Robotics Domain Task Force Final Agenda ver1.0.2 **robotics/2010-06-01**

OMG Technical Meeting - **Minneapolis, MN, USA** -- June 21-25, 2010

http://robotics.omg.org/

		TF/SIG				
		Host	**Joint (Invited)**	**Agenda Item**	**Purpose**	**Room**
Sunday: WG activites(pm)						
13:00	17:00			Robotics DDC and RoIS RFP submitter's meeting	Arrangement	Lake Calhoun, 5th FL
Monday: Robotics Plenary(am) and WG activites(pm)						
8:30	9:00	Robotics		Robotics-DTF Plenary Opening Session	Robotics plenary openning	Lake Superior B, 5th FL
10:00	10:30	MARS		Dynamic Deployment and Configuration (DDC) for RTC RFP 3rdReview - Noriaki Ando (AIST) and Jae-Yeong Jung(ETRI)	3rd Review Joint with **MARS**	Lake Superior A, 5th FL
10:30	11:00	Robotics		**Special Talk:** Future of the Deployment and Configuration Specification - **William R. Otte** (Vanderbilt Univ.)	discussion	Lake Superior B, 5th FL
11:00	12:00	Robotics		Robotic Interaction Service (RoIS) Framework RFP 2nd Review - Su-Young Chi(ETRI), Miki Sato(ATR) and Toshio Hori(AIST)	2nd Review	
12:00	13:00					Nicollet D, 1st FL
13:00	18:00			Architecture Board Plenary		Olsen 3rd FL
13:00	17:00			Robotic Infrastructure WG (4h) - Noriaki Ando(AIST) and Seung-Woog Jung(ETRI)	discussion	Prior Lake, 5th FL
				Robotic Functional Services WG(4h): - Su-Young Chi , Miki Sato and Toshio Hori	discussion	Lake of the Isles, 5th FL
Tuesday: WG activities (am) and Robotics Plenary (pm)						
9:00	12:00			Models in Robotics Meeting (3h) - Laurent Rioux (Thales)	discussion	Lake Calhoun, 5th FL
				Robotic Functional Services WG(3h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	Lake of the Isles, 5th FL
12:00	13:00			**LUNCH**		Nicollet D, 1st FL
15:30	16:00	Robotics		Models in Robotics - Laurent Rioux (Thales)	charter new WG	Lake Calhoun, 5th FL
16:00	16:40	Robotics		WG Reports and Discussion (Service WG, Profile WG, Robotic Localization Service WG)	presentation and discussion	
16:40	17:00	Robotics		Robotics-DTF Plenary Wrap-up Session (Roadmap and Next meeting Agenda)	Robotics plenary closing	
17:30	18:00			Robotics WG Co-chairs Planning Session (Preliminary Agenda for next TM, Draft report for Friday)	planning for next meeting	Lake Calhoun, 5th FL
Wednesday WG activity follow-up						
9:00	12:00			Models in Robotics (3h): - Laurent Rioux	discussion	Ceder Lake, 5th GL
				Robotic Functional Services WG(3h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	Lake of the Isles, 5th FL
12:00	14:00			**LUNCH and OMG Plenary**		Nicollet D, 1st FL
14:00	17:00			Robotic Functional Services WG(3h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	Lake of the Isles, 5th FL
18:00	20:00			**OMG Reception**		Nicollet D, 1st FL
Thursday Robotics-DTF Plenary						
9:00	9:45	MARS		Dynamic Deployment and Configuration for RTC (DDC4RTC) RFP 3rd Review - Noriaki Ando and Jae-Yeong Jung	Vote to Issue Joint with **MARS**	Lake Superior A, 5th FL
10:00	10:15	Robotics		Robotic Interaction Service (RoIS) Framework RFP 2nd Review - Su-Young Chi, Miki Sato and Toshio Hori	Vote to Issue	Lake of the Isles, 5th FL
10:15				Adjourn plenary meeting		
10:00	12:00			Robotics WG activity follow-up	discussion	
12:00	13:00			**LUNCH**		Nicollet C/D, 1st FL
13:00	18:00			Architecture Board Plenary		Olsen 3rd FL
Friday						
8:30	12:00			AB, DTC, PTC		Pterson, 3rd FL
12:00	13:00			**LUNCH**		Olsen 3rd FL

Other Meetings of Interest

Monday						
8:00	8:45	OMG		New Attendee Orientation		Skyway Suite A, 2nd FL
9:00	12:00	OMG		Tutorial - Introduction to OMG Specification Tutorial		Skyway Suite A, 2nd FL
18:00	19:00	OMG		New Attendee Reception (by invitation only)		??
Tuesday						
7:30	9:00	OMG		Liaison ABSC		Skyway Suite A, 2nd FL
9:00	17:30	OMG		Symposium on Eclipse Open Source Software & OMG Open Specification		Mirage, 2nd FL
17:00	18:00	OMG		RTF-FTF Chair's Workshop		Lake Nokomis, 5th FL
Wednesday						
9:30	17:30	OMG		System Asurance PTF / Architectur Driven Modernization TF (joint meeting)		Mirage, 2nd FL
Thursday						
9:00	17:00	OMG		System Asurance PTF		Peterson, 3rd FL
17:00	18:00	OMG		MARS Agenda Coordination		Lake Superior A, 5th FL

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

Minutes of the Robotics DTF Meeting

March 22-26, 2010

Jacksonville, FL, USA

(robotics/2010-06-02)

Meeting Highlights

- As the 2nd Review, the draft of Dynamic Deployment and Configuration (DDC) for Robotic Technology Component RFP was discussed. To coordinate views among Remedy IT and Thales, the issue of RFP was postponed to the upcoming Minneapolis meeting in June.
- As the 1st Review, the draft of Robotic Interaction Service (RoIS) Framework RFP was discussed.
- We have a Special Session on JAUS / RTC and have a Special Talk of Mr. David Martin (DeVivo, SAE AS4) in the DTF plenary meeting.
- A kick-off meeting of the Modeling in Robotics was held.
- We have two of new volunteers for the WG Co-Chair.

List of Generated Documents

ab/2010-03-02 RTC Deployment and Dynamic Reconfiguration (DDR) AB review Document (Noriaki Ando)
mars/2010-03-04 RTC Deployment and Dynamic Reconfiguration (DDR) 2nd draft Presentation (Noriaki Ando)
mars/2010-03-05 Dynamic Deployment and Configuration (DDC) for RTC RFP - Errata (Noriaki Ando)
mars/2010-03-06 Dynamic Deployment and Configuration (DDC) for RTC RFP - w/ change bars (Noriaki Ando)
mars/2010-03-07 Dynamic Deployment and Configuration (DDC) for RTC RFP - conv doc (no change bars) (Noriaki Ando)
mars/2010-03-21 Dynamic Deployment and Configuration (DDC) for RTC RFP Presentation (Noriaki Ando)
robotics/2010-03-01 Final Agenda (Tetsuo Kotoku)
robotics/2010-03-02 Long Beach Meeting Minutes [approved] (Geoffrey Biggs and Rockwon Kim)
robotics/2010-03-03 Opening Presentation (Tetsuo Kotoku)
robotics/2010-03-04 RTC Deployment and Dynamic Reconfiguration (DDR) 2nd draft Presentation [copy of mars/2010-03-04] (Noriaki Ando)
robotics/2010-03-05 A new middleware for unmanned systems (Laurent Rioux)
robotics/2010-03-06 RTC Deployment and Dynamic Reconfiguration (DDR) AB review Document (Noriaki Ando) [copy of ab/2010-03-02]
robotics/2010-03-07 Robot Interaction Service (RoIS) Framework (Toshio Hori)
robotics/2010-03-08 GostaiRTC: OMG RTC compliant middleware made by Gostai (Laurent Rioux)
robotics/2010-03-09 SAE JAUS Introductory Breifing (David Martin)
robotics/2010-03-10 Command-line Tools for OpenRTM-aist (Geoffrey Biggs)
robotics/2010-03-11 Introduction of RT-Middleware Tools (Takeshi Sakamoto)
robotics/2010-03-12 RTC / JAUS: 2 complementary standards (Laurent Rioux)
robotics/2010-03-13 Robotic Interaction Service (RoIS) Framework RFP - 1st Review Presentation (Su-Young Chi)
robotics/2010-03-14 Infrastructure WG Progress Report (Seung-Woog Jung)
robotics/2010-03-15 Contact Report: IEEE Standards Workshop (Young-Jo Cho)
robotics/2010-03-16 Status and Plan for Robotic Localization Service (RLS) (Shuichi Nishio)
robotics/2010-03-17 Contact Report: ISO/TC184/SC2 (Tetsuo Kotoku)
robotics/2010-03-18 Robotic Interaction Service (RoIS) Framework (rev.2) (Toshio Hori)
robotics/2010-03-19 Roadmap for Robotics Activities (Tetsuo Kotoku)
robotics/2010-03-20 Wrap-up Presentation (Tetsuo Kotoku)
robotics/2010-03-21 RLS Implementation and Issues (Jae-Yeong Lee)
robotics/2010-03-22 Robotic Interaction Service (RoIS) Framework RFP - Draft (Toshio Hori)

robotics/2010-03-23 Modeling in Robotics (Laurent Rioux)
robotics/2010-03-24 Dynamic Deployment and Configuration (DDC) for RTC RFP - Errata [copy of mars/2010-03-05] (Noriaki Ando)
robotics/2010-03-25 Dynamic Deployment and Configuration (DDC) for RTC RFP - Document with Change Bar [copy of mars/2010-03-06] (Noriaki Ando)
robotics/2010-03-26 Dynamic Deployment and Configuration for RTC (DDC) - Convenience Document without Change Bar [copy of mars/2010-03-07] (Noriaki Ando)
robotics/2010-03-27 Dynamic Deployment and Configuration (DDC) for RTC RFP
robotics/2010-03-28 Next Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)
robotics/2010-03-29 DTC Report Presentation (Tetsuo Kotoku)
robotics/2010-03-30 Jacksonville Meeting Minutes - DRAFT (Yoshihiro Nakabo and Jae-Yeong Lee)

Minutes

Monday, March 22, 2010, City Terrace 4, 3rd FL
Robotics DTF Plenary Meeting,

9:00-9:40Opening Session Chair: Dr Kotoku, Quorums: 3
Joined organizations: AIST, ETRI, JARA, Technologic Arts, ATR
- Minutes takers: Yoshihiro Nakabo(AIST), Jae-Yeong Lee(ETRI)
- Approval of minutes of Long Beach
- Correction: Document 2010-03-02 ISO TC number
- Approved: AIST(motion), ETRI(second), Technologic Art(white ballot)

10:00-10:30 Joint Plenary with MARS, City Terrace 12
- RTC Deployment and Dynamic Reconfiguration (DDR) RFP 2nd Review, Noriaki Ando, AIST, Japan
- Common service and interfaces for component repositories, searching, deployment, directory service,
and detecting/notifying of changes in components are needed.
- Reviced Draft RFP according to the comments and requirements from AB members.
- Adjustment of RFP timetable: discussion with AB on Monday afternoon, reflect AB’s comments on Tuesday and Wednesday, MARS voting on Thursday, AB voting on Friday.
- Revised RFP should be submitted with new document number
- Change on schedules: one meeting delay.

11:00-12:00 Special Talk: A new middleware for unmanned systems, Laurent Rioux (Thales)
- Collaborative work with Thatles and Gostai. First implementation of RTC in Europe (GostaiRTC).
- GostaiRTC is based on C/C++ and compatible with Urbi script. Implemented in Urbi: platform robot.
- Specification of GostaiRTC, Xenomai-GostaiRTC-urbiscript RTC/GostaiRTC interpreter.
- Video Demonstration of Gostai RTC, dynamic creation connection of RTCs by interpreter.
- RTC-related research activity in Europe including Munich, Germany

Tuesday, March 23, 2010, Daytona, 3rd FL
Robotics DTF Plenary Meeting

13:00-15:00 Special Session: JAUS and RTC, Laurent Rioux (Thales)
- RTC introduction/demonstration
- A new middleware for unmanned systems, Laurent Rioux, Thales
- Command-line tools for OpenRTM-aist, Geoffrey Biggs, AIST

- Introduction on JAUS
- JAUS: Joint Architecture for Unmanned System, David Martin, SAE
- RTC introduction/demonstration
- Introduction of RT-Middleware Tools, Takeshi Sakamoto, Technologic Arts Inc.
- RTC and JAUS: 2 complementary standards, Laurent Rioux, Thales
- Discussions/Questions

16:00-16:25 Robotic Interaction Service (RoIS) Framework RFP 1st review, Su-Young Chi (ETRI)
- RoIS framework: HRI engine, event subscription and cancellation, event notification, query, command
- ETRI and JARA to apply RFP.

16:25-16:30 Call for volunteer
- Co-Chair change in Infrastructure WG from Boem-Su Seo(ETRI) to Seung-Woog Jung(ETRI)
JARA(motion), AIST(second), Technologic Arts(white ballot)
- Co-Chair change in Robotic Interaction Service WG from Shuichi Nishio(ATR) to Miki Sato(ATR)
ETRI(motion), JARA(second), Technologic Arts(white ballot)

16:30-16:45 Robotic Infrastructure WG report, Seung-Woog Jung (ETRI)
- RTC DDR RFP 2nd review process:
- WG meeting, 2nd review in MARS, AB Plenary, 2nd WG meeting, MARS meeting for voting, AB Plenary.
- Mandatory requirements have been revised again. Revised version of RFP will be voted in MARS and reviewed in AB Plenary.
- Comments for 4-weeks document, and modification.
- Schedule: The initial submission was delayed by one meeting.
- LOI to submit to RFP due at next meeting on August 31. Initial submission on November 8th, 2010.

16:45-17:00 Contact Report, Young-Jo Cho (ETRI)
- IEEE Standards Workshop will be held at ICRA 2010, Anchorage, Alaska, May 3, 2010.
- http://www.ieee-ras.org/calendar/meetinglist
- Call for join: Anyone who concerns IEEE robot standardization can join the workshop freely.

17:00-17:08 Robotics Localization Service, Robotics Functional Service WG report, Shuichi Nishio(ATR)
- RLS-RTF liaison with ISO/TC211 approved on 5th Jan., 2010.
- OMG specification published on 16th Feb., 2010
- Future plans: revision and meeting at RLS-RTF, TC211 meetings, and coordination with geospatial consortium.

17:08-17:20 Contact Report, Tetsuo Kotoku (AIST)
- ISO/TC184/SC2 reports of WG7 (Personal care safety) February 15-17, 2010, 19 participants.

17:20-17:30 Closing presentation and next agenda, Tetsuo Kotoku (AIST)
- Two plans of next meeting agenda
- Next meeting: June 21-25, Minneapolis, MN, USA

Thursday, March 25, 2010, City Terrace 12, 3rd Floor
Robotics DTF Plenary Meeting

09:00-09:30 Joint Plenary with MARS,
- Presentation of revised RTC Deployment and Dynamic Reconfiguration (DDR) RFP, Noriaki Ando, AIST
- To coordinate views among Remedy IT and Thales, the issue of RFP was postponed to the upcoming Minn

eapolis meeting in June.

Adjourned plenary meeting at 9:30

ATTENDEE (21 Participants)

- Claude Baudoin (Cebe)
- David Martin (DeVivo / SAE AS4)
- David Miller (Boeing)
- Geoffrey Biggs (AIST)
- Hugues VINCENT (Thales)
- Itsuki Noda (AIST)
- Jae-Yeong Lee (ETRI)
- Jacek Skowronek (Thales)
- Laurent Rioux (Thales)
- Mike William (Zeligsoft)
- Miki Sato (ATR)
- Myung-Eun Kim (ETRI)
- Noriaki Ando (AIST)
- Seung-Woog Jung (ETRI)
- Shuichi Nishio (JARA/ATR)
- Su-Young Chi (ETRI)
- Takeshi Sakamoto (Technologic Arts)
- Tetsuo Kotoku (AIST)
- Toshio Hori (AIST)
- Yoshihiro Nakabo (AIST)
- Young-Jo Cho (ETRI)

Prepared and submitted by Yoshihiro Nakabo(AIST) and Jae-Yeong Lee (ETRI).

Robotics-DTF Plenary Meeting
Opening Session

June 21st, 2010

Minneapolis, MN, USA

Hyatt Regency Hotel

Approval of Minutes

Meeting Quorum : 3

AIST, ETRI, JARA, Shibaura IT, Technologic Arts,

Minutes taker(s):

Minutes review

Toshio Hori
Myung-Eun Kim

Jacksonville Meeting Summary

Robotics Plenary: (21 participants)

- Joint 2nd RFP Draft Review with MARS
  - Dynamic Deployment and Configuration (DDC) for Robotic Technology Component RFP [mars/2010-03-05, -06, -07, -21]
- 1st RFP Draft Review
  - Robotic Interaction Service (RoIS) Framework RFP [robotics/2010-03-18, -22]
- Special Session on JAUS and RTC
  - SAE JAUS Introductory Briefing (David Martin) [robotics/2010-03-09]
  - GostaiRTC: OMG RTC compliant middleware made by Gostai (Laurent Rioux) [robotics/2010-03-08]

Agenda Review

Mon:
10:00-11:00 RTC-DDR RFP 3rd Review (joint with MARS-PTF)
11:00-12:00 RoIS RFP 2nd Review

Tue:
13:00-16:00 Robotics-DTF Plenary
15:30-17:00 WG and Contact Report, Wrap-up

Thu:
09:00-09:45 Joint Plenary with MARS
Voting of RTC-DDR RFP
10:00-11:00 Voting of RoIS RFP

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

Object Management Group

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

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

Request for Proposal

Robotic Interaction Service (RoIS) Framework RFP

OMG Document: robotics/2010-06-04

Letters of Intent due: September 13, 2010
Submissions due: November 8, 2010

Objective of this RFP

This RFP solicits proposals for a Platform Independent Model (PIM) and a Platform Specific Model (PSM) for robotic interaction service (RoIS) that specify

- common interfaces between robotic service applications and components that provide functions for performing human-robot interaction.
- data structures for each interface.

For further details see Chapter 6 of this document.

6.0 Specific Requirements on Proposals

6.1 Problem Statement

Service robot that provides some services to people in daily life will become more and more popular in robotics market. These s~~S~~ervice robots ~~interacts with human to~~ provide appropriate services through human-robot interaction (HRI). For example, there are robotic services such as,

- Reception service
- Guide service
- Home security service
- Childcare robot service
- Elder person daily watching service
- ~~etc.~~

Service application is provided as a set of robot behaviors. The robot behavior r~~R~~ule ~~of robot behaviors~~ is defined based on the information collected from humans or environments. The information is collected by using functions such as,

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

Generally, several sensors and actuators are equipped with the~~a~~ robot body or the environment where the robot provides the services~~s~~. T~~, and~~ he~~a~~ service application programmer describes procedures for~~of~~ robot ~~behavior~~ action and relation between the ~~behavior~~ action and the information ~~-~~obtained by these~~the~~ sensors in the~~a~~ service application program. However,~~But~~ the various types of ~~the types of~~ sensors ~~and~~ and actuators are equipped with various robots, ~~s equipped with robots are usually different from each other~~ and~~,~~ moreover, ~~their~~ the application program interfaces (APIs)- of each robot are~~is~~ different by robot vendors even if their sensor types are the same. ~~That is~~Therefore, an application program developed for one specific robot will not run on the other robot. ~~and~~ ~~t~~This is one of the reasons of inefficiency in robot industry.

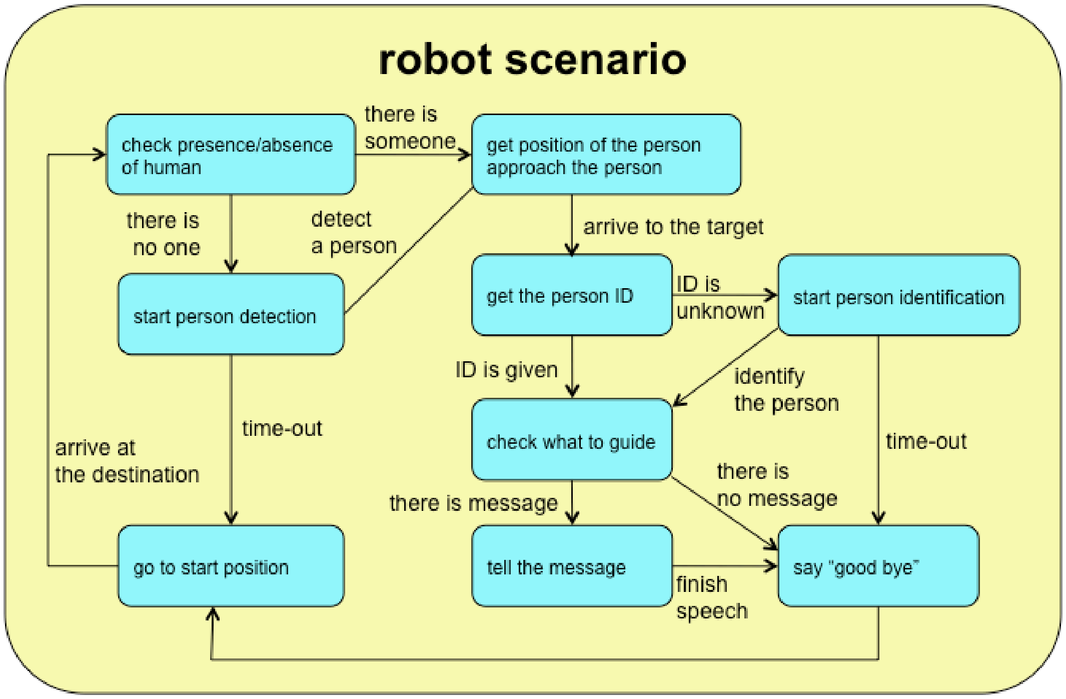


Fig.1: Example of robot scenario for robotic reception service.

In many service applications, the robot behavior rule is defined on a script, which is called a robot scenario. In the robot scenario, instructions to the functions and conditions based on the collected data are described in order to achieve the service task through the APIs that are specific to the robot. For example, the robotic reception service is constructed in the robot scenario as depicted in Fig. 1. In this scenario, when the robot detects someone, the robot approaches the person and tries to recognize who the person is, and then provides information appropriate to the person. Fig.2 shows the messages exchanged between the service application and the robot for the scenario. In this case, one robot (Robot 1) detects human by camera and move by wheels and another robot (Robot 2) detects human by RFID tag and move by legs. Because of the difference in the APIs, the service application programmer must write the different scenarios for each robot respectively.

~~For example in the case of robotic reception service, when a robot detects someone, it approaches the person and tries to recognize who is the person, and then guides the information appropriate to the person. One robot may detect human by camera and move by wheels. Another robot may detect human by RFID tag and move by legs. When there are two types of robots at the reception desks, service application programmer must write programs for each robot independently as depicted in Fig.1.~~

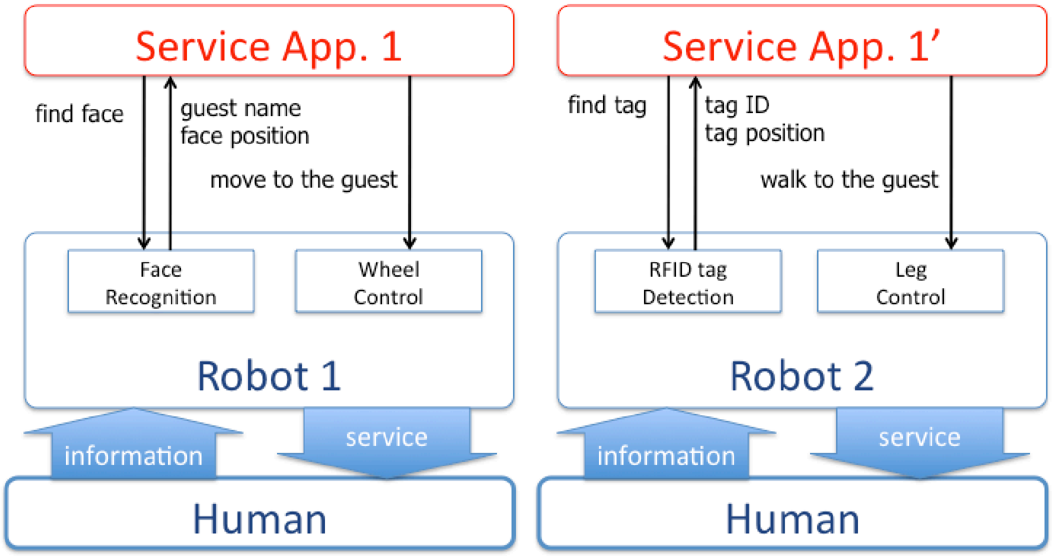


Fig.2~~1~~: Conventional style of service application programming. ~~Service application programmer must write service application programs for each robot independently because functions provided by each robot are different.~~

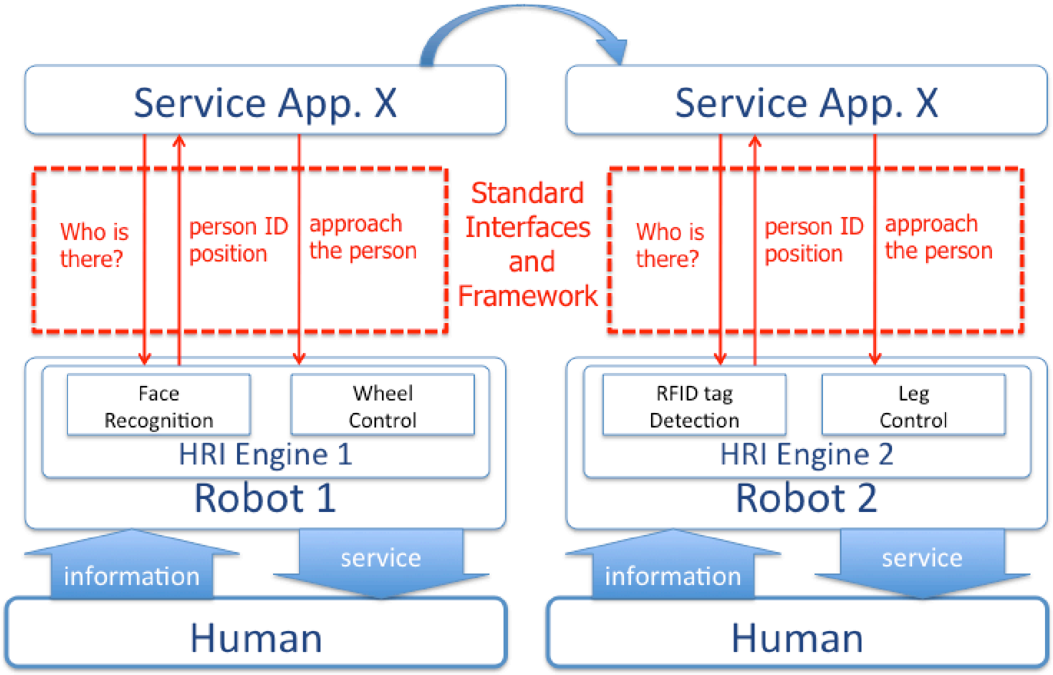


Fig.3~~2~~: RoIS service application programming style. ~~The same service application program works on different robot platforms with little modification.~~

If all the functional components are encapsulated in a package and the interfaces for obtaining information and controlling robots are standardized, they will enhance reusability of service application programs. A service application

program for a robot can work on the other robots regardless of the different robot platforms as depicted in Fig.3~~2~~. Therefore, new general framework architecture is needed. We call the package that encapsulates functional components as HRI Engine and this new framework architecture as Robotic Interaction Service (RoIS).

For the RoIS concept, it is important to standardize mechanisms for the information and the instruction exchanged between the service application and the HRI Engine. From the point of view of a service application, there are generally two types of information to be exchanged, i.e. active information and passive information. The active information is a type of information that should be obtained when the service application needs the target information, such as “check presence/absence of human” and “get position of the person”. The passive information is a type of information that should be provided when the target information is obtained or changed in the HRI Engine, such as “the robot has detected a person” and “the robot has arrived at the target”.

In general, interface types of “Query” and “Event notification” are used for exchanging the active information and the passive information, respectively. To deal with “Event notification”, a mechanism for “Event subscription / cancellation” is also required for selecting appropriate event notifications on demand. In addition, interface type of “Command” is naturally needed in order to instruct the HRI Engine to control its functions, such as “approach the person” and “go to start position”. RoIS framework should also include these interface types, i.e., Event subscription/cancellation, Event notification, Query and Command as illustrated in Fig.4.

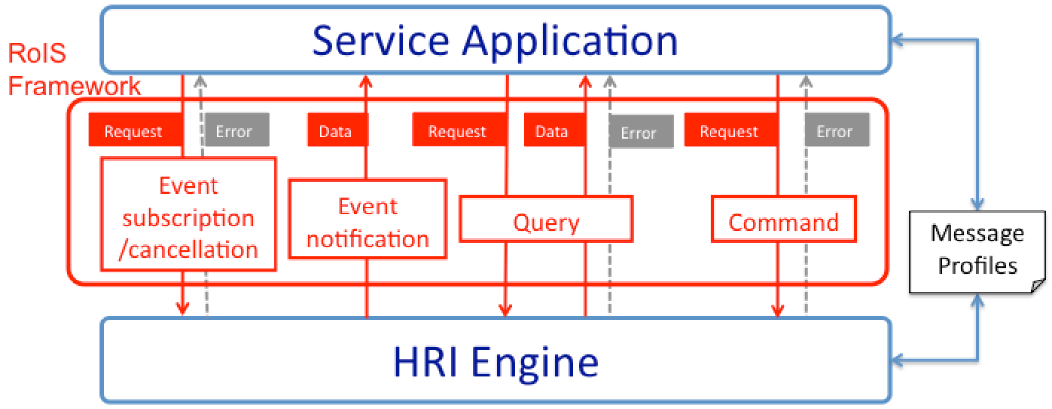


Fig.4: RoIS Framework. In the framework, application communicates with HRI Engine through Event(s) subscription/cancellation, Event(s) notification, Query and Command.

For RoIS framework, ~~must at least four types of interface: Event subscription/cancellation, Event notification, Query and Command as illustrated in Fig..~~

~~“Event subscription/cancellation” is an interface for service application to subscribe to HRI Engine for specific event type(s) or cancel the subscription(s). Available event types are obtained through the “Query” interface.~~

~~“Event notification” interface is required to notify service application of the occurrence of event(s) in real time. When a functional component in HRI Engine detects some status change, such as “a person appeared in a camera view,” “human voice recognized,” or “battery is running out,” then the component sends a specific event to service application through the “Event notification” interface when the application has subscribed to the event.~~

~~“Query” is used to obtain information actively from HRI Engine. Service application requires this interface type on HRI Engine to obtain the Engine’s capabilities (for example, what kind of information is available or what kind of event can be notified by the engine), current status of the Engine, and detailed information of the event notified. Each “Query” has its corresponding result(s).~~

~~“Command” is an interface type to control not only robot but also functional components in HRI Engine. Controlling robot actuators, changing sampling frequency of sensors, and replacing functional components in HRI Engine are the examples of “Commands” to be used.~~

i~~I~~t is desirable to define common messages~~interface items~~ of each interface type that are ~~available~~ suitable for all HRI Engines as possible. These common messages should be proposed. For example, there are common messages for each interface type such as,

- ▶ Event notification
  - Person detected
  - Person identified
  - Speech recognized
  - Sound detected
  - Action completed
  - Low battery
- ▶ Query
  - Presence/absence of human
  - Person ID
  - Position of person / robot
  - History of actions
  - Status of HRI Engine
- ▶ Command
  - Start / stop detection
  - Start / stop recognition
  - Approach the person
  - Follow the person
  - Go to XXX

- Talk XXX

On the other hand, there are also messages specific to each HRI Engine. Therefore, each HRI Engine should provide its message profiles with service applications. Message profiles are composed of common message types and domain specific message types, for example, educational domain, navigation domain and healthcare domain, and each profile may include a list of available messages for each interface type, name of each message, data format for the information exchanged through the message and required argument(s) for using the message. It is required not only to define common messages but also to make a scheme to describe these message profiles so that the service application can make a query about the appropriate message as necessary.

To specify RoIS framework, it is desirable to consider an abstraction level of the messages for the interface types. For example, it should be appreciated that a command message must not use some parameters based on a robot platform. Also, the information managed by "Event notification" and "Query" must be represented at the suitable level for service applications. The abstraction level appropriate to human robot interaction should be considered carefully by focusing on contents of the information and the robot control unique to human robot interaction. In regard to this point, there must be a unique structure for RoIS framework.

~~As for the messages to be exchanged through these interfaces, that is, the data specified and exchanged through queries or by events, and commands sent to control robotic systems, these can be classified into two categories; one for messages that are common to every HRI engine and the other for messages specific to certain service application domains. As services provided by robots may be applied to a variety of domains related to our daily activity such as route guidance or elderly care, it is much efficient to have dictionaries or profiles of messages specific to each domain rather than using a huge set of messages that covers all domains. On the other hand, there are messages that are commonly used in most of the applications that are typical to human-robot interaction. Therefore, the framework shall include the facility to define and to choose message profiles, and at the same time, shall specify profiles that contain basic common messages.~~

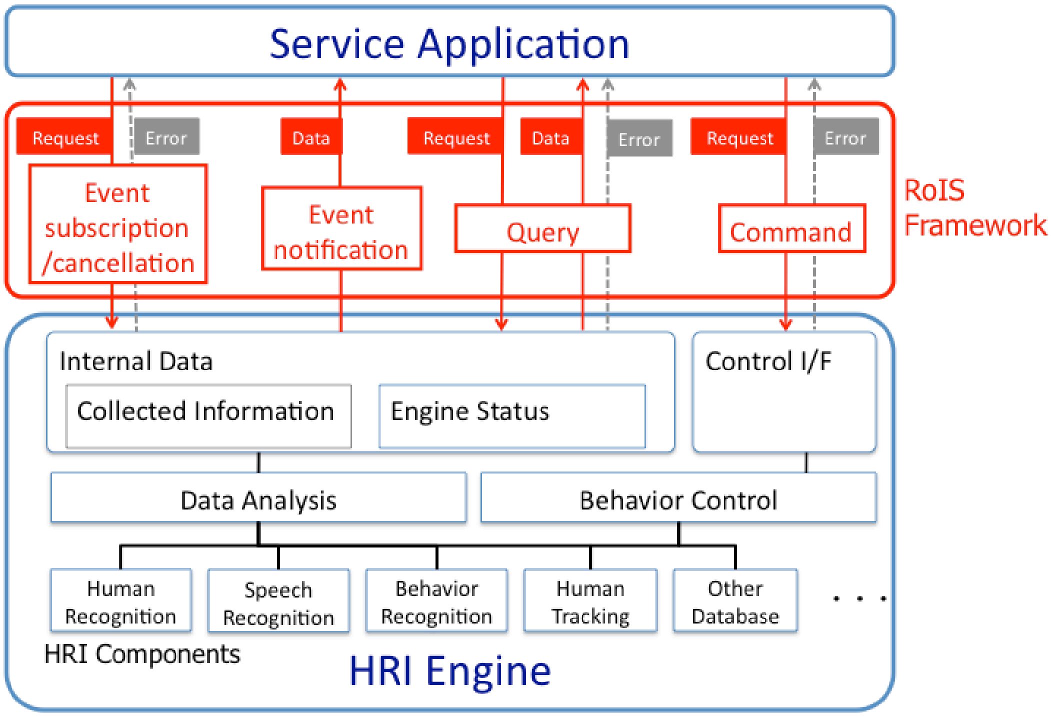


Fig.54: Example of RoIS Framework

Fig.54 illustrates a basic structure of RoIS Framework. HRI Engine integrates several functional components (HRI Components) and provides their functions with a service application through standardized interface. It collects and manages information of human around and environment by sensors, and provides collected data with the application on demand. As some HRI Components in HRI Engine provide robot control functions, the application can control robot through the engine's interface.

Separation and encapsulation of HRI Components into HRI Engine and providing standardized interface to the components will enhance not only the reusability of components but also the efficiency of service application and component development.

Therefore, the scope of the specification solicited includes the definition of interfaces in between service applications and HRI Engine, that is, Event subscription/cancellation, Event notification, Query and Command, and the structure of data transmitted through each interface. Error notification sent from HRI Engine to application may also be included. However, internal data structure that depends on each HRI Engine implementation or data structures defined in existing specifications such as user identification should not be included in this specification.

Considering that HRI Engines depend on their robot platforms, the HRI Engine developers should be able to define interfaces between functional components inside their engine in their independent way. RoIS framework should not concern about HRI Engine inside. For example, one developer can use some other standardized framework, such as RTM, for inside HRI Engine, and the other developer can use their original method. Also, HRI Engine can access to the other applications and databases, such as location data and map data for path generation, by using other framework as needed. The same can be said for service application program inside. For example, in the case that the service application has to be corporate with the other application, such as network robot service, the service application can use other framework to access to the other application.

In summary, following items are required for RoIS framework.

- ~~**Event(s) subscription/cancellation**~~
  - ~~Subscribe to specific event(s) and cancel subscription for specific event(s)~~
  - ~~Sent from application to HRI Engine~~
- ~~**Event(s) notification**~~
  - ~~Notify the occurrence of event to subscriber(s)~~
  - ~~Sent from HRI Engine to application~~
- ~~**Query**~~
  - ~~Retrieve detailed information of events notified by HRI Engine~~
  - ~~Sent from application to HRI Engine (i.e. *requests*) and from HRI Engine to application (i.e. *results*)~~
- ~~**Command**~~
  - ~~Give commands to a robot, components of HRI Engine and/or the engine itself~~

~~Sent from application to HRI Engine~~

- **Interface between service application and HRI Engine**
  - Interface to obtain information from HRI Engine according to the timing of the service application's needs
(Query)
  - Interface to receive information from HRI Engine according to the occurrences of the information in real time
(Event notification / subscription / cancellation)
  - Interface for the instruction to control HRI Engine functions
(Command)
- **Definition of common messages for all HRI Engines**

6.2 Scope of Proposals Sought

This RFP seeks proposals that specify RoIS framework, on top of which various robotic service applications are developed. This RFP targeted on human-robot interaction.~~It is the target for service robots interacting with human.~~

It is necessary to consider the followings~~s~~ in the specification of RoIS framework.

(1) Overview of~~all~~ architecture that consists of RoIS framework, a robotic service application, HRI Engine that is a set of robotic components ~~for RoIS framework~~ shall be ~~defined~~ provided (diagram or description for overview).

(2) The RoIS framework specification shall provide following interfaces between robotic service applications and HRI Engine.~~ that is a set of robotic components~~

- Interface to obtain information actively from HRI Engine
- Interface to receive information timely from HRI Engine
- Interface for the instruction to control HRI Engine functions

~~(2)~~(3) The RoIS framework specification must be general enough to incorporate robotic components for various sensors, actuators and algorithms in HRI Engine.

~~(3)~~(4) The RoIS framework specification shall satisfy interoperability and reusability. A HRI Engine should be able to be replaced with the other HRI engine with little efforts.

~~(4) The RoIS framework specification shall provide a minimum set of functionalities to satisfy the followings:~~

~~1. Interface types between robotic service applications and HRI Engine~~

~~Event subscription and cancellation~~

~~Event notifications~~

~~Query~~

~~Command~~

~~2. Data structure for each interface type~~

~~(5) The RoIS framework specification shall provide a scheme to manage profile of each interface type~~

~~(6)~~(5) The RoIS Framework specification shall specify ~~profiles of message and those of~~ common messages for all HRI Engines.~~each interface type~~

6.3 Relationship to Existing OMG Specifications

Submitters shall examine the following OMG specifications for possible benefit:

- Super Distributed Objects (SDO) Specification version 1.1 [formal/2008-10-11]

- Unified Modeling Language: Infrastructure version 2.1.2 [formal/2007-11-04]
- Unified Modeling Language: Superstructure version 2.1.2 [formal/2007-11-02]
- Lightweight CORBA Component Model 4.0 [formal/2006-04-01]
- Robotic Technology Component specification version 1.0 [formal/08-04-04]
- Robotic Localization Service version 1.0 [formal/2010-02-03]

6.4 Related Activities, Documents and Standards

Proposals may include existing systems, documents, user recognition service interface, and standards that are relevant to the problems discussed in this RFP. They can be used as background information for the proposal.

Example:

- IEEE Robotics and Automation Society, Technical Committee on Network Robot
- IEEE Robotics and Automation Society, Technical Committee on Programming Environment in Robotics and Automation
- ISO/ SC 37 Projects relate to ISO/IEC 19784-1(BioAPI Ver 2.0)
- ISO/TC184/SC2

6.5 Mandatory Requirements

(1) Proposals shall provide overview of~~all~~ architecture that consists of RoIS framework, a robotic service application, HRI Engine that is a set of robotic components ~~for RoIS framework~~ (diagram or description for overview)

~~(1)~~

~~(2) Proposals shall provide a Platform Independent Model (PIM) and at least one Platform Specific Model (PSM) of RoIS framework.~~

~~(3)~~(2)Proposals shall specify a general mechanism for RoIS framework

- Interfaces between applications and HRI Engine and their types:
  - Interface to obtain information actively from HRI Engine
  - Interface to receive information timely from HRI Engine
  - Interface for the instruction to control HRI Engine functions

- Data structure for each interface type

(3) Proposals shall specify communication protocols and middlewares to achieve functions required for RoIS framework.

(4) Proposal shall specify functions that cannot be achieved by existing communication protocols or middlewares.

- ~~The RoIS framework specification shall provide a scheme to manage profile of each interface type~~

~~(4)~~(5) ~~The RoIS Framework specification~~Proposals shall specify ~~profiles of message and those of~~ common messages for ~~each interface type~~all HRI Engines

6.6 Optional Requirements

(1) Proposals may specify schemes for the functionalities listed below.

- Error handling for each interface type
- Returning command results and status

(2) Proposals may provide a schema to describe message profiles.

(3) Proposals may provide a Platform Independent Model (PIM) and Platform Specific Model (PSM) as CORBA-specific model, XML or RTC for RoIS framework.

6.7 Issues to be discussed

These issues will be considered during submission evaluation. They should not be part of the proposed normative specification. (Place them in Part I of the submission.)

- Proposals shall demonstrate its feasibility by using a specific application based on the proposed model.
- Proposals shall demonstrate its applicability to existing robotic interaction service technologies.
- Proposals shall discuss simplicity of implementation.
- Proposals shall discuss the possibility to apply the proposed model to other fields of interest such as intelligent service robot applications
- Proposals shall specify on-the-wire protocol communication technology independent.

- Proposals shall discuss their relation and dependency to existing communication protocols or middlewares standards, such as CORBA [CORBA], DDS [DDS] or RTC [RTC].
- Proposals shall discuss the generality with respect to various sensors, actuators and algorithms in HRI Engine.

6.8 Evaluation Criteria

Proposals will be evaluated in terms of consistency in their specifications, feasibility and versatility across a wide range of different robot applications.

6.9 Other information unique to this RFP

None

6.10 RFP Timetable

The timetable for this RFP is given below. Note that the TF or its parent TC may, in certain circumstances, extend deadlines while the RFP is running, or may elect to have more than one Revised Submission step. The latest timetable can always be found at the OMG *Work In Progress* page at http://www.omg.org/schedules/ under the item identified by the name of this RFP. Note that "<month>" and "<approximate month>" is the name of the month spelled out; e.g., January.

Event or Activity	Actual Date
Preparation of RFP by TF	**24. May. 2010**
RFP placed on OMG document server	**24. May. 2010**
Approval of RFP by Architecture Board *Review by TC*	**21. June. 2010**
TC votes to issue RFP	**25. June. 2010**
LOI to submit to RFP due	**13. September. 2010**
Initial Submissions due and placed on OMG document server ("Three week rule")	**8. November. 2010**
Voter registration closes	**29, November, 2010**
Initial Submission presentations	**6. December. 2010**
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Three week rule")	
Revised Submission presentations	

Final evaluation and selection by TF *Recommendation to AB and TC*	
Approval by Architecture Board *Review by TC*	
TC votes to recommend specification	
BoD votes to adopt specification	

Appendix A References and Glossary Specific to this RFP

A.1 References Specific to this RFP

[RTC] Robotic Technology Component specification version 1.0, http://www.omg.org/spec/RTC/1.0/

A.2 Glossary Specific to this RFP

None

Appendix B General Reference and Glossary

B.1 General References

The following documents are referenced in this document:

[ATC] Air Traffic Control Specification, *http://www.omg.org/technology/documents/formal/air_traffic_control.htm*

[BCQ] OMG Board of Directors Business Committee Questionnaire, *http://www.omg.org/cgi-bin/doc?bc/02-02-01*

[CCM] CORBA Core Components Specification, *http://www.omg.org/technology/documents/formal/components.htm*

[CORBA] Common Object Request Broker Architecture (CORBA/IIOP), *http://www.omg.org/technology/documents/formal/corba_iiop.htm*

[CSIV2] [CORBA] Chapter 26

[CWM] Common Warehouse Metamodel Specification, *http://www.omg.org/technology/documents/formal/cwm.htm*

[DAIS] Data Acquisition from Industrial Systems, http://www.omg.org/technology/documents/formal/dais.htm

[EDOC] UML Profile for EDOC Specification, http://www.omg.org/techprocess/meetings/schedule/UML_Profile_for_EDOC_FTF.html

[EJB] “Enterprise JavaBeans™”, http://java.sun.com/products/ejb/docs.html

[FORMS] “ISO PAS Compatible Submission Template”. http://www.omg.org/cgi-bin/doc?pas/2003-08-02

[GE] Gene Expression, http://www.omg.org/technology/documents/formal/gene_expression.htm

[GLS] General Ledger Specification , http://www.omg.org/technology/documents/formal/gen_ledger.htm

[Guide] The OMG Hitchhiker's Guide,, http://www.omg.org/cgi-bin/doc?hh

[IDL] ISO/IEC 14750 also see [CORBA] Chapter 3.

[IDLC++] IDL to C++ Language Mapping, http://www.omg.org/technology/documents/formal/c++.htm

[MDAa] OMG Architecture Board, "Model Driven Architecture - A Technical Perspective*”, *http://www.omg.org/mda/papers.htm

[MDAb] “Developing in OMG's Model Driven Architecture (MDA),” http://www.omg.org/docs/omg/01-12-01.pdf

[MDAc] “MDA Guide” (http://www.omg.org/docs/omg/03-06-01.pdf)

[MDAd] “MDA "The Architecture of Choice for a Changing World™"”, http://www.omg.org/mda

[MOF] Meta Object Facility Specification, http://www.omg.org/technology/documents/formal/mof.htm

[MQS] “MQSeries Primer”, http://www.redbooks.ibm.com/redpapers/pdfs/redp0021.pdf

[NS] Naming Service, http://www.omg.org/technology/documents/formal/naming_service.htm

[OMA] “Object Management Architecture™”, http://www.omg.org/oma/

[OTS] Transaction Service, *http://www.omg.org/technology/documents/formal/transaction_service.htm*

[P&P] Policies and Procedures of the OMG Technical Process, *http://www.omg.org/cgi-bin/doc?pp*

[PIDS] Personal Identification Service, *http://www.omg.org/technology/documents/formal/person_identification_service.htm*

[RAD] Resource Access Decision Facility, *http://www.omg.org/technology/documents/formal/resource_access_decision.htm*

[RFC2119] IETF Best Practices: Key words for use in RFCs to Indicate Requirement Levels, (*http://www.ietf.org/rfc/rfc2119.txt*).

[RM-ODP] ISO/IEC 10746

[SEC] CORBA Security Service, *http://www.omg.org/technology/documents/formal/security_service.htm*

[TOS] Trading Object Service, *http://www.omg.org/technology/documents/formal/trading_object_service.htm*

[UML] Unified Modeling Language Specification, *http://www.omg.org/technology/documents/formal/uml.htm*

[UMLC] UML Profile for CORBA, *http://www.omg.org/technology/documents/formal/profile_corba.htm*

[XMI] XML Metadata Interchange Specification, *http://www.omg.org/technology/documents/formal/xmi.htm*

[XML/Value] XML Value Type Specification, *http://www.omg.org/technology/documents/formal/xmlvalue.htm*

B.2 General Glossary

Architecture Board (AB) - The OMG plenary that is responsible for ensuring the technical merit and MDA-compliance of RFPs and their submissions.

Board of Directors (BoD) - The OMG body that is responsible for adopting technology.

Common Object Request Broker Architecture (CORBA) - An OMG distributed computing platform specification that is independent of implementation languages.

Common Warehouse Metamodel (CWM) - An OMG specification for data repository integration.

CORBA Component Model (CCM) - An OMG specification for an implementation language independent distributed component model.

Interface Definition Language (IDL) - An OMG and ISO standard language for specifying interfaces and associated data structures.

Letter of Intent (LOI) - A letter submitted to the OMG BoD's Business Committee signed by an officer of an organization signifying its intent to respond to the RFP and confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements.

Mapping - Specification of a mechanism for transforming the elements of a model conforming to a particular metamodel into elements of another model that conforms to another (possibly the same) metamodel.

Metadata - Data that represents models. For example, a UML model; a CORBA object model expressed in IDL; and a relational database schema expressed using CWM.

Metamodel - A model of models.

Meta Object Facility (MOF) - An OMG standard, closely related to UML, that enables metadata management and language definition.

Model - A formal specification of the function, structure and/or behavior of an application or system.

Model Driven Architecture (MDA) - An approach to IT system specification that separates the specification of functionality from the specification of the implementation of that functionality on a specific technology platform.

Normative – Provisions that one must conform to in order to claim compliance with the standard. (as opposed to non-normative or informative which is explanatory material that is included in order to assist in understanding the standard and does not contain any provisions that must be conformed to in order to claim compliance).

Normative Reference – References that contain provisions that one must conform to in order to claim compliance with the standard that contains said normative reference.

Platform - A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented.

Platform Independent Model (PIM) - A model of a subsystem that contains no information specific to the platform, or the technology that is used to realize it.

Platform Specific Model (PSM) - A model of a subsystem that includes information about the specific technology that is used in the realization of it on a specific platform, and hence possibly contains elements that are specific to the platform.

Request for Information (RFI) - A general request to industry, academia, and any other interested parties to submit information about a particular technology area to one of the OMG's Technology Committee subgroups.

Request for Proposal (RFP) - A document requesting OMG members to submit proposals to the OMG's Technology Committee. Such proposals must be received by a certain deadline and are evaluated by the issuing task force.

Task Force (TF) - The OMG Technology Committee subgroup responsible for issuing a RFP and evaluating submission(s).

Technology Committee (TC) - The body responsible for recommending technologies for adoption to the BoD. There are two TCs in OMG – *Platform TC* (PTC), that focuses on IT and modeling infrastructure related standards; and *Domain TC* (DTC), that focus on domain specific standards.

Unified Modeling Language (UML) - An OMG standard language for specifying the structure and behavior of systems. The standard defines an abstract syntax and a graphical concrete syntax.

UML Profile - A standardized set of extensions and constraints that tailors UML to particular use.

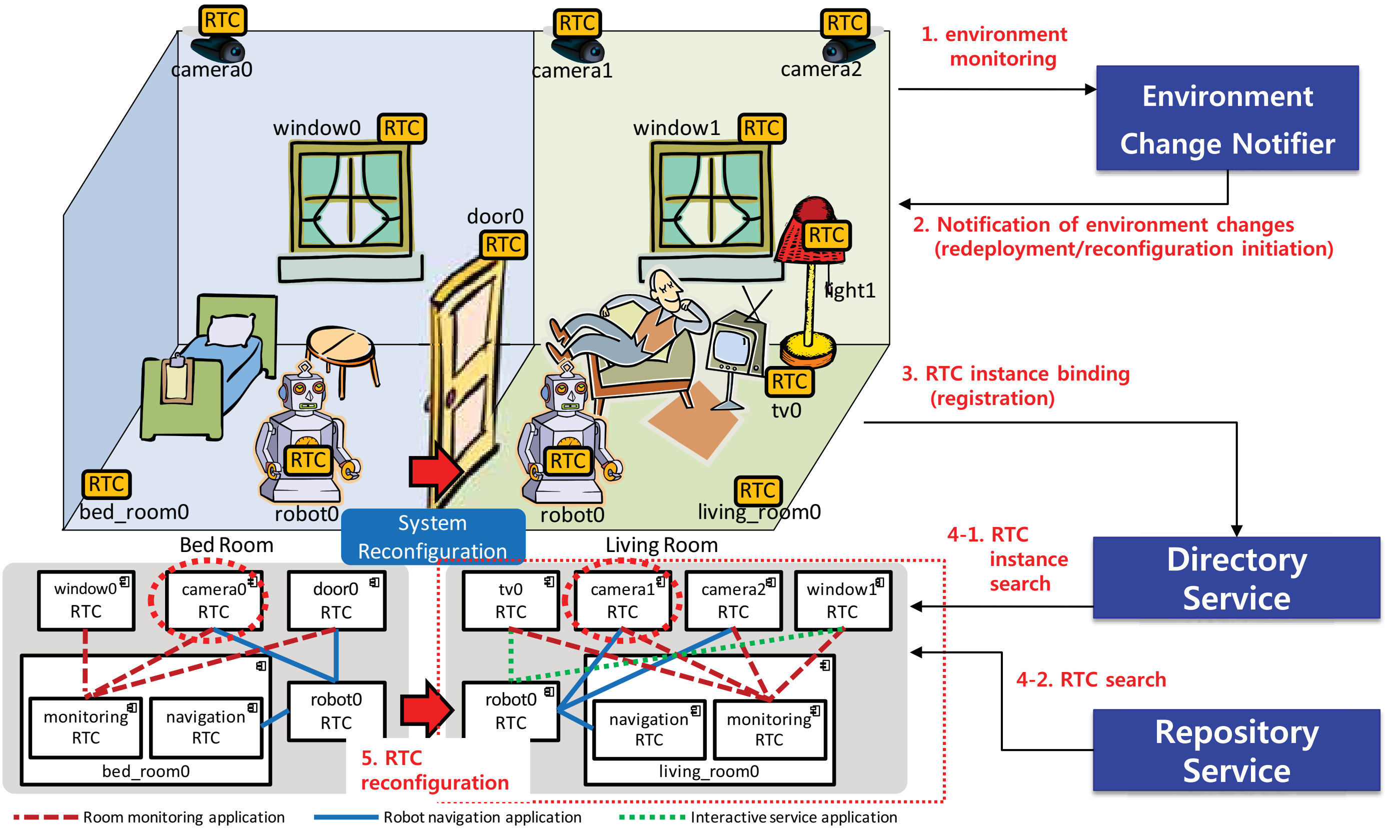
XML Metadata Interchange (XMI) - An OMG standard that facilitates interchange of models via XML documents.

Dynamic Deployment and Configuration (DDC) for RTC RFP 3rd draft

document number:
mars/2010-05-07
mars/2010-06-04 (errata)
mars/2010-06-05 (revised)
presentation: mars/2010-06-03

Seung-Woog Jung
Infrastructure WG, Robotics DTF
ETRI(Electronics and Telecommunications Research Institute)

Motivation and Scope



1. environment monitoring

2. Notification of environment changes (redeployment/reconfiguration initiation)

3. RTC instance binding (registration)

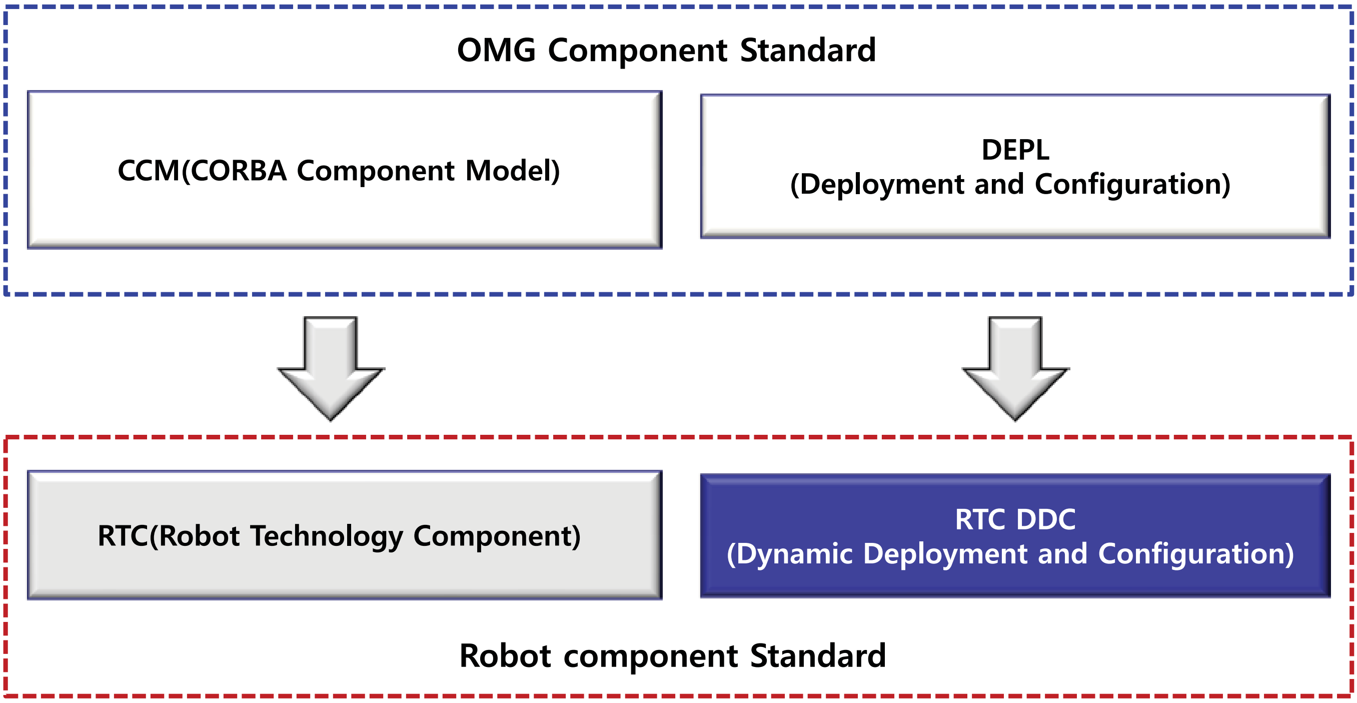
4-1. RTC instance search

4-2. RTC search

5. RTC reconfiguration

Room monitoring application
Robot navigation application
Interactive service application

Relation with other OMG Specs


OMG Component Standard
CCM(CORBA Component Model)
DEPL
(Deployment and Configuration)
RTC(Robot Technology Component)
RTC DDC
(Dynamic Deployment and Configuration)
Robot component Standard

Mandatory Requirements

- Platform independent deployment and configuration model
  - [6.5.1] Proposals shall specify services for dynamic configuration and deployment of RTCs.
  - [6.5.2] Proposal shall specify means to initiate RTC configuration based on external and/or internal events. A capability for event filtering shall be provided.
  - [6.5.3] Proposals shall reuse or extend the deployment architecture as defined by the Deployment and Configuration of Component-based Distributed Applications Specification [D&C].

