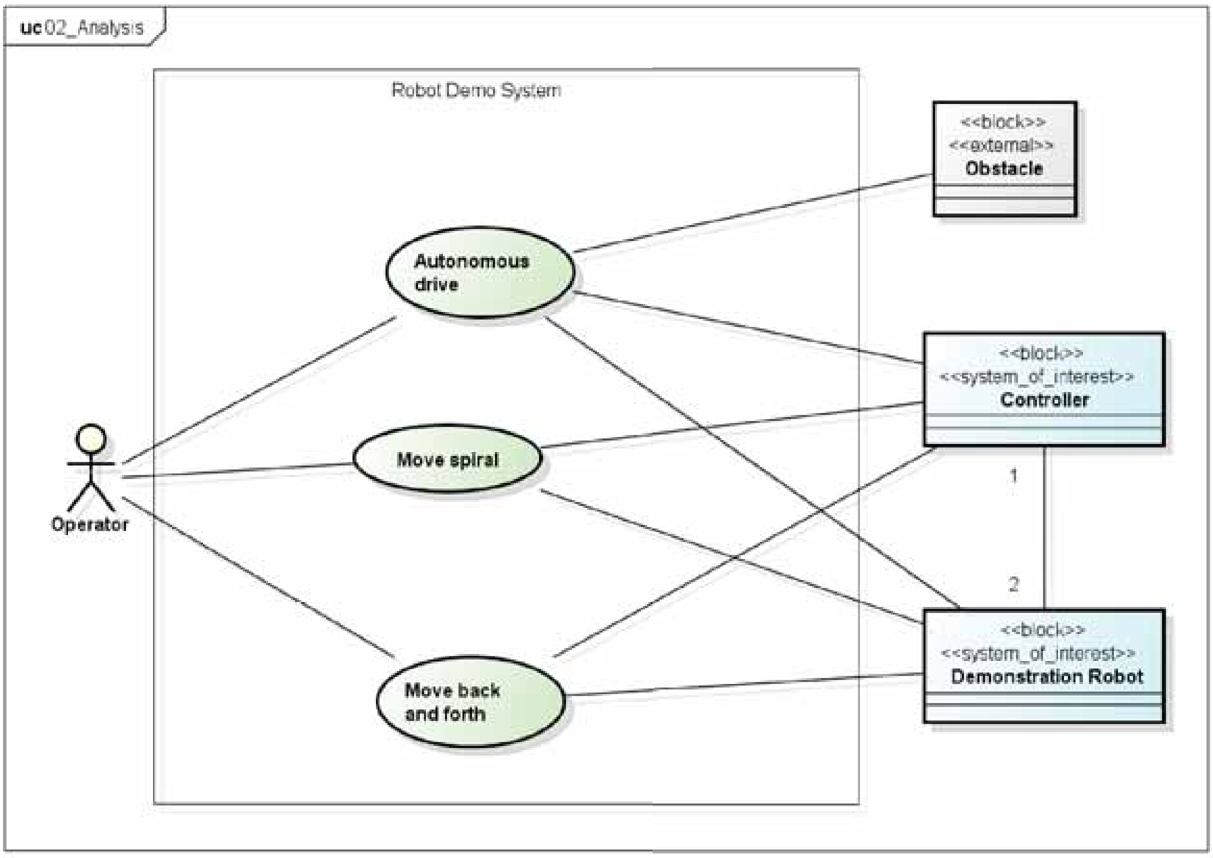


<<requirement>>
Operated by Kinect
Id = DR005
text = Use Kinect to switch between autonomous mode and the demonstration mode.

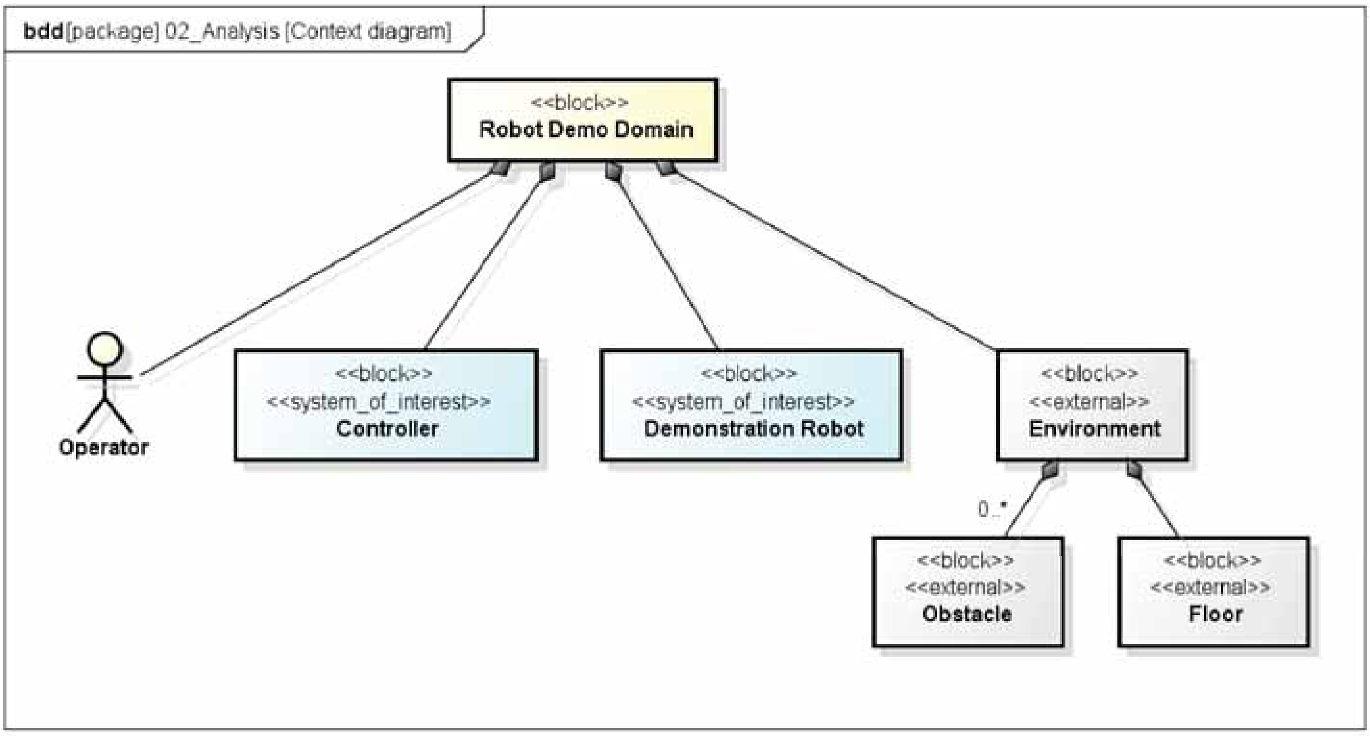
<<requirement>>
Operator control
Id = DR003
text = Operator can switch between the autonomous mode and demonstration mode.

<<requirement>>
Provide demonstration synchronously
Id = DR002
text = Perform a synchronized demonstration of the movements (figure-eight and back-and-forth movements) by controlling multiple robots.

uc [Use cases]

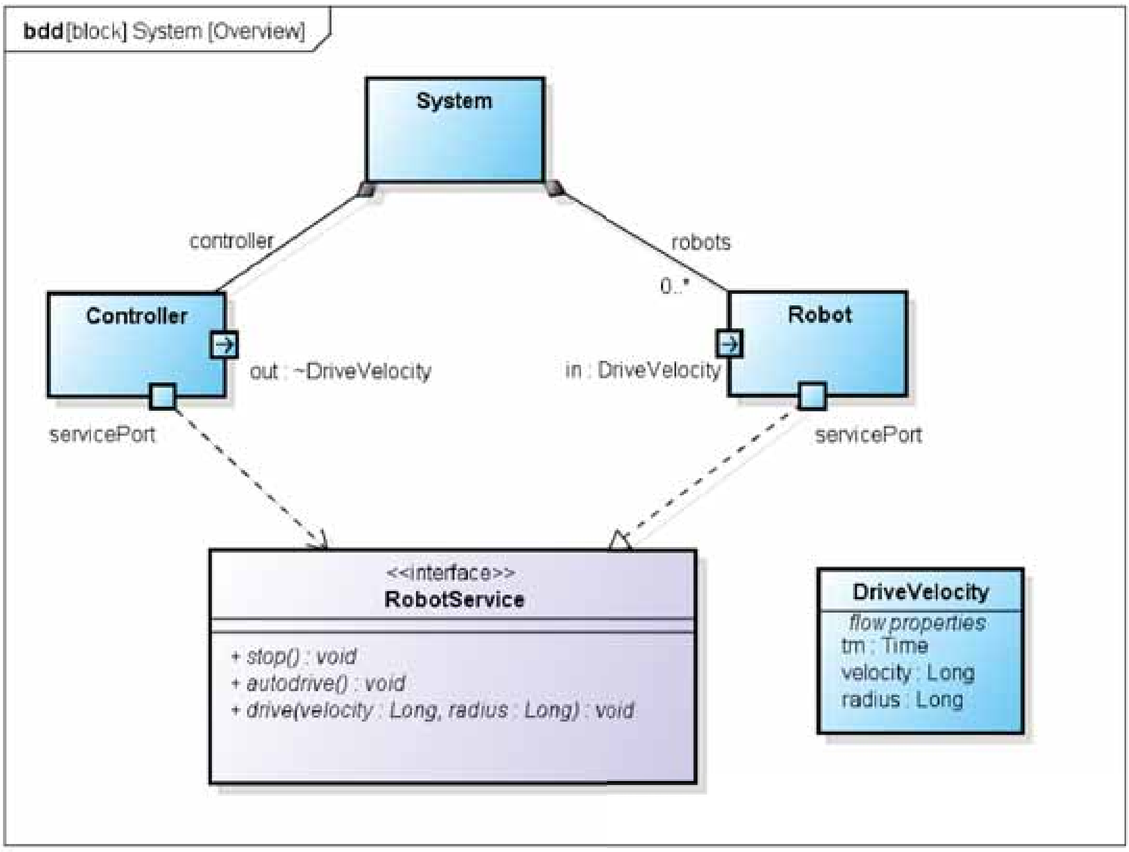


bdd [Context diagram]

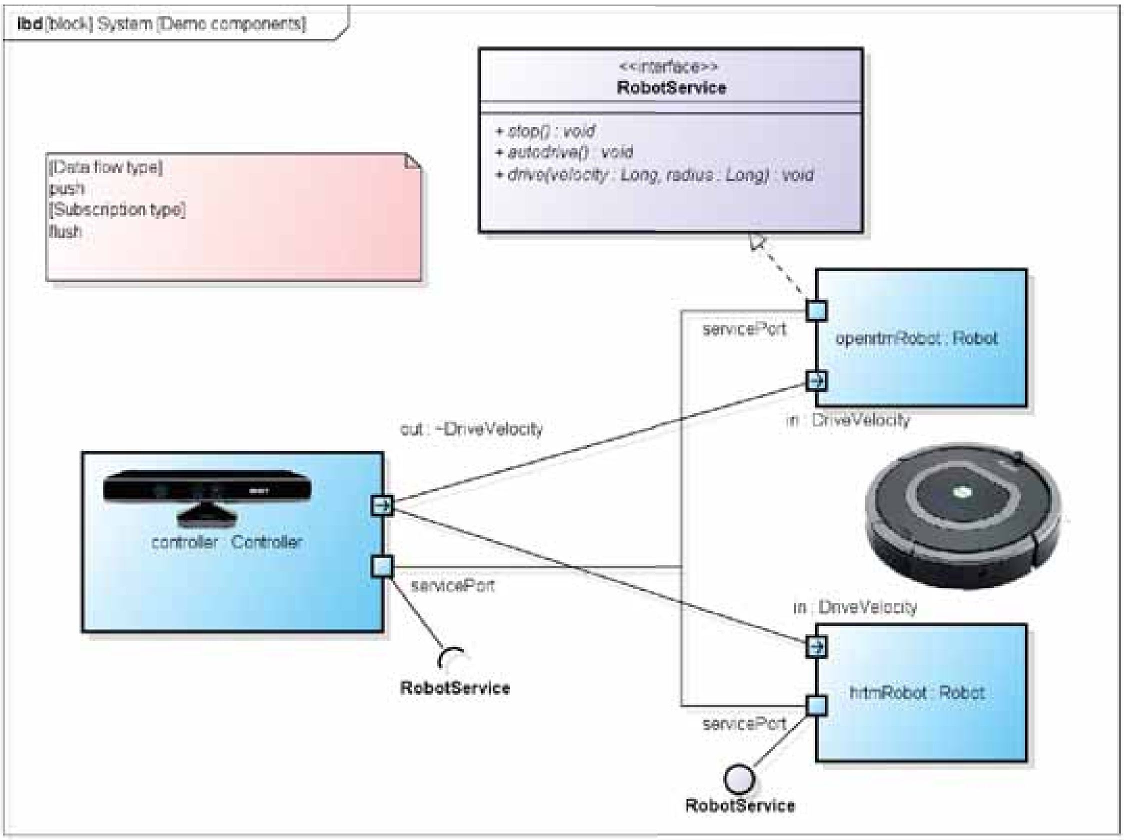


---

bdd [System overview]

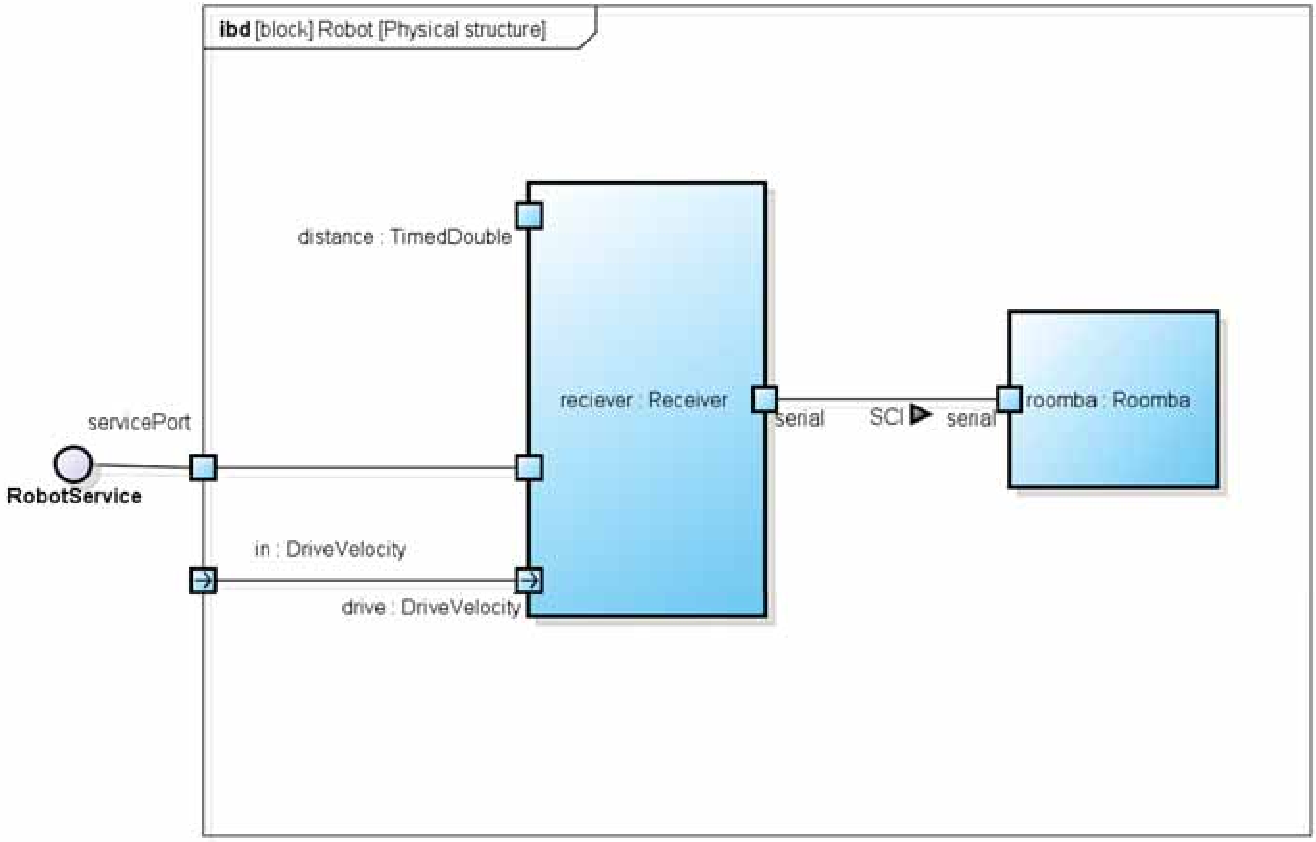


ibd [Demo system components]

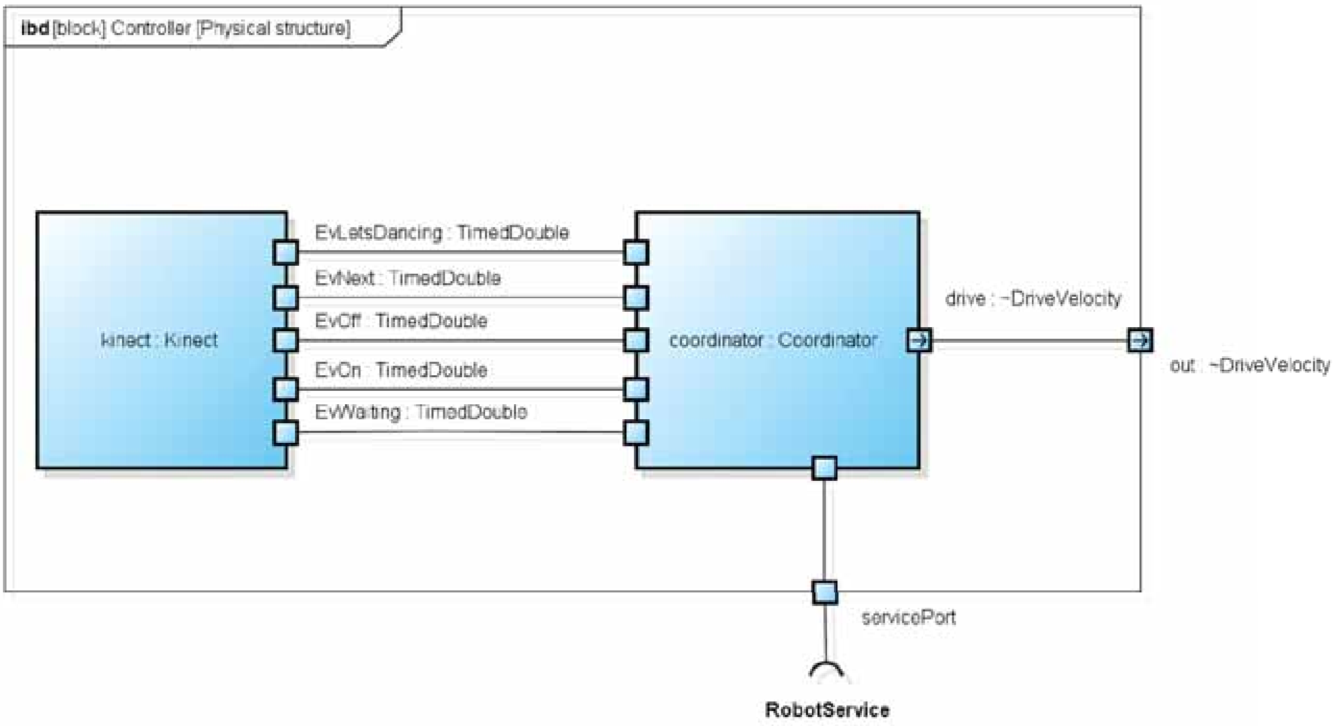


---

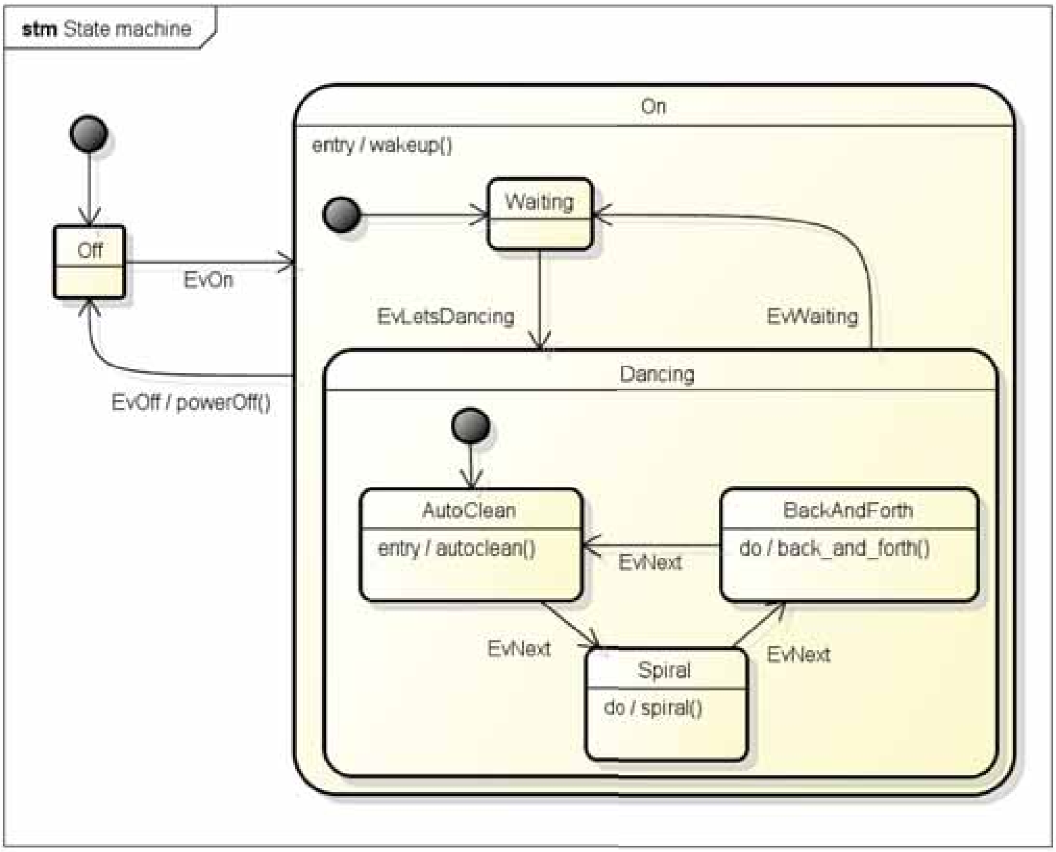
ibd [Physical structure of robot]



ibd [Physical structure of controller]



stm [State machine of controller]



Conclusion

- SysML “Block”s map to “RTC”s nicely.
- <<Satisfy>> relationships between “Requirements” and “Components” can be visualized to show the intentions of components reasonably.
- An Easy-to-use tool(Astah/SysML) boosted effectiveness of modeling.
- Communication between teams worked well using web-based model sharing feature of the tool.

Future Topics

- Real-time aspects into the model
- Relate Safety Case models(Software Assurance Case Model/Safe ML) with SysML models
- SysML Profile for RTC.
- Traceability and impact analysis from/to requirements to components via the tool.

ChangeVision

Thank You !

We are exhibiting the demo, and tools. Please visit us.

AIST


Noriaki Ando


Geoffrey Biggs


Isao Hara

ChangeVision


Kenji Hiranabe


Toshiki Iwanaga


Toshihiro Okamura

Honda R&D Team


Makoto Sekiya


Toyotaka Torii

robotics/2012-12-11

A New Robotic Technology Middleware
and
Robotic Technology Component Interoperability Demonstration

Makoto Sekiya
Honda R&D Co., Ltd.

