Robotics Domain Task Force Final Agenda ver1.0.2 **robotics/2010-03-01**

OMG Technical Meeting - Jacksonville, FL, USA -- March 22-26, 2010

http://robotics.omg.org/

		TF/SIG				
		Host	Joint (Invited)	Agenda Item	Purpose	Room
Monday: Robotics Plenary(am) and WG activites(pm)						
9:00	10:00			Robotics-DTF Plenary Opening Session	Robotics plenary openning	City Terrace 4, 3rd FL
10:00	11:00	MARS		RTC Deployment and Dynamic Reconfiguration RFP 2nd Review - Noriaki Ando(AIST) and Beom-Su Seo (ETRI)	presentation, discussion and Voting	City Terrace 12, 3rd FL
11:00	12:00	Robotics		Thales's Implementation of RTC and Introduction to Models in Robotics (tentative) - Laurent Rioux (Thales)	presentation and discussion	City Terrace 4, 3rd FL
12:00	13:00					River Terrace 1, 3rd FL
13:00	18:00			Architecture Board Plenary		**River Terrace 2, 3rd FL**
13:00	17:00			Robotic Infrastructure WG (4h) - Noriaki Ando and Beom-Su Seo	discussion	City Terrace 4, 3rd FL
				Robotic Functional Services WG(4h): - Su-Young Chi (ETRI) and Toshio Hori (AIST)	discussion	Boardroom 3, 3rd FL
Tuesday: WG activities (am) and Robotics Plenary (pm)						
9:00	12:00			Robotic Infrastructure WG (3h) - Noriaki Ando and Beom-Su Seo	discussion	Boardroom2, 3rd FL
				Robotic Functional Services WG(3h): - Su-Young Chi and Toshio Hori	discussion	Daytona, 3rd FL
12:00	13:00			**LUNCH**		River Terrace 1, 3rd FL
13:00	16:00	Robotics		**Special Session: JAUS and RTC (3h)** - Laurent Rioux (Thales) RTC introduction / demonstration SAE JAUS introduction & Update THALES RTC middleware: Presentation & Demo: THALES RTC & JAUS: 2 complementary standards: Discussions / questions:	presentation and discussion	Daytona, 3rd FL
16:00	16:45	Robotics		Robot Interaction Service (RoIS) Framework RFP - Su-Young Chi,	1st Review of RFP draft	
16:45	17:00	Robotics		Robotics-DTF Plenary Wrap-up Session (WG report, Contact Reports, Roadmap and Next meeting Agenda)	Robotics plenary closing	
17:00				Adjourn plenary meeting		
Wednesday WG activity follow-up						
9:00	12:00			Robotic Infrastructure WG (3h) - Noriaki Ando and Seung-Woog Jung (ETRI)	discussion	Daytona, 3rd FL
				Robotic Functional Services WG(3h): - Su-Young Chi, Toshio Hori and Miki Sato (ATR)	discussion	City Terrace 9, 3rd FL
12:00	14:00			**LUNCH and OMG Plenary**		River Terrace 1, 3rd FL
14:00	17:00			Models in Robotics (3h) - Laurent Rioux	discussion	Daytona, 3rd FL
				Robotic Functional Services WG(3h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	City Terrace 9, 3rd FL
				Robotic Localization Service-RTF(3h): - Shuichi Nishio	discussion	Boardroom 4, 3rd FL
18:00	20:00			**OMG Reception**		River Terrace1, 3rd FL
Thursday WG activity follow-up						
9:00	9:30	MARS		Robotics-DTF Joint Plenary with MARS Voting of Dynamic Deployment and Configuration (DDC) RFP	Vote to Issue	City Terrace 12, 3rd FL
9:30	12:00			WG activity follow-up(2.5h)	discussion	Boardroom 4, 3rd FL
12:00	13:00			**LUNCH**		River Terrace 1, 3rd FL
13:00	18:00			Architecture Board Plenary		River Terrace 2, 3rd FL
Friday						
8:30	12:00			AB, DTC, PTC		River Terrace 1, 3rd FL
12:00	13:00			**LUNCH**		River Terrace 3, 3rd FL

Other Meetings of Interest

Monday						
8:00	8:45	OMG		New Attendee Orientation		City Terrace 9, 3rd FL
9:00	12:00	OMG		Tutorial - Introduction to OMG's meeting and Middlewere Specifications		City Terrace 9, 3rd FL
18:00	19:00	OMG		New Attendee Reception (by invitation only)		Room 4104, 4th FL
Tuesday						
7:30	9:00	OMG		Liaison ABSC		City Terrace 10, 3rd FL
9:00	17:00	OMG		BPM/SOA Symposium		River Terrace 2, 3rd FL
9:00	17:00	SE		System Engineering DSIG		City Terrace 9, 3rd FL
17:00	18:00	OMG		RTF-FTF Chair's Workshop		Boardroom 3, 3rd FL
Wednesday						
9:00	17:00	OMG		BPM/SOA Symposium		River Terrace 2, 3rd FL
9:00	12:00	ECM		Emergency, Crisis & Major Event Management DSIG		City Terrace 4, 3rd FL
9:00	12:00	MF		Mathematical Formalization DSIG		Boardroom 4, 3rd FL
9:00	16:30	SysA		SysA-PTF Meeting		Boardroom 3, 3rd FL
Thursday						
8:00	12:00	OMG		Archtecture Ecosystem ABSIG		City Terrace 4, 3rd FL
9:00	17:00	SE		System Engineering DSIG - SysML-Modelica WG		City Terrace 9, 3rd FL
9:00	12:00	Ontology		Ontology PSIG		City Terrace 6, 3rd FL
9:00	12:00	SysA		SysA-PTF Meeting		Boardroom 3, 3rd FL
16:00	17:00	MARS		MARS Agenda Coordination		City Terrace 12, 3rd 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

December 7-11, 2009

Long Beach, CA, USA

(robotics/2010-03-02)

Meeting Highlights

- The 1st draft of "RTC deployment and Dynamic Reconfiguration RFP" was reviewed at the Joint Plenary with MARS and at the upcoming Long Beach Meeting. [robotics/2009-12-09]
- As a potential new work item, New Work Item Talks "Behavioral states and instructions for lifes tyle support service" was presented by Dr. Miwako Doi (Toshiba) [robotics/2009-12-03]
- As a special talk, "Introduction to DDR(Data Distribution Service" was presented by Dr. Rick W arren (RTI) [robotics/2009-12-13]
- Liaison between OMG and TC211 underway [robotics/2009-12-17]

List of Generated Documents

robotics/2009-12-01 Final Agenda (Tetsuo Kotoku)
robotics/2009-12-02 San Antonio Meeting Minutes [approved]
robotics/2009-12-03 Behavioral States and Instructions for Lifestyle Support Service (Miwako Doi)
robotics/2009-12-04 Steering Committee Presentation (Tetsuo Kotoku)
robotics/2009-12-05 Roadmap for Robotics Activities (Tetsuo Kotoku)
robotics/2009-12-06 High Level Task Description for Robotics (Rockwon Kim)
robotics/2009-12-07 Binding Symbols/Functions/Actions in World Model to APOIs (Rockwon Kim)
robotics/2009-12-08 OPRoS Component Tools = Snapshots = (Seung Woog Jung)
robotics/2009-12-09 RTC Deployment and Dynamic Reconfiguration RFP Draft [for 1st review revised version] (Noriaki Ando)
robotics/2009-12-10 OMG User Identification Service Interface (Su-Young Chi and Jaeyeon Lee)
robotics/2009-12-11 Scope of Standardization for UIS (Su-Young Chi and Jaeyeon Lee)
robotics/2009-12-12 Case Stady: UIS (JaeYeon Lee)
robotics/2009-12-13 Introduction to DDS (Rick Warren)
robotics/2009-12-14 Opening Presentation (Tetsuo Kotoku)
robotics/2009-12-15 Introduction of ISO/IEC JTC1/SC24 (Yun Koo Chung)
robotics/2009-12-16 RTC Deployment and Dynamic Reconfiguration (DDR) 1st draft [mars/2009-12-15] (Noriaki Ando)
robotics/2009-12-17 Contact Report: Status of Robotic Localization Service (RLS) in ISO/TC211 (Shuic hi Nishio)
robotics/2009-12-18 Contact Report: China/Korea/Japan Workshop (Young-Jo Cho)
robotics/2009-12-19 Contact Report: ISO/TC184/SC2 (Tetsuo Kotoku)
robotics/2009-12-20 Robotic Functional Services WG Meeting Report (Su-Young Chi)
robotics/2009-12-21 Infrastructure WG Progress Report (Noriaki Ando)
robotics/2009-12-22 Wrap-up Presentation (Tetsuo Kotoku)
robotics/2009-12-23 Next Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)
robotics/2009-12-24 DTC Report Presentation (Young-Jo Cho)
robotics/2009-12-25 Charter for Robotics Localization Service (RLS) RTF (Shuichi Nishio)
robotics/2009-12-26 Long Beach Meeting Minutes - DRAFT (Geoffrey Biggs and Rockwon Kim)

Minutes
Wednesday, December 10, 2009, Regency C, 3rd Floor
Robotics DTF Plenary Meeting,

14:00 - 14:10 Opening Session Chair: Dr Kotoku, Quorom: 4

Joined organizations: AIST, ETRI, JARA, Technologic Arts, KAR, View Five
- Minutes takers: Geoffrey Biggs (AIST), Rockwon Kim (ETRI)
- Approval of San Antonio minutes
  - Correction: Document 2009-09-14 was submitted by Dr. Hori (AIST)
  - Approved: AIST (motion), ETRI (second), JARA (white ballot)

14:10 - 15:20 Special talk: Introduction to DDS (Rick Warren, RTI)

- DDS has two parts:
  - Data Distribution Service for real-time systems.
  - Real-Time Publish-Subscribe (RTPS) Protocol (sometimes also called DDSI).
    - No other publisher-subscribe system has this sort of protocol.
- Several implementations of DDS and RTPS, including commercial and open-source.
- Used in a wide range of commercial control projects (military, driver safety systems, telescope control, MRI, PLCs, ...)
- Publish-Subscribe model, data-centric.

15:20 - 15:40 Contact report, Yun Koo Chung, ETRI

- Introduction to ISO/IEC JTC1/SC24
  - Joint Technical Committee of IEC and ISO 1: "Information Technology"
  - SC24: Computer graphics, image processing and environmental data representation.
- Yun Koo Chung was appointed as liaison to ISO/IEC JTC1/SC24 by OMG Robotics DTF.
  - Liase about standardization of imaging based applications for service robotics.
  - See document ISO/IEC JTC1 SC24N3182.
  - ETRI (motion), AIST (second), JARA (white ballot)

16:00 - 17:40 Joint Plenary with MARS

RTC Deployment and Dynamic Configuration RFP, Noriaki Ando, AIST
- Common services and interfaces for component repositories, searching, deployment, directory services, and detecting/notifying of changes in components are needed.
- Will be issued as a MARS RFP.
- Revise the RFP to call for a PSM based on the DnC PIM, extending it to add in the extra features not found in that model that are necessary for the RFP.
- Document number: MARS-2012

16:50 - 17:00 Contact report, Shuichi Nishio, JARA

Status of Robotic Localization Service at ISO/TC211
- Liaison between OMG and TC211 underway.
- Still not decided how the RLS will be standardized in TC211.
  - Fast-track or New Item Proposal.
- RLS spec under informal review at PMG.

- If "New Item", will be handled by WG10 (Ubiquitous Public Access).

17:00 - 17:08 Contact report, Young-Jo Cho, ETRI

4th China/Korea/Japan Joint Workshop on Robotics
- Service robot industrialization.
- Challenges of industrial robots.
- Discussion and exhibition of educational robots.
- Next workshop will be held in Korea.

17:08 - 17:10 Contact report, Tetsuo Kotoku, AIST

ISO/TC184/SC2 2009 Oct. Tokyo Meeting
- WG8: Setting up 3 Study Groups

17:10 - 17:20 User Identification Service WG report

- Discussed the name of the specification.
  - Voted for Robotic Interaction Service Framework (RIS) with 8 votes.
- Discussed the scope of standardization.
  - Application domain: service robot interacting with humans.
- Discussed standardization items.
- Discussed 4 steps to standardization. First is case studies.
- Will meet 19th and 20th January in Tokyo to present case studies.
- Roadmap: 1st review of RFP in March 2009, 2nd review in June 2009, submission in September 2009.

17:20 - 17:30 Infrastructure WG report

- Reviewing RTC DDR 1st draft.
  - Term consolidation and review presentation material.
- Presentation on High-Level Task Description and its Binding APIs for Robotics by Dr Rock won Kim.
- Joint plenary with MARS reviewing RFP 1st draft.
- Will submit 2nd RFP on February 22, 2010.
- Will be issued as a MARS RFP.

17:30 - 17:40 Closing presentation and next agenda by Tetsuo Kotoku

- Chair change in Infrastructure WG.
- Roadmap discussion.
  - Robotic Map Services RFP is still under discussion.
  - Behavior States and Instructions RFP was proposed.
- New member on the Contacts Sub-committee: Yun Koo Chung, ETRI
- Next meeting: March 22-26, Jacksonville, FL, USA.

Adjourned plenary meeting at 17:40

ATTENDEE (17 Participants)

- Akira Tanaka (View5)
- Chul Jong Hwang (KAR)
- Geoffrey Biggs (AIST)
- JaeYeon Lee (ETRI)
- Myung-Eun Kim (ETRI)
- Noriaki Ando (AIST)
- Rick Warren (RTI)
- Rockwon Kim (ETRI)
- Seung-Woog Jung (ETRI)
- Shuichi Nishio (JARA/ATR)
- Su-Young Chi (ETRI)
- Takashi Tsubouchi (Univ. of Tsukuba)
- Takeshi Sakamoto (Technologic Arts)
- Tetsuo Kotoku (AIST)
- Toshio Hori (AIST)
- Young-Jo Cho (ETRI)
- Yun Koo Chung (ETRI)

Prepared and submitted by Geoffrey Biggs(AIST) and Rockwon Kim (ETRI).

Robotics-DTF Plenary Meeting
Opening Session

March 22th, 2010

Jacksonville, FL, USA

Hyatt Regency Jacksonville Riverfront

Approval of Minutes

Meeting Quorum : 3

AIST, ETRI, JARA, Technologic Arts,

Minutes taker(s):
Minutes review

Yoshihiro Nakabo
Jaeyoung Lee

Long Beach Meeting Summary

Robotics Plenary: (17 participants)

–1 New Work Item Talks

- "Behavioral states and instructions for lifestyle support service", Miwako Doi, Toshiba [robotics/2009-12-03]

–1 Special Talk

- "Introduction to DDS(Data Distribution Service", Rick Warren, RTI [robotics/2009-12-13]

–2 WG Reports [robotics/2009-12-20,-21]

–4 Contact Reports [robotics/2009-12-15,-17,-18,-19]

AIST

Agenda Review

Mon:
10:00-11:00 RTC-DDR 2nd Review joint with MARS-PTF
11:00-12:00 Introduction to Models in Robotics

Tue:
13:00-16:00 Special Session on JAUS and RTC
16:00-17:00 WG and Contact Report, Wrap-up

Thu:
9:00-9:30 Joint Plenary with MARS
Voting of RTC-DDR RFP

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

RTC Deployment and Dynamic Reconfiguration (DDR) 2nd draft

document number: mars/2010-02-14
presentation: mars/2010-03-04

Noriaki Ando
Infrastructure WG, Robotics DTF
National Institute of Advanced Industrial Science and Technology

Purpose of Infra. WG

- The purpose of the Infrastructure Working Group of the Robotics Domain Task Force is *to standardize fundamental models, common facilities, and middleware to support the development and integration of a broad range of robotics applications*.

