Robotics DTF Final Agenda ver.1.1.6 robotics2006-02-04

OMG Technical Meeting - Tampa, FL, USA -- February 13-17, 2006

		TF/SIG				
		Host	Joint (Invited)	Agenda Item	Purpose	Room
Monday (Feb.13)						
12:00	13:00			**LUNCH**		Audubon A, 1st FL
13:00	18:00			Architecture Board Plenary		Audubon B, 1st FL
13:00	14:40	SDO, SBC	Robotics	RFP initial submission joint pre-discussion	Information exchange	Herring Gull, 2nd FL
				Break (20min)		
15:00	17:00	Robotics	SDO	Steering Committee of Robotics DTF	Volunteer recruit	Herring Gull, 2nd FL
17:00	18:00	BM&I	(Robotics)	Embedded Technology Skill Standard (ETSS)	Information exchange	Audubon C, 1st FL
Tuesday (Feb.14) Robotics Plenary						
8:40	**9:00**	Robotics, SDO		Welcome and Review Agenda	Robotics/SDO Joint Meeting Kick-off	Audubon B, 1st FL
9:30	10:30	MARS	SDO, Robotics	Robot Technology Components RFP initial submission - Noriaki Ando (AIST)	review	Audubon B, 1st FL
10:30	11:30	MARS	SDO, Robotics	Robot Technology Components RFP initial submission - Hung Pham (RTI)	review	
12:00	13:00			**LUNCH**		Audubon A, 1st FL
13:00	13:40	Robotics	(SDO)	Hitachi's needs for robotic system standards - Saku Egawa (Hitachi)	RFI response	Audubon B, 1st FL
13:40	14:20	Robotics	(SDO)	Towards Plug and Play Robotics - Abheek Kumar Bose (ADA Software Group)	RFI response	
14:20	15:00	Robotics	(SDO)	SEC's approach to the standardization of robotics systems - Hiroyuki Nakamoto and Masayuki Nagase (SEC)	RFI response	
				Break (20min)		
15:20	16:00	Robotics	(SDO)	Development of Food Robots and Meat Processing Robots, and Request for Standardization of RTC - Tomoki Yamashita (Mayekawa MFG)	RFI response	
16:00	16:40	Robotics	(SDO)	RT service framework using IT infrastructure - Wonpil Yu (ETRI)	RFI response	
16:40	17:20	Robotics	(SDO)	A mobile robot software system architecture with unified sensory data integration - Takashi Tsubouchi (Tsukuba Univ.)	RFI response	
17:20	18:00	Robotics	(SDO)	Navigation of mobile robots including mapping, localization, and motion - Wonpil Yu (ETRI)	RFI response	
Wednesday (Feb.15) Robotics Plenary						
8:10	8:50	Robotics	(SDO)	Response from AIST - Olivier Lemaire (AIST)	RFI response	Pelican, 2nd FL
8:50	9:30	Robotics	(SDO)	Current State of Robotics Script/Control Language Standards - Lloyd Spencer (CoroWare)	RFI response	
9:30	10:20	Robotics	(SDO)	***<Special Talk>*** ***"Lessons Learned About Software for Rescue Robots"*** - **Matt Long** (CRASAR, Univ. of South Florida)	Informative	
				Break (20min)		
10:40	11:20	Robotics	(SDO)	Development Framework for Mobile Robot based on JAUS and RT-Middleware - Wataru Inamura (IHI)	RFI response	
11:20	12:00	Robotics	(SDO)	an overview of PIM & PSM for SWRadio Components specification - Jerry Bickle (Prismtech)	RFI response	
12:00	14:00			**LUNCH and OMG Plenary**		Audubon A, 1st FL
14:00	14:40	Robotics	(SDO)	Response from Compare Project - Virginie Watine (THALES)	RFI response	Pelican, 2nd FL
14:40	15:20	Robotics	(SDO)	Robot Server Middleware: CAMU - Seung-Ik Lee (ETRI)	RFI response	
				Break (20min)		
15:40	16:20	Robotics	(SDO)	Toshiba's approach to RT standardization and where the standardization is needed - Fumio Ozaki (Toshiba)	RFI response	
16:20	17:40	Robotics	(SDO)	Summary of RFI responses and working group discussion	chartering WGs	
17:40	17:50	Robotics	(SDO)	Publicity Activity	chartering SC	
17:50	18:00	Robotics, SDO		Next Meeting Agenda Discussion, etc	Robotics/SDO Closing session	
18:00				Adjourn		
18:00	20:00			**OMG Reception**		**TBA**
Thursday						
12:00	13:00			**LUNCH**		Audubon A, 1st FL
13:00	18:00			Architecture Board Plenary		Audubon B, 1st FL
14:00	15:00	MARS	SDO, Robotics	Robotics RFI Summary Report		White Ibis, 1st FL
Friday						
8:30	12:00			AB, DTC, PTC		Audubon DEF, 1st FL
12:00	13:00			**LUNCH**		Armani&'s, 14th FL
Other Meetings of Interest						
Monday						
8:00	8:45	OMG		New Attendee Orientation		Audubon D, 1st FL
9:00	12:00	OMG		Introduction to the MetaObject Facility (MOF)		Audubon D, 1st FL
13:00	17:00	OMG		Architecture-Driven Modernization Concepts and Task Force Update		Audubon D, 1st FL
18:00	19:00	OMG		New Attendee Reception (by invitation only)		Pelican, 2nd FL
Tuesday						
9:00	12:00	OMG		Introduction to OMG's Modeling and Middleware Specifications		Audubon D, 1st FL
13:00	17:00	OMG		An Overview of MDA--Where it Came From and Where It's Going		Audubon D, 1st FL
Wednesday						
14:00	17:50	OMG		Software Assurance (SwA) Information Session		Audubon B, 1st FL

Robotics DTF Steering Committee Meeting

February 13, 2006

Tampa, FL, USA

Grand Hyatt Tampa Bay

Herring Gull, 2nd FL

Agenda

- Agenda Review
- Publicity
- Working Group Discussion
- Roadmap Discussion
- Next meeting Schedule

Review Agenda

Tuesday, Feb.14, 2006 Audubon B, 1st FL

08:40-09:00 Welcome and Review Agenda

09:30-11:30 RTCs initial submission review [MARS]

13:00-15:00 RFI response presentation (3)

15:20-18:00 RFI response presentation (4)

Joint Meeting with MARS/RTESS
Thursday, Feb.16, 2006
14:00-15:00 (White Ibis, 1st FL)

Review Agenda

Wednesday, Feb.15, 2006 Pelican, 2nd FL

08:10-09:30 RFI response presentation (2)
09:30-10:20 Special Talk (CRASAR, Univ. of South Florida)

10:40-12:00 RFI response presentation (2)

14:00-15:20 RFI response presentation (2)

15:40-16:20 RFI response presentation (1)
16:20-17:40 RFI summary and discussion
17:40-18:00 Next meeting, etc.
18:00 Adjourn

Joint Meeting with MARS/RTESS
Thursday, Feb.16, 2006
14:00-15:00 (White Ibis, 1st FL)

Publicity Activities

- 4 page fly sheet
  Draft of Abheek@ADA Software
  Abheek@ADA Soft, Olivier@AIST, Chung@ETRI, Yokomachi@NEDO
- RoboBusiness event
  June 20-21, 2006, Pittsburgh, PA, USA
  http://www.robobusiness2006.com/
  no volunteer
- IROS2006 Workshop
  October 9-15, Beijing, China
  http://www.iros2006.org/
  Kotoku@AIST, Chung@ETRI, Mizukawa@Sibaura-IT

Roadmap (WG organization)

Report of Lemaire's survey

Funding potential co-chairs:

- Infrastructure: Rick@RTI, Ando@AIST
- Service: Chi@ETRI, Lemaire@AIST
- Tool: Abheek@ADA soft
- Data Profile: Lee@ETRI
- Device Profile: Bruce@Systronics

Next Meeting Agenda

April 24-28, 2006 (St. Louis, MO, USA)

Monday :

Steering Committee

Tuesday-Wednesday :

Robotics-DTF Plenary Meeting

•RTCs RFP revised submission pre-review

(MARS joint meeting)

•RFI response presentation

•Roadmap discussion

•Contact reports

Robotics-DSIG Meeting Minutes – Burlingame, CA, USA
(robotics/2006-02-06)

Overview and votes

Following the issuance of the "Robotic Systems RFI" at the Boston meeting (June 2005) this meeting has mainly consisted of reviewing the responses to the RFI.
Although there have already been a large number of responses, it has been decided to extend the deadline for responding to the RFI to 3 weeks before the Tamp meeting. This decision has been made in order to allow some organizations who have expressed the will to respond to the RFI but had no time or learned only too recently about our activity.
Especially to be mentioned is the motion made by Tetsuo Kotoku for our activity to move from the status of a Domain Special Interest Group (DSIG) to the one of a Domain Task Force (DTF). The motion was approved by all voters. Along with the new status, the 3 new co-chairs of the Robotics DTF have been elected. From the next OMG meeting, the three co-chairs will be : Hung Pham (RTI), Yun Koo Chung (ETRI) and Tetsuo Kotoku (AIST).

OMG Documents Generated

robotics/2005-12-01 Final Agenda (Tetsuo Kotoku)
robotics/2005-12-02 Atlanta Meeting Minutes [approved] (Olivier Lemaire)
robotics/2005-12-03 Opening presentation (Tetsuo Kotoku)
robotics/2005-12-04 Robotics-DSIG Roadmap (Tetsuo Kotoku)
robotics/2005-12-05 Special Talk: "Robots with Agents!" (James Odell)
robotics/2005-12-06 "On Robotics Middleware" (Hung Pham)
robotics/2005-12-07 "Self-Configuring Smart Java Robots through Plag and Play I/O Boards" (Bruce Boyes)
robotics/2005-12-08 "RSCA: Robot Software Communications Architecture" (Seongsoo Hong)
robotics/2005-12-09 "Human Interface of the Robotic System RFI" (Soo-Young Chi)
robotics/2005-12-10 "High Assurance Security and Safety for Robotics" (Joseph M. Jacob)
robotics/2005-12-11 "Standardization on Interfaces to Robot Devices " (Seung-Ik Lee)
robotics/2005-12-12 "Research and Development of Personal Robot in NEC" (Yoshihiro Fujita)
robotics/2005-12-13 "The Player/Stage/Gazebo project: Open Source tools for robotics research" (Brian P. Gerkey)
robotics/2005-12-14 "Network Robot Platform for Information Sharing" (Ken-ichiro Shimokura)
robotics/2005-12-15 "Human Robot Interaction in Network Robots" (Norihiko Hagita)
robotics/2005-12-16 "Network Robots Standardization Activity in Japan" (Miwako Doi)
robotics/2005-12-17 Proposed Charter for Robotics Domain Task Force
robotics/2005-12-18 : Atlanta Robotics DSIG DTC Report Presentation (Tetsuo Kotoku)
robotics/2005-12-19 : Meeting Minutes (Olivier Lemaire, Seung-Ik Lee, Claude Baudoin)

Agenda

<u>05 December, Monday</u>
15:00-17:00 – Steering Committee of Robotics DSIG

<u>06 December, Tuesday</u>
13:00-13:10 – Welcome and Review Agenda
13:10-14:10 – "Introduction to the Agent-SIG activities" - James J. Odell (OMG Agent PSIG)
14:10-15:00 – "Response to RFI from Real-Time Innovation" – Hung Pham (RTI)
15:20-16:10 – "Response to RFI from Java.net" - - Bruce Boyes (Systronix)
16:10-17:00 – "Response to RFI from SNU - The Robot Software Communications Architecture (RSCA) : Embedded Middleware for Networked Service Robots" - Seongsoo Hong (Seoul National Univ.)
17:00-17:50 – "Capabilities: Human Interface of the Robotic Systems RFI" - Soo-Young Chi (ETRI)

07 December, Wednesday
9:00-10:00 – “High Assurance Security and Safety for Robotics” - Joseph M. Jacob (Objective Interface Systems)
10:20-11:10 – “Hardware Abstraction to the Robotic Systems RFI” - Seung-Ik Lee (ETRI)
11:10-12:00 – “Response to RFI from NEC” - Yoshihiro Fujita (NEC)
14:00-14:40 – "The Player/Stage/Gazebo project: Open Source tools for robotics research" - Brian P. Gerkey (SRI International)
14:50-15:30 – “Network Robot Platform for Information Sharing” - Ken-ichiro Shimokura (NTT)
15:30-16:10 – “Human Robot Interaction in Network Robots” - Norihiko Hagita (ATR)
16:10-16:50 – “Network Robots Standardization Activity in Japan” - Miwako Doi (Toshiba)
17:00-17:20 – Chartering Robotics Domain Task Force and voting
17:20-17:40 – Next Meeting Agenda Discussion, etc Robotics/SDO Closing session
17:40 Adjourn

Minutes

06 December, Tuesday

Opening Presentation (Tetsuo Kotoku)

Special Talk : "Robots with Agents!"
Jim Odell defined an agent as "an autonomous entity that can adapt and interact with its environment" and commented on the concepts of autonomy, interaction, and adaptation, as they apply to software agents. His point was that robots are very much like agents, and that a number of the concepts developed in the OMG's Agents SIG would be relevant to the work of the Robotics SIG. For example, agents can coordinate their actions using different processes: negotiation, competition, persuasion, and deliberation.

Presentation – RFI Response : " On Robotics Middleware " - Hung Pham (RTI)
RTI provides software tools and services to developers of complex, distributed control systems (including robotic systems).
Hung Pham presented a "robotic technology vertical layer" model, with 7 levels:
7 = application layer: integrated subsystem with user interface, mode switching, automatic detection and self-organization capability
6 = domain layer: a subsystem composed of engineering domain-specific software
5 = middleware layer: application framework
4 = data layer: platform-independent data and interface representations
3 = operating system layer
2 = hardware abstraction layer (device drivers, etc.)
1 = physical layer
He proposes that the biggest opportunities for standardization are in levels 5 and 4. Middleware services are "mundane and transparent to the end user" and are not a product differentiator; therefore vendors should be amenable to their standardization.
Proposed requirements for robotics middleware:
R1: offer a platform- and vendor-independent standard interface to which a system integrator can write
R2: extendable components, in the OO sense of "programming by difference"
R3: real-time executable with minimum operating system support
R4: real-time "interact-able": behavior can be observed and changed at runtime, there need to be integrated log/playback/debug capabilities.
Three technical characteristics of a standard should be met:
Retain component definitions and connection concepts from UML and CCM
Support data-centric designs and communications

Be agnostic with regard to underlying communication mechanisms (CORBA, DDS, etc.)

Presentation - RFI Response: "Self-Configuring Smart Java Robits through Plug-and-Play I/O Boards" - Bruce Boyes (Systronix)

Bruce Boyes explained the importance of self-descriptive hardware (especially for swarm robots) and introduced the concept of XML-tagging. By assuming that most hardware (sensors & actuators) is I/O based, the idea is to develop a generic I/O driver common to all devices and differentiate the functionalities provided by the hardware not in the base code but by using standardized XML tags (similarly to the IEEE1451 standard) describing these functionalities at the I/O pin level. The recent drop of the price of memory chips makes this solution viable and an example of application of the concept using the Systronix JCX board was presented.

http://www.omg.org/docs/robotics/05-12-07.pdf

Presentation - RFI Response: "RSCA: Robot Software Communication Architecture" by Seongsoo Hong of Seoul National University

The National Robotics Project in Korea is predicated on the assumption, made at the level of the Minister of Information and Communication, that home service robots will become ubiquitous in homes in the near future. Their approach is to develop affordable robots with limited capabilities and distribute the task processing to high performance remote servers.
The main requirements for the software platform supporting the development of such robotic application are:
- Distributed nature of hardware and software
- Component-based software development
- Dynamic deployment and reconfiguration
- Real-time and QoS capabilities

After a detailed presentation of RSCA, the speaker did address the mapping of RSCA to the robotics technology and where RSCA should be extended to fulfill all the requirements for the development of a robotics application

http://www.omg.org/docs/robotics/05-12-08.pdf

Presentation - RFI Response: "Human Interface of the Robotic System RFI" by Soo-Young Chi

This presentation by ETRI specifies the "user recognition component of the human-robot interface."
The presenter proposes an interface comprising three main high level primitive functions in a user recognition component: - Enroll, Verify and Identify. After describing different possible distributed implementations strategies (actual recognition process on client or server side), the main data structures used during the recognition process were presented.

http://www.omg.org/docs/robotics/05-12-09.pdf

06 December, Tuesday

Informative Talk : “High Assurance Security and Safety for Robotics” - Joseph M. Jacob (Objective Interface Systems)

The speaker first listed the requirements related to security and safety in robotics applications as well as already existing standards. He then introduced the concept of MILS (Multiple Independent Levels of Security/Safety) in which the code running in privileged mode is reduced to a minimum kernel (dealing only

with security policies) which then becomes mathematically verifiable, non-bypassable, always invoked and tamper-proof
The presentation then covered a tool called Partition Communication System (PCS), developed for military/intelligence needs, which applies the MILS principles

http://www.omg.org/docs/robotics/05-12-10.pdf

Presentation - RFI Response: "Hardware Abstraction to the Robotic Systems RFI" - Seung-Ik Lee (ETRI)
In order to facilitate interoperability between robotic systems, ETRI has developed an abstract "Common Robot Interface Framework" (CRIF) which clearly defines types of hardware devices (wheels, sensors, cameras, head, ...), interfaces (CRIF includes a Robot API Layer with about 50 APIs), data types and coordinate systems (local and global). The framework makes a clear distinction between the Application Programmers APIs (which are hardware independent) and the Robot Platform Developers APIs (which are application independent). The communication mechanism between the 2 sets of APIs is interchangeable and transparent to the developers (ETRI provides a socket based implementation).
Dr. Lee said that one challenge is going to be that there are many types of devices in the robotics area, and more will appear, so it will be impossible to avoid custom interfaces. Olivier Lemaire (JARA) suggested that a useful first step would be to create a UML Profile for Robotics.

http://www.omg.org/docs/robotics/05-12-11.pdf

Presentation - RFI Response: "Response to RFI from NEC" - Yoshihiro Fujita (NEC)
The presenter relates on the difficulty of developing a robotic application (not a robot itself) of which users will not get bored after just a few weeks. To overcome this difficulty, more developers from different backgrounds should be involved (even the end-user itself) in the development, which implies the availability of an easy to use application development platform as well as customizable graphical tools. After describing the two fundamental concepts of robotic application programming (Actions and Behaviors), the presenter introduced the framework developed by NEC to support the development of Robotic Applications and based on XML interface definition and scripting.

http://www.omg.org/docs/robotics/05-12-12.pdf

Informative Talk : "The Player/Stage/Gazebo project: Open Source tools for robotics research" - Brian P. Gerkey (SRI International)
The SRI team has developed, and placed in open-source, a robotic control library called Player, which contains about 90 drivers for many different devices. They also have written a 2-D and a 3-D simulator, which respond to the same commands as the real robots. This makes this library appealing for development, in particular in universities, since it is possible to move back and forth seamlessly from simulation to actual execution. One of the machines that can be controlled is the robot version of the Segway transporter.

http://www.omg.org/docs/robotics/05-12-13.pdf
http://playerstage.sourceforge.net

Presentation - RFI Response: "Network Robot Platform for Information Sharing" - Ken-ichiro Shimokura (NTT)
The speaker introduced the 4 year-long Network-robot Project funded by the Ministry of Internal Affairs and Communication of Japan and which mission is to develop the core technologies necessary to a successful integration of the robotic technology into everyday's life, focusing on human like behaviours and interaction with humans. The 4 main themes of the project are :
- Human Behaviour Recognition
- Platform-mediated communication

- Service allocation and execution
- Human-robot Interaction

After describing the 4 themes, especially "Platform mediated Communication", for which the Field Data Markup Language (FDML) was used as a mean to abstract information provided by a robot (a pointer to more detailed information was unfortunately not provided), a video showing the early results of the project was played.

http://www.omg.org/docs/robotics/05-12-14.pdf

Presentation - RFI Response: "Human Robot Interaction in Network Robots" - Norihiko Hagita (ATR)

In the context of social robots and after presenting several cases of Human-Robot interaction and results of investigations made by ATR regarding that matter, the speaker expressed the need of the standardization of :

- basic behavior languages for different-type robots
- Software modules for social intelligence
- Interactive primitives and corpus from various sensor data

http://www.omg.org/docs/robotics/05-12-15.pdf

Presentation - RFI Response: "Network Robots Standardization Activity in Japan" - Miwako Doi (Toshiba)

After describing the Robotic Technology standardization effort in Japan and especially within the Network Robot Project, the speaker presented her views on the way to fulfill users' needs in the context of distributed service oriented robotic architecture, as well as the requirements and the difficulties met to develop such a system. A call was then made for leveraging the work that has already been done in the domains of ubiquitous computing and network sensors by trying to bridge the existing standards. Finally a case study cooperation between human and networked robots was described.

http://www.omg.org/docs/robotics/05-12-14.pdf

Chartering Robotics Domain Task Force and voting

Tetsuo Kotoku, from AIST, presented the roadmap for the SIG. There is an RFI in progress, for which 4 responses have been received, and at least two more are expected. To allow these additional responses, a motion was adopted to extend the reply deadline to 1/23/06, three weeks before the Tampa meeting.
Following the RFI, there seem to be potentially four RFPs on the roadmap:

1 - robot middleware for controllers
2 - robot middleware for specific applications
3 - robot middleware for common services
4 - robot middleware for common data structures

A quick survey showed that for each potential RFP, at least 6 to 7 participants would be interested in pursuing the activity, which was encouraging.

This was followed by a discussion concerning the Robotics DSIG becoming a Task Force. Tetsuo Kotoku described the pros and cons of such a move :

Pros : More visibility and publicity; Faster process of adoption
Cons : Collaboration with other Task Force no more necessary -> risk of isolation : need to actively maintain relations to other task forces (joint meetings ?); when voting a quorum is necessary

A motion was then made by Tetsuo Kotoku for our activity to move from the status of a Domain Special Interest Group (DSIG) to the one of a Domain Task Force (DTF). The motion was approved by all voters.
The charter of the Task Force has been adopted (same as the DSIG), the word "adapt" in the first bullet being replace by "adopt"

Following, a call for volunteers to become the new co-chairs of the task force has been made. Three volunteers stood up (Hung Pham (RTI)-Tetsuo Kotoku(AIST)-YunKoo Chung (ETRI)) and have been approved unanimously.

Finally, a call for volunteers for being the official OMG contact with other organizations has been made. (No new volunteer)

Next Meeting in Tampa
Monday : Steering Committee
Tuesday-Wednesday : Robotics DSIG plenary

ADJOURNED @ 17:40PM

Participants (Sign-in)

05 December, Monday (15 participants)

- Makoto Mizukawa (Shibaura Institute of Technology)
- Yun Koo Chung (ETRI)
- Seongsoo Hong (SNU)
- Saehwa Kim (SNU)
- Hung Pham (RTI)
- Claude Baudoin (Schlumberger)
- Seung-Ik Lee (ETRI)
- Olivier Lemaire (JARA)
- Carlo Cloet (RTI)
- Gerardo Pardo (RTI)
- Masayoshi Yokomachi (NEDO)
- Takashi Suehiro (AIST)
- Noriaki Ando (AIST)
- Tetsuo Kotoku (AIST)
- Jaesoo Lee (SNU)

06 December, Tuesday (28 participants)

- Miwako Doi (Toshiba)
- Takashi Suehiro (AIST)
- Duane Clarkson (John Deer)
- Saku Egawa (Hitachi)
- Makoto Mizukawa (Shibaura Institute of Technology)
- Noriaki Ando (AIST)
- Olivier Lemaire (JARA)
- Yoshihoro Fujita (NEC)
- Bruce Boyes (Systronix)
- Adam Howell (Lockheed Martin)
- John Hogg (Zeligsoft)
- Roger Burkhart (John Deer)
- Masayoshi Yokomachi (NEDO)
- Rick Warren (RTI)
- Yun Koo Chung (ETRI)
- Seung-Ik Lee (ETRI)

- Soo-Young Chi (ETRI)
- Takashi Tsuboushi (University of Tsukuba)
- Hung Pham (RTI)
- Henri Choi (RTI)
- Seongsoo Hong (SNU)
- Saehwa Kim (SNU)
- Jaesoo Lee (SNU)
- Ken Shimokura (NTT)
- Roy Bell (Raytheon)
- Virginie Watine (Thales)
- Gerardo Pardo (RTI)
- Tetsuo Kotoku (AIST)

07 December, Wednesday (28 participants)

- Takashi Suehiro (AIST)
- Makoto Mizukawa (Shibaura Institute of Technology)
- Masayoshi Yokomachi (NEDO)
- Seung-Ik Lee (ETRI)
- Noriaki Ando (AIST)
- Bruce Boyes (Systronix)
- Claude Baudoin (Schlumberger)
- Jaesoo Lee (SNU)
- Joseph Jacob (Objective Interface)
- Yoshihoro Fujita (NEC)
- Stan Schneider (RTI)
- Tetsuo Kotoku (AIST)
- Yun Koo Chung (ETRI)
- Soo-Young Chi (ETRI)
- Miwako Doi (Toshiba)
- Saku Egawa (Hitachi)
- Ken Shimokura (NTT)
- John Hogg (Zeligsoft)
- Adam Howell (Lockheed Martin)
- Rick Warren (RTI)
- Olivier Lemaire (JARA)
- Saehwa Kim (SNU)
- Takashi Tsuboushi (University of Tsukuba)
- Roy Bell (Raytheon)
- Juergen Boldt (OMG)
- Regis Vincent (SRI International)
- Brian P. Gerkey (SRI International)
- Hung Pham (RTI)

Prepared and submitted by Olivier Lemaire with the assistance of Seung-Ik Lee, Claude Baudoin, Masayoshi Yokomachi and Makoto Mizukawa.

Robotics-DTF/SDO-DSIG
Plenary Meeting

February 14, 2005

Tampa, FL, USA

Grand Hyatt Tampa Bay

Audubon B, 1st FL

Approval of Burlingame Minutes

- Ask for a volunteer (minutes taker)
  - Saku Egawa (Hitachi)
  - Soo-Young Chi (ETRI)
- Burlingame Minutes review

[Robotics] The deadline of Robotic Systems RFI. was extended. We had 3 special talks (James J. Odell, Joseph M. Jacob, Brian P. Gerkey) and 9 RFI response presentations.

[SDO] We were waiting response of RTCs RFP.

Document Number

robotics/2006-02-04 Final Agenda
robotics/2006-02-05 Steering Committee presentation
robotics/2006-02-06 Atlanta Meeting Minutes [approved]
robotics/2006-02-07 Opening presentation
robotics/2006-02-08 Robotics-DSIG Roadmap
robotics/2006-02-09 **Saku Egawa** presentation
robotics/2006-02-10 **Abheek Kumar Bose** presentation
robotics/2006-02-11 **Hiroyuki Nakamoto and Masayuki Nagase** presentation
robotics/2006-02-12 **Tomoki Yamashita** presentation
robotics/2006-02-13 **Wonpil Yu** presentation
robotics/2006-02-14 **Takashi Tsubouchi** presentation
robotics/2006-02-15 **Wonpil Yu** (Nakuju Doh) presentation
robotics/2006-02-16 **Olivier Lemaire** presentation
robotics/2006-02-17 **Lloyed Spencer** presentation
robotics/2006-02-18 **Matt Long** presentation
robotics/2006-02-19 **Wataru Inamura** presentation
robotics/2006-02-20 **Jerry Bickle** presentation
robotics/2006-02-21 **Virginie Watine** presentation
robotics/2006-02-22 **Seung-Ik Lee** presentation
robotics/2006-02-23 **Fumio Ozaki** presentation
robotics/2006-02-24 Summary of RFI responses
robotics/2006-02-25 Query report
robotics/2006-02-26 Publicity Activity proposal
robotics/2006-02-27 RFI summary at MARS joint meeting presentation
robotics/2006-02-28 DTC Report Presentation
robotics/2006-02-29 Tampa Meeting Minutes – Draft

Review Agenda

Tuesday, Feb.14, 2006 **Audubon B, 1st FL**

08:40-09:00 Welcome and Review Agenda

09:30-11:30 RTCs initial submission review [MARS]

13:00-15:00 RFI response presentation (3)

15:20-18:00 RFI response presentation (4)

Joint Meeting with MARS/RTESS
Thursday, Feb.16, 2006
14:00-15:00 (White Ibis, 1st FL)

Review Agenda

Wednesday, Feb.15, 2006 **Pelican, 2nd FL**

08:10-09:30 RFI response presentation (2)

09:30-10:20 Special Talk (CRASAR, Univ. of South Florida)

11:40-12:00 RFI response presentation (2)

14:00-15:20 RFI response presentation (2)

15:40-16:20 RFI response presentation (1)

16:20-17:40 RFI summary and discussion

17:40-18:00 Next meeting, etc.

18:00 Adjourn

Joint Meeting with MARS/RTESS
Thursday, Feb.16, 2006
14:00-15:00 (White Ibis, 1st FL)

Publicity Activities

- 4 page fly sheet
  Draft of Abheek@ADA Software
  Abheek@ADA Soft, Olivier@AIST, Chung@ETRI, Yokomachi@NEDO
- RoboBusiness event
  June 20-21, 2006, Pittsburgh, PA, USA
  http://www.robobusiness2006.com/
  no volunteer (ask for OMG staff)
- IROS2006 Workshop
  October 9-15, Beijing, China
  http://www.iros2006.org/
  Kotoku@AIST, Chung@ETRI, Mizukawa@Sibaura-IT

Potential Working Groups

- Infrastructure WG:
  Rick@RTI, Ando@AIST
- Tool WG:
  Abheek@ADA software
- Service WG:
  Chi@ETRI, Olivier@AIST
- Profile WG:
  Lee@ETRI, Bruce@Systronix

Will be chartered at the St. Louis meeting

contact between the related organizations

Call for volunteers

One page PowerPoint presentation

post it robotics@omg.org 3weeks before the meeting

Two-minute presentation at the meeting

- JAUS: Hui-Ming Huang (NIST)
- ORiN: Makoto Mizukawa (Shibaura Institute of Technology)
- RTmiddleware: Tetsuo Kotoku (AIST)
- KIRSF: Yun Koo Chung (ETRI)
- NRF: Miwako Doi

FYI:

OMG official liaisons will have a privilege of OMG resource access.

Contact to the Liaisons Sub Committee, please.

Next Meeting Agenda

April 24-28, 2006 (St. Louis, MO, USA)

Monday :

Steering Committee

Tuesday-Wednesday :

Robotics-DTF Plenary Meeting

- RTCs RFP revised proposal progress report (MARS/SDO joint meeting)
- RFI response presentation
- Roadmap discussion
- Contact reports

Roadmap for Robotics Activities

robotics/06-02-08 & sdo/06-02-04

Item	Status	Burlingame Dec-2005	Tampa Feb-2006	St. Louis Apr-2006	Boston Jun-2006	Anaheim Sep-2006	DC Dec-2006	TBA Feb-2007
Robot Technology Components RFP (SDO model for robotics domain)	In Process		Initial Submittion	Pre-review	Revised Submittion	issue		
SDO model for xxx Domain	no plan			discussion	draft RFP	RFP		Initial Submittion
Charter on Robotics WG in SDO	done Oct-2004							
Robotic Systems RFI [Robotics: Initial Survey]	In Process	Response Presentation	Response Presentation	Response Presentation	review whitepaper	Whitepaper		
Flyer of Robotics-DTF [Publicity Sub-Committee]	Planned		discussion	issue ver.1.0				
Potential RFPs [Infrastracture (tentative) WG]	Planned		Grouping	Chartering WG	draft RFP	RFP		Initial Submittion
Potential RFPs [Services (tentative) WG]	Planned		Grouping	Chartering WG	draft RFP	RFP		Initial Submittion
Potential RFPs [Tools (tentative) WG]	Planned		Grouping	Chartering WG	draft RFP	RFP		Initial Submittion
Potential RFPs [Profiles (tentative) WG]	Planned		Grouping	Chartering WG	draft RFP	RFP		Initial Submittion
etc…	Future			Grouping	Chartering WG	draft RFP	RFP	
Charter on Robotics TF	done Dec-2005	issued						
Charter on Robotics SIG	done Feb-2005							
Robotics Information Day [Technology Showcase]	done Jan-2005							

OMG Technical Meeting, Feb. 14, 2006

robotics2006-02-09

Hitachi's needs for robotic system standards

Saku Egawa

Mechanical Engineering Research Laboratory

Hitachi, Ltd.

2 Purpose of Presentation

- Show our needs for robotic system standards.
  - What field do we expect to be standardized first?
  - How should it be?
  - Show examples of our robotic systems.

Contents

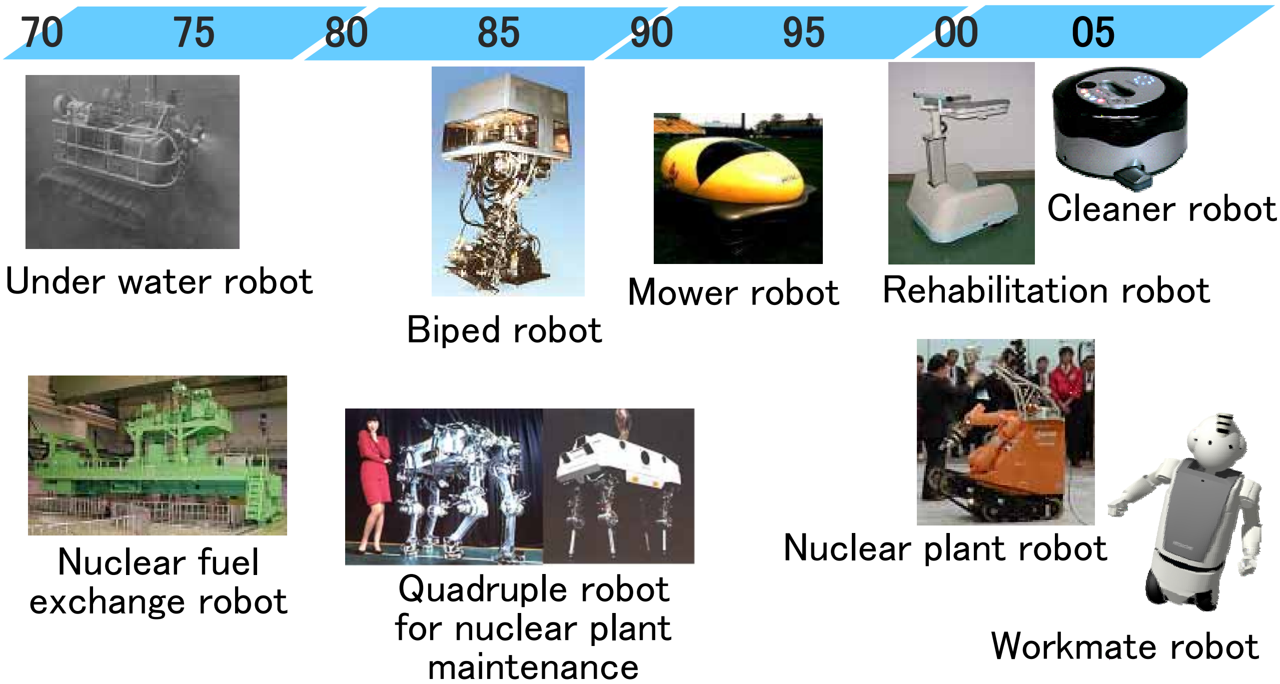
About HITACHI

- Global electronics company
- Wide business field including:
  - Systems for industry
    - Energy, Transportation, Medicine, Manufacturing, Construction…
  - Consumer products
    - Home appliance, Digital media device
  - Devices
    - Semiconductor, Hard disk drive, Automotive
  - Information Technology
    - IT system, Storage

About merl

- Corporate research center specialized in mechanical engineering.
- 350 employees.
- Participates in almost all Hitachi business.
- Base technology include:
  - Simulation & analysis of mechanical systems
  - MEMS (Micro Electro Mechanical Systems)
  - Robotics and mechatronics

Advanced Robots R&D History

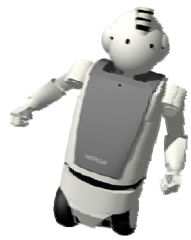


Objective of Advanced Robot R&D

- Create new technology
  - Control, Sensing, Embedded system
- Create future products
  - Security robots, Housekeeping robots, Rehabilitation robots, ….
- Bring new technology to existing fields.

Current projects:

Cleaner robot



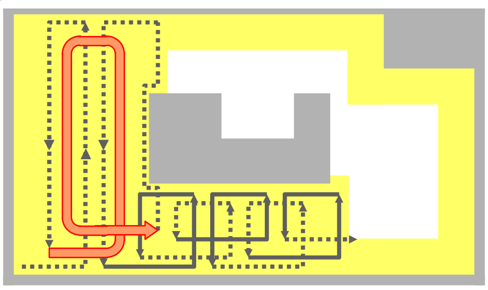
Workmate robot

Cleaner Robot

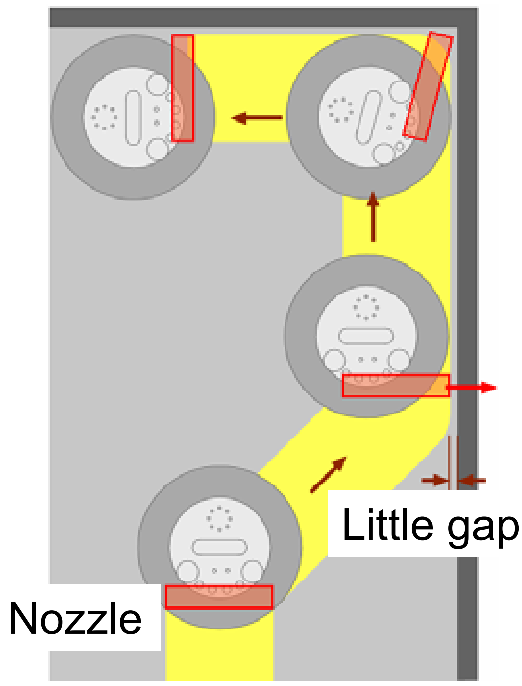
- Near future product

Compact size

Spiral motion



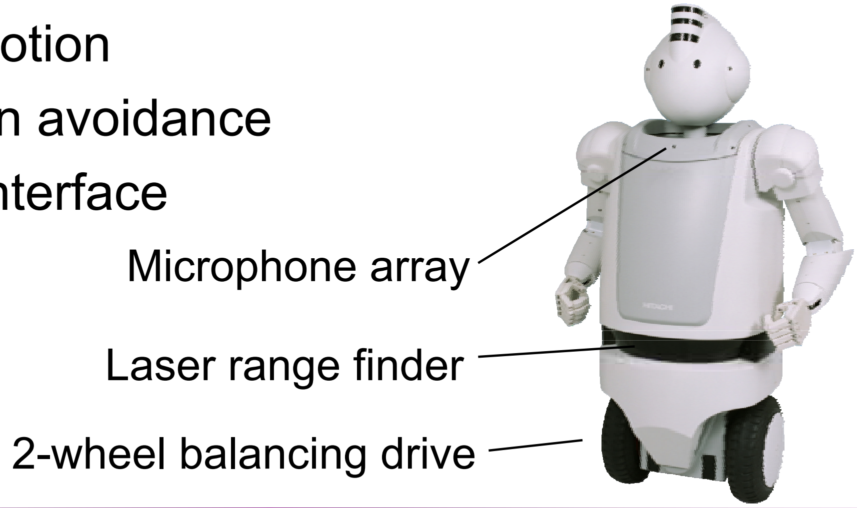
Movable nozzle



9 Workmate Robot EMIEW

Excellent Mobility and Interactive Existence as Workmate

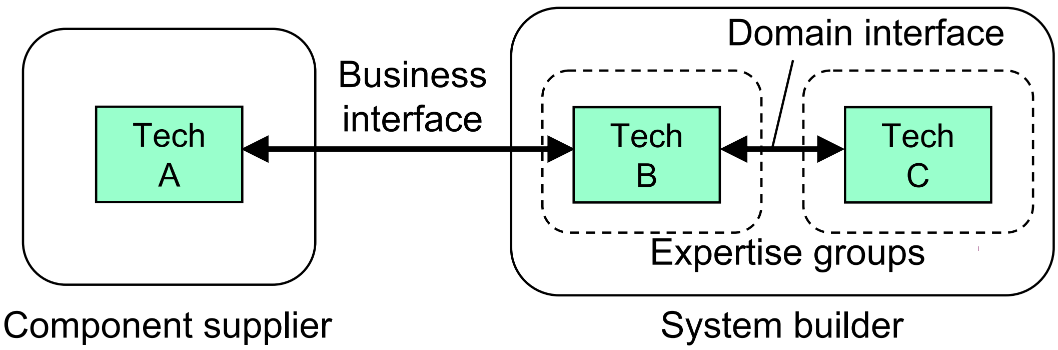
- Concept model for future product
  - Shown at Expo 2005 Aichi Japan
- Platform for new technology research
  - Agile motion
  - Collision avoidance
  - Voice interface



10 Contents

11 Where are Standards Needed?

- Business interface
  - Interface between companies
  - Clear need for open standard
  - Product design defines interface
- Domain interface
  - Interface between different expertise groups
  - Concerns productivity
  - In-house standard or open standard?



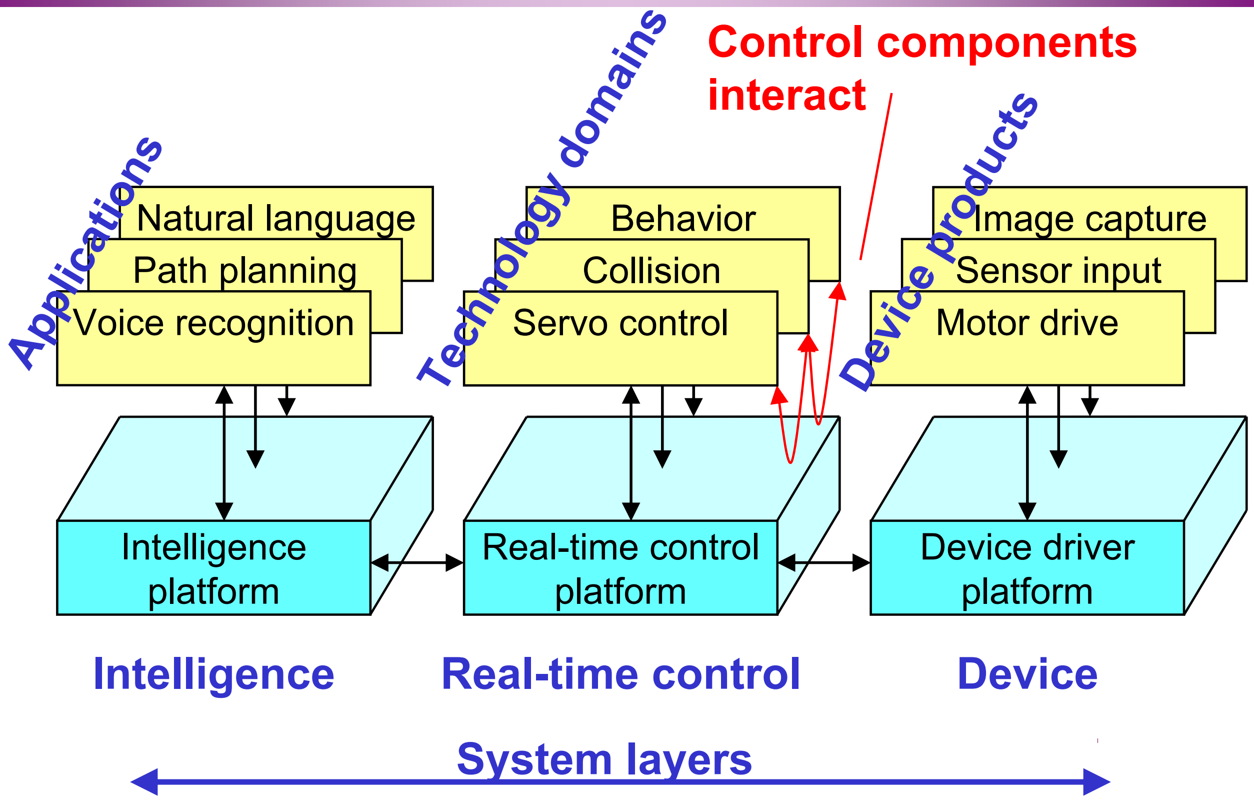
12 In-house vs. Open Standard

- Develop in-house standard
  - Can be optimized for the business area.
  - Better performance with less computing power.
  - But limitations…
    - Human resource, cost, scalability, support.
- Adopt open standard
  - Good design at low cost
  - Wider application area
  - Longer lifetime
  - Good solution for:
    - Difficult technical problem
    - Common request

13 Characteristics of Robotics

- Business environment
  - Tightly integrated product
    - Hardware depends on applications
    - Hardware and software is tightly connected
  - Component business is under development
- Technology problems
  - Multiple domains of knowledge needed
    - control, voice recognition, distributed computing, network service, …
  - Integration into real-time embedded system
  - Limited resource
  - Safety for physical system

14 Robotic Software Model



15 Priority Order for Standardization

1	Real-time Control	• Platform for integration needed • Technologies are being established
2	Device	• Component business is under development
3	Intelligence	• In research stage • Depends on application

16 Needs for Control System Standard

- Work with limited processing power
  - Efficient execution mechanism for real-time distributed tasks.
- Cover range of products
  - Flexibility to provide scalability
- Implementation & verification support
  - Support for control problems
    - Multi-rate, arbitration,…
  - Link with design tools
  - Simulation support
  - Logging mechanism

2-Layer Structure

- Real-time system middleware –
  – basic component architecture model
  - Asynchronous messaging
  - Real-time support
  - Scalability mechanism
- Robotic controller framework
  - Multi-rate sampling
  - Arbitration of requests for devices
  - Import controller model (servo)
  - Import sate machine model (behavior)

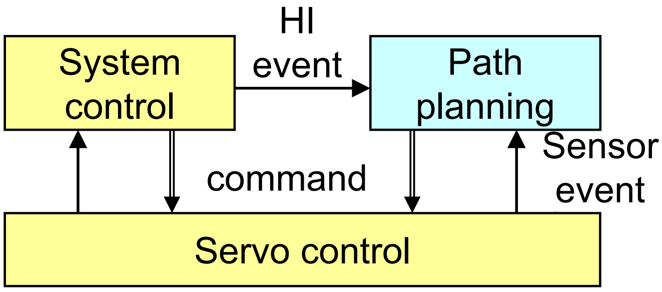
Contents

Cleaner Robot

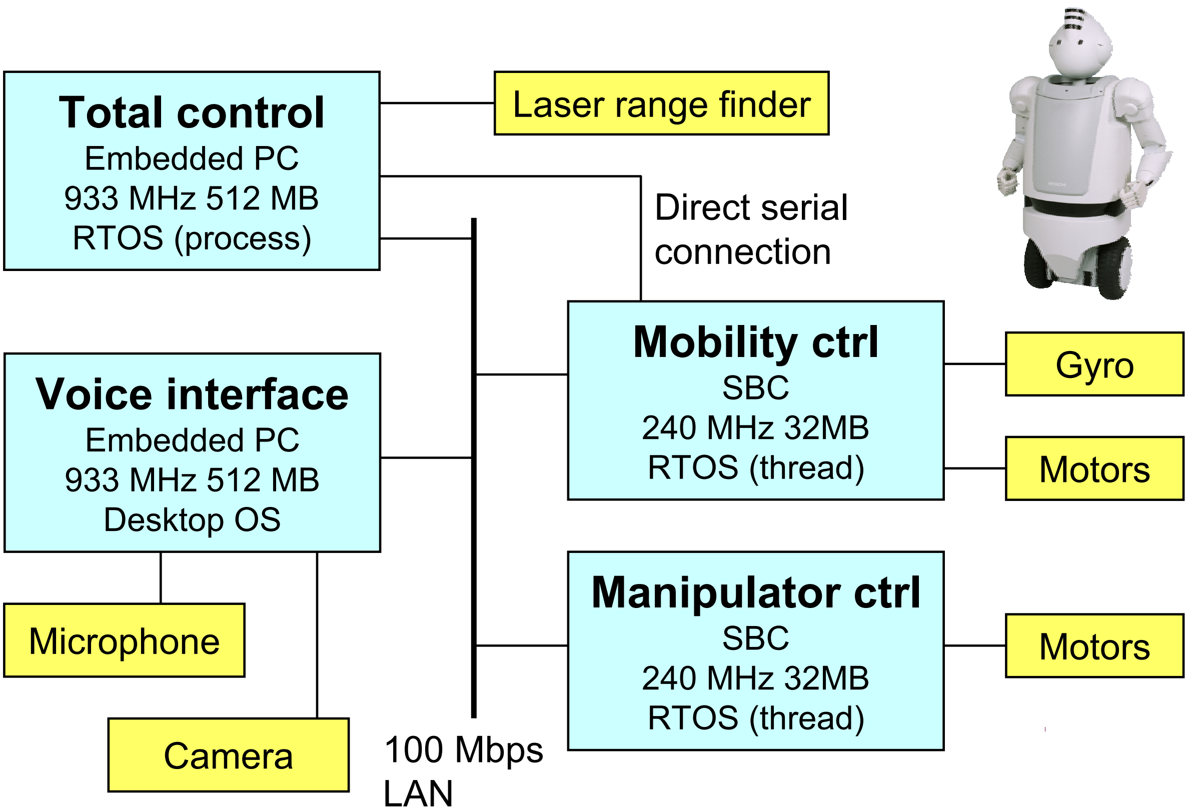
- Hardware
  - "Rich system" as a home appliance

CPU	32 bit, 28 MHz RISC
Memory	1 MB ROM, 2 MB RAM
Sensor	3 IR + 1US range sensors, Bumper SW 3 Encoders (L&R wheels, nozzle)
Actuator	5 DC motors (L&R wheels, nozzle, brush, fan)
Interface	Buttons, Remote commander

- Software
  - OS: none
  - Periodic & Event-driven

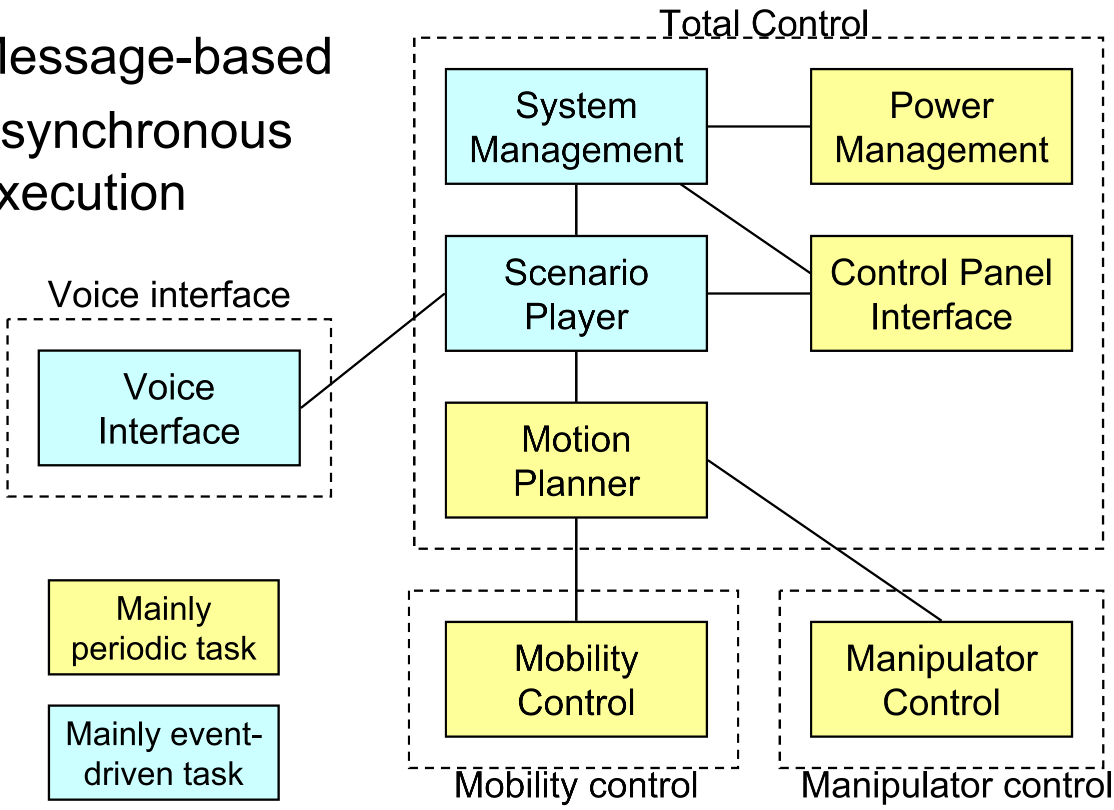


Workmate Robot – Hardware



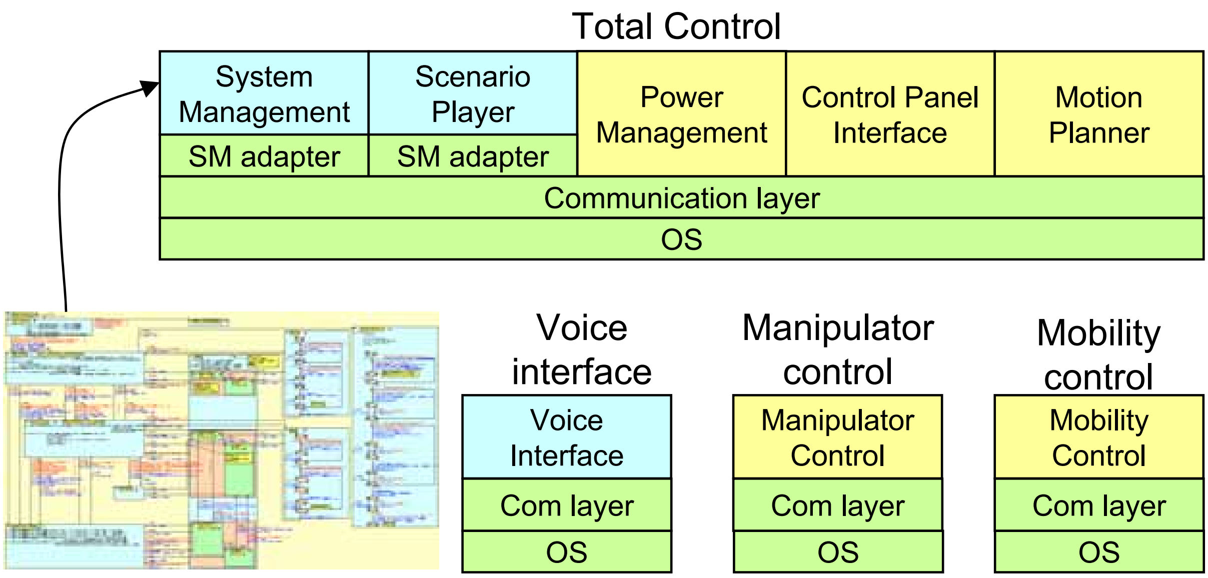
Workmate Robot – Software

- Message-based
- Asynchronous execution



Workmate Robot – Architecture

- Simple in-house communication layer
  - Selects path from IPC, UDP, serial
- State machine modeler generates code



Plan for Architecture

- Unified model for robotic products
  - Covers from a low-cost home appliance to rich industry use robot.
- Compatible with standards
  - Adopt existing standard.
  - Cooperate with Robotics TF

Conclusion

- Control system standard has our most priority.
- Expect Robotics TF to build scalable standard.
- We may contribute by providing information on practical needs especially in low-cost products.

OMG Technical Meeting, Tampa, Feb. 14, 2006
Copyright © Hitachi,Ltd. 2006 All rights reserved

Towards Plug & Play Robotics

Response to RFI

ADA Software Group, India

OMG Robotics DTF
Technical Meeting, TAMPA, Florida
Wednesday, February 15, 2006

ADA SOFTWARE technology stimulant

Fast Start

A Specialized Resource in OBJECT TRANSITION

robotics2006-02-10

www.adasoftware.com | abheek@adasoftware.com

About ADA Software Group

ADA SOFTWARE GROUP

A Specialized Resource in OBJECT TRANSITION

- Founded in 1991 simultaneously in India & Germany
- OMG International Partner since 1994
  - CEO, DK Bose - OMG International Partner
  - Active participation in UML standardisation
  - Currently the only MDA Fast Start QSP in India
  - Represents OMG interests in the Indian Subcontinent
  - Organised various events related to OMG Interests
    - Modeling 2006: OMG's India-Singapore Series
    - CORBA Competence Center
    - "THINK OBJECTS": Mentoring Legacy Professionals to Object World
    - Introduced Distributed Computing to India's premiere Management Institute – IIM Calcutta
- Entered the OMG Robotics scenario in November 2005

Robots: Here, there… Everywhere!

Manufacturing Industries

Research: RoboCup

Education

Entertainment

Exploration and Rescue

Human Assistant Systems

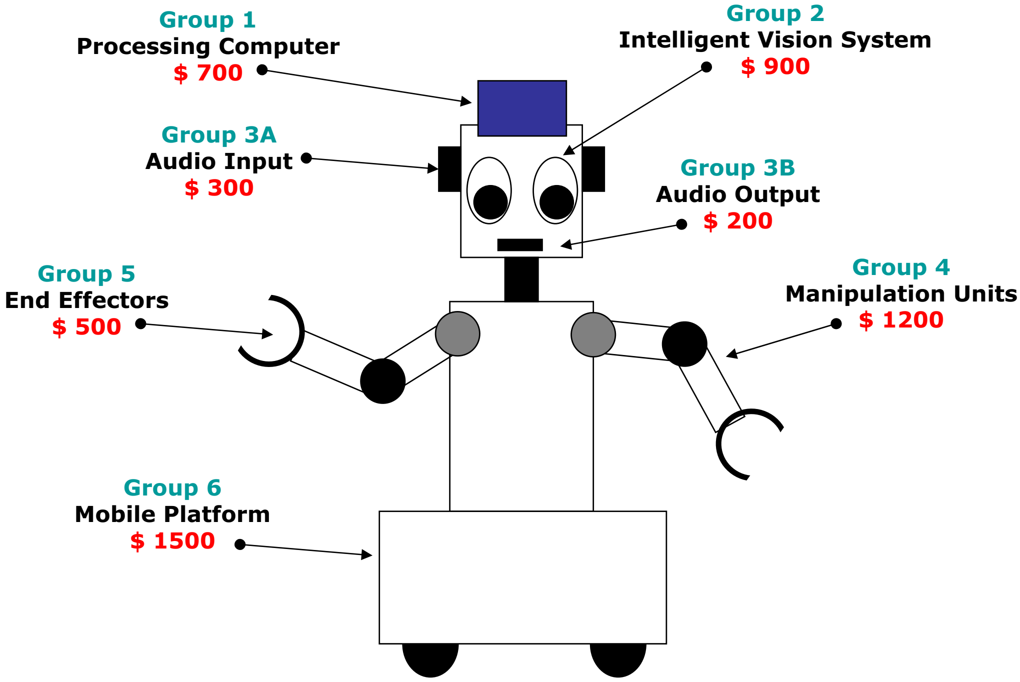
Cleaning & Housekeeping

Robotics @ ADA Software

- Prime focus: "intelligent components"
  - Distributed set of robot components having basic intelligence
  - Should seamlessly integrate / combine to form complex systems
- Target: To merge Robot Technologies with OMG Technologies by
  - Identifying the common merging grounds
  - Applying directly OMG Technologies on Robot System Development
  - Develop working prototypes
  - Collaboratively work with other organisations within OMG

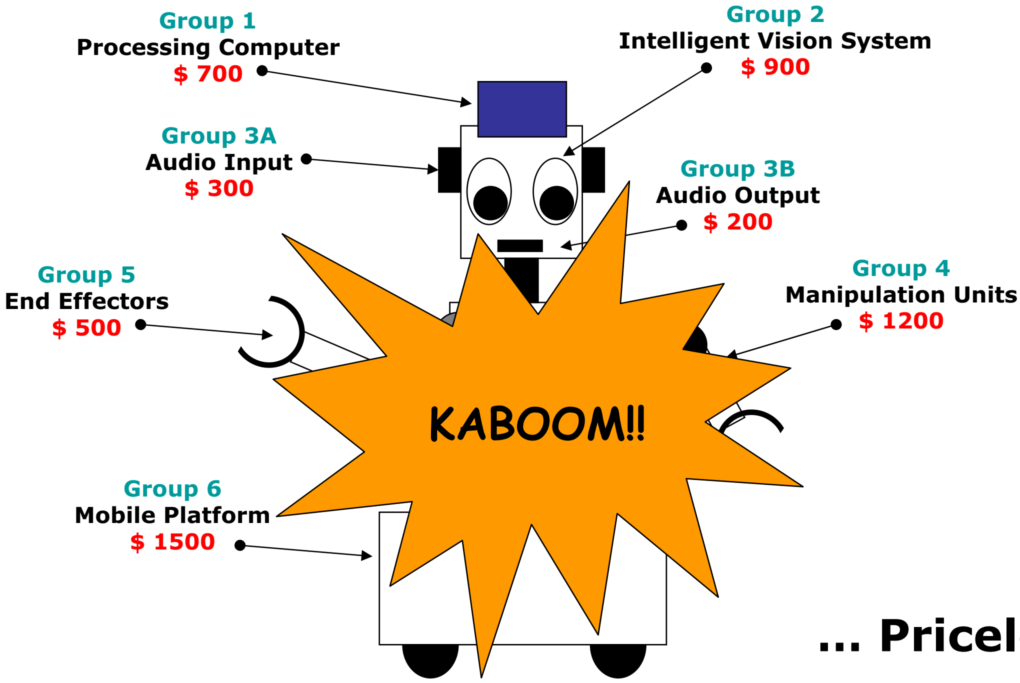
Intelligent Components
- the benefit -

ADA SOFTWARE GROUP
A Specialized Resource in OBJECT TRANSITION



Intelligent Components
- the challenge -

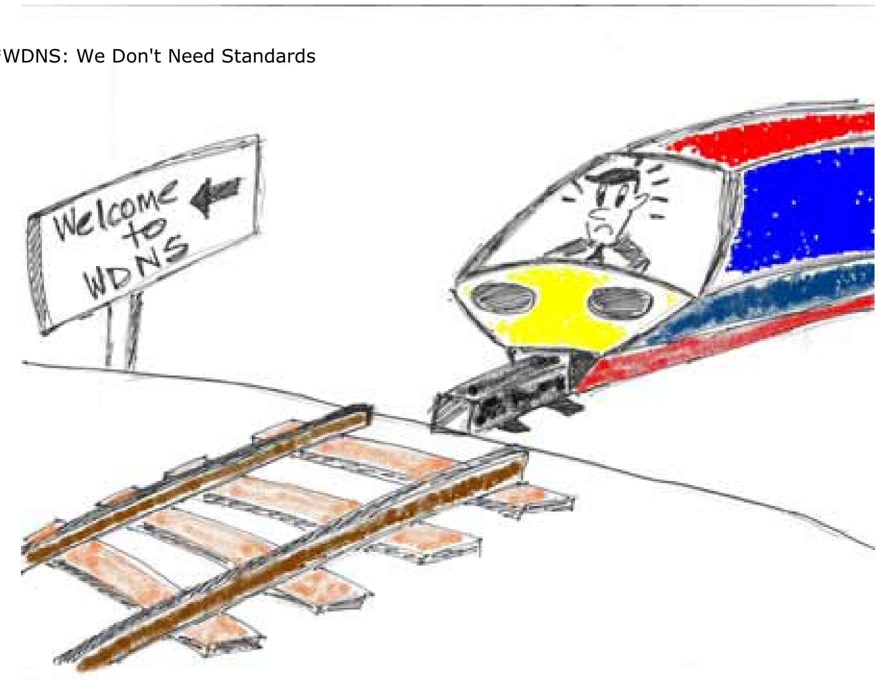
ADA SOFTWARE GROUP
A Specialized Resource in OBJECT TRANSITION



... Priceless!

Standards: A MUST!