Mandatory Requirements

- Platform independent RTC information model
  - [6.5.4] Proposals shall provide a schema describing RTC characteristics such as ....
  - [6.5.5] Proposals shall provide a schema describing RTC-based systems characteristics.
  - [6.5.6] Proposals shall specify query services to discover and interrogate characteristics of RTCs and RTC-based systems.
  - [6.5.7] Proposal shall specify query services to discover characteristics and location information of deployed RTCs and RTC-based systems.

Comments from AB

- Document : mars/2010-05-07 (submitted 4 weeks ago)

- Comments from AB members
  - Notation problems
  - 6.5 Mandatory Requirements
  - Schedule

RFP Revision

- Based on the comments from AB member,
  - In the 6.1 Problem Statement: 1. RTC profile, it is not clearly described what is RTC profile. At the last sentence, the text “which is described in the RTC specification” has been added.
  - According to the OMG's official abbreviations, all the "D&C"s are replaced with “DEPL."
  - In the mandatory requirement 6.5.1, "services for" has been replaced with "interfaces to services for.“
  - In the 6.5.1, to make the requirement clear, capabilities to be expected in this requirement have been listed below.
  - In the 6.5.2, "means" has been replaced with "interfaces.“

RFP Revision

- In the 6.5.3, to make the meaning of reuse of DEPL more clear, the text "at least the PIM, including terms and definitions, of" has been added into the first sentence.
- In the 6.5.4, the non-completed sentence has been completed: “Proposals shall provide a schema, the RTC Profile, describing RTC characteristics such as basic RTC information, ports information and so on, based on the RTC specification.”
- In the 6.5.5, to make the sentence clear, the sentence "such as port connection information, configuration information, deployment conditions and so on" has been added.
- In the 6.8.2, reuse of DEPL specification has been added as evaluation criteria.
- The timetable has been updated.

Revised Mandatory Requirements

- Platform independent deployment and configuration model
  - [6.5.1] Proposals shall specify interfaces to services for dynamic configuration and deployment of RTCs.
    - storing, searching and retrieving RTC,
    - storing, searching and retrieving RTC-based applications
    - RTC registration
  - [6.5.2] Proposal shall specify interfaces to initiate RTC configuration based on external and/or internal events. A capability for event filtering shall be provided.
  - [6.5.3] Proposals shall reuse or extend at least the PIM, including terms and definitions, of the deployment architecture as defined by the Deployment and Configuration of Component-based Distributed Applications Specification [DEPL].

Revised Mandatory Requirements

- Platform independent RTC information model
  - [6.5.4] Proposals shall provide a schema, the RTC Profile, describing RTC characteristics such as basic RTC information, ports information and so on, based on the RTC specification
  - [6.5.5] Proposals shall provide a schema, the RTC-based System Profile, describing RTC-based systems characteristics such as port connection information, configuration information, deployment conditions and so on.
  - [6.5.6] Proposals shall specify query services to discover and interrogate characteristics of RTCs and RTC-based systems.
  - [6.5.7] Proposal shall specify query services to discover characteristics and location information of deployed RTCs and RTC-based systems.

Revised RFP Timetable

Event or Activity	Actual Date
Preparation of RFP by TF	
RFP placed on OMG document server	*February 22nd, 2010*
Approval of RFP by Architecture Board *Review by TC*	*June, 2010*
TC votes to issue RFP	*June, 2010*
LOI to submit to RFP due	*August 31, 2010*
Initial Submissions due and placed on OMG document server ("Four week rule")	*November 8th, 2010*
Voter registration closes	*December, 2010*
Initial Submission presentations	*December, 2010*
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Four week rule")	*May , 2011*
Revised Submission presentations	*June, 2011*
Final evaluation and selection by TF *Recommendation to AB and TC*	
Approval by Architecture Board *Review by TC*	
TC votes to recommend specification	*June, 2011*
BoD votes to adopt specification	*September, 2011*

Current Status

- Two organizations are interested in the RTC-DDC RFP
  - Remedy IT & Vanderbilt University
  - Thales
- Professor Otte from Vanderbilt University, Laurent from Thales are attending this meeting (MARS plenary)
  - We would like to receive some comments from them about the scope of the RFP

Thank you

mars/2010-06-04

Errata to revised submission of the RTC Deployment and Dynamic Reconfiguration (DDR) Request For Proposal, mars/2010-05-07

This document lists some minor errors and corrections to the revised submission of the RTC Deployment and Dynamic Reconfiguration (DDR) Request For Proposal, mars/10-05-07

Change Overview

Based on the comments from AB member,

1. In the 6.1 Problem Statement: 1. RTC profile, it is not clearly described what is RTC profile. At the last sentence, the test “which is described in the RTC specification” has been added.
2. According to the OMG's official abbreviations, all the "D&C" are replaced with "DPEL."
3. In the 6.5.1, "services for" has been replaced with "interfaces to services for.“
4. In the 6.5.1, to make the requirement clear, capabilities to be expected in this requirement have been listed below.
5. In the 6.5.2, "means" has been replaced with "interfaces.”
6. In the 6.5.3, to make the meaning of reuse of DEPL more clear, the text "at least the PIM, including terms and definitions, of" has been added into the first sentence.
7. In the 6.5.4, the non-completed sentence has been completed: “Proposals shall provide a schema, the RTC Profile, describing RTC characteristics such as basic RTC information, ports information and so on, based on the RTC specification.”
8. In the 6.5.5, to make the sentence clear, the sentence "such as such as port connection information, configuration information, deployment conditions and so on" has been added.
9. In the 6.8.2, reuse of DEPL specification has been added as evaluation criteria.
10. The timetable has been updated.

Object Management Group

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

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

Dynamic Deployment and Configuration (DDC) of Robotic Technology Components

Request For Proposal Draft

OMG Document: mars/2010-~~05-07~~06-05

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

Objective of this RFP

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

In particular, the proposal shall provide:

- Ways to search for and deploy Robotic Technology Components (RTC) into robotic systems at run-time.
- Ways to notify the relevant RTC instances of environment changes.
- Ways to search for appropriate RTC instances and dynamically configure them.

6.0 Specific Requirements on Proposals

6.1 Problem Statement

Generally, most component-based software platforms have their own specifications for component deployment and configuration. We already have the Robotic Technology Component (RT-Component: RTC) Specification in the OMG for a component-based robot software platform. The component model for robotics domain-specific design patterns is described in the current RTC specification. However, functionality such as deployment and configuration, which are usually supported by middleware services or facilities, are not defined.

As the general UML (Unified Modeling Language) component model has been extended in the RTC specification, in order to apply it to the robotics domain, some services and facilities also should be extended with robot-specific characteristics. Existing specifications are inadequate to meet the requirements of robotics. They are general purpose and are oriented toward static software systems, not dynamic software systems such as robotic systems. This RFP describes dynamic deployment and configuration specific to RT components.

A robot is a mobile system that interacts with the real environment. Figure 1 shows the typical robotic application environment. A robot moves around from one place to another in the dynamic environment and it can use the environment's resources, which include sensors, robotic devices and other robots.

In the robot application development phase, we may not know what environment the robot will be installed to and, furthermore, what environment changes will occur while the robot is operating. These dynamic characteristics should be considered not at software build-time but at runtime. This means that RTC-based systems can be deployed and reconfigured at runtime according to environment changes. Therefore a new flexible, adaptive, and dynamically configurable mechanism and method are required to meet the dynamic characteristics of robot applications.

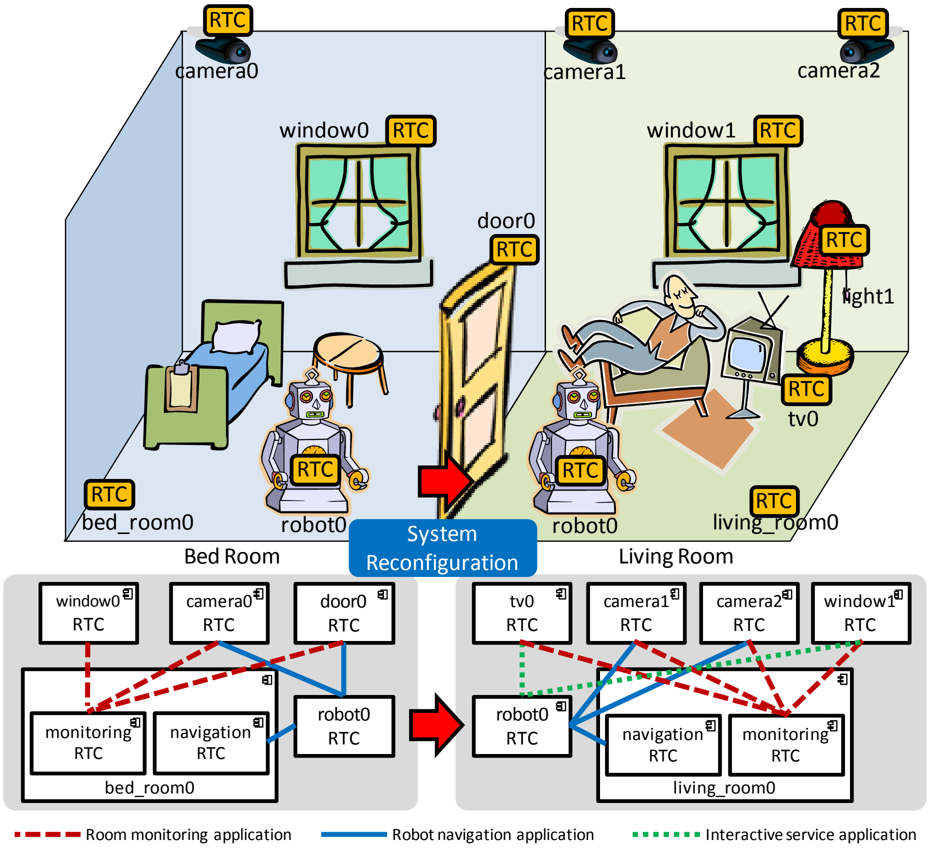


Figure 1 Typical robotic application environment

In order to address functionality of dynamic deployment and configuration, the following issues should be included:

1. RTC profile

A component can generally have common profile information, and as shown in Figure 2, this profile information can be used in the component development phase, system development phase, simulation, and so on. Furthermore, when using a repository server that accumulates many components, this information can be utilized for storing, searching and retrieving components from it. This is called a component profile, which is described in the RTC specification.

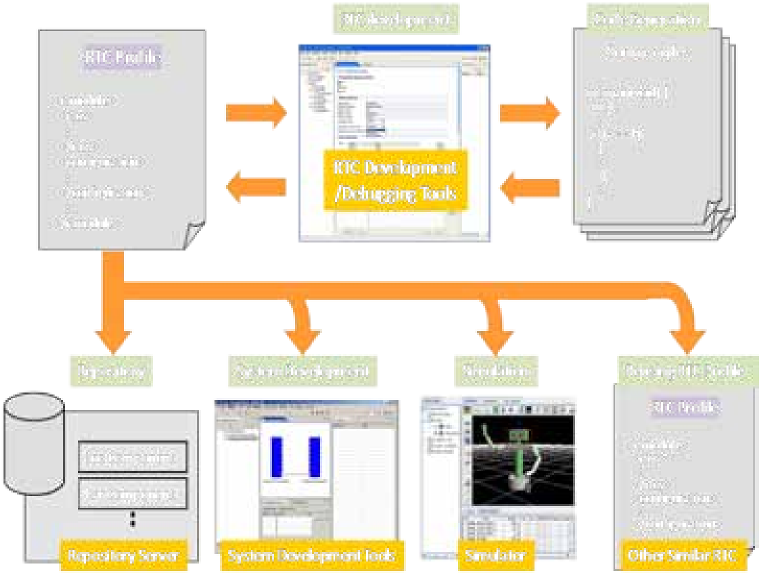


Figure 2 Use of the RTC Profile

2. RTC-based system profile

An RTC-based system is generally built by composing the RTCs or RTC-based subsystems. An RTC-based system or subsystem shall consist of connection information among RTCs, configuration information for RTCs, and so on. This information is called an RTC-based system profile. As shown in Figure 3, this information can be utilized for simulation or component deployment for actual systems. Usually, the components are installed on the target system prior to starting it. (Here, we are focusing on static systems only. The dynamic case will be addressed in the following issues.) Therefore, the person who wants to deploy components has to prepare all the components that constitute the target system. Also, as the number of RTCs and component developers (or developing organizations) is increasing, the person in charge of deployment cannot personally manage all the RTCs that are built. In these cases, a central repository, which manages all the RTCs built, is very helpful in deploying to robot systems. It enables people who want to deploy components to search for what they want in the repository and download/install the components found onto the target hardware. Moreover, if they describe the composing components in a computer-understandable form, the RT middleware is now able to automatically search, download, and install the RTCs while deploying the system.

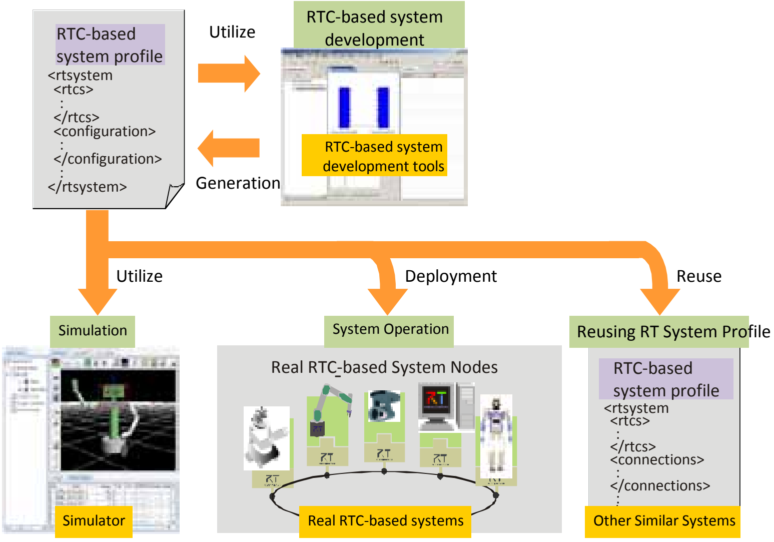


Figure 3 Use of RTC-based system profile

3. RTC-based system deployment

The current RTC specification does not provide a declarative way to compose RTCs to build a robot application or system. Many component based systems present a deployment method that can describe the target application (or system) by combining their components. However those descriptions are not suitable for the robotics domain, which inherently suffers from environment changes during operation time due to mobility. Links between components established at deployment time become obsolete as a robot moves to a new environment. In order to handle these situations, the method of describing the links should be declarative enough such that the description remains valid as the surrounding environment changes over time.

A robot consists of different kinds of sensor and actuator devices and usually includes multiple computing nodes. The RTC-based system should consider the automated deployment of RTCs to the distributed nodes. However, the existing RTC specification suffers from insufficient support for deployment and configuration of software components of distributed applications.

4. RTC instance lookup

As mentioned above, a robot application (or system) consists of RTCs and links among them. Here, the components which are participating in the link are not limited to a single node (or host) but are placed on separate nodes. In this case, it is necessary to search for appropriate component instances running throughout the distributed system. To fulfill these requirements, the specification should provide an RTC directory, which is in charge of searching for a candidate component instance to be linked with other component instances. Since a meta-

information-based component instance search is needed, the specification must also define the data model for the meta-information of RTC. Finally, in order for the RTC directory to find the right component instance that matches the requirements, all the meta-information of the component instances running throughout the distributed system must be known to the directory. Therefore the specification must also specify the registering (and conversely unregistering) processes by which all component instances register their own meta-information with the directory.

5. RTC instance tracking

As mentioned earlier, robotic systems have a unique characteristic in that their surrounding context may change during operation time. In such cases, a link between component instances could become invalid, and so need to be removed and re-established between different component instances. This kind of configuration commonly results from the impairment of the participating component instances and/or changes in the robot location. To support such configuration, the robot application (or system) needs to be notified whenever the situation changes. Since not all changes require configuration, it must be possible to specify the specific environment changes that trigger configuration. It is desirable that the specification is also based on meta-information of component instances and looks similar to that for the component instance searching.

We already have the RTC specification in the OMG for the reusability and interoperability of robot modules. We also have the ~~D&C~~DEPL (Deployment and Configuration of Component-based Distributed Applications specification) in the OMG for deployment and configuration of component based distributed applications.

RTC defines a component model and infrastructure services applicable to the domain of robotics software development. By extending the general-purpose component functionality of UML with direct support for domain-specific structural and behavioral design patterns, RTCs serve as powerful building blocks in an RTC-based system. The RTC specification provides a way to make RTCs and build RTC-based systems. However, it does not discuss how to deploy and configure RTCs at runtime.

~~D&C~~DEPL defines installation, configuration, planning, preparation, and launch process for component-based applications. ~~D&C~~DEPL could support the deployment and configuration of components at build time. However it cannot cover the deployment and configuration of components at run time and meet the dynamic characteristics for robotic systems.

To use ~~D&C~~DEPL in the robotics domain and expand RTC, the RFP proposes the specifications for the dynamic deployment and configuration specific to RT components.

6.2 Scope of Proposals Sought

This RFP solicits proposals to specify common interfaces and common data models for RTC dynamic deployment and configuration that is specific and relevant to robot applications. The proposals shall include a PIM, using UML in the most recent public available version, and one or more PSMs, including one based on OMG IDL (Interface Definition Language) and XML (eXtensible Mark-up Language).

The proposed specification shall provide functionality for component deployment and dynamic system configuration for RTC based systems. The specification must be general enough to allow a variety of robotic systems to be easily constructed, and must be provided for interoperability.

It is necessary to consider the following in the specification:

(1) The repository service interfaces for storing, searching, and retrieving RTCs, and the data model for the component profile description. The component profile might be extensible to include related hardware's functional, mechanical, electrical, physical or geometrical information. This information is helpful in the design and simulation processes.

(2) The repository service interfaces for storing, searching, and retrieving RTC-based systems, and the data model for the RTC-based system profile description.

(3) The service interfaces for the deployment of RTCs into the nodes that constitute RTC-based systems at run time, and the data model for describing the details of deployment.

(4) The directory service interfaces for RTC instance discovery, and the data model for describing the RTC instance. In addition to functions such as registration and searching, this service might provide certain functionality such as notifying environmental changes to RTC based applications or filtering such events based on previously registered condition.

6.3 Relationship to other OMG Specifications and activities

6.3.1 Relationship to OMG specifications

- Platform Independent Model and Platform Specific Model for super Distributed Object Specification Version 1.1 [formal/2008-10-01] http://www.omg.org/spec/SDO/1.1

- Robotic Technology Component Specification Version 1.0 [formal/2008-04-04] http://www.omg.org/spec/RTC/1.0

- Deployment and Configuration of Component-based Distributed Applications Specification OMG Available Specification Version 4.0 [formal/2006-04-02] http://www.omg.org/spec/DEPL/4.0

- Unified Modeling Language: Infrastructure Version 2.3 [formal/2009-02-04] http://www.omg.org/spec/UML/2.3/Infrastructure/PDF/

- Unified Modeling Language: Superstructure Version 2.3 [formal/2009-02-02] http://www.omg.org/spec/UML/2.3/Superstructure/PDF/

- Meta Object Facility (MOF) Core Specification OMG Available Specification Version 2.0 [formal/06-01-01] http://www.omg.org/spec/MOF/2.0/

- Common Object Request Broker Architecture (CORBA/IIOP) 3.1 [formal/2008-01-04, formal/2008-01-06, formal/2008-01-08] http://www.omg.org/spec/CORBA/3.1/Interfaces/PDF/

- CORBA Component Model OMG Available Specification Version 4.0 [formal/2006-04-01] http://www.omg.org/spec/CCM/4.0

- Lightweight Services Specification Version 1.0 [formal/04-10-01] http://www.omg.org/spec/LtSVC/1.0/

- Event Service Specification Version 1.2 [formal/04-10-02] http://www.omg.org/spec/EVNT/1.2/

- Naming Service Specification Version 1.3 [formal/04-10-03] http://www.omg.org/spec/NAM/1.3/

- Enhanced View of Time Specification Version 1.2 [formal/04-10-04] http://www.omg.org/spec/EVoT/2.0

- Property Service Specification Version 1.0 [formal/00-06-22] http://www.omg.org/spec/PROP/1.0/

- Mobile Agent Facility Specification Version 1.0 [formal/2000-01-02] http://www.omg.org/spec/MOBFAC/1.0/

- PIM and PSM for Software Radio Components (SDRP) Version 1.0 [formal/07-03-01] http://www.omg.org/spec/SDRP/

- UML Profile For MARTE: Modeling And Analysis Of Real-Time Embedded Systems [formal 2009-11-02] http://www.omg.org/spec/MARTE/1.0

- MARTE Profile XMI file [ptc/09-05-15] http://www.omg.org/spec/MARTE/20090501

- MARTE model library XMI file [ptc/09-05-16] http://www.omg.org/spec/MARTE/20090502

6.3.2 Relationship to other OMG Documents and work in progress

None

6.4 Related non-OMG Activities, Documents and Standards

- CLARAty: Coupled Layer Architecture for Robotic Autonomy http://robotics.jpl.nasa.gov/tasks/claraty/homepage.html

- Network Robot Forum http://www.scat.or.jp/nrf/

- IEEE Robotics and Automation Society, Technical Committee on Network Robot

- IEEE Robotics and Automation Society, Technical Committee on Programming Environments in Robotics and Automation

- OpenRT Platform http://www.openrtp.jp

- OpenRTM-aist http://www.openrtm.org

- OpenRAVE: http://openrave.programmingvision.com

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

- OROCOS: Open Robot Control Software, Open Realtime Control Service http://www.orocos.org/
- Orca: http://orca-robotics.sourceforge.net/
- ORiN :Open Robot/Resource Interface for the Network: http://www.orin.jp/
- Player/Stage: http://playerstage.sourceforge.net/
- Ptolemy Project: http://ptolemy.eecs.berkeley.edu/
- RCS (Realtime Control Systems Architecture): http://www.isd.mel.nist.gov/projects/rcs/
- ROS: http://www.ros.org
- RSi: Robot Service Initiative: http://www.robotservice.org/
- RT middleware Project: http://www.is.aist.go.jp/rt
- SAE AADL (Society for Automotive Engineers, Architecture Analysis and Design Language): http://www.aadl.info/
- RETF (Robotics Engineering Task Force): http://www.robo-etf.org/
- URC (Ubiquitous Robotic Companion) Project
- Yaorozu Project: http://www.8mg.jp/

6.5 Mandatory Requirements

Proposals shall provide a Platform Independent Model (PIM) expressed in UML and at least one Platform Specific Model (PSM) as CORBA-specific model and XML schema for RTC Dynamic Deployment and Configuration. The models shall meet the following requirements.

Platform independent deployment and configuration model

6.5.1 Proposals shall specify interfaces to services for dynamic configuration and deployment of RTCs.

- storing, searching and retrieving RTC,
- storing, searching and retrieving RTC-based applications

- RTC registration

~~6.5.1~~6.5.2 Proposal shall specify interfaces~~means~~ to initiate RTC configuration based on external and/or internal events. A capability for event filtering shall be provided.

~~6.5.2~~6.5.3 Proposals shall reuse or extend at least the PIM, including terms and definitions, of the deployment architecture as defined by the Deployment and Configuration of Component-based Distributed Applications Specification [~~D&C~~DEPL].

Platform independent RTC information model

~~6.5.3~~6.5.4 Proposals shall provide a schema, the RTC Profile, describing RTC characteristics such as basic RTC information, ports information and so on, based on the RTC specification

~~6.5.4~~6.5.5 Proposals shall provide a schema, the RTC-based System Profile, describing RTC-based systems characteristics such as port connection information, configuration information, deployment conditions and so on.

~~6.5.5~~6.5.6 Proposals shall specify query services to discover and interrogate characteristics of RTCs and RTC-based systems.

~~6.5.6~~6.5.7 Proposal shall specify query services to discover characteristics and location information of deployed RTCs and RTC-based systems.

6.6 Optional Requirements

6.6.1 Proposals may support coordinated RTC configuration of multiple robot systems to allow the performance of coordinated tasks.

6.7 Issues to be discussed

These issues will be considered during submission evaluation. They should not be part of the proposed normative specification. (Place them in Part I of the submission.)

6.7.1 Proposals shall discuss the possibility of applying the proposed model to other existing fields/projects of interest that deploy components such as CCM [CCM], SDRP [SDRP], ~~D&C~~DEPL [~~D&C~~DEPL] and other well-known component models.

6.7.2 Proposals shall discuss their relation to and dependency on existing communication protocols or middleware standards, such as CORBA [CORBA] or DDS [DDS].

6.7.3 Proposals shall discuss efficient methods/procedures to avoid the need for extensive information discovery activities when interacting with the environment or other robots.

6.8 Evaluation Criteria

6.8.1 Demonstration of a proposal with a working implementation may aid in selection.

6.8.2 Reuse of existing technology, such as the RTC specification and DEPL specification, is considered important.

6.9 Other information unique to this RFP

None.

6.10 RFP Timetable

The timetable for this RFP is given below. Note that the TF or its parent TC may, in certain circumstances, extend deadlines while the RFP is running, or may elect to have more than one Revised Submission step. The latest timetable can always be found at the OMG *Work In Progress* page at http://www.omg.org/schedules under the item identified by the name of this RFP.

Event or Activity	Actual Date
Preparation of RFP by TF	
RFP placed on OMG document server	*February 22nd, 2010*
Approval of RFP by Architecture Board *Review by TC*	*~~March~~June, 2010*
TC votes to issue RFP	*~~March~~June, 2010*

LOI to submit to RFP due	*August 31, 2010*
Initial Submissions due and placed on OMG document server ("Four week rule")	*November 8th, 2010*
Voter registration closes	*December, 2010*
Initial Submission presentations	*December, 2010*
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Four week rule")	*May , 2011*
Revised Submission presentations	*June ~~??~~, 2011*
Final evaluation and selection by TF *Recommendation to AB and TC*	
Approval by Architecture Board *Review by TC*	
TC votes to recommend specification	*June, 2011*
BoD votes to adopt specification	*September, 2011*

Appendix A References and Glossary Specific to this RFP

A.1 References Specific to this RFP

[CCM] CORBA Components Specification, http://www.omg.org/technology/documents/formal/components.htm

[DDS] Data Distribution Services Specification, http://www.omg.org/spec/DDS/1.2/

[~~D&C~~DEPL] Deployment and Configuration of Component-based Distributed Applications Specification OMG Available Specification, http://www.omg.org/spec/DEPL/4.0/

[RTC] Robotic Technology Component specification, http://www.omg.org/spec/RTC/1.0/

[SDO] Super distributed Object Specification, http://www.omg.org/spec/SDO/1.1/

A.2 Glossary Specific to this RFP

Robot application –A software application that controls a robot's behavior. Examples include a vacuum cleaning robot and a butler robot.

Super Distributed Object (SDO) – A logical representation of a hardware device or a software component that provides well-known functionality and services.

Robotic Technology Component (RTC) –A logical representation of a hardware and/or software entity that provides well-known functionality and services.

RTC-based system –A system comprised of RTCs connected in a network representing a robotic system, including robot hardware and software algorithms.

Robotic Technology (RT) – Robotic Technology (RT) is a general term of the technology originating in robotics, and it means not only the standalone robot but technical element which constitutes robots.

RT-component profile – A description that represents the static state of an RT Component that is referred to other RT Components.

RTC-based system profile - A description of how RT-components are connected and interact with each other, and RT-component configuration parameters.

Deployment profile - A description of information used in deploying components, including RT-component profiles.

Meta-information – Data that represents the properties of running RT component instance.

Directory – A storage that manages the references and the meta-information of running RT component instances.

Environment change – Situation that available resources in environment are changed such as sensors, actuators, and other robots, when a robotic system moves to new environment.

Deployment - all of the activities that make a set of components available for use and consist of installation and activation of the components.

Appendix B General Reference and Glossary

B.1 General References

The following documents are referenced in this document:

[ATC] Air Traffic Control Specification, *http://www.omg.org/technology/documents/formal/air_traffic_control.htm*

[BCQ] OMG Board of Directors Business Committee Questionnaire, *http://doc.omg.org/bc/07-08-06*

[CCM] CORBA Core Components Specification, *http://www.omg.org/technology/documents/formal/components.htm*

[CORBA] Common Object Request Broker Architecture (CORBA/IIOP), *http://www.omg.org/technology/documents/formal/corba_iiop.htm*

[CSIV2] [CORBA] Chapter 26

[CWM] Common Warehouse Metamodel Specification, *http://www.omg.org/technology/documents/formal/cwm.htm*

[DAIS] Data Acquisition from Industrial Systems, *http://www.omg.org/technology/documents/formal/dais.htm*

[EDOC] UML Profile for EDOC Specification, *http://www.omg.org/techprocess/meetings/schedule/UML_Profile_for_EDOC_FTF.html*

[EJB] "Enterprise JavaBeans™", *http://java.sun.com/products/ejb/docs.html*

[FORMS] "ISO PAS Compatible Submission Template". *http://www.omg.org/cgi-bin/doc?pas/2003-08-02*

[GE] Gene Expression, *http://www.omg.org/technology/documents/formal/gene_expression.htm*

[GLS] General Ledger Specification , *http://www.omg.org/technology/documents/formal/gen_ledger.htm*

[Guide] The OMG Hitchhiker's Guide,, *http://www.omg.org/cgi-bin/doc?hh*

[IDL] ISO/IEC 14750 also see [CORBA] Chapter 3.

[IDLC++] IDL to C++ Language Mapping, *http://www.omg.org/technology/documents/formal/c++.htm*

[Inventory] Inventory of Files for a Submission/Revision/Finalization, *http://doc.omg.org/smsc/2007-09-05*

[MDAa] OMG Architecture Board, "Model Driven Architecture - A Technical Perspective*”, http://www.omg.org/mda/papers.htm*

[MDAb] “Developing in OMG's Model Driven Architecture (MDA),” *http://www.omg.org/docs/omg/01-12-01.pdf*

[MDAc] “MDA Guide” (*http://www.omg.org/docs/omg/03-06-01.pdf*)

[MDAd] “MDA "The Architecture of Choice for a Changing World™"”, *http://www.omg.org/mda*

[MOF] Meta Object Facility Specification, *http://www.omg.org/technology/documents/formal/mof.htm*

[MQS] “MQSeries Primer”, *http://www.redbooks.ibm.com/redpapers/pdfs/redp0021.pdf*

[NS] Naming Service, *http://www.omg.org/technology/documents/formal/naming_service.htm*

[OMA] “Object Management Architecture™”, *http://www.omg.org/oma/*

[OTS] Transaction Service, *http://www.omg.org/technology/documents/formal/transaction_service.htm*

[P&P] Policies and Procedures of the OMG Technical Process*, http://www.omg.org/cgi-bin/doc?pp*

[PIDS] Personal Identification Service, *http://www.omg.org/technology/documents/formal/person_identification_service.htm*

[RAD] Resource Access Decision Facility, *http://www.omg.org/technology/documents/formal/resource_access_decision.htm*

[RFC2119] IETF Best Practices: Key words for use in RFCs to Indicate Requirement Levels, (*http://www.ietf.org/rfc/rfc2119.txt*).

[RM-ODP] ISO/IEC 10746

[SCA] Software Communications Architecture (SCA), http://sca.jpeojtrs.mil/sca.asp

[SDRP] Software Radio Components (SDRP), http://www.omg.org/spec/SDRP/

[SEC] CORBA Security Service, *http://www.omg.org/technology/documents/formal/security_service.htm*

[TOS] Trading Object Service, *http://www.omg.org/technology/documents/formal/trading_object_service.htm*

[UML] Unified Modeling Language Specification, *http://www.omg.org/technology/documents/formal/uml.htm*

[UMLC] UML Profile for CORBA, *http://www.omg.org/technology/documents/formal/profile_corba.htm*

[XMI] XML Metadata Interchange Specification, *http://www.omg.org/technology/documents/formal/xmi.htm*

[XML/Value] XML Value Type Specification, *http://www.omg.org/technology/documents/formal/xmlvalue.htm*

B.2 General Glossary

Architecture Board (AB) - The OMG plenary that is responsible for ensuring the technical merit and MDA-compliance of RFPs and their submissions.

Board of Directors (BoD) - The OMG body that is responsible for adopting technology.

Common Object Request Broker Architecture (CORBA) - An OMG distributed computing platform specification that is independent of implementation languages.

Common Warehouse Metamodel (CWM) - An OMG specification for data repository integration.

CORBA Component Model (CCM) - An OMG specification for an implementation language independent distributed component model.

Interface Definition Language (IDL) - An OMG and ISO standard language for specifying interfaces and associated data structures.

Letter of Intent (LOI) - A letter submitted to the OMG BoD's Business Committee signed by an officer of an organization signifying its intent to respond to the RFP and confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements.

Mapping - Specification of a mechanism for transforming the elements of a model conforming to a particular metamodel into elements of another model that conforms to another (possibly the same) metamodel.

Metadata - Data that represents models. For example, a UML model; a CORBA object model expressed in IDL; and a relational database schema expressed using CWM.

Metamodel - A model of models.

Meta Object Facility (MOF) - An OMG standard, closely related to UML, that enables metadata management and language definition.

Model - A formal specification of the function, structure and/or behavior of an application or system.

Model Driven Architecture (MDA) - An approach to IT system specification that separates the specification of functionality from the specification of the implementation of that functionality on a specific technology platform.

Normative – Provisions that one must conform to in order to claim compliance with the standard. (as opposed to non-normative or informative which is explanatory material that is included in order to assist in understanding the standard and does not contain any provisions that must be conformed to in order to claim compliance).

Normative Reference – References that contain provisions that one must conform to in order to claim compliance with the standard that contains said normative reference.

Platform - A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented.

Platform Independent Model (PIM) - A model of a subsystem that contains no information specific to the platform, or the technology that is used to realize it.

Platform Specific Model (PSM) - A model of a subsystem that includes information about the specific technology that is used in the realization of it on a specific platform, and hence possibly contains elements that are specific to the platform.

Request for Information (RFI) - A general request to industry, academia, and any other interested parties to submit information about a particular technology area to one of the OMG's Technology Committee subgroups.

Request for Proposal (RFP) - A document requesting OMG members to submit proposals to an OMG Technology Committee. Such proposals must be received by a certain deadline and are evaluated by the issuing Task Force.

Task Force (TF) - The OMG Technology Committee subgroup responsible for issuing a RFP and evaluating submission(s).

Technology Committee (TC) - The body responsible for recommending technologies for adoption to the BoD. There are two TCs in OMG – the *Platform TC* (PTC) focuses on IT and modeling infrastructure related standards; while the *Domain TC* (DTC) focuses on domain specific standards.

Unified Modeling Language (UML) - An OMG standard language for specifying the structure and behavior of systems. The standard defines an abstract syntax and a graphical concrete syntax.

UML Profile - A standardized set of extensions and constraints that tailors UML to particular use.

XML Metadata Interchange (XMI) - An OMG standard that facilitates interchange of models via XML documents.

Future of the Deployment and Configuration Specification

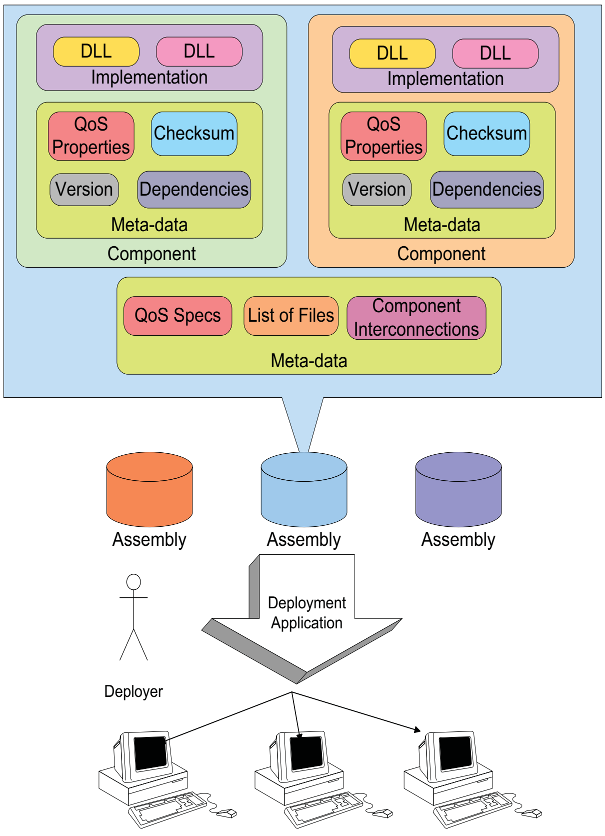
William R. Otte

wotte@dre.vanderbilt.edu

Vanderbilt University

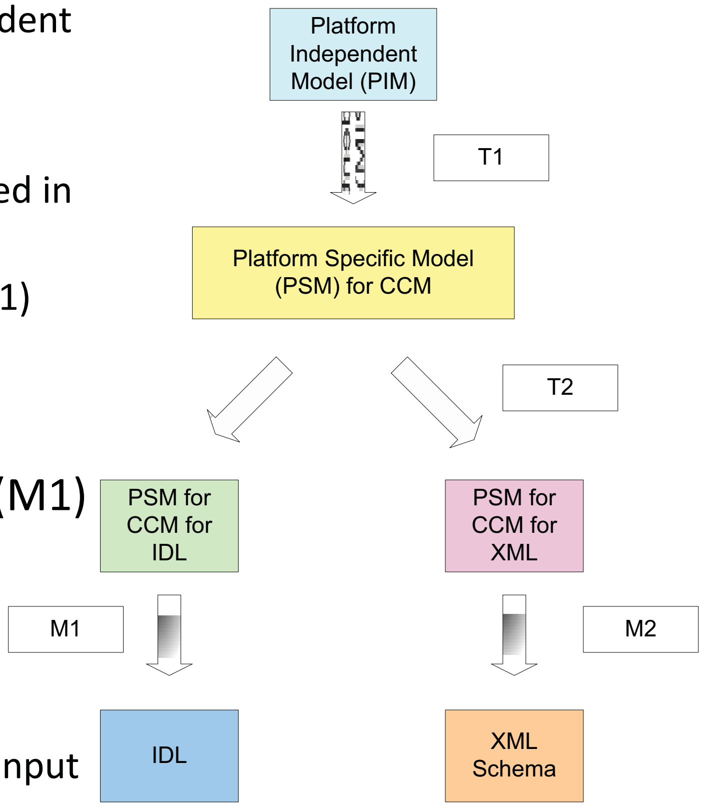
CCM Deployment & Configuration (D&C) Spec

- “D&C” spec was adopted by OMG in 2003
  - “Deployment & Configuration of Component-based Distributed Applications”
- Intended to replace *Packaging & Deployment* chapter of CCM (CORBA 3.0) specification
- Updated in 2005 to be an independent specification in it’s own right.
- Supports …
  - Resource management
  - QoS characteristics
  - Automated deployment
  - Vendor-independent deployment infrastructure



D&C & Model-Driven Architecture

- D&C is specified using a platform-independent model
  - Defines “deployment” model
  - Independent of CORBA & CCM (specified in UML)
- Can be refined into CCM-specific model (T1)
- Uses standard mappings to generate
  - IDL (for “on-line” data)
    - using UML Profile for CORBA (M1)
  - XML Schema (for “off-line” data)
    - using XMI (M2)
- Intermediate transformation T2
  - Transforms PSM for CCM into suitable input for M1 & M2



Platform Independent Model (PIM)

T1

Platform Specific Model (PSM) for CCM

T2

PSM for CCM for IDL

PSM for CCM for XML

M1

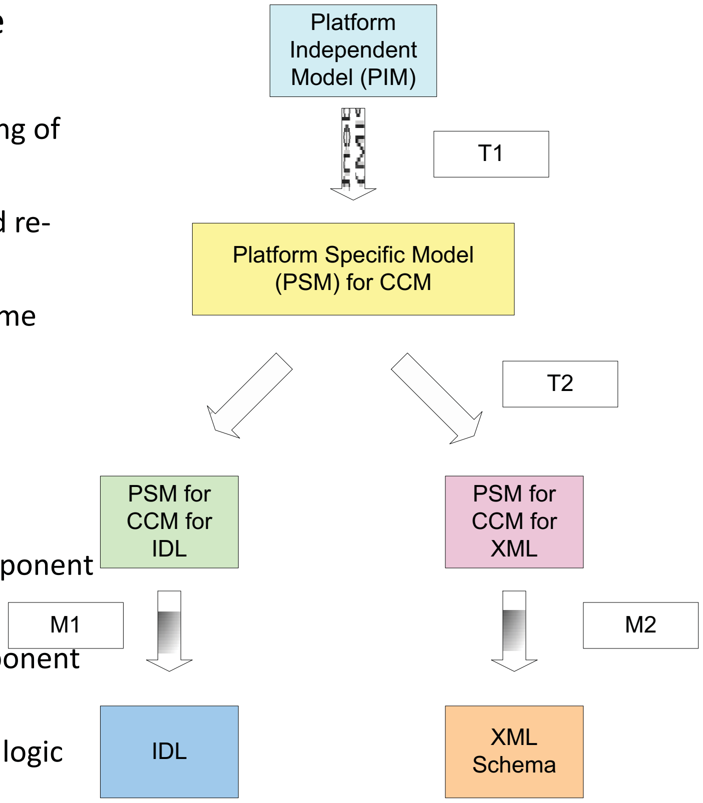
M2

IDL

XML Schema

Limitations of Existing D&C Standard

- Largely assumes a very “static” view of the deployment process
  - No standard functionality for run-time monitoring of existing deployments
  - No facility to provide mechanisms to modify and re-configure existing deployments
  - Limited facility to re-compose domains at run-time based on environmental changes
- Mapping the entire PIM to a “component model” is largely inflexible
  - Impossible to perform “mixed” component deployments consisting of several different component models
  - Represents a high barrier of entry for new component models
  - Limits the ability to re-use “generic deployment logic



Platform Independent Model (PIM)

T1

Platform Specific Model (PSM) for CCM

T2

PSM for CCM for IDL

PSM for CCM for XML

M1

M2

IDL

XML Schema

New D&C Design Goals (1/3)

- Separate *Generic Deployment Logic* from *Specific Deployment Logic*
- Provide a mechanism to customize *Generic Deployment Logic*
  - Preparing a *locality constrained* plan for deployment
  - Installing a generic *instance*
  - Creating connections in the context of a particular *Locality*
- Provide a mechanism to customize *Specific Deployment Logic*

By providing a standard interface through with *specific deployment logic* is implemented, we open the door varied *comunication & control* mechanisms

New D&C Design Goals (2/3)

- Provide a mapping to the *communication mechanism* rather than the *deployment target*
  - *Communication mechanism* is the distribution middleware used to disseminate meta-data and control at run-time
  - *Deployment target* is the component middleware/software intended to be deployed
- *Spefic deployment logic* could then be portable between D&C implementations.

Mappings for CORBA, DDS, or other domain specific communications mechanisms could be provided

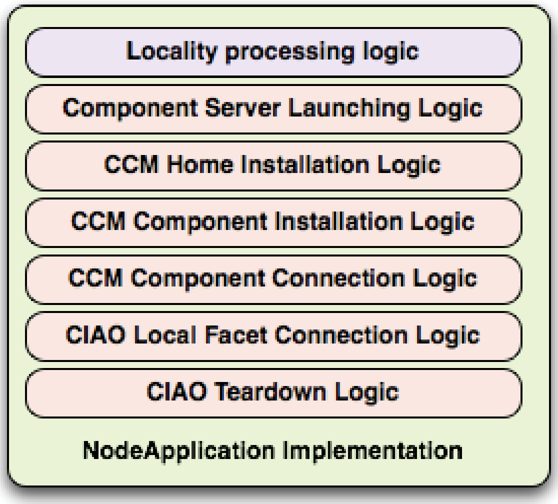
New D&C Design Goals (3/3)

- Extend the D&C interfaces to support dynamic aspects of deployment & configuration
  - Modifications of running deployments
  - Re-configuration of individual instances
  - Nodes/resources dynamically joining/leaving domains
  - Live monitoring of system health and status
- Make improvements to domain management & representation to improve scalability & applicability to grid/cloud computing

Mappings for CORBA, DDS, or other domain specific communications mechanisms could be provided

Where we are now (1/2)

- Current NodeApplication encourages stovepiped implementations
- Very tightly bound with installation and configuration of CIAO CCM entities
- D&C specification envisions creating new *NodeApplication* implementations to handle new/different CCM implementations
- Several *accidental* complexities make it difficult to maintain and extend
- Inherent complexities in the CCM D&C mapping make it difficult to apply to entirely new deployment scenarios



Locality processing logic

Component Server Launching Logic

CCM Home Installation Logic

CCM Component Installation Logic

CCM Component Connection Logic

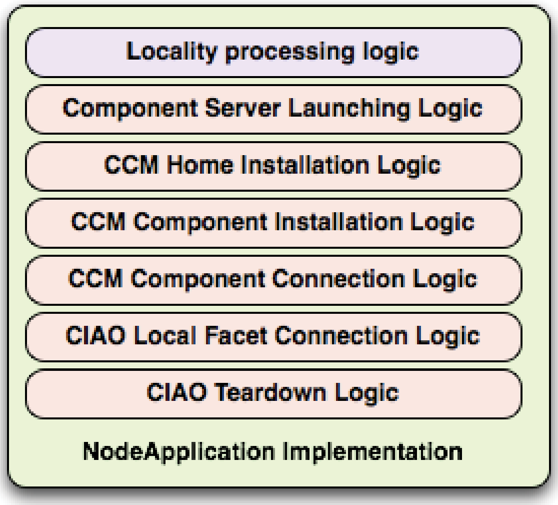
CIAO Local Facet Connection Logic

CIAO Teardown Logic

NodeApplication Implementation

Where we are now (2/2)

- *Accidental Complexities*
  - Generic deployment logic tightly bound with *CIAO Specific* deployment logic
  - CIAO/CCM Deployment logic tightly integrated
  - Individual *deployment steps* are tightly integrated and difficult to customize on a case-by-case basis
- *Inherent Complexities*
  - D&C specification envisions new NodeManager 'stacks' for different technology


Locality processing logic

Component Server Launching Logic

CCM Home Installation Logic

CCM Component Installation Logic

CCM Component Connection Logic

CIAO Local Facet Connection Logic

CIAO Teardown Logic

NodeApplication Implementation

Locality Manager (1/2)

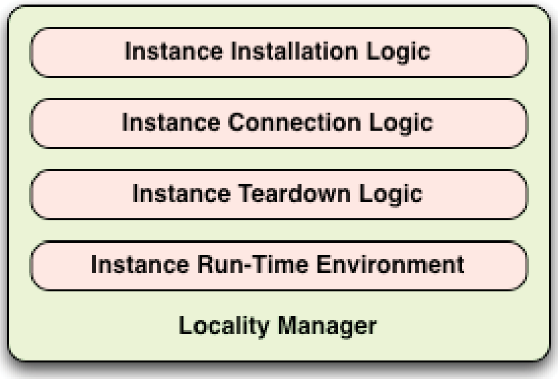
- D&C Specification can be viewed as a mechanism for establishing various *Localities* in different contexts
- A *Locality* is a grouping of installed instances in a particular context
  - Components & Homes installed in a *component server*
  - *Component servers* installed on a *Node*
  - *Nodes* installed in the context of a D&C *Domain*
- Individual *Localities* can have specific knowledge of how to install particular *instance types*
  - CIAO Components
  - CARDAMOM Components
  - EJB
  - Opaque *processes*

We would like to expand the horizons of D&C beyond a single component model, to be a generic deployment tool.

Locality Manager (2/2)

- Process of installing an *instance* can be generalized

1. Instantiate *a run time environment* (CIAO Container)
2. Interact with the run time environment to install *instances* (CIAO Components & Homes)
3. Interact with installed instances to create *connections*
4. Eventually, interact with the run time environment to *remove* an instance



Instance Installation Logic

Instance Connection Logic

Instance Teardown Logic

Instance Run-Time Environment

Locality Manager

These instance-specific steps can be provisioned with plug-ins to create a generic application server.

Locality Manager Interface

- Represents a fusion of three core D&C entities / concepts
  - Execution/Node Manager (i.e., preparePlan)
  - ApplicationManager (i.e., startLaunch)
  - Application (i.e., finishLaunch)
- Intended to fully manage a *locality constrained* portion of an application

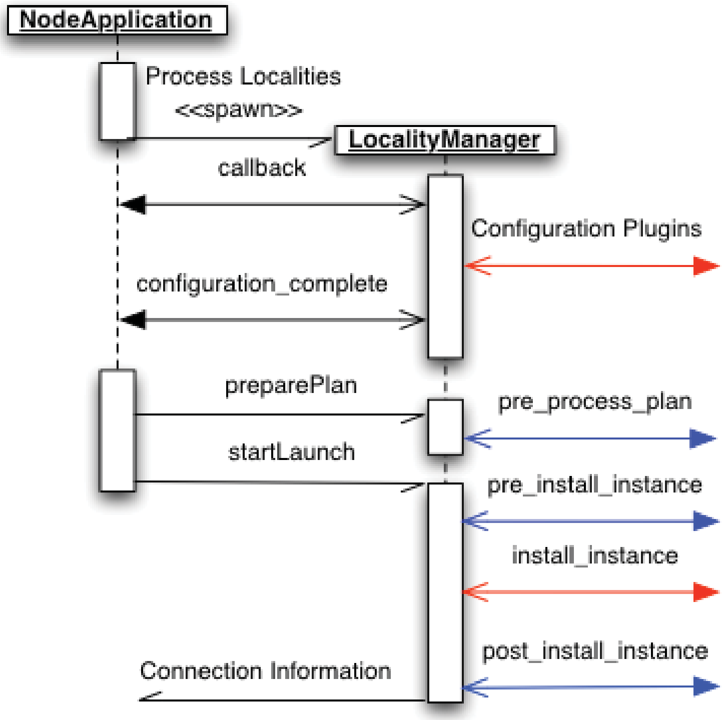
```
interface LocalityManager :
  Deployment::Application,
  Deployment::ApplicationManager
 {
  readonly attribute
     ::Deployment::Properties
     configuration;

  Deployment::ApplicationManager
    preparePlan (/*…*/);

  void destroyManager (/*…*/));

  oneway void shutdown ();
 };
```

(Some details omitted for brevity)

Keep in mind that “Locality Constrained” has different meanings depending on your context!