OMG RTC Specification

- Robotic Technology Component (RTC): RTC's component model provides typical functionality and services for robotic systems
  - "Robotic Technology Component Specification" [formal/2008-04-04]
- Implementations:
  - AIST: OpenRTM-aist (C++, Java, Python)
  - SEC: OpenRTM.NET (C#, VB, etc)
  - Korean National Project "OPRoS": partially complient with OMG RTC specification

Users of RTC Specification

- OpenRT Platform Project (Japan)
  - 15 consortium, more than 40 research institutes, universities and companies
  - Two missions
    - Software platform for robotic system development
    - Software component library development for service robots
- OPRoS (Open Platform for Robot Services) Project (Korea)
  - More than 25 research institutes, universities and companies
  - Software platform for robotic system development

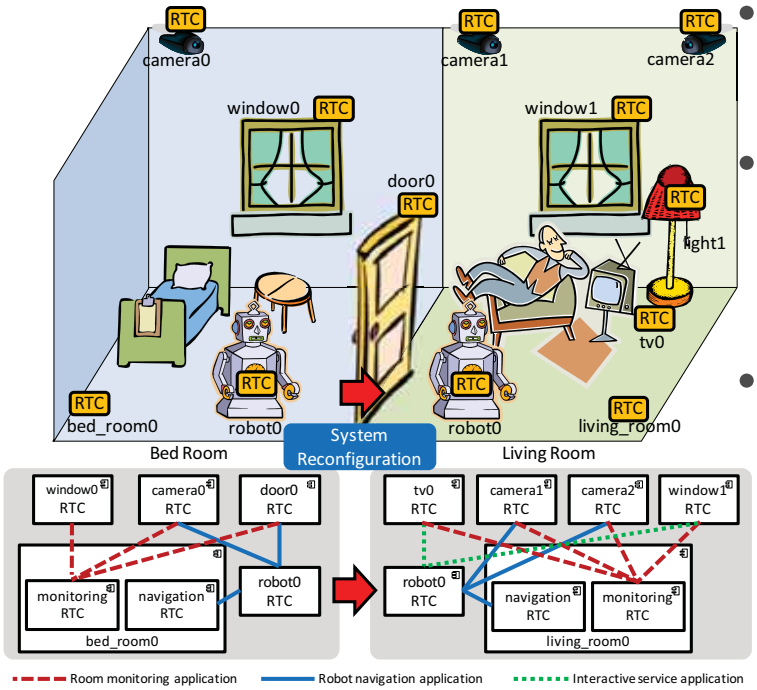
RTC Deployment and Dynamic Reconfiguration (DDR)

Motivation

- Common component repository service for RTC
  - Registering, storing, searching and downloading component
- Common component deployment interface for RTC
  - Deploying RTC on the distributed nodes at run-time
  - Configuration, making connection among RTCs
- Common directory services for RTC instances
  - Registering, searching component
- Common method for detection and a notice of change of a component
  - Notifying changing event into other RTC-based systems
  - Runtime reconfiguration based on changing event

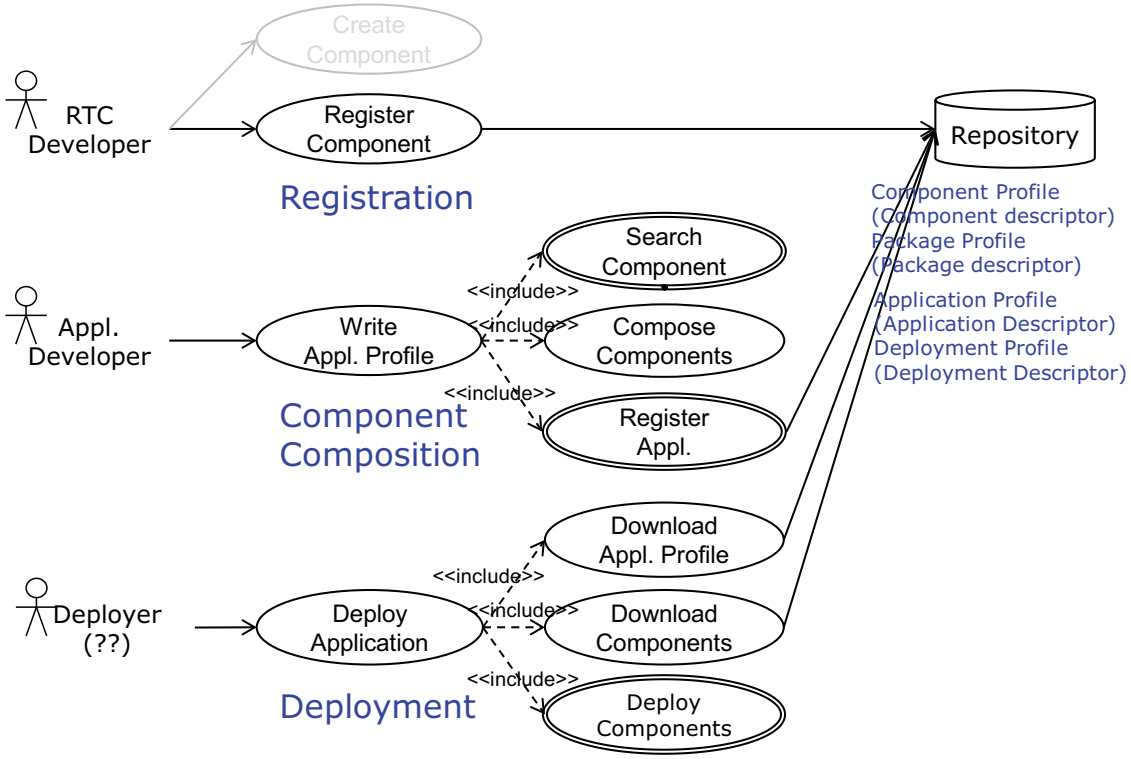
New specification defining these dynamic functionality is necessary

Assumption

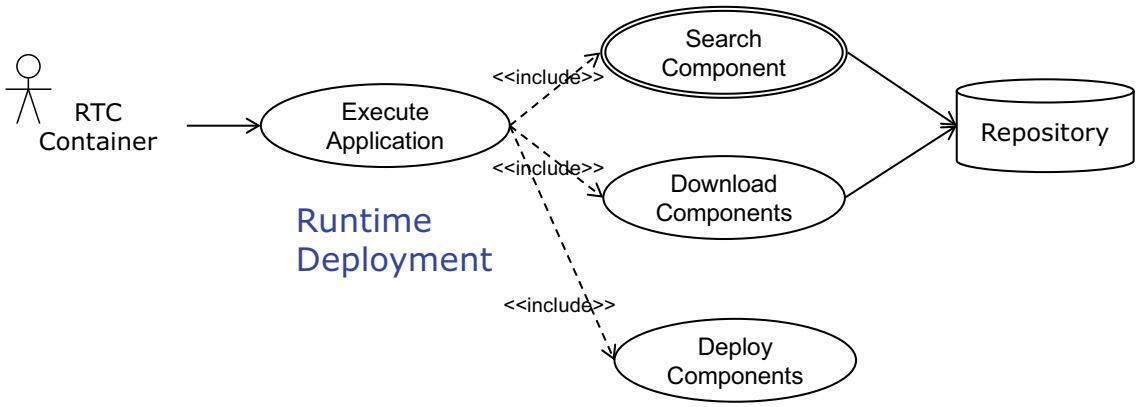


- Many RTCs are distributed spatially
- Systems would be constructed as RTCs aggregation
- System structure should be changed according to the environmental changes in run-time

Use case (1): Deployment

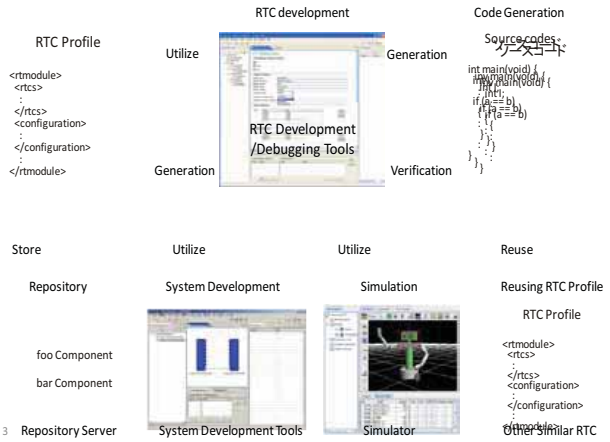


Use case (2): Deployment



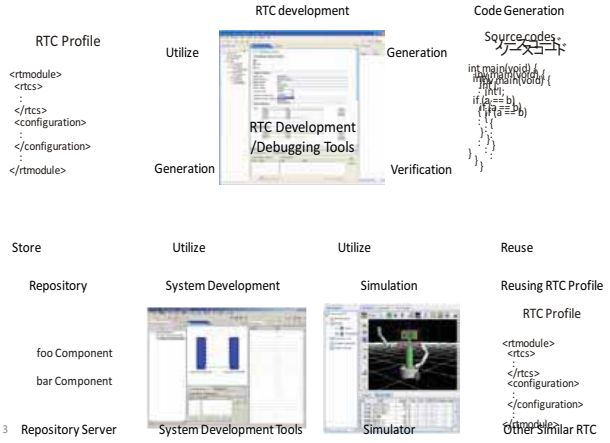
RT component profile

- Meta data structure that describe component profile
- Various usage
  - Code template generation
  - Repository database information
  - System development
  - Simulation
  - Re-use



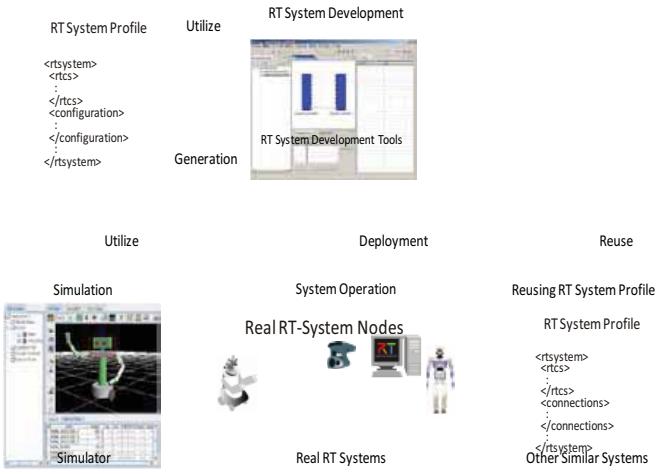
Repository Interfaces

- RTC source/binary data base
  - Registered by RTC developer
  - Searched/downloaded at system deployment time
- RTC-based system profile data base
  - Registered by system developer
  - Searched/downloaded at system deployment time



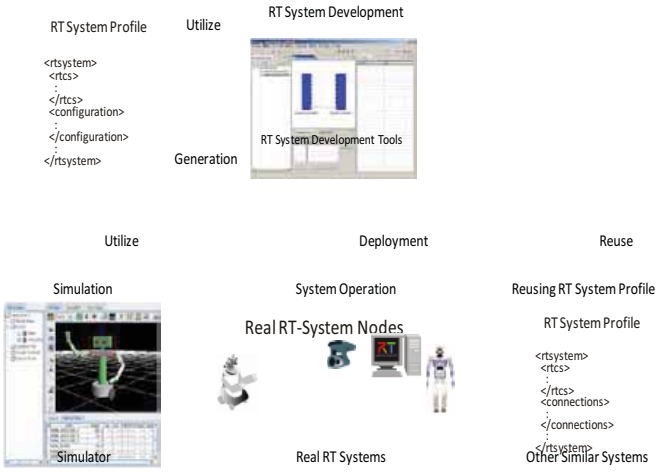
RTC-based system deployment profile

- Meta data structure that describe system structure
- Various usage
  - System design tools' data format
  - System deployment
  - Simulation
  - Re-use

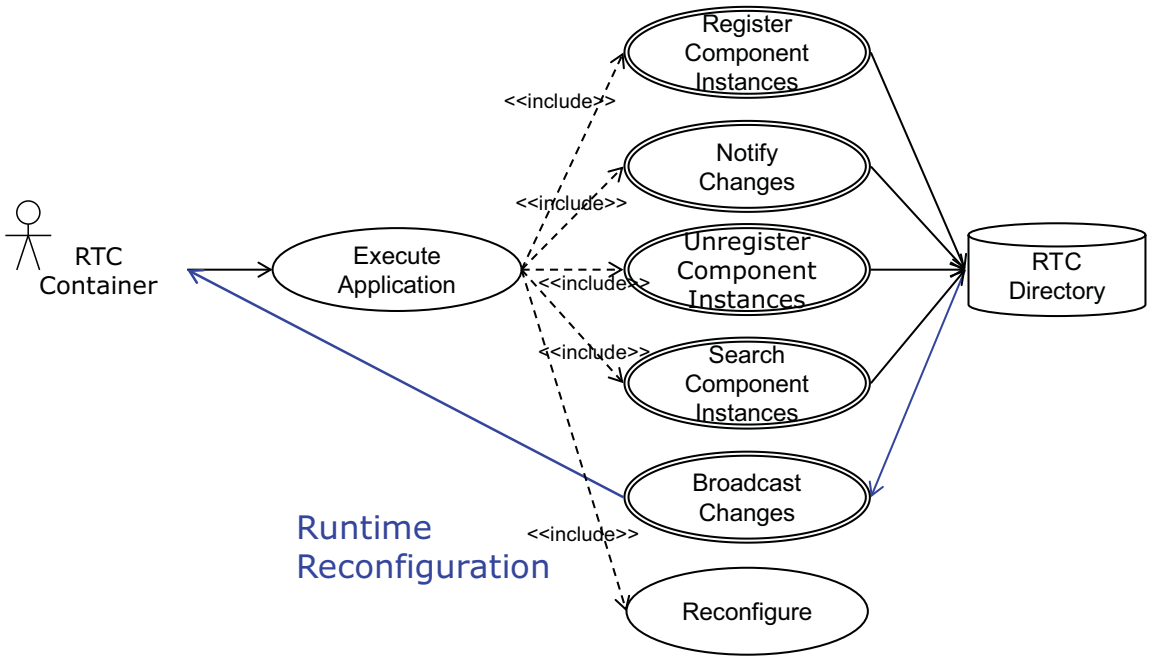


RTC-based system deployment interface

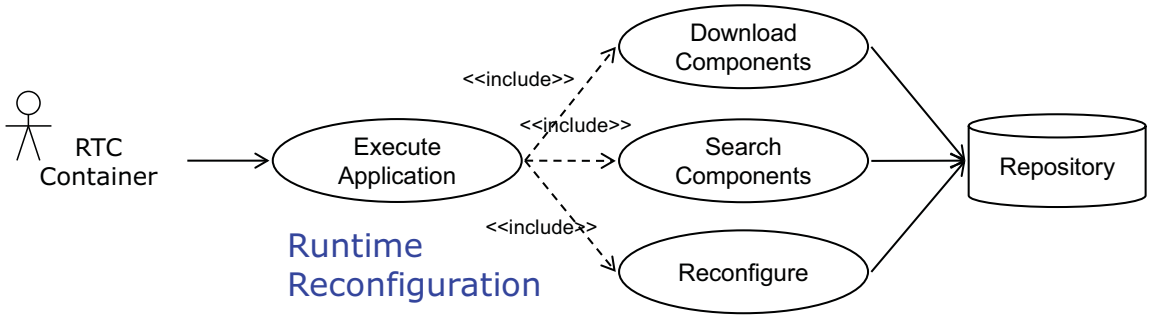
- Interfaces for RTC-based system deployment
  - It would be provided by distributed nodes
  - It manages component lifecycle including downloading, loading, creating and destroying
  - It would be used by application programs



Use case (1): Reconfiguration



Use case (1): Reconfiguratrion



RTC instance lookup

- Naming service, directory service
- It provides higher level search functionality based on component profile information

RTC instance tracking

- Tracking component internal status
- Tracking component internal parameters

What's Changed?

According to the comments from AB members….

- Notation problems
  - Abbreviations, font color, etc…
- 6.5 Mandatory Requirements
- 6.6 Optional Requirements
- 6.7 Issue to be Discussed
  - "Out of focus". No dynamic features are described.
- Schedule
  - The initial-submission was delayed by one meeting.

Mandatory Requirements (1)

- 6.5.1 Proposals shall specify a meta-model for the description of component meta-information necessary to support automatic searching and comparing of RT components in component repositories and in the run-time system.
- 6.5.2 Proposals shall specify a meta-model for the description of RTC interfaces, their compatibility criteria, and deployment requirements.