Today

We are pleased to announce

a NEW

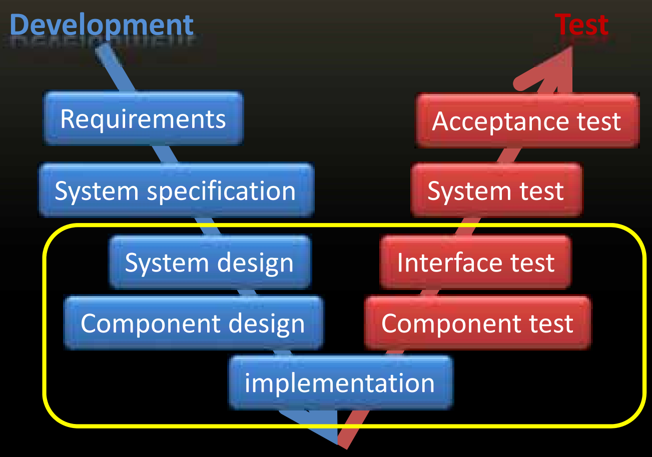
Robotic Technology Middleware

Agenda

- Concept
- Architecture
- Main features
- Demonstration

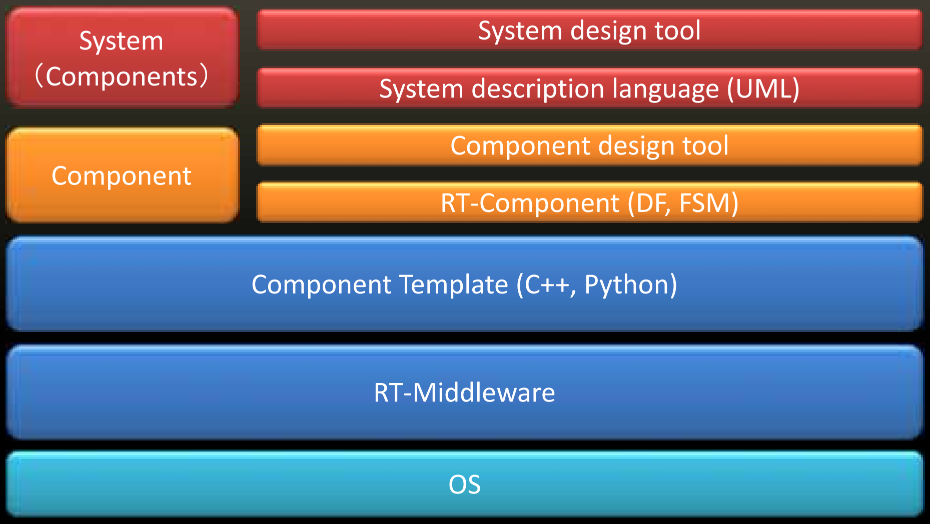
Concept

Bring power of MBD and V-model to real-time robotic software development



Architecture

System is modeled as UML diagram



System（Components）

System design tool

System description language (UML)

Component

Component design tool

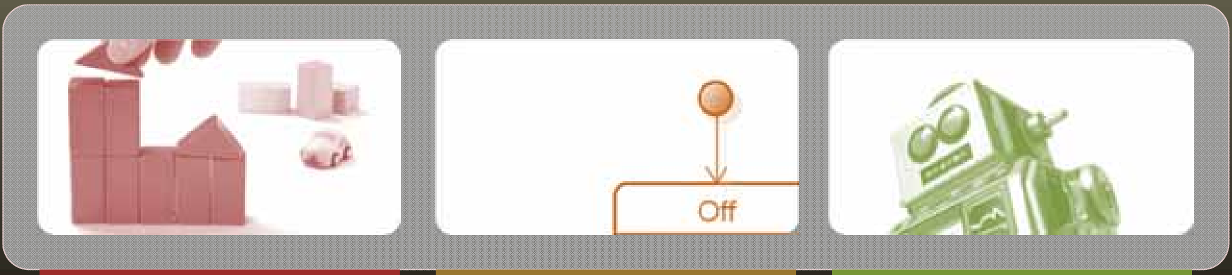
RT-Component (DF, FSM)

Component Template (C++, Python)

RT-Middleware

OS

Main Features

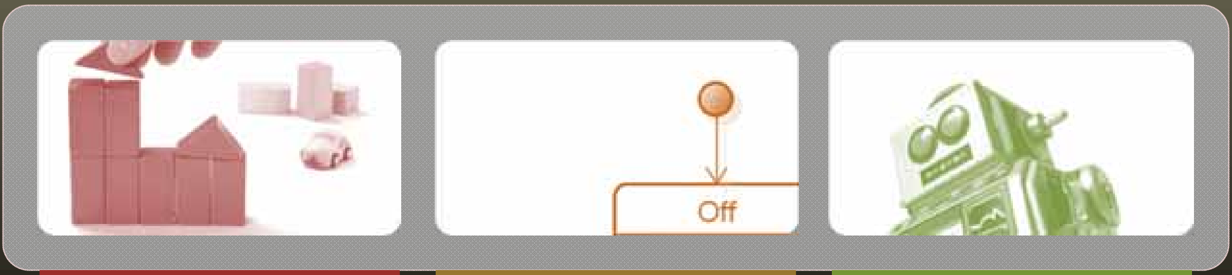


Model-Based Development

State Machine Component

Real-time Middleware

Main Features



Model-Based Development

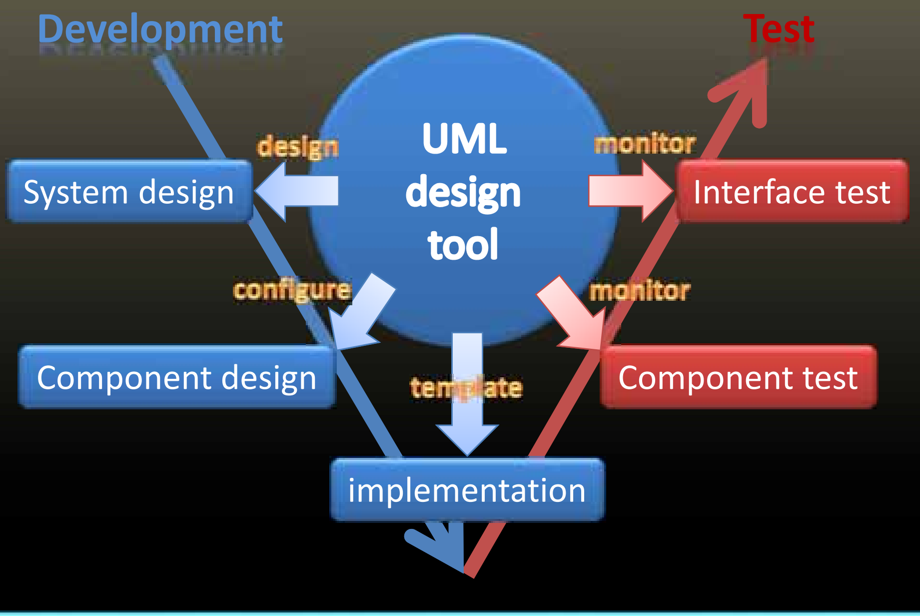
State Machine Component

Real-time Middleware

Model-Based Development

- Users can do with UML editor:
  - Design a system
  - Configure components in the system
  - Generate RT-Component template
  - Monitor components

Model-Based Development



Development

Test

UML design tool

design

System design

configure

Component design

template

implementation

monitor

Interface test

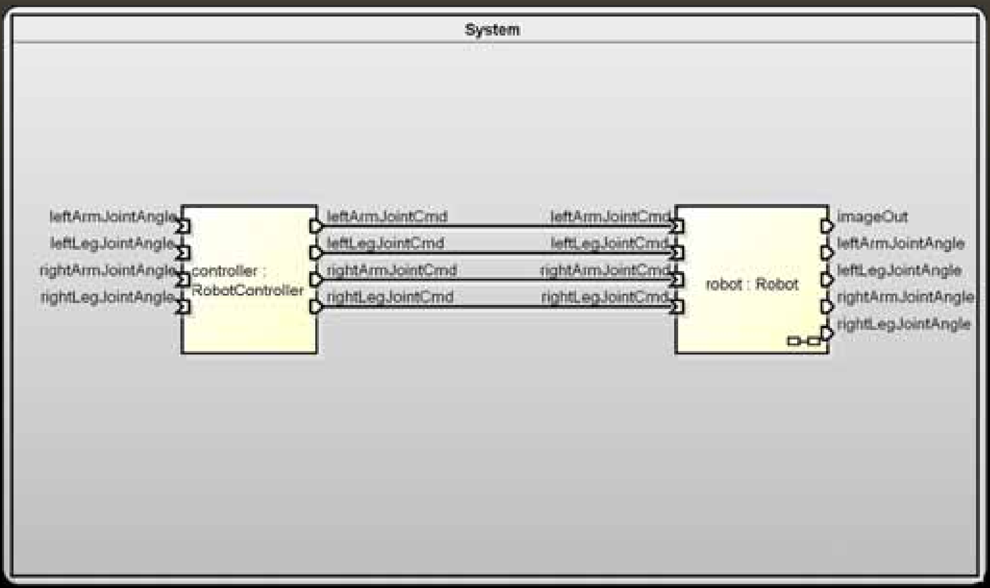
monitor

Component test

UML design tool is the center place of development process

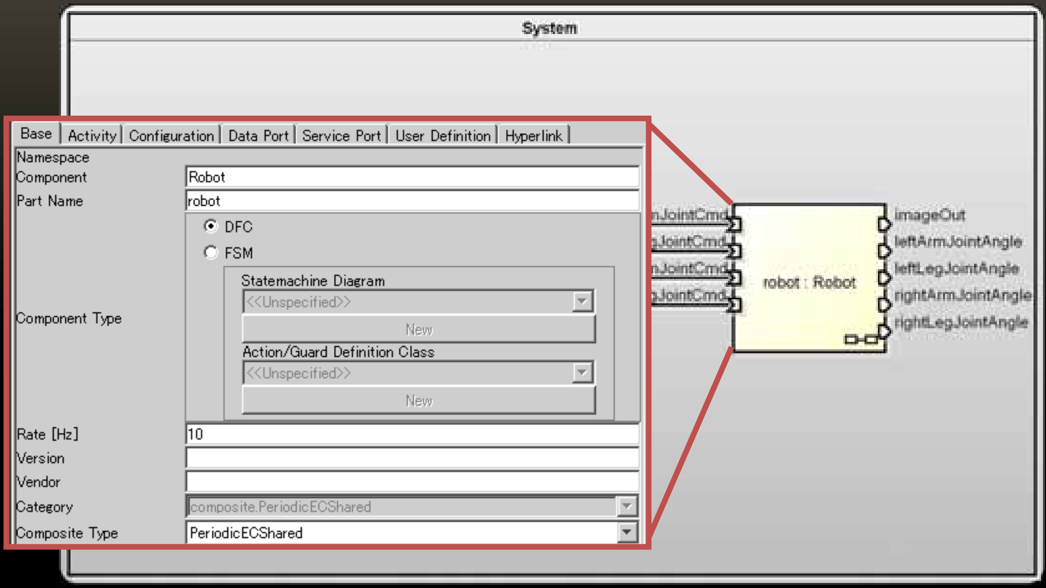
Model-Based Development

- Design a system with UML



Model-Based Development

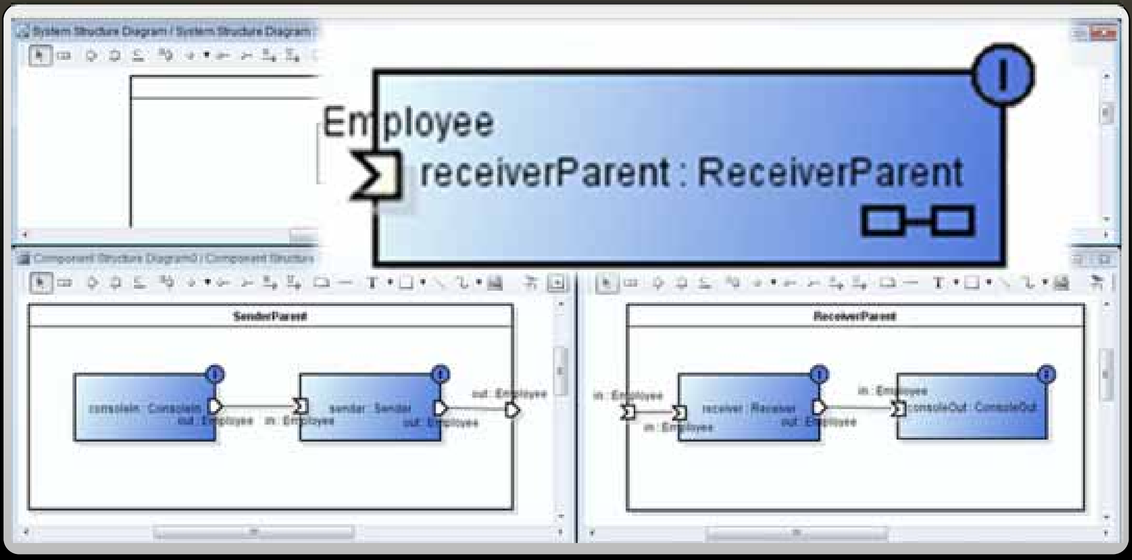
- Configure components in the system



Namespace
Component: Robot
Part Name: robot
Component Type: ● DFC ○ FSM
Statemachine Diagram: <<Unspecified>>
Action/Guard Definition Class: <<Unspecified>>
Rate [Hz]: 10
Version
Vendor
Category: composite.PeriodicECShared
Composite Type: PeriodicECShared

Model-Based Development

- Monitor components in the system



Main Features

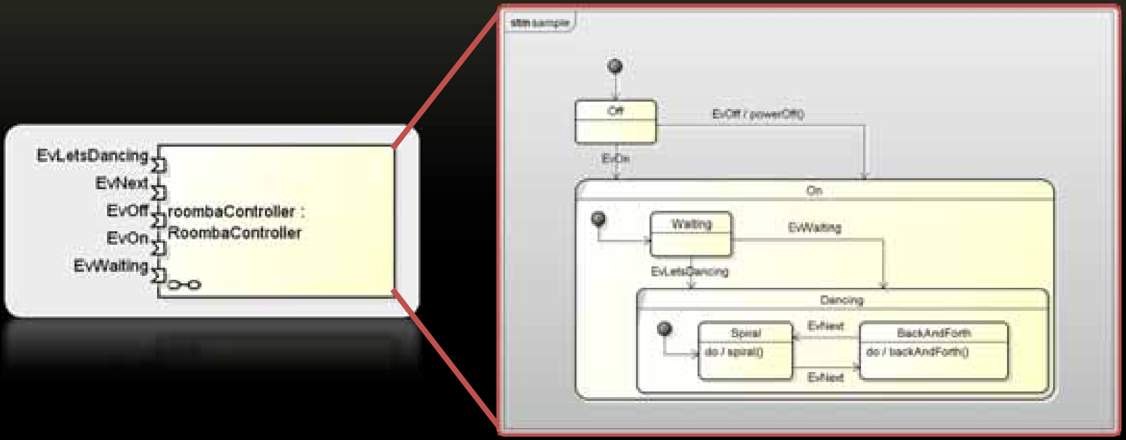
Model-Based Development

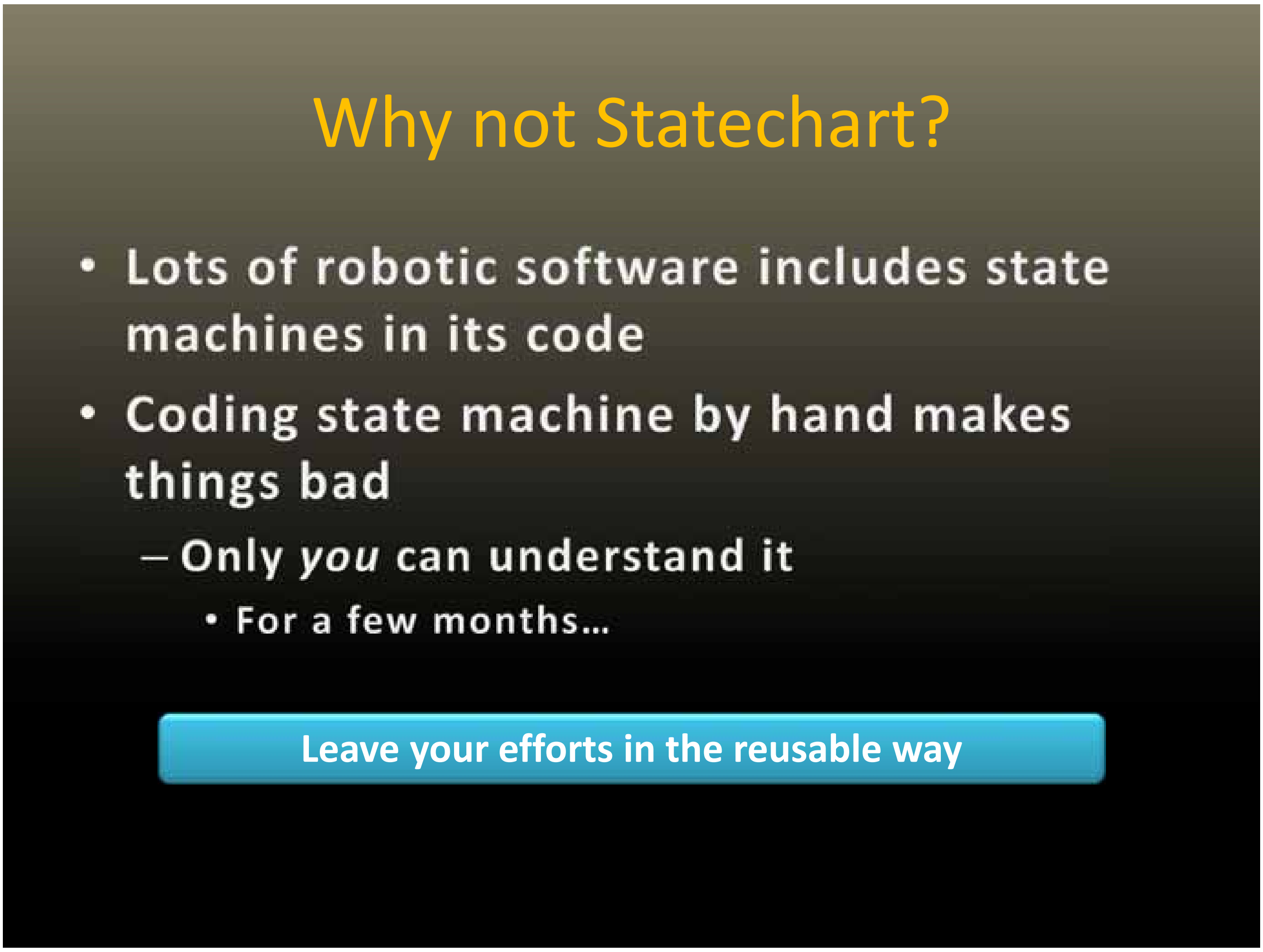
State Machine Component

Real-time Middleware

State Machine Component

- Embedded FSM in Data Flow Component
  - Events are input from data ports
- FSM is modeled with UML Statechart

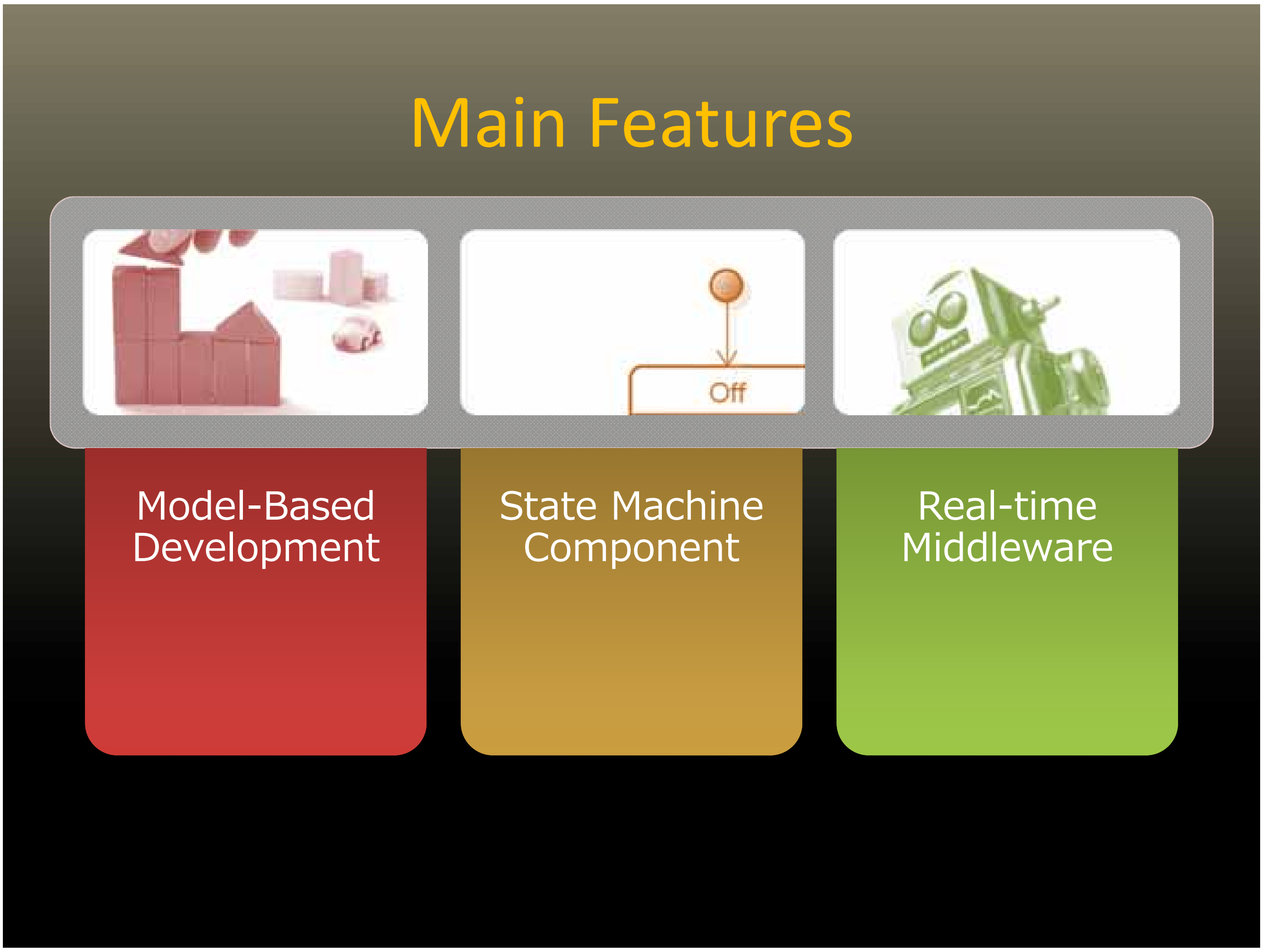




Why not Statechart?

- Lots of robotic software includes state machines in its code
- Coding state machine by hand makes things bad
  - Only *you* can understand it
    - For a few months…

Leave your efforts in the reusable way



Main Features

Model-Based Development

State Machine Component

Real-time Middleware

Real-time Middleware

- Enhancements for RTOS
  - Runtime configuration
    - Fast data port types
    - Bus communication protocols

Full functionality is available

Demonstration

2 Roombas are synchronously changed their action by gestures

- To show *interoperability* of Honda RTM and OpenRTM-aist
  - Honda RTM provides:
    - Action state component (mode changer)
    - Roomba component (controller)
  - OpenRTM-aist provides:
    - Kinect component (UI)
    - Roomba component (controller)

Enjoy it!

robotics/2012-12-12

Implementation of RoIS to robots in ETRI

Su-young Chi, Young-jo Cho
DoHyung Kim, Jaeyeon Lee, Youngwoo Yoon, Ho-Sub Yoon, Jaehong Kim
Electronics and Telecommunications Research Institute
Cognition Technology Research Team

Introduction

- **HRI technology** such as human detection and recognition is **very important** for commercialization of an **intelligent service robot**.

- But, the **performance** of HRI technology of commercial robots is relatively **lower than its importance**. **Why?**

What’s the problem? (1)

1. Lack of effort in HRI component integration

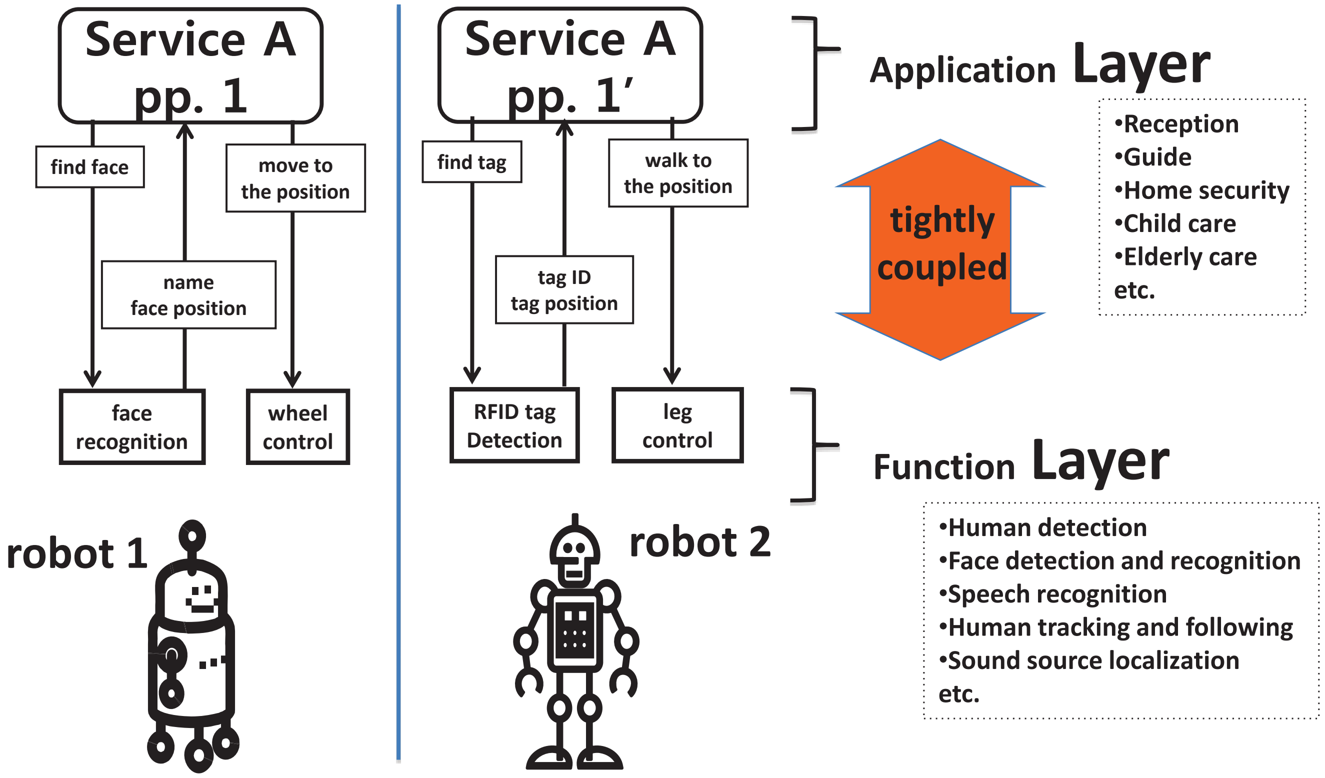
- Many researches have been **concentrated on the enhancement of each HRI core components.**
  - Person detection, face recognition, and so on.
- But, **how to combine unit HRI components effectively** is also **important,** because a HRI service consists of several HRI core components.

What’s the problem? (2)

2. Discontinuity of the recognition processes of HRI components

- **In real life**, **HRI** is bound to **occur continuously**.
- But, **HRI components operate for a short time** of span.
  - Especially when they get requests from an application.

Existing Robotic Service Applications



Service A pp. 1

find face

move to the position

name
face position

face recognition

wheel control

robot 1

Service A pp. 1′

find tag

walk to the position

tag ID
tag position

RFID tag Detection

leg control

robot 2

Application **Layer**

- Reception
- Guide
- Home security
- Child care
- Elderly care

etc.

tightly coupled

Function **Layer**

- Human detection
- Face detection and recognition
- Speech recognition
- Human tracking and following
- Sound source localization

etc.