Phase 2 Locality Manager Implementation

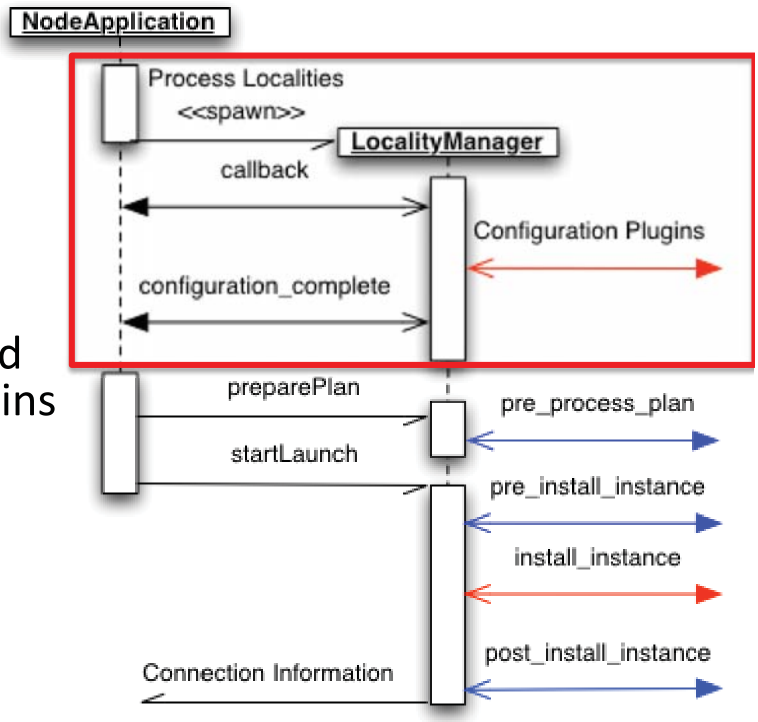
- Phase 2 will bootstrap the Locality Manager
- LocalityManager implementation will be statically configured with CCM-specific plug-ins
- NodeApplication will be modified to spawn Locality Manager processes
- Locality Manager processes will fully manage CCM-specific portion of deployment



Deployment Interceptors will be applied to existing code only on an as-needed basis

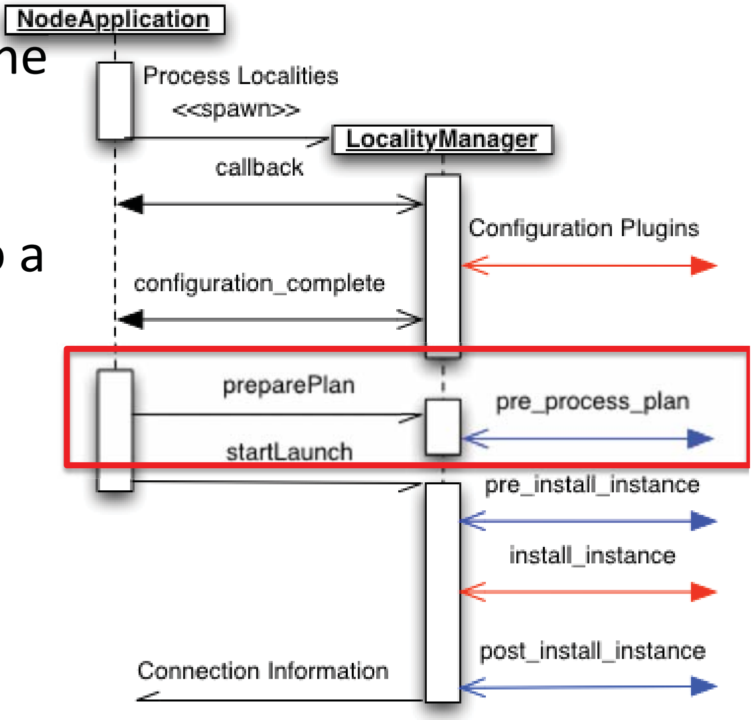
Locality Manager Startup

- Leverages existing CIAO component server startup procedure
- On startup, LM will call back to NodeApplication to receive configuration properties
- These properties will be provided to a series of configuration plug-ins
  - Instantiate CIAO Container
  - Configure process parameters
  - Etc.
- LM will notify NA when configuration is complete



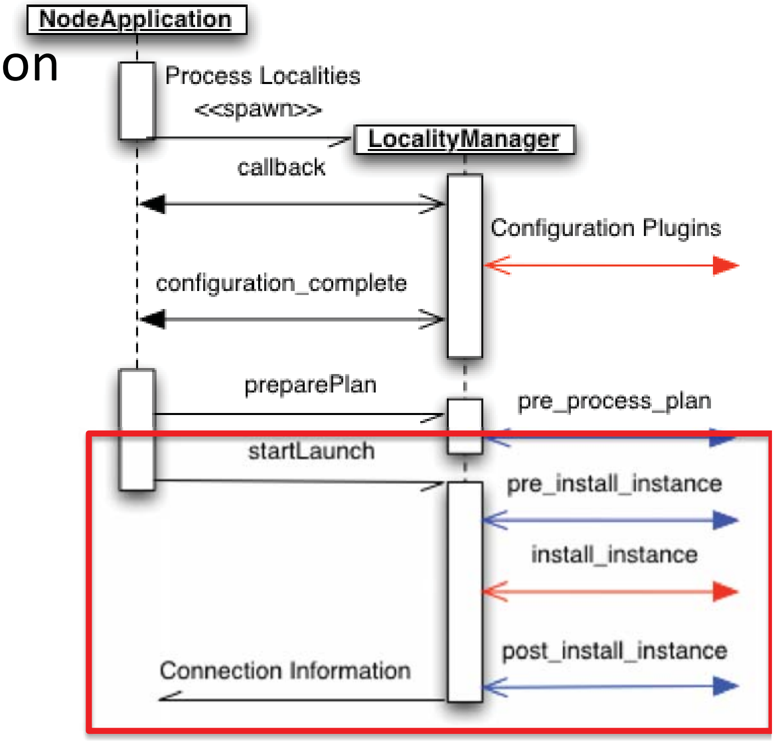
Locality Manager PreparePlan

- Node Application will split the plan into individual locality constrained plans
- Locality in this case refers to a single component server process
- Plan is provided to LM via preparePlan
- LM will invoke any installed plan-preprocessing deployment interceptors

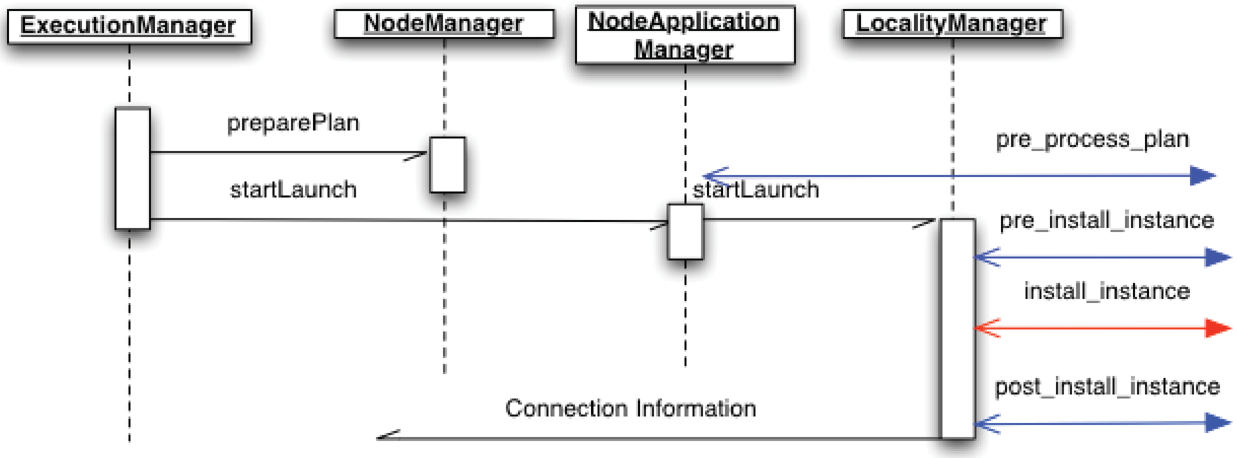


Locality Manager Start Launch

- NA will invoke startLaunch on all local LM processes



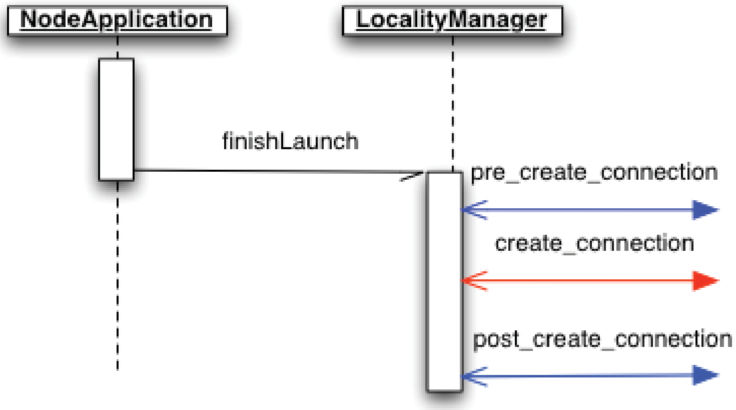
Global Start Launch Behavior (Phase 2)



- From a user perspective, the beginning of the deployment process will not change
- A plan pre-processing plug-in will be available to add LocalityManager instance information when needed

Locality Manager finishLaunch/start

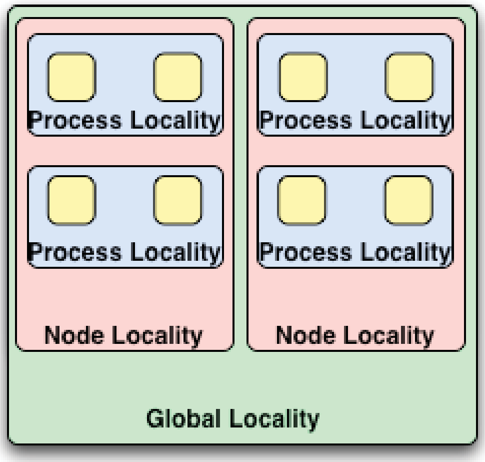
- NA will apportion connections necessary to each LM instance
- For each connection in it's local plan, the LM will invoke interception points and plug-in behavior to establish connections
- Behavior for application start is the same



In many ways, the NA/NAM will behave very much like the DA/DAM

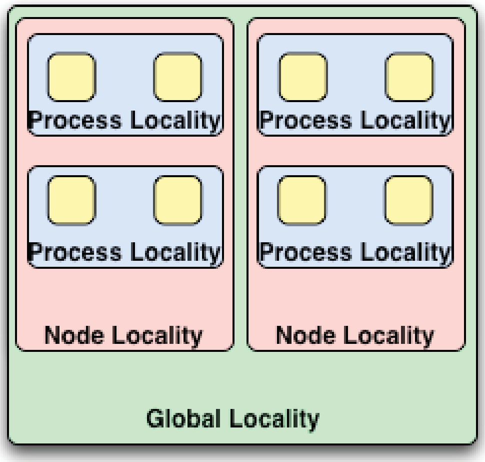
Phase 3 Locality Manager

- Global deployment plans can be viewed as hierarchical deployments of localities
  - "Global Locality" deploys instances of "Node Locality"
  - "Node Locality" deploys instances of "Process Locality"
  - "Process Locality" deploys component instances
- In this sense, EM/DAM/DA and NM/NAM/NA can be implemented in terms of the Locality Manager.
- Wrapper facades will provide traditional interfaces for standard compliance



Phase 3 Locality Manager

- Global deployment plans can be viewed as hierarchical deployments of localities
  - "Global Locality" deploys instances of "Node Locality"
  - "Node Locality" deploys instances of "Process Locality"
  - "Process Locality" deploys component instances
- In this sense, EM/DAM/DA and NM/NAM/NA can be implemented in terms of the Locality Manager.
- Wrapper facades will provide traditional interfaces for standard compliance



robotics/2010-06-09

– OMG Robotics DTF Infrastructure WG Meeting–

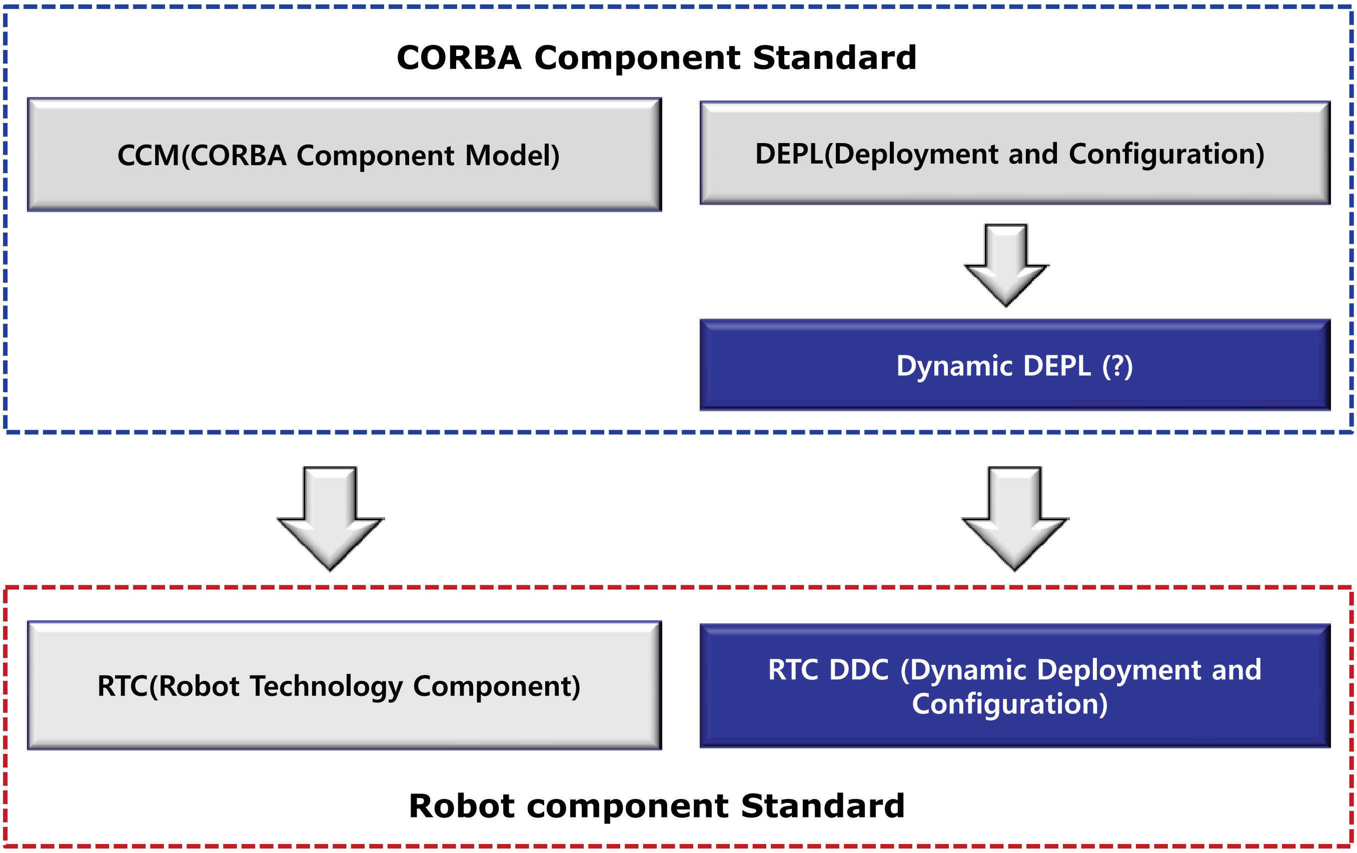
RTC Dynamic Deployment and Configuration(DDC) Specification

2010. 06

Seung-Woog Jung
Robot/Cognition System Research Department

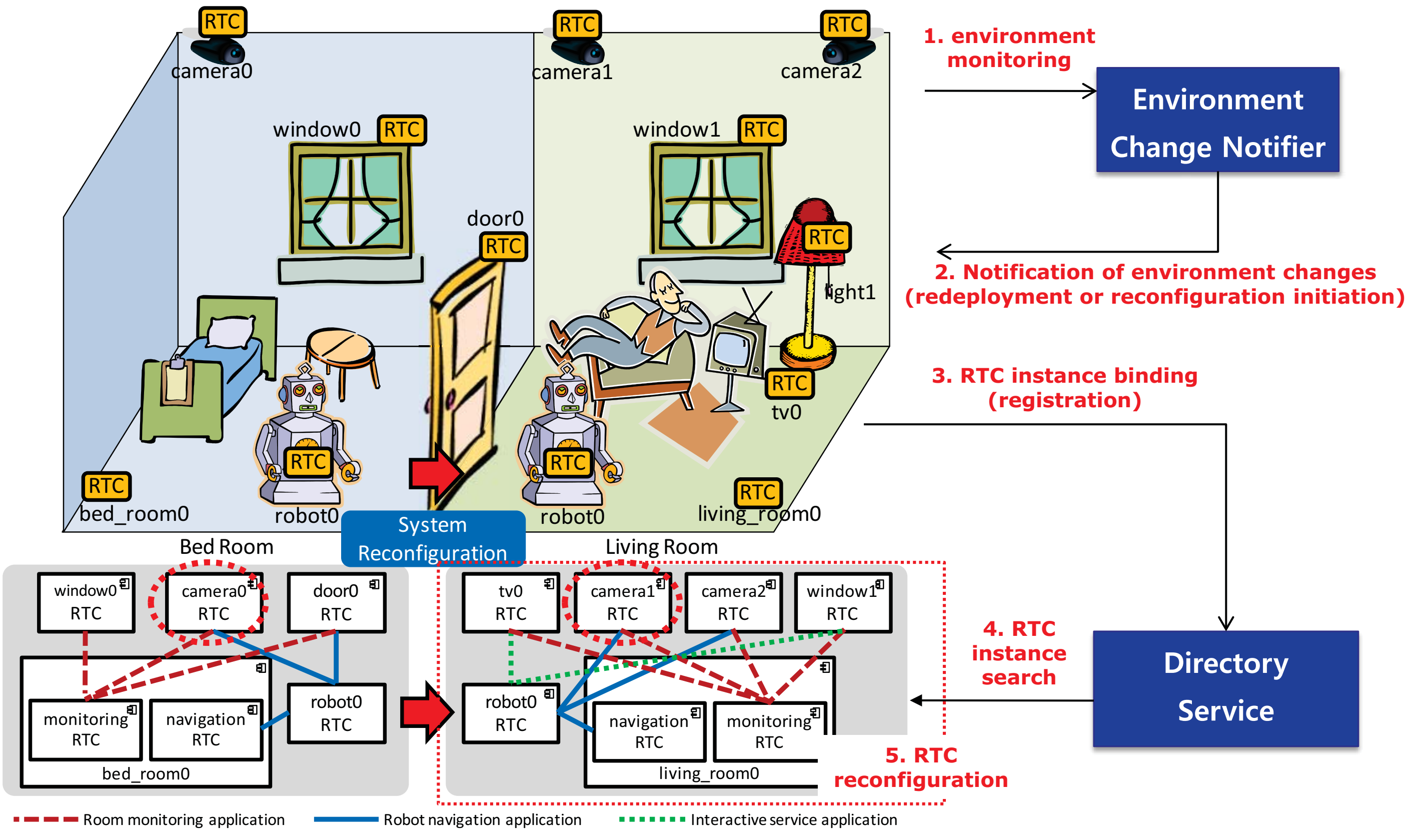
ETRI

Current Situation



CORBA Component Standard
CCM(CORBA Component Model)
DEPL(Deployment and Configuration)
Dynamic DEPL (?)
RTC(Robot Technology Component)
RTC DDC (Dynamic Deployment and Configuration)
Robot component Standard

RTC DDC RFP Scope



1. environment monitoring

Environment Change Notifier

2. Notification of environment changes (redeployment or reconfiguration initiation)

3. RTC instance binding (registration)

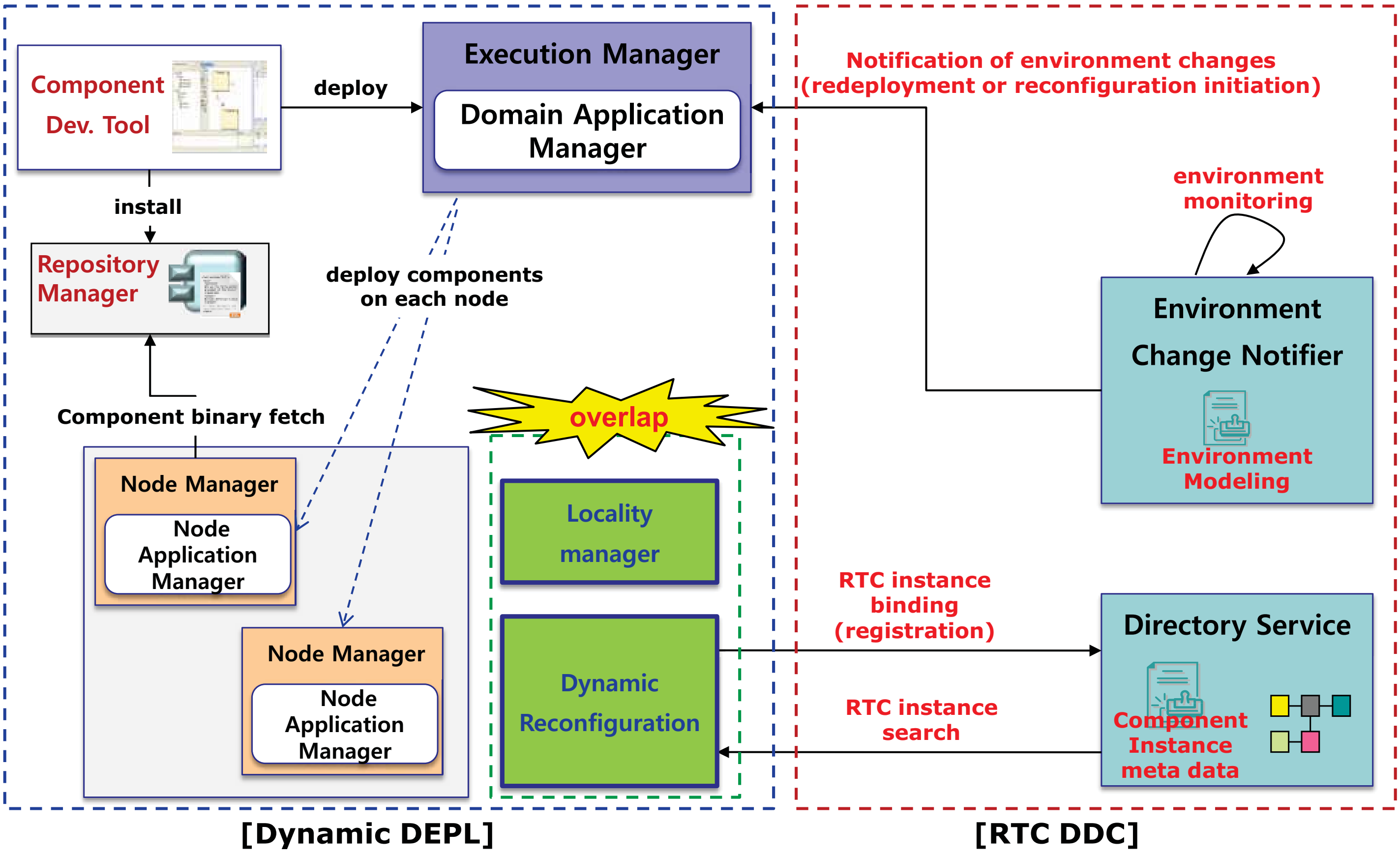
Directory Service

4. RTC instance search

5. RTC reconfiguration

System Reconfiguration

Dynamic DEPL vs RTC DDC



Component Dev. Tool

deploy

Execution Manager

Domain Application Manager

install

Repository Manager

deploy components on each node

Component binary fetch

Node Manager

Node Application Manager

Node Manager

Node Application Manager

overlap

Locality manager

Dynamic Reconfiguration

Notification of environment changes (redeployment or reconfiguration initiation)

environment monitoring

Environment Change Notifier

Environment Modeling

RTC instance binding (registration)

RTC instance search

Directory Service

Component Instance meta data

[Dynamic DEPL]

[RTC DDC]

Discussion Issues

RFP Process

- Propose I: We continue our RFP which includes minimal set of reconfiguration, and when dynamic DEPL be standardized we reflect it to our standard.
- Propose II: Work separately (Separate issues into two separate problems)
  - Approaches
    - Otte will newly propose "dynamic D&C" issue to the MARS
      - ETRI/AIST can use the specifications proposed from Dr. Otte.
    - ETRI/AIST will propose some features which are specific to the robot
      - In this case, we need to change the title that we've already proposed.
- Propose III: Work together
  - We can work together to make ONE proposal using current RFP.
  - In this case, while Otte may be in charge of dynamic DEPL, ETRI/AIST may be in charge of other issues

Discussion Issues

RFP Process

- Propose I: Work separately (Separate issues into two separate problems)
  - Two alternatives
    - In MARS
    - In Robotics DTF
  - Approaches
    - Otte will newly propose "dynamic D&C" issue to the MARS
      - ETRI/AIST can use the specifications proposed from Dr. Otte.
    - ETRI/AIST will propose dynamic D&C issues which are specific to the robot
      - In this case, we need to change the title that we've already proposed.
- We continue our RFP which includes minimal set of reconfiguration, and when dynamic DEPL be standardized we reflect it.
- Propose II: Work together
  - We can work together to make ONE proposal using current RFP.
  - In this case, while Otte may be in charge of dynamic DEPL, ETRI/AIST may be in charge of other issues

Discussion Issues

Checking Items

- Is it possible to make a major change on the current RFP?
- Is it possible to change the PTF proposal to the DTF one, even it's same issue?

RTC DDC Schedule

Event or Activity	Actual Date
Preparation of RFP by TF	
RFP placed on OMG document server	*February 22nd, 2010*
Approval of RFP by Architecture Board *Review by TC*	*March, 2010*
TC votes to issue RFP	*March, 2010*
LOI to submit to RFP due	**August 31, 2010**
Initial Submissions due and placed on OMG document server ("Four week rule")	**November 8th , 2010**
Voter registration closes	**December, 2010**
Initial Submission presentations	**December, 2010**
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Four week rule")	**May, 2011**
Revised Submission presentations	**June, 2011**
Final evaluation and selection by TF *Recommendation to AB and TC*	
Approval by Architecture Board *Review by TC*	
TC votes to recommend specification	**June, 2011**
BoD votes to adopt specification	**September, 2011**

Object Management Group

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

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

Request for Proposal

Robotic Interaction Service (RoIS) Framework RFP

OMG Document: robotics/2010-06-10

Letters of Intent due: September 13, 2010
Submissions due: November 8, 2010

Objective of this RFP

This RFP solicits proposals for a Platform Independent Model (PIM) and a Platform Specific Model (PSM) for robotic interaction service (RoIS) that specify

- common interfaces between robotic service applications and components that provide functions for performing human-robot interaction.
- data structures for each interface.

For further details see Chapter 6 of this document.

6.0 Specific Requirements on Proposals

6.1 Problem Statement

Service robot that provides some services to people in daily life will become more and more popular in robotics market. These s~~S~~ervice robots ~~interacts with human to~~ provide appropriate services through human-robot interaction (HRI). For example, there are robotic services such as,

- Reception service
- Guide service
- Home security service
- Childcare robot service
- Elder person daily watching service
- ~~etc.~~

Service application is provided as a set of robot behaviors. The robot behavior r~~R~~ule ~~of robot behaviors~~ is defined based on the information collected from humans or environments. The information is collected by using functions such as,

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

Generally, several sensors and actuators are equipped with the~~a~~ robot body or the environment where the robot provides the service~~s~~. T~~, and~~ he~~a~~ service application programmer describes procedures for~~of~~ robot ~~behavior~~ action and relation between the ~~behavior~~ action and the information -obtained by these~~the~~ sensors in the~~a~~ service application program. However,~~But~~ the various types of ~~the types of~~ sensors~~ and~~ and actuators are equipped with various robots, ~~s equipped with robots are usually different from each other~~ and~~,~~ moreover, ~~their~~ the application program interfaces (APIs)- of each robot are~~is~~ different by robot vendors even if their sensor types are the same. ~~That is~~Therefore, an application program developed for one specific robot will not run on the other robot. ~~and t~~This is one of the reasons of inefficiency in robot industry.

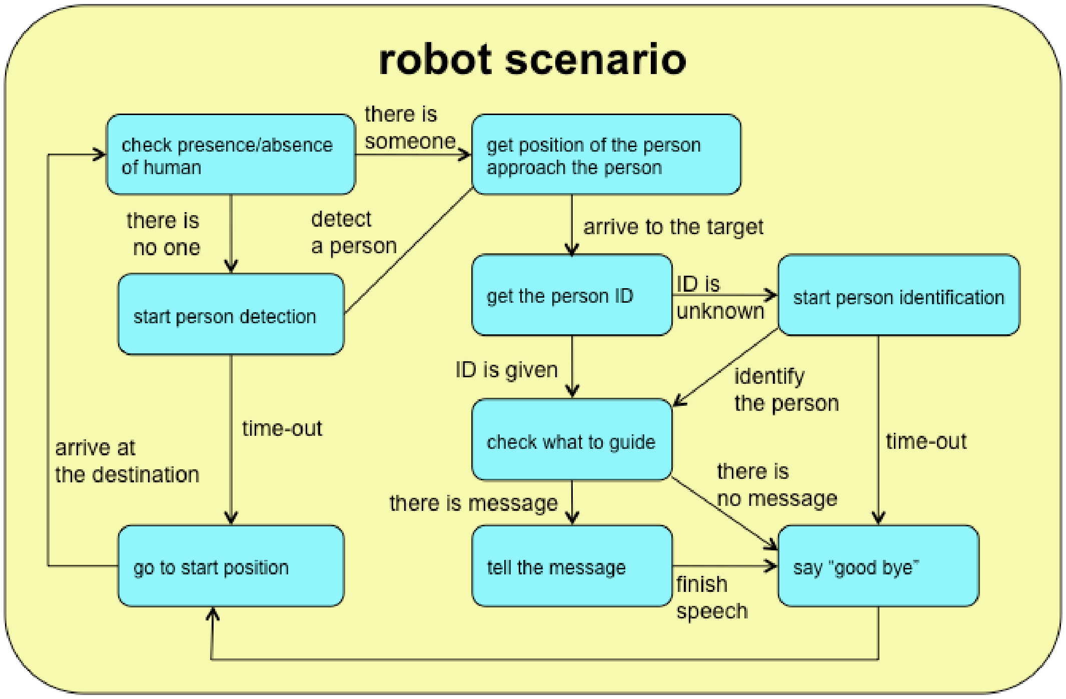


Fig.1: Example of robot scenario for robotic reception service.

In many service applications, the robot behavior rule is defined on a script, which is called a robot scenario. In the robot scenario, instructions to the functions and conditions based on the collected data are described in order to achieve the service task through the APIs that are specific to the robot. For example, the robotic reception service is constructed in the robot scenario as depicted in Fig. 1. In this scenario, when the robot detects someone, the robot approaches the person and tries to recognize who the person is, and then provides information appropriate to the person. Fig.2 shows the messages exchanged between the service application and the robot for the scenario. In this case, one robot (Robot 1) detects human by camera and move by wheels and another robot (Robot 2) detects human by RFID tag and move by legs. Because of the difference in the APIs, the service application programmer must write the different scenarios for each robot respectively.

~~For example in the case of robotic reception service, when a robot detects someone, it approaches the person and tries to recognize who is the person, and then guides the information appropriate to the person. One robot may detect human by camera and move by wheels. Another robot may detect human by RFID tag and move by legs. When there are two types of robots at the reception desks, service application programmer must write programs for each robot independently as depicted in Fig.1.~~

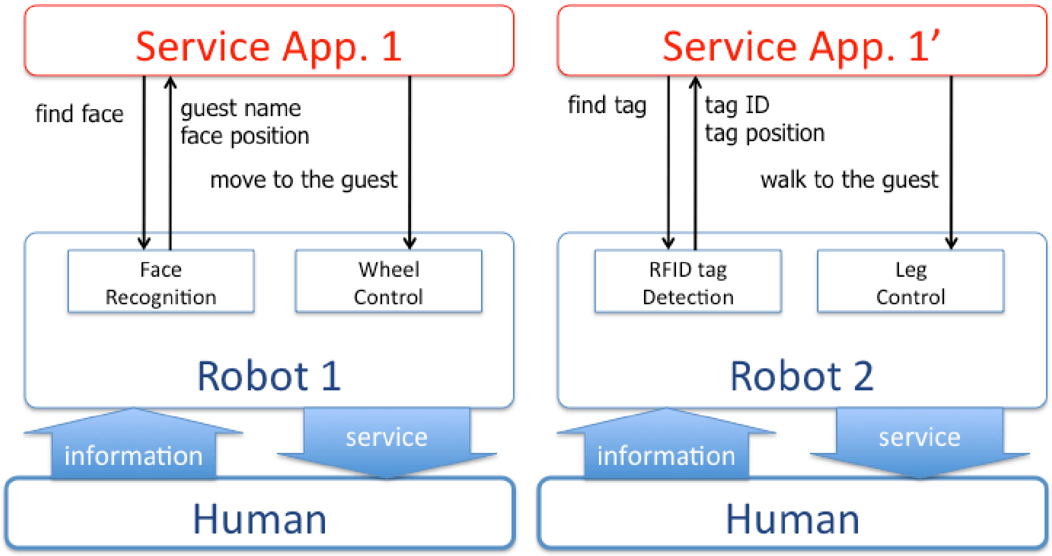


Fig.2~~1~~: Conventional style of service application programming. Service application programmer must write service application programs for each robot independently because functions provided by each robot are different.

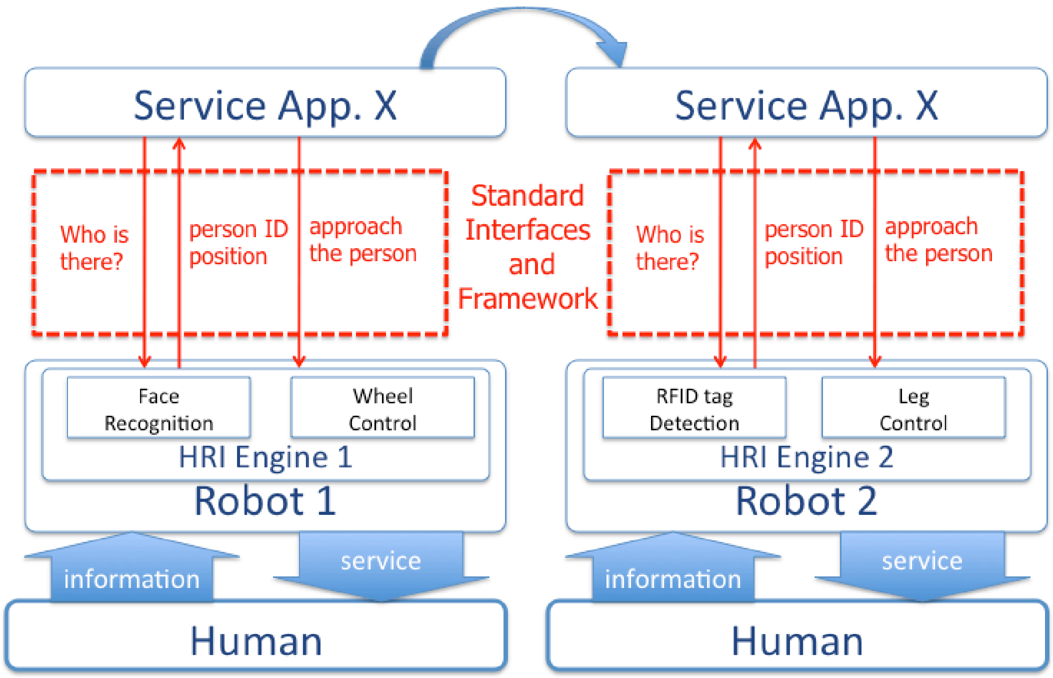


Fig.3~~2~~: RoIS service application programming style. The same service application program works on different robot platforms with little modification.

If all the functional components are encapsulated in a package and the interfaces for obtaining information and controlling robots are standardized, they will enhance reusability of service application programs. A service application

program for a robot can work on the other robots regardless of the different robot platforms as depicted in Fig.3~~2~~. Therefore, new general framework architecture is needed. We call the package that encapsulates functional components as HRI Engine and this new framework architecture as Robotic Interaction Service (RoIS).

For the RoIS concept, it is important to standardize mechanisms for the information and the instruction exchanged between the service application and the HRI Engine. From the point of view of a service application, there are generally two types of information to be exchanged, i.e. active information and passive information. The active information is a type of information that should be obtained when the service application needs the target information, such as “check presence/absence of human” and “get position of the person”. The passive information is a type of information that should be provided when the target information is obtained or changed in the HRI Engine, such as “the robot has detected a person” and “the robot has arrived at the target”.

In general, interface types of “Query” and “Event notification” are used for exchanging the active information and the passive information, respectively. To deal with “Event notification”, a mechanism for “Event subscription / cancellation” is also required for selecting appropriate event notifications on demand. In addition, interface type of “Command” is naturally needed in order to instruct the HRI Engine to control its functions, such as “approach the person” and “go to start position”. RoIS framework should also include these interface types, i.e., Event subscription/cancellation, Event notification, Query and Command as illustrated in Fig.4.

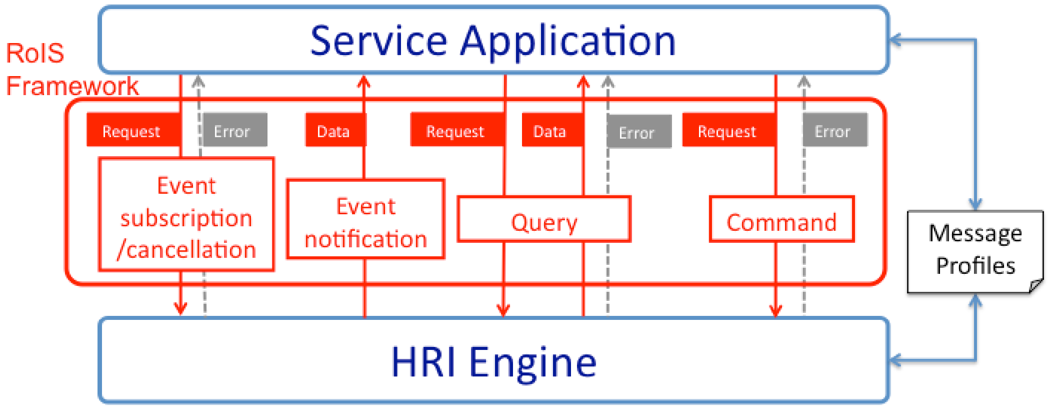


Fig.4: RoIS Framework. In the framework, application communicates with HRI Engine through Event(s) subscription/cancellation, Event(s) notification, Query and Command.

For RoIS framework, ~~must at least four types of interface: Event subscription/cancellation, Event notification, Query and Command as illustrated in Fig..~~

~~“Event subscription/cancellation” is an interface for service application to subscribe to HRI Engine for specific event type(s) or cancel the subscription(s). Available event types are obtained through the “Query” interface.~~

~~“Event notification” interface is required to notify service application of the occurrence of event(s) in real time. When a functional component in HRI Engine detects some status change, such as “a person appeared in a camera view,” “human voice recognized,” or “battery is running out,” then the component sends a specific event to service application through the “Event notification” interface when the application has subscribed to the event.~~

~~“Query” is used to obtain information actively from HRI Engine. Service application requires this interface type on HRI Engine to obtain the Engine’s capabilities (for example, what kind of information is available or what kind of event can be notified by the engine), current status of the Engine, and detailed information of the event notified. Each “Query” has its corresponding result(s).~~

~~“Command” is an interface type to control not only robot but also functional components in HRI Engine. Controlling robot actuators, changing sampling frequency of sensors, and replacing functional components in HRI Engine are the examples of “Commands” to be used.~~

i~~I~~t is desirable to define common messages~~interface items~~ of each interface type that are ~~available~~ suitable for all HRI Engines as possible. These common messages should be proposed at least for person detection, person identification, and person location. There are common messages for each interface type such as,

- ▶ Event notification
  - Person detected
  - Person identified
  - Robot action completed
- ▶ Query
  - Person ID
  - Position of person / robot
  - Status of HRI Engine
- ▶ Command
  - Start / stop detection
  - Start / stop identification

In regard to these common messages, there must be a unique structure for RoIS framework.

On the other hand, there are also messages specific to each HRI Engine. For example in guidance service, there may be specific messages for each interface type such as, event notification for “speech recognition”, query for “presence / absence of human” and command for “approach the person” in addition to the common messages. -Therefore, each HRI Engine should provide its message profiles with service applications.

Message profiles are composed of common message types and domain specific message types, for example, educational domain, navigation domain and healthcare domain, and each profile may include a list of available messages for each interface type, name of each message, data format for the information exchanged through the message and required argument(s) for using the message. It is required not only to define common messages but also to make a scheme to describe these message profiles so that the service application can make a query about the appropriate message as necessary.

To specify RoIS framework, it is desirable to consider an abstraction level of the messages for the interface types. For example, it should not include~~it should be appreciated that a command must not use~~ ~~some~~ any parameters ~~based on a~~specific to robot platform. Also, the information managed by "Event subscription/cancelation," "Event notification," ~~and~~ "Query" and "Command" must be represented ~~at the~~ suitable ~~level~~ for service applications. The abstraction level appropriate to human robot interaction should be considered carefully by focusing on contents of the information and the robot control unique to human robot interaction. ~~In regard to this point, there must be a unique structure for RoIS framework.~~

~~As for the messages to be exchanged through these interfaces, that is, the data specified and exchanged through queries or by events, and commands sent to control robotic systems, these can be classified into two categories; one for messages that are common to every HRI engine and the other for messages specific to certain service application domains. As services provided by robots may be applied to a variety of domains related to our daily activity such as route guidance or elderly care, it is much efficient to have dictionaries or profiles of messages specific to each domain rather than using a huge set of messages that covers all domains. On the other hand, there are messages that are commonly used in most of the applications that are typical to human-robot interaction. Therefore, the framework shall include the facility to define and to choose message profiles, and at the same time, shall specify profiles that contain basic common messages.~~

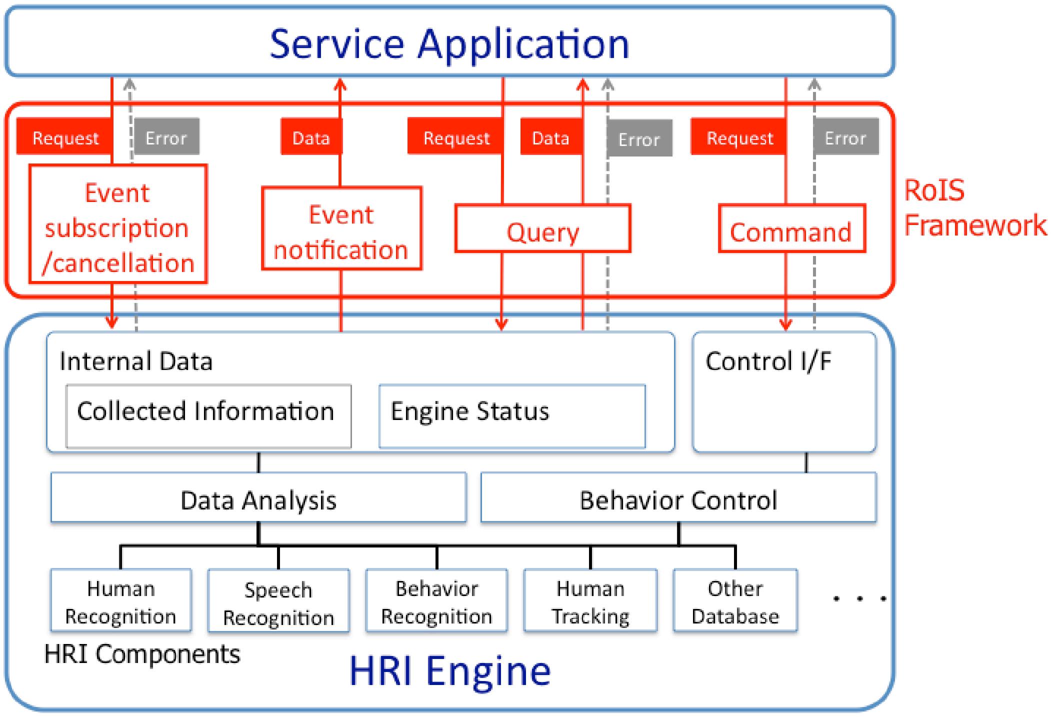


Fig.54: Example of RoIS Framework

Fig.54 illustrates a basic structure of RoIS Framework. HRI Engine integrates several functional components (HRI Components) and provides their functions with a service application through standardized interface. It collects and manages information of human around and environment by sensors, and provides collected data with the application on demand. As some HRI Components in HRI Engine provide robot control functions, the application can control robot through the engine's interface.

Separation and encapsulation of HRI Components into HRI Engine and providing standardized interface to the components will enhance not only the reusability of components but also the efficiency of service application and component development.

Therefore, the scope of the specification solicited includes the definition of interfaces in between service applications and HRI Engine, that is, Event subscription/cancellation, Event notification, Query and Command, and the structure of data transmitted through each interface. Error notification sent from HRI Engine to application may also be included. However, internal data structure that depends on each HRI Engine implementation or data structures defined in existing specifications such as user identification should not be included in this specification.

Considering that HRI Engines depend on their robot platforms, the HRI Engine developers should be able to define interfaces between functional components inside their engine in their independent way. RoIS framework should not concern about HRI Engine inside. For example, one developer can use some other standardized framework, such as RTM, for inside HRI Engine, and the other developer can use their original method. Also, HRI Engine can access to the other applications and databases, such as location data and map data for path generation, by using other framework as needed. The same can be said for service application program inside. For example, in the case that the service application has to be corporate with the other application, such as network robot service, the service application can use other framework to access to the other application.

In summary, following items are required for RoIS framework.

- ~~**Event(s) subscription/cancellation**~~
    - ~~Subscribe to specific event(s) and cancel subscription for specific event(s)~~
    - ~~Sent from application to HRI Engine~~
- ~~**Event(s) notification**~~
    - ~~Notify the occurrence of event to subscriber(s)~~
    - ~~Sent from HRI Engine to application~~
- ~~**Query**~~
    - ~~Retrieve detailed information of events notified by HRI Engine~~
    - ~~Sent from application to HRI Engine (i.e. *requests*) and from HRI Engine to application (i.e. *results*)~~
- ~~**Command**~~
    - ~~Give commands to a robot, components of HRI Engine and/or the engine itself~~

~~Sent from application to HRI Engine~~

- **Interface between service application and HRI Engine**
    - Interface to obtain information from HRI Engine according to the timing of the service application's needs
(Query)
    - Interface to receive information from HRI Engine according to the occurrences of the information in real time
(Event notification / subscription / cancellation)
    - Interface for the instruction to control HRI Engine functions
(Command)
- **Definition of common messages for all HRI Engines**

6.2 Scope of Proposals Sought

This RFP seeks proposals that specify RoIS framework, on top of which various robotic service applications are developed. This RFP targeted on human-robot interaction.~~It is the target for service robots interacting with human.~~

It is necessary to consider the following~~s~~ in the specification of RoIS framework.

(1) Overview of~~all~~ architecture that consists of RoIS framework, a robotic service application, HRI Engine that is a set of robotic components ~~for RoIS framework~~ shall be ~~defined~~ provided (diagram or description for overview).

(2) The RoIS framework specification shall provide following interfaces between robotic service applications and HRI Engine. ~~that is a set of robotic components~~

- Interface to obtain information actively from HRI Engine
- Interface to receive information timely from HRI Engine
- Interface for the instruction to control HRI Engine functions

~~(2)~~(3) The RoIS framework specification must be general enough to incorporate robotic components for various sensors, actuators and algorithms in HRI Engine.

~~(3)~~(4) The RoIS framework specification shall satisfy interoperability and reusability. A HRI Engine should be able to be replaced with the other HRI engine with little efforts.

~~(4) The RoIS framework specification shall provide a minimum set of functionalities to satisfy the followings:~~

~~1. Interface types between robotic service applications and HRI Engine~~

~~Event subscription and cancellation~~

~~Event notifications~~

~~Query~~

~~Command~~

~~2. Data structure for each interface type~~

~~(5) The RoIS framework specification shall provide a scheme to manage profile of each interface type~~

~~(6)~~(5) The RoIS Framework specification shall specify ~~profiles of message and those of~~ common messages for all HRI Engines.~~each interface type~~

6.3 Relationship to Existing OMG Specifications

Submitters shall examine the following OMG specifications for possible benefit:

- Super Distributed Objects (SDO) Specification version 1.1 [formal/2008-10-11]

- Unified Modeling Language: Infrastructure version 2.1.2 [formal/2007-11-04]
- Unified Modeling Language: Superstructure version 2.1.2 [formal/2007-11-02]
- Lightweight CORBA Component Model 4.0 [formal/2006-04-01]
- Robotic Technology Component specification version 1.0 [formal/08-04-04]
- Robotic Localization Service version 1.0 [formal/2010-02-03]

6.4 Related Activities, Documents and Standards

Proposals may include existing systems, documents, user recognition service interface, and standards that are relevant to the problems discussed in this RFP. They can be used as background information for the proposal.

Example:

- IEEE Robotics and Automation Society, Technical Committee on Network Robot
- IEEE Robotics and Automation Society, Technical Committee on Programming Environment in Robotics and Automation
- ISO/ SC 37 Projects relate to ISO/IEC 19784-1(BioAPI Ver 2.0)
- ISO/TC184/SC2

6.5 Mandatory Requirements

(1) Proposals shall provide overview of~~all~~ architecture that consists of RoIS framework, a robotic service application, HRI Engine that is a set of robotic components ~~for RoIS framework~~ (diagram or description for overview)

~~(1)~~

~~(2) Proposals shall provide a Platform Independent Model (PIM) and at least one Platform Specific Model (PSM) of RoIS framework.~~

~~(3)~~(2)Proposals shall specify a general mechanism for RoIS framework

- Interfaces between applications and HRI Engine and their types:
  - ▶ Interface to obtain information actively from HRI Engine
  - ▶ Interface to receive information timely from HRI Engine
  - ▶ Interface for the instruction to control HRI Engine functions

- Data structure for each interface type

(3) Proposals shall specify existing technologies to achieve functions required for RoIS framework.

(4) Proposal shall specify functions that cannot be achieved by existing technologies.

- ~~The RoIS framework specification shall provide a scheme to manage profile of each interface type~~

(5) ~~The RoIS Framework specification~~Proposals shall specify ~~profiles of message and those of~~ common messages for ~~each interface type~~all HRI Engines

(6) Proposals shall provide a Platform Independent Model (PIM).

6.6 Optional Requirements

(1) Proposals may specify schemes for the functionalities listed below.

- Error handling for each interface type
- Returning command results and status

(2) Proposals may provide a schema to describe message profiles.

(3) Proposals may provide a Platform Specific Model (PSM) as C++, CORBA-specific model, XML or RTC for RoIS framework.

6.7 Issues to be discussed

These issues will be considered during submission evaluation. They should not be part of the proposed normative specification. (Place them in Part I of the submission.)

- Proposals shall demonstrate its feasibility by using a specific application based on the proposed model.
- Proposals shall demonstrate its applicability to existing robotic interaction service technologies.
- Proposals shall discuss simplicity of implementation.
- Proposals shall discuss the possibility to apply the proposed model to other fields of interest such as intelligent service robot applications

- Proposals shall specify on-the-wire protocol communication technology independent.
- Proposals shall discuss their relation and dependency to existing communication protocols or middlewares standards, such as CORBA [CORBA], DDS [DDS] or RTC [RTC].
- Proposals shall discuss the generality with respect to various sensors, actuators and algorithms in HRI Engine.

6.8 Evaluation Criteria

Proposals will be evaluated in terms of consistency in their specifications, feasibility and versatility across a wide range of different robot applications.

6.9 Other information unique to this RFP

None

6.10 RFP Timetable

The timetable for this RFP is given below. Note that the TF or its parent TC may, in certain circumstances, extend deadlines while the RFP is running, or may elect to have more than one Revised Submission step. The latest timetable can always be found at the OMG *Work In Progress* page at http://www.omg.org/schedules/ under the item identified by the name of this RFP. Note that "<month>" and "<approximate month>" is the name of the month spelled out; e.g., January.

Event or Activity	Actual Date
Preparation of RFP by TF	**24. May. 2010**
RFP placed on OMG document server	**24. May. 2010**
Approval of RFP by Architecture Board *Review by TC*	**21. June. 2010**
TC votes to issue RFP	**25. June. 2010**
LOI to submit to RFP due	**13. September. 2010**
Initial Submissions due and placed on OMG document server ("Three week rule")	**8. November. 2010**
Voter registration closes	**29, November, 2010**
Initial Submission presentations	**6. December. 2010**
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Three week	

rule”)	
Revised Submission presentations	
Final evaluation and selection by TF *Recommendation to AB and TC*	
Approval by Architecture Board *Review by TC*	
TC votes to recommend specification	
BoD votes to adopt specification	

Appendix A References and Glossary Specific to this RFP

A.1 References Specific to this RFP

[RTC] Robotic Technology Component specification version 1.0, http://www.omg.org/spec/RTC/1.0/

A.2 Glossary Specific to this RFP

None

Appendix B General Reference and Glossary

B.1 General References

The following documents are referenced in this document:

[ATC] Air Traffic Control Specification, *http://www.omg.org/technology/documents/formal/air_traffic_control.htm*

[BCQ] OMG Board of Directors Business Committee Questionnaire, *http://www.omg.org/cgi-bin/doc?bc/02-02-01*

[CCM] CORBA Core Components Specification, *http://www.omg.org/technology/documents/formal/components.htm*

[CORBA] Common Object Request Broker Architecture (CORBA/IIOP), *http://www.omg.org/technology/documents/formal/corba_iiop.htm*

[CSIV2] [CORBA] Chapter 26

[CWM] Common Warehouse Metamodel Specification, http://www.omg.org/technology/documents/formal/cwm.htm

[DAIS] Data Acquisition from Industrial Systems, http://www.omg.org/technology/documents/formal/dais.htm

[EDOC] UML Profile for EDOC Specification, http://www.omg.org/techprocess/meetings/schedule/UML_Profile_for_EDOC_FTF.html

[EJB] "Enterprise JavaBeans™", http://java.sun.com/products/ejb/docs.html

[FORMS] "ISO PAS Compatible Submission Template". http://www.omg.org/cgi-bin/doc?pas/2003-08-02

[GE] Gene Expression, http://www.omg.org/technology/documents/formal/gene_expression.htm

[GLS] General Ledger Specification , http://www.omg.org/technology/documents/formal/gen_ledger.htm

[Guide] The OMG Hitchhiker's Guide,, http://www.omg.org/cgi-bin/doc?hh

[IDL] ISO/IEC 14750 also see [CORBA] Chapter 3.

[IDLC++] IDL to C++ Language Mapping, http://www.omg.org/technology/documents/formal/c++.htm

[MDAa] OMG Architecture Board, "Model Driven Architecture - A Technical Perspective", http://www.omg.org/mda/papers.htm

[MDAb] "Developing in OMG's Model Driven Architecture (MDA)," http://www.omg.org/docs/omg/01-12-01.pdf

[MDAc] "MDA Guide" (http://www.omg.org/docs/omg/03-06-01.pdf)

[MDAd] "MDA "The Architecture of Choice for a Changing World™"", http://www.omg.org/mda

[MOF] Meta Object Facility Specification, http://www.omg.org/technology/documents/formal/mof.htm

[MQS] "MQSeries Primer", http://www.redbooks.ibm.com/redpapers/pdfs/redp0021.pdf

[NS] Naming Service, http://www.omg.org/technology/documents/formal/naming_service.htm

[OMA] "Object Management Architecture™", http://www.omg.org/oma/

[OTS] Transaction Service, *http://www.omg.org/technology/documents/formal/transaction_service.htm*

[P&P] Policies and Procedures of the OMG Technical Process*, http://www.omg.org/cgi-bin/doc?pp*

[PIDS] Personal Identification Service, *http://www.omg.org/technology/documents/formal/person_identification_service.htm*

[RAD] Resource Access Decision Facility, *http://www.omg.org/technology/documents/formal/resource_access_decision.htm*

[RFC2119] IETF Best Practices: Key words for use in RFCs to Indicate Requirement Levels, (*http://www.ietf.org/rfc/rfc2119.txt*).

[RM-ODP] ISO/IEC 10746

[SEC] CORBA Security Service, *http://www.omg.org/technology/documents/formal/security_service.htm*

[TOS] Trading Object Service, *http://www.omg.org/technology/documents/formal/trading_object_service.htm*

[UML] Unified Modeling Language Specification, *http://www.omg.org/technology/documents/formal/uml.htm*

[UMLC] UML Profile for CORBA, *http://www.omg.org/technology/documents/formal/profile_corba.htm*

[XMI] XML Metadata Interchange Specification, *http://www.omg.org/technology/documents/formal/xmi.htm*

[XML/Value] XML Value Type Specification, *http://www.omg.org/technology/documents/formal/xmlvalue.htm*

B.2 General Glossary

Architecture Board (AB) - The OMG plenary that is responsible for ensuring the technical merit and MDA-compliance of RFPs and their submissions.

Board of Directors (BoD) - The OMG body that is responsible for adopting technology.

Common Object Request Broker Architecture (CORBA) - An OMG distributed computing platform specification that is independent of implementation languages.

Common Warehouse Metamodel (CWM) - An OMG specification for data repository integration.

CORBA Component Model (CCM) - An OMG specification for an implementation language independent distributed component model.

Interface Definition Language (IDL) - An OMG and ISO standard language for specifying interfaces and associated data structures.

Letter of Intent (LOI) - A letter submitted to the OMG BoD's Business Committee signed by an officer of an organization signifying its intent to respond to the RFP and confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements.

Mapping - Specification of a mechanism for transforming the elements of a model conforming to a particular metamodel into elements of another model that conforms to another (possibly the same) metamodel.

Metadata - Data that represents models. For example, a UML model; a CORBA object model expressed in IDL; and a relational database schema expressed using CWM.

Metamodel - A model of models.

Meta Object Facility (MOF) - An OMG standard, closely related to UML, that enables metadata management and language definition.

Model - A formal specification of the function, structure and/or behavior of an application or system.

Model Driven Architecture (MDA) - An approach to IT system specification that separates the specification of functionality from the specification of the implementation of that functionality on a specific technology platform.

Normative – Provisions that one must conform to in order to claim compliance with the standard. (as opposed to non-normative or informative which is explanatory material that is included in order to assist in understanding the standard and does not contain any provisions that must be conformed to in order to claim compliance).

Normative Reference – References that contain provisions that one must conform to in order to claim compliance with the standard that contains said normative reference.

Platform - A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented.

Platform Independent Model (PIM) - A model of a subsystem that contains no information specific to the platform, or the technology that is used to realize it.

Platform Specific Model (PSM) - A model of a subsystem that includes information about the specific technology that is used in the realization of it on a specific platform, and hence possibly contains elements that are specific to the platform.

Request for Information (RFI) - A general request to industry, academia, and any other interested parties to submit information about a particular technology area to one of the OMG's Technology Committee subgroups.

Request for Proposal (RFP) - A document requesting OMG members to submit proposals to the OMG's Technology Committee. Such proposals must be received by a certain deadline and are evaluated by the issuing task force.

Task Force (TF) - The OMG Technology Committee subgroup responsible for issuing a RFP and evaluating submission(s).

Technology Committee (TC) - The body responsible for recommending technologies for adoption to the BoD. There are two TCs in OMG – *Platform TC* (PTC), that focuses on IT and modeling infrastructure related standards; and *Domain TC* (DTC), that focus on domain specific standards.

Unified Modeling Language (UML) - An OMG standard language for specifying the structure and behavior of systems. The standard defines an abstract syntax and a graphical concrete syntax.

UML Profile - A standardized set of extensions and constraints that tailors UML to particular use.

XML Metadata Interchange (XMI) - An OMG standard that facilitates interchange of models via XML documents.

A Standards Based Architecture Using JAUS and RTC

Laurent Rioux Ph.D, Charles R. Robinson Ph.D,

[1] THALES Research And Technology
Campus Polytechnique
1, Av Augustin Fresnel
91767 PALAISEAU Cedex,
{Laurent.Rioux, Charles.Robinson}@thalesgroup.com

Abstract. The Joint Architecture for Unmanned Systems (JAUS) and Robotic Technology Component (RTC) standards represent two frameworks that have evolved independently, but they both share the goal of modularising the development and structure of machines. Where RTC addresses this with a fine granularity of algorithmic interaction, or components, JAUS defines a message-based protocol for interaction on the macro scale between and within unmanned vehicle platforms, operator control units and payloads. The purpose of this paper is to illustrate the suitability of developing autonomous systems using the combined strengths of the RTC and JAUS standards and to provide a solution to meet compliance with both standards.

Keywords: Unmanned Systems, Joint Architecture, JAUS, OMG RTC Standard, Robotics, Interoperability, Middleware.

1 Introduction

This paper describes the advantageous union between two enabling technologies for autonomous systems, the Joint Architecture for Unmanned Systems (JAUS) and the OMG standard for Robotic Technology Components (RTCs). Over the following sections, the RTC and JAUS standards are described and then there is a discussion of their complementary attributes that make them ideal to be used in an integrated approach for autonomous system development.

2 RTC: Robotics Technology Component (An OMG Standard)

When one talks about the OMG RTC standard, one refers to an approach for developing robotic technology in the form of components that may represent specific tasks or processes that take place inside the robot [4][5]. The component-based approach allows for rapid prototyping and testing of new technologies without requiring a complete redesign of the whole system. This style of development is being used to an ever greater extent in industry [1].

An RTC, which is shown in Figure 1, is a logical representation of a hardware and/or software entity that provides well-known functionality and services. The functionality of the RT-Component is as follows:

- Component metadata for dynamic component assembly.
- Component action and execution context for business logic execution.
- Data ports for data exchange between RTCs.

The RTC standard is a PIM (Platform Independent Model) which defines three PSMs (Platform Specific models for implementing the RTC standard): Local PSM (for non-distributed robotics applications), Lightweight Corba Component Model (CCM) and CORBA. The RTC component has been an OMG standard since April 2008, when the initial version 1.0 was released.

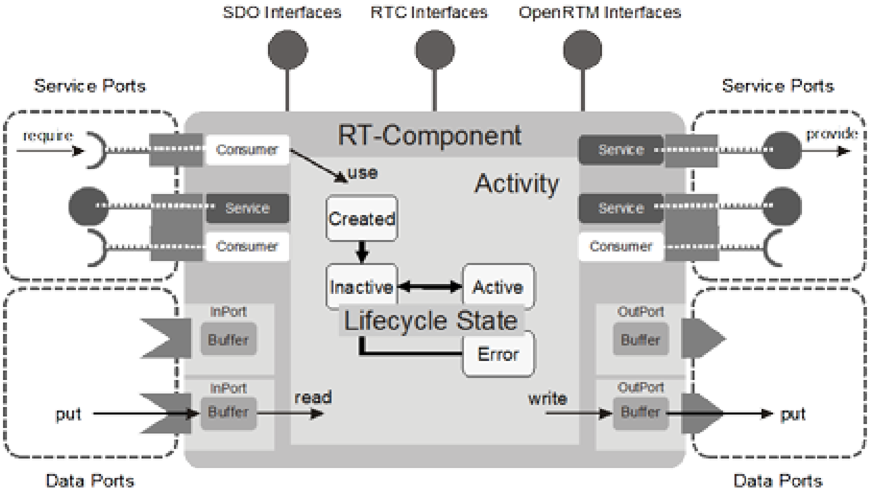


Fig. 1. Representation of a robotics technology component[1].

By extending the general-purpose component functionality of UML with direct support for domain-specific structural and behavioral design patterns, RTCs can serve as powerful building blocks in a robotics system. Developers can combine RTCs from multiple vendors into a single application, allowing them to create much more flexible designs, quicker than in the past.