Mandatory Requirements (2)

- 6.5.3 Proposal shall specify a platform independent model for information service to identify to locate deployed RTCs available for utilization by the requesting robot.
- 6.5.4 Proposals shall specify a data-model for a component information registry using meta-model requested in requirement 6.5.1 and 6.5.2. Proposal shall also specify query mechanism for this repository.

Mandatory Requirements (3)

- 6.5.5 Proposals shall specify a platform independent model for dynamic RTC configuration and deployment, which
  - allows an efficient configuration of RTCs.
  - initiates reconfiguration based on external and/or internal events. A capability for event filtering shall be provided.
  - supports coordinated reconfiguration of multiple robot systems to support performance of coordinated tasks.
  - defines a service interface for the deployment process.
- 6.5.6 Proposals shall reuse or extend the deployment architecture as defined Deployment and Configuration of Component-based Distributed Applications Specification[D&C].

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 6th, 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*
LOI to submit to RFP due	*August 31, 2010*

GOSTAI Robotics for everyone

THALES

A new middleware for unmanned systems.

L. Rioux

Research & Technology

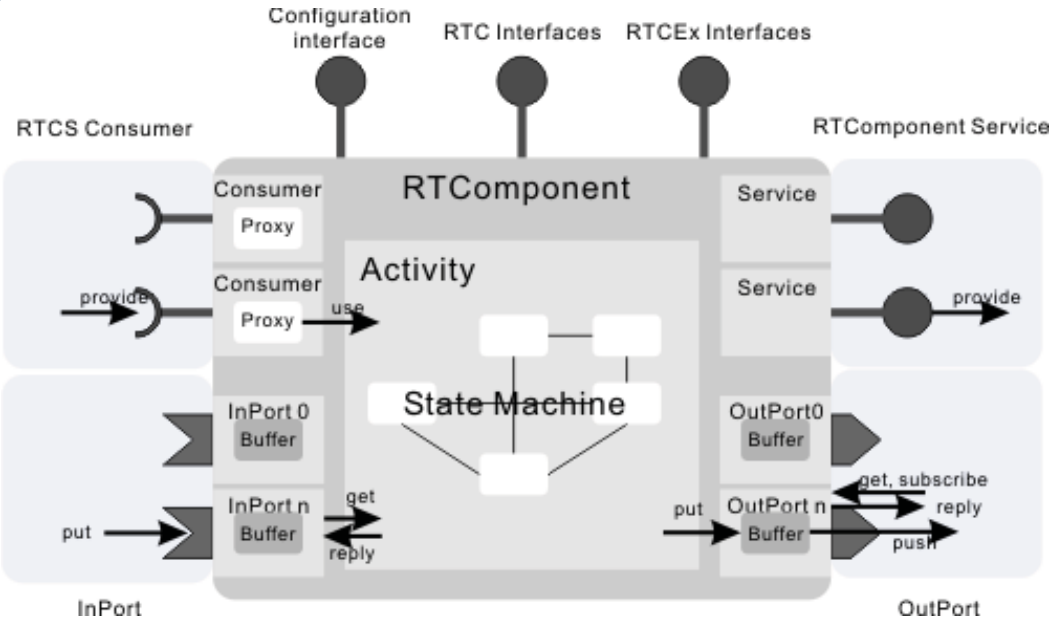
Problematics / Solution

- An Robotics middleware is required to manage the internal complexity of the autonomous systems
- Which one to choose ?
  - All robotics middleware have their own advantages
  - But also their own inconvenient
  - Too many middleware available in open source
  - Too few at commercial level
- Do not impose a specific software architecture.
- Do not create yet Another middleware
- Solution: Not Choose a middleware
  - Choose the standard approach
  - Choose an architecture style: Component based architecture
  - Only one standard available: RTC (Robotics Technology Component)

OMG RTC: Robotics Technology Component (RTC)

RTC v1.0:

- OMG standard finalized and voted in 2008
- Propose a component based approach for robotics and unmanned systems



GOSTAI Robotics for everyone

THALES

RTC implementations

Open-Source: OpenRTM: http://www.openrtm.org

Commercial: RTM.net: http://www.sec.co.jp/

Human-Robot Interaction Lab.

National Institute of Advanced Industrial Science and Technology AIST

RT CLUE

Ubiquitous Functions Research Group

SONY

Shibaura Institute of Technology

Chuo University

Advanced Intelligence

APPLIED VISION SYSTEMS

University of Tsukuba

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

PatternWeaver for RT-Middleware

RSi Robot Services Initiative

REDONE technologies

HMSL Human Machine Systems Lab.

MSTC 財団法人 製造科学技術センター Manufacturing Science and Technology Center

テクノロジック アート

The information contained in this document and any attachments are the property of THALES. You are hereby notified that any review, dissemination, distribution, copying or otherwise use of this document is strictly prohibited without Thales prior written approval. ©THALES 2008. Template rtp version 7.0.5

GOSTAI Robotics for everyone

THALES

Decision:

Collaboration between THALES and GOSTAI to implement the RTC standard

First standard RTC middleware implementation in Europe

GOSTAI Robotics for everyone

THALES

RTC Deployment and Dynamic Reconfiguration (DDR)

Document number: ab/2010-03-02
Original one: mars/2010-02-14

Noriaki Ando
Infrastructure WG, Robotics DTF
National Institute of Advanced Industrial Science and Technology

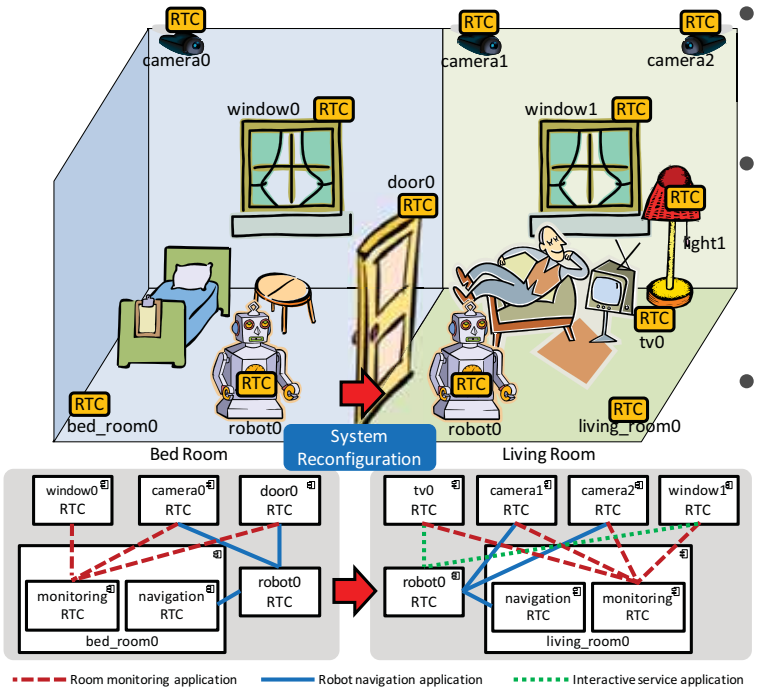
OMG RTC Specification

- Robotic Technology Component (RTC): RTC's component model provides typical functionality and services for robotic systems
  - "Robotic Technology Component Specification" [formal/2008-04-04]
- Implementations:
  - AIST: OpenRTM-aist (C++, Java, Python)
  - SEC: OpenRTM.NET (C#, VB, etc)
  - Korean National Project "OPRoS": partially complient with OMG RTC specification
  - THALES/GOSTAI: GostaiRTC

Objective of this RFP

- This RFP solicits proposals for the deployment and dynamic reconfiguration of RT components.
- In particular, the proposal shall provide:
  - Methods for searching for and deploying RT components into robotic systems at run-time.
  - Methods for notifying the relevant RT component instances of environment changes.
  - Methods for searching for appropriate RT component instances and dynamically reconfiguring them.

Motivation



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

Differences from other specifications

- Deployment interface for RTC
  - Deployment in D&C is static – we need run-time deployment.
  - Connections and configuration need to be alterable at run-time.
- CORBA trading service only specifies static properties
  - We also wish to search for run-time instances based on run-time properties
- The processes that must occur in response to events to execute dynamic reconfiguration are specific to RTCs.
  - Notification service just specifies how to send events; what happens next is not specified.

New specification defining this dynamic functionality is necessary

What's changed from mars/10-02-14

According to the comments from AB members….

- Notation problems
  - Abbreviations, font color, etc…
- "Out of focus". No dynamic features are described.
  - Objective of this RFP
  - 6.5 Mandatory Requirements
  - 6.6 Optional Requirements
  - 6.7 Issue to be Discussed
- Schedule
  - The initial-submission was delayed by one meeting.

Mandatory Requirements (1)

- 6.5.1 Proposals shall specify a meta-model for the description of component meta-information necessary to support automatic searching and comparing of RT components in component repositories and in the run-time system.
- 6.5.2 Proposals shall specify a meta-model for the description of RTC interfaces, their compatibility criteria, and deployment requirements.

Mandatory Requirements (2)

- 6.5.3 Proposal shall specify a platform independent model for information service to identify to locate deployed RTCs available for utilization by the requesting robot.
- 6.5.4 Proposals shall specify a data-model for a component information registry using meta-model requested in requirement 6.5.1 and 6.5.2. Proposal shall also specify query mechanism for this repository.

Mandatory Requirements (3)

- 6.5.5 Proposals shall specify a platform independent model for dynamic RTC configuration and deployment, which
  - allows an efficient configuration of RTCs.
  - initiates reconfiguration based on external and/or internal events. A capability for event filtering shall be provided.
  - supports coordinated reconfiguration of multiple robot systems to support performance of coordinated tasks.
  - defines a service interface for the deployment process.
- 6.5.6 Proposals shall reuse or extend the deployment architecture as defined Deployment and Configuration of Component-based Distributed Applications Specification[D&C].

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 6th, 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*
LOI to submit to RFP due	*August 31, 2010*

Object Management Group

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

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

RTC Deployment and Dynamic Reconfiguration (DDR)

Initial Draft Request For Proposal

OMG Document: ab/2010-03-02

Letters of Intent due: XX June 2010
Submissions due: 23 August 2010

Objective of this RFP

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

In particular, the proposal shall provide:

- Methods for searching for and deploying RT components into robotic systems at run-time.
- ~~Descriptions specific to robotics for the deployment of RT components.~~
- ~~Interfaces for deploying RT components into robotic systems at runtime.~~
- Methods ~~and interfaces~~ for notifying the relevant RT component instances of environment changes.

- Methods ~~and interfaces~~ for searching for appropriate RT component instances and dynamically reconfiguring them.

1.0 Introduction

1.1 Goals of OMG

The Object Management Group (OMG) is the world's largest software consortium with an international membership of vendors, developers, and end users. Established in 1989, its mission is to help computer users solve enterprise integration problems by supplying open, vendor-neutral portability, interoperability and reusability specifications based on Model Driven Architecture (MDA). MDA defines an approach to IT system specification that separates the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform, and provides a set of guidelines for structuring specifications expressed as models. OMG has established numerous widely used standards such as OMG IDL[IDL], CORBA[CORBA], Realtime CORBA [CORBA], GIOP/IIOP[CORBA], UML[UML], MOF[MOF], XMI[XMI] and CWM[CWM] to name a few significant ones.

1.2 Organization of this document

The remainder of this document is organized as follows:

Chapter 2 - *Architectural Context* - background information on OMG's Model Driven Architecture.

Chapter 3 - *Adoption Process* - background information on the OMG specification adoption process.

Chapter 4 - *Instructions for Submitters* - explanation of how to make a submission to this RFP.

Chapter 5 - *General Requirements on Proposals* - requirements and evaluation criteria that apply to all proposals submitted to OMG.

Chapter 6 - *Specific Requirements on Proposals* - problem statement, scope of proposals sought, requirements and optional features, issues to be discussed, evaluation criteria, and timetable that apply specifically to this RFP.

Appendix A – *References and Glossary Specific to this RFP*

Appendix B – General References and Glossary

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 deployment and dynamic reconfiguration 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.

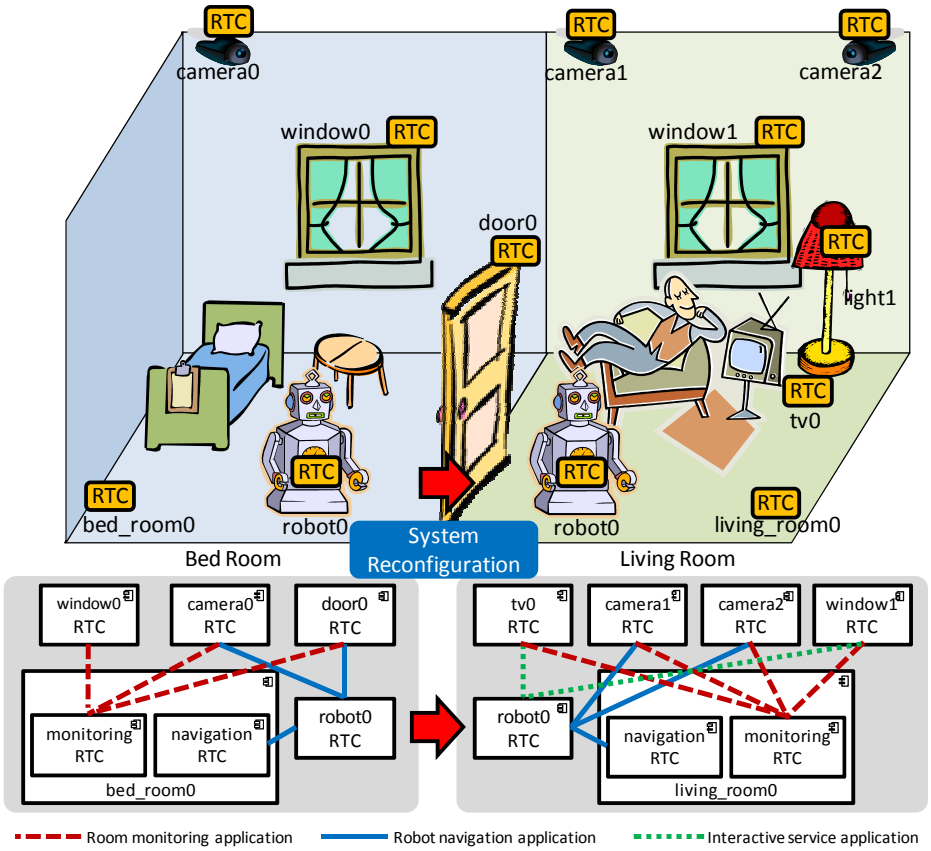


Figure 1 Typical robotic application environment

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

1. ~~RT-component~~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~~a lot of~~ components, this information can be utilized for storing, searching and retrieving components from it. This is called a component profile.

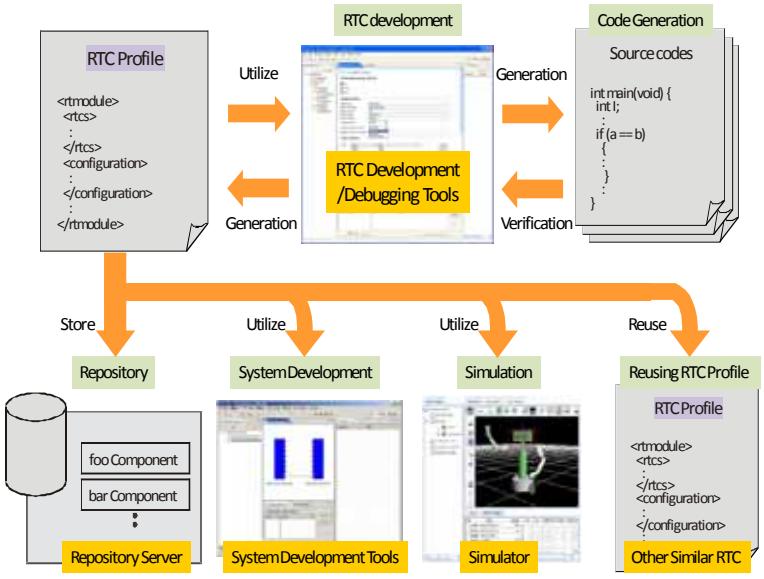


Figure 2 Use of the ~~RT Component~~RTC Profile

2. RTC-based system profile

An RTC-based system is generally built by composing the RTC~~-components~~s 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 RT~~-component~~Cs and component developers (or developing organizations) is increasing, the person in charge of deployment cannot personally manage all the RT~~-component~~Cs that are built. In these cases, a central repository, which manages all the RT ~~component~~Cs 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 RT~~-component~~Cs while deploying the system.