*WDNS: We Don't Need Standards



Our Experiences

The VolksBot Project

Our Experiences

ADA SOFTWARE GROUP

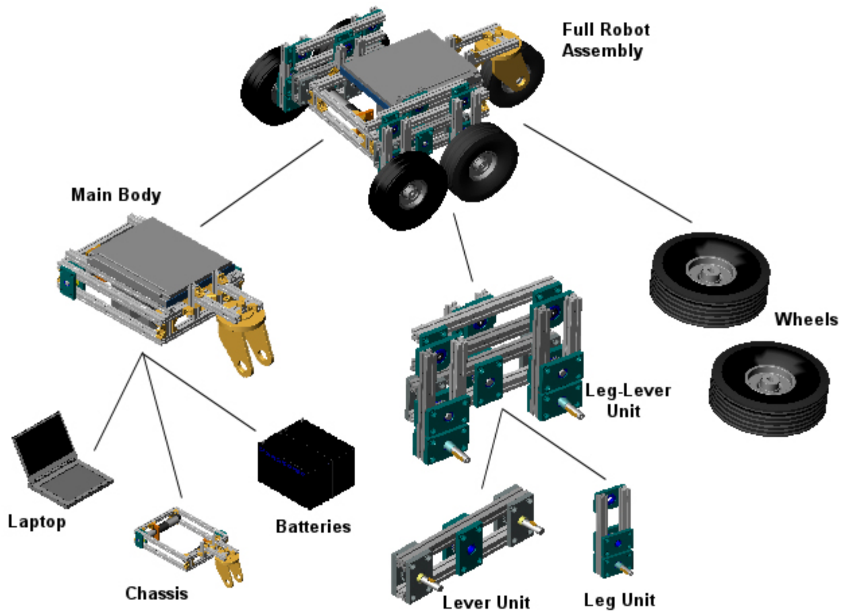
The VolksBot Project

- Research project undertaken in the Fraunhofer Institute for Autonomous Intelligent Systems, Germany
- Uses mainly standard off-the-shelf reusable components
- Components build Modules: Modules build robots
- Up to 6 different Variants of robots possible using the ***same*** modules
  - 2 indoor variants (differential drive & omni directional drive)
  - 4 outdoor variants
- Follows the modular approach in both Hardware and Software
  - Hardware modules include drive units, chassis frames, standard wheels etc
  - Electronics modules include Motor Controller (TMC) and IO Module (M-Board)
  - Software is a Visual Programming System (IConnect)
- More information on www.volksbot.de

Our Experiences

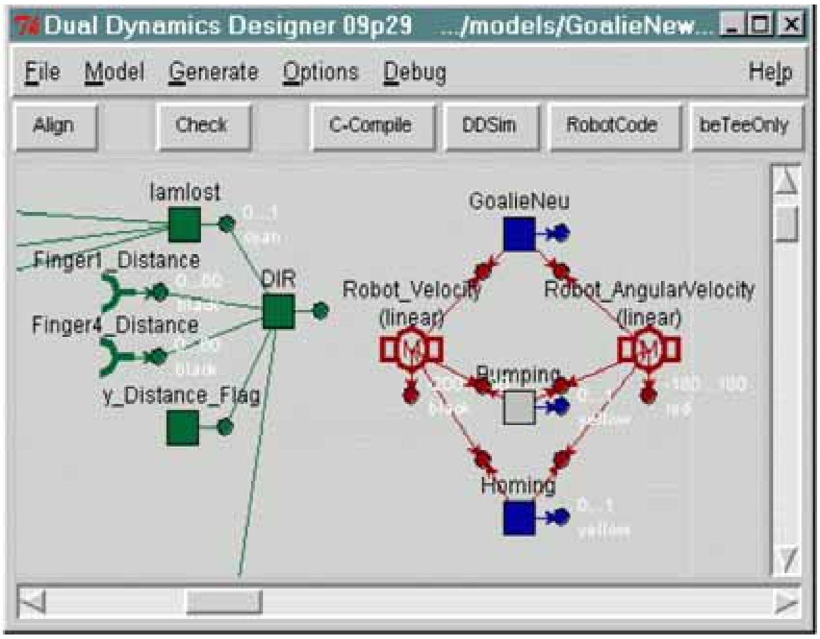
ADA SOFTWARE GROUP

The VolksBot XT: Advanced Mobility



ADA SOFTWARE GROUP
A Specialized Resource in OBJECT TRANSITION

Modelling Robot Intelligence: Dual Dynamics (DD) Designer



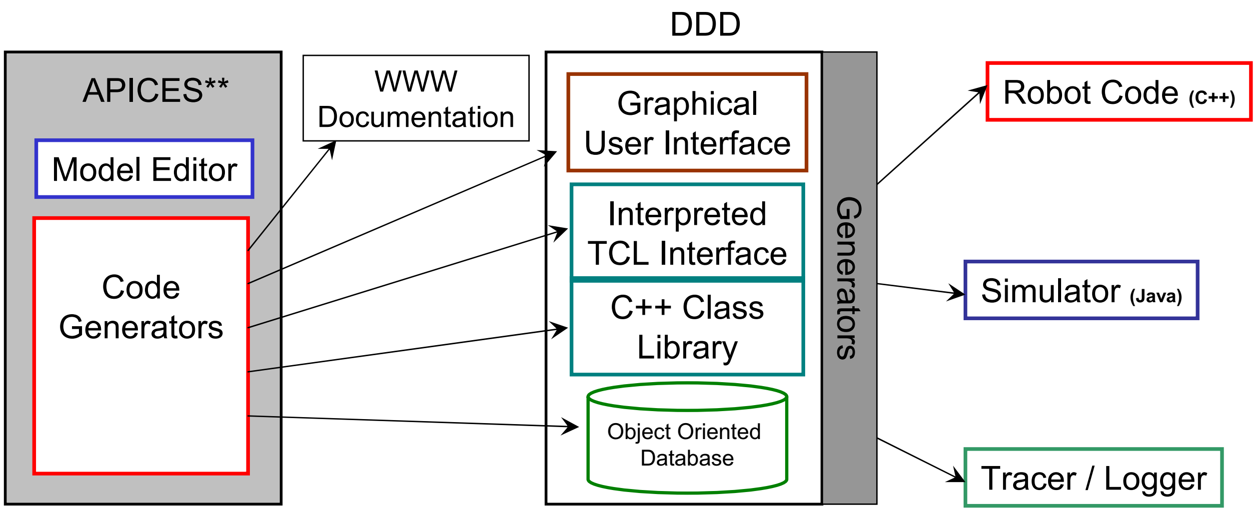
- Robot Behaviour Modelling
- Models include:
  - Sensor Inputs
  - Sensor Filters
  - Behaviour Hierarchy
  - Actuator Modules
- Developed by Fraunhofer AIS
- Based on APICES, a locally developed architecture
- www.ais.fhg.de/BE/env

Our Experiences

ADA SOFTWARE GROUP
A Specialized Resource in OBJECT TRANSITION

Modelling Robot Intelligence: Dual Dynamics (DD) Designer

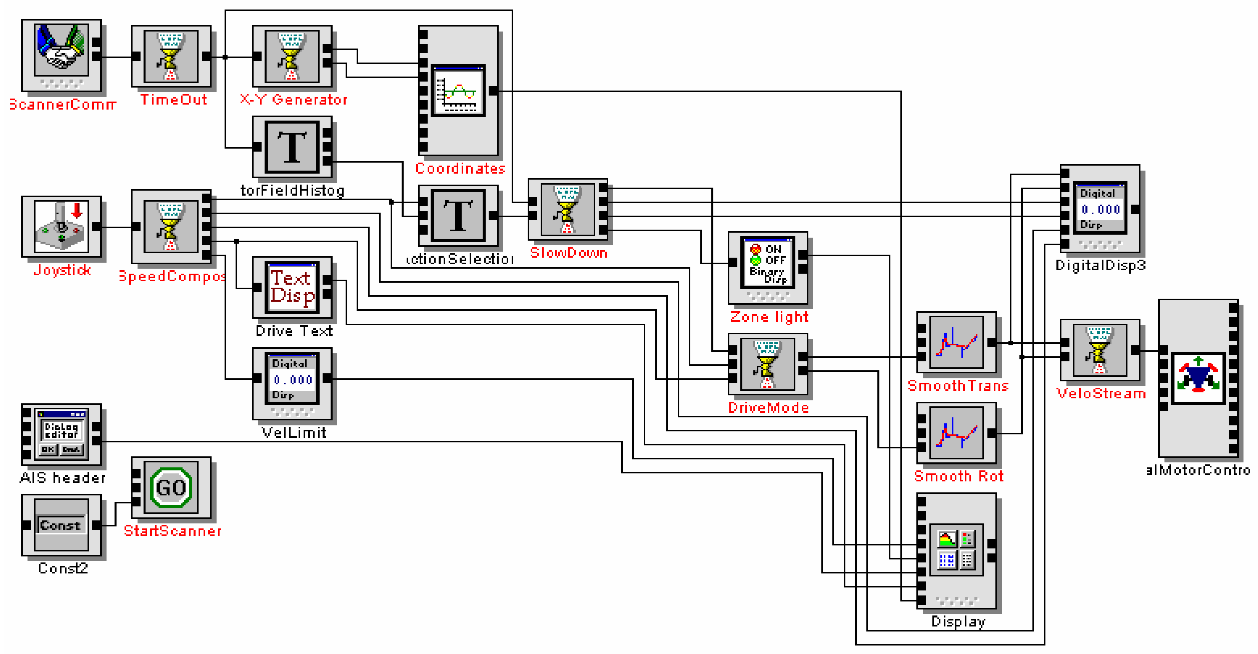
The working behind the DD Designer



** www.ais.fraunhofer.de/BE/apices/

Our Experiences

Towards Executable Modelling: IConnect



ADA SOFTWARE GROUP
A Specialized Resource in OBJECT TRANSITION
ADA SOFTWARE technology stimulant
Fast Start

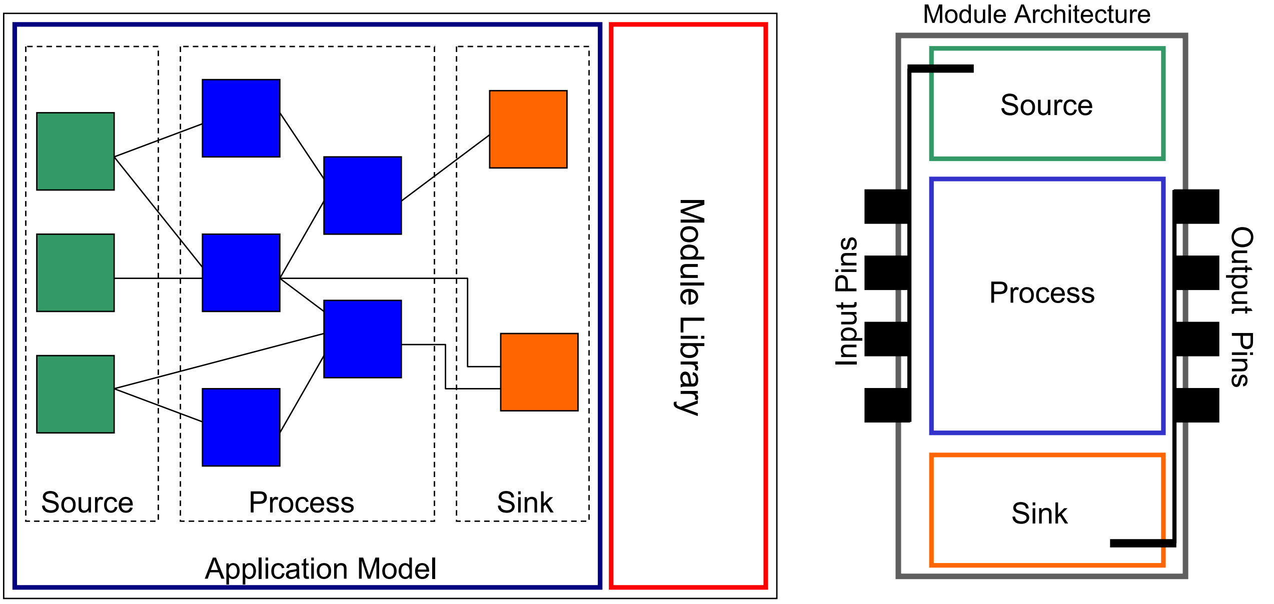
Our Experiences

Towards Executable Modelling: IConnect

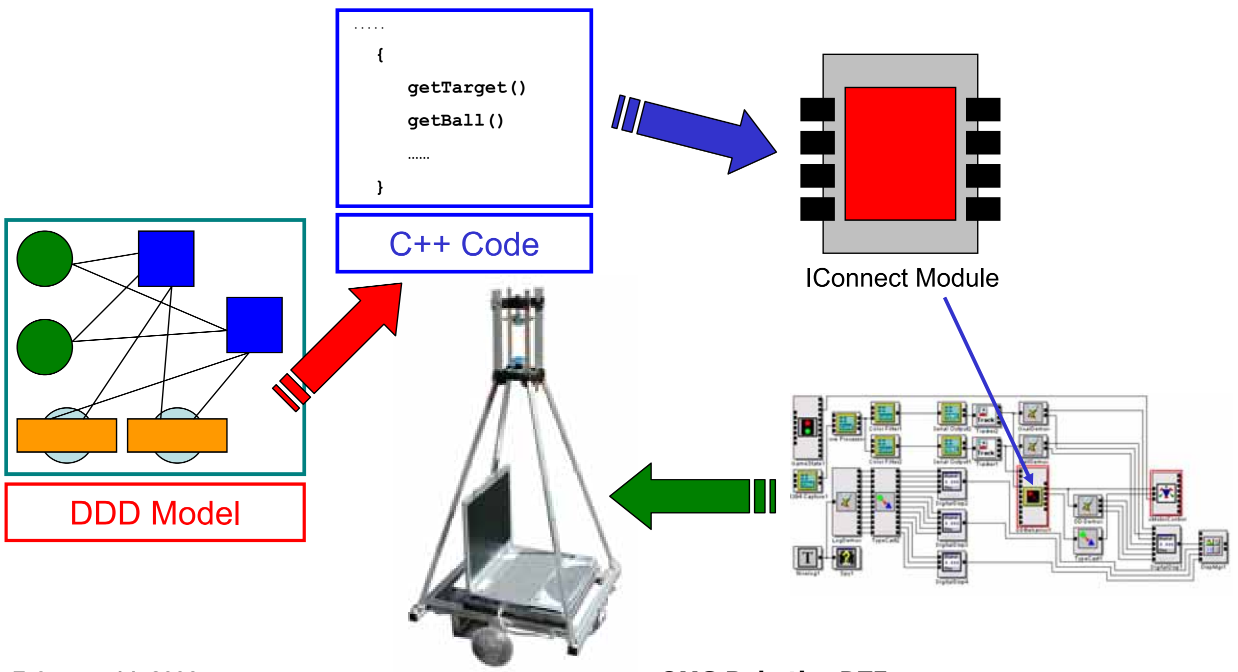
- Software tool for fast development of various applications including:
  - Data Acquisition
  - Process Control
  - Industrial Automation
  - Quality Control
  - Image Processing
  - R&D and Education
- Developed my Micro-Epsilon, Germany
- Every application "Graph" is composed of executable modules
  - pre – compiled
  - clearly defined interfaces (input / output pins)
- Modules fall under three categories
  - Source (e.g.. Sensors)
  - Process (e.g.. Behaviours)
  - Sink (e.g.. Actuators)
- Various applications possible by module recombination
- Executes in real time under Windows
- More on www.micro-epsilon.de/de-en/Navigation/Software/ICONNECT/

Towards Executable Modelling: IConnect

IConnect Architecture



Model Based Development of an Autonomous Robot



Fraunhofer AIS

Related Projects

DESIRE PROJECT

Cooperation Project in connection with "Deutsche ServiceRobotik" initiative

Project Focus:

- Methodical principles of service robotic design and engineering
- Systems for practical applications in particular everyday life performance capability
- Fraunhofer AIS is responsible for providing a standardized robot control architecture.

Fraunhofer AIS

Related Projects

CONTROL DESIGNER FRAMEWORK

A MDA approach, based on Eclipse

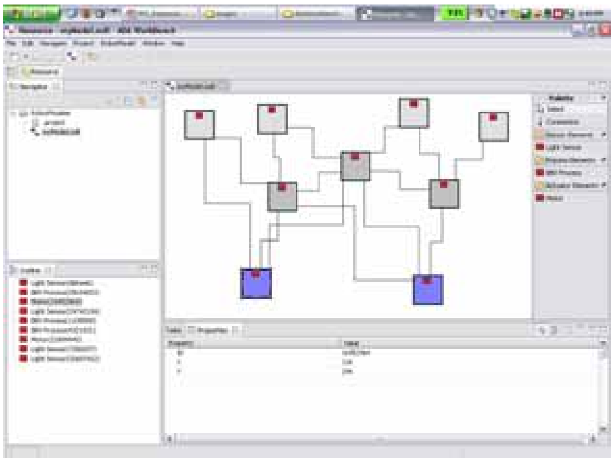
Project Focus:

- Eclipse based Framework using EMF and GEF
- Ability to transform any robot control model (PIM) to any language (PSM)
- Meta-Framework easily extended using Eclipse plugin technologies
- Users can develop their own plugins (PIM) and integrate them into the framework.
- Author: Peter Schoell, Fraunhofer AIS www.ais.fhg.de/~schoell/designer.html

Current Work in Progress

ROBOT MODELLER

Extending the Control Designer Framework



Project Focus:

- Extending the abilities of the CDF
- Models develop models
- Nested Source-Process-Sink architecture
- Modelling of component behaviours
- Ability to develop PSMs for
  - Simulation
  - Embedded Components
  - Complete Robot Systems

Thank You!

Abheek Kumar Bose,
Robotics Division,
ADA Software Group,
India
abheek@adasoftware.com

SEC's Approach to the Standardization of Robotic Systems

Robotic DTF
OMG Technical Meeting in Tampa
Feb. 14, 2006

Systems Engineering Consultants

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

Ⅰ. Introduction

In this presentation,

I. Introduction (this part ☺)
II. About SEC
III. Our Experience & Advantage
IV. Our Point of the Standardization of Robotic Systems
V. Conclusion

Ⅱ. About SEC

I. Introduction

II. About SEC

III. Our Experience & Advantage

IV. Our Point of the Standardization of Robotic Systems

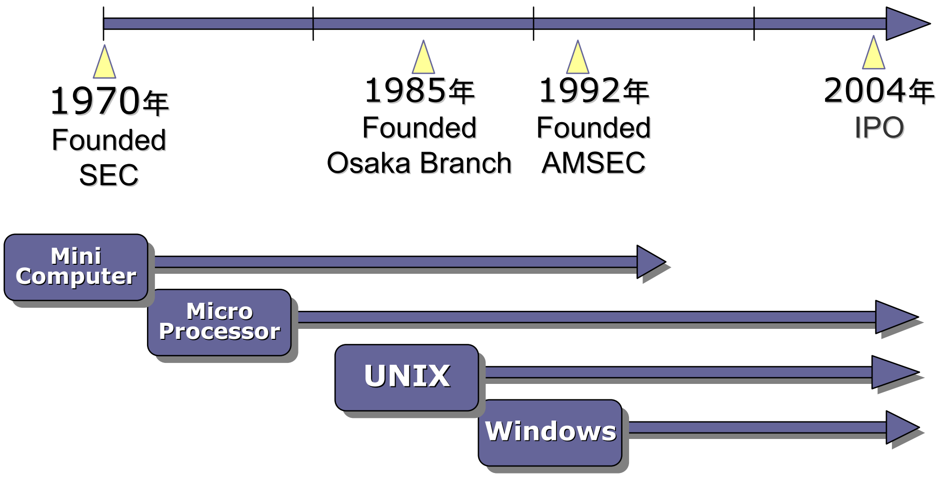
V. Conclusion

Ⅱ. About SEC

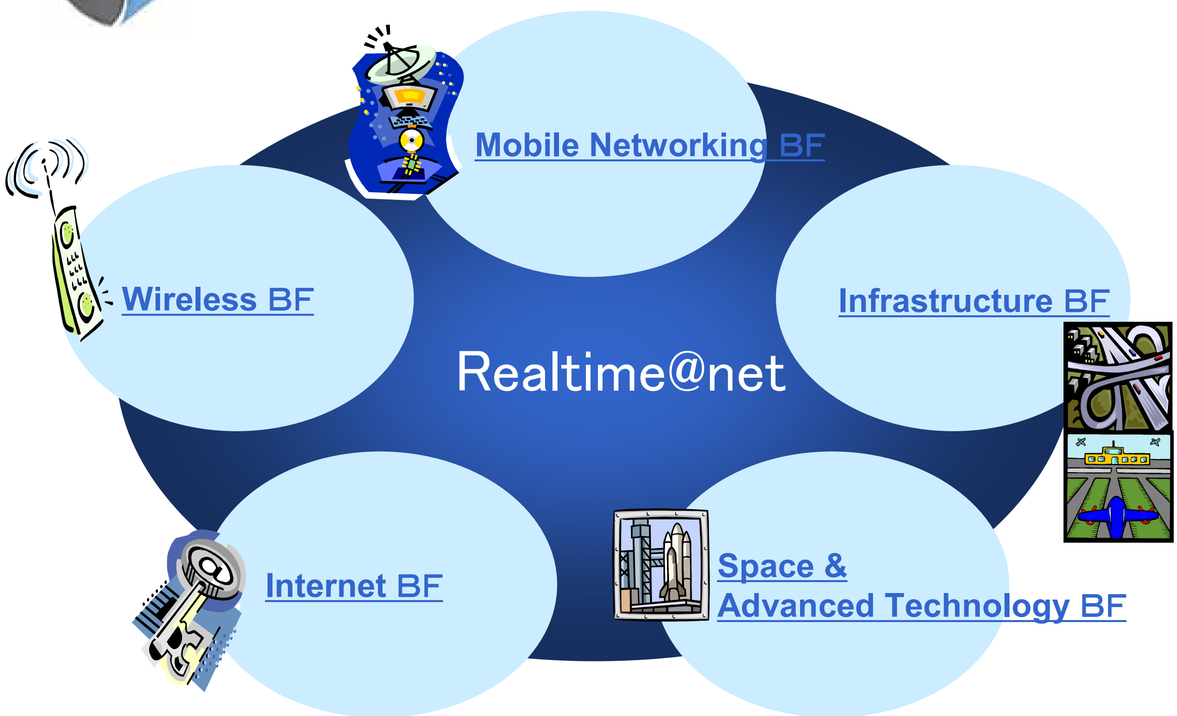
In this Section,

1. History of SEC
2. Business Target

1. History of SEC



2. Business Target



Ⅲ. Our Experience & Advantage

I. Introduction
II. About SEC
III. **Our Experience & Advantage**
IV. Our Point of the Standardization of Robotic Systems
V. Conclusion

Ⅲ. Our Experience & Advantage

In this Section,

1. Our Experience
2. Technology Advantage
3. Advantage in Robotic Systems

1. Our Experience

1)Satellite & Spacecraft

2)Cellular Phone

3)Internet Systems

1. Our Experience

1)Satellite & Spacecraft

- On-board computer of Spacecraft

Courtesy of JAXA

Courtesy of JAXA

1. Our Experience

2）Cellular Phone

- Browser software
- Mailer software
- Porting Java VM

1. Our Experience

3）Internet Systems

- Web Systems
- Web Services
- SVG

airLook Location Based Services

2. Technology Advantage

Business Field	Technology Services			
	Real-time Software			Solution Tools
	Embedded Software	Core Technology Services	Technical Applications	
Mobile Networking		OMA	Network Infrastructure	Real-time Power Series Karearea airLook
Wireless	Cellular Phone			
Internet	In-vehicle Terminal PDA	XML, BML, SVG, LBS	Web Systems Web Services	
Infrastructure			ITS, Broadcast, Defense, LBS, Logistics	
Space & Advanced Technology	Spacecraft, Satellite, Robotic	Object Oriented Design, RT Component	Rocket, Spacecraft, Astronomy	

Systems Engineering Consultants
Ⅲ. Our Experience & Advantage

3. Advantage in Robotic System

- Firmware on Robotic System
- Robot Control and Robot Monitoring server application
- Robotic Content Service System

Ⅳ. Our Point of the Standardization of Robotic Systems

Ⅳ. Our Point of the Standardization of Robotic Systems

In the last Section,

1. Motivation
2. Our Interest
3. Our Point of the Standardization

Ⅳ. Our Point of the Standardization of Robotic Systems

1. Motivation

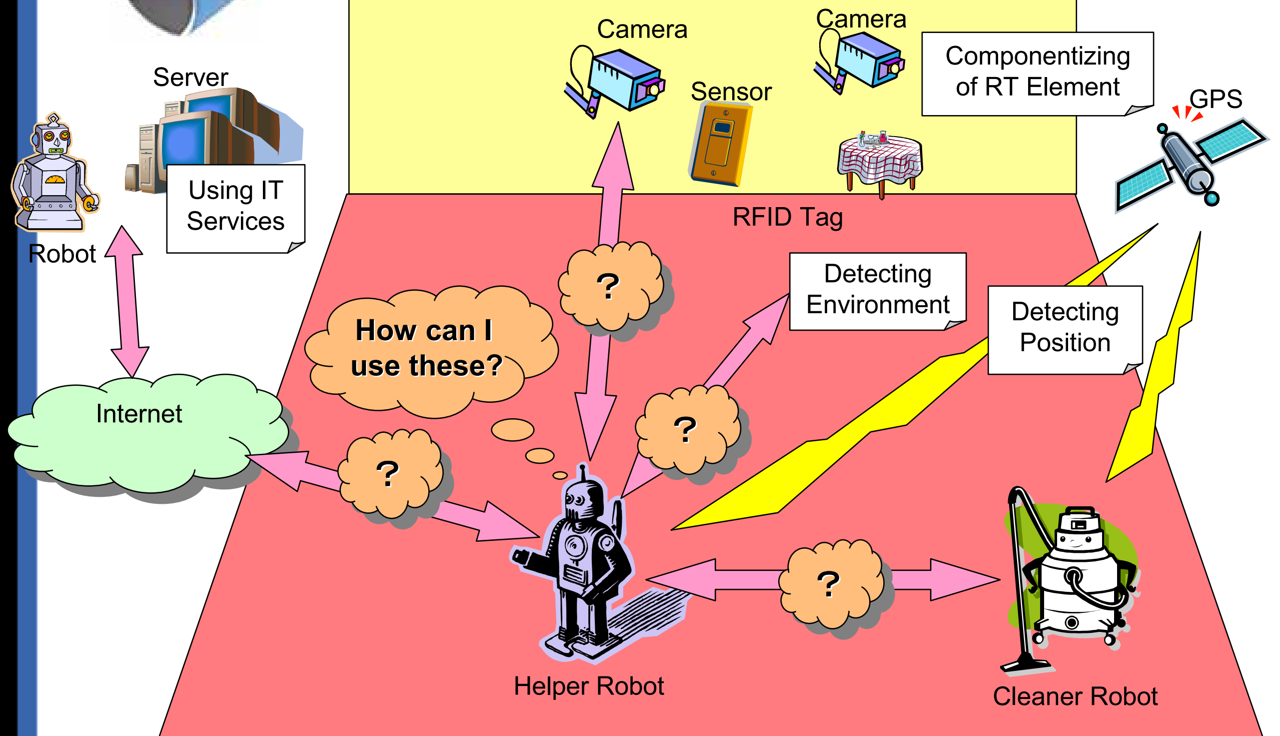


Figure - Present Situation, which surrounds Robots

Ⅳ. Our Point of the Standardization of Robotic Systems

1. Motivation

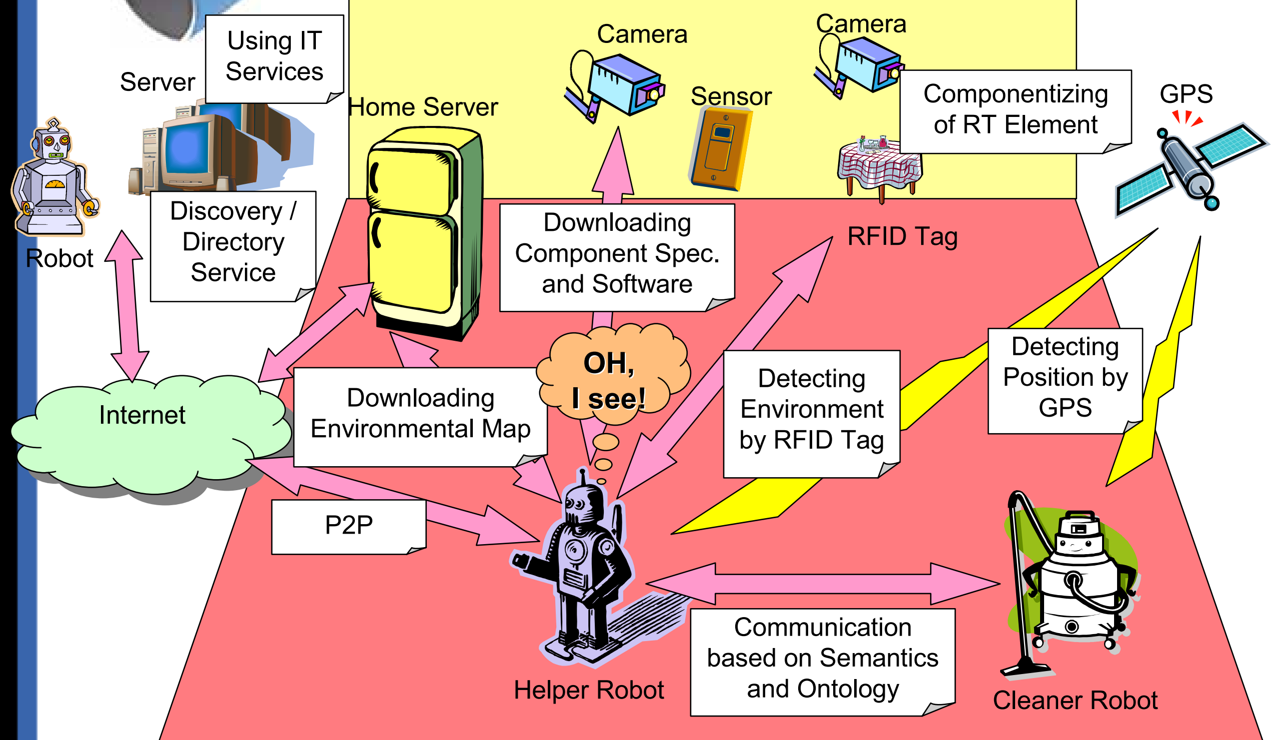
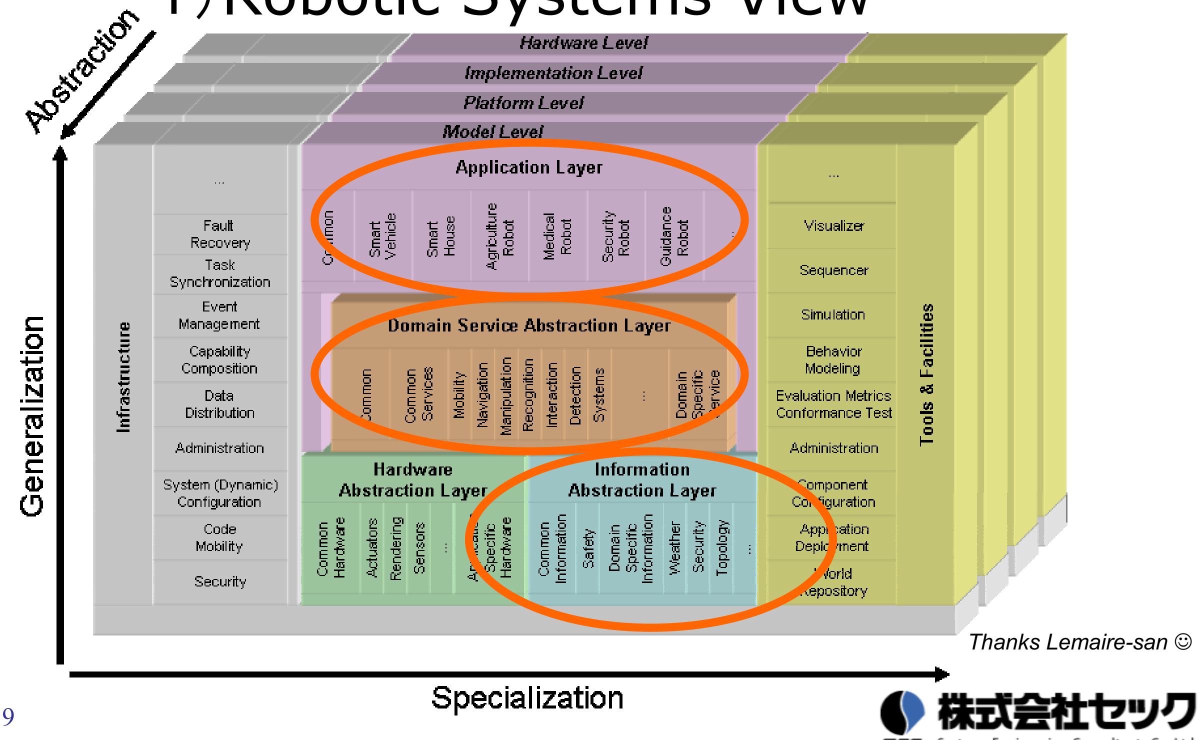


Figure - Future Robot services Use Cases with Technology Fusion

Ⅳ. Our Point of the Standardization of Robotic Systems

2. Our Interest

1）Robotic Systems View

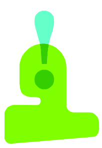


Thanks Lemaire-san ☺

Ⅳ. Our Point of the Standardization of Robotic Systems

2. Our Interest

2）Our Interest Point



Code	Infrastructure Layer / Middleware Layer
IM1	Component Model / Encapsulation
IM2	Data Flow / Data Distribution
IM3	Command Flow
IM4	Event Management
IM5	Security Management
IM6	Code Mobility
IM7	Activity Monitoring
IM8	Execution Synchronization / Prioritization
IM9	Capability Modeling / Description / Advertisement
IM10	Capability Access Policies
IM11	Capability Composition / Hierarchization
IM12	Resource Allocation / Management
IM13	Safety Management / Safety Procedure / Safety Policies
IM14	Fault Tolerances / Recovery Strategies
IM15	Physical Space Management
IM16	Time Management
IM17	System Configuration / Dynamic Reconfiguration

Ⅳ. Our Point of the Standardization of Robotic Systems

2. Our Interest

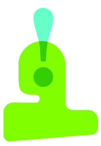
2) Our Interest Point

Code	Hardware Abstraction Layer
HAL1	Common Device Definition
HAL2	Composite Device Management
HAL3	Sensing Devices
HAL4	Actuating Devices
HAL5	Rendering Devices

Ⅳ. Our Point of the Standardization of Robotic Systems

2. Our Interest

2) Our Interest Point



Code	Data Abstraction Layer
DAL1	Common Data Structure
DAL2	Data Adaptation Mechanism / Data Relation Management
DAL3	Data Semantics / Ontology
DAL4	Geometry
DAL5	Imagery
DAL6	Topology

Ⅳ. Our Point of the Standardization of Robotic Systems

2. Our Interest

2）Our Interest Point

Code	Domain Service Layer
DS1	Navigation
DS2	Localization / Positioning
DS3	World Mapping
DS4	Path-Planning
DS5	Motion Control
DS6	Kinematics
DS7	Task Planning
DS8	Object Recognition / Person Recognition
DS9	Object Tracking / Person Tracking
DS10	Visual Processing
DS11	Audio Processing
DS12	Sensor Fusion
DS13	Human Interface
DS14	Energy Management
DS15	Neural Networks

Systems Engineering Consultants

Ⅳ. Our Point of the Standardization of Robotic Systems

2. Our Interest

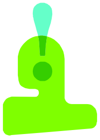
2）Our Interest Point

Code	Tools / Facilities Layer
TF1	Application Composition
TF2	Behavior Language
TF3	Internal Component Configuration
TF4	Application Deployment
TF5	Physical World Repository
TF6	Evaluation Metrics / Conformance Test
TF7	Modelization Language
TF8	Remote Monitoring & Control

Ⅳ. Our Point of the Standardization of Robotic Systems

2. Our Interest

2) Our Interest Point



Code	Application Layer
DA1	Unmanned Ground Vehicles
DA2	Humanoid Robots
DA3	Manipulator Robots
DA4	Robotic Space
DA5	Swarm Robots
DA6	Underwater Robots
DA7	Aerial Robots
DA8	Wheeled Indoor Robots
DA9	Micro Robots
DA10	Agriculture Robot
DA11	Home Indoor Service Robot
DA12	Factory Robot
DA13	Outdoor Service Robot
DA14	Outdoor Heavy Duty Robot
DA15	Rescue Robot
DA16	Communication Robots
DA17	Entertainment Robot

Ⅳ. Our Point of the Standardization of Robotic Systems

3. Our Point of the Standardization

1) For Detecting Environment

- What does the Robot cooperate with?
  - Human-being
  - Another Robot
  - Sensor (e.g. Camera, Floor Sensor,…)
  - RFID Tag
  - IT Services for the Robot
    - RSi Information Service Specification (e.g. Whether, Cooking Recipe, Entertainment,…)

3. Our Point of the Standardization

1) For Detecting Environment

- How does the Robot detect and identify the Object?
  - Protocol / Interface
  - Object Description / Format
    - Data Schema (XML)
    - W3C Semantic Web (Semantics, Ontology)
  - Ontology / Suitable Object Category Classification
    - Guideline

3. Our Point of the Standardization

1) For Detecting Environment

- How does the Robot retrieve the Object?
  - Directory / Discovery Service
    - Sun Service Registry
      - Function of registry and repository to pursue and to manage Web service
      - Support the UDDI v3 specification and the ebXML Registry 3.0 specification
    - Web Retrieval Technology
      - google
      - msn (Microsoft)

3. Our Point of the Standardization

2) For Positioning

- How does the Robot detect the Object's Position?
  - GPS
  - RFID
  - Wireless LAN
    - AirLocation™ (HITACHI)

3. Our Point of the Standardization

2) For Positioning

- Which map data is better for the Robot?
  - Format
    - SVG (Scalable Vector Graphics)
    - NVML (NaVigation Markup Language, Fujitsu Lab.)
    - 2D / 3D
  - How to Distribute
    - from the Internet?

3. Our Point of the Standardization

2) For Positioning

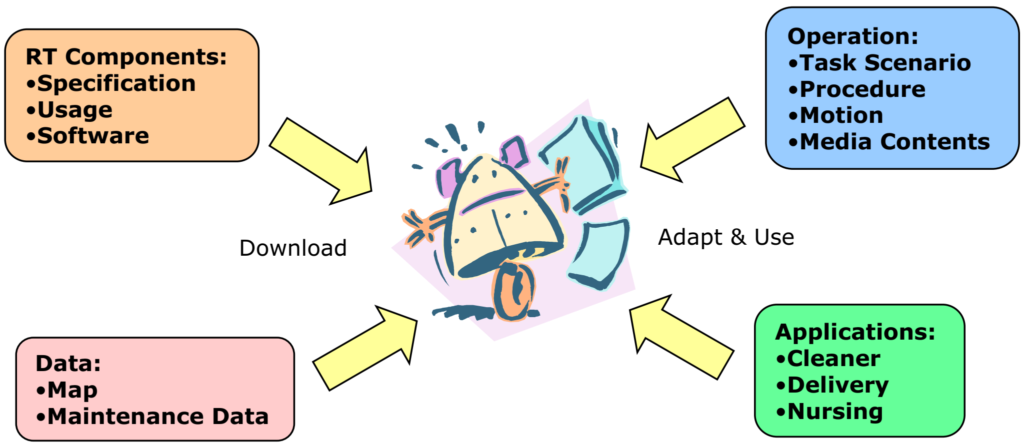
- How can we get the Position of Remote Robots?
  - LBS (Location Base Services)
    - airLook (SEC Co., Ltd.)

Ⅳ. Our Point of the Standardization of Robotic Systems

3. Our Point of the Standardization

3) For Adapting to Environment

- What does the Robot downloads?



3. Our Point of the Standardization

3) For Adapting to Environment

- What's Download Architecture like?
  - Specification for Cellular Phone:
    - OMA Download Architecture V1.0
    - OMA Generic Content Download Over The Air Specification V1.0

3. Our Point of the Standardization

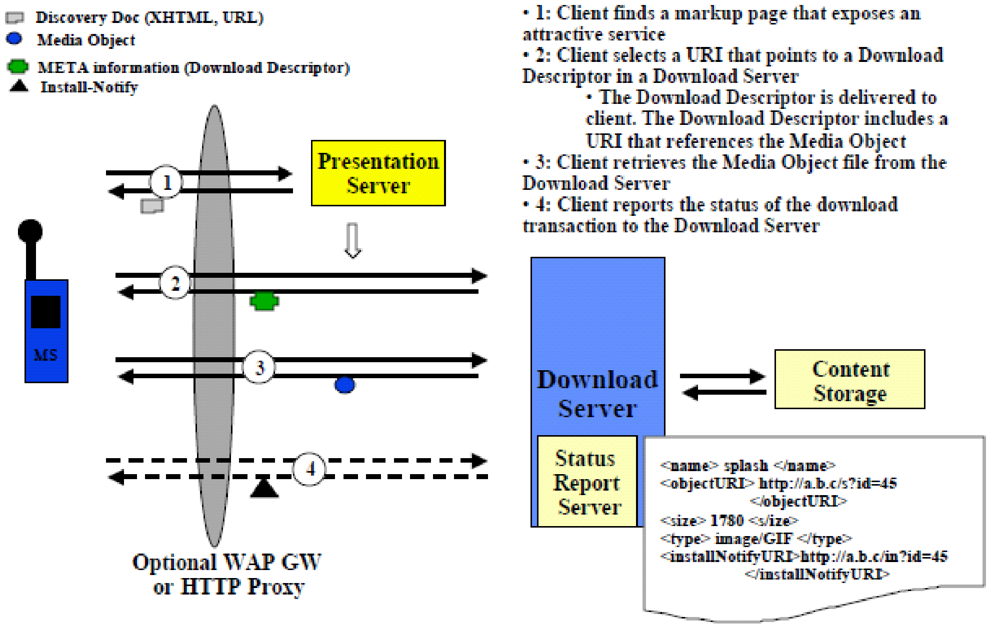


Figure – OMA Download Architecture

Reference: Generic Content Download Over The Air Specification Version 1.0, Version 21-Feb-2003

Ⅳ. Our Point of the Standardization of Robotic Systems

3. Our Point of the Standardization

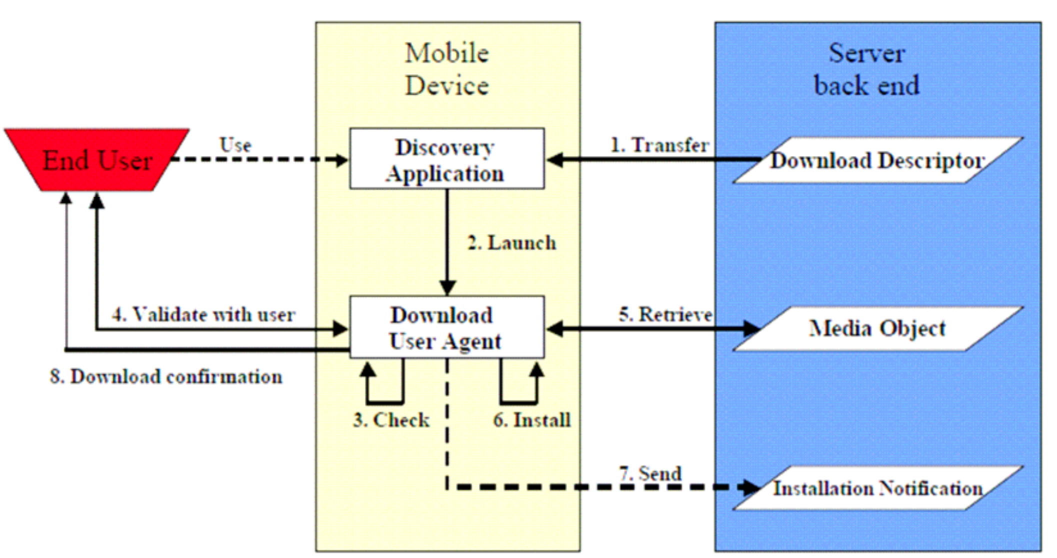


Figure – OMA Download Process

Reference: Generic Content Download Over The Air Specification Version 1.0, Version 21-Feb-2003

Ⅴ. Conclusion

I. Introduction
II. About SEC
III. Our Experience & Advantage
IV. Our Point of the Standardization of Robotic Systems
V. Conclusion

Ⅴ. Conclusion

In this Section,

1. Today's Review
2. Issues

1. Today's Review

Today’ Presentation, We talk about:

- Our Experience & Advantage
- Our Point of the Standardization of Robotic Systems
  - Interest WG
    - Robotic Services
    - Robotic Profiles

2. Issues

We have some more issues to deal with:

1. Joining OMG !
2. Our English !!

Again, Thank you all for your attention today!

We hope our presentation was helpful to you.

For the social safety and development
Realtime@net

http://www.sec.co.jp

E-mail: nakamoto@sec.co.jp, nagase@sec.co.jp

社団法人 日本ロボット工業会 Japan Robot Association

RiN RSi Robot Services initiative

W3C MEMBER

oma Open Mobile Alliance

XML Consortium

JAVA COMMUNITY PROCESS

robotics2006-02-12

Development of Food Robots ,Meat Processing Robots, and Request for Standardization of RTC

Tomoki YAMASHITA

Mayekawa Mfg. Co., Ltd.

tomo-3440@mayekawa.co.jp

MYCOM

Today’s topics of discussion

A. The Introduction of Mayekawa’s food robots and meat processing robots.

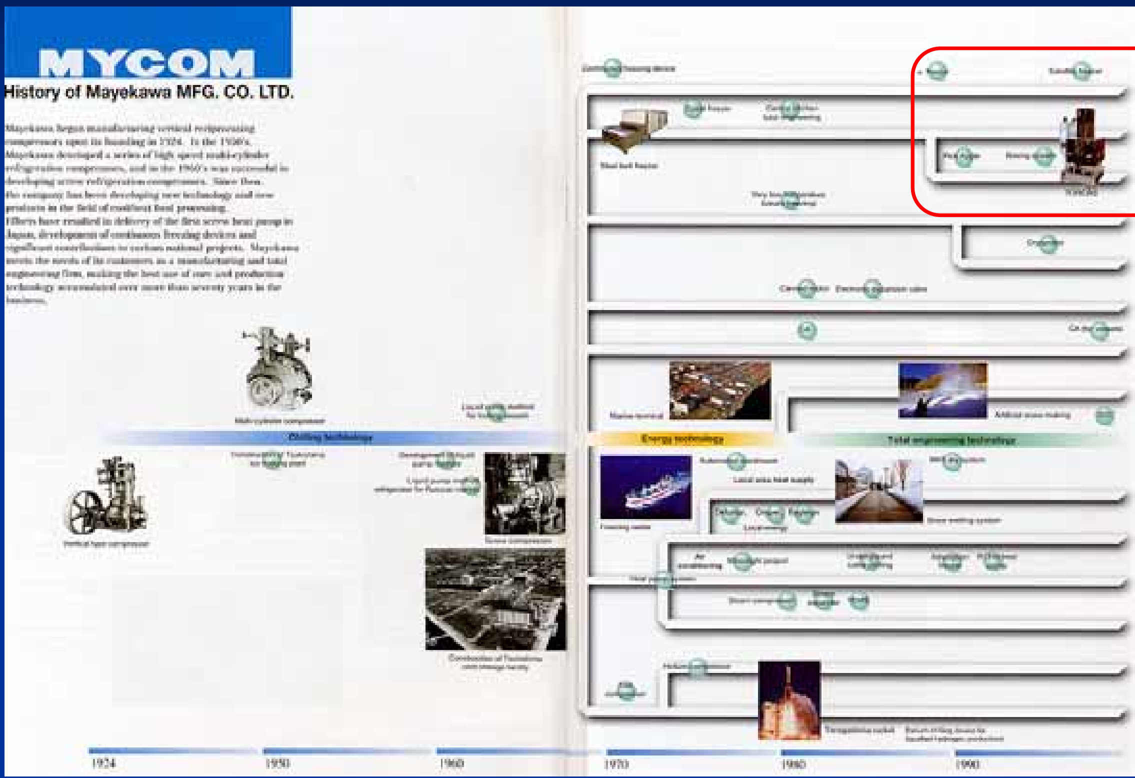
B. Related technical theme:

The adoption of the RT-Middleware “OpenRTM-aist” to the prototype robot

Requests for RTC (as RTC user)

MYCOM

History of Mayekawa technologies and products



MYCOM

Mayekawa's robot divisions

Meat processing robots

Poutly:TORIDAS,TAKIDAS,etc…
Pork:HAMDAS,WANDAS,SHOLDAS…
Beef:(Under development…)
Other: Salmon, scallop,

How to use RT in the food industry

Cold-resistant robots
(Cold storage at -30℃)

Food handling robot

MYCOM

Mayekawa's Meat / Fish Processing Robots

TORIDAS

YIELDAS

TAKIDAS

HAMDAS

WANDAS

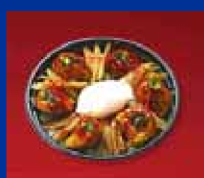
Salmon devoner

MYCOM

Food handling robots development projects (2003-2006)

•Goal

Packing "Bento" (lunch boxes) sold at convenience stores by the robot



•Joint research organization

Hiroshima University, Nihon Pisco Co., Ltd.,
Chiba Precision Co., Ltd. Mayekawa MFG. Co., Ltd.

Supported by Organization for Small & Medium Enterprises and Regional Innovation, JAPAN (SMRJ)

中小機構
独立行政法人
中小企業基盤整備機構

Packing line at box lunch factory
--This market grows up to 500 billion dollars--

MYCOM

Difficulties in adoption of RT in the food industry

- Taking into account the food sanitation = washable body

  Chemical and water resist body
  Care about electrical devices, optical devices, etc.…
  Easy-to-overhaul body

- Robot handling ability < Human handling ability

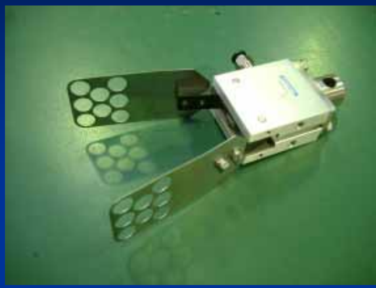
  Industrial robots cannot grip various foods with one hand ,because most of foods are soft and flexible shape .

Development of easy changeable hand-tools according to characteristics of foods.

MYCOM

Examples of the prototype hand

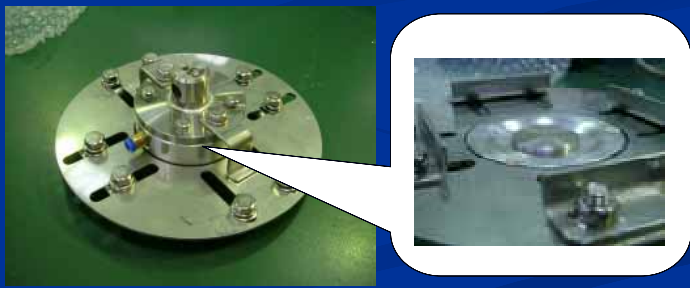
For large meals



Holding by the leaf spring
(For small and elastic meals)

Holding by the skewer

Vacuum type



Holding by the non-contact transport device “NCT”
(NCT is the Koganei Corp.’s product)

Test : Humburger putty, Fried fish, Fried chicken

Problem in Primary test

My customers said :

"It is a pain to make foods line up manually!"

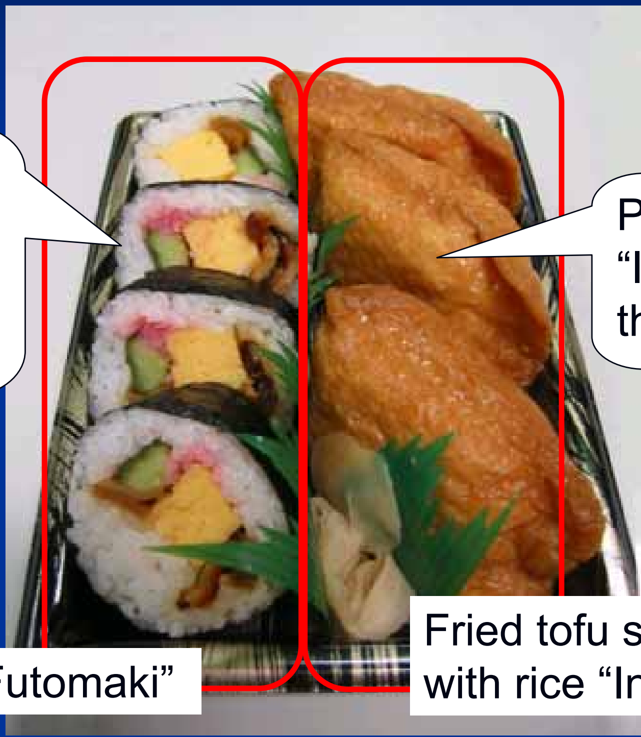
Auto-arrangement by the Parts-feeder

Foods auto-arrangement test (Fried Tofu/ Buns Bread)

link to the movie…

Current theme : “Sukeroku sushi ” --Most popular item--

Tilting pieces back to show the cutting plane.



Placing “Inari”s on the diagonal

Sushi roll”Futomaki”

Fried tofu stuffed with rice “Inari sushi”

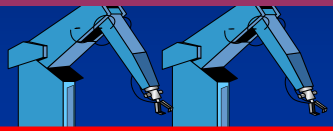
MYCOM

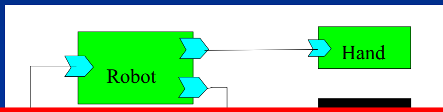
Sushi roll ”Futomaki”

MYCOM

Related Technical topics
--Using RT Middleware "OpenRTM-aist" --

We tried to use RT Middleware "OpenRTM-aist010" in the development of the food robot



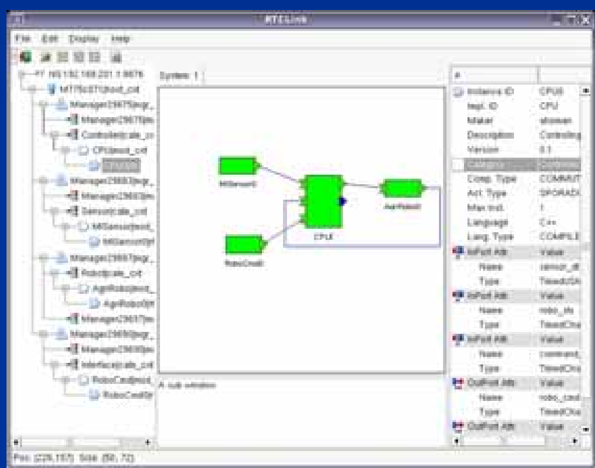


※OpenRTM-aist is not supported for the industrial robot controller and the programmable logic controller (PLC).

Give up …

MYCOM

Related Technical topics
--Development of Agri-robot using RT Middleware (OpenRTM-aist0.2.0)--



Prototype of harvest transportation robot
(Controlled by the laptop PC→Follow the people in the future)

MYCOM

Agri-robots in Japan
--Japanese farmland are small--

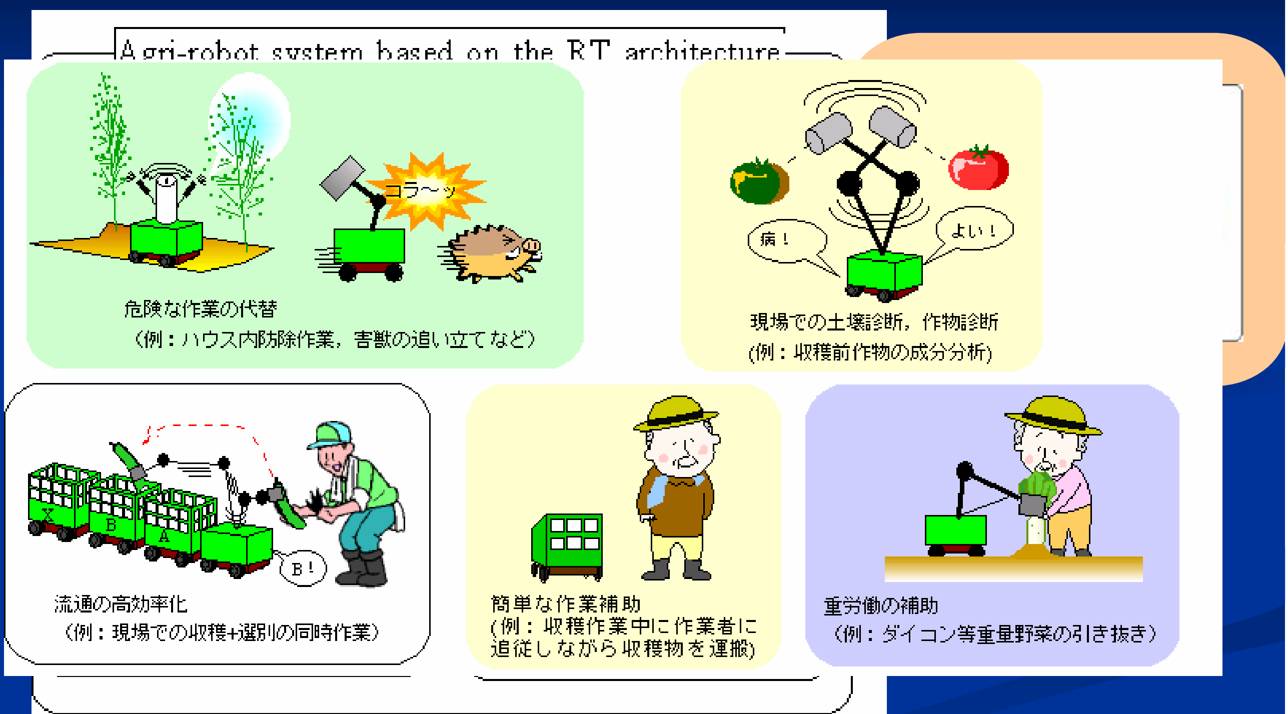
Unmanned Tractor

Self-steering rice transplanter

- Very Big!!!
- Very Expensive!!
- Work in limited time!!

MYCOM

User configurable Agri-robot system
--Based on RT-Architecture—



MYCOM

Test run

Last: Request for Standardization of RTC

→Request to OpenRTM development team in AIST

(a)Standard RTC development tools.
  (e.g.User-friendly GUI development tools)

(b)Platform independent: Including Non-pc -based controller device.
  (e.g. Programmable logic controller (PLC), tiny microchip for embedded controller)

(c)RTC's operation: Installing or starting / shut-downing RTC's program. (Instant starting like "Plug-n-play")

(d)Safety management (Hardware error, Data transmit/receive error, Software runaway, etc)

Thank you for your kind attention.

robotics/2006-02-13

IT R&D Global Leader

RT service framework using IT infrastructure

Response on
"RT (Robotics Technology) Services: Integration with IT Systems"

2006. 2. 14.
Wonpil Yu

ETRI
Electronics and Telecommunications Research Institute

ETRI

RT service for real-world

- To realize an RT service for real-world
  - Perception of objects, humans, environment
    - Vision, sonar, laser, radar, IR, etc.
    - Platform/domain-specific algorithms
    - Difficult to obtain a reliable solution
  - Behavior control (navigation)
    - Localization, mapping, path planning
    - Localization is a key factor for successful navigation
    - Partial success as of today due to lack of generality
  - Connectivity to IT infrastructure to be available for users
    - Integrate RT with the existing IT infrastructure
    - Reliable perception, complete control of robot behavior

New tools to realize RT service

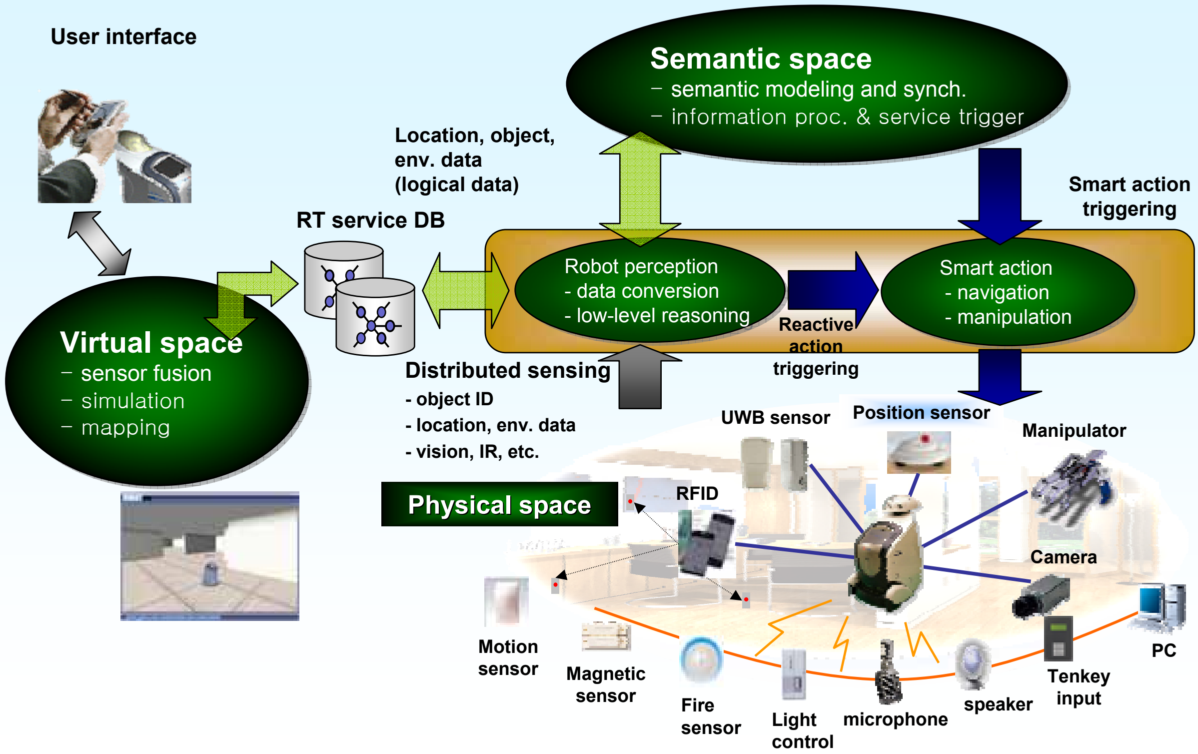
- Wireless sensor network
  - ZigBee for monitoring and maintenance of environment
  - RFID for object, environment recognition
- Localization network for robot localization
  - GPS for outdoor environment
  - UWB (IEEE 802.15.4a) for precision radio ranging
  - Custom localization solution based on various physical media
- Broadband communications network
  - Connectivity to the Internet
  - Connectivity to existing communications network (e.g., CDMA)

Our approach to implement RT service

- Robo-Care service (office care)
  - Monitoring environment and understand current situation
  - Carry out relevant robotic action
- Decompose RT service into three conceptual spaces
  - Physical space
    - Localization network, sensor network, RT service management (RSM) server, robot, devices, etc.
  - Semantic space
    - Domain knowledge management
    - Contextual information processing
    - Robotic service triggering
  - Virtual space
    - User interface to physical space
    - Direct interaction with user through PC, PDA, mobile phone, etc.

ETRI

Ubiquitous Robotic Space



ETRI

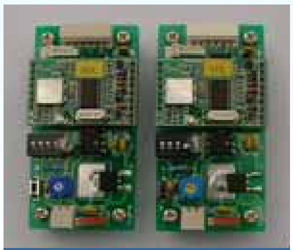
Components in physical space

- Localization network
  - *StarLITE*: IR-based precision robot localization system
  - Localization network should be separated from sensor network
    - To reduce communication/processing burden for sensor network
- Sensor network
  - ZigBee based environment monitoring
  - Temperature, light, humidity, motion, magnetic sensor
  - Sink nodes to connect to RSM server
- RSM (RT Service Management) server
  - Provide connectivity to the communications network
  - Manages the three spaces

Components in localization network

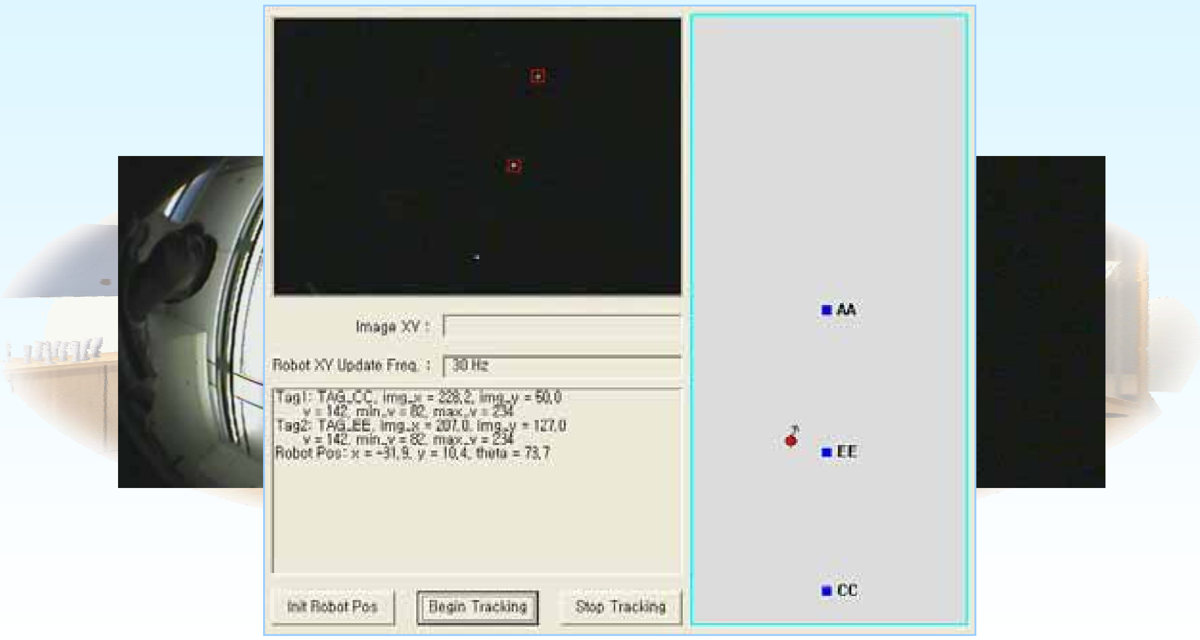
- Robot localization sensor STARLITE
  - Components
    - Detector (Camera + Tag Controller)
    - Wireless Tag (Infrared LED + Wireless Module)
  - Main features
    - Real-time localization (x, y, θ)
    - Easy installation
    - Low-cost
    - Robust to ambient lighting condition
  - Performance data
    - Accuracy: ± 4.1 cm / ±0.1 deg.
    - Jitter: 0.42 cm / 0.01 deg.