[1] Copyright AIST : National Institute of Advanced Industrial Science and Technology

1.1 RTC Details

Metadata Acquisition

The ‘Metadata Acquisition’ capability, which is, in effect, the querying and administering of RTCs at runtime, is also known as “Introspection”. An RTC has some interfaces to get metadata including profile and properties about ports. These capabilities can be used by other RTCs, tools or alternate application programs that support dynamic RTC composition. By using metadata acquisition, application programs can obtain information from an RTC at runtime, providing the capacity to dynamically compose RTCs at runtime. These metadata are also useful for developing debugging tools and composition tools for the components. This functionality has two features, one is a resource data model and the other is the stereotype and interfaces. Resource data, which is a kind of data-only class, describes component profiles. Interfaces define some methods to get or set profiles and properties.

Component Action

The ‘Component Action’ interface defines callbacks corresponding to the execution of the lifecycle operations of an RTC. These callbacks can be invoked by the execution entity named ‘Execution Context’ that is a logical thread object. In order to execute RT-component-specific logic, an RTC developer can implement callback operations using Component Actions. These are able to be invoked at each state of the Execution Context. An RTC can participate in Execution Contexts, and an Execution Context can accept multiple RTC participants. As mentioned above, the logic of an RTC and the logical thread is decoupled in the RTC model. This model is useful to implement tightly coupled RTCs in a single (real-time) thread. It is called the synchronous composite RT-Component.

Data Ports

In the low level real-time control layer, if a component is considered a functional unit consisting of inputs, processing, and outputs like a control block diagram, it will be easy to perform system configuration. However, this input/output model is not suitable for general usage in a distributed object model, because an object which sends its data to other objects must know the complete interface definition of each other object. On the other hand, at such low level control layers, the types of data, number of data and units of the data are more important than interface definition.

Service Ports

The software component should be equipped with enough interfaces to provide access to detailed functionality of the component from outside. The ‘Service Ports’ provide endpoints to attach interfaces and required interfaces to it. A component developer can provide his or her own defined interface through the Service Port. The developer also can use provided interfaces by the other components through the Service Ports.

Configuration

The Configuration interface provides interfaces to administrate user defined parameters of RTCs. As mentioned above, a component should not have the hardcoded configuration parameters which prevent reuse of the component.

The configuration consists of some configuration parameters as a list of values with names. An RTC is able to have some configurations as sets. This is called the ‘Configuration Set’. A Configuration Set can be replaced at runtime to adapt the RTC to the current application.

Lightweight RTC

In the RTC specification the basic RTC is called a Lightweight RTC. Here below is a simplified meta-model of the LwRTC (Figure 2).

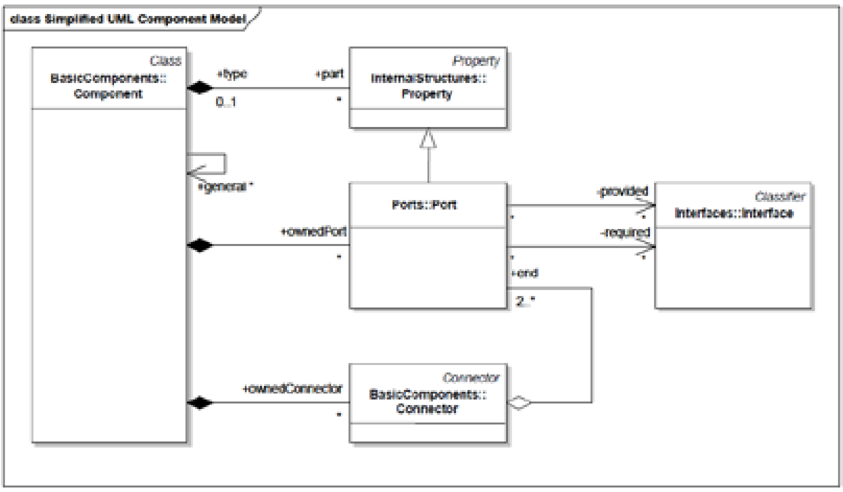


Fig. 2. Lightweight meta-model of an RTC.

A LwRTC consists solely of a package with content algorithm code. Typically, it is inside a LwRTC that a software developer will put their functional code (examples include SLAM, control algorithms or navigation). The RTC standard provides three specialized types of component, derived from the basic component:

• **Periodic Sampled Data Processing** defines the support for a design pattern corresponding to the periodic execution kind. In the context of periodic execution, data flow components are executed periodically in a well-defined order relative to one another. A data flow component may also be a composite component containing other data flow components. In this way, sampled data processing may be decomposed to an arbitrary level of hierarchy.

• **Stimulus Response Processing** may take the form of either asynchronous or discrete event processing. Applications use this pattern when they need to respond asynchronously to changes in their environment. The behaviour of this component can be modelled by using finite state machines. The state machine waits until the asynchronous arrival of an ‘event’, at which point it wakes up and transitions to a new state, executing an action associated with that transition.

- **Modes represent different** types of operation that provide support for applications that are required to transition between different implementations of a given functionality. For example, a car may throttle its engine based solely on the position of the gas pedal or alternatively on the desired speed set by the cruise control. This depends on whether the cruise control is activated or not. “Cruise control on” and “cruise control off” are examples of modes. (In this example, the choice of mode is a binary one, although in general any number of alternatives is possible as required).

In keeping with the RTC standard and the ability to have components composed of components, an RTC can contain several LwRTCs interconnected through their ports (see Figure 3). The RTC is mainly used to regroup LwRTC together to generate more complex and consistent components.

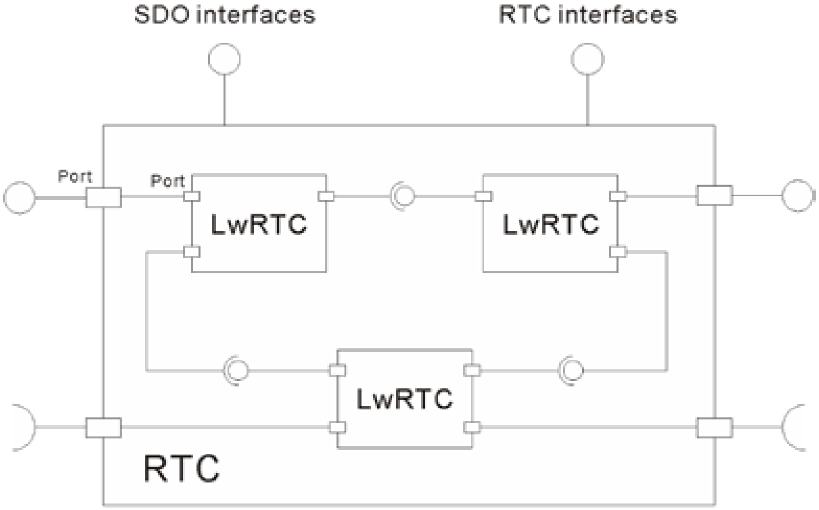


Fig. 3. RTC and LwRTC relationship.

1.2 Summary

The RTC is an OMG standard technology for developing robotics components. It also defines the execution semantics of an RTC component. Since it is a standard, an organization may capitalize on their robotics expertise by incorporating it into RTC components which may be migrated to and executed on any other RTC middleware.
Use of RTC enables a standardized, component based architecture for robotics systems with an inherent high flexibility of the architecture and configuration. Currently, several implementations of the standard are in use worldwide, including Europe.

3 JAUS – AS4: Joint Architecture for Unmanned Systems[2]

JAUS is primarily a standardized message-based protocol to enable interoperability among unmanned systems and associated elements, such as controllers and payloads. It was initially chartered as JAUGS by the Office of the Undersecretary of Defense for Acquisition, Technology and Logistics in America in 1998. Initially it was targeted at ground systems but later dropped the 'G' and focused on developing the protocol for unmanned systems in general.

In 2004, JAUS migrated from the ad-hoc working group to the Society of Automotive Engineers (SAE) Unmanned Systems Steering Group (AS-4) and was realigned as a services orientated architecture. The SAE is a standards development organisation for all forms of powered vehicles, including trucks, boats and aircraft. The JAUS standard may be used and is applicable to both civil and military unmanned systems.

JAUS has been founded on five principles. These are vehicle platform independence, mission isolation, computer hardware independence, technology independence and operator use independence. SAE JAUS is not a single standard, but rather a collection of documents that can be used by requirements writers, system engineers, designers, and implementers. For example, the Architecture Framework for Unmanned Systems (AFUS, SAE AIR 5665) provides a model and common vocabulary to describe the capabilities of a collaborative autonomous system from a user's or customer's perspective. The various JAUS Service Set (SAE AS 5710, SAE AS 6009) documents define a normative, unambiguous set of application-layer interfaces between elements of an unmanned system, where each interface definition is compliant with the machine-readable schema published in the JAUS Service Interface Definition Language (JSIDL, SAE AS 5684). Lastly, the JAUS Transport Specification (SAE AS 5669) defines a link-layer protocol suitable for the transmission of JAUS messages over IP and serial based networks.

JAUS does not define nor mandate a particular implementation; rather, it standardizes the interface between elements. For convenience, JAUS refers to three levels within a system, shown in Fig 4. Level 1 represents communication between subsystems such as the operator control unit (OCU) and one or more unmanned systems. Level 2 refers to transfer of information between nodes inside a subsystem. Each node represents a group of components whose conglomeration produces a computing platform responsible for a particular aspect of the unmanned system. The authors define this as a 'module' containing functionality such as mobility control, sensor fusion, decision making, or a payload. Thus Level 2 is the interface between these modules. The third level is for specifying interactions among software components, where each component hosts one or more service interfaces. The JAUS message set does not currently have the granularity for compliance at this level [8].

[2] Special thanks to David Martin, DeVivo AST, Inc., for technical information and participation on this section.

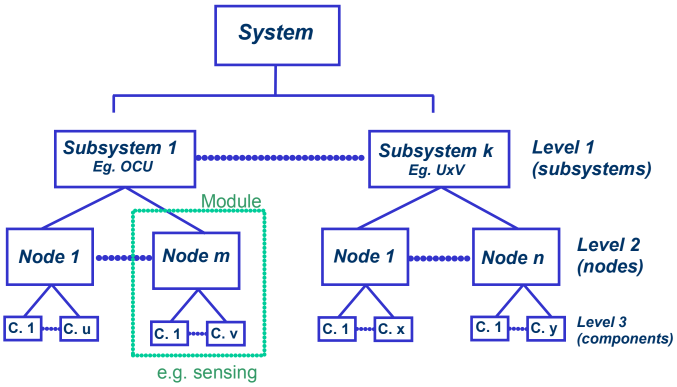


Fig. 4. JAUS architecture definition.

The advantages of using JAUS are derived from having interoperable unmanned systems and payloads provided by multiple vendors:

- The costs of product life-cycles are reduced.
- Provides a modular development approach.
- Reuse of subsystems and modules for other projects.
- Development and integration times are reduced.
- Maintenance costs of products are decreased.
- Wider availability of modules and range of potential vendors.

4 Features of JAUS and RTC That Augment Each Other

The focus of RTC and JAUS are on different aspects required for robotic development and usage. RTC provides a standard means of implementing and executing components, whereas the strength of JAUS is communication and interoperability between modules and unmanned systems. Whilst RTC provides an efficient embedded real-time implementation at the component level, JAUS delivers a powerful services-oriented architecture (SOA) between groups of the components. Due to these different capabilities, the two standards have the ability to complement each other to have a consistent robotic development and usage.

The enhancements that RTC can expect to gain from this alliance include the inheritance of the service orientated structure of JAUS, an exception being the Robotics Localisation Service. Examples of services include the core services, mobility services, UGV services or UUV services. The other key benefit is communication and system management at higher levels enabling inter-robot communication through firewall and over long distances. This leads on naturally to the systems of systems capabilities which JAUS enables.

The areas in which users of a JAUS system stand to gain are several. A standardised means to implement the services at the lower component level becomes available, where procedures are set in place for the implementation and execution for the algorithms oriented towards unmanned systems. This could allow a vendor to provide a JAUS-compliant component that can operate across numerous run-time environments via RTC middleware. Further to this, the capability of dynamic deployment of the components across systems and system reconfiguration at this level is where there is now the benefit of real-time embedded components delivering high performance.

5 An Autonomous Systems Architecture (ASA) with JAUS and RTC.

Using both JAUS and the RTC provides a more complete structural architecture for the interaction among internal mechanisms, tasks and interaction with outside entities. This ASA provides a sound foundation for developing plug-and-play type code where desired levels of functional granularity can be obtained.

Figure 5 provides a representation of the implementation of JAUS and RTC inside an autonomous system. Developing autonomous systems using this framework means one has the capability to address systems of systems and components within components as applied to unmanned systems, including the related service definitions.

This ASA provides a consistent development approach for the JAUS community enabling compatibility (and interoperability) of the middleware among different organizations. In addition one has middleware with the capacity of real-time performance. The RTC standard will ease the software development of robotics embedded components. Complementary to this is that the JAUS standard will ease system integration with other systems with a SOA.

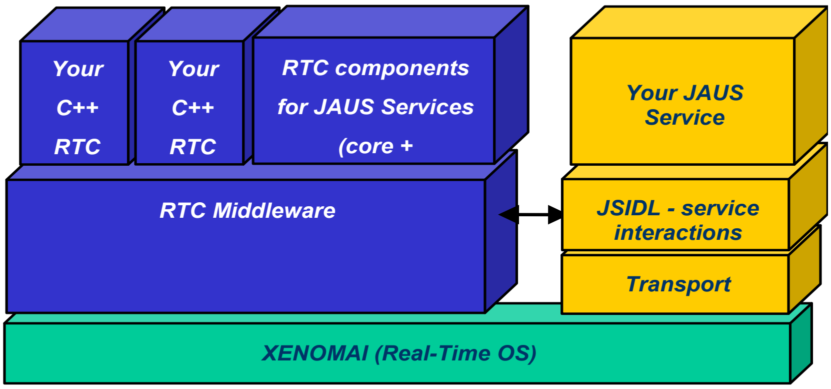


Fig. 5. Proposed interaction between JAUS and RTC: ASA

Thus, the ASA offers capabilities to plug a JAUS service into the architecture. As an example, consider integration of a LIDAR. The developer has three possible implementation solutions, depending on the needs. First, a pure RTC implementation provides a portable solution that can be executed on an compatible middleware. A second option would use a pure JAUS implementation. This provides for additional interoperability via access by remote components; or 3) A RTC-component with a JAUS wrapper. This provides the advantages of both standards, with system-to-system interoperability along with a portable implementation. The developer also has the freedom to choose the boundary at which to apply the wrapper. This makes for a truly flexible architecture.

6 Conclusion

The power of the combined usage of JAUS and the RTC in addressing a development framework for unmanned robotic systems has been described. Providing the means for their interaction will enable a greater extent of technology transfer and reuse - and a flexibility of development from components inside components to systems of systems. In addition, mutual compliance will help to strengthen the position of the two standards in the global market.

Other particular aspects that the RTC stands to gain include the introduction of higher levels of communication and extended capabilities such as interoperability with other JAUS based autonomous robots. A JAUS implementation would inherit those things at the opposite end of the spectrum, or rather all the functionality provided at the component level offered by the RTC standard including high performance real-time processing. In addition, developers would also have the opportunity to gain the use of integration tools used by RTC based on the OMG MDA (Model Driven Development) for its services design and development.

The advantageous are numerous and with collaboration between the two communities, the autonomous systems architecture implementing JAUS and RTC can be realised. The wheels for this have already been set in motion.

7 Acknowledgements

The authors would like to express their appreciation of the support provided, while this paper was developing, from both the JAUS and RTC communities. In particular special thanks go to David Martin (co-chair of SAE-AS4A committee) and Noriaki Ando (AIST) for their feedbacks.

References

[1] E. Prassler & A. Shakhimardanov, "State of the Art report Robot Middleware", *Robot Standards and Reference Architectures* (RoSta), October, 2007.

[2] D. Martin, "SAE JAUS Briefing", SAE AS-4 Meeting, March 2010, Jacksonville, Florida, U.S.A.

[3] L. Rioux, “RTC / JAUS: 2 complementary standards”, OMG meeting, Robotics Task Force (http://robotics.omg.org), March 2010, Jacksonville, Florida, U.S.A.

[4] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, W. Yoon, "RT-Middleware: Distributed Component Middleware for RT (Robot Technology)", 2005 IEEE/RSJ, *International Conference on Intelligent Robots and Systems* (IROS2005), pp.3555-3560, 2005.08, Edmonton, Canada.

[5] N. Ando, T. Suehiro, T. Kotoku, "A Software Platform for Component Based RT-System Development: OpenRTM-Aist", *International Conference on Simulation, Modeling and Programming for Autonomous Robots* (SIMPAR 2008), pp.87-98, 2008.11, Venice, Italy, ISSN 0302-9743.

[6] OpenRTM-aist official web site, http://www.openrtm.org

[7] RTC 1.0 Specification OMG, formal/08-04-04, http://www.omg.org/spec/RTC/1.0/, 2008.

[8] J. Pedersen, A Practical View and Future Look at JAUS, White Paper, May, 2006.

Modelling for Robotics
Working Group

Charter
Robotics/2010-06-12

This working group is a sub-group of the Robotics Task Force. This working group is focusing on modelling engineering for robotics.
Current engineering approaches for robotics have indeed been demonstrated to be not sufficient in order to bypass both following constraints that robotics systems are faced to now:

- *the problem space is huge:* as uncertainty of the environment and the number and type of resources available to the robot increase, the definition of the best matching between current situation and correct robot resource exploitation becomes overwhelming even for the most skilled robot engineer,
- *the solution space is huge:* in order to enhance robustness of complex robotic systems, existing cognitive methods and techniques need to adequately exploit robotic-specific resources. This means that the robotics engineer should master highly heterogeneous technologies in order to integrate them in a consistent and effective way.
- *the problem of integration and reuse*: is critical because robotics systems required a lot of assets and intelligence. The only way to be able to develop new robots with a reduced cost is by reusing and integrating very large and different existing hardware and software.

The ideal process for developing robots is the one that enables the design and implementation of highly complex and robust robotic systems involving as less effort as possible. Robots are complex and embedded systems, since models based approach has already demonstrate its efficiency for complex systems and also for embedded systems, we expect that models is really promising solution for the development process of robotics software & systems. This working group will focus to promote and standardise technologies related to model based engineering for robotics.

The objectives of this working group will be also to:

- Exchange information and techniques related to model based development processes for robotics.
- Identify and developing new enabling technologies based on model based processes.
- Explore issues from real world that will produce technology recommendations
- Promoting “early adopters” results and identifies issues related to models engineering.
- Promoting new and enabling technologies related to model based for robotics.

Chairs:

- Takeshi Sakamoto (Technologic Arts)
- Laurent Rioux (THALES)
- Toby McClean (Zeligsoft)

Object Management Group

140 Kendrick Street
Building A Suite 300
Needham, MA 02494

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

Models for robotics

Request For Information

OMG Document: *<prefix/YYYY-MM-NN>*

Responses due: *<month> <day>, <year>*

<Notes to RFI Editors: (1.) Spell out month name; e.g., January (2.) Don't forget to replace the running header and footer with the name of the RFI title and document number, date, and so on. (3.) Remove or replace any notes in RED and enclosed by angle brackets. >

1.0 Introduction

1.1 The Object Management Group (OMG)

With well-established standards covering software from design and development, through deployment and maintenance, and extending to evolution to future platforms, the Object Management Group (OMG) supports a full-lifecycle approach to enterprise integration which maximizes ROI, the key to successful IT. OMG's Modeling standards, the basis for the MDA, include the Meta Object Facility (MOF), the Unified Modeling Language (UML), Business Process Modelling Notation (BPMN) and the Common Warehouse Metamodel (CWM). CORBA, the Common Object Request Broker Architecture, and the Data Distribution Service (DDS) are OMG's standard open platforms, widely used in Enterprise and embedded applications. Headquartered in Needham, MA, USA, the Object Management Group is an international, open membership, not-for-profit computer industry specifications consortium. More information about OMG can be found at www.omg.org.

1.2 *Robotics Task force*

< Note to RFI Editors: Briefly summarize the mission and goals of your subgroup and explain how responses to this RFI will help meet your objectives.>

And the modeling working group

1.3 RFI Objectives

1.3.1 What is an OMG RFI?

The intent of an OMG Request for Information (RFI) is to gather information for the purpose of guiding a subgroup in its efforts to provide solutions to industry problems. The RFI process is used by a subgroup to canvass a targeted industry segment for one or more of the following purposes:

- Acquiring general or specific information about industry requirements.
- Soliciting assistance in identifying potential technology sources.
- Soliciting input to validate a subgroup’s roadmap.

Generally speaking, the RFI process determines which Request For Proposals (RFPs) will be issued (and, based on negative feedback, which won't) or influences the way a particular RFP is constructed.

2.0 Information Being Requested

2.1 Summary of this RFI

This RFI calls for needs on standards technologies required for modeling, designing robotics systems.

Submitters are invited to propose new standard technologies required for robotics systems. Note to RFI Editors: Provide a 2-3 paragraph summary of what information is being sought and how it will be used (e.g., develop a roadmap.>

2.2 Detail

< Note to RFI Editors: Be precise but not prescriptive about what information is being sought. Remember that an RFI is an information-gathering exercise, and you will not be aware of all the possible responses – if you were, you wouldn't be issuing an RFI. You must explicitly leave the door open to anyone with something relevant to contribute to send it as a response. Try to make it clear to your readers that this is not an RFP, so we're not looking for a specification – yet. >

1- About your organization

Q1.1: What industry is your organization?

Q1.2: what market your organization is targeting?

2- About Robotics

Q2.1: What are your experiences in robotics?

Q2.2: what does the robotics market represent for your organization?

Q2.3: what are robotic engineers' skills inside your organization?

Q2.4: what kind of robotic systems does your organization develop?

3- About Models

Q3.1: Which roles do models play (or could play) for you?

Q3.2 What are the skills on modeling inside your organization

Q3.3: Where in your engineering process are you using (or plan to use) models?

Q3.4: What is the experience of your organization with model based engineering?

4- About Robotics and models

Q4.1: What are the requirements a modeling language for robotics must have?

Q4.2: what challenges/difficulties of using models do you foresee in robotics fields?

Q4.3: what is for you the most missing standard technologies for your organization related to robotics and models?

Q4.4: What are the requirements for tooling and automation that you expect for your robotics activities?

3.0 Instructions for Responding to this RFI

3.1 Who May Respond

Responses from *anyone* in industry, government or academia with practical knowledge of *models for robotics* are welcome.

When and if OMG issues a subsequent Request for Proposals (RFP) in this area, OMG members at the appropriate membership level will be eligible to respond with detailed specifications. OMG is an open membership organization. Any company, university or organization is welcome to join and participate. For information, consult *http://www.omg.org/membership*.

3.2 How to Respond

One electronic copy in machine-readable format should be sent to ***omg-documents@omg.org***. Acceptable formats are ODF (ISO/IEC 26300), PDF (ISO 32000), ISO Latin-1 (ISO/IEC 8859-1) or MS Word .doc files. Please consult OMG before submitting documents in other machine-readable formats. One confirming paper copy of all documents should also be sent to the OMG postal address below.

Object Management Group, Inc.
140 Kendrick Street
Building A Suite 300
Needham, MA 02494
USA
Attn: *<title>* RFI

Responses to this RFI must be received at OMG no later than 5:00 PM US Eastern Time (typically 22:00 GMT) *<due date>*.

Other communication regarding this RFI should be sent to the contacts listed in paragraph 3.8.

3.3 RFI Response Contact

Companies responding to this RFI shall designate a single contact within that company for receipt of all subsequent information regarding this RFI and the forthcoming series of RFPs. The name of this contact will be made available to all OMG members.

3.4 Format of RFI Responses

The following outline is offered to assist in the development of your response. You should include:

- A cover letter -- the cover letter should include a brief summary of your response, such as indicating to which areas you are responding and must also indicate if supporting documentation is included in your response.

- The response itself, covering any or all of the areas of information requested by this RFI.

- If required, a glossary that maps terminology used in your response to OMG standard terminology. (For example, see OMG specifications [CORBA, MOF, UML, XMI] and a description of OMG's Model Driven Architecture [MDA] for OMG's standard terminology.)

Although the OMG does not limit the size of responses, you are asked to consider that the OMG will rely upon volunteer resources with limited time availability to review these responses. In order to assure that your response receives the attention it deserves, you are asked to consider limiting the size of your response (not counting any supporting documentation) to approximately 25 pages. If you consider supporting documentation to be necessary, please indicate which portions of the supporting documentation are relevant to this RFI.

3.5 Distribution of RFI Responses

Copies of all documentation submitted in response to this RFI will be available to all OMG members for review purposes.

3.6 Copyrighted Material

According to OMG Policies and Procedures, proprietary and confidential material shall not be included in any response to the OMG. Any material received is treated as a public document. If copyrighted material is sent in response to this RFI then a statement waiving that copyright for use by the OMG is required and a limited waiver of copyright that allows OMG members to make up to twenty-five (25) copies for review purposes is required. Consult Appendix B for a template for this copyright waiver.

3.7 Reimbursement

The OMG will not reimburse submitters for any costs in conjunction with their responses to this RFI.

3.8 Questions Regarding this RFI

Any technical questions regarding this RFI should be sent to:

*robotics-chairs@omg.org*Questions regarding the response process should be forwarded to:

Object Management Group, Inc.
140 Kendrick Street
Building A Suite 300
Needham, MA 02494
USA
Attn: RFI Response Desk

Phone: +1-781-444 0404
Fax: +1-781-444 0320
Email: rfi-responses@omg.org

4.0 Response Review Process and Schedule

4.1 Review Process

OMG RFIs are issued with the intent to survey industry to obtain information that provides guidance, which will be used in the preparation of RFPs. The OMG membership, specifically the *robotics task force*, will review responses to this RFI. Based on those responses, the *robotics task force and its modelling working group* will augment its roadmap and prepare one or more RFPs.

4.2 Clarification

To fully comprehend the information contained within a response to this RFI, the reviewing group may seek further clarification on that response. This clarification may be requested in the form of brief verbal communication by telephone; written communication; electronic communication; or a presentation of the response to a meeting of the robotics task force

4.3 RFI Response Presentations and Demonstrations

RFI Respondents may be invited to present their response to the *robotics task force*. The purpose of this presentation would be to seek clarification of information contained within the response (as noted above); to further explore issues raised; or to further meet the goals of the RFI.

In addition, a technology demonstration to the *robotics tsk force* may prove useful to support the RFI response. If desired, please coordinate with the Contact cited in paragraph 3.8.

<Note to Subgroup Chairs:>

<Contact OMG support personnel at *demonstrations@omg.org* in order that preparations can be made.>

4.4 Schedule

The schedule for responding to this RFI is as follows. Please note that early responses are encouraged.

RFI issued:	*<date>*
RFI responses due:	*<date – must be at least four (4) weeks before the first OMG meeting where those responses will be considered>*
Review of RFI responses:	*<date>*

Appendix A References and Glossary Specific to this RFI

A.1 References Specific to this RFI

< Note to RFI Editors: Insert any references specific to this RFI in alphabetical order. The five listed below are offered as examples. >

[CORBA] Common Object Request Broker Architecture
(CORBA), *http://www.omg.org/spec/CORBA*.

[MDA] MDA Guide, Version 1.0.1, *http://doc.omg.org/omg/2003-06-01*.

[MOF] Meta-Object Facility (MOF), *http://www.omg.org/spec/MOF*.

[UML] Unified Modeling Language (UML), *http://www.omg.org/spec/UML*.

[XMI] XML Metadata Interchange (XMI), *http://www.omg.org/spec/XMI*.

A.2 Glossary Specific to this RFI

< Note to RFI Editors: Insert any glossary items specific to this RFI in alphabetical order. >

Appendix B Template for Copyright Waiver for RFI Responses

[Date]

Object Management Group, Inc.
140 Kendrick Street
Building A Suite 300
Needham, MA 02494
Attn: RFI Response Desk

Fax: +1 781-444-0320

This letter constitutes a limited license to use certain materials copyrighted by the undersigned. We understand that the Object Management Group, Inc. ("OMG") is a not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications.

We understand that the Copyrighted Material identified below is being submitted to OMG as part of a response to the identified Request for Information (RFI), for use in connection with an OMG process that may result in the adoption of an OMG specification.

Source of Copyrighted Material:

Copyrighted Material to be submitted to OMG:

Submitter(s):

RFI Doc.-Title & No.

We hereby grant OMG the right to make an unlimited number of copies of the Copyrighted Material as part of the OMG adoption process.

We hereby grant each OMG member the limited right to make up to twenty-five (25) copies of the Copyrighted Material for review purposes only as part of the OMG adoption process.

Regards,

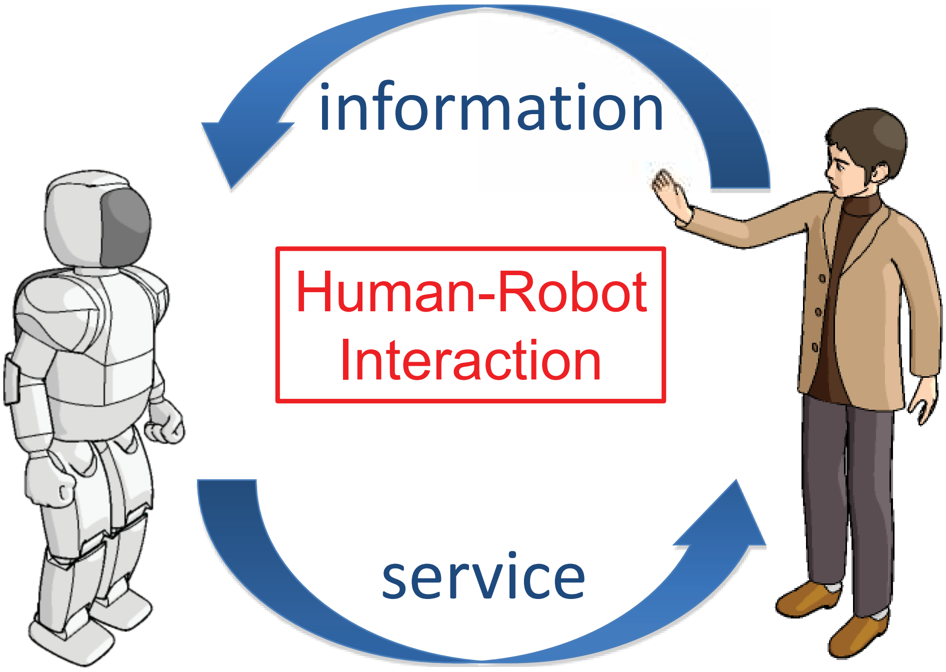
Robotic Interaction Service(RoIS) Framework RFP

Document number: robotics/2010-06-14

Robotics Functional Service WG, Robotic DTF

Background

Many robotic services are provided through human-robot interaction (HRI).



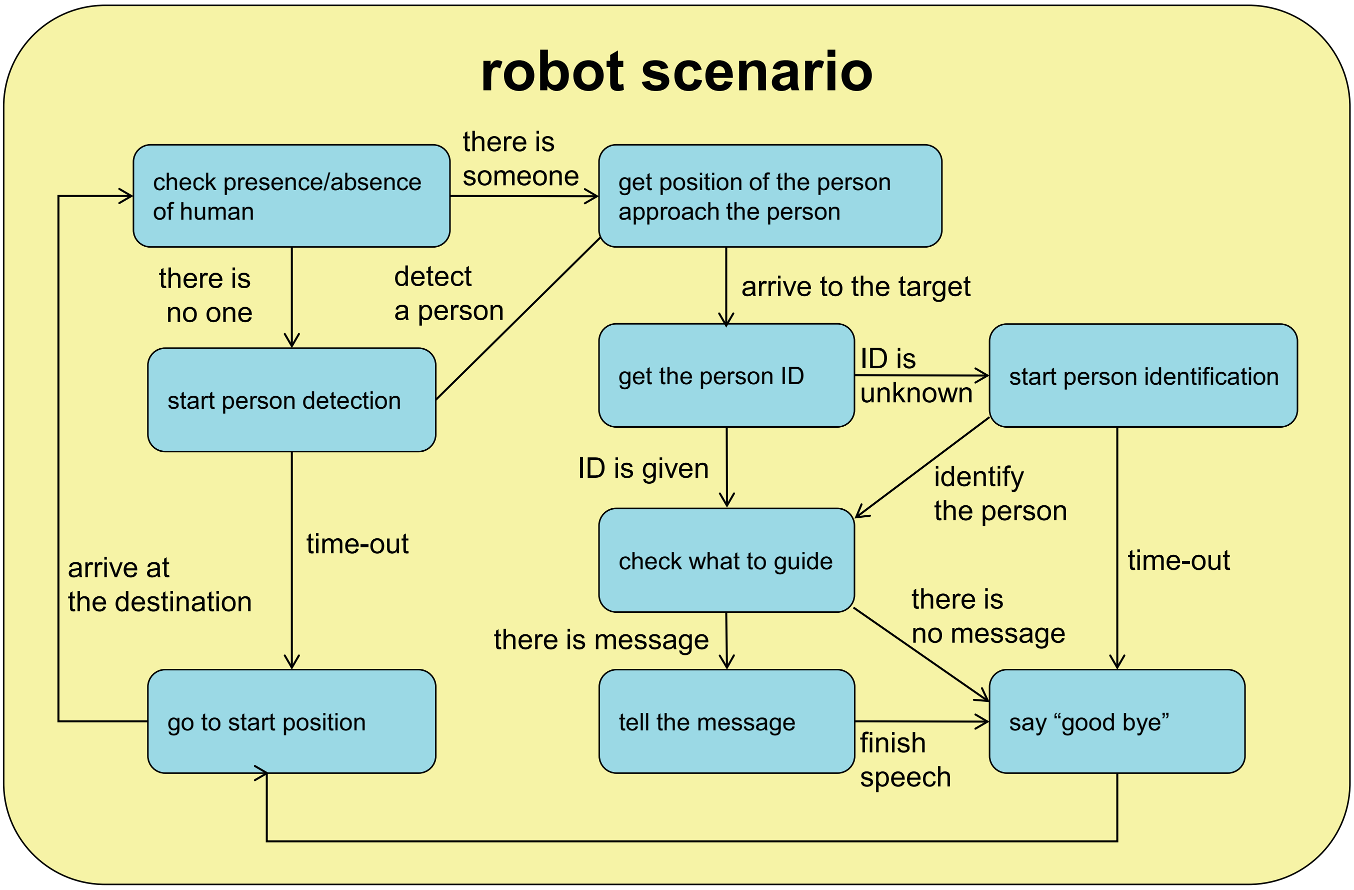
Motivation

- Robotic service applications are described as
  - procedures of robot behavior
  - relation between the behavior and the information provided by HRI components
- Each HRI component depends on the robot platform.
  - equipped sensors and actuators
  - APIs for the robotic functions

  are generally specific to the robot platforms.
- Programmer must master the specification of the robot platform
- Service application program must be written for each robot independently

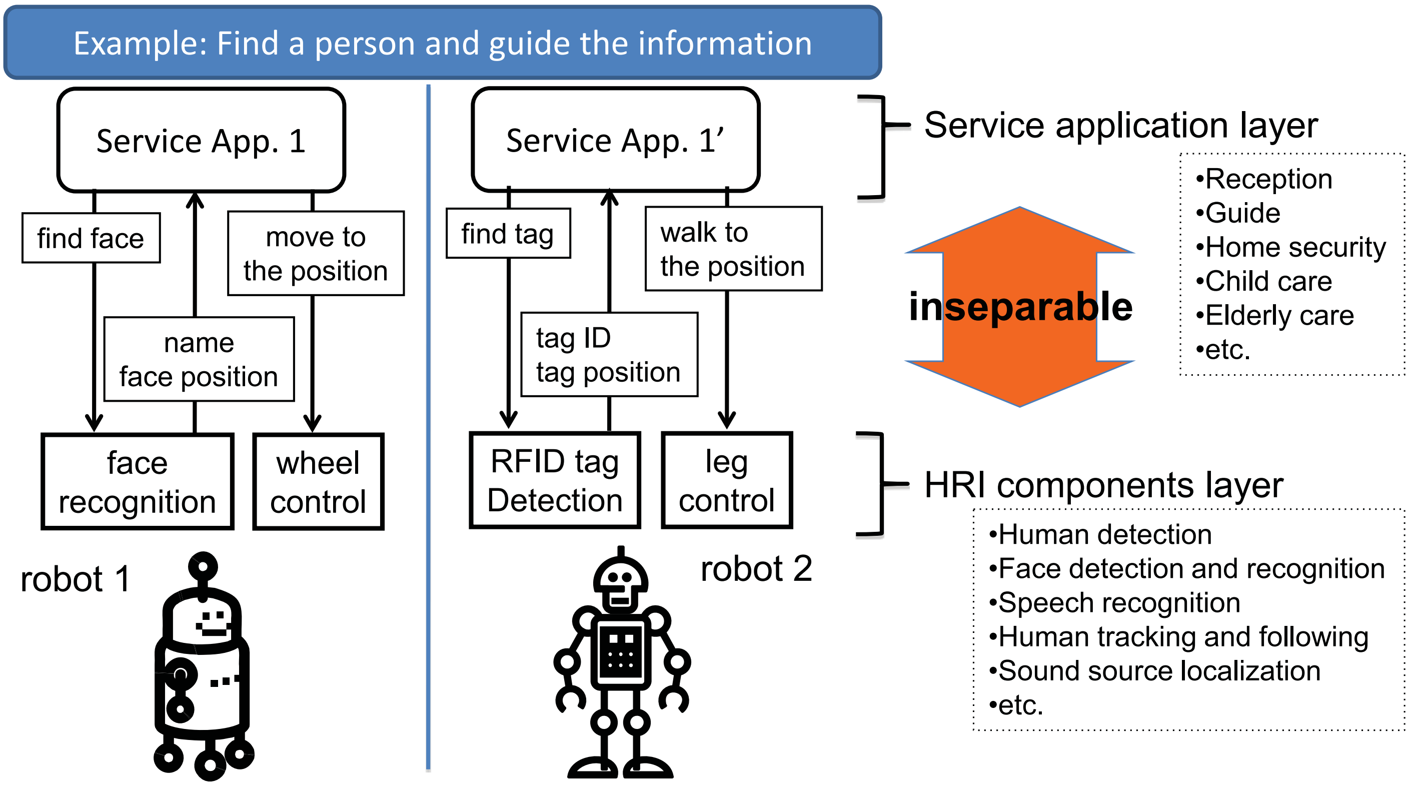
General framework for HRI functions is needed as Robotic Interaction Service (RoIS)

Example of Service Application



Conventional Style

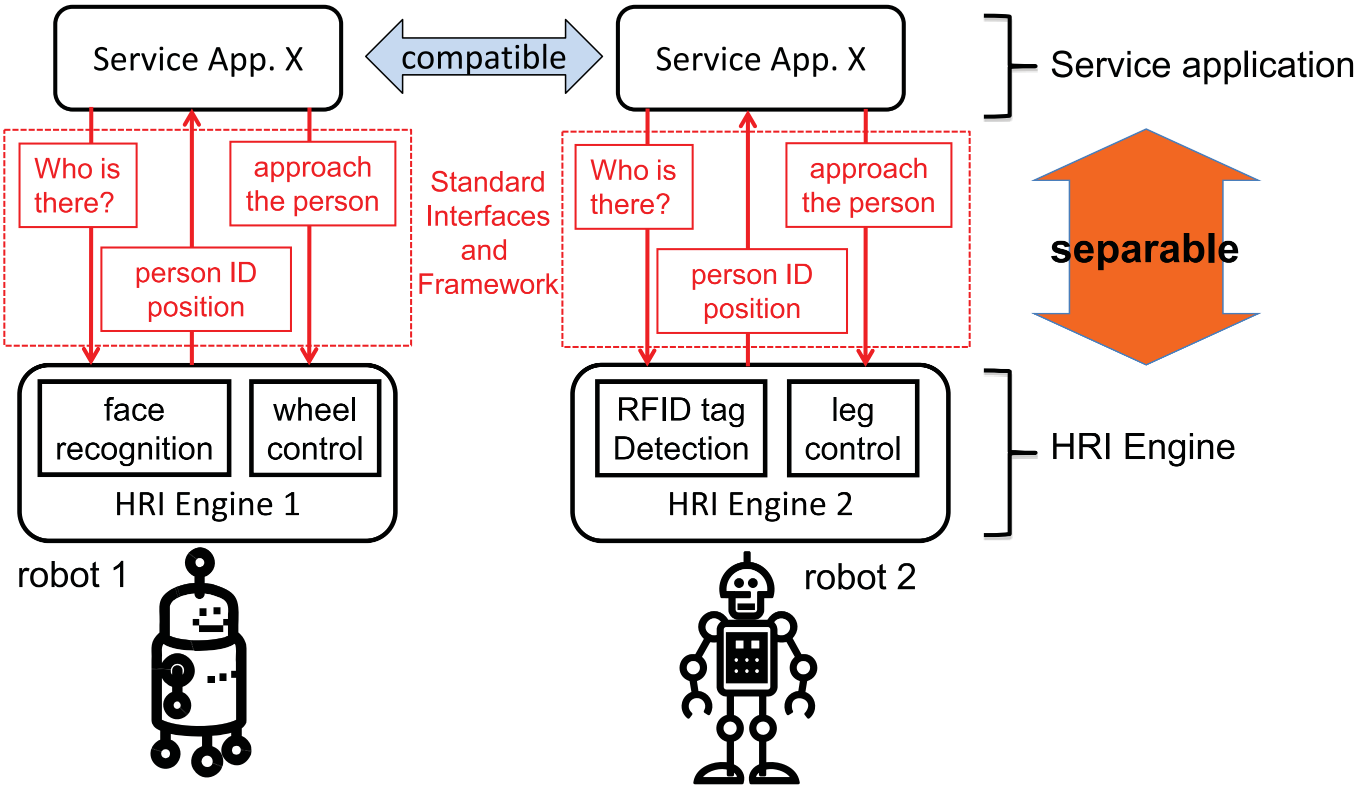
- Service applications must be developed for each robot
- Difficult to separate service app. layer from robot platform



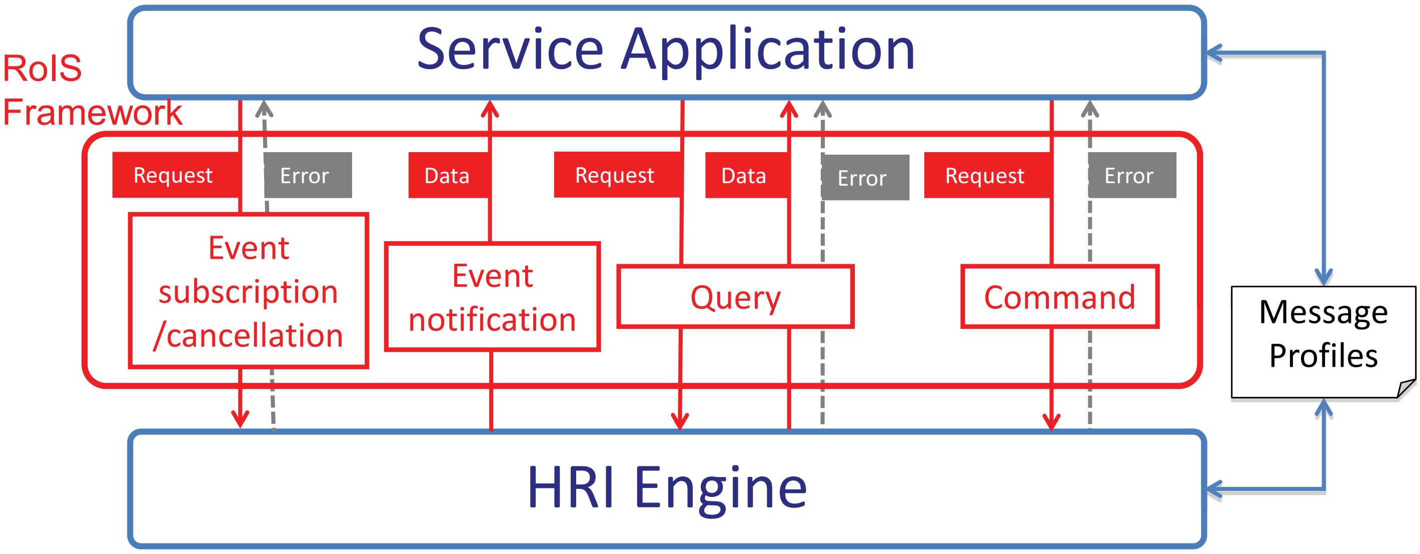
Example: Find a person and guide the information

Assumption

- HRI components should be encapsulated in HRI engine.
- Interfaces for HRI engine should be standardized.

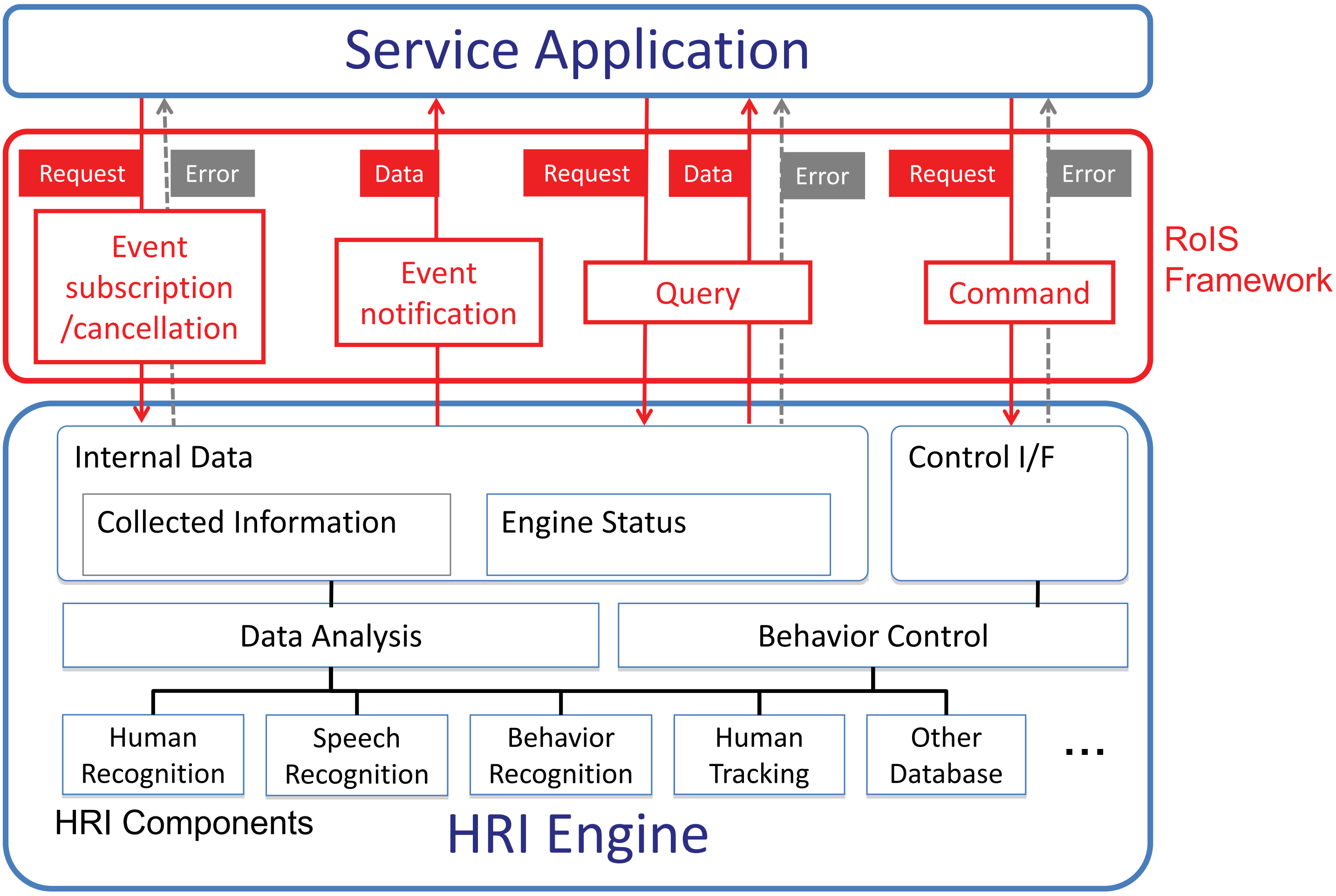


Concept of RoIS Framework



- **<u>Interfaces between service application and HRI Engine:</u>**
  - Interface to obtain information actively from HRI Engine (Query)
  - Interface to receive information timely from HRI Engine (Event)
  - Interface for the instruction to control HRI Engine functions (Command)
- **<u>Definition of common messages for all HRI Engines</u>**

Example of RoIS Framework



Common Messages

- Common messages for each interface type should be defined as possible.

  For Example,

- ➢ Event notification
  - Person detected
  - Person identified
  - Speech recognized
  - Sound detected
  - Action completed
  - Low battery
  - etc

- ➢ Query
  - Presence/absence of human
  - Person ID
  - Position of person / robot
  - History of actions
  - Status of HRI Engine
  - etc

- ➢ Command
  - Start / stop detection
  - Start / stop recognition
  - Approach the person
  - Follow the person
  - Go to XXX
  - Talk XXX
  - etc

Message Profile

- Messages for the interface types can be classified into two categories:
  - ➢ common message type
  - ➢ domain specific message type

    (ex. educational, navigation, healthcare domain)
- Scheme to describe “message profile” should be defined.

  each profile includes such as,
  - availability of messages
  - name of each message
  - data format
  - required argument (s)

What’s Changed?

According to the comments from AB reviewers

- Background of robotic service application
  - added example of robotic scenario concept
- Definition of interface types for RoIS
  - added example of common messages for each type
  - clarified the concept of “message profile” for each interface type that was called “profile of interface” in the previous draft
- 6.2 Scope of Proposals Sought (1) (2) (5)
- 6.5 Mandatory Requirements (1)-(5)
- 6.6 Optional Requirements (2) (3) (moved from 6.5)
- 6.7 Issue to be discussed
  - added the bullet about “the generality wrt sensors/actuators/algorithms”

Comments from Mr. Vincent (1)

- The figure 3 as well as the 5 following paragraphs are too broad: almost all API in the world can be split in these four types: Event subscription, Event notification, Query and Command.

→We have added an explanation and a figure (new figure 1) of robot scenario concept to clarify the necessity for the four interface types. Two paragraphs before figure 4 (previous figure 3) are also corrected for this purpose. The most important point of this RFP is that we should standardize the content and handling of messages for these interfaces. That is, it is desirable to define common interface messages specific to human-robot interaction. To express it in a better way, we added some examples of such common messages after figure 4.

Comments from Mr. Vincent (2)

- The concept of "Profiles of interface" as well as how to "manage" them are unclear. These concepts must be made clearer in the RFP before continuing the process.

→We changed the words "profiles of interface" to “message profiles” and explained it in detail after figure 4.

Comments from Mr. Vincent (3)

- Section 6.2 - 1st paragraph: "It is the target for ... human." -> "This RFP is targeted on human-robot interaction”

→We corrected the sentence.

- Section 6.2 - 1st bullet (1) AND Section 6.5 Requirement (1):

  What do you mean by “overall architecture”? The “architecture” term is too broad and has too many uses to let it alone without clarification - Please specify this requirement. Moreover, are yor sure you want to standardize such an "architecture" for RoIS? Isn’t it implementation-dependant?

→ We have corrected the expression "overall architecture" to clarify the requirement.

Comments from Mr. Vincent (4)

- Section 6.2 - 5th bullet (5):

  As already said, this is too broad for an RFP and can simply be removed from it.

→We have unified (2) and (5) into (2) and corrected according to the above corrections.

- Section 6.5 - Requirement (2): "at least one PSM"

  My guess is that you don't want whatever PSM so please specify what you want: RTC and XML?

→We have specified the requirement for PSM and changed to the optional requirement.

Comments from Mr. Vincent (5)

- Section 6.5 - Requirements (3) to (5)

  As already said the concepts of profile and profile management are not clear enough to keep this like that.

→We have dropped the requirement for “schema to manage profile” and separated these requirements to the requirements for “schema to describe message profile” and “common messages”. The requirement for “schema to describe message profile” is changed to the optional requirement.

- Section 6.7: Proposals should also be requested to discuss about their generality wrt various sensors, actuators and algorithms (cf section 6.2 - bullet (3)).

→ We have added the last bullet based on the comment.

Comments from Mr. Vincent (6)

- I'm puzzled by the lack of any specific interfaces you want to find in the future standard. For instance, I was waiting for a list of useful interfaces such as those found in figure 2: "Is there somebody?", "Get Person ID", "Approach the person”… If you really have no list for this, I believe that you need first either to work on it inside your TF or to draft an RFI to request this list before redrafting this RFP.

→Although we discussed about it in our TF through workshops with several robot vendors, it is difficult to fix a list of such messages without hearing more concrete opinions. We should discuss with submitters based on their proposals. Therefore, we have listed it in the the mandatory requirements.

Comments from Mr. Andrew (1)

1. Does the RFP seek to standardise a complete set of domain abstractions for the domestic robotic domain (e.g. person, speech, pick-up, carry, put-down, etc), or just to standardise a framework within which these concepts can be expressed?

→This RFP seeks the latter.

2. The RFP talks extensively about events. Does it direct submitters to re-use any of the existing event frameworks that OMG as already standardised?

→Yes. We expect to use existing communication protocols and middlewares to achieve this framework as possible. Therefore, we have added the requirements in Section 6.5 (3) (4)to specify the classification of the required functions into the existing standards and the other.

Comments from Mr. Andrew (2)

3. The Mandatory requirements section talks about events, but barely mentions the important issue of domain abstractions and mechanisms for defining new abstractions.

→We have explained about the issue as a part of "message profiles" and mentioned the requirement for message profiles in Section 6.6 (2).

4. Do you want the submitters to target any particular platform with their PSM? If so, which one?

→We have specified the requirement for PSM in Section 6.6 (3).

Mandatory Requirements (1)

1) Proposals shall provide overview of architecture that consists of RoIS framework, a robotic service application, HRI Engine that is a set of robotic components (diagram or description for overview)
2) Proposals shall specify a general mechanism for RoIS framework
   - Interfaces between applications and HRI Engine and their types:
     - Interface to obtain information actively from HRI Engine
     - Interface to receive information timely from HRI Engine
     - Interface for the instruction to control HRI Engine functions
   - Data structure for each interface type

Mandatory Requirements (2)

3) Proposals shall specify communication protocols and middlewares to achieve functions required for RoIS framework.
4) Proposal shall specify functions that cannot be achieved by existing communication protocols or middlewares.
5) Proposals shall specify common messages for all HRI Engines

Optional Requirements

1) Proposals may specify schemes for the functionalities listed below.
   - Error handling for each interface type
   - Returning command results and status
2) Proposals shall provide a schema to describe message profile for each interface type.
3) Proposals may provide a Platform Independent Model (PIM) and Platform Specific Model (PSM) as CORBA-specific model, XML or RTC for RoIS framework.

Schedule

Event or Activity	Actual Date
Preparation of RFP by TF	24. May. 2010
RFP placed on OMG document server	24. May. 2010
Approval of RFP by Architecture Board *Review by TC*	21. June. 2010
TC votes to issue RFP	25. June. 2010
LOI to submit to RFP due	13. September. 2010
Initial Submissions due and placed on OMG document server ("Three week rule")	8. November. 2010
Voter registration closes	29, November, 2010
Initial Submission presentations	6. December. 2010
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Three week rule")	
Revised Submission presentations	
Final evaluation and selection by TF *Recommendation to AB and TC*	
Approval by Architecture Board *Review by TC*	
TC votes to recommend specification	
BoD votes to adopt specification	

OMG Document: robotics/2010-06-15

<Mr. Vincent's comments>

1. The figure 3 as well as the 5 following paragraphs are too broad: almost all API in the world can be split in these four types: Event subscription, Event notification, Query and Command.

→ We have added an explanation and a figure (new figure 1) of robot scenario concept to clarify the necessity for the four interface types. Two paragraphs before figure 4 (previous figure 3) are also corrected for this purpose. The most important point of this RFP is that we should standardize the content and handling of messages for these interfaces. That is, it is desirable to define common interface messages specific to human-robot interaction. To express it in a better way, we added some examples of such common messages after figure 4.

2. The concept of "Profiles of interface" as well as how to "manage" them are unclear. These concepts must be made clearer in the RFP before continuing the process.

→ We changed the words "profiles of interface" to "message profiles" and explained it in detail after figure 4.

3. Section 6.2 - 1st paragraph: "It is the target for ... human." -> "This RFP is targeted on human-robot interaction"

→ We corrected the sentence.

4. Section 6.2 - 1st bullet (1) AND Section 6.5 Requirement (1): What do you mean by "overall architecture"? The "architecture" term is too broad and has too many uses to let it alone without clarification - Please specify this requirement. Moreover, are yor sure you want to standardize such an "architecture" for RoIS? Isn't it implementation-dependant?

→ We have corrected the expression "overall architecture" to clarify the requirement.

5. Section 6.2 - 5th bullet (5): As already said, this is too broad for an RFP and can simply be removed from it.

→ We have unified (2) and (5) into (2) and corrected according to the above corrections.

6. Section 6.5 - Requirement (2): "at least one PSM": My guess is that you don't want whatever PSM so please specify what you want: RTC and XML?

→ We have specified the requirement for PSM and changed to the optional requirement.

7. Section 6.5 - Requirements (3) to (5): As already said the concepts of profile and profile management are not clear enough to keep this like that.

→ We have dropped the requirement for "schema to manage profile" and separated these requirements to the requirements for "schema to describe message profile" and "common messages". The requirement for "schema to describe message profile" is changed to the optional requirement.

8. Section 6.7: Proposals should also be requested to discuss about their generality wrt various sensors, actuators and algorithms (cf section 6.2 - bullet (3)).

→ We have added the last bullet based on the comment.

9. I'm puzzled by the lack of any specific interfaces you want to find in the future standard. For instance, I was waiting for a list of useful interfaces such as those found in figure 2: "Is there somebody?", "Get Person ID", "Approach the person"… If you really have no list for this, I believe that you need first either to work on it inside your TF or to draft an RFI to request this list before redrafting this RFP.