What’s the problem? (3)

3. Difficulty in adapting HRI components to the real robot.

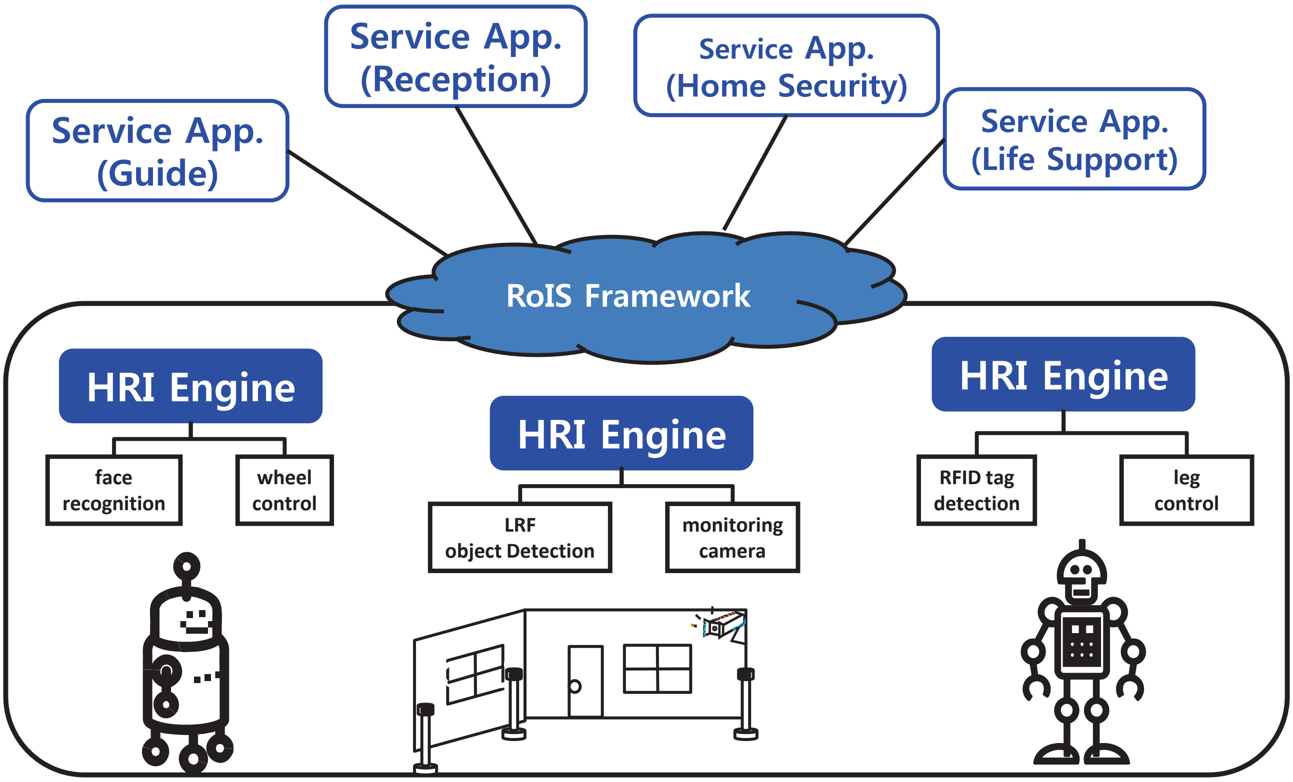
- Current HRI service applications **directly** receive sensor data from the robot and **process the data by using their own HRI components running in them.**
- So, service developers **have burdened** themselves **with works for making the best use of their HRI components.**

The proposed framework: HRIDemon (1)

- The HRIDemon is a **HRI component integration framework** for recognizing users' locations, identities and behaviors in human-robot interaction.

- The HRIDemon features **a constant observation of users** for collecting suitable evidences and **fusion of the diverse components.**
  - It ensure more reliable recognition performance and consequently higher quality of HRI services.

Schematic Picture of RoIS Framework



Service App.
(Reception)

Service App.
(Home Security)

Service App.
(Guide)

Service App.
(Life Support)

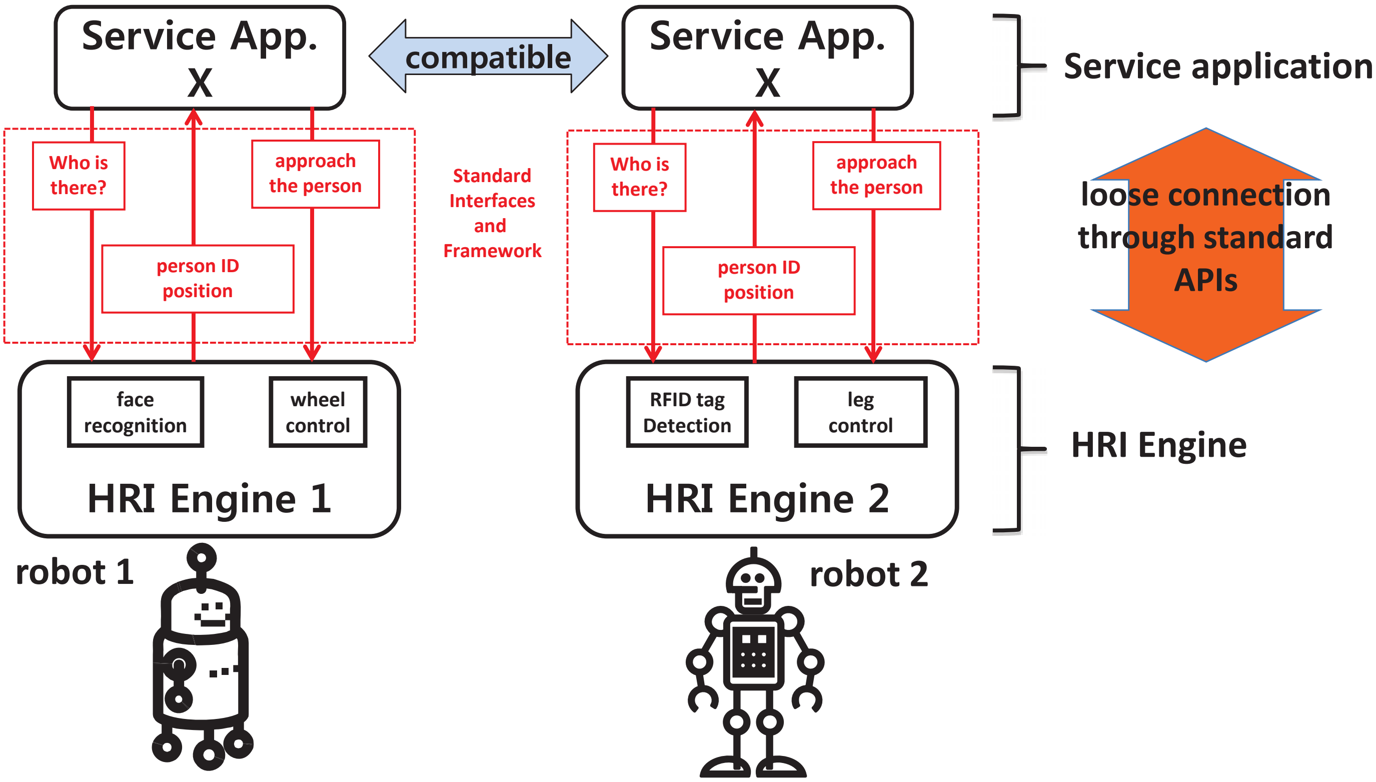
RoIS Framework

HRI Engine
face recognition
wheel control

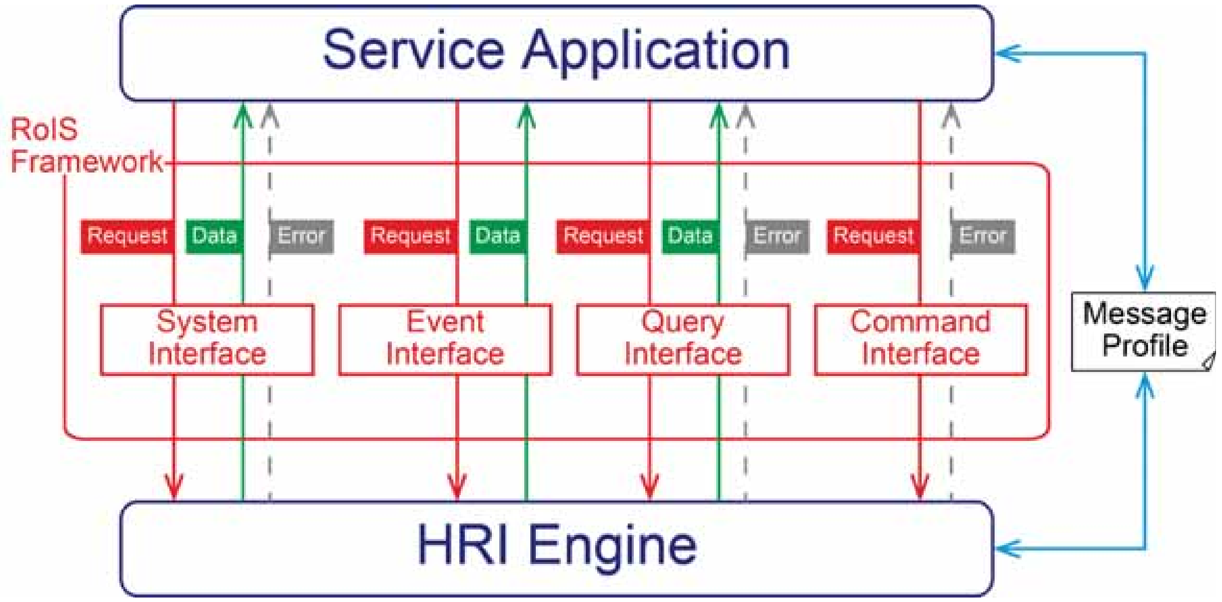
HRI Engine
LRF object Detection
monitoring camera

HRI Engine
RFID tag detection
leg control

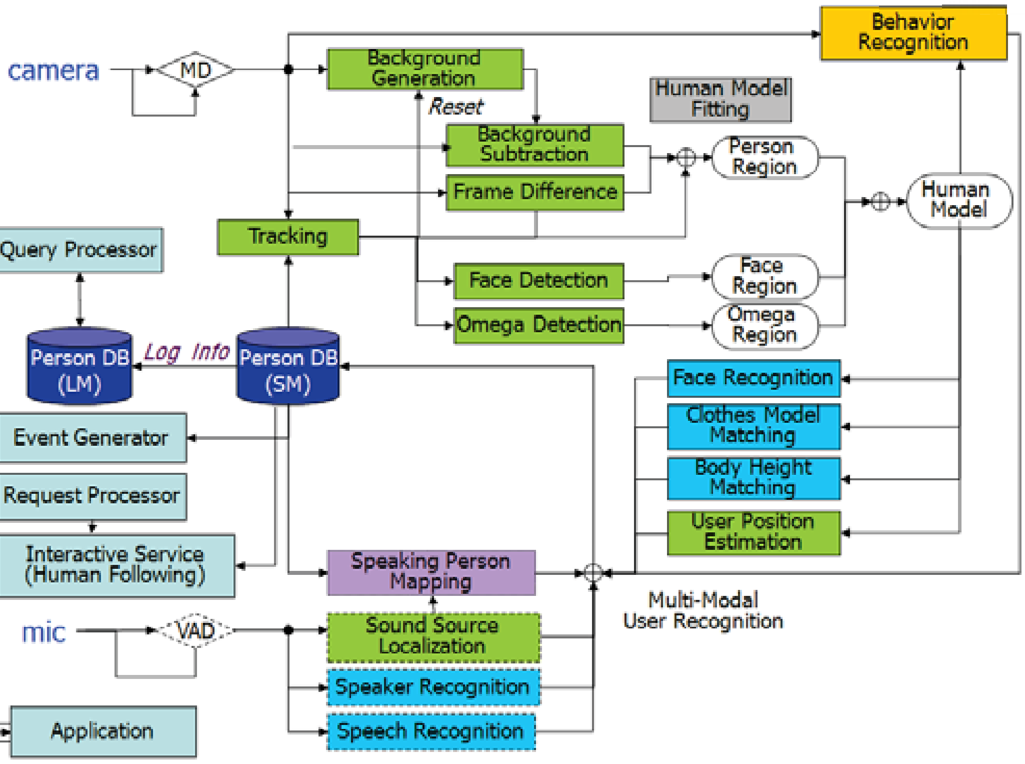
Possible Solution of Software Reuse



Interfaces of RoIS Framework and its message flows



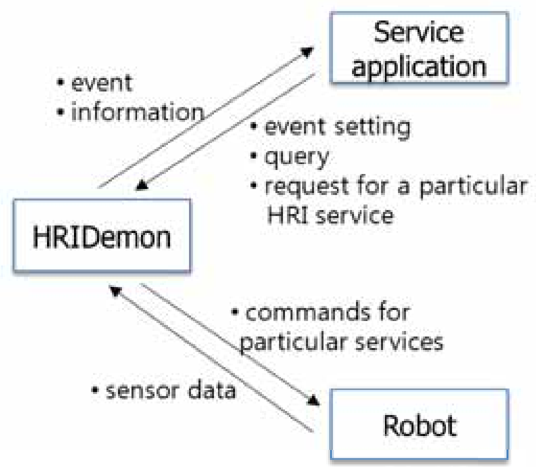
Designed architecture of the HRIDemon in ETRI



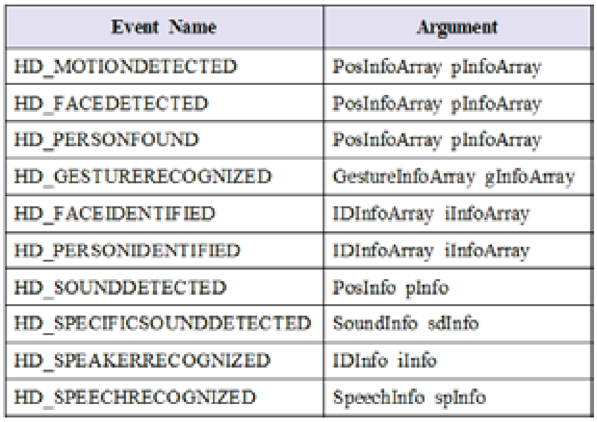
Example of designed architecture of the HRIDemon

The proposed framework: HRIDemon (2)

- The HRIDemon **continuously analyzes sensor data** taken from a robot and **accumulates information about users**.
- When a service application hopes to get some information about users, it can **easily get more reliable information** just by **registering some events** on the HRIDemon.



Event sample



Event Name	Argument
HD_MOTIONDETECTED	PosInfoArray pInfoArray
HD_FACEDETECTED	PosInfoArray pInfoArray
HD_PERSONFOUND	PosInfoArray pInfoArray
HD_GESTURERECOGNIZED	GestureInfoArray gInfoArray
HD_FACEIDENTIFIED	IDInfoArray iInfoArray
HD_PERSONIDENTIFIED	IDInfoArray iInfoArray
HD_SOUNDDETECTED	PosInfo pInfo
HD_SPECIFICSOUNDDETECTED	SoundInfo sdInfo
HD_SPEAKERRECOGNIZED	IDInfo iInfo
HD_SPEECHRECOGNIZED	SpeechInfo spInfo

Implementation SCENARIO

Scenario	HRIDemon	ServiceApp
	Start	
Connection		[System:request] Connection to the HRIEngine
	[System:receive] Connection from service application	
	[System:send] Connected	
		[System:receive] Connected
GetUsername		[Query:request] List of registered user names
	[Query:receive] List of registered user names	
	[Query:send] User name list (1) aaa (2) bbb	
		[Query:receive] User name list (1) aaa (2) bbb
SetEvent		[Event:request] Event registration (ID:xxx, name:xxx)
	[Event:receive] Event registration (ID:xxx, name:xxx)	
	[Event:send] Event registration success	
		[Event:receive] Event registration success

---

* Message display on a console window

GetEvent List		
	[Query:receive] List of registered events	
	[Query:send] events list (1) id: xx, name: xx (2) id: xx, name: xx	
		[Query:receive] events list (1) id: xx, name: xx (2) id: xx, name: xx (person_identified, gesture_recognized)
actors entered	HRIEngine> motion_detected... HRIEngine> face_detected... HRIEngine> person_detected... HRIEngine> person_identified... HRIEngine> gesture_recognized... scroll........	
Event occurred	[Event occured] person_identified(event_id: %x) 1) person information 2) person information	[EventHandler] person_identified(event_id: %x)

RoIS Event Register

RoIS Engine

Sample service application

Implemented HRIDemon

Intelligent robot

Conclusions

- **The concept of HRI component integration framework** and a sample design of its architecture were proposed.
- The HRIDemon provides **more reliable recognition results** because it can get **abundant evidences constantly from diverse recognition components**.
- Many developers of HRI service applications **can easily get information on users** from the proposed HRIDemon without concerns about making the best use of individual HRI components.

Thank you!

OMG Meeting, Burlingame, CA, USA, 2012

Component Management in OPRoS

Seungwoog Jung(swjung@etri.re.kr,
Electronics and Communications Research Institute
KOREA

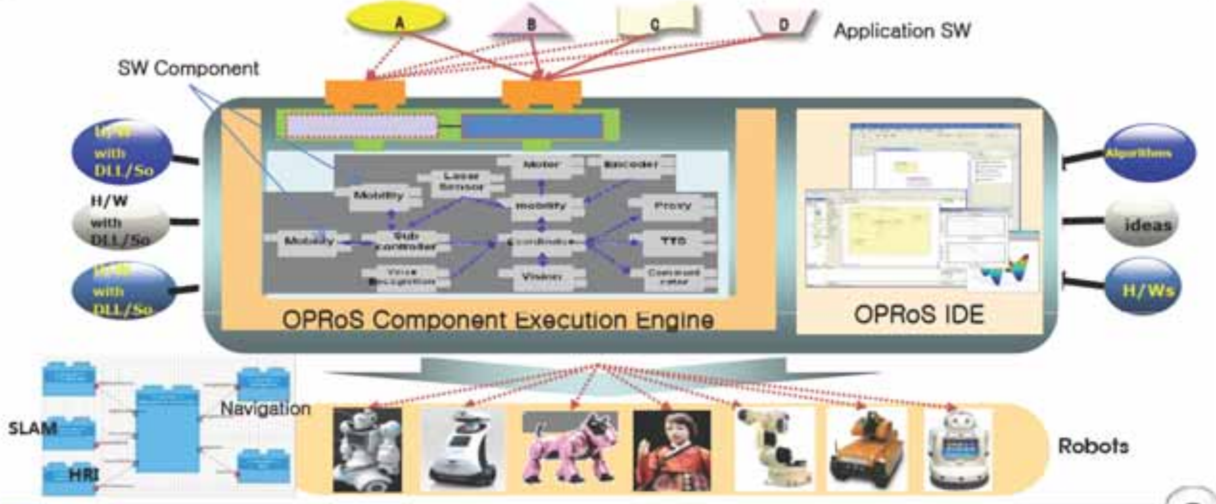
Contents

- Introduction of OPRoS
- OPRoS Component Model
- OPRoS Component Execution Engine
- Conclusion

Introduction of OPRoS

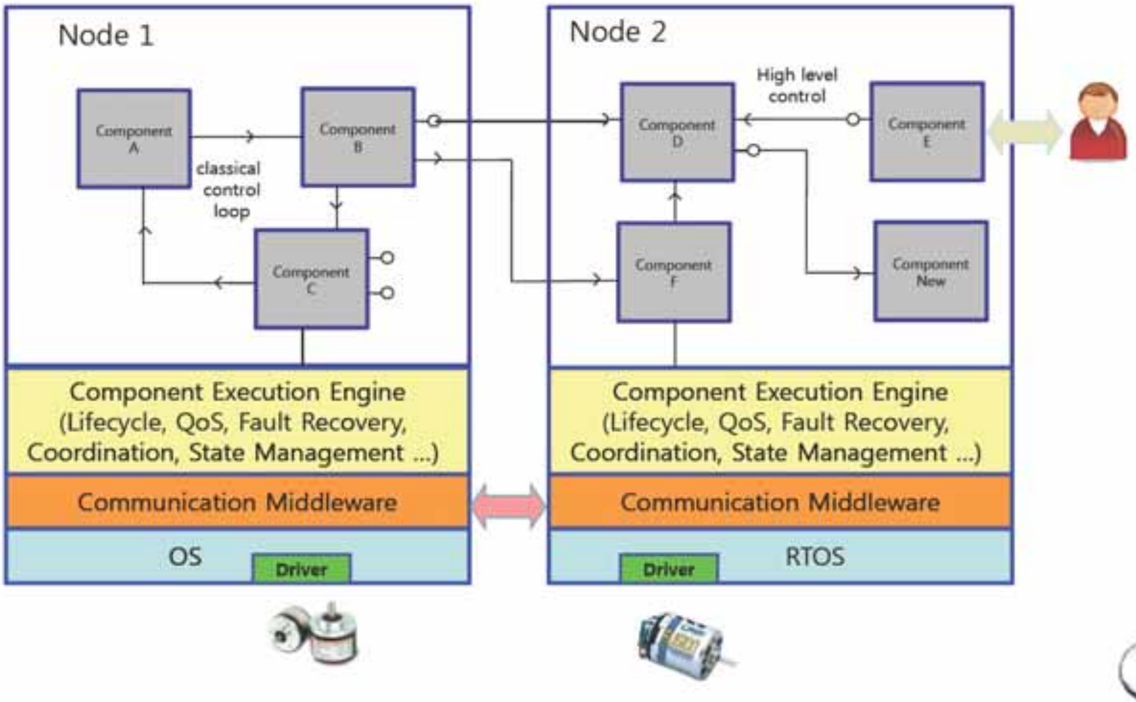
OPRoS

- **OPRoS :** Open Platform for Robotic Services
- Robot software platform that provides component-based development and execution environments for robot software
- providing the robot software component model, component execution engine, various middleware services, development tools, and simulation environment



Introduction of OPRoS Component

Role of Component Execution Engine



Component Execution Engine (Lifecycle, QoS, Fault Recovery, Coordination, State Management ...)

Communication Middleware

OS

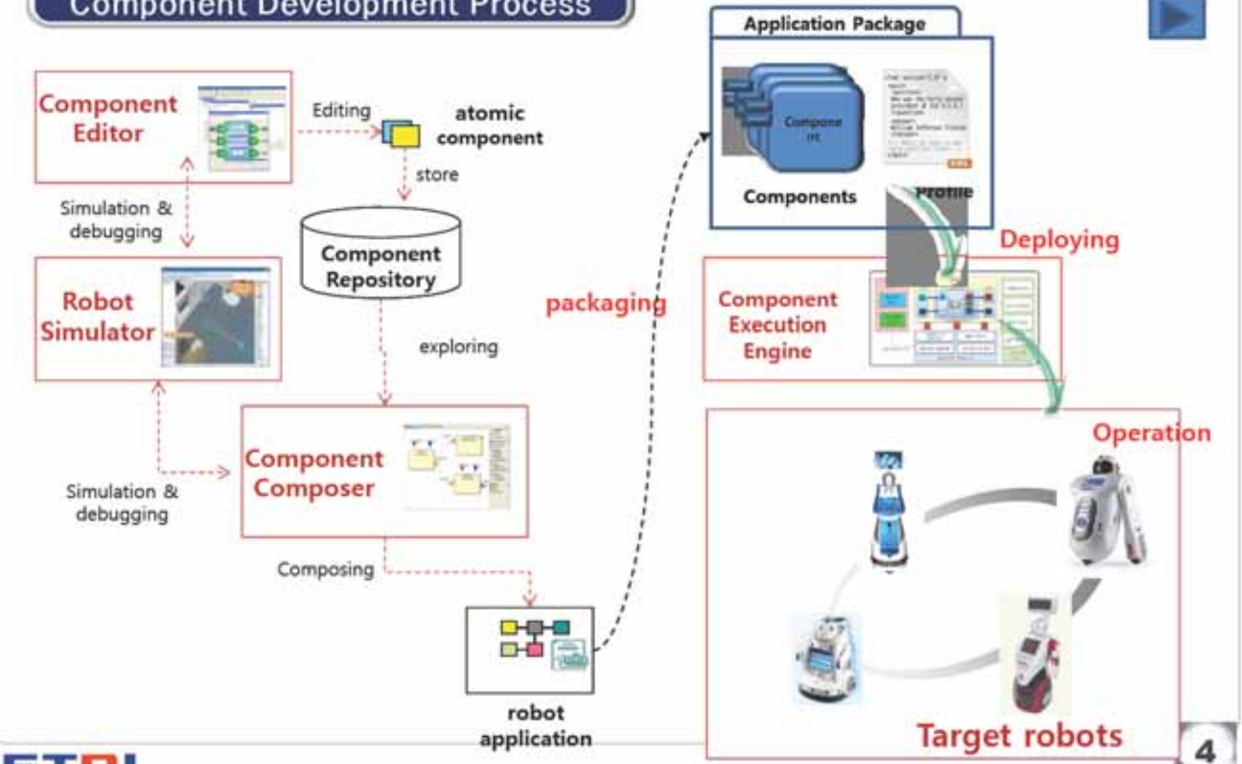
Component Execution Engine (Lifecycle, QoS, Fault Recovery, Coordination, State Management ...)