Prototype Detector (Camera + Transceiver)



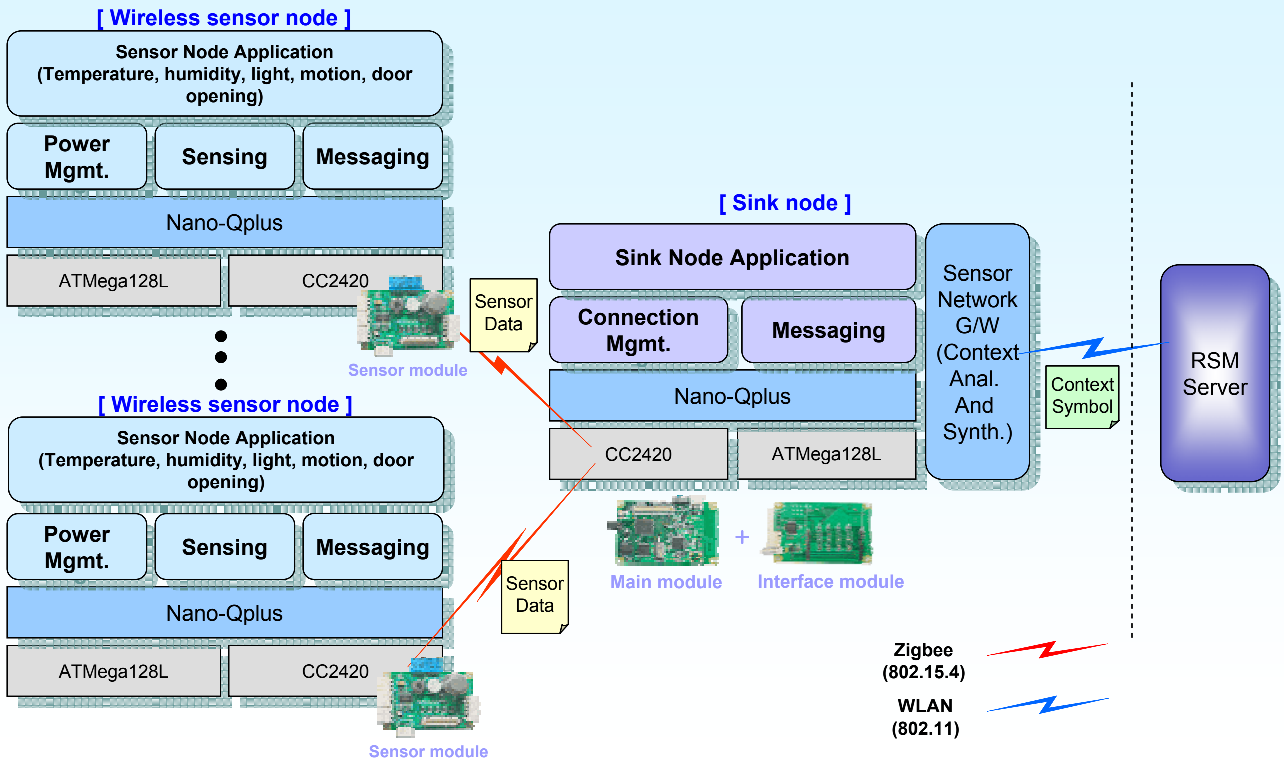
Prototype Tag with Infrared LED

Operating principles of localization sensor



ETRI

Components in sensor network

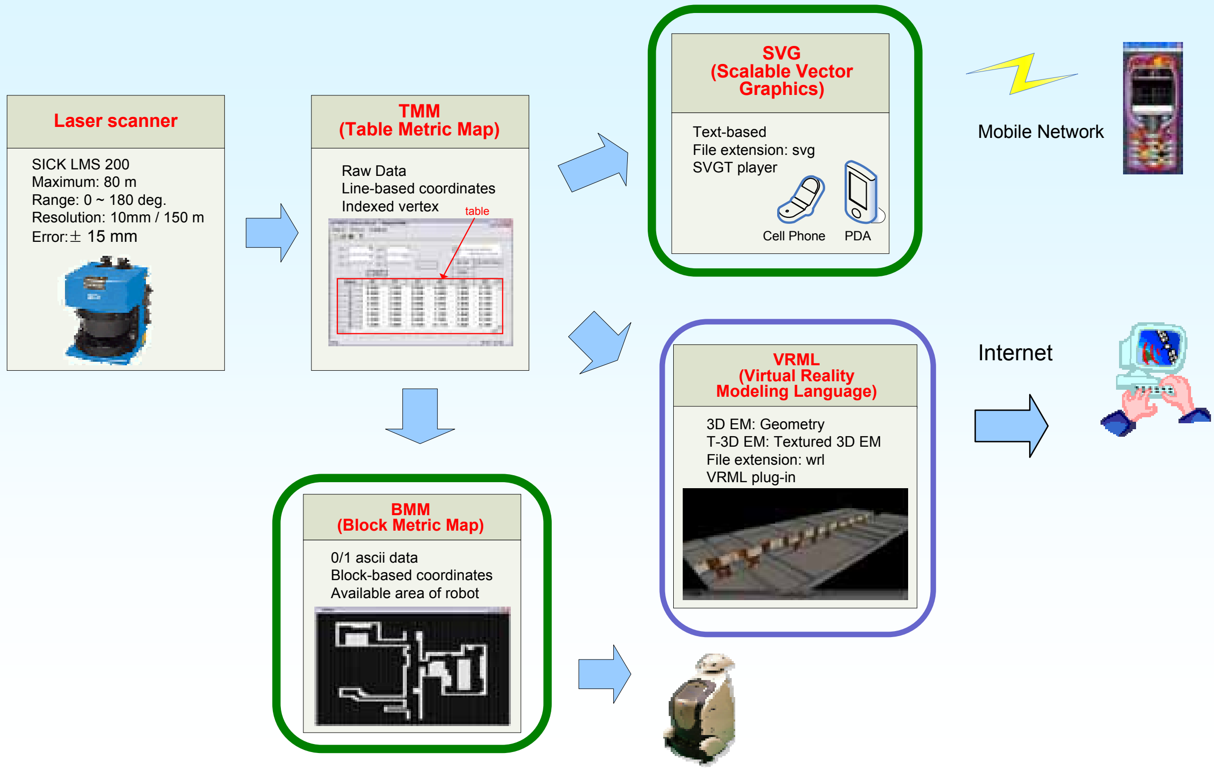


ETRI

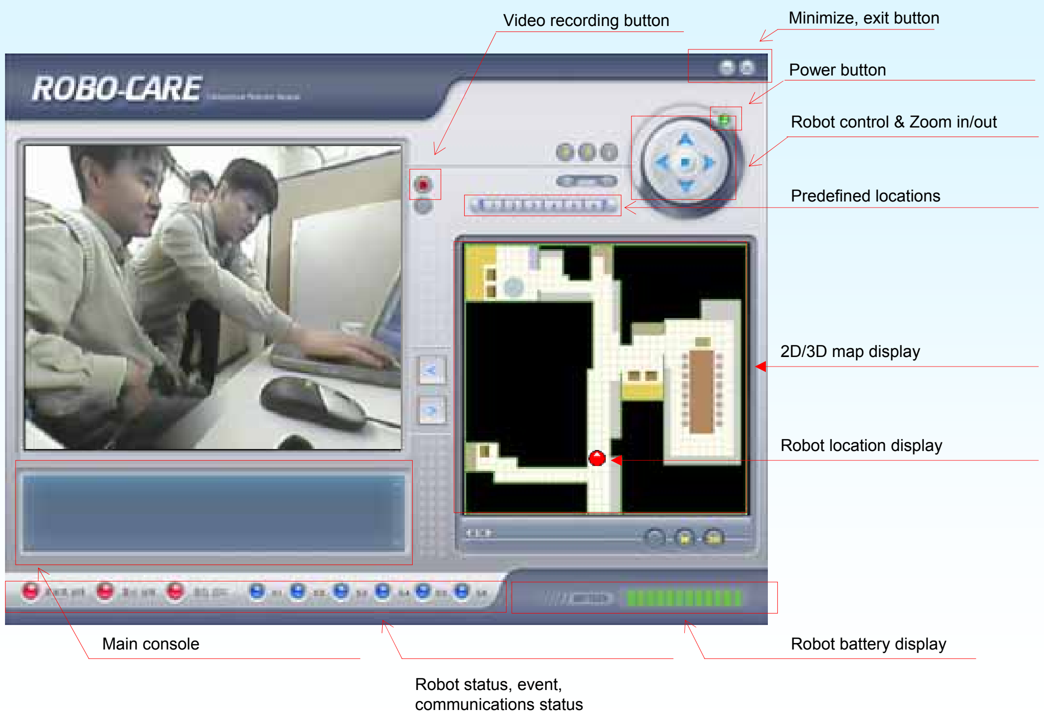
Functionalities in semantic space

- Situation understanding
  - Input: context data
    - Recognition Results: Vision, Speech Recognition, etc
    - Sensor Outputs: motion, door opening
    - Robot Status: Location, Battery Level, Velocity, etc
    - Web Data: Weather, Stock Quotes, Traffic Status, etc
  - Output
    - Semantic context descriptions
- Service generation
  - Input: semantic context descriptions, service knowledge
  - Output: service commands

Components in virtual space



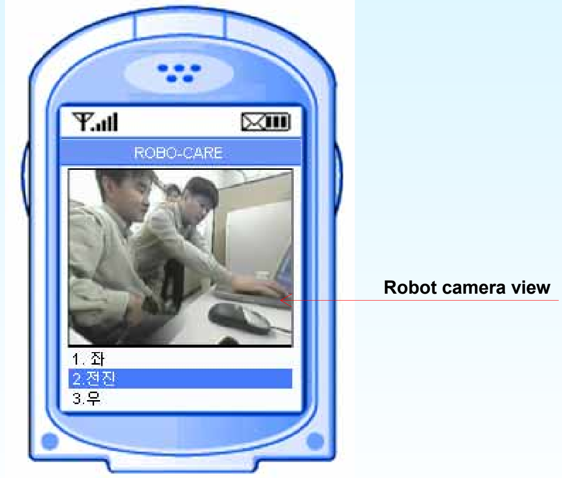
RT service client (PC)



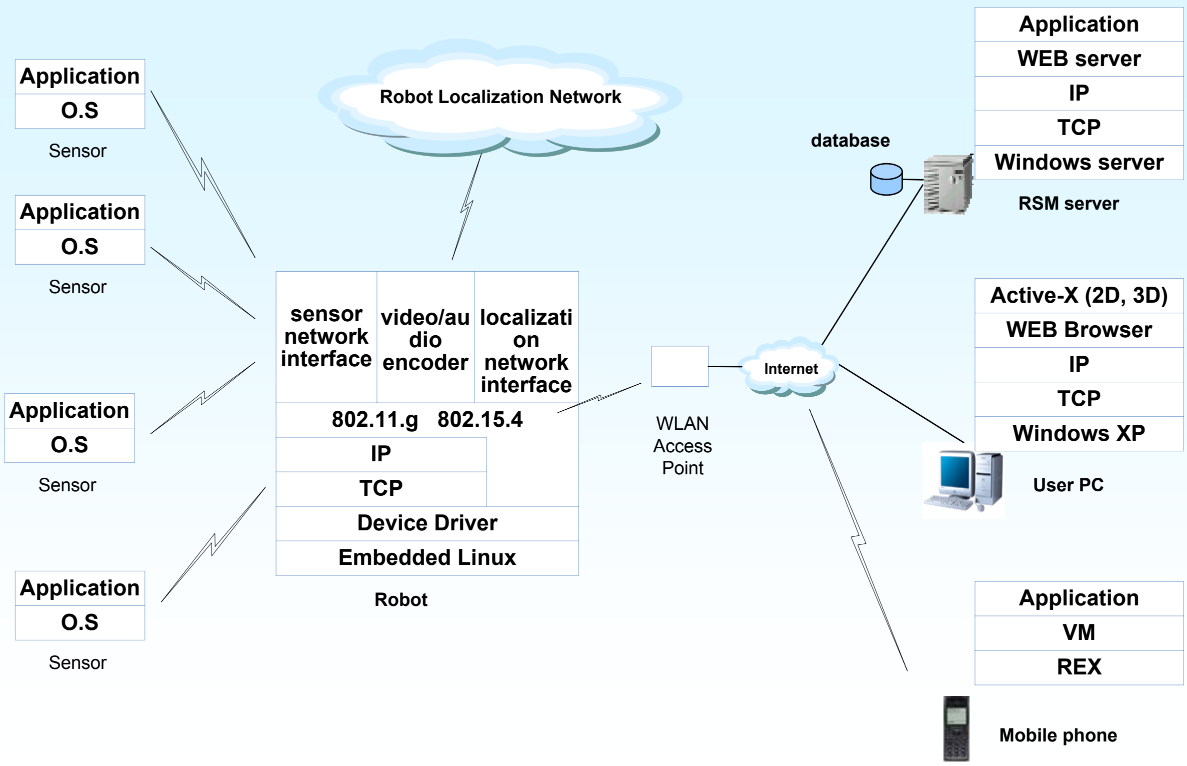
RT service client (PDA/mobile phone)

< PDA GUI >

< Mobile Phone GUI >



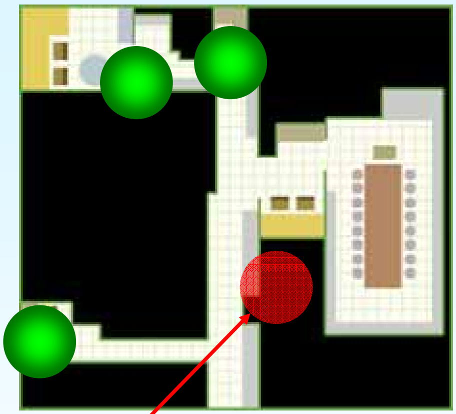
RT service system structure

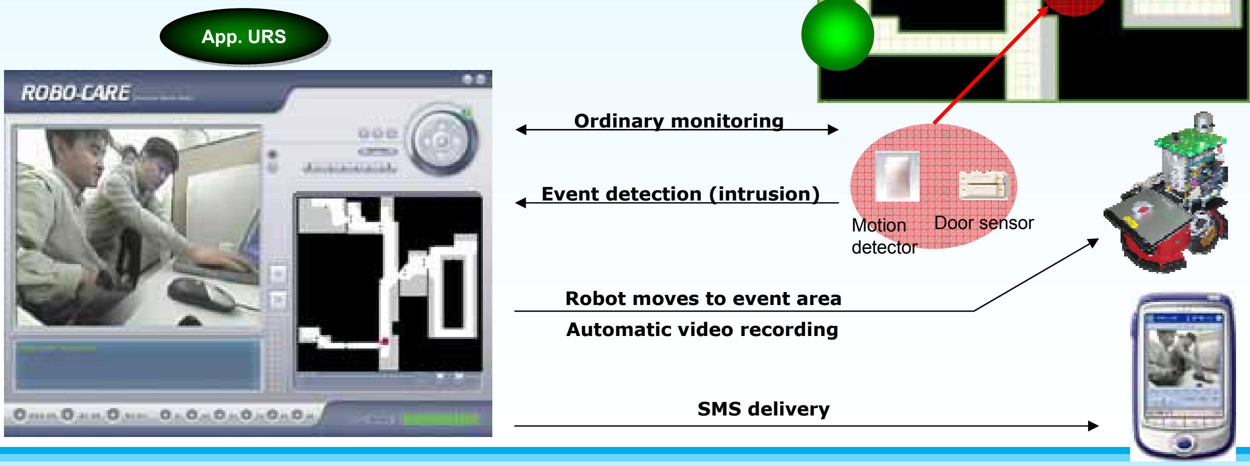


ETRI

RT service scenario example

- Robot monitoring (localization network)
- Event detection (intrusion)
  - Semantic space triggers robot actions
  - Alarm message to user (SMS, etc)
  - Robot moves to event area
  - …





ETRI

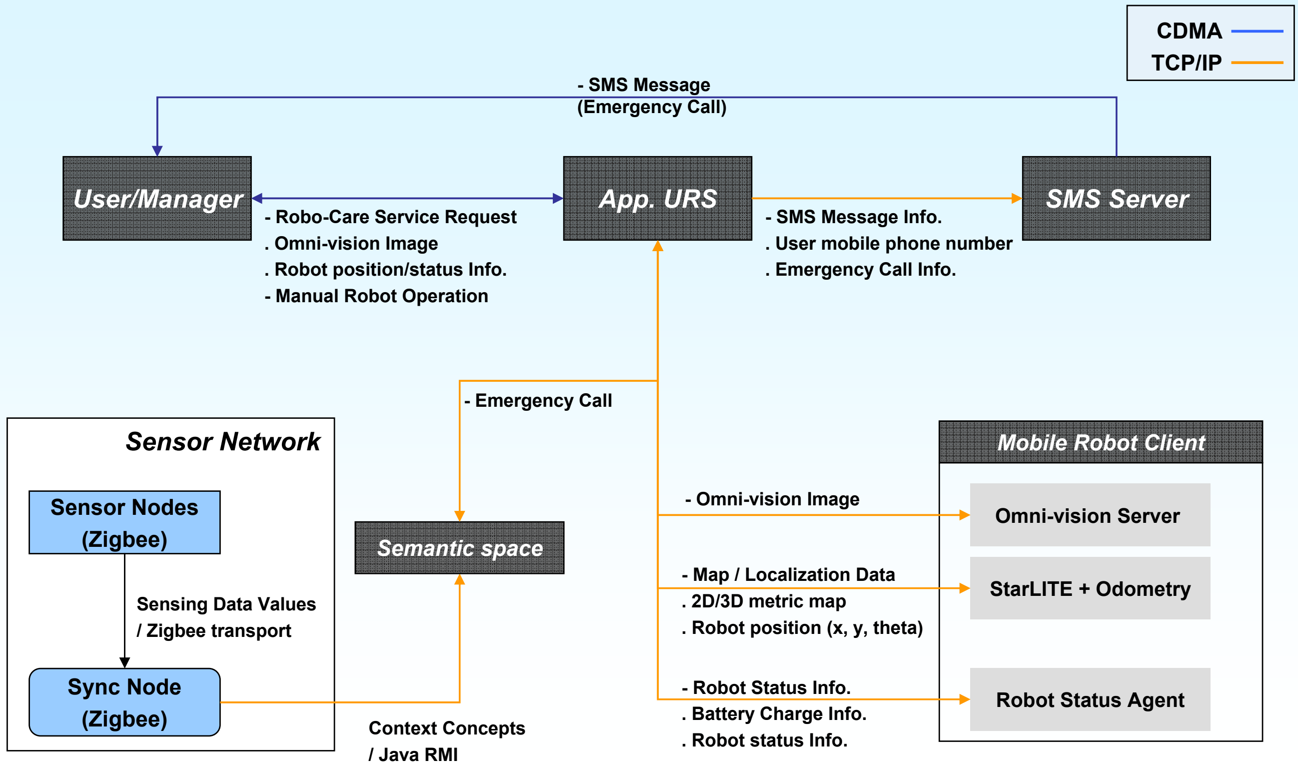
RT service video demo

Concluding remarks

- The employed three spaces structure is efficient to develop RT service
  - *Independent development of each space*
    - *Each space can be independently modified or improved*
- What to standardize?
  - *Interface between the three spaces*
  - *Which is actually S/W API interfaces among the spaces*
- What is obtained?
  - *Reduction of development complexity of RT service based on IT infrastructure, which is, in fact, very complex*
  - *No extra investment at the user's side*

Thank you for your attention!

RT service message flow



RT service protocol

- Common header

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Flags |  Protocol ID  |   Robot ID    |    Op Code    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Message Length                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

