Robotics Domain Task Force Final Agenda ver.1.0.2 robotics/2011-06-01

OMG Technical Meeting - Salt Lake City, UT, USA -- June 20-24, 2011

http://robotics.omg.org/

		TF/SIG				
		Host	Joint (Invited	Agenda Item	Purpose	Room
Sunday: WG activites(pm)						
13:00	17:00			Robotics DDC4RTC and RoIS submitters meeting	Arrangement	snowbasin, 2nd FL Tower
Monday: Robotics-DTF Plenary(am) and WG activity(pm)						
10:45	11:00	Robotics		Robotics-DTF Plenary Opening Session	presenta ion and discussion	Sun Valley, 2nd FL Twr
11:00	12:00	Robotics		**Revised Submission for RoIS RFP Review, Vote-to-Vote, and Voting** - Su-Young Chi (ETRI), Miki Sato (JARA/ATR) and Toshio Hori (AIST)	Robo ics plenary closing	
12:00	13:00			**LUNCH**		Ballroom B, Main Lvl
13:00	18:00			Architecture Board Plenary		Arizona, Main Lvl
13:00	18:00			**DDC4RTC** (Robotic Infrastructure) WG(5h) - Noriaki Ando (AIST) and Seung-Woog Jung (ETRI)	discussion	Wasatch, 2nd FL Twr
				RoIS (Robotic Functional Services) WG(5h): - Su-Young Chi (ETRI), Miki Sato (JARA/ATR) and Toshio Hori (AIST)	discussion	Tucson, 2nd FL Twr
Tuesday: WG activity(am) and Robotics-DTF Plenary(pm)						
9:00	12:00			**DDC4RTC** (Robotic Infrastructure) WG(3h) - Noriaki Ando (AIST) and Seung-Woog Jung (ETRI)	discussion	Teton, 2nd FL Twr
				RoIS (Robotic Functional Services) WG(3h): - Su-Young Chi (ETRI), Miki Sato (JARA/ATR) and Toshio Hori (AIST)	discussion	Teton, 2nd FL Twr
12:00	13:00			**LUNCH**		Ballroom C, Main Lvl
13:00	14:00	Robotics		**Special Talk: OPRoS: Open Platform for Robotic Services** - Hong Seong Park (Kangwan Na ional Univ.)	presenta ion and discussion	Sun Valley, 2nd FL Twr
14:00	14:30	Robotics		Conformance Testing Method for Robo ic Software Components - Mi-Sook Kim (Kangwon National Univ.)	presenta ion and discussion	
				Break (30min)		
15:00	15:30	Robotics		Robotics Technology Applied to Great East Japan Ear hquake - Miwako Doi (Toshiba)	presenta ion and discussion	
15:30	16:10	Robotics		WG Reports and Discussion (Service WG, Infrastructure WG, Models in Robotics WG)	presenta ion and discussion	
16:10	16:30	Robotics		Contact Reports: - Makoto Mizukawa(Shibaura-IT), and Young-Jo Cho(ETRI)	Information Exchange	
16:30	17:00	Robotics		Robotics-DTF Plenary Wrap-up Session (**DTF Co-Chair Election**, Roadmap and Next mee ing Agenda)	Robo ics plenary closing	
17:00				Adjourn joint plenary meeting		
17:00	17:30			Robotics WG Co-chairs Planning Session (Preliminary Agenda for next TM, Draft report for Friday)	planning for next meeting	
Wednesday: WG activitiy						
9:00	12:00	SysA	Robotics	**Seminar on Systems Assurance & Safety for Consumer Devices: Automotive, Robotic & Building Automation Systems of the Future**		Ballroom A, Main Lvl
12:00	14:00			**LUNCH and OMG Plenary**		Ballroom C, Main Lvl
14:00	17:00	SysA	Robotics	**Seminar on Systems Assurance & Safety for Consumer Devices: Automotive, Robotic & Building Automation Systems of the Future**		Ballroom A, Main Lvl
18:00	20:00			**OMG Reception**		Foyer ABC, Main Lvl
Thursday: WG activitiy						
9:00	12:00	Robotics		Robotics WG activity follow-up (reserved for RoIS RFP Re-Review and Vo ing)	discussion	Wasatch, 2nd FL Twr
12:00	13:00			**LUNCH**		Ballroom B, Main Lvl
13:00	18:00			Architecture Board Plenary		Arizona, Main Lvl
13:00	18:00			Robotics WG activity follow-up	discussion	Wasatch, 2nd FL Twr
Friday						
8:30	12:00			AB, DTC, PTC		Arizona, Main Lvl
12:00	13:00			**LUNCH**		Idaho, Main Lvl
Other Meetings of Interest						
Monday						
8:00	8:45	OMG		New Attendee Orientation		Teton, 2nd FL Twr
9:00	12:00	OMG		Tutorial - Introduc ion to OMG's mee ing and Middlewere Specifications		Teton, 2nd FL Twr
Tuesday						
7:30	9:00	OMG		Liaison ABSC		Tucson, 2nd FL Twr
9:00	17:00	SysA		System Assurance PTF		Ballroom B, Main Lvl
9:00	17:30	OMG		Cloud Standards Customer Council Mee ing		Olympus, 2nd FL Lvl
17:00	18:00	OMG		RTF-FTF Chair's Workshop		snowbasin, 2nd FL Twr
Thursday						
9:00	17:00	SysA		System Assurance PTF		Tucson, 2nd FL Twr

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

Minutes of the Robotics DTF Meeting
March 21-25, 2011
Arlington, VA, USA
(robotics/2011-06-02)

Meeting Highlights

- Robotics Infrastructure WG meetings for integrating two initial submissions of the Dynamic Deployment and Configuration for Robotic Technology
- Robotics RoIS meetings for revision of submission of the Robotic Interaction Service (RoIS) Framework.
- Two special talks
  - PROTEUS Project, Laurent Rioux (Thales)
  - Conformance Testing Method for Robot Software Components, Mi-Sook Kim (Kangwon National University, KOREA)

List of Generated Documents

robotics/2011-03-01 Final Agenda (Tetsuo Kotoku)
robotics/2011-03-02 Santa Clara Meeting Minutes [approved] (Miki Sato and Myung-Eun Kim)
robotics/2011-03-03 Review of ETRI Initial Submission (Seung-Woog Jung)
robotics/2011-03-04 How to apply policy concept for integration (Seung-Woog Jung)
robotics/2011-03-05 Discussion issues for supervision tree (Myung-Eun Kim)
robotics/2011-03-06 How to apply the repository and directory to the supervision tree? (Myung-Eun Kim)
robotics/2011-03-07 Robotic Functional Service WG - RoIS Framework - (Miki Sato)
robotics/2011-03-08 Proteus - French National Initiative - (Laurent Rioux)
robotics/2011-03-09 Conformance Testing Method for Robot Software Components (Mi-Sook Kim)
robotics/2011-03-10 Opening Presentation (Tetsuo Kotoku)
robotics/2011-03-11 Robotic Functional Services WG Report (Toshio Hori)
robotics/2011-03-12 Robotic Infrastructure WG Report (Noriaki Ando)
robotics/2011-03-13 Roadmap for Robotics Activities (Tetsuo Kotoku)
robotics/2011-03-14 Wrap-up Presentation (Tetsuo Kotoku)
robotics/2011-03-15 The number of downloads (Tetsuo Kotoku)
robotics/2011-03-16 Next Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)
robotics/2011-03-17 DTC Report Presentation (Young-Jo Cho)
robotics/2011-03-18 Washington DC Meeting Minutes - DRAFT (Geoffrey Biggs and Seung-woog Jung)

Minutes

Thursday, March 24, 2011, Prince William, 3rd FL

- 13:00 - 14:00 Special talk : PROTEUS Project, Laurent Rioux (Thales)
  - Nationwide project for robot development in France
  - Incorporating several challenges of varying complexity
- 14:00 - 14:20 Special talk : Conformance Testing Method for Robot Software Components, Mi-Sook Kim (Seoul / Kangwon National University)
  -Tools for testing components
  - Tools for verifying components comply with their specification and interfaces
- 13:00-16:00 Robotics DTF Plenary Meeting, Chair: Dr Kotoku
  - Participant organizations : AIST, ETRI, JARA, Technologic Arts
  - Minutes takers: Geoffrey Biggs (AIST) and Seung-Woog Jung (ETRI)
  - Santa Clara meeting minutes approved
    (Motion: AIST, Second: ETRI, White valet: Technologic Arts)

  - 15:05 - 15:15 Functional Service WG Report, Toshi Hori (AIST)
    . Discussion on RoIS framework draft proposal.
    . Parameters for each basic HRI component
    . Definition of "Procedure"
    . Discussion on data structures for IDs
    . Differences between "navigation" and "move" need further discussions
    . Deadline for revised submission is late May
  - 15:15 - 15:30 Infrastructure WG Report, Noriaki Ando (AIST)
    . Need to further understand the DEPL specification, so spent time reviewing it
    . Discussed scenarios for DDC4RTC
    . Have postponed the merged submission deadline until the December meeting.
  - 15:30 - 15:40 Wrap-up session
    . Call for volunteers for Robotics DTF co-chair
    . No volunteers postponed one meeting.
    . Changes in organization
    . New Robotic Technology Component 1.1 WG
    . Schedule for next meeting
    . Must have vote-to-vote for RoIS on Monday morning
    . Plenary meeting on Tuesday afternoon

ATTENDEE (17 participants)

- Andrey Sadovykh (Softeam)
- Amaud Cuccuru (CEA LIST)
- Geoffrey Biggs (AIST)
- Joeng-Sook Kang (KNU)
- Laurent Rioux (THALES)
- Miki Sato (ATR)
- Mi-Sook Kim (KNU)
- Myung-Eun Kim (ETRI)
- Noriaki Ando (AIST)
- Sebastien GERARD (CEA LIST)
- Seung-Woog Jung (ETRI)
- Su-Young Chi (ETRI)
- Takeshi Sakamoto (Technologic Arts)
- Tetsuo Kotoku (AIST)
- Toshio Hori (AIST)
- Young-Jo Cho (ETRI)
- Yves BEKNARD (Airbus)

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

Revised Submission to Robotic Interaction Service (RoIS) Framework

Toshio Hori (JARA/AIST), Miki Sato (JARA/ATR), Su-Young Chi (ETRI)
Jun. 20, 2011

Problem Statement & Scope of RoIS Framework RFP

- Application program for a specific service robot does not run on other robots because:
  - Different robots may be equipped with a variety of types of sensors and actuators.
  - APIs of robots from different vendors vary even if they use the same sensor types.
- This lack of application portability is an obstacle to the success of the service robot industry.
- Establishing a specification of **Robotic Interaction Service (RoIS) Framework** that provides
  - A set of common interfaces between service application and robot control program components,
  - A set of common messages exchanged through the common interfaces.

History: RFP issued & Proposals submitted

- **2009/03 (Washington D.C. meeting)**
  - The Robotic Functional Service WG agreed to make a standard for service robots.
- **2010/06 (Minneapolis meeting)**
  - RoIS Framework RFP passed AB and DTC.
  - The RFP was issued by OMG.
- **2010/12 (Santa Clara meeting)**
  - JARA (robotics/10-10-01) and ETRI (robotics/10-11-01) submitted proposals.
  - Two proposals were reviewed at the Robotic Functional Service WG and the WG agreed to merge them based on JARA's proposal.
- **2011/05 (4 weeks before Salt Lake City meeting)**
  - **Revised submission** was posted to the OMG server.

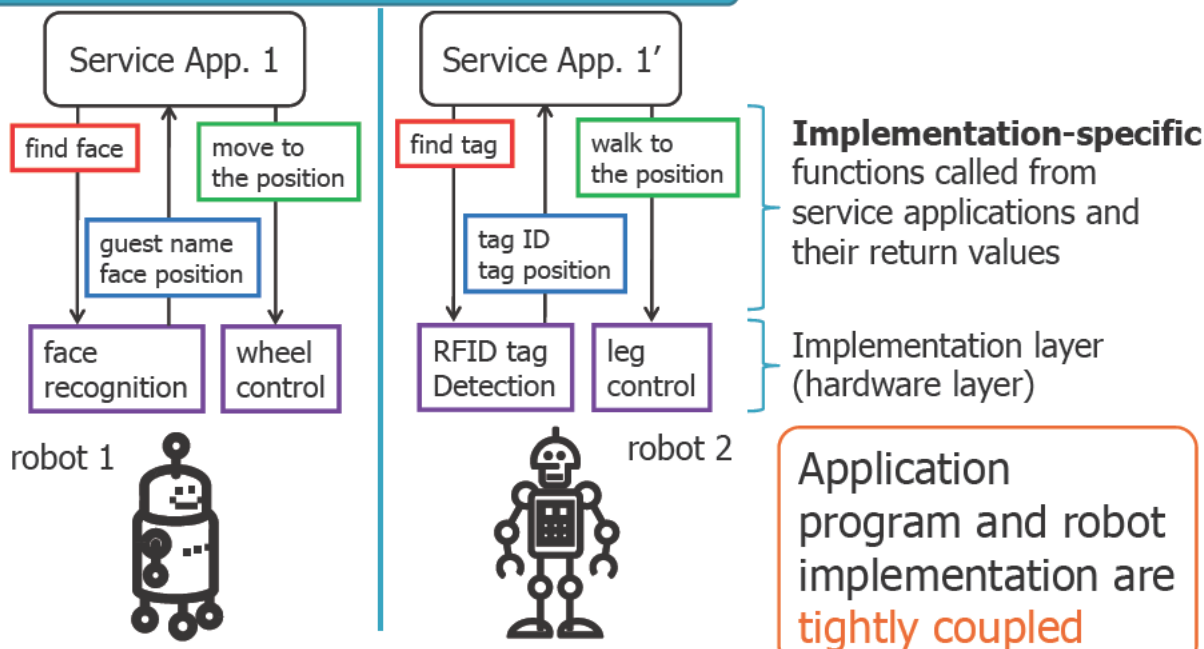
robotics/2011-06-06

Highlights of the Revised Submission

- Clarified the definitions of "HRI Component" and "HRI Engine"
- Fixed Basic HRI Components and their parameters
  - 15 basic components and the common methods for all HRI components
- Defined a data type for ID with its reference
- Revised Command interface to deal with "Procedure"
  - Procedure is a predefined set of HRI Components which shall be executed in parallel or sequentially
- Revised PSM to conform to the submission

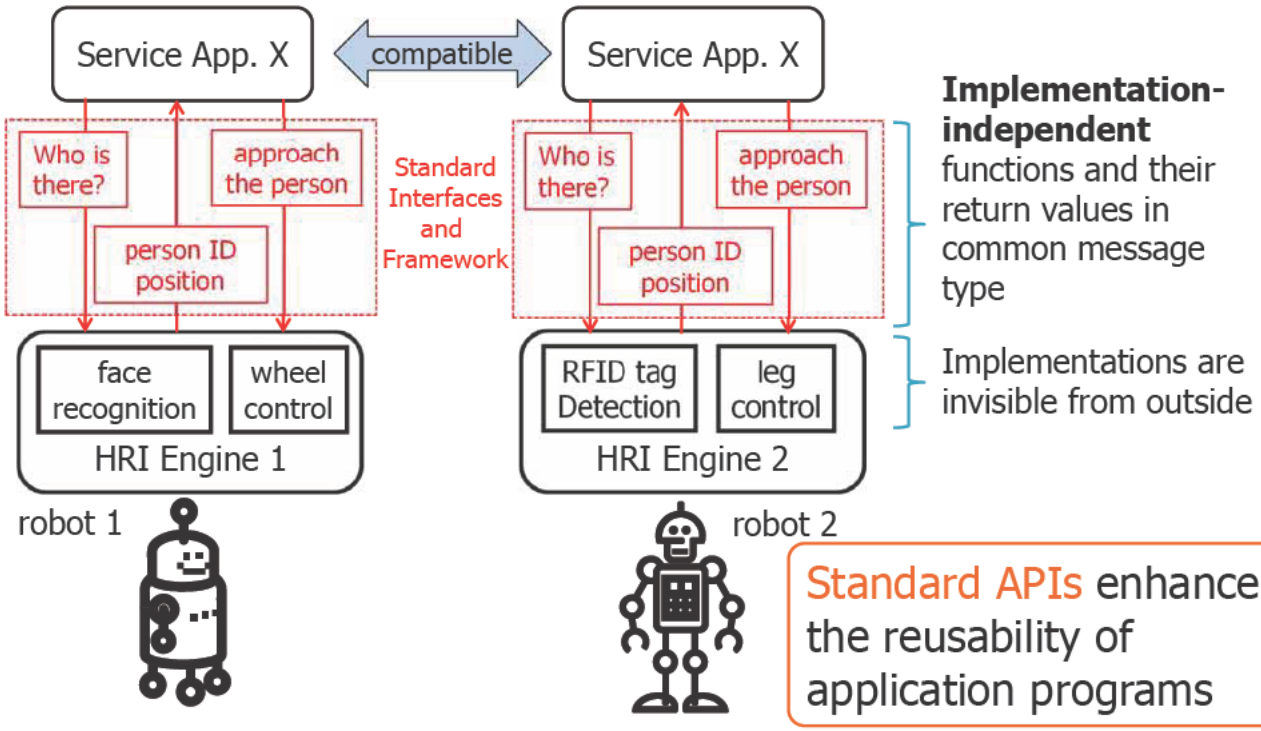
Conventional Style of Service Application Programming

Example: Find a person and approach the person



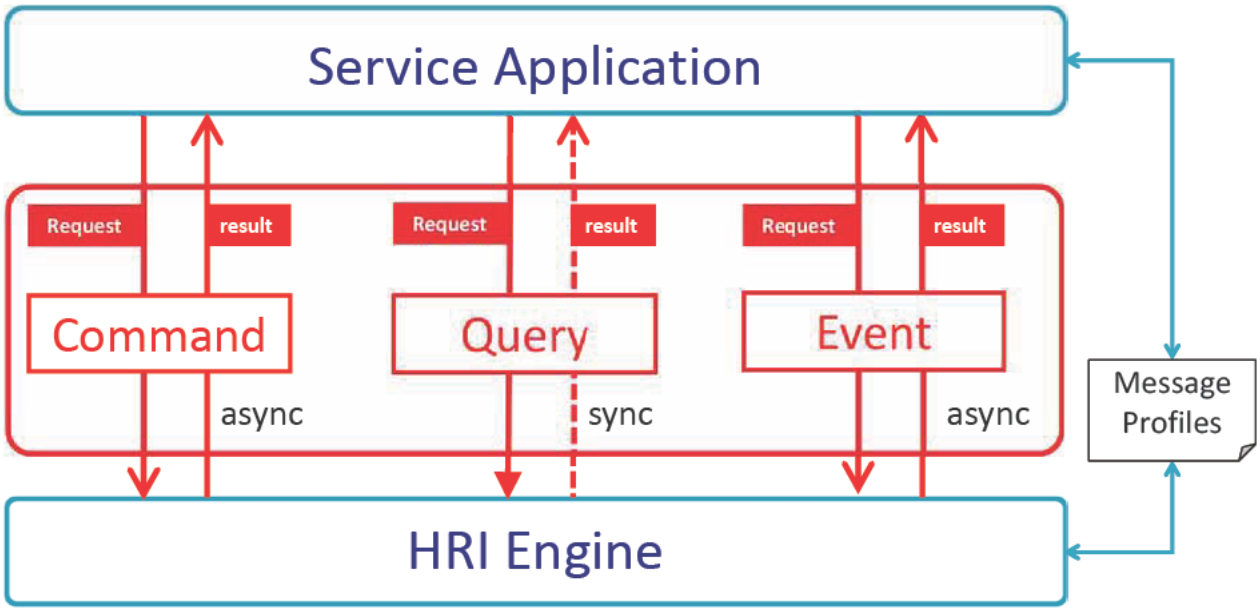
Application program and robot implementation are tightly coupled

Proposed Service Application Programming Style

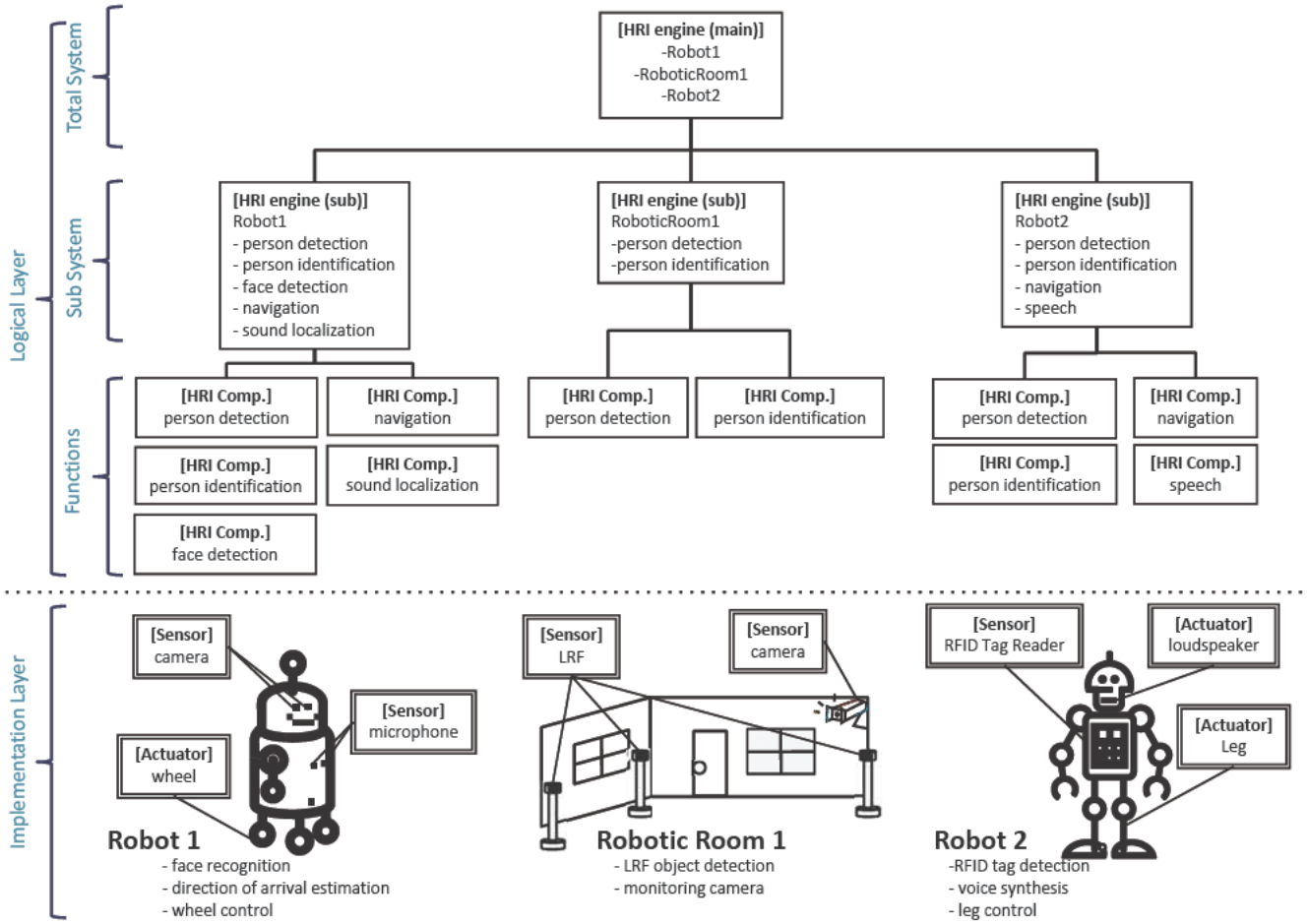


Standard APIs enhance the reusability of application programs

Concept of RoIS Framework



AIST ATR ETRI



Definitions of HRI Component and HRI Engine

- **HRI Component**
  - HRI Components are **functional units** making up the description of HRI scenario.
  - HRI Component provides **hardware-independent APIs**.
  - Only symbolic data is exchanged between HRI Components and service applications.
  - Symbolic data is used in the service applications without special handling such as pattern recognition, signal processing and human judgment.
- **HRI Engine**
  - HRI Engine is an entity which encapsulates HRI Components and works as a logical implementation of a robot system.
  - It may include several HRI Engines. In this case, internal HRI Engines are called sub HRI Engines.

robotics/2011-06-06

Interface types

- RoIS Framework is composed of **4 interface types**:
  - **System Interface**: Manages connection status between a service application and an HRI Engine
  - **Command Interface**: Enables a service application to send commands to HRI Components
  - **Query Interface**: Enables a service application to query the HRI Engine on information it holds.
  - **Event Interface**: Enables a service application to receive notifications on changes in the HRI Engine status.

Revised Submission to RoIS Framework RFP 2011/06/20
AIST ATR ETRI

Definition of Basic HRI Component

- RoIS Framework specifies **Basic HRI Components** that are commonly used for human-robot interaction.
  - Each Basic HRI Component shall be a **functional unit** that is developed with mature technologies from the viewpoint of the usage.
  - Methods for each Basic HRI Component shall be simple as possible.
  - Mandatory parameters for the operation of each component shall be minimized.

- **Notes:**
  - HRI Components other than Basic HRI Components are called "**User-defined HRI Components**."

List of Basic HRI Components (for sensing)

Component Name	Description
system information	Provides the information of the system such as the system status and position of the physical unit.
person detection	Detects number of people.
person localization	Detects position(s) of people.
person identification	Identifies ID(s), such as name(s), of people.
face detection	Detects number of human faces.
face localization	Detects position of human faces.
sound detection	Detects number of sound sources.
sound localization	Detects position/direction of sound source(s).
speech recognition	Recognizes person's speech.
gesture recognition	Recognizes person's gesture(s).

List of Basic HRI Components (for action)

Component Name	Description
speech synthesis	Generates robot speech.
reaction	Performs specified reaction.
navigation	Moves to specified target location.
follow	Follows specified target object.
move	Moves relatively, such as back and forth or in a curve.

- **Notes:**
  - A data type "**RoIS_Identifier**" is defined to describe ID(s) in the Framework.
  - A data type "**RLS Data**" [RLS] is used to describe position data in the Framework.

Common Message for All HRI Components

- All HRI Components shall incorporates the following messages and parameters in common.
  - **for Command Interface:**
    - **start**
    - **stop**
    - **suspend**
    - **resume**
  - **for Query Interface:**
    - **component_status**

    (UNINITIALIZED, READY, BUSY, WARNING, ERROR)

RoIS Profiles

- Configuration of the HRI Engine and HRI Components are defined using 4 types of profiles:
  - **HRI Engine Profile:** defines engine name, a list of HRI Components and parameters of the HRI Engine.
  - **HRI Component Profile:** defines component name, a list of messages and parameters of the HRI Component.
  - **Message Profile:** defines message name and required arguments and results of the command, query and event messages.
  - **Parameter Profile:** defines data name, data type and default value of parameters. (i.e. parameters of HRI Engine/HRI Component, arguments/results of messages)

robotics/2011-06-06

Related Standards

- **Normative**
  - [CORBA] Object Management Group, Common Object Request Broker Architecture (CORBA), Version 3.1, 2008
  - [DDS] Object Management Group, Data Distribution Services (DDS), Version 1.2, 2007
  - [ISO639] International Organization for Standardization, Codes for the representation of names of languages
  - [ISO19111] International Organization for Standardization, Geographic information - Spatial referencing by coordinates, 2007
  - [ISO19115] International Organization for Standardization, Geographic information - Metadata, 2003
  - [ISO19784] International Organization for Standardization, Biometric Application Programming Interface (BioAPI), Version 2.0, 2006.
  - [RLS] Object Management Group, Robotic Localization Service (RLS), Version 1.0, 2010
  - [RTC] Object Management Group, Robotic Technology Component (RTC), Version 1.0, 2008
  - [W3C-SRGS] W3C, Speech Recognition Grammar Specification Version 1.0, 2004
  - [W3C-SSML] W3C, Speech Synthesis Markup Language (SSML) Version 1.0, 2004
- **Non-normative**
  - [W3C-DT] World Wide Web Consortium, Date and Time Formats, http://www.w3.org/TR/NOTE-datetime, 1998

Revised Submission to RoIS Framework RFP 2011/06/20
AIST ATR ETRI

Resolution of RFP Mandatory Requirements (1/4)

1. Proposals shall provide a diagram or description giving an overview of the architecture, including the RoIS framework, robotic service application and HRI Engine (composed of robotic components).
   - **Section 7.2 "Structure of the RoIS Framework"**

2. Proposals shall provide a specification of the following interfaces between robotic service applications and the HRI Engine:
   - Interface to obtain information actively from HRI Engine
   - Interface to receive notification of real-time events from the HRI Engine
   - Control interface to send commands to HRI Engine functions
     - **Section 7.4.1.3 "Query Interface,"**
     - **Section 7.4.1.4 "Event Interface,"**
     - **Section 7.4.1.2 "Command Interface," respectively**

Note: All section numbers correspond to those in the Revised Submission ([updated] robotics/11-06-03).

robotics/2011-06-06

Resolution of RFP Mandatory Requirements (2/4)

3. Proposals shall specify data structure for each interface
   - **Section 7.4.3 "Message Data"**

4. Proposals shall provide a specification of the following common messages for all HRI Engines:
   - Event notification
     - person detected, person identified, command action completed
   - Query (message types are omitted)
     - person ID, position of person/robot, status of HRI Engine
   - Command (message types are omitted)
     - start/stop person detection/identification
     - **Section 7.6 "Common Message"**

Revised Submission to RoIS Framework RFP 2011/06/20

AIST ATR ETRI

Resolution of RFP Mandatory Requirements (3/4)

5. The specification shall meet the following criteria:
   - Be general enough to incorporate robotic components for various sensors, actuators and algorithms in HRI Engine
   - Satisfy interoperability and reusability, to allow an HRI Engine to be replaced with another without difficulty
     - All messages in the submission are defined independently of any specific sensors, actuators and algorithms.
     - Common messages defined in the submission include not only messages presented in the Mandatory Requirements but also messages that are common among several service robot systems which we surveyed.
     - Message types proposed in the submission are extensible by message profiles, whose structure is described in **Section 7.5 "Profiles."**
     - HRI engine is treated at abstraction level that is independent of implementation. HRI engine will be replaced without difficulty.

Resolution of RFP Mandatory Requirements (4/4)

6. Proposals shall specify existing technologies to achieve functions required for RoIS framework.
   - **Section 4 "References"**

7. Proposal shall specify functions that cannot be achieved by existing technologies.
   - This submission stands out by specializing in physical and spatial characteristics in the real world, which has not been done before. For example, structure of HRI engine and HRI component are determined in a form that makes use of physical and spatial characteristics as described in **Section 7.2 "Structure of the RoIS Framework."**

8. Proposals shall provide a Platform Independent Model (PIM).
   - **Section 7 "Platform Independent Model"**

Resolution of RFP Optional Requirements (1/2)

1. Proposals may specify schemes for the functionalities listed below.
   - Error handling for each interface type
     - Error handling for each method of the interfaces
       - **"ReturnCode_t" in Section 7.3 "Return Codes" (See 7.4.1.2-4)**
     - Error handling for the system
       - **"receive_error" and "get_error_detail" in Section 7.4.1.1 "System Interface"**
   - Returning command results and status
     - **"completed" and "get_command_result" in Section 7.4.1.2 "Command Interface"**

2. Proposals may provide a schema to describe message profiles.
   - **Section 7.5 "Profiles"**
   - **Section 9 "Examples of Profile in XML"** provides a possible description of concrete message profiles.

robotics/2011-06-06

Resolution of RFP Optional Requirements (2/2)

3. Proposals may provide a Platform Specific Model (PSM) as C++, CORBA-specific model, XML or RTC for RoIS framework.
   - C++ PSM is described in **Section 8.1**
   - CORBA PSM is described in **Section 8.2**
   - XML PSM is described in **Section 8.3**

Revised Submission to RoIS Framework RFP 2011/06/20
AIST ATR ETRI

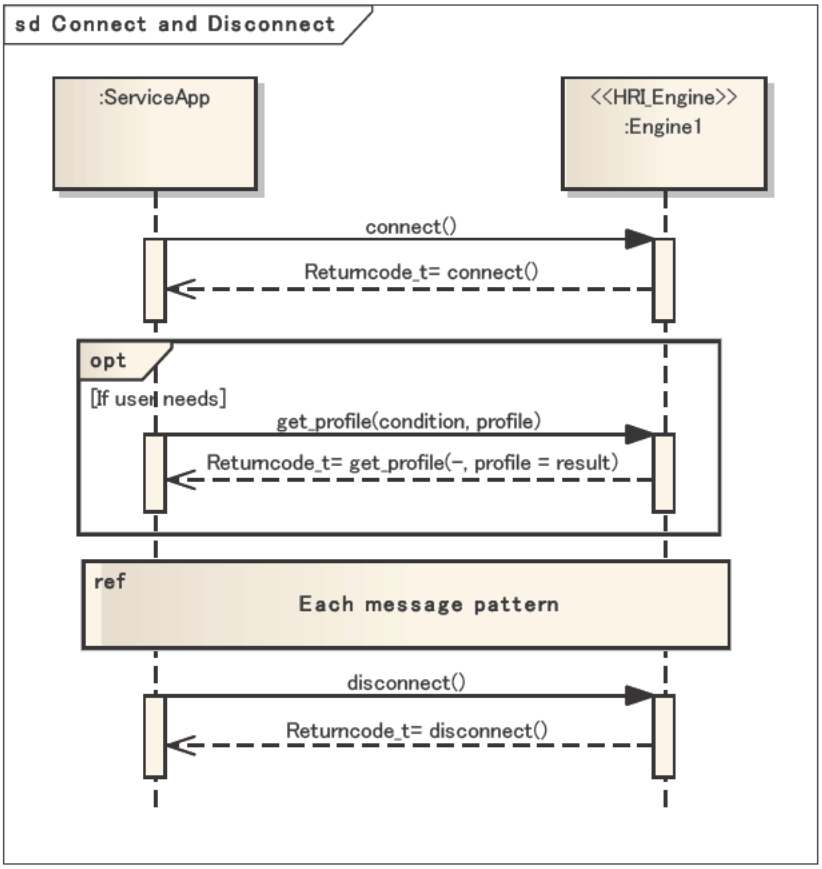
AB Comment to the Revised Submission (from Andrew Watson)

- No technical comments
- One procedural comment on the copyright:
  - ATR and AIST in the copyright list are not submitters so OMG needs to know that the submitters (ETRI & JRA) either own, or have transferable rights to, all copyrighted parts of the submission, and have licensed OMG to publish the whole submission, including any parts licensed from others.
- Solution:
  - We remove ATR and AIST from the copyright list.