Communication Middleware

RTOS

Introduction of OPRoS

Component Development Process



Application Package

Components

Profile

Component Editor

Editing

atomic component

store

Simulation & debugging

Component Repository

Robot Simulator

exploring

packaging

Deploying

Component Execution Engine

Operation

Component Composer

Simulation & debugging

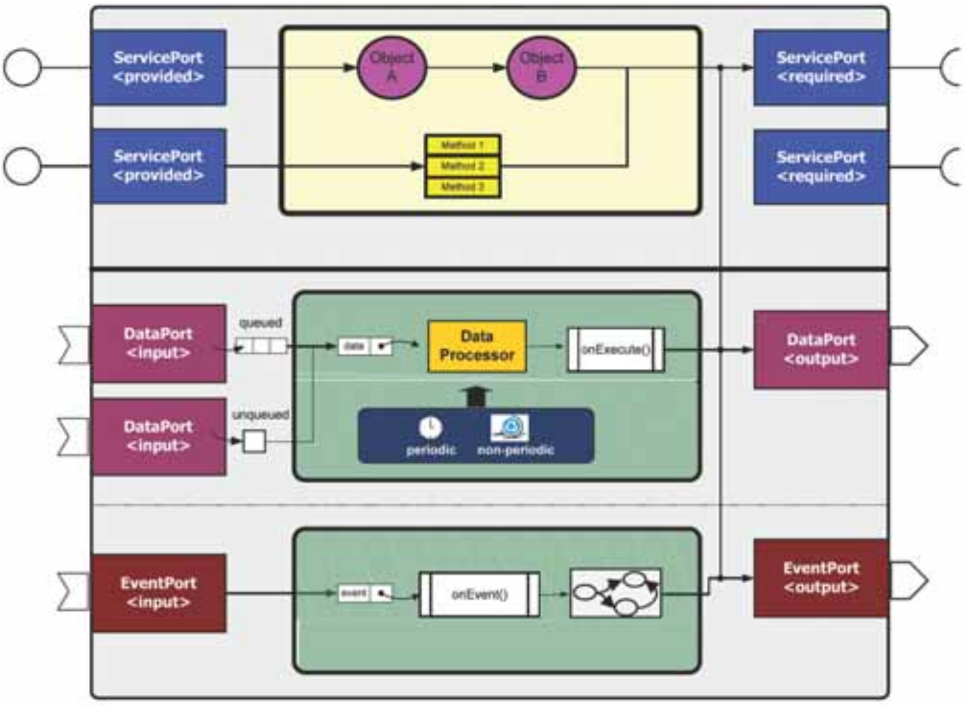
Composing

robot application

Target robots

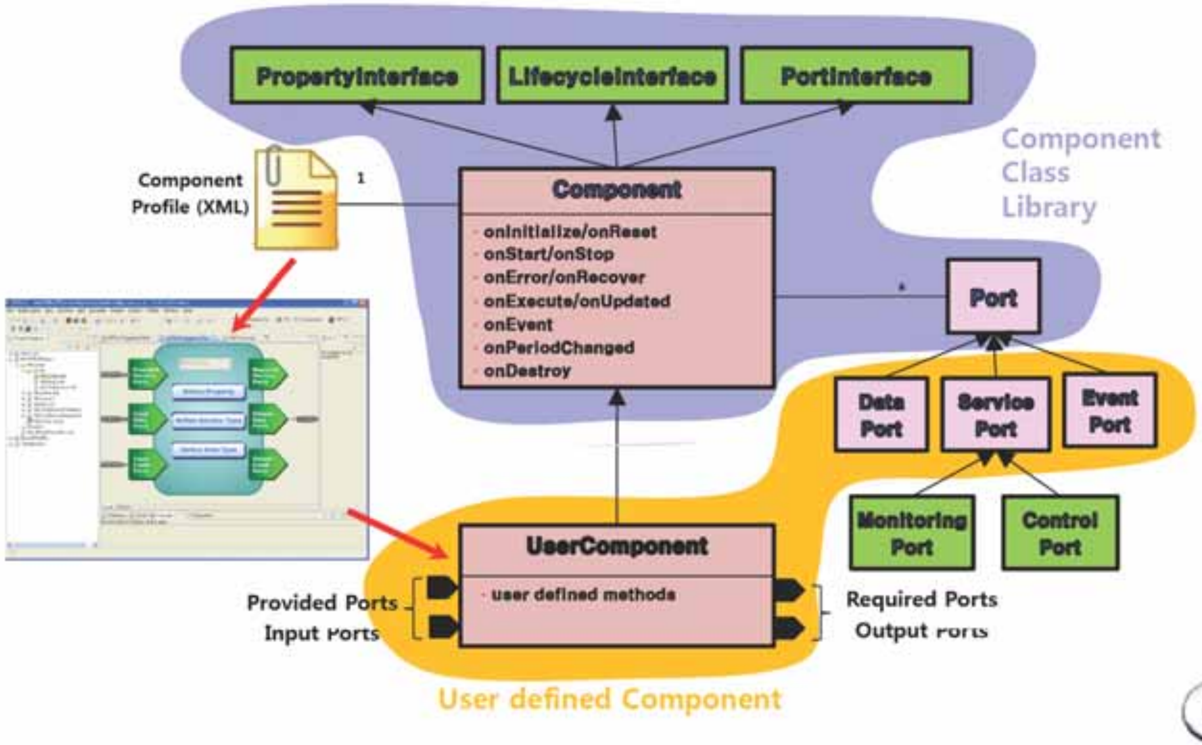
OPRoS Component Model

Component Communication Model



OPRoS Component Model

Component Class Diagram



PropertyInterface
LifecycleInterface
PortInterface

Component Profile (XML)
1

Component
- onInitialize/onReset
- onStart/onStop
- onError/onRecover
- onExecute/onUpdated
- onEvent
- onPeriodChanged
- onDestroy

*
Port
Data Port
Service Port
Event Port
Monitoring Port
Control Port

Component Class Library

UserComponent
- user defined methods

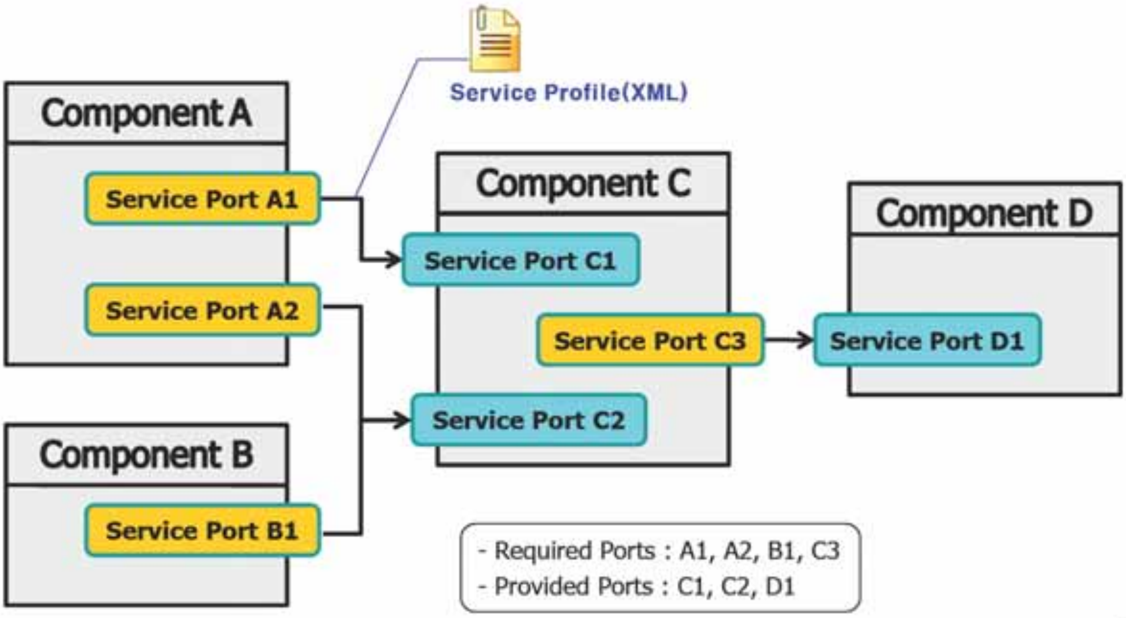
Provided Ports
Input Ports
Required Ports
Output Ports

User defined Component

OPRoS Component Model

Service Port

- Interface for method invocation of component



Service Profile(XML)

Component A
- Service Port A1
- Service Port A2

Component B
- Service Port B1

Component C
- Service Port C1
- Service Port C3
- Service Port C2

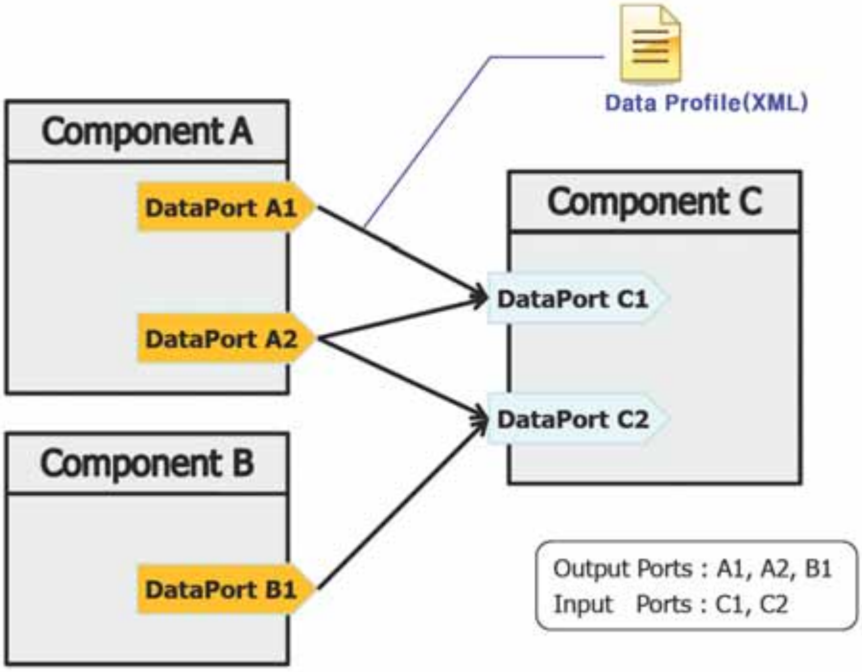
Component D
- Service Port D1

- Required Ports : A1, A2, B1, C3
- Provided Ports : C1, C2, D1

OPRoS Component Model

Data Port

- Port for transferring data to components



Output Ports : A1, A2, B1
Input Ports : C1, C2

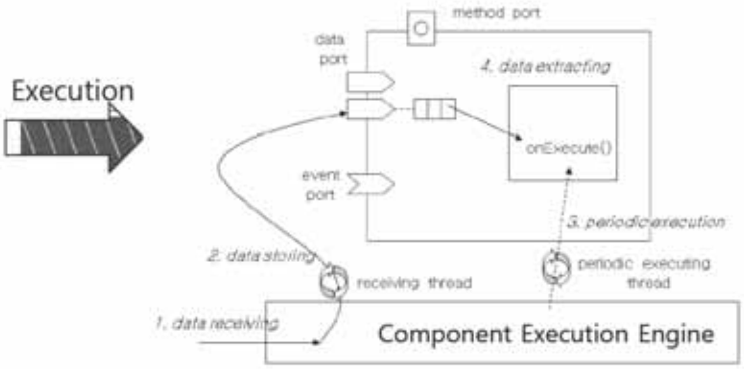
OPRoS Component Model

Data Port

- asynchronous processing
- data processing after queuing

```
<execution_semantics>
            <priority>5</priority>
            <type>periodic</type>
            < period>100</period>
</execution _semantics>
```

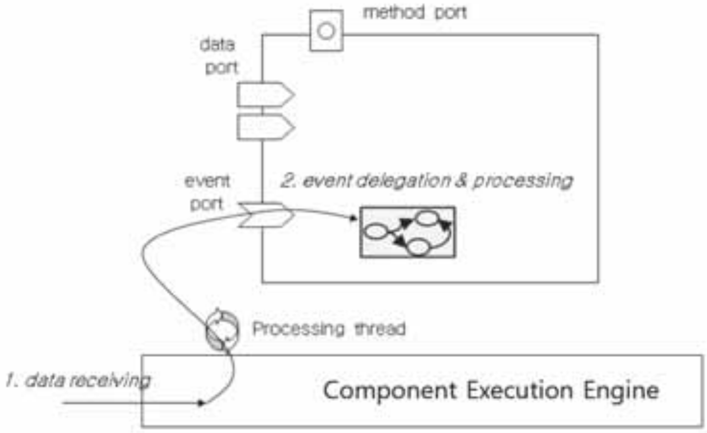
<Component Profile>



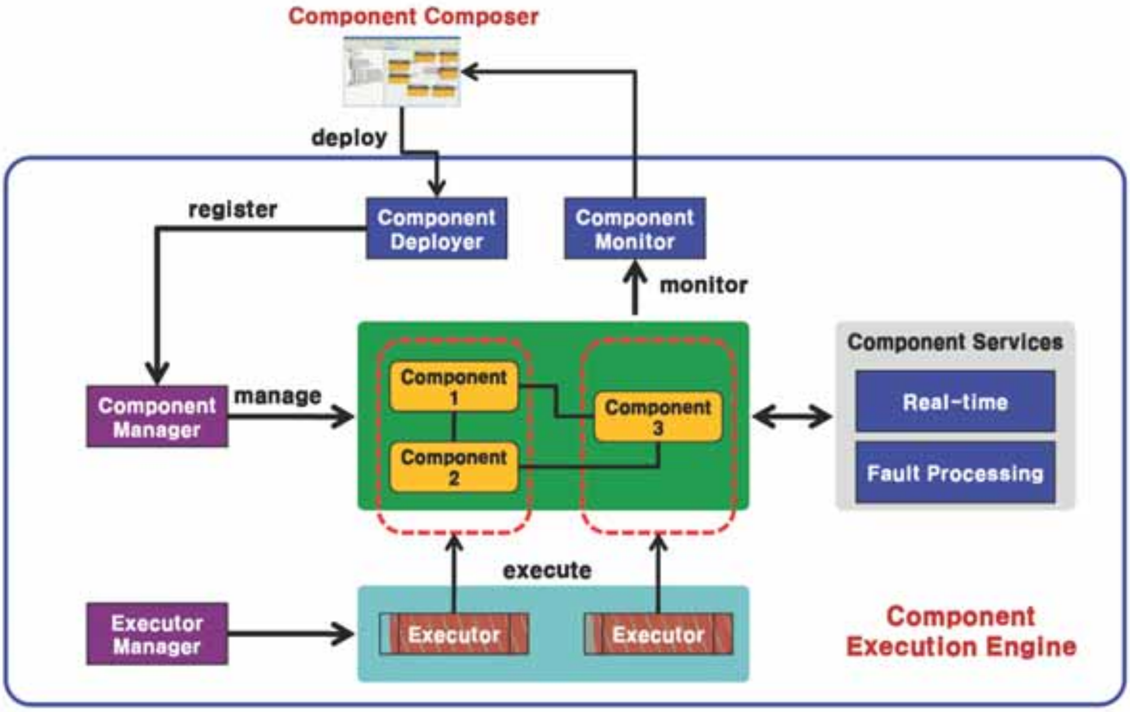
OPRoS Component Model

Event Port

- Port for transferring event to components
  - Immediate processing of data without queuing



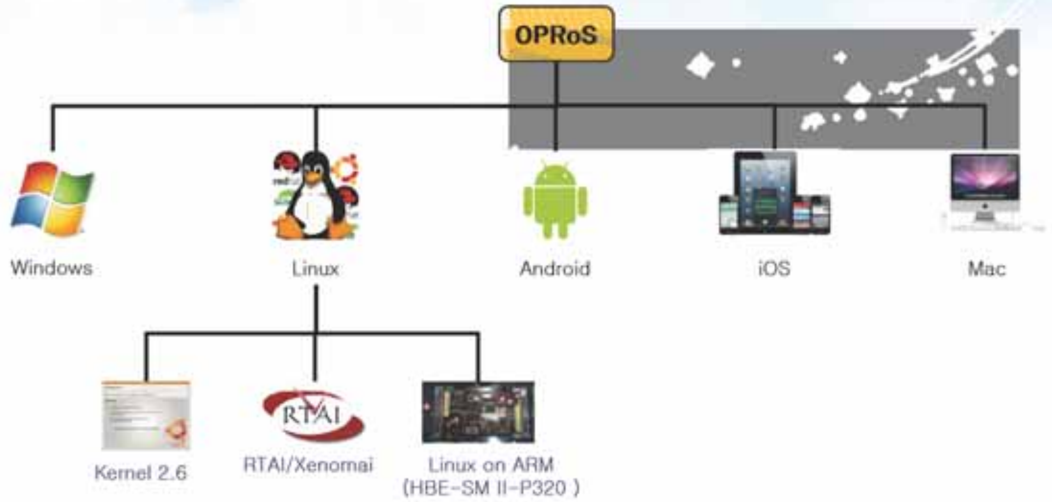
Component Execution Engine



Component Execution Engine

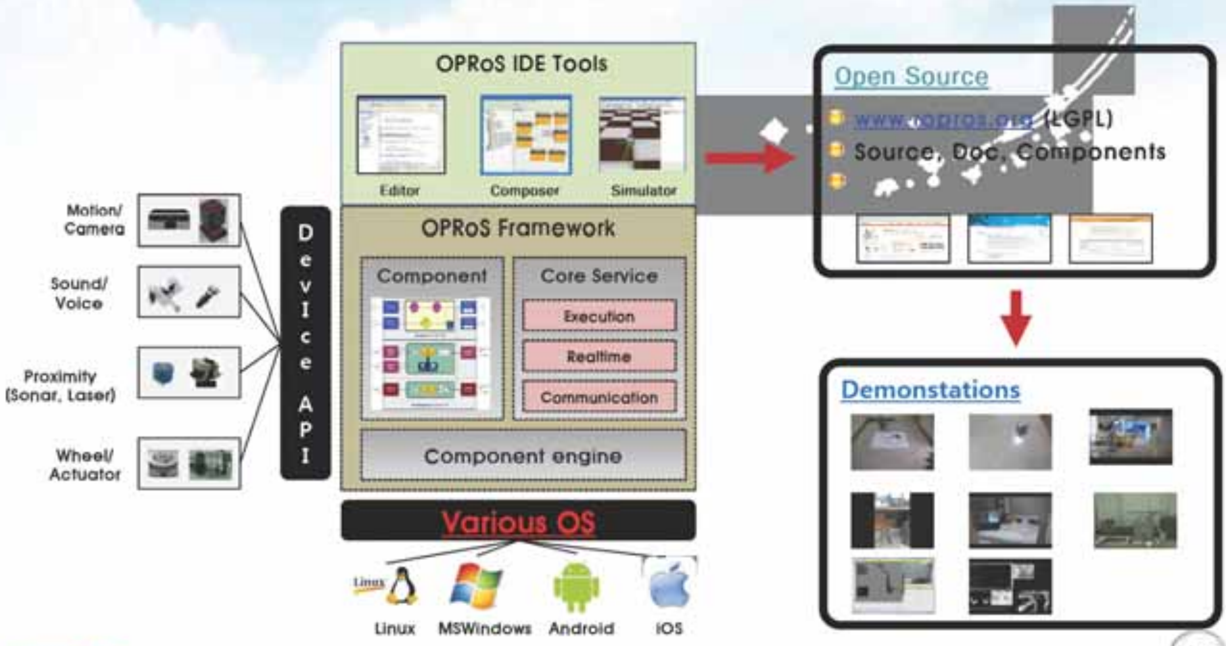
OS Support

- Various OS support
- Windows, Linux
- Android, iOS, Mac



Component Execution Engine

-Open source (www.ropros.org)
-source code, documents, components



Q & A

Thank you!

ETRI 한국전자통신연구원

TOSHIBA
Leading Innovation >>>

Cloud Networked Robotics and Acceleration Based Sensing

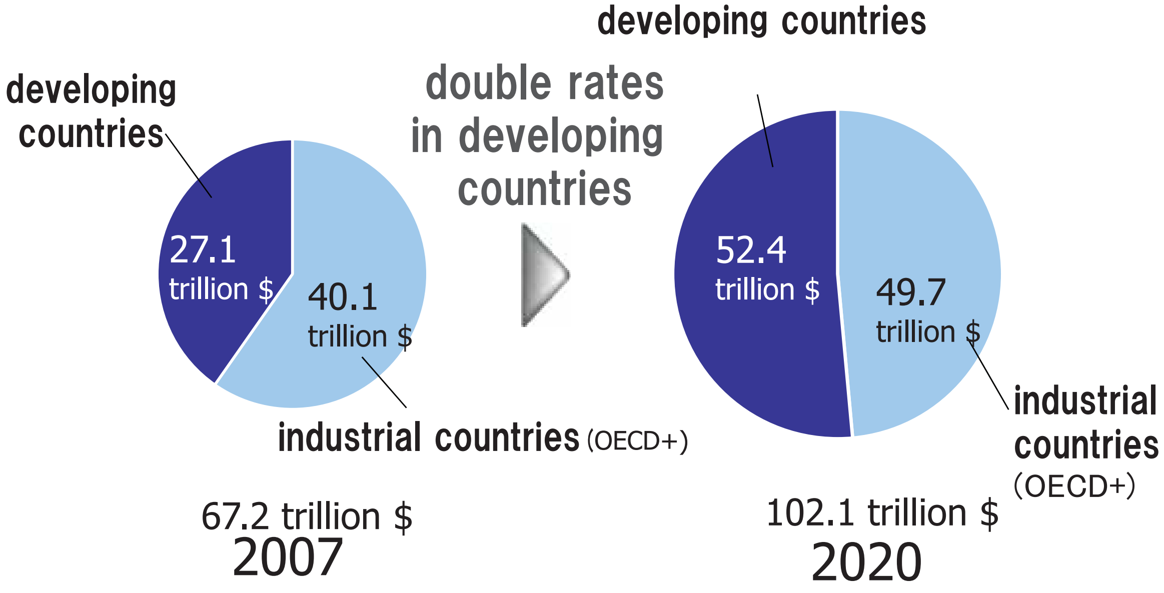
Miwako Doi, Toshiba Corporation
Tokyo, Japan
Robotics DTF, OMG
Dec. 11, 2012

eco スタイル
東芝グループは、持続可能な
地球の未来に貢献します。

Mega Trends

Economic developing center shifting to developing countries

Market center shifting to developing countries

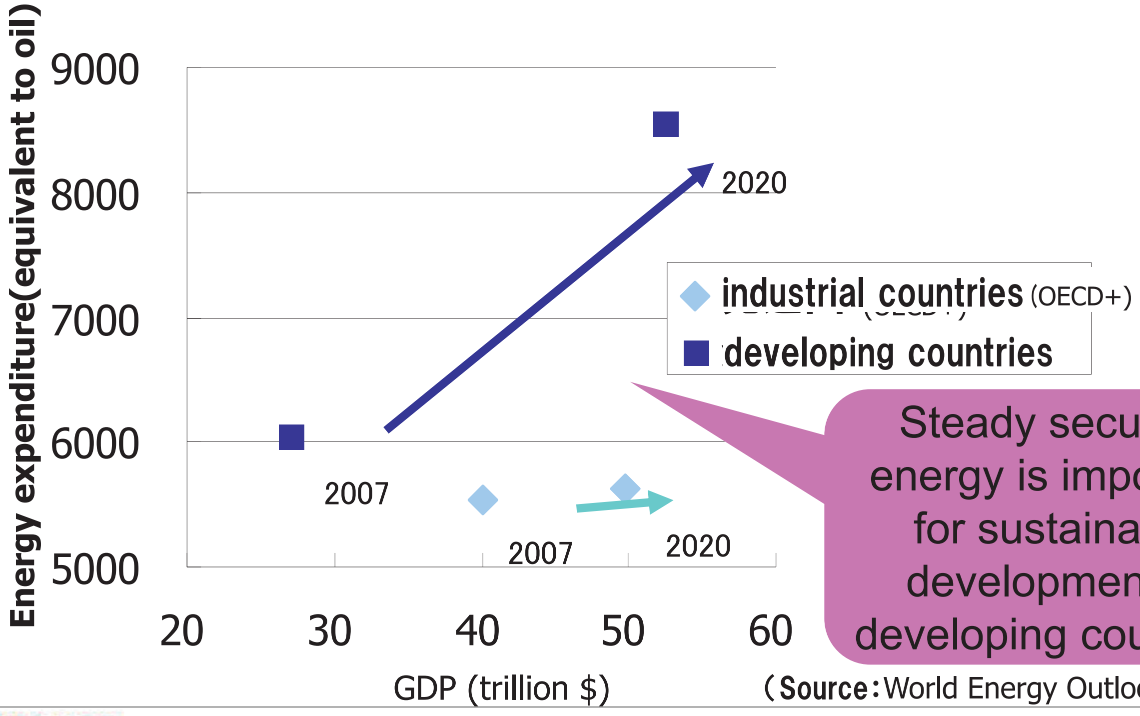


GDP changing

（source:World Energy Outlook 2010

Energy expenditure center shifting to developing countries

Sharply-rising energy expenditure in developing countries

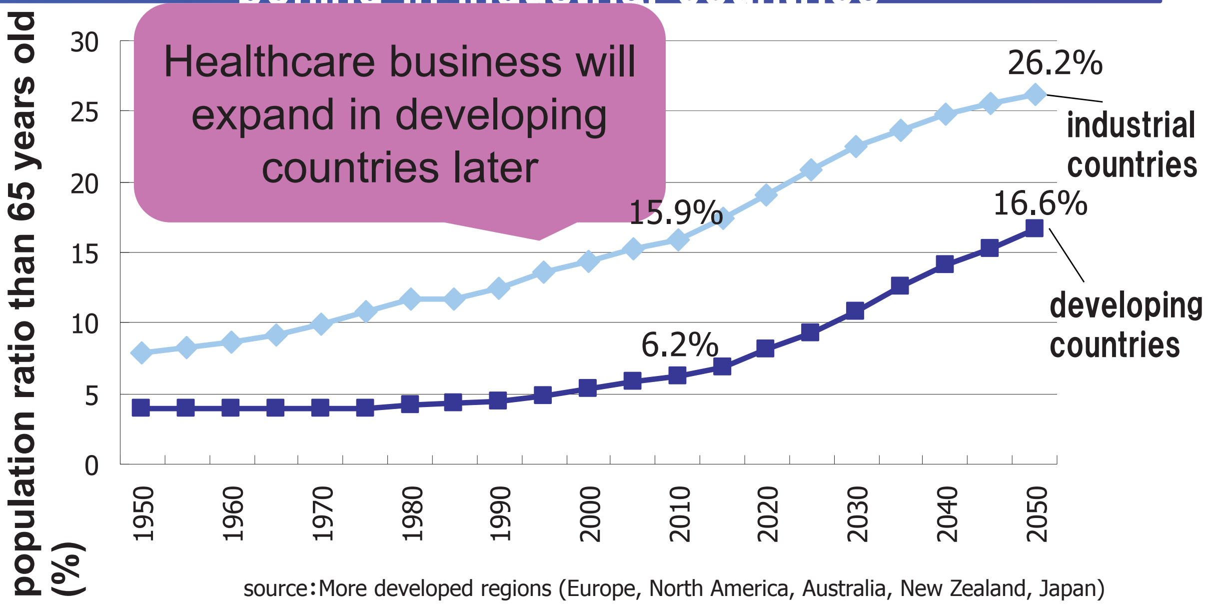


Steady securing energy is important for sustainable development of developing countries

（Source:World Energy Outlook 2010）

Dwindling birthrate and an aging population in industrial countries

Developing countries follow 30 or 40 years behind in industrial countries



Healthcare business will expand in developing countries later

source:More developed regions (Europe, North America, Australia, New Zealand, Japan)
:Less developed regions, excluding least developed 49 countries