- Version(4) : RCSP Version
- Flags(4) : Option
- Protocol ID : RCSP ID
- Robot ID(8) : Robot Identifier
- Op Code(8) : RCSP Operation Command definition
  . 1 : Connection Open (CO)         S <--  C
  . 2 : Connection Accept (CA)       S  --> C
  . 3 : Connection Close (CC)        S <--> C
  . 4 : Request (REQ)                S <--> C
  . 5 : Response (RES)               S <--> C
  . 6 : Report (RPT)                 S <--> C
  . 7 : Ack (ACK)                    S <--> C
  . 8 : Nack (NCK)                   S <--> C
  . 9 : KeepAlive (KA)               S <--> C
- Message Length(32) : Total Message Length for decoding/encoding
```

RT service protocol

- REQ handler
  - Define types of request message
  - Define image/video recording start
- Image handler
  - Define image size, format, recording interval
- Response handler
  - Robot status, robot position, image/video transfer, etc.
- Request, response, report message

A Mobile Robot Software System Architecture with Unified Sensory Data Integration

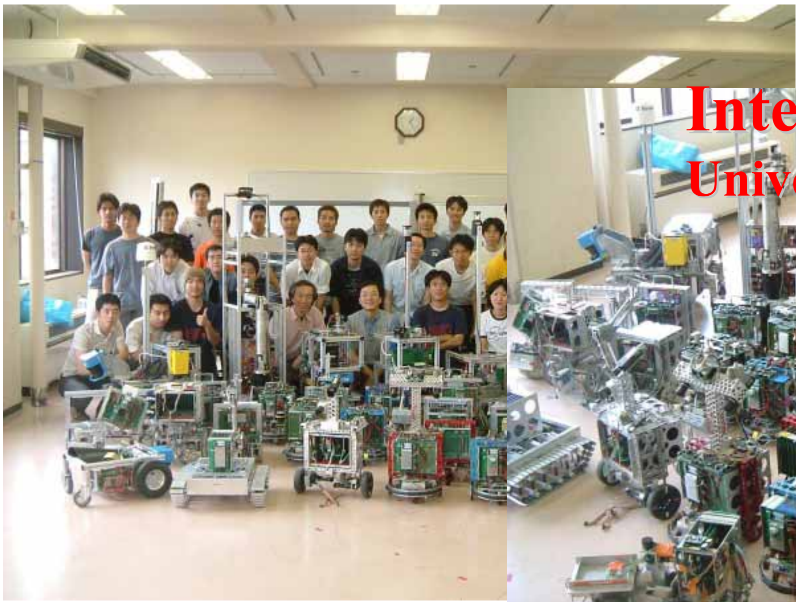
Takashi Tsubouchi and Eijiro Takeuchi

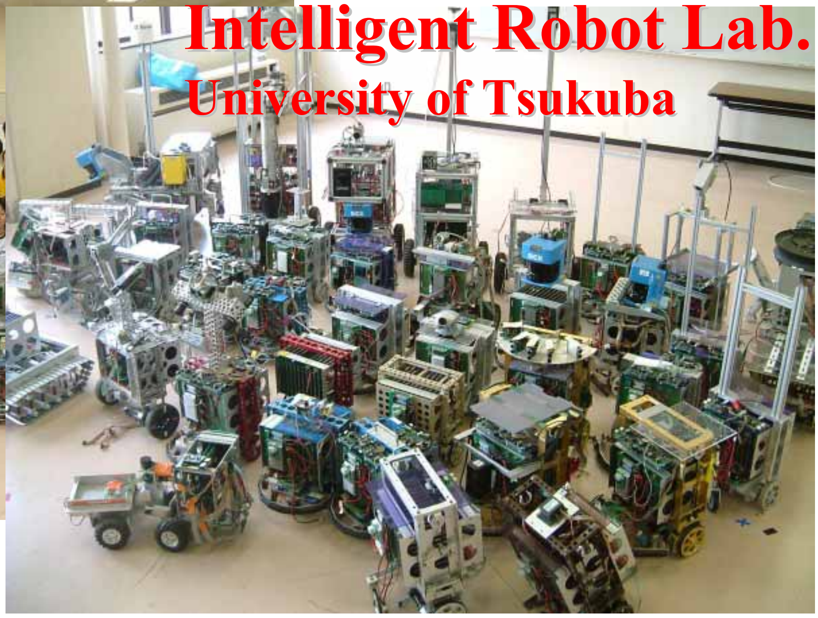
Intelligent Robot Laboratory,

University of Tsukuba, Japan

Introduction of Mobile Robots at Intelligent Robot Lab.

Lab. Members and Our Robot

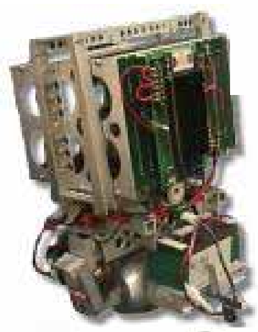




Intelligent Robot Laboratory "Robo-ken"

at University of Tsukuba

- ***Self-contained***
- ***Autonomous***
- ***Mobile Robots***



Approx. 40 robots
In the lab.

Autonomous Mobile Robots "*YAMABICO*" family

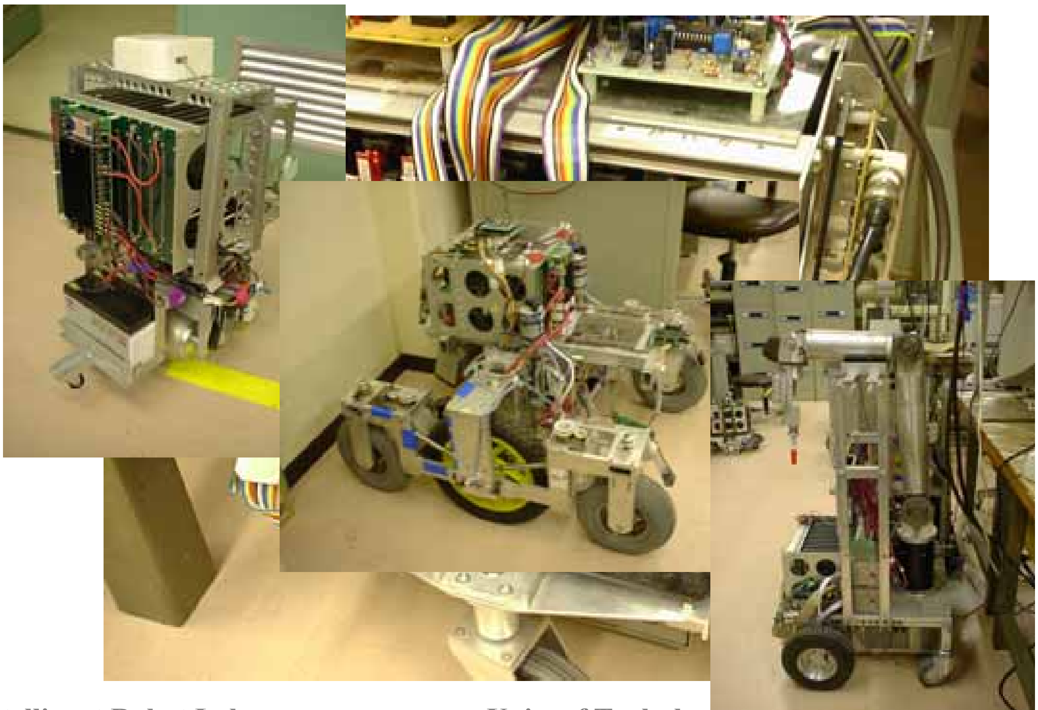

- ***Autonomy***
  - **Behavior decision by itself**
- ***Self-containedness***
  - **Energy source, computers, sensors and actuators are all in one.**






"YAMABICO (山彦)" stands for "mountain echo"

Range finding principle by ultra sound
is the same with the mountain echo.

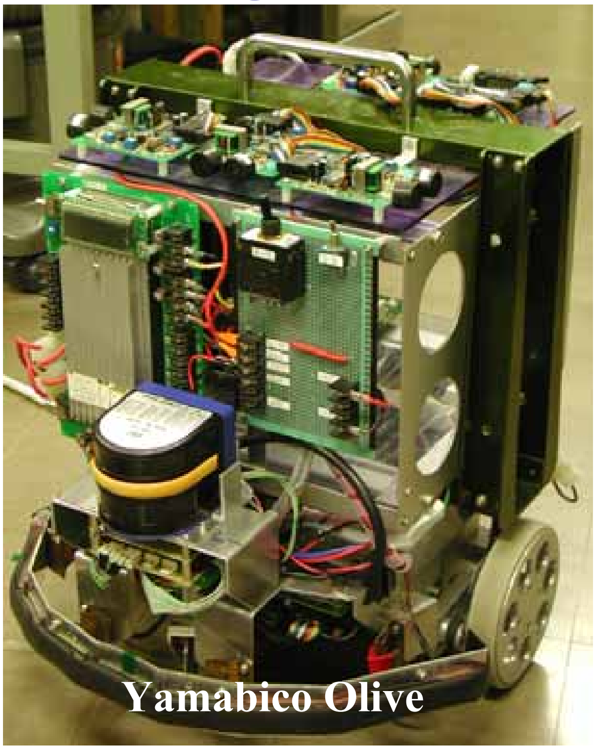


筑波大学
University of Tsukuba

A Prototype of Working Robot

(To be presented at
IEEE ICRA 2006 in May,
Takeuchi and Tsubouchi)

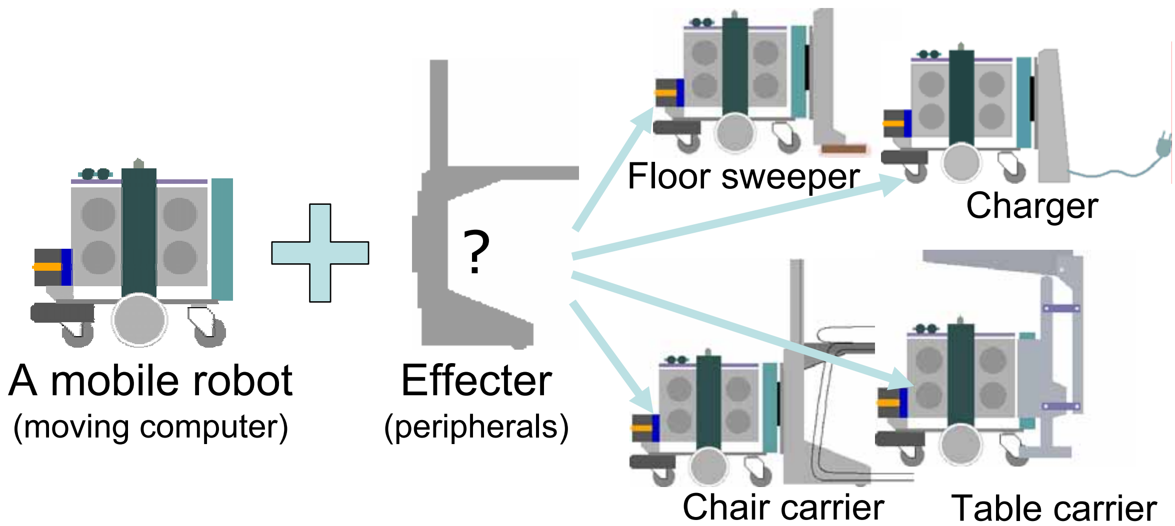
- Use in house or office rooms
- One work per one effecter
- Several types of effecter
- Automatic effecter exchange



筑波大学
University of Tsukuba

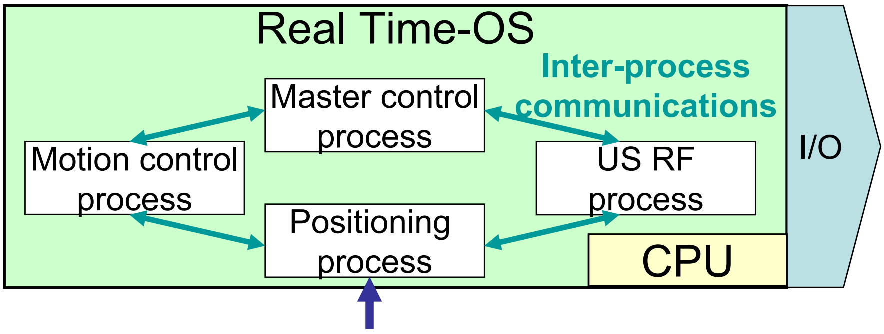
Can be a new business model?

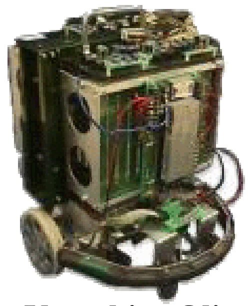
Some venders provide a robot, and
other venders provide each kind of the effecter.



Controller of This Robot

- A PC-board with single Pentium MPU and ART-Linux.



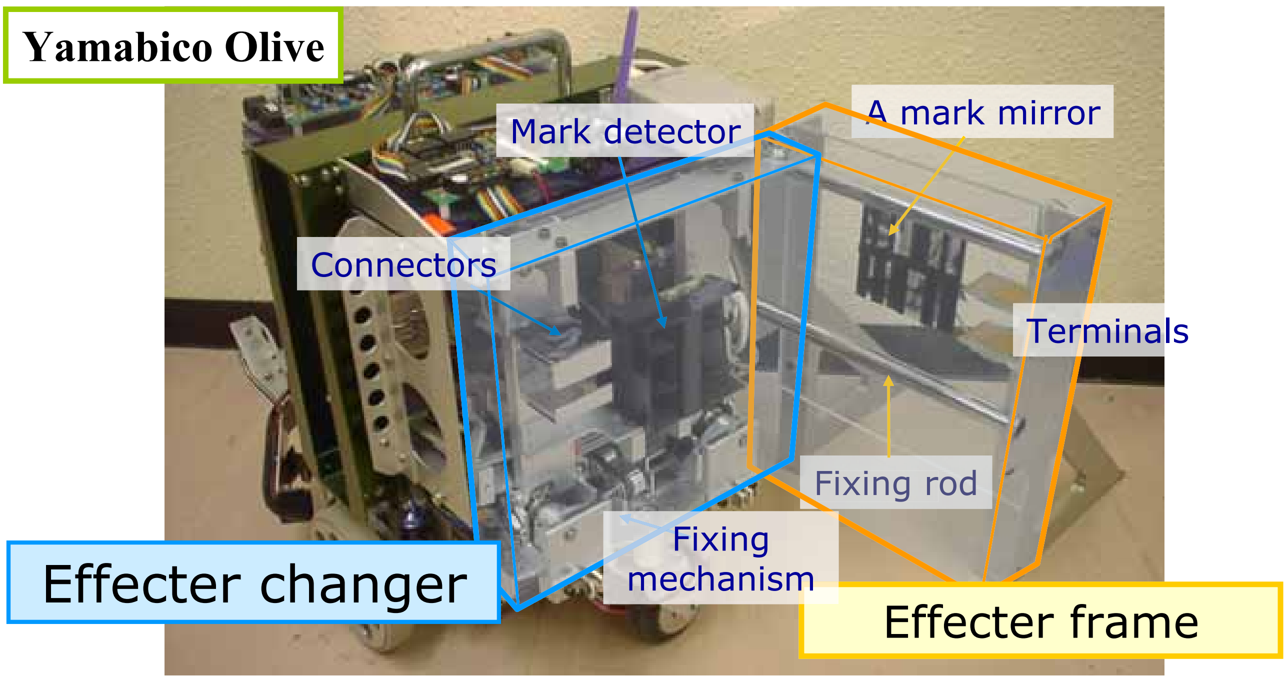


Yamabico Olive

Functions are modularized into software processes

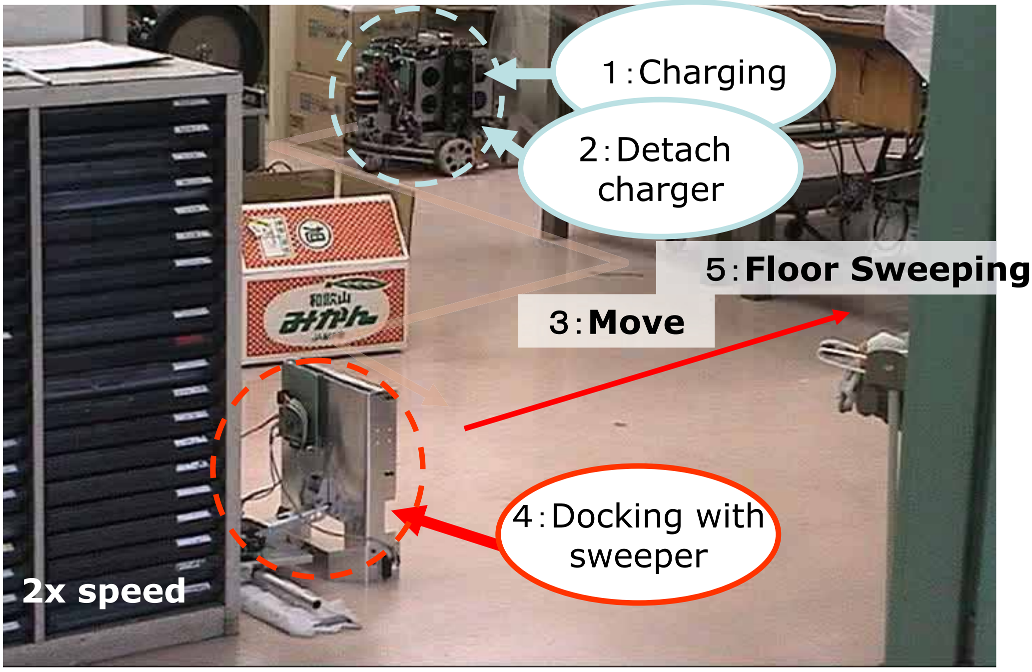
  - Multi processes
  - Modularization method needs skills

A Robot with an Effecter Frame



筑波大学
University of Tsukuba

Demo (Floor Sweeping in Free Space)



筑波大学
University of Tsukuba

Demo (Floor Sweeping in Free Space)

2x speed

Main Issue
(Identify Free Space)

Fundamental Function of a Mobile Robot

- Mobility
- Navigation

Behavior Description Program
(= Application Software)

- Description of Free Space

Sensory Information

Application Software to
Realize Navigation

My Comment at Last Robotics DSIG Meeting in Burlingame

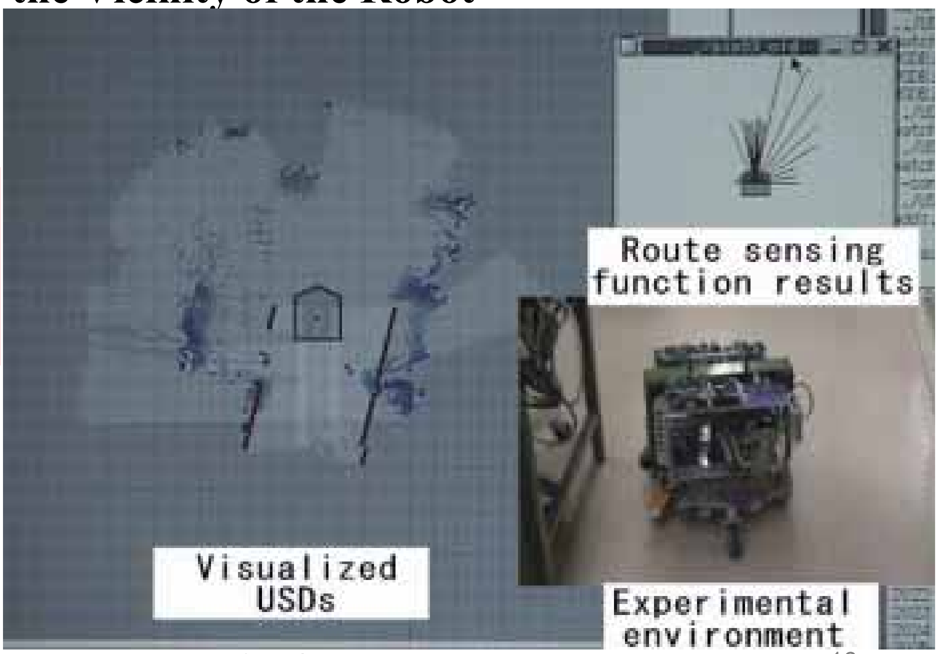
"Not only an software architecture or data flow, but also data structure must be discussed."

Integration and Unified Sensory Data

A Proposed (Proposing?) Solution

- Use of **"Unified Sensory Data Bank (USD)"**
- A 2D Scrolling Ring Buffer
- Grid-map of the Vicinity of the Robot

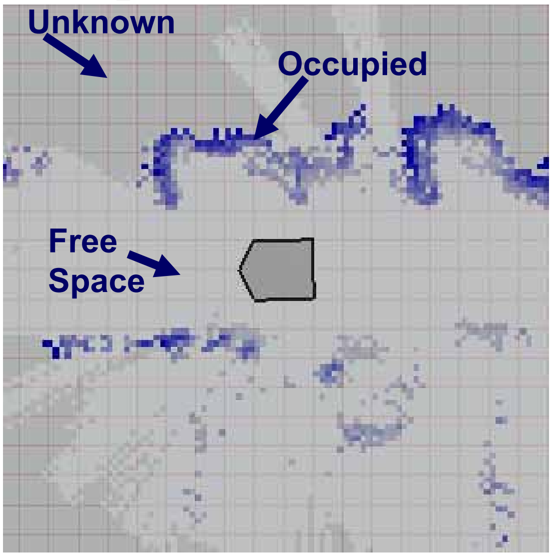
All info about environment from sensors will be mapped in the USD



筑波大学 University of Tsukuba

The Idea - Unification of Sensory Information

- **Unified (abstracted) information independent from specific type of sensors**
  - **Description of "free space" is essential for obstacle detection**



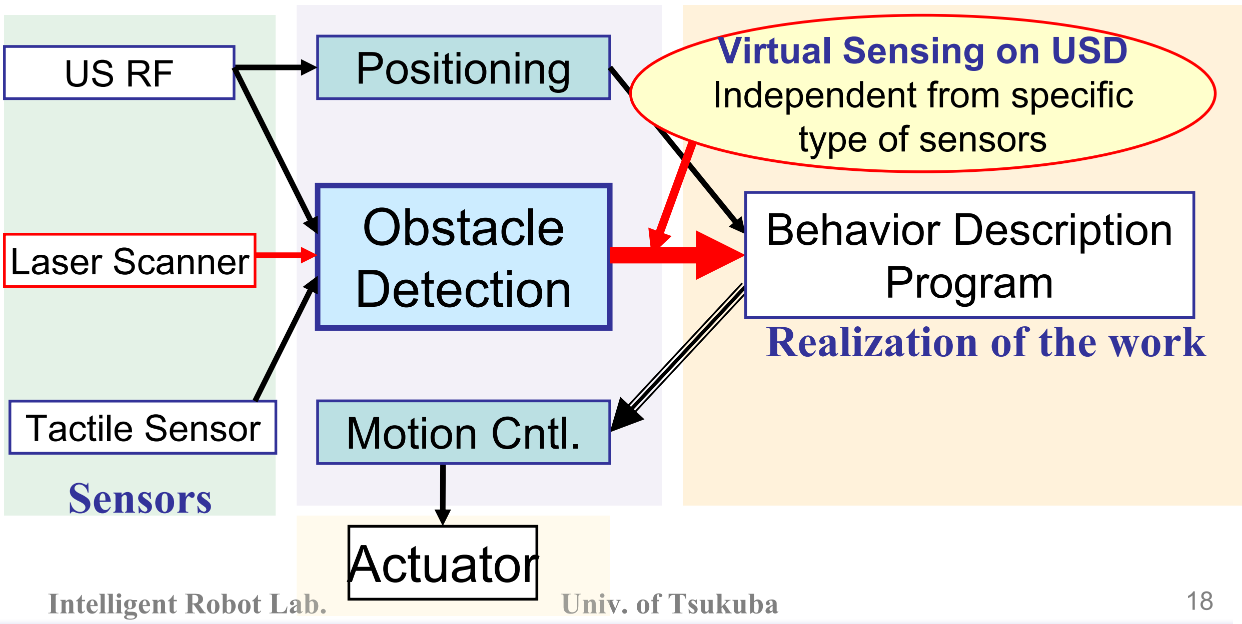
Accumulation w.r.t. the robot position on the Occupancy Grid Map

- **USD（Unified Sensory Data）**
  - **Occupancy Grid Map**
  - **Vicinity of the Robot**
  - **Integration of Every Data from the Sensors**

筑波大学 University of Tsukuba

Unification of Behavior Description Program

- **Try to make the program independent from specific types of the sensors**



筑波大学 University of Tsukuba

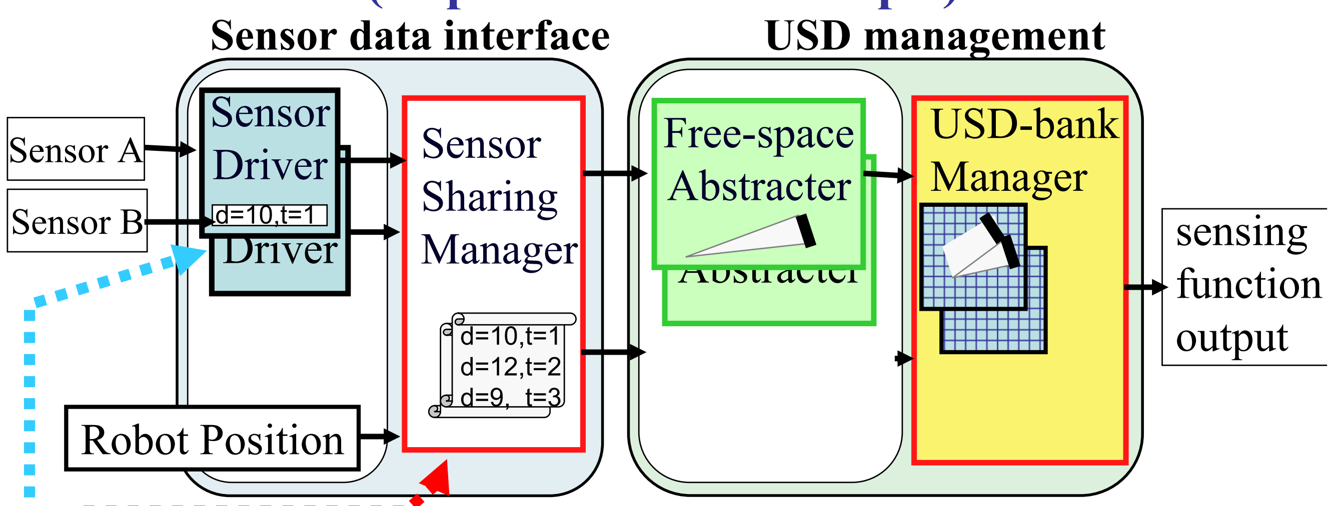
Implementation Issue (Example)

A Mobile Robot Software System Architecture with Unified Sensory Data Integration

筑波大学 University of Tsukuba

Integration of Sensory Information

(Implementation Example)



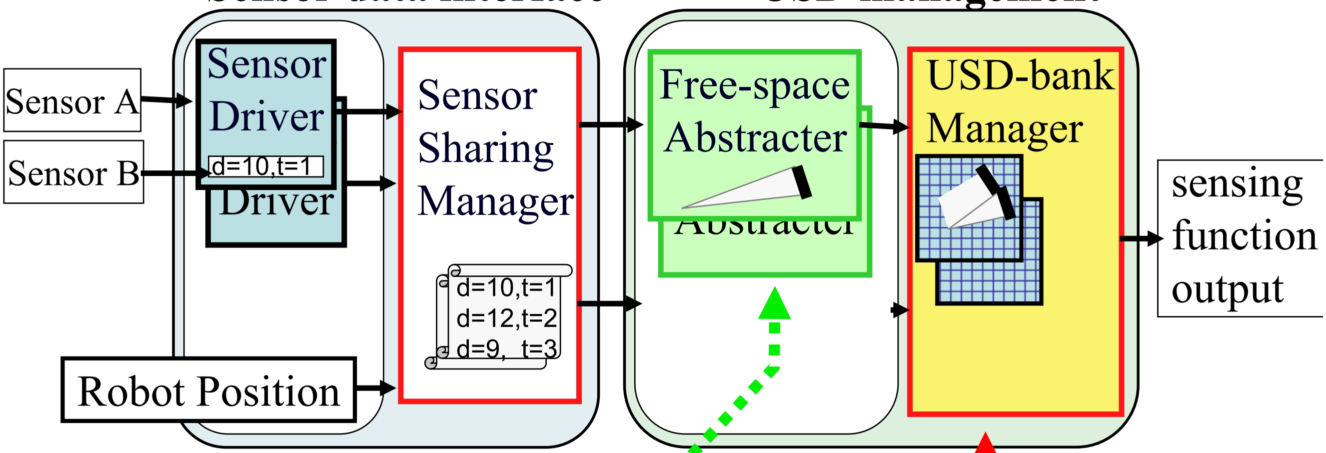
Sensor Driver

-Get sensory data continually and put a time stamp

Sensor Sharing Manager

-Shared memory to hold sensory data temporarily

Integration of Sensory Information
(Implementation Example)



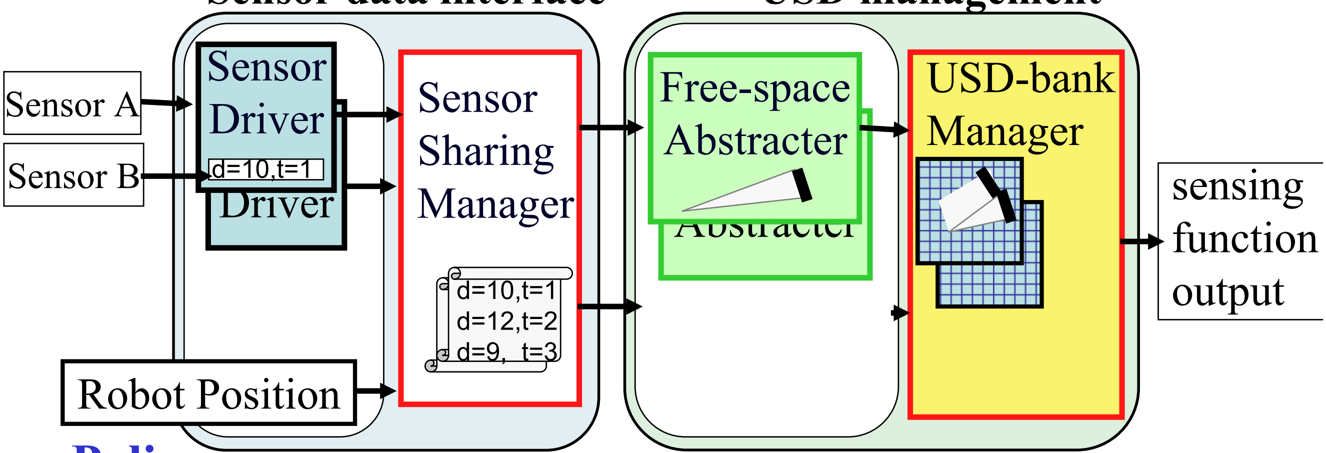
Free Space Abstracter

-Probabilistic map of the space from sensory data and robot position

USD-bank Manager

-Draw/erase management for the grid attributes in the map

Integration of Sensory Information
(Implementation Example)



Policy

- Modularization of sensor intrinsic part

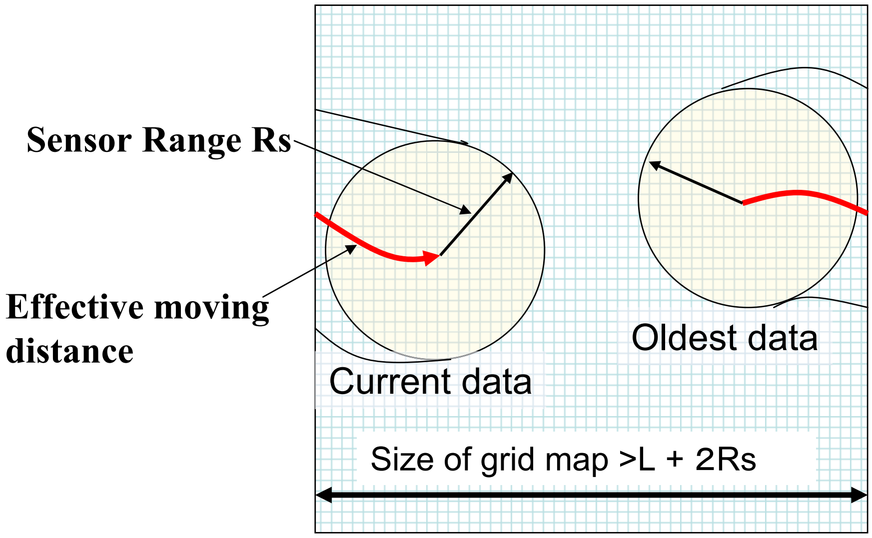
- Sharing common processing parts

Merit

- Easy supplementary of the new sensor

Contraction of USD-bank for Size Limit

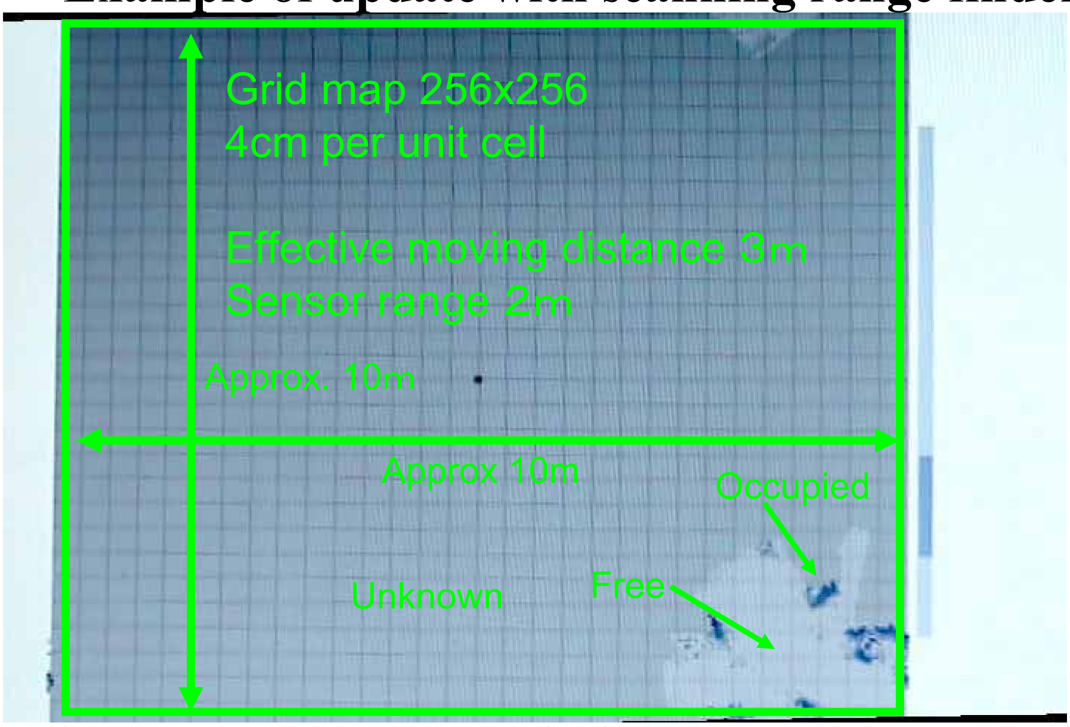
- **Limit the size of the map in the vicinity of the robot**
  - 2-dimensional ring-buffering
  - Erase old data. Maintain data within effective moving distance



筑波大学
University of Tsukuba

Contraction of USD-bank for Size Limit

- **Example of update with scanning range finder**

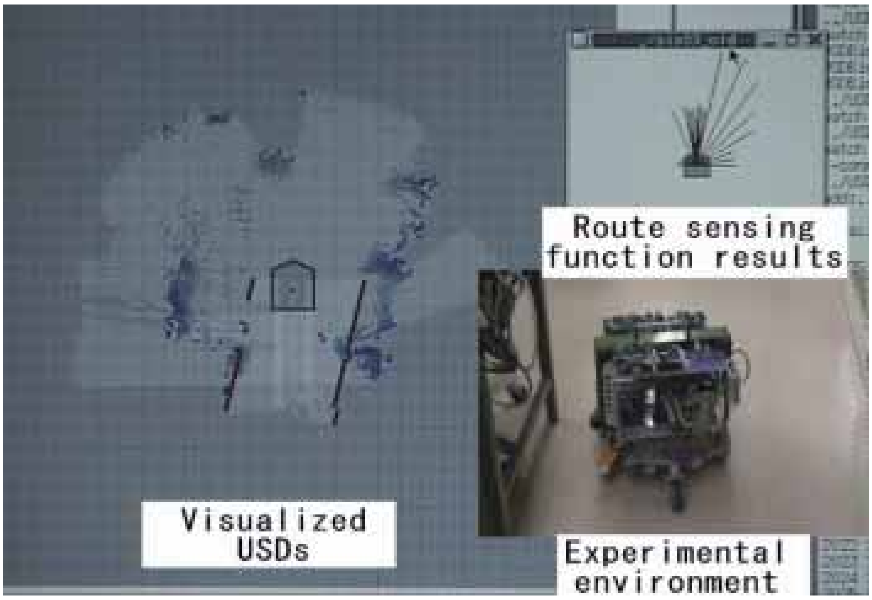


筑波大学
University of Tsukuba

Time-stamping for Sensory data

- Sensory data processing is time consuming
  - Retro-active synchronization between robot position and sensory information

➡Accurate Mapping



筑波大学
University of Tsukuba

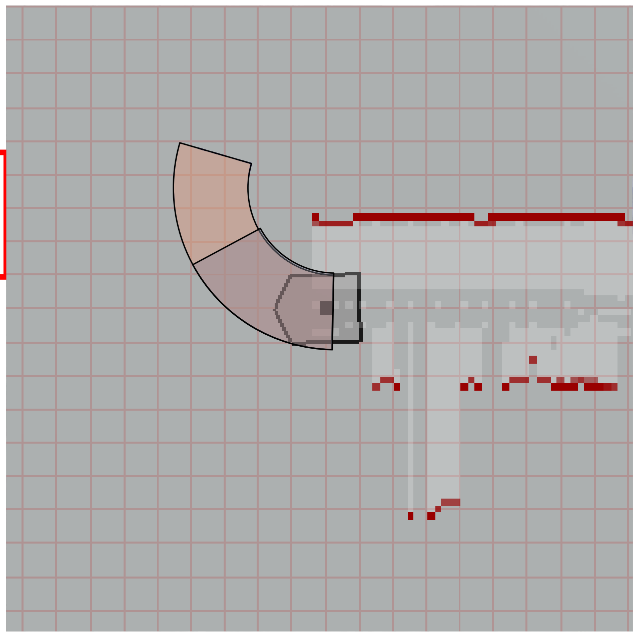
An Example of the Use of USD-bank

- Virtual sensing utilizing the occupancy grid map (=USD-bank)

Virtual Sensing Function =

Return TRUE if Free
in expected path in near future

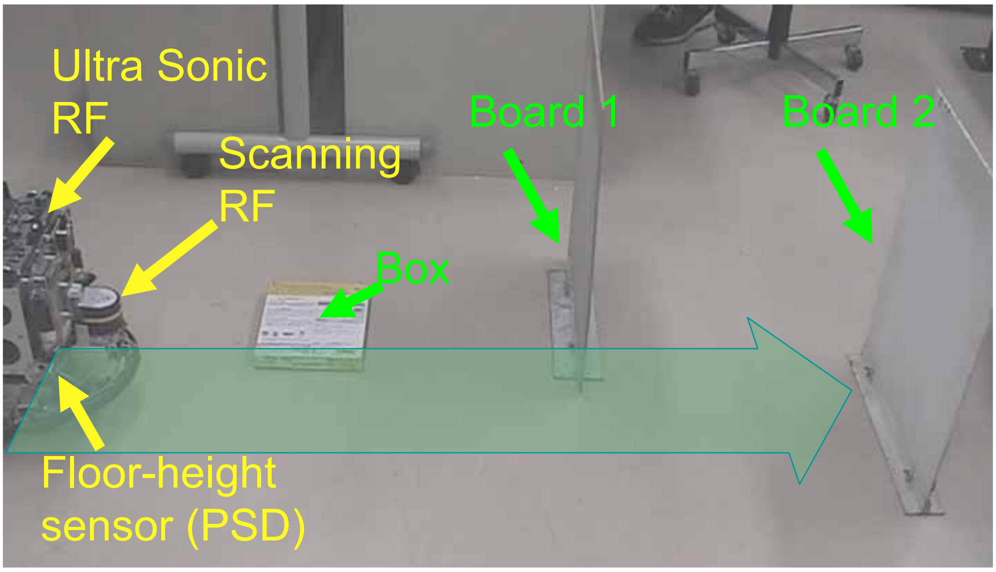
If there are several mapping from different sensors, most inside obstacle is examined.

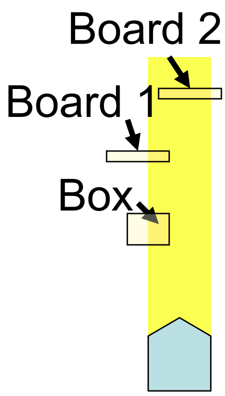


筑波大学 University of Tsukuba

Unified Behavior Description Program

"Go forward, if obstacle within 50cm in front, then stop and go backward"



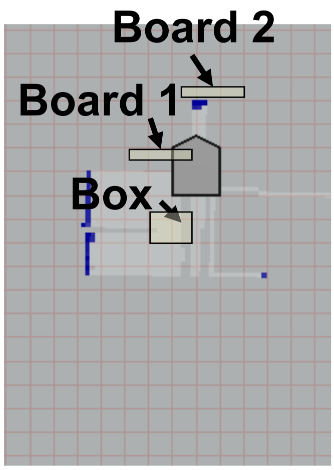


In the following 3 videos, the same program will be demonstrated, but supplement sensors.

筑波大学 University of Tsukuba

Unified Behavior Description Program

"Go forward, if obstacle within 50cm in front, then stop and go backward"

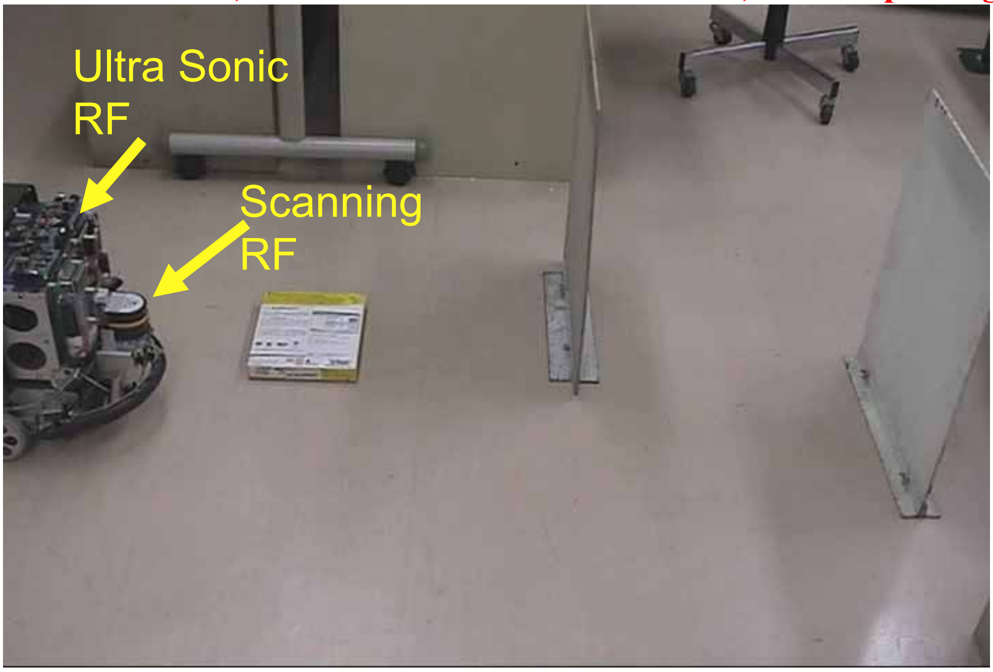


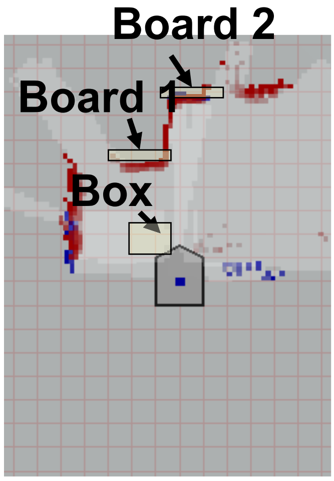
Ultra Sonic Range Finder Only

筑波大学
University of Tsukuba

Unified Behavior Description Program

"Go forward, if obstacle within 50cm in front, then stop and go backward"



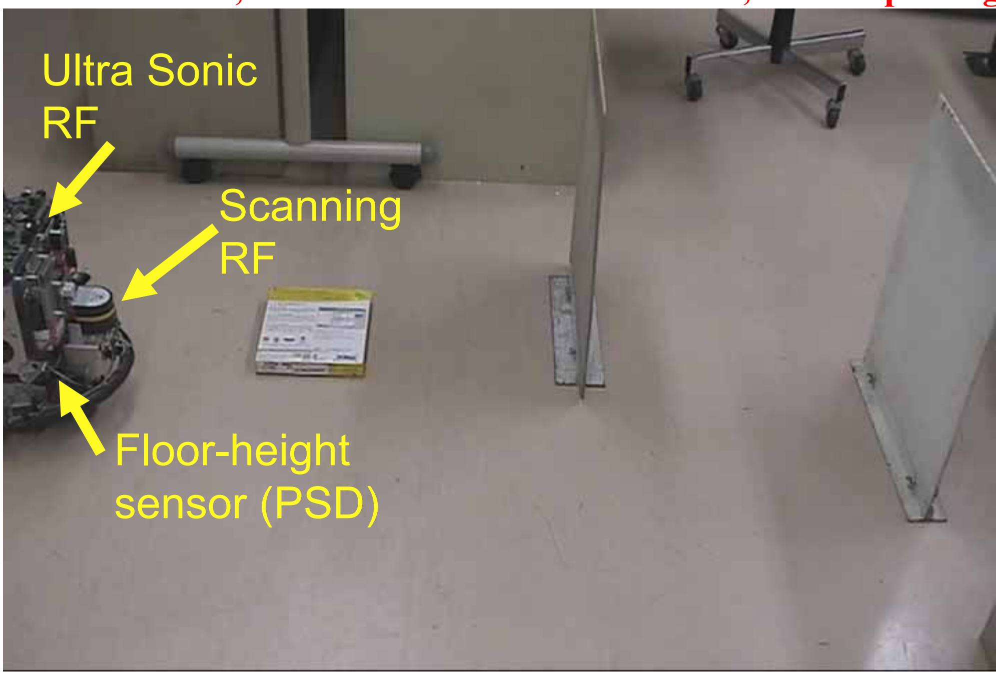


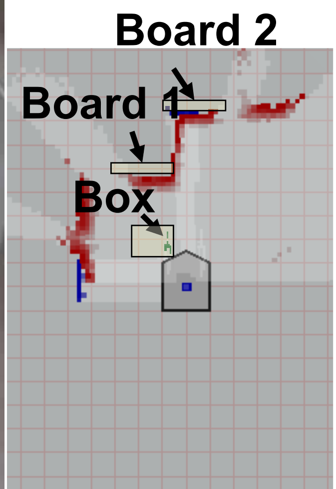
Ultra Sonic and Scanning Range Finder are uilized.

筑波大学
University of Tsukuba

Unified Behavior Description Program

"Go forward, if obstacle within 50cm in front, then stop and go backward"





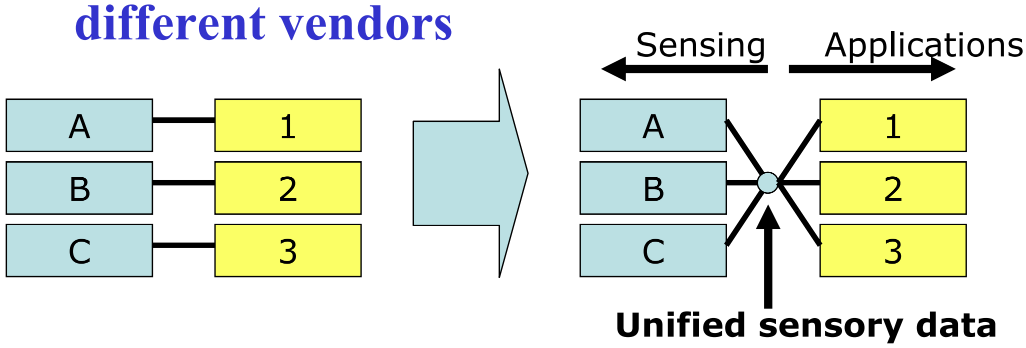
All the three sensors are utilized

What is Realized?
(Major Contributions)

- The behavior description program employs "virtual sensing function" which refers USD-bank.
- The behavior description program was not modified, even though sensors are supplemented.

Implications?

- Making the system with flexibility
  - Increase combinations by modularization
- Consider "independent one is independent"
  - Make it easy to independent development by different vendors

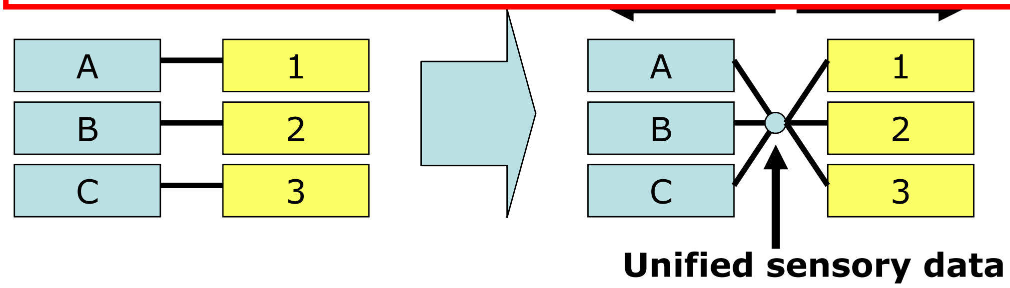


筑波大学 University of Tsukuba

Implications?

- Making the system with flexibility
  - Increase combinations by modularization
- ... ent”

Establishment of business model for robots



筑波大学 University of Tsukuba

Unification, Abstraction and Modularization (toward standard)

- Unify the treatment, to examine same kinds of materials within several types
- Search simple cross-section for the unification
  - with possibly larger “resolution” to realize desired function.

robotics2006-02-15

Navigation of mobile robots

including mapping, localization, and mapping

2006. 2. 15.

Nakju Lett Doh
Yucheol Lee
Wonpil Yu

Intelligent Task Control Team
Intelligent Robot Research Division
ETRI Electronics and Telecommunications Research Institute

Introduction

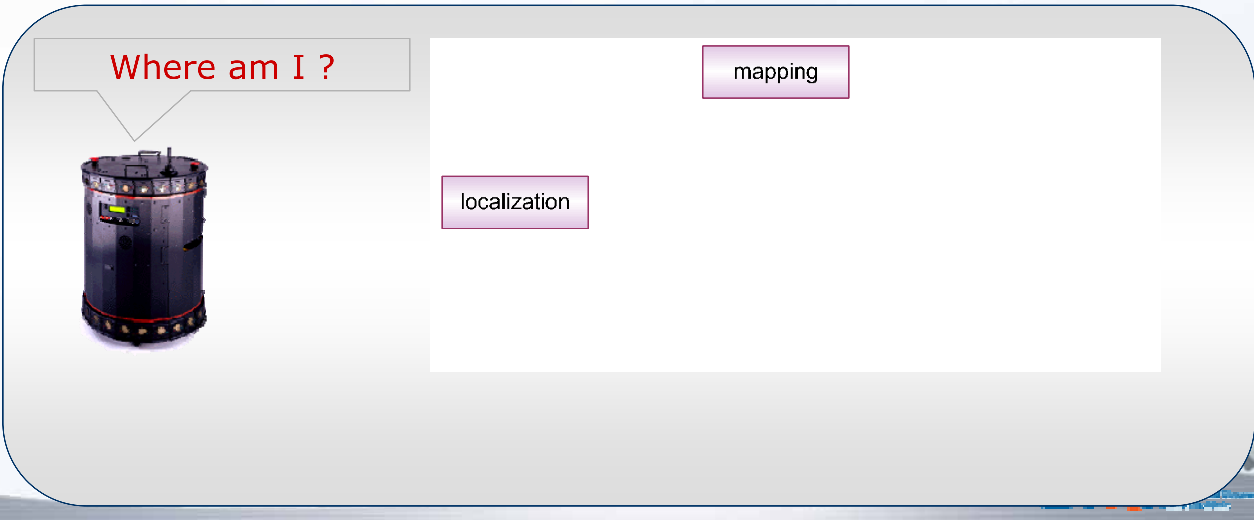
- **Navigation: Key component in mobile robots**
- **Sub-techniques of navigation**
  - **[Mapping]** representing world in a consistent manner using accumulated sensor data

Where I have been?

mapping

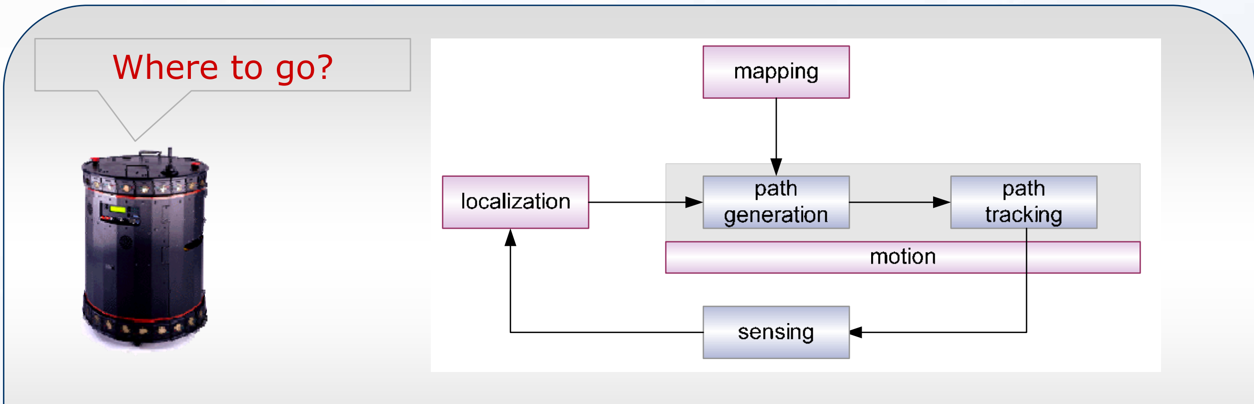
Introduction

- Navigation: Key component in mobile robots
- Sub-techniques of navigation
  - **[Mapping]** representing world in a consistent manner using accumulated sensor data
  - **[Localization]** knowing the posture of robot



Introduction

- Navigation: Key component in mobile robots
- Sub-techniques of navigation
  - **[Mapping]** representing world in a consistent manner using accumulated sensor data
  - **[Localization]** knowing the posture of robot
  - **[Motion]** generating path and tracking the path

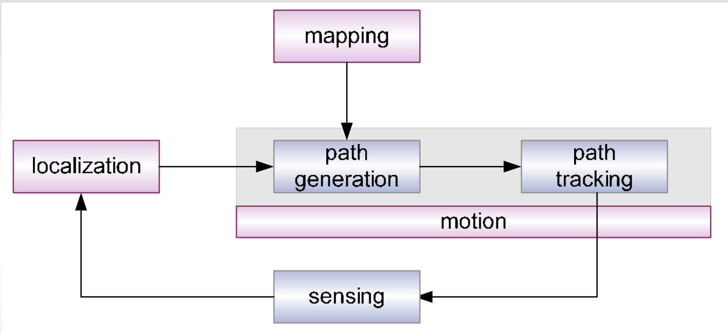


We suggest a standard for navigation

Introduction

- **Navigation: Key component in mobile robots**
- **Sub-techniques of navigation**
  - **[Mapping]** representing world in a consistent manner using accumulated sensor data
  - **[Localization]** knowing the posture of robot
  - **[Motion]** generating path and tracking the path

Where to go?



We suggest a standard for navigation

Another implementation

☐ Current problem of robots

- Stand-alone robot -> high cost
- Large calculation -> low performance

High performance

- Main calculation by server
- High level multimedia service

conventional robot

Network ability

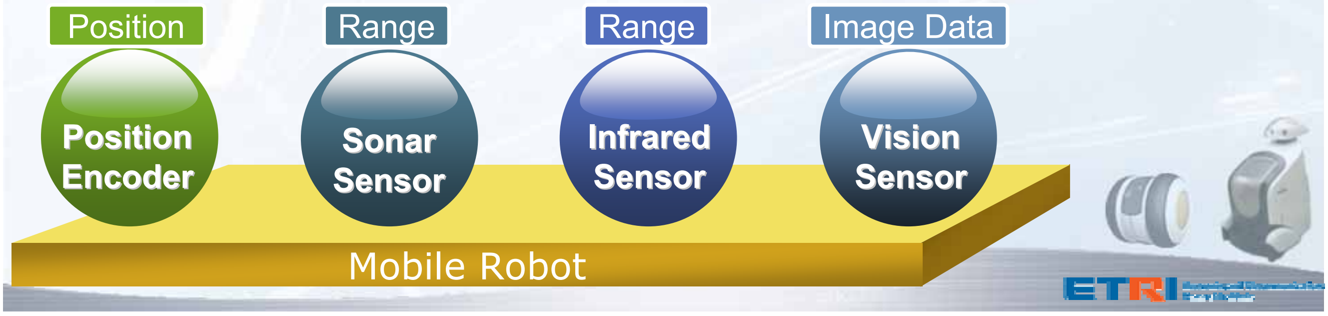
Cost down

- Mobile Platform을 갖춘 Multi function 로봇에서 개념 확대
- 플랫폼 가격 인하
- 다양한 기술적 접근

Another implementation: URC (Ubiquitous Robotic Companion)

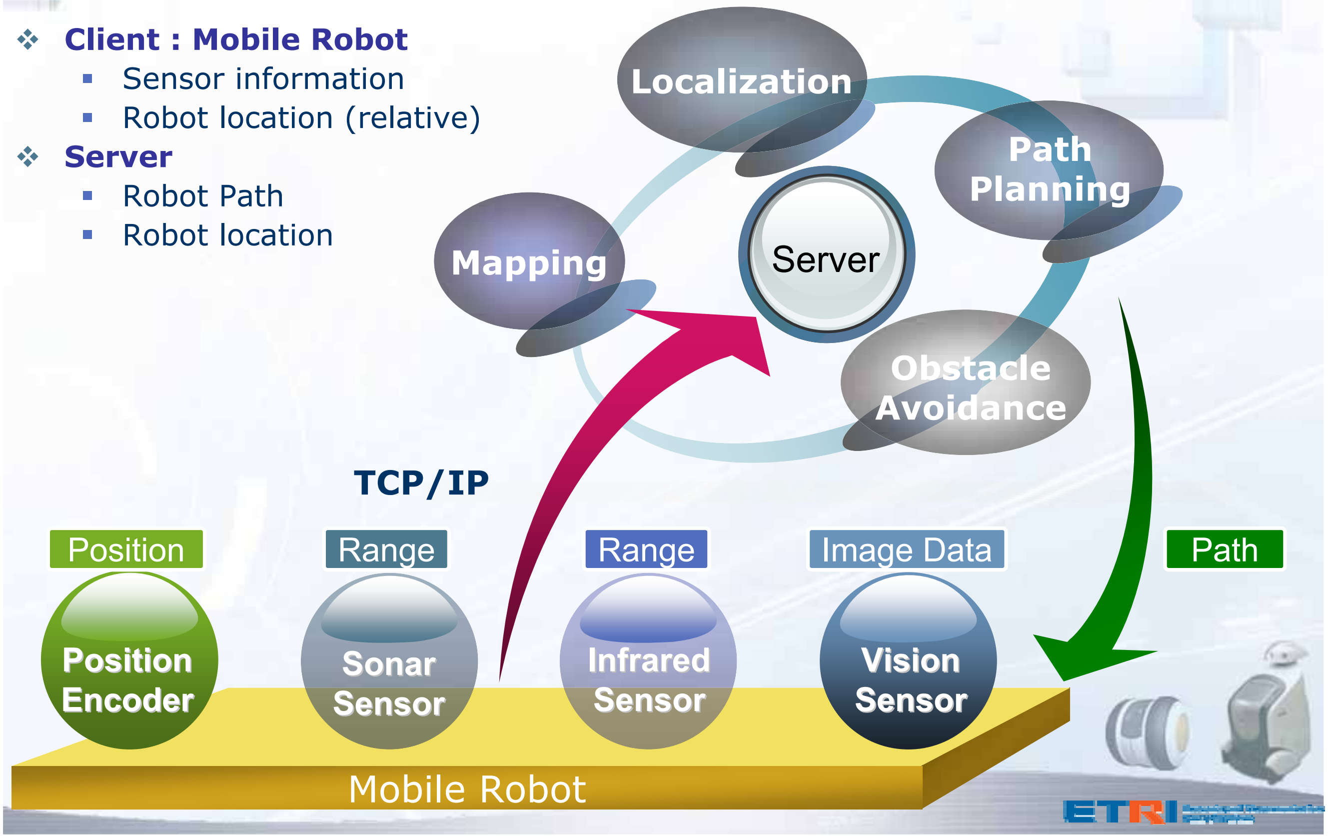
URC Implementation

- **Client : Mobile Robot**
  - Sensor information
  - Robot Position



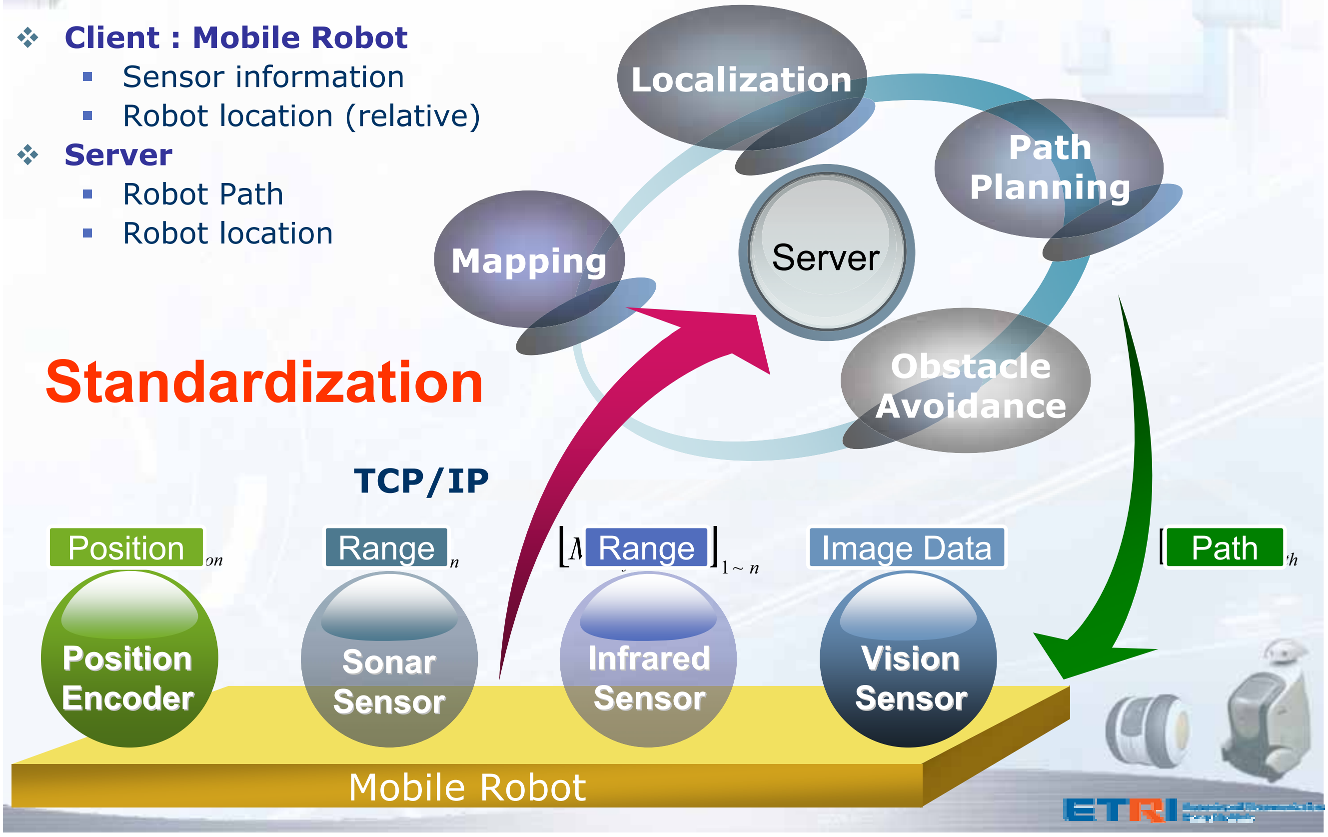
URC Implementation

- **Client : Mobile Robot**
  - Sensor information
  - Robot location (relative)
- **Server**
  - Robot Path
  - Robot location



URC Implementation

- **Client : Mobile Robot**
  - Sensor information
  - Robot location (relative)
- **Server**
  - Robot Path
  - Robot location



Concluding remarks

Standard for navigation

- Mapping / localization / motion
- A candidate is suggested

Usage of server / client system

- As an alternative way, a usage of server / client system is suggested

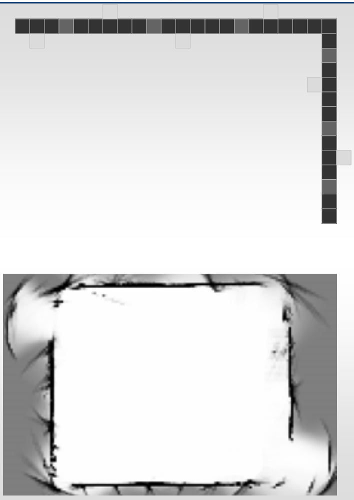
Thank you for your attention!

Map representation

- **3 types widely used maps**
  - Grid map
  - Feature map
  - Topological map

Map representation

- 3 types widely used maps
  - Grid map
  - Feature map
  - Topological map

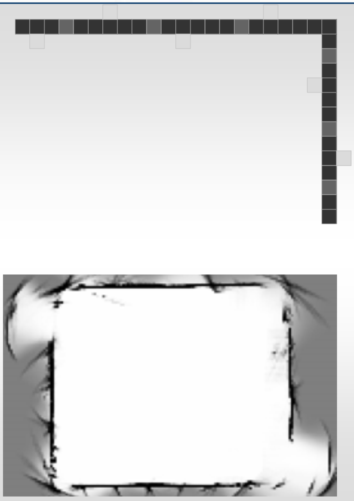


- Representing the space as a combination of grids
- Each grid can be filled by an integer value to represent its occupancy
- Grid position is denoted with respect to a global coordinate.

cont'd

Map representation

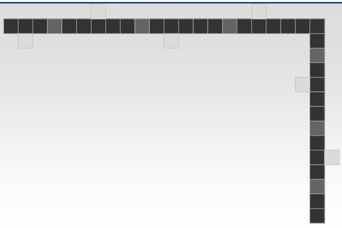
- 3 types widely used maps
  - Grid map
  - Feature map
  - Topological map



- Advantage
  - Easy to use
  - Arbitrary object can be represented
- Disadvantage
  - Large memory requirements

Map representation

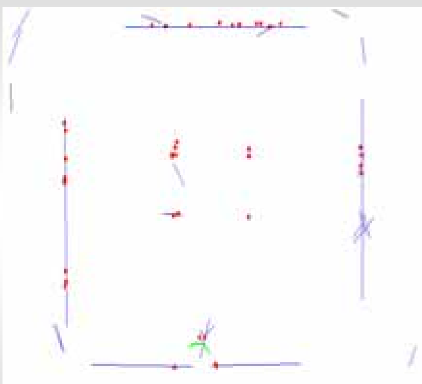
- **3 types widely used maps**
  - Grid map
  - Feature map
  - Topological map



- [Map size] physical size of the map (L_x, L_y)
- [Map position] Left/top & right/bottom position of the map $(X,Y)_{left/top}$ & $(X,Y)_{right/bottom}$
- [Cell size] size of each cell S_{cell}
- [Number of cell] Number of cells in x, y coordinate (N_x, N_y)
- [Coordinate] A coordinate from which the map is represented $(X_{coord}, Y_{coord}, \theta_{coord})$

Map representation

- **3 types widely used maps**
  - Grid map
  - Feature map
  - Topological map



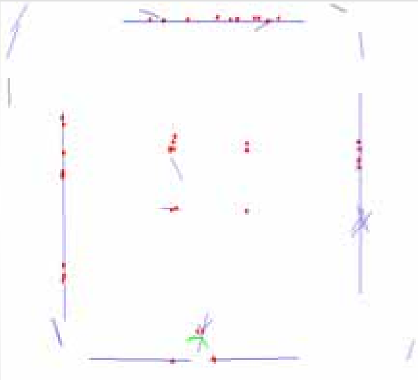
- **Representing the space as a combination of features such as line, are, point, etc**
- **Low memory requirement**
- **Hard to represent an arbitrary shape**

cont'd

Map representation

❖ **3 types widely used maps**

- Grid map
- Feature map
- Topological map



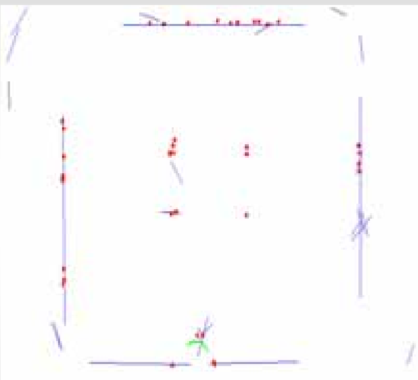
- [Map size] physical size of the map (L_x, L_y)
- [Map position] Left/top & right/bottom position of the map $(X,Y)_{left/top}$ & $(X,Y)_{right/bottom}$
- [Coordinate] A coordinate from which the map is represented $(X_{coord}, Y_{coord}, \theta_{coord})$

cont'd

Map representation

❖ **3 types widely used maps**

- Grid map
- Feature map
- Topological map



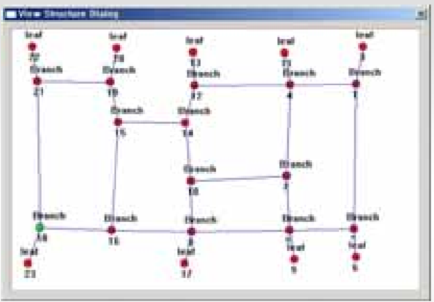
❖ **[Point]**
- position: $(X,Y)_{point}$
- visible angle: $(\theta_{low}, \theta_{upper})$

❖ **[Line]**
- Position: $(X,Y)_{start}$ & $(X,Y)_{end}$
- Visible angle $(\theta_{low}, \theta_{upper})_{start}$ & $(\theta_{low}, \theta_{upper})_{end}$

❖ **[Arc]**
- position: $(X,Y)_{start}$ & $(X,Y)_{end}$
- radius: R_{arc}
- visible angle: $(\theta_{low}, \theta_{upper})_{start}$ & $(\theta_{low}, \theta_{upper})_{end}$

Map representation

- **3 types widely used maps**
  - Grid map
  - Feature map
  - Topological map

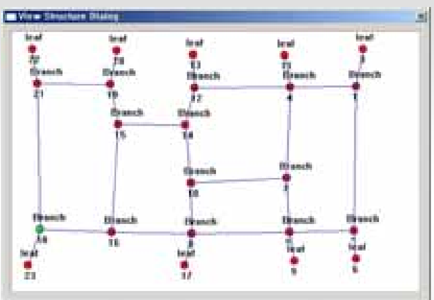


- **Representing the space as a combination of**
  - nodes: topologically meaningful place
  - edges: connections between nodes
- **An abstraction of the environment**
- **Good for a corridor like environment**

cont'd

Map representation

- **3 types widely used maps**
  - Grid map
  - Feature map
  - Topological map



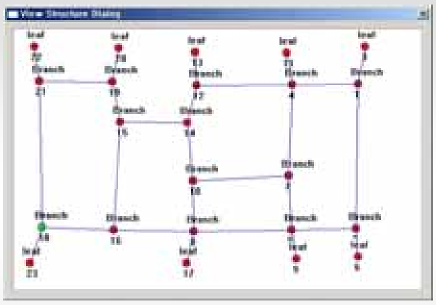
- [Map size] physical size of the map (L_x, L_y)
- [Map position] Left/top & right/bottom position of the map $(X,Y)_{left/top}$ & $(X,Y)_{right/bottom}$
- [Coordinate] A coordinate from which the map is represented $(X_{coord}, Y_{coord}, \theta_{coord})$

cont'd

Map representation

❖ 3 types widely used maps

- Grid map
- Feature map
- Topological map



- Node
  - position: $(X,Y)_{node}$
  - node ID: ID_{node}
  - number of edges: N_{edge}
  - edge data
- Edge
  - connected node's ID: $ID_{connected}$
  - length: L_{edge}
  - shape: $(X,Y)_0, (X,Y)_1, \cdots, (X,Y)_i$

Localization

❖ Localization

- Relative localization (dead-reckoning)
  - Knowing the position of robots with respect to a certain coordinate (or start point)
  - In general, robot uses internal sensors such as odometry, gyroscope, compass, etc.
- Absolute localization
  - Knowing the position of robots with respect to a global coordinate (or origin of map)
  - In general, external sensors (such as GPS, artificial landmarks, etc) are used
  - Kidnap recovery: knowing the position of the robot during a initialization of robots

absolute localization

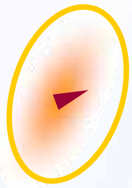
relative localization

Internal sensor
Gyroscope, Compass
Odometry

Localization

- **Standard candidate**
  - robot posture: $(X_{robot}, Y_{robot}, \theta_{robot})$
  - based coordinate: $(X_{coord}, Y_{coord}, \theta_{coord})$
  - data flag (success, failure): $flag$
  - uncertainty representation

$$\Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{y\theta} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta\theta} \end{bmatrix}$$



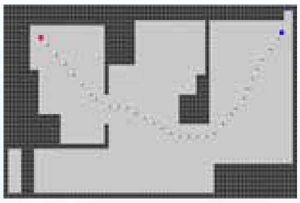
Motion

- **Motion**
  - Path planning
    - Use of localization and mapping
    - Generate an optimal path

Motion

Motion

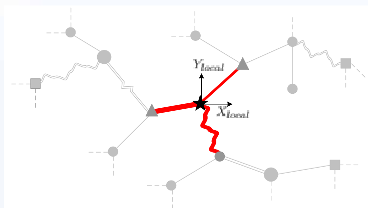
- Path planning
  - Use of localization and mapping
  - Generate an optimal path
  - Grid / feature map
    - **Series points:** $(X,Y)_0,(X,Y)_1,\cdots,(X,Y)_t$
    - **Coordinate:** $(X_{coord},Y_{coord},\theta_{coord})$



Motion

Motion

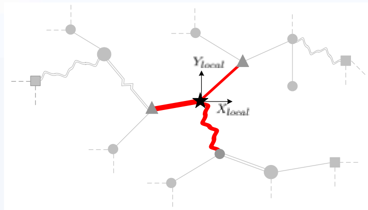
- Path planning
  - Use of localization and mapping
  - Generate an optimal path
  - Grid / feature map
    - **Series points:** $(X,Y)_0,(X,Y)_1,\cdots,(X,Y)_t$
    - **Coordinate:** $(X_{coord},Y_{coord},\theta_{coord})$
  - Topological map
    - **ID of nodes and edges:** $(ID_{node},ID_{edge})_0,(ID_{node},ID_{edge})_1,\cdots,(ID_{node},ID_{edge})_t$



Motion

Motion

- Path planning
  - Use of localization and mapping
  - Generate an optimal path



  - Grid / feature map
    - **Series points:** $(X,Y)_0,(X,Y)_1,\cdots,(X,Y)_n$
    - **Coordinate:** $(X_{coord}, Y_{coord}, \theta_{coord})$
  - Topological map
    - **ID of nodes and edges:** $(ID_{node}, ID_{edge})_0, (ID_{node}, ID_{edge})_1, \cdots, (ID_{node}, ID_{edge})_n$

- Path tracking
  - Give command to robot for path tracking
  - Command
    - **Linear velocity:** v_{robot}
    - **Angular velocity:** ω_{robot}

robotics/2006-02-16

OMG Robotics Systems RFI Response

from

The National Institute of Advanced Industrial Science and Technology – Ubiquitous Functions Research Group

OMG Technical Meeting – Robotics DTF
Tampa (USA - Florida) – February 15th , 2006

Olivier LEMAIRE

National Institute of Advanced Industrial Science and Technology (AIST)

Ubiquitous Functions Research Group

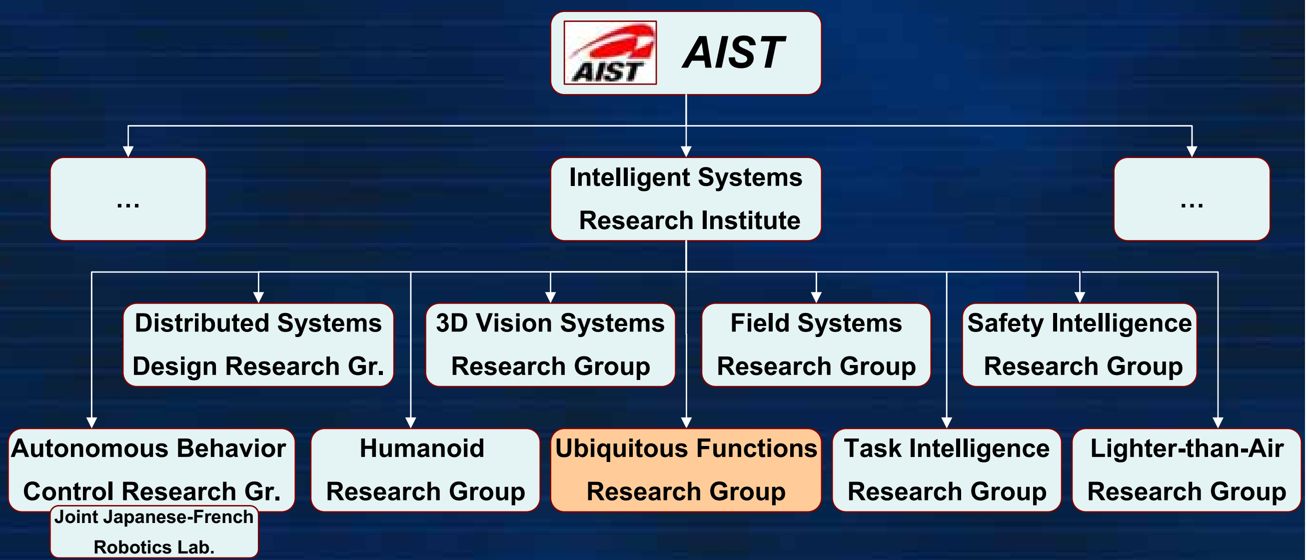
Usage of Robotics Technology

Robotics @ AIST / ISRI

AIST

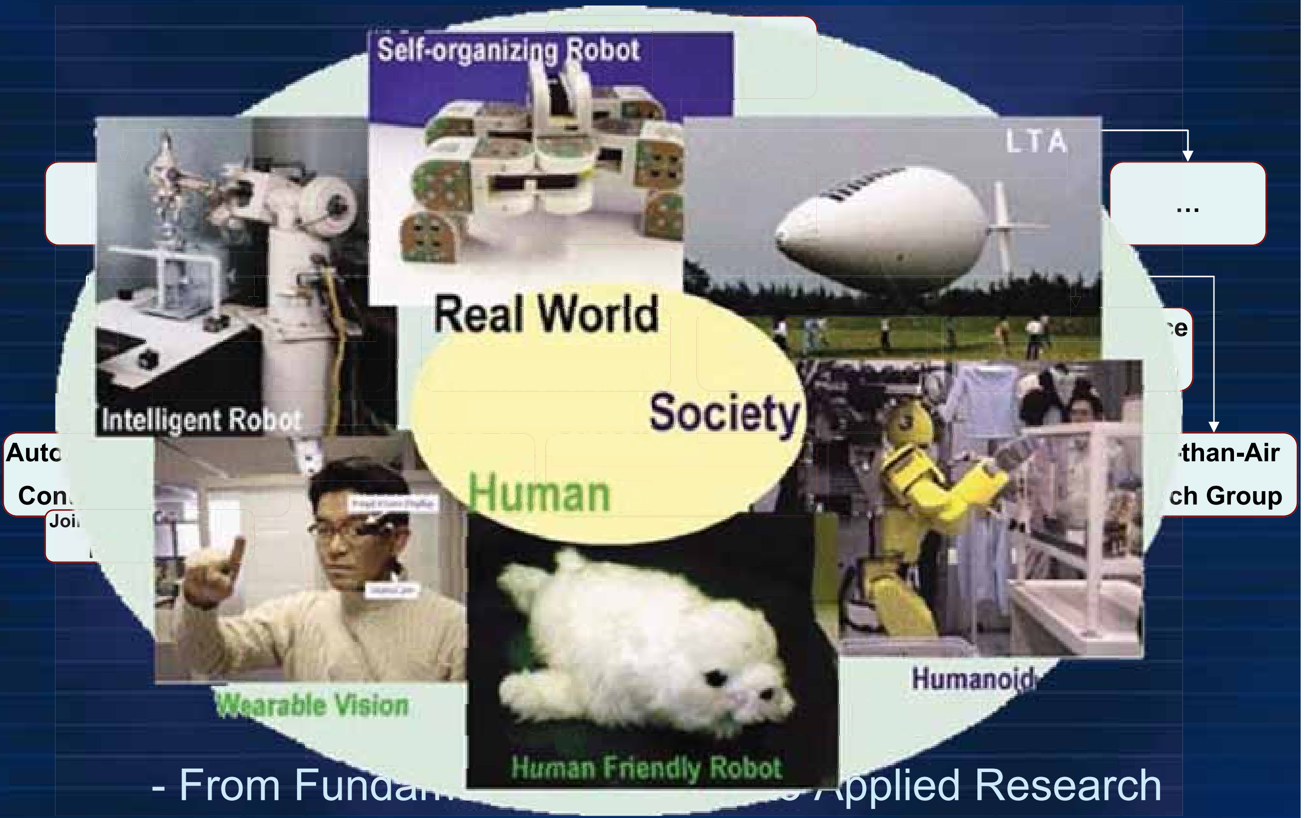
- Largest public research organization in Japan
- 3200 Employees
- 50 Research Centers and Research Institutes
- Research fields including :
    * Life Science & Technology
    * Information Technology
    * Nanotechnology, Materials & Manufacturing
    * Environment & Energy
    * Geological Survey & Applied Geoscience
    * Metrology and Measurement Technology
- From Fundamental Research to Applied Research

Robotics @ AIST / ISRI

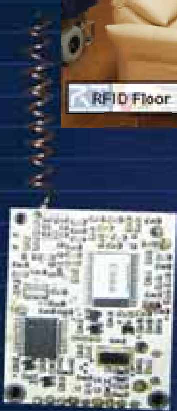


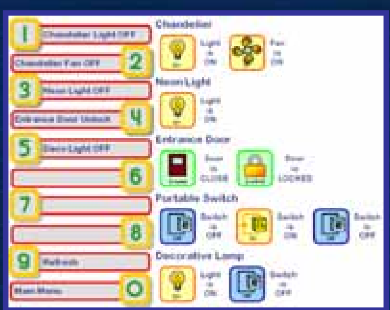
Intelligent Systems Research Institute – Ubiquitous Functions Research Group

Robotics @ AIST / ISRI



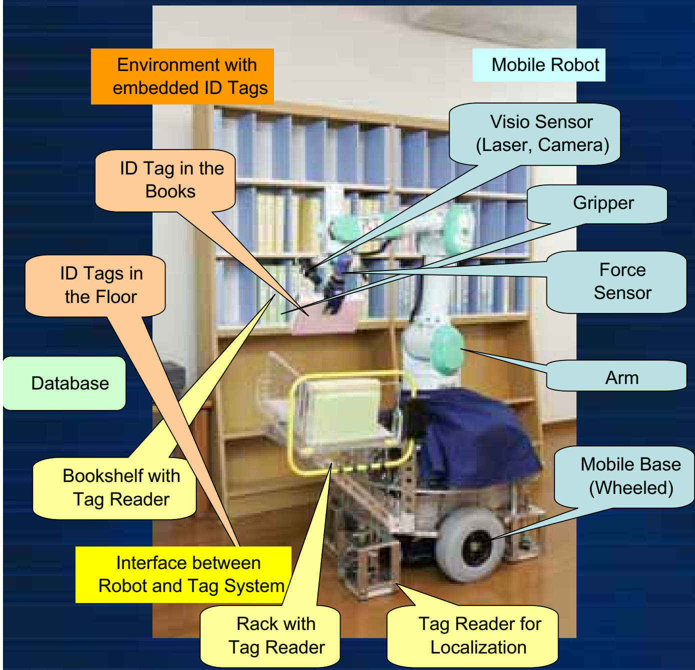
Robotics @ AIST / ISRI / UFRG – Robotic Space





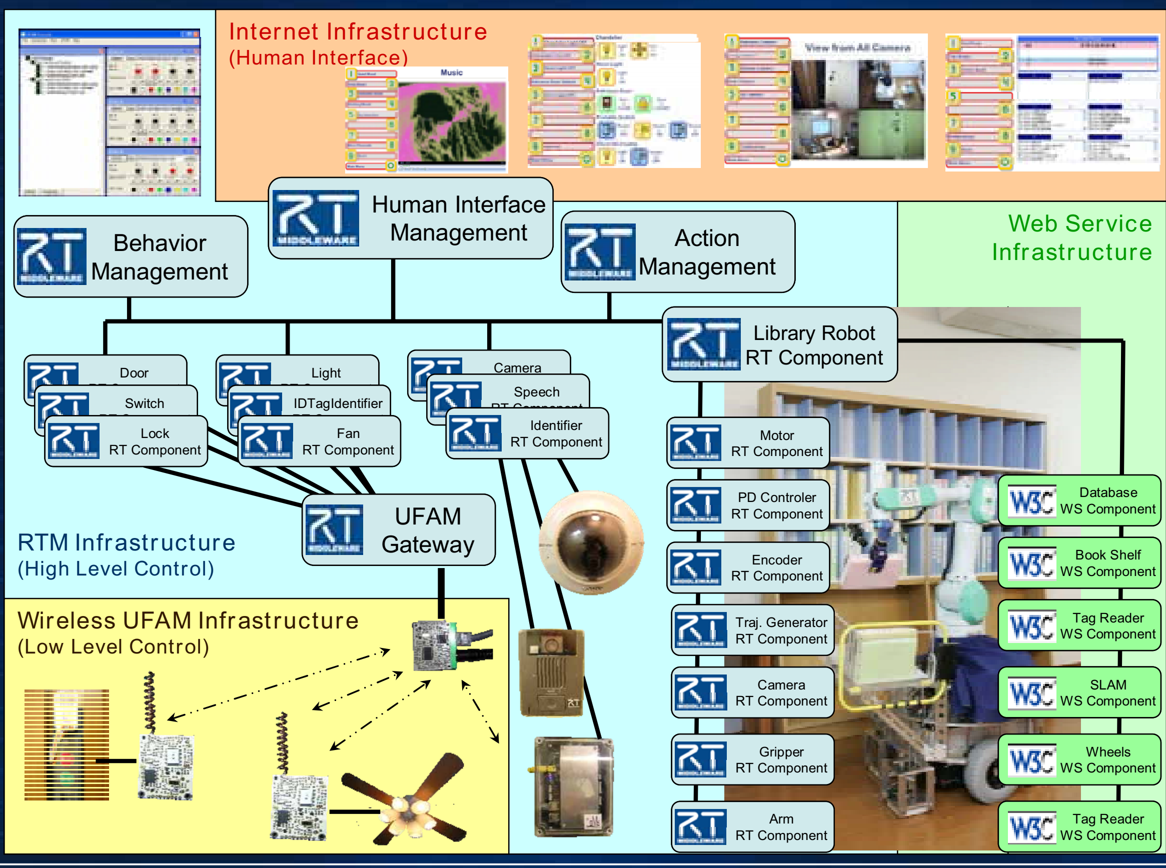
- Robotic Room for Daily life support
- The Room is the Robot where each device provides a simple function
- Sum of functions provides more intelligent functions
- Constraints :
  - Need High Flexibility/Customization
  - Event Driven
  - No a priori design time decision
  - More about Human /system Interaction than control
- Use of wireless Active RFID tags with Digital I/O to control appliances and gather sensor states
  - Use I/O tagging to achieve PnP

Usage of Robotics Technology – Ubiquitous Robots



- Librarian Robot
- Tidy books back to bookshelf
- Minimal on board processing (only sensors)
- All intelligence distributed
- Taking advantage of infrastructure (ID Tags) for many usual robotic tasks
  - Localization, Book Recognition…

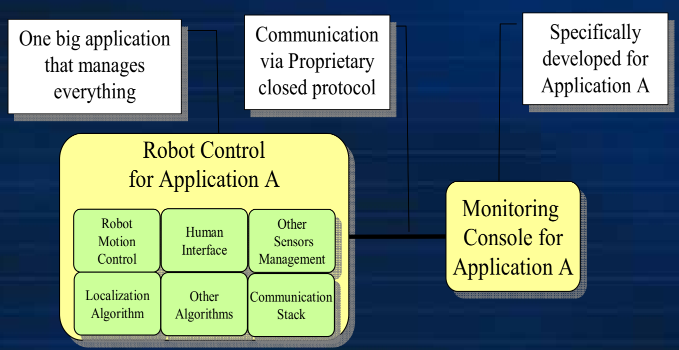
Usage of Robotics Technology – Integration



Needs for Standardization of Robotics Technology

Technology Re-use

- Develop new robots faster and cheaper by integrating COTS

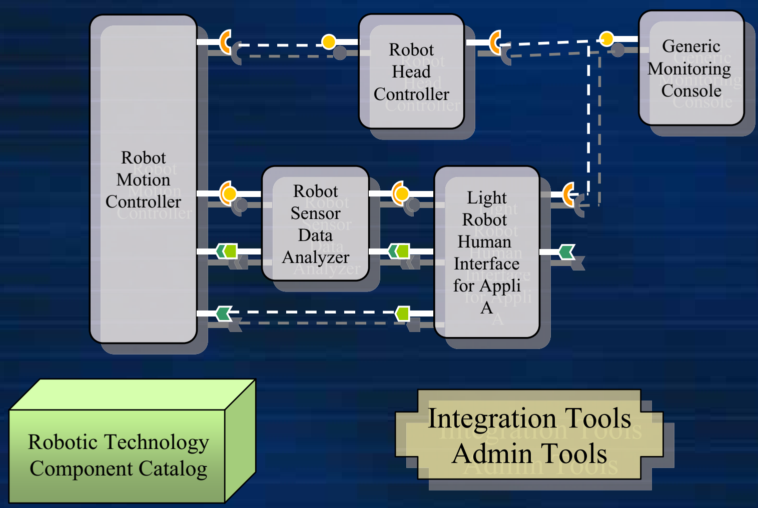


<u>This kind of Monolithic and Specific Application :</u>

- Cannot address a wide enough market to be very profitable
- Takes a long time to develop and to maintain
- Is Error-prone
- Reaches resources saturation for high intelligent system
- Cannot easily integrate with other systems
- Is bound to the hardware manufacturer

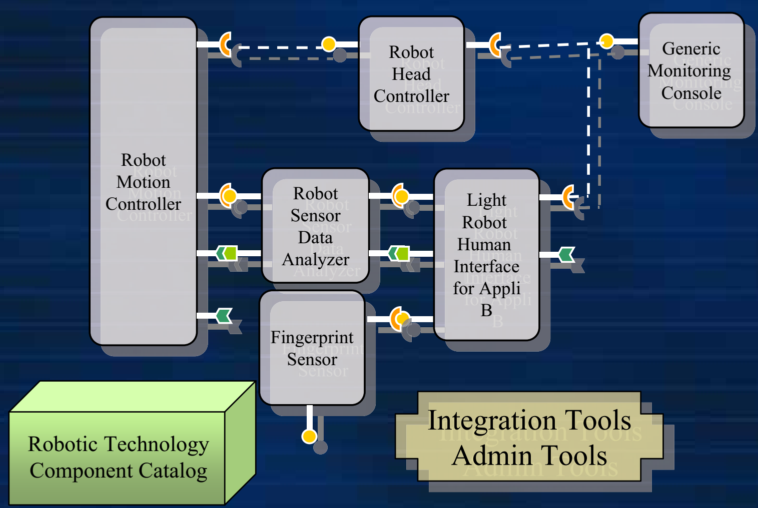
Technology Re-use

- Develop new robots faster and cheaper by integrating COTS



Technology Re-use

- Develop new robots faster and cheaper by integrating COTS

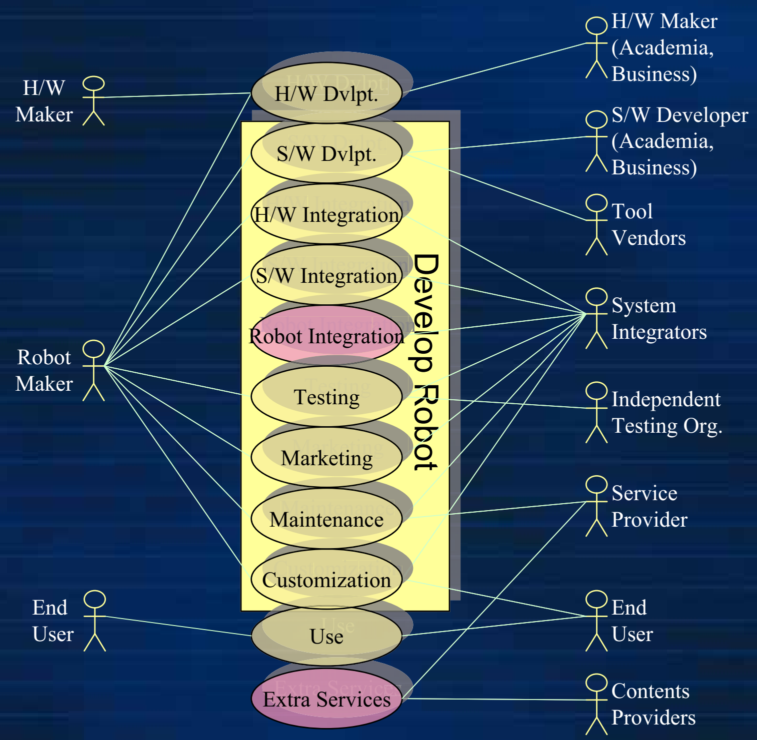


This kind of Modular and Flexible Application :

- Emphasize **reuse** of smaller specialized components
- Takes a short Time to **Integrate**
- Can be **distributed** over a network to share resources
- Is **totally open** to other systems and to customization
- As more accessible, can generate interest from different business actors and stimulate the market

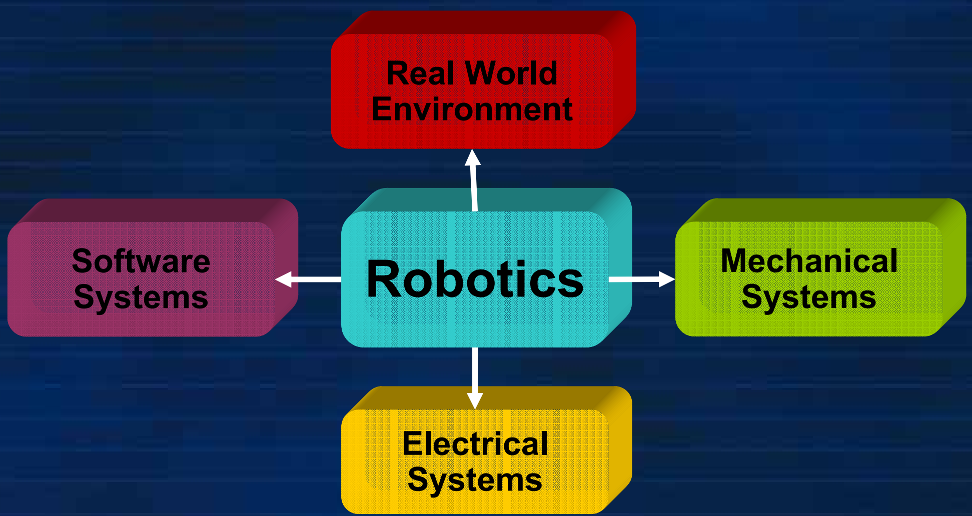
Technology Re-use

- Expand the Robotics Business by sharing the burden of development



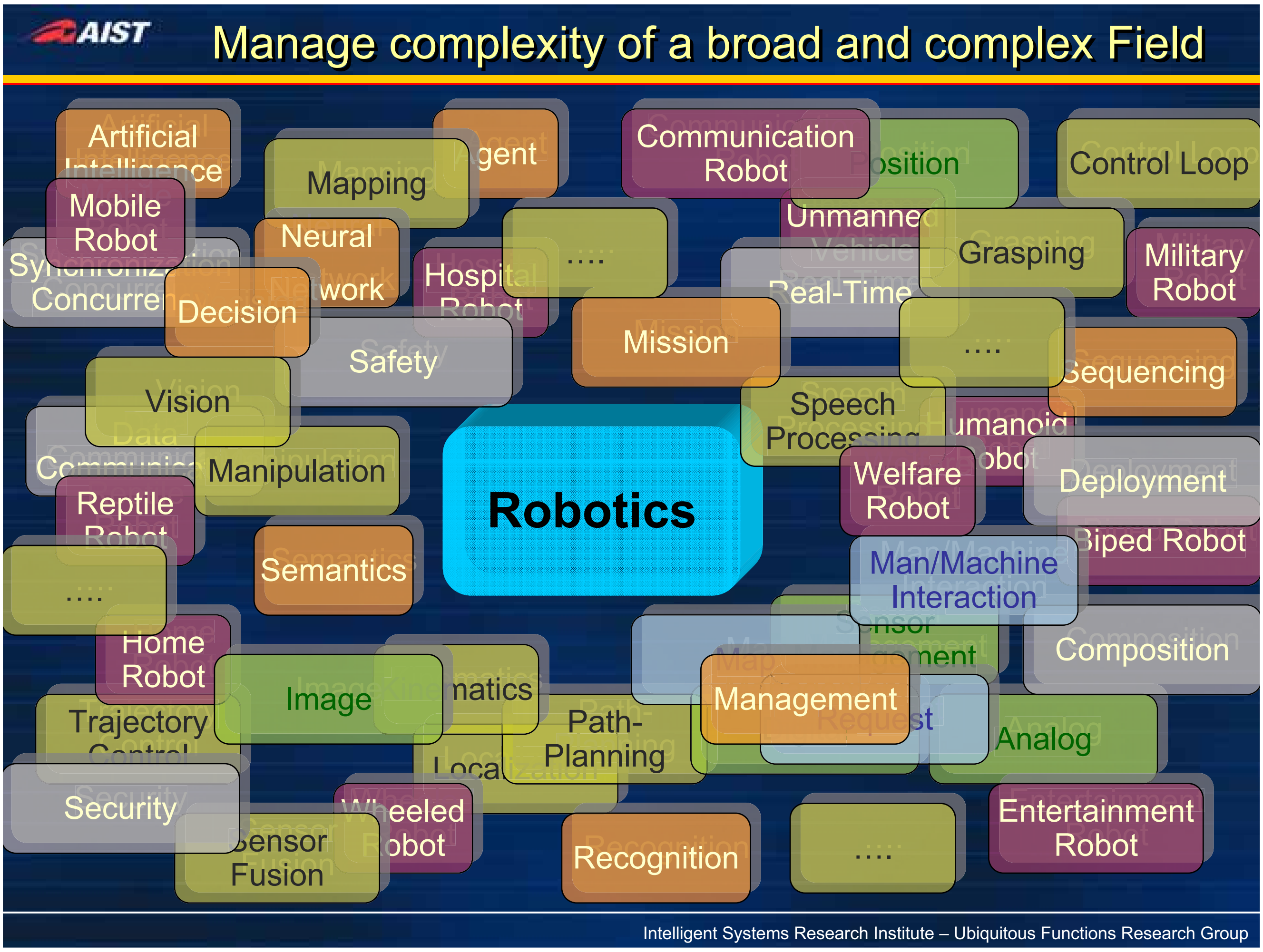
Manage complexity of a broad and complex Field

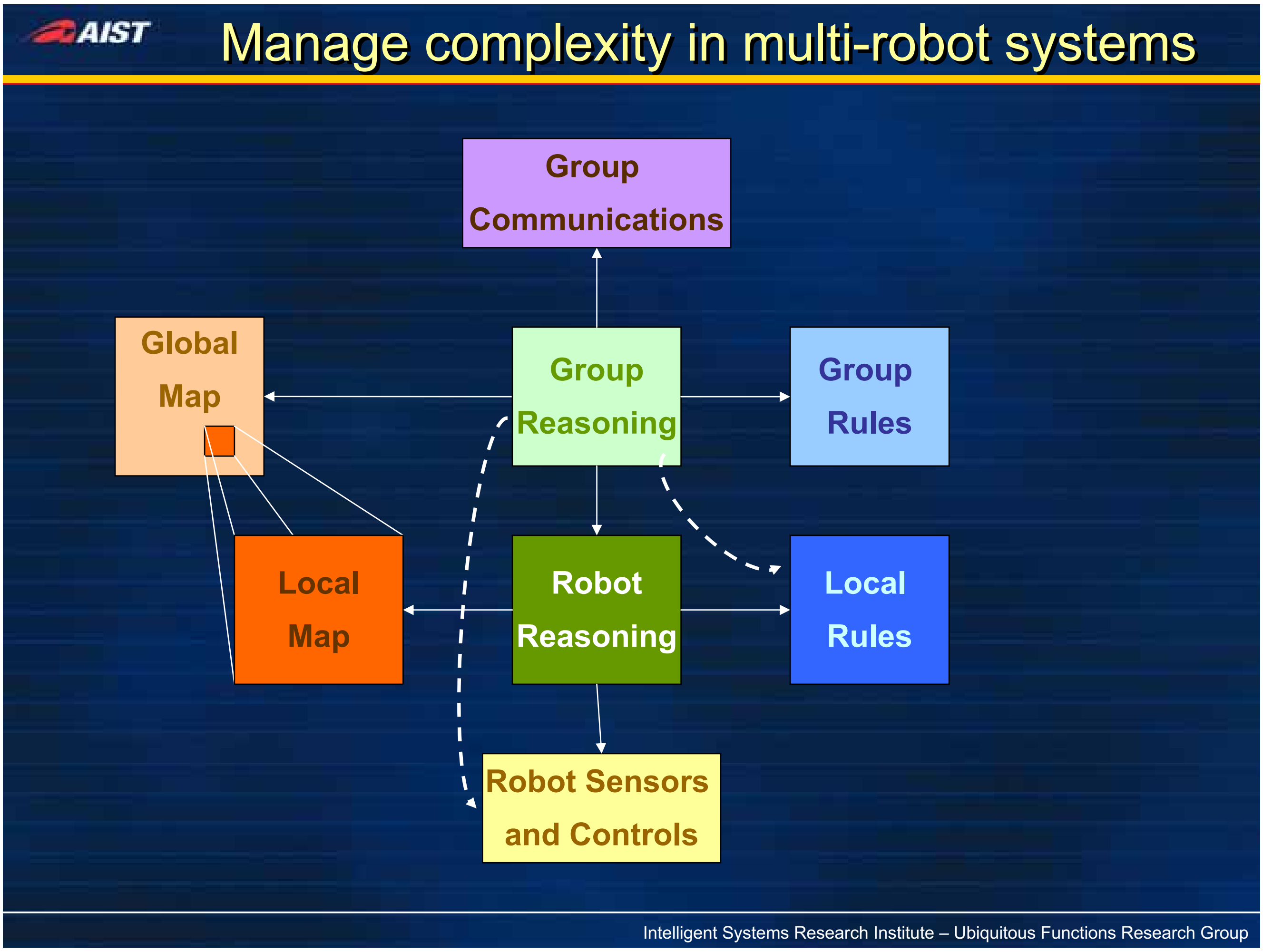
- Robotics is a multi-dimensional discipline
  - Integration of existing sciences
  - Dealing real world environment



- A roboticist must master all Mechanical, Electrical, Software sciences AND deal with the complexity and uncertainties of a real world uncontrolled environment -> Only Superman can do that !!

Intelligent Systems Research Institute – Ubiquitous Functions Research Group

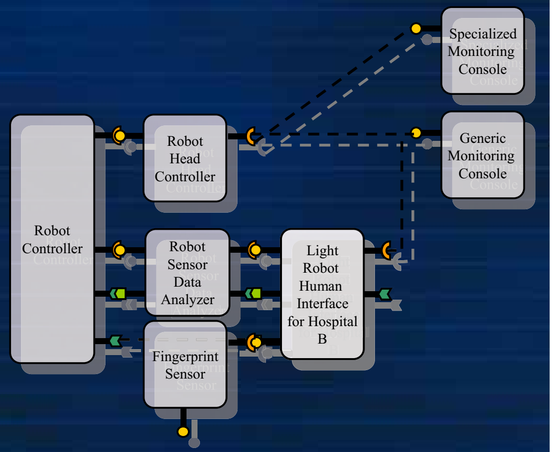

AIST
Manage complexity of a broad and complex Field
Artificial Intelligence
Mapping
Agent
Communication Robot
Position
Control Loop
Mobile Robot
Neural Network
Synchronization
Concurrency
Decision
Hospital Robot
....
Unmanned Vehicle
Grasping
Military Robot
Real-Time
Mission
Safety
....
Sequencing
Vision
Speech Processing
Humanoid Robot
Data Communication
Manipulation
Robotics
Welfare Robot
Deployment
Reptile Robot
Biped Robot
....
Semantics
Man/Machine Interaction
Sensor
Home Robot
Composition
Image
Kinematics
Map
Management
Request
Trajectory Control
Path-Planning
Analog
Localization
Security
Wheeled Robot
Entertainment Robot
Sensor Fusion
Recognition
....
Intelligent Systems Research Institute – Ubiquitous Functions Research Group


AIST
Manage complexity in multi-robot systems
Group Communications
Global Map
Group Reasoning
Group Rules
Local Map
Robot Reasoning
Local Rules
Robot Sensors and Controls
Intelligent Systems Research Institute – Ubiquitous Functions Research Group

Everything but Static

(Dynamic Configuration)

- We cannot shutdown a Robotic System every time an entity is added or removed
- System configuration needs to adapt to hardware mobility in a real-time manner
- **We need a way to handle dynamic connection between components as well as their modeling**
- **We need a way to manage dynamic properties of components**



Question of Point of View

(Inheritance, Composition and Granularity of Components)

- The robot can be considered as a whole or as the sum of its components... Or anything else
- **We need a way to let the same entity provide different views**
- Light with a dimmer is still just a light
- **We need a way to define and advertise complex services without breaking the semantic of the underlying basic services**
- A camera on a mobile robot becomes a mobile camera
- **We need a way to compose, at integration time, functionalities defined at design time**

+

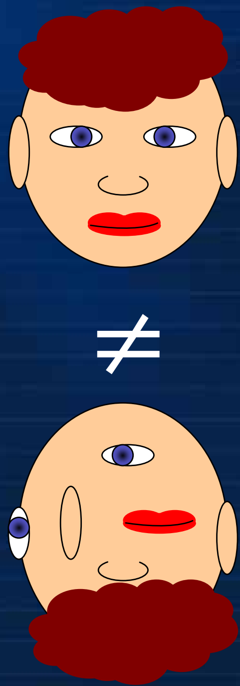
=

A **mobile Robot**
Or
A **Camera**
Or
A **Mobile Camera**
Or
A **Robot with Camera**

Relationships are important

(Spatial and Temporal Integrity)

- Software distributed entities are location independent, but the pieces of hardware they control are not
- **We need a shared representation of the Robotic World and a way to clearly define robotic entities physical relationship**
- Data generated from Robotic Entities may go through several other entities before reaching their destination
  - Delay may be induced
  - Information concerning the origin of the data might be lost
- **We need a way to ensure spatial and temporal integrity of the data**



To go or not to go ?

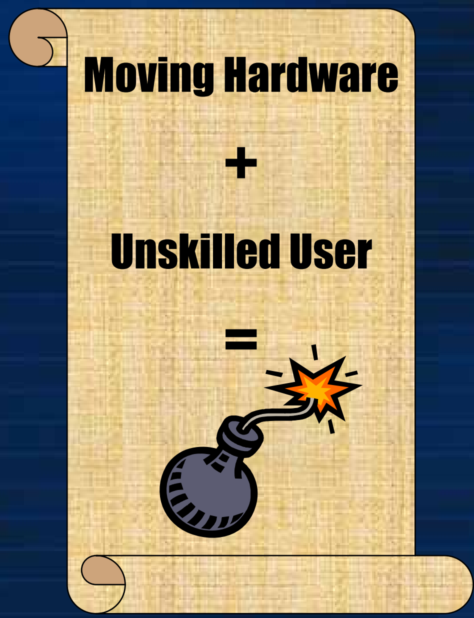
(Conflict Management, Prioritization of Tasks)

- In a distributed Robotic System, we cannot assume anything about the synchronization of task
- Conflicting requests to (especially hardware) entities may lead to incoherent behavior of the system
- Some request may be more important than others (especially safety related ones)
- **We need a way to manage conflicting requests to robotic entities and to prioritize them in an homogenous way**

Safety First

(Conformance)

- Even if a Robotic System cannot be 100% reliable, **it has to be 100% safe !**
- It is **NOT** possible to assert that a robotic system is 100% safe in an unmanaged real world environment
- Companies are reluctant to put on the market a robotic system that could cost them fortunes in potential product liability claims
- **Standards in robotics could help developing a conformance framework defining acceptable safety and performance levels**
- Conforming Robot makers, protected against abusive PL claims, can develop more challenging robots
- User’s **trust** in the technology improves

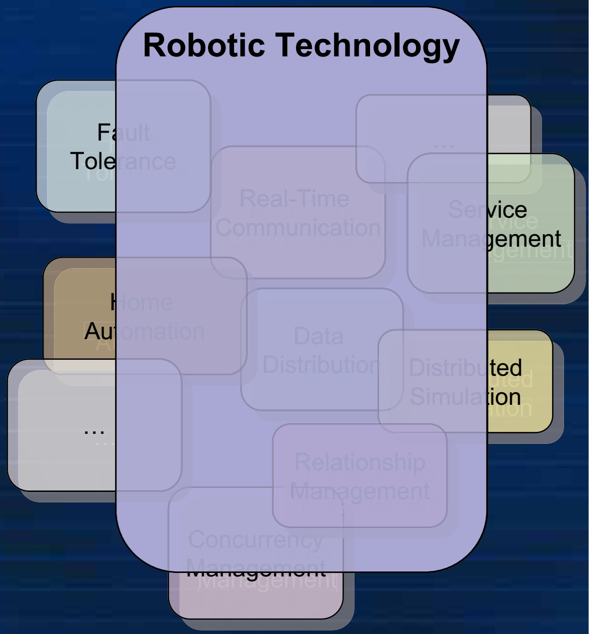


Tools for the people !

- Robotic Systems will be more often used by people with minimal technical skills
- **Robotic Systems must become easier to program and to manage**
- Many tools are necessary to manage the different aspects of the construction of a Robotic System
  - Functionality design
  - Application design
  - Application configuration
  - Simulation
  - Monitoring
- **We need a way to let all these tools homogenously and interchangeably work together**

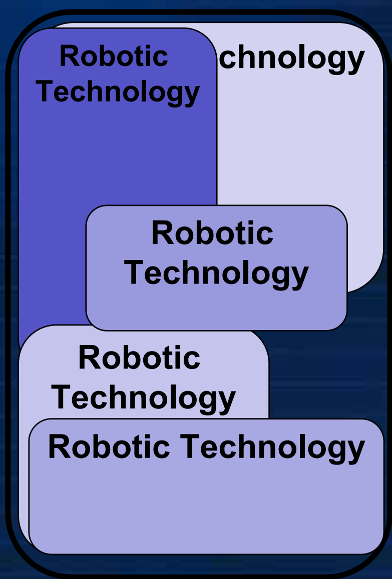
We need a Framework,… not a Patchwork !

- *There are standard frameworks addressing problems related to distributed objects*
- *Most of them are either too general*
  - *- Hard to understand and support*
- *… or too limited*
  - *- Some specific aspects of the robotic technology are not covered*
- *They can't always integrate well*
  - *- Based on different technologies*
  - *- Overlapping and conflicting*
- ***We need a framework that would homogeneously address the problematic of robotic system***

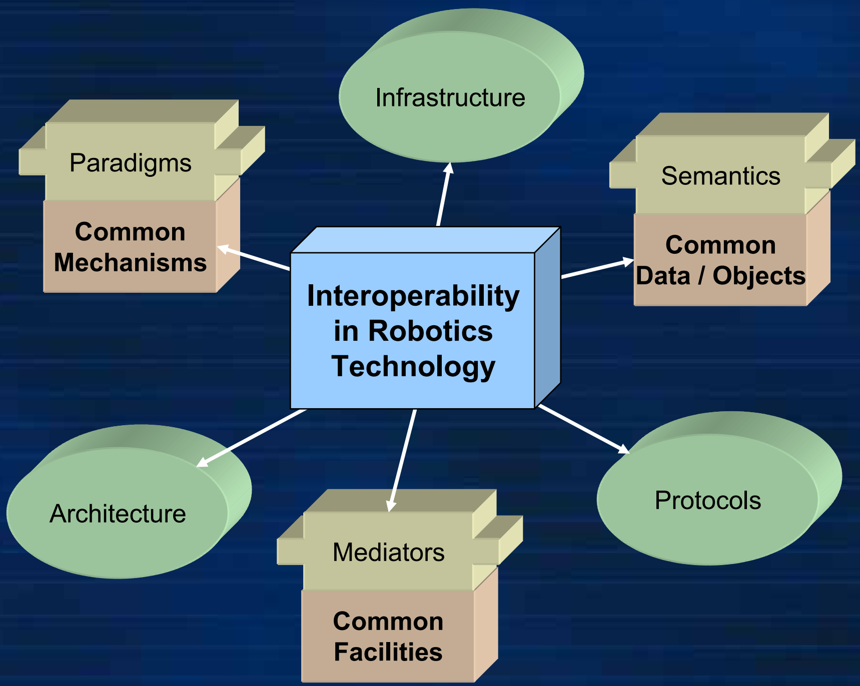


Some frameworks do exist ...

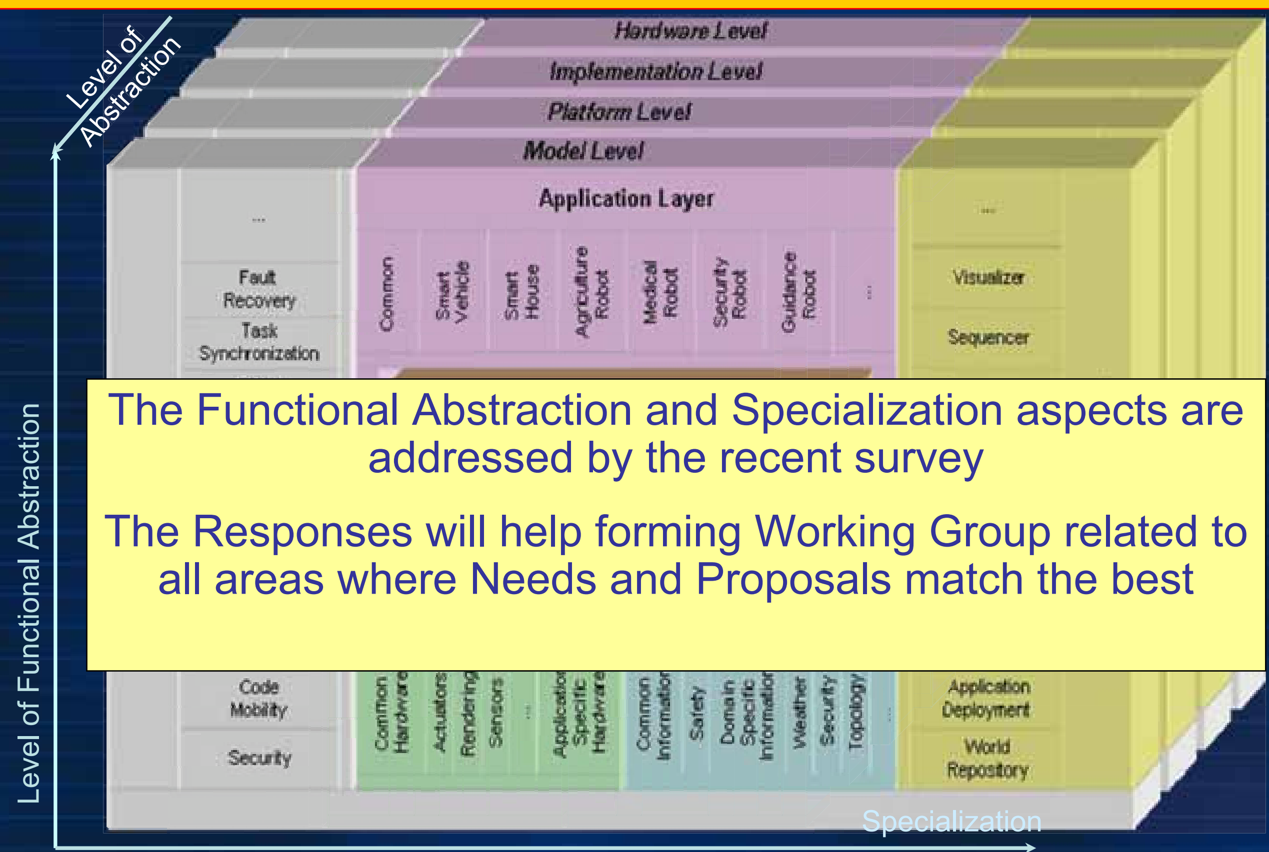
- Many research groups around the world have been and are still trying to address
- Most of them take a similar approach which indicates that **a consensus could be reached**, but...
- Most of them are research oriented
  - Technically correct, unusable as is
  - None of them as yet been backed up by the industry
- Some companies are trying to develop their own solution which usually
  - Cover only their needs
  - Is proprietary
- None of these solutions is even close to reach the volume to make it a de-facto standard



Requirements for Interoperability



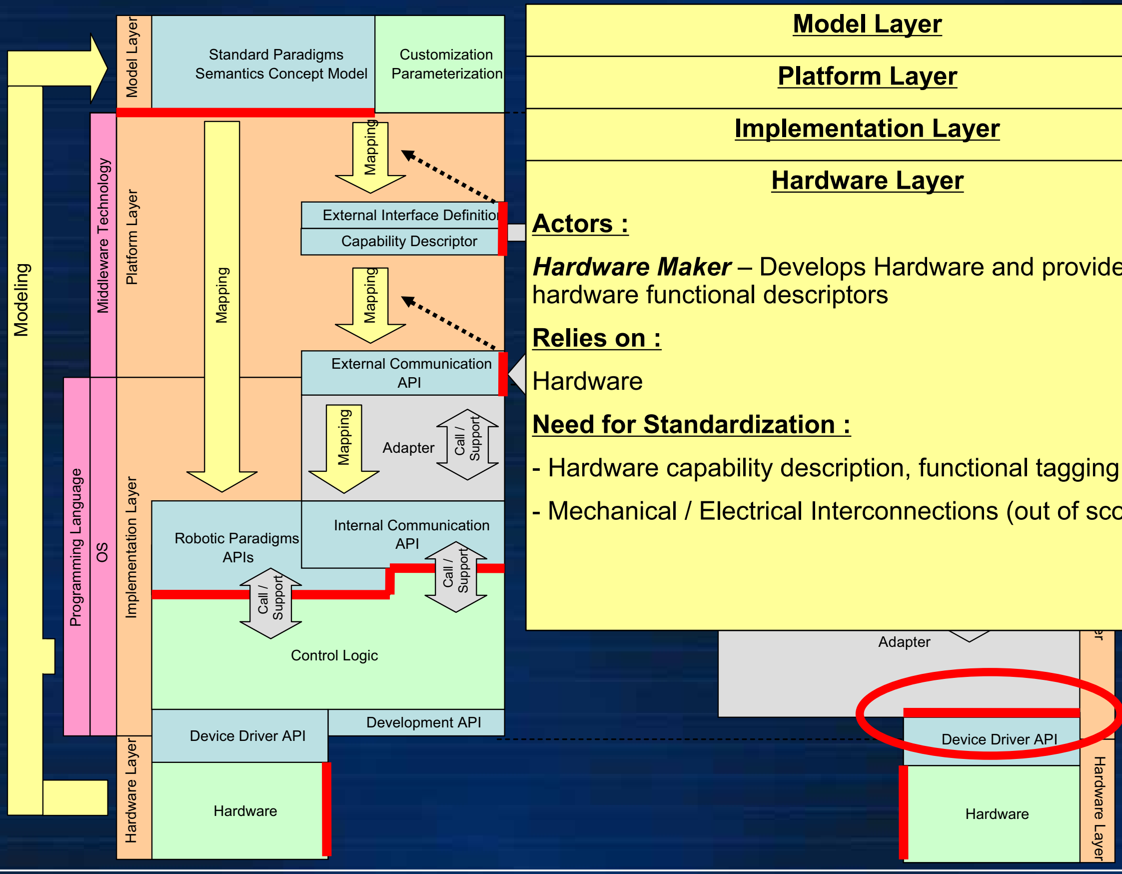
Application to Robotics : a 3 Dimension Space



The Functional Abstraction and Specialization aspects are addressed by the recent survey

The Responses will help forming Working Group related to all areas where Needs and Proposals match the best

Standards at all Levels of Abstraction



Model Layer

Platform Layer

Implementation Layer

Hardware Layer

Actors :

Hardware Maker – Develops Hardware and provides hardware functional descriptors

Relies on :

Hardware

Need for Standardization :

- Hardware capability description, functional tagging
- Mechanical / Electrical Interconnections (out of scope)

The RT Middleware

Robot Technology Middleware (RTM) : what is it anyway ?!?!

- Joint national project between :
  - AIST
  - JARA
  - MEW (Panasonic)
- Sponsored by :
  - NEDO

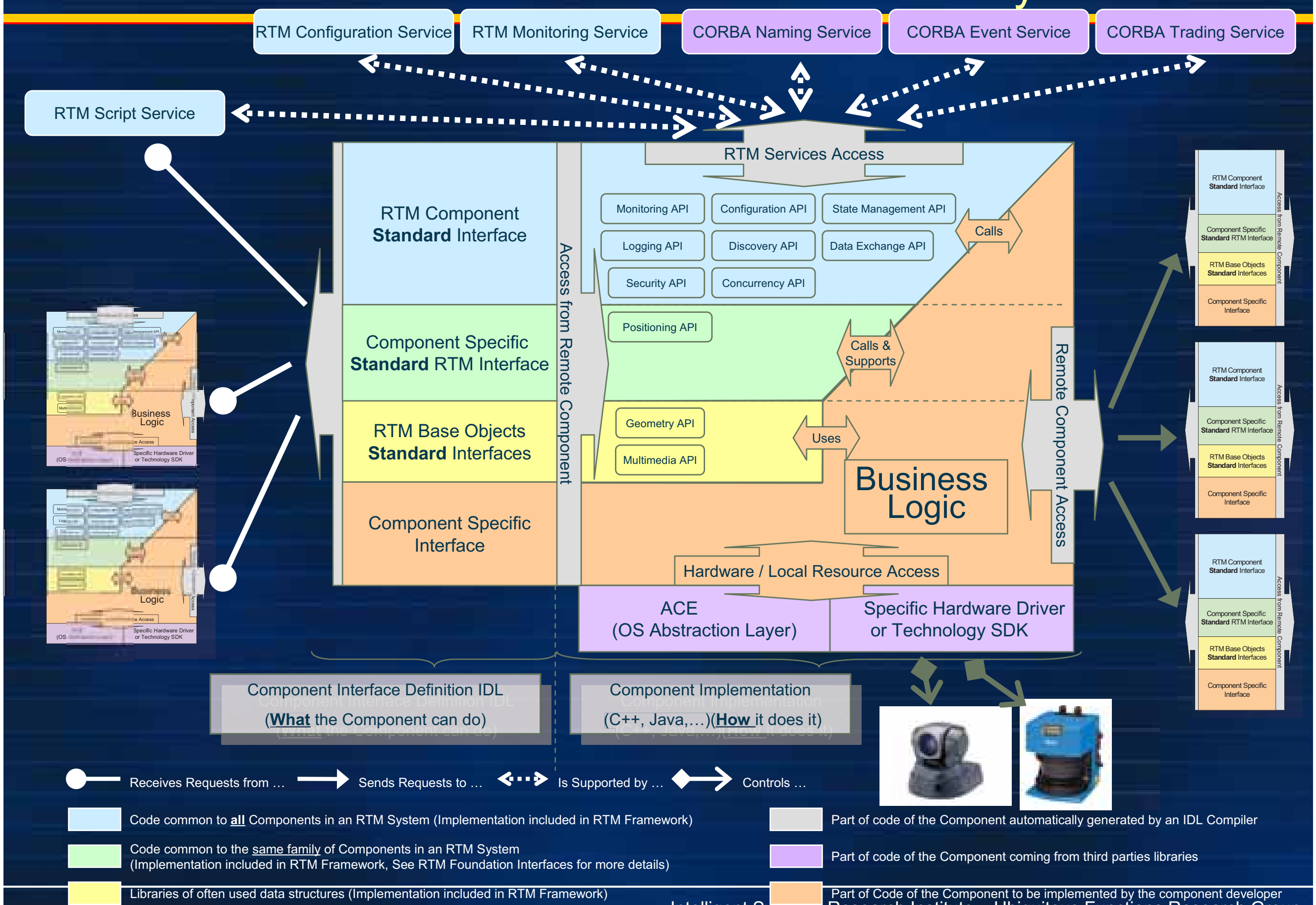
Objective

Increase the competitivity of the Robotic Industry by facilitating the development of large scale multi-vendor Robotic systems through software modularization and standardization

2 Organizations, 2 Approaches… 2 Middlewares ?

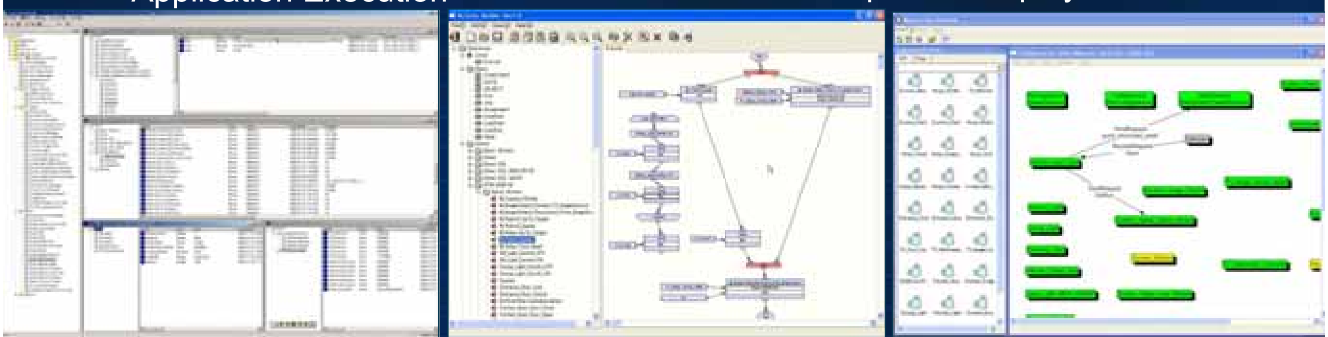
- AIST considered mostly the Component Developer case and emphasized :
  - Component Model and Data Exchange
  - Component Internal Execution
  - Integration by weakly typed components through data exchange
  - Control Systems, High Performance, Real-Time
  - Debugging/Visualization tools, Static System Composition Tools
- MEW considered mostly the System Integrator case and emphasized :
  - Component Model and Data Exchange
  - Component Integration
  - Integration by strongly typed components through well defined interfaces
  - Event Driven, Sequential Behaviors, Flexibility, Scalability
  - Dynamic Application Composition Tools, Configuration Tools

The RT Middleware Concretely



The RT Middleware : Concretely

- **A set of standard mechanisms**:
  - State Machine
  - Data Exchange Mechanism
  - Uniform Global Positioning
  - Activity Monitoring
- **A set of supporting services**:
  - Centralized Configuration Manager
  - Component Repository
  - Event Dispatcher
  - Event Reactor
  - Application Execution
- **A set of standard interfaces** :
  - Common Sensors (IO, Laser, Camera, Mobile Base...)
  - Common Actuators (Motor, Arm, Home Appliances...)
  - Common Rendering Devices
- **A set of tools for** :
  - Component Configuration Management
  - Real-time System Monitoring
  - Graphical Application Composition
  - Component Deployment



AIST

In the End…

- After sharing both our experiences (around a beer…)
  – We agreed on a common component model
  – We agreed on a common data exchange mechanism
  – We agreed on a common set of component states
- … which allowed us to let both approaches co-exist in the same system.

Conclusion : Let's all share our experiences around a beer

AIST

The RT Middleware in Action

Finally :

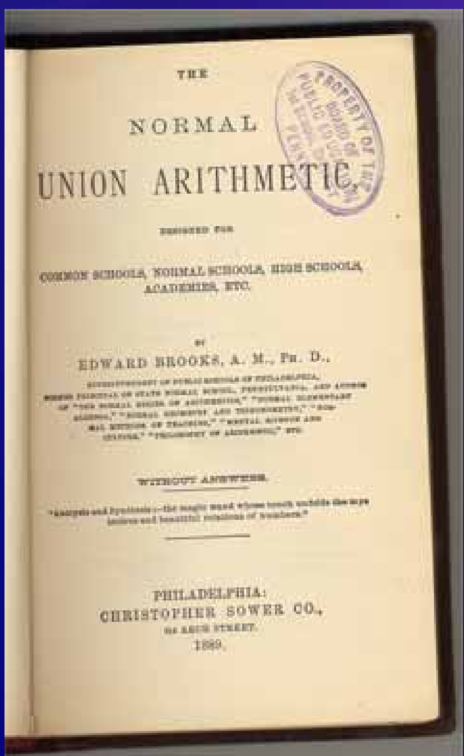
A Nice Video…

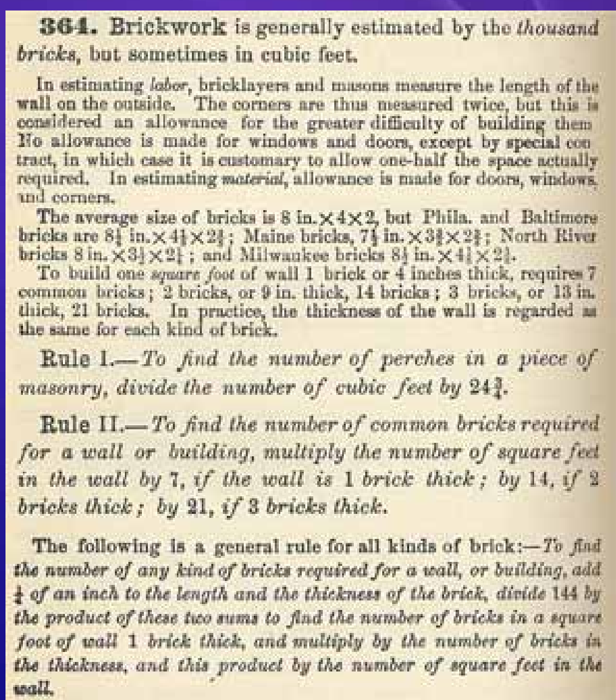
Current State of Robotics Script and Control Languages [and Standards]

Lloyd Spencer
President and CEO
CoroWare, Inc.

February 15, 2006

Robotics Components and Bricks





364. Brickwork is generally estimated by the *thousand bricks*, but sometimes in cubic feet.

In estimating *labor*, bricklayers and masons measure the length of the wall on the outside. The corners are thus measured twice, but this is considered an allowance for the greater difficulty of building them. No allowance is made for windows and doors, except by special contract, in which case it is customary to allow one-half the space actually required. In estimating *material*, allowance is made for doors, windows, and corners.

The average size of bricks is 8 in.$\times 4 \times 2$, but Phila. and Baltimore bricks are $8\frac{1}{4}$ in.$\times 4\frac{1}{8} \times 2\frac{3}{8}$; Maine bricks, $7\frac{1}{2}$ in.$\times 3\frac{3}{8} \times 2\frac{3}{8}$; North River bricks 8 in.$\times 3\frac{1}{2} \times 2\frac{1}{4}$; and Milwaukee bricks $8\frac{1}{2}$ in.$\times 4\frac{1}{8} \times 2\frac{3}{8}$.

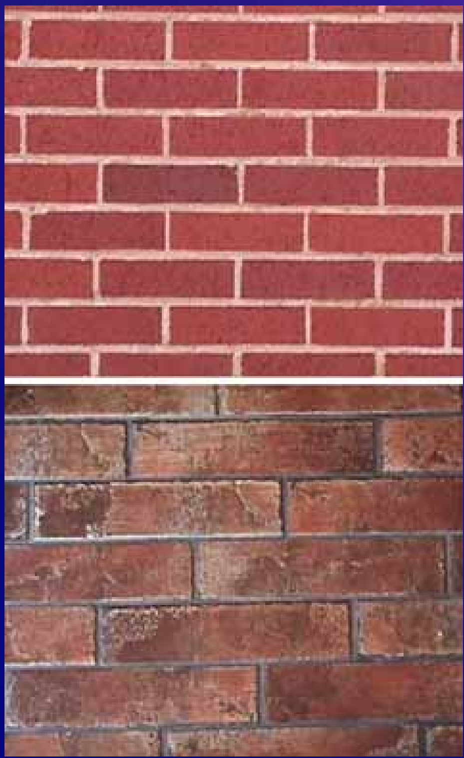
To build one *square foot* of wall 1 brick or 4 inches thick, requires 7 common bricks; 2 bricks, or 9 in. thick, 14 bricks; 3 bricks, or 13 in. thick, 21 bricks. In practice, the thickness of the wall is regarded as the same for each kind of brick.

Rule I.—*To find the number of perches in a piece of masonry, divide the number of cubic feet by* $24\frac{3}{4}$.

Rule II.—*To find the number of common bricks required for a wall or building, multiply the number of square feet in the wall by 7, if the wall is 1 brick thick; by 14, if 2 bricks thick; by 21, if 3 bricks thick.*

The following is a general rule for all kinds of brick:—*To find the number of any kind of bricks required for a wall, or building, add* $\frac{1}{4}$ *of an inch to the length and the thickness of the brick, divide 144 by the product of these two sums to find the number of bricks in a square foot of wall 1 brick thick, and multiply by the number of bricks in the thickness, and this product by the number of square feet in the wall.*

Robotics Components and Bricks and Standards



364. Brickwork is generally estimated by the *thousand bricks*, but sometimes in cubic feet.

In estimating *labor*, bricklayers and masons measure the length of the wall on the outside. The corners are thus measured twice, but this is considered an allowance for the greater difficulty of building them. No allowance is made for windows and doors, except by special contract, in which case it is customary to allow one-half the space actually required. In estimating *material*, allowance is made for doors, windows, and corners.

The average size of bricks is 8 in.$\times 4 \times 2$, but Phila. and Baltimore bricks are $8\frac{1}{4}$ in.$\times 4\frac{1}{8} \times 2\frac{3}{8}$; Maine bricks, $7\frac{1}{2}$ in.$\times 3\frac{3}{8} \times 2\frac{3}{8}$; North River bricks 8 in.$\times 3\frac{1}{2} \times 2\frac{1}{4}$; and Milwaukee bricks $8\frac{1}{2}$ in.$\times 4\frac{1}{4} \times 2\frac{3}{8}$.

To build one *square foot* of wall 1 brick or 4 inches thick, requires 7 common bricks; 2 bricks, or 9 in. thick, 14 bricks; 3 bricks, or 13 in. thick, 21 bricks. In practice, the thickness of the wall is regarded as the same for each kind of brick.

Rule I.—*To find the number of perches in a piece of masonry, divide the number of cubic feet by* $24\frac{3}{4}$.

Rule II.—*To find the number of common bricks required for a wall or building, multiply the number of square feet in the wall by 7, if the wall is 1 brick thick; by 14, if 2 bricks thick; by 21, if 3 bricks thick.*

The following is a general rule for all kinds of brick:—*To find the number of any kind of bricks required for a wall, or building, add* $\frac{1}{2}$ *of an inch to the length and the thickness of the brick, divide 144 by the product of these two sums to find the number of bricks in a square foot of wall 1 brick thick, and multiply by the number of bricks in the thickness, and this product by the number of square feet in the wall.*

CoroWare

Robotics Interfaces

Applications
- Non-Destructive Inspection
- Security and Monitoring
- Unmanned Vehicles

Robotic Functions
- Vision
- Navigation
- Manipulation

Operating System Platform
- Plug and Play
- Wireless Communications
- Device Drivers

Hardware Platform
- Sensors
- Motors
- Mainboards

Scripting and Control Interfaces

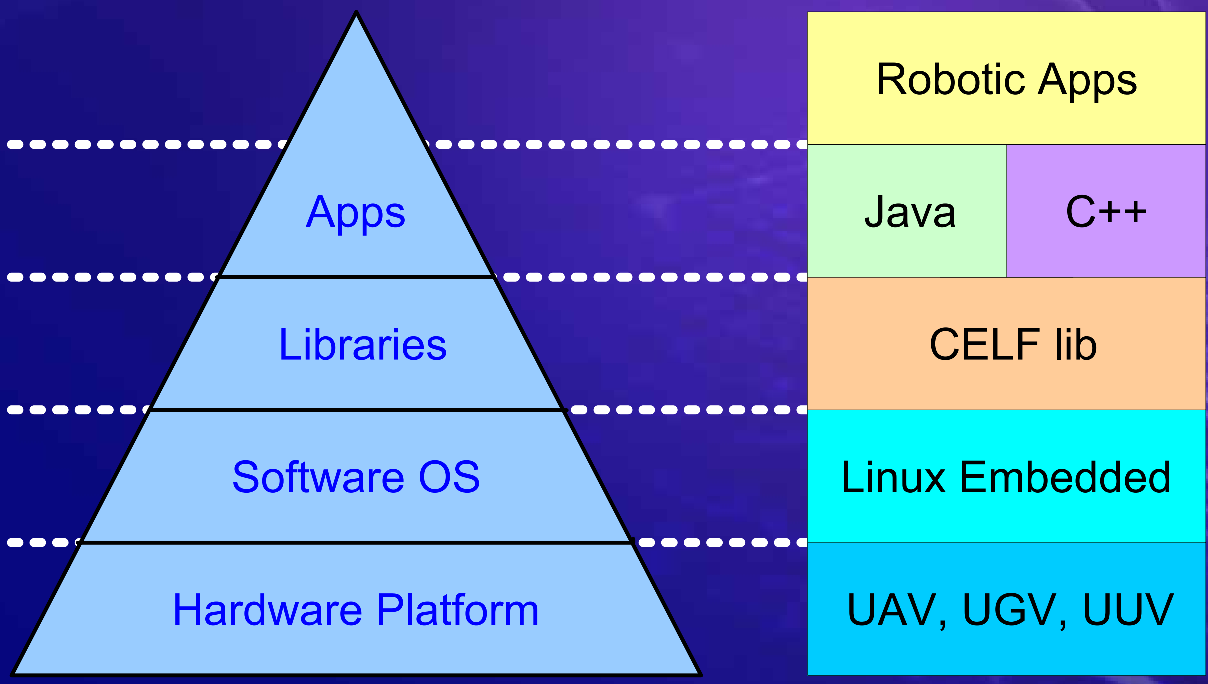
Protocol Interfaces

Programmatic Interfaces

CoroWare

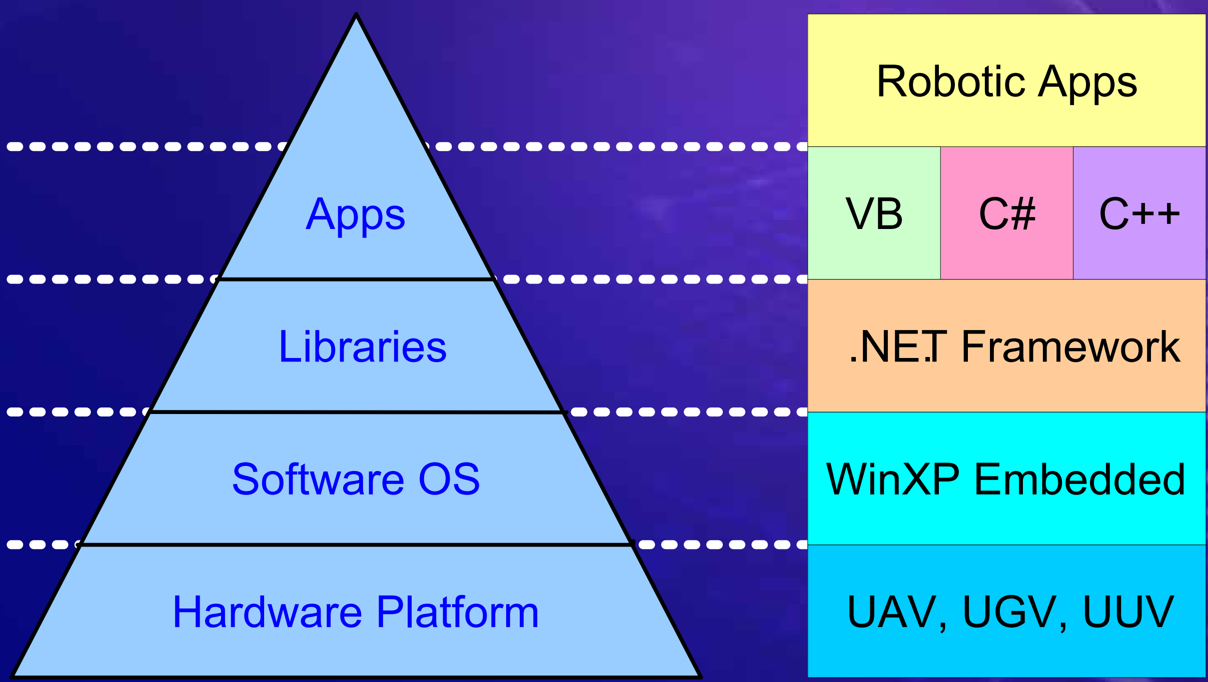
Robotics Interfaces

Programmatic Interfaces (Linux CELF)



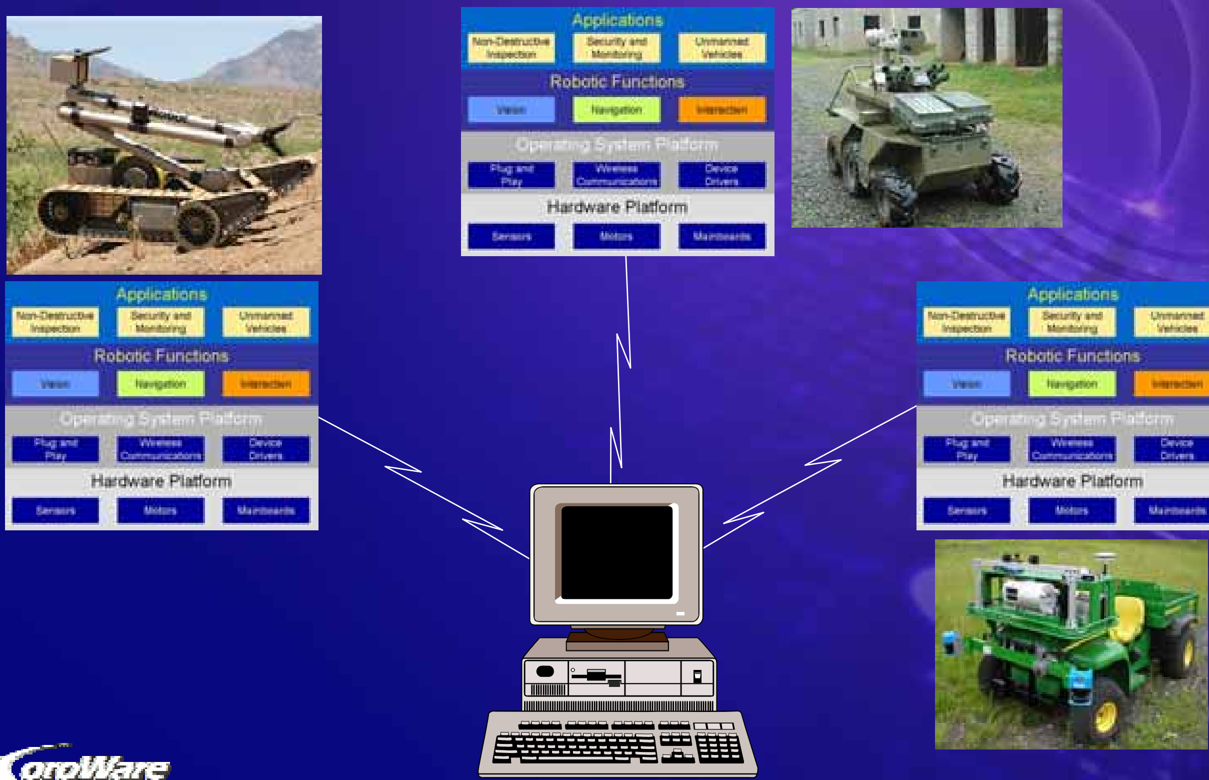
Robotics Interfaces

Programmatic Interfaces (Microsoft .NET)



Robotics Interfaces

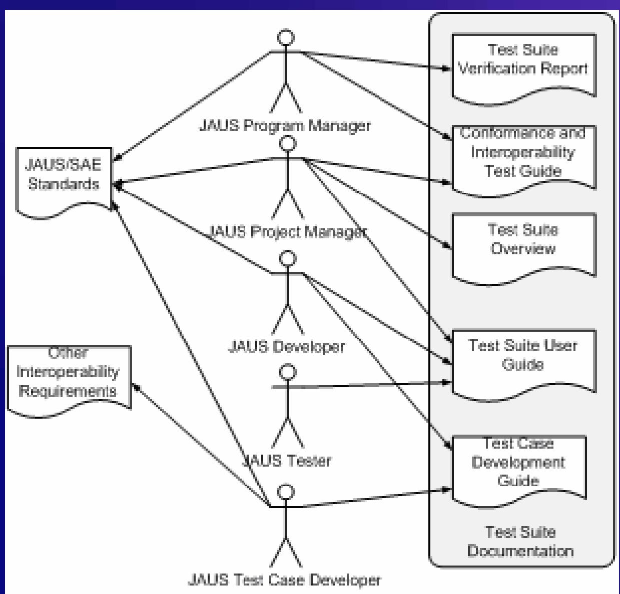
Protocol Interfaces (JAUS)



CoroWare

Robotics Interfaces

Protocol Interfaces (JAUS)



SAE International

JAUS

CoroWare Test Labs, Inc.
Interoperability Testing for Robotics

CoroWare

Robotics Interfaces
Scripting and Control Interfaces

- Industrial Automation
  - Principally for controlling arms, conveyers and gantry devices
  - Proprietary
    - Allen Bradley
    - ABB
    - Fanuc
    - Motoman
    - MicroRobotics
  - Multi-Vendor
    - G-Code
    - Robotic Workspace Technologies

Robotics Interfaces
Scripting and Control Interfaces

Proprietary Scripting

MOVL P000 V=138 SRCH
RIN#(1)=ON T=1.00 DIS=10.00

MOVS V=120 PL=0

SPEED VJ=50.00

WAIT IN# (12)=ON T=10.00

Robotics Interfaces
Scripting and Control Interfaces

Multi-Vendor RobotScript

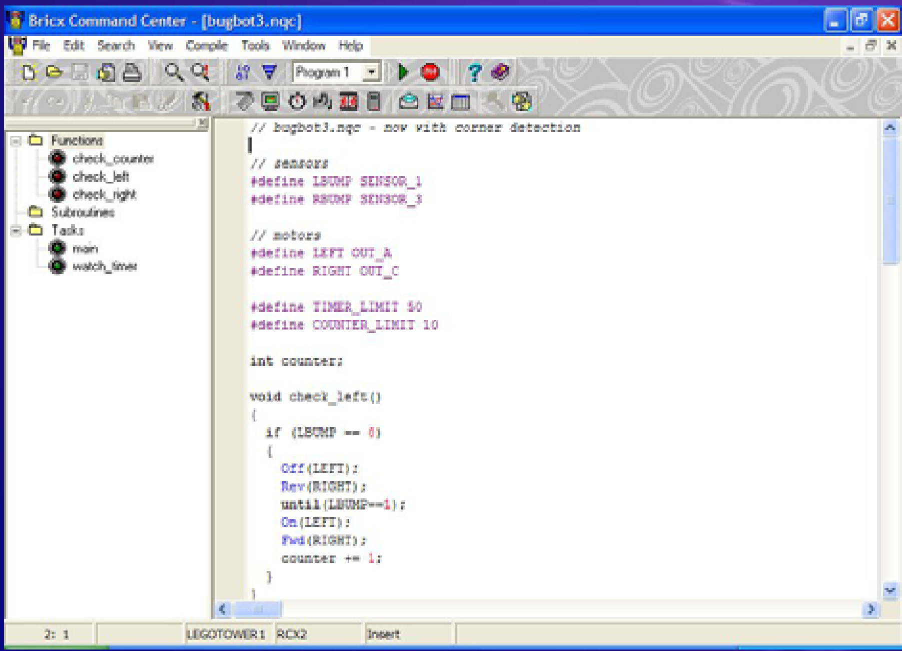
```
Include ProgramPath & "TitanDemo.xyz"