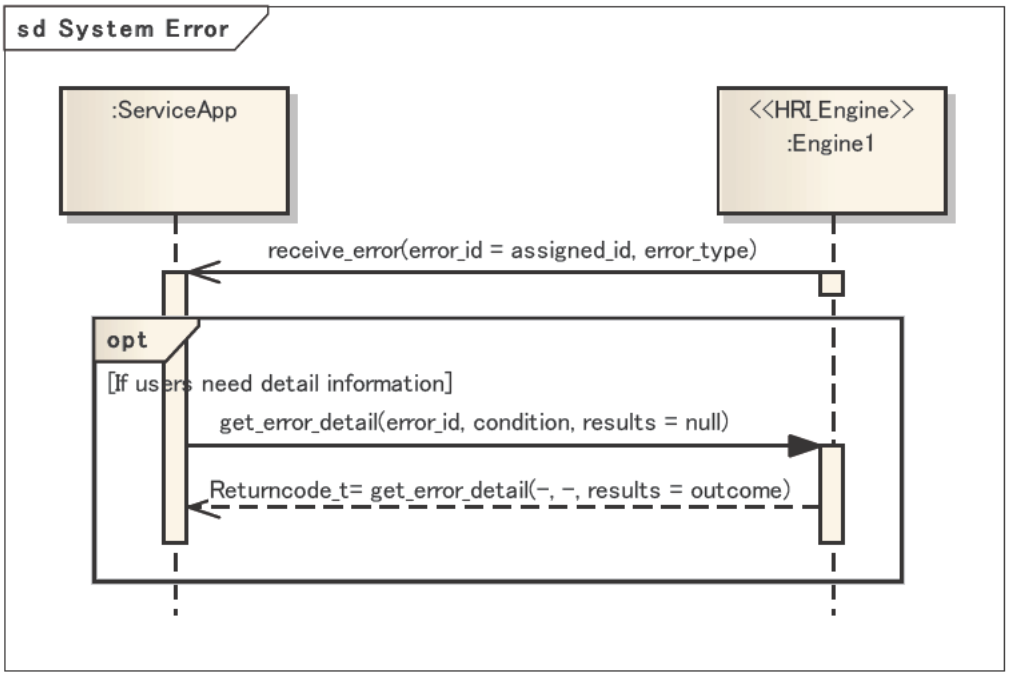
Revised Submission to RoIS Framework RFP 2011/06/20

Sequence Diagrams

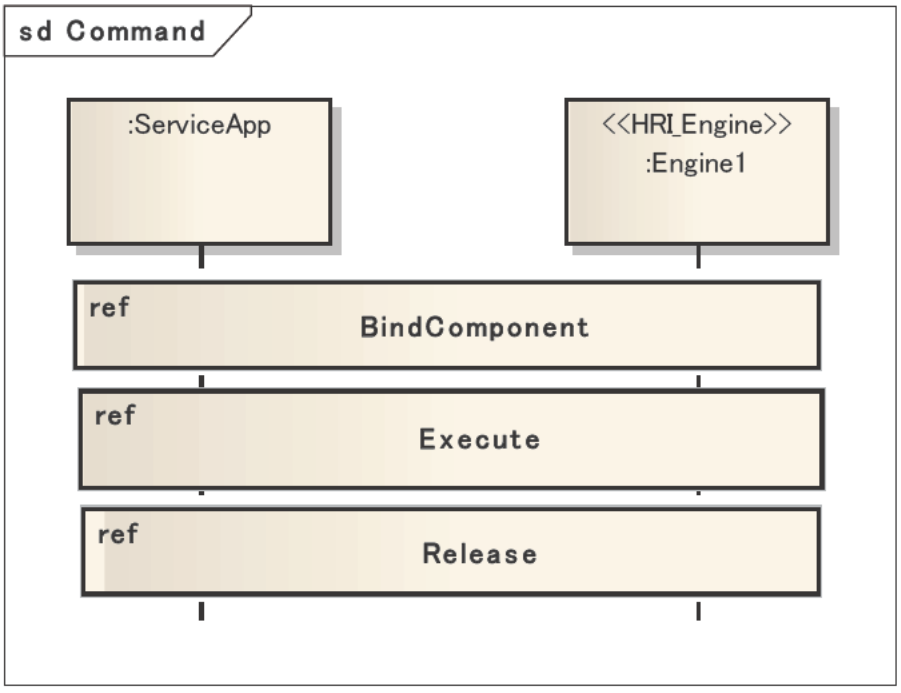
System Interface (connect / disconnect)

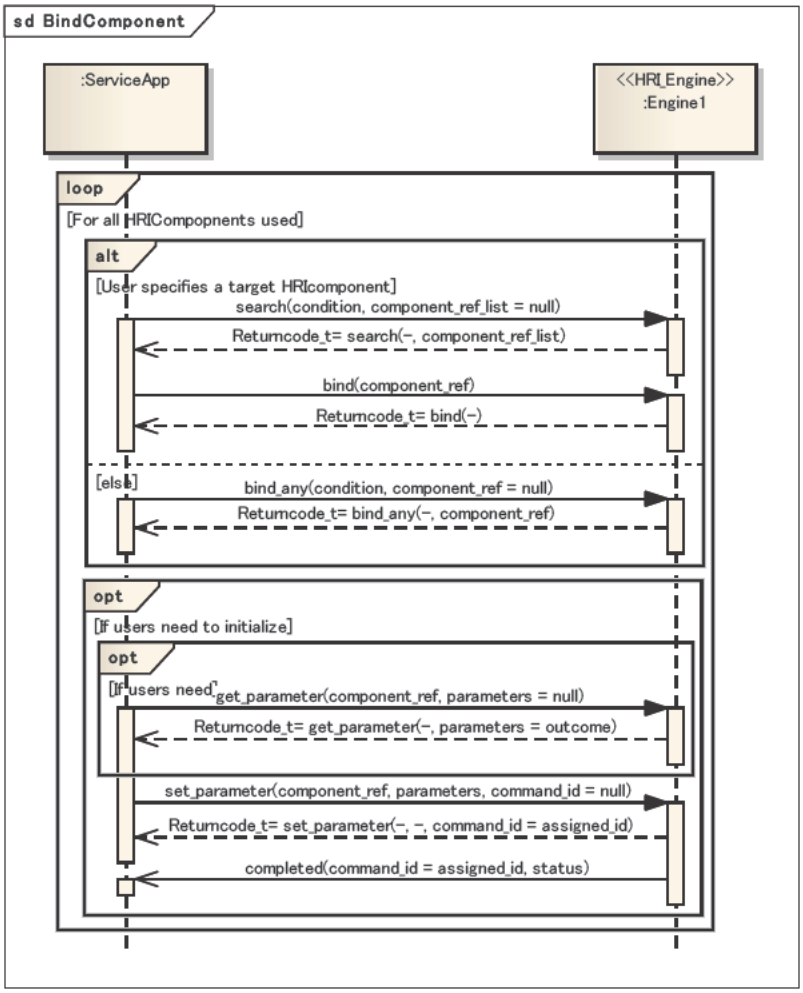


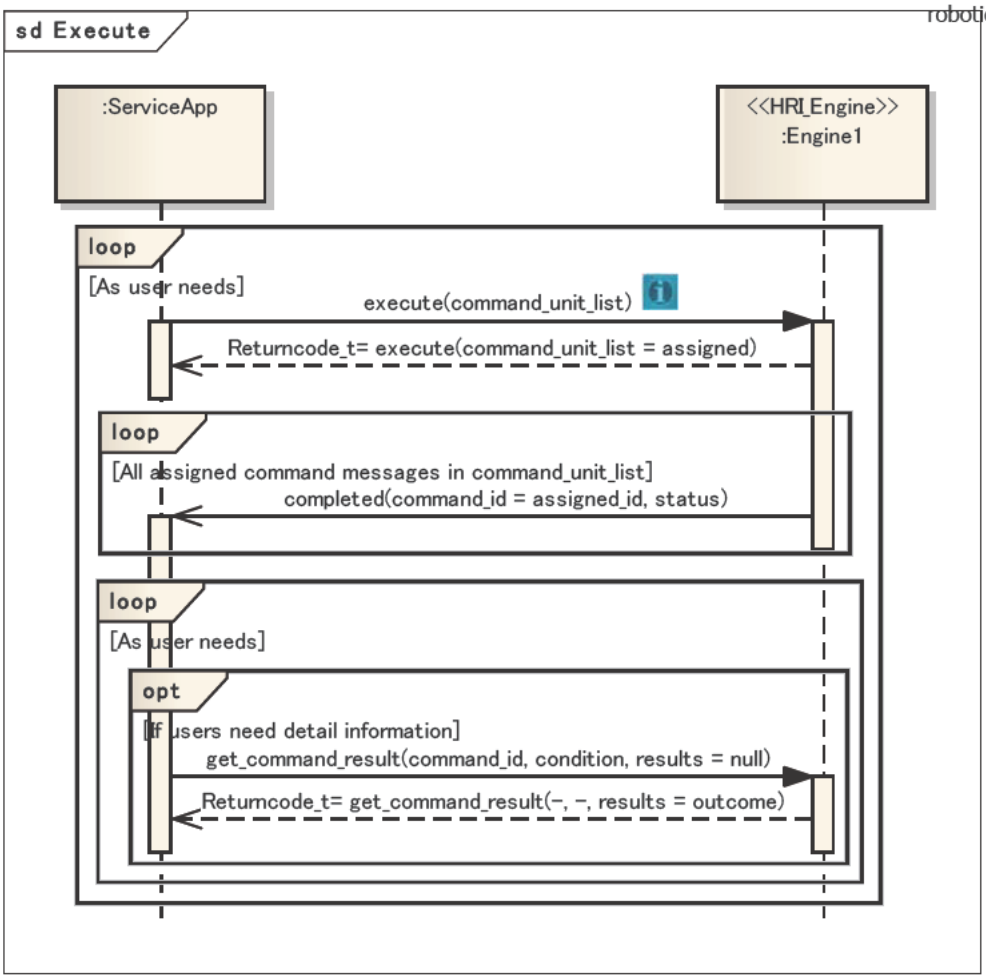
System Interface (error notification)

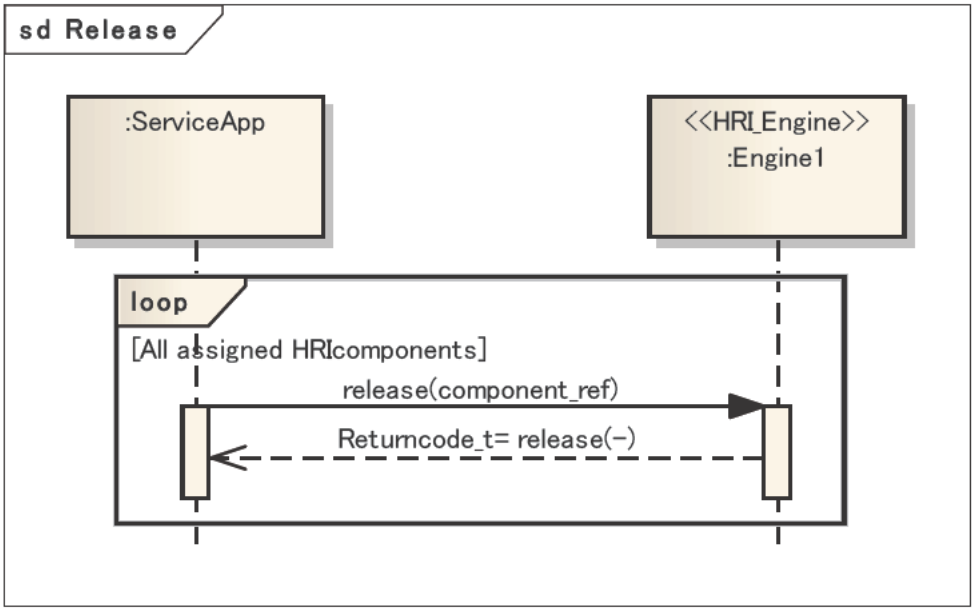


Command Interface

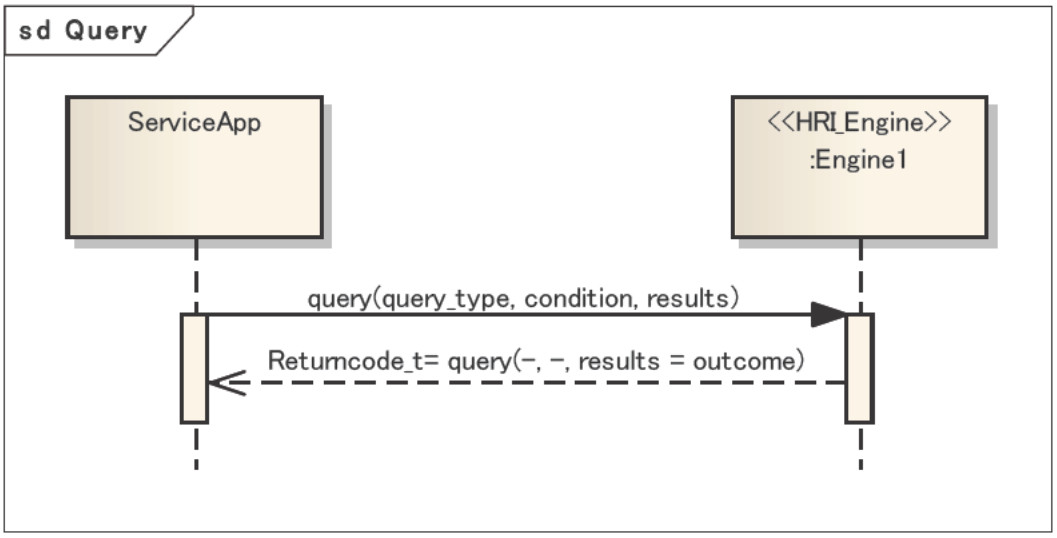



sd BindComponent
:ServiceApp
<<HRI_Engine>>
:Engine1
loop
[For all HRIComponents used]
alt
[User specifies a target HRIcomponent]
search(condition, component_ref_list = null)
Returncode_t= search(–, component_ref_list)
bind(component_ref)
Returncode_t= bind(–)
[else]
bind_any(condition, component_ref = null)
Returncode_t= bind_any(–, component_ref)
opt
[If users need to initialize]
opt
[If users need]
get_parameter(component_ref, parameters = null)
Returncode_t= get_parameter(–, parameters = outcome)
set_parameter(component_ref, parameters, command_id = null)
Returncode_t= set_parameter(–, –, command_id = assigned_id)
completed(command_id = assigned_id, status)

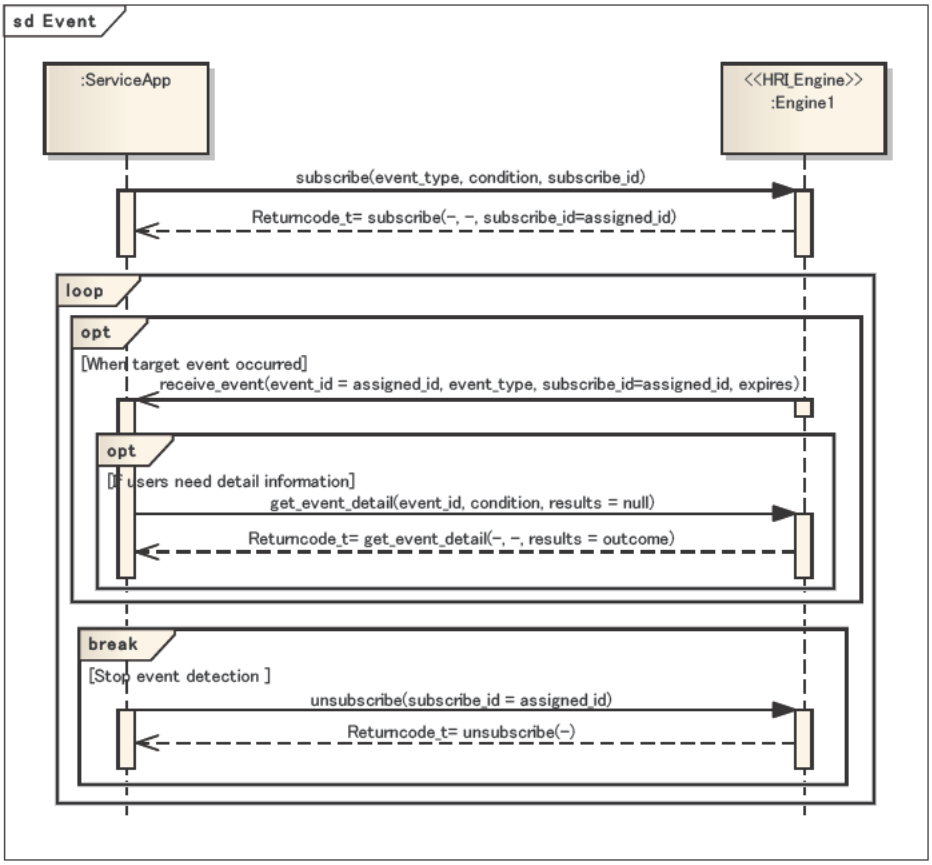

sd Execute
:ServiceApp
<<HRI_Engine>>
:Engine1
loop
[As user needs]
execute(command_unit_list)
Returncode_t= execute(command_unit_list = assigned)
loop
[All assigned command messages in command_unit_list]
completed(command_id = assigned_id, status)
loop
[As user needs]
opt
[If users need detail information]
get_command_result(command_id, condition, results = null)
Returncode_t= get_command_result(–, –, results = outcome)



Query Interface

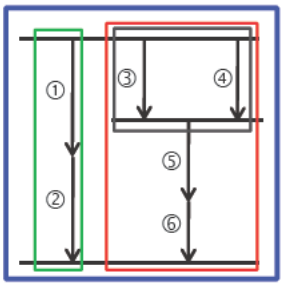


Event Interface



Command Unit List

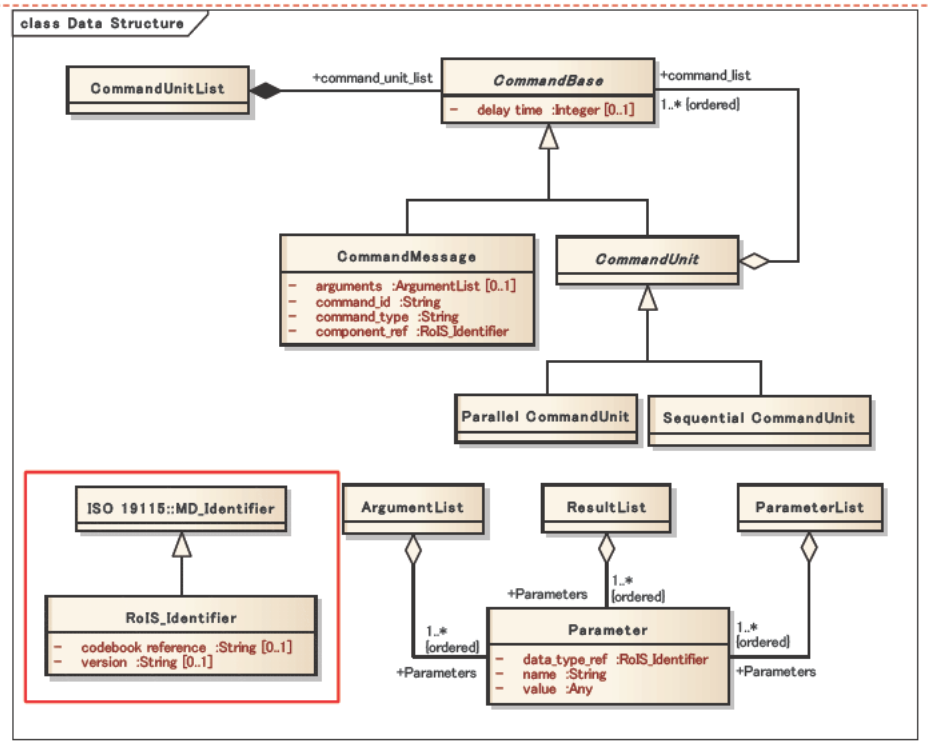
Example of sequential/parallel commands execution in a procedure



XML notation

```
<command_unit_list>
  <command_list_ type="parallel">
    <command_list type="sequential">
      <command_list type="message">①</···>
      <command_list type="message">②</···>
    </command_list>
    <command_list type="sequential">
      <command_list type="parallel">
        <command_list type="message">③</···>
        <command_list type="message">④</···>
      </command_list>
      <command_list type="message">⑤</···>
      <command_list type="message">⑥</···>
    </command_list>
  </command_list>
</command_unit_list>
```


Data Structure of Command Message



Basic HRI Component

robotics/2011-06-06

System Information

Description: This is a component for providing system information. The system information includes the status and the location of the system. This information belongs to the HRI engine that is treated as a unified physical unit of several HRI components. Therefore this component is different from other HRI components and does not include RoIS_Common methods.				
Query				
robot position	Returns location information.			
result	timestamp	DateTime [W3C-DT]	M	Measurement time.
result	position of unit	Set<Data>[RLS]	M	Set of location data.
engine status	Returns status information of the HRI engine.			
result	status	Component_Status	M	Status information of this engine.
result	operable time	DateTime [W3C-DT]	O	Operatable time of this engine.

RoIS_Common

Description: Common method for all HRI components.				
Command				
start	Start the functionality of the HRI component.			
stop	Stop the functionality of the HRI component.			
suspend	Pause the functionality of the HRI component.			
resume	Resume the functionality of the HRI component.			
Query				
component_status	Obtain status information of the component.			
result	status	Component_Status	M	Status information of this component.

Person Detection

Description: This is a component for detecting number of persons. This component notifies a number of the detected people when the number has changed.				
Command (In addition to RoIS_Common)				
Query (In addition to RoIS_Common)				
Event				
person_detected	notifies when number of people has changed			
result	timestamp	DateTime [W3C-DT]	M	Measurement time.
result	number of person	Integer	M	Number of detected persons

Person Localization

Description: This is a component for detecting position of persons. This component notifies position of the detected people when the position has been localized.				
Command (In addition to RoIS_Common)				
Query (In addition to RoIS_Common)				
Event				
person_localized	Notifies position of people when the position has localized.			
result	timestamp	DateTime [W3C-DT]	M	Measurement time.
result	person_ref	List<RoIS_Identifier>	M	List of detected person IDs.
result	position data	List<Data>[RLS]	M	List of detected person data. Each data entry at least contains position of the detected person.

Person Identification

Description: This is a component for identifying person ID. This component notifies ID(s) of the detected people when the ID(s) has been identified.				
Command (In addition to RoIS_Common)				
Query (In addition to RoIS_Common)				
Event				
person_identified	notifies when person ID is identified			
result	timestamp	DateTime [W3C-DT]	M	Measurement time.
result	person_ref	List<RoIS_Identifier>	M	List of detected person IDs. Reference information related to the ID shall be provided with the each ID.

Face Detection

Description: This is a component for detecting number of human faces. This component notifies a number of the detected faces when the number has changed.				
Command (In addition to RoIS_Common)				
Query (In addition to RoIS_Common)				
Event				
face_detected	Notifies number of human face when the number has changed.			
result	timestamp	DateTime [W3C-DT]	M	Measurement time.
result	number	Integer	M	Number of human faces

Face Localization

Description: This is a component for detecting position of human faces. This component notifies position of the detected human face(s) when the position has been localized.				
Command (In addition to RoIS_Common)				
Query (In addition to RoIS_Common)				
Event				
face_localized	Notifies position of human face when the position has localized.			
result	timestamp	DateTime [W3C-DT]	M	Measurement time.
result	face_ref	List<RoIS_Identifier>	M	List of detected human face IDs.
result	position data	List<Data>[RLS]	M	List of detected human face data. Each data entry at least contains position of the detected face.

Sound Detection

Description: This is a component for detecting number of sound sources. This component notifies a number of detected sound sources when the number has changed.				
Command (In addition to RoIS_Common)				
Query (In addition to RoIS_Common)				
Event				
sound detected	Notifies number of sound sources when the number has changed.			
result	timestamp	DateTime [W3C-DT]	M	Measurement time.
result	number	Integer	M	Number of sound sources.

Sound Localization

Description: This is a component for detecting position of sound sources. This component notifies position of detected sound source(s) when the position has been localized.				
Command (In addition to RoIS_Common)				
Query (In addition to RoIS_Common)				
Event				
sound localized	Notifies position of sound sources when the position has localized.			
result	timestamp	DateTime [W3C-DT]	M	Measurement time.
result	person_ref	List<RoIS_Identifier>	M	List of detected sound source IDs.
result	position data	List<Data>[RLS]	M	List of detected sound source data. Each data entry at least contains position of the detected sound source. .

Speech Recognition

Description: This is a component for recognizing human speech. This component notifies text data of the recognized speech when the speech has been recognized.				
Command (In addition to RoIS_Common)				
set_parameter	Specifies speech recognition parameters.			
argument	languages	Set<String>[ISO639-1]	M	Specifies languages the speech recognizer will recognize.
argument	grammar	String	O	Specifies grammar for the speech recognizer.
argument	rule	String	O	Specifies active rule in the grammar.
Query (In addition to RoIS_Common)				
get_parameter	Obtains speech recognition parameters. (In addition to the parameters treated in "set_parameter")			
result	recognizable languages	Set<String>[ISO639-1]	M	Obtains languages the speech recognizer can recognize.
Event				
speech recognized	Notifies recognized result when the speech has been recognized.			
result	timestamp	DateTime [W3C-DT]	M	Measurement time.
result	recognized_text	List<String>	M	List of speech recognition results.
speech input started	Notifies the recognizer has detected start of speech input.			
result	timestamp	DateTime [W3C-DT]	M	Measurement time.
speech input finished	Notifies the recognizer has detected end of speech input.			
result	timestamp	DateTime [W3C-DT]	M	Measurement time.

Gesture Recognition

Description: This is a component for recognizing human gesture. This component notifies ID of the recognized gesture when the gesture has been recognized.				
Command (In addition to RoIS_Common)				
Query (In addition to RoIS_Common)				
get parameter	Obtains gesture parameter			
result	recognizable_gestures	Set<RoIS_Identifier>	M	Obtains gestures that this component can recognize. The gestures are expressed as IDs.
Event				
gesture recognized	Notifies recognized result when the gesture has been recognized.			
result	timestamp	DateTime [W3C-DT]	M	Measurement time.
result	gesture_ref	List<RoIS_Identifier>	M	List of gesture recognition results. The result is provided as gesture types. The type is specifyed by gesture IDs.

Speech Synthesis

Description: This is a component for generating synthesized speech. This component act to generate synthesized speech by specifying the speech text.				
Command (In addition to RoIS_Common)				
set parameter	sets parameter for speech synthesis			
argument	speech_text	String	C	Text to synthesize (in plain text format).
argument	SSML text	String [W3C-SSML]	C	Text to synthesize (in W3C-SSML format).
argument	volume	Integer	O	Volume.
argument	language	String [ISO639-1]	O	Language of the speech.
argument	character	String	O	Character of the voice.
Query (In addition to RoIS_Common)				
get parameter	Obtains speech synthesis parameters. (In addition to the parameters treated in "set_parameter")			
result	synthesizable_languages	Set<String>[ISO639-1]	M	Information about languages that can be synthesized.
result	Synthesizable_charcters	Set<String>	M	Information about languages that can be synthesized.
Event				

※Condition: These elements shall be selected according to the speech text format.

Reaction

Description: This is a component for executing specified reaction. This component acts to execute specified reaction by specifying the reaction ID.				
Command (In addition to RoIS_Common)				
set parameter	Specifies reaction paramters.			
argument	reaction type	RoIS_Identifier	M	Reaction type. The type is specified by reaction ID.
Query (In addition to RoIS_Common)				
get parameter	Obtains reaction paramters. (In addition to the parameters treated in "set_parameter")			
result	available_reactions	Set<RoIS_Identifier>	M	Obtains reaction types the robot can execute. The reaction type is expressed as ID reference information.
result	reaction_ref	RoIS_Identifier	M	Obtains the current specified reaction type.
Event				

Navigation

Description: This is a component for commanding navigation toward specified destinations. This component act to move to the destination by specifying the position data of the destination.				
Command (In addition to RoIS_Common)				
set parameter	Specifies parameters for navigation.			
argument	target_position	List<Data>[RLS]	M	List of target position data.
argument	time_limit	Integer	O	Time limit for determining whether it is impossible to continue the navigation. [ms]
argument	routing_policy	String	O	Policy for determining the navigation route.
Query (In addition to RoIS_Common)				
get parameter	Obtains parameters for navigation. (Same as the parameters treated in "set_parameter")			
Event				

Follow

Description: This is a component for following specified object. This component act to follow an object by specifying the ID of the object.				
Command (In addition to RoIS_Common)				
set parameter	Specifies parameters for follow.			
argument	target_object_ref	RoIS_Identifier	M	Taget object. The object is specifyed by object IDs.
argument	distance	Integer	O	Minimum distance between the target and the robot. [mm]
argument	time_limit	String	O	Time limit for determining whether it is impossible to continue following. [ms]
Query (In addition to RoIS_Common)				
get parameter	Obtains parameters for follow. (Same as the parameters treated in "set_parameter")			
Event				

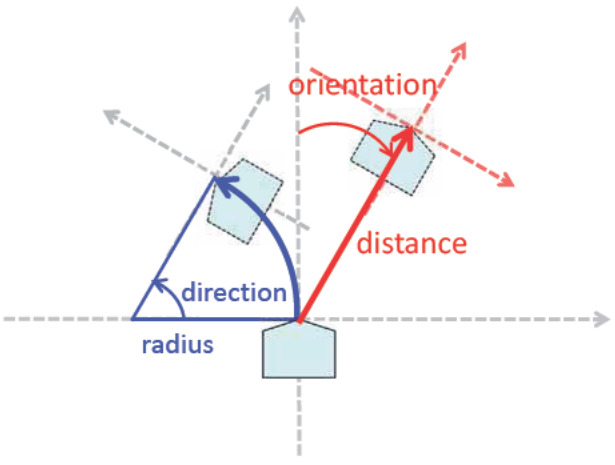
Move

Description: This is a component for moving based on a specified motion. The motion is simply specified by a line or a curve.				
Command (In addition to RoIS_Common)				
set parameter	Specifies parameters for move.			
argument	line	List<Integer>	C	Distance and orientation for specifying the line. [mm. degree]
argument	curve	List<Integer>	C	Radius and direction for specify the curve. [mm, degree]
argument	time	Integer	O	Operating time for the motion. [ms]
Query (In addition to RoIS_Common)				
get parameter	Obtains parameters for move. (Same as the parameters treated in "set_parameter")			
Event				

※Condition: These elements shall be selected according to the motion.

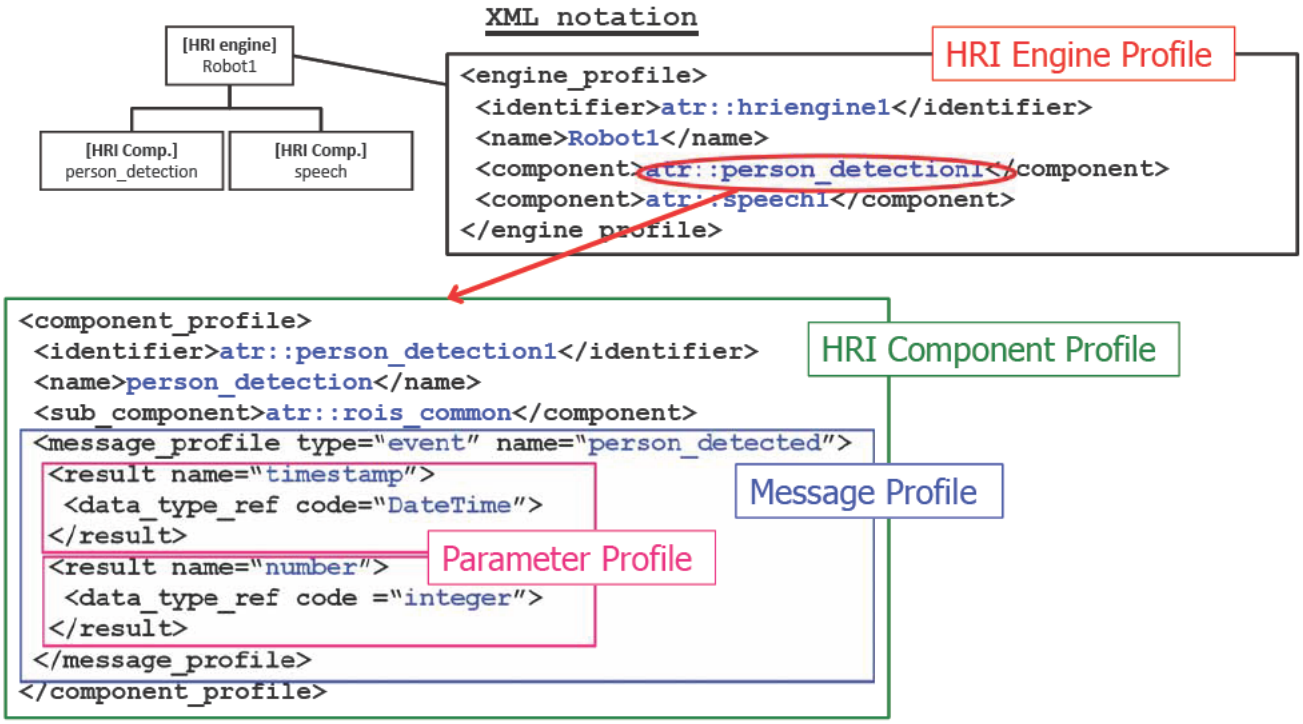
Arguments for "Move"

line: distance & orientation
curve: radius & direction
} conditional

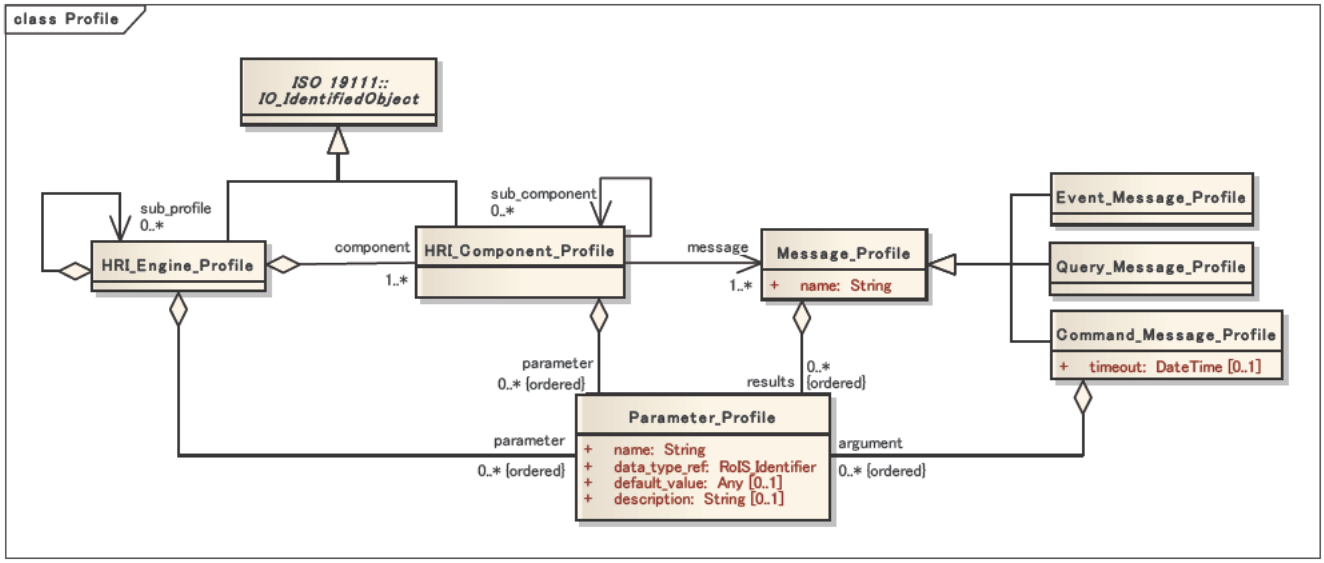


Profile

Example of RoIS Profiles



Class Diagram of RoIS Profiles



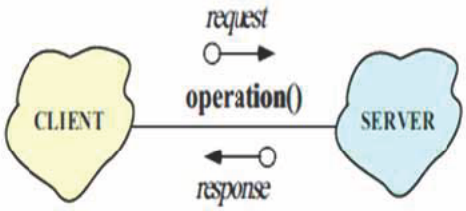
Event Management Model for DDC4RTC

OMG Meeting June 20-24, 2011 Salt Lake City

Seungwoog Jung
Infrastructure WG, Robotics DTF
ETRI, KOREA

OMG Event Service Spec

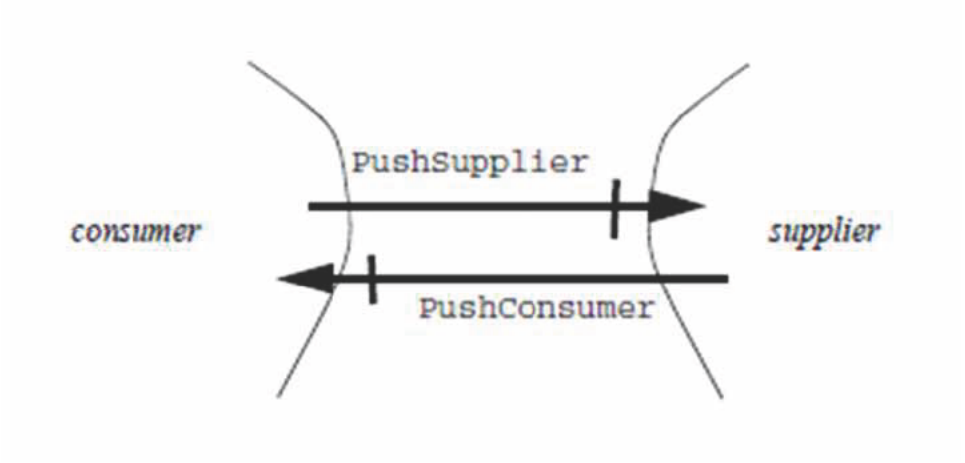
- Standard CORBA method invocations result in synchronous execution of an operation provided by an object



- For many applications, a more decoupled communication model between objects is required (i.e., asynchronous communication with multiple suppliers and consumers)
- OMG defines a set of event service interfaces that enable decoupled, asynchronous communication between objects

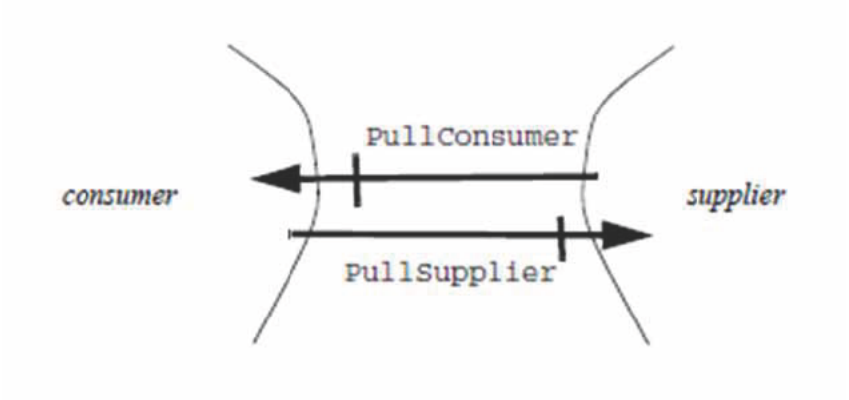
Event Communication Model

- Push Model
  - suppliers “push” event data to consumers; that is, suppliers communicate event data by invoking push operations on the PushConsumer interface



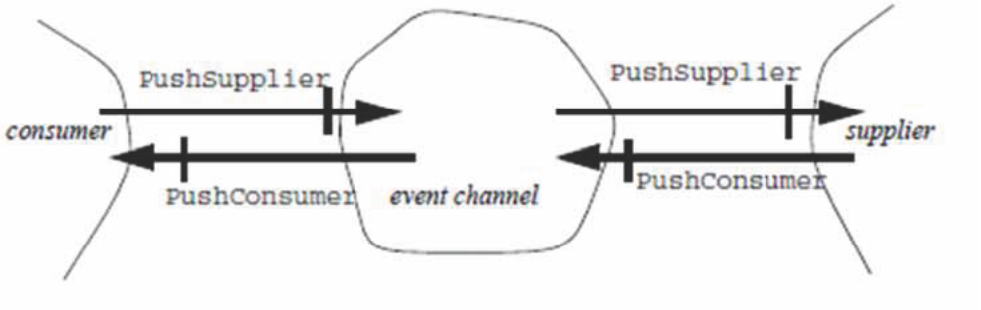
Event Communication Model

- Pull Model
  - suppliers “pull” event data from consumers; that is, suppliers request event data by invoking pull operations on the PullComsuer interface



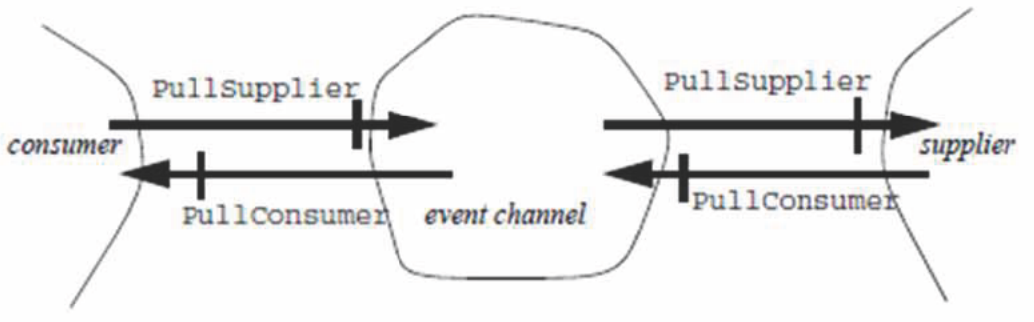
Event Channel

- The event channel is a service that decouples the communication between suppliers and consumers
- The event channel is itself both a consumer and a supplier of the event data
- An event channel can provide asynchronous communication of event data between suppliers and consumers
- Push-Style Communication with an Event Channel
  - The supplier pushes event data to the event channel; the event channel, in turn, pushes event data to the consumer.