（Source:United Nation World Population Prospect 2008）

Cloud Networked Robotics

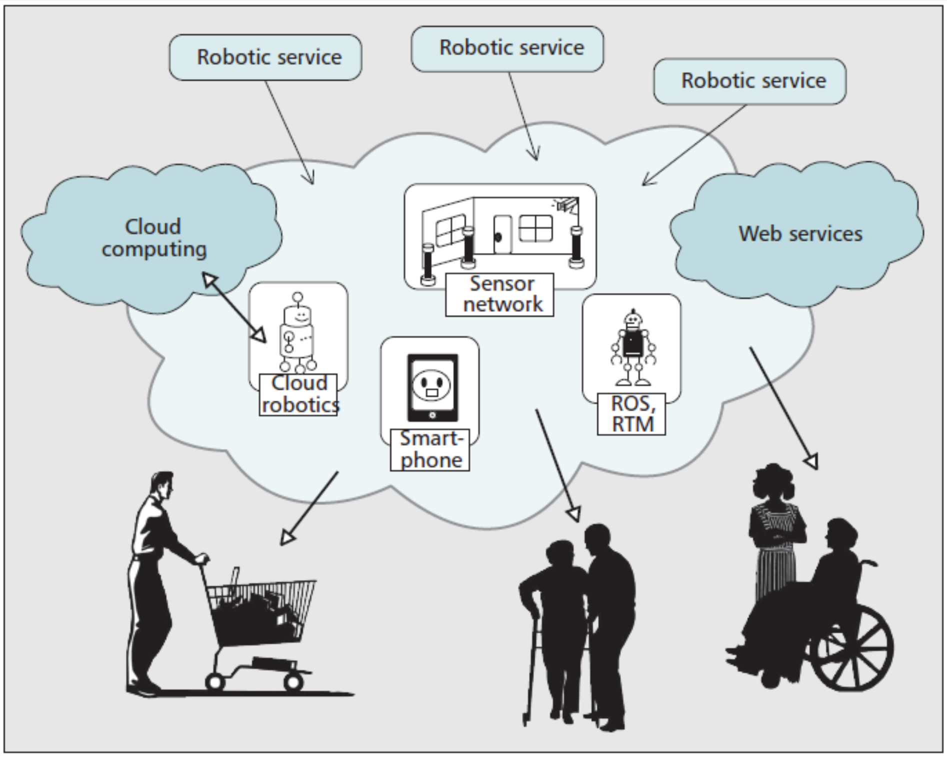


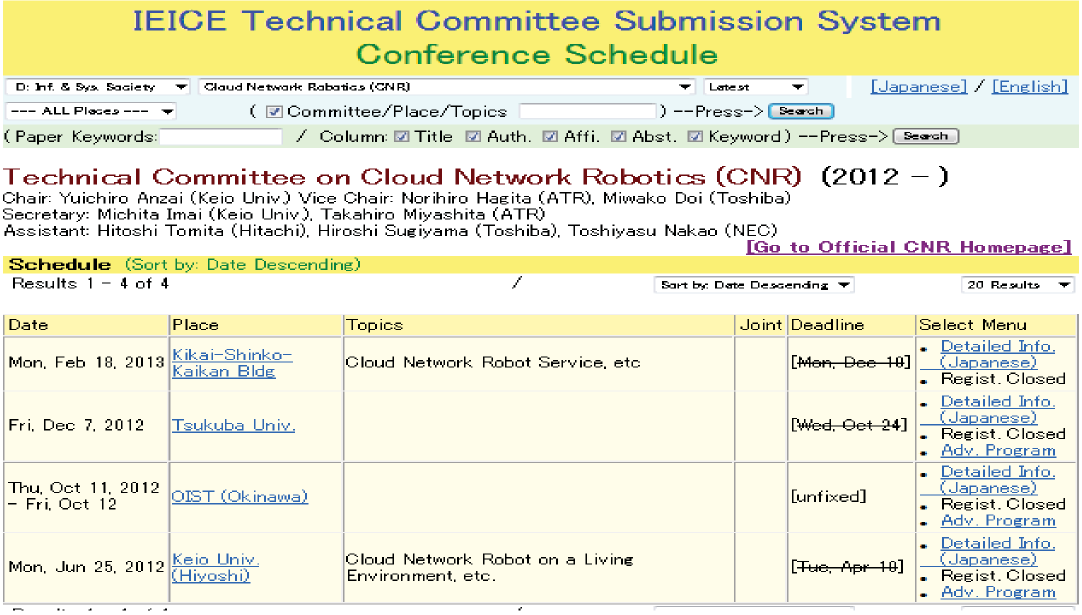
Figure 1. *In Cloud Networked Robotics, a platform layer located between service applications and robotic components isolates and coordinates them to realize multi-area, multi-robot networked robotic services.*

Courtesy of Dr. Kamei

IEICE Technical Committee on Cloud Networked Robotics

- **Interaction between heterogeneous robots**
- **Interaction between human and robots**
- **Cloud data collaboration for networked robots**

http://www.ayu.ics.keio.ac.jp/cnr/



IEICE Technical Committee Submission System
Conference Schedule

Technical Committee on Cloud Network Robotics (CNR) (2012 -)
Chair: Yuichiro Anzai (Keio Univ.) Vice Chair: Norihiro Hagita (ATR), Miwako Doi (Toshiba)
Secretary: Michita Imai (Keio Univ.), Takahiro Miyashita (ATR)
Assistant: Hitoshi Tomita (Hitachi), Hiroshi Sugiyama (Toshiba), Toshiyasu Nakao (NEC)
[Go to Official CNR Homepage]

Schedule (Sort by: Date Descending)
Results 1 - 4 of 4 / Sort by: Date Descending ▼ 20 Results ▼

Date	Place	Topics	Joint	Deadline	Select Menu
Mon, Feb 18, 2013	Kikai-Shinko-Kaikan Bldg	Cloud Network Robot Service, etc		[~~Mon, Dec 10~~]	Detailed Info. (Japanese) Regist. Closed
Fri, Dec 7, 2012	Tsukuba Univ.			[~~Wed, Oct 24~~]	Detailed Info. (Japanese) Regist. Closed Adv. Program
Thu, Oct 11, 2012 - Fri, Oct 12	OIST (Okinawa)			[unfixed]	Detailed Info. (Japanese) Regist. Closed Adv. Program
Mon, Jun 25, 2012	Keio Univ. (Hiyoshi)	Cloud Network Robot on a Living Environment, etc.		[~~Tue, Apr 10~~]	Detailed Info. (Japanese) Regist. Closed Adv. Program

Brains on Clouds

- **Cleverbot http://www.cleverbot.com/**
- **Siri (Speech Interpretation and Recognition Interface)**
- **しゃべってコンシェルShabette_concier**

http://www.nttdocomo.co.jp/service/information/shabette_concier/

- **Animetrics**

http://animetrics.com/cloud-face-recognition-services/

:

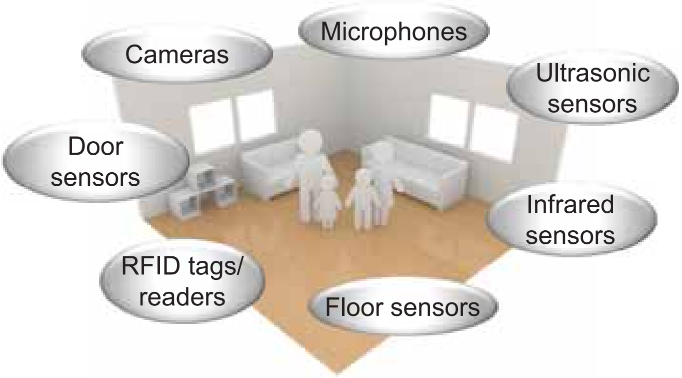
:

Elements of Cloud Networked Robotics

- **Data sensing**
- **Recognition**
- **Data mining**
- **Visualization and feedback**
- **Modeling**
- **Simulation**
- **Prediction**
- **Control**
- **Actuation**
- **Utilization**
- **Harmonization with human and robots**

Acceleration Based Sensing

Related Works on Activity Recognition (1)

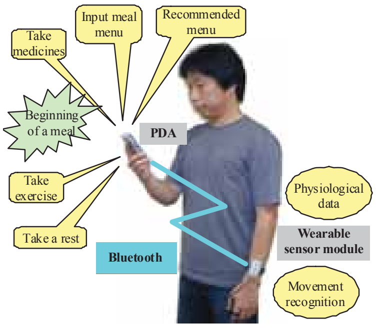


- Aware Home
  Georgia Tech. (1999)
- EasyLiving Project
  Microsoft Research (2000)
- YUKARI Project
  NiCT (2004)
- Sensing Room
  Univ. of Tokyo (2004)

Many sensors need to be installed in the environment.

➔ too costly to implement

Related Works on Activity Recognition (2)



Toshiba (2002)

MIT (2004)

Fig. 1. (a) Conceptual image of wristband type sensor device and (b) prototype device

NTT (2010)

Various living activities have become recognizable by wearing sensors.

➔ <u>Impracticable to continuously wear many accelerometers in daily life</u>
<u>to use a special device from the viewpoint of the cost</u>

Mobile phones / Smartphones

- are widely used.
- are always located close to users.
- are equipped with sensors such as an accelerometer and a GPS.

Problems for Activity Recognition @ Smartphone

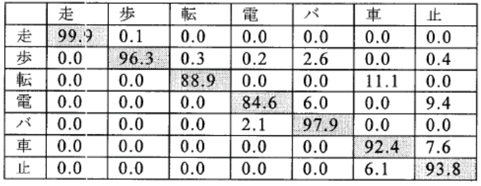
- **Problem 1: Focusing only outdoor migration activities**

(Related works)

(a) A three-axis accelerometer mounted on a cell phone
(b) Gait analysis browsing by Java applications

NTT DoCoMo (2006)

表 9：釈迦の推定性能[%]



	走	歩	転	電	バ	車	止
走	99.9	0.1	0.0	0.0	0.0	0.0	0.0
歩	0.0	96.3	0.3	0.2	2.6	0.0	0.4
転	0.0	0.0	88.9	0.0	0.0	11.1	0.0
電	0.0	0.0	0.0	84.6	6.0	0.0	9.4
バ	0.0	0.0	0.0	2.1	97.9	0.0	0.0
車	0.0	0.0	0.0	0.0	0.0	92.4	7.6
止	0.0	0.0	0.0	0.0	0.0	6.1	93.8

KDDI (2008)

- It was difficult to recognize various indoor activities.

Objective 1:

Recognizing various indoor activities (ADL and IADL)

- **Problem 2: Power consumption**

Objective 2:

Developing a low power consumption algorithm

Indoor Activity Recognition @ Smartphone

Recognizing various indoor activities (ADL and IADL)

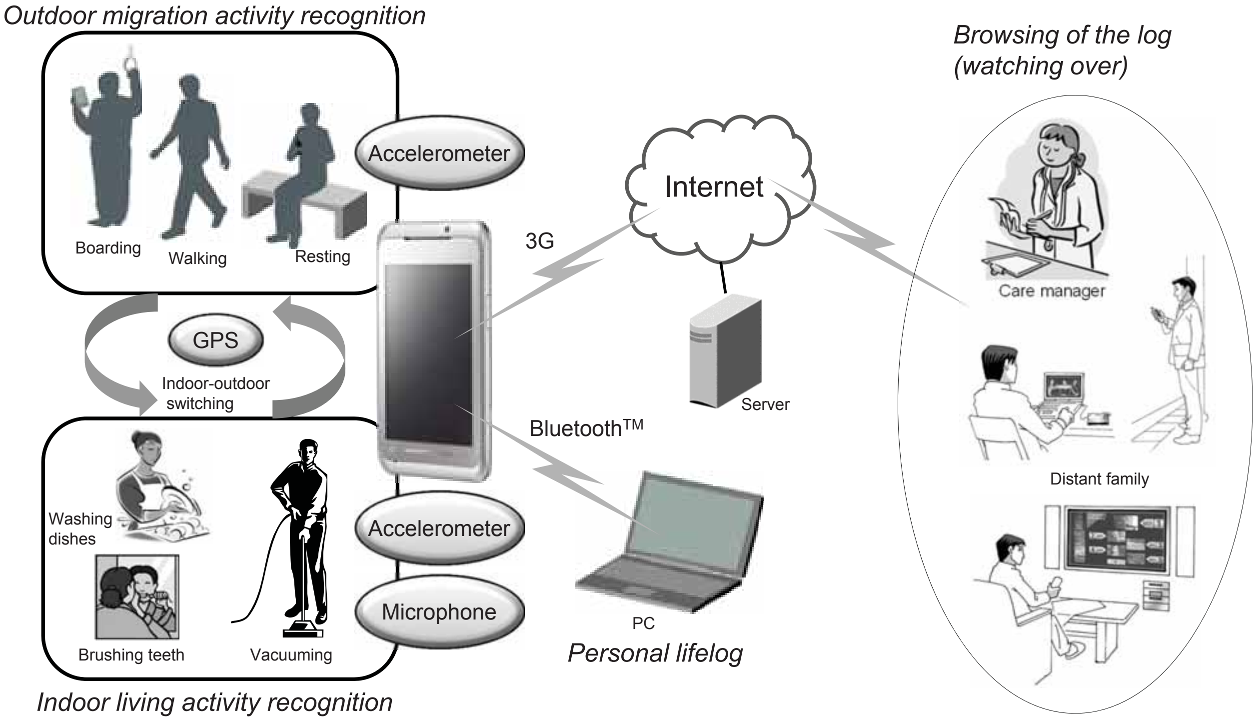
- **Using not only an accelerometer but also a microphone**
  - Hybrid activity recognition focused on the sound

Developing a low power consumption algorithm

- **Rough classification of movement by acceleration**
  - "Walking," "Quiet," and "Performing a living activity"
- **Classification of the nature of the task by sound**

Data	Resolution, Sampling Frequency	Data size per 1 sec. (Bytes)
3-axis acceleration	10bit, 20Hz	75
Sound	16bit, 16kHz	32,000

Indoor-Outdoor Activity Recognition @ Smartphone



Indoor-Outdoor Activity Recognition by a Smartphone

2012/03/16 11:31:46

Engine: ActivityAnalyzer

Place: -----

Activity: Hairdrier

Basic activities
- 🟥 Running
- 🟩 Walking
- 🟦 Stay
- ■ Boarding

Working activities
- Brushing Teeth
- Washing Dishes
- Hairdrier
- Vacuuming
- undefined
- undefined
- undefined
- undefined

Features

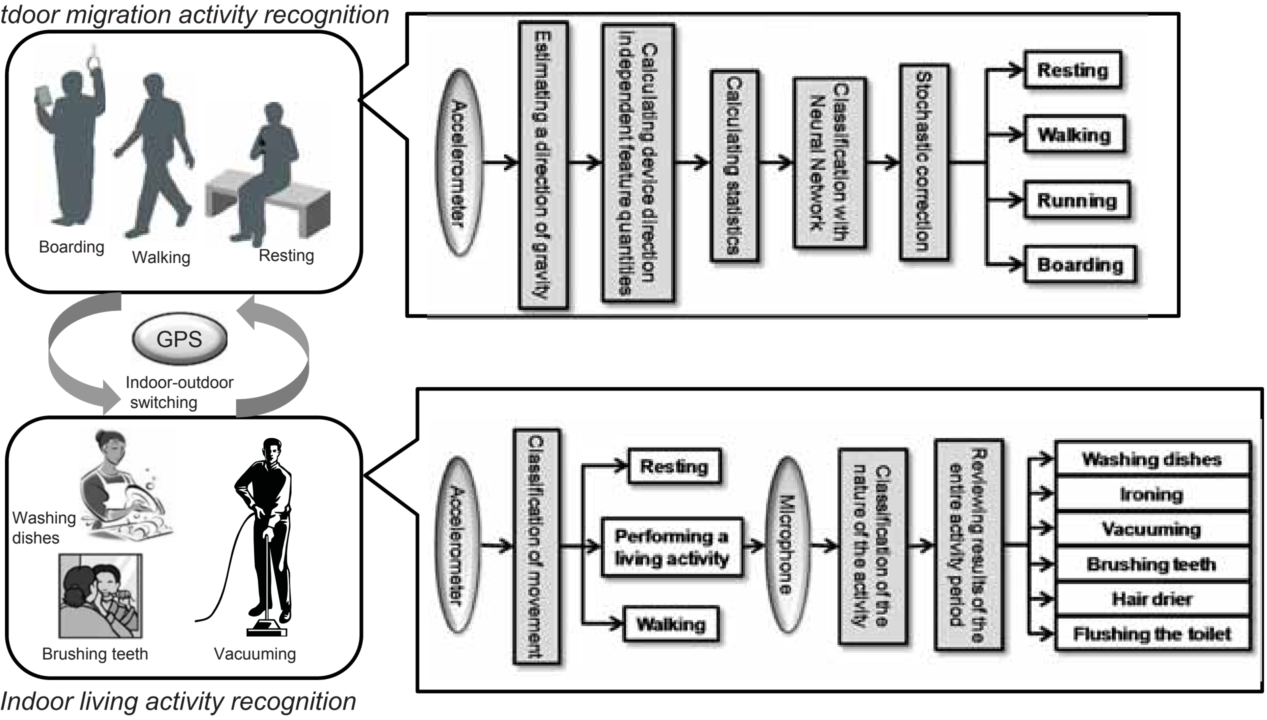
- **Recognizing various indoor-outdoor activities in real time by using a commonly-used smartphone**
  - Sensors
    - 3-axis accelerometer, microphone, GPS
  - Combined an indoor living activity recognition engine and an outdoor migration activity recognition engine

- **By switching between the 2 engines depending on an acquisition condition of GPS satellites**
  - Enables users to continuously recognize indoor-outdoor activities

- **A transmitting function to a cloud server or an external terminal via 3G networks or Bluetooth^{TM}**
  - In anticipation of various practical services

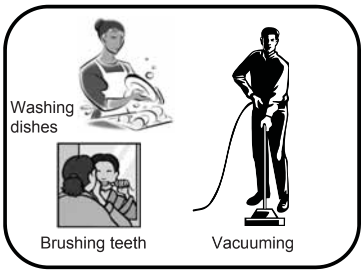
Processing Flow

Outdoor migration activity recognition

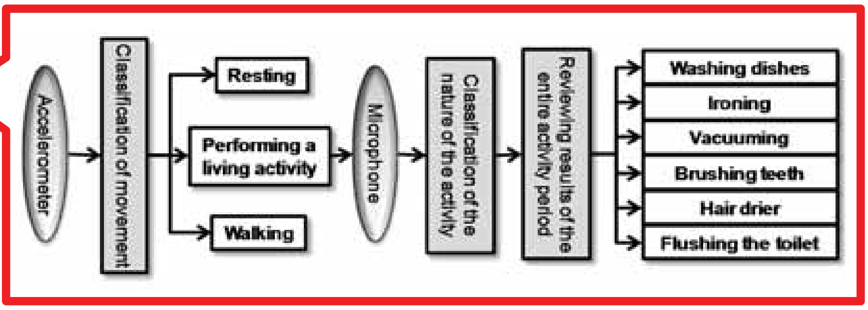


Indoor living activity recognition

Indoor Living Activity Recognition Engine



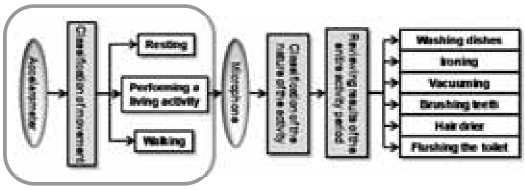
Indoor living activity recognition

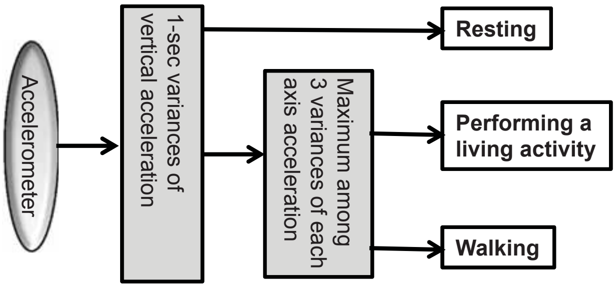

Accelerometer
Classification of movement
Resting
Performing a living activity
Walking
Microphone
Classification of the nature of the activity
Reviewing results of the entire activity period
Washing dishes
Ironing
Vacuuming
Brushing teeth
Hair drier
Flushing the toilet

1. **Roughly classifies the user's movement into "Resting," "Walking," and "Performing a living activity"**
2. **When it classifies "Performing a living activity," it activates the microphone to recognize various living activities.**

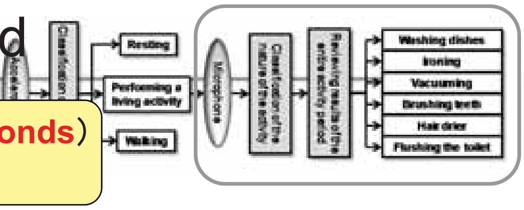
Movement Classification by Acceleration

- **Low throughput algorithm**
  - 10bit, 20Hz sampling
  - Using variances of 1-sec data series

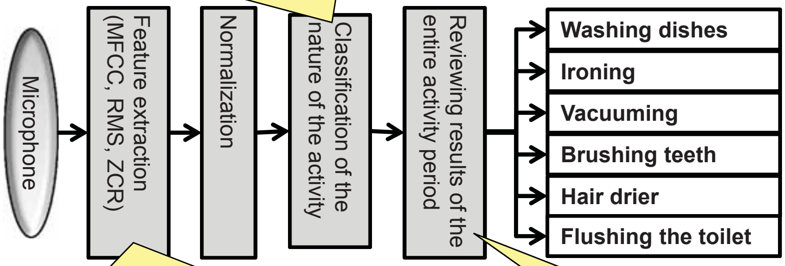



Accelerometer
1-sec variances of vertical acceleration
Maximum among 3 variances of each axis acceleration
Resting
Performing a living activity
Walking

Activity Classification by Sound



- **Train each target activity beforehand（10 seconds）**
- **Classifier: SVM（Support Vector Machine）**



•**Classifying every 1 second**

Microphone → Feature extraction (MFCC, RMS, ZCR) → Normalization → Classification of the nature of the activity → Reviewing results of the entire activity period →

- **Washing dishes**
- **Ironing**
- **Vacuuming**
- **Brushing teeth**
- **Hair drier**
- **Flushing the toilet**

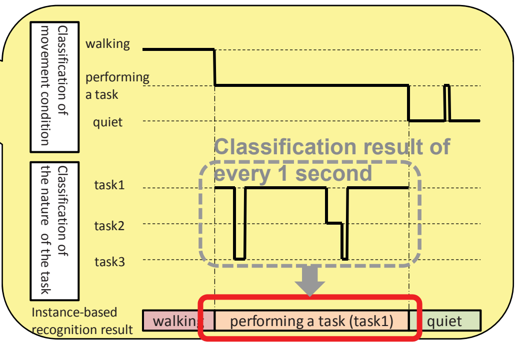
- MFCC (Mel Frequency Cepstral Coefficients)
- RMS (Root Mean Square)

$$a_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} a_i^2}$$

- ZCR (Zero-Crossing Rate)

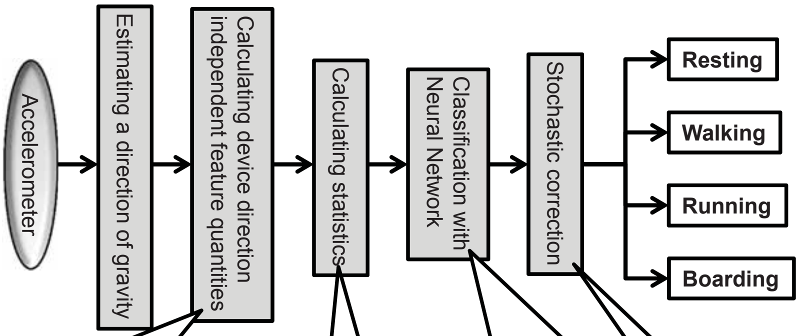
$$a_{zcr} = \frac{1}{N-1}\sum_{i=2}^{N} F\{a_i a_{i-1} < 0\}$$

•**16bit, 16kHz sampling**
•**Calculating averages of each feature every 1 second**



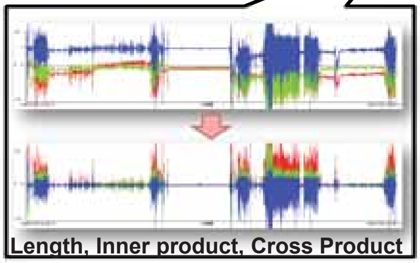
Additional recognition result by majority voting

Outdoor Migration Activity Recognition Engine

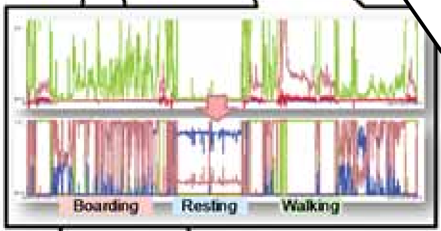


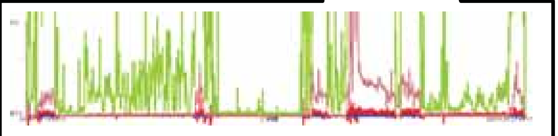
Accelerometer → Estimating a direction of gravity → Calculating device direction independent feature quantities → Calculating statistics → Classification with Neural Network → Stochastic correction →

- **Resting**
- **Walking**
- **Running**
- **Boarding**

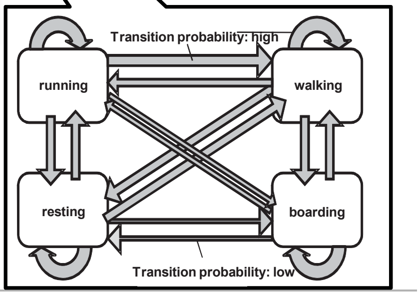


Length, Inner product, Cross Product





average, minimum, maximal and variance of these quantities in a certain time window



Evaluation Experiment

- **Indoor Living Activity Recognition**
  - **21 subjects**
    - 6 men and 6 women in their 60s
    - 5 men and 4 women in their 20s to 40s

    @ a mock living room

  - **Target living activities**
    - “Washing dishes,” “Ironing,” “Vacuuming,” “Brushing teeth,” “Hair drier,” and “Flushing the toilet.”
      - First, they performed each activity for 10 seconds.
      - Then, we directed them to perform all the target activities.
      - We did not direct them how to spend the intervals between the target activities.
      - An untrained task might occur during the intervals.
      - It should be considered to be an untrained task.