UnitType   = Metric
Speed      = 200 ' mm/s
JointSpeed = 30 '% of max joint velocity

'Move to press:
SendOutputMsg "Moving to press..."
MoveJointTo "StartPos"
MoveJointTo "PressPrePick"
MoveLinearTo "PressPick"
WaitUntilStop
```

Robotics Interfaces
Scripting and Control Interfaces

- Mobile Service Robotics
  - Education and research
  - Proprietary
    - Lego
  - Open and Multi-Vendor
    - Not-Quite-C (NQC)
      - Based on a C-like syntax (David Baum)
    - Java and Visual Basic
    - PolyKinetic System (Xerox PARC)
      - Based on XML Web Services

Robotics Interfaces

Scripting and Control Interfaces



Not Quite C (NQC)

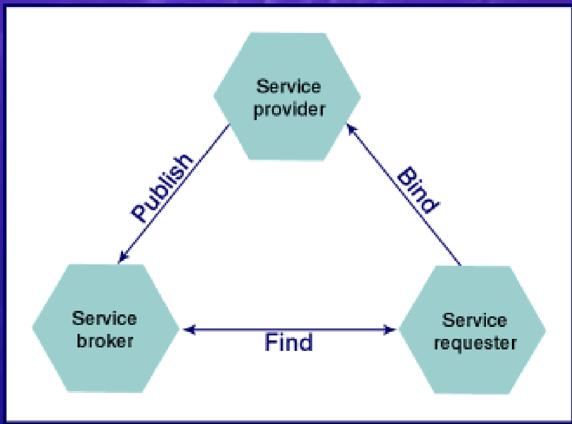
Robotics Interfaces

Scripting and Control Interfaces

- Mobile Service Robotics
  - Education and research
  - Proprietary
    - Lego
  - Open and Multi-Vendor
    - Not-Quite-C (NQC)
      - Based on a C-like syntax (David Baum)
    - Java and Visual Basic
    - PolyKinetic System (Xerox PARC)
      - Based on XML Web Services

Robotics Interfaces
XML Web Service Interfaces

- XML Web Service
  - Modular application that is published, located, and invoked [over a network]
- Robotic Web Service
  - Service Provider
    Robotic platform invoked via web service agent
  - Service Requester
    Operations Control Unit that locates and remotely invokes robotic platform
- Robotic Markup Language



Robotics Interfaces
XML Web Service Interfaces

- Extended Markup Language
  - Optimized for Software Developers and Testers
- Simple Object Access Protocol
  - Optimized for System and Communications Performance
- Underlying Transport Protocol Options
  - TCP/IP Sockets
  - HTTP over TCP/IP