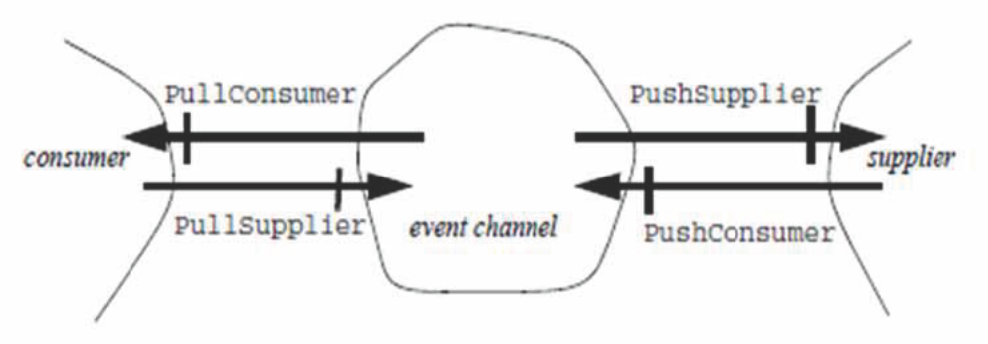
Event Channel

- Pull-Style Communication with an Event Channel
  - The consumer pulls event data from the event channel; the event channel, in turn, pulls event data from the supplier



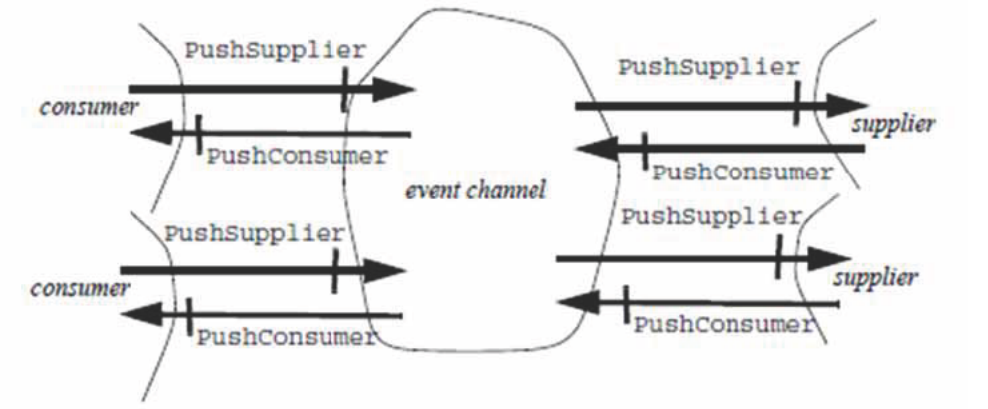
Event Channel

- Mixed Style Communication with an Event Channel
  - An event channel can communicate with a supplier using one style of communication, and communicate with a consumer using a different style of communication



Event Channel

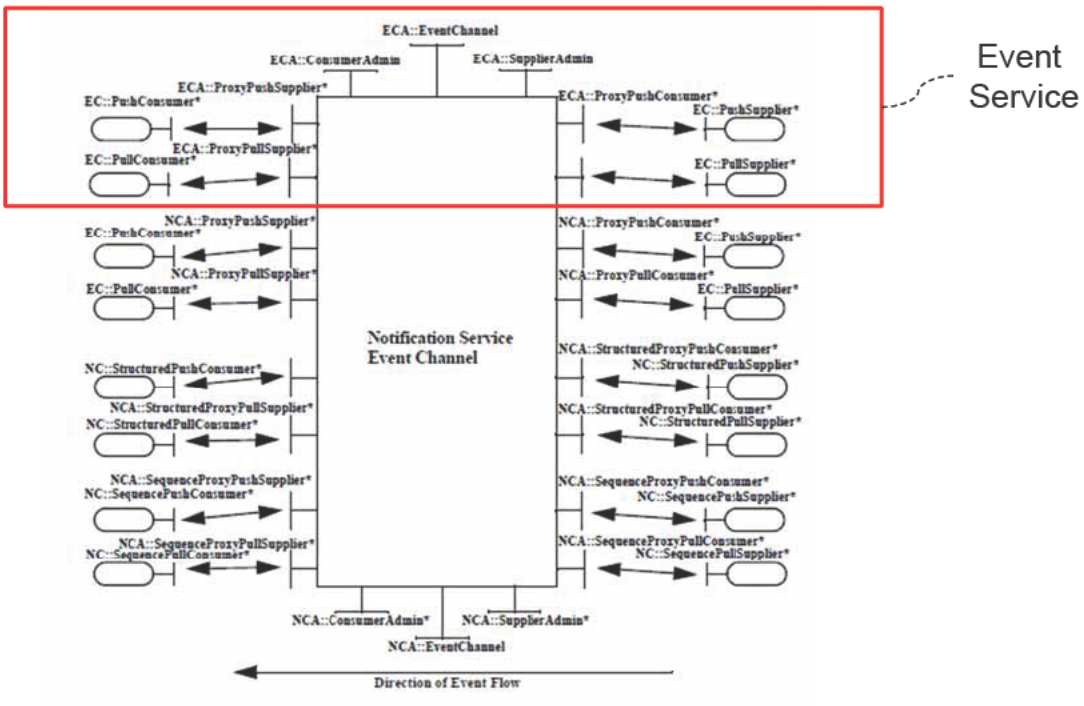
- Multiple Consumers and Multiple Suppliers



OMG Notification Service Spec

- CORBA-based Notification Service extends the existing OMG Event Service, adding to it the following new capabilities:
  - The ability to transmit events in the form of a **well-defined data structure**, in addition to Anys and Typed-events as supported by the existing Event Service.
  - The ability for clients to specify exactly which events they are interested in receiving, by attaching **filters** to each proxy in a channel.
  - The ability for the event types offered by suppliers to an event channel to be discovered by consumers of that channel so that consumers may subscribe to new event types as they become available.
  - The ability to configure various **quality of service properties** on a per-channel, perproxy, or per-event basis
  - An **optional event type repository** which, if present, facilitates the formation of filter constraints by end-users, by making information about the structure of events which will flow through the channel readily available.

Event Service vs. Notification Service

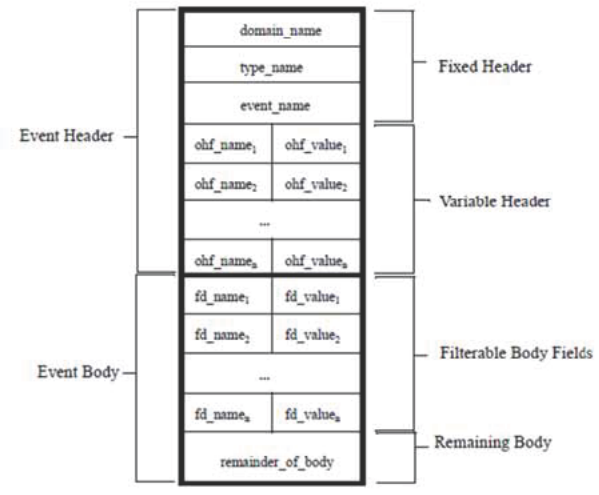


Notification Channel

- Notification Service event channel
- The default behavior of the Notification Channel is to deliver every event it receives to every subscribed consumer (This is also the behavior of the Event Channel)
- Notification Channel has the facility to filter events and thereby provide selective delivery.
  - To use this facility, consumers specify which events they are interested in receiving by registering a filter expression with the Notification Channel
  - The Channel then applies the filter expression to each event to determine whether it should be delivered to that consumer or not
- The Notification Service specifies four different kinds of Notification Channel: the Untyped-Event Channel, the Structured-Event Channel, the Sequenced-Event Channel, and the Typed-Event Channel

Notification Channel

- The Untyped-Event Notification Channel uses the CORBA::Any datatype to represent events
- The Structured-Event Channels use predefined structured datatypes to represent events



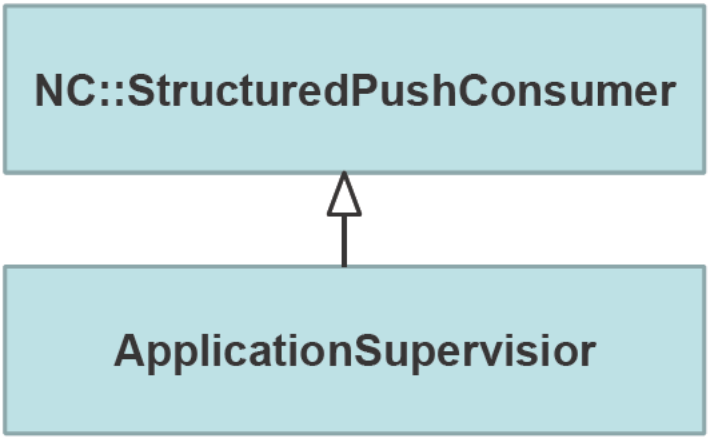
<The structure of a Structured Event>

Notification Channel

- The Sequenced-Event Channel's event datatype is a sequence of the Structured-Event Channel's event datatype
- A significant advantage of both Structured-and Sequenced-Event Channels is that their event filtering activities are much more efficient than the other kinds of channels because the Event Structure is known to the Channel *a priori meaning that event demarshalling and inspection* algorithms that must be implemented in the Channel can be significantly optimized at design time
- In the Typed-Event Notification Channel, clients communicate with Typed Channels through application-defined interfaces.
  - The Notification specification refers to these interfaces as <I> interfaces. Operations of an <I> interface which are to be used to transfer events must specify **void** return type and not use any out or inout parameters

Event Management Model of DDC4RTC

- We don't have to define additional event system for DDC4RTC
- We can use OMG notification service specification

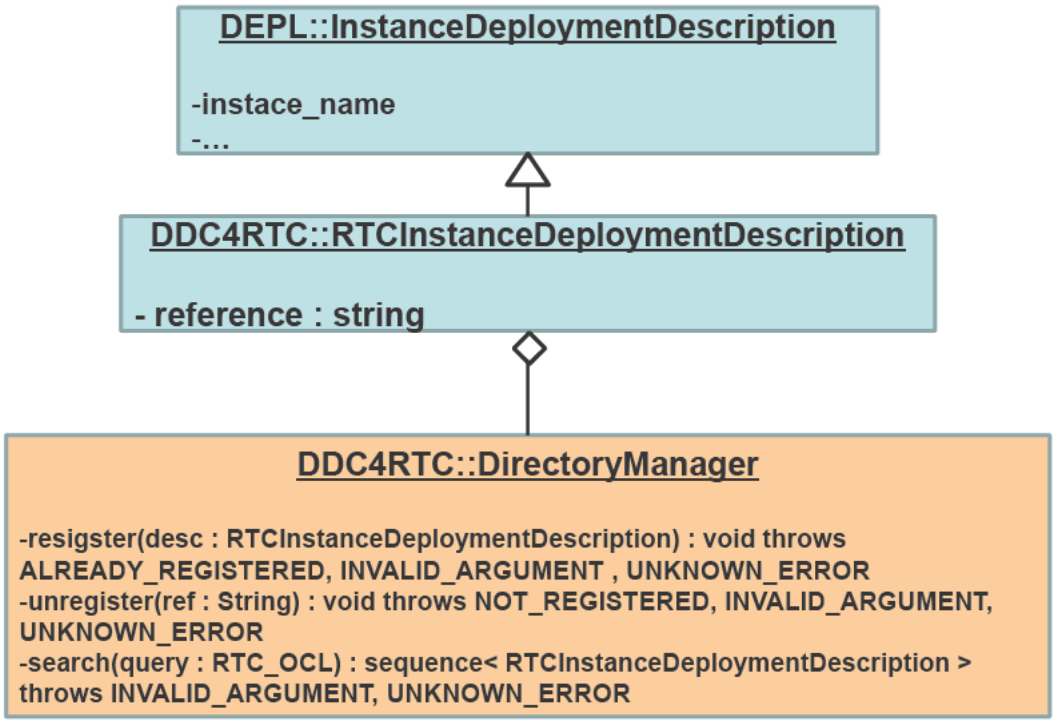


Repository & Directory Managers for DDC4RTC

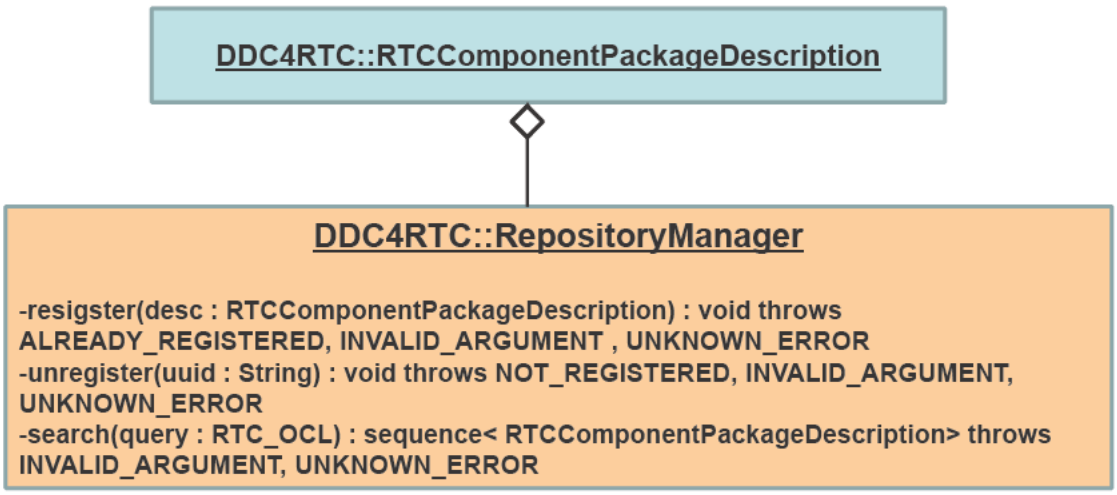
OMG Meeting June 20-24, 2011 Salt Lake City

Seungwoog Jung
Infrastructure WG, Robotics DTF
ETRI, KOREA

Directory Manager for DDC4RTC



Repository Manager for DDC4RTC



OPRoS

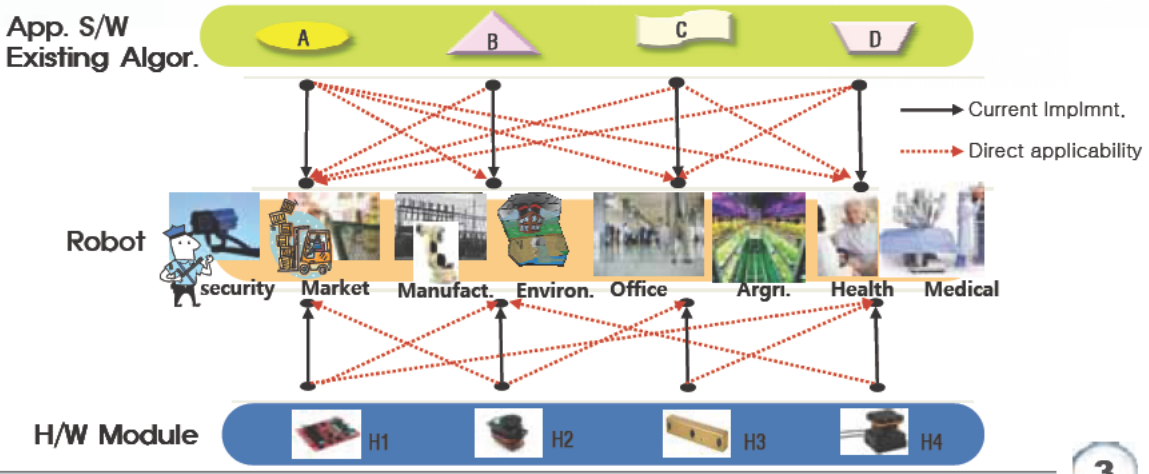
(Open Platform for Robotic Services)

2011.6.21

Hong Seong Park
Kangwon National University
hspark@kangwon.ac.kr

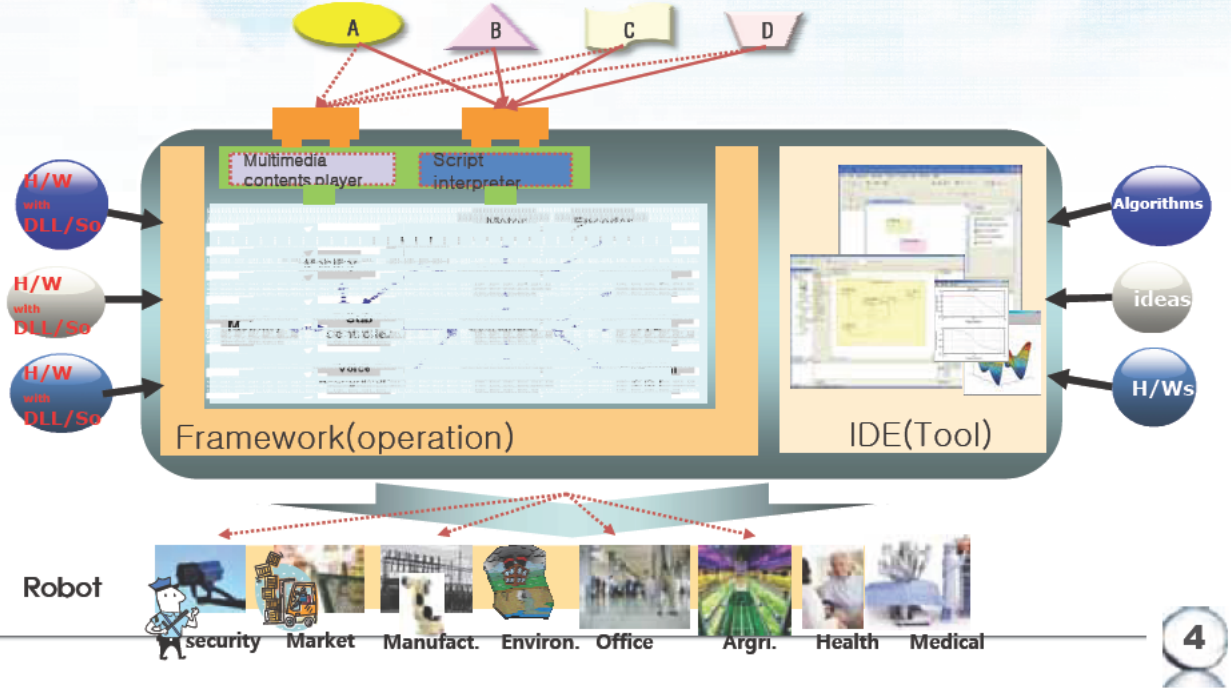
1. Introduction : Problem Statement

- Hurdles preventing from vitalizing the robot market (compared to PC market)
  - Incompatibility of application SW
    - App. SW A in Robot I is not immediately operated in Robot II, III, and IV. Some additional tasks are required.
  - Incompatibility of HW modules
    - HW module H2 in Robot II is not operated in Robot I, III, or IV because their driver SW is not suitable to robots.
  - Difficult recycling of existing algorithms
    - There have been lots of useful algorithms in the navigation, manipulation, and object recognition fields. But it is difficult to reuse the algorithms because they have been developed with some constraints, such as special interfaces.



1. Introduction : Objective of OPRoS

- The Objective of OPRoS
  - Make the development of Robot and Robot SW easy, convenient, and fast
  - Remove the hurdles
  - Do easy adoption of new IT technology to Robots.



1. Introduction : Architecture of OPRoS

Component based S/W

- Robot modeler & environment modeler
- Device Modeler
- Debugger

Simulator (Text, Graphic)

Server technology

- Server technology
  - Service provider/binder
  - Service Description
  - SOA
  - Repository

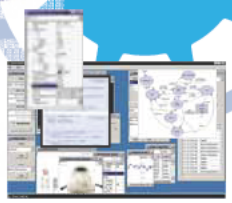
Component

Evaluating and Testing tool

- Content composing tool
  - C, C++
  - Task scripts
- Component composing tool
  - Component composer
  - Component Editor

Editor (Text, Graphic, Block) Compiler

Component Execution Eng & Script Interpreter

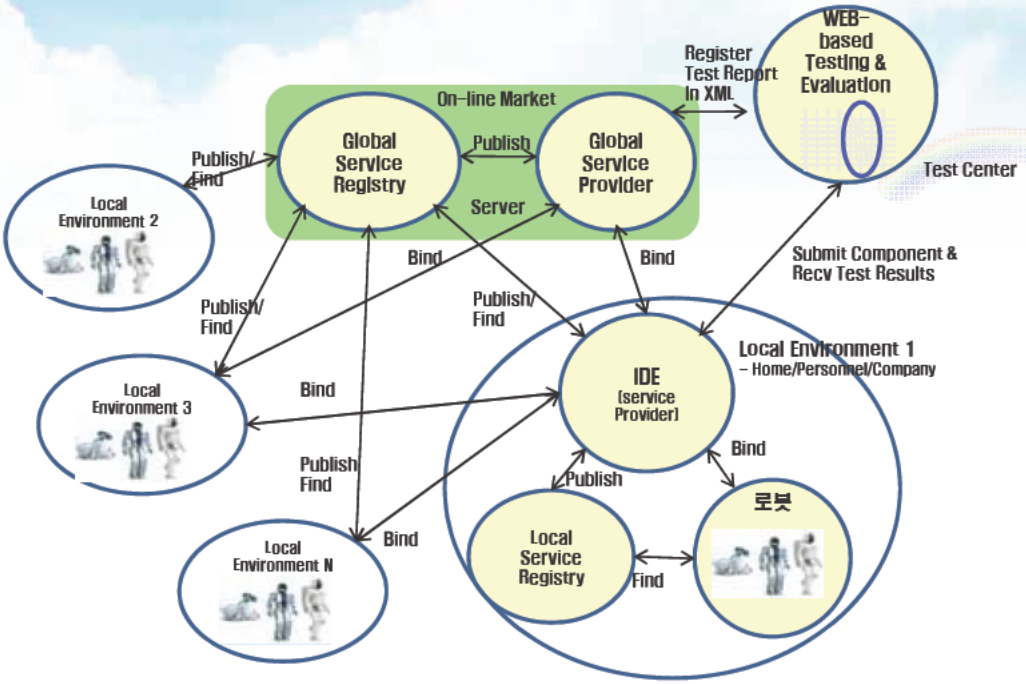


- Component Scheduler
- Component Manager
- Task Script Execution

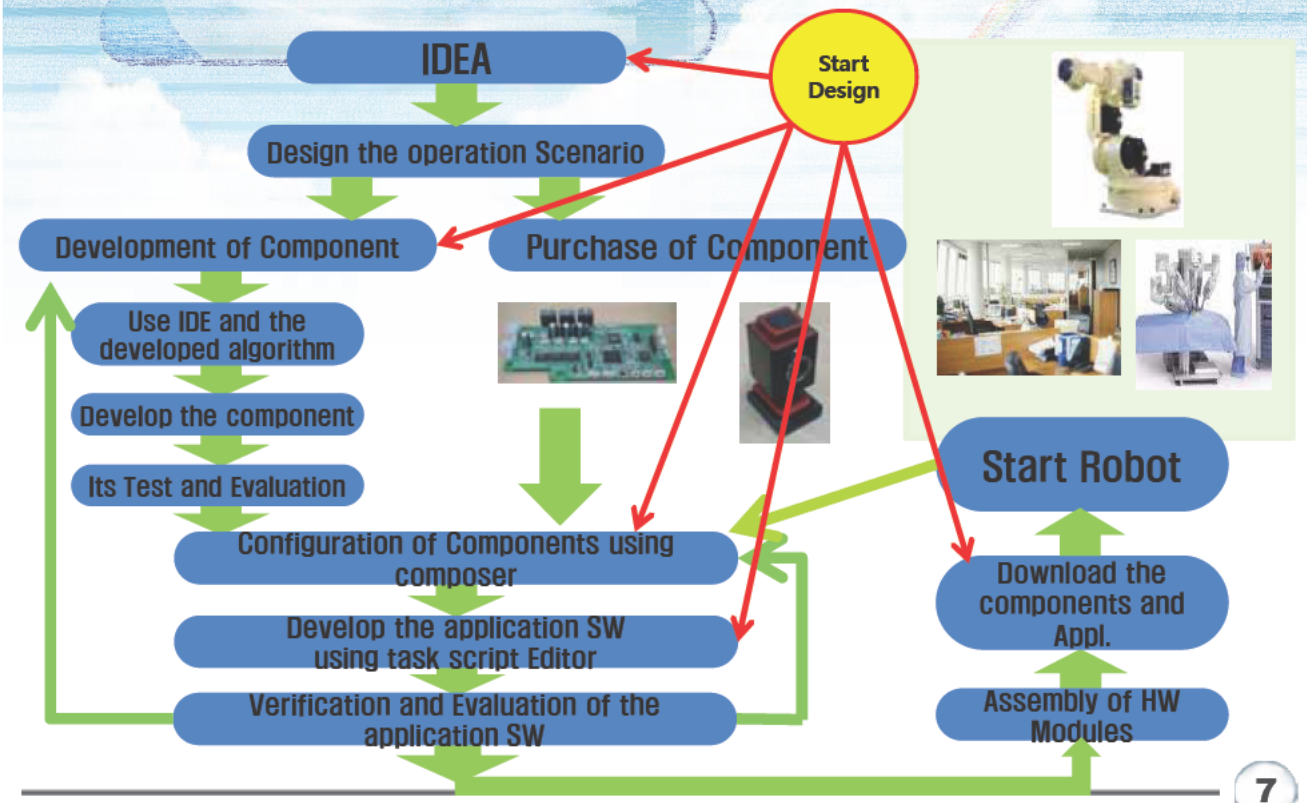
- Evaluation/Testing technology
  - WEB/Local Test
  - Support stability of components and Contents
  - Automation of generation and test of test cases

1. Introduction : Operation Environment

Feature : Focus on APP store

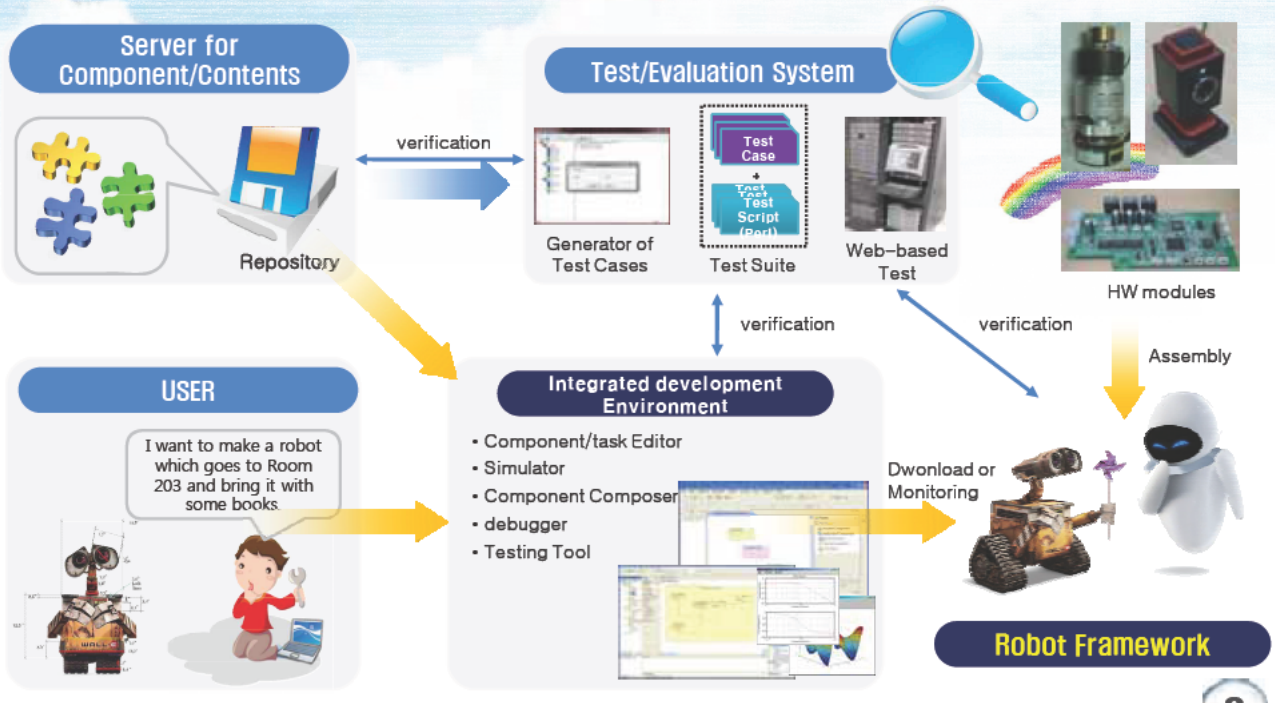


1. Introduction : Usage of OPRoS



1. Introduction : Component/Content Flow

컴포넌트/컨텐츠 흐름



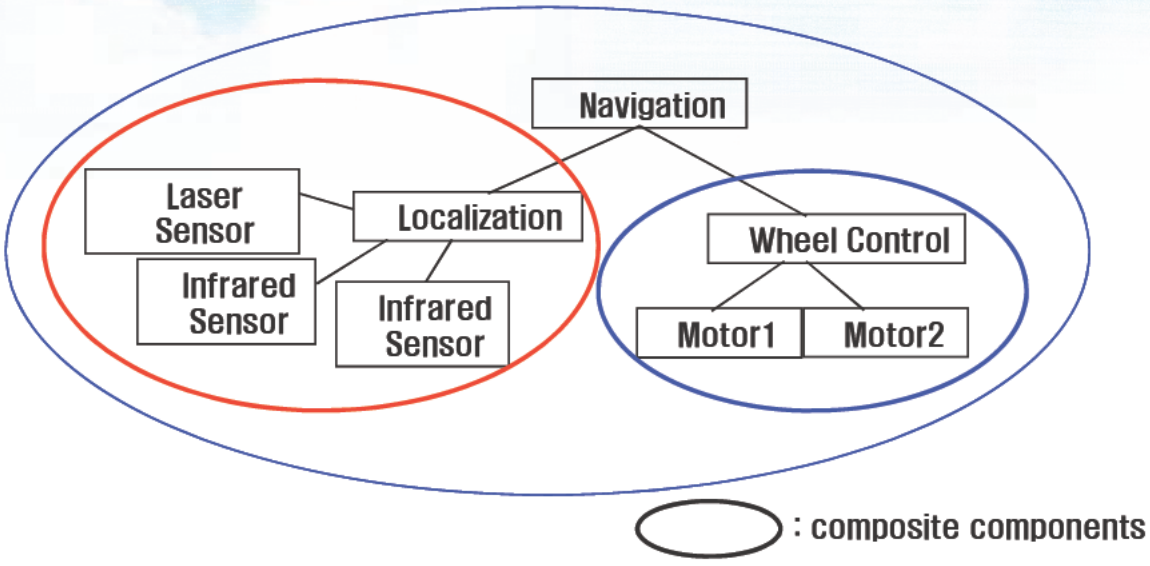
Operation of OPRoS : Video Clip

2. Component

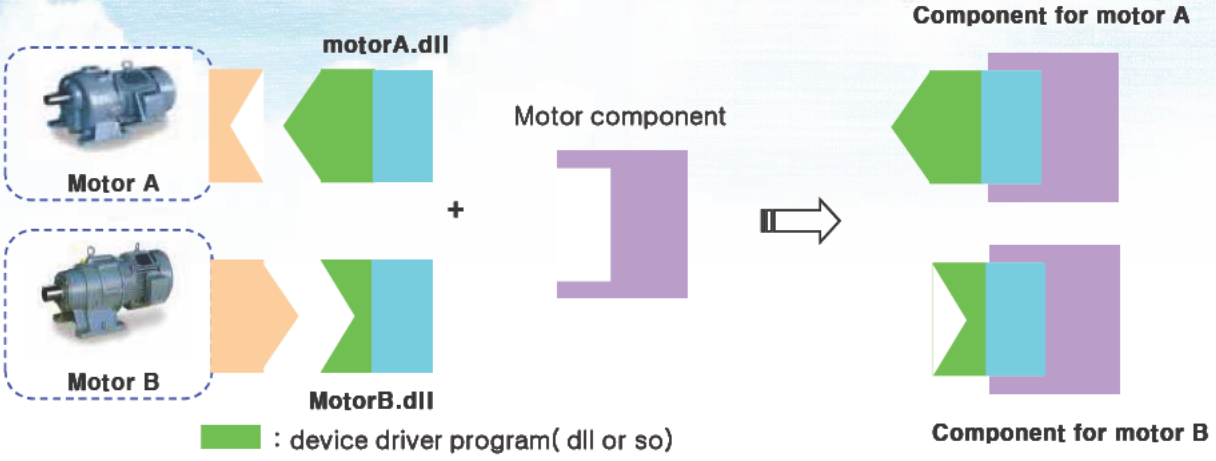
- The basic SW module for operating robots
- Support Data Port, Service Port, and Event Port, and their multiple connections are possible
  - Data Port : transmit data between components: periodic / non-periodic transmission
  - Service Port: execute service functions
  - Event Port : send and receive events that occur in each component
- HW linked component
  - Sensor Component, Actuator Component, Vision Component, etc.
- SW component
  - Components using HW linked component, Navigation / DB Component, etc.
- Depend on the kinds of building robots
  - Navigation Component can be connected to every HW linked component directly, or to a Sensor Component and a Actuator Component composed of multiple components
- For the same kinds HW, use the same Component and switch HW linked DLLs
  - Provide standardized API for each sensors and actuators

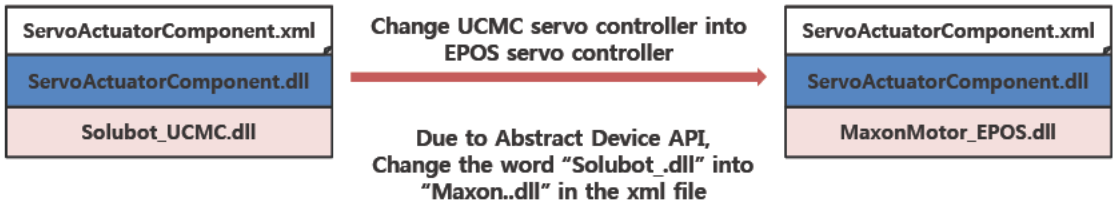
2. Component

- Examples of composite components



2. Component : ex) Servo Motor component





3. Framework

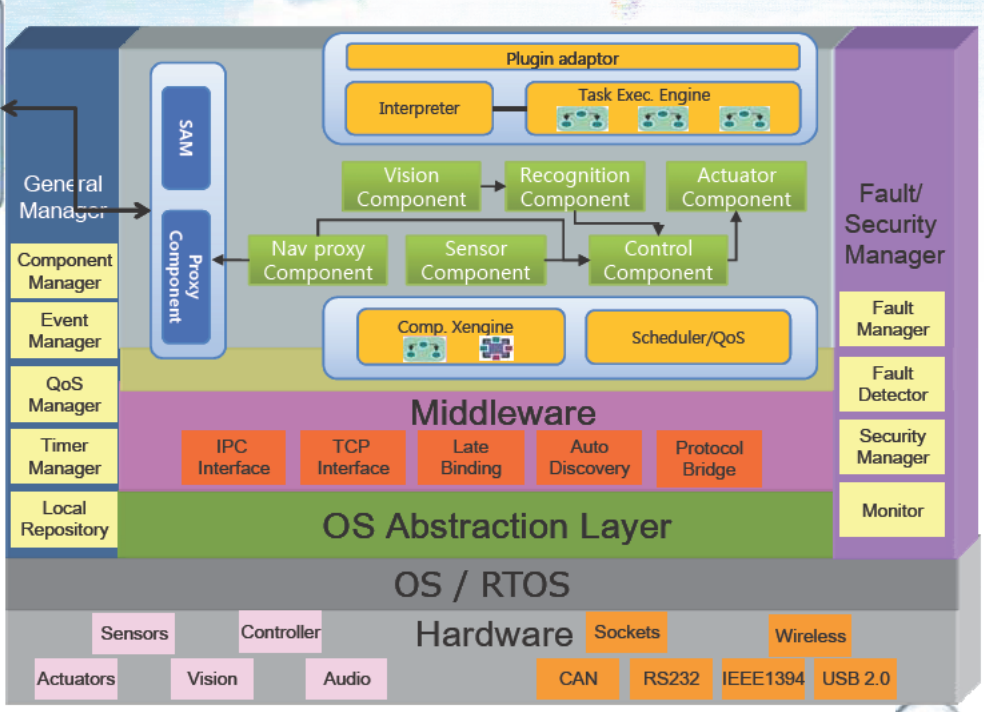
- **Characteristics**
  - Component based structure
    - Simple structure for using different kinds of components
    - To change the linked component, simply loading the HW driver routine (.dll or .so)
  - Work in Windows XP , Linux, and Embedded OS, RTOS(QNX)
  - Various functionalities are available, that selective functions can be added into the required functions
    - Use an interpreter, that is composed of script Language, for program portability
    - Use compiled component for program speed
    - Build easily various robot SW(Reactive model, Behavior model, Hybrid Model, User Creative model, etc.)
  - Guarantee the portability of applied program (used in Windows and Linux both)
    - Provide application programs that are composed of task scripts and VPL language
    - (an applied program means a program that makes robots moving as the user wanted)
- **Specification**
  - 32bit CPU (plan for 16 bit CPU to be supported later)
  - Minimum memory usage: 16MB (plan to be reduced)
  - Used programming language
    - Task Script, VPL
    - C++/C, Control Block

3. Framework

- **Functionality**
  - Execution Engine (required)
    - Minimum period : 1 ms for Linux, 50 ms with jitter about 1.3ms for Windows
    - No missing period,
    - Scheduling according to precedence relationship
  - Component / Application Program Manager (required)
  - Communication Manager (required)
  - Middleware/Component Adapter (required)
  - Event Manager (selective)
  - Fault Manager (selective)
  - Multimedia Execution Engine (selective)
  - Script Interpreter/Plug-in Manger (selective)
  - Security Manager (selective)
  - Time Manager (selective)
- **Supported OS : Windows/Linux/RTOS(QNX)**
- **So far applied robots**
  - LEGO NXT, Mini robot Robonova, Dasa robot TETRA, Yujin robot iRobiQ,
  - Robotware E3, ED robot ED 7270,
  - Industrial robot
  - rehabilitation robot/humanoid (plan to be implemented)

3. Framework

Server



Middleware
: UPnP & CORBA
Programming Language
: C++/C
OS
: Windows, Linux, Embedded OS

4. Integrated Development Environment

- **Characteristic**
  - Provide various integrated Eclipse based editors
  - Component Editor for creating, editing, testing, and debugging components
    - Provide templates to make HW and SW components easily
    - In case of homogenous devices, using API's to make HW devices add easily
  - Component Editor composes robot SW easy using various components
    - Independence from OS, and SW stability and verification functionalities are provided
    - Easy constructing various kinds robot models (reactive model, behavior model, mixed model, user created model, etc.)
  - Task GUI/Text based Editor makes creating various applied programs (content) easy
    - Independent from OS, and debugging function linked with the simulator is provided (plan to provide debugging function)
    - Support various robot controls and applications linked with component variables (under development)
  - Control Block Editor helps developing control programs like motors easy, and simulate the programs
    - Provide various commends using control algorithms, and models of motors
    - Support user defined function calls (planning)
  - Graphic programming Editors for developing various application programs (under development)
    - Graphic language program
    - Provide user defined function calls

4. Integrated Development Environment

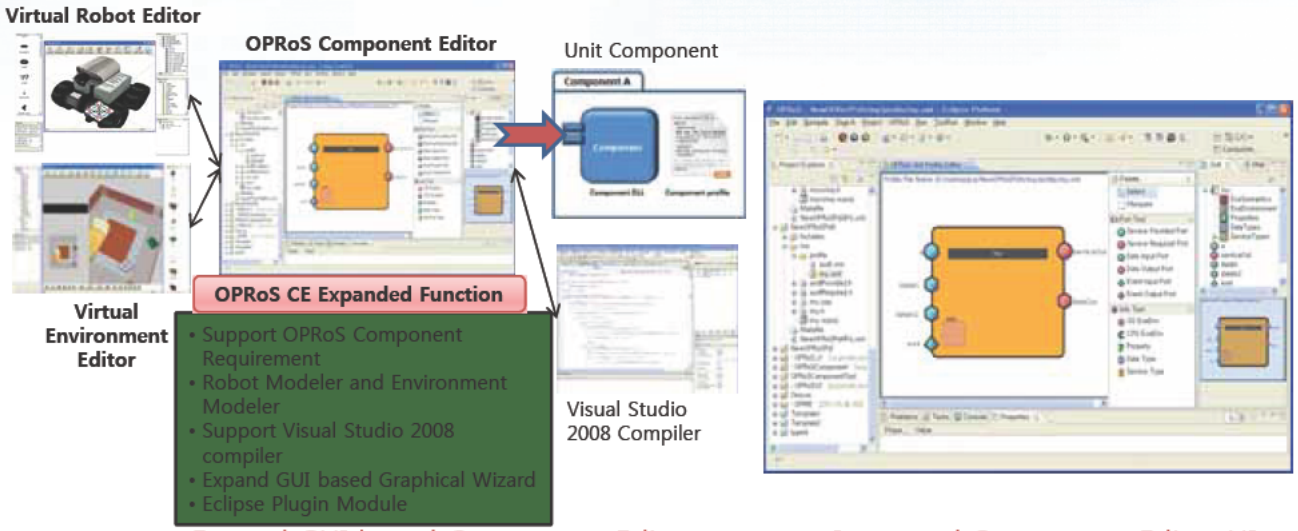
- **Characteristics**
  - Provide editors tailored to applications