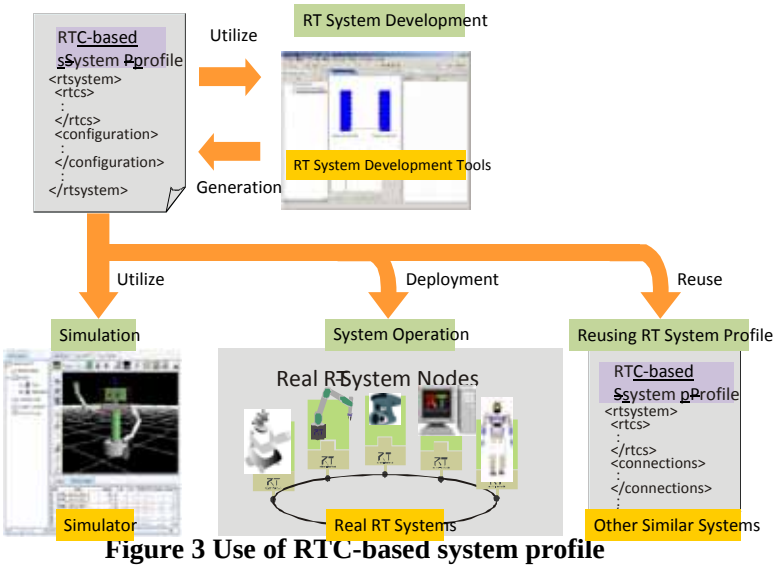


Figure 3 Use of RTC-based system profile

3. RTC-based system deployment

The current RTC specification does not ~~cover~~ provide a declarative way to compose RTC~~ components~~s to build a robot application or system. Many component based systems present ~~their~~ 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 ~~an~~other component instances. Since ~~the~~ a

meta-information-~~-~~based component instance search is needed, the specification must also define the data model for the meta-information of ~~the~~ RT~~-component instance~~Cs~~s~~. 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~~ies~~ the registering (and conversely unregistering) processes by which all component instances register their own meta-information with the directory.

5. ~~RT-component~~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 reconfiguration commonly results from the impairment of the participating component instances and/or changes in the robot location. To support such reconfiguration, the robot application (or system) needs to be notified whenever the situation changes. Since not all changes require reconfiguration, it must be possible to specify the specific environment changes that trigger reconfiguration. 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~~D&C (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~~the~~ way to make RT~~-component~~Cs and build RTC-based systems. However, it does not discuss how to deploy and reconfigure RT~~-component~~Cs at runtime.

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

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

6.2 Scope of Proposals Sought

This RFP solicits proposals to specify common interfaces and common data models for ~~the~~ RTC deployment and dynamic reconfiguration ~~which~~ that is specific and ~~competent~~ relevant to robot applications. The proposals shall include a PIM, using UML ~~2.2~~in the most recent public available version, and one or more PSMs, including one based on CORBA IDL (Interface Definition Language) and XML ~~.~~(eXtensible Mark-up Language).

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

It is necessary to consider the following in the specification:

(1) The repository service interfaces for storing, searching, and retrieving RT~~-component~~Cs, 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 ~~applications~~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]
- Robotic Technology Component Specification Version 1.0 [formal/2008-04-04]
- Deployment and Configuration of Component-based Distributed Applications Specification OMG Available Specification Version 4.0 [formal/2006-04-02]
- Unified Modeling Language: Infrastructure Version 2.2 [formal/2009-02-04]
- Unified Modeling Language: Superstructure Version 2.2 [formal/2009-02-02]
- Meta Object Facility (MOF) Core Specification OMG Available Specification Version 2.0 [formal/06-01-01]
- Common Object Request Broker Architecture (CORBA/IIOP) 3.1 [formal/2008-01-04, formal/2008-01-06, formal/2008-01-08]
- CORBA Component Model OMG Available Specification Version 4.0 [formal/2006-04-01]
- Lightweight Services Specification Version 1.0 [formal/04-10-01]
- Event Service Specification Version 1.2 [formal/04-10-02]
- Naming Service Specification Version 1.3 [formal/04-10-03]
- Enhanced View of Time Specification Version 1.2 [formal/04-10-04]
- Property Service Specification Version 1.0 [formal/00-06-22]
- Mobile Agent Facility Specification Version 1.0 [formal/2000-01-02]

6.3.2 Relationship to other OMG Documents and work in progress

- UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded systems, beta 3 – convenience document with change bars [ptc/09-05-13]
- MARTE model library XMI file [ptc/09-05-16]
- MARTE Profile XMI file [ptc/09-05-15]

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

For all the mandatory requirements, proposals shall provide a Platform Independent Model (PIM) and at least one CORBA-specific model or XML schema for RTC Deployment and Dynamic Reconfiguration. The models shall meet the following requirements.

- ~~—~~Proposals ~~shall specify common interfaces~~ shall specify a meta-model for ~~storing,~~ the description of component meta-information necessary to support automatic searching and ~~retrieving RTCs,~~comparing of RT components in component repositories and ~~shall also provide data models describing RTC profiles.~~

~~6.5.1 Proposals shall specify common interfaces for storing, searching and retrieving RTC-based applications, and shall also provide data models for RTC-based~~in the run-time system ~~profile describing RTCs connection structure. The interfaces for searching appropriate RT-component instances. The scope of the search shall not be restricted to a single host~~.

6.5.2 Proposals shall specify a ~~common~~meta-model for the description of RTC interfaces, their compatibility criteria, and deployment requirements.

6.5.3 Proposal shall specify a platform independent model for information service to identify to locate deployed RTCs available for utilization by the requesting robot.

6.5.4 Proposals shall specify a data-model for a component information registry using the meta-model requested in requirements 6.5.1 and 6.5.2. The proposal shall also specify a query mechanism for this repository.

6.5.5 Proposals shall specify a platform independent model for dynamic RTC configuration and deployment, which

a) allows an efficient configuration of RTCs.

b) initiates reconfiguration based on external and/or internal events. A capability for event filtering shall be provided.

c) supports coordinated reconfiguration of multiple robot systems to allow performing coordinated tasks.

~~d)~~ defines a service interface for ~~RTC deployment into the nodes that constitute an RTC-based system, and shall also provide the data models for describing the details of deployment~~the deployment process.

~~*6.5.6*~~ Proposals shall ~~specify common interfaces for RTC registration, searching, discovery and notification of environmental changes.~~reuse or extend the deployment architecture as defined by the Deployment and Configuration of Component-based Distributed Applications Specification[D&C].

6.6 Optional Requirements

~~Proposals shall follow CORBA CCM and DEPL specification for component deployment.~~

None

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 demonstrate its feasibility by using a specific application based on the proposed model.

- 6.7.2 Proposals shall demonstrate its applicability to existing technology such as the RTC specification [RTC].

- 6.7.3 Proposals shall discuss simplicity of implementation.

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

- 6.7.5 Proposals shall discuss the possibility of providing a standard mechanism for advertising and~~,~~ querying component instances and receiving change notifications

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

6.7.7 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

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.

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	~~*June*~~*August 31, 2010*
Initial Submissions due and placed on OMG document server ("Four week rule")	~~*August 23rd*~~*November 6th, 2010*
Voter registration closes	~~*September*~~*December, 2010*
Initial Submission presentations	~~*September*~~*December, 2010*
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Four week rule")	~~*February 21st*~~*May , 2011*
Revised Submission presentations	~~*March,*~~*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	~~*March*~~*June, 2011*
BoD votes to adopt specification	~~*June*~~*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~~D&C] 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

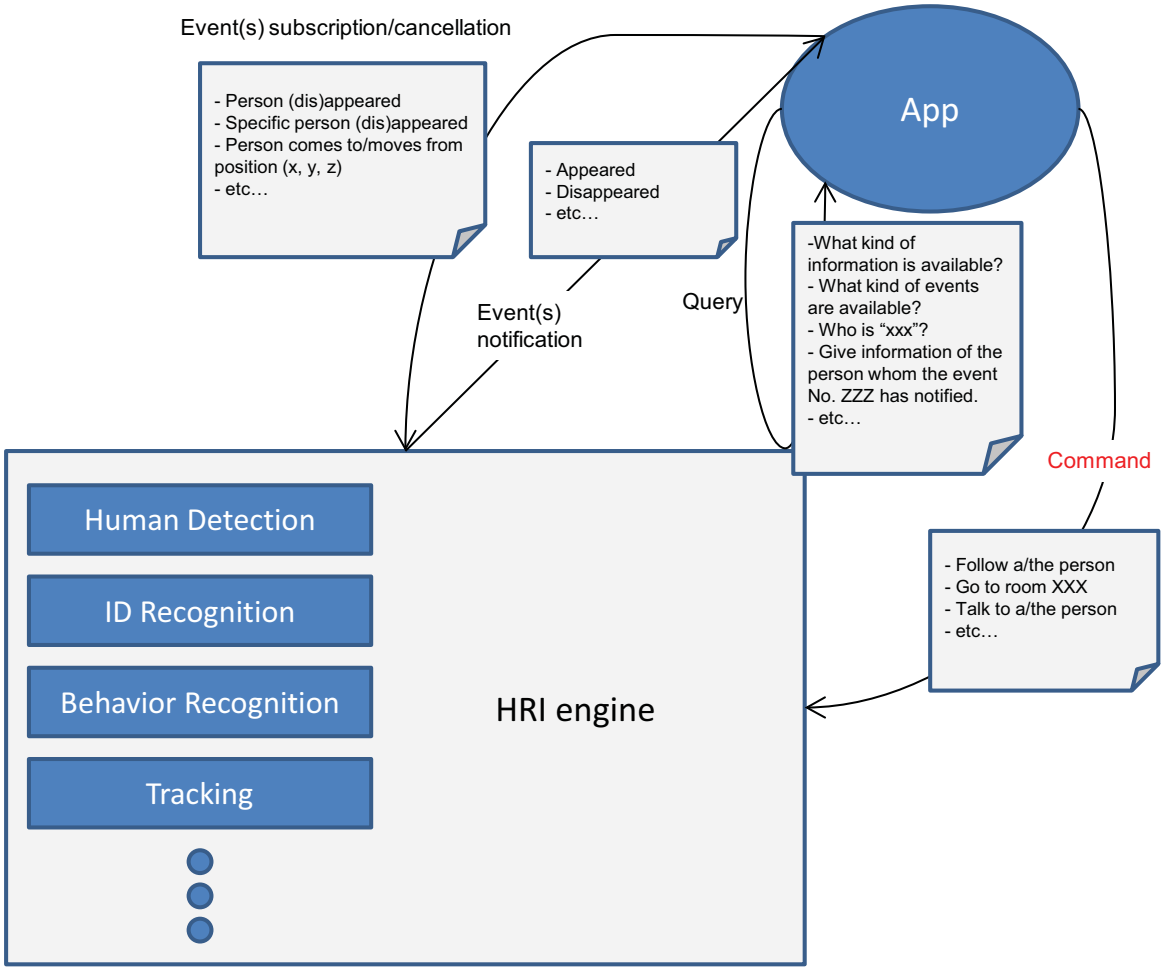
[CORBA] Common Object Request Broker Architecture (CORBA/IIOP),
http://www.omg.org/technology/documents/formal/corba_iiop.htm

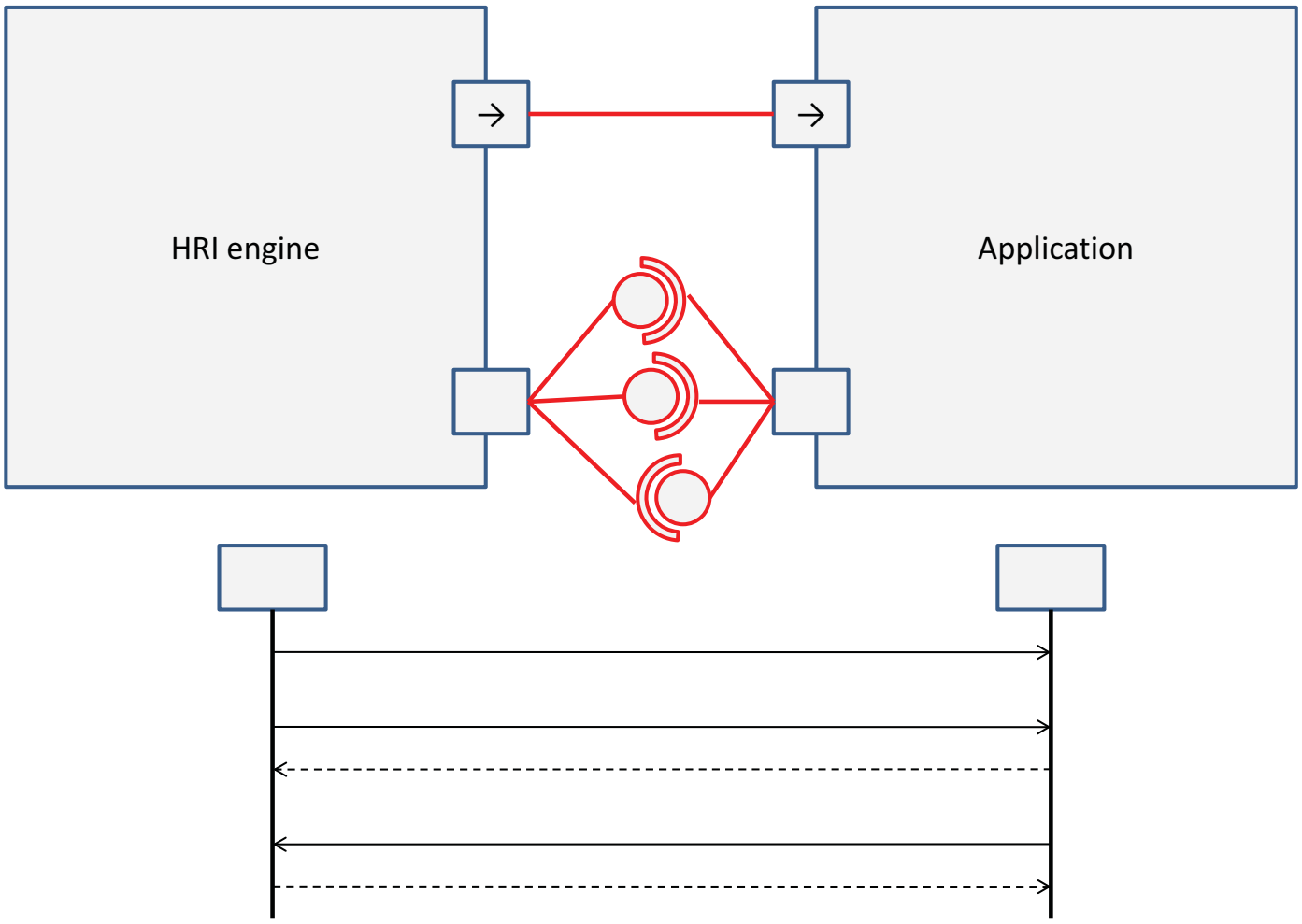
[CSIV2] [CORBA] Chapter 26

[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

Robot Interaction Service (RoIS) Framework





What should be discussed in proposal?