Indoor Living Activity Recognition Accuracy

Movement classification accuracy by acceleration

Rough movement	Accuracy(%)
Walking	95.2
Performing a living activity	96.8
Resting	96.5

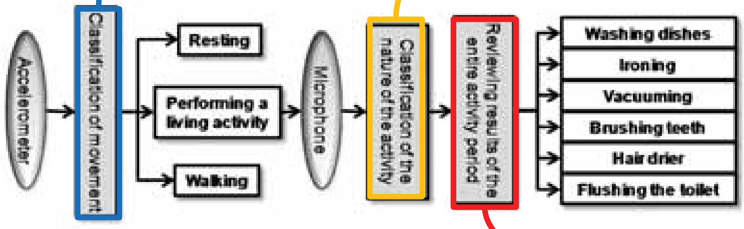
Raw activity classification accuracy every 1 second

84.5%

Living activity	Accuracy(%)
Washing dishes	80.2
Ironing	76.7
Vacuuming	86.3
Brushing teeth	74.1
Hair drier	96.4
Flushing the toilet	93.2

91.4%

Accuracy(%)
88.1
85.7
100
85.7
100
97.6



	Trained tasks							
	Washing dishes	Ironing	Vacuuming	Brushing teeth	Hair drier	Flushing the toilet	Untrained task	Accuracy (%)
Washing dishes	37			3		1	1	88.1
Ironing		36		4			2	85.7
Vacuuming			42					100
Brushing teeth		3		36			3	85.7
Hair drier					42			100
Flushing the toilet						41	1	97.6
Untrained task	3	7		4		4	87	82.9

Final classification accuracy by majority voting

Outdoor Migration Activity Recognition Accuracy

- **Trial subjects carry the cellular phone in daily life**
  - Number of subjects: 4 (our project members)
  - Total number of hours: 56 hours 31 minutes

Correct answer

Estimated result

	Running	Walking	Resting	Boarding
Running	92.5	0.0	0.0	0.0
Walking	7.4	99.4	1.1	1.4
Resting	0.0	0.1	97.5	2.6
Boarding	0.0	0.4	1.3	95.6
Unknown	0.1	0.1	0.1	0.3

Future Work

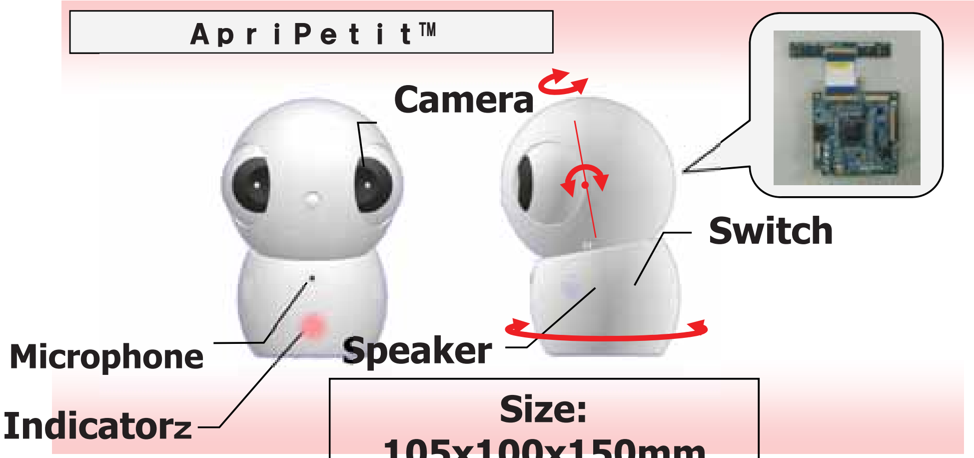
- **Performance evaluation in an actual usage environment**
  - Recognition accuracy
  - Power consumption
- **Considering the measures in case that users doesn’t carry their smartphone on them inside the house**
  - Use of sensors equipped with commonly-used digital products and home appliances

- **A number of business under consideration**
  - Tele-monitoring service
  - Healthcare service
  - Energy saving in cooperation with HEMS
  - Routine inspection task support system
  - BEMS, Smart office, etc.

Hand held IF Robot : ApriPetit™

Hand held size

- Image processing inside and speech processing cloud



Size:
105x100x150mm
Weight 500g

Thank you.

TOSHIBA
Leading Innovation >>>

This research was partly supported by Ministry of Internal Affairs and Communications

Introduction to OpenEL (Embedded Library) for Robots

December 11, 2012
OMG Robotics Information Day

Kenichi Nakamura
Director
Chairman of Applied Technology Research Committee
Chairman of Platform Research Group
Japan Embedded Systems Technology Association(JASA)

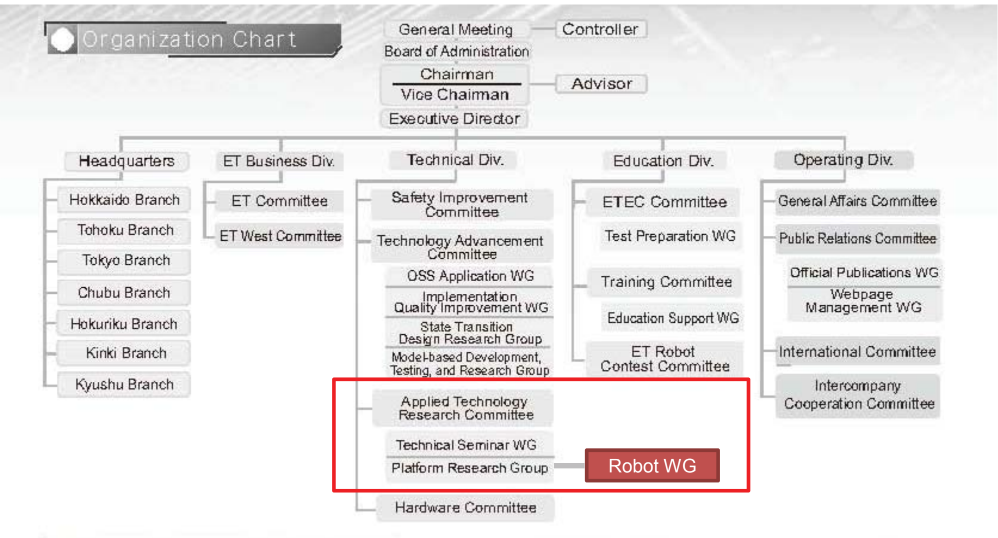
Agenda

- Introduction of JASA
  - Association profile, Main Activities, Organization, Embedded Technology Robot Software Design Contest, Platform Research Group etc.
- Introduction of OpenEL
  - Concept, Activities, API specification of Version 0.1.1, examples, video, Roadmap etc.

About JASA(Japan Embedded Systems Technology Association)

- Established in 1986.
- More than 200 embedded systems companies in Japan
  - ALPINE, CORE, dSPACE, Hitachi, Imagination, Microsoft, Mentor, MontaVista, Panasonic, RICOH, RENESAS, Toshiba etc.
- Main Activities
  - Embedded Technology, a Comprehensive Exhibit of Embedded Systems Technology(Yokohama and Osaka)
  - Implementation and Expansion of ETEC(Embedded Technology Engineer Certification)
- Study and Research Activities for Technological Advancement
  - Case studies of safe design, surveying of techniques and methods recommended by safety standards, research and study into safety-related products, and support for IEC 61508 and ISO 26262.
  - Research and study on modeling and verification for the achievement of reliable embedded software development and public awareness activities and dissemination of case studies for the education of engineers.
- Embedded Technology Robot Software Design Contest

JASA Organization Chart



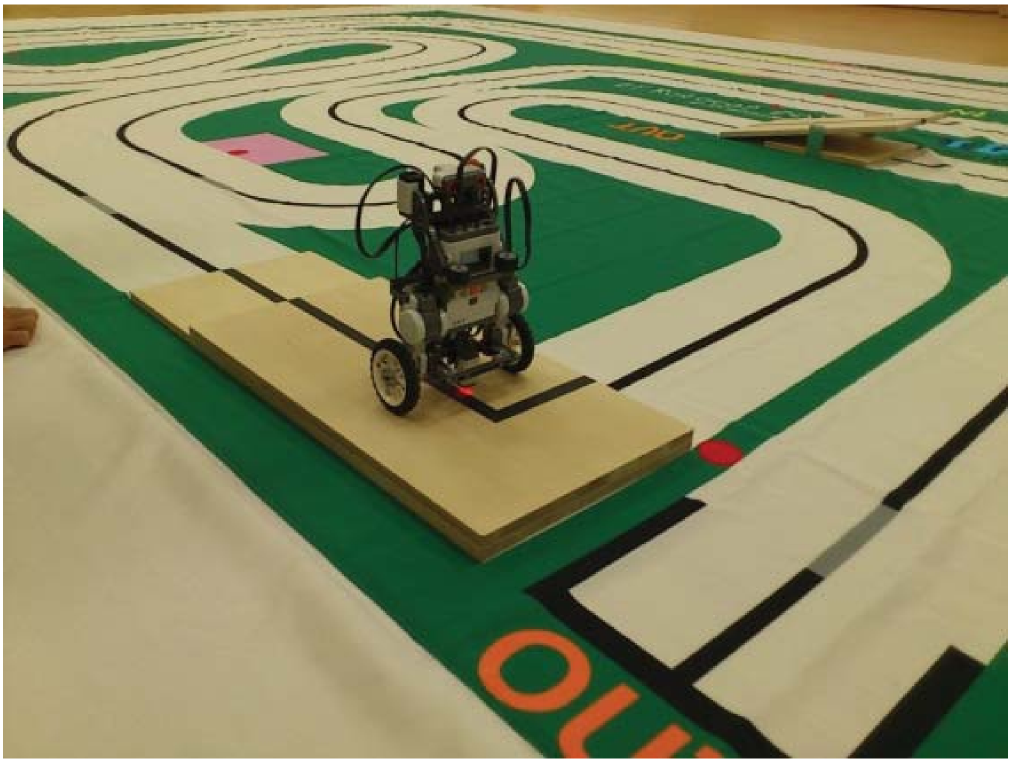
Platform Research Group

- Started to work in 2000.
- Members
  - CORE, NDD, CIC, ZUKEN ELMIC, Oriental Motor, Upwind Technology etc.
- Advisors
  - Tetsuo Kotoku Dr.Eng. The National Institute of Advanced Industrial Science and Technology (AIST)
  - Naoyuki Takesue, Associate Professor, Intelligent System Design, Tokyo Metroporitan University
  - Akihito Sano, Professor, Department of Mechanical Engineering, Department of Engineering Physics, Electronics and Mechanics, Nagoya Institute of Technology
  - Junji Furusho, Professor, Faculty of Engineering, Department of Management Information Science, Fukui University of Technology
- Activities
  - Research and study into technological and business trends in the platforms that serve as the common foundation for our business.
  - Drafting of the specifications of “OpenEL for Robots”, a software platform for robotics that is being proposed by JASA.

Embedded Technology Robot Software Design Contest

- In 2002, former OMG Japan started the contest named "UML Robot Contest".
- Since 2004, JASA has hosted the contest named “ET Robot Contest”.
- This is a software development contest that helps to educate embedded software engineers. Engineer training and competition are held in 11 areas within Japan. At the comprehensive exhibit of embedded systems technology, “Embedded Technology,” which is held in November, a championship tournament is held between winning teams from each area.
- Upwind Technology, Inc. is one of Bronze sponsor and provide a development environment.
  - GNUWing™ for ARM– Embedded system development toolchain
  - UTOS ® for LEGO NXT - Real-Time Operating System
    - NOTE:UTOS is a registered trademark of Upwind Technology, Inc..
  - OpenEL™ for LEGO NXT - Open Embedded Library for Robots
    - NOTE:OpenEL is a trademark of JASA.

Embedded Technology Robot Software Design Contest



What's OpenEL(Embedded Library)?

- OpenEL for Robots is an open platform to standardize the specifications of the software implementation of robotics and control systems.
- Currently, porting existing software on different systems, including the device driver in the development of embedded systems has been considerable effort required.
- For example, turning on the LED or just to operate the motor on different hardware, there may spend many days.
- Because an application program interface to control the output of the sensors and motors, were each uniquely defined by the device manufacturer, has been implemented since.

What's OpenEL(Embedded Library)?

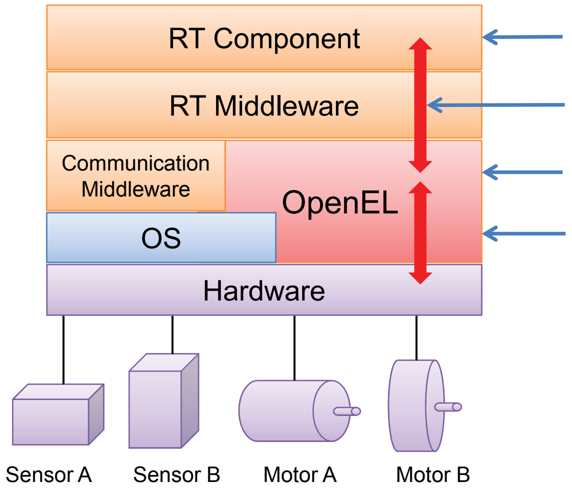
- Therefore, JASA propose to unify these interfaces which were different for each device manufacturer.
- JASA focus on robotics and control systems, has started drafting specifications OpenEL for Robots.
- In OpenEL for Robots, by the base portion of the software platform for robotics and control systems, and aims to enable applications running on different hardware too soon.
- This increases the portability and reusability of the software, resulting in improved quality, lower costs and lead to improved productivity is expected to improve convenience for users and developers.

Why OpenEL?

- Some Problems in ROS and RT Middleware
  - Limitation of Software Development Environment
    - OS, Languages etc.
  - No standard to use Sensors and Motors etc.
  - No standard to use A/D, D/A, DIO etc.
- In non-competitive areas, we often have a lot of trouble.
- OpenEL solves above problems.

Concept and Architecture of OpenEL platform

- Specifically, OpenEL is API (Application Program Interface) standardized on the layer below the middleware.
  - Naming Convention : el + Device + Command (ex. elMotorSetAngle())
- It is a mechanism for device control, such as the output to the motor, the input from the sensor and so on.
- We are targeting only implementation, the bottom of V-model.



RT Component

RT Middleware

Communication Middleware

OpenEL

OS

Hardware

Sensor A

Sensor B

Motor A

Motor B

If you use OpenEL , you can change the device anytime. No need to rewrite the application code.

elMotorPowerOn(), elMotorSetAngle(), elMotorPowerOff(), elMotorSetSpeed() etc.

OpenEL layer absorb the difference of devices such as sensors and motors.

The device vendors or we implement OpenEL layer.

Advantage of OpenEL

- OpenEL is new Application Programming Interface for Robots and Embedded Devices.
- OpenEL redefine embedded system programming.
- Very Easy Naming Convention for programmer.
- OpenEL standard to use Sensors and Motors etc.
- OpenEL standard to use A/D, D/A, DIO etc.
- In non-competitive areas, we will never have any trouble.
- OpenEL improves software portability, reusability and productivity.

Activities of OpenEL

- JASA Platform Research Group started work on the specification of OpenEL for Robots in April, 2011.
- JASA announced OpenEL on 16 November 2011.
- JASA released OpenEL version 0.1 in April, 2012.
  - The initial target robot is LEGO Mindstorms NXT.
  - Open Source Software(BSD License)
  - You can download from www.jasa.or.jp/top/activity/platform.html
- JASA Platform Research Group is working on the specification of OpenEL version 1.0.
- JASA Platform Research Group introduce OpenEL at OMG Technical meeting in December, 2012.

OpenEL API Version 0.1.1

Macros

```
#define    EL_TRUE    1
#define    EL_FALSE    0
#define    EL_NXT_PORT_A    0
#define    EL_NXT_PORT_B    1
#define    EL_NXT_PORT_C    2
#define    EL_NXT_PORT_S1    0
#define    EL_NXT_PORT_S2    1
#define    EL_NXT_PORT_S3    2
#define    EL_NXT_PORT_S4    3
#define    OPENEL_MAJOR    0
#define    OPENEL_MINOR    1
#define    OPENEL_VERSION    "OpenEL 0.1.1"
```

Typedefs

```
typedef signed char           ELChar
typedef unsigned char         ELUChar
typedef signed char           ELInt8
typedef signed short          ELInt16
typedef signed int            ELInt32
typedef signed long long      ELInt64
typedef unsigned char         ELUInt8
typedef unsigned short        ELUInt16
typedef unsigned int          ELUInt32
typedef unsigned long long    ELUInt64
typedef float                 ELFloat32
typedef double                ELFloat64
typedef unsigned char         ELBool
```

OpenEL API Version 0.1.1

■ Functions for Motors

```
ELFloat64  elMotorGetAngle (ELUInt32 portid)
ELFloat64  elMotorSetAngle (ELUInt32 portid, ELFloat64 angle, ELInt32 speed)
void       elMotorResetEncoder (ELUInt32 portid)
ELInt32    elMotorGetSpeed (ELUInt32 portid)
void       elMotorSetSpeed (ELUInt32 portid, ELInt32 speed)
ELBool     elMotorGetBrake (ELUInt32 portid)
void       elMotorSetBrake (ELUInt32 portid, ELBool brake)
```

■ Functions for Sensors

```
ELUInt16  elGyroSensorGetValue (ELUInt32 portid)
ELUInt16  elGyroSensorGetOffset (ELUInt32 portid)
void      elGyroSensorSetOffset (ELUInt32 portid, ELUInt16 offset)
ELUInt16  elLightSensorGetValue (ELUInt32 portid)
ELBool    elLightSensorGetLED (ELUInt32 portid)
void      elLightSensorSetLED (ELUInt32 portid, ELBool light)
ELBool    elTouchSensorGetState (ELUInt32 portid)
ELUInt16  elBatteryGetVoltage (void)
ELBool    elSpeakerOutput (ELUInt32 freq, ELUInt32 ms, ELUInt32 vol)
void      elSonarSensorInitialize (ELUInt32 portid)
void      elSonarSensorTerminate (ELUInt32 portid)
ELInt32   elSonarSensorGetValue (ELUInt32 portid)
```

OpenEL API Version 0.1.1

■ Functions for Bluetooth

```
void      elBluetoothInitializeMaster (const ELUChar *addr, const char *pin)
void      elBluetoothInitializeSlave (const char *pin)
void      elBluetoothTerminate (void)
ELUInt32 elBluetoothSendData (const void *buf, ELUInt32 offset, ELUInt32 len)
ELUInt32 elBluetoothReceiveData (void *buf, ELUInt32 offset, ELUInt32 len)
ELBool    elBluetoothGetDeviceName (char *name)
ELBool    elBluetoothSetDeviceName (const char *name)
ELInt32   elBluetoothGetStatus (void)
ELInt16   elBluetoothGetSignalStrength (void)
```

Example of elMotorSetAngle()

```
ELFloat64 elMotorSetAngle ( ELUInt32       portid,
                            ELFloat64      angle,
                            ELInt32        speed
                          )
```

Rotates the motor to the specified angle.

If it is unable to do so, this function is finished.

A motor angle is defined as base angle(0radian) at the time of starting program or doing elMotorResetEncoder.

If This return value is difference between the parameter angle and the actual rotation angle.

Parameters:

[in]	portid	the port id of the motor.
[in]	angle	the angle specifies to the encoder. (unit: radian)
[in]	speed	the pwm value specifies to the motor. (range: [-100,100])

Returns:

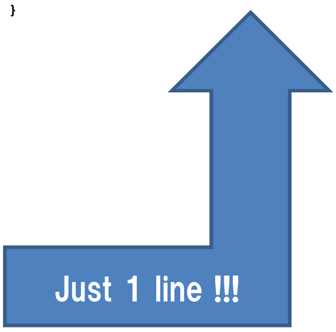
the difference between the parameter angle and the actual rotation angle (unit: radian)

Example to use elMotorSetAngle()

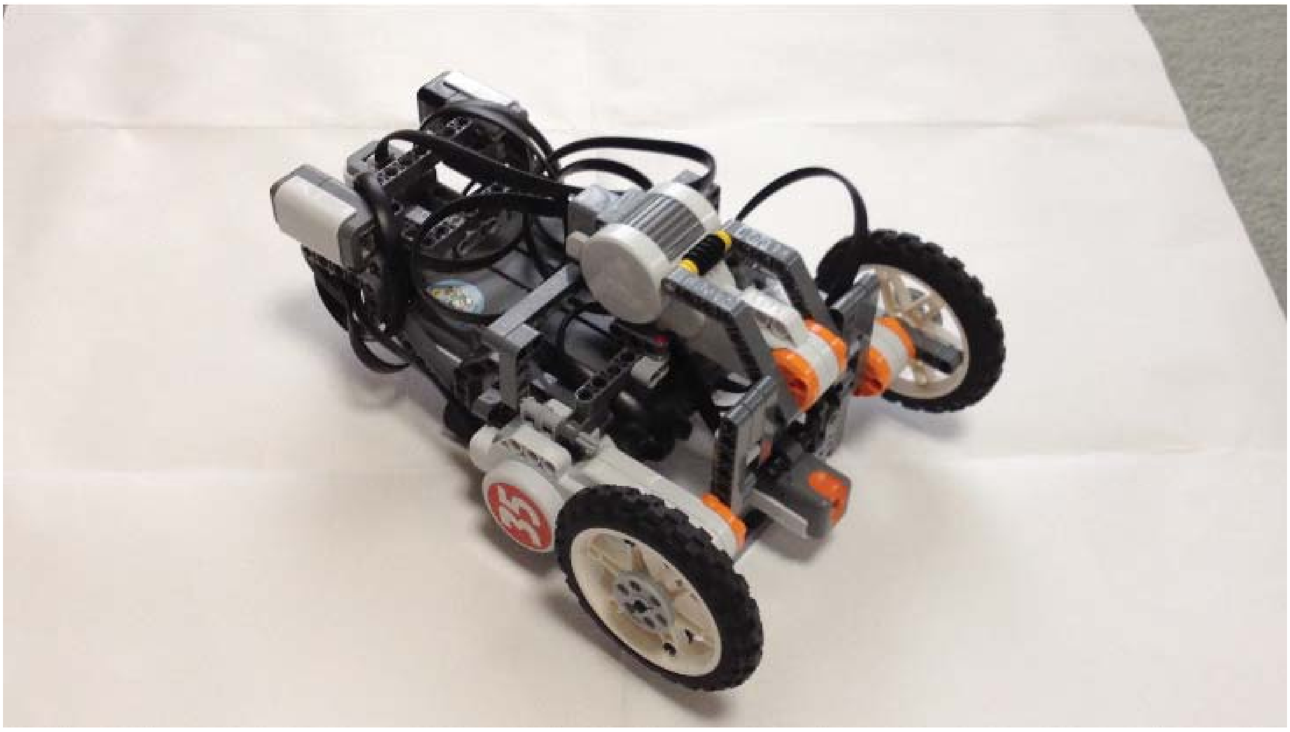
```
ELFloat64 elMotorSetAngle(ELUInt32 portid, ELFloat64
angle, ELInt32 speed)
{
  ELFloat64 pwm;
  ELInt32 loop_count=1, motor_count=0,
motor_count_prev=0;
  ELInt64 sum_motor_count_diff=0;

  motor_count = nxt_motor_get_count(portid);
  pwm = angle * 180.0 / PI - (ELFloat64)motor_count;
  while (fabs(pwm) > ANGLE_MARGIN) {
    if(loop_count == LOOP_MOTOR_COUNT){
      if(sum_motor_count_diff < 2)
        break;
      loop_count = 0;
      sum_motor_count_diff = 0;
    }
 pwm *= P_GAIN;
    if (pwm > speed)
      pwm = speed;
    else if (pwm < -speed)
      pwm = -speed;
    nxt_motor_set_speed(portid, pwm, 1);
    motor_count = nxt_motor_get_count(portid);
    sum_motor_count_diff += abs(motor_count -
motor_count_prev);
    pwm = angle * 180.0 / PI - (ELFloat64)motor_count;
    loop_count++;
    motor_count_prev = motor_count;
  }
  nxt_motor_set_speed(portid, 0, 1);
  return angle - (ELFloat64)motor_count / 180.0 * PI;
}
```

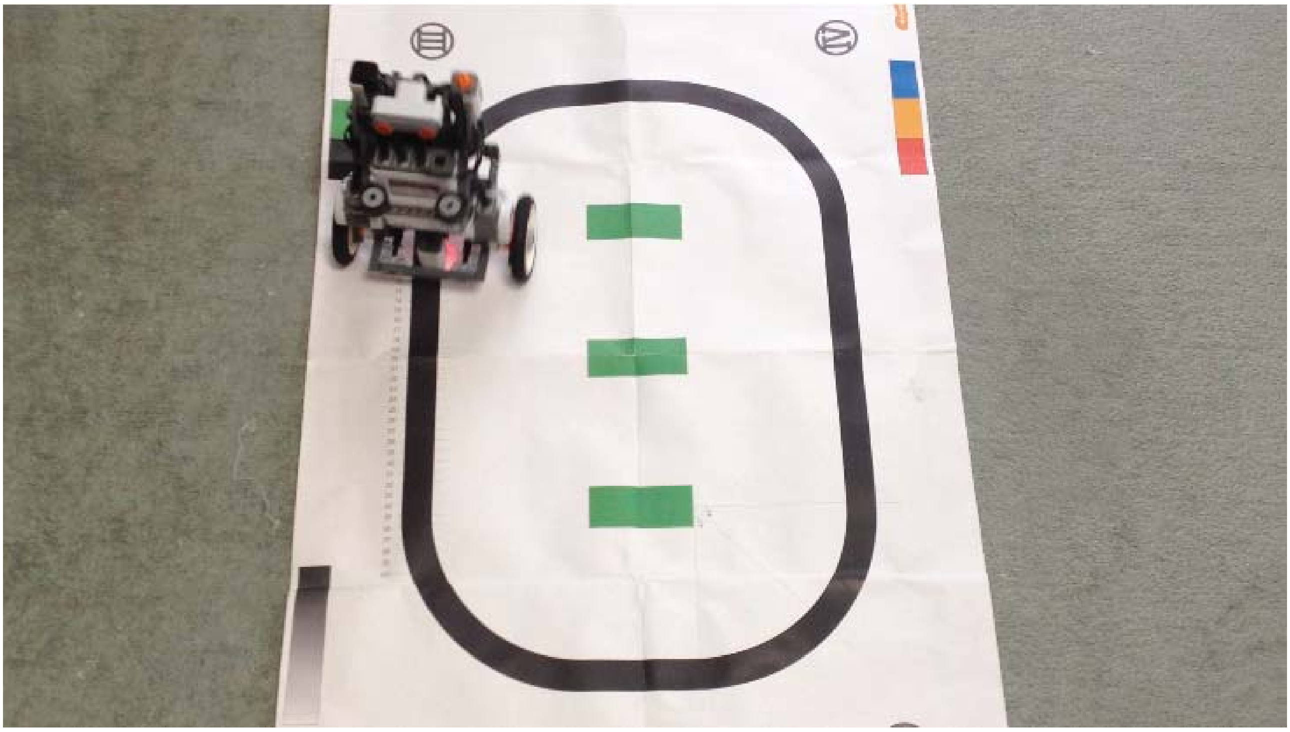
```
static void tail_control(signed int angle)
{
  double rad = (double)angle * PI / 180.0;
  elMotorSetAngle(EL_NXT_PORT_A, rad, 50);
}
```