Type of SW		Editors	Application Examples	Debugging
Applied program	•Contents •Tasks	•Task Editor •VPL editor	•Errand Services, Public Guide Service •Assembling, Dancing etc.	Verification and Evaluation Tool Simulator
Combined SW Modules	•Robot Model SW	•Component Block Editor	•Guide/Public/Educational Service Robot •Industrial Robot	
SW Modules	•HW Linked Component •Only SW linked Component	•Component Editor	•HW Control, Environment Awareness •Location Control, Driving •Voice, Vision etc.	
		•Control Block Editor	Server/DC Motor, etc.	Special Simulator

  - Verification/Evaluation Tool
    - Tools for verifying and evaluating SW components, and application programs
    - Generate test cases, and tests them automatically
  - Simulator simulate components and the application programs
    - Provide environment modeler modeling various environments easy, and robot modeler modeling various robots easy (under development, partial functionalities are implemented)
    - The framework and the application program, run the real robots & run simulator
  - Not only just open the source code of the editors, but also commercializing is granted
  - Integrate and link editors and unify the UI (under development)

4. Integrated Development Environment

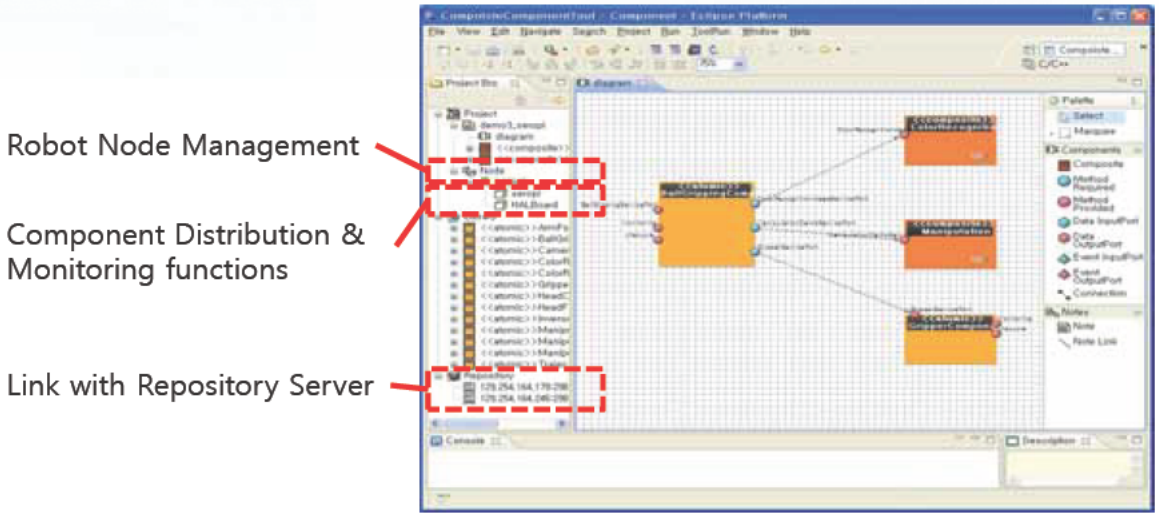
- **Component Editor**
  - Improve user accessibility, usability, convenience
  - Provide functionalities of GUI based component editing and link to VS 2008 compiler/GNU Compiler
  - Link with robot modeler and environment modeler for simulation



<Expand GUI based Component Editor> <Improved Component Editor UI>

4. Integrated Development Environment

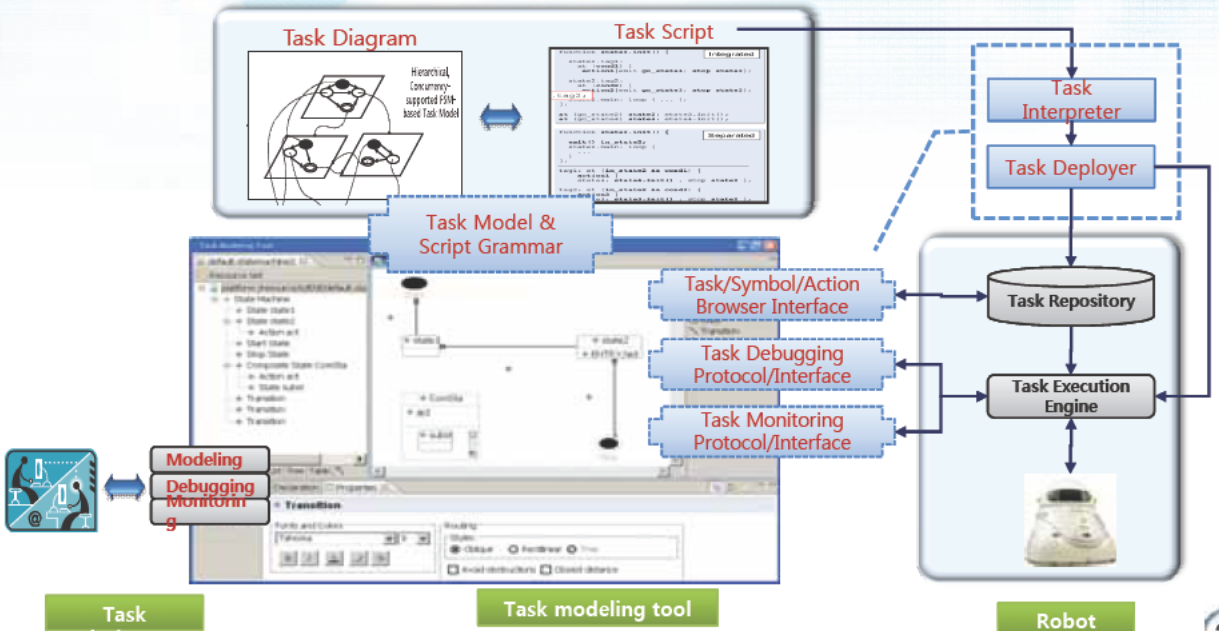
- **Component Composer**
  - Link Component Execution Engine and Repository Server
  - Multiple node management
  - Component remote distribution / monitoring / controlling functions



19

4. Integrated Development Environment

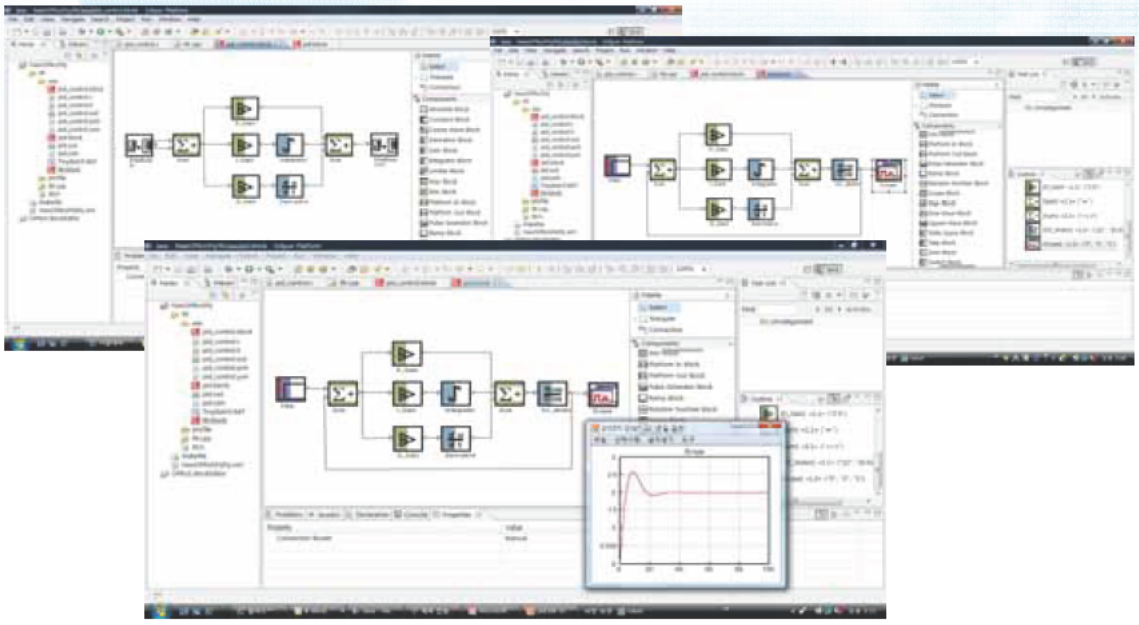
- **Task Script Editor**
  - Based on Finite State Machine
  - Generation of Script Prgram



20

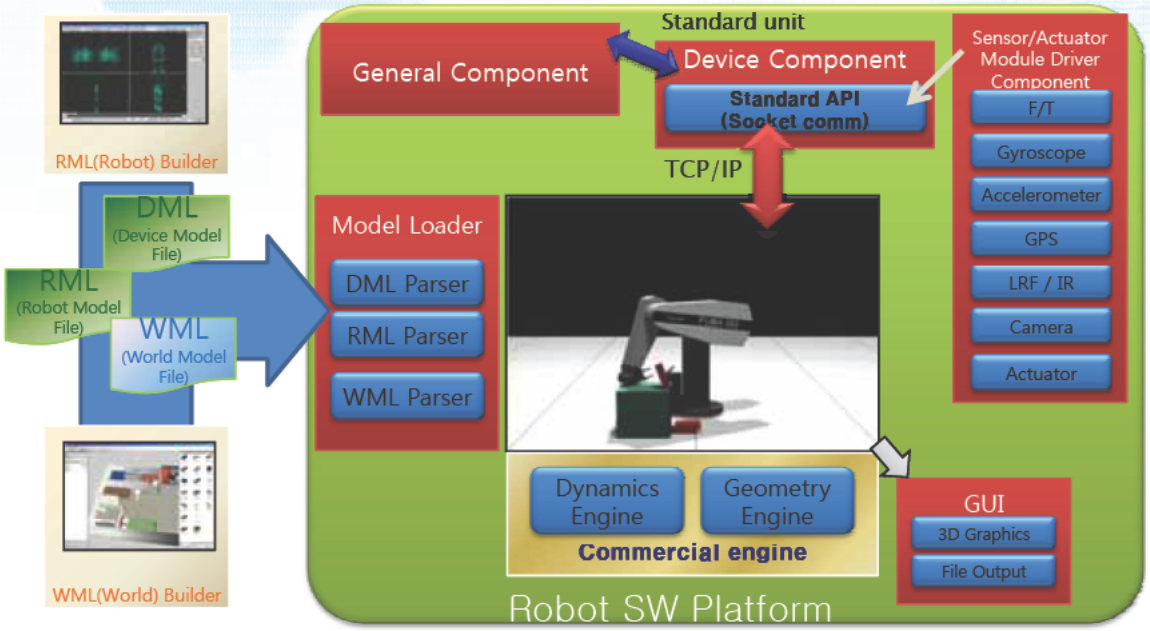
4. Integrated Development Environment

- **Control block Editor**
  - **Based on CemTool**



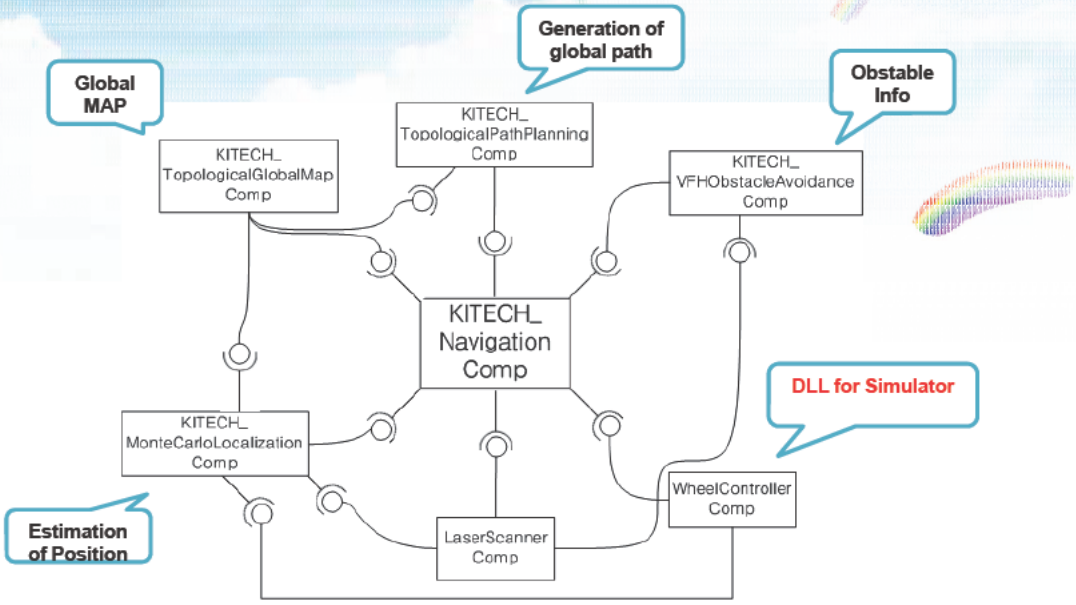
4. Integrated Development Environment

- **Simulator**
  - **Generation of Script Prgram**



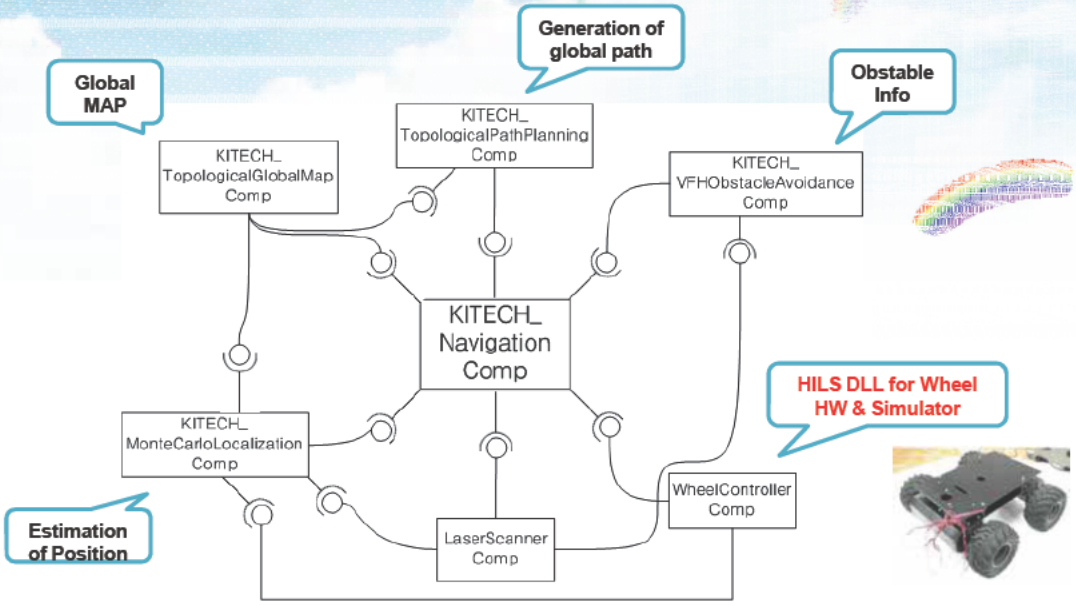
4. Integrated Development Environment

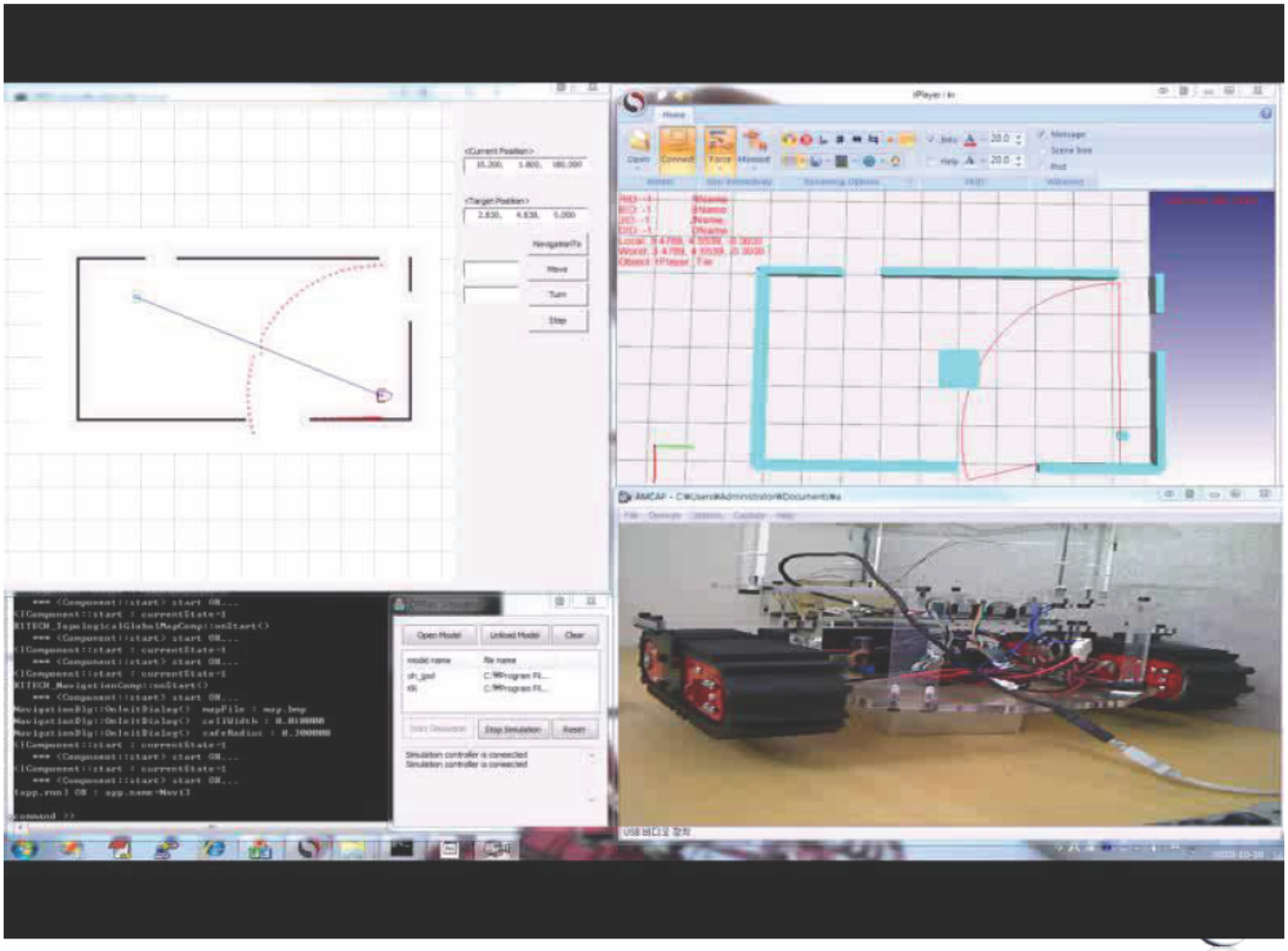
◆ Components for Simulation

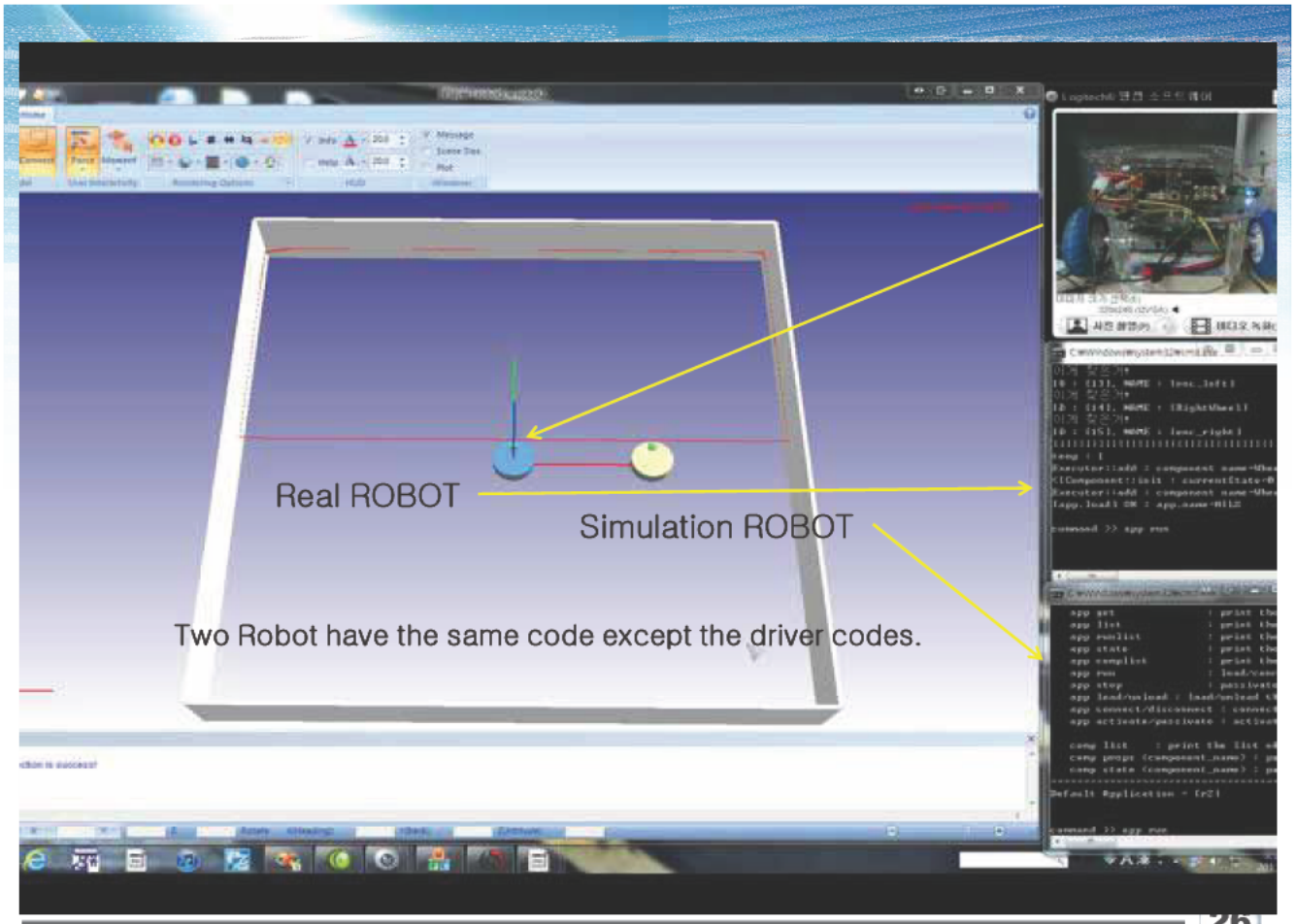


4. Integrated Development Environment

◆ Components for HW in the Loop Simulation(HILS)

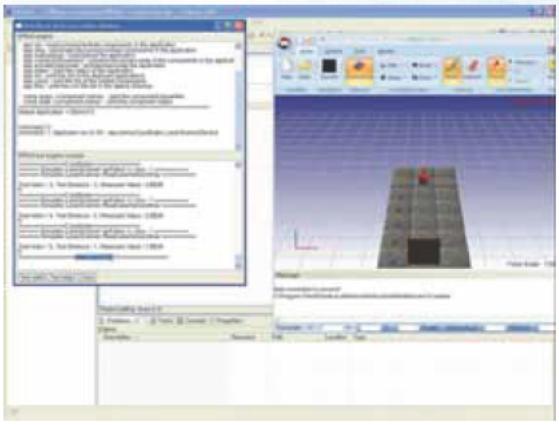






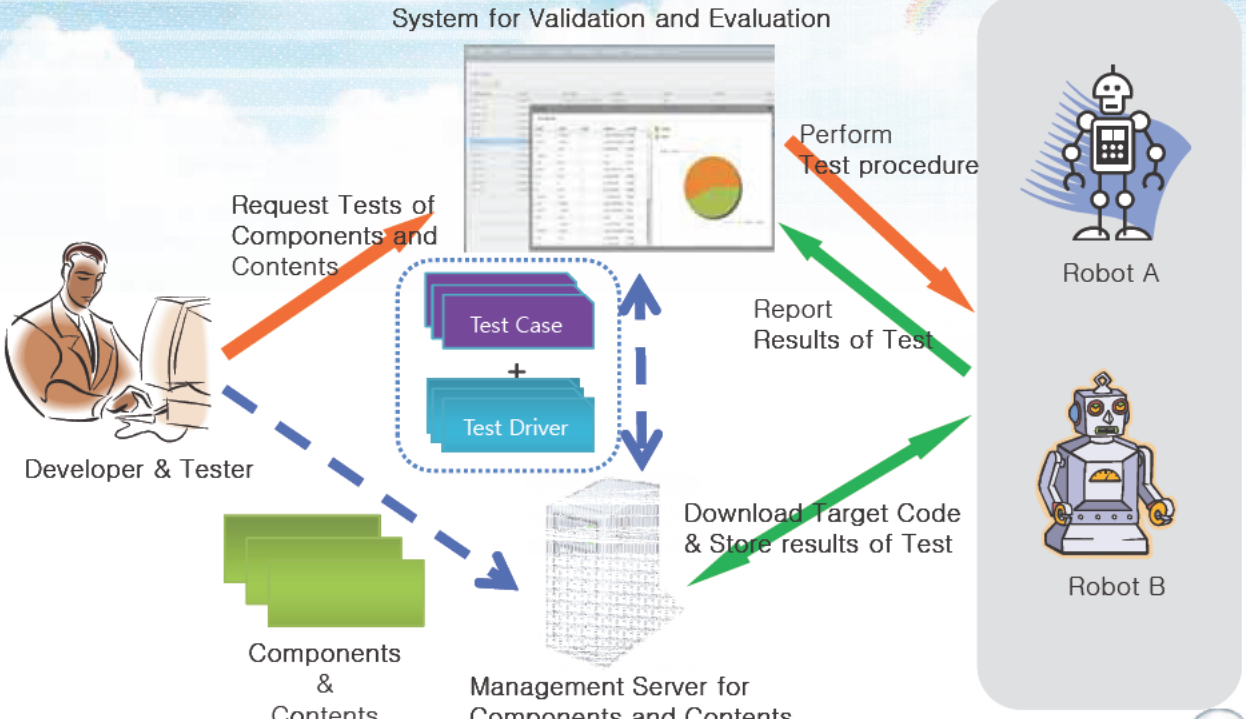
5. Testing & Evaluation

- **Characteristics**
  - Including components, robot SW tools for reliablity and function evaluation
    - Linking robot HW with simulator, verification and evaluation can be done without HW
  - Saving time from automatic robot SW module test
    - Reservation of testing time is capable of testing in difficult time, such as the middle of the night
  - Generating various test cases automatically for robot SW unit module
    - Unit module stability and performance evaluation results are in XML
  - Web based unit test, component test, GUI test, and robot system test are available
  - Eclipse based unit test, component test, GUI test and Content's test are possible
    - Linked to Component/Component Block/Task editors (plan to develop)



Example of testing Inferard sensor component with simulator

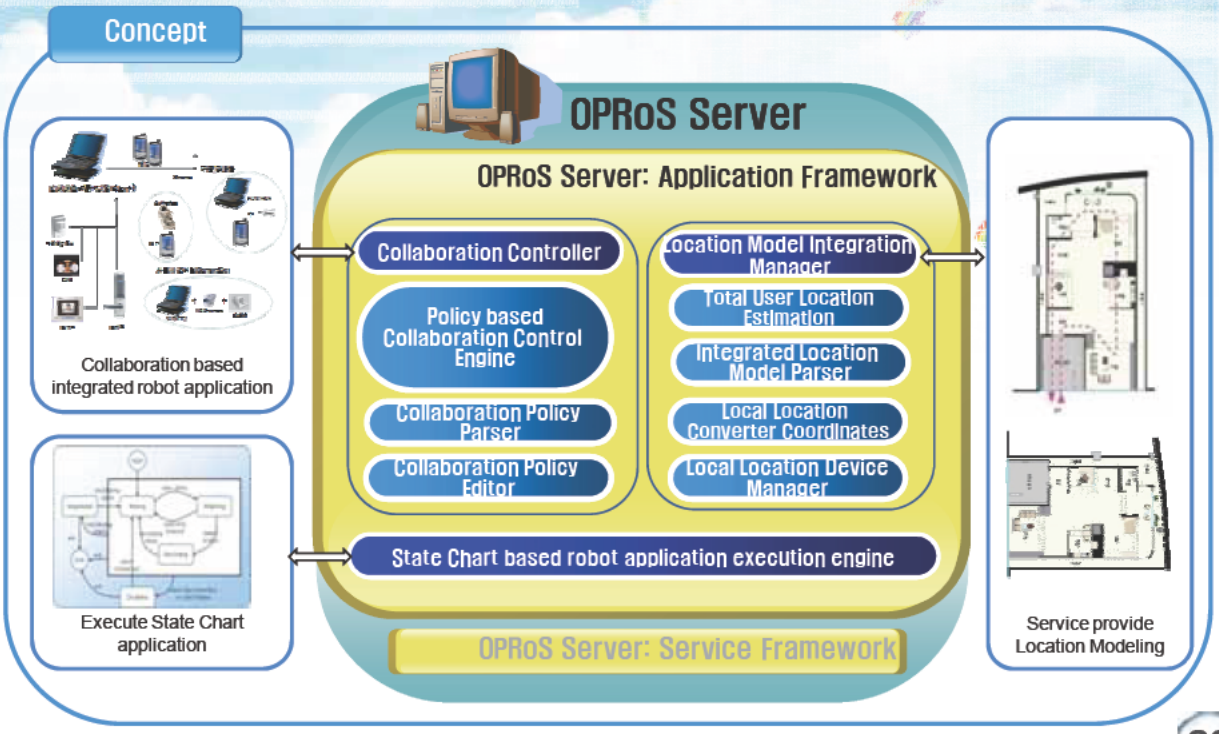
5. Testing & Evaluation



6. Server

- **Characteristics**
  - Support linking various robots and sensors in physical space
  - Gather situation information from various environmental inputs, and supports controlling embedded devices in the environment
  - manage component/service for robots and support computing resource
  - Support linking with various robot application system
  - Provide exclusive communication middleware between robot and environment
- **Specification**
  - Supported Operating Systems
    - Windows/Linux
  - Used programming languages
    - JAVA/C++
- **Application field**
  - Collaboration work more than 2 robots
  - Developing robot service business, and developing home service business linked with robots
  - Application development with robot and Smart Space

6. Server



License for Source Code and Download

- License for source code
  - Dual License mode
    - LGPL (GNU Lesser General Public License) for free usage
    - individual license for commercialization

- Download Site
  - http://www.opros.or.kr/

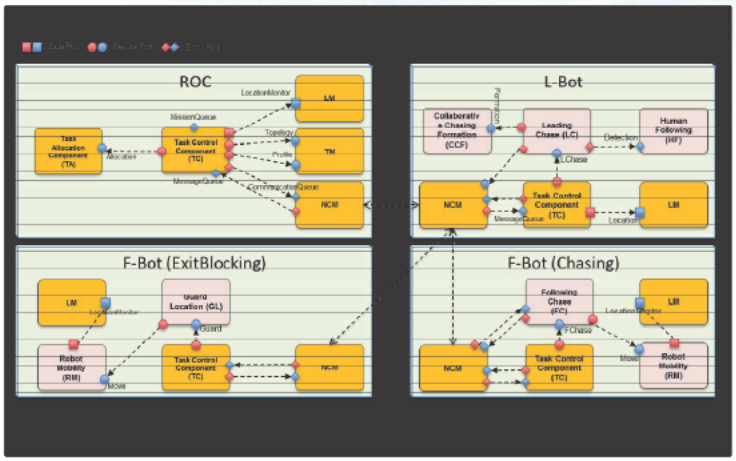
Summary

- Component based, and HW vendors Independent structure development
- Flexible Framework
  - Provide the scheduling function to process real-time components on the general OS such as Linux and Windows under some constraints
  - Provide various functions for fault management (fault-safe and fault-stop)
- Provide various tools for general users, and developers
  - For general users, task editor and simulator are enough to run robots
  - For developers,
    - Develop for motor control algorithm using control block editor
    - Then convert component and development
    - Develop reactive model and behavior model for user wanted robot
      : use pre-developed sensors/control component, navigation, vision
    - Developers utilize simulator and test verification tools for robot run
    - Complete robot development
    - General users, who are not developers, uses task editor to run the program

Group Robot

- consists of one Leader Robot and Multiple Robots

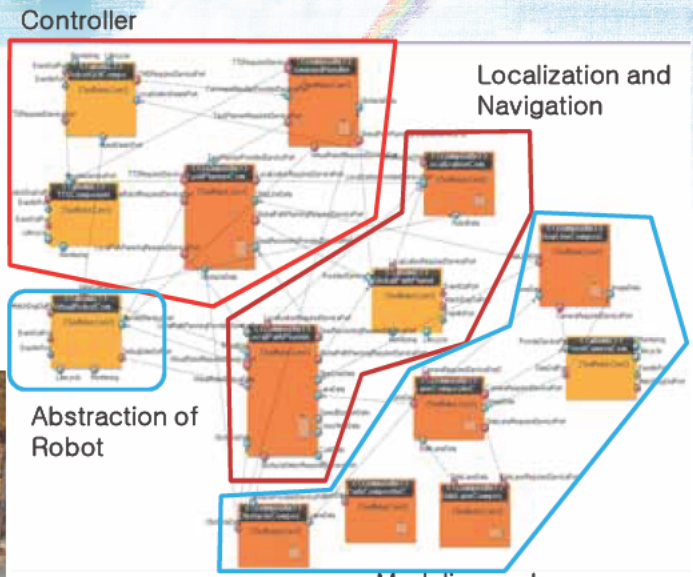
Platform



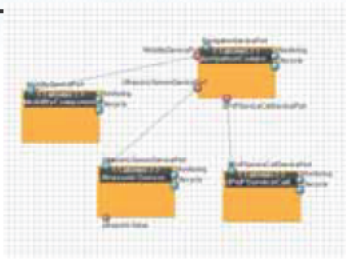
Configuration of OPRoS Components

Smart Transport Robot(ESTRO)

- ESTRO(Etri Smart Transport RObot)
- Sensors
  - Camera, LRF, LRF, GPS, Odometer



Serving Robot

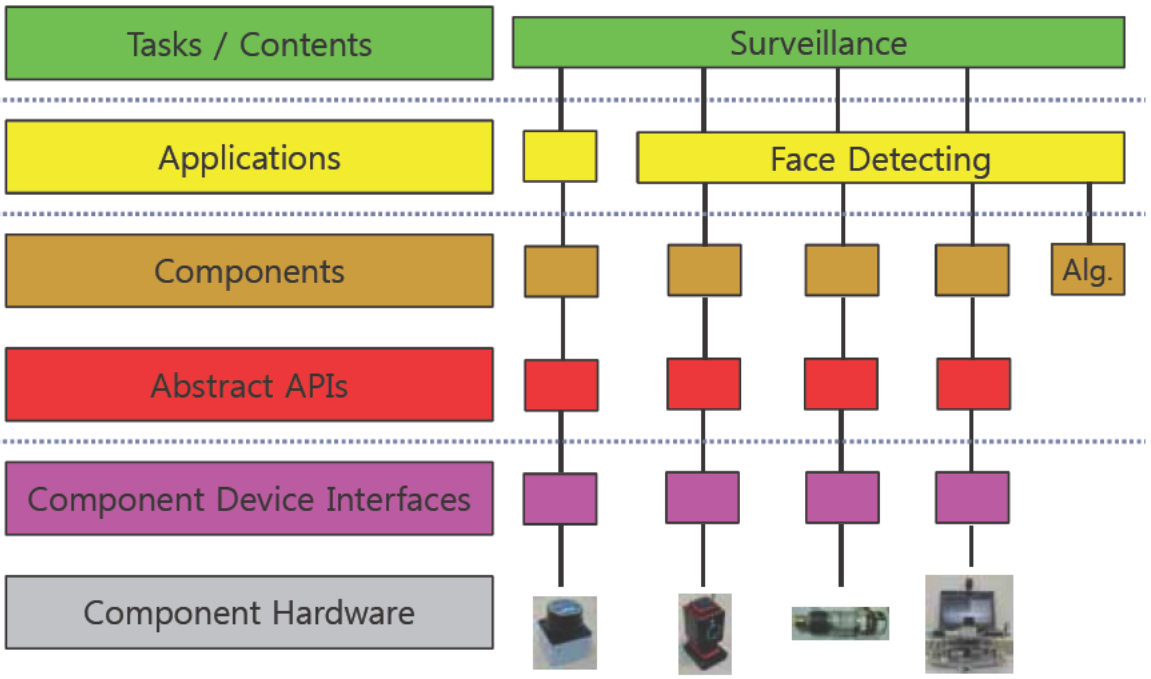
Vision Processing Unit	Mobile Platform
Pan/Tilt Camera Control Component	Navigation Component
Face Recognition Component	Ultrasonic sensor component
Camera Component	Wheel Control Component
	

Conformance Testing Method for Robot Software Components

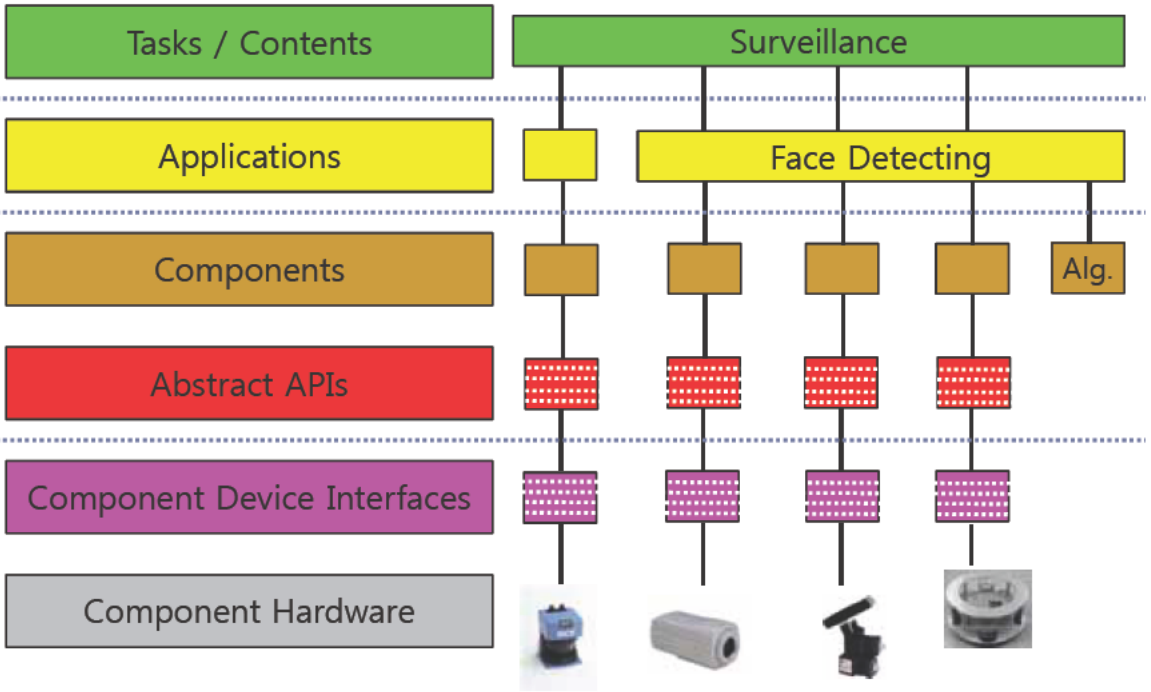
Kangwon National University Robot S/W Research Center
Hong Seong Park, and Mi-Sook Kim 2010.06.21

Contents

Robot Software Component



Robot Software Component



Testing Component

- If the components are
  - Written in abnormal ways or
  - Differ from defined component specifications,
  - It is difficult to achieve a normal robot application
- Verification is required to check
  - If the robot software component follows the standard

Objectives

- Defining conformance testing items for Robot SW Components
  - Structural conformance items
  - Dynamic conformance items
- Standards for defining evaluation systems and procedural documents of the test
  - Constituents of the evaluation system for testing Robot SW Components
  - Defining procedures for testing items
  - Format for testing results

Testing items

Test Classification	Test details
Structural Conformance	▸Verification whether the component profile follows the standard or not ▸Verification of consistency between component profile and source code ▸Verification of essential item implementation within the source code
Dynamic Conformance	▸Verification of basic operation of the component ▸Verification of state transition of the component ▸Verification of the interface of the component

Structural Conformance Test Procedure

	Test procedures	Note
1)	Test target materials, robot component standard document and structural conformance test tool are installed at specified environment	
2)	Checks if necessary information is prepared for standard conformance test. 2.1) Check if there is a component library file. 2.2) Check if there is a component profile. 2.3) Check if there is a source code.	
3)	Check the essential items of component related profile. 3.1) Check if all essential items of component profile are stated. (Refer to the component standard) 3.2) For the case that service port profile exists, check if all essential items of profile are described. (Refer to the component standard) 3.3) For the case that data port profile exists, check if all essential items of profile are described. (Refer to the component standard)	
4)	Check on the limiting cardinality for each item of component related profile. 4.1) Check if each item described at component profile follows the limiting cardinality standard. (Refer to the component standard) 4.2) In case of service port profile existence, check if each item described at profile is following the limiting cardinality standard. (Refer to the component standard.) 4.3) In case of data port profile existence, check if each item described at profile is following the limiting cardinality standard. (Refer to the component standard)	
5)	Check the consistency between component items for the described port standard. 5.1) Check related files to see they satisfy the consistency test conditions between component items. (Refer to the component standard.)	