- Data types of human profile
  - For event notifications, queries, etc.
  - Combination with RLS standard
- Event types of human action
- Query types
  - Details of HRI engine functions
  - HRI engine types
  - etc…
- Command sets for human-robot interaction
- Query interface for accessing detailed data of event source
- etc…

GostaiRTC

OMG RTC compliant middleware
made by Gostai

www.gostai.com

Gostai

- French SME located in Paris
- Founded 2006
- 15 employees
- Creator of Urbi, a parallel and event-driven middleware for robotics and complex systems
- Identified as one of the 16 key players of today's personnal robotics (ABI Research, 2008-2009)

Urbi is a **platform** for robotics and complex systems which includes:

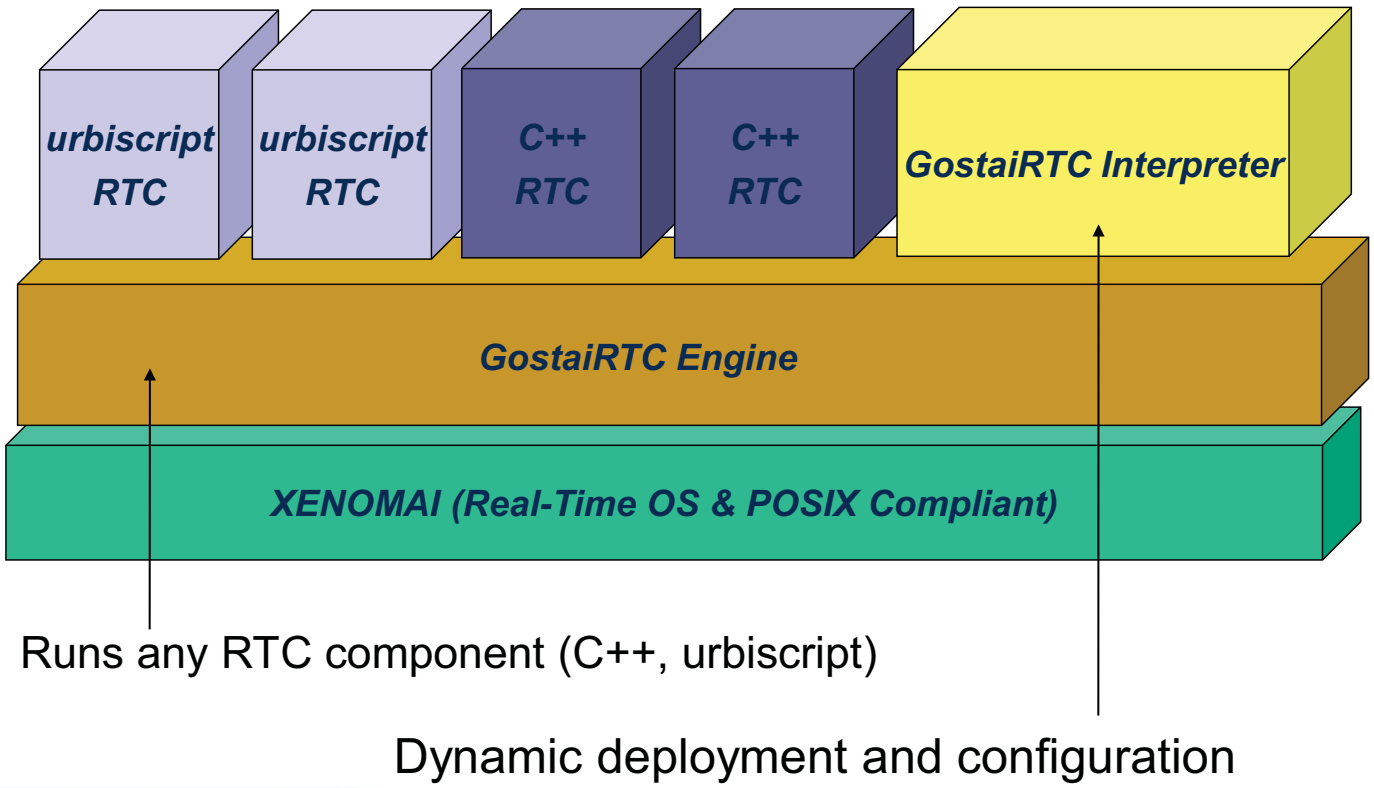
- A C++ **component** architecture
- **Interfaces** to Java/Matlab/Python/...
- The **urbiscript** language
- The **Gostai Studio** IDE

Our customers also wanted a **standard** middleware for robotics, so we are proud to announce the creation of **GostaiRTC**, Gostai's implementation of the RTC standard :

- OMG RTC 1.0 (formal/2008-04-04)
- Local PSM

OMG OBJECT MANAGEMENT GROUP

- Real-Time features
- Low latency
- Low memory footprint (12MB)
- Publish/subscribe model
- Dynamic deployment and configuration
- Language supported: C/C++
- Rapid prototyping / links with simulator
- Compatible with Urbi



Functions:

- Configure and reconfigure dynamically RTC system
- Manage events (internal or external changes)
- Manage components directory

Features:

- parallel & event-driven semantics
- object language
- introspection
- live components graphical visualisation
- strong binding with C++
- easy syntax inspired by known languages

SCENARIO

- Introspect RTC components and execution context
- Create FileSink component
  - writes data from REQUIRED port to a file
- Create FileSource component
  - reads PROVIDED port content from a file (FIFO in the video), of type float
- Connect FileSource to FileSink and test
- Force error in FileSource with invalid data
- Reset components and reactivate source
- Create hypothetical sensor component, connect it to FileSink and test it live from the interpreter

Source source code in urbiscript

```
/** This RT DataFlowComponent has one PROVIDED port that is
 * filled from a File. The port is filled from the next
 * line of the file at each cycle.  */

var Global.FileSource = USDataFlow.new | do (FileSource)
{
  function on_initialize()
  {
    var this.stream = nil;
    var this.asfloat;
    create_output_port("val");
    0;
  };
  // Set the source location. If asfloat is true, data is
  // interpreted as a floating point value. Otherwise, as a
  // string.
  function set_input_file(location, asfloat = true)
  {
    stream = InputStream.new(File.new(location));
    this.asfloat = asfloat
  };
  function on_execute(ctx)
  {
    try
    {
      var v = stream.getLine;
      if (asfloat) v = v.asFloat;
      val = v;
      0
    }
    catch(var e)
    {
      // Put us in error state
      1
    }
  };
};
```

Sink source code in C++

```
/* FileSink: RTObject with DataFlow interface.
 * This component has one REQUIRED port that dumps everything that is written
 * in it to a file.
 */
#include <fstream>
#include <urtc.hh>
#include <uportservice.hh>

URTC_DATAFLOW_OBJECT_START(FileSink)
URTC_DATAFLOW_METHODS_DECLARE(FileSink)
// Put your class member declarations here.
public:
  bool set_output_file(const std::string& filename);
private:
  std::ofstream ofile;
  URTC::UPortService* port;
URTC_DATAFLOW_OBJECT_END(FileSink)
UStart(FileSink);

bool FileSink::set_output_file(const std::string& filename)
{
  ofile.open(filename.c_str(), std::ios::out);
  return ofile.good();
}
// Port declaration goes here.
RTC::ReturnCode_t FileSink::on_initialize()
{
  set_component_profile("Write what is written to the val port to a file",
                        "1.0",
                        "Gostai",
                        "examples");
  port = create_input_port("val");
  // make set_output_file accessible in urbiscript
  UBindFunction(FileSink, set_output_file);
  return RTC::RTC_OK;
}

RTC::ReturnCode_t FileSink::on_finalize()
{
  return RTC::RTC_OK;
}

RTC::ReturnCode_t FileSink::on_startup(ExecutionContextHandle_t exec_context)
{
  return RTC::RTC_OK;
}

RTC::ReturnCode_t FileSink::on_shutdown(ExecutionContextHandle_t exec_context)
{
  return RTC::RTC_OK;
}

RTC::ReturnCode_t FileSink::on_activated(ExecutionContextHandle_t exec_context)
{
  return RTC::RTC_OK;
}

RTC::ReturnCode_t FileSink::on_deactivated(ExecutionContextHandle_t exec_context)
{
  return RTC::RTC_OK;
}

RTC::ReturnCode_t FileSink::on_aborting(ExecutionContextHandle_t exec_context)
{
  return RTC::RTC_OK;
}

RTC::ReturnCode_t FileSink::on_error(ExecutionContextHandle_t exec_context)
{
  return RTC::RTC_OK;
}

RTC::ReturnCode_t FileSink::on_reset(ExecutionContextHandle_t exec_context)
{
  return RTC::RTC_OK;
}

RTC::ReturnCode_t FileSink::on_execute(ExecutionContextHandle_t exec_context)
{
  ReturnCode_t res = RTC::RTC_OK;
  if (port->has_changed())
  {
    if (!ofile.good())
      return RTC::PRECONDITION_NOT_MET;
    urbi::UValue v = port->read_uvalue(res);
    v.print(ofile) << std::endl;
  }
  return res;
}

RTC::ReturnCode_t FileSink::on_state_update(ExecutionContextHandle_t exec_context)
{
  return RTC::RTC_OK;
}

RTC::ReturnCode_t FileSink::on_rate_changed(ExecutionContextHandle_t exec_context)
{
  return RTC::RTC_OK;
}
```


Create source and sink components, and connect them

```
// Create components
create_component(FileSink, "sink")|
sink.set_output_file("/tmp/sink.out")|
defaultExecutionContext.add_component(sink)|
defaultExecutionContext.activate_component(sink)|;

create_component(FileSource, "source")|
source.set_input_file("/tmp/source.in")|
defaultExecutionContext.add_component(source)|
defaultExecutionContext.activate_component(source)|;


// Run
defaultExecutionContext.start();


// Connect
var cp = sink.create_connector_profile;
sink.connect("val", cp);
source.connect("val", cp);
```

Check error and recover

```
// Check error status
defaultExecutionContext.status;

// Recover
defaultExecutionContext.status;
defaultExecutionContext.reset_component(source);
defaultExecutionContext.status;
defaultExecutionContext.activate_component(source);
```


[Video]

23 March 2010
Jacksonville, Fl

SAE JAUS
Joint Architecture for Unmanned Systems

Introductory Briefing

David Martin
DaveMartin@DeVivoAST.com
AS-4A Chair

SAE JAUS Status Briefing

23 March 2010
Jacksonville, Fl

"The nice thing about standards is that there are so many of them to choose from."

Andrew S. Tanenbaum

SAE JAUS Status Briefing

23 March 2010
Jacksonville, Fl

Introduction

- The **Joint Architecture for Unmanned Systems (JAUS)** is an open message standard to ensure Unmanned Systems interoperability and evolution with resultant cost savings.
- JAUS has five objectives:
  - All classes of unmanned systems
  - Rapid technology insertion
  - Interoperable control units
  - Interchangeable/interoperable payloads
  - Interoperable unmanned systems

SAE JAUS Status Briefing

- Introduction
- Overview
- Interoperability
- Background
- Migration
- Services
- Example
- Transport
- Status
- Challenges
- Contact

23 March 2010
Jacksonville, Fl

What is JAUS?

- An authoritative description of the jurisdiction and application of the standard,
- A common language, or set of messages, to facilitate the communication of information pertinent to the operation of unmanned systems,
- The rules and constructs associated with the use of the language, and
- The recommended practices for integration of the language for transmission through electrical and radio frequency mediums.

What is JAUS?

(short version)

JAUS defines a common message set, the protocol governing message exchange, and the rules for message transmission.

JAUS promotes interoperability through formal definition of local interfaces.

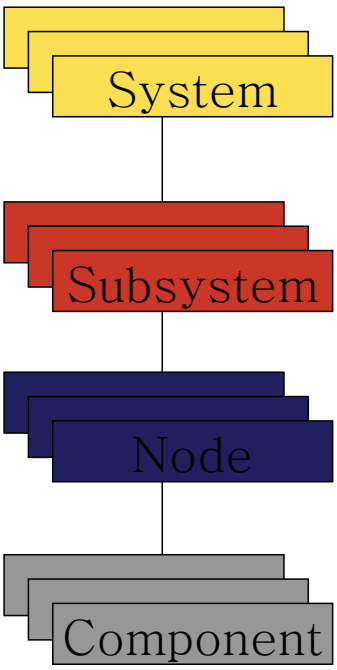
SAE JAUS Status Briefing

"***The primary purpose of JAUS is interoperability.***"

- JAUS was created to resolve issues such as:
  - Subsystems common to all Unmanned Systems are built from scratch for each unique system.
  - Performance gains made by one system cannot be leveraged by a different system
  - New technologies cannot be rapidly incorporated into existing systems.
  - Systems become "locked in" to one vendor's solution.
  - Systems become "locked out" of technology advancements.
- The net effect of JAUS is more efficient development, reduced ownership cost, and an expanded range of vendors.

JAUS can be applied at one or more layers

- JAUS Defines Interoperability at all levels of Unmanned Systems Designs
- Robots, or Subsystems, are replaceable units within a System
- Nodes provide modularity at the Payload and hardware design level
- A JAUS Component supports software modularity



SAE JAUS Status Briefing

23 March 2010
Jacksonville, Fl

JAUS Reduces Risk for an Unmanned System Buyer

- Open consensus standards allow for multiple vendors
- Open consensus standards have the largest group of people examining the product
- Open consensus standards are maintained by well defined processes
- Numerous systems and projects are already incorporating JAUS
  - Army Future Combat System
  - Navy Littoral Combat Ship
  - Navy Unmanned Sea Surface Vehicle
  - …
- Supported by the Department of Defense

JAUS Supports an Expanded Market for Unmanned System Developers

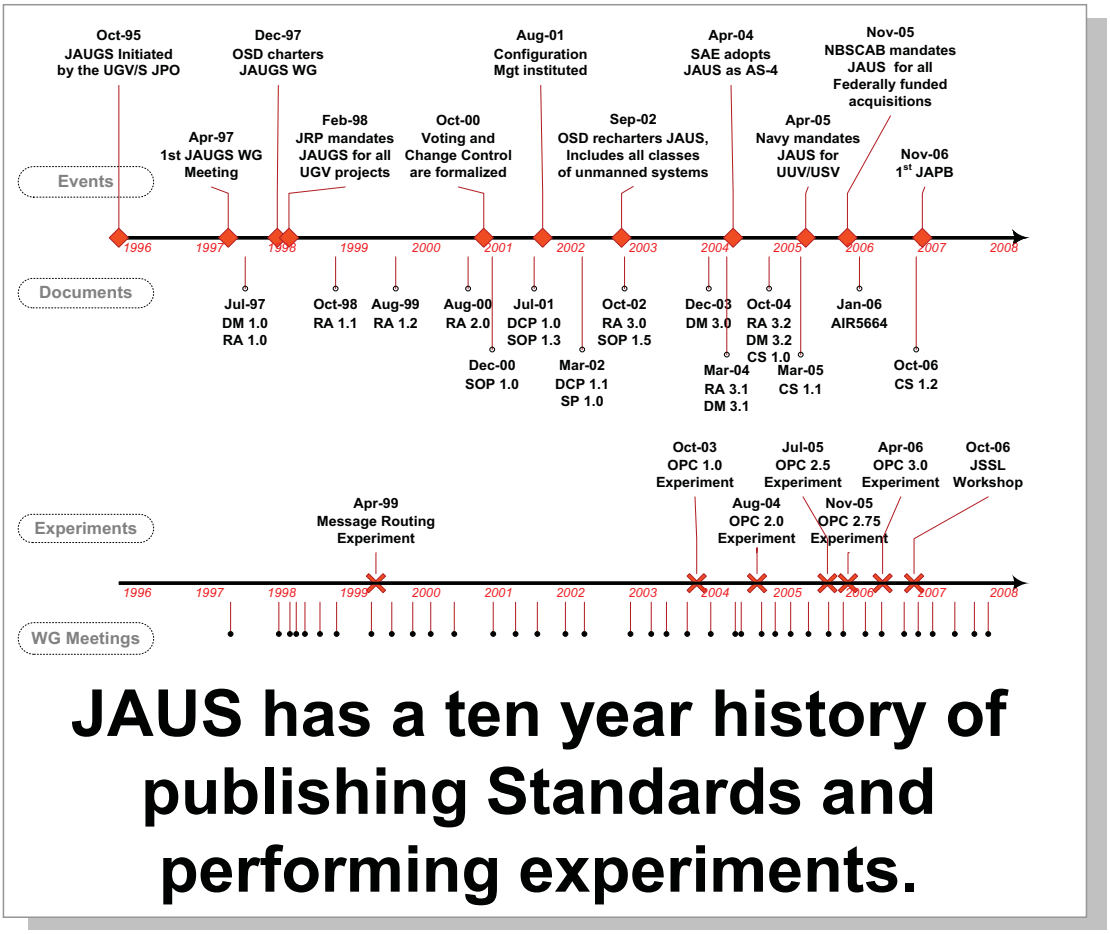
- Standard is open and available for any vendor
  - International standards body with open processes
- Standard focuses on interface definition
- Standard can be expanded using a consensus based open process
- Standard avoids technology or system implementation details
  - Allows vendors to protect intellectual property
  - Allows vendors to design different configurations

SAE JAUS Status Briefing

History

- The Office of the Undersecretary of Defense for Acquisition, Technology and Logistics initially chartered the JAUS WG in 1998 as JAUGS.
- The ‘G’ was formally dropped from JAUGS in August 2002.
- In August 2004 the JAUS WG voted to become the SAE Technical Committee AS-4, Unmanned Systems.