```
<script doc="Move to conveyer...">
   <command cmd="SendOutputMsg">
    <param name="str1">"Moving to conveyor..."</param>
   </command>
   <command cmd="MoveLinearTo">
    <param name="str1">"BancroftPostPlace"</param>
   </command>
   <command doc="ConvPrePlace" cmd="MoveJointTo">
    <param name="loc">"ConvPrePlace"</param>
   </command>
   <command doc="ConvPlace" cmd="MoveLinearTo">
    <param name="loc">"ConvPlace"</param>
   </command>
   <command cmd="WaitUntilStop"/>
   <script doc="Time delay of 2 seconds:">
    <command cmd="Delay>
      <param name="seconds">2</param>
    </command>
   </script>
```

In Summary

- Robotics Interfaces can be categorized as
  - Programmatic Interfaces
  - Protocol Interfaces
  - Scripting and Control Interfaces
  - All of the above using XML Web Services
- XML Web Services should be considered as a viable model for:
  - Local resource interfaces
    - arms, motors, sensors
  - Local robotic function interfaces
    - localization, mapping, vision recognition
  - Remote robotic function interfaces
    - Higher level navigation functions and tactical objectives
- OMG should create an RFI for Robotics Markup Language
  - Simple Object Access Protocol (SOAP)

Lessons Learned About Software for Rescue Robots

Matt Long
Prof. Robin Murphy
University of South Florida
{mtlong,murphy}@cse.usf.edu

Software Needs for Rescue Robots

- Require an architecture to enable robot operation (autonomous, supervisory, and direct), medical reachback, training, multi-robot operations
- Must have the following:
  - reactive and deliberative robotics; open standards / open source; fault tolerant; adaptability; consistent programming model; dynamic design
- Our experience favors:
  - Modular, service-oriented architecture
  - Built on prior experience with SFX single-agent architecture
  - Layered on standard middleware

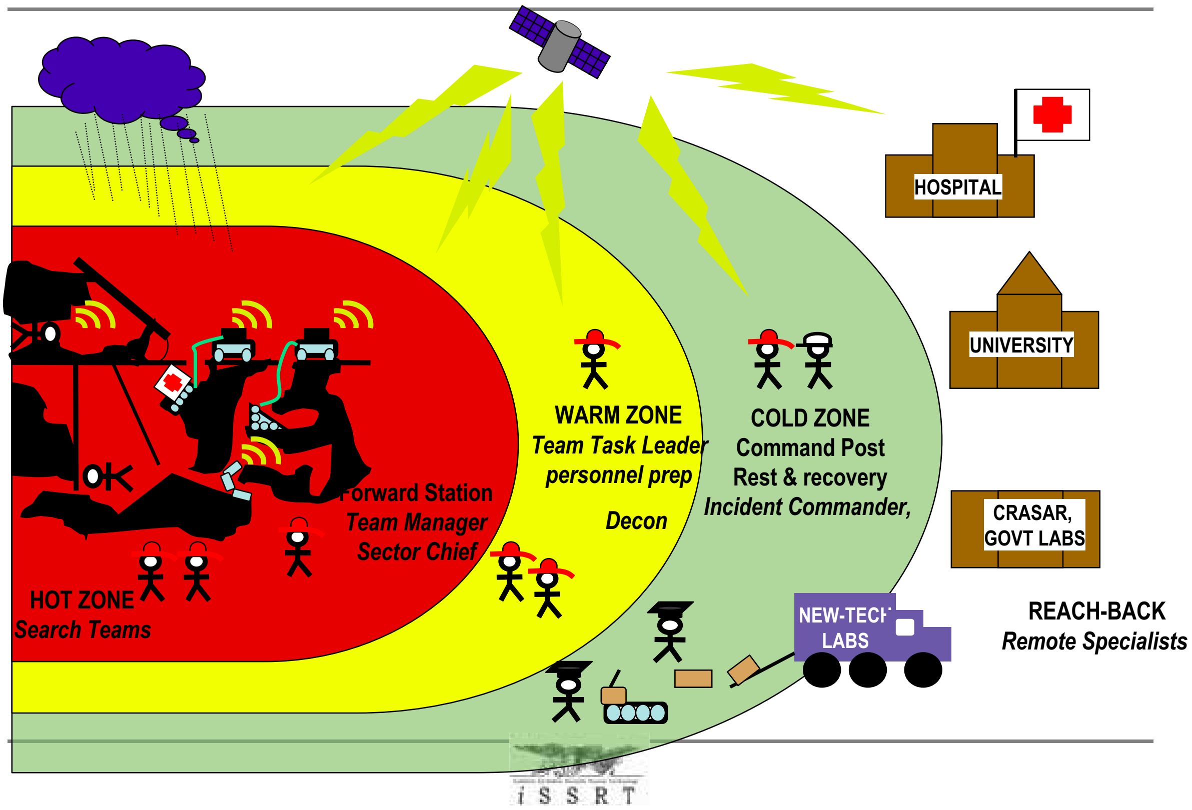
Outline

- USF Urban Search and Rescue Experience
- The Problems Posed by Rescue Robots (land, sea, and air)
- Current Thinking on Software Organization
  - General Design Goals and Issues

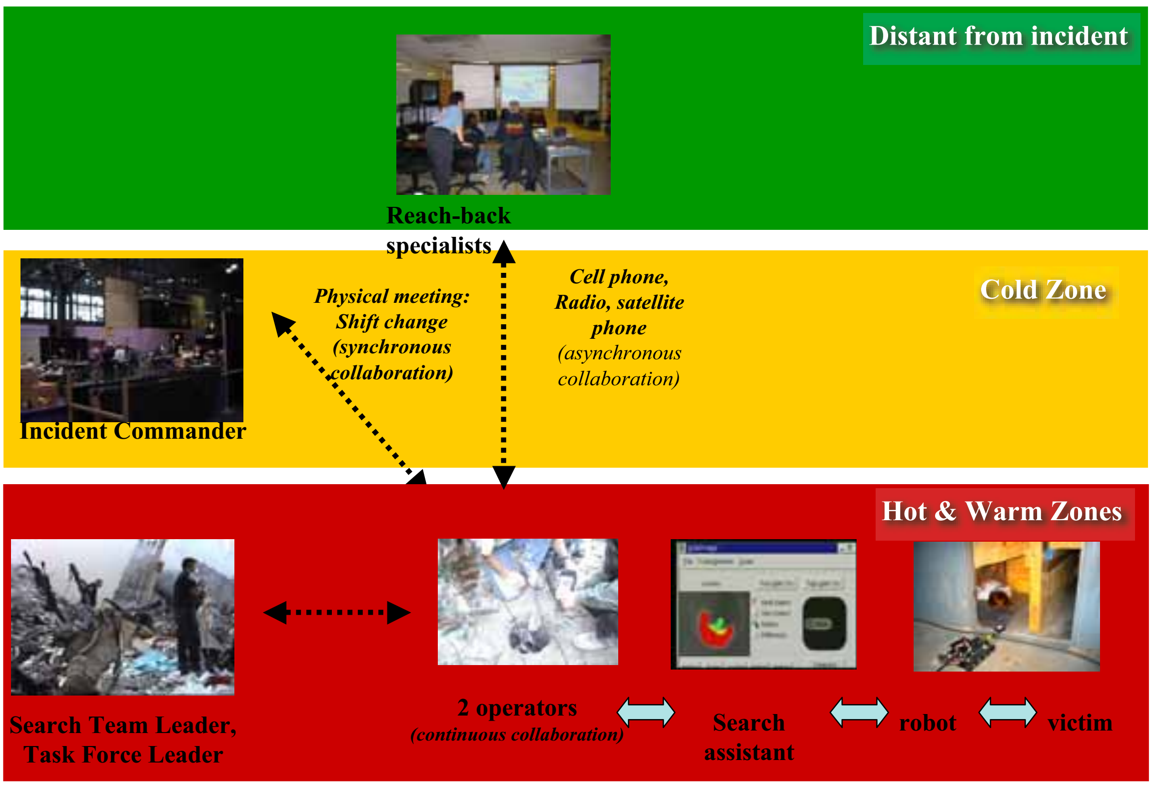
Field Data on Emergency Response
6 responses: UGV, UAV, USV

- 2001
  - 2 evolutions: FLTF-3 (Mar, Aug)
  - **WTC response**
  - 1 evolution: RTA@Miami (Nov)
- 2002
  - 2 evolutions: RTA@Connecticut (Oct), Rescue South Africa (Dec)
  - Medical field evaluation: USMC CBIRF (Aug)
- 2003
  - 1 joint exercise: ShadowBowl (Jan)
  - 3 evolutions: RTA@OKC (Apr), INTF-1 (Aug), International Rescue Systems Institute Japan (Sept)
  - Sensor field evaluation: RTA@OKC (Apr)
  - Mine rescue field evaluation: SIMTARS Australia (Jun-Dec)
- 2004
  - 3 evolutions: CATF-2 (Apr), NASA Ames Technologists Meet the Responders (May), LSU FETI (July)
  - **Hurricane Charley FLTF3 (Aug), standby for Frances, Ivan, Jeanne**
- 2005
  - **La Conchita Mudslides LaCOFD/Ventura Co (Jan)**
  - 3 evolution: NJTF-1 (Feb), Tampa Fire Rescue (Jul), Kansas City (Sep)
  - **Hurricane Dennis (Jun)**
  - **Hurricane Katrina (Sep)**
  - **Hurricane Wilma (Oct)**
  - **Hurricane Katrina (Nov-Dec) photo-documentation**

USAR



Distributed Data



Distributed Systems

- "Essentially everyone, when they first build a distributed application, makes the following eight assumptions. All prove to be false in the long run and all cause big trouble and painful learning experiences.
  1. The network is reliable
  2. Latency is zero
  3. Bandwidth is infinite
  4. The network is secure
  5. Topology doesn't change
  6. There is one administrator
  7. Transport cost is zero
  8. The network is homogeneous"
    - – Attributed to Peter Deutch

iSSRT

What are the desirable characteristics of a distributed **field** robot architecture?

- Incorporate *reactive and deliberative* components
- *Open* standards and / or open source
- **Fault tolerant** at both the system and component levels
- **Adaptable** in the face of changing operating conditions
- *Long-lived*: Modify, administer, log and maintain the system at runtime
- *Consistent programming model*
- **Dynamic** from the start

iSSRT

Reuse from existing architectures

- Robot architectures already specify interaction between system components on a single system
  - Support for deliberative agents, reactive components
  - E.g. Subsumption (Brooks, 1986), 3T (Bonasso, 1997), AuRA (Arkin, 1998), SFX (Murphy, 2000), 4D/RCS (2002)
- Software agent architectures specify interactions between distributed agents, but don't map directly to a robot architecture
  - Typically good at using / creating standards
  - E.g. KAoS (Bradshaw), RETSINA (Sycara), CoABS, FIPA
- Existing middleware can provide the "glue" to tie the two together
  - Abstraction of a distributed computing environment (Bernstein, 1996)

Provides **reactive, deliberative** robotics, **AI-based fault tolerance**

- Robot architectures already specify interaction between system

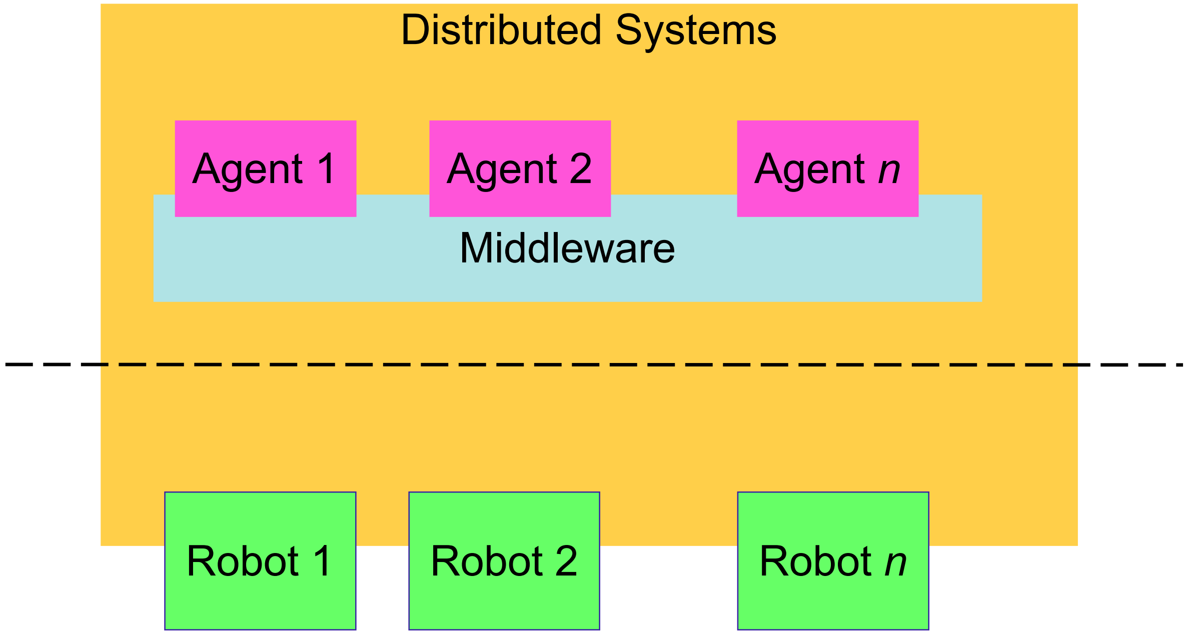
Designed for **adaptability**, **scalability**, **dynamicity**, typically **open standards**

1998), SFX (Murphy, 200 … RCS (2002)
- Software agent architectures specify interactions between

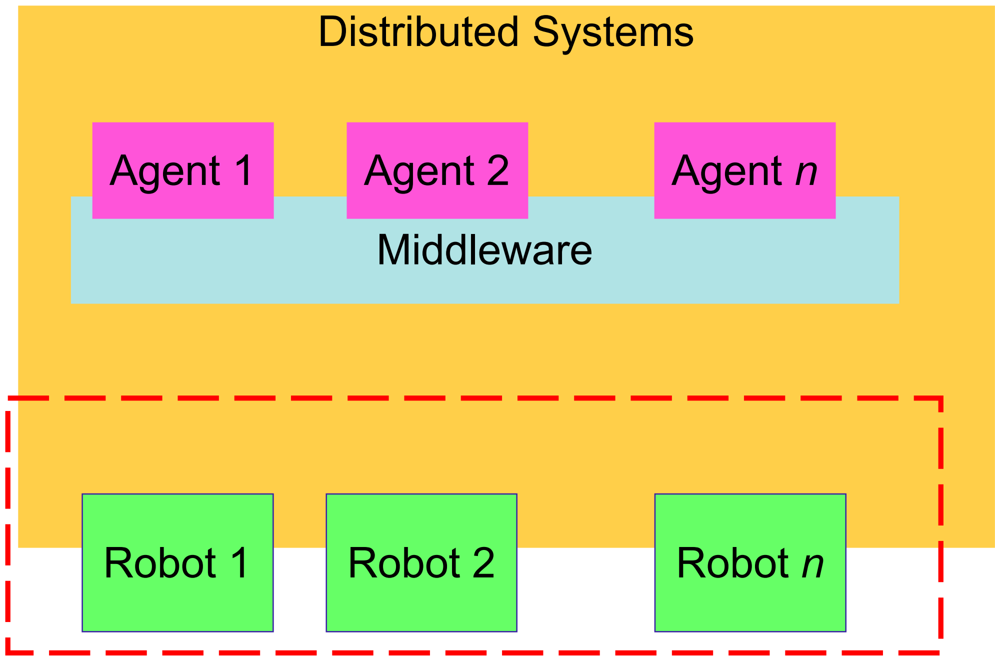
Aids **system-level fault tolerance, consistent programming model**

- E.g. KAoS (Bradshaw), RE … (Sycara), CoABS, FIPA
- Existing middleware can provide the "glue" to tie the two together
  - Abstraction of a distributed computing environment (Bernstein, 1996)

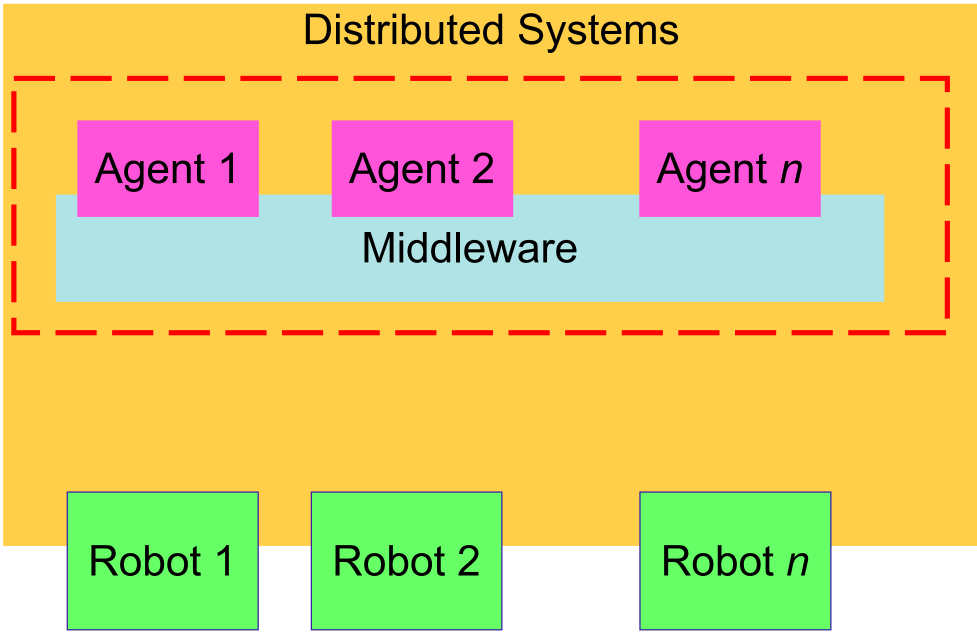
Another view: Multi-Robot + MAS



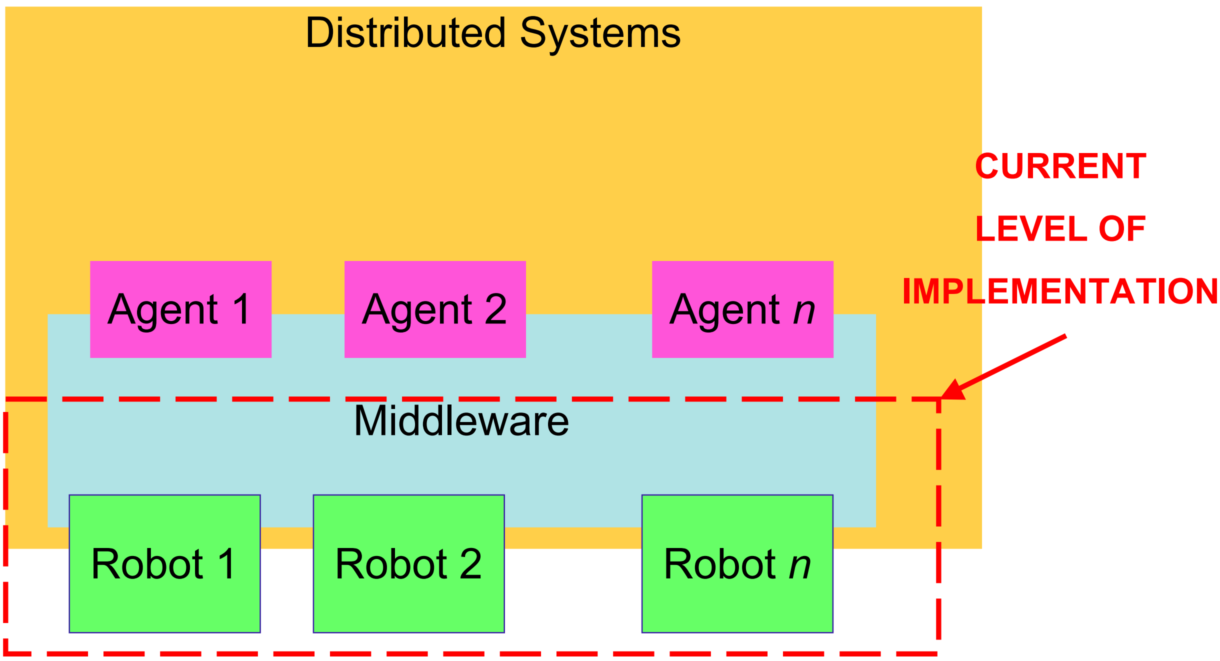
Multi-Robot Work Doesn't Leverage MAS



MAS Not Suitable for Real-Time Control



Goal



DFRA

Why Java and Jini?

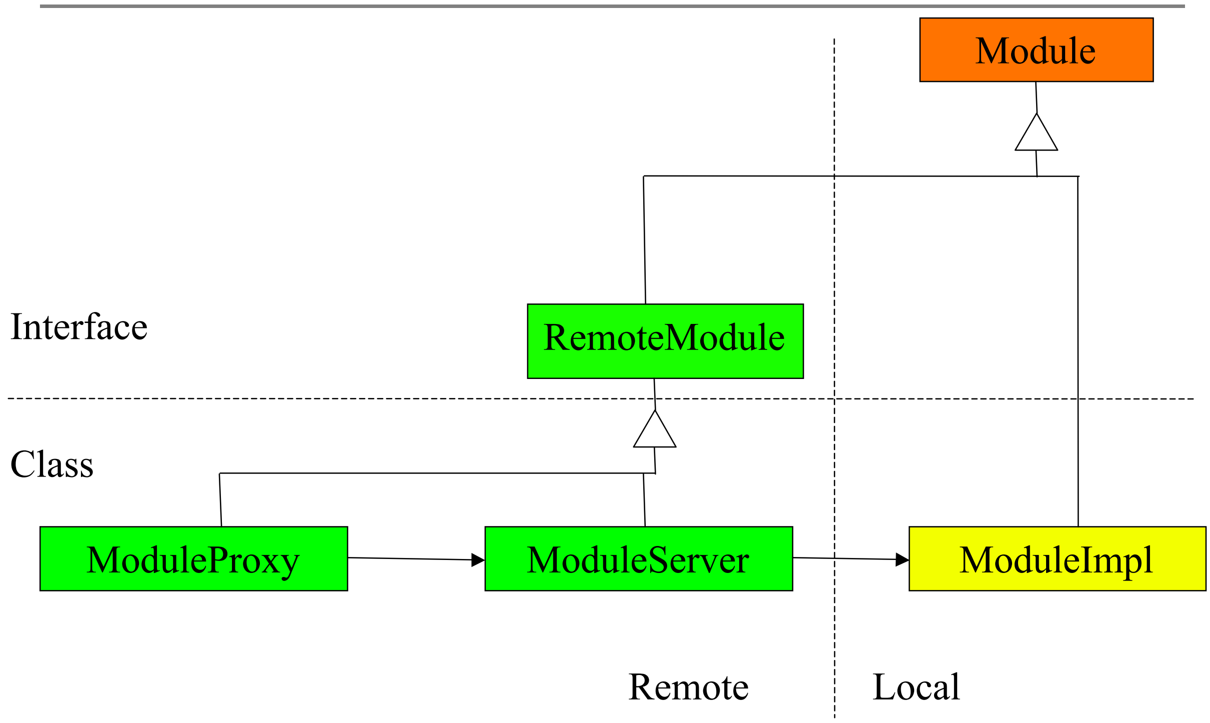
- Java + Jini provide a consistent programming model
- Jini enables a service-oriented architecture
  - Robot capabilities can be utilized across the network
  - Included utilities: activation, naming, lookup
- Multi-platform code mobility
  - Code downloaded from codebase as needed
  - Particularly important for flexible design, execution
- Complexity of Java is in the libraries, not the language
  - Java core, Jini, Java3D, JDOM, etc.
- Familiarity
  - Prior experience with Java fundamentals and Jini technology
  - Programmer productivity
- Freely available, active developer community

Design Issues

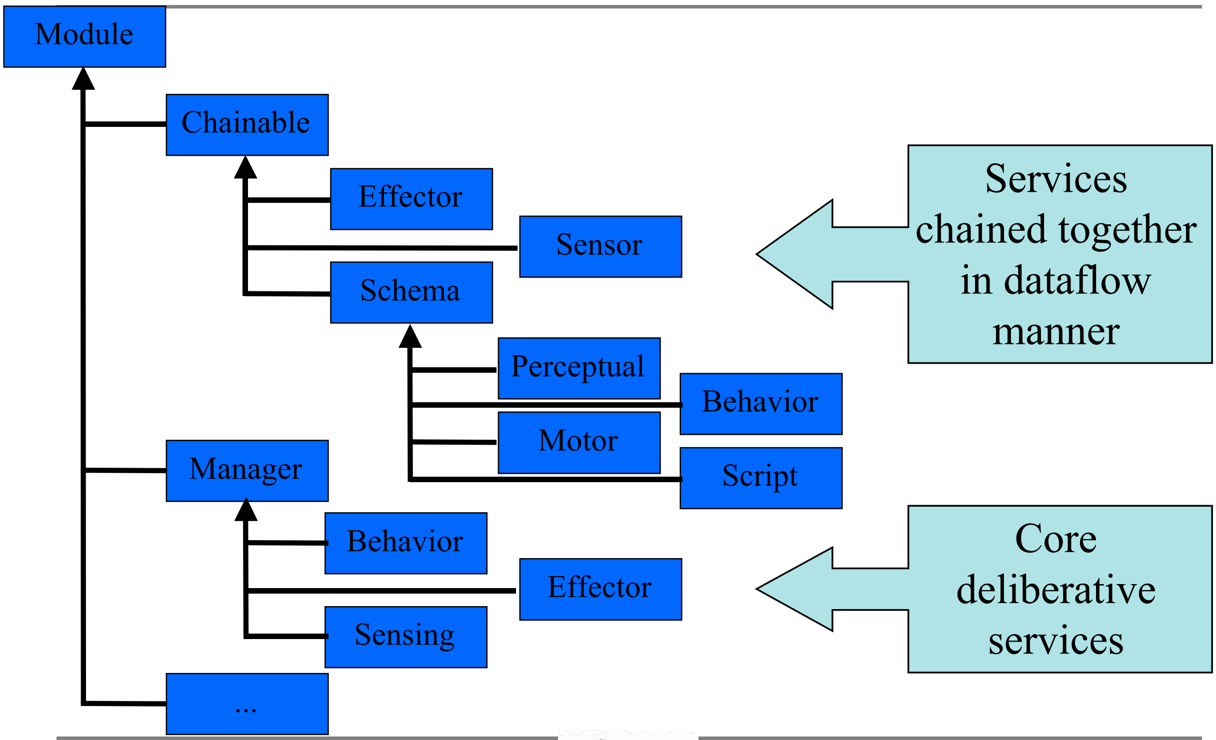
- Extensive use of design patterns
  - Factory Methods to defer class creation to runtime objects
  - Observer for distributed event handling
  - Proxy / Smart Proxy to control access to distributed objects and flexible control of execution context

- 2-way "orthogonal" class design
  - "Horizontal" classes related to the distributed system glue (Jini)
  - "Vertical" classes related to underlying robot architecture (SFX)

"Horizontal" Class Hierarchy



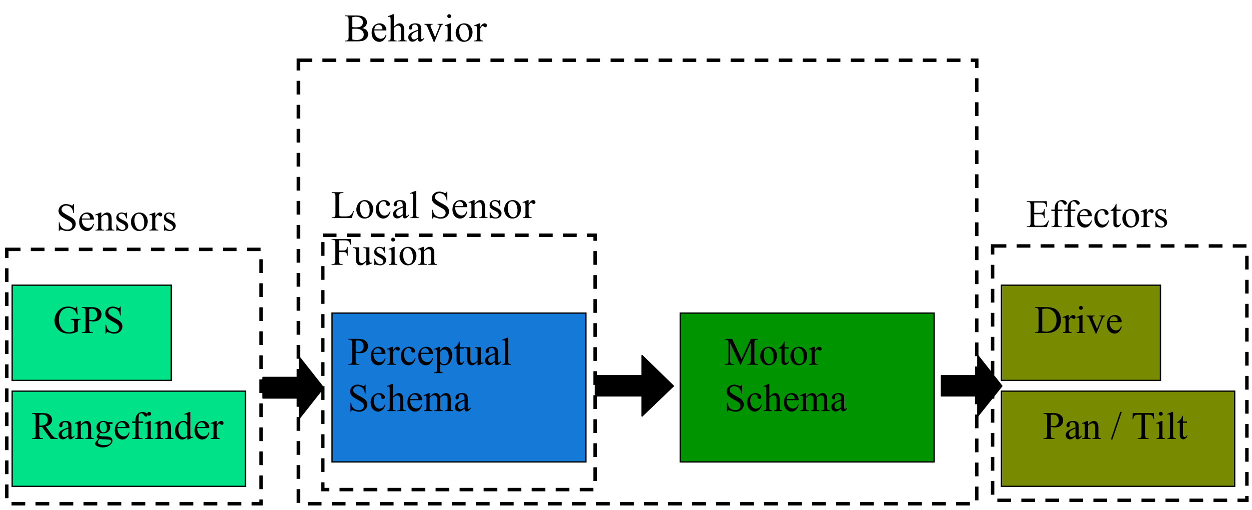
"Vertical" Class Hierarchy



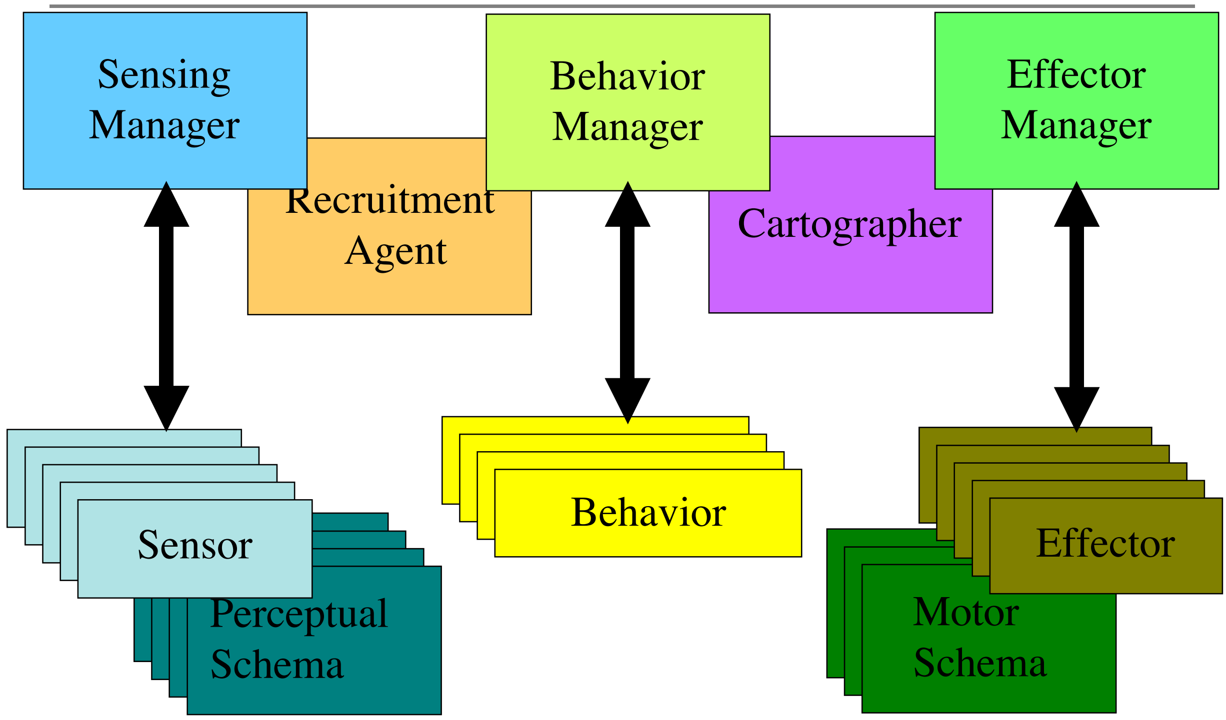
Architectural Model

- DFRA built on SFX components
  - Sensors, effectors, perceptual and motor schemas, behaviors
  - Sensing-, effector-, and task managers
  - Supports cartographer, recruitment and other deliberative services
- DFRA object model provides a number of key datatypes
  - Extensive sensor and percept **data formats** (images, GPS, laser readings, etc.)
  - Many services can have a **pose**, relating the position of the component to others
  - Common set of **events** (sensor, behavior, robot state, etc.)
  - Each service has an **entry** into the system-wide directory services
  - Entry **filters** are used to search for arbitrary services
  - **Logging** built into each component

A Reactive Behavior



Deliberative Services



Design Lessons

- Benefit: Allowing for flexibility is key
  - Can vary code execution with smart proxies
  - Low-power/computation devices can be proxied into larger network
  - Implementation can use most languages (e.g. MATLAB through JNI)
  - Each service can operate at own speed (reactive -> fast, deliberative -> slower)
  - Configuration through independent Jini and service config files
- Benefit: Robust middleware provides critical capabilities
  - Service activation and management, naming, lookup, discovery
  - Distributed event system, transactions for concurrency support
  - Leasing to handle partial failure
  - Transport agnostic (RMI, JERI, IIOP, custom)

Design Lessons

- Tradeoff: Complexity
  - Configuration (both Jini and DFRA)
  - Future: Can be simplified / eliminated with automated tools or GUIs
- Tradeoff: Performance Overhead
  - Desktop Java is not real-time, but is good enough
  - ~5-15ms to read sensors, ~50-100ms for MATLAB control processing, 10ms for motors
  - Small-systems efficiency really pays off in inner loop control
- Tradeoff: Steeper learning curve
  - Familiarity with Java libraries required
  - Large set of additional libraries for Jini, etc.
  - Fixed with experience and time

Conclusions

- Needed an architecture to cover robot operation (autonomous, supervisory, and direct), medical reachback, training, multi-robot operations
- Followed our requirements:
  - reactive and deliberative robotics; open standards / open source; fault tolerant; adaptability; consistent programming model; dynamic design
  - DFRA satisfies these constraints
- Our work with USAR led us to:
  - Modular, service-oriented architecture
  - Built on prior experience with SFX single-agent architecture
  - Layered on standard middleware
- Used for a variety of platforms
  - ATRV-class robots, Inuktun VGTVs, custom-designed VTOLs, and pure-software services

OMG Robotic Systems RFI Response

Development Framework for Mobile Robot based on JAUS and RT-Middleware

Explore the Engineering Edge

Ishikawajima-Harima Heavy Industries Co., Ltd.

http://www.ihi.co.jp/index-e.html

0. Agenda

Explore the Engineering Edge

1. Expectation for Standardization
2. Approach of IHI
3. JAUS
4. OpenRTM-aist
5. Our Mobile Robot Framework
6. Experimental examples
7. Summary

Reduction in development cost, Improvement in reliability

We wish the robotic standard bring us those things.

- Reform the current state of developing almost entire systems at each new development.
- Concentrates on the control algorithm that improves the performance. The common system part should be left to a standard framework as much as possible.
- Develop only specialized parts by myself. Other parts should be left to outsourcers.
- Enable step by step function enhancements.

1.1. Expectation for Standardization

Explore the Engineering Edge

Reduction in development cost, Improvement in reliability

We wish the robotic standard bring us those things.

- Reform the current state of developing almost entire systems at each new development.
- Co ... roves the ... ould be lef ... le.
- De ... er parts sh...

Many useful robots in a short term

- Enable step by step function enhancements.

1.2. Approach of IHI (1)

Two existing open architectures are positively used.

- **JAUS** (Joint Architecture for Unmanned Systems)

  Application specification for unmanned systems.It doesn't depend on the implementation technologies.

- **OpenRTM-aist** (Robot Technology Middleware, implemented by AIST)

  Basic specification/ implementation framework of Robot Technology Component, which doesn't depend on application.

1.3. Approach of IHI (2)

We are not a member of JAUS Working Group.

But, we think that the JAUS specification is very useful for mobile robots and adopt the concept of OpenRTM-aist.

Construct a suitable mobile robot development framework by appropriate combination of JAUS and OpenRTM-aist.

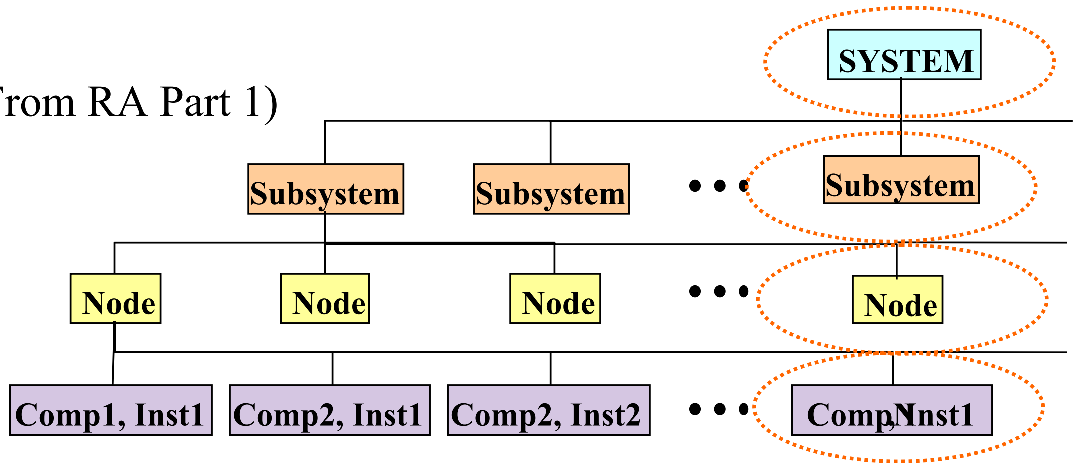
2.1. JAUS – Outline

JAUS (Joint Architecture for Unmanned Systems)

- Standard specification of the U.S. Department of Defense for unmanned systems.
- Specification independent from technology, computer hardware, control means and platform.
- Aiming to decrease the life cycle cost and the development period.
- Behaviors and messages of the software component are defined.
- Website: http://www.jauswg.org/

2.2. JAUS – System Topology

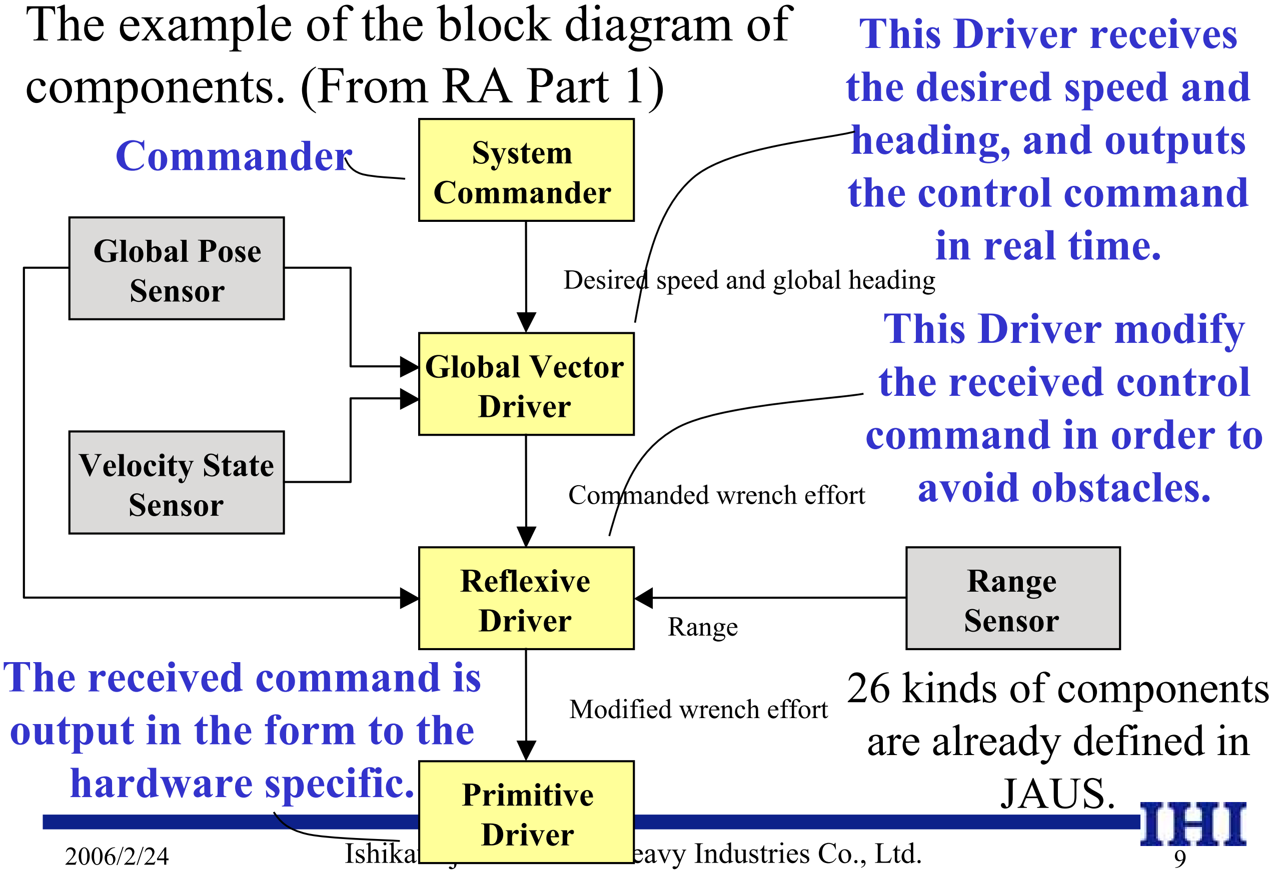
(From RA Part 1)



- System : Expressing the whole
- Subsystem : Single unit/machine (Vehicle, etc.)
- Node : Computer resource
- Component : Functional unit (Software module)
- Instance : Execution form of a component (Process)

2.3. JAUS – Components

The example of the block diagram of components. (From RA Part 1)



26 kinds of components are already defined in JAUS.

2.4. JAUS – Message

The system of JAUS is driven by the message communication between components.

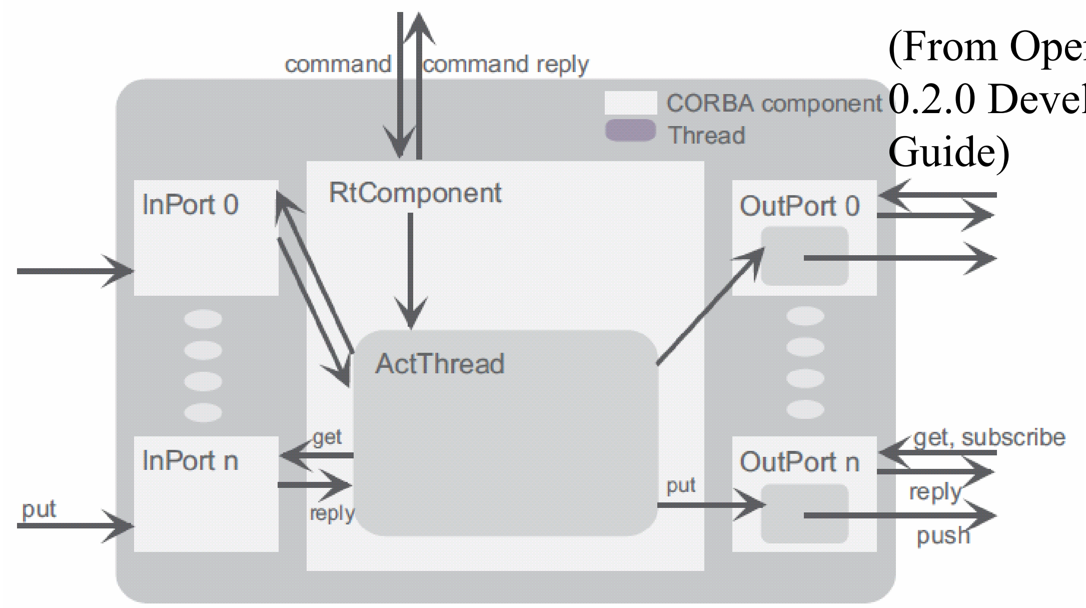
- The messages for unmanned systems are already defined. (currently, 110 kinds)
- The messages are classified into six categories. （Command、Query、Inform、Event Setup、Event Notification、Node Management）
- Message format ＝ header(16bytes) + option data
- The option data is specific to each kind of message.

3.1. OpenRTM-aist – Outline

OpenRTM-aist

- Basic specification/ implementation framework of RT-Component, which doesn't depend on application.
- The communication between RT-Components is realized with the CORBA objects that are called InPort/OutPort.
- The RT-Component is active, and the behavior (state transition) is defined in the standard.
- Website: http://www.is.aist.go.jp/rt/

3.2. OpenRTM-aist – RT-Component



(From OpenRTM-aist-0.2.0 Developers Guide)

- RtComponent : RT-Component main body.
- InPort : Object that receives data.
- OutPort : Object that sends data.

4.1. Framework – Design

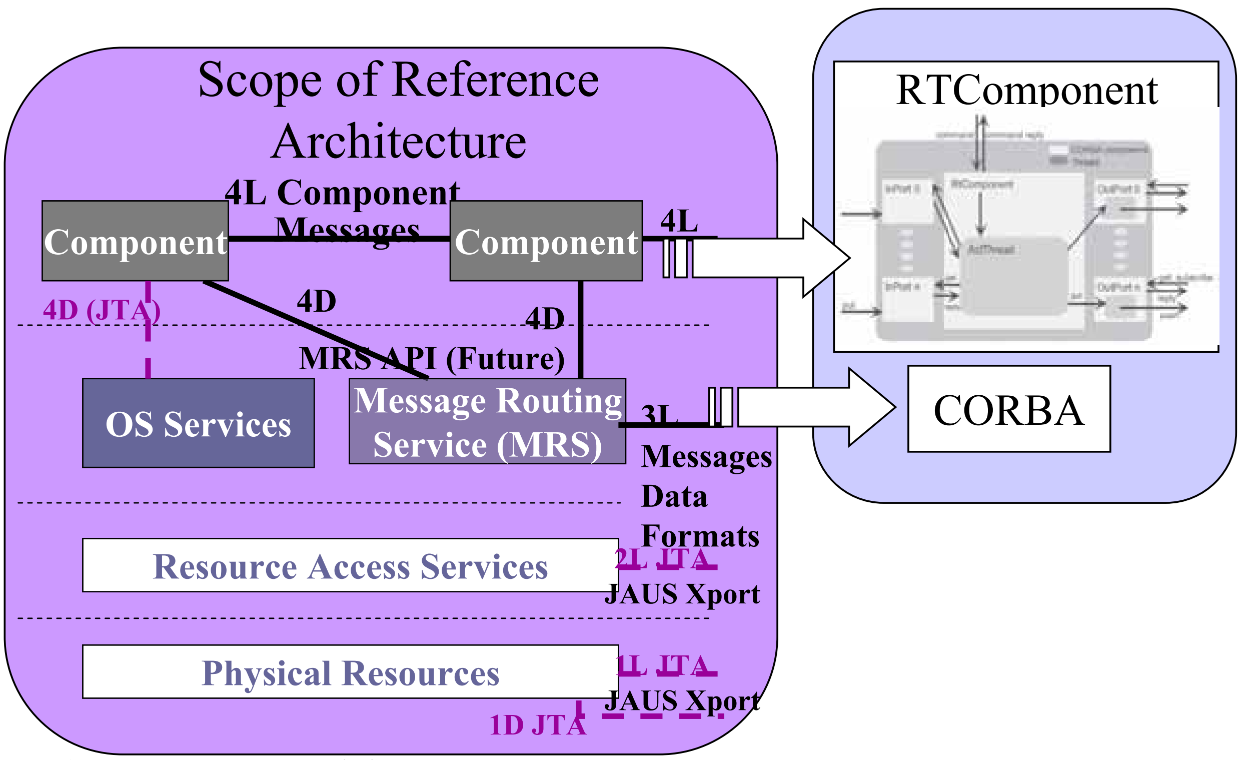
JAUS : Application specification of mobile robot (Defines "What", independent from "How")

- Software structure of mobile robot.
- Function assignment to each component.
- What to communicate between components. = messages
- Roll and Behavior of component.

OpenRTM-aist : Implementation of Components and Message Communication (Defines "How", "What" is general)

- The JAUS component can be realized by the RT-Component.
- How to communicate, which can be realized with InPort and OutPort.

4.2. Framework – JAUS - RTM Mapping



(From JAUS Tutorial)

4.3. Framework – Implementation

Implementation of JAUS specification

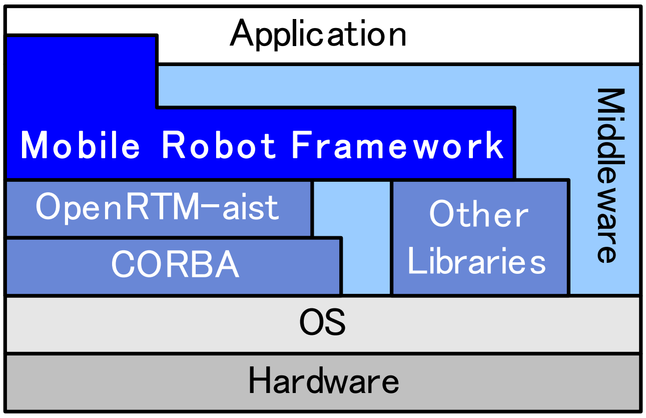
- Base class that behaves as JAUS Component utilizing RT-Component.
- JAUS Component classes that are already defined .
- IDL definition of JAUS Messages.

Tool

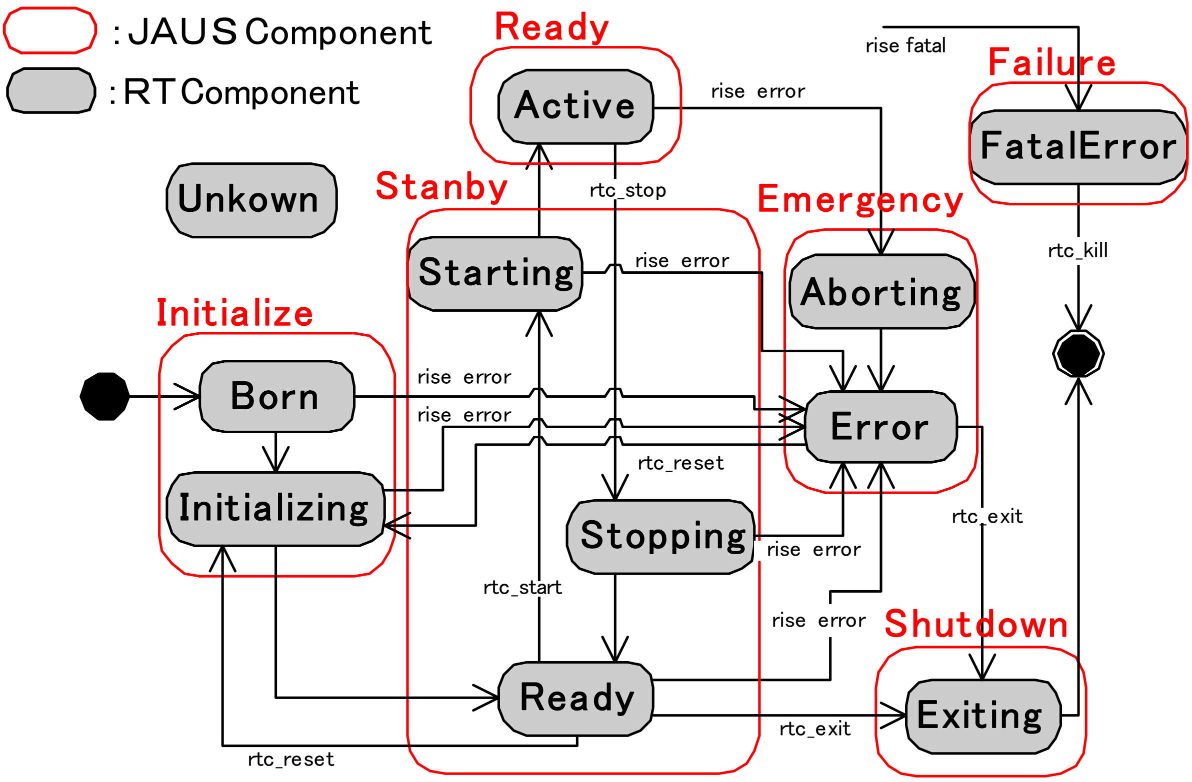
- Start program.
- In/OutPort connection program.

Programming language:

C++

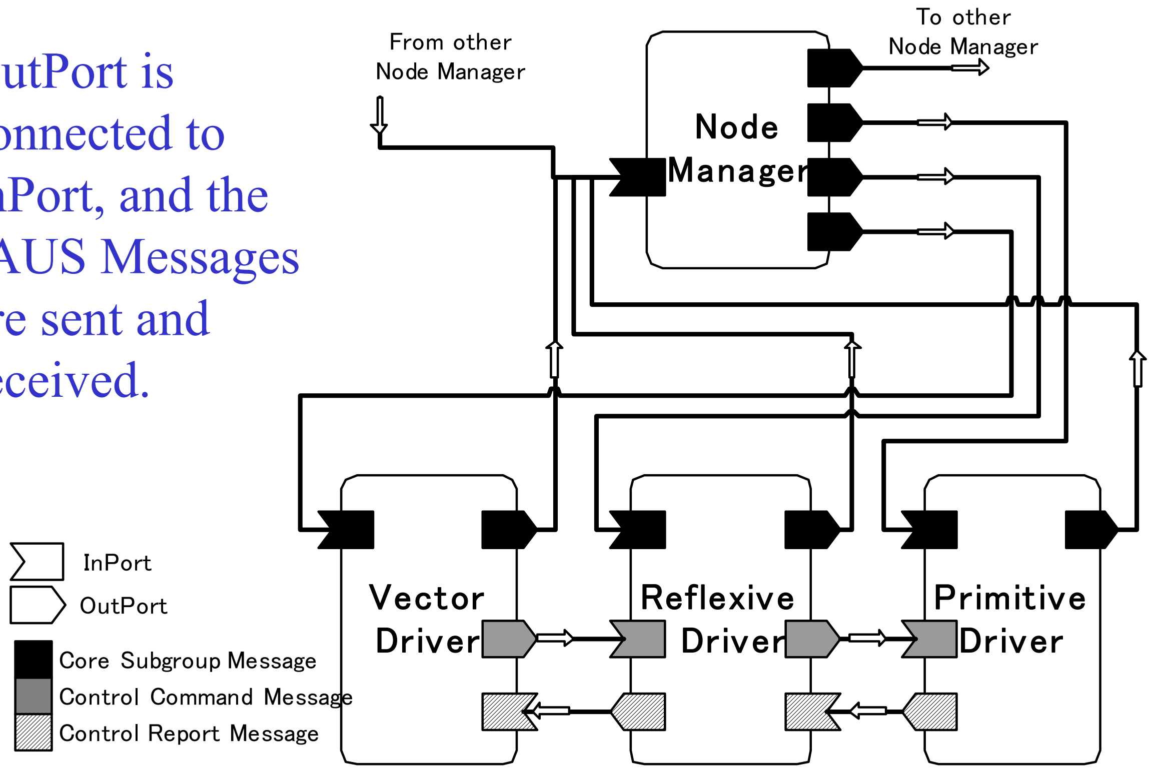


4.4. Framework–Mapping of State Transition of Component

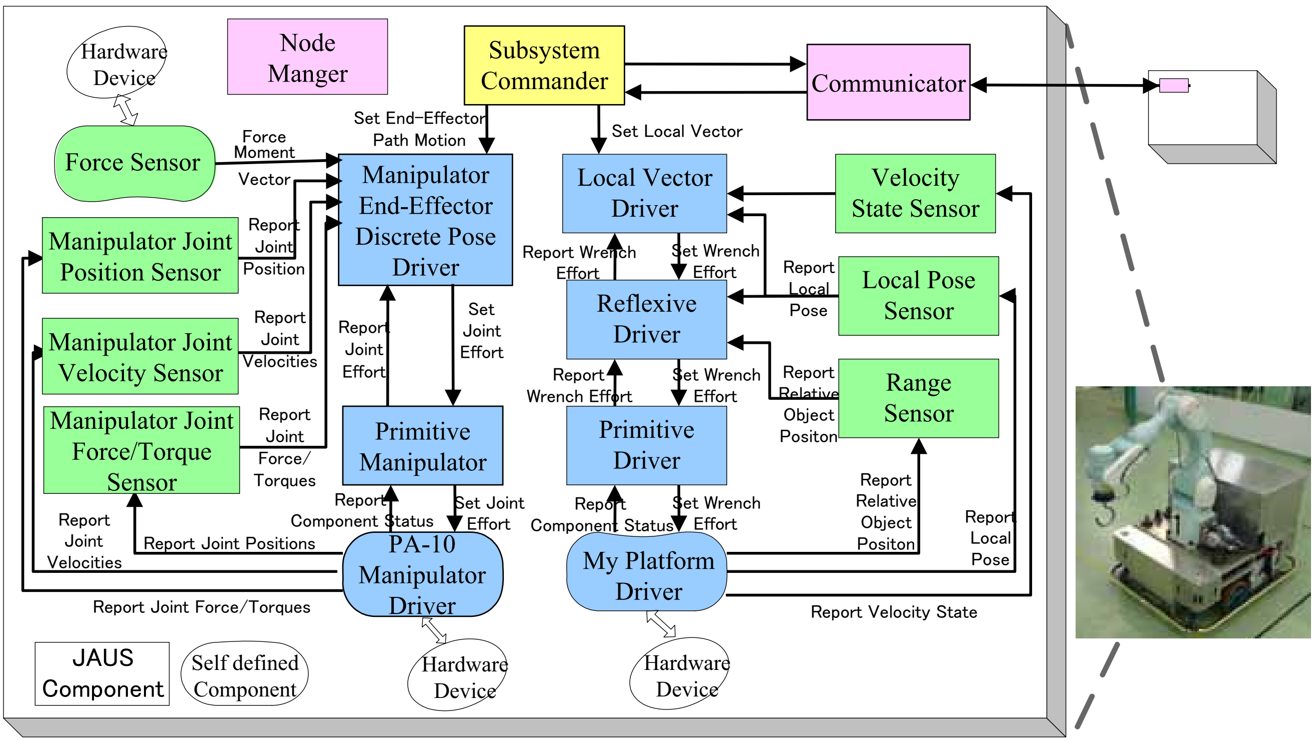


4.5. Framework – Inter Component Communication

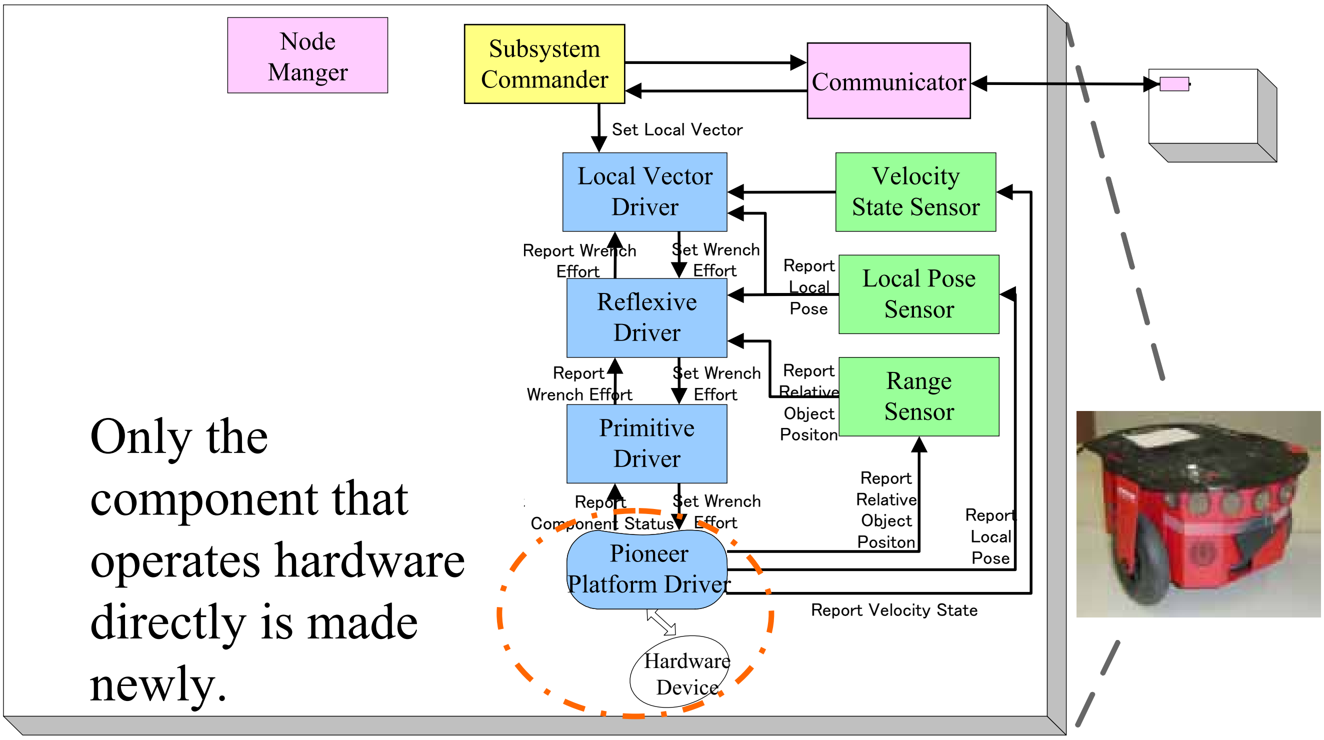
OutPort is connected to InPort, and the JAUS Messages are sent and received.



5.1. Experimental example – Mobile Robot subsystem 1



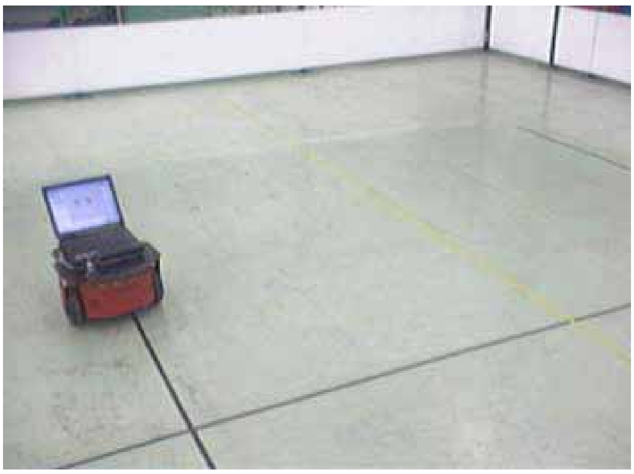
5.2. Experimental example – Mobile Robot subsystem 2



Only the component that operates hardware directly is made newly.

5.3. Experimental example – Simple Demo

Running along a circular trajectory with a radius of 1.0m.



5.4. Future work

Many robots work together !

6.1. Summary

Proposal of utilizing open architectures

- JAUS and RT-Middleware combination are shown.
- A development framework for mobile robot is proposed.
- We developed the software program for two mobile robots by our framework.

Requirements for Standard of RoboticsDTF

- It can be implemented with RT-Middleware easily.
- The robot integrator can use it easily. Usually, they are not software vendors bat software users.
- It must have compatibilities between old and new version.

We want to utilize standards very much !

Thank you for your attention.

PRISMTECH

Applicable SWRadio Spec Concepts for Robotics Domain

Jerry Bickle
Chief Scientist
PrismTech Corporation - USA

Spectra Software Defined Radio Products

SWRadio Specs

- UML Profile for SWRadio
- SWRadio Facilities

- Compliance is at the level of usage
- Current PSMs at this time are XML and CORBA interfaces
  - Other PSMs could easily be defined.

UML Profile for SWRadio

- Basic Elements
  - Extensions of UML Concepts
    - Component, Device, Port, Interface, Property, Artifacts
  - Model Libraries
    - Basic Generic Interfaces and Types
- Consist Of
  - Generic Component Framework
  - Comm Channel & Equipment

Component Framework

- Rich set of semantics for component and container based development
  - Basic Types and Properties for defining components configurable, query, characteristic, and capacity properties
  - Basic port and interface types for indicating the type of ports.
  - Basic interfaces for generic component and domain management.
  - Basic component types for application development, device, and domain management.
- Heavy lifting is done by the tools that implement the profiles.
- PIMs base upon this Component Framework can be transformed into any technology.
- Deployment & Configuration Implementation is not dictated (e.g., static, dynamic, or both)
- Extensible
  - By addition
    - *QoS and FT profile*
    - *DDS profile*
  - By another domain (e.g., robotics) extending or constraining

Communication Channel & Equipment

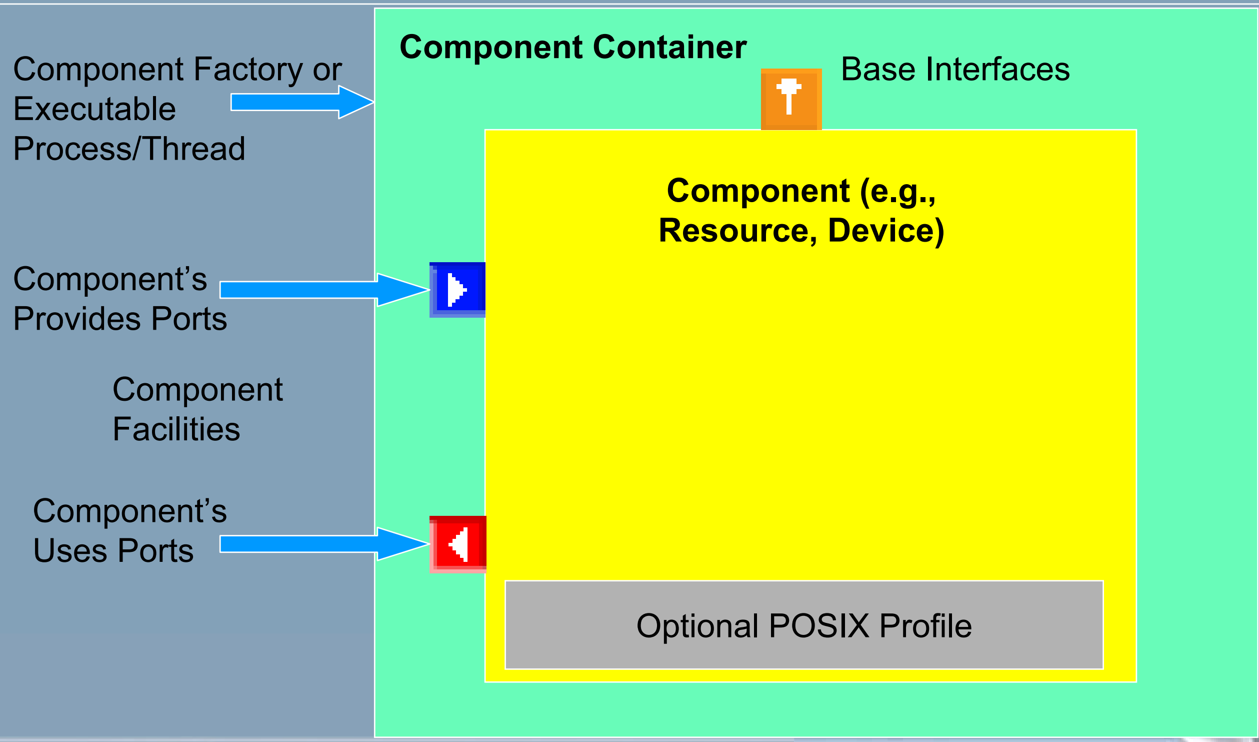
- Rich set of semantics for communication channel and equipment definition
  - Basic Types and Properties for communication channel
  - Basic Types and Properties for communication equipment
    - *Amplifier*
    - *Analog Port*
    - *Antenna*
    - *Converters*
    - *Digital Port*
    - *Filters*
    - *IO Device*
    - *Processor*
    - *Switch*
    - *Etc.*

SWRadio Facilities

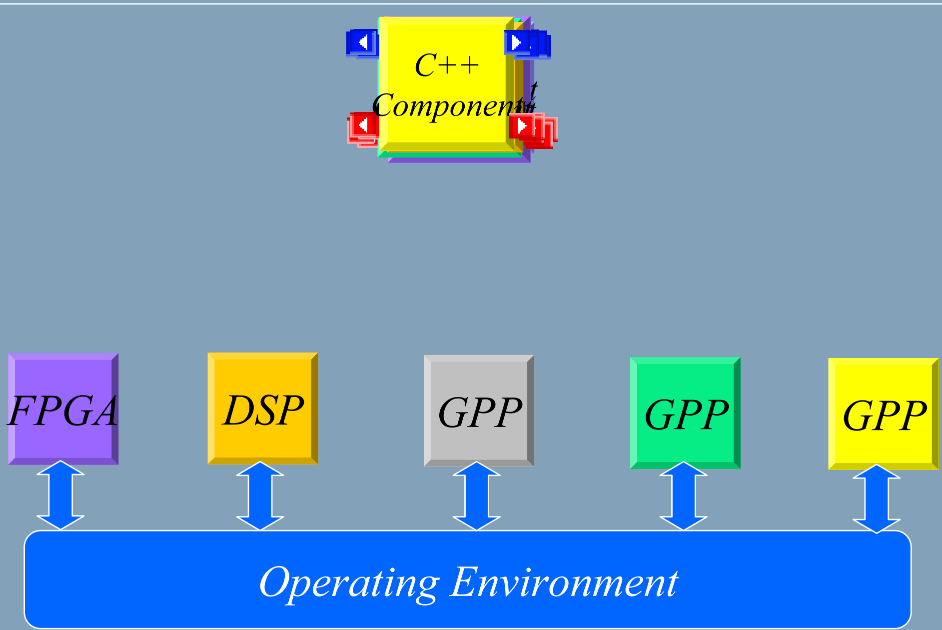
- Optional Interfaces for Component Behavior Extensions
  - Common layer
    - PDUs, Error Control, Flow Control, etc.
  - Networking
    - Link Layer, MAC
  - Physical Layer
  - Radio Control

- Robotics may use these facilities as appropriate and supplement these facilities with their own that are specific to Robotics domain