Dynamic Conformance Test Procedure

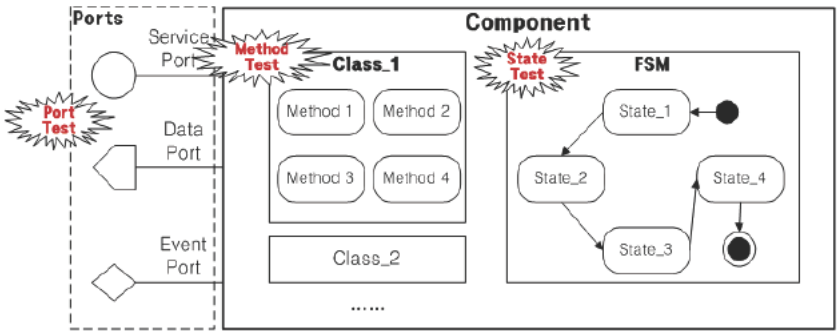
	Test procedures	Note
1)	Prepare for the dynamic conformance test. 1.1) Prepare the port standard profile, dynamic conformance test tool and component executing middleware. 1.2) Create a test component to test the target component.	
2)	Check if the target component is executed using the component executing middleware.	
3)	Check the compatibility with component executing middleware. 3.1) Check if onInitialize() callback function is called at the time of the initialization of the target component. 3.2) Check if onStart() callback function is called at the time of the start of the target component. 3.3) Check if onExecute() callback function is called according to the period specified on component profile in case that the target component is a periodic component. 3.4) Check if onPeriodChanged() callback function is called when the period is changed in case that the target component is a periodic component. 3.5) Check if onExecute() callback function is called only once in case the target component is a non-periodic component.. 3.6) Check if onStop() callback function is called when the target component is stopped during the process. 3.7) Check if onStart() callback function is called when the target component is restarted after it's been stopped. 3.8) Check if onError() callback function is called when error occurred during the target component process. 3.9) Check if onRecover() callback function is called when it is recovered from the error after the error has occurred. 3.10) Check if onReset(), onInitialize(), onStart() callback functions are called continuously at the time of reset of target component. 3.11) Check if onDestroy() callback function is called at the time of target component termination.	
4)	Test on the basic state transition of the component. 4.1) Execute the target component using the component executing engine. 4.2) By applying the state testing value, check if the state of target component is transit according to the component standard. (Refer to the component standard)	
5)	Verify on the port standard.	

Test Documents

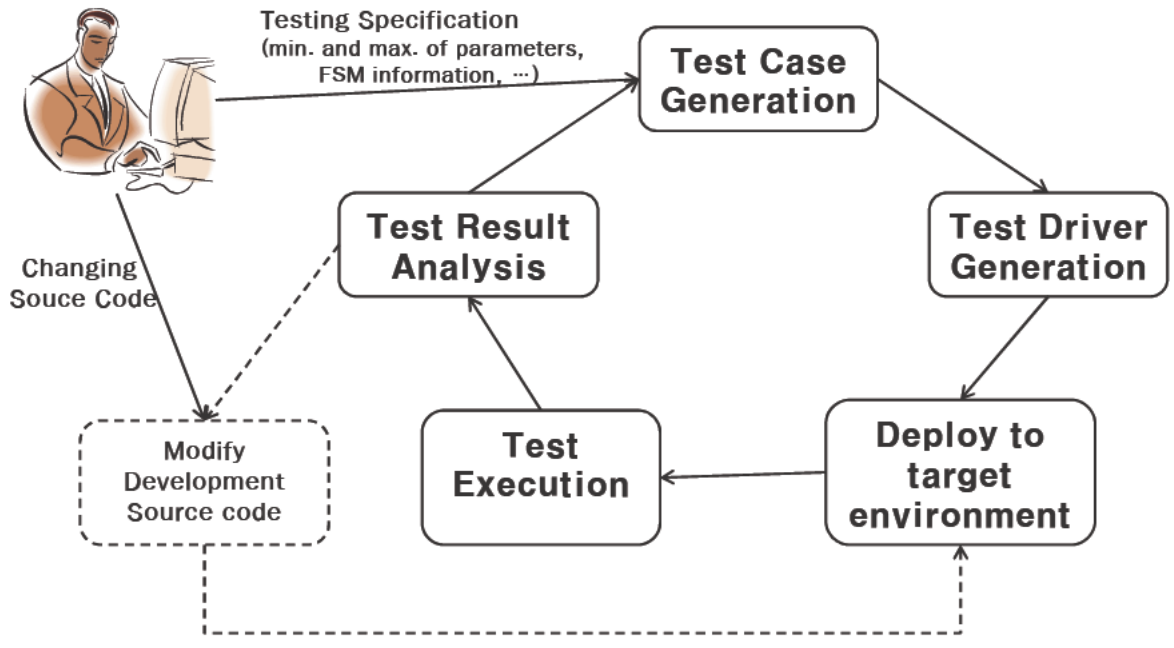
Test documents	Details
Test request documents	▸Application form to request the component conformance test ▸Information on person who requests and component are given. ▸Refer to the Appendix Ⅱ for this test request document
Test procedure document	▸Test items and procedures for component conformance test ▸Refer to the part 6 for the test procedures
Test result document	▸Results of the component standard conformance test ▸Refer to the Appendix Ⅲ for the test result document form

Robot SW Component Test System

- Testing Target Language
  - C, C++ (planned for Java)
- Testing Methods
  - Method Testing (Unit Testing) : Method of Class Test
  - State Testing : SW State Transition Test
  - Interface Testing (Port Testing) : Software Interface Test

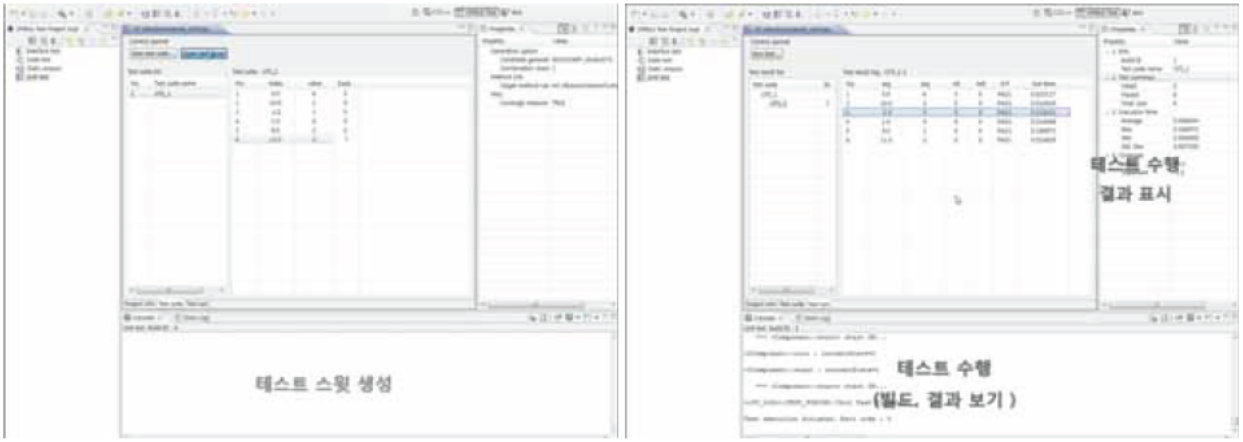


Robot SW Component Test System Process



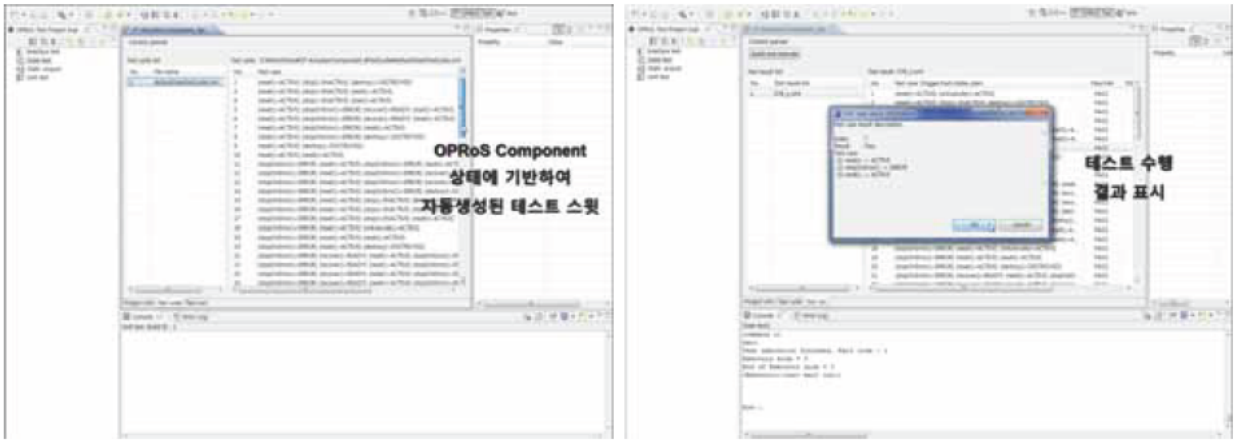
Eclipse based Testing Tool – Unit Test

- Test Component: Ultrasonic Sensor Component
  - Target Method:
    - getValue(int index, std::valarry<double>* value)
  - Testcase Generation Method:
    - BOUNDARY_ANALISYS / EQUIV_PARTITIONING



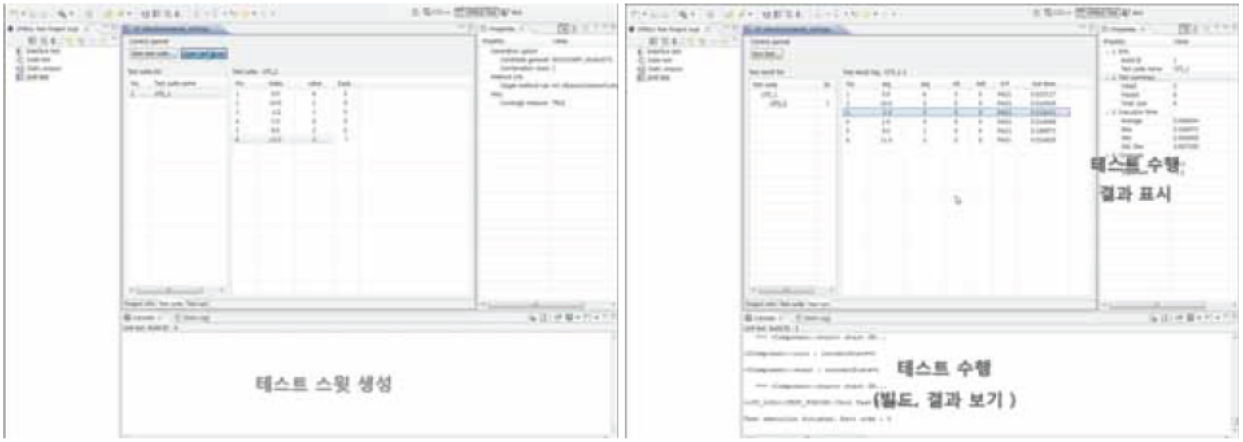
Eclipse based Testing Tool – State Test

- Test Component: Actuator Component
  - Required Resource:
    - Profile: ActuatorComp.xml
    - Binary file: ActuatorComp.dll
    - API: TestStubActuator.dll



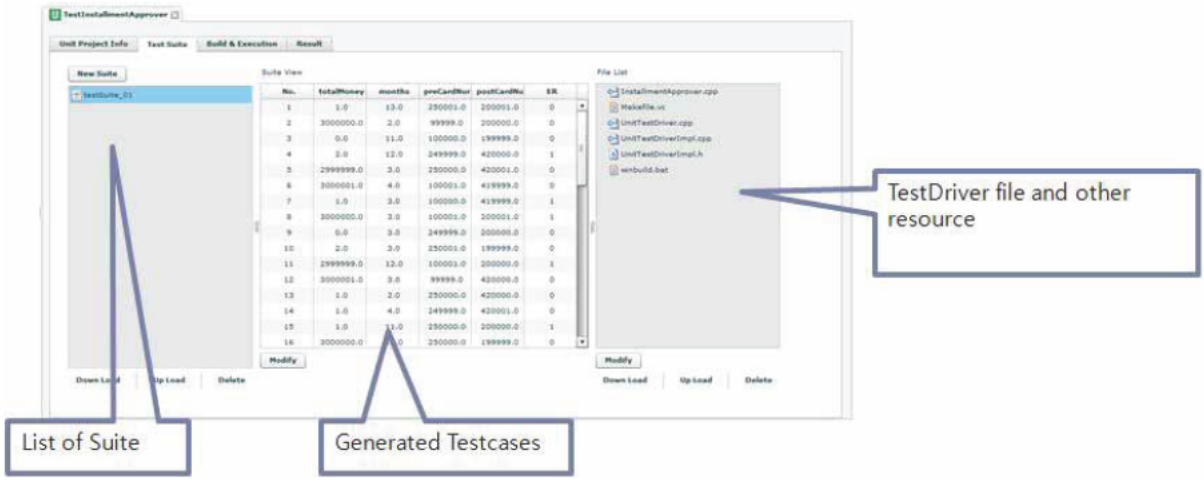
Eclipse based Testing Tool – Interface Test

- Test Component: Ultrasonic Sensor Component
  - Target Method:
    - getValue(int index, std::valarry<double>* value)
  - Testcase Generation Method:
    - BOUNDARY_ANALISYS / EQUIV_PARTITIONING



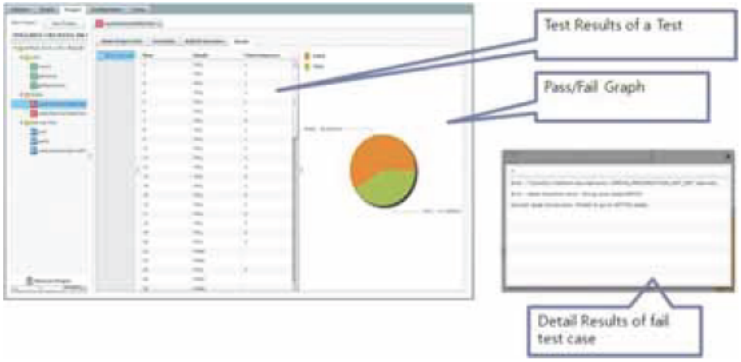
Web-based Robot Component Test Tool

- Executing the test and checking its result over the Internet anytime



Web-based Robot Component Test Tool

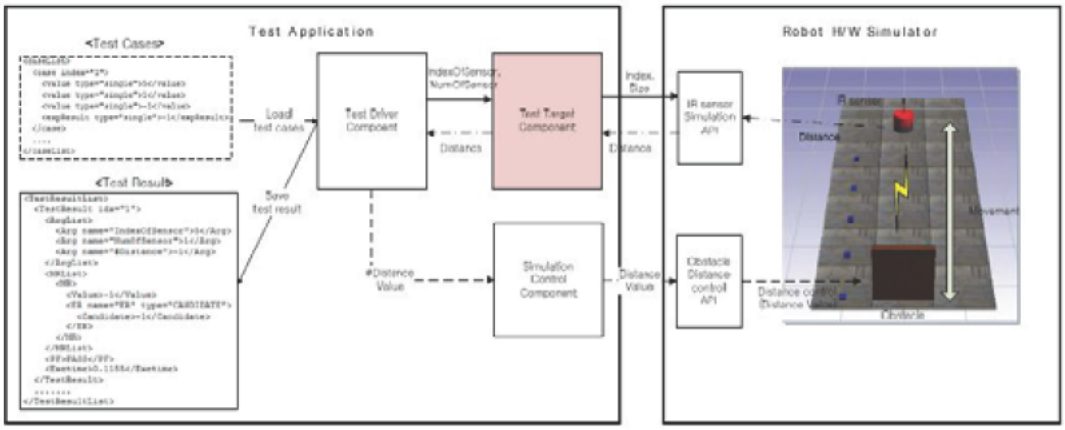
- Unit Test and Interface Test Results



- State Test Result

Simulator-based Robot Component Test

- HW related Component can have its extreme values from the virtual environment for testing purposes



Component Conformance Testing System

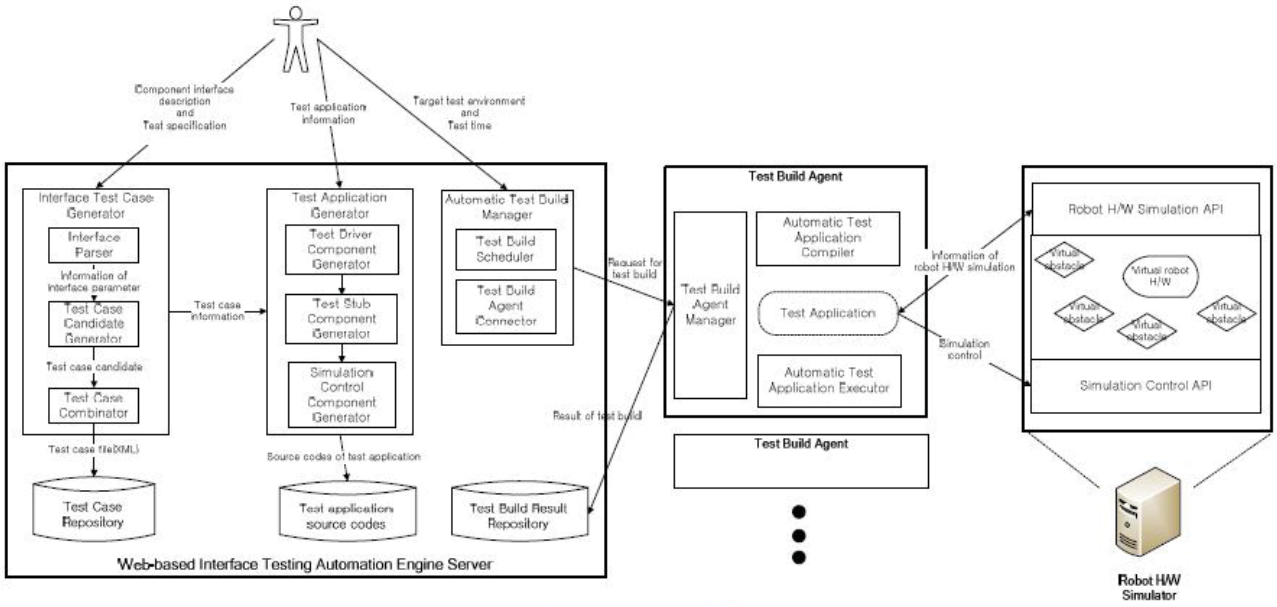


Fig. 1 The SITAT Architecture

Conclusion

- Verification is required to check
  - If the robot software component follow the standard
- Testing categories can be structural conformance and dynamic conformance
- Testing procedures for each conformance tests
- Documents for testing request, procedure, and result are suggested
- OPRoS Component Testing System:
  - Unit testing
  - State testing
  - Interface testing – simulator based component testing

Thank you

Structural Conformance Testing Items

Test items	Details
Check inclusion of essential items	▸Check on essential descriptive items of component profile ▸Refer to component standard for essential descriptive items
Check limiting cardinality for each item	▸Check limiting cardinality for each item of component profile ▸Refer to component standard for limiting cardinality for each item
Check consistency between components	▸Check consistency between profiles or profile and source code
Checking essential items for source code implementation	▸Check essential implementation Class and Method

Dynamic Conformance Testing items

Test items	Details
Compatibility test between component execution middleware	▸Operation checked by calling the callback function on the execution engine ▸Refer to component standard for component basic callback function
State transition	▸Check basic state transition of component ▸Refer to component standard for component state transition diagram ▸Refer to component standard for state transition test value
Port standard	▸Check port standard described in component profile ▸Refer to Appendix I for port standard test value
Exception process	▸Check process related to abnormal standard or situation

TOSHIBA
Leading Innovation

Robotics Technology Applied to Great East Japan Earthquake

Miwako Doi
TOSHIBA Corp.

OMG Robotics DTF
June 21, 2011.

The 5th emergency recommendation regarding the response to the Great East Japan Earthquake

Science Council of Japan, Great East Japan Earthquake Task Force

1.Regarding the handling of the accident of the Fukushima Daiichi Nuclear Power Plant

(1) Operation period: from immediately after accident until the completion of a cold shutdown of the damaged reactors

Actions: A joint team shall be organized immediately comprised of the electric power company , fire department, Japan Self-Defense Forces, the ministries and agencies concerned, comprehensive disaster prevention teams of robot specialists, nuclear power and radiation specialists, and a collaborative network (see Note 1) for support comprised of domestic and overseas universities, research institutions, and com. anies involved in robots ,see Note 2) under the direction of the central government and with the cooperation of robot-related academic societies and the Science Council of Japan.

Note 1: A hyper-academic organization, "Robotics Task Force to Deal with Disaster"has already been organized and has started work. Reference: http://www.rsj.or.jp/shinsai/RoboticsTF_1.pdf

Quotation from http://www.scj.go.jp/en/

The 5th emergency recommendation regarding the response to the Great East Japan Earthquake

(2) Operation period: from the completion of a cold shutdown of the damaged reactors –the completion of the decommissioning

Actions: Making the above system into a specialized area suited for the handling for longer periods, promoting systematically the development of a new robot capable of medium to long-term handling in addition to the immediate work onsite, planned development of a new operation system etc.

(3) Operation period: until the completion of the decontamination of the terrestrial areas

Actions: In addition to the plans as described above (2), the central government shall integrate and utilize various methods suggested from within and outside Japan and monitor continuously with patrolling autonomous mobile robots, and aim to realize a complete automation of certain aspects of the decontamination work, by autonomous working robots working in coordination with the monitoring.

The 5th emergency recommendation regarding the response to the Great East Japan Earthquake

2. Operation of the nuclear power plant and other high risk facilities and the fundamental reexamination and improvement of the safety system

Measures: For the strengthening of the safety functions along with immediate response to unexpected situations, a system design that presupposes the use of robots in various parts of the plant shall be carried out. Through the introduction of general purpose and autonomous robots, the aim is to achieve responsiveness to unexpected situations and safety maintenance after system function damage. The central government shall establish a system that facilitates close cooperation between the site, related companies, nuclear power plant specialists, and robot scientists, and that conducts operations, tests, and research and development continuously for a long time. The societies concerned and the Science Council of Japan will circulate relevant information while promoting cross-field cooperation.

The 5th emergency recommendation regarding the response to the Great East Japan Earthquake

3. Re-inspection and improvement of the disaster management support technology

Actions: Robot scientists shall coo¸erate with disaster ¸revention related institutions, analyze the disaster situation and restoration measures, provide effective technologies suited to the current situation, analyze technical and systematic issues to be addressed for effective robot use in the future, and provide solutions. Especially, the central government shall maintain and improve disaster management support robots and establish organizations or systems that constitutively conduct training in robot operation. The Science Council of Japan will support the necessary collaboration between universities, research institutions, local governments, and business enterprises as well as conducting fair and objective information disclosure to the Japanese people and to international communities.

Robot Uses in the Fukushima

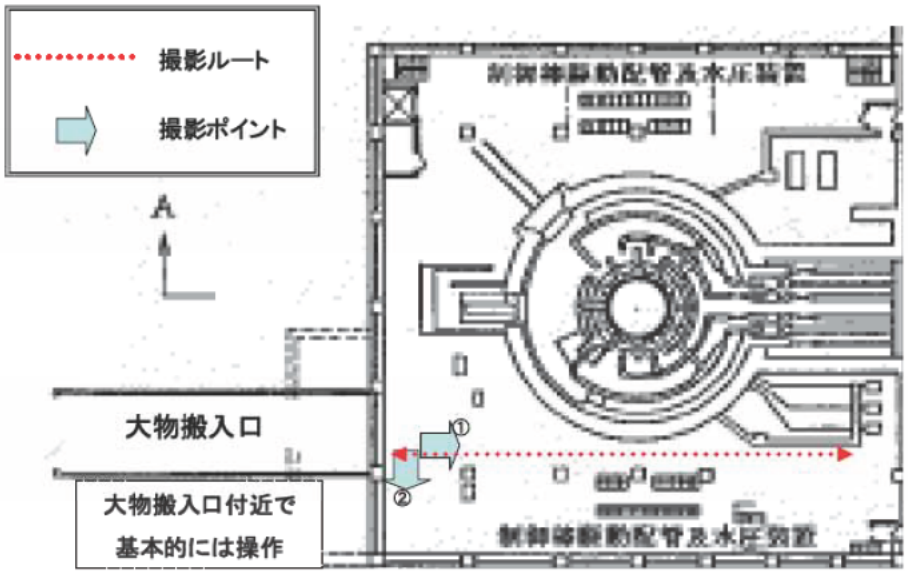
Maker	Robot		April	May	Usage
iRobot	Packbot		① In operation April 1, shooting rubbles outdoors ② April 17–18, monitoring daiichi, daini, and daisan April 22, monitoring daiich		Indoor and outdoor monitoring
QinetiQ (DOE)	Talon		① 訓練・調整 @小名浜 4/14：産総研（つくば）にて受入 4/22：小名浜にて操作訓練を実施 4/27：ロボット操作車との組合せ 4/E：サイト導入に向けて最終調整	Not in operation	Outdoor surveying with map generation
It-Chiba, Tosho ku U., et.al.	Quince		訓練・調整 @千葉 操作性能確認、各種調整を実施（4/11,15,21,26）	Operation at the end of June http://www.youtube.com/watch?v=oNyBFAdW_6E	Indoor and outdoor monitoring

Quotation from http://www.tepco.co.jp/nu/fukushima-np/f1-roadmap/images/11042801a-j.pdf

Radiation Dose Monitoring in the Fukushima Daiichi

5月20日　1号機原子炉建屋内γカメラ線量測定箇所

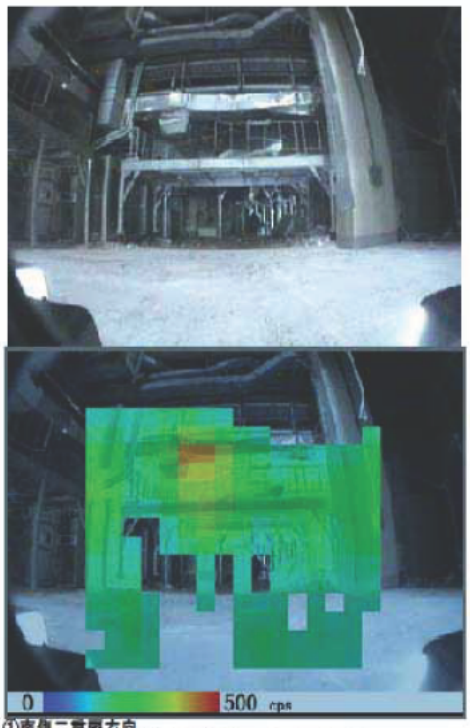
Quotation from http://www.tepco.co.jp/nu/fukushima-np/images/handouts_110522_01-j.pdf



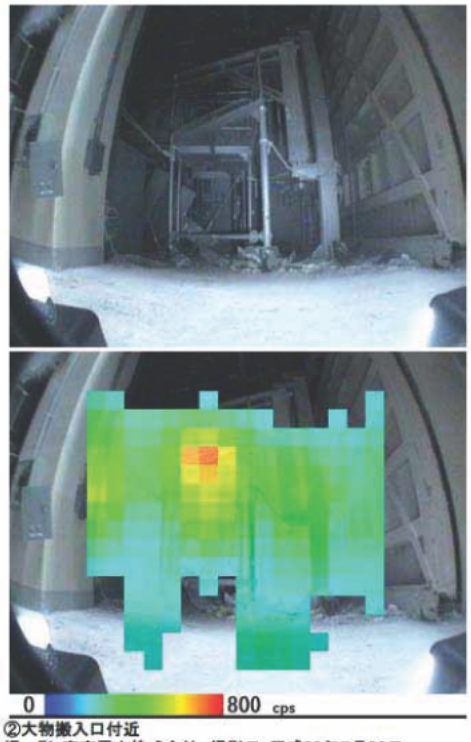
1号機原子炉建屋1階平面図

TOSHIBA Crawler Robot No.1 with a Gamma Camera

Gamma Camera Images



①南側二重扉方向
撮　影:東京電力株式会社　撮影日:平成23年5月20日



②大物搬入口付近
撮　影:東京電力株式会社　撮影日:平成23年5月20日

TOSHIBA Crawler Robot No.2 with a Manipulator

TOSHIBA

Leading Innovation

Infrastructure WG Progress Report

(Salt Lake City meeting)

Noriaki Ando (AIST)

robotics/2011-06-12

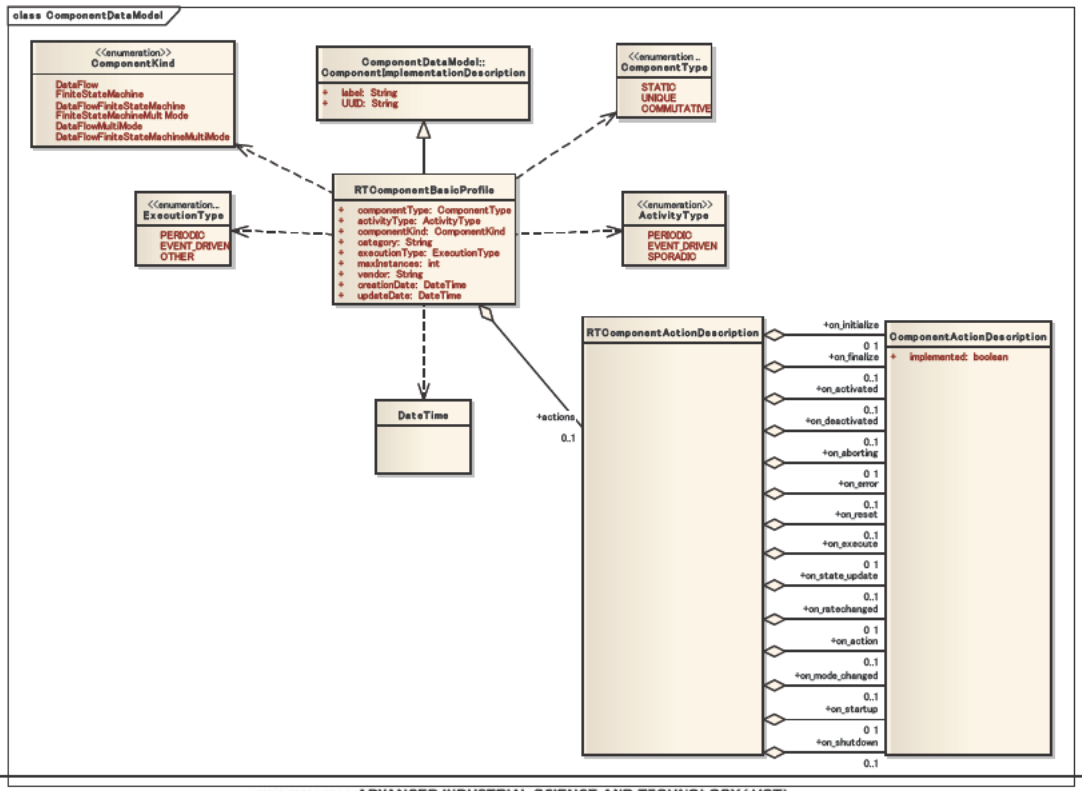
Topics of This Meeting

- Merging homework from each member
- Reviewing homework
  - Make clear the relation between existing DEPL spec and our specification.
  - Some diagrams are created from the discussion
- Discussing rest of specification
  - Repository/Directory Manager
  - Details in supervisor FSM

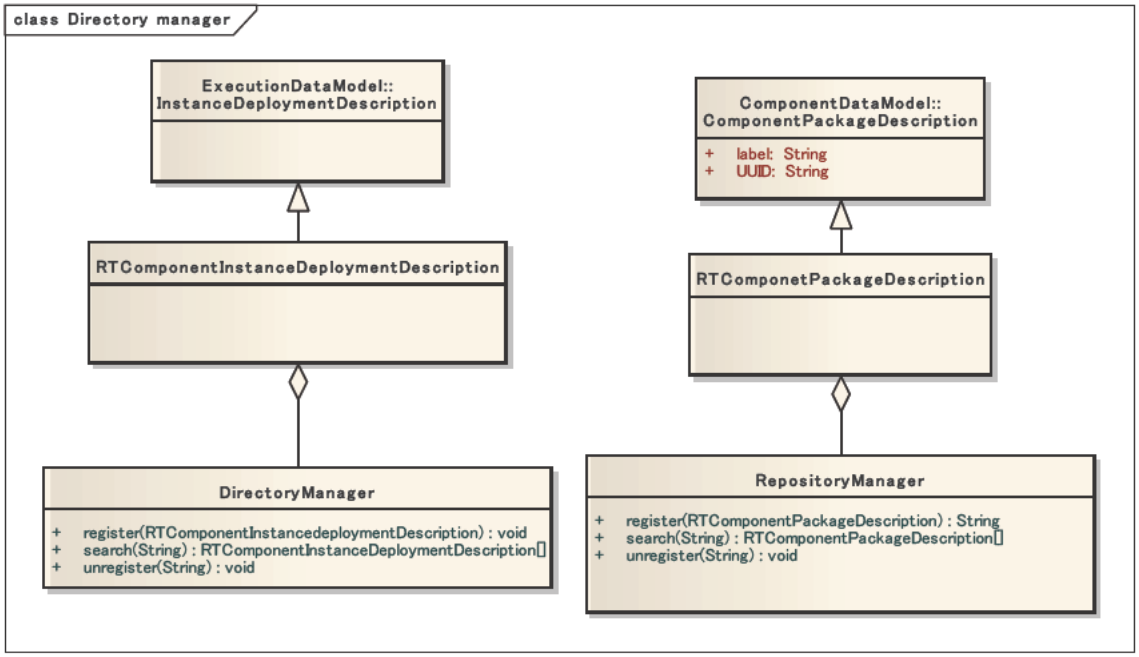
Platform Independent Model

1. Segmentation of the Model
2. Model Diagram Conventions
3. Component Data Model (AIST)
   1. RTCBasicProfile
   2. RTCPortDescription (ComponentPortDescription)
   3. RTCConnectionDescription (AssemblyConnectionDescription)
   4. SupervisorFSMDescription
4. Component Management Model (ETRI)
   1. **RepositoryManager**
5. Target Data Model
6. Execution Data Model (AIST)
   1. SupervisorFSMInstance
7. Execution Management Model
   1. EventManagement (ETRI)
   2. DirectoryManager (ETRI)
   3. ApplicationSupervisor (AIST)
8. Common Elements (AIST)
   1. Constraint
   2. Query Language for Directory Manager or OCL
9. Exceptions
10. Relations to Other Standards

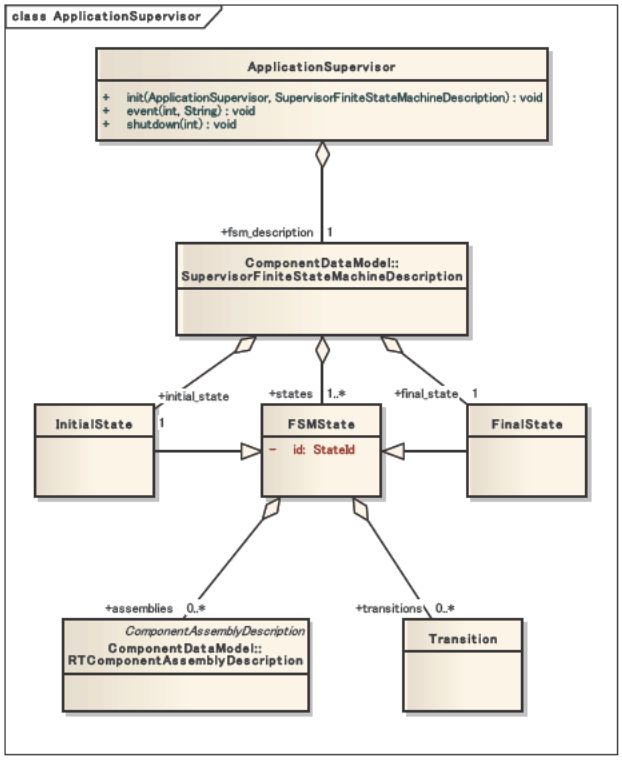
RTComponentBasicProfile



DirectoryManager/RepositoryManager



ApplicationSupervisor



Next

- Continues merging process
- Complete a draft merged specification before the next (or next next) meeting.