SAE JAUS Status Briefing



JAUS has a ten year history of publishing Standards and performing experiments.

SAE JAUS Status Briefing

- Introduction
- Overview
- Interoperability
- Background
- Migration
- Services
- Example
- Transport
- Status
- Challenges
- Contact

- SAE AS-4 Unmanned Systems Technical Committee
  - Develops standards for Unmanned Systems domain
  - Comprised of subcommittees that manage standards or documents
  - Meets periodically and conducts business using SOP based on Robert's Rules of Order
  - Follows guidelines of Aerospace Council Organization and Operating Guide and SAE Technical Standards Board Governance Policy
- SAE Document Control Process
  - Defines and manages change requests for AS-4 standards
- Membership
  - Members vote on issues & standards
  - Chairperson conducts Technical Committee meetings

SAE JAUS Status Briefing

AS-4 Unmanned Systems
Chair: English
Vice Chair: Kotora
Secretary: Carroll

AS-4A	AS-4B	AS-4C	AS-4D
Architecture Framework	Network Environment	Information Modeling	Performance Measures
Chair: Martin	Chair: Wienhold	Chair: Wit	Chair: Huang
Vice Chair: Culbertson	Vice Chair: Hinton	Vice Chair: Gray	Vice Chair: Pavek

Establishes the **interoperability requirements** and language for AS-4. Past efforts include the Architecture Framework for Unmanned Systems (AFUS) and the Compliance and Interoperability Policy (CIP).

Documents **performance metrics** for evaluation of unmanned systems. Ideal measures are cost effective and independent of underlying technology. Works closely with NIST.

Defines the transport mechanisms for use with JAUS based systems. Establishes **"on-the-wire" message formats** including data specific to a transport medium.

Specifies individual **message format** and utilization rules. Service interfaces formalize the **protocols for message transactions** and describe message details.

Interoperability of Systems and System Components for Unmanned Vehicles

SAE JAUS Status Briefing

23 March 2010
Jacksonville, Fl

Migration to "Services" Approach

- **Early JAUS ran into an issue in which a lack of protocol (sequencing) caused ambiguity**
- **Borrowed concepts from Service Oriented Architecture (SOA) to draft the JAUS Service Interface Definition Language**
  - **Set of semantics for describing an interface**
  - **Schema based on Relax NG (an XML variant)**
  - **Based the five tenets of interface design in book by Gerard Holzmann.**
  - **Enabled re-use through declared types and inheritance**
  - **Protocol description based on state machine design**

Define Essential Elements in a Protocol…*

Service to be provided by the protocol

Assumptions about the environment in which the protocol is executed

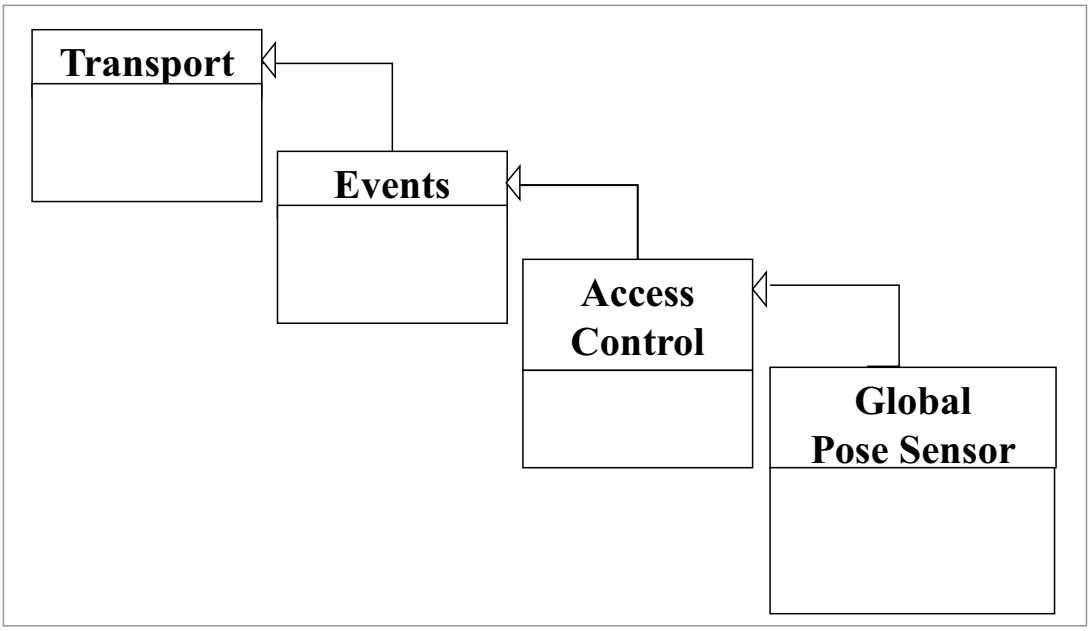
Vocabulary of messages used to implement the protocol

Encoding (format) of each message in the vocabulary

Procedural Rules guarding the consistency of message exchanges

** G.J. Holzmann, Design and Validation of Computer Protocols, Prentice Hall Software Series, 1991*

SAE JAUS Status Briefing

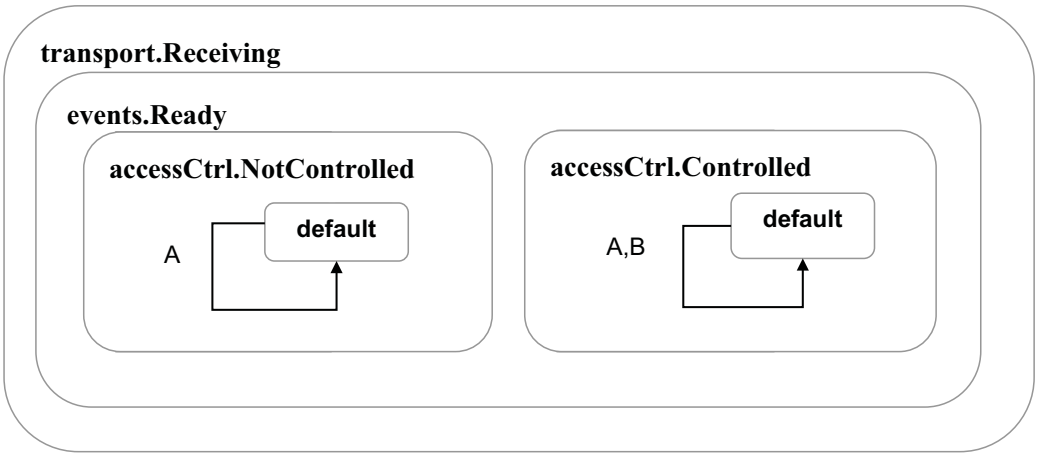


name=GlobalPoseSensor
version=1.0
id= urn:jaus:jss:mobility:GlobalPoseSensor
Inherits-from **AccessControl**
id= urn:jaus:jss:core:AccessControl
version=1.0

SAE JAUS Status Briefing

#	Name	Type	Units	Optional?	Interpretation
1	**<presence_vector>**	unsigned short integer	one	false	
2	**<fixed_field> Latitude**	unsigned integer	degrees	True	Scaled Integer Lower Limit= -90 Upper Limit= 90
3	**<fixed_field> Longitude**	unsigned integer	degrees	True	Scaled Integer Lower Limit= -180 Upper Limit= 180
4	**<fixed_field> Altitude**	unsigned integer	meters	True	Scaled Integer Lower Limit= -10000 Upper Limit= 35000
5	**<fixed_field> Position RMS**	unsigned integer	meters	True	An RMS value indicating the validity of the position data. Scaled Integer Lower Limit= 0 Upper Limit= 100

SAE JAUS Status Briefing



Label	Trigger	Guard	Actions
A	Query Local Pose		sendReportLocalPose
B	Set Local Pose	*isControllingClient*	*updateLocalPose*

Condition	Interpretation
isControllingClient	True if the message that triggered the transition is received from the client that is in control of this service.

SAE JAUS Status Briefing

23 March 2010
Jacksonville, Fl

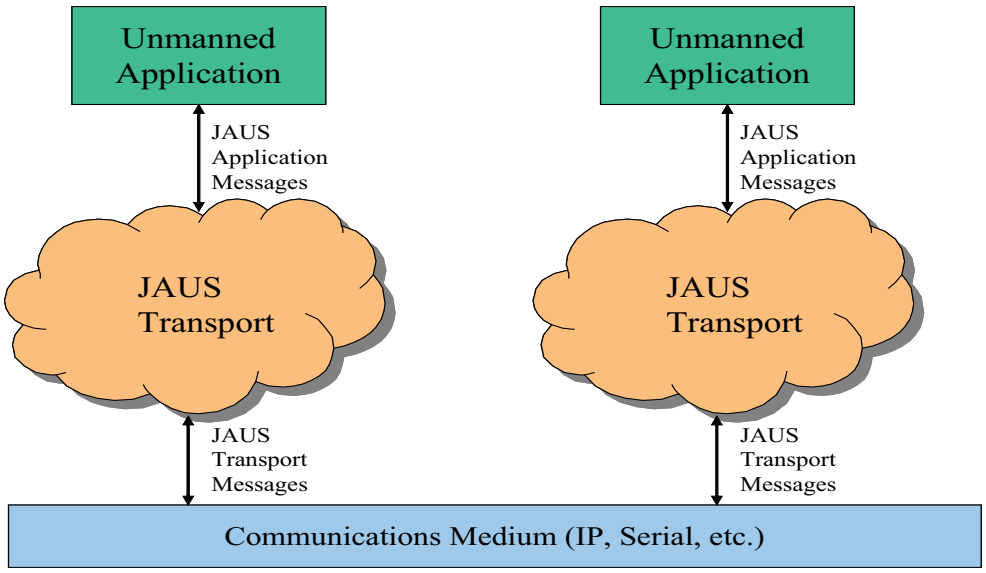
Machine Readable Service Definition

```
<message_def name="SetElement" message_id="041A"
    is_command="true">
      <description xml:space="preserve">
       This message is used to add, insert or replace one or more
          elements in a list.
      </description>
      <declared_header name="AppHeader"
          declared_type_ref="basicTypes.JAUSHeader"/>
      <body name="Body">
       <sequence name="SetElementSeq" optional="false">
        <record name="RequestIDRec" optional="false">
         <fixed_field name="RequestID" field_type="unsigned byte"
                      field_units="one"    optional="false"
                      interpretation="ID of the request."/>
        </record>
        <list name="ElementList" optional="false">
         <count_field field_type_unsigned="unsigned byte"/>
         <declared_record name="ElementRec"
                      declared_type_ref="mobilityTypes.ElementRec"
         optional="false"/>
```


SAE JAUS Status Briefing

- Introduction
- Overview
- Interoperability
- Background
- Migration
- Services
- Example
- Transport
- Status
- Challenges
- Contact

23 March 2010
Jacksonville, Fl

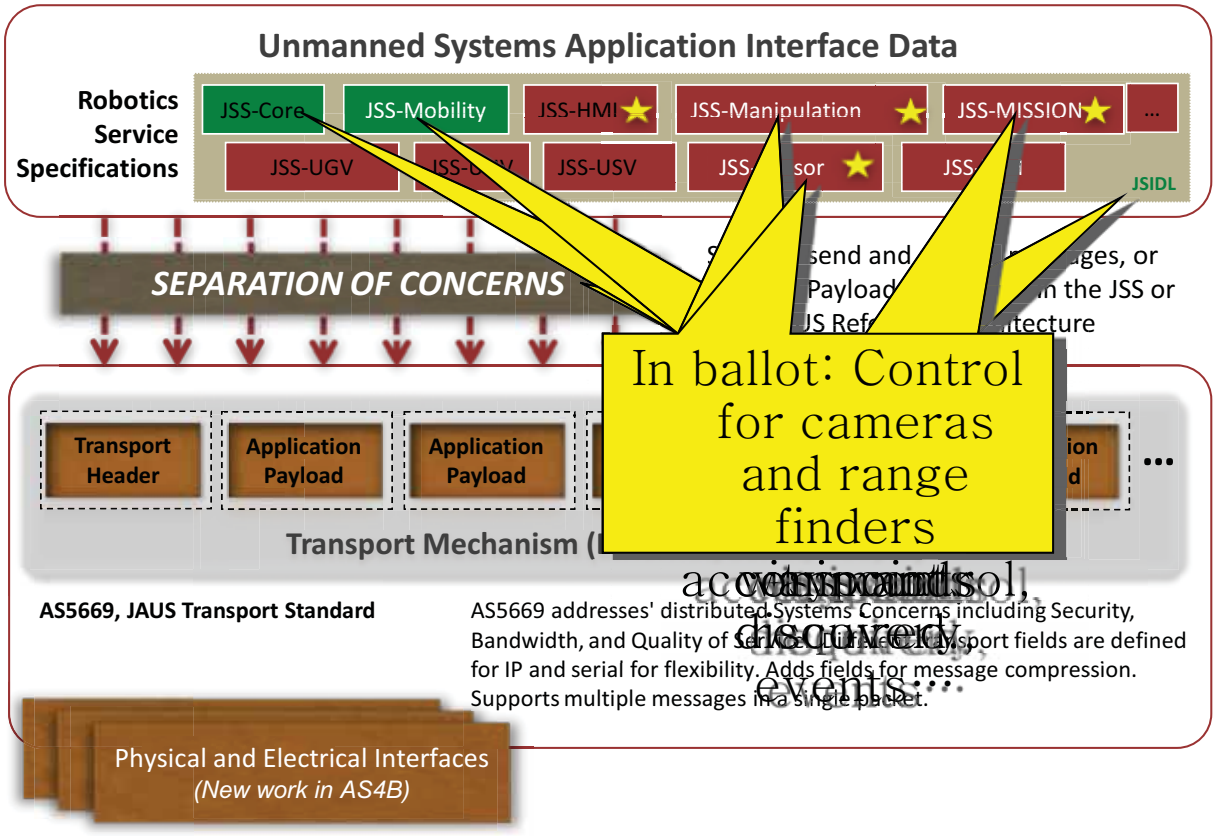


SAE AS 5669 defines 'on-the-wire' behavior for transmission over IP and Serial networks.

- Little Endian
- Binary encoding
- Message routing
- Prioritized delivery
- ACK/NAK behavior
- Data compression

SAE JAUS Status Briefing

JAUS is a collection of Standards.