→ We made a list of common messages. It is described after figure 4 and in the mandatory requirement (in Section 6.5 (5)).

<Mr. Watson's comments>

10. Does the RFP seek to standardise a complete set of domain abstractions for the domestic robotic domain (e.g. person, speech, pick-up, carry, put-down, etc), or just to standardise a framework within which these concepts can be expressed?

→ This RFP seeks the latter. In our latest draft, it is mentioned in the paragraphs after figure 3. This concept includes several robotic domains.

11. The RFP talks extensively about events. Does it direct submitters to re-use any of the existing event frameworks that OMG as already standardised?

→ Yes. We expect to use existing communication protocols and middlewares to achieve this framework as possible. Therefore, we have added the requirements in Section 6.5 (3) (4) to specify the classification of the required functions into the existing standards and the other.

12. The Mandatory requirements section talks about events, but barely mentions the important issue of domain abstractions and mechanisms for defining new abstractions.

→ We have explained about the issue as a part of "message profiles" and mentioned the requirement for message profiles in Section 6.6 (2).

13. Do you want the submitters to target any particular platform with their PSM? If so, which one?

→ We have specified the requirement for PSM in Section 6.6 (3).

Object Management Group

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

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

Request for Proposal

Robotic Interaction Service (RoIS) Framework RFP

OMG Document: robotics/2010-06-16

Letters of Intent due: September 13, 2010
Submissions due: November 8, 2010

Objective of this RFP

This RFP solicits proposals for a Platform Independent Model (PIM) and a Platform Specific Model (PSM) for robotic interaction service (RoIS) that specify

- common interfaces between robotic service applications and components that provide functions for performing human-robot interaction.
- data structures for each interface.

For further details see Chapter 6 of this document.

6.0 Specific Requirements on Proposals

6.1 Problem Statement

Service robot that provides some services to people in daily life will become more and more popular in robotics market. These service robots provide appropriate services through human-robot interaction (HRI). For example, there are robotic services such as,

- Reception service
- Guide service
- Home security service
- Childcare robot service
- Elder person daily watching service

Service application is provided as a set of robot behaviors. The robot behavior rule is defined based on the information collected from humans or environments. The information is collected by using functions such as,

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

Generally, several sensors and actuators are equipped with the robot body or the environment where the robot provides the service. The service application programmer describes procedures for robot action and relation between the action and the information obtained by these sensors in the service application program. However, the various types of sensors and actuators are equipped with various robots, and moreover, the application program interfaces (APIs) of each robot are different by robot vendors even if their sensor types are the same. Therefore, an application program developed for one specific robot will not run on the other robot. This is one of the reasons of inefficiency in robot industry.

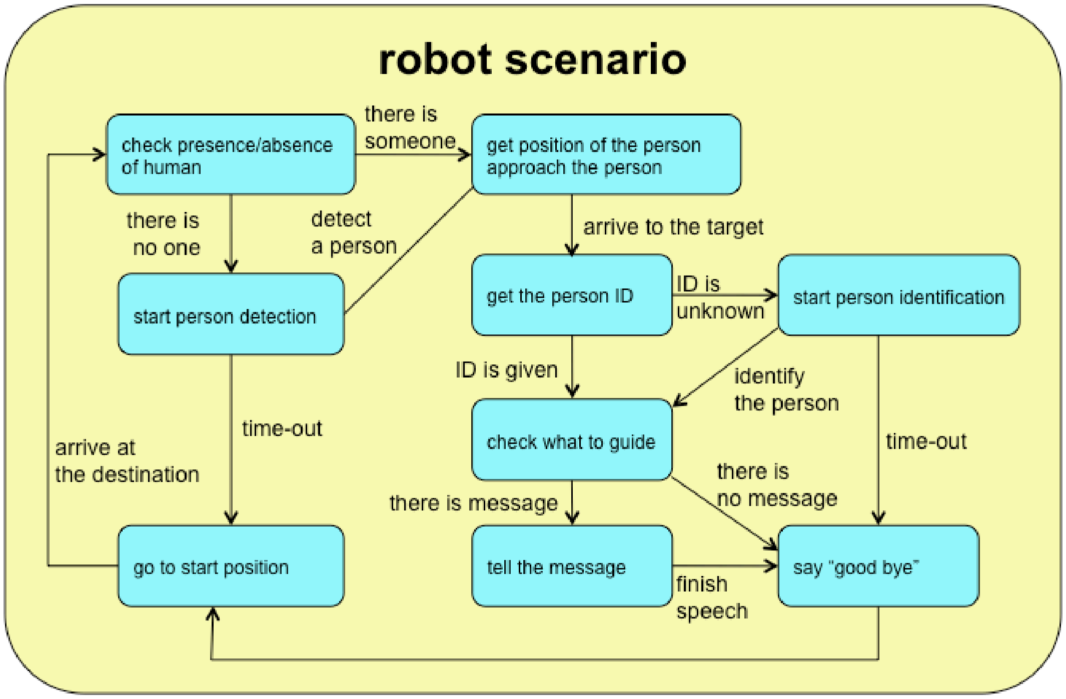


Fig.1: Example of robot scenario for robotic reception service.

In many service applications, the robot behavior rule is defined on a script, which is called a robot scenario. In the robot scenario, instructions to the functions and conditions based on the collected data are described in order to achieve the service task through the APIs that are specific to the robot. For example, the robotic reception service is constructed in the robot scenario as depicted in Fig. 1. In this scenario, when the robot detects someone, the robot approaches the person and tries to recognize who the person is, and then provides information appropriate to the person.

Fig.2 shows the messages exchanged between the service application and the robot for the scenario. In this case, one robot (Robot 1) detects human by camera and move by wheels and another robot (Robot 2) detects human by RFID tag and move by legs. Because of the difference in the APIs, the service application programmer must write the different scenarios for each robot respectively.

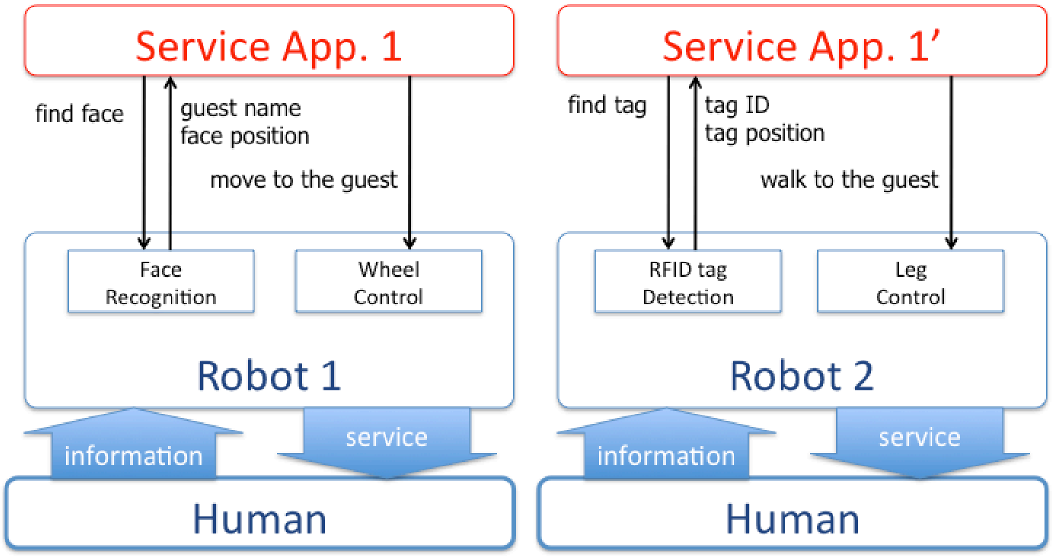


Fig.2: Conventional style of service application programming. Service application programmer must write service application programs for each robot independently because functions provided by each robot are different.

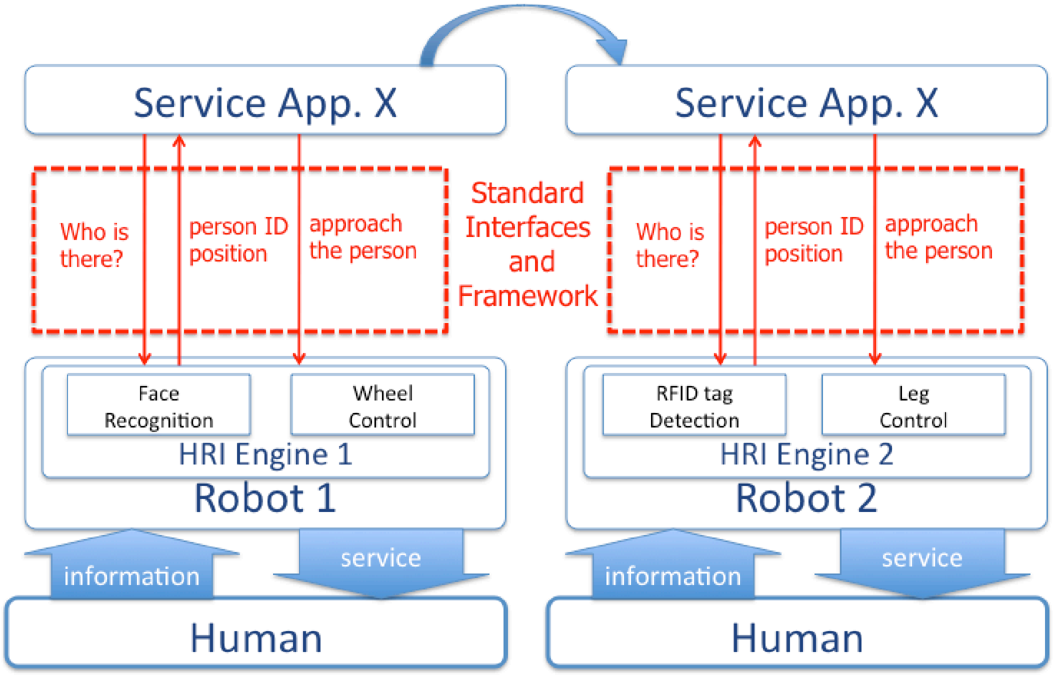


Fig.3: RoIS service application programming style. The same service application program works on different robot platforms with little modification.

If all the functional components are encapsulated in a package and the interfaces for obtaining information and controlling robots are standardized, they will enhance reusability of service application programs. A service application

program for a robot can work on the other robots regardless of the different robot platforms as depicted in Fig.3. Therefore, new general framework architecture is needed. We call the package that encapsulates functional components as HRI Engine and this new framework architecture as Robotic Interaction Service (RoIS).

For the RoIS concept, it is important to standardize mechanisms for the information and the instruction exchanged between the service application and the HRI Engine. From the point of view of a service application, there are generally two types of information to be exchanged, i.e. active information and passive information. The active information is a type of information that should be obtained when the service application needs the target information, such as "check presence/absence of human" and "get position of the person". The passive information is a type of information that should be provided when the target information is obtained or changed in the HRI Engine, such as "the robot has detected a person" and "the robot has arrived at the target".

In general, interface types of "Query" and "Event notification" are used for exchanging the active information and the passive information, respectively. To deal with "Event notification", a mechanism for "Event subscription / cancellation" is also required for selecting appropriate event notifications on demand. In addition, interface type of "Command" is naturally needed in order to instruct the HRI Engine to control its functions, such as "approach the person" and "go to start position". RoIS framework should also include these interface types, i.e., Event subscription/cancellation, Event notification, Query and Command as illustrated in Fig.4.

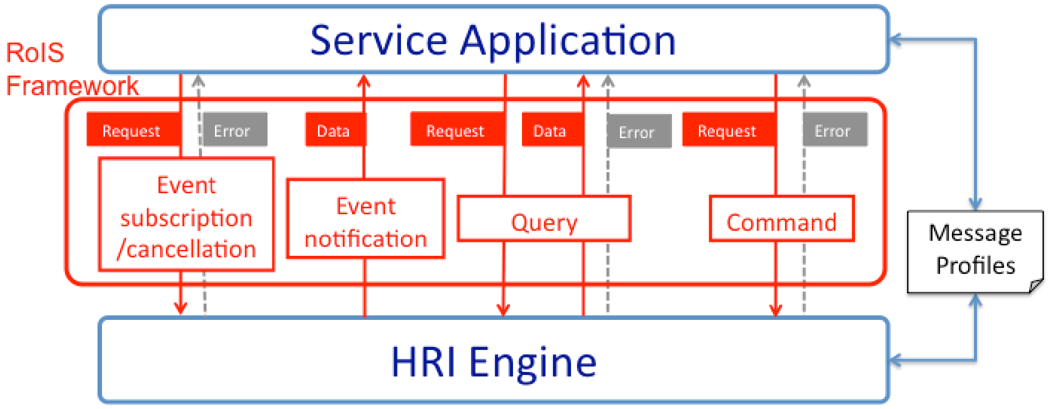


Fig.4:   RoIS Framework. In the framework, application communicates with HRI Engine through Event(s) subscription/cancellation, Event(s) notification, Query and Command.

For RoIS framework, it is desirable to define common messages of each interface type that are suitable for all HRI Engines as possible. These common messages should be proposed at least for person detection, person identification, and person location. There are common messages for each interface type such as,

- Event notification
  - Person detected
  - Person identified
  - Robot action completed
- Query
  - Person ID
  - Position of person / robot
  - Status of HRI Engine
- Command
  - Start / stop detection
  - Start / stop identification

In regard to these common messages, there must be a unique structure for RoIS framework.

On the other hand, there are also messages specific to each HRI Engine. For example in guidance service, there may be specific messages for each interface type such as, event notification for “speech recognition”, query for “presence / absence of human” and command for “approach the person” in addition to the common messages. Therefore, each HRI Engine should provide its message profiles with service applications.

Message profiles are composed of common message types and domain specific message types, for example, educational domain, navigation domain and healthcare domain, and each profile may include a list of available messages for each interface type, name of each message, data format for the information exchanged through the message and required argument(s) for using the message. It is required not only to define common messages but also to make a scheme to describe these message profiles so that the service application can make a query about the appropriate message as necessary.

To specify RoIS framework, it is desirable to consider an abstraction level of the messages for the interface types. For example, it should not include any parameters specific to robot platform. Also, the information managed by “Event subscription/cancelation,” “Event notification,” “Query” and “Command” must be represented suitable for service applications. The abstraction level appropriate to human robot interaction should be considered carefully by focusing on contents of the information and the robot control unique to human robot interaction.

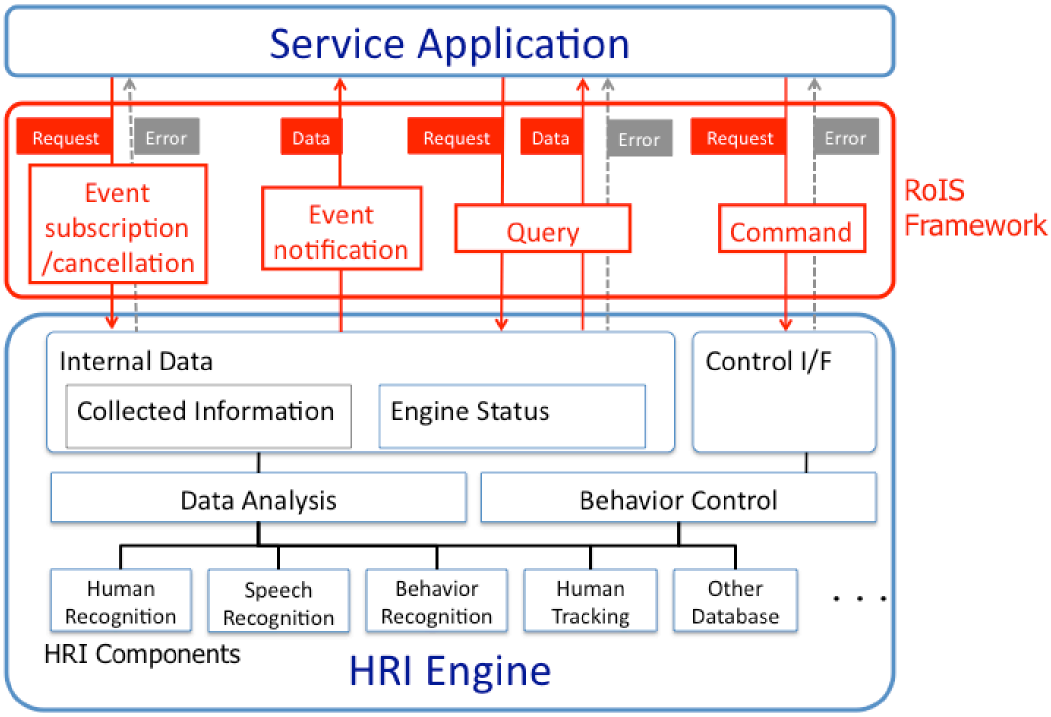


Fig.5: Example of RoIS Framework

Fig.5 illustrates a basic structure of RoIS Framework. HRI Engine integrates several functional components (HRI Components) and provides their functions with a service application through standardized interface. It collects and manages information of human around and environment by sensors, and provides collected data with the application on demand. As some HRI Components in HRI Engine provide robot control functions, the application can control robot through the engine’s interface.

Separation and encapsulation of HRI Components into HRI Engine and providing standardized interface to the components will enhance not only the reusability of components but also the efficiency of service application and component development.

Therefore, the scope of the specification solicited includes the definition of interfaces in between service applications and HRI Engine, that is, Event subscription/cancellation, Event notification, Query and Command, and the structure of data transmitted through each interface. Error notification sent from HRI Engine to application may also be included. However, internal data structure that depends on each HRI Engine implementation or data structures defined in existing specifications such as user identification should not be included in this specification.

Considering that HRI Engines depend on their robot platforms, the HRI Engine developers should be able to define interfaces between functional components inside their engine in their independent way. RoIS framework should not concern about HRI Engine inside. For example, one developer can use some other standardized framework, such as RTM, for inside HRI Engine, and the other developer can use their original method. Also, HRI Engine can access to the other applications and databases, such as location data and map data for path generation, by using other framework as needed. The same can be said for service application program inside. For example, in the case that the service application has to be corporate with the other application, such as network robot service, the service application can use other framework to access to the other application.

In summary, following items are required for RoIS framework.

- **<u>Interface between service application and HRI Engine</u>**
  - Interface to obtain information from HRI Engine according to the timing of the service application's needs
(Query)
  - Interface to receive information from HRI Engine according to the occurrences of the information in real time
(Event notification / subscription / cancellation)
  - Interface for the instruction to control HRI Engine functions
(Command)
- **<u>Definition of common messages for all HRI Engines</u>**

6.2 Scope of Proposals Sought

This RFP seeks proposals that specify RoIS framework, on top of which various robotic service applications are developed. This RFP targeted on human-robot interaction.

It is necessary to consider the following~~s~~ in the specification of RoIS framework.

(1) Overview of architecture that consists of RoIS framework, a robotic service application, HRI Engine that is a set of robotic components shall be provided (diagram or description for overview).

(2) The RoIS framework specification shall provide following interfaces between robotic service applications and HRI Engine.

- Interface to obtain information actively from HRI Engine
- Interface to receive information timely from HRI Engine
- Interface for the instruction to control HRI Engine functions

(3) The RoIS framework specification must be general enough to incorporate robotic components for various sensors, actuators and algorithms in HRI Engine.

(4) The RoIS framework specification shall satisfy interoperability and reusability. A HRI Engine should be able to be replaced with the other HRI engine with little efforts.

(5) The RoIS Framework specification shall specify common messages for all HRI Engines.

6.3 Relationship to Existing OMG Specifications

Submitters shall examine the following OMG specifications for possible benefit:

- Super Distributed Objects (SDO) Specification version 1.1 [formal/2008-10-11]
- Unified Modeling Language: Infrastructure version 2.1.2 [formal/2007-11-04]
- Unified Modeling Language: Superstructure version 2.1.2 [formal/2007-11-02]
- Lightweight CORBA Component Model 4.0 [formal/2006-04-01]
- Robotic Technology Component specification version 1.0 [formal/08-04-04]
- Robotic Localization Service version 1.0 [formal/2010-02-03]

6.4 Related Activities, Documents and Standards

Proposals may include existing systems, documents, user recognition service interface, and standards that are relevant to the problems discussed in this RFP. They can be used as background information for the proposal.

Example:

- IEEE Robotics and Automation Society, Technical Committee on Network Robot
- IEEE Robotics and Automation Society, Technical Committee on Programming Environment in Robotics and Automation
- ISO/ SC 37 Projects relate to ISO/IEC 19784-1(BioAPI Ver 2.0)
- ISO/TC184/SC2

6.5 Mandatory Requirements

(1) Proposals shall provide overview of architecture that consists of RoIS framework, a robotic service application, HRI Engine that is a set of robotic components (diagram or description for overview)

(2) Proposals shall specify a general mechanism for RoIS framework

- Interfaces between applications and HRI Engine and their types:
  - Interface to obtain information actively from HRI Engine
  - Interface to receive information timely from HRI Engine
  - Interface for the instruction to control HRI Engine functions
- Data structure for each interface type

(3) Proposals shall specify existing technologies to achieve functions required for RoIS framework.

(4) Proposal shall specify functions that cannot be achieved by existing technologies.

(5) Proposals shall specify the following common messages for all HRI Engines.

- Event notification
  - Person detected
  - Person identified
  - Robot action completed
- Query
  - Person ID
  - Position of person / robot
  - Status of HRI Engine
- Command
  - Start / stop detection
  - Start / stop identification

(6) Proposals shall provide a Platform Independent Model (PIM).

6.6 Optional Requirements

(1) Proposals may specify schemes for the functionalities listed below.

- Error handling for each interface type
- Returning command results and status

(2) Proposals may provide a schema to describe message profiles.

(3) Proposals may provide a Platform Specific Model (PSM) as C++, CORBA-specific model, XML or RTC for RoIS framework.

6.7 Issues to be discussed

These issues will be considered during submission evaluation. They should not be part of the proposed normative specification. (Place them in Part I of the submission.)

- Proposals shall demonstrate its feasibility by using a specific application based on the proposed model.
- Proposals shall demonstrate its applicability to existing robotic interaction service technologies.
- Proposals shall discuss simplicity of implementation.
- Proposals shall discuss the possibility to apply the proposed model to other fields of interest such as intelligent service robot applications
- Proposals shall specify on-the-wire protocol communication technology independent.
- Proposals shall discuss their relation and dependency to existing communication protocols or middlewares standards, such as CORBA [CORBA], DDS [DDS] or RTC [RTC].
- Proposals shall discuss the generality with respect to various sensors, actuators and algorithms in HRI Engine.

6.8 Evaluation Criteria

Proposals will be evaluated in terms of consistency in their specifications, feasibility and versatility across a wide range of different robot applications.

6.9 Other information unique to this RFP

None

6.10 RFP Timetable

The timetable for this RFP is given below. Note that the TF or its parent TC may, in certain circumstances, extend deadlines while the RFP is running, or may elect to have more than one Revised Submission step. The latest timetable can always be found at the OMG *Work In Progress* page at http://www.omg.org/schedules/ under the item identified by the name of this RFP. Note that "<month>" and "<approximate month>" is the name of the month spelled out; e.g., January.

Event or Activity	Actual Date
Preparation of RFP by TF	**24. May. 2010**
RFP placed on OMG document server	**24. May. 2010**

Approval of RFP by Architecture Board *Review by TC*	**21. June. 2010**
TC votes to issue RFP	**25. June. 2010**
LOI to submit to RFP due	**13. September. 2010**
Initial Submissions due and placed on OMG document server ("Three week rule")	**8. November. 2010**
Voter registration closes	**29, November, 2010**
Initial Submission presentations	**6. December. 2010**
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Three week rule")	
Revised Submission presentations	
Final evaluation and selection by TF *Recommendation to AB and TC*	
Approval by Architecture Board *Review by TC*	
TC votes to recommend specification	
BoD votes to adopt specification	

Appendix A References and Glossary Specific to this RFP

A.1 References Specific to this RFP

[RTC] Robotic Technology Component specification version 1.0, http://www.omg.org/spec/RTC/1.0/

A.2 Glossary Specific to this RFP

None

Appendix B General Reference and Glossary

B.1 General References

The following documents are referenced in this document:

[ATC] Air Traffic Control Specification, http://www.omg.org/technology/documents/formal/air_traffic_control.htm

[BCQ] OMG Board of Directors Business Committee Questionnaire, http://www.omg.org/cgi-bin/doc?bc/02-02-01

[CCM] CORBA Core Components Specification, http://www.omg.org/technology/documents/formal/components.htm

[CORBA] Common Object Request Broker Architecture (CORBA/IIOP), http://www.omg.org/technology/documents/formal/corba_iiop.htm

[CSIV2] [CORBA] Chapter 26

[CWM] Common Warehouse Metamodel Specification, http://www.omg.org/technology/documents/formal/cwm.htm

[DAIS] Data Acquisition from Industrial Systems, http://www.omg.org/technology/documents/formal/dais.htm

[EDOC] UML Profile for EDOC Specification, http://www.omg.org/techprocess/meetings/schedule/UML_Profile_for_EDOC_FTF.html

[EJB] "Enterprise JavaBeans™", http://java.sun.com/products/ejb/docs.html

[FORMS] "ISO PAS Compatible Submission Template". http://www.omg.org/cgi-bin/doc?pas/2003-08-02

[GE] Gene Expression, http://www.omg.org/technology/documents/formal/gene_expression.htm

[GLS] General Ledger Specification , http://www.omg.org/technology/documents/formal/gen_ledger.htm

[Guide] The OMG Hitchhiker's Guide,, http://www.omg.org/cgi-bin/doc?hh

[IDL] ISO/IEC 14750 also see [CORBA] Chapter 3.

[IDLC++] IDL to C++ Language Mapping, http://www.omg.org/technology/documents/formal/c++.htm

[MDAa] OMG Architecture Board, "Model Driven Architecture - A Technical Perspective", http://www.omg.org/mda/papers.htm

[MDAb] "Developing in OMG's Model Driven Architecture (MDA)," http://www.omg.org/docs/omg/01-12-01.pdf

[MDAc] “MDA Guide” (*http://www.omg.org/docs/omg/03-06-01.pdf*)

[MDAd] “MDA "The Architecture of Choice for a Changing World™"”, *http://www.omg.org/mda*

[MOF] Meta Object Facility Specification, *http://www.omg.org/technology/documents/formal/mof.htm*

[MQS] “MQSeries Primer”, *http://www.redbooks.ibm.com/redpapers/pdfs/redp0021.pdf*

[NS] Naming Service, *http://www.omg.org/technology/documents/formal/naming_service.htm*

[OMA] “Object Management Architecture™”, *http://www.omg.org/oma/*

[OTS] Transaction Service, *http://www.omg.org/technology/documents/formal/transaction_service.htm*

[P&P] Policies and Procedures of the OMG Technical Process, *http://www.omg.org/cgi-bin/doc?pp*

[PIDS] Personal Identification Service, *http://www.omg.org/technology/documents/formal/person_identification_service.htm*

[RAD] Resource Access Decision Facility, *http://www.omg.org/technology/documents/formal/resource_access_decision.htm*

[RFC2119] IETF Best Practices: Key words for use in RFCs to Indicate Requirement Levels, (*http://www.ietf.org/rfc/rfc2119.txt*).

[RM-ODP] ISO/IEC 10746

[SEC] CORBA Security Service, *http://www.omg.org/technology/documents/formal/security_service.htm*

[TOS] Trading Object Service, *http://www.omg.org/technology/documents/formal/trading_object_service.htm*

[UML] Unified Modeling Language Specification, *http://www.omg.org/technology/documents/formal/uml.htm*

[UMLC] UML Profile for CORBA, *http://www.omg.org/technology/documents/formal/profile_corba.htm*

[XMI] XML Metadata Interchange Specification, *http://www.omg.org/technology/documents/formal/xmi.htm*

[XML/Value] XML Value Type Specification, *http://www.omg.org/technology/documents/formal/xmlvalue.htm*

B.2 General Glossary

Architecture Board (AB) - The OMG plenary that is responsible for ensuring the technical merit and MDA-compliance of RFPs and their submissions.

Board of Directors (BoD) - The OMG body that is responsible for adopting technology.

Common Object Request Broker Architecture (CORBA) - An OMG distributed computing platform specification that is independent of implementation languages.

Common Warehouse Metamodel (CWM) - An OMG specification for data repository integration.

CORBA Component Model (CCM) - An OMG specification for an implementation language independent distributed component model.

Interface Definition Language (IDL) - An OMG and ISO standard language for specifying interfaces and associated data structures.

Letter of Intent (LOI) - A letter submitted to the OMG BoD's Business Committee signed by an officer of an organization signifying its intent to respond to the RFP and confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements.

Mapping - Specification of a mechanism for transforming the elements of a model conforming to a particular metamodel into elements of another model that conforms to another (possibly the same) metamodel.

Metadata - Data that represents models. For example, a UML model; a CORBA object model expressed in IDL; and a relational database schema expressed using CWM.

Metamodel - A model of models.

Meta Object Facility (MOF) - An OMG standard, closely related to UML, that enables metadata management and language definition.

Model - A formal specification of the function, structure and/or behavior of an application or system.

Model Driven Architecture (MDA) - An approach to IT system specification that separates the specification of functionality from the specification of the implementation of that functionality on a specific technology platform.

Normative – Provisions that one must conform to in order to claim compliance with the standard. (as opposed to non-normative or informative which is explanatory material that is included in order to assist in understanding the standard and does not contain any provisions that must be conformed to in order to claim compliance).

Normative Reference – References that contain provisions that one must conform to in order to claim compliance with the standard that contains said normative reference.

Platform - A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented.

Platform Independent Model (PIM) - A model of a subsystem that contains no information specific to the platform, or the technology that is used to realize it.

Platform Specific Model (PSM) - A model of a subsystem that includes information about the specific technology that is used in the realization of it on a specific platform, and hence possibly contains elements that are specific to the platform.

Request for Information (RFI) - A general request to industry, academia, and any other interested parties to submit information about a particular technology area to one of the OMG's Technology Committee subgroups.

Request for Proposal (RFP) - A document requesting OMG members to submit proposals to the OMG's Technology Committee. Such proposals must be received by a certain deadline and are evaluated by the issuing task force.

Task Force (TF) - The OMG Technology Committee subgroup responsible for issuing a RFP and evaluating submission(s).

Technology Committee (TC) - The body responsible for recommending technologies for adoption to the BoD. There are two TCs in OMG – *Platform TC* (PTC), that focuses on IT and modeling infrastructure related standards; and *Domain TC* (DTC), that focus on domain specific standards.

Unified Modeling Language (UML) - An OMG standard language for specifying the structure and behavior of systems. The standard defines an abstract syntax and a graphical concrete syntax.

UML Profile - A standardized set of extensions and constraints that tailors UML to particular use.

XML Metadata Interchange (XMI) - An OMG standard that facilitates interchange of models via XML documents.

Robotics-DTF Plenary Meeting
Wrap-up Session

June 22nd, 2010

Minneapolis, MN, USA

Hyatt Regency Hotel

Document Number

robotics/2010-06-01 Final Agenda (Tetsuo Kotoku)

robotics/2010-06-02 Jacksonville Meeting Minutes [approved] (Yoshihiro Nakabo and Jae-Yeong Lee)

robotics/2010-06-03 Opening Presentation (Tetsuo Kotoku)

robotics/2010-06-04 Revised dradt of Robotic Interaction Service (RoIS) Framework RFP (Toshio Hori)

robotics/2010-06-05 MARS Presentation of Dynamic Deployment and Configuration (DDC) for RTC RFP (copy of mars/2010-06-03) (Seung-Woog Jung)

robotics/2010-06-06 Errata of Dynamic Deployment and Configuration (DDC) for RTC RFP (copy of mars/2010-06-04) (Noriaki Ando)

robotics/2010-06-07 Revised draft of Dynamic Deployment and Configuration (DDC) for RTC RFP (copy of mars/2010-06-05) (Noriaki Ando)

robotics/2010-06-08 Future of the Deployment and Configuration Specification (William R. Otte)

robotics/2010-06-09 RTC Dynamic Deployment and Configuration (DDC) Specification (Seung-Woog Jung)

robotics/2010-06-10 2nd revised draft of Robotic Interaction Service (RoIS) Framework RFP (Toshio Hori)

Document Number (cont.)

robotics/2010-06-11 Draft JAUS-RTC White Paper (Laurent Rioux)

robotics/2010-06-12 Charter of the Modelling for Robotics Working Group (Laurent Rioux)

robotics/2010-06-13 Draft RFI on Models for Robotics (Laurent Rioux)

robotics/2010-06-14 Presentation of Robotic Interaction Service (RoIS) Framework RFP (Miki Sato)

robotics/2010-06-15 AB Reviewer's Comments for RoIS (Miki Sato)

robotics/2010-06-16 3rd revised draft of Robotic Interaction Service (RoIS) Framework RFP (Miki Sato)

robotics/2010-06-17 Wrap-up Presentation (Tetsuo Kotoku)

robotics/2010-06-18 Infrastructure WG Progress Report (Noriaki Ando)

robotics/2010-06-19 Robotic Functional Services WG Report (Toshio Hori)

robotics/2010-06-20 Modelling WG Report (Laurent Rioux)

robotics/2010-06-21 Roadmap for Robotics Activities (Tetsuo Kotoku)

robotics/2010-06-22 4th Draft of Robotic Interaction Service (RoIS) Framework RFP (Miki Sato)

robotics/2010-06-23 Convenience Document of Robotic Interaction Service (RoIS) Framework RFP with change bars (Toshio Hori)

robotics/2010-06-24 Errata of RoIS RFP (Toshio Hori)

robotics/2010-06-25 Robotic Interaction Service (RoIS) Framework RFP (Toshio Hori)

Document Number (cont.)

robotics/2010-06-26 Dynamic Deployment and Configuration for Robot Technology Component (DDC4RTC) RFP (copy of mars/2010-06-16) (Noriaki Ando)

robotics/2010-06-27 Errata (copy of mars/2010-06-17) (Noriaki Ando)

robotics/2010-06-28 Convenient Document of Dynamic Deployment and Configuration for Robot Technology Component (DDC4RTC) RFP with change bars (copy of mars/2010-06-18) (Noriaki Ando)

robotics/2010-06-29 Presentation of Dynamic Deployment and Configuration for Robot Technology Component (DDC4RTC) RFP (copy of mars/2010-06-19) (Noriaki Ando)

robotics/2010-06-30 Next Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)

robotics/2010-06-31 DTC Report Presentation (Tetsuo Kotoku)

robotics/2010-06-32 Minneapolis Meeting Minutes - DRAFT (Toshio Hori and Myung-Eun Kim)

New WG Charter

Modelling for Robotics

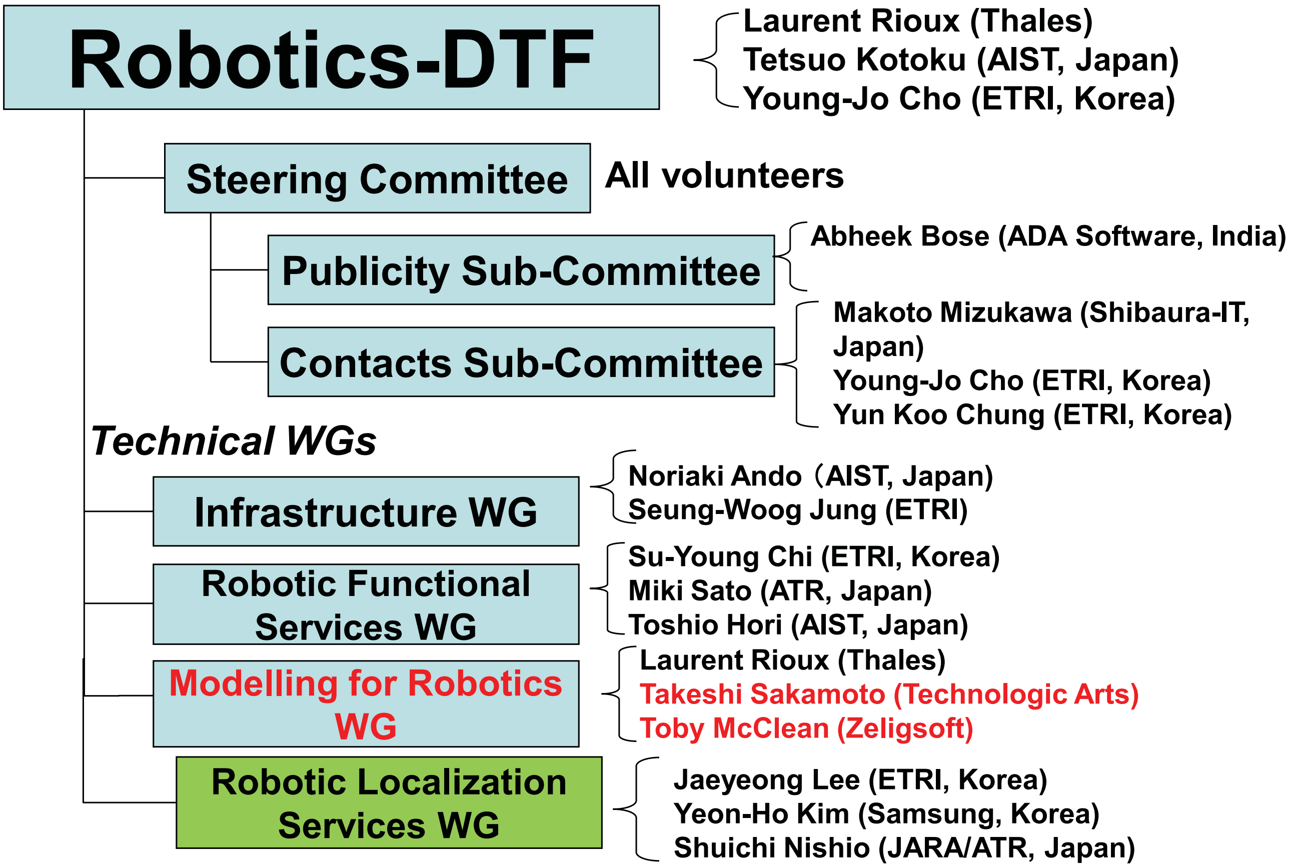
Charter [robotics/2010-06-12]

This working group is a sub-group of the Robotics Task Force. This working group is focusing on modelling engineering for robotics.

Chair:

- Takeshi Sakamoto (Technologic Arts)
- Laurent Rioux (THALES)
- Toby McClean (Zeligsoft)

Organization (from 22nd June, 2010)



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

Steering Committee — All volunteers

Publicity Sub-Committee
- Abheek Bose (ADA Software, India)

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

Technical WGs

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

Robotic Functional Services WG
- Su-Young Chi (ETRI, Korea)
- Miki Sato (ATR, Japan)
- Toshio Hori (AIST, Japan)

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

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

Plenary Attendee (17 participants)

- Geoffrey Biggs (AIST)
- Hugues VINCENT (Thales)
- Jae-Yeon Lee (ETRI)
- Laurent Rioux (Thales)
- Makoto Mizukawa (Shibaura-IT)
- Miki Sato (ATR)
- Miwako Doi (Toshiba)
- Myung-Eun Kim (ETRI)
- Noriaki Ando (AIST)
- Seung-Woog Jung (ETRI)
- Su-Young Chi (ETRI)
- Takeshi Sakamoto (Technologic Arts)
- Tetsuo Kotoku (AIST)
- Toshio Hori (AIST)
- William Otte (Vanderbilt Univ.)
- Yun Koo Chung (ETRI)

IROS2010 Workshop / Tutorial

IEEE/RSJ International Conference on Intelligent Robots and Systems
Aug 18-22 Taipei, Taiwan
http://www.iros2010.org.tw/

- T02 (Aug.18 PM)
  Introduction to OpenRTM-aist-1.0.0
- W08 (Aug.18 full day)
  Workshop on Standardization for Service Robots: Current Status and Future Directions
- W18 (Aug. 22 PM)
  Best Practice in Robot Control Architectures for Service Robots

Infrastructure WG Progress Report

(Minneapolis meeting)

Noriaki Ando (AIST)

robotics/2010-06-18

Topics of This Meeting

- RTC DDC RFP 3rd review process
  - Infra. WG meeting (on Sunday)
  - 3rd review in MARS (on Monday morning)
    - mars/10-05-07, mars/10-06-04,05
  - AB Plenary (on Monday afternoon)
  - MARS meeting for voting (on Thursday)
  - AB Plenary (on Thursday)

Comments from Vincent (1/4)

- The objective, problem statement and scope of proposal sought are clear and sound.
- The fact that this is a request for an extension of DEPL should be nevertheless emphasized.
- Section 6.1 need to more clearly specify what a RTC profile is made of.

Comments from Vincent (2/4)

- Requirement 6.5.1:
  - Please, replace "services for" with "interfaces to services for“
  - This requirement need to list what capabilities are expected:
    - storing, searching and retrieving RTC,
    - storing, searching and retrieving RTC-based applications
    - RTC registration
    - and others you (ie the TF) may have in mind

Comments from Vincent (3/4)

- Requirement 6.5.2:
  - Please, replace "means" with "interfaces“
  - This requirement should also list what capabilities are expected.
- Requirement 6.5.3:
  - What do you mean by "reuse or extend" D&C? There are a lot of things in D&C:
    - terms and definitions
    - a PIM
    - a UML profile for D+C Tool support
    - a list of actors and a deployment process
    - a PSM for CCM
  - All these things could be of interest for reusing or extending in your context. Please specify what you mean.

Comments from Vincent (4/4)

- Requirement 6.5.4:
  - the "..." need to be completed
  - I guess that the proposal should also specify what the units used for TC characteristics are.
- Requirement 6.5.5:
  - add "such as" at the end of the sentence and complete.
- There is no requirement about "RTC profile" which seems to be central in the description; I believe this is an oversight.
- Section 6.8: Add that the reuse of D&C is mandatory and that, with that respect, the more is the better.
- Section 6.10: Fix the dates in the table.

Comments from Andrew

- The requirements are clearly set out. Related specifications seem appropriate. The RFP is well-written in idiomatic English.
- Only small issue - we'll find out when the 3Q2011 OMG meeting is taking place, so that we can put the appropriate revised submission date in the timetable.

Current Status

- Two organizations are interested in the RTC-DDC RFP
  - Remedy IT & Vanderbilt University
  - Thales
- Monday meeting
  - William Otte’s presentation about new generic dynamic D&C architecture
- Tuesday meeting
  - Infrastructure WG had a discussion with William and Laurent.

Agreements

- We changed the name of the proposal
  - DDC4RTC
    - Dynamic Deployment and Configuration for Robotic Technology Component
- RFP process
  - DDC4RTC and generic dynamic D&C will be done in parallel
    - RemedyIT, Vanderbilt Univ., Zeligsoft and Thales will start a new working group in MARS to make new generic dynamic D&C standard
    - Some people from RemedyIT, Vanderbilt Univ., Zeligsoft and Thales will join the DDC4RTC RFP process and give some comments and their ideas in order to make some consistency

Next

On Thursday

- MARS meeting for review and voting
- AB Plenary Meeting

and

On Friday

- TC approval

Robotic Functional Service WG Report

2010/06/22

Status of RoIS Framework RFP draft

- Initial draft [robotics/2010-05-01] was submitted 4W before the meeting (May 25)
- According to AB comments from Hugues Vincent and Andrew Watson, the draft was revised [robotics/2010-06-04] in the Sunday meeting (June 20)
- 2nd Review of RFP draft in Robotics-DTF in Monday morning (June 21)
- According to the comments in the review, the draft was revised [robotics/2010-06-10]

Status of RoIS Framework RFP draft (cont.)

- AB Plenary in Monday afternoon
- According to the comments in the AB Plenary, the draft was revised [robotics/2010-06-16]. Errata of the draft was created [robotics/2010-06-15]

- Review and voting on the draft in Thursday morning
- AB Plenary in Thursday afternoon
- TC Approval on Friday

Documents are/will be available on the OMG server.

FYI: Comments from Mr. Vincent (1)

- The figure 3 as well as the 5 following paragraphs are too broad: almost all API in the world can be split in these four types: Event subscription, Event notification, Query and Command.

→We have added an explanation and a figure (new figure 1) of robot scenario concept to clarify the necessity for the four interface types. Two paragraphs before figure 4 (previous figure 3) are also corrected for this purpose. The most important point of this RFP is that we should standardize the content and handling of messages for these interfaces. That is, it is desirable to define common interface messages specific to human-robot interaction. To express it in a better way, we added some examples of such common messages after figure 4.

FYI: Comments from Mr. Vincent (2)

- The concept of "Profiles of interface" as well as how to "manage" them are unclear. These concepts must be made clearer in the RFP before continuing the process.

→We changed the words "profiles of interface" to “message profiles” and explained it in detail after figure 4.

FYI: Comments from Mr. Vincent (3)

- Section 6.2 - 1st paragraph: "It is the target for ... human." -> "This RFP is targeted on human-robot interaction”

→We corrected the sentence.

- Section 6.2 - 1st bullet (1) AND Section 6.5 Requirement (1):

  What do you mean by “overall architecture”? The “architecture” term is too broad and has too many uses to let it alone without clarification - Please specify this requirement. Moreover, are yor sure you want to standardize such an "architecture" for RoIS? Isn’t it implementation-dependant?

→ We have corrected the expression "overall architecture" to clarify the requirement.

FYI: Comments from Mr. Vincent (4)

- Section 6.2 - 5th bullet (5):

  As already said, this is too broad for an RFP and can simply be removed from it.

→We have unified (2) and (5) into (2) and corrected according to the above corrections.

- Section 6.5 - Requirement (2): "at least one PSM"

  My guess is that you don't want whatever PSM so please specify what you want: RTC and XML?

→We have specified the requirement for PSM and changed to the optional requirement.

FYI: Comments from Mr. Vincent (5)

- Section 6.5 - Requirements (3) to (5)

  As already said the concepts of profile and profile management are not clear enough to keep this like that.

→We have dropped the requirement for “schema to manage profile” and separated these requirements to the requirements for “schema to describe message profile” and “common messages”. The requirement for “schema to describe message profile” is changed to the optional requirement.

- Section 6.7: Proposals should also be requested to discuss about their generality wrt various sensors, actuators and algorithms (cf section 6.2 - bullet (3)).

→ We have added the last bullet based on the comment.

FYI: Comments from Mr. Vincent (6)

- I'm puzzled by the lack of any specific interfaces you want to find in the future standard. For instance, I was waiting for a list of useful interfaces such as those found in figure 2: "Is there somebody?", "Get Person ID", "Approach the person"… If you really have no list for this, I believe that you need first either to work on it inside your TF or to draft an RFI to request this list before redrafting this RFP.

→Although we discussed about it in our TF through workshops with several robot vendors, it is difficult to fix a list of such messages without hearing more concrete opinions. We should discuss with submitters based on their proposals. Therefore, we have listed it in the the mandatory requirements.

FYI: Comments from Mr. Andrew (1)

1. Does the RFP seek to standardise a complete set of domain abstractions for the domestic robotic domain (e.g. person, speech, pick-up, carry, put-down, etc), or just to standardise a framework within which these concepts can be expressed?

→This RFP seeks the latter.

2. The RFP talks extensively about events. Does it direct submitters to re-use any of the existing event frameworks that OMG as already standardised?

→Yes. We expect to use existing communication protocols and middlewares to achieve this framework as possible. Therefore, we have added the requirements in Section 6.5 (3) (4)to specify the classification of the required functions into the existing standards and the other.

FYI: Comments from Mr. Andrew (2)

3. The Mandatory requirements section talks about events, but barely mentions the important issue of domain abstractions and mechanisms for defining new abstractions.

→We have explained about the issue as a part of “message profiles” and mentioned the requirement for message profiles in Section 6.6 (2).

4. Do you want the submitters to target any particular platform with their PSM? If so, which one?

→We have specified the requirement for PSM in Section 6.6 (3).

Modelling WG report

L.Rioux (THALES)

Activities at this meeting

- Working group charter definition
- White Paper RTC-JAUS review
  - Final version in july.
- Draft RFI « Models for robotics »
  - Plan to be issues in december 10.

promotion

- RoSym Workshop
  - 1st international workshop on MBE for robotics
    - Deadline 26 july 2010
    - In conjunction with models'2010
  - Website:

  http://www.artist-embedded.org/artist/RoSym-2010

Roadmap for Robotics Activities

robotics/2010-06-21

Item	Status	Jacksonville FL Mar-2010	Minneapolis MN Jun-2010	Cambridge MA Sep-2010	Santa Clara CA Dec-2010	Washington DC Mar-2011	? Jun-2011	POC / Comment
Flyer of Robotics-DTF [Publicity Sub-Committee]	**Suspended**							
Robot Interaction Service (RoIS) Framework RFP [Robotic Functional Services WG]	**In Process**	**1st review RFP**	**2nd Review & RFP issue**		**Initial Submittion**		**Revised Submittion & Voting**	
Dynamic Deployment and Configuration for RTC (DDC4RTC) RFP [Robotic Infrastructure WG] in MARS	**In Process**	**2nd Review**	**3rd Review & RFP issue**		**Initial Submittion**		**Revised Submittion & Voting**	Sponsor: MARS
JAUS-RTC White Paper [Modelling for Robotics WG]	**In Process**	**discussion**	**1st review**		**2nd Review & issue**			
Models for Robotics RFI [Modelling for Robotics WG]	**In Process**		**1st review**		**2nd Review & issue**			
Robotic Map Services RFP [Robotic Functional Services WG]	**Planned**							IEEE R&A?
etc…	**Future**							
Robotics Information Day [Technology Showcase]	**Future**							
RLS Revision Task Force	**In Process**							will go to ISO/TC211

Related Events

Chu-suk
(Special Holidays in Korea)

Object Management Group

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

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

Request for Proposal

Robotic Interaction Service (RoIS) Framework RFP

OMG Document: robotics/2010-06-22

Letters of Intent due: September 13, 2010
Submissions due: November 8, 2010

Objective of this RFP

This RFP solicits proposals for a Platform Independent Model (PIM) and a Platform Specific Model (PSM) for robotic interaction service (RoIS) that specify

- common interfaces between robotic service applications and components that provide functions for performing human-robot interaction.
- data structures for each interface.

For further details see Chapter 6 of this document.

6.0 Specific Requirements on Proposals

6.1 Problem Statement

Service robots that provide~~s~~ ~~some~~ services to people in daily life will become more and more popular in the robotics market. These service robots provide appropriate services through human-robot interaction (HRI). E~~For e~~xample~~,~~ ~~there are~~ robotic services ~~such as,~~include;

- Reception service
- Guide service
- Home security service
- Childcare robot service
- Elder person daily watching service

A s~~S~~ervice application is provided as a set of robot behaviors. The robot's behavior ~~rule~~ is defined based on the information collected from humans or environments. ~~The~~ I~~I~~nformation is collected ~~by~~ using functions such as:~~,~~

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

Generally~~,~~ ~~several sensors and actuators are equipped with~~ the robot body, or the environment where the robot provides the service, is equipped with several sensors and actuators~~provides the service~~. The service application programmer ~~describes~~ defines procedures for the robot's actions ~~and relation between the action~~based (in part) ~~and~~ on the information obtained by these sensors ~~in the service application program~~. However, different robots may be equipped with ~~the~~ a ~~various~~ variety of types of sensors and actuators ~~are equipped with various robots~~, and moreover~~,~~ the application program interfaces (APIs) of ~~each~~ robots from different vendors could vary ~~are different by robot vendors~~ even if ~~their~~ they use the same sensor types ~~are the same~~. ~~Therefore~~As a result, an application program developed for one specific robot ~~will~~ may not run on an~~the~~ other robot. This lack of application portability is an obstacle to the success of the ~~This is one of the reasons of inefficiency in~~ robot industry.

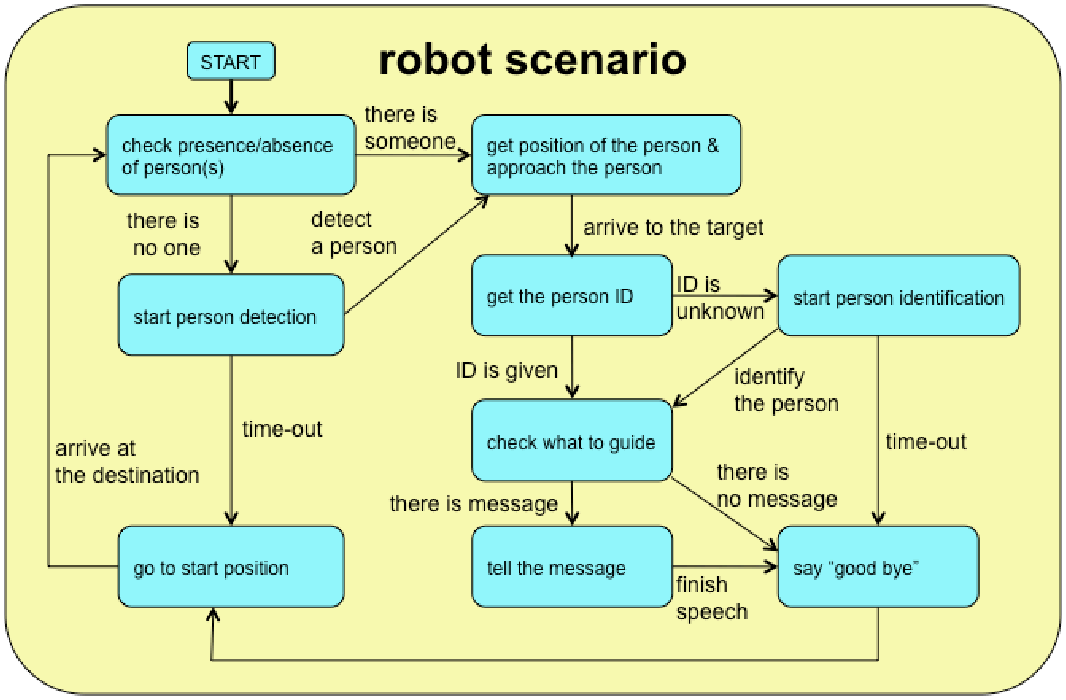


Fig.1: Example of robot scenario for robotic reception service.

In many service applications~~,~~ the robot's behavior ~~rule~~ is defined ~~on~~ using a script~~,~~ ~~which is~~ called a "robot scenario". ~~In the robot scenario~~This defines how to achieve the service task through the APIs that are specific to the robot, ~~instructions to the functions and conditions~~ based on the collected data ~~are described in order to achieve the service task through the APIs that are specific to the robot~~. For example, see the robotic reception service ~~is constructed in the robot scenario as~~ depicted in Fig. 1. In this scenario, when the robot detects someone~~,~~ ~~the robot~~it approaches the person, ~~and~~ tries to recognize who the person is, and then provides appropriate information ~~appropriate~~ to the person.

Fig.2 shows the messages exchanged between the service application and the robot ~~for~~ in this~~e~~ scenario. In this case, one robot (Robot 1) detects the human ~~by~~ using a camera and moves ~~by~~ using wheels, ~~and~~ while another robot (Robot 2) detects the human ~~by~~ using an RFID tag and moves ~~by~~ using legs. Because of the resulting difference in the APIs~~,~~ the service application programmer ~~must~~ would have to write ~~the different~~separate scenarios for each robot ~~respectively~~.

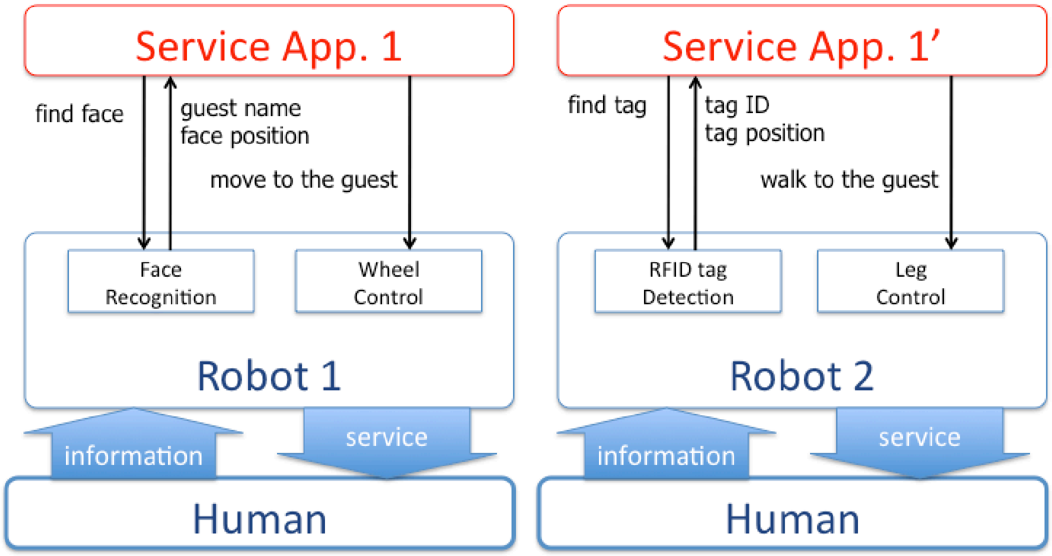


Fig.2: Conventional style of service application programming. Service application programmer must write service application programs for each robot independently because functions provided by each robot are different.