Robotic Functional Service WG
WG Report

WG Co-Chairs: SuYoung Chi, Miki Sato, Toshio Hori

2011/06/21

robotics/2011-06-13

WG activities before Salt Lake City meeting

- Revised submission to the RoIS Framework RFP was submitted last month (4-weeks rule)
- AB comments from Andrew Watson
  - No technical comments
  - Procedural comment on the copyright

WG activities during this meeting (1)

- Monday – morning
  - Presentation of the Revised Submission at Robotics-DTF
    - Voting: passed
    - Revised Submission goes to AB (in the afternoon)
- Monday – afternoon
  - Presentation of the Revised Submission at the Architecture Board
    - Before the presentation, Hori and Sato prepared LOIs to permit reproduction of this submission (to deal with the comment from Andrew)
    - Comments from AB members
      - The submission must mention how the XMI file was created.
      - The submission must mention how the XSD (XML Schema) file was created, too. Namespace of XML Schema should be modified.
      - These are homework to FTF.
    - Voting: passed (with conditions, see above)

robotics/2011-06-13

WG activities during this meeting (2)

- Tuesday – morning
  - Drafting a charter of the RoIS FTF
    - Members:
      - Co-Chairs: Suyoung Chi, Toshio Hori and Miki Sato
      - Other Members: Miwako Doi, ChulJong Hwang, Takeshi Sakamoto and Takashi Tsubouchi (plus Co-Chairs)
    - Deadlines: See next slide
  - Demonstration of RoIS Framework developed by Sato

Schedule after this meeting (tentative)

- FTF starts its activity, if the proposed charter is approved

- Deadlines (indicated in the charter):
  - Beta Specification Publication: 31st July, 2011
  - Comments Due: 20th February, 2012
  - Report Due Date: 21st May, 2012
  - Report Deadline: 29th June, 2012

Robotics DTF Contact Report Standardization of RTC-CANopen

2011/6/21

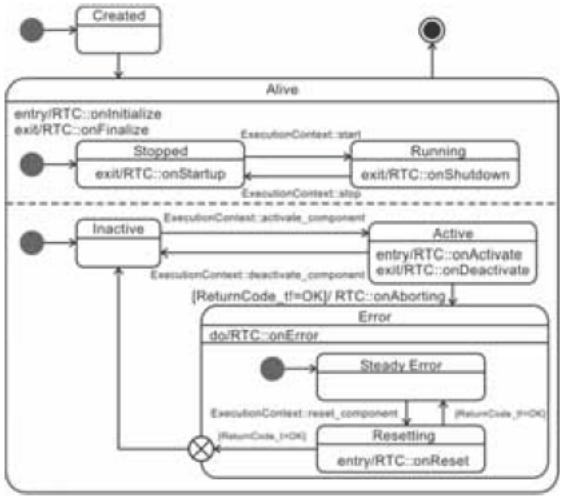
Makoto Mizukawa
Dean, College of Engineering
Professor, Human-Robot-Interaction Lab.
Department of Electrical Engineering,
College of Engineering
Shibaura Institute of Technology

3-7-5, Toyosu, Koto-ku, Tokyo 135-8548

Acknowledgement
This work is supported by NEDO (New Energy and Industrial Technology Development Organization, Japan) project "Intelligent RT Software Project".

RTC: Robotic Technology Component Specification

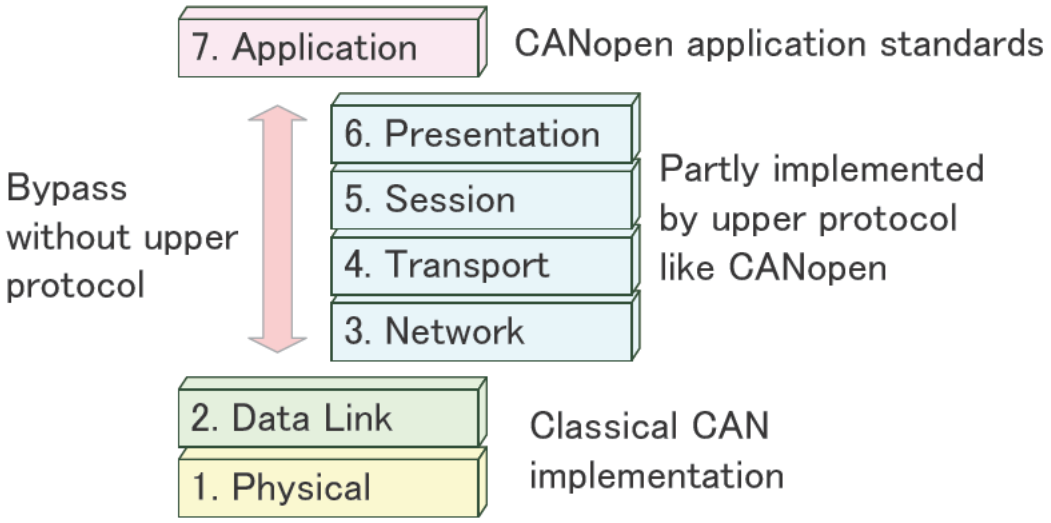
- Standardized in Object Management Group(OMG).
  - Ver.1.0 was opened to the public in April, 2008.
  - Specify a basic part of RT-Component.
    - RT-Component Interfaces
    - Basic state transition



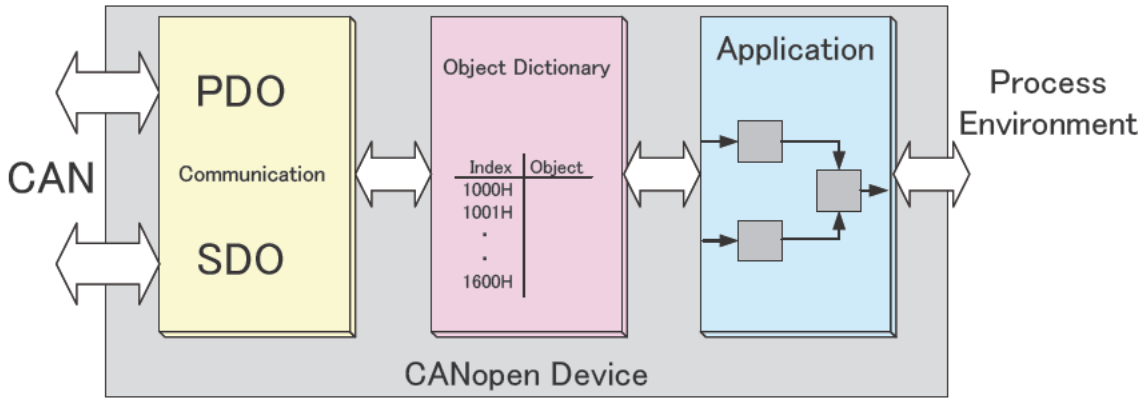

Robotic Technology Component Specification

CANopen

- Standardized upper protocol based on Controller Area Network (CAN) that suits embedded network for machine control.
- CANopen decreases message overhead.



CANopen



- Application and communication are independent of each other by Object Dictionary that is a kind of data table.
- Some data defined in the profile specialized device and application.

SDO: Service Data Object=2 data frames+ID
PDO: Process Data Object

RTC-CANopen

RT-Middleware

- RT-Middleware can improve software reusability.
- It's difficult to use RT-Component on embedded MPU, because RT-Component needs CORBA environment.

CANopen

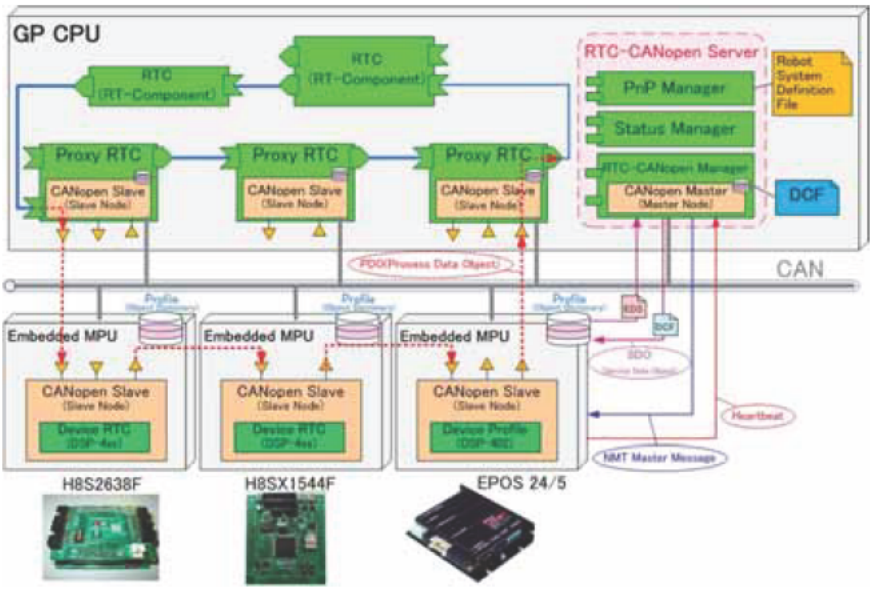
- CAN-based upper protocol for embedded network
- CANopen can improve hardware reusability, because this protocol is standardized.

Development RT-Middleware for embedded system

RTC-CANopen

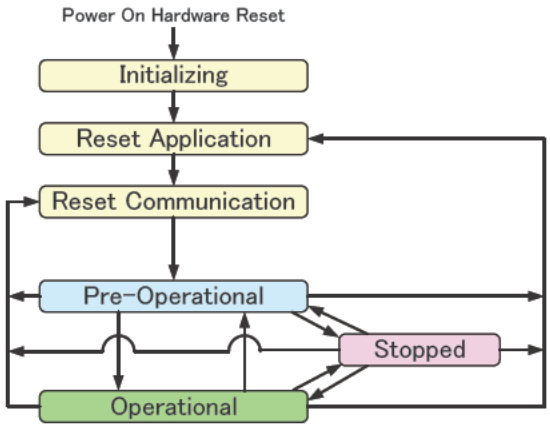
What is RTC-CANopen？

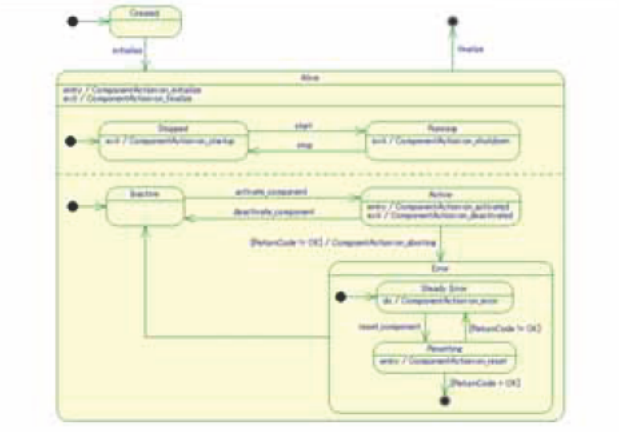
- Platform for distributed control robotic development
  - Middleware that runs RTC on CANopen
  - Each embedded devices (components) communicates using CANopen.



- RTC-CANopen is middleware to make CANopen devices cooperate with RT-Component running on a general-purpose PC.

Proposal example –State Machine Mapping-





CANopen		RT-Component	Process
Initializing		onInitialize	Initialization. Invoked only once when the RTC starts.
Pre-Operational		onActivated	Invoked only once when the INACTIVE RTC is activated.
Operational		onExecute	Invoked periodically while in the ACTIVE state.
Stopped		onDeactivated	Invoked only once when the ACTIVE RTC is deactivated.
–		onAborting	Invoked only once before the RTC goes to the ERROR state.
Reset Application (Reset Communication)	⇔	onReset	Invoked when the RTC recovers from the ERROR state to the INACTIVE state by RESET.
–		onError	Invoked periodically while in the ERROR state.
–		onFinailize	Invoked only once when the RTC finalizes.

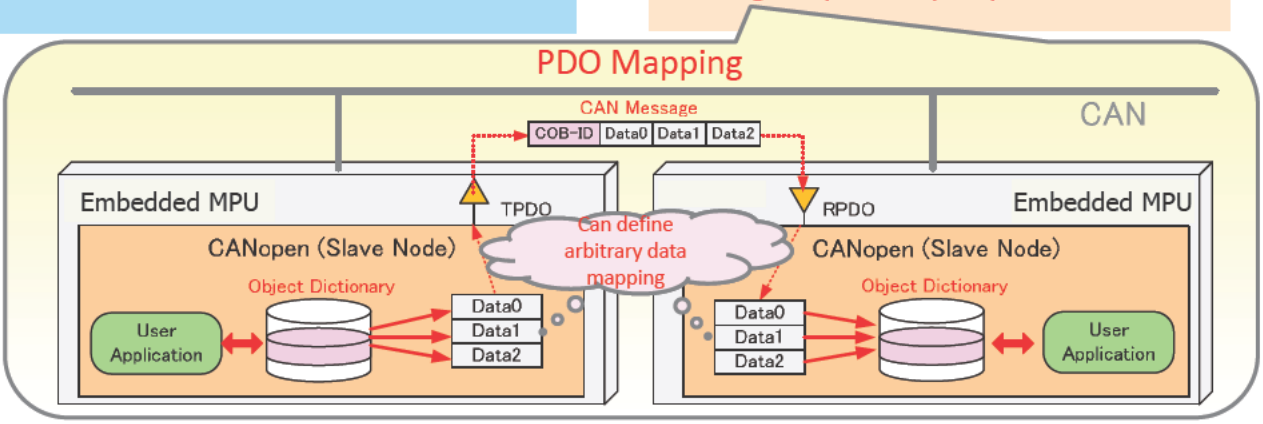
CANopen and RT-Middleware

RT-Middleware (RTC)

- Modularized with Software level
- Specified State transitions.
- Own input/output ports
- Trigger of data transmission and reception

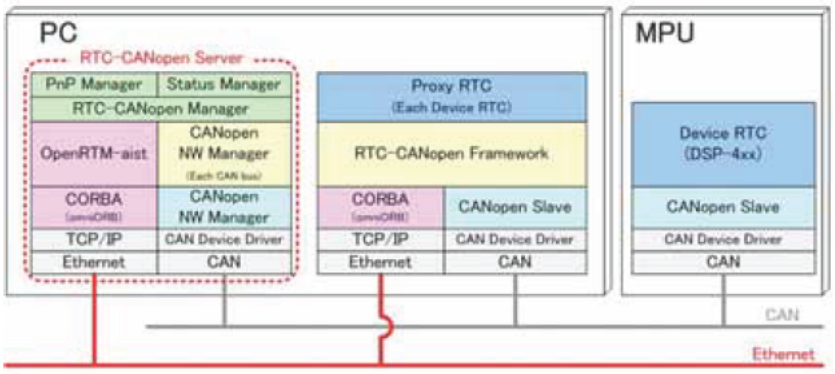
CANopen (Slave Node)

- Modularized with Hardware level
- Specified State transitions.
- Own input/output ports
- Trigger of data transmission and reception
- Changeable mapping definition about data to transmit and receive through input/output ports



Features of RTC-CANopen

- Lightweight – Implement light RT middleware providing only necessary functions for embedded devices.
- Robustness - Improve system robustness introducing monitoring methods for operations on native buses.
- Real-time - Implement fast and reliable communication between components with the aid of adventages of native buses.
- Flexibility - Possible to change system configurations flexibly, switching parts or changing combinations.
- Reusability - Improve reusability of each element, since both hardware and software can be dealt as components.

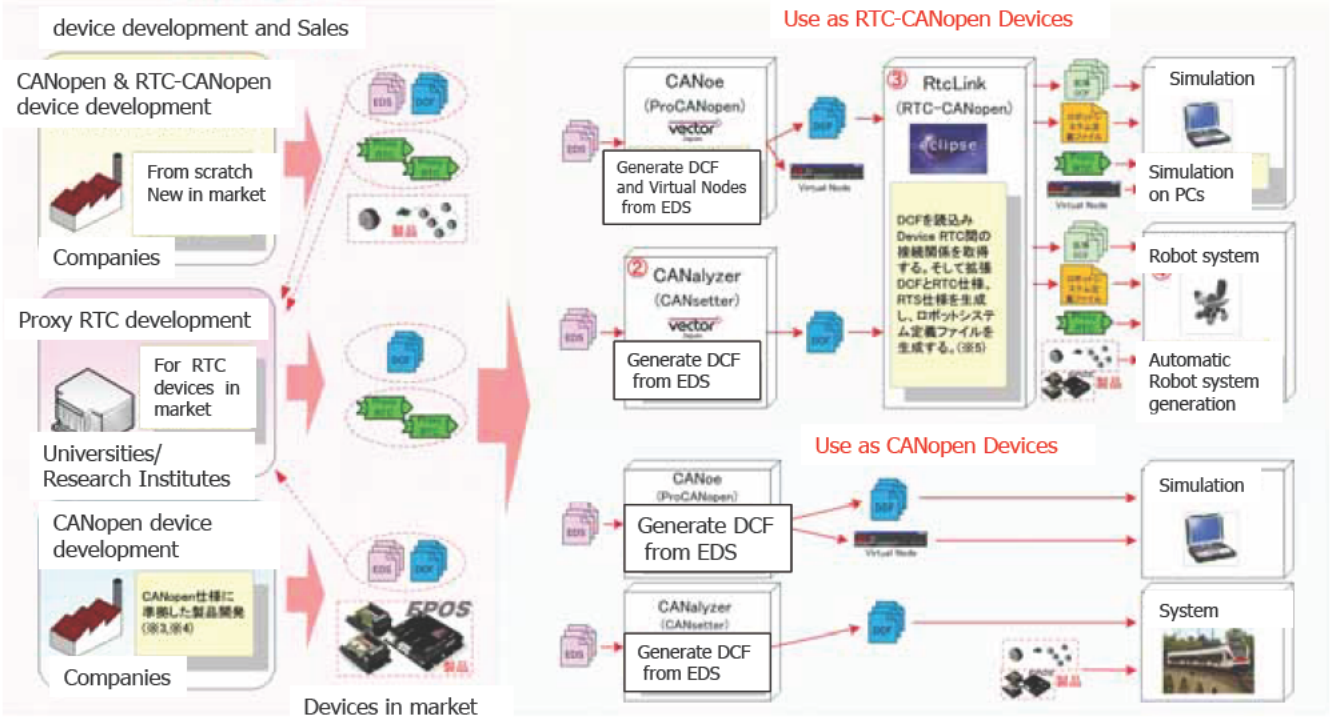


Advantage of RTC-CANopen

- Available to convert existing hardware for robots.
  - Available to use existing CANopen compliant devices in robot systems.
  - Reduction of a hardware cost.

- Available to run in many embedded MPU
  - All MPU that can implement CANopen.
    - M16C, R8C, H8S, SH series, V850, PIC18Fxx8, and so on.
- Improvement of reliability of communication between components
  - High reliability and high speed communication can be realized.

System Development using RTC-CANopen



Specification Contents (Proposal)

DS-4XX : Device profile for RTC-CANopen

Part1 : General definitions

Future work plan

- Part2 : Device profile for Device and motion control device (RTC-CANopen)
- Part3 : Device profile for battery modules (RTC-CANopen)
- Part4 : Device profile for battery chargers (RTC-CANopen)
- PartX : Device profile for ····· (RTC-CANopen)

-Proposal profiles don't replace existing profiles.
-Proposal profiles take the part of existing profiles as much as possible. However they will add/modify the function for robot systems.

RTC-CANopen Standardization in CiA

- Negotiation with CiA (CAN in Automation) started in 2008
- RTC-CANopen article is published in CAN (2010.6)
- CFE(Call for Expert) of Service Robot issued from CiA (2010.7.14)
- Kick off meeting of CiA in Tokyo (2010.11.15)
- Service Robot SIG started (2011.3.5)
  - Chair Makoto Mizukawa (Shibaura Inst. Tech.)
- CiA 460 WD (2011.6.21)

CAN Newsletter(June 2010)

Scope

This document specifies the CANopen interface of a service robot controller device, which is compliant to the Robotic technology component (RTC) specification. The profile specifies the NMT master application and its object dictionary with NMT master parameter as well as CANopen device proxies representing the connected CANopen devices in the service robot controller device

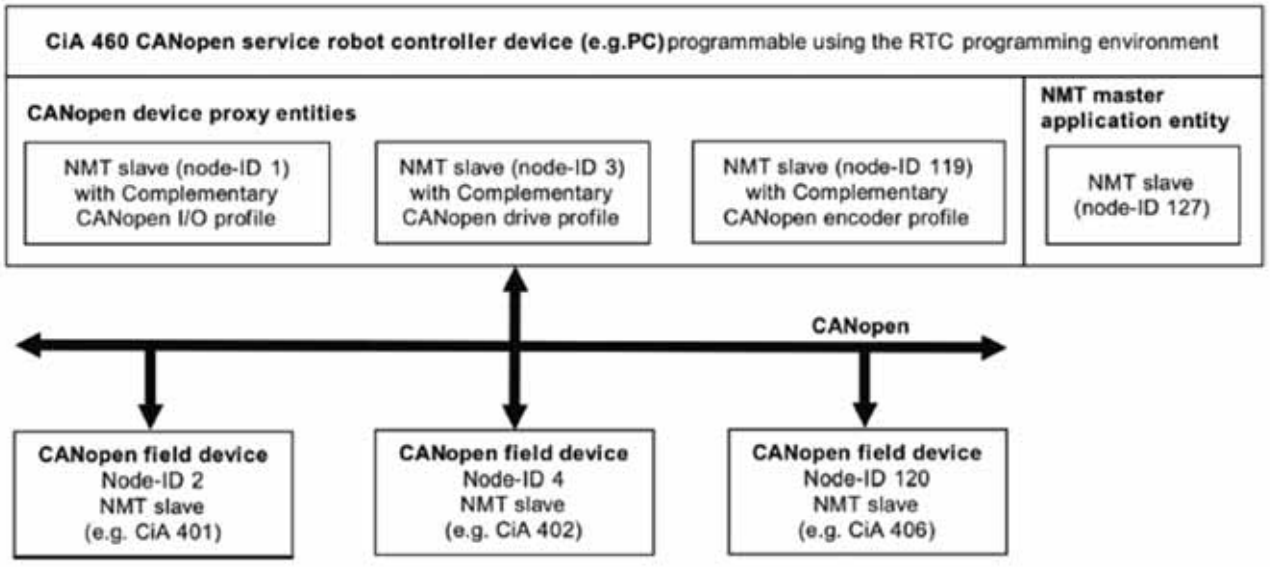
References

/CiA301/ CiA 301, CANopen application layer and communication profile
/CiA302/ CiA 302 (all parts), Additional CANopen application layer functions
/CiA303-2/ CiA 303-2, CANopen additional specifications
– Part 2: Representation of SI units and prefix

/CiA401/ CiA 401, CANopen device profile for generic I/O modules
/CiA402/ CiA 402 (all parts), CANopen device profile for drives and motion controllers

/OMG-RTC/ OMG-RTC (formal 4-4-2008), Robotic technology component specification
/OMG-SDO/ OMG-SDO (formal 1-11-2004), Platform independent model (PIM) and platform specific model (PSM) for super distributed objects (SDO)

CANopen sub-network for service robot systems



Contact Report: ISO/TC184/SC2/WG1

June 21st, 2011

Salt Lake City ,UT , USA

Little American Hotel

ISO/CD9787

Coordinate systems and motion nomenclatures
(Revision of ISO 9787:1999)

- ISO 9787:1999
  (for Industrial Robots only)
  right hand coordinate system
  tool, industrial robot coordinate system

- CD 9787
  (add Personal Service Robots)

  mobile platform coordinate system
  task, object, camera coordinate system

Voting terminate on 2011-05-24

ISO/DIS 8373

Robots and Robotic Devices-- Vocabulary (Revision of ISO 8373:1994)

- ISO 8373:1994
  (for Industrial Robots only)

- DIS 8373
  (add Personal Service Robots)

Voting terminate on 2011-05-09

Proposed Charter for Robotic Interaction Service (RoIS) Framework 1.0 FTF

TC Meeting Date: 24 June, 2011
Presenter: Suyoung Chi
Group email: rois-ftf@omg.org
WIP page (URL):http://www.omg.org/techprocess/meetings/schedule/RoiS_1.0_FTF.html

- **Adopted Specification:**
  - robotics/2011-05-01 (Specification document)
  - robotics/2011-05-02 (C++ Header)
  - robotics/2011-05-03 (CORBA IDL)
  - robotics/2011-05-04 (XMI)
  - robotics/2011-05-05 (Inventory)

- **Members:**
  - Suyoung Chi (Co-Chair), ETRI
  - Miwako Doi, The Univercity of Tokyo
  - Toshio Hori (Co-Chair), AIST
  - Chuljong Hwang, KAR
  - Takeshi Sakamoto, Shibaura Institute of Technology
  - Miki Sato (Co-Chair), JARA
  - Takashi Tsubouchi, Univercity of Tsukuba
  -

- **Deadlines:**
  - Beta Specification Publication: 31st July, 2011
  - Comments Due: 20th February, 2012
  - Report Due Date: 21st May, 2012
  - Report Deadline: 29th June, 2012

TOSHIBA
Leading Innovation

Contact Report : IEEE SA(Standardization Activities)

Miwako Doi
TOSHIBA Corp.

OMG Robotics DTF
June 21, 2011.

IEEE SA

- **2 study groups**
  Map data representing SG by Wonpil Yu (ETRI)
  Terminology (NIST)
- **Map data representing SG**
  - Type of PAR(Project Authorization Request)
  - 2D first, 3D later; for a quick start, we begin with 2d, you can do with 3D anytime
  - What is difference between IEEE and existing SDOs
  - Metric, topology together: definition of metric, topology, New map is hybrid topology basis, with metric elements
  - Target schedule: July 29 (prepare PAR)

Robotics-DTF Plenary Meeting
Opening Session

June 20th, 2011

Salt Lake City, UT, USA

Little America Hotel

Approval of Minutes

Meeting Quorum : 3

AIST, ETRI, JARA, KAR, Shibaura-IT, UEC, Univ. of Tsukuba,

Minutes taker(s):

- **Miki Sato**
- **Seung-Woog Jung**

Minutes review

Washington DC Meeting Summary

Robotics Plenary: (17 participants)

–2 Special Talks

- **"PROTEUS (Platform for Robotic modeling and Transformation for End-Users and Scientific communities) project", Rioux Laurent, Thales [robotics/2011-03-08]**
- **"Robotic component testing method", Mi-Sook Kim, Kangwon National University [robotics/2011-03-09]**

–2 WG Reports

- **Robotic Functional Service WG**: RoIS(Robot Interaction Service Framework) **[robotics/2011-03-11]**
- **Robotic Infrastructure WG:** DDC4RTC(Dynamic Deployment & Configuration for RTC **[robotics/2011-03-12]**

Roadmap for Robotics Activities

robotics/2011-06-19

Item	Status	Washington DC Mar-2011	Salt Lake City UT Jun-2011	Orlando FL Sep-2011	Santa Clara CA Dec-2011	Washington DC Mar-2012	Cambridge MA Jun-2012	POC / Comment
Flyer of Robotics-DTF [Publicity Sub-Committee]	Suspended							
Robot Interaction Service (RoIS) Framework RFP [Robotic Functional Services WG]	In Process		Revised Submittion & Voting				FTF Report	
Dynamic Deployment and Configuration for RTC (DDC4RTC) RFP [Robotic Infrastructure WG] in MARS	In Process		Pre-review		Revised Submittion & Voting			Sponsor: MARS
Robotic Map Services RFP [Robotic Functional Services WG]	Planned							IEEE R&A?
etc…	Future							
Robotics Information Day [Technology Showcase]	Future							
RoIS Finalization Task Force	Planned		Charter				FTF Report	
RTC Revision Task Force	In Process			Comments Due: ?	RTF Report			
RLS Revision Task Force	In Process		RTF Report					will go to ISO/TC211

Related Events

Chu-suk (Special Holidays in Korea)

Robotics-DTF Plenary Meeting
Wrap-up Session

June 21st, 2011

Arlington, VA, USA

Hyatt Regency Crystal City Hotel

Document Number

robotics/2011-05-01 Revised submission to the RoIS Framework RFP
robotics/2011-05-02 Revised submission for the RoIS Framework RFP - C++ PSM header file
robotics/2011-05-03 Revised submission for the RoIS Framework RFP - CORBA PSM IDL file
robotics/2011-05-04 Revised submission for the RoIS Framework RFP - XML PSM schema file
robotics/2011-05-05 Revised submission for the RoIS Framework RFP - XMI file
robotics/2011-05-06 Revised submission for the RoIS Framework RFP - inventory file

dtc/2011-05-08 Robotic Localization Service 1.1 RTF report
dtc/2011-05-09 Specification with change bars
dtc/2011-05-10 Specification without change bars
dtc/2011-05-12 C++ PSM
dtc/2011-05-13 XML schema
dtc/2011-05-14 XMI
dtc/2011-05-11 Inventory

Document Number (cont.)

robotics/2011-06-01 Final Agenda (Tetsuo Kotoku)

robotics/2011-06-02 Washington DC Meeting Minutes [approved] (Geoffrey Biggs and Seung-woog Jung)

robotics/2011-06-03 Draft: Revised submission ver.2 to the RoIS Framework RFP (Miki Sato)

robotics/2011-06-04 Draft: Revised submission ver.2 to the RoIS Framework RFP with Change bar (Miki Sato)

robotics/2011-06-05 Draft: Errata for ver.2 from 4weeks before original version (robotics/2011-05-01) (Miki Sato)

robotics/2011-06-06 Presentation: Revised Submission to Robotic Interaction Service (RoIS) Framework (Toshio Hori, Miki Sato, and Su-Young Chi)

robotics/2011-06-07 Event Management Model for DDC4RTC (Seung-Woog Jung)

robotics/2011-06-08 Repository & Directory Managers for DDC4RTC (Seung-Woog Jung)

robotics/2011-06-09 OPRoS: Open Platform for Robotic Services (Hong Seong Park)

robotics/2011-06-10 Conformance Testing Method for Robotic Software Components (Mi-sook Kim)

AIST

Document Number (cont.)

robotics/2011-06-11 Robotics Technology Applied to Great East Japan Earthquake (Miwako Doi)

robotics/2011-06-12 Infrastructure WG Progress Report (Noriaki Ando)

robotics/2011-06-13 Robotic Functional Services WG Report (Toshio Hori)

robotics/2011-06-14 Contact Report: Standardization of RTC-CANopen (Makoto Mizukawa)

robotics/2011-06-15 Contac Report: ISO/TC184/SC2/WG1 (Tetsuo Kotoku)

robotics/2011-06-16 Charter RoIS-FTF (Miki Sata)

robotics/2011-06-17 Contact Report: IEEE SA (Standardization Activities)

robotics/2011-06-18 Opening Presentation (Tetsuo Kotoku)

robotics/2011-06-19 Roadmap for Robotics Activities (Tetsuo Kotoku)

robotics/2011-06-20 Wrap-up Presentation (Tetsuo Kotoku)

robotics/2011-06-21 Kissimmee Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)

robotics/2011-06-22 Santa Clara Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)

robotics/2011-06-23 DTC Report Presentation (Su-Young Chi)

robotics/2011-06-24 Salt Lake City Meeting Minutes - DRAFT (Miki Sato and Seung-woog Jung)

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

Call for volunteer

- Robotics-DTF Co-Chair

  => Postpone voting one more meeting

Organization (from Mar. 24th, 2011)

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)
- Miki Sato (ATR, Japan)
- Toshio Hori (AIST, Japan)

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

Robotic Localization Services 1.1 WG
- Jaeyeong Lee (ETRI, Korea)
- Yeon-Ho Kim (Samsung, Korea)
- Shuichi Nishio (JARA/ATR, Japan)

Robotic Technology Component 1.1 WG
- Geoffrey Biggs (JARA/AIST, Japan)
- Noriaki Ando (AIST, Japan)
- Seung Woog Jung (ETRI)

AIST

Next Meeting Agenda

September 19-23 (Kissimmee, FL, USA)

Monday:

WG activity

Tuesday:

WG activity

AIST

Santa Clara Meeting Agenda

December 12-16 (Santa Clara, CA, USA)

Monday:

DDC4RTC revised submission review, vote-to-vote, voting (am)
WG activity (pm)

Tuesday:

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

Wednesday:

WG activity follow-up

Thursday:

WG activity follow-up [if necessary]

Plenary Attendee (17 participants)

- Andrew Watson (OMG)
- Chul-Jong Hwang (KAR)
- Geoffrey Biggs (AIST)
- Hong Seong Park (Kangwan National Univ.)
- Koji Kamei (ATR)
- Makoto Mizukawa (Shibaura-IT)
- Miki Sato (ATR)
- Mi-Sook Kim (KNU)
- Miwako Doi (Toshiba)
- Noriaki Ando (AIST)
- Seiichi Shin (UEC)
- Seung-Woog Jung (ETRI)
- Su-Young Chi (ETRI)
- Takashi Tsubouchi (Univ. of Tsukuba)
- Tetsuo Kotoku (AIST)
- Toshio Hori (AIST)
- Yoshihiro Nakabo (AIST)

Robotics Domain Task Force Preliminary Agenda ver.0.0.1 robotics/2011-06-21

OMG Technical Meeting - Kissimmee, FL, USA -- September 19-23, 2011

http://robotics.omg.org/

		TF/SIG				
		Host	Joint (Invited	Agenda Item	Purpose	Room
Monday:						
9:00	12:00			**RoIS** (Robotic Functional Services) WG(3h): - Su-Young Chi (ETRI), Miki Sato (JARA/ATR) and Toshio Hori (AIST)	discussion	
12:00	13:00			**LUNCH**		
13:00	18:00			Architecture Board Plenary		
14:00	17:00			**RoIS** (Robotic Functional Services) WG(3h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	
Tuesday: WG activity						
9:00	12:00			**RoIS** (Robotic Functional Services) WG(3h): - Su-Young Chi (ETRI), Miki Sato (JARA/ATR) and Toshio Hori (AIST)	discussion	
12:00	13:00			**LUNCH**		
13:00	17:00			**RoIS** (Robotic Functional Services) WG(4h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	
Wednesday: WG activitiy						
12:00	14:00			**LUNCH and OMG Plenary**		
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 Domain Task Force Preliminary Agenda ver.0.0.1 robotics/2011-06-22

OMG Technical Meeting - Santa Clara, CA, USA -- December 12-16, 2011

http://robotics.omg.org/

		TF/SIG				
		Host	Joint (Invited	Agenda Item	Purpose	Room
Sunday: WG activites(pm)						
13:00	17:00			Robotics DDC4RTC submitters meeting	Arrangement	
Monday: WG activity and Robotics-DTF Plenary(am)						
9:00	11:00			Robotics DDC4RTC submitters meeting	Arrangement	
11:00	12:00	MARS	Robotics	**Revised Submission for DDC4RTC RFP Review, Vote-to-Vote, and Voting** - Noriaki Ando (AIST) and Seung-Woog Jung (ETRI)	Joint with **MARS**	
12:00	13:00	**LUNCH**				
13:00	18:00			Architecture Board Plenary		
13:00	18:00			**DDC4RTC** (Robotic Infrastructure) WG(5h) - Noriaki Ando (AIST) and Seung-Woog Jung (ETRI)	discussion	
				RoIS (Robotic Functional Services) WG(5h): - Su-Young Chi (ETRI), Miki Sato (JARA/ATR) and Toshio Hori (AIST)	discussion	
Tuesday: WG activity(am) and Robotics-DTF Plenary(pm)						
9:00	12:00			**DDC4RTC** (Robotic Infrastructure) WG(3h) - Noriaki Ando (AIST) and Seung-Woog Jung (ETRI)	discussion	
				RoIS (Robotic Functional Services) WG(3h): - Su-Young Chi (ETRI), Miki Sato (JARA/ATR) and Toshio Hori (AIST)	discussion	
12:00	13:00	**LUNCH**				
13:00	13:45	Robotics		Special Talk: (45min) - TBA	presenta ion and discussion	
13:45	14:30	Robotics		Special Talk: (45min) - TBA	presenta ion and discussion	
				Break (30min)		
15:00	15:30	Robotics		Talk: (30min) - TBA	presenta ion and discussion	
15:30	16:10	Robotics		WG Reports and Discussion (Service WG, Infrastructure WG, Models in Robotics WG)	presenta ion and discussion	
16:10	16:30	Robotics		Contact Reports: - Makoto Mizukawa(Shibaura-IT), and Young-Jo Cho(ETRI)	Information Exchange	
16:30	17:00	Robotics		Robotics-DTF Plenary Wrap-up Session (**DTF Co-Chair Election**, Roadmap and Next mee ing Agenda)	Robo ics plenary closing	
17:00				Adjourn joint plenary meeting		
17:00	17:30			Robotics WG Co-chairs Planning Session (Preliminary Agenda for next TM, Draft report for Friday)	planning for next meeting	
Wednesday: WG activitiy						
9:00	12:00			**DDC4RTC** (Robotic Infrastructure) WG(3h) - Noriaki Ando and Seung-Woog Jung	discussion	
				RoIS (Robotic Functional Services) WG(3h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	
12:00	14:00	**LUNCH and OMG Plenary**				
14:00	18:00			**DDC4RTC** (Robotic Infrastructure) WG(4h) - Noriaki Ando and Seung-Woog Jung	discussion	
				RoIS (Robotic Functional Services) WG(4h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	
18:00	20:00	**OMG Reception**				
Thursday: WG activitiy						
9:00	9:30	MARS		Joint Plenary with MARS (tentative) (reserved for DDC4RTC RFP Re-Review and Voting)	Joint with **MARS**	
9:00	12:00			Robotics WG activity follow-up	discussion	
12:00	13:00	**LUNCH**				
13:00	18:00			Architecture Board Plenary		
13:00	18:00			Robotics WG activity follow-up	discussion	
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, 24th June, 2011
Chair:, T. Kotoku and Y. –J. Cho
URL: http://robotics.omg.org/
email: robotics@omg.org

➢Highlights from this Meeting:

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]

Robotics-DTF

Date: Friday, 24th June, 2011
Chair:, T. Kotoku and Y. –J. Cho
URL: http://robotics.omg.org/
email: robotics@omg.org

➢Future deliverables (In-Process):

- Dynamic Deployment and Configuration for RTC (DDC4RTC) revised submission

➢Next Meeting (Kissimmee):

- WG activity only

➢Next Meeting (Santa Clara):

- Election of a Robotics-DTF Co-Chair
- Review of the revised submission of DDC4RTC
- Guest presentation
- Contact reports
- Roadmap discussion

Minutes of the Robotics DTF Meeting - DRAFT
June 20-24, 2011
Salt Lake City, UT, USA
(robotics/2011-06-24)

Meeting Highlights

- Revised submission of the Robotic Interaction Service (RoIS) Framework RFP was accepted and the Finalization Task Force (RoIS-FTF) was chartered.
- The final report of Robotic Localization Services (RLS-1.1) was accepted to issue.
- Three presentations;
  - "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]

List of Generated Documents

robotics/2011-06-01 Final Agenda (Tetsuo Kotoku)
robotics/2011-06-02 Washington DC Meeting Minutes [approved] (Geoffrey Biggs and Seung-woog Jung)
robotics/2011-06-03 Draft: Revised submission ver.2 to the RoIS Framework RFP (Miki Sato)
robotics/2011-06-04 Draft: Revised submission ver.2 to the RoIS Framework RFP with Change bar (Miki Sato)
robotics/2011-06-05 Draft: Errata for ver.2 from 4weeks before original version (robotics/2011-05-01) (Miki Sato)
robotics/2011-06-06 Presentation: Revised Submission to Robotic Interaction Service (RoIS) Framework (Toshio Hori, Miki Sato, and Su-Young Chi)
robotics/2011-06-07 Event Management Model for DDC4RTC (Seung-Woog Jung)
robotics/2011-06-08 Repository & Directory Managers for DDC4RTC (Seung-Woog Jung)
robotics/2011-06-09 OPRoS: Open Platform for Robotic Services (Hong Seong Park)
robotics/2011-06-10 Conformance Testing Method for Robotic Software Components (Mi-sook Kim)
robotics/2011-06-11 Robotics Technology Applied to Great East Japan Earthquake (Miwako Doi)
robotics/2011-06-12 Infrastructure WG Progress Report (Noriaki Ando)
robotics/2011-06-13 Robotic Functional Services WG Report (Toshio Hori)
robotics/2011-06-14 Contact Report: Standardization of RTC-CANopen (Makoto Mizukawa)
robotics/2011-06-15 Contac Report: ISO/TC184/SC2/WG1 (Tetsuo Kotoku)
robotics/2011-06-16 Charter RoIS-FTF (Miki Sata)
robotics/2011-06-17 Contact Report: IEEE SA (Standardization Activities)
robotics/2011-06-18 Opening Presentation (Tetsuo Kotoku)
robotics/2011-06-19 Roadmap for Robotics Activities (Tetsuo Kotoku)
robotics/2011-06-20 Wrap-up Presentation (Tetsuo Kotoku)
robotics/2011-06-21 Kissimmee Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)
robotics/2011-06-22 Santa Clara Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)
robotics/2011-06-23 DTC Report Presentation (Su-Young Chi)
robotics/2011-06-24 Salt Lake City Meeting Minutes - DRAFT (Miki Sato and Seung-woog Jung)
robotics/2011-06-25 OPRoS Catalog (Hong Seong Park)

Minutes

Monday, June 20, 2011, Sun Valley, 2nd FL
Robotics DTF Plenary Meeting
AIST, ATR, ETRI, JARA, KAR, KNU, Sibaura-IT, Toshiba, UEC, Univ. of Tokyo, Univ. of Tsukuba

10:55-11:00 Robotics DTF Opening Session, Chair: Dr. Kotoku (AIST)
- Minutes takers: Miki Sato (ATR) and Seung-Woog Jung (ETRI)

11:00-12:00 Revised Submission Presentation for RoIS Framework RFP, Sun Valley, 2nd FL
- Presented by Dr. Hori, AIST, Japan.
- Review of the revised RoIS framework and discussion.
- Vote to Vote for the RoIS Framework revised submission [robotics/11-06-03].
  Approved: Shibaura IT(motion), AIST(second), ETRI(white ballot)
- Vote for the RoIS Framework revised submission [robotics/11-06-03].
  Approved: JARA(motion), ETRI(second), Shibaura IT(white ballot)
- AB review schedule
  RLS: Mon. 13:40-14:00 at Arizona, Main Level.
  RoIS Framework: Mon. 16:20-16:40 at Arizona, Main Level.