SAE JAUS Status Briefing

Current Challenges

- **Community Acceptance**
  - "On the wire" encoding has changed very little
  - Perceived complexities with new standard
  - Community tools and libraries require update
  - Waiting on customer (DoD) demand
- **Compliance**
  - Still no compliance body for testing
  - JSIDL removes some ambiguity, making self-certification somewhat more useful
  - ARP 6012 proposed Interoperability Profiles, acquisition offices should test for compliance
- **Coordination with other bodies**
  - STANAG 4586, ASTM, OMG RTC

SAE JAUS Status Briefing

SAE AS-4 Participation

- Membership is open
  - New participants start as liaisons
  - Membership is gained through participation
  - Voting is mandatory for all members
  - Members are technical contributors
- Next meeting: SAE AS-D
  - San Antonio, Texas
  - April 13-14, 2010
- Contact
  - DaveMartin@DeVivoAST.com
  - www.sae.org

SAE JAUS Status Briefing

- Introduction
- Overview
- Interoperability
- Background
- Migration
- Services
- Example
- Transport
- Status
- Challenges
- Contact

23 March 2010
Jacksonville, Fl

Command-line tools for OpenRTM-aist

Geoffrey Biggs

RT Synthesis Group
Intelligent Systems Research Institute
National Institute of Advanced Industrial Science and Technology
Japan

March 24, 2010

Background

- RTSystemEditor offers a lot of power, but:
  - It requires a graphical environment, which may not be available.
  - It uses a lot of resources, which may be limited.
  - It cannot be scripted easily.
- UNIX philosophy of tools that do one thing only and do it well.
  - Create several small tools, each of which performs one task.
- UNIX philosophy of "everything is a file."
  - Use a similar interaction scheme.
  - Components and other objects used by OpenRTM represented as files and directories of a pseudo-file system.
- Two tool kits created:

  rtcshell Tools for interacting with and managing individual components.

  rtsshell Tools for interacting with and managing complete component networks.

rtcshell

rtcwd	Changes the current working directory in the pseudo-file system.
rtpwd	Prints the current working directory.
rtls	Lists the contents of the current working directory.
rtfind	Searches for components.
rtcat	Shows the contents of "files."
rtconf	Manages component configuration parameters.
rtmgr	Interacts with managers.

rtcshell

rtact	Activates a component.
rtdeact	Deactivates a component.
rtreset	Resets a component.
rtcon	Connects two ports.
rtdis	Removes connections.
rtinject	Inject data into an input port.
rtprint	Display data being sent by an output port.

G. Biggs (AIST) rtcshell March 24, 2010 3 / 4

rtsshell

- Used to manage complete RT Systems.
- Works with RTSProfile XML files.

rtcryo	Examines a running system and stores all settings into an RTSProfile file.
rtteardown	Removes all connections in an RT System.
rtresurrect	Reconstructs all connections in an RT System based on an RTSProfile file.
rtstart	Activates all components in an RT System. Will obey component state-change orderings.
rtstop	Deactivates all components in an RT System. Will obey component state change orderings.

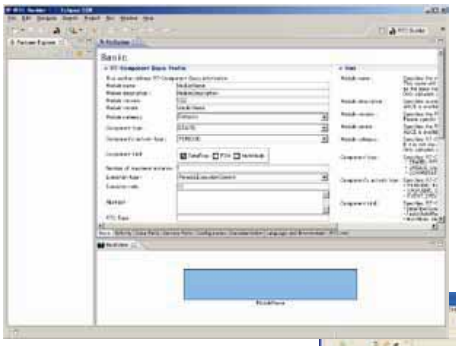
Introduction of RT-Middleware Tools

2010/3/23

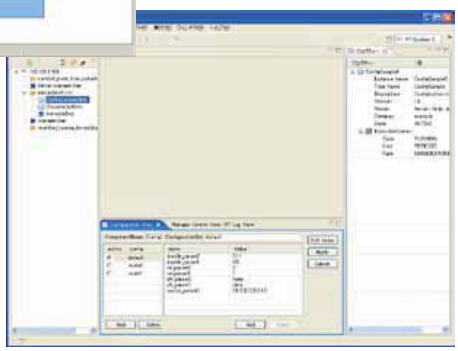
株式会社テクノロジックアート
TECHNOLOGIC ARTS INCORPORATED

OpenRT Platform

- Tool chain for OpenRTM-aist
  - IDE for each phase of system development
  - IDE based on Eclipse



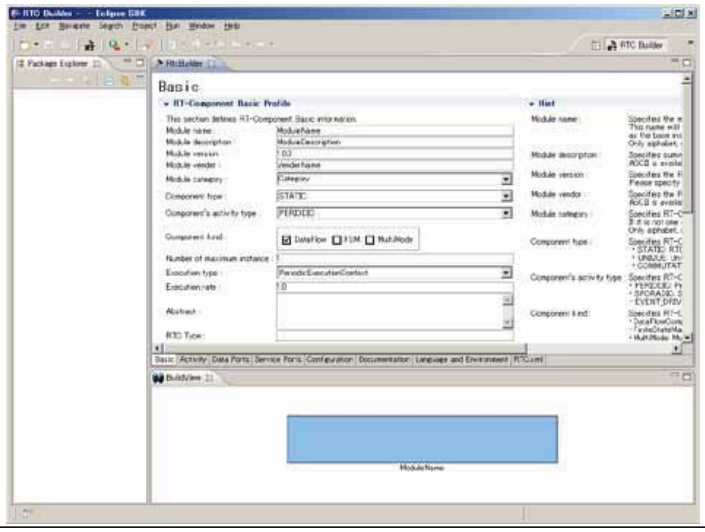
- Tools
  - RTCBuilder
  - RTCDebuger
  - RTSystemEditor
  - Hardware design tool
  - RT Repository
  - Motion Pattern Generator
  - Scenario Designer
  - Real-time software design tool
  - OpenHRP3 (dynamics simulator)



These tools, except for RTCBuilder and RTSystemEditor, are developed by third party organizations.

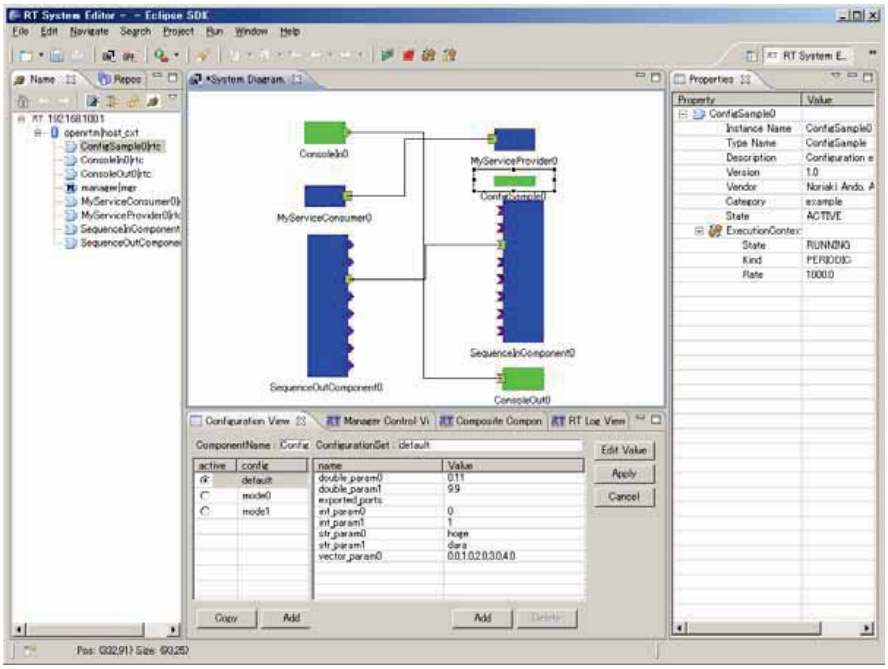
Overview of RtcBuilder

- RTCBuilder
  - Template code generator for RT-Components
  - Code generated from RTC profile
  - Several languages are supported
    - C++
    - Java
    - Python
    - C#
    - VB.NET

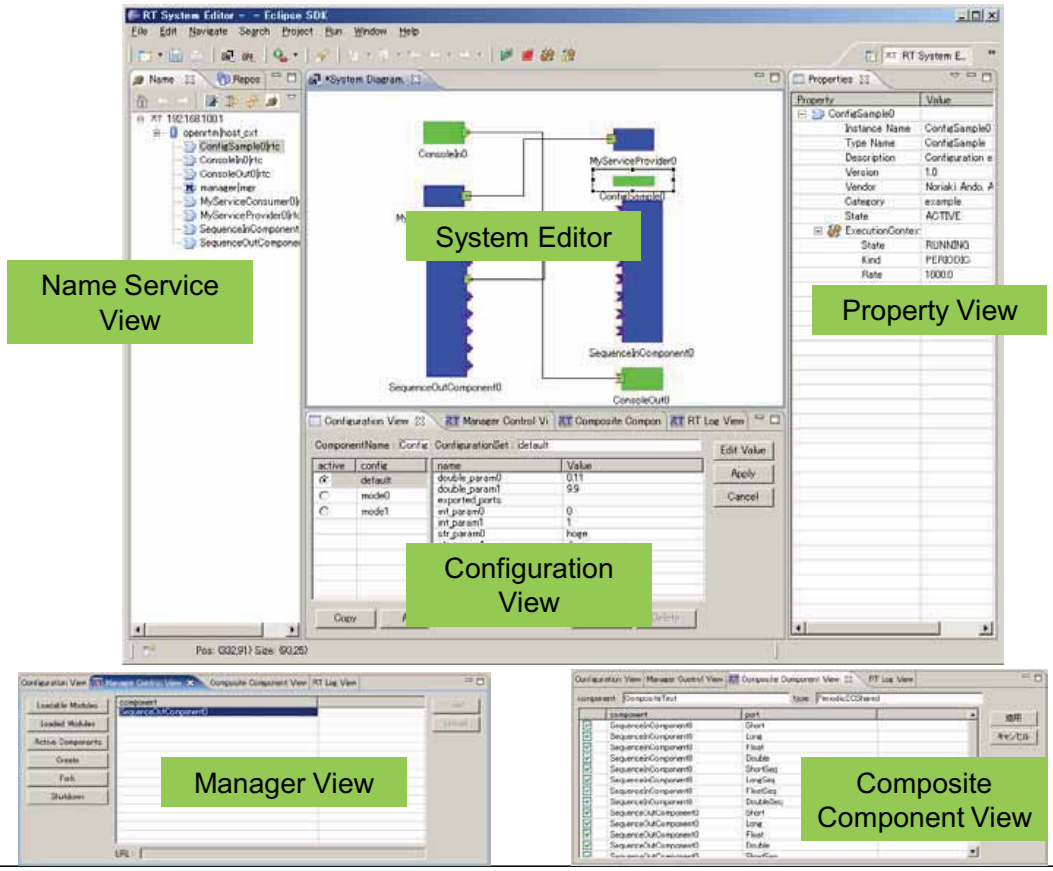


RTSystemEditor

- What is RTSystemEditor
  - RT-Component authoring tool, like simulink, LabView, etc..

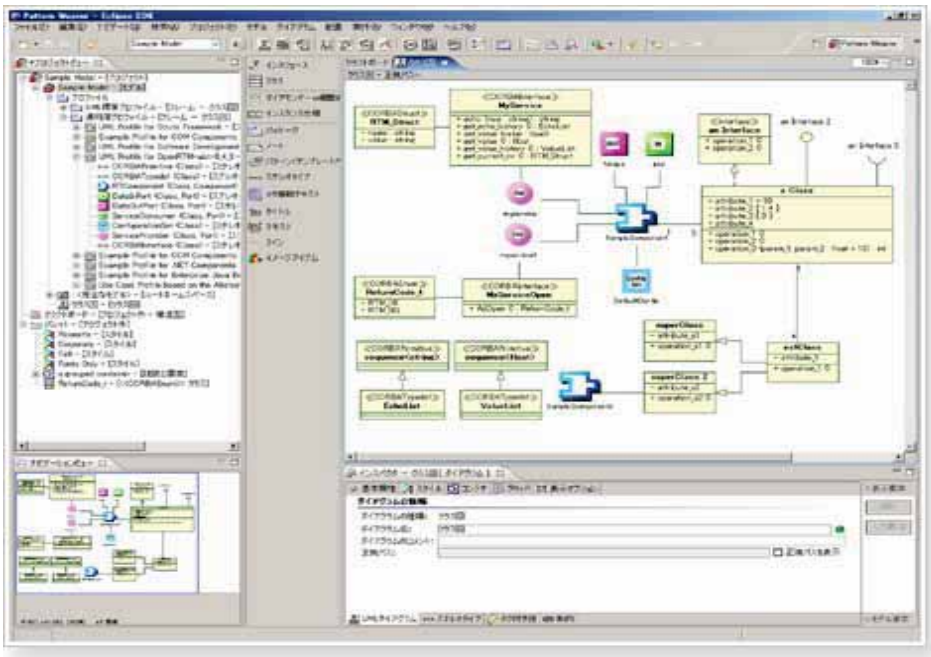


Window Layout



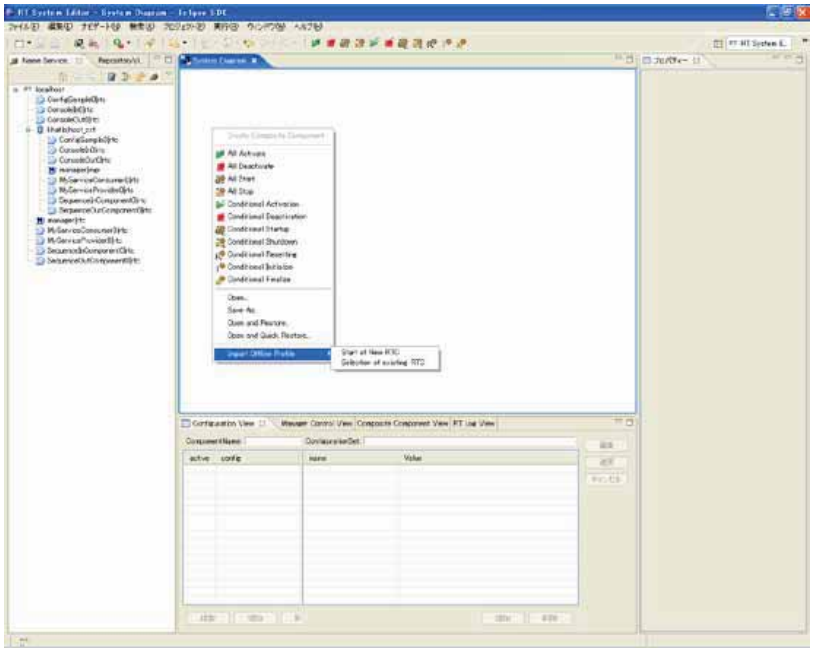
PatternWeaver for RT-Middleware

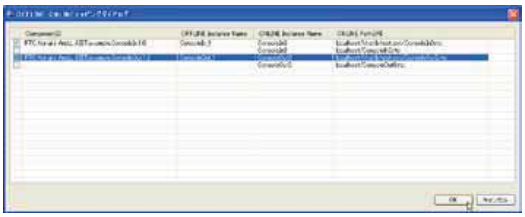
- Template Code Genarator for RT-Componet
  - Based on RtcBuilder (Official Tool)
  - Based on UML Modeling Tool – PatternWeaver
  - Several languages are supported
    - C++
    - Java
    - Python
    - CORBA IDL

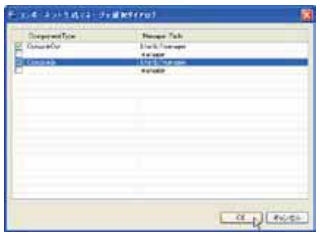


RTSystemLoader

- Build a real system from RT System Profile
  - RT System Profile is defined with an OFFLINE Editor.
  - Build a real system from specification.