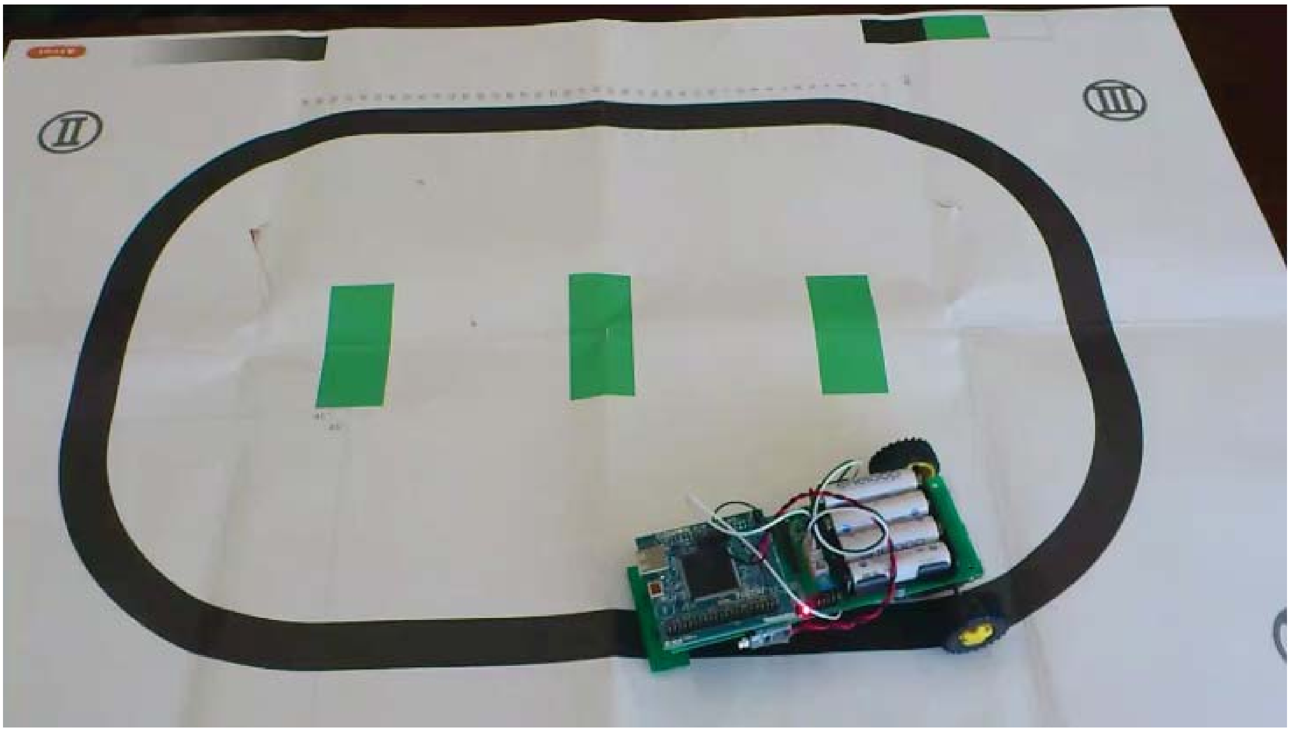
Demo video of tail_control() using elMotorSetAngle()



Demo video of OpenEL for LEGO Mindstorms NXT

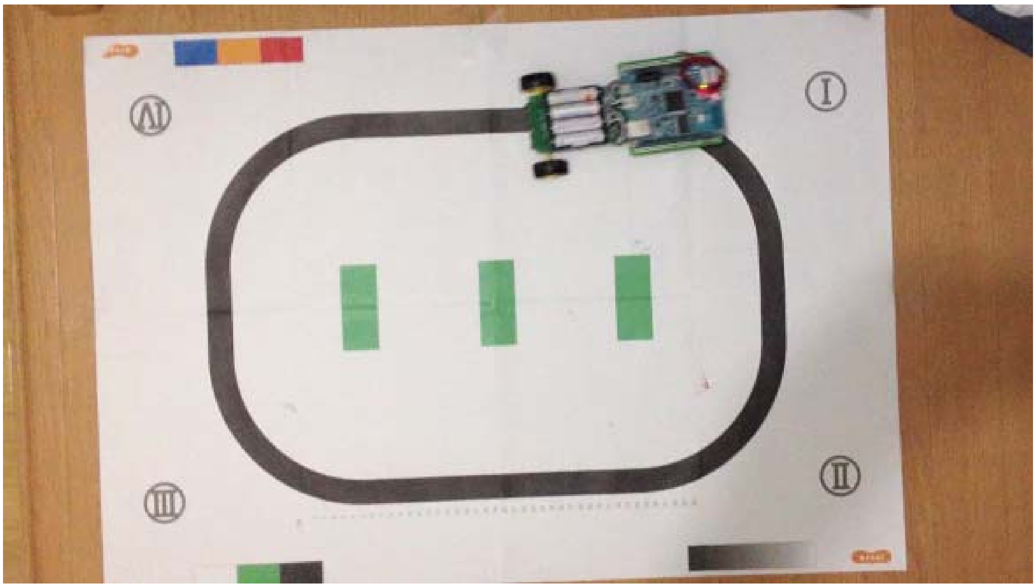


Demo video of OpenEL for FM3(Cortex-M3)



Demo video of OpenEL for Renesas RX63N

Same application runs on both FM3 and RX63N without any changes!



OpenEL Roadmap

- JASA will release OpenEL version 1.0 in March, 2013.
  - The target robots are Factory Automation robots etc..
  - The supported motors are Oriental motor etc..
  - The supported devices are A/D, D/A, DIO boards, etc.
  - The supported OS are UTOS, Linux, Windows 7.
- JASA will continue to update OpenEL.
  - More motors, sensors, devices and Robots
  - More OS like Android etc.
  - Support for safety standards, IEC 61508, ISO 13482 etc.
- JASA want to start standardization work at OMG next year.
- JASA are looking for people who agree with OpenEL in the world.

Conclusion

- OpenEL for Robots is an open platform to standardize the specifications of the software implementation of robotics and control systems.
- OpenEL is API (Application Program Interface) standardized on the layer below the middleware.
- OpenEL is a mechanism for device control, such as the output to the motor, the input from the sensor and so on.
- OpenEL increases the portability and reusability of the software, resulting in improved quality, lower costs and lead to improved productivity is expected to improve convenience for users and developers.
- JASA want to start standardization work at OMG.
- JASA are looking for people who agree with OpenEL in the world.

Thank you!

OpenEL

Title : Introduction of OpenEL(Embedded Library) for Robots

Date of publication : December 11, 2012

Publisher : Japan Embedded Systems Technology Association (JASA）
Head Office : 1-8-12 Nihonbashi-Hamacho, Chuo-ku, Tokyo, Japan
TEL : +81-3-5821-7973
FAX : +81-3-5821-0444
URL : http://www.jasa.or.jp/top/en/index.html

openel.h File Reference

Macros

#define **EL_TRUE** 1
#define **EL_FALSE** 0
#define **EL_NXT_PORT_A** 0
#define **EL_NXT_PORT_B** 1
#define **EL_NXT_PORT_C** 2
#define **EL_NXT_PORT_S1** 0
#define **EL_NXT_PORT_S2** 1
#define **EL_NXT_PORT_S3** 2
#define **EL_NXT_PORT_S4** 4
#define **EL_BT_NO_INIT** 4
#define **EL_BT_INITIALIZED** 5
#define **EL_BT_CONNECTED** 6
#define **EL_BT_STREAM** 7
#define **OPENEL_MAJOR** 0
#define **OPENEL_MINOR** 1
#define **OPENEL_PATHLEVEL** 1
#define **OPENEL_VERSION** "OpenEL 0.1.1"

Typedefs

typedef signed char	**ELChar**
typedef unsigned char	**ELUChar**
typedef signed char	**ELInt8**
typedef signed short	**ELInt16**
typedef signed int	**ELInt32**
typedef signed long long	**ELInt64**
typedef unsigned char	**ELUInt8**
typedef unsigned short	**ELUInt16**
typedef unsigned int	**ELUInt32**
typedef unsigned long long	**ELUInt64**
typedef float	**ELFloat32**
typedef double	**ELFloat64**
typedef unsigned char	**ELBool**

Functions

ELFloat64	**elMotorGetAngle** (**ELUInt32** portid)
ELFloat64	**elMotorSetAngle** (**ELUInt32** portid, **ELFloat64** angle, **ELInt32** speed)
void	**elMotorResetEncoder** (**ELUInt32** portid)
ELInt32	**elMotorGetSpeed** (**ELUInt32** portid)
void	**elMotorSetSpeed** (**ELUInt32** portid, **ELInt32** speed)
ELBool	**elMotorGetBrake** (**ELUInt32** portid)
void	**elMotorSetBrake** (**ELUInt32** portid, **ELBool** brake)
ELUInt16	**elGyroSensorGetValue** (**ELUInt32** portid)
ELUInt16	**elGyroSensorGetOffset** (**ELUInt32** portid)
void	**elGyroSensorSetOffset** (**ELUInt32** portid, **ELUInt16** offset)
ELUInt16	**elLightSensorGetValue** (**ELUInt32** portid)
ELBool	**elLightSensorGetLED** (**ELUInt32** portid)
void	**elLightSensorSetLED** (**ELUInt32** portid, **ELBool** light)
ELBool	**elTouchSensorGetState** (**ELUInt32** portid)
ELUInt16	**elBatteryGetVoltage** (void)
ELBool	**elSpeakerOutput** (**ELUInt32** freq, **ELUInt32** ms, **ELUInt32** vol)
void	**elSonarSensorInitialize** (**ELUInt32** portid)
void	**elSonarSensorTerminate** (**ELUInt32** portid)
ELInt32	**elSonarSensorGetValue** (**ELUInt32** portid)
void	**elBluetoothInitializeMaster** (const **ELUChar** *addr, const char *pin)
void	**elBluetoothInitializeSlave** (const char *pin)
void	**elBluetoothTerminate** (void)
ELUInt32	**elBluetoothSendData** (const void *buf, **ELUInt32** offset, **ELUInt32** len)
ELUInt32	**elBluetoothReceiveData** (void *buf, **ELUInt32** offset, **ELUInt32** len)
ELBool	**elBluetoothGetDeviceName** (char *name)
ELBool	**elBluetoothSetDeviceName** (const char *name)
ELInt32	**elBluetoothGetStatus** (void)
ELInt16	**elBluetoothGetSignalStrength** (void)

Macro Definition Documentation

#define EL_BT_CONNECTED 6

#define EL_BT_INITIALIZED 5

#define EL_BT_NO_INIT 4

#define EL_BT_STREAM 7

#define EL_FALSE 0

#define EL_NXT_PORT_A 0

#define EL_NXT_PORT_B 1

#define EL_NXT_PORT_C 2

#define EL_NXT_PORT_S1 0

#define EL_NXT_PORT_S2 1

#define EL_NXT_PORT_S3 2

#define EL_NXT_PORT_S4 4

#define EL_TRUE 1

#define OPENEL_MAJOR 0

#define OPENEL_MINOR 1

#define OPENEL_PATHLEVEL 1

#define OPENEL_VERSION "OpenEL 0.1.1"

Typedef Documentation

typedef unsigned char ELBool

typedef signed char ELChar

typedef float ELFloat32

typedef double ELFloat64

typedef signed short ELInt16

typedef signed int ELInt32

typedef signed long long ELInt64

typedef signed char ELInt8

typedef unsigned char ELUChar

typedef unsigned short ELUInt16

typedef unsigned int ELUInt32

typedef unsigned long long ELUInt64

typedef unsigned char ELUInt8

Function Documentation

ELUInt16 elBatteryGetVoltage (void)

Gets the battery voltage.

Returns:
the current battery voltage.

ELBool elBluetoothGetDeviceName (char * name)

Gets the Bluetooth device name.

Parameters:
`[in]` **name** the head address of the buffer that stores the device name.

Returns:
true if succeeded to get the device name, false otherwise.

ELInt16 elBluetoothGetSignalStrength (void)

Gets the signal strength of the Bluetooth.
If the connection has not been established, it returns -1.

Returns:
the strength of the Bluetooth. (range: [0,100])

ELInt32 elBluetoothGetStatus (void)

Gets the status of the connection of the Bluetooth.

List of constants representing the status of Bluetooth connection:

- **EL_BT_NO_INIT(Uninitialized state)**
- **EL_BT_INITIALIZED(Initialized state)**
- **EL_BT_CONNECTED(Connection established state)**
- **EL_BT_STREAM(State data can be transmitted and received)**

Returns:
the constant representing the status of Bluetooth connection.

void elBluetoothInitializeMaster (const ELUChar * addr, const char * pin)

Initializes the Bluetooth as a master device.

Parameters:

[in]	**addr**	the head address of the Bluetooth device address of a slave device.
[in]	**pin**	the head address of the pin code for the passkey exchange.

void elBluetoothInitializeSlave (const char * pin)

Initializes the Bluetooth as a slave device.

Parameters:

[in]	**pin**	the head address of the pin code for the passkey exchange.

ELUInt32 elBluetoothReceiveData (void * buf, ELUInt32 offset, ELUInt32 len)

Receives the data to the buffer via the Bluetooth.

Parameters:

[in]	**buf**	the head address of the receiving data buffer.
[in]	**offset**	the offset of the receiving data buffer.
[in]	**len**	the receiving data buffer size.

Returns:

the number of bytes of data received.

ELUInt32 elBluetoothSendData (const void * buf, ELUInt32 offset, ELUInt32 len)

Sends the data in the buffer via the Bluetooth.

Parameters:

[in]	**buf**	the head address of the sending data buffer.
[in]	**offset**	the offset of the sending data buffer.
[in]	**len**	the sending data buffer size.

Returns:

the number of bytes of data sent.

ELBool elBluetoothSetDeviceName (const char * name)

Sets the Bluetooth device name.

Parameters:

[in]	**name**	the device name.

Returns:

true if succeeded to set the device name, false otherwise.

void elBluetoothTerminate (void)

Ternimates the Bluetooth.
This function can be used in both the master device and slave devices.

ELUInt16 elGyroSensorGetOffset (ELUInt32 portid)

Gets the offset value of the gyro sensor.

Parameters:

[in]	**portid**	the port id of the gyro sensor

Returns:

the offset value of the gyro sensor.

ELUInt16 elGyroSensorGetValue (ELUInt32 portid)

Gets the value of the gyro sensor.

Parameters:

[in]	**portid**	the port id of the gyro sensor.

Returns:

the value of the gyro sensor.

void elGyroSensorSetOffset (ELUInt32 portid, ELUInt16 offset)

Sets the offset value of the gyro sensor.

Parameters:

[in]	**portid**	the port id of the gyro sensor.
[in]	**offset**	the offset value which is set to the gyro sensor.

ELBool elLightSensorGetLED (ELUInt32 portid)

Gets whether the LED of light sensor is turned on.

Parameters:

`[in]` **portid** the port id of the light sensor.

Returns:

true if the LED is turned on, false otherwise.

ELUInt16 elLightSensorGetValue (ELUInt32 portid)

Gets the value of the light sensor.

Parameters:

`[in]` **portid** the port id of the light sensor.

Returns:

the value of the light sensor.

void elLightSensorSetLED (ELUInt32 portid,
ELBool light
)

Sets the lighting state of the LED of light sensor.

Parameters:

`[in]` **portid** the port id of the light sensor.
`[in]` **light** true if turn on the LED of light sensor, false otherwise.

ELFloat64 elMotorGetAngle (ELUInt32 portid)

Gets the angle of the encoder of the motor.

Parameters:

`[in]` **portid** the port id of the motor.

Returns:

the angle of the encoder. (unit: radian)

ELBool elMotorGetBrake (ELUInt32 portid)

Gets whether the brake of motor is enabled.

Parameters:

`[in]` **portid** the port id of the motor.

Returns:

true if the brake of motor is enabled, false otherwise.

ELInt32 elMotorGetSpeed (ELUInt32 portid)

Gets the rotational velocity(PWM value) which is set to the motor.

Parameters:

`[in]` **portid** the port id of the motor.

Returns:

the rotational velocity which is set to the motor. (range: [-100,100])

void elMotorResetEncoder (ELUInt32 portid)

Resets the encoder value, and set the current angle as a criteria(zero radian).

Parameters:

`[in]` **portid** the port id of the motor.

ELFloat64 elMotorSetAngle (ELUInt32 portid,
ELFloat64 angle,
ELInt32 speed
)

Rotates the motor to the specified angle.
If it is unable to do so, this function is finished.
A motor angle is defined as base angle(0radian) at the time of starting program or doing elMotorResetEncoder.
If This return value is difference between the parameter angle and the actual rotation angle.

Parameters:

`[in]` **portid** the port id of the motor.
`[in]` **angle** the angle specifies to the encoder. (unit: radian)
`[in]` **speed** the pwm value specifies to the motor. (range: [-100,100])

Returns:

the difference between the parameter angle and the actual rotation angle (unit: radian)

void elMotorSetBrake (ELUInt32 portid,
ELBool brake
)

Enables/Disables the brake of motor.

Parameters:

`[in]` **portid** the port id of the motor.
`[in]` **brake** true enables the brake of motor, and false disables.

void elMotorSetSpeed (ELUInt32 portid, ELInt32 speed)

Sets the rotational velocity(PWM value) to the motor.

Parameters:

- `[in]` **portid** the port id of the motor.
- `[in]` **speed** the rotational velocity which is set to the motor. (range: [-100,100])

ELInt32 elSonarSensorGetValue (ELUInt32 portid)

Gets the value of sonar sensor.
If the sonar sensor has not been initialized, it returns -1.

Parameters:

- `[in]` **portid** the port id of the sonar sensor.

Returns:

the measured distance. (unit: cm)

void elSonarSensorInitialize (ELUInt32 portid)

Initializes the sonar sensor.
This function should be called only once before using the sonar sensor.

Parameters:

- `[in]` **portid** the port id of the sonar sensor.

void elSonarSensorTerminate (ELUInt32 portid)

Terminates the sonar sensor.
This function should be called only once before terminating the use of the sonar sensor.

Parameters:

- `[in]` **portid** the port id of the sonar sensor.

ELBool elSpeakerOutput (ELUInt32 freq, ELUInt32 ms, ELUInt32 vol)

Outputs the tone sound from the speaker.

Parameters:

- `[in]` **freq** the frequency. (unit: Hz)
- `[in]` **ms** the output duration. (unit: 10ms)
- `[in]` **vol** the volume. (unit: %)

Returns:

true if the output was succeeded, false otherwise.

ELBool elTouchSensorGetState (ELUInt32 portid)

Gets the touch sensor state.

Parameters:

- `[in]` **portid** the port id of the touch sensor.

Returns:

true if the touch sensor is on, false otherwise.

Generated on Wed Jul 4 2012 01:21:01 for OpenEL API by doxygen 1.8.1.1

Robotics-DTF Plenary Meeting
Opening Session

December 12, 2012

Burlingame, CA, USA

Hyatt Regency San Francisco Airport

Approval of Minutes

Meeting Quorum : 3

AIST, Change Vision, ETRI, Honda, JARA, UEC,

Minutes taker(s):

- **Takashi Suehiro**
- **Seung-Woog Jung**

Minutes review

Cambridge Meeting Summary

Robotics Plenary: (10 participants)

- The DDC4RTC submission was adopted in MARS-PTF, and charter DDC4RTC-FTF.
- The RoIS-FTF Final Report was reviewed in AB, but was voted by e-mail. (There was no time to review the revised report)
- We decide to plan the OMG Robotics Information Day in Burlingame Technical Meeting.

Robotics Information Day

34 participants

- Antonello Corevola (Honda-RI-EU)
- Beom-Su Seo (ETRI)
- Byung-Tae Chun (Hankyoung Univ.)
- Chuck Zublic (NGC)
- Daniel Siegl (LieberLiever)
- Geoffrey Biggs (AIST)
- Gerardo Pardo-Castellote (RTI)
- Hajime Ueno (Fuji Xerox)
- Hugues Vincent (Thales)
- Isao Hara (AIST)
- Isashi Uchida (IPA)
- Julien Deantoni (INRIA)
- Kenichi Nakamura(JASA)
- Kenji Hiranabe (ChangeVision)
- Koji Kamei (ATR)
- Makoto Sekiya (Honda)
- Miwako Doi (Toshiba)
- Noriaki Ando (AIST)
- Paul Evans (SwRI)
- Russell Peak (Georgia Tech)
- Seiichi Shin (UEC)
- Se-Kyung Song (Future Robot)
- Seungbin Moon (Sejong Univ.)
- Seung-Woog Jung (ETRI)
- Su-Young Chi (ETRI)
- Takashi Suehiro (UEC)
- Takashi Tsubouchi (Univ. of Tsukuba)
- Tetsuo Kotoku (AIST)
- Tomomasa Sato (Univ. of Tokyo)
- Toshihiro Okamura (ChangeVison)
- Toshiki Iwanaga (ChangeVision)
- Toyotaka Torii (Honda)
- Young-Jo Cho (ETRI)
- Yutaka Matsuno (Nagoya Univ.)

Reston Meeting Summary

Robotics Plenary: (6 participants)

- The DDC4RTC submission was reviewed and adopted in MARS-PTF, but rejected in AB review.
- The deadline of DDC4RTC revised submission extended to the upcoming Cambridge Meeting.
- One presentation;
  "A Trial Approach for Automation in Open Cut Mine", Takashi Tsubouchi (Univ. of Tsukuba)

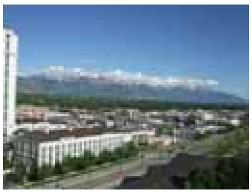
Santa Clara Meeting Summary

Robotics Plenary: (18 participants)

- **3 Talks**
  - "The Legal Aspects of Autonomous cars", Bryant Walker Smith (Stanford Univ.)
  - "Proteus: An ontology for experimental validation of solutions to robotic problems", Laurent Rioux (THARES) [robotics/2011-12-04]
  - "Domestic Standardization Activity for Standardizing Voice Interface for Service Robots in Japan", Yosuke Matsusaka(AIST) [robotics/2011-12-05]
- **2 WG Reports**
  - Robotic Infrastructure WG **[robotics/2011-12-09]**
  - Robotic Functional Services WG **[robotics/2011-12-07]**
- The deadline of the DDC4RTC revised submission was extended to the upcoming Reston
- The final report of Robotic Technology Component (RTC-1.1) was accepted to issue.

Salt Lake City Meeting Summary

Revised Submission to Robotic Interaction Service (RoIS) Framework [robotics/2011-05-01,02,03,04,05]



Robotics Plenary: (17 participants)

- –3 Talks
  - “OPRoS: Open Platform for Robotic Services”, Hong Seong Park, Kangwon National University [robotics/2011-06-09]
  - “Conformance Testing Method for Robotic Software Components”, Mi-Sook Kim, Kangwon National University [robotics/2011-06-10]
  - “Robotics Technology Applied to Great East Japan Earthquake”, Miwako DOi, Toshiba [robotics/2011-06-11]
- –2 WG Reports
  - Robotic Infrastructure WG [robotics/2011-06-12]
  - Robotic Functional Services WG [robotics/2011-06-13]

Kissimmee Meeting Summary

No Plenary Meeting:

Robotic Functional Services WG :
(4 participants)

-

Agenda Review

Mon:
08:40-12:00 Robotics-DTF Plenary (with GoToMeeting)
13:00-18:00 Infrastructure WG (with GoToMeeting)
13:00-18:00 Service WG
Tue:
09:00-18:00 Infrastructure WG/ Service WG
Wed:
09:00-12:00 Infrastructure WG/ Service WG
14:00-16:00 Joint Plenary with MARS-PTF
16:00-18:00 Robotics-DTF Plenary
Thu:
09:00-18:00 WG activity follow-up

please check our up-to-date agenda
http://staff.aist.go.jp/t.kotoku/omg/RoboticsAgenda.pdf

Infrastructure WG Progress Report

Infra. WG, Robotics DTF

Noriaki Ando, AIST

robotics/2012-12-18

Overview

- Infrastructure WG meeting on Monday
- New Work Item has been proposed by Honda R&D
  - FSM component specification for RTC
  - Data port related specification for RTC
  - Data types for data port

FSM component

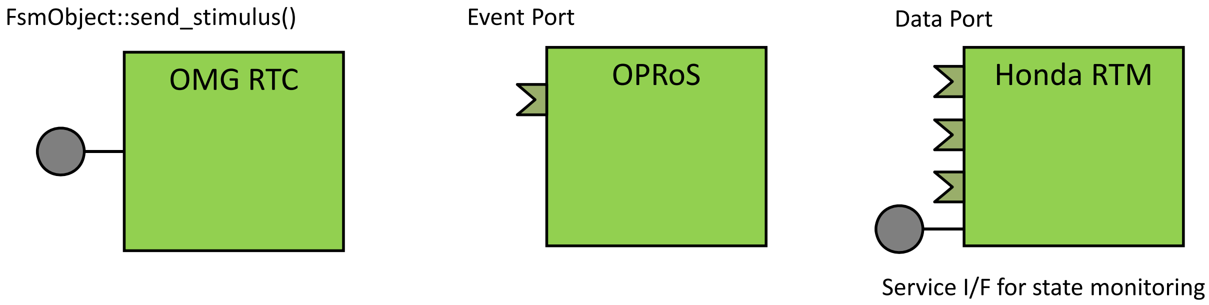
- RTC 1.1 FSM component is heavy and complicated.
- No FSM structure description and monitoring service interfaces are defined
- RTC spec's FSM receives event by an operation FsmObject::send_stimulus()

```
ReturnCode_t send_stimulus(in string message,
                                                 in ExecutionContextHandle_t exec_handle);
```

- Honda's RTM and OPRoS receive event by DataPort or EventPort

---

FSM components



- How do components receive events?
- How do applications and tools do obtain state machine profile?
- How do applications and tools do monitor state machine?