Tuesday, June 21, 2011, Sun Valley, 2nd FL
Robotics DTF Plenary Meeting
AIST, ATR, ETRI, JARA, KAR, KNU, Sibaura-IT, Toshiba, UEC, Univ. of Tokyo, Univ. of Tsukuba

13:00-14:00 Special Talk: Prof. Hong Seong Park (Kangwon National Univ.)
- Design Tools for Robot Software: OPRoS (Open Platform for Robotic Service)
- Introduction of OPRoS framework, its functions and tools.
- Several demonstrations using OPRoS.

14:00-14:30 Special Talk: Prof. Mi-Sook Kim (Kangwon National Univ.)
- Testing Methods for Robot Software Component
- Introduction of OpRoS Test System.
- Introduction of 3 types of SW component testing, methods, states and ports.

15:00-15:30 Special Talk: Dr. Miwako Doi (Toshiba)
- Robotics Technology Applied to Tohoku Earthquake and Tsunami
- Introduction of robots working at the nuclear plant in Fukushima.
- Introduction of Toshiba crawler robots with a Gamma Camera.

15:30-15:35 Wrap-up Session, Chair: Dr. Kotoku (AIST)
Quorums: 3
AIST, ETRI, JARA, KAR, Shibaura-IT, UEC, Univ. of Tsukuba
- Washington meeting minute review
  Approved: AIST(motion), Univ. of Tsukuba(second), Shibaura-IT(white ballot)

15:35-15:45 WG report: Infrastructure WG, Dr. Ando
- Reviewing and merging homeworks of each others..
- Merging two initial submissions: Component Data Model, Event Management, Reposity Manager, Directory Manager, Application Supervisor.
- Review and discussion about PIM and UML diagrams.

15:45-15:55 WG report: Functional Service WG, Dr. Hori (AIST)
- Presentation and voting for the revised submission to the RoIS.
- Presentation at the AB and passed with condition.
- Drafting a charter of RoIS FTF.

15:55-16:10 Contact Report, Prof. Makoto Mizukawa (Shibaura IT)
- Introduction of RTC-CANopen standardization in CiA.

16:10-16:25, Demonstration of Prototype of RoIS Framework, Dr. Sato (JARA/ATR)
- Demonstrate the prototype of RoIS Framework libraries.

16:25-16:35 Contact Report, Dr. Kotoku (AIST)
- Introduction of ISO/CD9787 Coordinate system and motion
- Introduction of ISO/DIS8373 Vocabulary

16:35-16:45 Contact Report, Dr. Miwako Doi (Toshiba)
- IEEE standardization meeting at ICRA2011

16:45-17:00 Wrap-up Session, Chair: Dr. Kotoku (AIST)
- Call for volunteer Co-Chair of Robotics-DTF: No volunteers, postponed one meeting.
- Roadmap for Robotics Activities
  RTC-RTF Report deadline postponed to Santa Clara meeting, Dec. 2011
- Schedule for next meeting: skip the next Kissimmee OMG meeting.

ATTENDEE (17 Participants)
- Andrew Watson (OMG)
- Noriaki Ando (AIST)
- Geoffrey Biggs (AIST)
- Su-Young Chi (ETRI)
- Miwako Doi (Toshiba/Univ of Tokyo)
- Toshio Hori (AIST)
- Chul-Jong Hwang (KAR)
- Seung-Woog Jung (ETRI)
- Koji Kamei (JARA/ATR)
- Mi-Sook Kim (Kangwan National Univ.)
- Tetsuo Kotoku (AIST)
- Makoto Mizukawa (Shibaura IT)
- Yoshihiro Nakabo (AIST)
- Hong-Seong Park (Kangwan National Univ.)
- Miki Sato (JARA/ATR)
- Seiichi Shin (UEC)
- Takashi Tsubouchi (Univ. of Tsukuba)

Prepared and submitted by Seung-Woog Jung (ETRI) and Miki Sato (JARA/ATR).

OPRoS Framework

Characteristics

Component based structure
> Convenient structure for various components
> Loads HW driver routine (.dll or .so) only when changing linked component
> Operation in both Windows XP / Linux / RTOS(QNX) this year

Various optional functions, and optional functions selectable
> Script language component for portability
> Compiled components for computing speed
> Easy configuration of components (Reactive model, Behavior model, Hybrid model, User creative model, etc.)

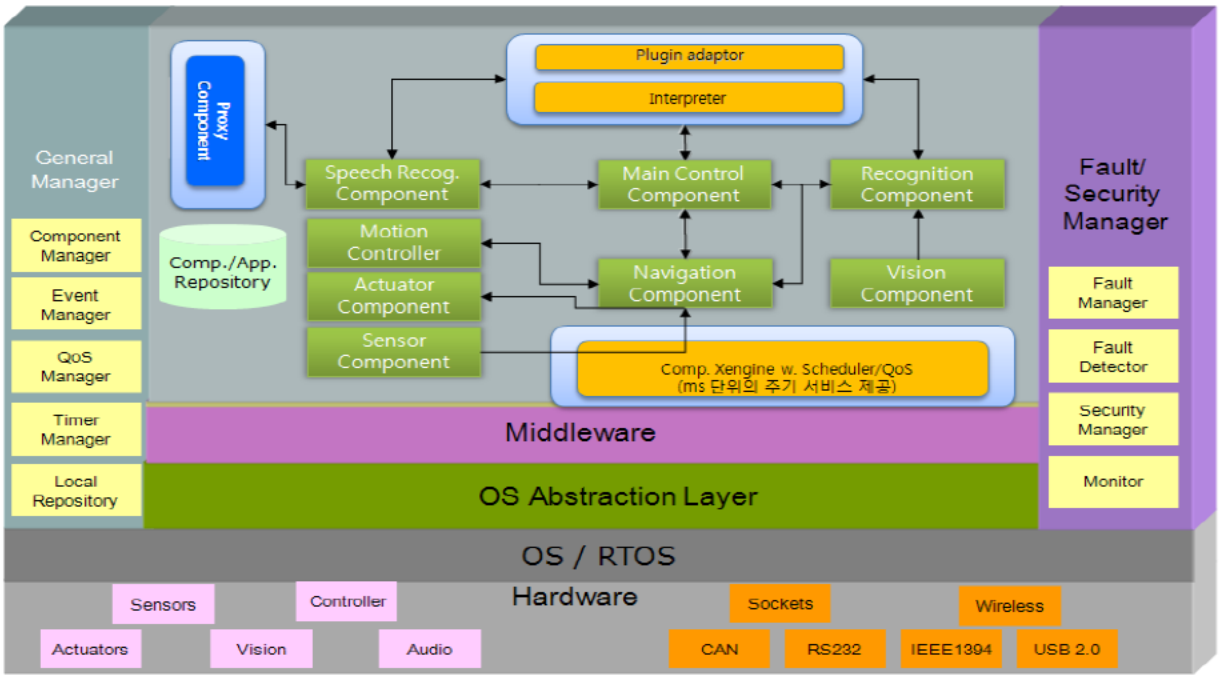
Portability of application programs (Ensured in both Windows and Linux)
> Application program composite with task script and VPL language
i.e.) Application program is for user execution of desired robot behavior

Specifications

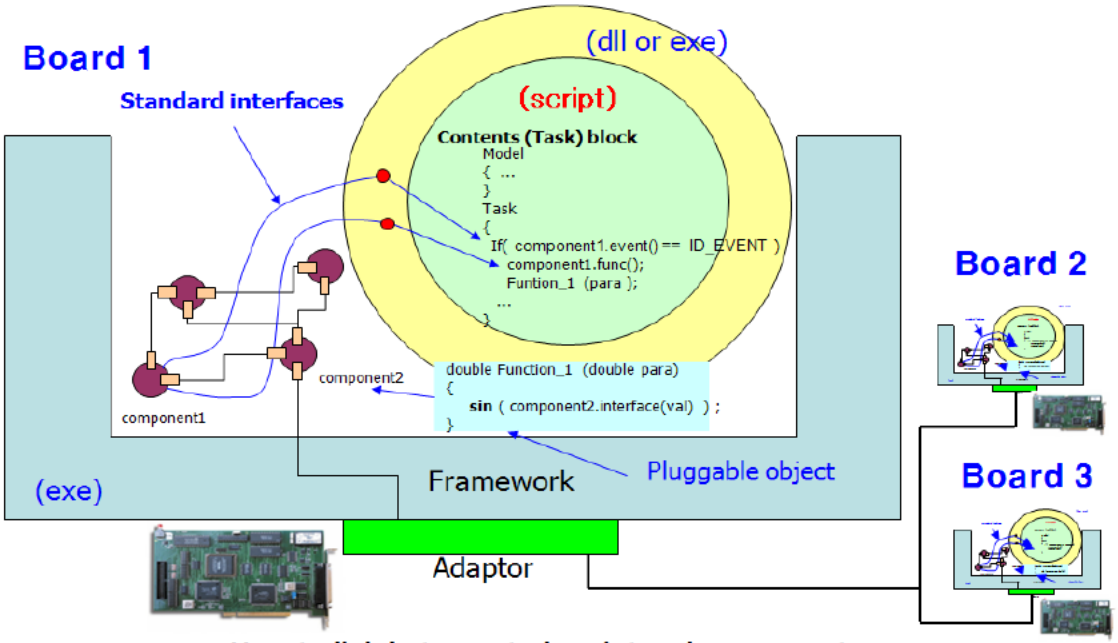
Operate in 32 bit CPU (Support 16 bit planed)
Minimum memory usage: 16 MB (all functions supported)
Application program languages
> Task script, VPL
> C/C++, Control block
> Python, Java (To be added)

Applications

Robots using OPRoS
> NXT of LEGO
> Rovonova of Minirobot
> TETRA of Dasarobot
> iRobiQ of Yujin Robot
> E3 of Robotware
> robot ED 7270 of ED
> Other various prototype robots
> Industrial robots / Rehabilitation robots / Humanoid (Forthcoming)



OPRoS 프레임워크 구조



How to link between task script and component

Functions

Execution Engine (Mandatory)
> Scheduling of tasks in components
  * Periodic task (currently 50ms for windows and 1ms for Linux)
  * Aperiodic task

> Adjustable execution time according to component frequency and priority
  * Check executability of component within the period
  * Execution by user's intention

> Support proxy component
  * Local component corresponding to remote service

Component/Application Program Manager (Mandatory)
> Management of component/application program
> Loading/Unloading components/application program
> Connection and management of data flow between components

Communication Manager (Mandatory)
> Communication between robot and IDE
> Communication between robot and server
> Communication between robots
※ Currently working on the standardization of communication protocol for general use
(Binary message format, to be in text or ML types afterward)

Middleware/Component adapter (Mandatory)
> UPnP and CORBA embedded communication middleware between a component and a component in the boards.
> No involvement of user with middleware data exchange between components treated through adaptor.
> Direct communication between components in the same board

Event manager (Option)
> Manager to register and release events generated in each component / manager / engine
> Each component checks necessary event registered in the event manager. If registered, event notifies corresponding components when generated

Fault manager (Option)
> Manage fault-safe or fault-operation actions against defined faults
e.g.) Fault-safe: Stop safely when corresponding fault occurs
> Fault-operation: Keep operating even when fault occurs
> Fault Manager operates with Event Manager.

Script Interpreter
> Task script language support sequence, parallel and state transitions, and with components
> Support Multimedia Service

Security Manager (Option)
> Decides robot security level
> Treats hacking / virus / spying, and operates existing security system
> Responds with special security management system, especially, when remote controlling the vision system, sensing system, and mobile operation (Planned)

Time Manager (Option)
> Manage time synchronization between multiple boards in the robot

OPRoS Editor and Tools

Characteristics

Eclipse based Integrated Development Editors

> Custom-made editor for each application

SW Type	Exclusive Editor	Application Examples	Debugging
Application Program * Contents * Task	* Task Editor * VPL Editor	Errand service, public announcement service, etc. * Assembly, dancing, etc.	* Verification and evaluation tool * simulator
Combination of unit SW *Robot model SW	*Component Composer	Guide/Public/Education service robot * Industrial robot	
Unit SW * HW linking components * Pure SW linking components	Component Editor	HW control, environment recognition * Position control, navigation * Voice, vision, etc.	
Motor Control SW	Control Block Editor	* Servo DC motor, etc.	* Exclusive simulator

> Component Editor easily edits, tests, and debugs components
 * Template for each SW type for easy development of HW and SW components
 * Easy linking of HW using API provided by the same type of H/W device independent of OS
> Component Block Editor configures robot SW using components
 * Easy configuration of various types of robot model (reactive model, behavior model, mixed model, use creative model, etc.)
> Graphic/Text based Task Editors for easy development of application programs(or contents)
 * Provides debugging function linked with simulator
 * link with component variables in components for various control and application of robots
> control Block Editor for easy development of motor control programs, etc. and can do simulation
 * Provides various commands applicable to the control algorithm and motor models
 * Provides user calling function
> Graphic programming editor assists easy developments of various application programs
 * Graphic language programming
 * User function calls

SW Verification / Evaluation Tool

> Verifies and evaluates SW components and application programs
> Provides automatic creation of various test cases and automatic execution of the tests

Simulator for components and application programs

> Provide environment modeler for various environments and robot modeler for various robots
> Operate robot's actual framework and application programs in the simulator

Provide source code

Specifications

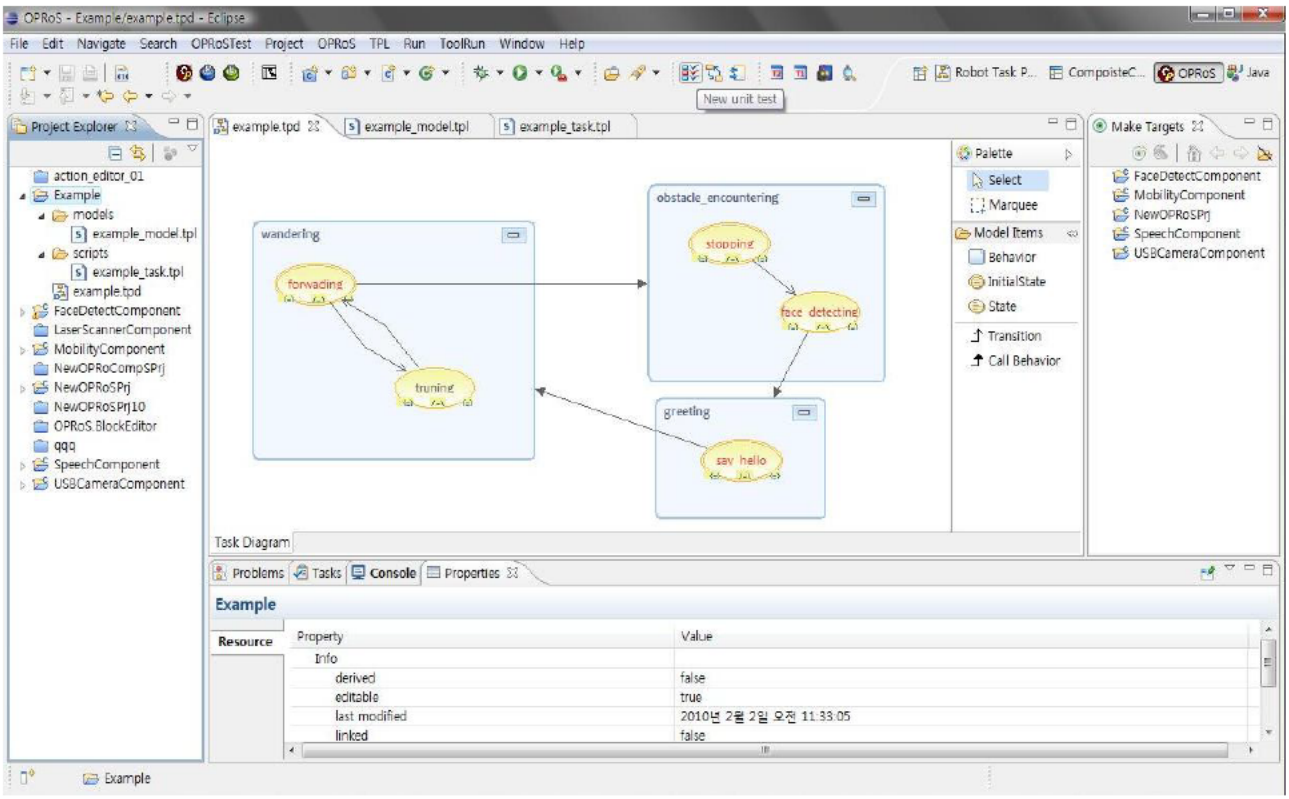
Program size
> 345MB (including Eclipse Java version and Eclipse plug-in needed for editor)

Supported OS
> Windows, Linux

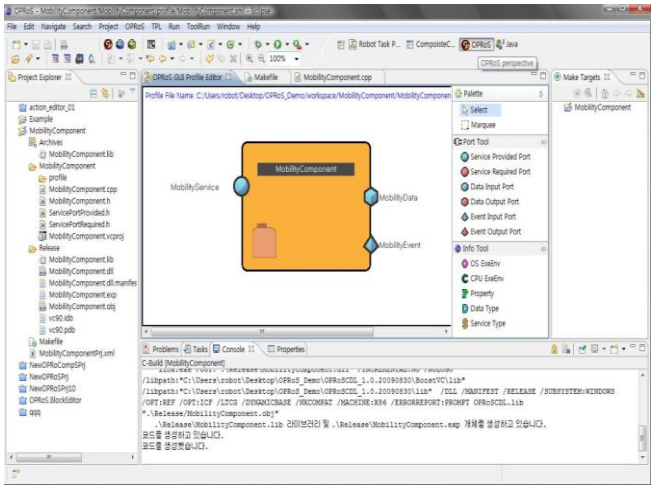
Applications

- **Development of robot application programs (contents)**
- **Development of HW module-related SW manufacturing and design of robot**
- **Robot related business**
- **SW verification and evaluation**
- **Robots using OPRoS**
  > NXT of LEGO
  > Robonova of Minirobot
  > Tetra of Dasarobot
  > iRobiQ of Yujin robot
  > E3 of Robotware
  > Robot ED 7270 of ED
  > Other various own developed robot
- **Industrial robots / Rehabilitation robots / Humanoid** (Forthcoming)

Example of Task Editor



Example of Component Editor

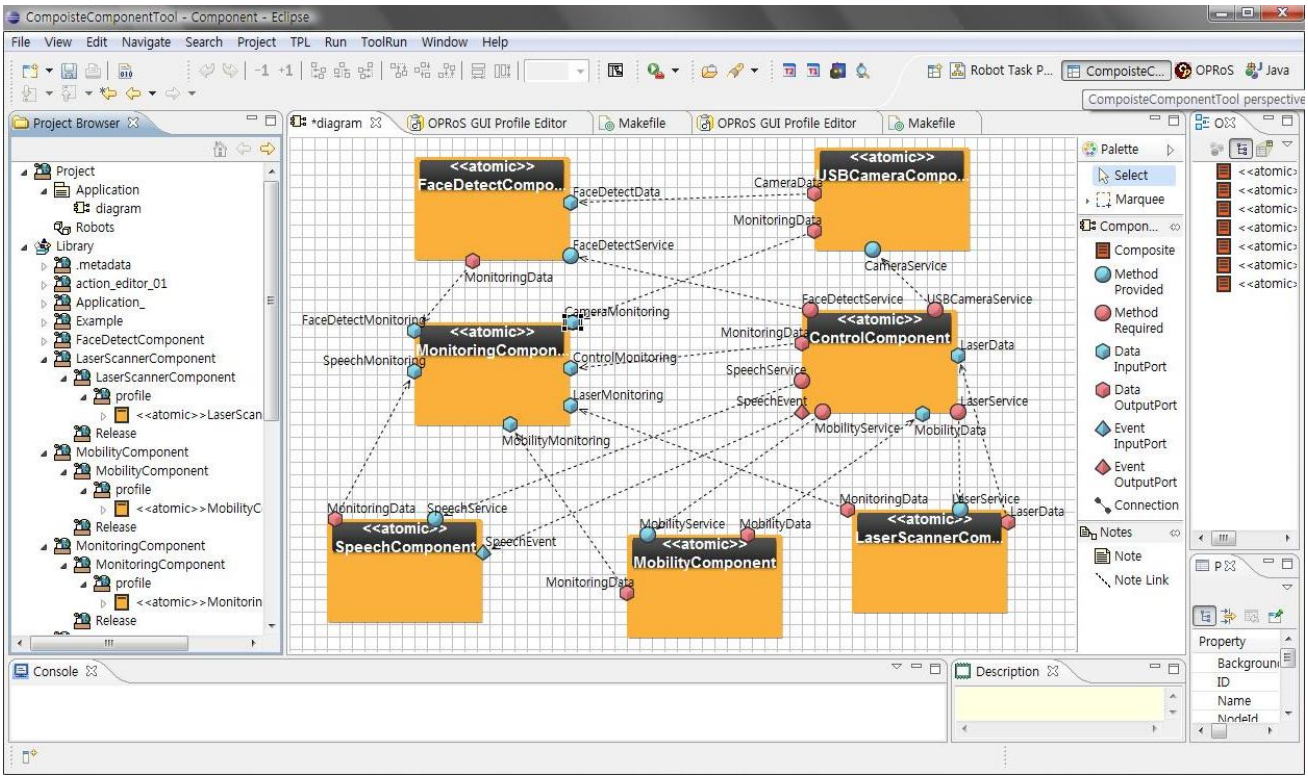


MobilityComponent.cpp | CameraComponent.cpp | HelloMaker.cpp | MessagePrinter.cpp

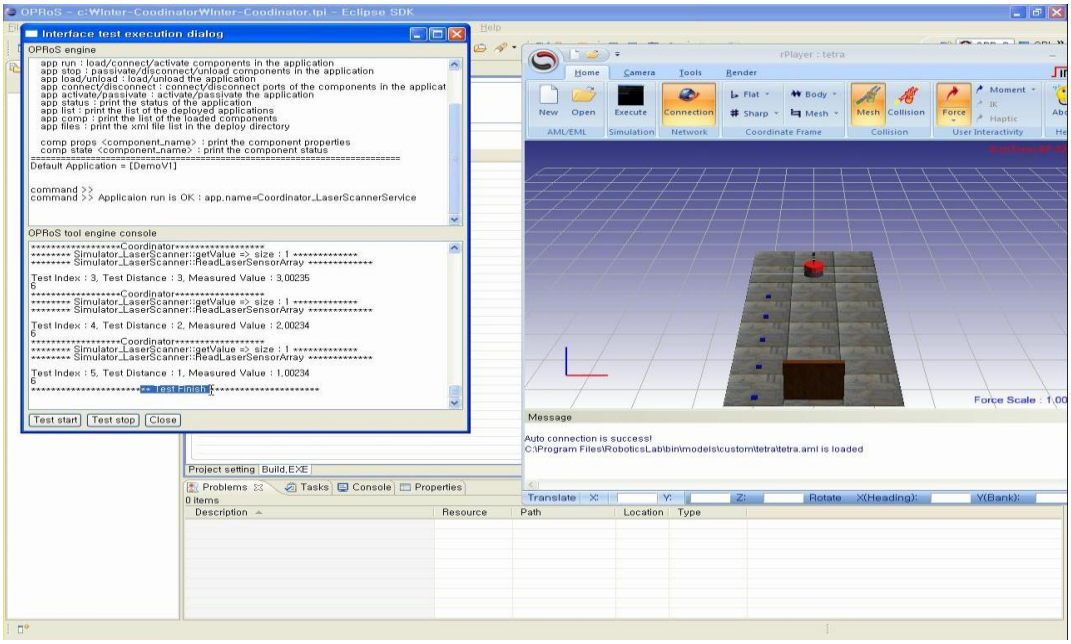
```
ReturnType MobilityComponent::onExecute()
{
    // user code here
    if (mobility == NULL) {
        return OPROS_CALLER_ERROR;
    }

    if (!mobility->execute ()) {
        return OPROS_CALLER_ERROR;
    }

    return OPROS_SUCCESS;
}

ReturnType MobilityComponent::onUpdated()
{
    // user code here
    if (mobility == NULL) {
        return OPROS_CALLER_ERROR;
    }

    if (!mobility->update ()) {
        return OPROS_CALLER_ERROR;
    }

    return OPROS_SUCCESS;
}

ReturnType MobilityComponent::onPeriodChanged()
```

Example code of component

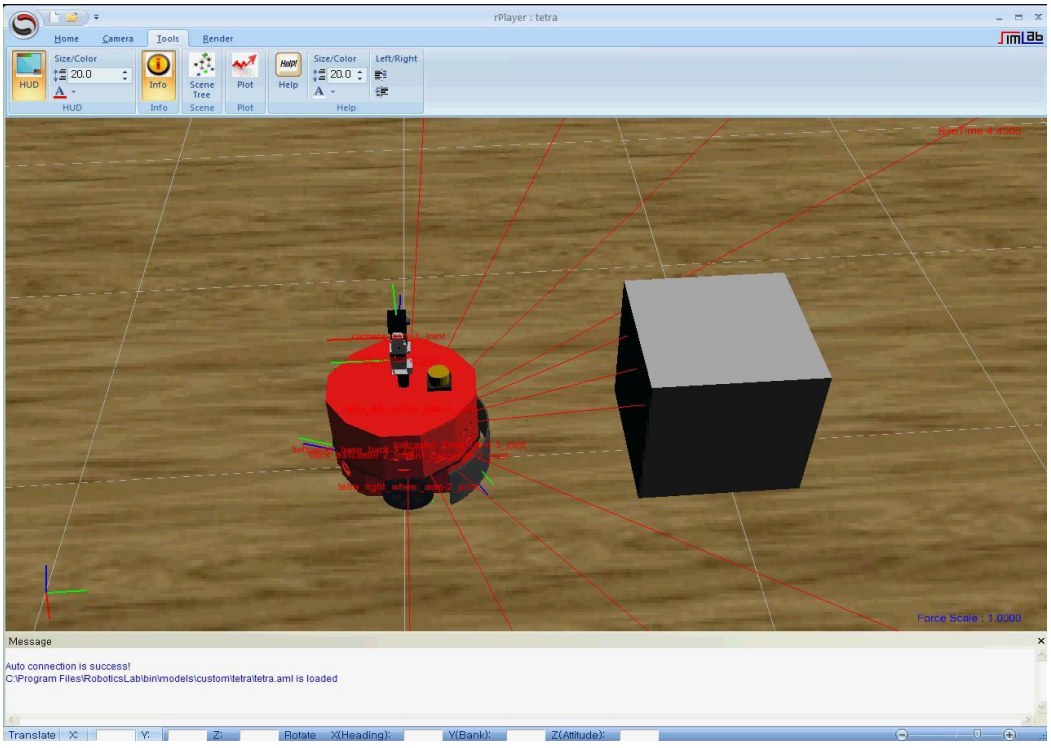
Example of Component Composer



Example of Verification and Evaluation Tool



Example of Simulation

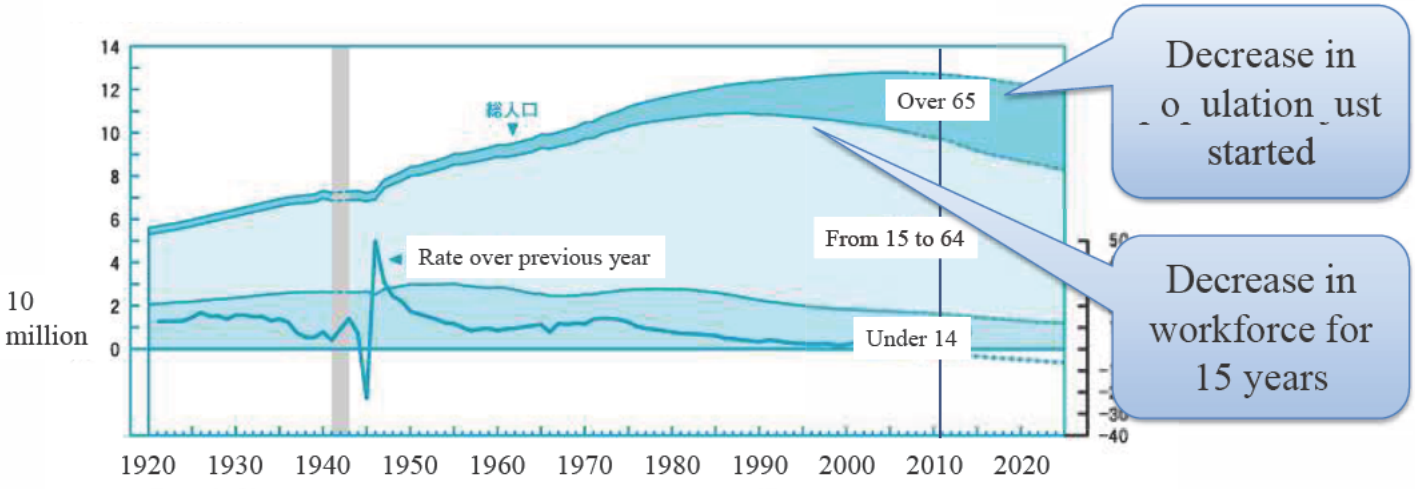


Safety of Personal Care Robots as an Example of Consumer Devices

OMG TECHNICAL MEETING SPECIAL EVENT
Seminar On Systems Assurance & Safety For Consumer Devices:
Automotive, Robotic & Building Automation Systems Of The Future
June 22th, 2011

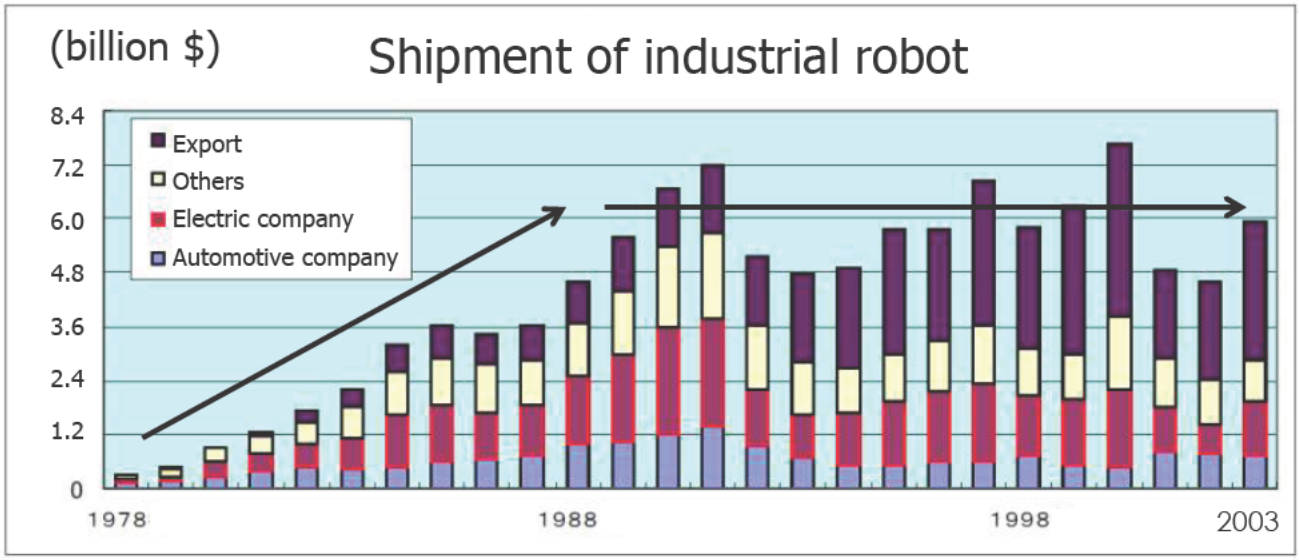
Yoshihiro Nakabo
Senior Research Scientist,
Dependable Systems Research Group, Intelligent Systems Research Institute,
National Institute of Advanced Industrial Science and Technology(AIST)

Aging Society with Falling Birthrate



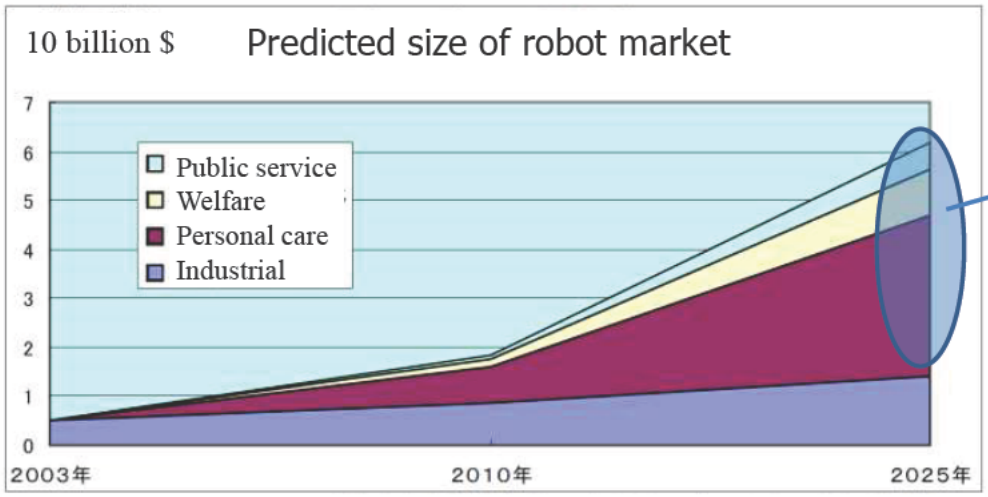
- Innovation is necessary to maintain Japanese economy and society.

Japanese Robot Production



- Robot production = Industrial robot production
- Around 6 billion $ for 20years
- Export is growing recently

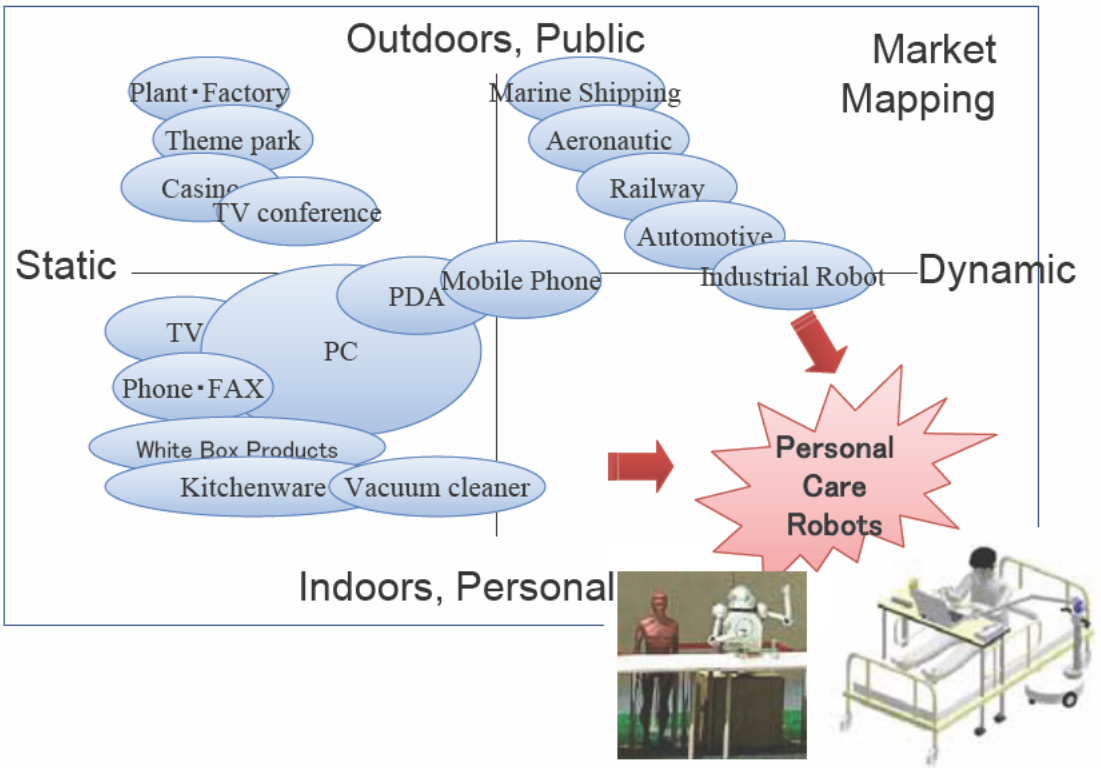
Expectation for Personal Care Robots



- 4.7 millions of worker decrease in 20 years
- If ¼ of worker decrease replaced by robots >> 80 billion $

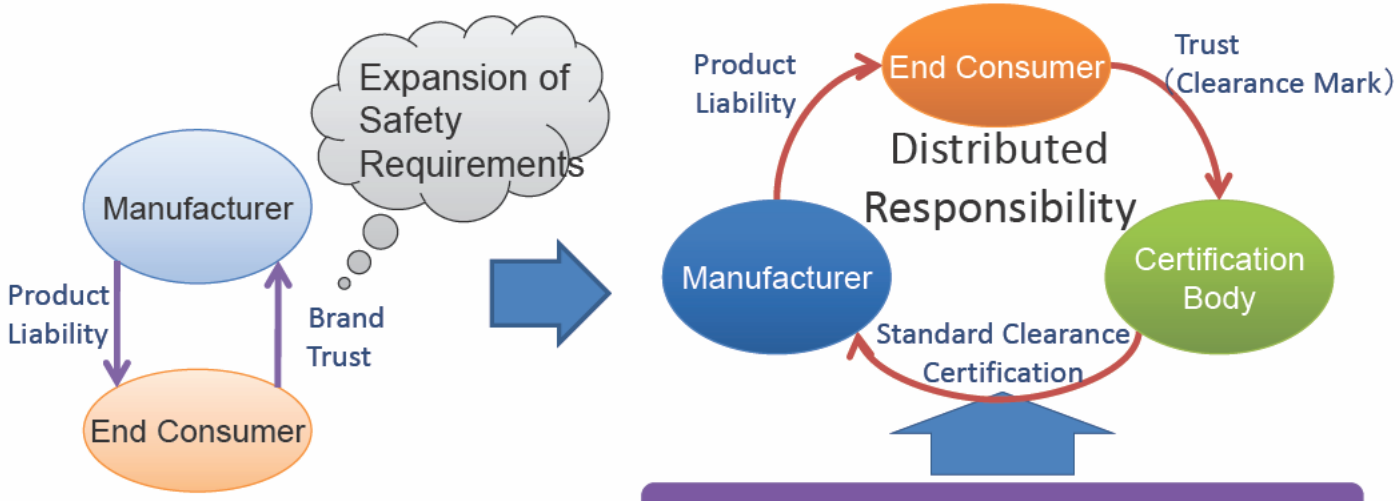
Product Position of Service Robots

Needs to enter a completely new market area



Distribution of Safety Responsibility

- Can just one country or corporate be held responsible?