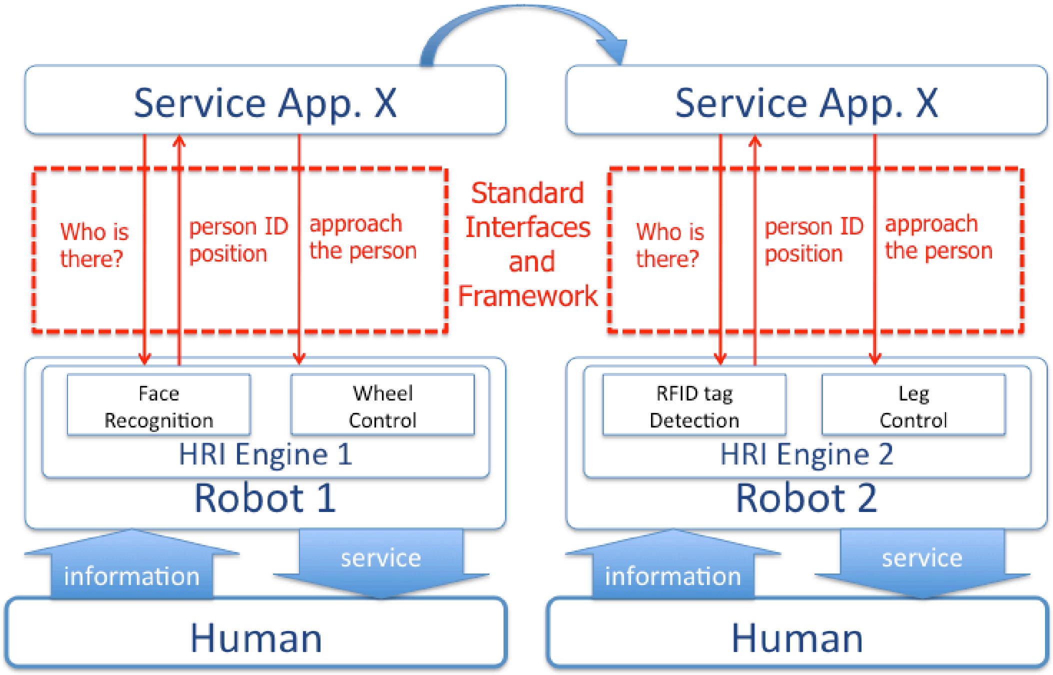


Fig.3: RoIS service application programming style. The same service application program works on different robot platforms with little modification.

By contrast, i~~I~~f ~~all~~ the interfaces between the service application and the robot’s functional components ~~are~~are standardized, using a higher level of abstraction based on concepts from the target domain (such as “detect person”) rather than implementation dependent concepts (like “detect RFID”), -it would be possible to ~~encapsulated in a package and the interfaces for obtaining information and controlling robots are standardized, they will~~ enhance reusability of service application programs. A service application program for a robot ~~can~~ could then work on ~~the~~ other robots regardless of the different robot platforms, as depicted in Fig.3. We term this proposed ~~Therefore,~~ new general framework architecture, ~~is needed. We call the package~~ that encapsulates functional components such as HRI Engines, ~~and this new framework architecture as~~the Robotic Interaction Service (RoIS).

~~For~~ When using the RoIS concept~~,~~ it is important to standardize mechanisms for the information and ~~the~~ instructions exchanged between the service application and the HRI Engine. From the point of view of a service application, there are generally two types of information to be exchanged, ~~i.e.~~ A~~a~~ctive I~~i~~nformation and P~~p~~assive I~~i~~nformation. A~~The a~~ctive I~~i~~nformation is ~~a type of information that should be~~actively solicited by the service application~~obtained when the service application needs the target information~~, ~~such as~~for example “check presence/absence of human” and “get position of the person”. P~~The p~~assive I~~i~~nformation is ~~a type of information that should be~~ provided when relevant ~~the target information~~data is obtained or changed in the HRI Engine, ~~such~~ for example ~~as~~ “the robot has detected a person” ~~and~~ or “the robot has arrived at the target”.

In general, interface types of “Query” and “Event notification” are used for exchanging ~~the~~ A~~a~~ctive I~~i~~nformation and ~~the~~ P~~p~~assive I~~i~~nformation~~,~~ respectively. To deal with “Event notification”, a mechanism for “Event subscription / cancellation” is also required for selecting appropriate event notifications on demand. In addition, the ~~interface type of~~ “Command” interface type is naturally needed in order to instruct the HRI Engine to control its functions, such as “approach the person” and “go to start position”. The RoIS framework should also include these interface types, i.e., Event subscription/cancellation, Event notification, Query and Command as illustrated in Fig.4.

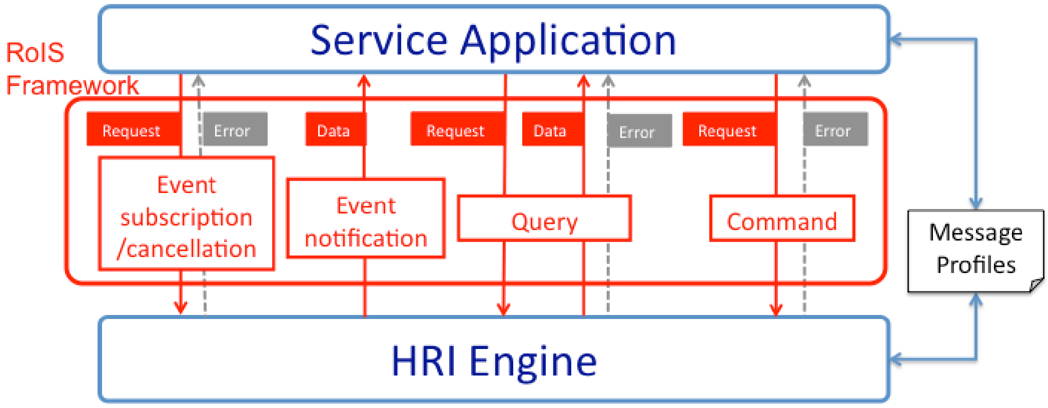


Fig.4: RoIS Framework. In the framework, application communicates with HRI Engine through Event(s) subscription/cancellation, Event(s)

notification, Query and Command.

~~For~~ Within the RoIS framework~~,~~ it is desirable to define ~~common~~ messages for each interface type that are common to all service domains (and therefore also common to all~~of~~ ~~each interface type that are suitable for all~~ HRI Engines)~~ as possible~~. ~~These common messages should be proposed at least for person detection, person identification, and person location.~~ These common messages should include:

~~There are common messages for each interface type such as,~~

- Event notification
  - Person detected
  - Person identified
  - ~~Robot~~ Command action completed
- Query
  - Person ID
  - Position of person / robot
  - Status of HRI Engine
- Command
  - Start / stop person detection
  - Start / stop person identification

~~In regard to these common messages, there must be a unique structure for RoIS framework.~~

On the other hand, there ~~are~~ must also be message types~~s~~ that are specific to each service domain and HRI Engine. For example, in the ~~guidance~~ reception service~~,~~ there may be specific messages for each interface type, such as:~~,~~ message “speech recognised” ~~event notification~~ for the “event notification” interface~~“speech recognition”~~, message ~~query for~~ “presence / absence of human” for the query interface, and message ~~command for~~ “approach the person” for the command interface.~~in addition to the common messages.~~ ~~Therefore, each~~ HRI Engines should be able to inform service applications about which message profiles they support.
~~should provide its message profiles with service applications.~~

Message profiles are specific to particular application domains, such as the educational domain, navigation domain or healthcare domain. Each message profile is ~~Message profiles are~~ composed of both common message types and domain specific message types. ~~, for example, educational domain, navigation domain and healthcare domain, and~~ E~~e~~ach profile ~~may~~ includes a list of available messages for each interface type, specifying the name of each message, data format for the information exchanged ~~through~~ using the message and required argument(s) for ~~using~~ the message. An RoIS specification must not only define ~~It is required not only to define~~ common messages but also ~~to make a scheme~~specify how to describe these message profiles, so that the service application can ~~make a query about the~~discover the ~~appropriate ~~supported messages as necessary.

~~To ~~In specifying the RoIS framework, ~~it is desirable to consider ~~the~~an~~ abstraction level of the messages for the interface types should be considered. For example, ~~it ~~the RoIS framework should not include any parameters specific to a particular robot platform. The abstractions should be carefully designed to be appropriate for service robot control (as opposed to other robotic applications, such as industrial robots).

~~ Also, the information managed by "Event subscription/cancelation," "Event notification," "Query" and "Command" must be represented suitable for service applications. The abstraction level appropriate to human robot interaction should be considered carefully by focusing on contents of the information and the robot control unique to human robot interaction.~~

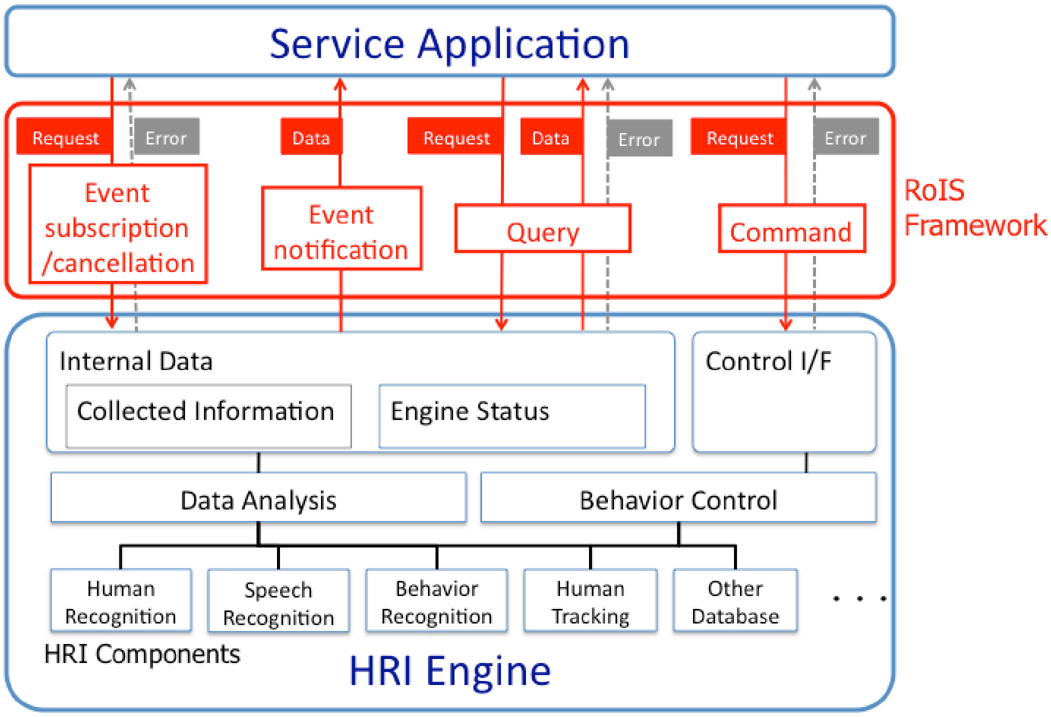


Fig.5: Example of RoIS Framework

Fig.5 illustrates the~~a~~ basic structure of the RoIS Framework. The HRI Engine integrates several functional components (the HRI Components) and provides their functions ~~with ~~to a service application through standardized interfaces. It collects and manages information ~~of ~~about nearby humans ~~around ~~and its environment ~~by ~~via sensors, and provides collected data ~~with ~~to the application on demand. As some of the HRI Components in the HRI Engine provide robot control functions, the application can control the robot through the engine's interfaces.

Separation and encapsulation of HRI Components ~~into ~~within the HRI Engine, and providing standardized interface to the components, will enhance ~~not only~~ the reusability of HRI C~~c~~omponents, and also improve the ~~but also the~~ efficiency of both service application and component development.

~~Therefore, t~~The scope of the specification solicited includes the definition of interfaces ~~in~~ between service applications and the HRI Engine (i.e. ~~, that is,~~ Event subscription/cancellation, Event notification, Query and Command), and the structure of data transmitted through each interface~~ –~~. Error notification sent from the HRI Engine to applications may also be ~~included~~specified. However, internal data structures that depend~~s~~ on each HRI Engine implementation, or data structures defined in existing specifications ~~such as~~(e.g the BioAPI specification for user identification), should not be included in this specification.

Considering that HRI Engines depend on their robot platforms, ~~the~~ HRI Engine developers should be able to define interfaces between functional components ~~inside~~ within ~~their~~ each engine ~~in their~~ independently~~ way~~. The ~~–~~RoIS framework standard should not ~~concern~~ address the internals of the ~~about~~ HRI Engine~~ inside~~. ~~–~~For example, ~~one~~ a developer ~~can~~ could use some other standardized framework, such as OMG Robotic Technology Components~~M~~, ~~for~~ inside an HRI Engine, ~~and~~ while an~~the~~ other developer ~~can~~ could use ~~their~~ a proprietary ~~original method~~framework. In addition, ~~Also,~~ HRI Engines ~~can~~ may access ~~to the~~ other applications and databases, such as location ~~data~~ and map data for path generation, ~~by~~ using other appropriate frameworks~~ as needed~~. Similarly, the RoIS framework must not constrain the internal implementation details of ~~ The same can be said for~~ service application~~-~~s~~program inside~~. ~~–~~For example, ~~in the case that the~~if a service application ~~has~~ must ~~to be corporate~~communicate with the ~~other~~ another application, such as a network robot service, the service application ~~can~~ may use another framework to access ~~to the~~ that ~~other~~ application.

~~In summary, the following items are requirementsd for the RoIS framework are:.~~

- ~~**Interfaces between service application and HRI Engine**~~
- ~~Interface to obtain information from HRI Engine according to the timing of the service application's needs~~
  ~~(Query)~~
- ~~Interface to receive information from HRI Engine according to the occurrences of the informationtriggered by in real- time events~~
  ~~(Event notification / subscription / cancellation)~~
- ~~Interface for the instructions to control HRI Engine functions~~
  ~~(Command)~~
- ~~**Definition of common messages for all HRI Engines**~~
  ~~Definition of a set of messages that are common to all service robot domains~~

6.2 Scope of Proposals Sought

This RFP seeks proposals that specify a~~n~~ RoIS framework, on top of which various service robot~~ic service~~ applications are developed.

~~This RFP targetsed on human-robot interaction.~~The scope of proposals is summarized in the following items:

- Interfaces between service application and HRI Engine
  - ▸ Interface to obtain information from HRI Engine according to the timing of the service application's needs (Query)
  - ▸ Interface to receive information from HRI Engine triggered by real-time events (Event notification / subscription / cancellation)
  - ▸ Interface for instructions to control HRI Engine functions (Command)
- Definition of common messages for all HRI Engines
  - ▸ Definition of a set of messages that are common to all service robot domains

~~It is necessary to consider Tthe followings must be provided in the specification of the RoIS framework:.~~

~~(1)A diagram or description giving an oOverview of the architecture, including that consists ofthe RoIS framework, a robotic service application and, HRI Engine (composed of that is a set of robotic components). shall be provided (diagram or description for overview).~~

~~(2)A specification of the The RoIS framework specification shall provide following interfaces between robotic service applications and the HRI Engine:.~~

- ~~▸Interface to obtain information actively from HRI Engine~~
- ~~▸Interface to receive information timelynotification of real-time events from the HRI Engine~~
- ~~▸Control iInterface for the instruction to controlto send commands to HRI Engine functions~~

~~(3) A specification of common messages for all HRI Engines.~~

~~▸ The specification shall meet the following criteria:~~

~~(3)(1) The RoIS framework specification must bBe general enough to incorporate robotic components for various sensors, actuators and algorithms in HRI Engine.~~

~~(4)(2) The RoIS framework specification shall sSatisfy interoperability and reusability, to allow. Aan HRI Engine should to be able to be replaced with the another HRI engine without difficulty little efforts.~~

~~(5)The RoIS Framework specification shall specify common messages for all HRI Engines.~~

6.3 Relationship to Existing OMG Specifications

Submitters shall examine the following OMG specifications for possible benefit:

- Super Distributed Objects (SDO) Specification version 1.1 [formal/2008-10-01~~11~~]
- Unified Modeling Language: Infrastructure version 2.3~~1.2~~ [formal/2010~~07~~-05~~11~~-034]
- Unified Modeling Language: Superstructure version 2.3~~1.2~~ [formal/2010~~10~~-05~~5~~-05~~3formal/2007-11-02~~]
- Lightweight CORBA Component Model 4.0 [formal/~~20~~06-04-01]
- Robotic Technology Component specification version 1.0 [formal/08-04-04]
- Robotic Localization Service version 1.0 [formal/2010-02-03]

6.4 Related Activities, Documents and Standards

Proposals may include existing systems, documents, user recognition service interface, and standards that are relevant to the problems discussed in this RFP. They can be used as background information for the proposal.

Examples:

- IEEE Robotics and Automation Society, Technical Committee on Network Robot
- IEEE Robotics and Automation Society, Technical Committee on Programming Environment in Robotics and Automation
- ISO/ SC 37 Projects- relate to ISO/IEC 19784-1(BioAPI Ver 2.0)
- ISO/TC184/SC2 Robots and robotic devices

6.5 Mandatory Requirements

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

(2) Proposals shall provide a specification of the following interfaces between robotic service applications and the HRI Engine:

- Interface to obtain information actively from HRI Engine
- Interface to receive notification of real-time events from the HRI Engine
- Control interface to send commands to HRI Engine functions

(3) Proposals shall specify data structure for each interface.

(4) Proposals shall provide a specification of the following common messages for all HRI Engines.

- Event notification
    - Person detected
    - Person identified
    - ~~Robot~~Command action completed
- Query
    - Person ID
    - Position of person / robot
    - Status of HRI Engine
- Command
    - Start / stop person detection
    - Start / stop person identification

(5) The specification shall meet the following criteria:

- Be general enough to incorporate robotic components for various sensors, actuators and algorithms in HRI Engine.
- Satisfy interoperability and reusability, to allow an HRI Engine to be replaced with another without difficulty.

(6) Proposals shall specify existing technologies to achieve functions required for RoIS framework.

(7) Proposal shall specify functions that cannot be achieved by existing technologies.

(8) Proposals shall provide a Platform Independent Model (PIM).

~~(1)Proposals shall provide an overview of the propsed architecture, that consistings of the RoIS framework, a robotic service application, HRI~~

~~Engine that is a set of robotic components (diagram or description for overview)~~

~~(2)Proposals shall specify a general mechanism for RoIS framework~~

- ~~Interfaces between applications and HRI Engine and their types:~~
  - ~~Interface to obtain information actively from HRI Engine~~
  - ~~Interface to receive information timely from HRI Engine~~
  - ~~Interface for the instruction to control HRI Engine functions~~
- ~~Data structure for each interface type~~

~~(3)Proposals shall specify existing technologies to achieve functions required for RoIS framework.~~

~~(4)Proposal shall specify functions that cannot be achieved by existing technologies.~~

~~(5)Proposals shall specify common messages for all HRI Engines~~

~~(6)Proposals shall provide a Platform Independent Model (PIM).~~

6.6 Optional Requirements

(1) Proposals may specify schemes for the functionalities listed below.

- Error handling for each interface type
- Returning command results and status

(2) Proposals may provide a schema to describe message profiles.

(3) Proposals may provide a Platform Specific Model (PSM) as C++, CORBA-specific model, XML or RTC for RoIS framework.

6.7 Issues to be discussed

These issues will be considered during submission evaluation. They should not be part of the proposed normative specification. (Place them in Part I of the submission.)

- Proposals shall demonstrate ~~its~~ their feasibility by ~~using~~ giving a specific example application based on the proposed model.
- Proposals shall demonstrate ~~its~~ their applicability to existing robotic interaction service technologies.

- Proposals shall discuss simplicity of implementation.
- Proposals shall discuss the possibility ~~to~~ of applying the proposed model to other fields of interest such as intelligent service robot applications
- Proposals shall ~~shall~~ show that ~~specify~~ they are independent of on-the-wire protocol communication technology ~~independent~~.
- Proposals shall discuss their relationship to and dependency ~~to~~ on existing ~~communication protocols or~~ middlewares standards, such as CORBA [CORBA], DDS [DDS] or RTC [RTC].
- Proposals shall discuss their generality with respect to ~~various~~ the anticipated range of sensors, actuators and the algorithms in the HRI Engine.

6.8 Evaluation Criteria

Proposals will be evaluated in terms of consistency in their specifications, feasibility and versatility across a wide range of different robot applications.

6.9 Other information unique to this RFP

None

6.10 RFP Timetable

The timetable for this RFP is given below. Note that the TF or its parent TC may, in certain circumstances, extend deadlines while the RFP is running, or may elect to have more than one Revised Submission step. The latest timetable can always be found at the OMG *Work In Progress* page at http://www.omg.org/schedules/ under the item identified by the name of this RFP. Note that "<month>" and "<approximate month>" is the name of the month spelled out; e.g., January.

Event or Activity	Actual Date
Preparation of RFP by TF	**24. May. 2010**
RFP placed on OMG document server	**24. May. 2010**
Approval of RFP by Architecture Board *Review by TC*	**21. June. 2010**
TC votes to issue RFP	**25. June. 2010**
LOI to submit to RFP due	**13. September. 2010**
Initial Submissions due and placed on OMG document server ("Three week rule")	**8. November. 2010**
Voter registration closes	**29, November, 2010**

Initial Submission presentations	**6. December. 2010**
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Three week rule")	**June 2011**
Revised Submission presentations	**June 2011**
Final evaluation and selection by TF *Recommendation to AB and TC*	
Approval by Architecture Board *Review by TC*	
TC votes to recommend specification	
BoD votes to adopt specification	

Appendix A References and Glossary Specific to this RFP

A.1 References Specific to this RFP

[RTC] Robotic Technology Component specification version 1.0, http://www.omg.org/spec/RTC/1.0/

A.2 Glossary Specific to this RFP

None

Appendix B General Reference and Glossary

B.1 General References

The following documents are referenced in this document:

[ATC] Air Traffic Control Specification, *http://www.omg.org/technology/documents/formal/air_traffic_control.htm*

[BCQ] OMG Board of Directors Business Committee Questionnaire, *http://www.omg.org/cgi-bin/doc?bc/02-02-01*

[CCM] CORBA Core Components Specification, *http://www.omg.org/technology/documents/formal/components.htm*

[CORBA] Common Object Request Broker Architecture (CORBA/IIOP), http://www.omg.org/technology/documents/formal/corba_iiop.htm

[CSIV2] [CORBA] Chapter 26

[CWM] Common Warehouse Metamodel Specification, http://www.omg.org/technology/documents/formal/cwm.htm

[DAIS] Data Acquisition from Industrial Systems, http://www.omg.org/technology/documents/formal/dais.htm

[EDOC] UML Profile for EDOC Specification, http://www.omg.org/techprocess/meetings/schedule/UML_Profile_for_EDOC_FTF.html

[EJB] “Enterprise JavaBeans™”, http://java.sun.com/products/ejb/docs.html

[FORMS] “ISO PAS Compatible Submission Template”. http://www.omg.org/cgi-bin/doc?pas/2003-08-02

[GE] Gene Expression, http://www.omg.org/technology/documents/formal/gene_expression.htm

[GLS] General Ledger Specification , http://www.omg.org/technology/documents/formal/gen_ledger.htm

[Guide] The OMG Hitchhiker's Guide,, http://www.omg.org/cgi-bin/doc?hh

[IDL] ISO/IEC 14750 also see [CORBA] Chapter 3.

[IDLC++] IDL to C++ Language Mapping, http://www.omg.org/technology/documents/formal/c++.htm

[MDAa] OMG Architecture Board, "Model Driven Architecture - A Technical Perspective”, http://www.omg.org/mda/papers.htm

[MDAb] “Developing in OMG's Model Driven Architecture (MDA),” http://www.omg.org/docs/omg/01-12-01.pdf

[MDAc] “MDA Guide” (http://www.omg.org/docs/omg/03-06-01.pdf)

[MDAd] “MDA "The Architecture of Choice for a Changing World™"”, http://www.omg.org/mda

[MOF] Meta Object Facility Specification, http://www.omg.org/technology/documents/formal/mof.htm

[MQS] “MQSeries Primer”, http://www.redbooks.ibm.com/redpapers/pdfs/redp0021.pdf

[NS] Naming Service, *http://www.omg.org/technology/documents/formal/naming_service.htm*

[OMA] “Object Management Architecture™”, *http://www.omg.org/oma/*

[OTS] Transaction Service, *http://www.omg.org/technology/documents/formal/transaction_service.htm*

[P&P] Policies and Procedures of the OMG Technical Process, *http://www.omg.org/cgi-bin/doc?pp*

[PIDS] Personal Identification Service, *http://www.omg.org/technology/documents/formal/person_identification_service.htm*

[RAD] Resource Access Decision Facility, *http://www.omg.org/technology/documents/formal/resource_access_decision.htm*

[RFC2119] IETF Best Practices: Key words for use in RFCs to Indicate Requirement Levels, (*http://www.ietf.org/rfc/rfc2119.txt*).

[RM-ODP] ISO/IEC 10746

[SEC] CORBA Security Service, *http://www.omg.org/technology/documents/formal/security_service.htm*

[TOS] Trading Object Service, *http://www.omg.org/technology/documents/formal/trading_object_service.htm*

[UML] Unified Modeling Language Specification, *http://www.omg.org/technology/documents/formal/uml.htm*

[UMLC] UML Profile for CORBA, *http://www.omg.org/technology/documents/formal/profile_corba.htm*

[XMI] XML Metadata Interchange Specification, *http://www.omg.org/technology/documents/formal/xmi.htm*

[XML/Value] XML Value Type Specification, *http://www.omg.org/technology/documents/formal/xmlvalue.htm*

B.2 General Glossary

Architecture Board (AB) - The OMG plenary that is responsible for ensuring the technical merit and MDA-compliance of RFPs and their submissions.

Board of Directors (BoD) - The OMG body that is responsible for adopting technology.

Common Object Request Broker Architecture (CORBA) - An OMG distributed computing platform specification that is independent of implementation languages.

Common Warehouse Metamodel (CWM) - An OMG specification for data repository integration.

CORBA Component Model (CCM) - An OMG specification for an implementation language independent distributed component model.

Interface Definition Language (IDL) - An OMG and ISO standard language for specifying interfaces and associated data structures.

Letter of Intent (LOI) - A letter submitted to the OMG BoD's Business Committee signed by an officer of an organization signifying its intent to respond to the RFP and confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements.

Mapping - Specification of a mechanism for transforming the elements of a model conforming to a particular metamodel into elements of another model that conforms to another (possibly the same) metamodel.

Metadata - Data that represents models. For example, a UML model; a CORBA object model expressed in IDL; and a relational database schema expressed using CWM.

Metamodel - A model of models.

Meta Object Facility (MOF) - An OMG standard, closely related to UML, that enables metadata management and language definition.

Model - A formal specification of the function, structure and/or behavior of an application or system.

Model Driven Architecture (MDA) - An approach to IT system specification that separates the specification of functionality from the specification of the implementation of that functionality on a specific technology platform.

Normative – Provisions that one must conform to in order to claim compliance with the standard. (as opposed to non-normative or informative which is explanatory material that is included in order to assist in understanding the standard and does not contain any provisions that must be conformed to in order to claim compliance).

Normative Reference – References that contain provisions that one must conform to in order to claim compliance with the standard that contains said normative reference.

Platform - A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented.

Platform Independent Model (PIM) - A model of a subsystem that contains no information specific to the platform, or the technology that is used to realize it.

Platform Specific Model (PSM) - A model of a subsystem that includes information about the specific technology that is used in the realization of it on a specific platform, and hence possibly contains elements that are specific to the platform.

Request for Information (RFI) - A general request to industry, academia, and any other interested parties to submit information about a particular technology area to one of the OMG's Technology Committee subgroups.

Request for Proposal (RFP) - A document requesting OMG members to submit proposals to the OMG's Technology Committee. Such proposals must be received by a certain deadline and are evaluated by the issuing task force.

Task Force (TF) - The OMG Technology Committee subgroup responsible for issuing a RFP and evaluating submission(s).

Technology Committee (TC) - The body responsible for recommending technologies for adoption to the BoD. There are two TCs in OMG – *Platform TC* (PTC), that focuses on IT and modeling infrastructure related standards; and *Domain TC* (DTC), that focus on domain specific standards.

Unified Modeling Language (UML) - An OMG standard language for specifying the structure and behavior of systems. The standard defines an abstract syntax and a graphical concrete syntax.

UML Profile - A standardized set of extensions and constraints that tailors UML to particular use.

XML Metadata Interchange (XMI) - An OMG standard that facilitates interchange of models via XML documents.

Object Management Group

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

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

Request For Proposal

Robotic Interaction Service (RoIS) Framework RFP

OMG Document: robotics/2010-06-23 with change bars
OMG Document: robotics/2010-06-25 without change bars

Letters of Intent due: September 13, 2010
Submissions due: November 8, 2010

Objective of this RFP

This RFP solicits proposals for a Platform Independent Model (PIM) and a Platform Specific Model (PSM) for robotic interaction service (RoIS) that specify

- common interfaces between robotic service applications and components that provide functions for performing human-robot interaction.
- data structures for each interface.

For further details see Chapter 6 of this document.

6.0 Specific Requirements on Proposals

6.1 Problem Statement

Service ~~robot~~ robots that ~~provides some~~ provide services to people in daily life will become more and more popular in the robotics market.~~ Service robot interacts with human to~~ These service robots provide appropriate services~~. For example, there are~~ through human-robot interaction (HRI). Example robotic services ~~such as,~~include;

- Reception service
- Guide service
- Home security service
- Childcare robot service
- Elder person daily watching service
- ~~etc.~~

~~Service~~
A service application is provided as a set of robot behaviors. ~~Rule of~~ The robot ~~behaviors~~’s behavior is defined based on the information collected from ~~human~~humans or ~~environment. The information~~environments. Information is collected ~~by~~ using functions such as~~,~~;

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

Generally~~, several sensors and actuators are equipped with a~~ the robot body, or the environment where the robot provides ~~services, and a~~the service, is equipped with several sensors and actuators. The service application programmer ~~describes~~defines procedures ~~of~~ for the robot~~ behavior and relation between the behavior and~~ ’s actions based (in part) on the information obtained by ~~the~~these sensors~~ in~~. However, different robots may be equipped with a ~~service application program. But the~~ variety of types of sensors and actuators~~ equipped with robots are usually different~~, and moreover the application program interfaces (APIs) of robots from ~~each other and, moreover, their application program interface (API) is~~ different ~~by~~ vendors could vary even if ~~their~~they use the same sensor types~~ are the same. That is~~. As a result, an application program developed for one specific robot ~~will~~ may not run on ~~the other and this is one of the reasons of inefficiency in~~another robot. This lack of application portability is an obstacle to the success of the robot industry.

~~For example in the case of~~

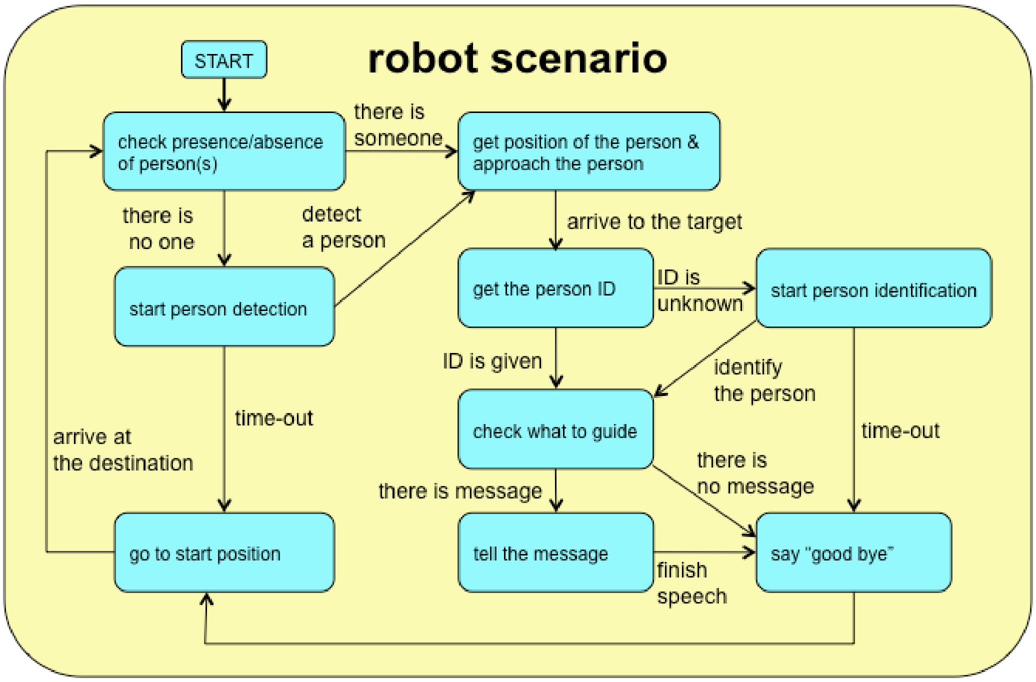


Fig.1: Example of robot scenario for robotic reception service.

In many service applications the robot's behavior is defined using a script called a "robot scenario". This defines how to achieve the service task through the APIs that are specific to the robot, based on the collected data. For example, see the robotic reception service depicted in Fig. 1. In this scenario, when ~~a~~the robot detects someone~~,~~ it approaches the person~~ and~~, tries to recognize who ~~is~~ the person is, and then ~~guides the~~ provides appropriate information ~~appropriate~~ to the person. ~~One~~

Fig.2 shows the messages exchanged between the service application and the robot ~~may detect~~in this scenario. In this case, one robot (Robot 1) detects the human ~~by~~using a camera and ~~move by~~moves using wheels~~. Another~~, while another robot ~~may detect~~(Robot 2) detects the human ~~by~~using an RFID tag and ~~move by~~moves using legs. ~~When there are two types~~Because of ~~robots at~~ the ~~reception desks,~~resulting difference in the APIs the service application programmer ~~must~~would have to write ~~programs~~separate scenarios for each robot ~~independently as depicted in Fig.1~~.

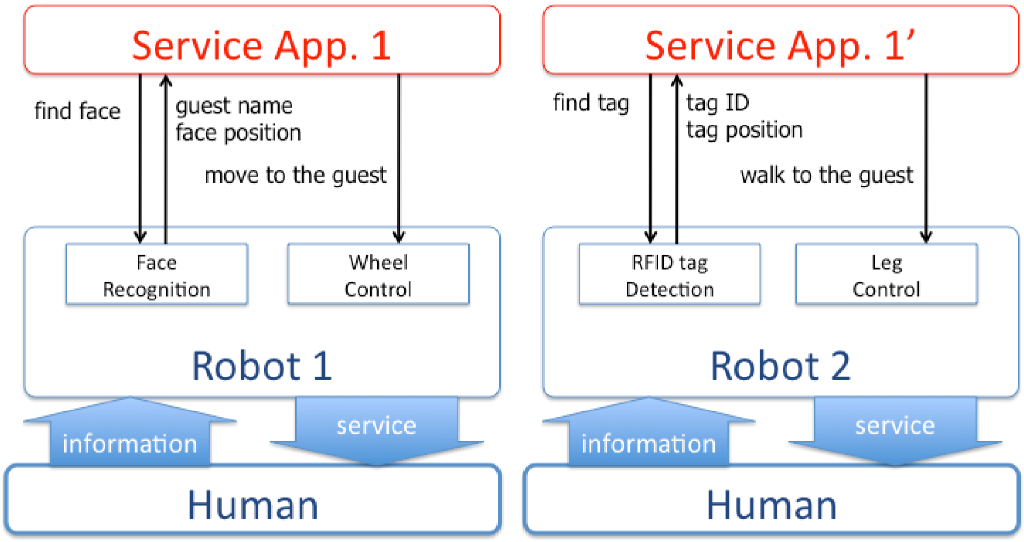


Fig.~~1~~2: Conventional style of service application programming. Service application programmer must write service application programs for each robot independently because functions provided by each robot are different.

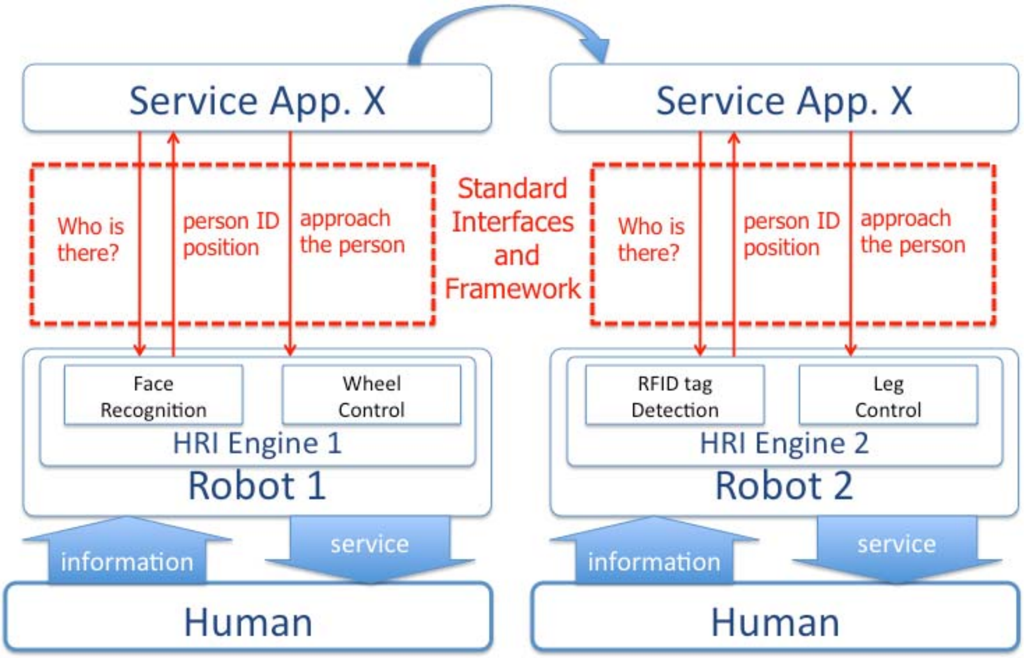
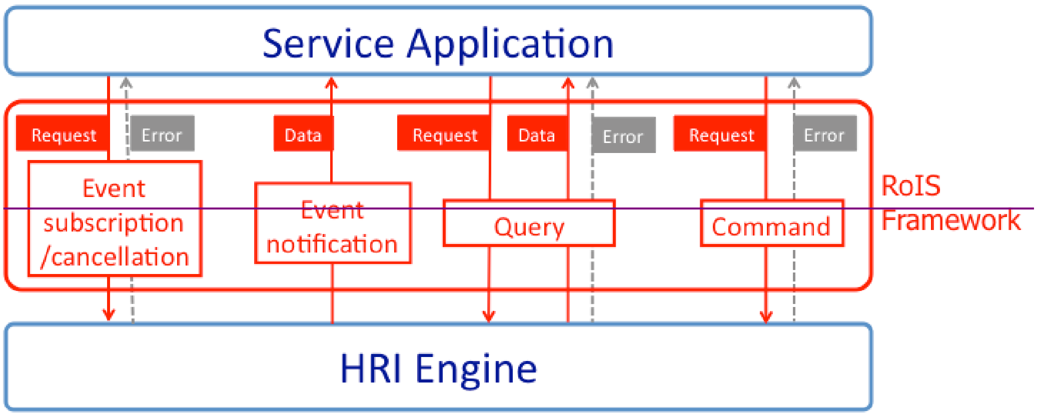


Fig.~~2~~3: RoIS service application programming style. The same service application program works on different robot platforms with little modification.

~~If all the~~By contrast, if the interfaces between the service application and the robot's functional components are ~~encapsulated in a package and the interfaces for obtaining information and controlling robots are~~ standardized, ~~they will~~using a higher level of abstraction based on concepts from the target domain (such as "detect person") rather than implementation dependent concepts (like "detect RFID"), it would be possible to enhance reusability of service application programs. -A service application program for a robot ~~can~~could then work on ~~the~~ other robots regardless of the different robot platforms, as depicted in Fig.~~2. Therefore,~~3. We term this proposed new general framework architecture~~ is needed. We call the package~~, that encapsulates functional components such as HRI ~~Engine and this new framework architecture as~~Engines, the Robotic Interaction Service (RoIS).



~~Fig.3~~When using the RoIS concept it is important to standardize mechanisms for the information and instructions exchanged between the service application and the HRI Engine. From the point of view of a service application, there are generally two types of information to be exchanged, Active Information and Passive Information. Active Information is actively solicited by the service application, for example "check presence/absence of human" and "get position of the person". Passive Information is provided when relevant data is obtained or changed in the HRI Engine, for example "the robot has detected a person" or "the robot has arrived at the target".

In general, interface types of "Query" and "Event notification" are used for exchanging Active Information and Passive Information respectively. To deal with "Event notification", a mechanism for "Event subscription / cancellation" is also required for selecting appropriate event notifications on demand. In addition, the "Command" interface type is naturally needed in order to instruct the HRI Engine to control its functions, such as "approach the person" and "go to start position". The RoIS framework should also include these interface types, i.e., Event subscription/cancellation, Event notification, Query and Command as illustrated in Fig.4.

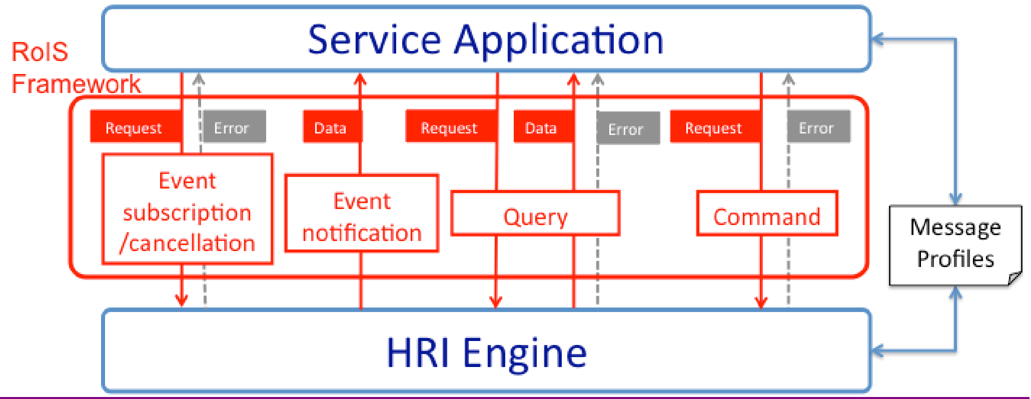


Fig.4: RoIS Framework. In the framework, application communicates with HRI Engine through Event(s) subscription/cancellation, Event(s) notification, Query and Command.

~~There must be at least four types of interface in~~ Within the RoIS framework~~: Event subscription/cancellation, Event notification, Query and Command as illustrated in Fig.3.~~

~~"Event subscription/cancellation" is an interface for service application to subscribe to HRI Engine for specific event type(s) or cancel the subscription(s). Available event types are obtained through the "Query" interface.~~

~~"Event notification" interface is required to notify service application of the occurrence of event(s) in real time. When a functional component in HRI Engine detects some status change, such as "a person appeared in a camera view," "human voice recognized," or "battery is running out," then the component sends a specific event to service application through the "Event notification" interface when the application has subscribed to the event.~~

~~"Query" is used to obtain information actively from HRI Engine. Service application requires this interface type on HRI Engine to obtain the Engine's capabilities (for example, what kind of information is available or what kind of event can be notified by the engine), current status of the Engine, and detailed information of the event notified. Each "Query" has its corresponding result(s).~~

~~"Command" is an interface type to control not only robot but also functional components in HRI Engine. Controlling robot actuators, changing sampling frequency of sensors, and replacing functional components in HRI Engine are the examples of "Commands" to be used.~~

~~It~~ it is desirable to define ~~common interface items of~~ messages for each interface type that are ~~available for~~ common to all ~~HRI Engines as possible. However, there are two types of items: One is the items~~ service domains (and therefore also common to all HRI Engines ~~and the other is specific to each HRI Engine. Therefore, it is important to make a scheme to manage profile of each interface~~

~~type.~~). These common messages should include:

- ~~To specify~~ Event notification
  - Person detected
  - Person identified
  - Command action completed
- Query
  - Person ID
  - Position of person / robot
  - Status of HRI Engine
- Command
  - Start / stop person detection
  - Start / stop person identification

On the other hand, there must also be message types that are specific to each service domain and HRI Engine. For example, in the reception service there may be specific messages for each interface type, such as: message “speech recognised” for the “event notification” interface, message “presence / absence of human” for the query interface, and message “approach the person” for the command interface. HRI Engines should be able to inform service applications about which message profiles they support.

Message profiles are specific to particular application domains, such as the educational domain, navigation domain or healthcare domain. Each message profile is composed of both common message types and domain specific message types. Each profile includes a list of available messages for each interface type, specifying the name of each message, data format for the information exchanged using the message and required argument(s) for the message. An RoIS specification must not only define common messages but also specify how to describe these message profiles, so that the service application can discover the supported messages as necessary.

In specifying the RoIS framework, ~~it is desirable to consider an~~ the abstraction level of the messages for the interface ~~items~~types should be considered. For example, ~~it~~the RoIS framework should ~~be appreciated that a command must~~ not ~~use some~~include any parameters ~~based on a~~specific to a particular robot platform. ~~Also, the information managed by “Event notification” and “Query” must be represented at the suitable level for service applications. The abstraction level~~ The abstractions should be carefully designed to be appropriate ~~to human robot interaction should be considered carefully by focusing on contents of the information and the robot control unique to human robot interaction. In regard to this point, there must be a unique structure for RoIS framework.~~

~~As for the messages to be exchanged through these interfaces, that is, the data specified and exchanged through queries or by events, and commands sent to control~~ for service robot control (as opposed to other robotic ~~systems, these can~~

~~be classified into two categories; one for messages that are common to every HRI engine and the other for messages specific to certain service application domains. As services provided by robots may be applied to a variety of domains related to our daily activity such as route guidance or elderly care, it is much efficient to have dictionaries or profiles of messages specific to each domain rather than using a huge set of messages that covers all domains. On the other hand, there are messages that are commonly used in most of the~~ applications~~ that are typical to human-robot interaction. Therefore, the framework shall include the facility to define and to choose message profiles, and at the same time, shall specify profiles that contain basic common messages.~~, such as industrial robots).

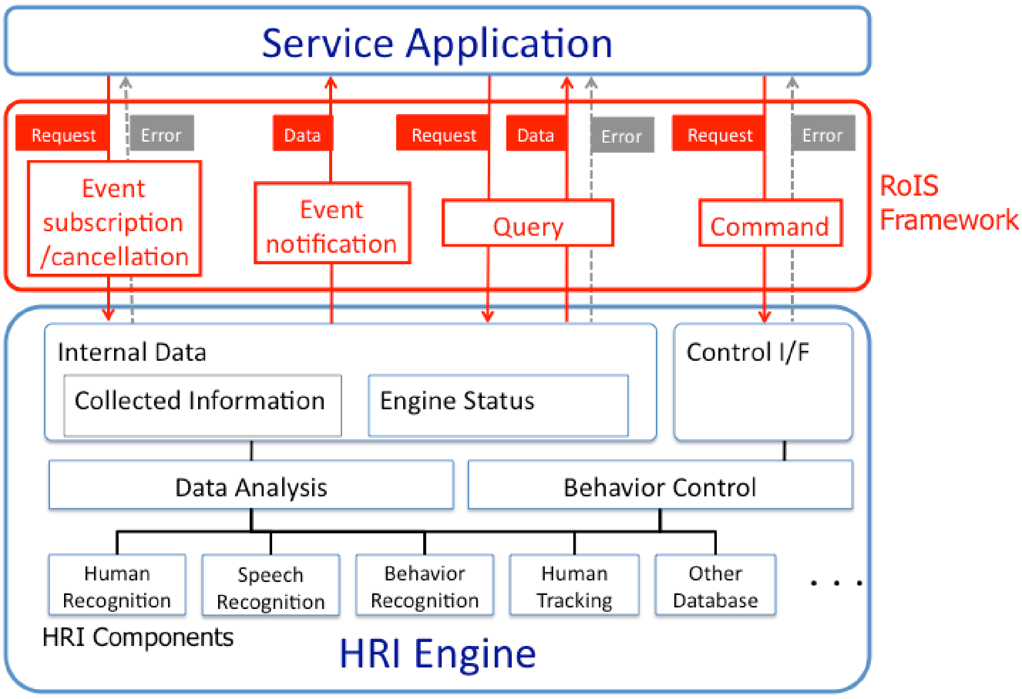


Fig.~~4~~5: Example of RoIS Framework

Fig.~~4~~5 illustrates ~~a~~the basic structure of the RoIS Framework. The HRI Engine integrates several functional components (the HRI Components) and provides their functions ~~with~~ to a service application through standardized ~~interface~~interfaces. It collects and manages information ~~of human around~~about nearby humans and its environment ~~by~~via sensors, and provides collected data ~~with~~to the application on demand. As some of the HRI Components in the HRI Engine provide robot control functions, the application can control the robot through the engine's ~~interface~~interfaces.

Separation and encapsulation of HRI Components ~~into~~within the HRI Engine, and providing standardized interface to the components, will enhance ~~not only~~ the reusability of ~~components but~~HRI Components, and also improve the efficiency of both service application and component development.

~~Therefore, the~~The scope of the specification solicited includes the definition of interfaces ~~in~~ between service applications and the HRI Engine~~, that is,~~ (i.e. Event subscription/cancellation, Event notification, Query and Command~~,~~), and the structure of data transmitted through each interface. ~~-~~Error notification sent from the HRI Engine to ~~application~~ applications may also be ~~included.~~specified. However, internal data ~~structure~~structures that ~~depends~~depend on each HRI Engine implementation, or data structures defined in existing specifications ~~such as~~ (e.g the BioAPI specification for user identification), should not be included in this specification.

Considering that HRI Engines depend on their robot platforms, ~~the~~ HRI Engine developers should be able to define interfaces between functional components ~~inside their~~within each engine ~~in their independent way.~~ independently. The RoIS framework standard should not ~~concern about~~address the internals of the HRI Engine~~ inside. ~~. For example, ~~one~~ a developer ~~can~~could use some other standardized framework, such as ~~RTM, for~~OMG Robotic Technology Components, inside an HRI Engine, ~~and the other~~while another developer ~~can~~could use ~~their original method. Also~~a proprietary framework. In addition, HRI ~~Engine can~~Engines may access ~~to the~~ other applications and databases, such as location ~~data~~ and map data for path generation, ~~by~~ using other appropriate frameworks. Similarly, the RoIS framework ~~as needed. The same can be said for service application program inside.~~ must not constrain the internal implementation details of service applications. For example, ~~in the case that the~~if a service application ~~has to be corporate~~must communicate with ~~the other~~another application, such as a network robot service, the service application ~~can~~may use ~~other~~another framework to access ~~to the other~~that application.

~~In summary, following items are required for RoIS framework.~~

- **~~Event(s) subscription/cancellation~~**
  - ~~Subscribe to specific event(s) and cancel subscription for specific event(s)~~
  - ~~Sent from application to HRI Engine~~
- **~~Event(s) notification~~**
  - ~~Notify the occurrence of event to subscriber(s)~~
  - ~~Sent from HRI Engine to application~~
- **~~Query~~**
  - ~~Retrieve detailed information of events notified by HRI Engine~~
  - ~~Sent from application to HRI Engine (i.e. *requests*) and from HRI Engine to application (i.e. *results*)~~
- **~~Command~~**
  - ~~Give commands to a robot, components of HRI Engine and/or the engine itself~~
- ~~Sent from application to HRI Engine~~

6.2 Scope of Proposals Sought

This RFP seeks proposals that specify a RoIS framework, on top of which various ~~robotic~~ service robot applications are developed. ~~It~~

The scope of proposals is ~~the target for service robots interacting with human.~~summarized in the following items:

~~It is necessary to consider the followings in the specification of RoIS framework.~~

~~(1)Overall architecture for RoIS framework shall be defined (diagram or description for overview).~~

~~(3)~~● ~~The RoIS framework specification shall provide interface~~Interfaces between ~~robotic~~ service ~~applications~~application and HRI Engine ~~that is a set of robotic components.~~

~~(3)The RoIS framework specification must be general enough to incorporate robotic components for various sensors, actuators and algorithms in HRI Engine.~~

- ~~The RoIS framework specification shall satisfy interoperability and reusability. A~~Interface to obtain information from HRI Engine according to the timing of the service application's needs (Query)

~~(4)~~Interface to receive information from HRI Engine ~~should be able to be replaced with the other HRI engine with little efforts.~~

~~(5)The RoIS framework specification shall provide a minimum set of functionalities to satisfy the followings:~~

~~●Interface types between robotic service applications and HRI Engine~~

- triggered by real-time events (Event notification / subscription ~~and~~/ cancellation)

  - ~~Event notifications~~
  - ~~Query~~
  - ~~Command~~

~~●Data structure for each interface type~~

~~(6)The RoIS framework specification shall provide a scheme to manage profile of each interface type~~

- ~~The RoIS Framework specification shall specify profiles of message and those~~Interface for instructions to control HRI Engine functions (Command)

~~(4)~~● Definition of common messages for ~~each interface type~~all HRI Engines

- Definition of a set of messages that are common to all service robot domains

6.3 Relationship to other OMG Specifications and activities

6.3.1 Relationship to OMG specifications

Submitters shall examine the following OMG specifications for possible benefit:

- Super Distributed Objects (SDO) Specification version 1.1 [formal/2008-10-~~11~~01]
- Unified Modeling Language: Infrastructure version 2.~~1.2~~3 [formal/~~2007-11-04~~2010-05-03]
- Unified Modeling Language: Superstructure version 2.~~1.2~~3 [formal/~~2007-11-02~~2010-05-05]
- Lightweight CORBA Component Model 4.0 [formal/~~2006~~06-04-01]
- Robotic Technology Component specification version 1.0 [formal/08-04-04]
- Robotic Localization Service version 1.0 [formal/2010-02-03]

6.3.2 Relationship to other OMG Documents and work in progress

None

6.4 Related non-OMG Activities, Documents and Standards

Proposals may include existing systems, documents, user recognition service interface, and standards that are relevant to the problems discussed in this RFP. They can be used as background information for the proposal.

~~Example~~Examples:

- IEEE Robotics and Automation Society, Technical Committee on Network Robot
- IEEE Robotics and Automation Society, Technical Committee on Programming Environment in Robotics and Automation
- ISO/ SC 37 Projects ~~-~~relate to ISO/IEC 19784-1(BioAPI Ver 2.0)
- ISO/TC184/SC2 Robots and robotic devices

6.5 Mandatory Requirements

~~(1)~~Proposals shall provide ~~overall architecture for RoIS framework (~~a diagram or description ~~for~~giving an overview~~)~~

~~(2)Proposals shall provide a Platform Independent Model (PIM) and at least one Platform Specific Model (PSM)~~ of the architecture, including the RoIS framework~~.~~

~~(3)Proposals shall specify a general mechanism for RoIS framework~~

~~●~~(1) ~~Interfaces between applications~~, robotic service application and HRI Engine ~~and their types:~~(composed of robotic components).

- ~~Event subscription and cancellation~~
- ~~Event notifications~~
- ~~Query~~
- ~~Command~~

(2) ~~Data~~Proposals shall provide a specification of the following interfaces between robotic service applications and the HRI Engine:

- Interface to obtain information actively from HRI Engine

- Interface to receive notification of real-time events from the HRI Engine

- Control interface to send commands to HRI Engine functions

~~●~~(3) Proposals shall specify data structure for each interface~~ type~~.

~~(5)~~(4)~~The RoIS framework specification~~ Proposals shall provide a ~~scheme to manage profile of each interface type~~specification of the following common messages for all HRI Engines.

- ▶ Event notification
  - Person detected
  - Person identified
  - Command action completed
- ▶ Query
  - Person ID
  - Position of person / robot
  - Status of HRI Engine
- ▶ Command
  - Start / stop person detection
  - Start / stop person identification

(5) The ~~RoIS Framework~~ specification shall meet the following criteria:

- Be general enough to incorporate robotic components for various sensors, actuators and algorithms in HRI Engine.

- Satisfy interoperability and reusability, to allow an HRI Engine to be replaced with another without difficulty.

(6) Proposals shall specify ~~profiles of message and those of common messages for each interface type~~existing technologies to achieve functions required for RoIS framework.

(7) Proposal shall specify functions that cannot be achieved by existing technologies.

(8) Proposals shall provide a Platform Independent Model (PIM).

6.6 Optional Requirements

(1) Proposals may specify schemes for the functionalities listed below.

- Error handling for each interface type
- Returning command results and status

(2) Proposals may provide a schema to describe message profiles.

(3) Proposals may provide a Platform Specific Model (PSM) as C++, CORBA-specific model, XML or RTC for RoIS framework.

6.7 Issues to be discussed

These issues will be considered during submission evaluation. They should not be part of the proposed normative specification. (Place them in Part I of the submission.)

- Proposals shall demonstrate ~~its~~their feasibility by ~~using~~giving a specific example application based on the proposed model.
- Proposals shall demonstrate ~~its~~their applicability to existing robotic interaction service technologies.
- Proposals shall discuss simplicity of implementation.
- Proposals shall discuss the possibility ~~to apply~~of applying the proposed model to other fields of interest such as intelligent service robot applications.
- Proposals shall ~~specify~~show that they are independent of on-the-wire protocol communication technology ~~independent~~.
- Proposals shall discuss their ~~relation~~ relationship to and dependency ~~to~~ on existing ~~communication protocols or middlewares~~middleware standards, such as CORBA [CORBA], DDS [DDS] or RTC [RTC].
- Proposals shall discuss their generality with respect to the anticipated range of sensors, actuators and the algorithms in the HRI Engine.

6.8 Evaluation Criteria

Proposals will be evaluated in terms of consistency in their specifications, feasibility and versatility across a wide range of different robot applications.

6.9 Other information unique to this RFP

None

6.10 RFP Timetable

The timetable for this RFP is given below. Note that the TF or its parent TC may, in certain circumstances, extend deadlines while the RFP is running, or may elect to have more than one Revised Submission step. The latest timetable can always be found at the OMG *Work In Progress* page at http://www.omg.org/schedules under the item identified by the name of this RFP. Note that "<month>" and "<approximate month>" is the name of the month spelled out; e.g., January.