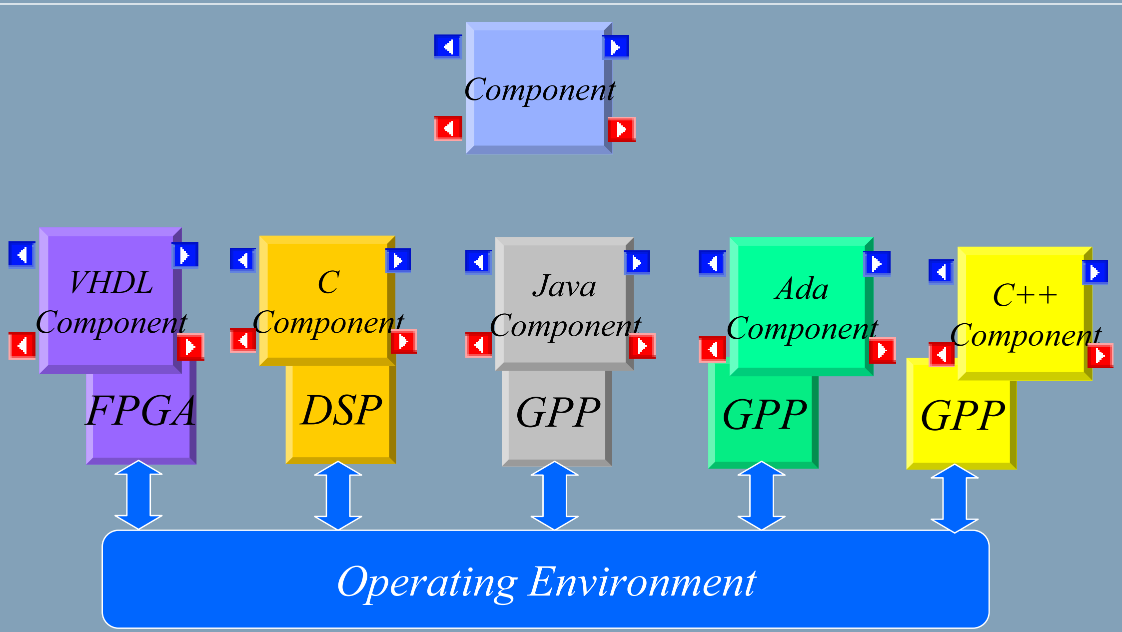
Component



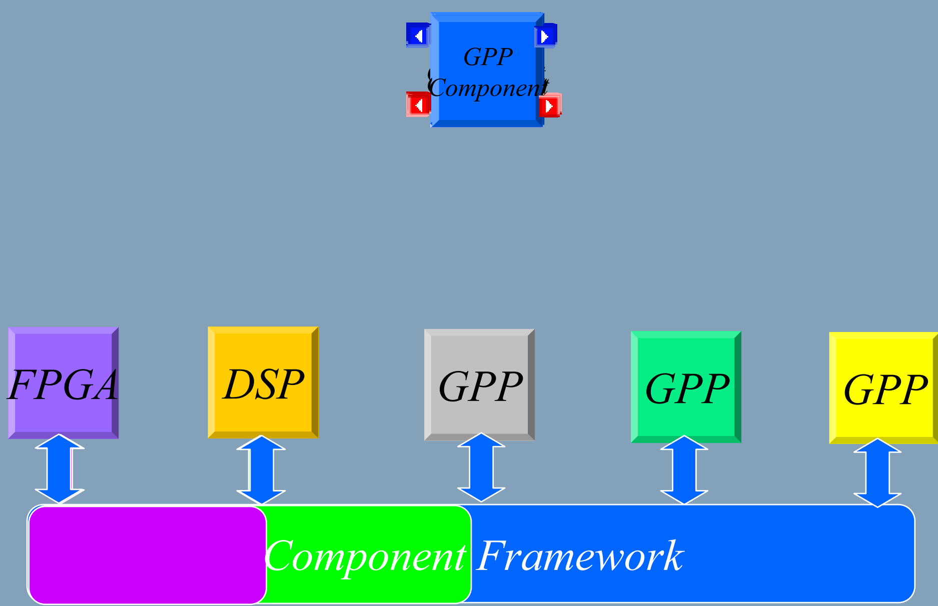
MDA Component PIM to PSM



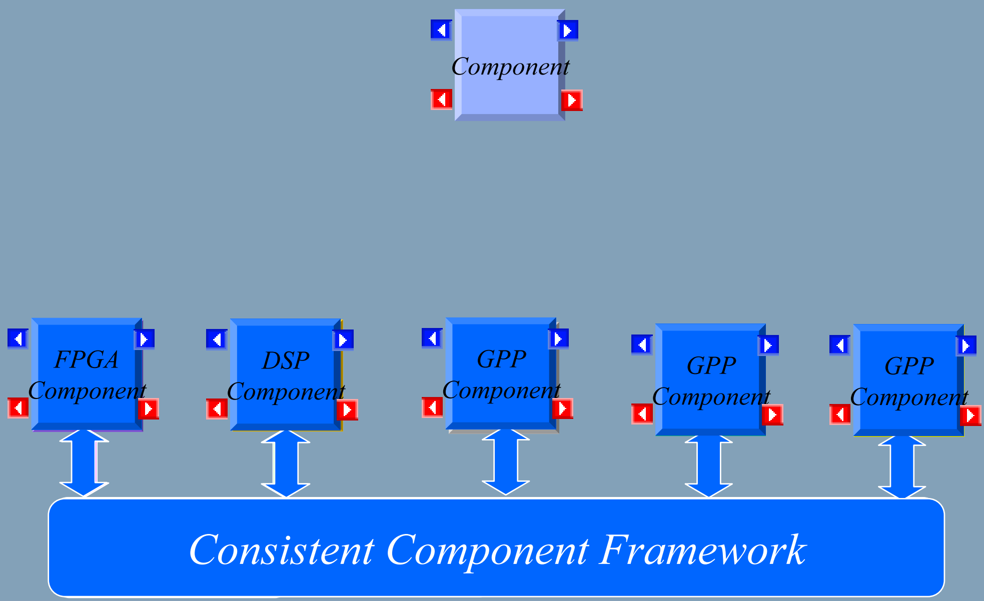
MDA Component PIM to PSM



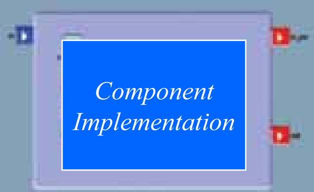
Consistent Component Framework



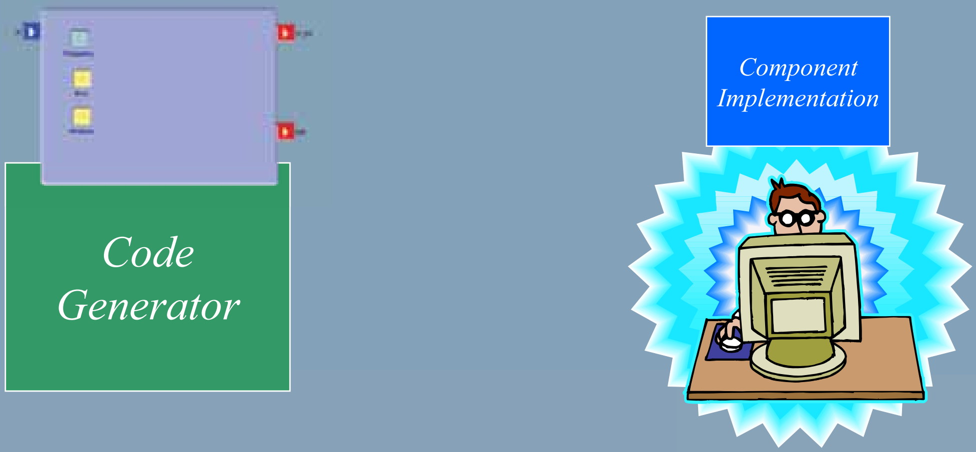
Consistent Component Framework



Separation of the Component Infrastructure from the Component Implementation



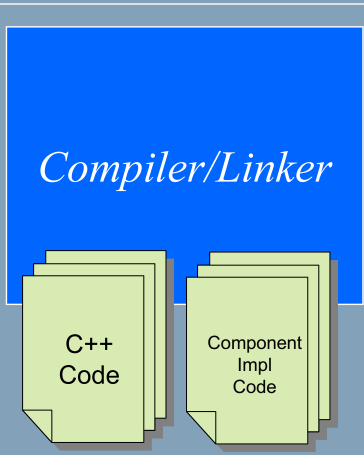
Separation of the Component Infrastructure from the Component Implementation



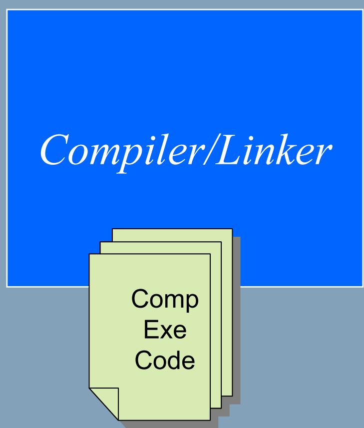
Separation of the Component Infrastructure from the Component Implementation

Proper separation allows Component infrastructure and Component Implementations to Mature Independently

Separation of the Component Infrastructure from the Component Implementation



Separation of the Component Infrastructure from the Component Implementation



SWRadio RTF

- The specification is being broken up into multi volumes:
  - to be more accessible by other domains and
  - keeping profiles and PIM facilities independent of technology concepts that realize those concepts.
  - Profiles (XMI file)
    - Component Framework
    - Comm Channel & Equipment
  - Component Facilities (XMI file)

  - Technologies
    - POSIX Profiles
    - Component Descriptors (e.g., XML DTDs, Schemas, etc.)
    - CORBA Interfaces
    - Other Technologies can be easily defined

Illustration of Profile Concepts

- Illustrate the profile definition using a UML Tool
- Illustrate the profile in action using a MDD tool

Response to the Robotics RFI

robotics/06-02-21

Virginie Watine

virginie.watine@fr.thalesgroup.com

COMPonent Approach for Real-time and Embedded

Introduction

- This response answers partially to the RFI
  - Subject
    - Software Engineering enforced by ***Software Design***
      - Component-oriented
      - For Real-time and Embedded systems
      - Main objectives:
        - Wide applicability - potential support for various architectural styles and large range of targets
        - Enforcement of separation of concerns in the development process - allowing business experts to focus on their business expertise
        - Portability, Reusability
        - Ease of integration
  - Context
    - Research work sustained by 2 European research projects:
      - ***IST (COMPARE)*** and ITEA (MERCED)
      - ***Adaptation of a standard Component Model for RTE***
        - OMG Lightweight CCM
    - Application to several domains including SDR, Industrial Control Systems

Presentation Outline

- Motivations - Trends in RT/E Software
  - ➔ Are not specific to Robotics
  - ➔ Are relevant also for the Robotics domain
- Component / Container Model characteristics
- Adaptations of the model ➔ RT/E
  - ➔ Based on a standard Lw-CCM
  - ➔ With two main extensions
    - Open Container and Container Services
    - Connectors
- Conclusion

RT/E Systems Characteristics

- Software part in RT/E Systems (including Robotics) is increasing
  - ➔ Move from fixed wired hardware ➔ flexible logics (software)
  - ➔ ... but still heterogeneous hardware
    - Reconciled approach is needed
- Software in RT/E Systems is becoming more complex:
  - ➔ More functionality, heterogeneity
  - ➔ More variability, versatility
  - ➔ More integration in large-scale systems:
    - More connectivity
    - More remote manageability
  - ➔ Move towards dynamic (re)configuration - Self-xxx

RT/E Systems Characteristics

- RT/E Systems have still to deal with 'real' world:
  - Time constraints
  - Safety constraints
  - Security constraints
  - Resource constraints
    - Footprint
    - Energy

Some of those characteristics are conflicting

How to Manage those Conflicting Trends?

- Big challenges are:
  - Heterogeneity, Conflicting constraints, Complexity...
  - And also: Cost, Time to market, Reuse...
- Traditional approach for building RT/E software is no more appropriate
  - Used to focus only on performance and timeliness
  - Ad-hoc, fragmented

Need to move from performance-centric to complexity-centric ...
...without loosing the performance and time support!

- Need proper execution support (a.k.a. ***Middleware***)
  - Modelling is not enough: Code generation is made easier (or even feasible...) if what is modelled finds its natural realisation
  - Dynamic reconfiguration mandates a runtime support

What is Middleware?

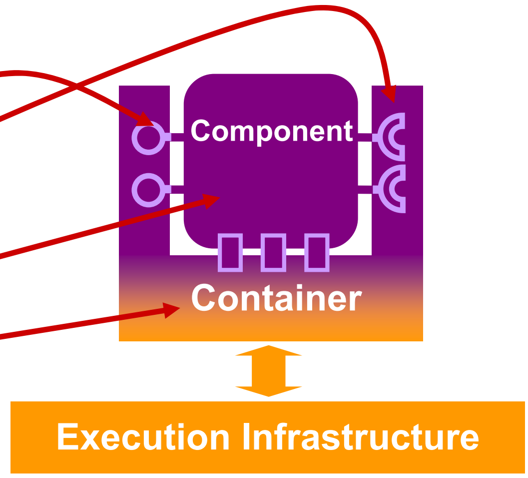
- ***Anything*** that stands between the pure application code and the raw (networked) platform
  - Not only the communication support
  - Should be the ***mediator*** between the application code and the platform resources and services
- What characteristics for new RT/E middleware?
  - Affordable
  - Providing
    - Suitable support for application break-down in manageable (reusable) parts
    - Suitable support for RT/E non-functional properties
    - Separation of concerns (SoC)
    - Isolation, partitions
  - Sound abstractions that can be realised by various implementations
  - Programmatic ➔ declarative

Component/Container Model

- A key architectural pattern
  - ➔ Explicit description of:
    - provided services
    - requested services
  - ➔ Separation of concerns:
    - business logic
    - 'technical' properties

(containers are provided as part of the infrastructure)

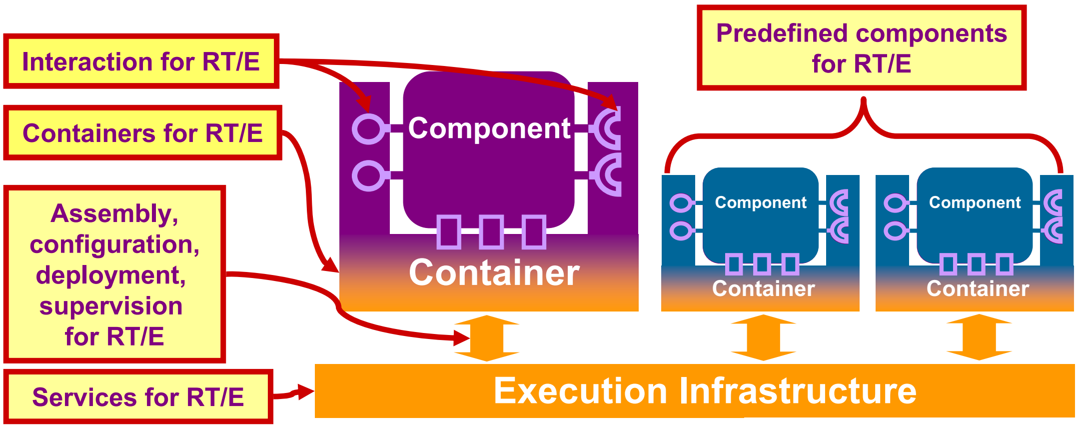
(based on descriptors ➔ move from fully programmatic to declarative)



- An existing model - Lightweight CCM
  - ➔ Open ***standard***
  - ➔ Tailored ***for Embedded***
  - ➔ CCM is ***not (only) CORBA!***
    - CCM is exactly meant for abstracting the underlying platform

Adaptation of CCM -> RT/E

- Real-Time & Embedded CCM framework
  - ➔ Based on Lw-CCM (including D&C) - CCM != CORBA
  - ➔ With adaptations specific to RT/E



  - ➔ Sustained by 2 European research projects
    - IST/***COMPARE*** - ITEA/MERCED
    - Focus on ***Extensibility*** and ***Usability***

COMPARE Project

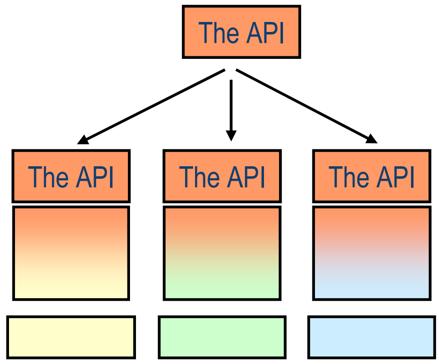
- ***COMP****onent* ***A****pproach for* ***R****eal-time and* ***E****mbedded*
  - ➔ T0 = June 2004 - Duration 30 months
  - ➔ Small focused consortium
  - ➔ 2 implementations / 3 use-cases
    - RT-CORBA (SwRadio / Industrial Control System)
    - OSEK-VDX (Electrical Circuit Breaker)

THALES — PRISMTECH The Integration Server Company — cea — TRIALOG — Universidad Politécnica de Madrid — Schneider Electric

www.ist-compare/org

Compared Approaches for Portability

Classical approach for portability



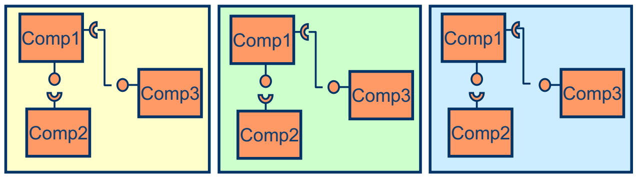
Same interface and semantics on top of different platforms

- ➔ may be tricky to define and implement
- ➔ forbids the best use of the underlying platform (RTOS, middleware)
- ➔ costly to maintain if the API is huge

Promoted approach

No "portability layer"

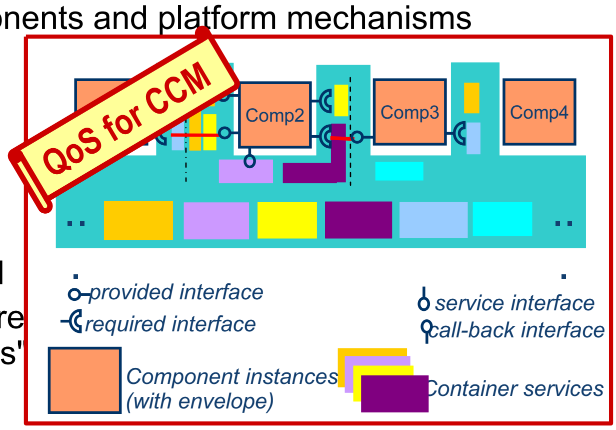
- ➔ Same component assembly projected on different targets
- ➔ Focus on integration of technology-neutral components
- ➔ Components hosted in a container, itself using specific platform services
- ➔ ***Containers as full platform mediators and independently configured***



- ➔ Need to provide adequate technical support

First LwCCM Adaptation - Containers

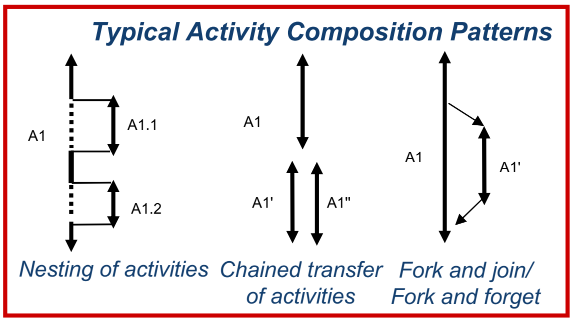
- Containers
  - Allow to actually implement separation of concerns
  - Are responsible for:
    - Components creation and activation
    - Communication between components
    - Mediation between components and platform mechanisms
- Implementation
  - Container optimised for its target
  - Modular and composable
    - Only what is really needed
    - Open container architecture to host "Container Services"
      - Set of related objects
      - Inserted at specific Integration Points (component creation, connection, interaction...)
      - With well defined interfaces (service and call-back) to ease integration



➔ *Focus on extensibility*

Example : CS for RT Properties

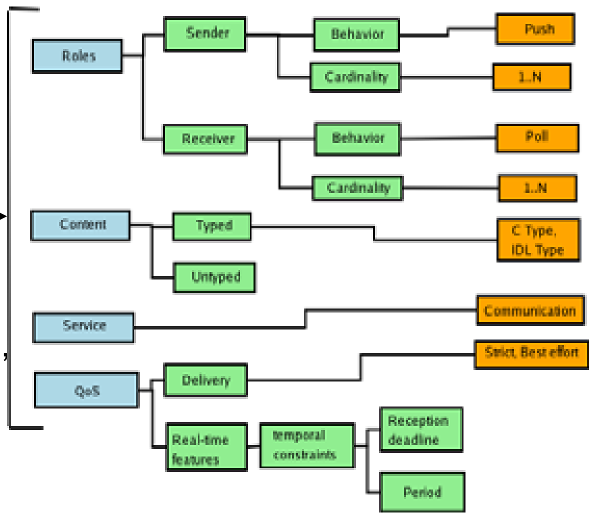
- How to isolate timing properties?
  - ➔ Concept of Activity as a behavioural design abstraction
    - Close to the RT-CORBA Distributable Thread
    - = An execution path whose starting point and ending point are identified
      - Associated to a component instance, a facet, and a method
      - Can span multiple component instances
    - Activity instance: realisation of an Activity
      - at a particular time, under a particular context



  - ➔ Real-time formal vocabulary used to annotate activities
    - Classical concepts: release time, relative deadline,importance...
  - ➔ Container Services to create and propagate Activity instances and to set accordingly the basic mechanisms
    - Realisation is highly platform-specific (achieved using RT-CORBA mechanisms, but could be made using directly RTOS primitives)

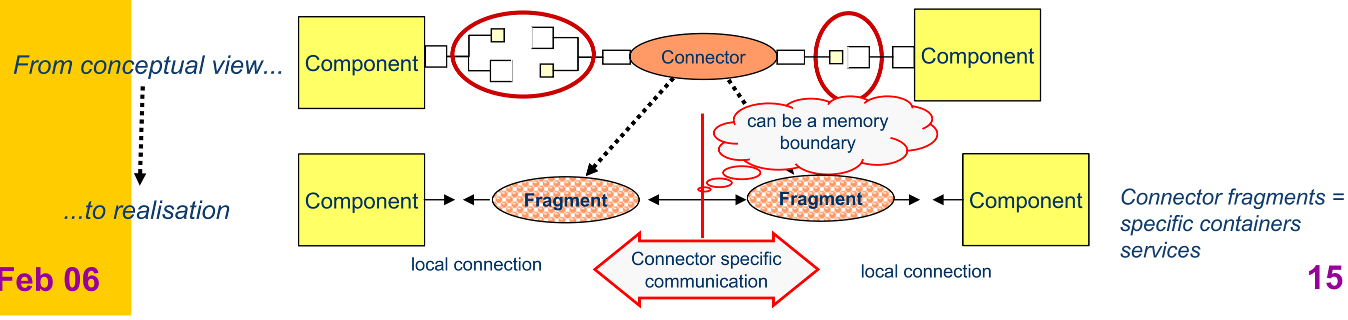
2nd LwCCM Adaptation - Connectors

- Interaction between Components
  - ➔ Initial CCM interaction support is limited
    - Synchronous method call (without...
    - Event send/receive ... QoS)
    - Recent addition of Streams
  - ➔ Other modes are required
    - To support various architectural styles
    - Proper interactions condition components existence
      - ➔ ***Focus on usability***
    - Examples:
      - Deferred synchronous call
      - Event passing with priorities, buffering...
      - Various pub/sub
      - Blackboard
      - ...

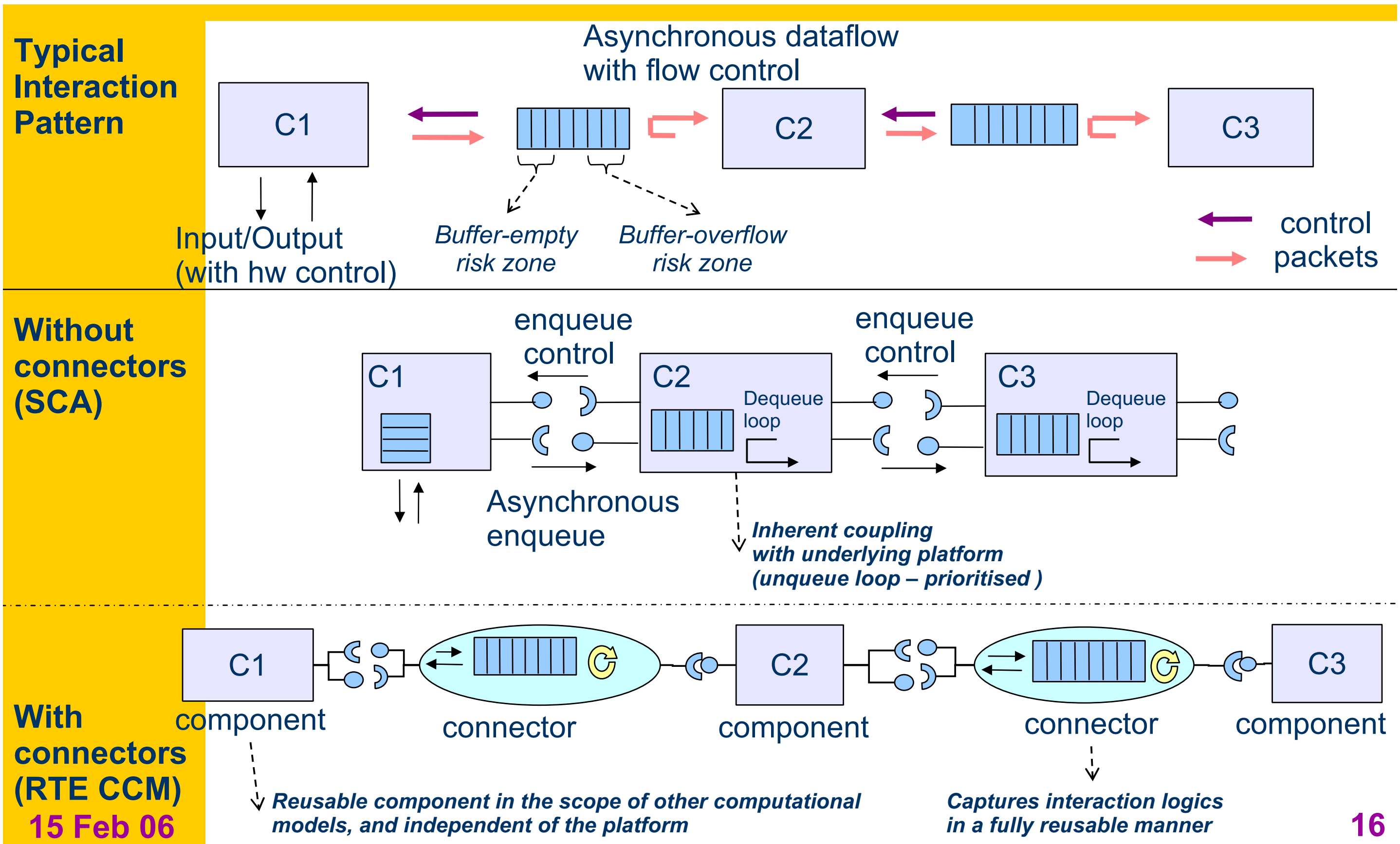


Interaction Support => Connectors

- Constructs from ADL concepts...
  - Components = functional entities
  - Connectors = interaction entities
- ...Introduced in Lightweight CCM...
  - To realise proper interactions
    - from generic to domain-specific
    - With suitable configuration mechanisms
- ... In a reusable manner
  - Packaging of connectors
    - including connector binaries, description, and code generation plugins for particular connectors that need it

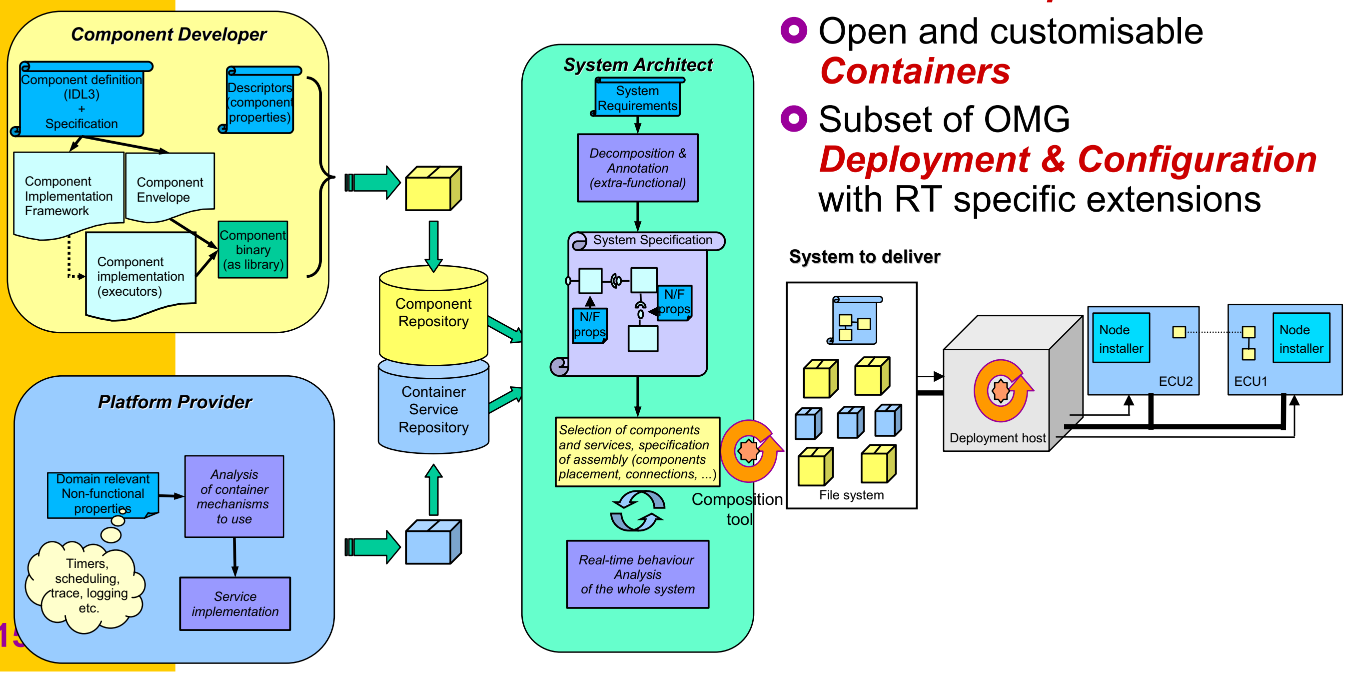


Example of Connectors: Waveform Design

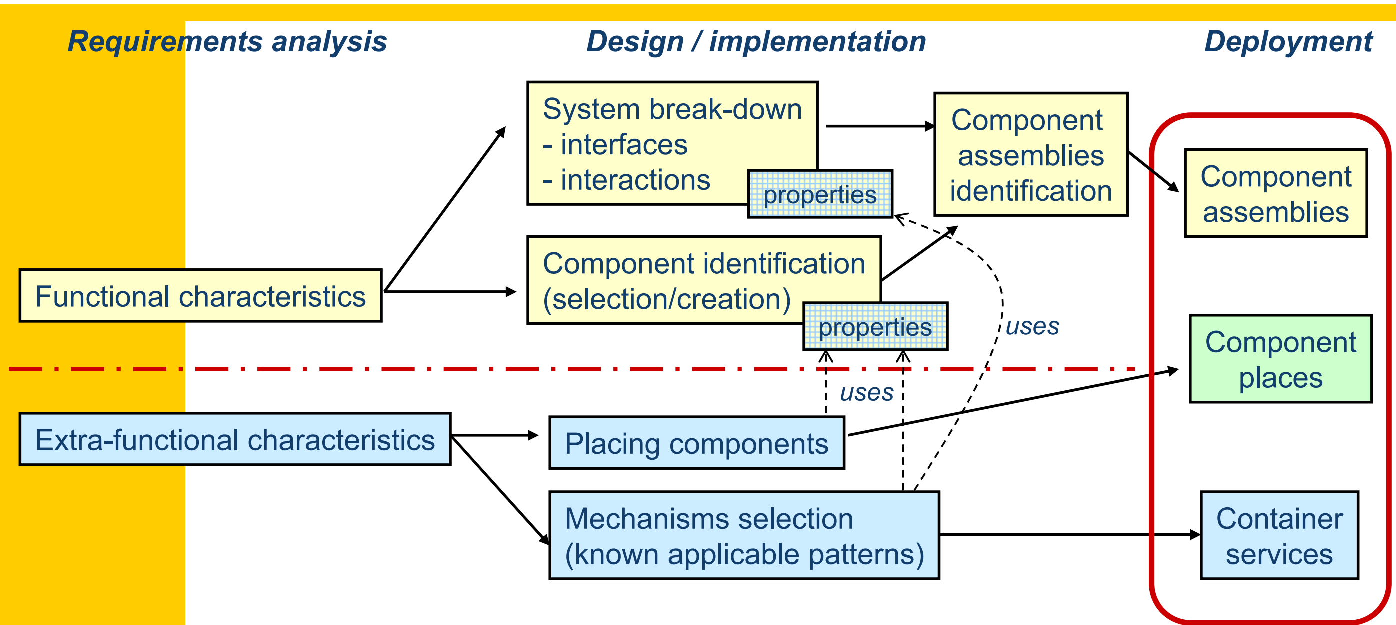


Component-based Development Process

- RT/E Component Framework ➔ Support for building ***OMG Lw-CCM*** compliant applications
  - Portable ***Components***
  - Open and customisable ***Containers***
  - Subset of OMG ***Deployment & Configuration*** with RT specific extensions



Impact of Separation of Concerns



➔ Benefits
- Clear separation of roles: ***focus on expertise***
- Actual component isolation: ***better reuse***
- Latest design decisions (component localisation...): ***faster easier integration***

OMPARE

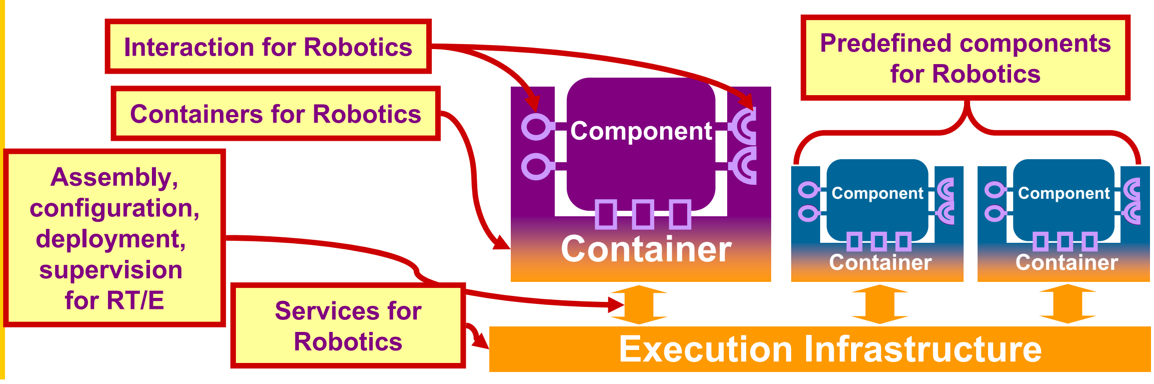
Conclusion 1/2

- New challenges for RT/E Software
  ➔ proper execution support:
  - Support reusable components
  - Provide adequate non-functional properties
  - Abstract the underlying platform as much as possible
  - Support heterogeneous requirements
- ***Component/Container*** is a key model:
  - That helps actual reuse
  - SoC and QoS declaration are very powerful
  - Can be instantiated according to the underlying platform
  - ➔ Its adaptation to RT/E characteristics is on going
    - Sustained by several Research initiatives
    - ***Lightweight CCM as starting point - Open Standard***
    - Initial focus has been put on ***suitability*** and ***extensibility***
  - ➔ Its validation on realistic use cases has started and is promising

OMPARE

Conclusion 2/2

- Such model ***with its separation of concerns*** would be very beneficial for Robotics domain
- The current realisation targets wide applicability
- ***Profiling for Robotics*** could consist in:
  - ➔ Definition of ***dedicated Connectors***
  - ➔ Definition of ***dedicated Container Services***
  - ➔ Definition of ***dedicated Predefined Components***

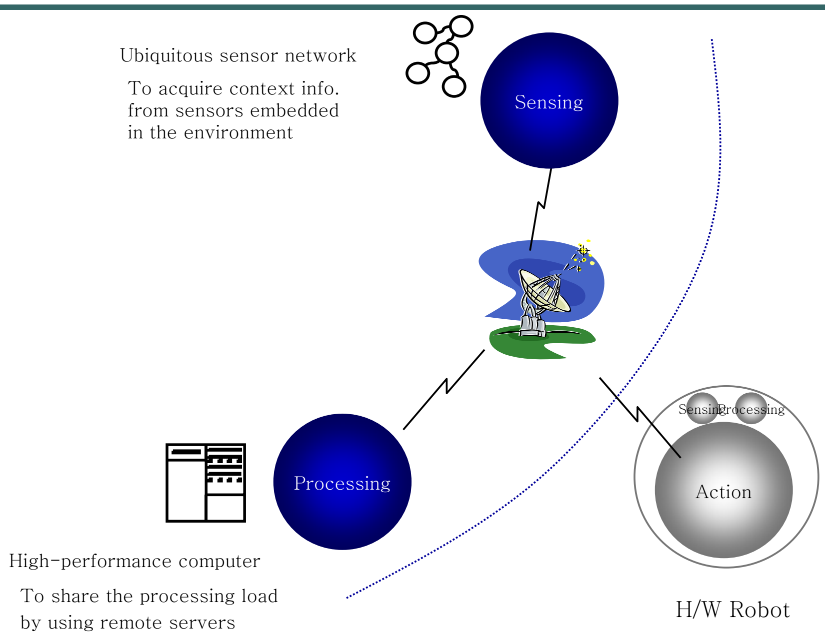


Robot Server Middleware

Response to the OMG Robotic Systems RFI

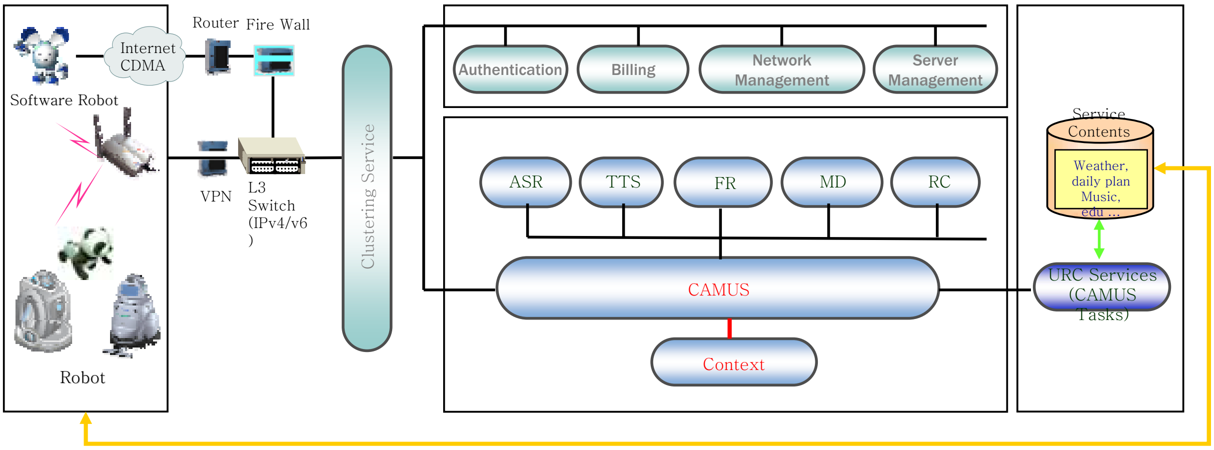
Seung-Ik Lee, Hyun Kim

Basic Concept of URC

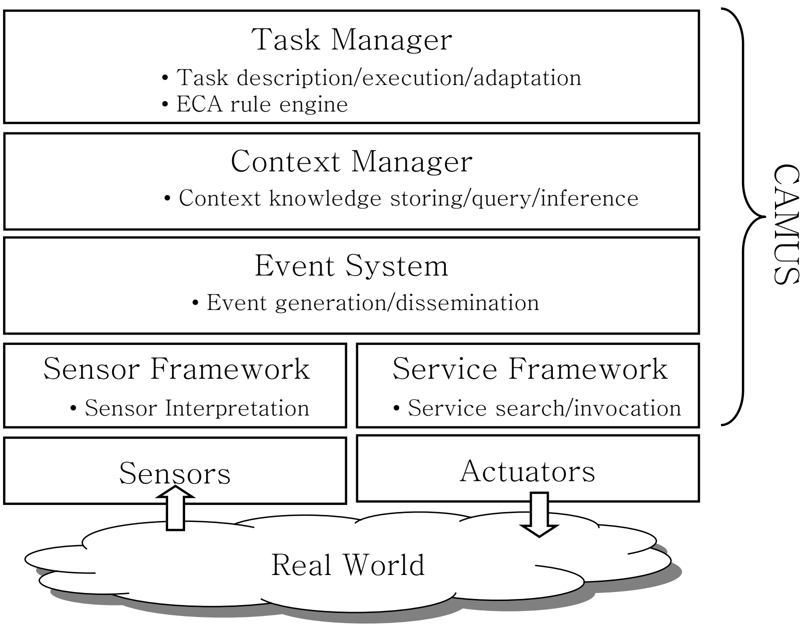


URC Server

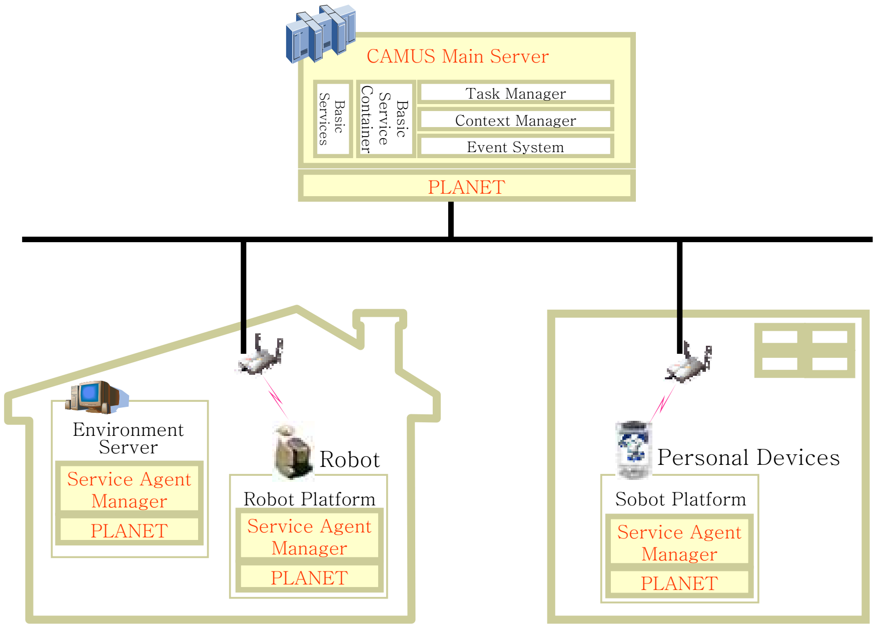
- It expands the robot functions and services, improves the context-awareness, and enhance the robot intelligence.



Conceptual System Architecture

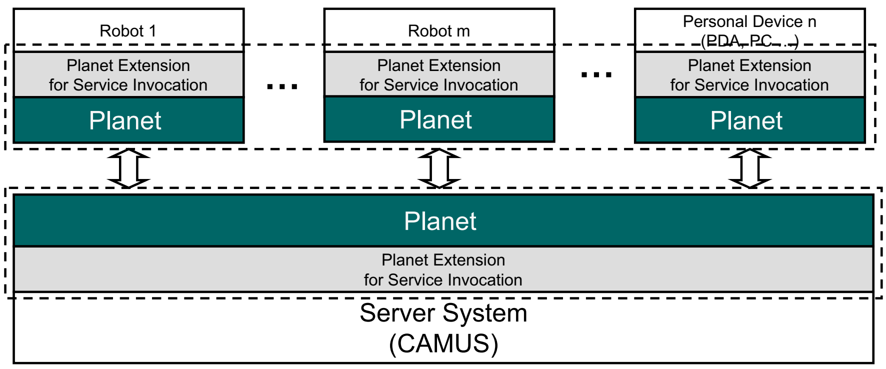


System Configuration



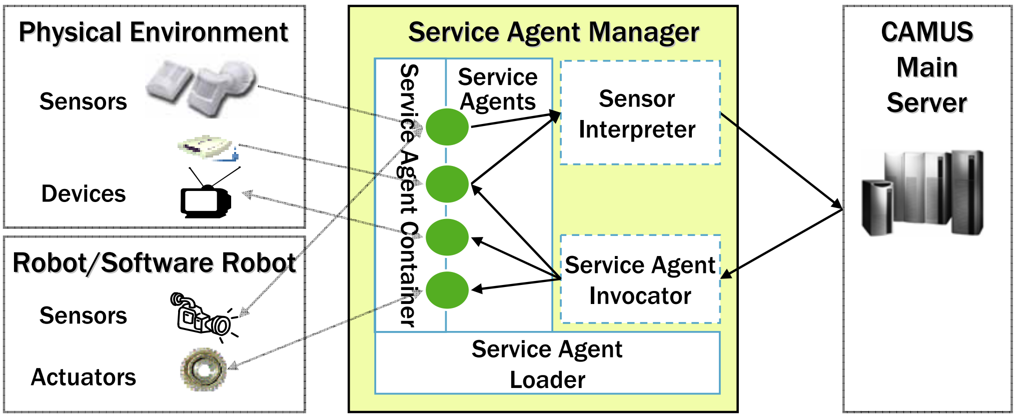
PLANET

- The communication framework between robot platforms and the robot server
  - It is based on the remote method invocation
  - It has light-weight protocol to minimize the sending/receiving messages
  - It uses fault-tolerant communication mechanism
  - It supports different kinds of operating systems and programming languages.



Service Agent Manager

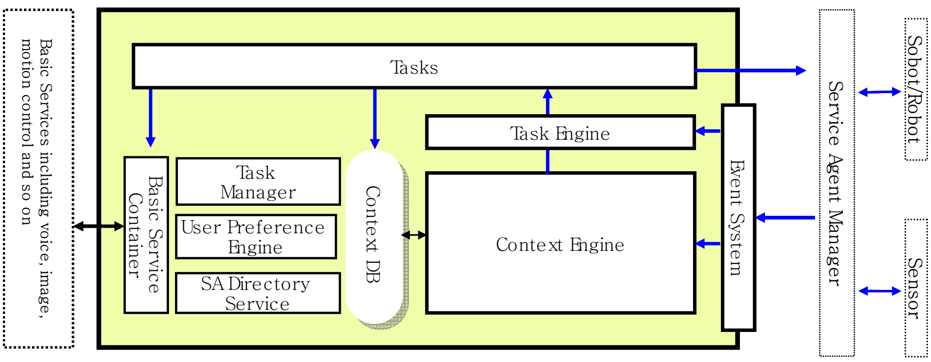
- The system for managing and controlling sensors and actuators (Service Agents) in an environment or a robot platform.
  - It acquires information from various sensors and disseminates them to the CAMUS Main Server.
  - It controls actuators according to the control commands from a CAMUS Main Server.



CAMUS Main Server

- It manages context information including User and Environment Contexts.
- It generates and disseminates appropriate events to the Task Engine according to the context change, which enables tasks to take actions required to the current context.
- It provides the framework to develop context-aware applications.
- It provides the service container which manages service components that provide basic functions of robots

CAMUS Main Server

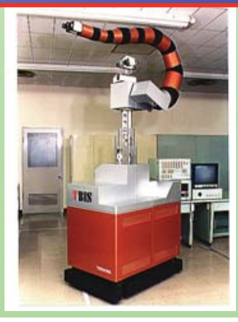


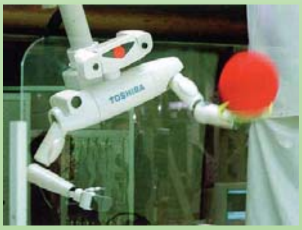
URC Field Test Services

Toshiba's Approach to RT Standardization and Where the Standardization is Needed

OZAKI, Fumio Toshiba Corporation

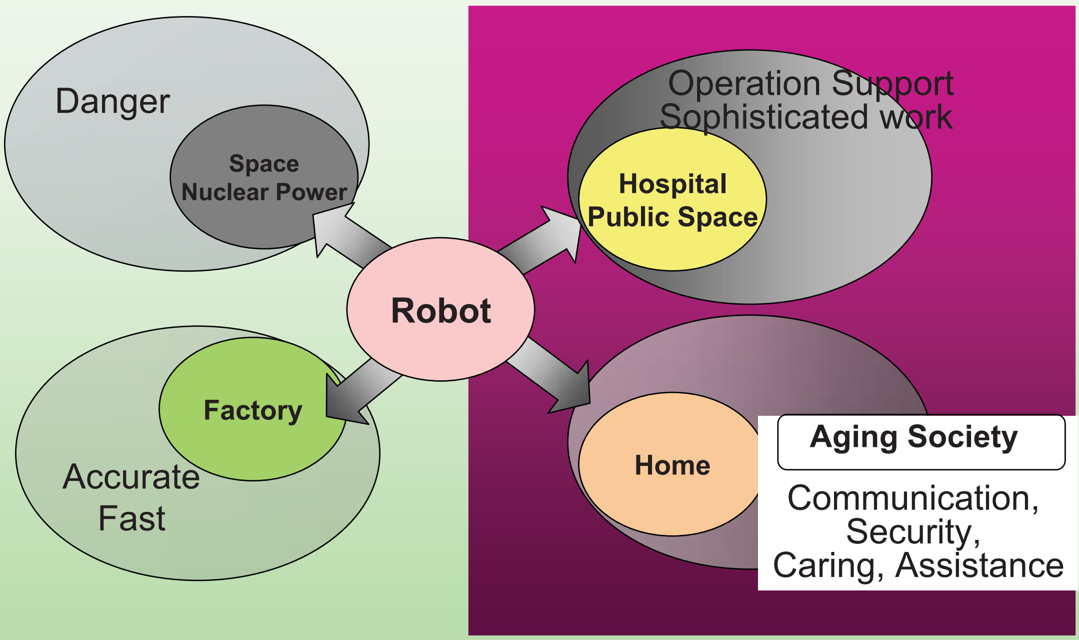
- Toshiba's Approach
  - Target Area
  - Toshiba Robot History
  - Open Robot Controller Architecture
- RT Standardization





Some of the above robot images are courtesy of AIST, and NEDO.

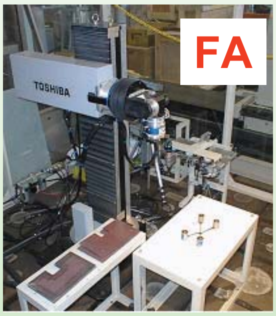
Target Area



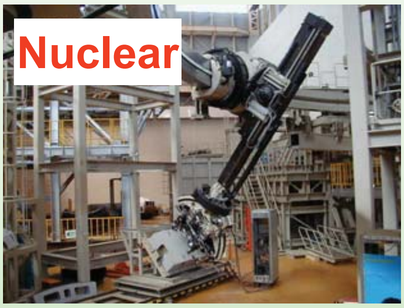
Accumulated Fundamental Technologies ⇒ Expansion of Knowledge & Human Capabilities
Safe & Secure Society

TOSHIBA

Toshiba Robot History



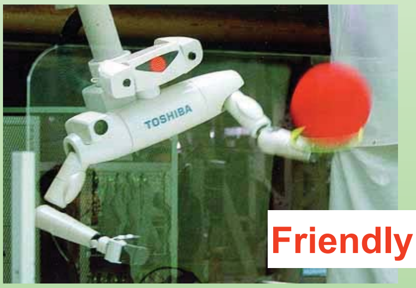
Finishing Robot



ITER Robot

Micro

Pipe Inspection Robot



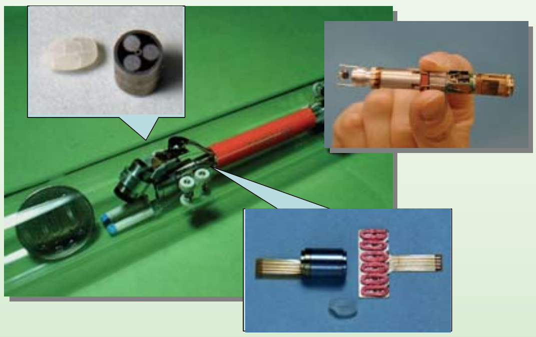
Human Friendly Robot

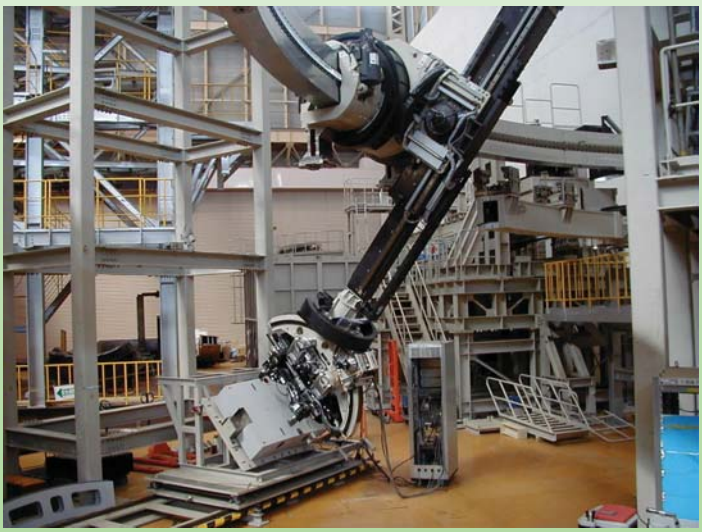
Space Robot

Surgical Assistant Robot

TOSHIBA

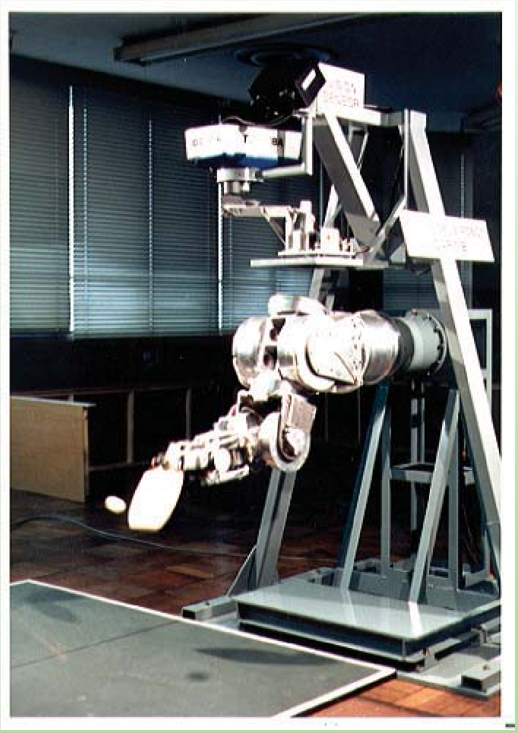
Maintenance Robots for Nuclear Facilities

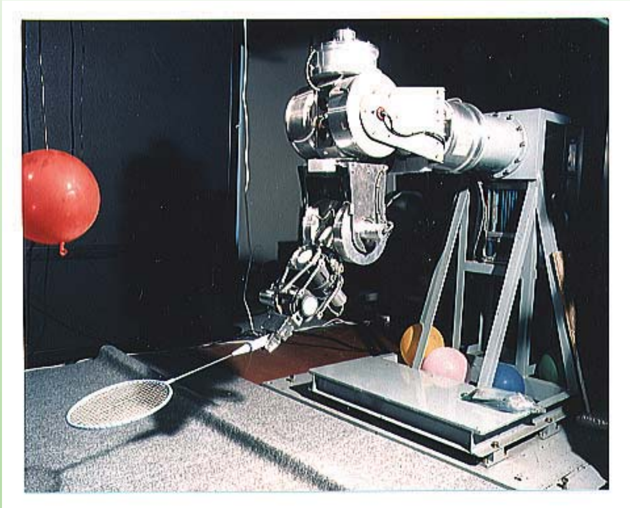




Courtesy of Japan Atomic Energy Research Institute

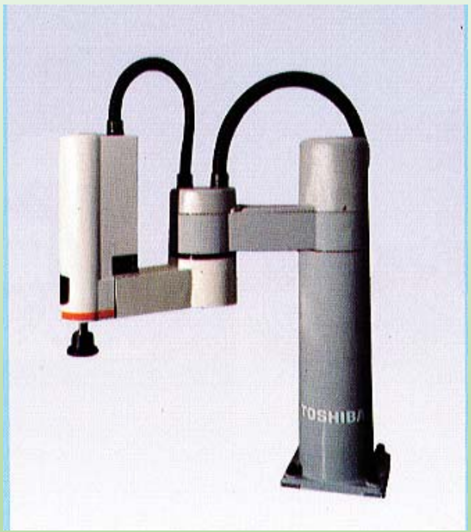
Juggling Robots

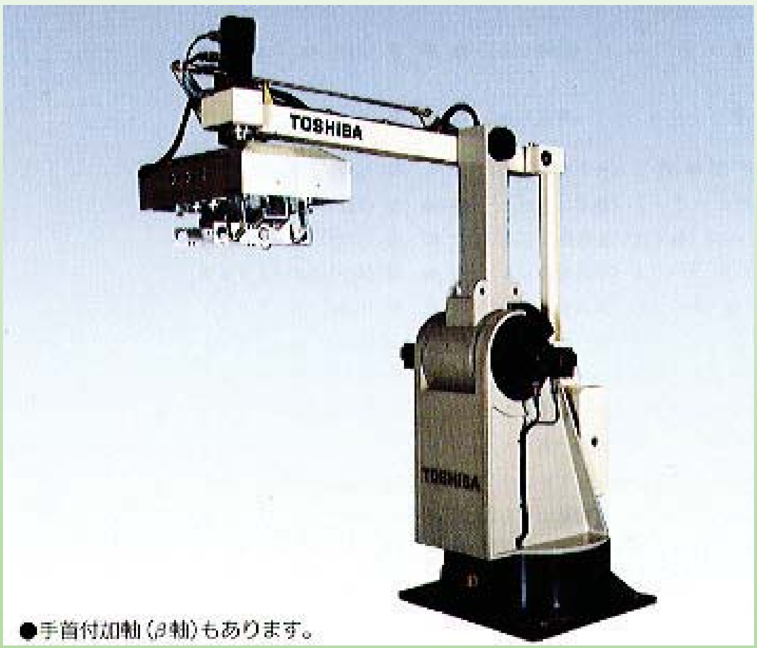




TOSHIBA
Toshiba Robots History

Industrial Robots

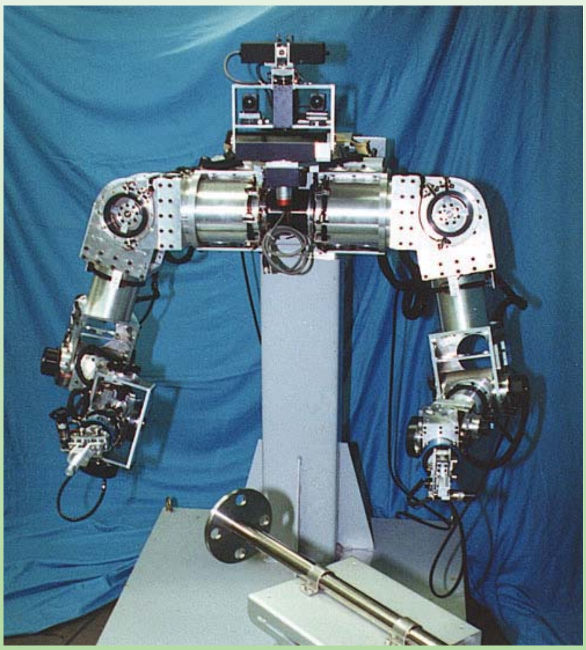




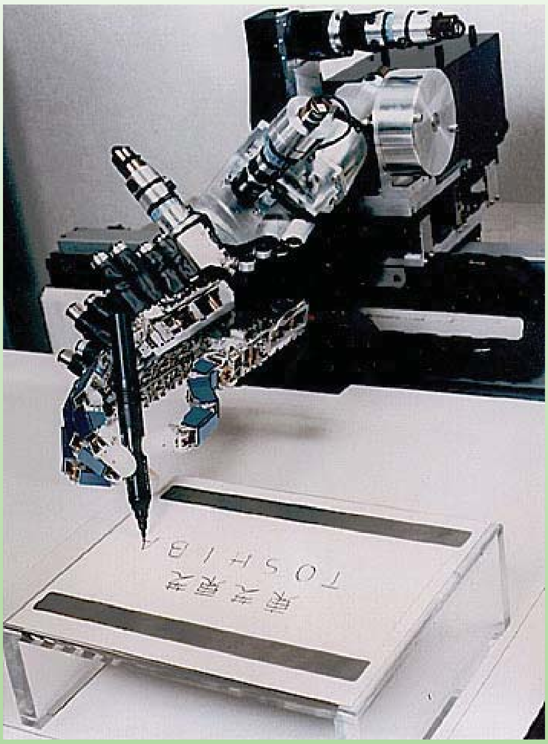
Grinding Robots

Remote Maintenance Robots for Overhead Power Distribution Lines

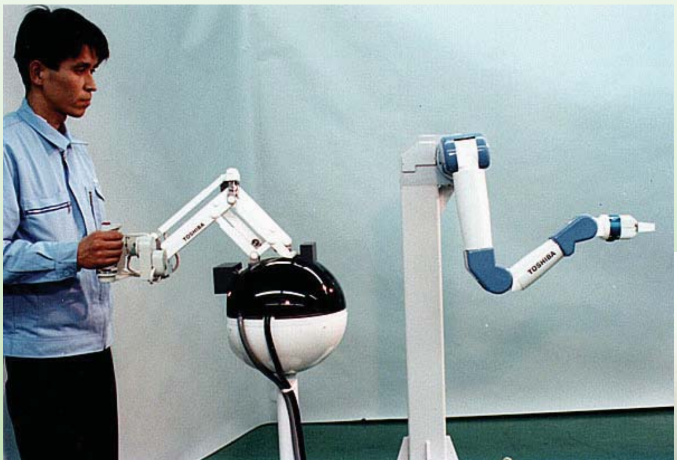
Courtesy of TEPCO

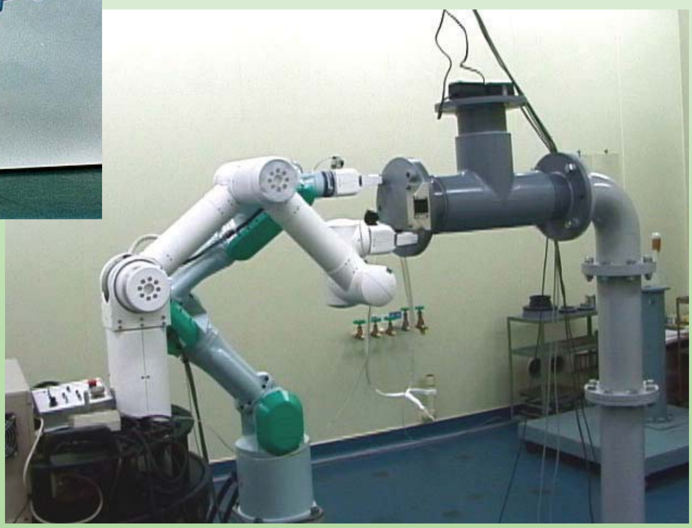


Multi-Fingered Hand



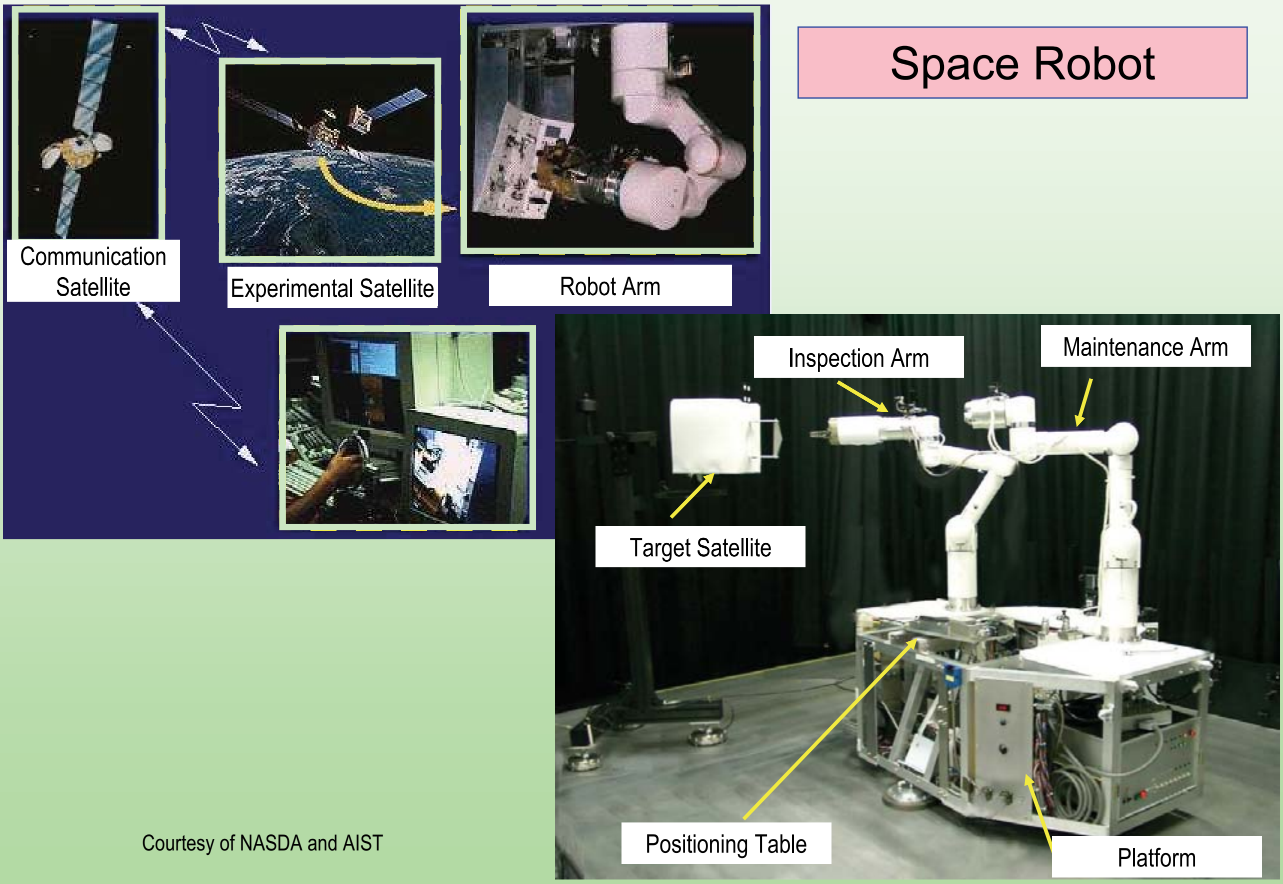
Remote Operation Robots





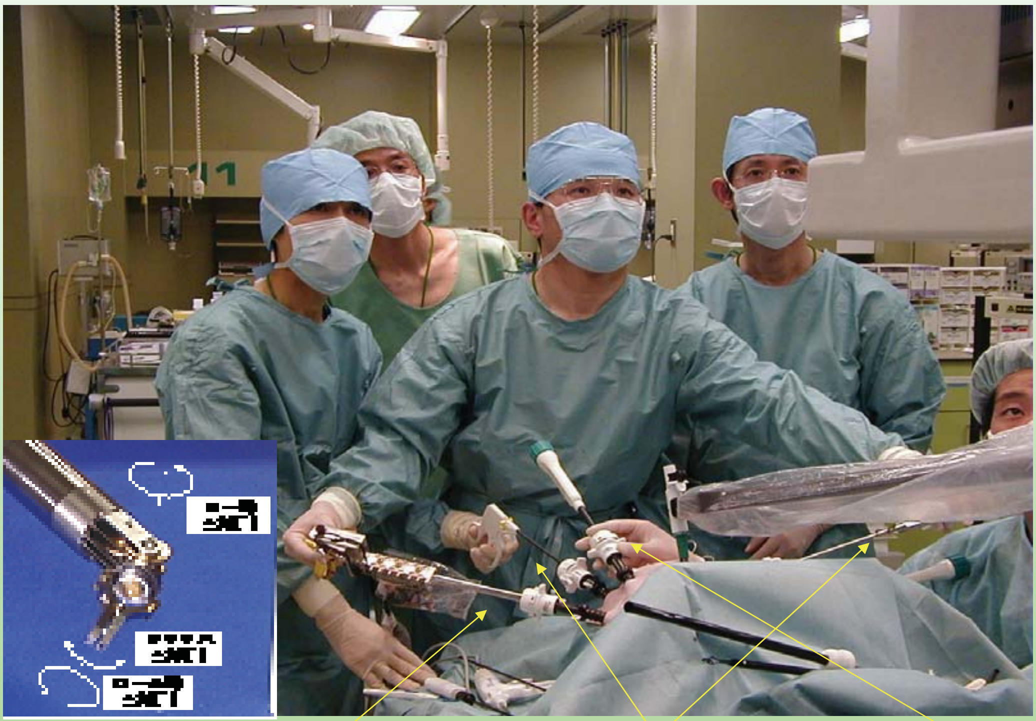
Courtesy of AIST

Space Robot



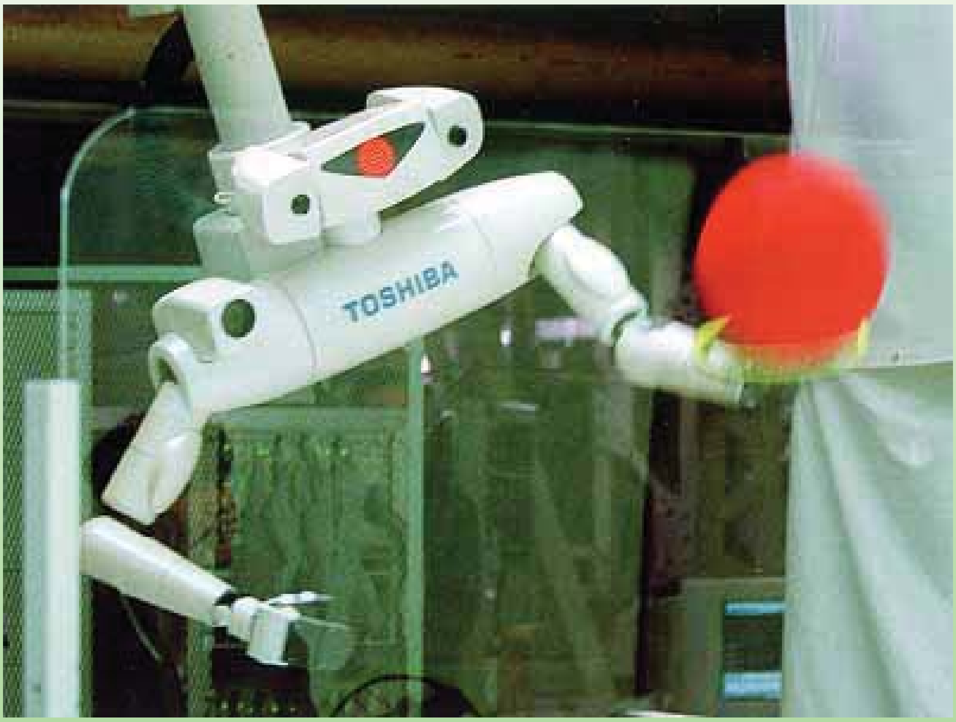
Courtesy of NASDA and AIST

Medical Robots



Courtesy of Keio University

Human Friendly Robot



Technologies in Newly Developed Robot

※Apri: **A**dvanced **p**ersonal **r**obot with **i**ntelligence

ApriAlpha_v3
--Apri, the sharp ear--

ApriAttenda
--Apri, the go-with-you robot--

* The new robots were developed as part of the New Energy and Industrial Technology Development Organization's (NEDO) "Next-Generation Robot Commercialization Project (Prototype Development Support Enterprise)" Some of the image processing technologies used for the robot that can follow a person were jointly developed with the Tokyo University of Science.
* These robots were demonstrated at AICHI EXPO's "Prototype Robot Exhibition," at the Morizo and Kiccoro Exhibition Center from June 9 to June 19, 2005.

Developing History of Robot Software in Toshiba

1980 - Developing System from Scratch
- VME Bus System, Multi Bus System
- Assembler, C
- No OS

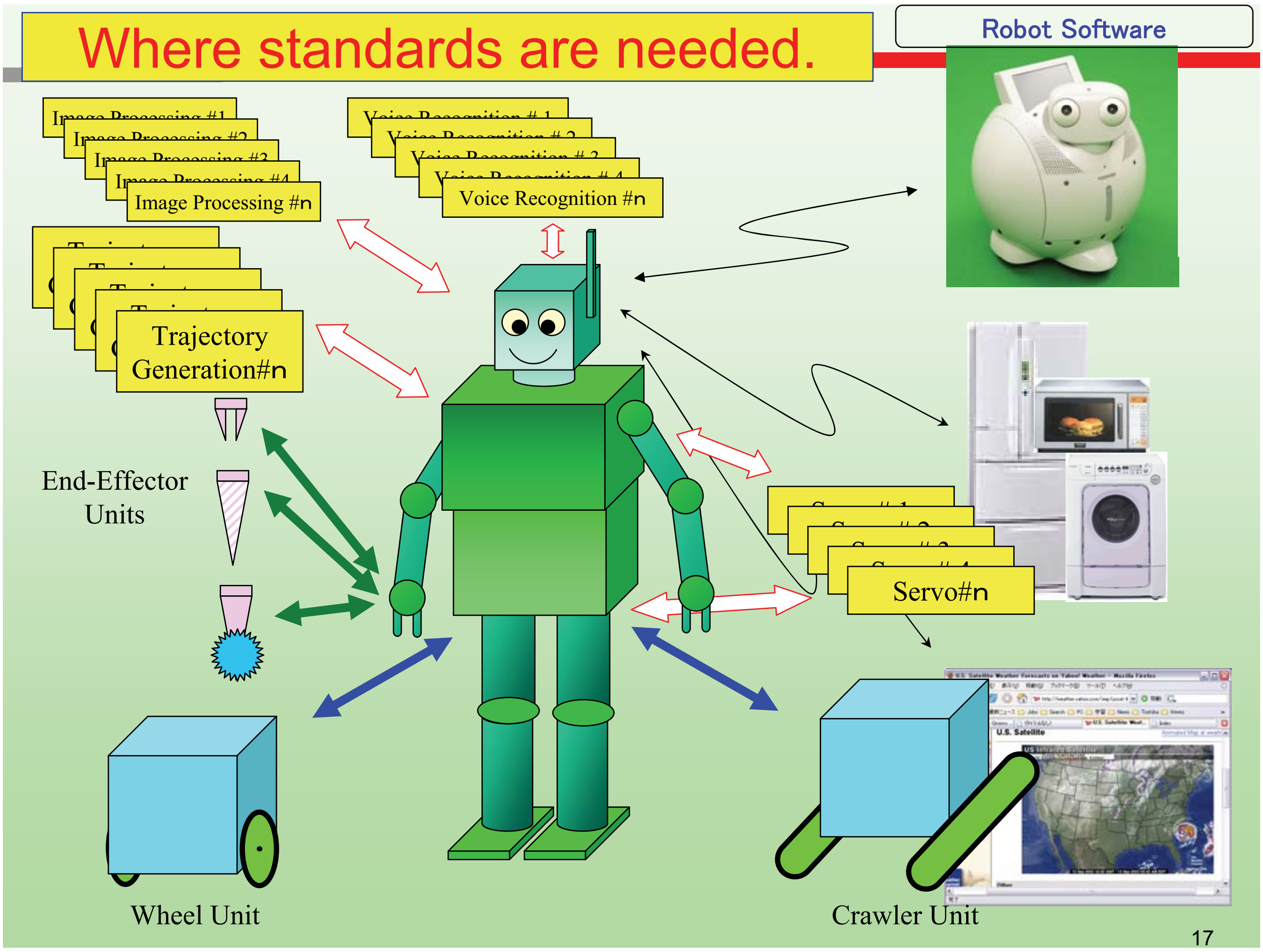
1990 - Introducing General Purpose Real Time OS （VxWorks）
- VME Bus System
- C/C++

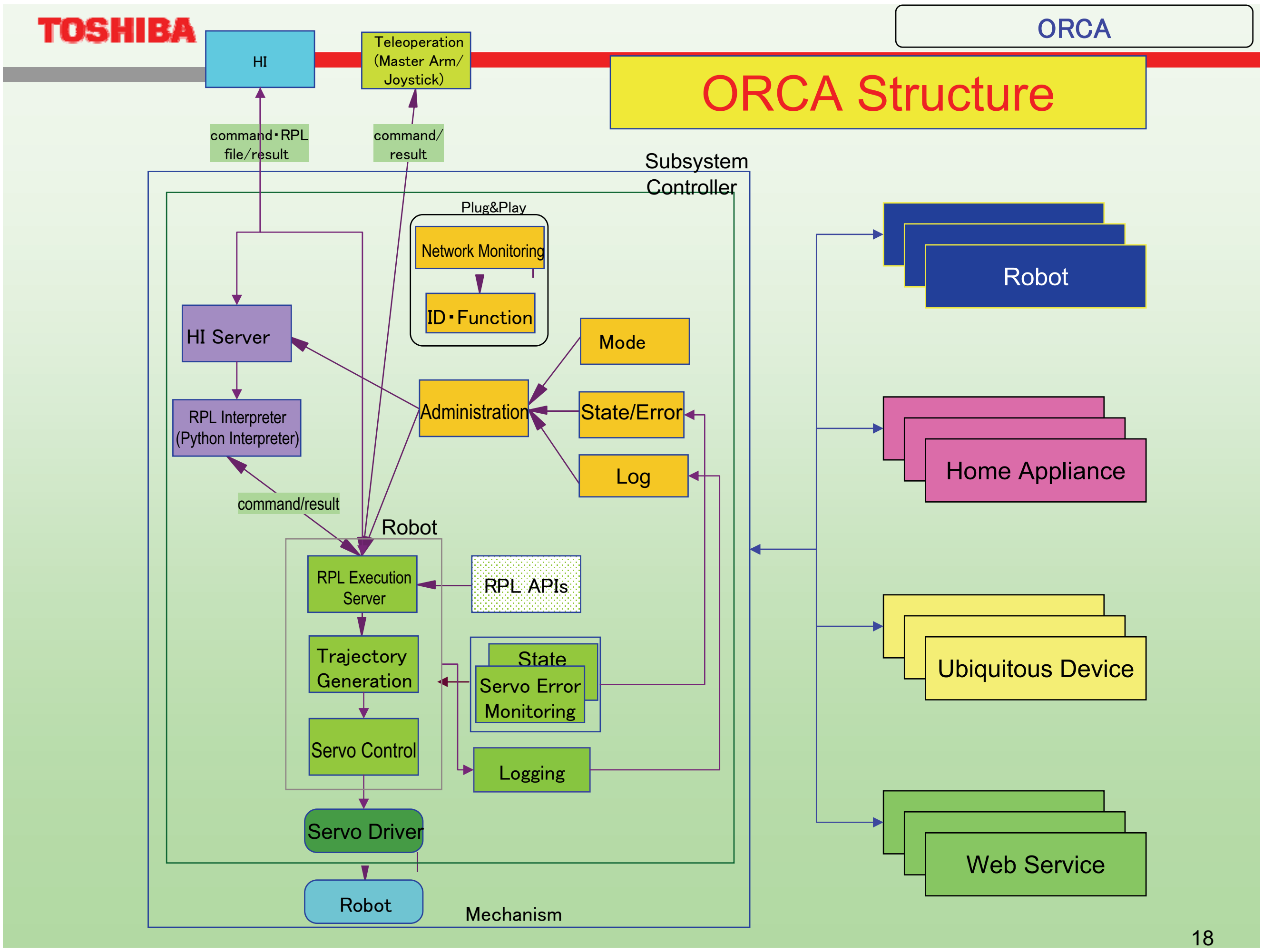
1998 - Making Software System Reusable
- Object Oriented Technology
- Using IBM-PC Compatibles
- Network(HORB, UPnP)

Lessons Learned from the Experience

How can we be more productive?

- Scripting
  - Making Many Experiments
  - Using Robots thru APIs
  - General Purpose Scripting Language -> Python
- Framework
  - Developers Can Concentrate on Parts
- Network
  - DOT -> HORB
- Other Systems Connection
  - Home Appliances -> UPnP


Where standards are needed.
Robot Software
Image Processing #1
Image Processing #2
Image Processing #3
Image Processing #4
Image Processing #n
Voice Recognition # 1
Voice Recognition # 2
Voice Recognition # 3
Voice Recognition # 4
Voice Recognition #n
Trajectory Generation#n
End-Effector Units
Servo#n
Wheel Unit
Crawler Unit
17


TOSHIBA
ORCA
ORCA Structure
HI
Teleoperation (Master Arm/ Joystick)
command・RPL file/result
command/ result
Subsystem Controller
Plug&Play
Network Monitoring
ID・Function
HI Server
Mode
Administration
State/Error
Log
RPL Interpreter (Python Interpreter)
command/result
Robot
RPL Execution Server
RPL APIs
Trajectory Generation
State
Servo Error Monitoring
Servo Control
Logging
Servo Driver
Robot
Mechanism
Robot
Home Appliance
Ubiquitous Device
Web Service
18

TOSHIBA

ORCA Idea

Using Standard IT

Concentrating on Robot Control

↓

Standard APIs

Framework

Jython/Groovy → Java → (JNI) → C++

UPnP

HORB

TOSHIBA

Experimental Robot System

HI PC

LAN

Wireless LAN Station

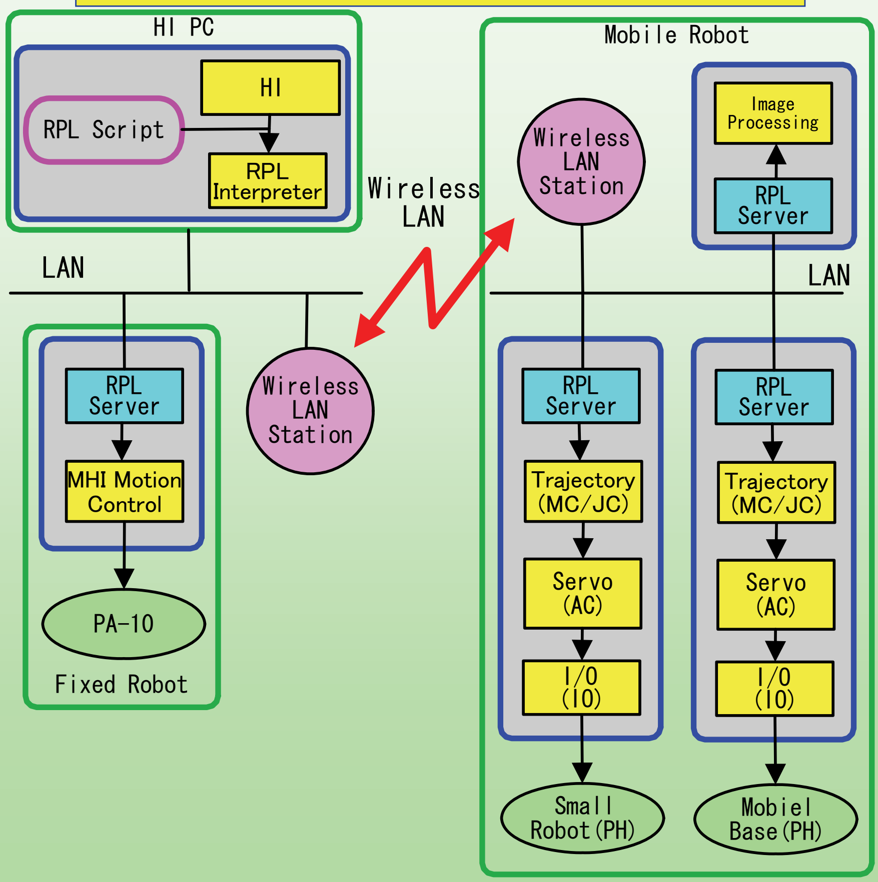
Wireless LAN

Fixed Robot

Mobile Robot

TOSHIBA

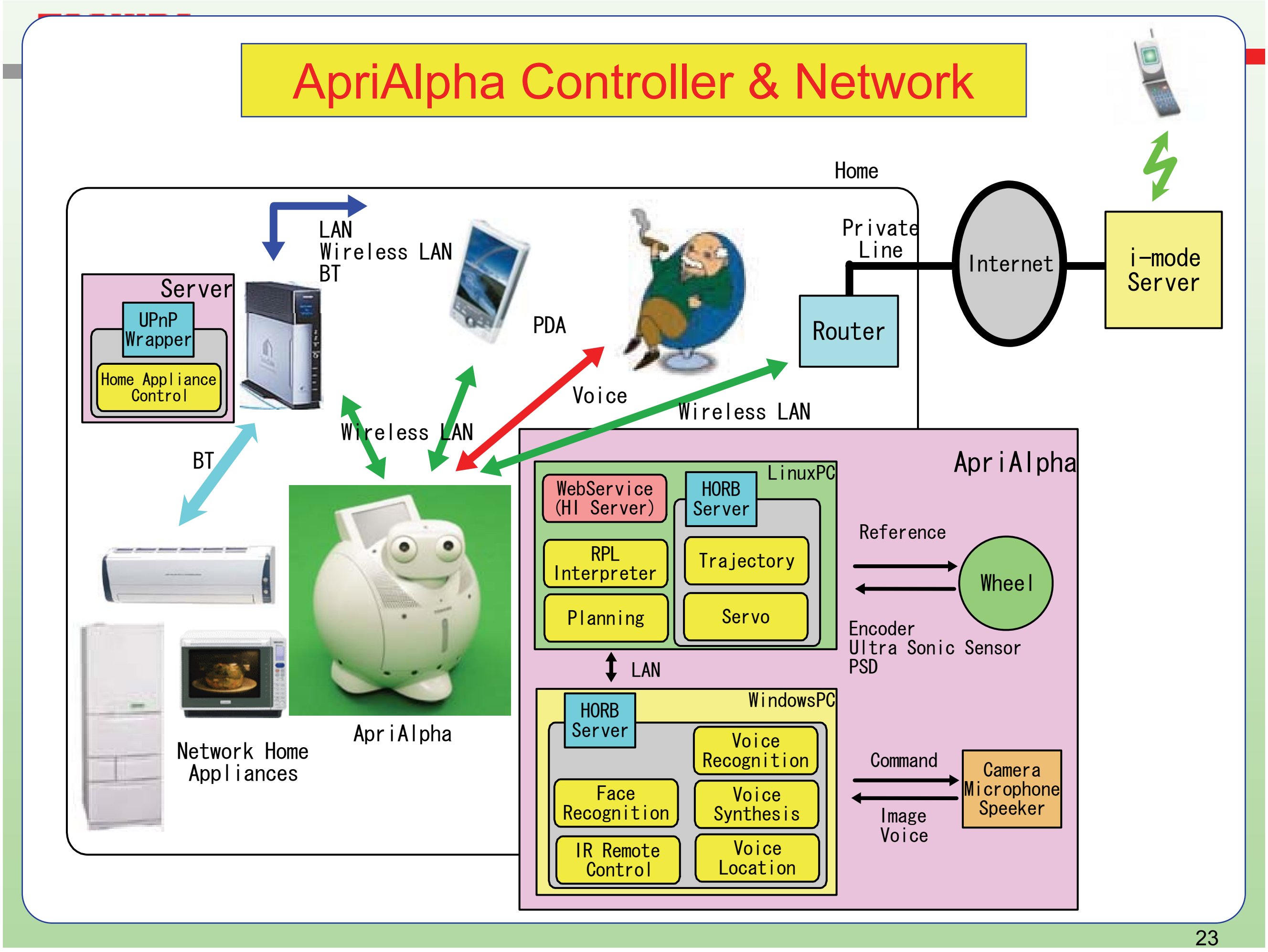
Controller Structure

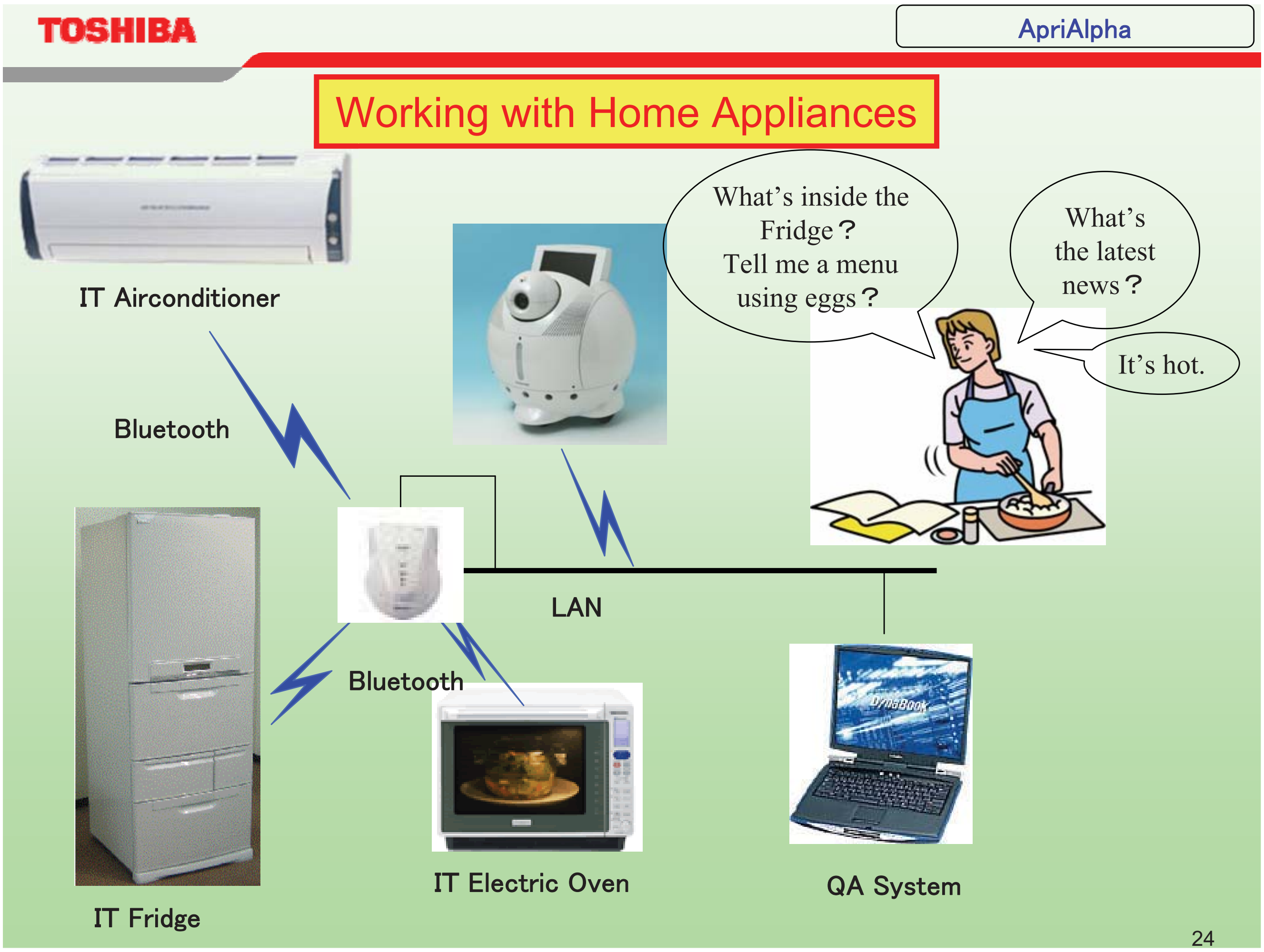


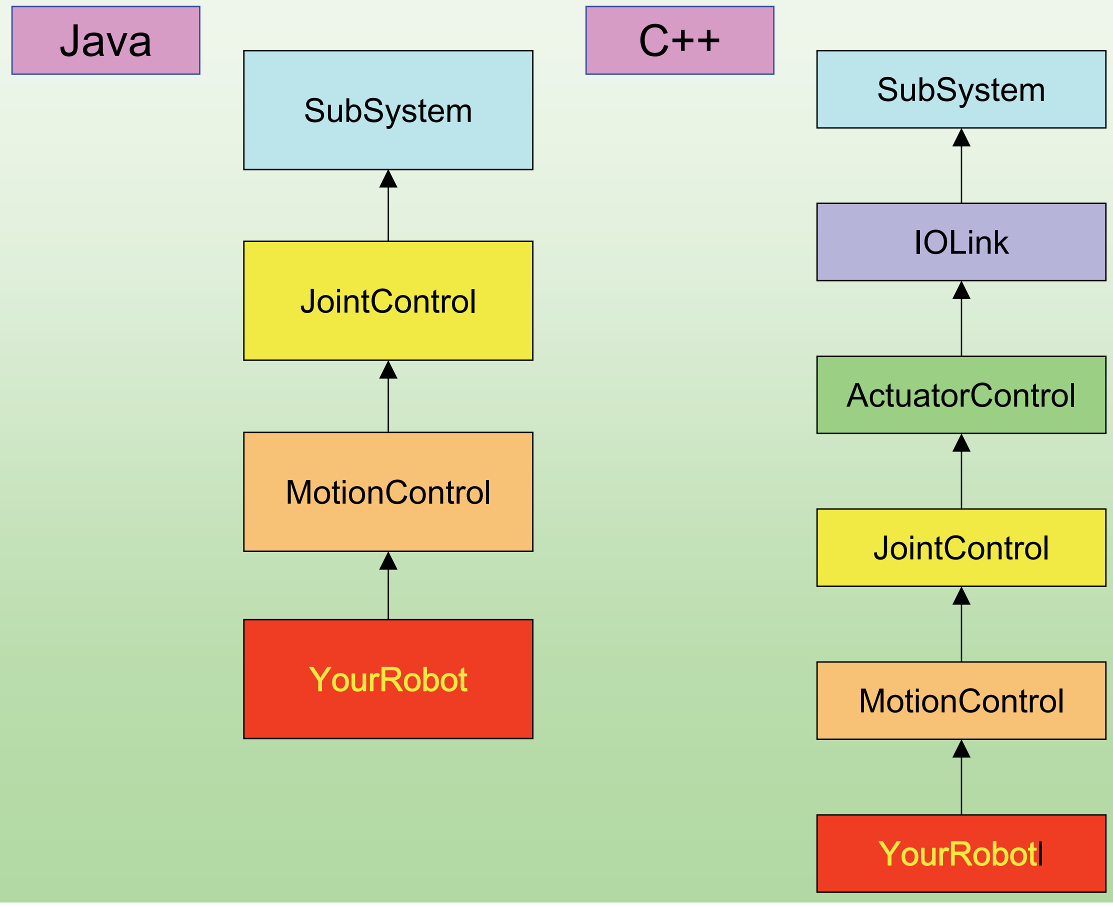
TOSHIBA

Robot Program Example