NEDO Safety Verification Technology Project on Personal Care Robots

NEDO (New Energy and Industrial Technology Development Organization, Japan)

- Research and Development Schedule

1st FY(2009)	2nd FY(2010)	3rd fy(2011)	4th FY(2012)	5th FY2013)
1st phase				
R&D of safety verification/testing technology				
		2nd phase		
Budget scale about 2 million $ for the 1st fiscal year (2009)		Demonstration by service robots		

Project Promotion

R&D item①
Development of safety verification/testing technology

(1) Establishment of safety criteria, safety verification/testing methodology
- development of risk assessment process for service robot
- development of testing methods for service robots: such as mechanical/ electrical safety, functional safety

(2) Establishment of safety conformity assessment methodology

linkage

risk assessment

safety testing

safety verification

(R&D items ② ~ ⑤)
Development of service robots (SRs) with safety technology installed

mobile SRs centered by manual operation

Autonomously controlled mobile SRs

physical assistance SRs

person carrier SRs

(1) Development of safety technology for SRs
- risk assessment

(2) Safety verification of SRs
- testing
- demonstration

Establish safety criteria
Contribute to making safety standards

Robots are put into practical market with safety technology installed.

Service Robots in this Project

METI and NEDO expects 4 types of service robots to cultivate a market within 10 years. R&D activities of safety evaluation technique for these 4 types of service robots will be performed in the project.

1) Mobile servant robots (with manipulator)

A robot that is capable of moving freely to perform an intended task and allowing physical contact with humans.

R&D: Robotic Bed (Panasonic Corporation)

2) Mobile servant robots (without manipulator)

A robot that is capable of moving freely to perform an intended task and allowing physical contact with humans.

R&D:
Cleaning Robot (Fuji Heavy Industries Ltd.)
Security Robot (Sohgo Security Services Co.,Ltd.)

AIST Intelligent Systems Research Institute
NEDO Safety Verification Technology Project

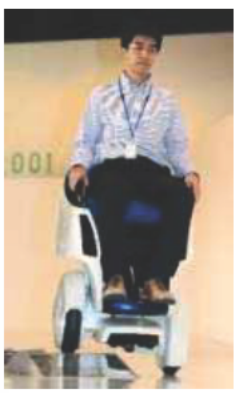
3) Physical assistance robots (Restrained type)

A robot that is typically worn by human or fastened to human in order to supplement or augment one's physical ability, depending upon one's physical situations or physical impairment levels. A few examples include wearable suits f the physically-impaired for their rehabilitation and those for factory hands to overcome their physical fatigue.

R&D: ROBOT SUIT HAL (CYBERDYNE)

4) People carrier robots

A robot that possesses a cabin or is equipped with a seat o footboard for the purpose of transporting a human to a different location with one's intention (manual navigation) by means of autonomous locomotion.



R&D: Winglet, MoBiRo, etc. (Toyota Motor Corp.)

Research Center for the NEDO project at Tsukuba, Japan

Personal Care Robots

Personal mobility

Example usage:
- Transportation in public area
- Sports or fun
- frequent movement

Automatic vehicle robot

Example usage:
- Cleaning or security guard in public area or office building/supermarket

Powered suite

Example usage:
- Rehabilitation
- Supporting ones walk
- Help for nursing care

Robotic bed

Example usage:
- Mobility in room
- Help for nursing care

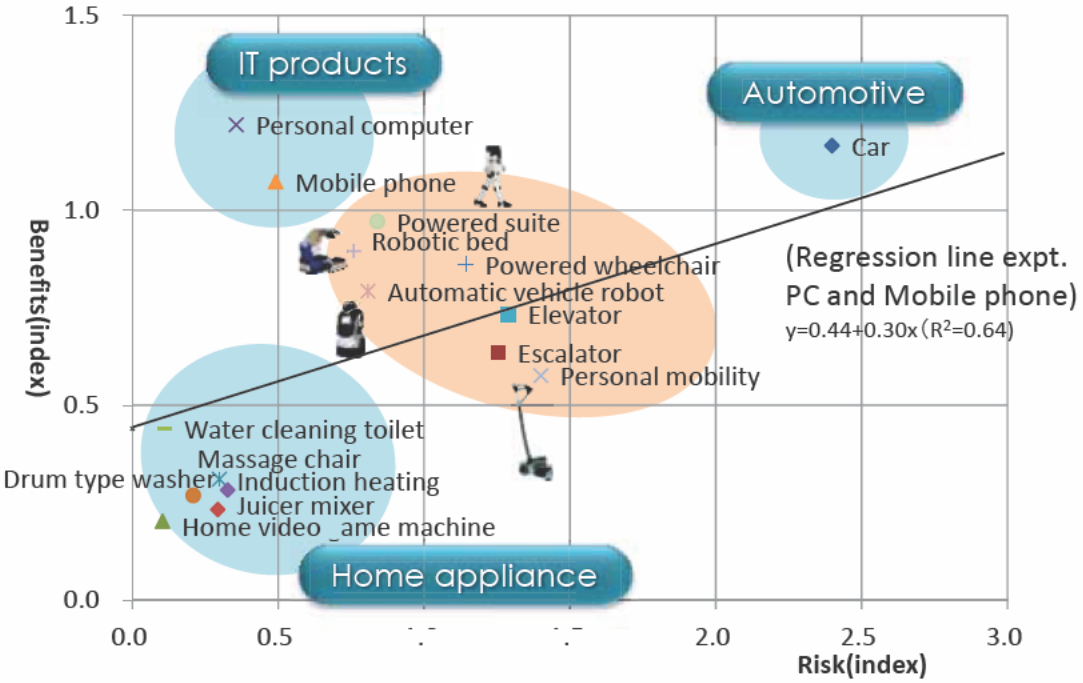
- Research for 4 types of personal care robots as samples.(from NEDO safety verification technology project on personal care robots)

- And other 12 familiar consumer products.(car, escalator, elevator, home video game machine, personal computer, massage chair, drum type washer, powered wheelchair, juicer mixer, water cleaning toilet, induction heating, and mobile phone)

Pictures from NEDO project

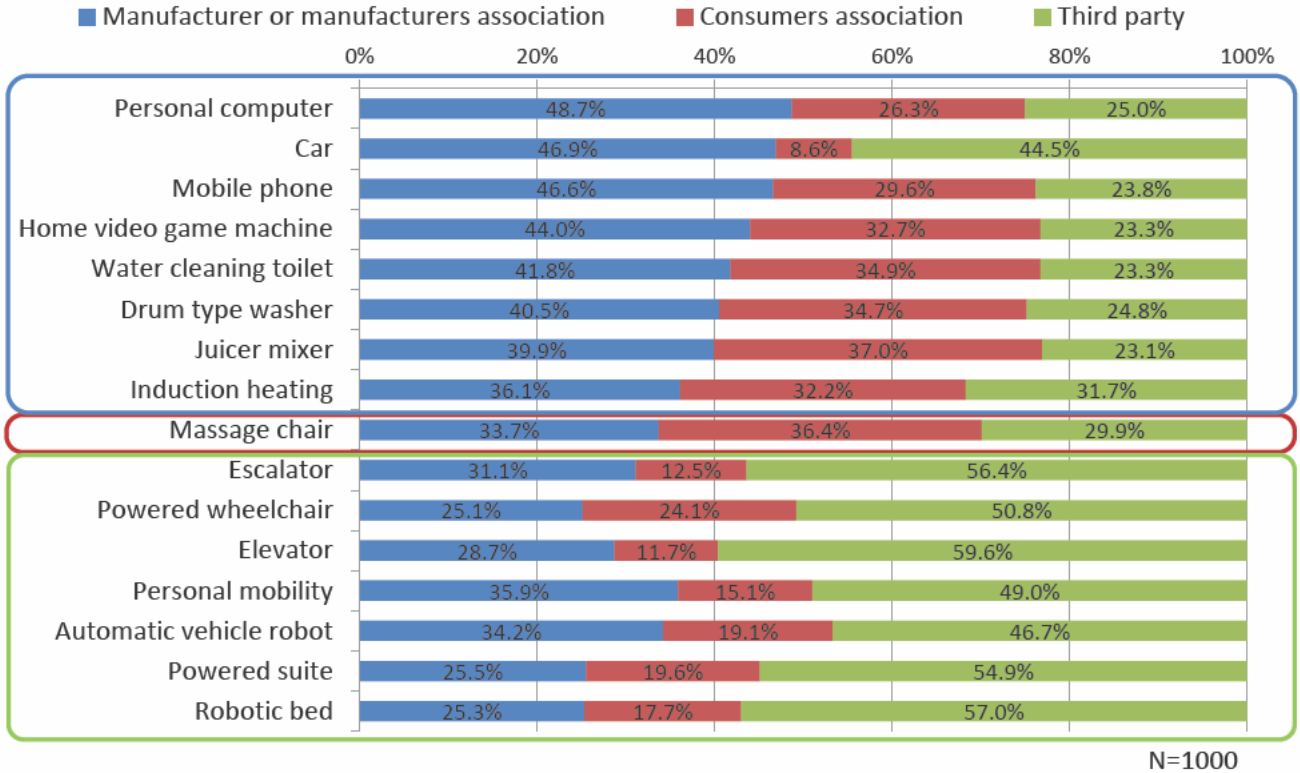
Balance of Risk and Benefits

- Risk image of personal care robots are not smaller than other products
- Personal care robots can be classified as a same group with “Powered wheelchair”, “Escalator” and “Elevator”



Who shall assure the safety of products

- Third party is expected for personal care robots



International Standards

- Industrial Robots:
  - Safety standardization completed
- Service Robots:
  - NEDO project for practical applications of service robot
  - ISO13482 standardization in progress

Hierarchical Structure of International Standards (ISO/IEC) Concerning Machinery Safety

ISO12100 Safety of Machinery

Definition

- Safety:

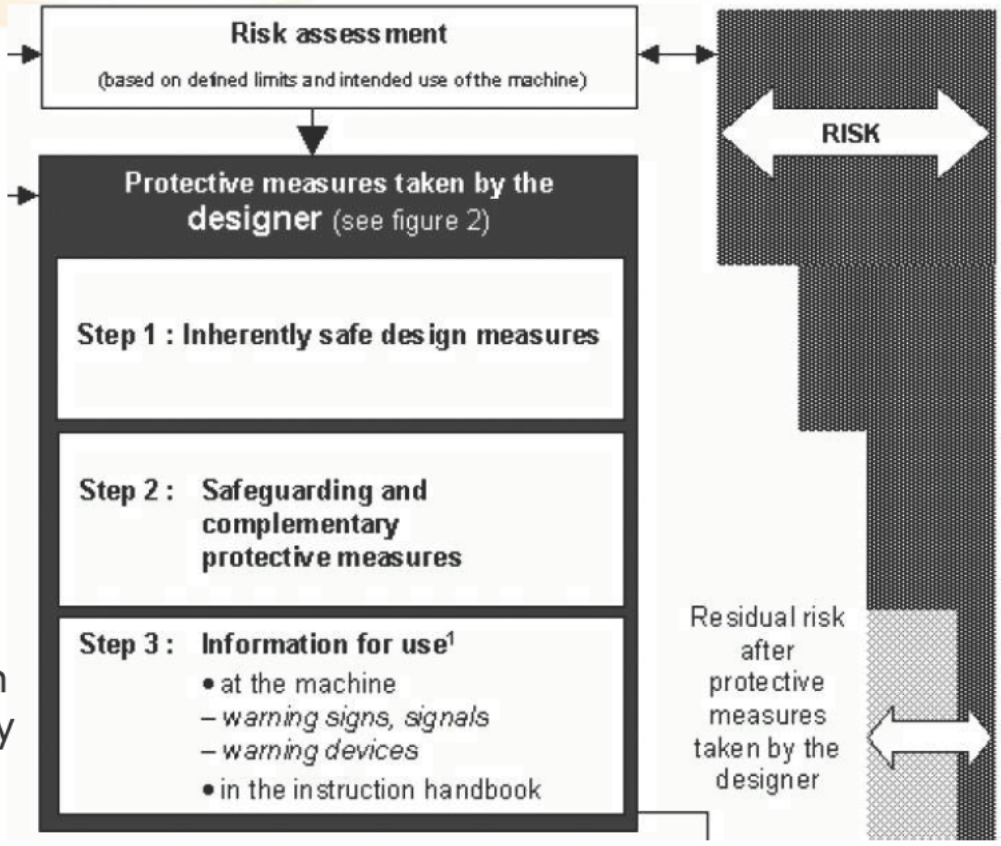
freedom from unacceptable risk

- Risk:

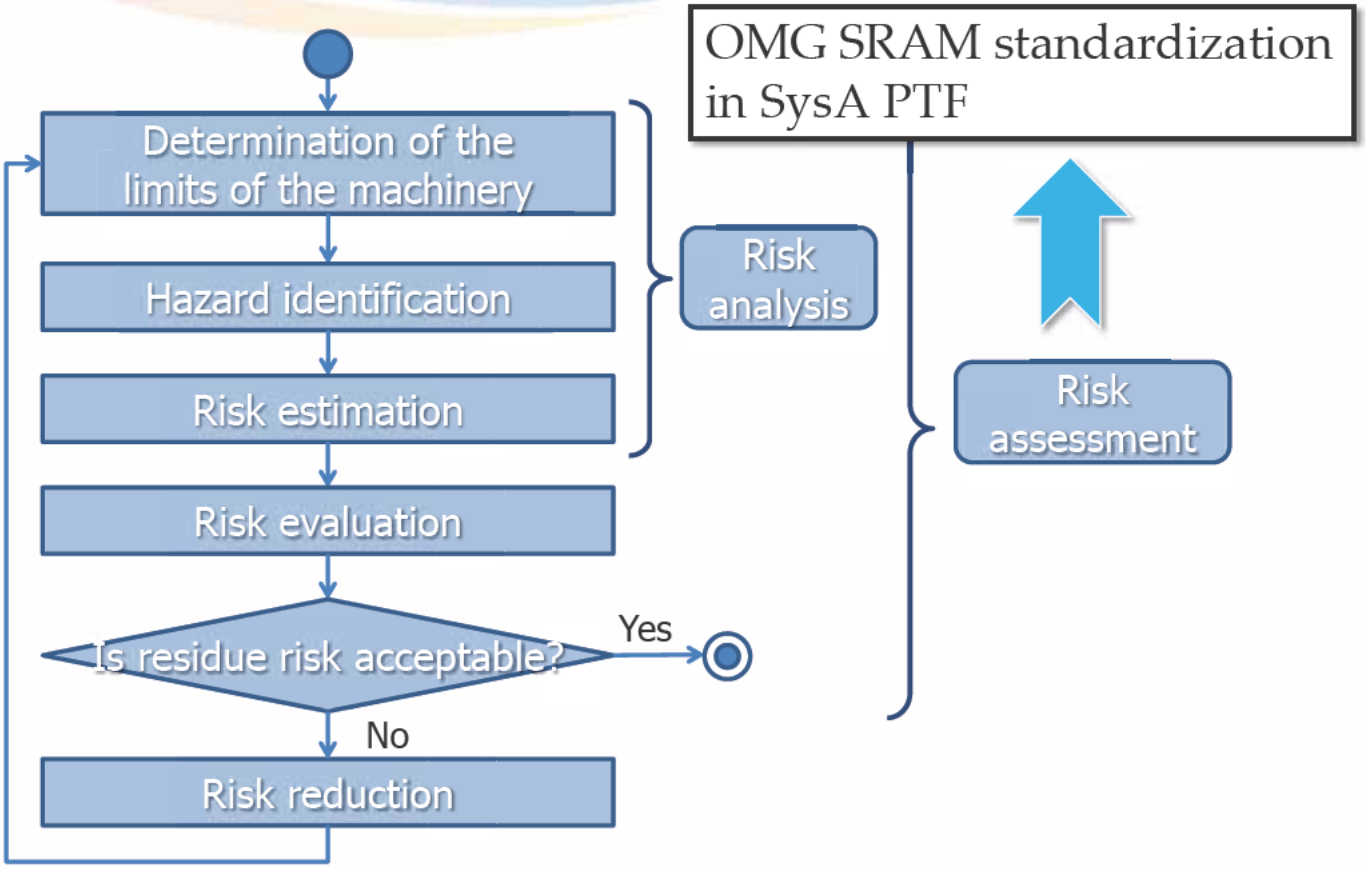
combination of the probability of the occurrence of harm and the severity of that harm

- Harm:

physical injury or damage to the health of people, or property or the environment

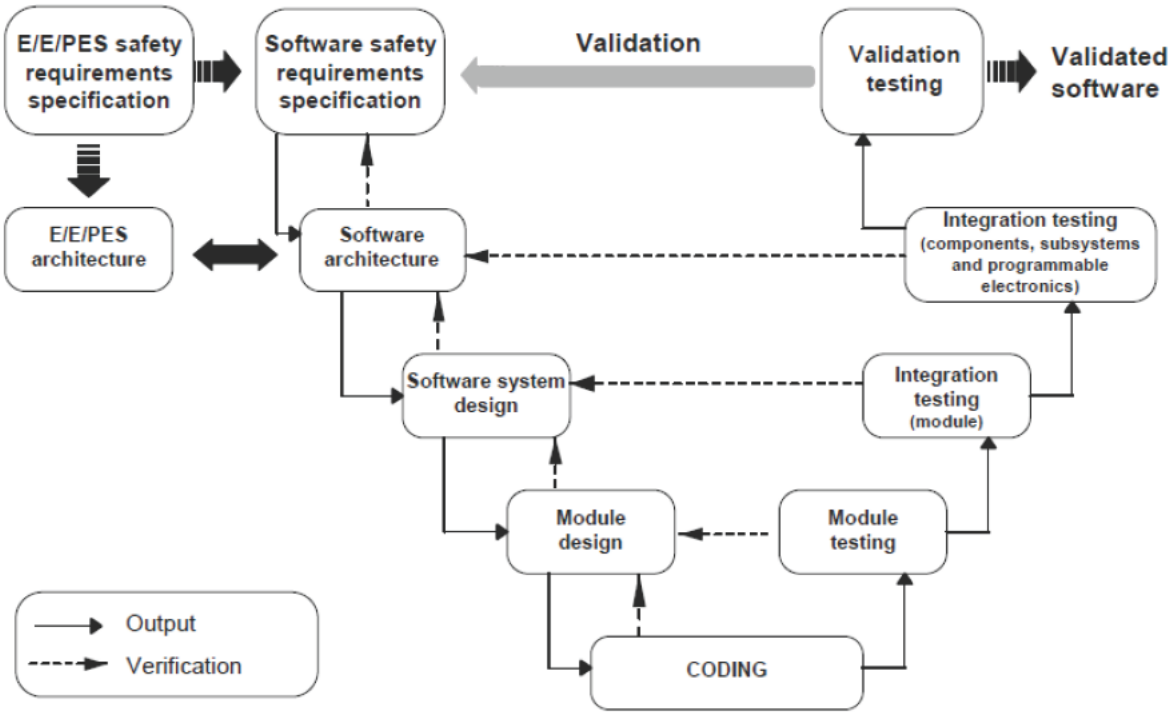


ISO 14121 Risk Assessment



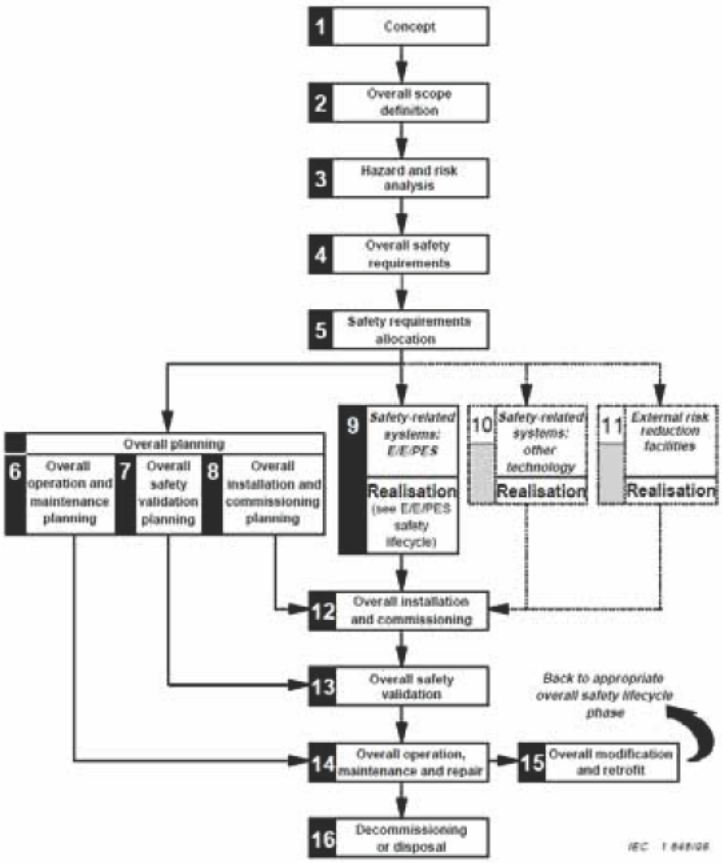
IEC 61508 Functional Safety

The V Model: software safety integrity and the development lifecycle



from IEC61508-3

IEC61508 Safety Life Cycle

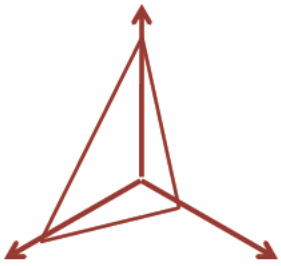


- Certification body concerns from early stage(safety concepts)
- All PDCA development cycle(Plan, Do, Check, Analyze)
- All Information related to safety must be opened
- Safety case = Huge documents
- Need re-development if not meet safety std.

  → Cost consuming

Relation Between Safety and Service

Service・Application



Safety Cost

Trade-off problem

Solved by platform technology and model-based rapid iteration

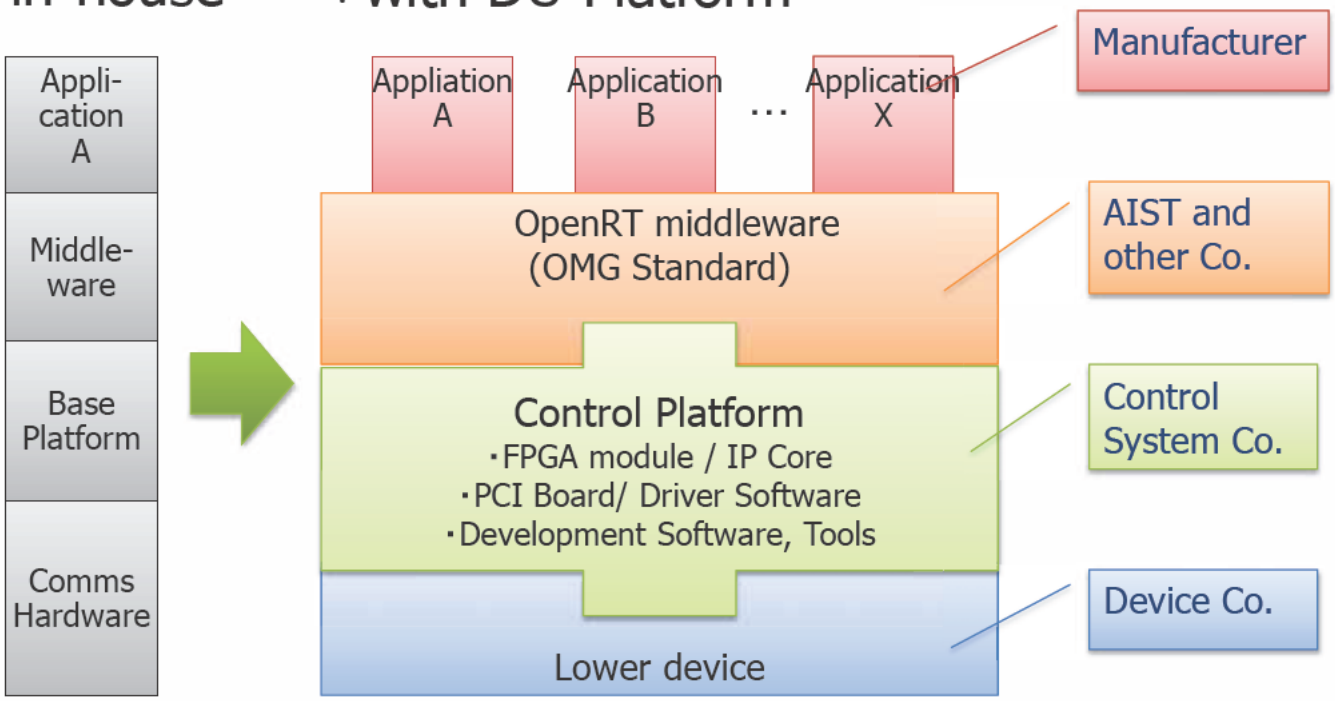
Application = Enough>>OK
Price = High>>NG
Safety = Safe>>OK
>>> No business

Application = Not enough>>NG
Price = Low>>OK
Safety = Safe>>OK
>>> No business

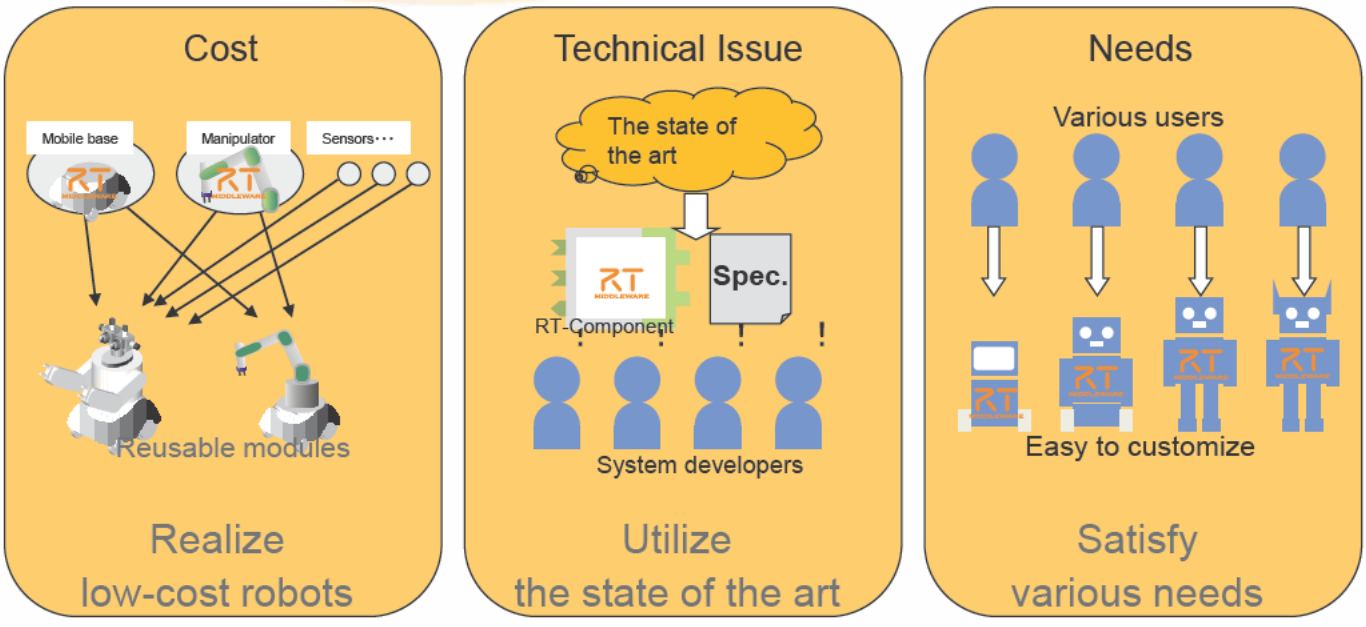
Application = Enough>>OK
Price = Low>>OK
Safety = Not safe>>NG
>>> **Should not sell**

Idea behind use of Platform Technologies

- All in-house
- Service Robot Development with D3 Platform

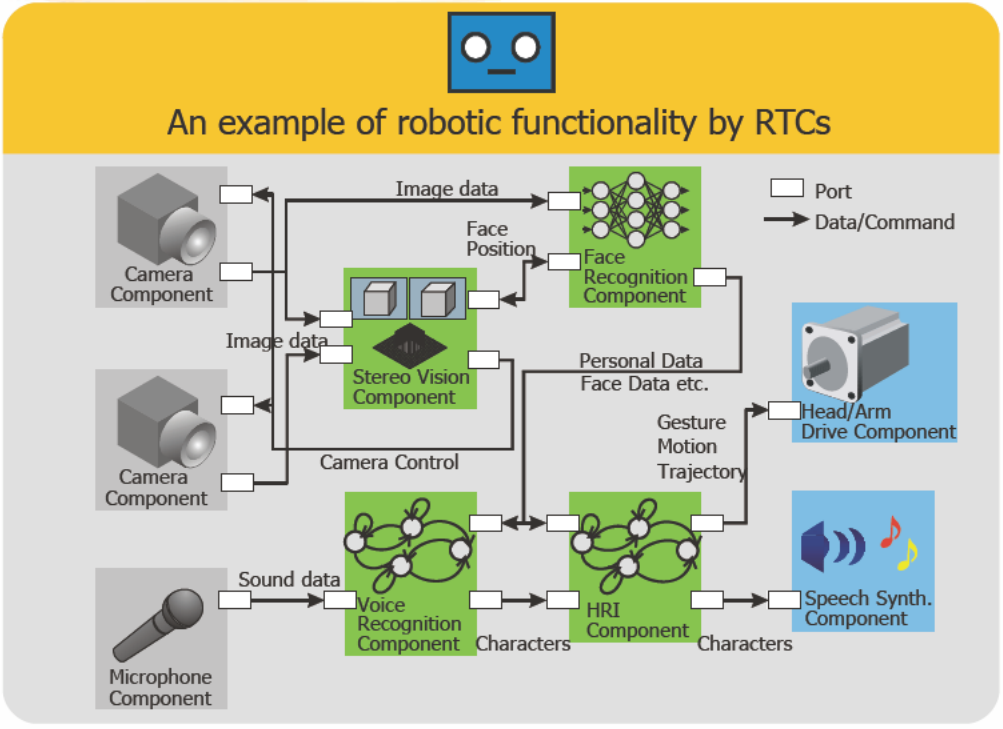


Problem Solving by Modularization



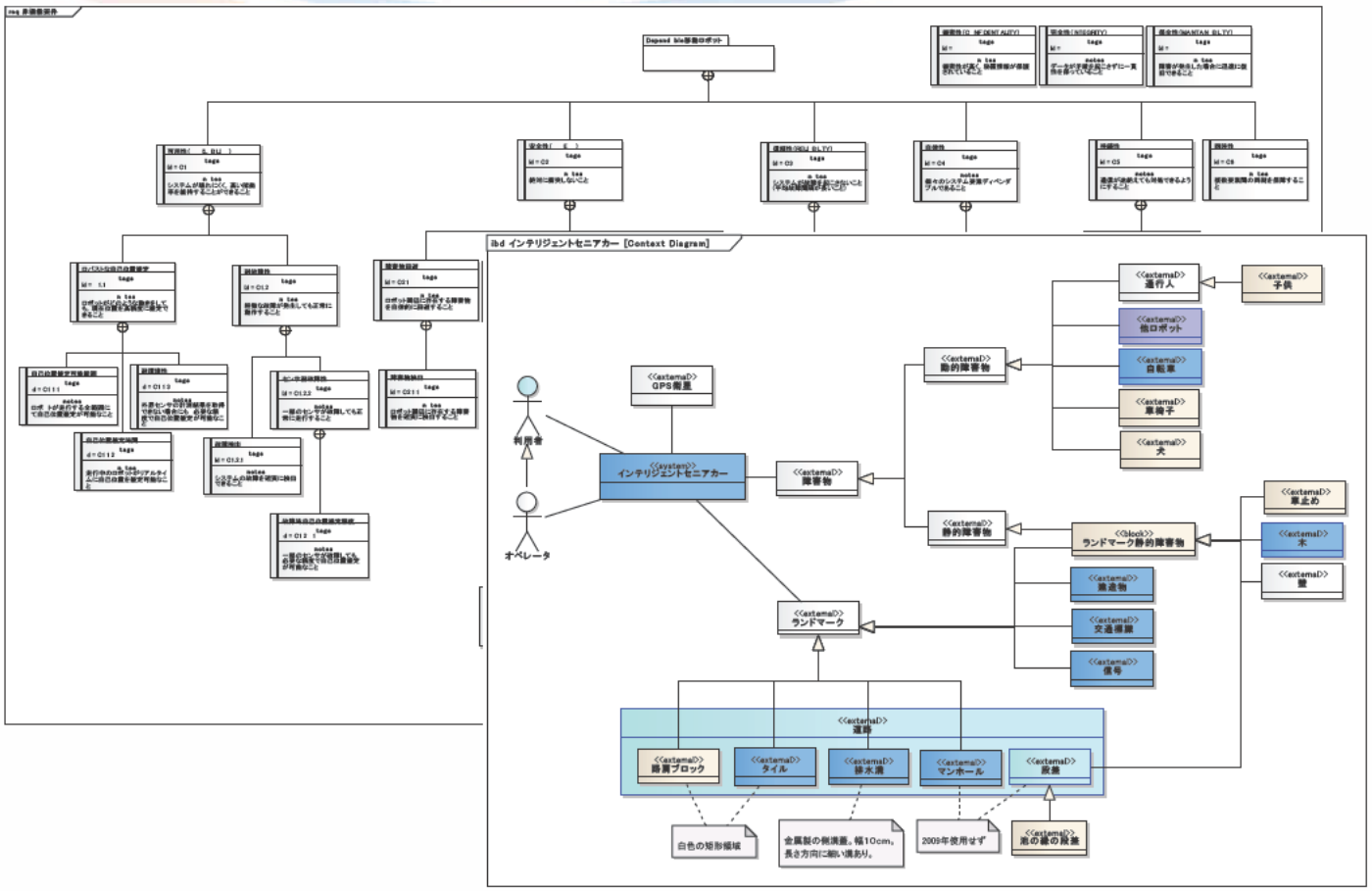
- Clear specification
- Rapid introduction of new technology
- Easy to start

RTC decomposition and integration



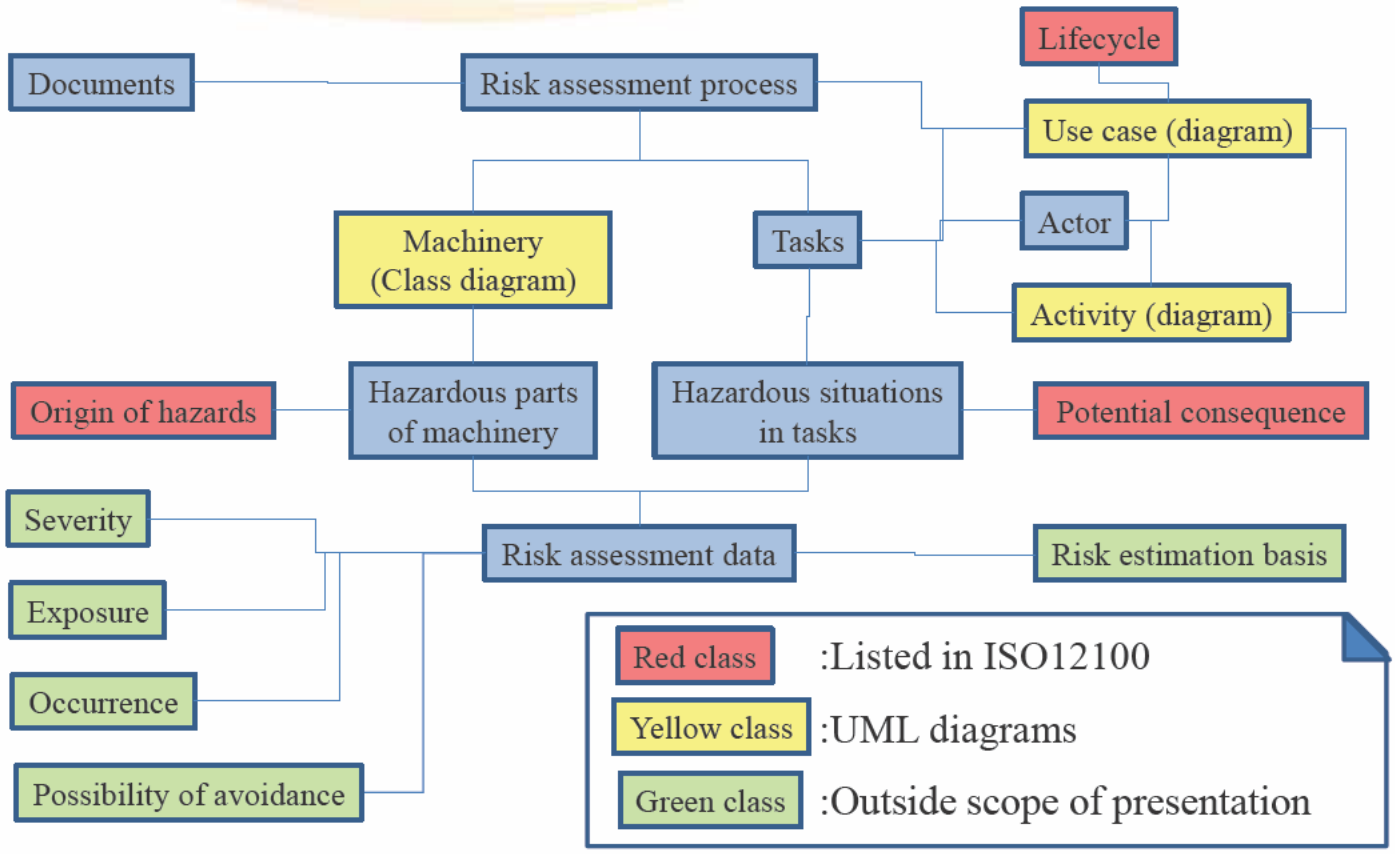
RTCs allow the concealment and appropriate disclosure of module information

SysML modeling of Personal Care Robot



UML model of risk assessment

System Risk Assessment Metamodel(SRAM) standardization at OMG



Conclusion

- Trial to marketize personal care robots
- Requirements from safety standard
- Model-based and module based development to fulfill requirements of both safety and reasonable cost