Event or Activity	Actual Date
Preparation of RFP by TF	*24. May* 2010
RFP placed on OMG document server	24. May 2010
Approval of RFP by Architecture Board *Review by TC*	21. June 2010
TC votes to issue RFP	25. June 2010
LOI to submit to RFP due	13. September 2010
Initial Submissions due and placed on OMG document server ("Four week rule")	8. November 2010
Voter registration closes	29. November 2010
Initial Submission presentations	~~6.~~ December 2010
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Four week rule")	June 2011
Revised Submission presentations	June 2011
Final evaluation and selection by TF *Recommendation to AB and TC*	
Approval by Architecture Board *Review by TC*	
TC votes to recommend specification	
BoD votes to adopt specification	

Appendix A References and Glossary Specific to this RFP

A.1 References Specific to this RFP

[RTC] Robotic Technology Component specification version 1.0, http://www.omg.org/spec/RTC/1.0/

A.2 Glossary Specific to this RFP

None

Appendix B General Reference and Glossary

B.1 General References

The following documents are referenced in this document:

[ATC] Air Traffic Control Specification, *http://www.omg.org/technology/documents/formal/air_traffic_control.htm*

[BCQ] OMG Board of Directors Business Committee Questionnaire, http://doc.omg.org/bc/07-08-06

[CCM] CORBA Core Components Specification, *http://www.omg.org/technology/documents/formal/components.htm*

[CORBA] Common Object Request Broker Architecture (CORBA/IIOP), *http://www.omg.org/technology/documents/formal/corba_iiop.htm*

[CSIV2] [CORBA] Chapter 26

[CWM] Common Warehouse Metamodel Specification, *http://www.omg.org/technology/documents/formal/cwm.htm*

[DAIS] Data Acquisition from Industrial Systems, *http://www.omg.org/technology/documents/formal/dais.htm*

[EDOC] UML Profile for EDOC Specification, *http://www.omg.org/techprocess/meetings/schedule/UML_Profile_for_EDOC_FTF.html*

[EJB] “Enterprise JavaBeans™”, *http://java.sun.com/products/ejb/docs.html*

[FORMS] “ISO PAS Compatible Submission Template”. *http://www.omg.org/cgi-bin/doc?pas/2003-08-02*

[GE] Gene Expression, *http://www.omg.org/technology/documents/formal/gene_expression.htm*

[GLS] General Ledger Specification , *http://www.omg.org/technology/documents/formal/gen_ledger.htm*

[Guide] The OMG Hitchhiker's Guide,, *http://www.omg.org/cgi-bin/doc?hh*

[IDL] ISO/IEC 14750 also see [CORBA] Chapter 3.

[IDLC++] IDL to C++ Language Mapping, *http://www.omg.org/technology/documents/formal/c++.htm*

[Inventory] Inventory of Files for a Submission/Revision/Finalization, *http://doc.omg.org/smsc/2007-09-05*

[MDAa] OMG Architecture Board, "Model Driven Architecture - A Technical Perspective”, *http://www.omg.org/mda/papers.htm*

[MDAb] “Developing in OMG's Model Driven Architecture (MDA),” *http://www.omg.org/docs/omg/01-12-01.pdf*

[MDAc] “MDA Guide” (*http://www.omg.org/docs/omg/03-06-01.pdf*)

[MDAd] “MDA "The Architecture of Choice for a Changing World™"”, *http://www.omg.org/mda*

[MOF] Meta Object Facility Specification, *http://www.omg.org/technology/documents/formal/mof.htm*

[MQS] “MQSeries Primer”, *http://www.redbooks.ibm.com/redpapers/pdfs/redp0021.pdf*

[NS] Naming Service, *http://www.omg.org/technology/documents/formal/naming_service.htm*

[OMA] “Object Management Architecture™”, *http://www.omg.org/oma/*

[OTS] Transaction Service, *http://www.omg.org/technology/documents/formal/transaction_service.htm*

[P&P] Policies and Procedures of the OMG Technical Process, *http://www.omg.org/cgi-bin/doc?pp*

[PIDS] Personal Identification Service, *http://www.omg.org/technology/documents/formal/person_identification_service.htm*

[RAD] Resource Access Decision Facility, *http://www.omg.org/technology/documents/formal/resource_access_decision.htm*

[RFC2119] IETF Best Practices: Key words for use in RFCs to Indicate Requirement Levels, (*http://www.ietf.org/rfc/rfc2119.txt*).

[RM-ODP] ISO/IEC 10746

[SEC] CORBA Security Service, *http://www.omg.org/technology/documents/formal/security_service.htm*

[TOS] Trading Object Service, *http://www.omg.org/technology/documents/formal/trading_object_service.htm*

[UML] Unified Modeling Language Specification, *http://www.omg.org/technology/documents/formal/uml.htm*

[UMLC] UML Profile for CORBA, *http://www.omg.org/technology/documents/formal/profile_corba.htm*

[XMI] XML Metadata Interchange Specification, *http://www.omg.org/technology/documents/formal/xmi.htm*

[XML/Value] XML Value Type Specification, *http://www.omg.org/technology/documents/formal/xmlvalue.htm*

B.2 General Glossary

Architecture Board (AB) - The OMG plenary that is responsible for ensuring the technical merit and MDA-compliance of RFPs and their submissions.

Board of Directors (BoD) - The OMG body that is responsible for adopting technology.

Common Object Request Broker Architecture (CORBA) - An OMG distributed computing platform specification that is independent of implementation languages.

Common Warehouse Metamodel (CWM) - An OMG specification for data repository integration.

CORBA Component Model (CCM) - An OMG specification for an implementation language independent distributed component model.

Interface Definition Language (IDL) - An OMG and ISO standard language for specifying interfaces and associated data structures.

Letter of Intent (LOI) - A letter submitted to the OMG BoD's Business Committee signed by an officer of an organization signifying its intent to respond to the RFP and confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements.

Mapping - Specification of a mechanism for transforming the elements of a model conforming to a particular metamodel into elements of another model that conforms to another (possibly the same) metamodel.

Metadata - Data that represents models. For example, a UML model; a CORBA object model expressed in IDL; and a relational database schema expressed using CWM.

Metamodel - A model of models.

Meta Object Facility (MOF) - An OMG standard, closely related to UML, that enables metadata management and language definition.

Model - A formal specification of the function, structure and/or behavior of an application or system.

Model Driven Architecture (MDA) - An approach to IT system specification that separates the specification of functionality from the specification of the implementation of that functionality on a specific technology platform.

Normative – Provisions that one must conform to in order to claim compliance with the standard. (as opposed to non-normative or informative which is explanatory material that is included in order to assist in understanding the standard and does not contain any provisions that must be conformed to in order to claim compliance).

Normative Reference – References that contain provisions that one must conform to in order to claim compliance with the standard that contains said normative reference.

Platform - A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented.

Platform Independent Model (PIM) - A model of a subsystem that contains no information specific to the platform, or the technology that is used to realize it.

Platform Specific Model (PSM) - A model of a subsystem that includes information about the specific technology that is used in the realization of it on a specific platform, and hence possibly contains elements that are specific to the platform.

Request for Information (RFI) - A general request to industry, academia, and any other interested parties to submit information about a particular technology area to one of the OMG's Technology Committee subgroups.

Request for Proposal (RFP) - A document requesting OMG members to submit proposals to an OMG Technology Committee. Such proposals must be received by a certain deadline and are evaluated by the issuing Task Force.

Task Force (TF) - The OMG Technology Committee subgroup responsible for issuing a RFP and evaluating submission(s).

Technology Committee (TC) - The body responsible for recommending technologies for adoption to the BoD. There are two TCs in OMG – the *Platform TC* (PTC) focuses on IT and modeling infrastructure related standards; while the *Domain TC* (DTC) focuses on domain specific standards.

Unified Modeling Language (UML) - An OMG standard language for specifying the structure and behavior of systems. The standard defines an abstract syntax and a graphical concrete syntax.

UML Profile - A standardized set of extensions and constraints that tailors UML to particular use.

XML Metadata Interchange (XMI) - An OMG standard that facilitates interchange of models via XML documents.

< Note to RFP Editors: Append additional appendices if needed here and update the list and brief description of appendices in Chapter 1. >

OMG Document:　robotics/2010-06-24

Errata to revised submission of the Robotic Interaction Service (RoIS) Framework RFP

Initial Submission: robotics/2010-05-01

Revised Submission: robotics/2010-06-23(with change bars), 06-25(without change bars)

<Mr. Vincent’s comments>

1. The figure 3 as well as the 5 following paragraphs are too broad: almost all API in the world can be split in these four types: Event subscription, Event notification, Query and Command.

→ We have added an explanation and a figure (new figure 1) of robot scenario concept to clarify the necessity for the four interface types. Two paragraphs before figure 4 (previous figure 3) are also corrected for this purpose. The most important point of this RFP is that we should standardize the content and handling of messages for these interfaces. That is, it is desirable to define common interface messages specific to human-robot interaction. To express it in a better way, we added some examples of such common messages after figure 4.

2. The concept of "Profiles of interface" as well as how to "manage" them are unclear. These concepts must be made clearer in the RFP before continuing the process.

→ We changed the words "profiles of interface" to “message profiles” and explained it in detail after figure 4.

3. Section 6.2 - 1st paragraph: "It is the target for ... human." -> "This RFP is targeted on human-robot interaction”

→ We corrected the sentence.

4. Section 6.2 - 1st bullet (1) AND Section 6.5 Requirement (1): What do you

mean by “overall architecture”? The “architecture” term is too broad and has too many uses to let it alone without clarification - Please specify this requirement. Moreover, are yor sure you want to standardize such an "architecture" for RoIS? Isn’t it implementation-dependant?

→ We have corrected the expression "overall architecture" to clarify the requirement.

5. Section 6.2 - 5th bullet (5): As already said, this is too broad for an RFP and can simply be removed from it.

→ We have unified (2) and (5) into (2) and corrected according to the above corrections.

6. Section 6.5 - Requirement (2): "at least one PSM": My guess is that you don't want whatever PSM so please specify what you want: RTC and XML?

→ We have specified the requirement for PSM and changed to the optional requirement.

7. Section 6.5 - Requirements (3) to (5): As already said the concepts of profile and profile management are not clear enough to keep this like that.

→ We have dropped the requirement for “schema to manage profile” and separated these requirements to the requirements for “schema to describe message profile” and “common messages”. The requirement for “schema to describe message profile” is changed to the optional requirement.

8. Section 6.7: Proposals should also be requested to discuss about their generality wrt various sensors, actuators and algorithms (cf section 6.2 - bullet (3)).

→ We have added the last bullet based on the comment.

9. I'm puzzled by the lack of any specific interfaces you want to find in the future standard. For instance, I was waiting for a list of useful interfaces

such as those found in figure 2: "Is there somebody?", "Get Person ID", "Approach the person"… If you really have no list for this, I believe that you need first either to work on it inside your TF or to draft an RFI to request this list before redrafting this RFP.

→ We made a list of common messages. It is described after figure 4 and in the mandatory requirement (in Section 6.5 (5)).

<Mr. Watson's comments>

10. Does the RFP seek to standardise a complete set of domain abstractions for the domestic robotic domain (e.g. person, speech, pick-up, carry, put-down, etc), or just to standardise a framework within which these concepts can be expressed?

→ This RFP seeks the latter. In our latest draft, it is mentioned in the paragraphs after figure 3. This concept includes several robotic domains.

11. The RFP talks extensively about events. Does it direct submitters to re-use any of the existing event frameworks that OMG as already standardised?

→ Yes. We expect to use existing communication protocols and middlewares to achieve this framework as possible. Therefore, we have added the requirements in Section 6.5 (3) (4) to specify the classification of the required functions into the existing standards and the other.

12. The Mandatory requirements section talks about events, but barely mentions the important issue of domain abstractions and mechanisms for defining new abstractions.

→ We have explained about the issue as a part of "message profiles" and mentioned the requirement for message profiles in Section 6.6 (2).

13. Do you want the submitters to target any particular platform with their PSM? If so, which one?

→ We have specified the requirement for PSM in Section 6.6 (3).

<After AB Plenary on Monday (June 21)>

According to the comments from the AB Plenary

・ Section 6.1 Problem statement

–Improved the style of English with the help of Andrew.

・Section 6.2 Scope of Proposals Sought

–Changed into the summary of 6.1. (The last bullets in the previous 6.1.)

・Section 6.3 Relationship to Existing OMG Specifications

–Updated the version of each specification.

・Section 6.5 Mandatory Requirements

–Unified the items in the previous 6.1 and 6.5

・Section 6.6 Issues to be discussed

–Improved the style of English with the help of Andrew.

・Updated the RFP template to the latest version.

Object Management Group

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

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

Request For Proposal

Robotic Interaction Service (RoIS) Framework RFP

OMG Document: robotics/2010-06-23 with change bars
OMG Document: robotics/2010-06-25 without change bars

Letters of Intent due: September 13, 2010
Submissions due: November 8, 2010

Objective of this RFP

This RFP solicits proposals for a Platform Independent Model (PIM) and a Platform Specific Model (PSM) for robotic interaction service (RoIS) that specify

- common interfaces between robotic service applications and components that provide functions for performing human-robot interaction.
- data structures for each interface.

For further details see Chapter 6 of this document.

6.0 Specific Requirements on Proposals

6.1 Problem Statement

Service robots that provide services to people in daily life will become more and more popular in the robotics market. These service robots provide appropriate services through human-robot interaction (HRI). Example robotic services include;

- Reception service
- Guide service
- Home security service
- Childcare robot service
- Elder person daily watching service

A service application is provided as a set of robot behaviors. The robot's behavior is defined based on the information collected from humans or environments. Information is collected using functions such as;

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

Generally the robot body, or the environment where the robot provides the service, is equipped with several sensors and actuators. The service application programmer defines procedures for the robot's actions based (in part) on the information obtained by these sensors. However, different robots may be equipped with a variety of types of sensors and actuators, and moreover the application program interfaces (APIs) of robots from different vendors could vary even if they use the same sensor types. As a result, an application program developed for one specific robot may not run on another robot. This lack of application portability is an obstacle to the success of the robot industry.

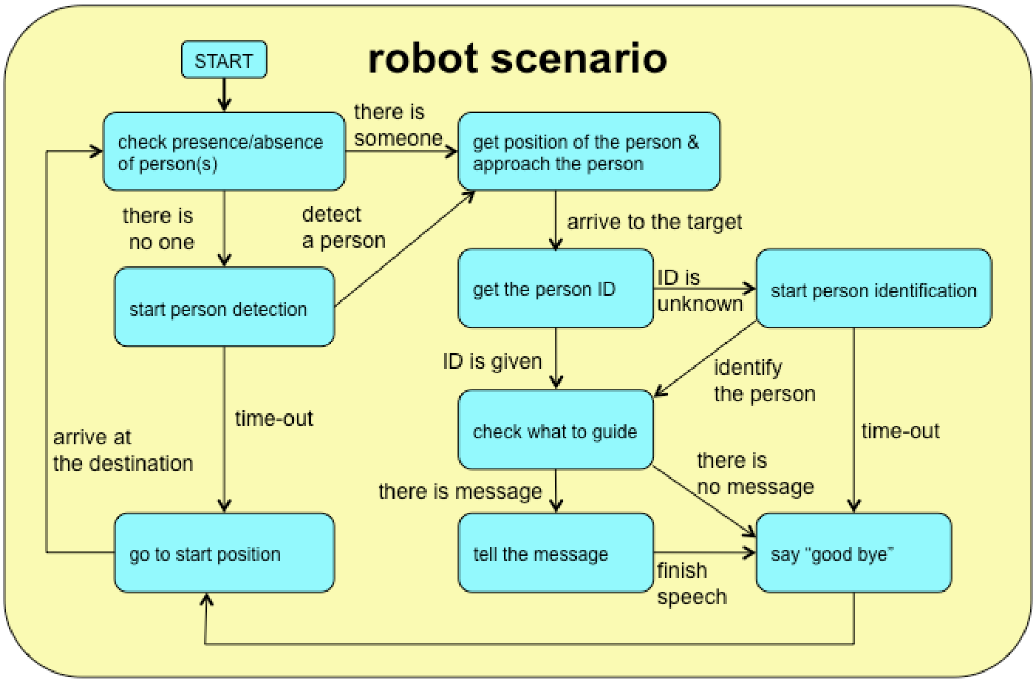


Fig.1: Example of robot scenario for robotic reception service.

In many service applications the robot's behavior is defined using a script called a "robot scenario". This defines how to achieve the service task through the APIs that are specific to the robot, based on the collected data. For example, see the robotic reception service depicted in Fig. 1. In this scenario, when the robot detects someone it approaches the person, tries to recognize who the person is, and then provides appropriate information to the person.

Fig.2 shows the messages exchanged between the service application and the robot in this scenario. In this case, one robot (Robot 1) detects the human using a camera and moves using wheels, while another robot (Robot 2) detects the human using an RFID tag and moves using legs. Because of the resulting difference in the APIs the service application programmer would have to write separate scenarios for each robot.

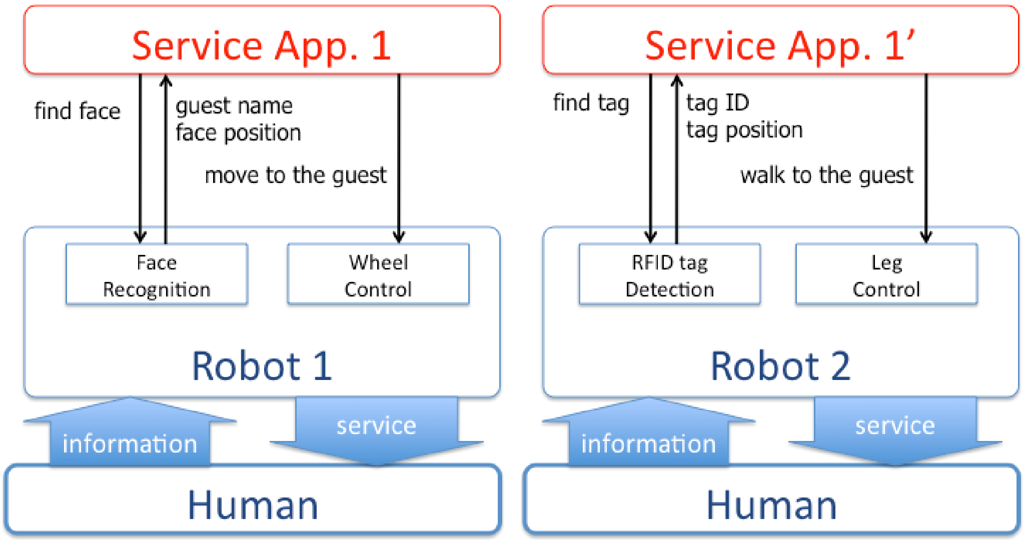


Fig.2: Conventional style of service application programming. Service application programmer must write service application programs for each robot independently because functions provided by each robot are different.

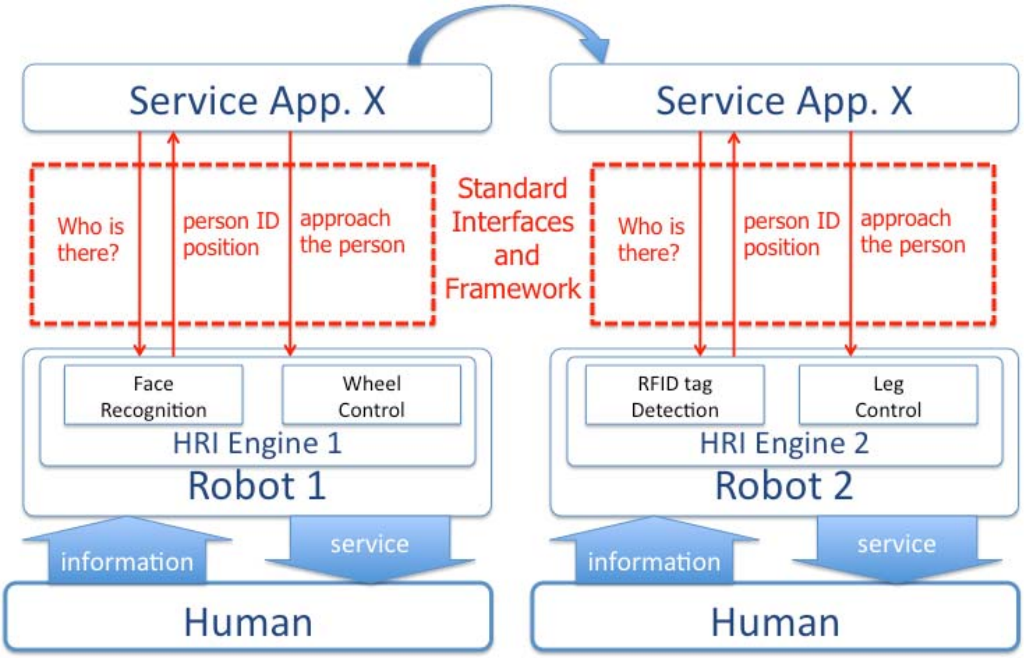


Fig.3: RoIS service application programming style. The same service application program works on different robot platforms with little modification.

By contrast, if the interfaces between the service application and the robot's functional components are standardized, using a higher level of abstraction based on concepts from the target domain (such as "detect person") rather than implementation dependent concepts (like "detect RFID"), it would be possible to enhance reusability of service application programs. A service application program for a robot could then work on other robots regardless of the different robot platforms, as depicted in Fig.3. We term this proposed new general framework architecture, that encapsulates functional components such as HRI Engines, the Robotic Interaction Service (RoIS).

When using the RoIS concept it is important to standardize mechanisms for the information and instructions exchanged between the service application and the HRI Engine. From the point of view of a service application, there are generally two types of information to be exchanged, Active Information and Passive Information. Active Information is actively solicited by the service application, for example "check presence/absence of human" and "get position of the person". Passive Information is provided when relevant data is obtained or changed in the HRI Engine, for example "the robot has detected a person" or "the robot has arrived at the target".

In general, interface types of "Query" and "Event notification" are used for exchanging Active Information and Passive Information respectively. To deal with "Event notification", a mechanism for "Event subscription / cancellation" is also required for selecting appropriate event notifications on demand. In addition, the "Command" interface type is naturally needed in order to instruct the HRI Engine to control its functions, such as "approach the person" and "go to start position". The RoIS framework should also include these interface types, i.e., Event subscription/cancellation, Event notification, Query and Command as illustrated in Fig.4.

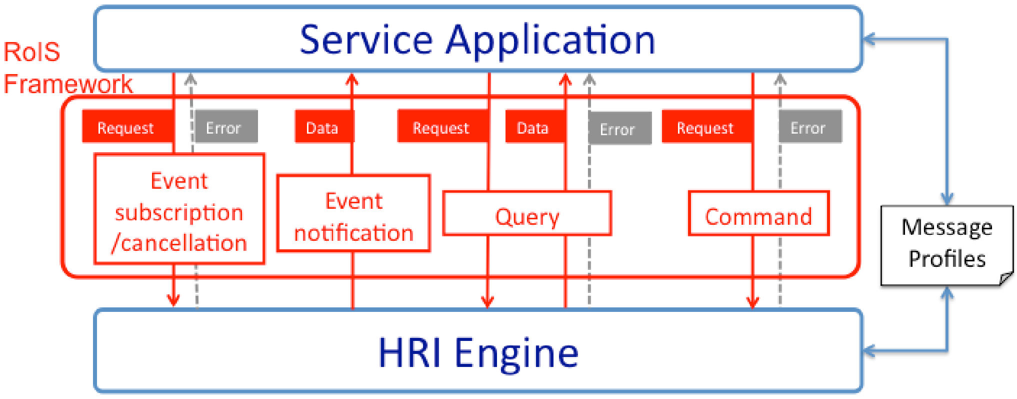


Fig.4: RoIS Framework. In the framework, application communicates with HRI Engine through Event(s) subscription/cancellation, Event(s) notification, Query and Command.

Within the RoIS framework it is desirable to define messages for each interface type that are common to all service domains (and therefore also common to all HRI Engines). These common messages should include:

- Event notification

    - Person detected
    - Person identified
    - Command action completed
  - Query
    - Person ID
    - Position of person / robot
    - Status of HRI Engine
  - Command
    - Start / stop person detection
    - Start / stop person identification

On the other hand, there must also be message types that are specific to each service domain and HRI Engine. For example, in the reception service there may be specific messages for each interface type, such as: message “speech recognised” for the “event notification” interface, message “presence / absence of human” for the query interface, and message “approach the person” for the command interface. HRI Engines should be able to inform service applications about which message profiles they support.

Message profiles are specific to particular application domains, such as the educational domain, navigation domain or healthcare domain. Each message profile is composed of both common message types and domain specific message types. Each profile includes a list of available messages for each interface type, specifying the name of each message, data format for the information exchanged using the message and required argument(s) for the message. An RoIS specification must not only define common messages but also specify how to describe these message profiles, so that the service application can discover the supported messages as necessary.

In specifying the RoIS framework, the abstraction level of the messages for the interface types should be considered. For example, the RoIS framework should not include any parameters specific to a particular robot platform. The abstractions should be carefully designed to be appropriate for service robot control (as opposed to other robotic applications, such as industrial robots).

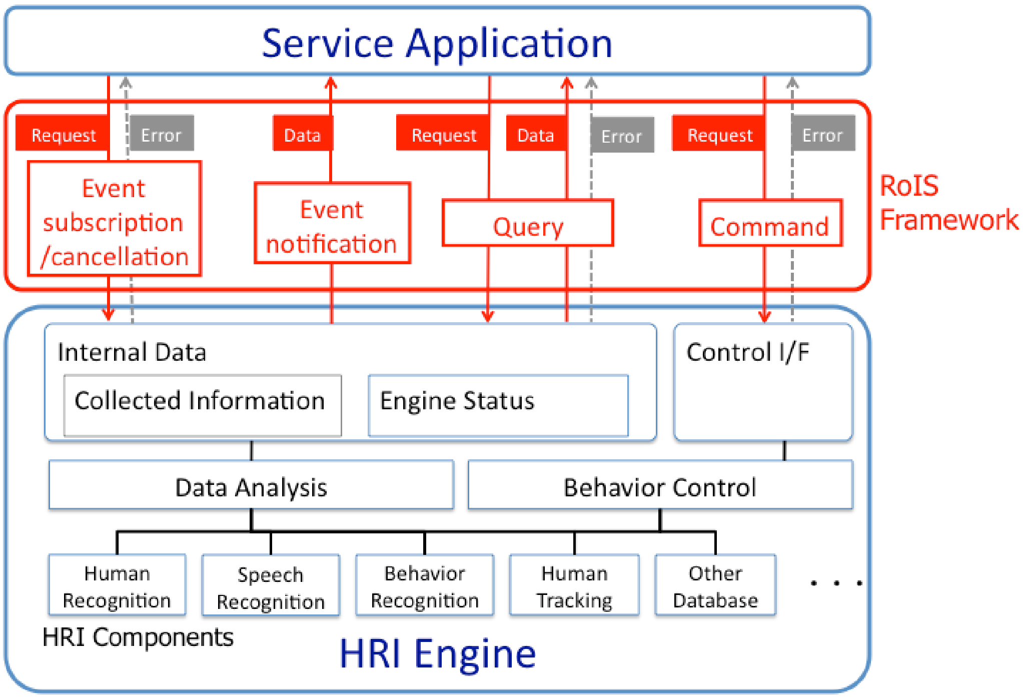


Fig.5: Example of RoIS Framework

Fig.5 illustrates the basic structure of the RoIS Framework. The HRI Engine integrates several functional components (the HRI Components) and provides their functions to a service application through standardized interfaces. It collects and manages information about nearby humans and its environment via sensors, and provides collected data to the application on demand. As some of the HRI Components in the HRI Engine provide robot control functions, the application can control the robot through the engine’s interfaces.

Separation and encapsulation of HRI Components within the HRI Engine, and providing standardized interface to the components, will enhance the reusability of HRI Components, and also improve the efficiency of both service application and component development.

The scope of the specification solicited includes the definition of interfaces between service applications and the HRI Engine (i.e. Event subscription/cancellation, Event notification, Query and Command), and the structure of data transmitted through each interface. Error notification sent from the HRI Engine to applications may also be specified. However, internal data structures that depend on each HRI Engine implementation, or data structures defined in existing specifications (e.g the BioAPI specification for user identification), should not be included in this specification.

Considering that HRI Engines depend on their robot platforms, HRI Engine developers should be able to define interfaces between functional components within each engine independently. The RoIS framework standard should not address the internals of the HRI Engine. For example, a developer could use some other standardized framework, such as OMG Robotic Technology Components, inside an HRI Engine, while another developer could use a proprietary framework. In addition, HRI Engines may access other applications and databases, such as location and map data for path generation, using other appropriate frameworks. Similarly, the RoIS framework must not constrain the internal implementation details of service applications. For example, if a service application must communicate with another application, such as a network robot service, the service application may use another framework to access that application.

6.2 Scope of Proposals Sought

This RFP seeks proposals that specify a RoIS framework, on top of which various service robot applications are developed.

The scope of proposals is summarized in the following items:

- Interfaces between service application and HRI Engine
  - Interface to obtain information from HRI Engine according to the timing of the service application’s needs (Query)
  - Interface to receive information from HRI Engine triggered by real-time events (Event notification / subscription / cancellation)
  - Interface for instructions to control HRI Engine functions (Command)

- Definition of common messages for all HRI Engines
  - Definition of a set of messages that are common to all service robot domains

6.3 Relationship to other OMG Specifications and activities

6.3.1 Relationship to OMG specifications

Submitters shall examine the following OMG specifications for possible benefit:

- Super Distributed Objects (SDO) Specification version 1.1 [formal/2008-10-01]

- Unified Modeling Language: Infrastructure version 2.3 [formal/2010-05-03]
- Unified Modeling Language: Superstructure version 2.3 [formal/2010-05-05]
- Lightweight CORBA Component Model 4.0 [formal/06-04-01]
- Robotic Technology Component specification version 1.0 [formal/08-04-04]
- Robotic Localization Service version 1.0 [formal/2010-02-03]

6.3.2 Relationship to other OMG Documents and work in progress

None

6.4 Related non-OMG Activities, Documents and Standards

Proposals may include existing systems, documents, user recognition service interface, and standards that are relevant to the problems discussed in this RFP. They can be used as background information for the proposal.

Examples:

- IEEE Robotics and Automation Society, Technical Committee on Network Robot
- IEEE Robotics and Automation Society, Technical Committee on Programming Environment in Robotics and Automation
- ISO/ SC 37 Projects relate to ISO/IEC 19784-1(BioAPI Ver 2.0)
- ISO/TC184/SC2 Robots and robotic devices

6.5 Mandatory Requirements

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

(2) Proposals shall provide a specification of the following interfaces between robotic service applications and the HRI Engine:

- Interface to obtain information actively from HRI Engine
- Interface to receive notification of real-time events from the HRI Engine
- Control interface to send commands to HRI Engine functions

(3) Proposals shall specify data structure for each interface.

(4) Proposals shall provide a specification of the following common messages for all HRI Engines.

- Event notification
  - Person detected
  - Person identified
  - Command action completed
- Query
  - Person ID
  - Position of person / robot
  - Status of HRI Engine
- Command
  - Start / stop person detection
  - Start / stop person identification

(5) The specification shall meet the following criteria:

- Be general enough to incorporate robotic components for various sensors, actuators and algorithms in HRI Engine.
- Satisfy interoperability and reusability, to allow an HRI Engine to be replaced with another without difficulty.

(6) Proposals shall specify existing technologies to achieve functions required for RoIS framework.

(7) Proposal shall specify functions that cannot be achieved by existing technologies.

(8) Proposals shall provide a Platform Independent Model (PIM).

6.6 Optional Requirements

(1) Proposals may specify schemes for the functionalities listed below.

- Error handling for each interface type
- Returning command results and status

(2) Proposals may provide a schema to describe message profiles.

(3) Proposals may provide a Platform Specific Model (PSM) as C++, CORBA-specific model, XML or RTC for RoIS framework.

6.7 Issues to be discussed

These issues will be considered during submission evaluation. They should not be part of the proposed normative specification. (Place them in Part I of the submission.)

- Proposals shall demonstrate their feasibility by giving a specific example application based on the proposed model.
- Proposals shall demonstrate their applicability to existing robotic interaction service technologies.
- Proposals shall discuss simplicity of implementation.
- Proposals shall discuss the possibility of applying the proposed model to other fields of interest such as intelligent service robot applications.
- Proposals shall show that they are independent of on-the-wire protocol communication technology.
- Proposals shall discuss their relationship to and dependency on existing middleware standards, such as CORBA [CORBA], DDS [DDS] or RTC [RTC].
- Proposals shall discuss their generality with respect to the anticipated range of sensors, actuators and the algorithms in the HRI Engine.

6.8 Evaluation Criteria

Proposals will be evaluated in terms of consistency in their specifications, feasibility and versatility across a wide range of different robot applications.

6.9 Other information unique to this RFP

None

6.10 RFP Timetable

The timetable for this RFP is given below. Note that the TF or its parent TC may, in certain circumstances, extend deadlines while the RFP is running, or may elect to have more than one Revised Submission step. The latest timetable can always be found at the OMG *Work In Progress* page at http://www.omg.org/schedules under the item identified by the name of this RFP. Note that "<month>" and "<approximate month>" is the name of the month spelled out; e.g., January.

Event or Activity	Actual Date
Preparation of RFP by TF	*24. May* 2010
RFP placed on OMG document server	24. May 2010
Approval of RFP by Architecture Board *Review by TC*	21. June 2010
TC votes to issue RFP	25. June 2010
LOI to submit to RFP due	13. September 2010
Initial Submissions due and placed on OMG document server ("Four week rule")	8. November 2010
Voter registration closes	29. November 2010
Initial Submission presentations	December 2010
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Four week rule")	June 2011
Revised Submission presentations	June 2011
Final evaluation and selection by TF *Recommendation to AB and TC*	
Approval by Architecture Board *Review by TC*	
TC votes to recommend specification	

BoD votes to adopt specification	

Appendix A References and Glossary Specific to this RFP

A.1 References Specific to this RFP

[RTC] Robotic Technology Component specification version 1.0, http://www.omg.org/spec/RTC/1.0/

A.2 Glossary Specific to this RFP

None

Appendix B General Reference and Glossary

B.1 General References

The following documents are referenced in this document:

[ATC] Air Traffic Control Specification, *http://www.omg.org/technology/documents/formal/air_traffic_control.htm*

[BCQ] OMG Board of Directors Business Committee Questionnaire, http://doc.omg.org/bc/07-08-06

[CCM] CORBA Core Components Specification, *http://www.omg.org/technology/documents/formal/components.htm*

[CORBA] Common Object Request Broker Architecture (CORBA/IIOP), *http://www.omg.org/technology/documents/formal/corba_iiop.htm*

[CSIV2] [CORBA] Chapter 26

[CWM] Common Warehouse Metamodel Specification, *http://www.omg.org/technology/documents/formal/cwm.htm*

[DAIS] Data Acquisition from Industrial Systems, *http://www.omg.org/technology/documents/formal/dais.htm*

[EDOC] UML Profile for EDOC Specification, *http://www.omg.org/techprocess/meetings/schedule/UML_Profile_for_EDOC_FTF.html*

[EJB] “Enterprise JavaBeans™”, *http://java.sun.com/products/ejb/docs.html*

[FORMS] “ISO PAS Compatible Submission Template”. *http://www.omg.org/cgi-bin/doc?pas/2003-08-02*

[GE] Gene Expression, *http://www.omg.org/technology/documents/formal/gene_expression.htm*

[GLS] General Ledger Specification , *http://www.omg.org/technology/documents/formal/gen_ledger.htm*

[Guide] The OMG Hitchhiker's Guide,, *http://www.omg.org/cgi-bin/doc?hh*

[IDL] ISO/IEC 14750 also see [CORBA] Chapter 3.

[IDLC++] IDL to C++ Language Mapping, *http://www.omg.org/technology/documents/formal/c++.htm*

[Inventory] Inventory of Files for a Submission/Revision/Finalization, *http://doc.omg.org/smsc/2007-09-05*

[MDAa] OMG Architecture Board, "Model Driven Architecture - A Technical Perspective”, *http://www.omg.org/mda/papers.htm*

[MDAb] “Developing in OMG's Model Driven Architecture (MDA),” *http://www.omg.org/docs/omg/01-12-01.pdf*

[MDAc] “MDA Guide” (*http://www.omg.org/docs/omg/03-06-01.pdf*)

[MDAd] “MDA "The Architecture of Choice for a Changing World™"”, *http://www.omg.org/mda*

[MOF] Meta Object Facility Specification, *http://www.omg.org/technology/documents/formal/mof.htm*

[MQS] “MQSeries Primer”, *http://www.redbooks.ibm.com/redpapers/pdfs/redp0021.pdf*

[NS] Naming Service, *http://www.omg.org/technology/documents/formal/naming_service.htm*

[OMA] “Object Management Architecture™”, http://www.omg.org/oma/

[OTS] Transaction Service, http://www.omg.org/technology/documents/formal/transaction_service.htm

[P&P] Policies and Procedures of the OMG Technical Process, http://www.omg.org/cgi-bin/doc?pp

[PIDS] Personal Identification Service, http://www.omg.org/technology/documents/formal/person_identification_service.htm

[RAD] Resource Access Decision Facility, http://www.omg.org/technology/documents/formal/resource_access_decision.htm

[RFC2119] IETF Best Practices: Key words for use in RFCs to Indicate Requirement Levels, (http://www.ietf.org/rfc/rfc2119.txt).

[RM-ODP] ISO/IEC 10746

[SEC] CORBA Security Service, http://www.omg.org/technology/documents/formal/security_service.htm

[TOS] Trading Object Service, http://www.omg.org/technology/documents/formal/trading_object_service.htm

[UML] Unified Modeling Language Specification, http://www.omg.org/technology/documents/formal/uml.htm

[UMLC] UML Profile for CORBA, http://www.omg.org/technology/documents/formal/profile_corba.htm

[XMI] XML Metadata Interchange Specification, http://www.omg.org/technology/documents/formal/xmi.htm

[XML/Value] XML Value Type Specification, http://www.omg.org/technology/documents/formal/xmlvalue.htm

B.2 General Glossary

Architecture Board (AB) - The OMG plenary that is responsible for ensuring the technical merit and MDA-compliance of RFPs and their submissions.

Board of Directors (BoD) - The OMG body that is responsible for adopting technology.

Common Object Request Broker Architecture (CORBA) - An OMG distributed computing platform specification that is independent of implementation languages.

Common Warehouse Metamodel (CWM) - An OMG specification for data repository integration.

CORBA Component Model (CCM) - An OMG specification for an implementation language independent distributed component model.

Interface Definition Language (IDL) - An OMG and ISO standard language for specifying interfaces and associated data structures.

Letter of Intent (LOI) - A letter submitted to the OMG BoD's Business Committee signed by an officer of an organization signifying its intent to respond to the RFP and confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements.

Mapping - Specification of a mechanism for transforming the elements of a model conforming to a particular metamodel into elements of another model that conforms to another (possibly the same) metamodel.

Metadata - Data that represents models. For example, a UML model; a CORBA object model expressed in IDL; and a relational database schema expressed using CWM.

Metamodel - A model of models.

Meta Object Facility (MOF) - An OMG standard, closely related to UML, that enables metadata management and language definition.

Model - A formal specification of the function, structure and/or behavior of an application or system.

Model Driven Architecture (MDA) - An approach to IT system specification that separates the specification of functionality from the specification of the implementation of that functionality on a specific technology platform.

Normative – Provisions that one must conform to in order to claim compliance with the standard. (as opposed to non-normative or informative which is explanatory material that is included in order to assist in understanding the standard and does not contain any provisions that must be conformed to in order to claim compliance).

Normative Reference – References that contain provisions that one must conform to in order to claim compliance with the standard that contains said normative reference.

Platform - A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented.

Platform Independent Model (PIM) - A model of a subsystem that contains no information specific to the platform, or the technology that is used to realize it.

Platform Specific Model (PSM) - A model of a subsystem that includes information about the specific technology that is used in the realization of it on a specific platform, and hence possibly contains elements that are specific to the platform.

Request for Information (RFI) - A general request to industry, academia, and any other interested parties to submit information about a particular technology area to one of the OMG's Technology Committee subgroups.

Request for Proposal (RFP) - A document requesting OMG members to submit proposals to an OMG Technology Committee. Such proposals must be received by a certain deadline and are evaluated by the issuing Task Force.

Task Force (TF) - The OMG Technology Committee subgroup responsible for issuing a RFP and evaluating submission(s).

Technology Committee (TC) - The body responsible for recommending technologies for adoption to the BoD. There are two TCs in OMG – the *Platform TC* (PTC) focuses on IT and modeling infrastructure related standards; while the *Domain TC* (DTC) focuses on domain specific standards.

Unified Modeling Language (UML) - An OMG standard language for specifying the structure and behavior of systems. The standard defines an abstract syntax and a graphical concrete syntax.

UML Profile - A standardized set of extensions and constraints that tailors UML to particular use.

XML Metadata Interchange (XMI) - An OMG standard that facilitates interchange of models via XML documents.

< Note to RFP Editors: Append additional appendices if needed here and update the list and brief description of appendices in Chapter 1. >

Object Management Group

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

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

Dynamic Deployment and Configuration for Robotic Technology Components (DDC4RTC)

Request For Proposal

OMG Document: mars/2010-06-16 Convenience doc without change bars
mars/2010-06-18 Convenience doc with change bars

Letters of Intent due: 17 September 2010
Submissions due: 8 November 2010

Objective of this RFP

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

In particular, the proposal shall provide:

- Ways to search for and deploy Robotic Technology Components (RTC) into robotic systems at run-time.
- Ways to notify the relevant RTC instances of environment changes.
- Ways to search for appropriate RTC instances and dynamically configure them.

6.0 Specific Requirements on Proposals

6.1 Problem Statement

Generally, most component-based software platforms have their own specifications for component deployment and configuration. We already have the Robotic Technology Component (RT-Component: RTC) Specification in the OMG for a component-based robot software platform. The component model for robotics domain-specific design patterns is described in the current RTC specification. However, functionality such as deployment and configuration, which are usually supported by middleware services or facilities, are not defined.

As the general UML (Unified Modeling Language) component model has been extended in the RTC specification, in order to apply it to the robotics domain, some services and facilities also should be extended with robot-specific characteristics. Existing specifications are inadequate to meet the requirements of robotics. They are general purpose and are oriented toward static software systems, not dynamic software systems such as robotic systems. This RFP describes dynamic deployment and configuration specific to RT components.

A robot is a mobile system that interacts with the real environment. Figure 1 shows the typical robotic application environment. A robot moves around from one place to another in the dynamic environment and it can use the environment's resources, which include sensors, robotic devices and other robots.

In the robot application development phase, we may not know what environment the robot will be installed to and, furthermore, what environment changes will occur while the robot is operating. These dynamic characteristics should be considered not at software build-time but at runtime. This means that RTC-based systems can be deployed and reconfigured at runtime according to environment changes. Therefore a new flexible, adaptive, and dynamically configurable mechanism and method are required to meet the dynamic characteristics of robot applications.

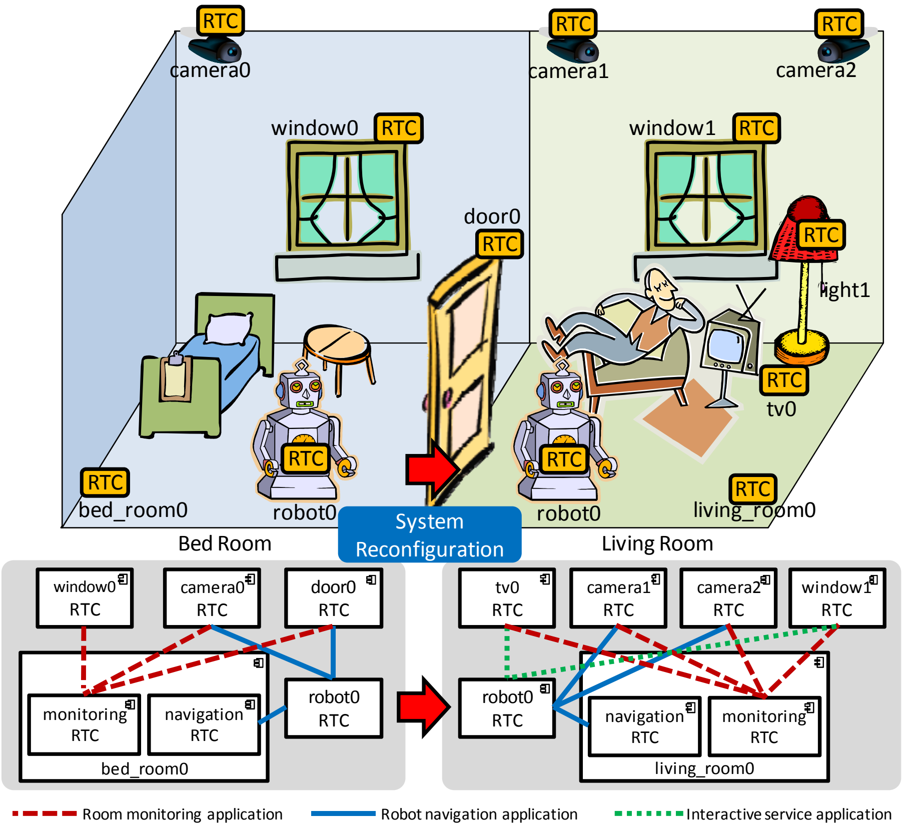


Figure 1 Typical robotic application environment

In order to address functionality of dynamic deployment and configuration, the following issues should be included:

1. RTC profile

A component can generally have common profile information, and as shown in Figure 2, this profile information can be used in the component development phase, system development phase, simulation, and so on. Furthermore, when using a repository server that accumulates many components, this information can be utilized for storing, searching and retrieving components from it. This is called a component profile, which is described in the RTC specification.

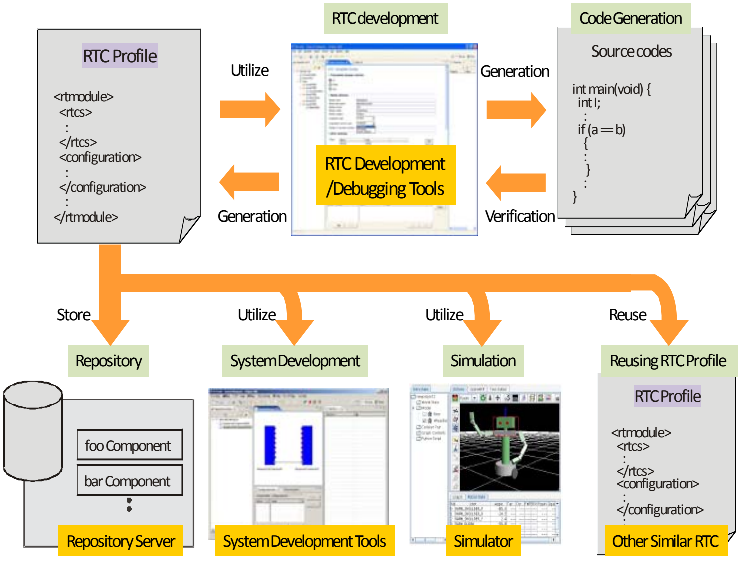


Figure 2 Use of the RTC Profile

2. RTC-based system profile

An RTC-based system is generally built by composing the RTCs or RTC-based subsystems. An RTC-based system or subsystem shall consist of connection information among RTCs, configuration information for RTCs, and so on. This information is called an RTC-based system profile. As shown in Figure 3, this information can be utilized for simulation or component deployment for actual systems. Usually, the components are installed on the target system prior to starting it. (Here, we are focusing on static systems only. The dynamic case will be addressed in the following issues.) Therefore, the person who wants to deploy components has to prepare all the components that constitute the target system. Also, as the number of RTCs and component developers (or developing organizations) is increasing, the person in charge of deployment cannot personally manage all the RTCs that are built. In these cases, a central repository, which manages all the RTCs built, is very helpful in deploying to robot systems. It enables people who want to deploy components to search for what they want in the repository and download/install the components found onto the target hardware. Moreover, if they describe the composing components in a computer-understandable form, the RT middleware is now able to automatically search, download, and install the RTCs while deploying the system.

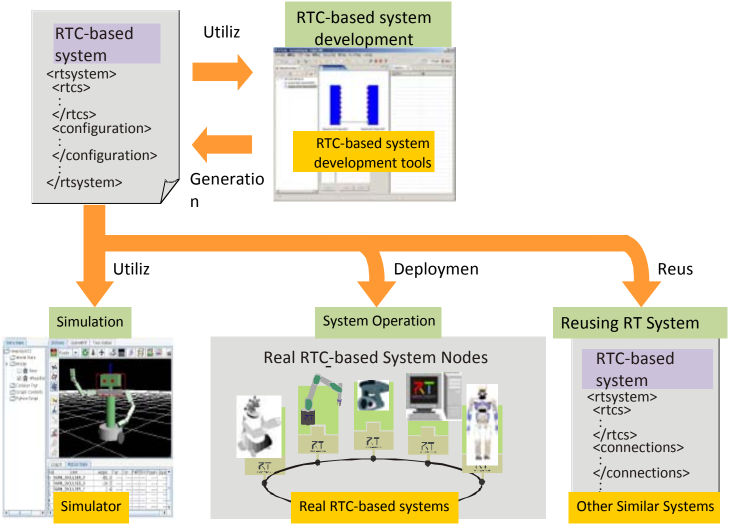


Figure 3 Use of RTC-based system profile

3. RTC-based system deployment

The current RTC specification does not provide a declarative way to compose RTCs to build a robot application or system. Many component based systems present a deployment method that can describe the target application (or system) by combining their components. However those descriptions are not suitable for the robotics domain, which inherently suffers from environment changes during operation time due to mobility. Links between components established at deployment time become obsolete as a robot moves to a new environment. In order to handle these situations, the method of describing the links should be declarative enough such that the description remains valid as the surrounding environment changes over time.

A robot consists of different kinds of sensor and actuator devices and usually includes multiple computing nodes. The RTC-based system should consider the automated deployment of RTCs to the distributed nodes. However, the existing RTC specification suffers from insufficient support for deployment and configuration of software components of distributed applications.

4. RTC instance lookup

As mentioned above, a robot application (or system) consists of RTCs and links among them. Here, the components which are participating in the link are not limited to a single node (or host) but are placed on separate nodes. In this case, it is necessary to search for appropriate component instances running throughout the distributed system. To fulfill these requirements, the specification should provide an RTC directory, which is in charge of searching for a candidate component instance to be linked with other component instances. Since a meta-

information-based component instance search is needed, the specification must also define the data model for the meta-information of RTC. Finally, in order for the RTC directory to find the right component instance that matches the requirements, all the meta-information of the component instances running throughout the distributed system must be known to the directory. Therefore the specification must also specify the registering (and conversely unregistering) processes by which all component instances register their own meta-information with the directory.

5. RTC instance tracking

As mentioned earlier, robotic systems have a unique characteristic in that their surrounding context may change during operation time. In such cases, a link between component instances could become invalid, and so need to be removed and re-established between different component instances. This kind of configuration commonly results from the impairment of the participating component instances and/or changes in the robot location. To support such configuration, the robot application (or system) needs to be notified whenever the situation changes. Since not all changes require configuration, it must be possible to specify the specific environment changes that trigger configuration. It is desirable that the specification is also based on meta-information of component instances and looks similar to that for the component instance searching.

We already have the RTC specification in the OMG for the reusability and interoperability of robot modules. We also have the DEPL (Deployment and Configuration of Component-based Distributed Applications specification) in the OMG for deployment and configuration of component based distributed applications.

RTC defines a component model and infrastructure services applicable to the domain of robotics software development. By extending the general-purpose component functionality of UML with direct support for domain-specific structural and behavioral design patterns, RTCs serve as powerful building blocks in an RTC-based system. The RTC specification provides a way to make RTCs and build RTC-based systems. However, it does not discuss how to deploy and configure RTCs at runtime.

DEPL defines installation, configuration, planning, preparation, and launch process for component-based applications. DEPL could support the deployment and configuration of components at build time. However it cannot cover the deployment and configuration of components at run time and meet the dynamic characteristics for robotic systems.

To use DEPL in the robotics domain and expand RTC, the RFP proposes the specifications for the dynamic deployment and configuration specific to RT components.

6.2 Scope of Proposals Sought

This RFP solicits proposals to specify common interfaces and common data models for RTC dynamic deployment and configuration that is specific and relevant to robot applications. The proposals shall include a PIM, using UML in the most recent public available version, and one or more PSMs, including one based on OMG IDL (Interface Definition Language) and XML (eXtensible Mark-up Language).

The proposed specification shall provide functionality for component deployment and dynamic system configuration for RTC based systems. The specification must be general enough to allow a variety of robotic systems to be easily constructed, and must be provided for interoperability.

It is necessary to consider the following in the specification:

(1) The repository service interfaces for storing, searching, and retrieving RTCs, and the data model for the component profile description. The component profile might be extensible to include related hardware's functional, mechanical, electrical, physical or geometrical information. This information is helpful in the design and simulation processes.

(2) The repository service interfaces for storing, searching, and retrieving RTC-based systems, and the data model for the RTC-based system profile description.

(3) The service interfaces for the deployment of RTCs into the nodes that constitute RTC-based systems at run time, and the data model for describing the details of deployment.

(4) The directory service interfaces for RTC instance discovery, and the data model for describing the RTC instance. In addition to functions such as registration and searching, this service might provide certain functionality such as notifying environmental changes to RTC based applications or filtering such events based on previously registered condition.

6.3 Relationship to other OMG Specifications and activities

6.3.1 Relationship to OMG specifications

- Platform Independent Model and Platform Specific Model for super Distributed Object Specification Version 1.1 [formal/2008-10-01] http://www.omg.org/spec/SDO/1.1
- Robotic Technology Component Specification Version 1.0 [formal/2008-04-04] http://www.omg.org/spec/RTC/1.0
- Deployment and Configuration of Component-based Distributed Applications Specification OMG Available Specification Version 4.0 [formal/2006-04-02] http://www.omg.org/spec/DEPL/4.0
- Unified Modeling Language: Infrastructure Version 2.3 [formal/2010-05-03] http://www.omg.org/spec/UML/2.3/Infrastructure/PDF/
- Unified Modeling Language: Superstructure Version 2.3 [formal/2010-05-05] http://www.omg.org/spec/UML/2.3/Superstructure/PDF/
- Meta Object Facility (MOF) Core Specification OMG Available Specification Version 2.0 [formal/06-01-01] http://www.omg.org/spec/MOF/2.0/
- Common Object Request Broker Architecture (CORBA/IIOP) 3.1 [formal/2008-01-04, formal/2008-01-06, formal/2008-01-08] http://www.omg.org/spec/CORBA/3.1/Interfaces/PDF/
- CORBA Component Model OMG Available Specification Version 4.0 [formal/2006-04-01] http://www.omg.org/spec/CCM/4.0
- Lightweight Services Specification Version 1.0 [formal/04-10-01] http://www.omg.org/spec/LtSVC/1.0/
- Event Service Specification Version 1.2 [formal/04-10-02] http://www.omg.org/spec/EVNT/1.2/
- Naming Service Specification Version 1.3 [formal/04-10-03] http://www.omg.org/spec/NAM/1.3/
- Enhanced View of Time Specification Version 2.1 [formal/08-08-01] http://www.omg.org/spec/EVoT/2.0

- Property Service Specification Version 1.0 [formal/00-06-22] http://www.omg.org/spec/PROP/1.0/

- Mobile Agent Facility Specification Version 1.0 [formal/2000-01-02] http://www.omg.org/spec/MOBFAC/1.0/

- PIM and PSM for Software Radio Components (SDRP) Version 1.0 [formal/07-03-01] http://www.omg.org/spec/SDRP/

- UML Profile For MARTE: Modeling And Analysis Of Real-Time Embedded Systems [formal 2009-11-02] http://www.omg.org/spec/MARTE/1.0

- MARTE Profile XMI file [ptc/09-05-15] http://www.omg.org/spec/MARTE/20090501

- MARTE model library XMI file [ptc/09-05-16] http://www.omg.org/spec/MARTE/20090502

6.3.2 Relationship to other OMG Documents and work in progress

None

6.4 Related non-OMG Activities, Documents and Standards

- CLARAty: Coupled Layer Architecture for Robotic Autonomy http://robotics.jpl.nasa.gov/tasks/claraty/homepage.html

- Network Robot Forum http://www.scat.or.jp/nrf/

- IEEE Robotics and Automation Society, Technical Committee on Network Robot

- IEEE Robotics and Automation Society, Technical Committee on Programming Environments in Robotics and Automation

- OpenRT Platform http://www.openrtp.jp

- OpenRTM-aist http://www.openrtm.org

- OpenRAVE: http://openrave.programmingvision.com

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

- OROCOS: Open Robot Control Software, Open Realtime Control Service http://www.orocos.org/
- Orca: http://orca-robotics.sourceforge.net/
- ORiN :Open Robot/Resource Interface for the Network: http://www.orin.jp/
- Player/Stage: http://playerstage.sourceforge.net/
- Ptolemy Project: http://ptolemy.eecs.berkeley.edu/
- RCS (Realtime Control Systems Architecture): http://www.isd.mel.nist.gov/projects/rcs/
- ROS: http://www.ros.org
- RSi: Robot Service Initiative: http://www.robotservice.org/
- RT middleware Project: http://www.is.aist.go.jp/rt
- SAE AADL (Society for Automotive Engineers, Architecture Analysis and Design Language): http://www.aadl.info/
- RETF (Robotics Engineering Task Force): http://www.robo-etf.org/
- URC (Ubiquitous Robotic Companion) Project
- Yaorozu Project: http://www.8mg.jp/

6.5 Mandatory Requirements

Proposals shall provide a Platform Independent Model (PIM) expressed in UML and at least one Platform Specific Model (PSM) as CORBA-specific model and XML schema for RTC Dynamic Deployment and Configuration. The models shall meet the following requirements.

Platform independent deployment and configuration model

6.5.1 Proposals shall specify interfaces to services for dynamic configuration and deployment of RTCs.

- storing, searching and retrieving RTC,
- storing, searching and retrieving RTC-based applications

- RTC registration

6.5.2 Proposal shall specify interfaces to initiate RTC configuration based on external and/or internal events. A capability for event filtering shall be provided.