These are possible mandatory requirement of a RFP

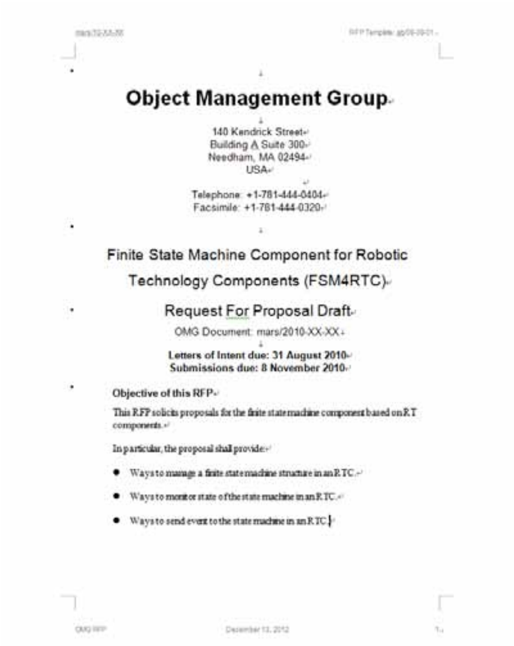
Standardization Process

- Update from RTC 1.1 to RTC 2.0
  - Add some features to FSM part in RTC spec.
  - This process would be managed by MARS.
- Create new specification
  - Create new spec of FSM related functionalities.
  - This spec might be reusable.
  - RTC's FSM still exists (but FSM component is optional)
  - This process would be managed by Robotics or MARS

For the FSM4RTC RFP

- Specification name
- Objective of the RFP
- General Requirements on Proposals
- Specific Requirements on Proposals
- Scope of proposals
- Mandatory requirements
- Optional requirements
- Timetable

should be decided



Object Management Group

140 Kendrick Street
Building A Suite 300
Needham, MA 02494
USA

Telephone: +1-781-444-0404
Facsimile: +1-781-444-0320

Finite State Machine Component for Robotic Technology Components (FSM4RTC)

Request For Proposal Draft

OMG Document: mars/2010-XX-XX

Letters of Intent due: 31 August 2010
Submissions due: 8 November 2010

Objective of this RFP

This RFP solicits proposals for the finite state machine component based on RT components.

In particular, the proposal shall provide:

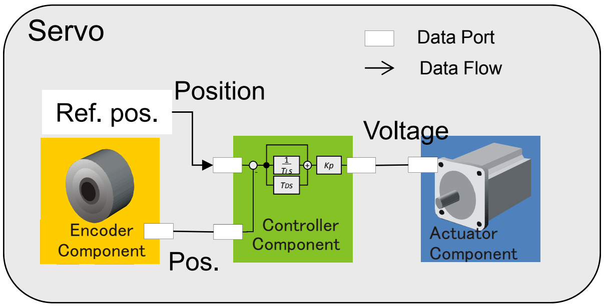
- Ways to manage a finite state machine structure in an RTC.
- Ways to monitor state of the state machine in an RTC.
- Ways to send event to the state machine in an RTC.

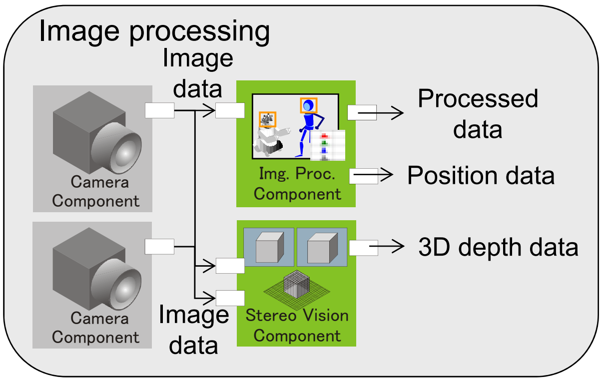
Data Port and Data Types

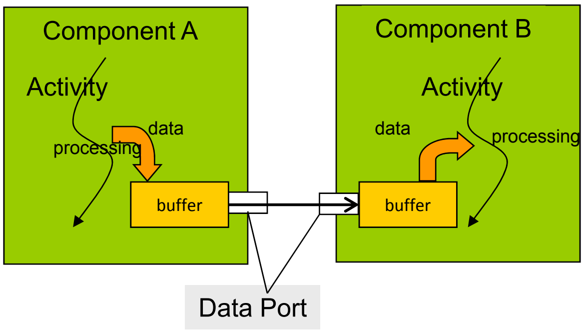
- OMG RTC only defines generic ports.
- But data centric ports are often required robotic applications.
- OpenRTM, OPRoS, Honda's RTM have data ports.
- Each concepts and functionalities are almost same.
- Data types for data ports are not standardized.

Data Port

- Port for data centric interaction
- Data stream
  - Position control
    - Ex. position, voltage
  - Image processing
    - Ex. image data
- For lower level processing for robot systems
- Same data-typed ports are connectable
- Dynamic connection/disconnection



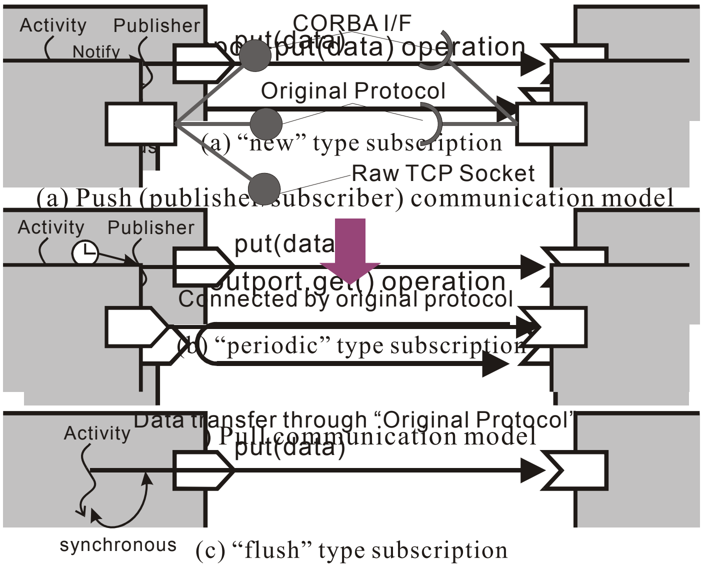




Data are sent automatically

Data Port Conection

- Connector Profile
  - Interface type
    - CORBA,TCP socket, other protocol, etc…
  - Data flow type
    - push/pull
  - Subscription type
    - Flush, New, Periodic



For the DCP4RTC RFP

- Specification name
- Objective of the RFP
- General Requirements on Proposals
- Specific Requirements on Proposals
- Scope of proposals
- Mandatory requirements
- Optional requirements
- Timetable

should be decided.

And should be divide data port spec and data type spec?

Object Management Group

140 Kendrick Street
Building A Suite 300
Needham, MA 02494
USA

Telephone: +1-781-444-0404
Facsimile: +1-781-444-0320

Data-centric Port for Robotic Technology Components (DCP4RTC)

Request For Proposal Draft

OMG Document: mars/2010-XX-XX

Letters of Intent due: 31 August 2010
Submissions due: 8 November 2010

Objective of this RFP

This RFP solicits proposals for the dynamic deployment and configuration of RT components.

In particular, the proposal shall provide:

- Ways to specify various profiles of data-centric ports for RTC.
- Ways to specify method of data transfer of data-centric ports for RTC.
- Basic data type definitions that are frequently used in robotic application.

Future Plan

- Discussion will be continued in robotics infra WG mailing list.

Schedule

- Submit RFP draft 4week before of Reston meeting
- 1st review in Reston (March 2013) meeting
- 2nd review in Berlin (June 2013) -> AB
- Initial submission in December 2013 meeting
- Merged submission in June (2014) meeting -> AB
- Starting FTF in June 2014 meeting
- FTF report in June 2015 meeting
- Specification will be published at the end of 2015

New Infrastructure WG Co-Chairs

Additional co-chairs

- BeomSu Seo (bsseo@etri.re.kr)
- Makoto Sekiya (Makoto_Sekiya@n.f.rd.honda.co.jp)

robotics/2012-12-19

Robotics-DTF Plenary Meeting
Wrap-up Session

December 12, 2012

Burlingame, CA, USA

Hyatt Regency San Francisco Airport

Document Number

robotics/2012-12-01 Final Agenda (Tetsuo Kotoku)
robotics/2012-12-02 Cambridge Meeting Minutes [approved] (Geoffrey Biggs and Seung-woog Jung)
robotics/2012-12-03 Intelligent RT Software Project - Natinal Project in Japan (Tomomasa Sato)
robotics/2012-12-04 Implementation of OPRoS to a human-friendly guide robot, FURO (Se-Kyeong Song)
robotics/2012-12-05 ISO Activity of Service Robot (Seungbin Moon)
robotics/2012-12-06 Disruptive Community Approach to Industrial Robotics Services (Paul Evans)
robotics/2012-12-07 Introduction to Robotic Technology Component (RTC-1.1) Specification (Geoffrey Biggs)
robotics/2012-12-08 An Inprementation of RoIS and RLS Spec. in Japan (Kenji Kamei)
robotics/2012-12-09 Dynamic Deployment and Configuration Standard for Robotic Technology Component: DDC4RTC (Noriaki Ando)
robotics/2012-12-10 Using SysML in a RTC-based Robotics Application : a case study with a demo (Kenji Hiranabe)
robotics/2012-12-11 A New Robotic Technology Middleware and Robotic Technology Component Interoperability demonstration (Makoto Sekiya)

Document Number (cont.)

robotics/2012-12-12 Implementation of RoIS to robots in ETRI (Su-Young Chi)
robotics/2012-12-13 Component Management in OPRoS (Seung-woog Jung)
robotics/2012-12-14 Cloud Networked Robotics and Acceleration Based Sensing (Miwako Doi)
robotics/2012-12-15 Introduction to OpenEL (Enbedded Library) for Robot (Kenichi Nakamura)
robotics/2012-12-16 OpenEL API specification ver.0.1.1 (Kenichi Nakamura)
robotics/2012-12-17 Opening Presentation (Tetsuo Kotoku)
robotics/2012-12-18 Infrastructure WG Progress Report (Noriaki Ando)
robotics/2012-12-19 Wrap-up Presentation (Tetsuo Kotoku)
robotics/2012-12-20 Roadmap for Robotics Activities (Tetsuo Kotoku)
robotics/2012-12-21 Next Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)
robotics/2012-12-22 DTC Report Presentation (Tetsuo Kotoku)
robotics/2012-12-23 Burlingame Meeting Minutes - DRAFT (Takashi Suehiro and Seung-woog Jung)

Organization (from December 12th, 2012)

Robotics-DTF
- **Tetsuo Kotoku (AIST, Japan)**
- **Young-Jo Cho (ETRI, Korea)**

- **Publicity Committee**
  - **Abheek Bose (ADA Software, India)**
- **Contacts Committee**
  - **Makoto Mizukawa (Shibaura-IT, Japan)**
  - **Young-Jo Cho (ETRI, Korea)**
  - **Yun Koo Chung (ETRI, Korea)**
- **Information Day 2013 Organizing Committee**
  - **Antonello Ceravola (Honda-RI-EU, Germany)**
  - **[Laurent Rioux (Thales, France)]**

Technical WGs

- **Infrastructure WG**
  - **Noriaki Ando（AIST, Japan)**
  - **Beom Su Seo (ETRI, Korea)**
  - **Makoto Sekiya (Honda, Japan)**
- **Robotic Functional Services WG**
  - **Su-Young *Chi* (ETRI, Korea)**
  - **Koji Kamei (ATR, Japan)**
  - **Toshio Hori (AIST, Japan)**
- **Modelling for Robotics WG**
  - **Takeshi Sakamoto (Technologic Arts)**
  - **Toby McClean (Zeligsoft)**

Call for volunteer

- Robotics-DTF Co-Chair

=> Postpone voting one more meeting

Next Meeting Agenda (tentative)

March 18-22, 2013 (Reston, VA, USA)

Monday:

WG activity

Tuesday:

WG activity (am)
Robotics-DTF Plenary Meeting (pm)
- **Guest and Member Presentation**
- **Contact reports**

Wednesday:

WG activity follow-up

Plenary Attendee (17 participants)

- Antonello Ceravola (Honda-RI-EU)
- Beom-Su Seo (ETRI)
- Byung-Tae Chun (Hankyoung Univ.)
- Geoffrey Biggs (AIST)
- Kenichi Nakamura (JASA)
- Kenji Hiranabe (ChangeVision)
- Koji Kamei (JARA/ATR)
- Makoto Sekiya (Honda)
- Noriaki Ando (AIST)
- Seung-Woog Jung (ETRI)
- Su-Young Chi (ETRI)
- Takashi Suehiro (UEC)
- Tetsuo Kotoku (AIST)
- Toshihiro Okamura (ChangeVision)
- Toshiki Iwanaga (ChangeVision)
- Toyotaka Torii (Honda)
- Young-Jo Cho (ETRI)

Roadmap for Robotics Activities

robotics/2012-12-20

Item	**Status**	**Jacksonville** **CANCEL** **Sep-2012**	**Burlingame** **CA** **Dec-2012**	**Reston** **VA** **Mar-2013**	**Berlin** **Germany** **Jun-2013**	**New Bunswick** **Sep-2013**	**Santa Clara** **CA** **Dec-2013**	**POC / Comment**
Flyer of Robotics-DTF **[Publicity Sub-Committee]**	**Suspended**							?
Finite State Machine Component for Robotic Technology Components (FSM4RTC) **[Robotic Infrastructure WG]**	**Planned**			Draft Review	RFP Issue		Initial Submission	Makoto Sekiya (Honda)
Data-Centric Port for Robotic Technology Components (DCP4RTC) **[Robotic Infrastructure WG]**	**Planned**			Draft Review	RFP Issue		Initial Submission	Makoto Sekiya (Honda)
OpenEL for Robitcs **[Hardware and Profile WG]?**	**Future**							Kenichi Nakamura (JASA)
etc…	**Future**							
Robotics Exhibition	**Planned**		**Exhibition**		**Exhibition**			
Robotics Information Day **[Technology Showcase]**	**Planned**		**Info. Day**		**Info. Day**			Antonello Corevola (Honda-RI-EU)
DDC4RTC Finalization Task Force	**In Process**				**FTF Report**			Noriaki Ando (AIST)
Related Events		Chu-suk (Special Holidays in Korea)				Chu-suk (Special Holidays in Korea)		

Robotics Domain Task Force Preliminary Agenda ver.0.0.1 robotics/2012-12-21

OMG Technical Meeting - Reston, VA, USA -- March 18-22, 2013

		TF/SIG		http://robotics.omg.org/		
		Host	Joint (Invited)	Agenda Item	Purpose	Room
Monday: Plannning Committee (pm)						
10:00	12:00			Robotics Infrastructure WG - Noriaki Ando(AIST), Makoto Sekiya(Honda), and Beom-Su Seo (ETRI)	RFP drafting	
12:00	13:00			**LUNCH**		
13:00	18:00			Architecture Board Plenary		
13:00	17:00			Robotics Infrastructure WG - Noriaki Ando, Makoto Sekiya, and Beom-Su Seo	RFP drafting	
Tuesday: Robotics Information Day						
9:00	12:00			Robotics Infrastructure WG - Noriaki Ando, Makoto Sekiya, and Beom-Su Seo	RFP drafting	
12:00	13:00			**LUNCH**		
13:00	15:00			Robotics Infrastructure WG - Noriaki Ando, Makoto Sekiya, and Beom-Su Seo	RFP drafting	
				Afternoon Break (30min)		
15:30	15:40	Robotics		Robotics-DTF Plenary Opening Session (minitues approval, minutes taker)	presentation and discussion	
16:45	17:00	Robotics		WG Reports and Discussion (Service WG, Infrastructure WG, Models in Robotics WG)	presentation and discussion	
17:00	17:15	Robotics		Contact Reports: - Makoto Mizukawa(Shibaura-IT), and Young-Jo Cho(ETRI)	Information Exchange	
17:15	17:30	Robotics		Robotics-DTF Plenary Wrap-up Session (**DTF Co-Chair Election**, Roadmap and Next meeting Agenda)	Robotics plenary closing	
17:30				Adjourn Information Day meeting		
Wednesday: WG activitiy						
9:00	12:00			Robotics WG activity follow-up	discussion	
12:00	14:00			**LUNCH and OMG Plenary**		
14:00	17:00			Robotics WG activity follow-up	discussion	
18:00	20:00			**OMG Reception**		
Thursday: WG activitiy						
12:00	13:00			**LUNCH**		
13:00	18:00			Architecture Board Plenary		
Friday						
8:30	12:00			AB, DTC, PTC		
12:00	13:00			**LUNCH**		
				Other Meetings of Interest		
Monday						
8:00	8:45	OMG		New Attendee Orientation		
Tuesday						
7:30	9:00	OMG		Liaison ABSC		

Please get the up-to-date version from http://staff.aist.go.jp/t.kotoku/omg/RoboticsAgenda.pdf

Robotics-DTF

Date: Friday, December 14th, 2012
Reporter : Toyotaka Torii
URL: http://robotics.omg.org/
email: robotics@omg.org

➢Highlights from this Meeting:

Robotics Information Day: (34 participants)

- 4 Keynotes, 4 Specification Introductions, 6 Talks (includes 2 demonstrations) [robotics/2012-12-03 - 16]

Robotics Demonstrations:

- AIST: Small humanoid robot (OpenRTM-aist)
- Change Vision: RTC application in SysML
- Honda R&D: two mobile robots (Interoperability of Honda RTM and OpenRTM-aist)

Robotics Plenary: (17 participants)

- 1 WG Report
  - Robotic Infrastructure WG [robotics/2012-12-18]
- Organizing Committee for Robotics Information Day 2013 in Berlin

---

Robotics-DTF

Date: Friday, December 14th, 2012
Reporter : Toyotaka Torii
URL: http://robotics.omg.org/
email: robotics@omg.org

Future deliverables (In-Process):

- Finite State Machine Component for Robotic Technology Components (FSM4RTC)
- Data-Centric Port for Robotic Technology Components (DCP4RTC)

Next Meeting (in Reston):

- 1st Discussion of FSM4RTC
- 1st Discussion of DCP4RTC
- Robotics Information Day Planning

After the Next Meeting (in Berlin):

- Robotics Information day 2013
- Several Exhibitions related to RTC, RLS, RoIS

Minutes of the Robotics Domain Task Force Meeting – DRAFT -
December 10-14, 2012
Burlingame, CA, USA
(robotics/2012-12-23)

Meeting Highlights

- **The Robotics Information Day 2012 was successfully held with 13 talks and 34 participants.**
- **The Robotics Demonstrations attracted lots of OMG participants. ETRI presented HRI demonstration (implementation of OMG RoIS specification), AIST exhibited a small humanoid robot controlled by OpenRTM-aist (implementation of OMG RTC-1.1 specification), Change Vision exhibited a newly released SysML tool, and Honda R&D exhibits the interoperability of two mobile robots controlled by OpenRTM-aist and Honda RTM respectively (implementations of OMG RTC-1.1 specification).**
- **Makoto Sekiya (Honda R&D) and Beom Su Seo (ETRI) were elected as additional Infrastructure WG co-chairs.**
- **We are planning to have the Robotics Information Day 2013 in Berlin collaborated with European Robotics Projects. Antonello Ceravola (Honda-RI-EU) was elected as an Organizing Committee chair.**

List of Generated Documents

robotics/2012-12-01 Final Agenda (Tetsuo Kotoku)
robotics/2012-12-02 Cambridge Meeting Minutes [approved] (Geoffrey Biggs and Seung-woog Jung)
robotics/2012-12-03 Intelligent RT Software Project - Natinal Project in Japan (Tomomasa Sato)
robotics/2012-12-04 Implementation of OPRoS to a human-friendly guide robot, FURO (Se-Kyeong Song)
robotics/2012-12-05 ISO Activity of Service Robot (Seungbin Moon)
robotics/2012-12-06 Disruptive Community Approach to Industrial Robotics Services (Paul Evans)
robotics/2012-12-07 Introduction to Robotic Technology Component (RTC-1.1) Specification (Geoffrey Biggs)
robotics/2012-12-08 An Inprementation of RoIS and RLS Spec. in Japan (Kenji Kamei)
robotics/2012-12-09 Dynamic Deployment and Configuration Standard for Robotic Technology Component: DDC4RTC (Noriaki Ando)
robotics/2012-12-10 Using SysML in a RTC-based Robotics Application : a case study with a demo (Kenji Hiranabe)
robotics/2012-12-11 A New Robotic Technology Middleware and Robotic Technology Component Interoperability demonstration (Makoto Sekiya)
robotics/2012-12-12 Implementation of RoIS to robots in ETRI (Su-Young Chi)
robotics/2012-12-13 Component Management in OPRoS (Seung-woog Jung)
robotics/2012-12-14 Cloud Networked Robotics and Acceleration Based Sensing (Miwako Doi)
robotics/2012-12-15 Introduction to OpenEL (Enbedded Library) for Robot (Kenichi Nakamura)
robotics/2012-12-16 OpenEL API specification ver.0.1.1 (Kenichi Nakamura)
robotics/2012-12-17 Opening Presentation (Tetsuo Kotoku)
robotics/2012-12-18 Infrastructure WG Progress Report (Noriaki Ando)
robotics/2012-12-19 Wrap-up Presentation (Tetsuo Kotoku)
robotics/2012-12-20 Roadmap for Robotics Activities (Tetsuo Kotoku)
robotics/2012-12-21 Next Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)
robotics/2012-12-22 DTC Report Presentation (Tetsuo Kotoku)
robotics/2012-12-23 Burlingame Meeting Minutes - DRAFT (Seung-woog Jung and Takashi Suehiro)

Minutes

Wednesday 12 December, 2012, Sandpebble E, 1st FL
Robotics DTF Plenary Meeting Chair: Tetsuo Kotoku (AIST)
AIST, Change Vision, ETRI, Honda, JARA, UEC (Quorum: 3)

16:00 - 16:10 Robotics-DTF Opening Session, Tetsuo Kotoku (AIST)
- Minutes takers: Seung-Woog Jung (ETRI) and Takashi Suehiro(UEC)
- A Brief Summary of Cambridge meeting
  - 10 participants
  - 2 Contact Reports
  - 2 WG reports
- Cambridge Meeting minutes (robotics/2012-12-02) was approved.
  : AIST (motion), ETRI (second), UEC (white ballot)

16:10 - 16:45 WG Activity Reports
- **Infrastructure WG, Noriaki Ando (AIST)**
  . New work item meeting on Monday
  . New work items
    * Data port and data type
    * FSM component
  . 2 possible standardization processes for the new work items
    * RTC 2.0 on MARS
    * New Spec on Robotics DTF or MRAS
  . Possible standardization schedule of the new work items (Best Scenario)
    * submit RFP draft 4 weeks before of the next Reston meeting
    * 1st review in Reston (March 2013) meeting
    * 2nd review and AB review in Berlin (June 2013) meeting
    * Initial submission in Dec. 2013 meeting
    * Starting FTF in June 2014 meeting
    * FTF report in Jun 2015 meeting
    * Specification might be published at the end of 2015

16:45-17:00 **Robotics-DTF Plenary Wrap-up Session, Tetsuo Kotoku (AIST)**
Robotics-DTF Co-Chair (call for volunteer): postpone voting one more meeting
New Organization was approved
  : AIST(motion), ETRI(second), UEC(white ballot)
  * Infrastructure WG:
    Seung-Woog Jung is difficult to attend upcoming meetings.
    New Co-Chairs: BeomSu Seo (ETR) and Makoto Sekiya(Honda)
  * Organizing Committee for Robotics Information Day in Berlin
    New Chair: Antnello Ceravola (Honda-RI-EU)
Next meeting schedule

Plenary meeting attendee (17 attendees):

- Antonello Ceravola (Honda-RI-EU)
- Beom-Su Seo (ETRI)
- Byung-Tae Chun (Hanyang Univ)
- Geoffrey Biggs (AIST)

- Kenichi Nakamura (JASA)
- Kenji Hiranabe (ChangeVision)
- Koji Kamei (JARA/ATR)
- Makoto Sekiya (Honda)
- Noriaki Ando (AIST)
- Seung-Woog Jung (ETRI)
- Su-Young Chi (ETRI)
- Takashi Suehiro (UEC)
- Tetsuo Kotoku (AIST)
- Toshihiro Okamura (ChangeVision)
- Toshiki Iwanaga (ChangeVision)
- Toyotaka Torii (Honda)
- Young-Jo Cho (ETRI)

Robotics Information Day 2012 attendee (34 attendees):

- Antonello Corevola (Honda-RI-EU)
- Beom-Su Seo (ETRI)
- Byung-Tae Chun (Hankyoung Univ.)
- Chuck Zublic (NGC)
- Daniel Siegl (LieberLiever)
- Geoffrey Biggs (AIST)
- Gerardo Pardo-Castellote (RTI)
- Hajime Ueno (Fuji Xerox)
- Hugues Vincent (Thales)
- Isao Hara (AIST)
- Isashi Uchida (IPA)
- Julien Deantoni (INRIA)
- Kenichi Nakamura(JASA)
- Kenji Hiranabe (ChangeVision)
- Koji Kamei (ATR)
- Makoto Sekiya (Honda)
- Miwako Doi (Toshiba)
- Noriaki Ando (AIST)
- Paul Evans (SwRI)
- Russell Peak (Georgia Tech)
- Seiichi Shin (UEC)
- Se-Kyung Song (Future Robot)
- Seungbin Moon (Sejong Univ.)
- Seung-Woog Jung (ETRI)
- Su-Young Chi (ETRI)
- Takashi Suehiro (UEC)
- Takashi Tsubouchi (Univ. of Tsukuba)
- Tetsuo Kotoku (AIST)
- Tomomasa Sato (Univ. of Tokyo)
- Toshihiro Okamura (ChangeVison)
- Toshiki Iwanaga (ChangeVision)
- Toyotaka Torii (Honda)
- Young-Jo Cho (ETRI)
- Yutaka Matsuno (Nagoya Univ.)

Prepared and submitted by Seung-Woog Jung (ETRI) and Takashi Suehiro (UEC).