```
fixedArm = FixedArm()                        # Initialize a Fixed Arm
smallArm = SmallArm()                        # Initialize a Small Arm
vehicle = Vehicle()                          # Initialize a Vehicle
image = Image()                              # Initialize a Image System

fixedArm.parallel(true)                      #Parallel Execution
fixedArm.move(x1,y1,z1,a1,b1,c1,v1)
smallArm.move(x2,y2,z2,a2,b2,c2,v2)
vehicle.move(x3,y3,v3)
for i in range(0, 3):                        # Repeat 3 times
    vehicle.rMove(x4,y4,v4)
    ans = image.measure(obj)
    if ans < 100 :
        break
fixedArm.parallel(false)                     #Sequential Execution
```


ApriAlpha Controller & Network
Home
LAN
Wireless LAN
BT
Server
UPnP Wrapper
Home Appliance Control
PDA
Private Line
Internet
i-mode Server
Router
Voice
Wireless LAN
Wireless LAN
BT
ApriAlpha
WebService (HI Server)
HORB Server
LinuxPC
RPL Interpreter
Trajectory
Planning
Servo
Reference
Wheel
Encoder
Ultra Sonic Sensor
PSD
LAN
HORB Server
WindowsPC
Voice Recognition
Face Recognition
Voice Synthesis
IR Remote Control
Voice Location
Command
Camera
Microphone
Speeker
Image
Voice
Network Home Appliances
ApriAlpha
23

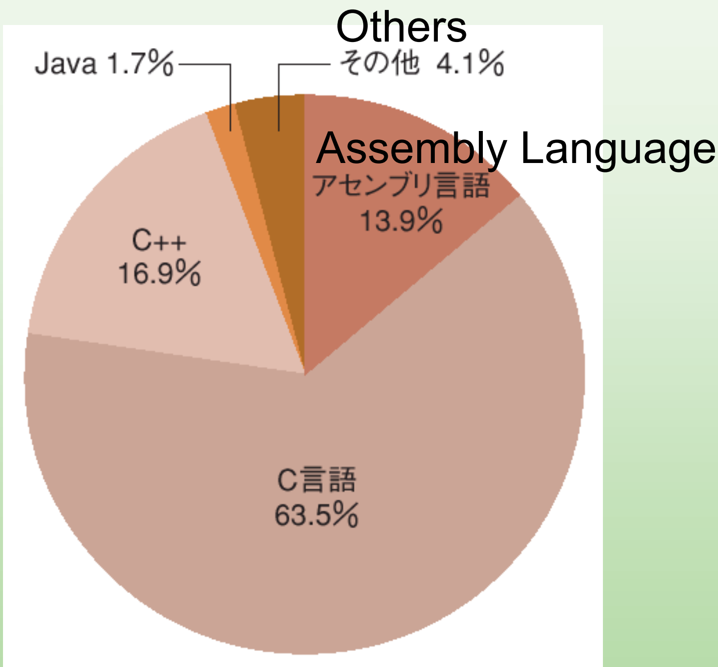

TOSHIBA
ApriAlpha
Working with Home Appliances
IT Airconditioner
What's inside the Fridge？
Tell me a menu using eggs？
What's the latest news？
It's hot.
Bluetooth
LAN
Bluetooth
IT Fridge
IT Electric Oven
QA System
24



Issues needed to be discussed

Performance vs. Standardization
- poor resources
  - for commercial robots, we cannot use rich resource such as high-end CPUs
- poor latest software knowledge of developers in embedded world

TOSHIBA



Programming Languages Used in Embedded Systems in Japan

http://itpro.nikkeibp.co.jp/members/NBY/techsquare/20050713/164540/

TOSHIBA

Issues needed to be discussed

What is the target of RT standardization?
The target robots? -> Service robots/Industrial robots/Robots in hazardous environment
The target developer
researchers, industry developers

a:The Number of Java Users
b:The Number of Java Developers
I think a>>b

a:The Number of RTC Users
b:The Number of RTC Developers
I think a>>b

Issues needed to be discussed

- API standardization
  Classification of API levels
- Framework standardization

Issues needed to be discussed

Integrated system behavior
arm with hand
mobile robot with arm

TOSHIBA

Issues needed to be discussed

RT includes

- Robot Control
- Image Processing
- Sound Processing
- Home Appliance Connection
- and many others

Communication with Other Communities

TOSHIBA

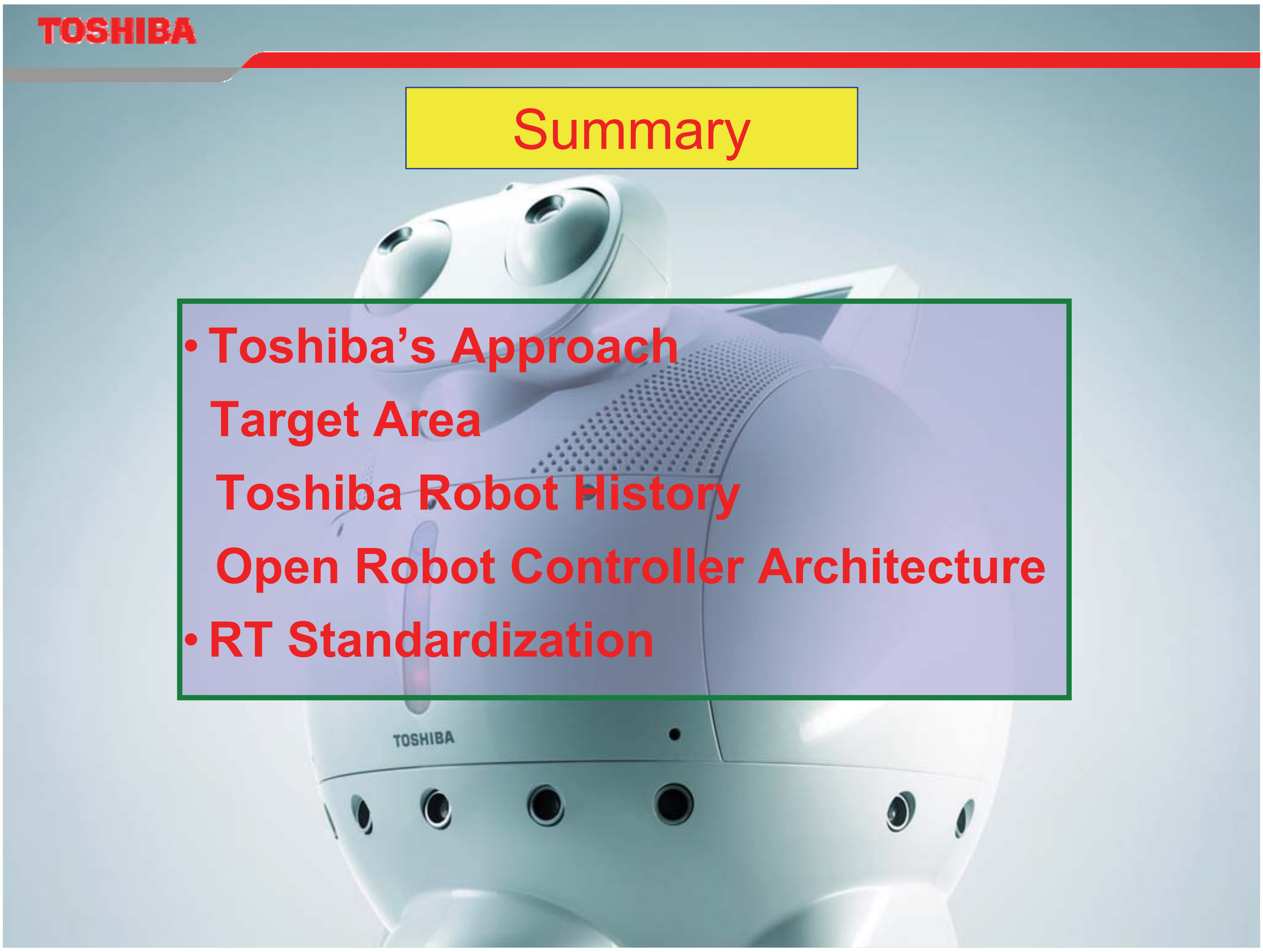
Issues needed to be discussed

Standardization

- Practical Side
- Off-the-Shelf Technologies
- Not Novel Technologies

High level technologies (eg. AI) should be developed in the future
Not for standardization

Standard should not prevent the development of new technologies


TOSHIBA
Summary
• Toshiba's Approach
Target Area
Toshiba Robot History
Open Robot Controller Architecture
• RT Standardization
TOSHIBA

OMG Robotics Task Force RFI Survey Result

Tampa (Florida – USA)

Participation

- So far 17 organizations responded :
  - Japan : 8
    - Mayekawa Manufacturing, Hitachi, IHI, SEC, ATR, AIST (2), Toshiba
  - Korea : 5
    - Samsung Electronics, ETRI, Hannool Robotics Corporation, KangWon University, Seoul National University
  - US : 2
    - RTI, SRI International
  - France : 1
    - URBI
  - India : 1
    - ADA Software Group

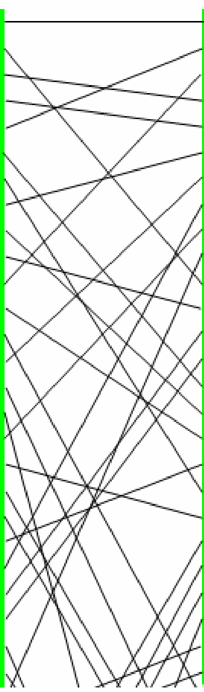
Survey Results in Brief

(as of 02/13/2006)

- Needs for standards are High
  – Especially for Robotic System Infrastructure
- Intention of Participation is High
  – Not only research centers but also businesses
- Needs match proposals rather well
  – Some unaddressed needs

Probable Activities

	Proposals	
1	Component Model / Encapsulation	30
2	Ressource Allocation / Management	27
3	Data Flow / Data Distribution	24
4	Event Management	24
5	Exectution Synchronization / Prioritization	24
6	Command Flow	21
7	Common Data Structure	21
8	Localization / Positioning	21
9	Visual Processing	21
10	Time Management	18
11	System Configuration / Dynamic Reconfiguration	18
12	Common Device Definition	18
13	Actuating Devices	18
14	Navigation	18
15	Composite Device Management	15
16	Sensing Devices	15
17	Path-Planning	15
18	Task Planning	15
19	Audio Processing	15
20	Sensor Fusion	15
21	Behavior Language	15
22	Internal Component Configuration	15
23	Modeling Language	15
24	Activity Monitoring	12
25	Physical Space Management	12
26	Data Adaptation Mechanism / Data Relation Management	12



	Needs
41	Component Model / Encapsulation
39	Exectution Synchronization / Prioritization
39	System Configuration / Dynamic Reconfiguration
38	Data Flow / Data Distribution
38	Event Management
37	Localization / Positioning
37	Navigation
36	Internal Component Configuration
35	Path-Planning
35	World Mapping
34	Ressource Allocation / Management
34	Time Management
34	Modeling Language
34	Activity Monitoring
33	Command Flow
33	Visual Processing
33	Common Device Definition
33	Behavior Language
32	Common Data Structure
32	Task Planning
32	Capability Modelling / Description / Advertisement
32	Capability Composition / Hierarchization
32	Fault Tolerances / Recovery Strategies
32	Home Indoor Service Robot
31	Object Recognition / Person Recognition
31	Object Tracking / Person Tracking

Possible Activities

27	World Mapping	12	30	Actuating Devices
28	Object Recognition / Person Recognition	12	30	Code Mobility
29	Object Tracking / Person Tracking	12	30	Capability Access Policies
30	Human Interface	12	30	Motion Control
31	Robotic Space	12	29	Composite Device Management
32	Entertainment Robot	12	29	Audio Processing
33	Code Mobility	9	29	Sensor Fusion
34	Capability Modelling / Description / Advertisement	9	29	Human Interface
35	Capability Access Policies	9	29	Application Composition
36	Capability Composition / Hierarchization	9	29	Security Management
37	Fault Tolerances / Recovery Strategies	9	28	Data Adaptation Mechanism / Data Relation Management
38	Data Semantics / Onthology	9	28	Robotic Space
39	Geometry	9	28	Entertainment Robot
40	Topology	9	28	Geometry
41	Motion Control	9	28	Safety Management / Safety Procedure / Safety Policies
42	Application Composition	9	27	Physical Space Management
43	Home Indoor Service Robot	9	27	Evaluation Metrics / Conformance Test
44	Communication Robots	9	26	Sensing Devices
45	Security Management	6	26	Data Semantics / Onthology
46	Rendering Devices	6	26	Communication Robots
47	Kinematics	6	26	Kinematics
48	Application Deployment	6	26	Application Deployment
49	Unmanned Ground Vehicules	6	26	Wheeled Indoor Robots
50	Manipulator Robots	6	25	Physical Wolrd Repository
51	Swarm Robots	6	24	Outdoor Service Robot
52	Wheeled Indoor Robots	6	22	Topology
53	Micro Robots	6	22	Energy Management
54	Outdoor Service Robot	6	21	Rendering Devices
55	Rescue Robot	6	21	Agriculture Robot

Improbable Activities

56	Safety Management / Safety Procedure / Safety Policies	3	20	Rescue Robot
57	Imagery	3	19	Manipulator Robots
58	Remote Monitoring	3	17	Unmanned Ground Vehicules
59	Simulation	3	17	Micro Robots
60	Humanoid Robots	3	17	Imagery
61	Aerial Robots	3	16	Factory Robot
62	Agriculture Robot	3	16	Outdoor Heavy Duty Robot
63	Factory Robot	3	14	Swarm Robots
64	Outdoor Heavy Duty Robot	3	14	Humanoid Robots
65	Communication Devices	0	14	Aerial Robots
66	Energy Management	0	11	Underwater Robots
67	Neural Networks	0	7	Neural Networks
68	Artificial Intelligence	0	4	Remote Monitoring *
69	Physical Wolrd Repository	0	4	Simulation *
70	Evaluation Metrics / Conformance Test	0	3	Food Processing Robots *
71	Underwater Robots	0	1	Communication Devices *
72	Food Processing Robots	0	1	Artificial Intelligence *

Possible Working Groups

- 1- Execution@Robotic Infrastructure
  - Synchronization, Event, Command Flow, Time Management
    - RTI, IHI, Toshiba, AIST , ETRI, Hanool RC, KangWon U., SRI International, ATR
  - Resource Management
- 2- Navigation@Robotic Service
  - Localization, Mapping
    - Toshiba, AIST , ETRI, Hanool RC, KangWon U., SRI International, ATR
  - Navigation, Path Planning
    - AIST,ETRI, SRI, ATR, Hanool RC, Kangwon U.
- 2 - Sensor Processing@Robotic Service
  - Audio Processing, Video Processing, Sensor Fusion
    - SEC, ADA, Toshiba, ETRI, Kangwon U., SNU, ATR, SRI
- 3 - Modeling@Robotic Tools
  - Behavior Language
    - Toshiba, ADA Software, ETRI, SRI, ATR
  - Internal Component Configuration
    - ADA Software, ASIT, Toshiba, SNU
- 4 - Data@Robotic Profiles
  - Common Data Types, Data Semantics, Data Adaptation
    - SEC, Toshiba, AIST, ETRI , SRI
- 4 - Hardware@Robotics Profiles
  - Common Device Definition, Composite Device, Actuating Devices
    - ETRI, URBI, SRI, ATR

robotics/2006-02-25

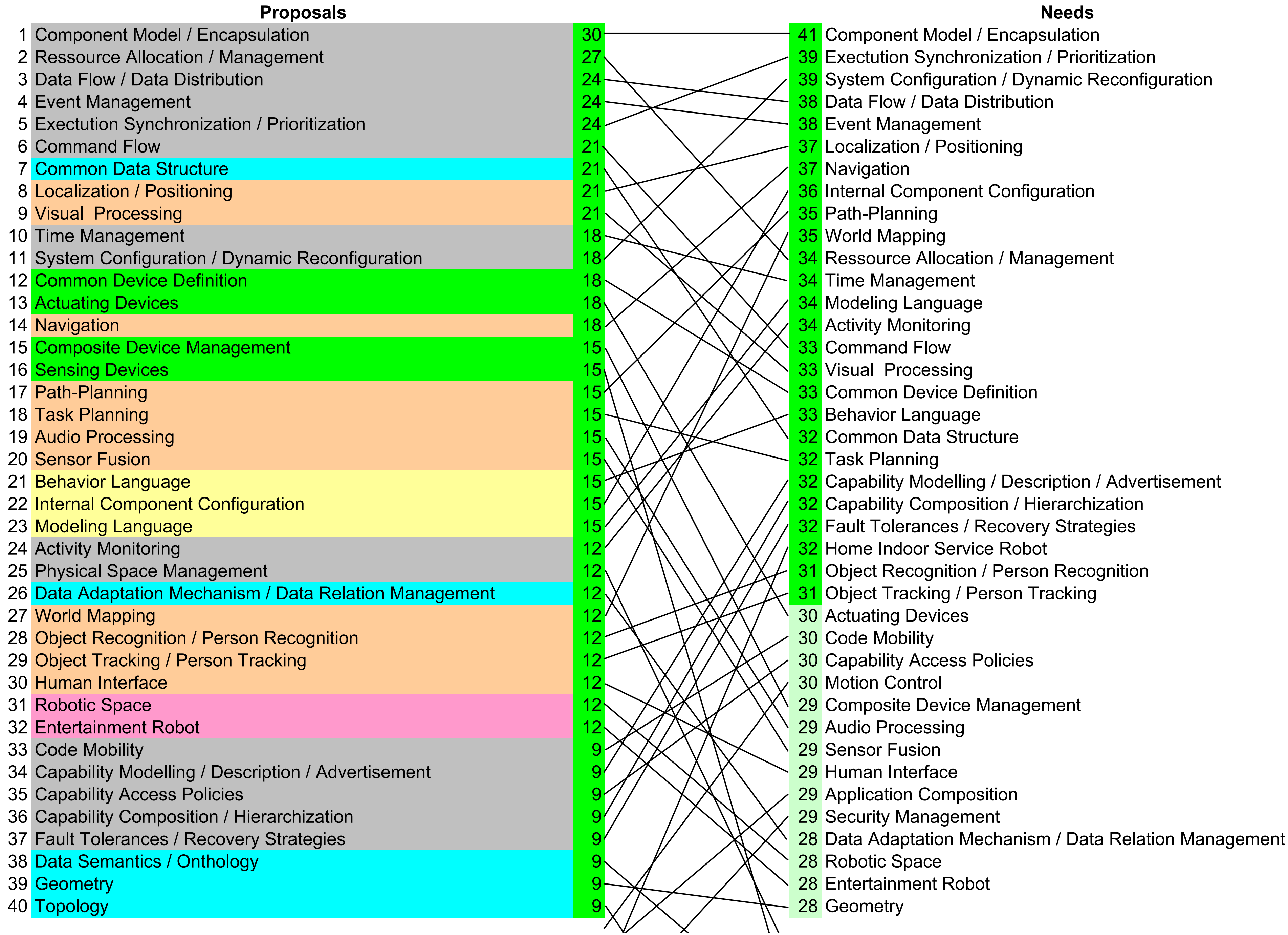
Organization	Infrastructure Layer / Middleware Layer	Component Model / Encapsulation	Data Flow / Data Distribution	Command Flow	Event Management	Security Management	Code Mobility	Activity Monitoring	Exectution Synchronization / Prioritization	Capability Modelling / Description / Advertise	Capability Access Policies	Capability Composition / Hierarchization	Ressource Allocation / Management	Safety Management / Safety Procedure / Safe	Fault Tolerances / Recovery Strategies	Physical Space Management	Time Management	System Configuration / Dynamic Reconfigura	(add here new items...)	Hardware Abstraction Lyer	Common Device Definition	Composite Device Management	Sensing Devices	Actuating Devices	Rendering Devices	Communi
	0	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	0	0	32	32	32	32	32	32
	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	0	32	32	32	32	32	32	0
	0	11	11	11	11	32	32	32	32	32	32	32	32	32	32	32	32	32	0	0	31	31	31	31	31	0
	21	21	21	21	21	21	21	11	21	21	21	21	21	21	21	21	21	21	0	32	32	32	32	32	32	0
	11	11	23	13	13	22	11	21	11	21	21	21	13	22	21	32	13	21	0	13	11	11	13	13	13	0
		11	11	32	11	23	22	11	11	22	22	11	11	22	33	33	11	22	0	0	33	33	33	33	33	0
	11	11	32	32	32	32	32	32	21	21	32	21	32	32	32	32	32	32	0	21	21	32	21	21	21	0
	12	12	12	32	22	32	32	22	12	22	32	22	12	32	22	22	22	21	0	12	12	12	12	12	12	0
	0	22	11	11	11	11	32	32	22	22	22	22	11	11	11	11	11	11	11	0	11	11	22	11	22	0
	0	21	21	21	21	32	32	21	11	21	21	32	11	32	32	11	32	21	0	0	21	21	22	21	31	0
	11	21	11	11	11	31	31	21	11	0	0	0	32	31	31	33	11	11	0	22	22	22	22	22	22	0
	11	11	21	21	21	21	33	21	11	22	22	22	11	22	22	11	21	11	0	21	21	21	21	21	22	0
	00	32	32	32	32	32	32	32	32	32	32	32	32	32	32	34	32	32	0	0	32	32	32	32	32	0
	11	11	11	0	11	2	11	11	11	11	11	11	11	1	11	11	1	11	0	11	11	11	11	11	21	0
	22	12	22	12	22	21	22	22	22	22	22	22	22	22	22	22	12	22	34	12	11	22	11	11	22	34
	11	11	11	11	11	11	11	11	11	11	11	21	11	21	11	21	11	11	11	0	11	11	11	11	0	0
	11	11	11	11	11	34	21	34	21	11	11	11	11	34	21	34	21	11	34	34	34	34	34	34	34	34

Relative Result																										
Will propose Standard	8	10	8	7	8	2	3	4	8	3	3	3	9	1	3	4	6	6	2	4	6	5	5	6	2	0
Will Influence Standard	3	5	6	4	6	6	5	7	6	10	8	9	3	7	7	5	5	7	0	3	4	4	5	4	6	0
No Opinion regarding Standard	0	2	3	5	3	8	9	6	3	3	5	4	5	8	7	8	5	4	2	3	7	8	7	7	8	3
Think it is Impractical	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A Standard will help a lot	9	12	11	9	11	7	7	9	11	8	7	8	9	6	8	7	9	11	2	3	9	7	6	8	4	0
Will use existing Standard	2	5	5	6	5	8	9	7	6	8	9	8	7	10	8	6	7	6	0	5	6	8	8	6	9	1
Do not need standard	0	0	1	1	1	1	1	0	0	0	0	0	1	0	1	2	1	0	0	1	1	1	2	2	2	0
Do not want standard	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	2	0	0	2	1	1	1	1	1	1	2
Activity Index																										
Positive Points	36	47	40	33	40	18	21	28	41	27	24	26	38	16	24	24	31	36	8	17	31	26	25	29	15	0
Negative Points	0	0	0	0	0	-2	-1	-2	0	0	0	0	0	-2	-1	-6	0	0	-4	-2	-3	-3	-3	-3	-3	-4
Need Coverage																										
Need (point)	29	41	38	33	38	29	30	34	39	32	30	32	34	28	32	27	34	39	6	14	33	29	26	30	21	1
Solution (point)	24	30	24	21	24	6	9	12	24	9	9	9	27	3	9	12	18	18	6	12	18	15	15	18	6	0
Need %	57	80	75	65	74.5	56.9	58.8	66.7	76.5	62.7	58.8	62.7	66.7	54.9	62.7	52.9	66.7	76.5	11.8	27.451	64.7	56.9	51	58.8	41.2	1.96
Solution %	47	59	47	41	47.1	11.8	17.6	23.5	47.1	17.6	17.6	17.6	52.9	5.88	17.6	23.5	35.3	35.3	11.8	23.529	35.3	29.4	29.4	35.3	11.8	0
Uncovered Needs	0	0	0	0	0	0	0	0	0	0	0	0	0	54.9	0	0	0	0	0	0	0	0	0	0	0	1.96
Uneeded Solution	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11.8	0	0	0	0	0	0	0

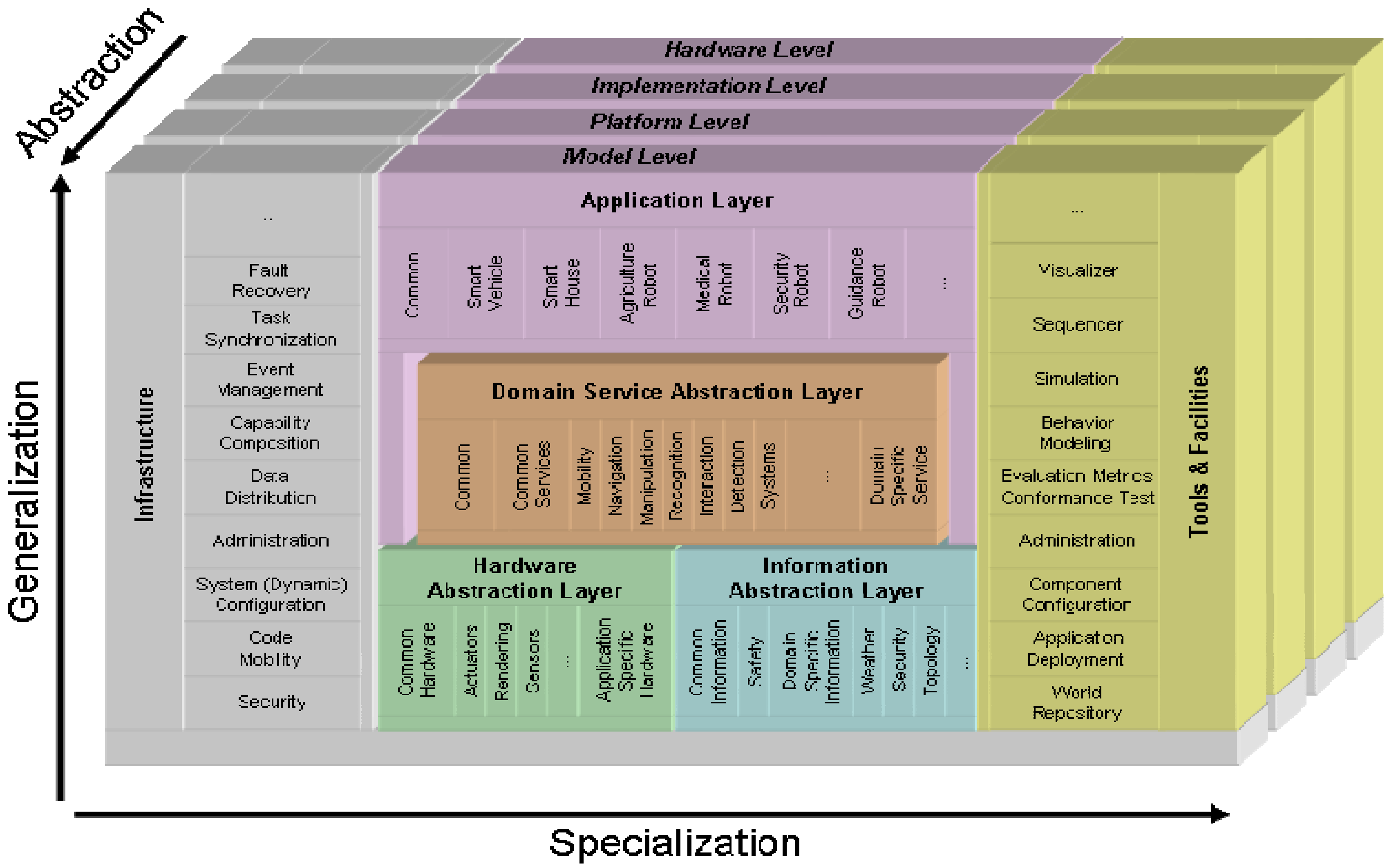
…ication Devices / Data Abstraction Layer	Common Data Structure	Data Adaptation Mechanism / Data Relation M…	Data Semantics / Onthology	Geometry	Imagery	Topology	(add here new items...)	Domain Service Layer	Navigation	Localization / Positioning	World Mapping	Path-Planning	Motion Control	Kinematics	Task Planning	Object Recognition / Person Recognition	Object Tracking / Person Tracking	Visual Processing	Audio Processing	Sensor Fusion	Human Interface	Energy Management	Neural Networks	Artificial Intelligence	Tools / Facilities Layer	Application Composition	Behavior Language	Internal Component Configuration	Application Deployment	Physical Wolrd Repository	Evaluation Metrics / Conformance Test	Modeling Language	Remote Monitoring	Simulation	Application Layer	Unmanned Ground Vehicles	Humanoid Robots	Manipulator Robots	Robotic Rob…
0	32	32	32	32	32	32	0	0	32	22	32	32	32	32	32	22	32	22	32	32	32	32	44	32	0	32	32	32	32	32	32	32	32	0	0	32	32	32	32
32	32	32	32	31	32	32	0	32	31	31	31	32	32	32	32	32	32	32	32	32	32	31	33	0	32	32	22	32	32	32	32	32	0	0	32	32	32	32	32
0	21	21	21	31	31	31	0	0	21	21	21	21	21	21	21	21	21	21	31	21	21	31	44	0	0	31	31	31	31	31	31	31	0	0	0	21	33	21	31
11	11	11	11	22	22	32	0	11	32	32	21	32	32	32	22	32	32	11	32	11	22	32	22	0	21	21	21	21	21	21	21	21	11	0	11	32	32	32	11
21	11	21	21	32	32	32	0	32	32	32	32	32	32	22	22	32	32	32	32	32	32	22	32	0	21	13	11	21	11	31	31	11	0	0	21	32	21	32	32
0	33	33	33	33	33	33	0	0	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33			22	22	22	22	33	33	22		0	0	0	0	0	0
32	32	32	32	32	32	32	0	11	21	21	32	21	21	21	21	32	32	11	32	11	21	32	32	0	11	21	11	11	32	32	32	11	0	11	11	32	32	32	32
11	11	11	12	22	22	11	0	12	11	11	11	11	32	32	12	32	22	22	12	12	12	22	32	0	22	12	32	32	32	21	21	32	0	0	21	12	32	32	32
0	11	11	22	11	22	22	0	0	11	11	11	11	11	11	11	11	11	11	11	11	11	22	22	0	0	22	11	11	22	22	22	22	0	0	0	22	11	11	22
0	21	21	21	21	32	32	0	0	21	11	21	21	21	32	31	21	21	32	32	21	21	31	32	0	0	21	32	11	32	31	32	21	0	0	0	32	32	32	11
11	11	22	11	11	44	11	0	11	11	11	11	11	11	21	11	11	11	11	11	44	22	22	33	0	21	31	11	11	32	0	32	11	0	32	44	44	44	44	44
21	11	22	22	21	32	32	0	22	22	22	22	22	21	32	21	22	22	22	21	32	22	22	43	0	21	22	21	21	22	21	22	22	0	0	22	21	32	21	21
0	32	32	32	32	32	32	0	0	21	21	31	31	31	31	31	31	31	21	21	31	21	31	44	0	0	21	31	21	31	32	31	31	0	0	0	34	33	33	22
11	11	11	21	11	11	11	0	11	11	11	32	11	32	32	11	11	11	11	11	21	11	32	43	0	0	32	11	32	32	32	32	11	0	0	21	33	33	33	21
22	22	22	22	22	22	22	0	11	11	11	11	11	11	11	11	11	11	11	22	11	22	22	22	0	22	22	22	22	22	22	22	22	0	0	11	11	22	11	11
0	0	0	0	0	0	0		0	11	11	21	21	0	0	0	21	21	11	11	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11
34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	34	11	11	34	11	11	34	21	11	34	0	34	34	34	34	34
4	7	4	3	3	1	3	0	6	6	7	4	5	3	2	5	4	4	7	5	5	4	0	0	0	2	3	5	5	2	0	0	5	1	1	3	2	1	2	4
3	3	6	7	5	4	2	0	1	5	5	5	5	4	4	5	5	5	5	3	3	8	6	3	0	6	8	5	6	5	5	6	6	0	0	4	3	2	2	4
3	6	6	6	8	10	11	1	3	6	5	8	7	9	10	6	8	8	5	9	7	5	10	8	2	1	5	6	5	9	10	10	5	2	1	2	9	11	10	7
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	5	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
6	9	7	6	7	2	4	0	5	11	11	10	10	8	6	9	8	8	9	7	8	7	4	0	0	6	7	9	10	5	6	6	9	1	1	6	3	2	4	7
3	5	7	8	7	11	10	0	4	4	4	5	5	6	8	5	7	7	6	8	5	8	10	7	1	3	8	6	6	11	7	9	7	1	1	2	8	8	7	7
0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	0	0	1	0	0	0	1	1	0	0	0	0	1	3	2	0
1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	4	1	0	0	1	0	0	1	0	0	1	0	2	3	2	2	2
21	33	25	22	21	9	15	0	24	34	37	27	30	21	16	29	25	25	35	25	26	27	10	3	0	18	23	29	31	16	11	12	30	4	4	19	12	7	12	23
-2	-3	-3	-3	-3	-6	-3	-2	-2	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-6	-3	-3	-18	-2	0	0	-2	0	0	-3	-1	0	-2	0	-5	-8	-8	-7	-5
21	32	28	26	28	17	22	0	19	37	37	35	35	30	26	32	31	31	33	29	29	29	22	7	1	21	29	33	36	26	25	27	34	4	4	20	17	14	19	28
12	21	12	9	9	3	9	0	18	18	21	12	15	9	6	15	12	12	21	15	15	12	0	0	0	6	9	15	15	6	0	0	15	3	3	9	6	3	6	12
41.176	62.7	54.9	51	54.9	33.3	43.1	0	37	73	73	69	69	59	51	63	61	61	65	57	57	57	43	14	2	41	57	65	71	51	49	53	67	7.8	8	39.2	33	27	37	55
23.529	41.2	23.5	17.6	17.6	5.88	17.6	0	35	35	41	24	29	18	12	29	24	24	41	29	29	24	0	0	0	12	18	29	29	12	0	0	29	5.9	6	17.6	12	5.9	12	24
0	0	0	0	0	33.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	43	14	2	0	0	0	0	0	49	53	0	7.8	8	0	0	27	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5.9	6	0	0	0	0	0

…obots
…tic Space

Swarm Robots	Underwater Robots	Aerial Robots	Wheeled Indoor Robots	Micro Robots	Agriculture Robot	Home Indoor Service Robot	Factory Robot	Outdoor Service Robot	Outdoor Heavy Duty Robot	Rescue Robot	Communication Robots	Entertainment Robot	Food Processin…	0
32	32	32	32	32	21	32	32	32	32	32	32	32	21	
32	32	32	32	32	32	32	32	32	32	32	32	32	0	
33	31	31	21	33	31	33	21	21	21	21	31	31	0	
32	32	32	32	32	32	11	32	11	32	21	11	11	0	
32	32	32	21	32	32	21	32	32	32	32	21	11	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	
32	32	11	32	32	11	21	11	21	21	32	32	21	0	
12	32	22	21	22	31	21	32	32	11	12	22	22	0	
22	22	22	11	11	22	11	22	22	22	22	11	11	0	
21	33	32	21	32	32	21	32	32	32	32	32	32	0	
44	44	44	44	44	44	44	44	44	44	44	44	44	44	
22	22	22	22	32	21	21	21	21	22	22	21	21	0	
33	34	34	21	21	34	21	34	21	34	21	0	33	0	
33	33	33	33	33	33	21	33	33	33	33	21	21	0	
34	44	44	11	11	22	11	44	11	44	11	11	11	0	
11	0	0	0	0	0	0	0	0	0	0	0	0	0	
34	34	34	34	34	34	34	34	34	34	34	34	34	34	
2	0	1	2	2	1	3	1	2	1	2	3	4	0	0
3	2	3	6	2	4	7	3	5	4	5	4	4	1	
10	11	9	6	10	9	4	9	7	8	7	6	6	1	
1	2	2	1	1	1	1	2	1	2	1	1	1	1	
2	1	2	7	3	5	10	3	6	3	4	7	8	1	
8	8	8	5	8	6	2	7	6	7	8	5	4	0	
3	2	1	1	2	1	1	1	1	1	1	0	1	0	
3	4	4	2	2	3	2	4	2	4	2	2	2	2	
11	3	8	19	11	12	26	9	17	10	15	20	24	2	
-10	-12	-11	-6	-7	-8	-6	-11	-6	-11	-6	-5	-6	-5	
14	11	14	26	17	21	32	16	24	16	20	26	28	3	
6	0	3	6	6	3	9	3	6	3	6	9	12	0	
27	22	27	51	33	41	63	31	47	31	39	51	55	5.9	
12	0	5.9	12	12	5.9	18	5.9	12	5.9	12	18	24	0	
0	22	27	0	0	41	0	31	0	31	0	0	0	5.9	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	


Proposals
1 Component Model / Encapsulation 30
2 Ressource Allocation / Management 27
3 Data Flow / Data Distribution 24
4 Event Management 24
5 Exectution Synchronization / Prioritization 24
6 Command Flow 21
7 Common Data Structure 21
8 Localization / Positioning 21
9 Visual Processing 21
10 Time Management 18
11 System Configuration / Dynamic Reconfiguration 18
12 Common Device Definition 18
13 Actuating Devices 18
14 Navigation 18
15 Composite Device Management 15
16 Sensing Devices 15
17 Path-Planning 15
18 Task Planning 15
19 Audio Processing 15
20 Sensor Fusion 15
21 Behavior Language 15
22 Internal Component Configuration 15
23 Modeling Language 15
24 Activity Monitoring 12
25 Physical Space Management 12
26 Data Adaptation Mechanism / Data Relation Management 12
27 World Mapping 12
28 Object Recognition / Person Recognition 12
29 Object Tracking / Person Tracking 12
30 Human Interface 12
31 Robotic Space 12
32 Entertainment Robot 12
33 Code Mobility 9
34 Capability Modelling / Description / Advertisement 9
35 Capability Access Policies 9
36 Capability Composition / Hierarchization 9
37 Fault Tolerances / Recovery Strategies 9
38 Data Semantics / Onthology 9
39 Geometry 9
40 Topology 9
Needs
41 Component Model / Encapsulation
39 Exectution Synchronization / Prioritization
39 System Configuration / Dynamic Reconfiguration
38 Data Flow / Data Distribution
38 Event Management
37 Localization / Positioning
37 Navigation
36 Internal Component Configuration
35 Path-Planning
35 World Mapping
34 Ressource Allocation / Management
34 Time Management
34 Modeling Language
34 Activity Monitoring
33 Command Flow
33 Visual Processing
33 Common Device Definition
33 Behavior Language
32 Common Data Structure
32 Task Planning
32 Capability Modelling / Description / Advertisement
32 Capability Composition / Hierarchization
32 Fault Tolerances / Recovery Strategies
32 Home Indoor Service Robot
31 Object Recognition / Person Recognition
31 Object Tracking / Person Tracking
30 Actuating Devices
30 Code Mobility
30 Capability Access Policies
30 Motion Control
29 Composite Device Management
29 Audio Processing
29 Sensor Fusion
29 Human Interface
29 Application Composition
29 Security Management
28 Data Adaptation Mechanism / Data Relation Management
28 Robotic Space
28 Entertainment Robot
28 Geometry

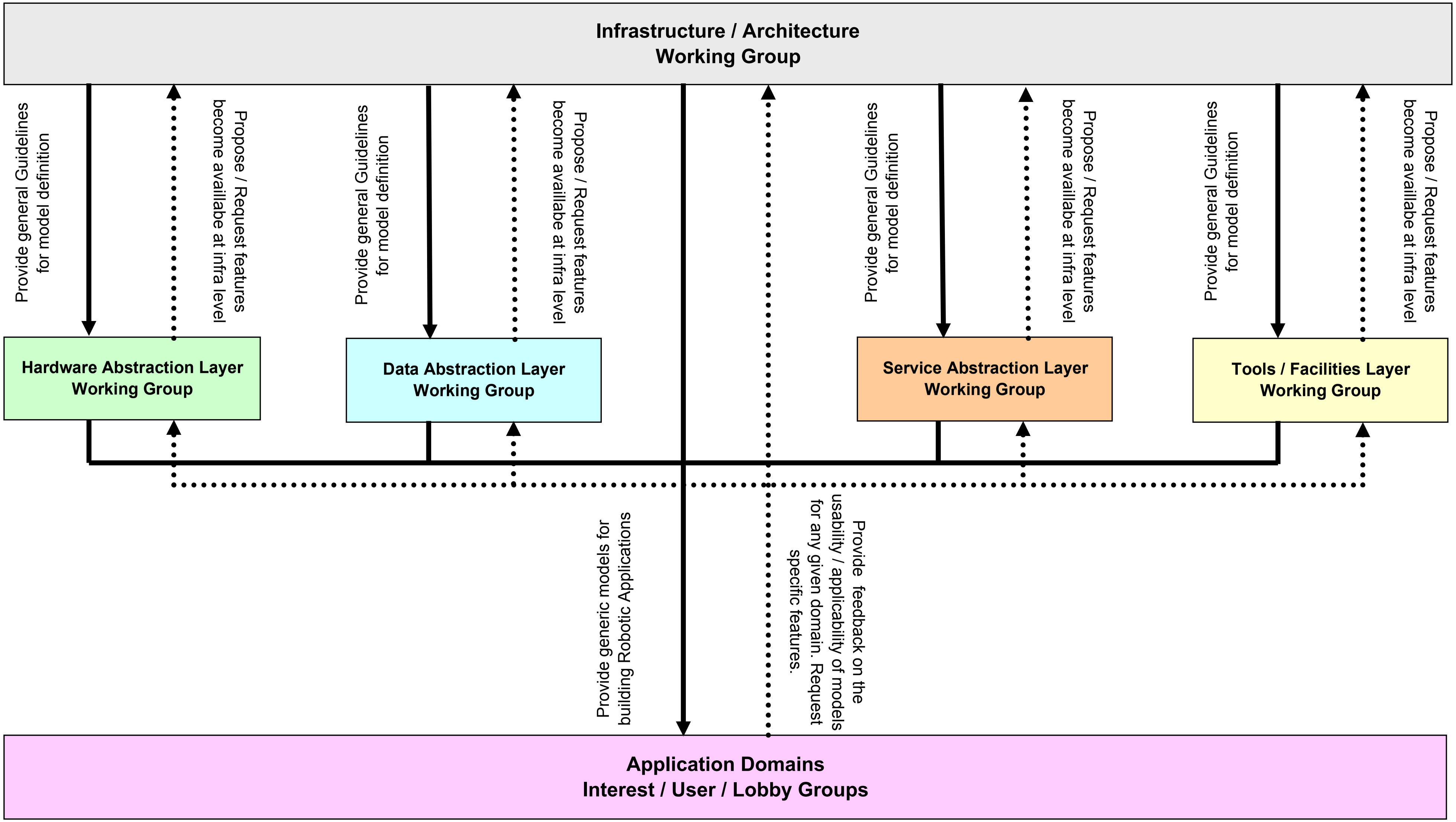
41 Motion Control 9
42 Application Composition 9
43 Home Indoor Service Robot 9
44 Communication Robots 9
45 Security Management 6
46 Rendering Devices 6
47 Kinematics 6
48 Application Deployment 6
49 Unmanned Ground Vehicules 6
50 Manipulator Robots 6
51 Swarm Robots 6
52 Wheeled Indoor Robots 6
53 Micro Robots 6
54 Outdoor Service Robot 6
55 Rescue Robot 6
56 Safety Management / Safety Procedure / Safety Policies 3
57 Imagery 3
58 Remote Monitoring 3
59 Simulation 3
60 Humanoid Robots 3
61 Aerial Robots 3
62 Agriculture Robot 3
63 Factory Robot 3
64 Outdoor Heavy Duty Robot 3
65 Communication Devices 0
66 Energy Management 0
67 Neural Networks 0
68 Artificial Intelligence 0
69 Physical Wolrd Repository 0
70 Evaluation Metrics / Conformance Test 0
71 Underwater Robots 0
72 Food Processing Robots 0
28 Safety Management / Safety Procedure / Safety Policies
27 Physical Space Management
27 Evaluation Metrics / Conformance Test
26 Sensing Devices
26 Data Semantics / Onthology
26 Communication Robots
26 Kinematics
26 Application Deployment
26 Wheeled Indoor Robots
25 Physical Wolrd Repository
24 Outdoor Service Robot
22 Topology
22 Energy Management
21 Rendering Devices
21 Agriculture Robot
20 Rescue Robot
19 Manipulator Robots
17 Unmanned Ground Vehicules
17 Micro Robots
17 Imagery
16 Factory Robot
16 Outdoor Heavy Duty Robot
14 Swarm Robots
14 Humanoid Robots
14 Aerial Robots
11 Underwater Robots
7 Neural Networks
4 Remote Monitoring *
4 Simulation *
3 Food Processing Robots *
1 Communication Devices *
1 Artificial Intelligence *


Abstraction
Generalization
Specialization
Hardware Level
Implementation Level
Platform Level
Model Level
Infrastructure
...
Fault Recovery
Task Synchronization
Event Management
Capability Composition
Data Distribution
Administration
System (Dynamic) Configuration
Code Mobility
Security
Application Layer
Common
Smart Vehicle
Smart House
Agriculture Robot
Medical Robot
Security Robot
Guidance Robot
...
Domain Service Abstraction Layer
Common
Common Services
Mobility
Navigation
Manipulation
Recognition
Interaction
Detection
Systems
...
Domain Specific Service
Hardware Abstraction Layer
Common Hardware
Actuators
Rendering
Sensors
...
Application Specific Hardware
Information Abstraction Layer
Common Information
Safety
Domain Specific Information
Weather
Security
Topology
...
Tools & Facilities
...
Visualizer
Sequencer
Simulation
Behavior Modeling
Evaluation Metrics Conformance Test
Administration
Component Configuration
Application Deployment
World Repository

Code	Infrastructure Layer / Middleware Layer	Related Items
IM1	Component Model / Encapsulation	
IM2	Data Flow / Data Distribution	
IM3	Command Flow	
IM4	Event Management	
IM5	Security Management	
IM6	Code Mobility	
IM7	Activity Monitoring	
IM8	Exectution Synchronization / Prioritization	
IM9	Capability Modelling / Description / Advertisement	
IM10	Capability Access Policies	
IM11	Capability Composition / Hierarchization	
IM12	Ressource Allocation / Management	
IM13	Safety Management / Safety Procedure / Safety Policies	
IM14	Fault Tolerances / Recovery Strategies	
IM15	Physical Space Management	
IM16	Time Management	
IM17	System Configuration / Dynamic Reconfiguration	
	(add here new items...)	
Code	Hardware Abstraction Lyer	
HAL1	Common Device Definition	
HAL2	Composite Device Management	
HAL3	Sensing Devices	
HAL4	Actuating Devices	
HAL5	Rendering Devices	
HAL6	Communication Devices	
Code	Data Abstraction Layer	
DAL1	Common Data Structure	
DAL2	Data Adaptation Mechanism / Data Relation Management	
DAL3	Data Semantics / Onthology	
DAL4	Geometry	
DAL5	Imagery	
DAL6	Topology	
	(add here new items...)	
Code	Domain Service Layer	
DS1	Navigation	
DS2	Localization / Positioning	
DS3	World Mapping	
DS4	Path-Planning	
DS5	Motion Control	
DS6	Kinematics	
DS7	Task Planning	
DS8	Object Recognition / Person Recognition	
DS9	Object Tracking / Person Tracking	
DS10	Visual Processing	
DS11	Audio Processing	
DS12	Sensor Fusion	
DS13	Human Interface	
DS14	Energy Management	
DS15	Neural Networks	
DS16	Artificial Intelligence	

Code	Tools / Facilities Layer	
TF1	Application Composition	
TF2	Behavior Language	
TF3	Internal Component Configuration	
TF4	Application Deployment	
TF5	Physical Wolrd Repository	
TF6	Evaluation Metrics / Conformance Test	
TF7	Modeling Language	
TF8	Remote Monitoring	
TF9	Simulation	

Code	Application Layer	
DA1	Unmanned Ground Vehicules	
DA2	Humanoid Robots	
DA3	Manipulator Robots	
DA4	Robotic Space	
DA5	Swarm Robots	
DA6	Underwater Robots	
DA7	Aerial Robots	
DA8	Wheeled Indoor Robots	
DA9	Micro Robots	
DA10	Agriculture Robot	
DA11	Home Indoor Service Robot	
DA12	Factory Robot	
DA13	Outdoor Service Robot	
DA14	Outdoor Heavy Duty Robot	
DA15	Rescue Robot	
DA16	Communication Robots	
DA17	Entertainment Robot	
DA18	Food Processing Robots	
	(add here new items...)	



Infrastructure / Architecture Working Group
Provide general Guidelines for model definition
Propose / Request features become availlabe at infra level
Hardware Abstraction Layer Working Group
Provide general Guidelines for model definition
Propose / Request features become availlabe at infra level
Data Abstraction Layer Working Group
Provide general Guidelines for model definition
Propose / Request features become availlabe at infra level
Service Abstraction Layer Working Group
Provide general Guidelines for model definition
Propose / Request features become availlabe at infra level
Tools / Facilities Layer Working Group
Provide generic models for building Robotic Applications
Provide feedback on the usability / applicability of models for any given domain. Request specific features.
Application Domains Interest / User / Lobby Groups

OMG Robotics DTF
Publicity Drive

Robotics DTF

The need for publicity…

- OMG is "Object Management Group" and not "Oh My God!"
- People know about UML, CORBA but not OMG!!
- OMG = Standards for Computer Industry… but for Robotics…???

Target Organisations…

- Mainly in Japan, EU, India & SE Asia
- Many research organisations with their own proprietary standards
- Lots to learn from… and collaborate

Points to stress on…

- Success of OMG in standardizing UML, CORBA and other success stories
- Clear description of the OMG targets to standardize robotics
  - Focus, targets, time-plan and current status
- Publicize the member list to assert the seriousness of the drive

robotics2006-02-26

OMG Robotics DTF
Publicity Drive

Robotics DTF

Publicity Sub Committee (as of now)

- ADA Software Group
- AIST
- ETRI
- NEDO

Immediate Goals…

- 4 Page Flyer following the guidelines
- A more descriptive web page for the OMG Robotics DTF
- …

Contact..

- Abheek Bose (ADA Software): abheek@adasoftware.com
- Olivier Lemaire (AIST): olivier.lemaire@aist.go.jp
- Yokomachi (NEDO): yokomachimsy@nedo.go.jp

Summary of Activity
Robotics TF – Tampa TM

Mars/2006-02-19

Responses to the RFI (so far)

- We received another 11 formal responses to the RFI from :
  - **Real-Time Innovation (RTI)**
  - **Seoul National University (SNU)**
  - **Electronics and Telecommunications Research Institute (ETRI) x 2**
  - **National Agricultural Research Center (Narc)**
  - **COMPARE consortium**
  - **SEC Co., Ltd.**
  - **Hitachi**
  - **Electronics and Telecommunications Research Institute (ETRI) + 3**
  - **Network Robot Forum**
  - **Ishikawajima-Harima Heavy Industries Co.,Ltd.**
  - **ADA Software Group response to the Robotics RFI.pdf**
  - **NEC**
- Altogether : 15 responses (still expecting some)
- One last extension of the deadline to respond to the RFI :

April 3rd, 2006

Responses to the RFI (so far)

- We had 14presentations in regard to the RFI
  - Hitachi's needs for robotic system standards- Saku Egawa (Hitachi)
  - Towards Plug and Play Robotics- Abheek Kumar Bose (ADA Software Group)
  - SEC's approach to the standardization of robotics systems- Hiroyuki Nakamoto and Masayuki Nagase (SEC)
  - Development of Food Robots and Meat Processing Robots, and Request for Standardization of RTC - Tomoki Yamashita (Maekawa MFG)
  - RT service framework using IT infrastructure - Wonpil Yu (ETRI)
  - A mobile robot software system architecture with unified sensory data integration - Takashi Tsubouchi (Tsukuba Univ.)
  - Navigation of mobile robots including mapping, localization, and motion - Nakju Doh (ETRI)
  - Response from AIST- Olivier Lemaire (AIST)
  - Current State of Robotics Script/Control Language Standards- Lloyd Spencer (CoroWare)
  - Development Framework for Mobile Robot based on JAUS and RTMiddleware - Wataru Inamura (IHI)
  - An overview of PIM & PSM for SWRadio Components specification - Jerry Bickle (Prismtech)
  - Response from Compare Project - Virginie Watine (THALES)
  - Robot Server Middleware: CAMU - Yun Koo Chung (ETRI)
  - Toshiba's approach to RT standardization and where the standardization is needed- Fumio Ozaki (Toshiba)

Survey regarding Standardization in Robotics Technology

- We issued a survey aiming at precisely identifying :
  - The needs for standardization
  - The organization potentially willing to propose technology
- After listing the most common robotics domain specific functionalities (around 70 items), organizations were asked, for each item, to give a mark according to their opinion regarding its standardization

Participation

- So far 17 organizations responded :
  - Japan : 8
    - Mayekawa Manufacturing, Hitachi, IHI, SEC, ATR, AIST (2), Toshiba
  - Korea : 5
    - Samsung Electronics, ETRI, Hannool Robotics Corporation, KangWon University, Seoul National University
  - US : 2
    - RTI, SRI International
  - France : 1
    - URBI
  - India : 1
    - ADA Software Group

Survey Results in Brief

(as of 02/13/2006)

- Needs for standards are High
  - Especially for Robotic System Infrastructure
- Intention of Participation is High
  - Not only research centers but also businesses
- Needs match proposals rather well
  - Some unaddressed needs

Probable Activities

	Proposals	
1	Component Model / Encapsulation	30
2	Ressource Allocation / Management	27
3	Data Flow / Data Distribution	24
4	Event Management	24
5	Exectution Synchronization / Prioritization	24
6	Command Flow	21
7	Common Data Structure	21
8	Localization / Positioning	21
9	Visual Processing	21
10	Time Management	18
11	System Configuration / Dynamic Reconfiguration	18
12	Common Device Definition	18
13	Actuating Devices	18
14	Navigation	18
15	Composite Device Management	15
16	Sensing Devices	15
17	Path-Planning	15
18	Task Planning	15
19	Audio Processing	15
20	Sensor Fusion	15
21	Behavior Language	15
22	Internal Component Configuration	15
23	Modeling Language	15
24	Activity Monitoring	12
25	Physical Space Management	12
26	Data Adaptation Mechanism / Data Relation Management	12

	Needs
41	Component Model / Encapsulation
39	Exectution Synchronization / Prioritization
39	System Configuration / Dynamic Reconfiguration
38	Data Flow / Data Distribution
38	Event Management
37	Localization / Positioning
37	Navigation
36	Internal Component Configuration
35	Path-Planning
35	World Mapping
34	Ressource Allocation / Management
34	Time Management
34	Modeling Language
34	Activity Monitoring
33	Command Flow
33	Visual Processing
33	Common Device Definition
33	Behavior Language
32	Common Data Structure
32	Task Planning
32	Capability Modelling / Description / Advertisement
32	Capability Composition / Hierarchization
32	Fault Tolerances / Recovery Strategies
32	Home Indoor Service Robot
31	Object Recognition / Person Recognition
31	Object Tracking / Person Tracking

Possible Working Groups

- 1- Execution@Robotic Infrastructure
  - Synchronization, Event, Command Flow, Time Management
    - RTI, IHI, Toshiba, AIST , ETRI, Hanool RC, KangWon U., SRI International, ATR
  - Resource Management
- 2- Navigation@Robotic Service
  - Localization, Mapping
    - Toshiba, AIST , ETRI, Hanool RC, KangWon U., SRI International, ATR
  - Navigation, Path Planning
    - AIST,ETRI, SRI, ATR, Hanool RC, Kangwon U.
- 2 - Sensor Processing@Robotic Service
  - Audio Processing, Video Processing, Sensor Fusion
    - SEC, ADA, Toshiba, ETRI, Kangwon U., SNU, ATR, SRI
- 3 - Modeling@Robotic Tools
  - Behavior Language
    - Toshiba, ADA Software, ETRI, SRI, ATR
  - Internal Component Configuration
    - ADA Software, ASIT, Toshiba, SNU
- 4 - Data@Robotic Profiles
  - Common Data Types, Data Semantics, Data Adaptation
    - SEC, Toshiba, AIST, ETRI , SRI
- 4 - Hardware@Robotics Profiles
  - Common Device Definition, Composite Device, Actuating Devices
    - ETRI, URBI, SRI, ATR

New Working Groups

- The following working groups have been started :
  - Robotic System Infrastructure
  - Robotic Domain Profiles
  - Robotic Services
  - Tools for Robotic Technology
- Founding chairs have been designated

What's next

- Founding Chairs are in charge to :
  - Write a mission statement for their group
  - Create sub working groups according to the result of the survey
- Official start of working group and election of chairs during next meeting in St Louis

Robotics-DTF

Date: Friday, 17th February, 2006
Chair: Tetsuo Kotoku, YunKoo Chung, Hung Pham
Group URL: http://robotics.omg.org/
Group email: robotics@omg.org

➢Highlights from this Meeting:

Robotics/SDO Joint Plenary (Tue. & Wed.):

- **14 RFI response presentations**
  (Hitachi, ADA Software, SEC, Mayekawa MFG, ETRI*3, Tsukuba Univ., AIST, Coroware, IHI, PrismTech, THALES, Toshiba)
- **1 Special Talk**
  - Matt Long (CRASAR, Univ. of South Florida) [robotics/06-02-18]

Joint Meeting with MARS-PTF (Thu.):

- RFI Summary report

Robotics-DTF

Date: Friday, 17th February, 2006
Chair: Tetsuo Kotoku, YunKoo Chung, Hung Pham
Group URL: http://robotics.omg.org/
Group email: robotics@omg.org

➢Deliverables from this Meeting:

- **Robotic Systems RFI** [mars2005-06-12] **deadline re-extension** (until April. 3rd 2006, 3 weeks before St. Louis Mtg.)
- **4 Potential WGs** (will be chartered at St. Louis Mtg.)

➢Future deliverables (In-Process):

- **RFI responses**

➢Next Meeting (St. Louis, MO, USA):

- **RTCs RFP revised proposal progress report**
  (joint with MARS-PTF, SDO-DSIG)
- **Robotic Systems RFI response presentation**
- **Roadmap discussion**
- **Contact report**