6.5.3 Proposals shall reuse or extend at least the PIM, including terms and definitions, of the deployment architecture as defined by the Deployment and Configuration of Component-based Distributed Applications Specification [DEPL].

Platform independent RTC information model

6.5.4 Proposals shall provide a schema, the RTC Profile, describing RTC characteristics such as basic RTC information, ports information and so on, based on the RTC specification

6.5.5 Proposals shall provide a schema, the RTC-based System Profile, describing RTC-based systems characteristics such as port connection information, configuration information, deployment conditions and so on.

6.5.6 Proposals shall specify query services to discover and interrogate characteristics of RTCs and RTC-based systems.

6.5.7 Proposal shall specify query services to discover characteristics and location information of deployed RTCs and RTC-based systems.

6.6 Optional Requirements

6.6.1 Proposals may support coordinated RTC configuration of multiple robot systems to allow the performance of coordinated tasks.

6.7 Issues to be discussed

These issues will be considered during submission evaluation. They should not be part of the proposed normative specification. (Place them in Part I of the submission.)

6.7.1 Proposals shall discuss the possibility of applying the proposed model to other existing fields/projects of interest that deploy components such as CCM [CCM], SDRP [SDRP], DEPL [DEPL] and other well-known component models.

6.7.2 Proposals shall discuss their relation to and dependency on existing communication protocols or middleware standards, such as CORBA [CORBA] or DDS [DDS].

6.7.3 Proposals shall discuss efficient methods/procedures to avoid the need for extensive information discovery activities when interacting with the environment or other robots.

6.8 Evaluation Criteria

6.8.1 Demonstration of a proposal with a working implementation may aid in selection.

6.8.2 Reuse of existing technology, such as the RTC specification and DEPL specification, is considered important.

6.9 Other information unique to this RFP

None.

6.10 RFP Timetable

The timetable for this RFP is given below. Note that the TF or its parent TC may, in certain circumstances, extend deadlines while the RFP is running, or may elect to have more than one Revised Submission step. The latest timetable can always be found at the OMG *Work In Progress* page at http://www.omg.org/schedules under the item identified by the name of this RFP.

Event or Activity	Actual Date
Preparation of RFP by TF	
RFP placed on OMG document server	*February 22nd, 2010*
Approval of RFP by Architecture Board *Review by TC*	*June, 2010*
TC votes to issue RFP	*June, 2010*
LOI to submit to RFP due	*September 17th, 2010*

Initial Submissions due and placed on OMG document server ("Four week rule")	*November 8th, 2010*
Voter registration closes	*December, 2010*
Initial Submission presentations	*December, 2010*
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Four week rule")	*May , 2011*
Revised Submission presentations	*June, 2011*
Final evaluation and selection by TF *Recommendation to AB and TC*	
Approval by Architecture Board *Review by TC*	
TC votes to recommend specification	*June, 2011*
BoD votes to adopt specification	*September, 2011*

Appendix A References and Glossary Specific to this RFP

A.1 References Specific to this RFP

[CCM] CORBA Components Specification, http://www.omg.org/technology/documents/formal/components.htm

[DDS] Data Distribution Services Specification, http://www.omg.org/spec/DDS/1.2/

[DEPL] Deployment and Configuration of Component-based Distributed Applications Specification OMG Available Specification, http://www.omg.org/spec/DEPL/4.0/

[RTC] Robotic Technology Component specification, http://www.omg.org/spec/RTC/1.0/

[SDO] Super distributed Object Specification, http://www.omg.org/spec/SDO/1.1/

A.2 Glossary Specific to this RFP

Robot application **–**A software application that controls a robot's behavior. Examples include a vacuum cleaning robot and a butler robot.

Super Distributed Object (SDO) **–** A logical representation of a hardware device or a software component that provides well-known functionality and services.

Robotic Technology Component (RTC) **–**A logical representation of a hardware and/or software entity that provides well-known functionality and services.

RTC-based system **–**A system comprised of RTCs connected in a network representing a robotic system, including robot hardware and software algorithms.

Robotic Technology (RT) **–** Robotic Technology (RT) is a general term of the technology originating in robotics, and it means not only the standalone robot but technical element which constitutes robots.

RT-component profile **–** A description that represents the static state of an RT Component that is referred to other RT Components.

RTC-based system profile - A description of how RT-components are connected and interact with each other, and RT-component configuration parameters.

Deployment profile - A description of information used in deploying components, including RT-component profiles.

Meta-information **–** Data that represents the properties of running RT component instance.

Directory **–** A storage that manages the references and the meta-information of running RT component instances.

Environment change **–** Situation that available resources in environment are changed such as sensors, actuators, and other robots, when a robotic system moves to new environment.

Deployment - all of the activities that make a set of components available for use and consist of installation and activation of the components.

Appendix B General Reference and Glossary

B.1 General References

The following documents are referenced in this document:

[ATC] Air Traffic Control Specification, *http://www.omg.org/technology/documents/formal/air_traffic_control.htm*

[BCQ] OMG Board of Directors Business Committee Questionnaire, *http://doc.omg.org/bc/07-08-06*

[CCM] CORBA Core Components Specification, *http://www.omg.org/technology/documents/formal/components.htm*

[CORBA] Common Object Request Broker Architecture (CORBA/IIOP), *http://www.omg.org/technology/documents/formal/corba_iiop.htm*

[CSIV2] [CORBA] Chapter 26

[CWM] Common Warehouse Metamodel Specification, *http://www.omg.org/technology/documents/formal/cwm.htm*

[DAIS] Data Acquisition from Industrial Systems, *http://www.omg.org/technology/documents/formal/dais.htm*

[EDOC] UML Profile for EDOC Specification, *http://www.omg.org/techprocess/meetings/schedule/UML_Profile_for_EDOC_FTF.html*

[EJB] “Enterprise JavaBeans™”, *http://java.sun.com/products/ejb/docs.html*

[FORMS] “ISO PAS Compatible Submission Template”. *http://www.omg.org/cgi-bin/doc?pas/2003-08-02*

[GE] Gene Expression, *http://www.omg.org/technology/documents/formal/gene_expression.htm*

[GLS] General Ledger Specification , *http://www.omg.org/technology/documents/formal/gen_ledger.htm*

[Guide] The OMG Hitchhiker's Guide,, *http://www.omg.org/cgi-bin/doc?hh*

[IDL] ISO/IEC 14750 also see [CORBA] Chapter 3.

[IDLC++] IDL to C++ Language Mapping, *http://www.omg.org/technology/documents/formal/c++.htm*

[Inventory] Inventory of Files for a Submission/Revision/Finalization, *http://doc.omg.org/smsc/2007-09-05*

[MDAa] OMG Architecture Board, "Model Driven Architecture - A Technical Perspective*", http://www.omg.org/mda/papers.htm*

[MDAb] "Developing in OMG's Model Driven Architecture (MDA)," *http://www.omg.org/docs/omg/01-12-01.pdf*

[MDAc] "MDA Guide" (*http://www.omg.org/docs/omg/03-06-01.pdf*)

[MDAd] "MDA "The Architecture of Choice for a Changing World™"", *http://www.omg.org/mda*

[MOF] Meta Object Facility Specification, *http://www.omg.org/technology/documents/formal/mof.htm*

[MQS] "MQSeries Primer", *http://www.redbooks.ibm.com/redpapers/pdfs/redp0021.pdf*

[NS] Naming Service, *http://www.omg.org/technology/documents/formal/naming_service.htm*

[OMA] "Object Management Architecture™", *http://www.omg.org/oma/*

[OTS] Transaction Service, *http://www.omg.org/technology/documents/formal/transaction_service.htm*

[P&P] Policies and Procedures of the OMG Technical Process, *http://www.omg.org/cgi-bin/doc?pp*

[PIDS] Personal Identification Service, *http://www.omg.org/technology/documents/formal/person_identification_service.htm*

[RAD] Resource Access Decision Facility, *http://www.omg.org/technology/documents/formal/resource_access_decision.htm*

[RFC2119] IETF Best Practices: Key words for use in RFCs to Indicate Requirement Levels, (*http://www.ietf.org/rfc/rfc2119.txt)*.

[RM-ODP] ISO/IEC 10746

[SCA] Software Communications Architecture (SCA), http://sca.jpeojtrs.mil/sca.asp

[SDRP] Software Radio Components (SDRP), http://www.omg.org/spec/SDRP/

[SEC] CORBA Security Service, *http://www.omg.org/technology/documents/formal/security_service.htm*

[TOS] Trading Object Service, *http://www.omg.org/technology/documents/formal/trading_object_service.htm*

[UML] Unified Modeling Language Specification, *http://www.omg.org/technology/documents/formal/uml.htm*

[UMLC] UML Profile for CORBA, *http://www.omg.org/technology/documents/formal/profile_corba.htm*

[XMI] XML Metadata Interchange Specification, *http://www.omg.org/technology/documents/formal/xmi.htm*

[XML/Value] XML Value Type Specification, *http://www.omg.org/technology/documents/formal/xmlvalue.htm*

B.2 General Glossary

Architecture Board (AB) - The OMG plenary that is responsible for ensuring the technical merit and MDA-compliance of RFPs and their submissions.

Board of Directors (BoD) - The OMG body that is responsible for adopting technology.

Common Object Request Broker Architecture (CORBA) - An OMG distributed computing platform specification that is independent of implementation languages.

Common Warehouse Metamodel (CWM) - An OMG specification for data repository integration.

CORBA Component Model (CCM) - An OMG specification for an implementation language independent distributed component model.

Interface Definition Language (IDL) - An OMG and ISO standard language for specifying interfaces and associated data structures.

Letter of Intent (LOI) - A letter submitted to the OMG BoD's Business Committee signed by an officer of an organization signifying its intent to respond to the RFP and confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements.

Mapping - Specification of a mechanism for transforming the elements of a model conforming to a particular metamodel into elements of another model that conforms to another (possibly the same) metamodel.

Metadata - Data that represents models. For example, a UML model; a CORBA object model expressed in IDL; and a relational database schema expressed using CWM.

Metamodel - A model of models.

Meta Object Facility (MOF) - An OMG standard, closely related to UML, that enables metadata management and language definition.

Model - A formal specification of the function, structure and/or behavior of an application or system.

Model Driven Architecture (MDA) - An approach to IT system specification that separates the specification of functionality from the specification of the implementation of that functionality on a specific technology platform.

Normative – Provisions that one must conform to in order to claim compliance with the standard. (as opposed to non-normative or informative which is explanatory material that is included in order to assist in understanding the standard and does not contain any provisions that must be conformed to in order to claim compliance).

Normative Reference – References that contain provisions that one must conform to in order to claim compliance with the standard that contains said normative reference.

Platform - A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented.

Platform Independent Model (PIM) - A model of a subsystem that contains no information specific to the platform, or the technology that is used to realize it.

Platform Specific Model (PSM) - A model of a subsystem that includes information about the specific technology that is used in the realization of it on a specific platform, and hence possibly contains elements that are specific to the platform.

Request for Information (RFI) - A general request to industry, academia, and any other interested parties to submit information about a particular technology area to one of the OMG's Technology Committee subgroups.

Request for Proposal (RFP) - A document requesting OMG members to submit proposals to an OMG Technology Committee. Such proposals must be received by a certain deadline and are evaluated by the issuing Task Force.

Task Force (TF) - The OMG Technology Committee subgroup responsible for issuing a RFP and evaluating submission(s).

Technology Committee (TC) - The body responsible for recommending technologies for adoption to the BoD. There are two TCs in OMG – the *Platform TC* (PTC) focuses on IT and modeling infrastructure related standards; while the *Domain TC* (DTC) focuses on domain specific standards.

Unified Modeling Language (UML) - An OMG standard language for specifying the structure and behavior of systems. The standard defines an abstract syntax and a graphical concrete syntax.

UML Profile - A standardized set of extensions and constraints that tailors UML to particular use.

XML Metadata Interchange (XMI) - An OMG standard that facilitates interchange of models via XML documents.

robotics/2010-06-27

mars/10-06-17

Errata to revised submission of the Deployment and Dynamic Configuration (DDC) of Robotic Technology Component Request For Proposal, mars/10-06-05

This document lists some minor errors and corrections to the revised submission of the Dynamic Deployment and Configuration (DDC) of RTC Request For Proposal, mars/10-06-05

Change Overview

1. Based on the comments from AB, LOI date has been updated from August 31st to September 17th
2. According to the WG discussion with Vanderbilt University, Thales, Zeligsoft, ETRI and AIST, the title of RFP has been changed to "Dynamic Deployment and Configuration for Robotic Technology Component (DDC4RTC) RFP."
3. Some mistakes in specification references have been corrected.

Object Management Group

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

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

Dynamic Deployment and Configuration ~~(DDC)~~ for~~of~~ Robotic Technology Components (DDC4RTC)

Request For Proposal ~~Draft~~

OMG Document: mars/2010-06-16 Convenience doc without change bars
mars/2010-06-18 Convenience doc with change bars

Letters of Intent due: 17~~31~~ September~~August~~ 2010
Submissions due: 8 November 2010

Objective of this RFP

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

In particular, the proposal shall provide:

- Ways to search for and deploy Robotic Technology Components (RTC) into robotic systems at run-time.
- Ways to notify the relevant RTC instances of environment changes.
- Ways to search for appropriate RTC instances and dynamically configure them.

6.0 Specific Requirements on Proposals

6.1 Problem Statement

Generally, most component-based software platforms have their own specifications for component deployment and configuration. We already have the Robotic Technology Component (RT-Component: RTC) Specification in the OMG for a component-based robot software platform. The component model for robotics domain-specific design patterns is described in the current RTC specification. However, functionality such as deployment and configuration, which are usually supported by middleware services or facilities, are not defined.

As the general UML (Unified Modeling Language) component model has been extended in the RTC specification, in order to apply it to the robotics domain, some services and facilities also should be extended with robot-specific characteristics. Existing specifications are inadequate to meet the requirements of robotics. They are general purpose and are oriented toward static software systems, not dynamic software systems such as robotic systems. This RFP describes dynamic deployment and configuration specific to RT components.

A robot is a mobile system that interacts with the real environment. Figure 1 shows the typical robotic application environment. A robot moves around from one place to another in the dynamic environment and it can use the environment's resources, which include sensors, robotic devices and other robots.

In the robot application development phase, we may not know what environment the robot will be installed to and, furthermore, what environment changes will occur while the robot is operating. These dynamic characteristics should be considered not at software build-time but at runtime. This means that RTC-based systems can be deployed and reconfigured at runtime according to environment changes. Therefore a new flexible, adaptive, and dynamically configurable mechanism and method are required to meet the dynamic characteristics of robot applications.

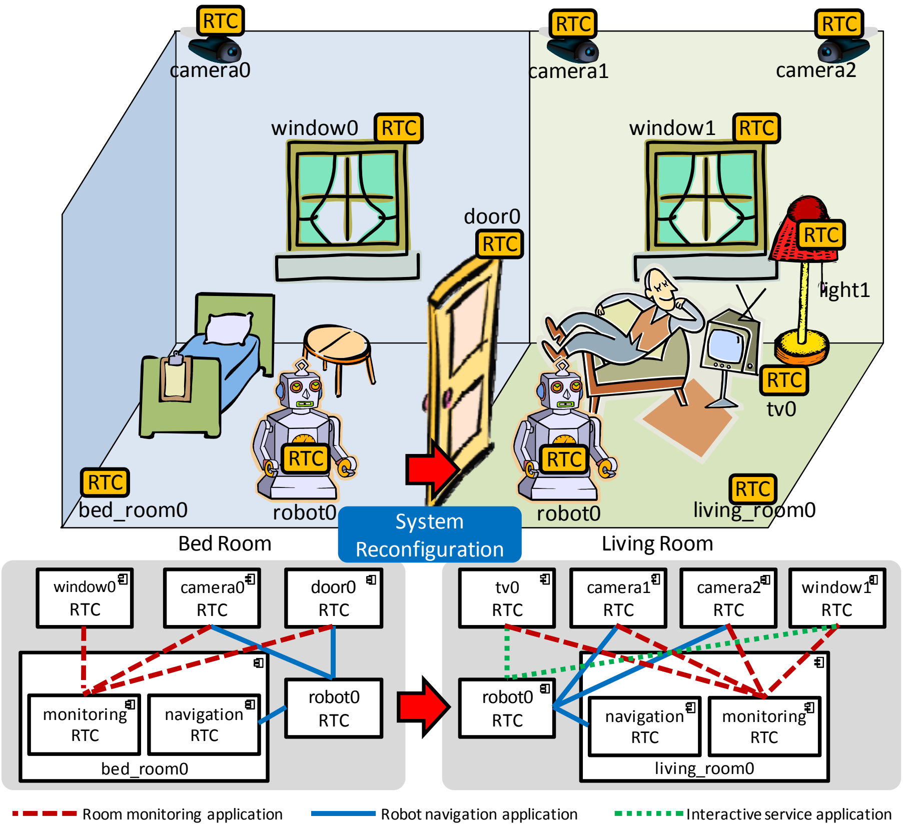
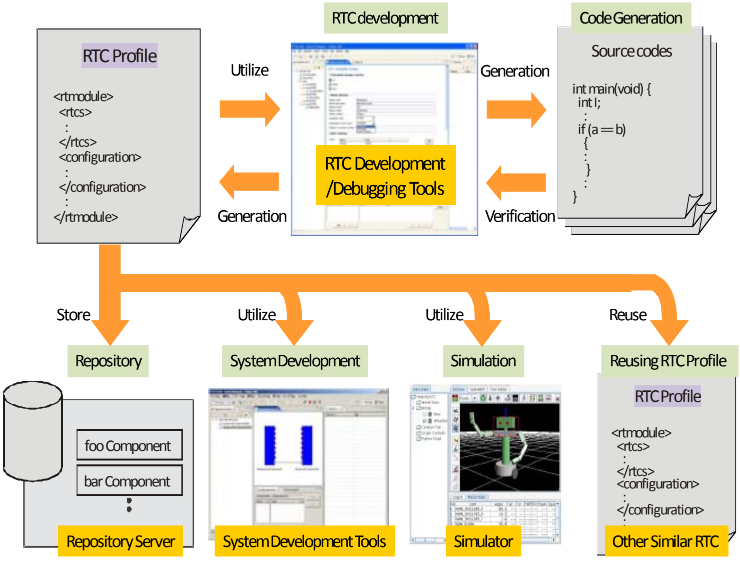


Figure 1 Typical robotic application environment

In order to address functionality of dynamic deployment and configuration, the following issues should be included:

1. RTC profile

A component can generally have common profile information, and as shown in Figure 2, this profile information can be used in the component development phase, system development phase, simulation, and so on. Furthermore, when using a repository server that accumulates many components, this information can be utilized for storing, searching and retrieving components from it. This is called a component profile, which is described in the RTC specification.



```
<rtmodule>
 <rtcs>
  :
 </rtcs>
 <configuration>
  :
 </configuration>
  :
</rtmodule>
```

```
int main(void) {
 int I;
  :
 if (a == b)
  {
  :
  }
  :
}
```

```
<rtmodule>
 <rtcs>
  :
 </rtcs>
 <configuration>
  :
 </configuration>
  :
```

Figure 2 Use of the RTC Profile

2. RTC-based system profile

An RTC-based system is generally built by composing the RTCs or RTC-based subsystems. An RTC-based system or subsystem shall consist of connection information among RTCs, configuration information for RTCs, and so on. This information is called an RTC-based system profile. As shown in Figure 3, this information can be utilized for simulation or component deployment for actual systems. Usually, the components are installed on the target system prior to starting it. (Here, we are focusing on static systems only. The dynamic case will be addressed in the following issues.) Therefore, the person who wants to deploy components has to prepare all the components that constitute the target system. Also, as the number of RTCs and component developers (or developing organizations) is increasing, the person in charge of deployment cannot personally manage all the RTCs that are built. In these cases, a central repository, which manages all the RTCs built, is very helpful in deploying to robot systems. It enables people who want to deploy components to search for what they want in the repository and download/install the components found onto the target hardware. Moreover, if they describe the composing components in a computer-understandable form, the RT middleware is now able to automatically search, download, and install the RTCs while deploying the system.

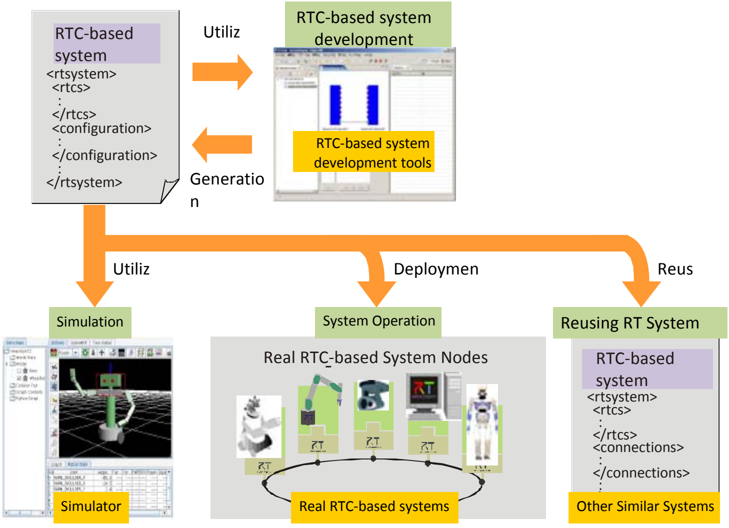


Figure 3 Use of RTC-based system profile

3. RTC-based system deployment

The current RTC specification does not provide a declarative way to compose RTCs to build a robot application or system. Many component based systems present a deployment method that can describe the target application (or system) by combining their components. However those descriptions are not suitable for the robotics domain, which inherently suffers from environment changes during operation time due to mobility. Links between components established at deployment time become obsolete as a robot moves to a new environment. In order to handle these situations, the method of describing the links should be declarative enough such that the description remains valid as the surrounding environment changes over time.

A robot consists of different kinds of sensor and actuator devices and usually includes multiple computing nodes. The RTC-based system should consider the automated deployment of RTCs to the distributed nodes. However, the existing RTC specification suffers from insufficient support for deployment and configuration of software components of distributed applications.

4. RTC instance lookup

As mentioned above, a robot application (or system) consists of RTCs and links among them. Here, the components which are participating in the link are not limited to a single node (or host) but are placed on separate nodes. In this case, it is necessary to search for appropriate component instances running throughout the distributed system. To fulfill these requirements, the specification should provide an RTC directory, which is in charge of searching for a candidate component instance to be linked with other component instances. Since a meta-

information-based component instance search is needed, the specification must also define the data model for the meta-information of RTC. Finally, in order for the RTC directory to find the right component instance that matches the requirements, all the meta-information of the component instances running throughout the distributed system must be known to the directory. Therefore the specification must also specify the registering (and conversely unregistering) processes by which all component instances register their own meta-information with the directory.

5. RTC instance tracking

As mentioned earlier, robotic systems have a unique characteristic in that their surrounding context may change during operation time. In such cases, a link between component instances could become invalid, and so need to be removed and re-established between different component instances. This kind of configuration commonly results from the impairment of the participating component instances and/or changes in the robot location. To support such configuration, the robot application (or system) needs to be notified whenever the situation changes. Since not all changes require configuration, it must be possible to specify the specific environment changes that trigger configuration. It is desirable that the specification is also based on meta-information of component instances and looks similar to that for the component instance searching.

We already have the RTC specification in the OMG for the reusability and interoperability of robot modules. We also have the DEPL (Deployment and Configuration of Component-based Distributed Applications specification) in the OMG for deployment and configuration of component based distributed applications.

RTC defines a component model and infrastructure services applicable to the domain of robotics software development. By extending the general-purpose component functionality of UML with direct support for domain-specific structural and behavioral design patterns, RTCs serve as powerful building blocks in an RTC-based system. The RTC specification provides a way to make RTCs and build RTC-based systems. However, it does not discuss how to deploy and configure RTCs at runtime.

DEPL defines installation, configuration, planning, preparation, and launch process for component-based applications. DEPL could support the deployment and configuration of components at build time. However it cannot cover the deployment and configuration of components at run time and meet the dynamic characteristics for robotic systems.

To use DEPL in the robotics domain and expand RTC, the RFP proposes the specifications for the dynamic deployment and configuration specific to RT components.

6.2 Scope of Proposals Sought

This RFP solicits proposals to specify common interfaces and common data models for RTC dynamic deployment and configuration that is specific and relevant to robot applications. The proposals shall include a PIM, using UML in the most recent public available version, and one or more PSMs, including one based on OMG IDL (Interface Definition Language) and XML (eXtensible Mark-up Language).

The proposed specification shall provide functionality for component deployment and dynamic system configuration for RTC based systems. The specification must be general enough to allow a variety of robotic systems to be easily constructed, and must be provided for interoperability.

It is necessary to consider the following in the specification:

(1) The repository service interfaces for storing, searching, and retrieving RTCs, and the data model for the component profile description. The component profile might be extensible to include related hardware's functional, mechanical, electrical, physical or geometrical information. This information is helpful in the design and simulation processes.

(2) The repository service interfaces for storing, searching, and retrieving RTC-based systems, and the data model for the RTC-based system profile description.

(3) The service interfaces for the deployment of RTCs into the nodes that constitute RTC-based systems at run time, and the data model for describing the details of deployment.

(4) The directory service interfaces for RTC instance discovery, and the data model for describing the RTC instance. In addition to functions such as registration and searching, this service might provide certain functionality such as notifying environmental changes to RTC based applications or filtering such events based on previously registered condition.

6.3 Relationship to other OMG Specifications and activities

6.3.1 Relationship to OMG specifications

- Platform Independent Model and Platform Specific Model for super Distributed Object Specification Version 1.1 [formal/2008-10-01] http://www.omg.org/spec/SDO/1.1

- Robotic Technology Component Specification Version 1.0 [formal/2008-04-04] http://www.omg.org/spec/RTC/1.0

- Deployment and Configuration of Component-based Distributed Applications Specification OMG Available Specification Version 4.0 [formal/2006-04-02] http://www.omg.org/spec/DEPL/4.0

- Unified Modeling Language: Infrastructure Version 2.3 [formal/2010~~09~~-0~~2~~5-0~~4~~3] http://www.omg.org/spec/UML/2.3/Infrastructure/PDF/

- Unified Modeling Language: Superstructure Version 2.3 [formal/2010~~09~~-0~~2~~5-0~~2~~5] http://www.omg.org/spec/UML/2.3/Superstructure/PDF/

- Meta Object Facility (MOF) Core Specification OMG Available Specification Version 2.0 [formal/06-01-01] http://www.omg.org/spec/MOF/2.0/

- Common Object Request Broker Architecture (CORBA/IIOP) 3.1 [formal/2008-01-04, formal/2008-01-06, formal/2008-01-08] http://www.omg.org/spec/CORBA/3.1/Interfaces/PDF/

- CORBA Component Model OMG Available Specification Version 4.0 [formal/2006-04-01] http://www.omg.org/spec/CCM/4.0

- Lightweight Services Specification Version 1.0 [formal/04-10-01] http://www.omg.org/spec/LtSVC/1.0/

- Event Service Specification Version 1.2 [formal/04-10-02] http://www.omg.org/spec/EVNT/1.2/

- Naming Service Specification Version 1.3 [formal/04-10-03] http://www.omg.org/spec/NAM/1.3/

- Enhanced View of Time Specification Version 2~~1~~.~~2~~1 [formal/0~~4~~8-08~~10~~-01~~04~~] http://www.omg.org/spec/EVoT/2.0

- Property Service Specification Version 1.0 [formal/00-06-22] http://www.omg.org/spec/PROP/1.0/
- Mobile Agent Facility Specification Version 1.0 [formal/2000-01-02] http://www.omg.org/spec/MOBFAC/1.0/
- PIM and PSM for Software Radio Components (SDRP) Version 1.0 [formal/07-03-01] http://www.omg.org/spec/SDRP/
- UML Profile For MARTE: Modeling And Analysis Of Real-Time Embedded Systems [formal 2009-11-02] http://www.omg.org/spec/MARTE/1.0
- MARTE Profile XMI file [ptc/09-05-15] http://www.omg.org/spec/MARTE/20090501
- MARTE model library XMI file [ptc/09-05-16] http://www.omg.org/spec/MARTE/20090502

6.3.2 Relationship to other OMG Documents and work in progress

None

6.4 Related non-OMG Activities, Documents and Standards

- CLARAty: Coupled Layer Architecture for Robotic Autonomy http://robotics.jpl.nasa.gov/tasks/claraty/homepage.html
- Network Robot Forum http://www.scat.or.jp/nrf/
- IEEE Robotics and Automation Society, Technical Committee on Network Robot
- IEEE Robotics and Automation Society, Technical Committee on Programming Environments in Robotics and Automation
- OpenRT Platform http://www.openrtp.jp
- OpenRTM-aist http://www.openrtm.org
- OpenRAVE: http://openrave.programmingvision.com
- OPRoS: http://www.opros.or.kr

- OROCOS: Open Robot Control Software, Open Realtime Control Service http://www.orocos.org/
- Orca: http://orca-robotics.sourceforge.net/
- ORiN :Open Robot/Resource Interface for the Network: http://www.orin.jp/
- Player/Stage: http://playerstage.sourceforge.net/
- Ptolemy Project: http://ptolemy.eecs.berkeley.edu/
- RCS (Realtime Control Systems Architecture): http://www.isd.mel.nist.gov/projects/rcs/
- ROS: http://www.ros.org
- RSi: Robot Service Initiative: http://www.robotservice.org/
- RT middleware Project: http://www.is.aist.go.jp/rt
- SAE AADL (Society for Automotive Engineers, Architecture Analysis and Design Language): http://www.aadl.info/
- RETF (Robotics Engineering Task Force): http://www.robo-etf.org/
- URC (Ubiquitous Robotic Companion) Project
- Yaorozu Project: http://www.8mg.jp/

6.5 Mandatory Requirements

Proposals shall provide a Platform Independent Model (PIM) expressed in UML and at least one Platform Specific Model (PSM) as CORBA-specific model and XML schema for RTC Dynamic Deployment and Configuration. The models shall meet the following requirements.

Platform independent deployment and configuration model

6.5.1 Proposals shall specify interfaces to services for dynamic configuration and deployment of RTCs.

- storing, searching and retrieving RTC,
- storing, searching and retrieving RTC-based applications

- RTC registration

6.5.2 Proposal shall specify interfaces to initiate RTC configuration based on external and/or internal events. A capability for event filtering shall be provided.

6.5.3 Proposals shall reuse or extend at least the PIM, including terms and definitions, of the deployment architecture as defined by the Deployment and Configuration of Component-based Distributed Applications Specification [DEPL].

Platform independent RTC information model

6.5.4 Proposals shall provide a schema, the RTC Profile, describing RTC characteristics such as basic RTC information, ports information and so on, based on the RTC specification

6.5.5 Proposals shall provide a schema, the RTC-based System Profile, describing RTC-based systems characteristics such as port connection information, configuration information, deployment conditions and so on.

6.5.6 Proposals shall specify query services to discover and interrogate characteristics of RTCs and RTC-based systems.

6.5.7 Proposal shall specify query services to discover characteristics and location information of deployed RTCs and RTC-based systems.

6.6 Optional Requirements

6.6.1 Proposals may support coordinated RTC configuration of multiple robot systems to allow the performance of coordinated tasks.

6.7 Issues to be discussed

These issues will be considered during submission evaluation. They should not be part of the proposed normative specification. (Place them in Part I of the submission.)

6.7.1 Proposals shall discuss the possibility of applying the proposed model to other existing fields/projects of interest that deploy components such as CCM [CCM], SDRP [SDRP], DEPL [DEPL] and other well-known component models.

6.7.2 Proposals shall discuss their relation to and dependency on existing communication protocols or middleware standards, such as CORBA [CORBA] or DDS [DDS].

6.7.3 Proposals shall discuss efficient methods/procedures to avoid the need for extensive information discovery activities when interacting with the environment or other robots.

6.8 Evaluation Criteria

6.8.1 Demonstration of a proposal with a working implementation may aid in selection.

6.8.2 Reuse of existing technology, such as the RTC specification and DEPL specification, is considered important.

6.9 Other information unique to this RFP

None.

6.10 RFP Timetable

The timetable for this RFP is given below. Note that the TF or its parent TC may, in certain circumstances, extend deadlines while the RFP is running, or may elect to have more than one Revised Submission step. The latest timetable can always be found at the OMG *Work In Progress* page at http://www.omg.org/schedules under the item identified by the name of this RFP.

Event or Activity	Actual Date
Preparation of RFP by TF	
RFP placed on OMG document server	*February 22nd, 2010*
Approval of RFP by Architecture Board *Review by TC*	*June, 2010*
TC votes to issue RFP	*June, 2010*
LOI to submit to RFP due	*September 17th ,~~August 31,~~*

	2010
Initial Submissions due and placed on OMG document server ("Four week rule")	*November 8th, 2010*
Voter registration closes	*December, 2010*
Initial Submission presentations	*December, 2010*
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Four week rule")	*May , 2011*
Revised Submission presentations	*June, 2011*
Final evaluation and selection by TF *Recommendation to AB and TC*	
Approval by Architecture Board *Review by TC*	
TC votes to recommend specification	*June, 2011*
BoD votes to adopt specification	*September, 2011*

Appendix A References and Glossary Specific to this RFP

A.1 References Specific to this RFP

[CCM] CORBA Components Specification, http://www.omg.org/technology/documents/formal/components.htm

[DDS] Data Distribution Services Specification, http://www.omg.org/spec/DDS/1.2/

[DEPL] Deployment and Configuration of Component-based Distributed Applications Specification OMG Available Specification, http://www.omg.org/spec/DEPL/4.0/

[RTC] Robotic Technology Component specification, http://www.omg.org/spec/RTC/1.0/

[SDO] Super distributed Object Specification, http://www.omg.org/spec/SDO/1.1/

A.2 Glossary Specific to this RFP

Robot application **–**A software application that controls a robot's behavior. Examples include a vacuum cleaning robot and a butler robot.

Super Distributed Object (SDO) **–** A logical representation of a hardware device or a software component that provides well-known functionality and services.

Robotic Technology Component (RTC) **–**A logical representation of a hardware and/or software entity that provides well-known functionality and services.

RTC-based system **–**A system comprised of RTCs connected in a network representing a robotic system, including robot hardware and software algorithms.

Robotic Technology (RT) **–** Robotic Technology (RT) is a general term of the technology originating in robotics, and it means not only the standalone robot but technical element which constitutes robots.

RT-component profile **–** A description that represents the static state of an RT Component that is referred to other RT Components.

RTC-based system profile - A description of how RT-components are connected and interact with each other, and RT-component configuration parameters.

Deployment profile - A description of information used in deploying components, including RT-component profiles.

Meta-information **–** Data that represents the properties of running RT component instance.

Directory **–** A storage that manages the references and the meta-information of running RT component instances.

Environment change **–** Situation that available resources in environment are changed such as sensors, actuators, and other robots, when a robotic system moves to new environment.

Deployment - all of the activities that make a set of components available for use and consist of installation and activation of the components.

Appendix B General Reference and Glossary

B.1 General References

The following documents are referenced in this document:

[ATC] Air Traffic Control Specification, *http://www.omg.org/technology/documents/formal/air_traffic_control.htm*

[BCQ] OMG Board of Directors Business Committee Questionnaire, *http://doc.omg.org/bc/07-08-06*

[CCM] CORBA Core Components Specification, *http://www.omg.org/technology/documents/formal/components.htm*

[CORBA] Common Object Request Broker Architecture (CORBA/IIOP), *http://www.omg.org/technology/documents/formal/corba_iiop.htm*

[CSIV2] [CORBA] Chapter 26

[CWM] Common Warehouse Metamodel Specification, *http://www.omg.org/technology/documents/formal/cwm.htm*

[DAIS] Data Acquisition from Industrial Systems, *http://www.omg.org/technology/documents/formal/dais.htm*

[EDOC] UML Profile for EDOC Specification, *http://www.omg.org/techprocess/meetings/schedule/UML_Profile_for_EDOC_FTF.html*

[EJB] "Enterprise JavaBeans™", *http://java.sun.com/products/ejb/docs.html*

[FORMS] "ISO PAS Compatible Submission Template". *http://www.omg.org/cgi-bin/doc?pas/2003-08-02*

[GE] Gene Expression, *http://www.omg.org/technology/documents/formal/gene_expression.htm*

[GLS] General Ledger Specification , *http://www.omg.org/technology/documents/formal/gen_ledger.htm*

[Guide] The OMG Hitchhiker's Guide,, *http://www.omg.org/cgi-bin/doc?hh*

[IDL] ISO/IEC 14750 also see [CORBA] Chapter 3.

[IDLC++] IDL to C++ Language Mapping, *http://www.omg.org/technology/documents/formal/c++.htm*

[Inventory] Inventory of Files for a Submission/Revision/Finalization, *http://doc.omg.org/smsc/2007-09-05*

[MDAa] OMG Architecture Board, "Model Driven Architecture - A Technical Perspective*"*, *http://www.omg.org/mda/papers.htm*

[MDAb] "Developing in OMG's Model Driven Architecture (MDA)," *http://www.omg.org/docs/omg/01-12-01.pdf*

[MDAc] "MDA Guide" (*http://www.omg.org/docs/omg/03-06-01.pdf*)

[MDAd] "MDA "The Architecture of Choice for a Changing World™"", *http://www.omg.org/mda*

[MOF] Meta Object Facility Specification, *http://www.omg.org/technology/documents/formal/mof.htm*

[MQS] "MQSeries Primer", *http://www.redbooks.ibm.com/redpapers/pdfs/redp0021.pdf*

[NS] Naming Service, *http://www.omg.org/technology/documents/formal/naming_service.htm*

[OMA] "Object Management Architecture™", *http://www.omg.org/oma/*

[OTS] Transaction Service, *http://www.omg.org/technology/documents/formal/transaction_service.htm*

[P&P] Policies and Procedures of the OMG Technical Process, *http://www.omg.org/cgi-bin/doc?pp*

[PIDS] Personal Identification Service, *http://www.omg.org/technology/documents/formal/person_identification_service.htm*

[RAD] Resource Access Decision Facility, *http://www.omg.org/technology/documents/formal/resource_access_decision.htm*

[RFC2119] IETF Best Practices: Key words for use in RFCs to Indicate Requirement Levels, (*http://www.ietf.org/rfc/rfc2119.txt)*.

[RM-ODP] ISO/IEC 10746

[SCA] Software Communications Architecture (SCA), http://sca.jpeojtrs.mil/sca.asp

[SDRP] Software Radio Components (SDRP), http://www.omg.org/spec/SDRP/

[SEC] CORBA Security Service, *http://www.omg.org/technology/documents/formal/security_service.htm*

[TOS] Trading Object Service, *http://www.omg.org/technology/documents/formal/trading_object_service.htm*

[UML] Unified Modeling Language Specification, *http://www.omg.org/technology/documents/formal/uml.htm*

[UMLC] UML Profile for CORBA, *http://www.omg.org/technology/documents/formal/profile_corba.htm*

[XMI] XML Metadata Interchange Specification, *http://www.omg.org/technology/documents/formal/xmi.htm*

[XML/Value] XML Value Type Specification, *http://www.omg.org/technology/documents/formal/xmlvalue.htm*

B.2 General Glossary

Architecture Board (AB) - The OMG plenary that is responsible for ensuring the technical merit and MDA-compliance of RFPs and their submissions.

Board of Directors (BoD) - The OMG body that is responsible for adopting technology.

Common Object Request Broker Architecture (CORBA) - An OMG distributed computing platform specification that is independent of implementation languages.

Common Warehouse Metamodel (CWM) - An OMG specification for data repository integration.

CORBA Component Model (CCM) - An OMG specification for an implementation language independent distributed component model.

Interface Definition Language (IDL) - An OMG and ISO standard language for specifying interfaces and associated data structures.

Letter of Intent (LOI) - A letter submitted to the OMG BoD's Business Committee signed by an officer of an organization signifying its intent to respond to the RFP and confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements.

Mapping - Specification of a mechanism for transforming the elements of a model conforming to a particular metamodel into elements of another model that conforms to another (possibly the same) metamodel.

Metadata - Data that represents models. For example, a UML model; a CORBA object model expressed in IDL; and a relational database schema expressed using CWM.

Metamodel - A model of models.

Meta Object Facility (MOF) - An OMG standard, closely related to UML, that enables metadata management and language definition.

Model - A formal specification of the function, structure and/or behavior of an application or system.

Model Driven Architecture (MDA) - An approach to IT system specification that separates the specification of functionality from the specification of the implementation of that functionality on a specific technology platform.

Normative – Provisions that one must conform to in order to claim compliance with the standard. (as opposed to non-normative or informative which is explanatory material that is included in order to assist in understanding the standard and does not contain any provisions that must be conformed to in order to claim compliance).

Normative Reference – References that contain provisions that one must conform to in order to claim compliance with the standard that contains said normative reference.

Platform - A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented.

Platform Independent Model (PIM) - A model of a subsystem that contains no information specific to the platform, or the technology that is used to realize it.

Platform Specific Model (PSM) - A model of a subsystem that includes information about the specific technology that is used in the realization of it on a specific platform, and hence possibly contains elements that are specific to the platform.

Request for Information (RFI) - A general request to industry, academia, and any other interested parties to submit information about a particular technology area to one of the OMG's Technology Committee subgroups.

Request for Proposal (RFP) - A document requesting OMG members to submit proposals to an OMG Technology Committee. Such proposals must be received by a certain deadline and are evaluated by the issuing Task Force.

Task Force (TF) - The OMG Technology Committee subgroup responsible for issuing a RFP and evaluating submission(s).

Technology Committee (TC) - The body responsible for recommending technologies for adoption to the BoD. There are two TCs in OMG – the *Platform TC* (PTC) focuses on IT and modeling infrastructure related standards; while the *Domain TC* (DTC) focuses on domain specific standards.

Unified Modeling Language (UML) - An OMG standard language for specifying the structure and behavior of systems. The standard defines an abstract syntax and a graphical concrete syntax.

UML Profile - A standardized set of extensions and constraints that tailors UML to particular use.

XML Metadata Interchange (XMI) - An OMG standard that facilitates interchange of models via XML documents.

Dynamic Deployment and Configuration for Robotic Technology Component (DDC4RTC) RFP

document number:
mars/2010-06-17: errata
mars/2010-06-18: DDC4RTC RFP with change bar
mars/2010-06-16: DDC4RTC RFP without change bar
presentation:
mars/2010-06-19

Noriaki Ando
Infrastructure WG, Robotics DTF
National Institute of Advanced Industrial Science and Technology

Changes form mars/10-06-05

- Based on the comments from AB, LOI date has been updated from August 31st to September 17th
- According to the WG discussion with Vanderbilt University, Thales, Zeligsoft, ETRI and AIST, the title of RFP has been changed to “Dynamic Deployment and Configuration for Robotic Technology Component (DDC4RTC) RFP.”
- Some mistakes in specification references have been corrected.

Discussion

- Two organizations are interested in the RTC-DDC RFP
  - – Remedy IT & Vanderbilt University
  - – Thales
- Monday meeting
  - – The idea of new generic dynamic D&C architecture has been proposed from Vanderbilt University
- Tuesday meeting
  - – Future plans have been discussed with Vanderbilt University, Thales, Zeilogsoft, ETRI and AIST

Point of Agreement

- We changed the name of the proposal
  - – DDC4RTC
    - Dynamic Deployment and Configuration for Robotic Technology Component
- RFP process
  - – DDC4RTC and generic dynamic D&C will be done in parallel
    - Some people from RemedyIT, Vanderbilt Univ., Zeligsoft and Thales will join the DDC4RTC RFP process and give some comments and their ideas in order to make some consistency
    - RemedyIT, Vanderbilt Univ., Zeligsoft and Thales will start a new working group in MARS to make new generic dynamic D&C standard

		Robotics Domain Task Force Preliminary Agenda ver0.0.1				**robotics/2010-06-30**
		OMG Technical Meeting - Cambridge, MA, USA -- September 20-24, 2010				
		TF/SIG		**http://robotics.omg.org/**		
		Host	**Joint (Invited)**	**Agenda Item**	**Purpose**	**Room**
Tuesday: Robotics-DTF Plenary						
8:40	9:00	Robotics		Robotics-DTF Plenary Opening Session	Robotics plenary openning	
9:00	10:00	Robotics		**Special Talk:** - TBA	presentation and discussion	
				Break (30min)		
10:30	11:30	Robotics		**Special Talk:** - TBA	presentation and discussion	
11:30	12:00	Robotics		Robotics-DTF Plenary Wrap-up Session (Roadmap and Next meeting Agenda)	Robotics plenary closing	
12:00	13:00	**LUNCH**				
13:00	16:00			Joint Dynamic Deployment and Configuration / Infrastructure WG -	RFP drafting	
16:00	17:00	MARS		Dynamic Deployment and Configuration WG (Time is tentative) - Wiiliam Otte (Vanderbuilt Univ.)? and Olivier Hachet (THALES)?	presentation and discussion	
Wednesday WG activitiy follow-up						
9:00	12:00			Robotics WG activity follow-up	discussion	
12:00	14:00	**LUNCH and OMG Plenary**				
18:00	20:00	**OMG Reception**				
Thursday Robotics-DTF Plenary						
12:00	13:00	**LUNCH**				
13:00	18:00			Architecture Board Plenary		
Friday						
8:30	12:00			AB, DTC, PTC		
12:00	13:00	**LUNCH**				
Other Meetings of Interest						
Monday						
8:00	8:45	OMG		New Attendee Orientation		
18:00	19:00	OMG		New Attendee Reception (by invitation only)		
Tuesday						
7:30	9:00	OMG		Liaison ABSC		
Please get the up-to-date version from http://staff.aist.go.jp/t.kotoku/omg/RoboticsAgenda.pdf						

robotics/2010-06-31

Robotics-DTF

Date: Friday, 25th June, 2010
Chair: T. Kotoku, Y. –J. Cho and L. Rioux
URL: http://robotics.omg.org/
email: robotics@omg.org

➢Highlights from this Meeting:

Robotics Plenary: (17 participants)

- Joint 3rd RFP Draft Review with MARS
  - Dynamic Deployment and Configuration for Robotic Technology Component (DDC4RTC) RFP
    [mars/2010-05-07, 06-16, -17, -18]
- 2nd RFP Draft Review
  - Robotic Interaction Service (RoIS) Framework RFP
    [robotics/2010-05-01, 06-23,-24, -25]
- Charter New Working Group
  - Modelling for Robotics [robotics/2010-06-12]
- 3 WG Reports [robotics/2010-06-18,-19,-20]

Robotics-DTF

Date: Friday, 25th June, 2010
Chair: T. Kotoku, Y. –J. Cho and L. Rioux
URL: http://robotics.omg.org/
email: robotics@omg.org

➢Deliverables from this Meeting:

- Dynamic Deployment and Configuration for RTC (DDC4RTC) RFP thru MARS-PTF
- Robotic Interaction Service (RoIS) Framework RFP

➢Future deliverables (In-Process):

- Models for Robotics RFI (will be issued in Santa Clara)
- JAUS-RTC White Paper (will be issued in Santa Clara)
  Draft : [robotics/2010-06-11]

➢Next Meeting (Boston, MA, USA):

- Chu-suk (Special Holidays in Korea)
- Joint Meeting with MARS DDC WG

Minutes of the Robotics DTF Meeting - DRAFT

June 21-25, 2010

Minneapolis, MN, USA

(robotics/2010-06-32)

Meeting Highlights

- The Dynamic Deployment and Configuration for Robotic Technology Component (DDC4RTC) RFP was recommended to issue in MARS-PTF and AB, and approved in the Platform TC.
- The Robotic Interaction Service (RoIS) Framework RFP was recommended to issue in Robotics-DTF and AB, and approved in the Domain TC.
- The Models for Robotics RFI was reviewed, and will be issued in the Santa Clara meeting.
- New Modelling for Robotics WG was chartered.
- We have two of new volunteers for the new WG Co-Chair.

List of Generated Documents

robotics/2010-06-01 Final Agenda (Tetsuo Kotoku)
robotics/2010-06-02 Jacksonville Meeting Minutes [approved] (Yoshihiro Nakabo and Jae-Yeong Lee)
robotics/2010-06-03 Opening Presentation (Tetsuo Kotoku)
robotics/2010-06-04 Revised dradt of Robotic Interaction Service (RoIS) Framework RFP (Toshio Hori)
robotics/2010-06-05 MARS Presentation of Dynamic Deployment and Configuration (DDC) for RTC RFP (copy of mars/2010-06-03) (Seung-Woog Jung)
robotics/2010-06-06 Errata of Dynamic Deployment and Configuration (DDC) for RTC RFP (copy of mars/2010-06-04) (Noriaki Ando)
robotics/2010-06-07 Revised draft of Dynamic Deployment and Configuration (DDC) for RTC RFP (copy of mars/2010-06-05) (Noriaki Ando)
robotics/2010-06-08 Future of the Deployment and Configuration Specification (William R. Otte)
robotics/2010-06-09 RTC Dynamic Deployment and Configuration (DDC) Specification (Seung-Woog Jung)
robotics/2010-06-10 2nd revised draft of Robotic Interaction Service (RoIS) Framework RFP (Toshio Hori)
robotics/2010-06-11 Draft JAUS-RTC White Paper (Laurent Rioux)
robotics/2010-06-12 Charter of the Modelling for Robotics Working Group (Laurent Rioux)
robotics/2010-06-13 Draft RFI on Models for Robotics (Laurent Rioux)
robotics/2010-06-14 Presentation of Robotic Interaction Service (RoIS) Framework RFP (Miki Sato)
robotics/2010-06-15 AB Reviewer's Comments for RoIS (Miki Sato)
robotics/2010-06-16 3rd revised draft of Robotic Interaction Service (RoIS) Framework RFP (Miki Sato)
robotics/2010-06-17 Wrap-up Presentation (Tetsuo Kotoku)
robotics/2010-06-18 Infrastructure WG Progress Report (Noriaki Ando)
robotics/2010-06-19 Robotic Functional Services WG Report (Toshio Hori)
robotics/2010-06-20 Modelling WG Report (Laurent Rioux)
robotics/2010-06-21 Roadmap for Robotics Activities (Tetsuo Kotoku)
robotics/2010-06-22 4th Draft of Robotic Interaction Service (RoIS) Framework RFP (Miki Sato)
robotics/2010-06-23 Convenience Document of Robotic Interaction Service (RoIS) Framework RFP with change bars (Toshio Hori)
robotics/2010-06-24 Errata of RoIS RFP (Toshio Hori)
robotics/2010-06-25 Robotic Interaction Service (RoIS) Framework RFP (Toshio Hori)
robotics/2010-06-26 Dynamic Deployment and Configuration for Robot Technology Component (DDC4RTC) RFP (copy of mars/2010-06-16) (Noriaki Ando)
robotics/2010-06-27 Errata (copy of mars/2010-06-17) (Noriaki Ando)
robotics/2010-06-28 Convenient Document of Dynamic Deployment and Configuration for Robot Technology Component (DDC4RTC) RFP with change bars (copy of mars/2010-06-18) (Noriaki Ando)
robotics/2010-06-29 Presentation of Dynamic Deployment and Configuration for Robot Technology Component (DDC4RTC) RFP (copy of mars/2010-06-19) (Noriaki Ando)
robotics/2010-06-30 Next Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)
robotics/2010-06-31 DTC Report Presentation (Tetsuo Kotoku)
robotics/2010-06-32 Minneapolis Meeting Minutes - DRAFT (Toshio Hori and Myung-Eun Kim)

Minutes

Monday, Jun 21, 2010, Lake Superior B, 5th FL
Robotics DTF Plenary Meeting

8:30-9:00 Robotics DTF Opening Session, Chair: Dr. Kotoku, Quorums: 3
AIST, JARA, Shibaura IT, Technologic Arts, ETRI

- Minutes takers: Toshio Hori (AIST) and Myung-Eun Kim (ETRI)
- Jacksonville Meeting Summary
  . Robotics Plenary: (21 participants)
  . Joint 2nd RFP Draft Review with MARS: DDC RFP (mars/2010-03-05,-06,-07,-21)
  . 1st RFP Draft Review: RoIS RFP (robotics/2010-03-18,-22)
  . Special Session on JAUS and RTC
- Agenda Review
  . Monday:
    RTC-DDC RFP 3rd Review with MARS/ RoIS RFP 2nd Review
  . Tuesday:
    Robotics-DTF Plenary/ WG and Contact Report, Wrap-up
  . Thursday:
    Joint Plenary with MARS/ Voting of DDC RFP/ Voting of RoIS RFP
- New WG needs for volunteers
  . Modeling in Robotics WG (potential co-Chair: Laurent Rioux, Takeshi Sakamoto)
- AB schedule (Mon.)
  . RTC-DDC RFP: 15:20-15:40
  . RoIS Framework RFP: 15:40-16:00

10:00-10:30 Joint Plenary with MARS, Lake Superior A, 5th FL
- RTC DDC RFP 3rd review, Seung-Woog Jung, ETRI, Korea
- Revised the draft of RFP according to the comments of AB members
- Discussion with Vanderbilt Univ. and Thales will be on Monday and Tuesday.

10:30-11:00 Special Talk: William Otte (Vanderbilt Univ.)
- Title: Future of the Deployment and Configuration Specification
- The limitation of the existing DEPL specification
- Introduction Locality Manager for various running environments and dynamic deployment

11:00-12:00 Robotic Interactive Service (RoIS) Framework 2nd Review, Miki Sato (ATR)
- Revised the draft of RFP according to the comments of AB members
- Discussion the relationship of the existing OMG specification such as CORBA DDS

15:20-16:00 Architecture Board Plenary, Olsen, 3rd FL
- Presentation of RoIS RFP, Toshio Hori, AIST, Japan
  . The revised RFP should be submitted by Thursday.
- Presentation of DDC RFP, Seung-Woog Jung, ETRI, Korea
  . If the RFP will be approved in MARS, there is no problem in AB voting on Thursday

Tuesday, Jun 22, 2010, Lake Calhoun, 5th FL
Robotics DTF Plenary Meeting

15:30-15:50 Modeling for Robotics WG, Laurent Rioux (Thales)
- Presentation of DRAFT Charter including background, objectives, and chairs
- Co-Chairs: Takeshi Sakamoto (Technologic Arts), Laurent Rioux (THALES), Toby McClean (Zeligsoft)
- Making a vote for this charter
  . Approved: Thales (Motion), Shibaura IT(Second), AIST(White ballot)
- Making a vote for resolution of Robotic Data and Profiles WG and creation of Modeling in Robotics WG
  . Approved: AIST (Motion), Shibaura IT(Second), Thales(White ballot)

- White Paper RTC-JAUS review: Final version in July
- Draft RFI <<Models for robotics >>: Plan to be issues in December 10.
- Introduction of RoSym Workshop
  . 1st international workshop on MBE for robotics: Deadline 26 July 2010
  . http://www.artist-embedded.org/artist/RoSym-2010

15:50-16:10 Infrastructure WG, Noriaki Ando (AIST)
- Revised the RFP according to AB member's comments
- Two organizations are interested in the RTC DDC RFP: Remedy IT and Vanderbilt Univ, Thales
- William Otte presented about new generic dynamic D&C architecture
- Discussed the future plan of RTC DDC RFP with Laurent Rioux(Thales), William Otte (Vanderbilt Univ.), and Toby McClean (Zeligsoft)
- Changed the title of the RFP: DDC4RTC
- DDC4RTC and generic dynamic D&C will be done in parallel
  . Remedy IT, Vanderbilt Univ., Zeligsoft and Thales will start a new working group in MARS
  . Some people will join to making the DDC4RTC and give some comments for consistency
- MARS meeting for review and voting and AB Plenary will be on Thursday
- TC voting will be on Friday

16:10-16:30 Functional Service WG, Toshio Hori (AIST)
- Status of RoIS Framework RFP draft
  . Initial draft was submitted 4W before the meeting
  . Revised the RFP according to AB members' comments
  . 2nd review of draft in Robotics DTF was on Monday and we got some comments from AB plenary
  . Review and voting on the draft will be on Thursday
  . AB plenary will be on Thursday
  . TC approval will be on Friday

16:30-17:00 Wrap-up Session, Chair: Dr. Kotoku (AIST)
- The Schedule of Functional Service WG and Infrastructure WG in next meeting will depends on the result of this voting
- The Modeling for Robotics WG won't have a schedule in next meeting

Thursday, Jun 24, 2010, Lake of the Isles, 5th FL
Robotics DTF Plenary Meeting

9:00-9:45 Joint Plenary with MARS, Lake Superior A, 5th FL
- Presentation of RTC4DDC RFP, Noriaki Ando, AIST, Japan
- RTC4DDC RFP is approved
  . AIST(Motion), Thales(Second), Technologies Art(White ballot)

10:00-10:25 Robotic Interaction Service(RoIS) Framework RFP 2nd Review, Miki Sato (ATR)
- Presentation of the revised RoIS RFP
- RoIS RFP is approved
  . JARA(Motion), ETRI(Second), Technologies Art(White ballot)

10:25-10:30 Review Final Minutes of Jacksonville Meeting, Tetsuo Kotoku (AIST)
- The Minutes of Jacksonville meeting is approved
  . AIST(Motion), ETRI(Second), Technologies Art(White ballot)

Adjourned at 10:30

ATTENDEE (17)

- Geoffrey Biggs (AIST)
- Hugues VINCENT (Thales)
- Jae-yeon Lee (ETRI)
- Laurent Rioux (Thales)
- Makoto Mizukawa (Shibaura Inst. of Tech.)
- Miki Sato (ATR)
- Miwako Doi (Toshiba)
- Myung-Eun Kim (ETRI)
- Noriaki Ando (AIST)
- Seung-Woog Jung (ETRI)
- Su-Young Chi (ETRI)
- Takeshi Sakamoto (Technologic Arts)
- Tetsuo Kotoku (AIST)
- Toby McClean (Zeligsoft)
- Toshio Hori (JARA/AIST)
- William Otte (Vanderbilt Univ.)
- Yun Koo Chung (ETRI)

Prepared and submitted by Toshio Hori (AIST) and Myung-Eun Kim (ETRI)