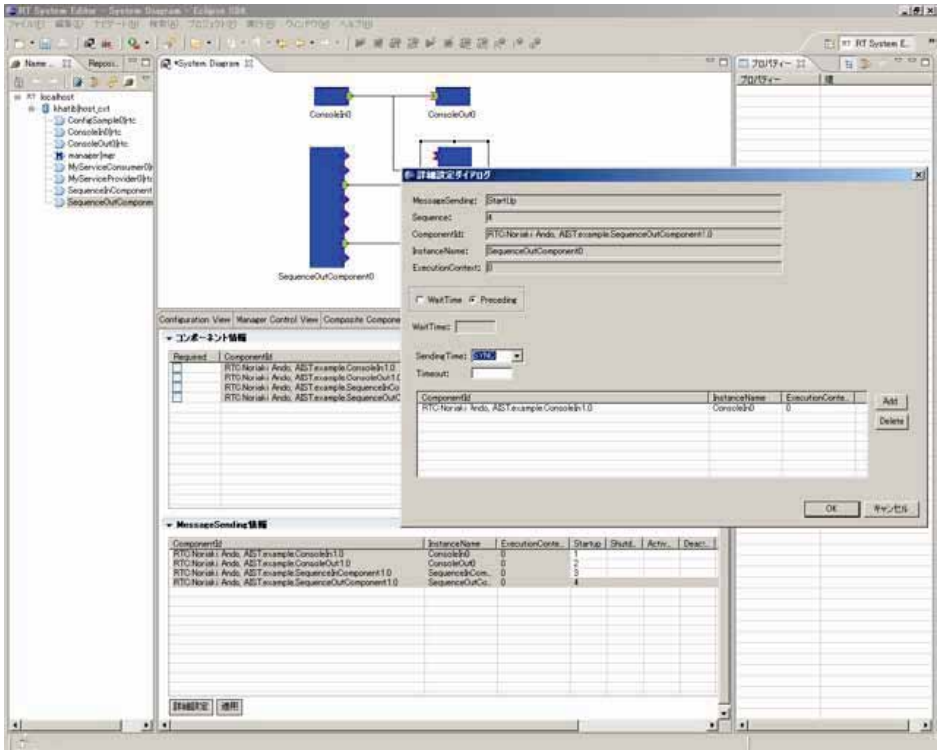






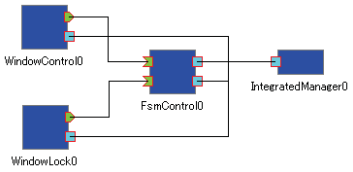
Plug and Play Setting Tool

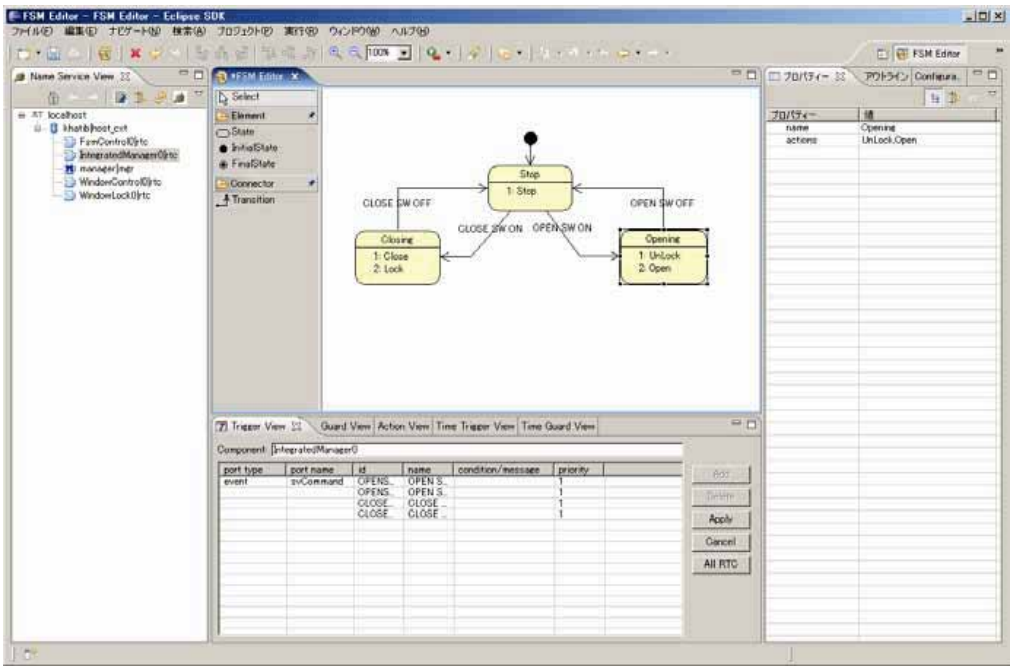
- Setting the action order of components.
- Setting the condition of action.



State Machine Definition Tool

- Define the system state machine.
  - Definition tool & Control Component
  - Control Component manages RTSystem state transitions.

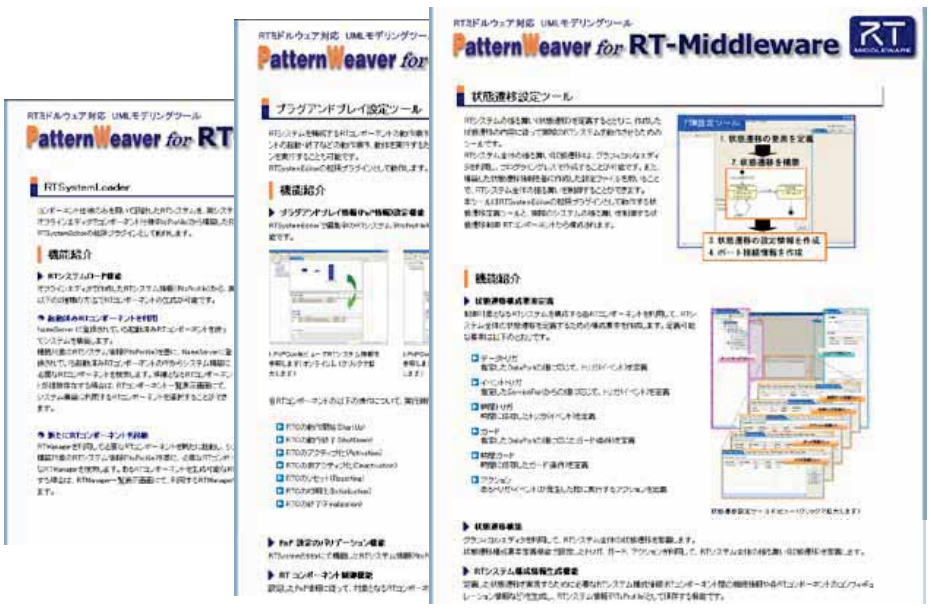




RTMiddleware Tools

http://pw.tech-arts.co.jp/pw/rt_middleware/

※Japanese page Only



THALES

RTC / JAUS: 2 complementary standards

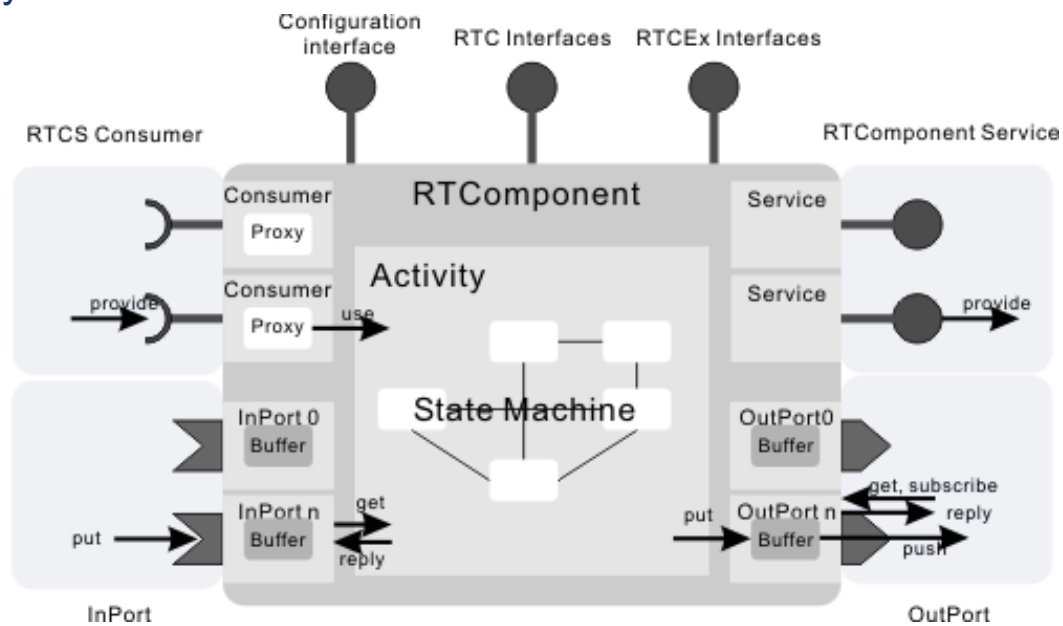
L. Rioux, Robotics TF Co-Chair

Research & Technology

OMG RTC: Robotics Technology Component (RTC)

RTC v1.0:

- OMG standard finalized and voted in 2008
- Propose a component based approach for robotics and unmanned systems



THALES

4 types of RTC components

Basic component:

- Just encapsulates robotic functions and provides interfaces
- A basic component may also contain sub-components

Periodic Sampled Data Processing

- Periodic dataflow component

Stimulus Response Processing

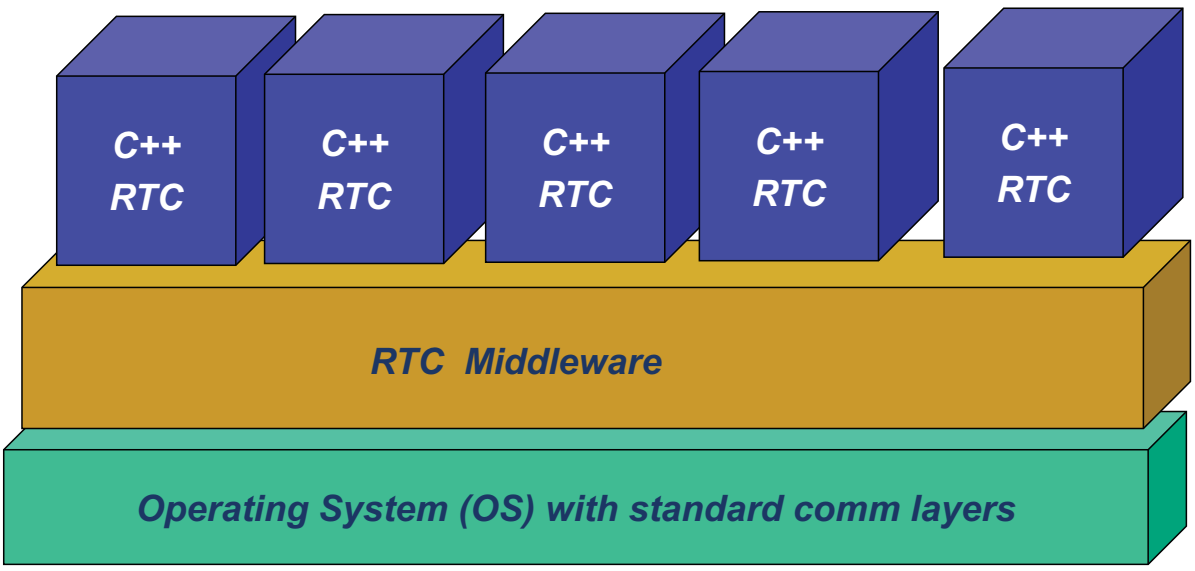
- Manage asynchronous or discrete events
- Manage FSMs

Modes of Operation

- Manage Modes and Configurations

THALES

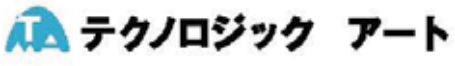
RTC middleware



THALES

Open-Source: OpenRTM: http://www.openrtm.org

Commercial: RTM.net: http://www.sec.co.jp/

THALES

Robotics Technology Components

- PIM component model for Robotics
- PSM: CORBA, Lw-CCM, and others
- Standard execution formalism
  - Interoperability of execution

RTC for robotics algorithm implementation

- Enables Real-Time features
- High Embedded performance
- Mode and lifecycle management of component
- Capitalization at runtime component level
  - Like Corba component, Java code, …

THALES

The goal of JAUS:

- The SAE Joint Architecture for Unmanned Systems (JAUS) is a technology enabler for air, ground, water surface, and underwater unmanned systems.
- SAE JAUS is platform independent and thus provides a standard for interfacing with **different** types of vehicles, sensor platforms, operator control stations, and payloads.
- Open, scalable, and responsive to the unmanned systems communities' needs.

JAUS Service Definition Language

- Provides a common language enabling **external** communication between unmanned systems.
- It incorporates a service based architecture and specifies data formats & protocols.
- Promotes stability of capabilities by projecting anticipated requirements as well as those currently needed.

THALES

36 Vehicles in Semi-Finals, 5 used JAUS

3rd Place team = Victor-Tango [JAUS]

Benefits:

- **Route & Mission Data Formats**
- **Protocol (Emergency Remote)**
- **Processes (safety, operation, …)**
- **Testing / Qualification / Certification**
- **Validation of JAUS Communications**

THALES

SOME FRAMEWORKS using JAUS

RI-JAUS: http://www.repinvatiant.com

OpenJAUS: http://www.openjaus.com

Kairos Autonomi

- ProntoJAUS Software Design Kit

RE²: RE² JAUS Software Development Kit

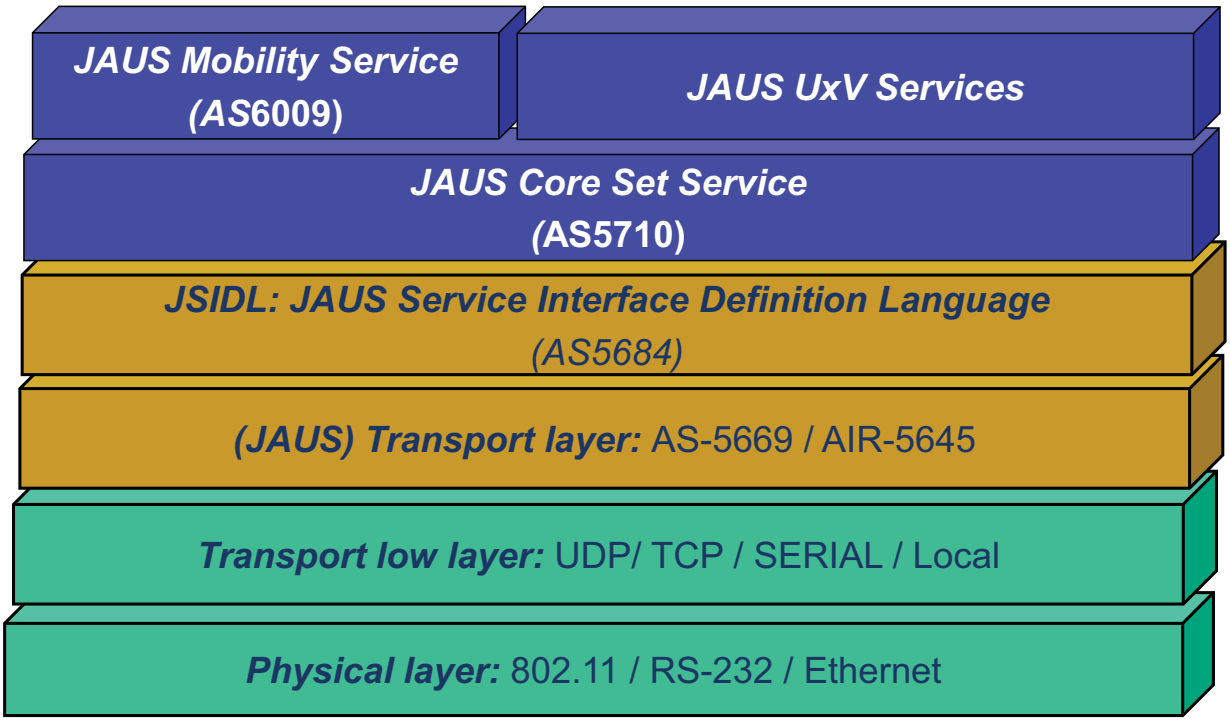
JAUS software

Autonomous Solutions: JAUS Now

DeVivo AST: jr (JAUS Router) Middleware

Skilligent: Jaus Software kit

JAUS: A Service Oriented Architecture



JAUS Services *JAUS middleware* *Comm. layers*

JAUS: Service Oriented Architecture for Unmanned Systems

- Platform independent
- Language independent
- Permits interoperability between Unmanned Systems, Sensors, Payload etc…

JAUS for interoperability between unmanned (sub) Systems

- Integrates unmanned System into one system
- Enable interoperability through Service Oriented Architecture (XML)

JSIDL: A language to create JAUS Service Definitions (JSDs)

JAUS Service written in XML

JAUS Service validated with the JSIDL Schema

THALES

Both JAUS and RTC do not specify these attributes:

RTC does not specify:

- Any robotics Services (except RLS: Robotics Localisation Service)
- RTC do not specify any transport level
- Communication through firewall, long distance communication

JAUS does not specify:

- How JAUS Services are implemented.
- How the runtime environment is implemented.
- How Unmanned algorithms are implemented and executed.

THALES

Join RTC and JAUS: A win-win situation

For RTC:

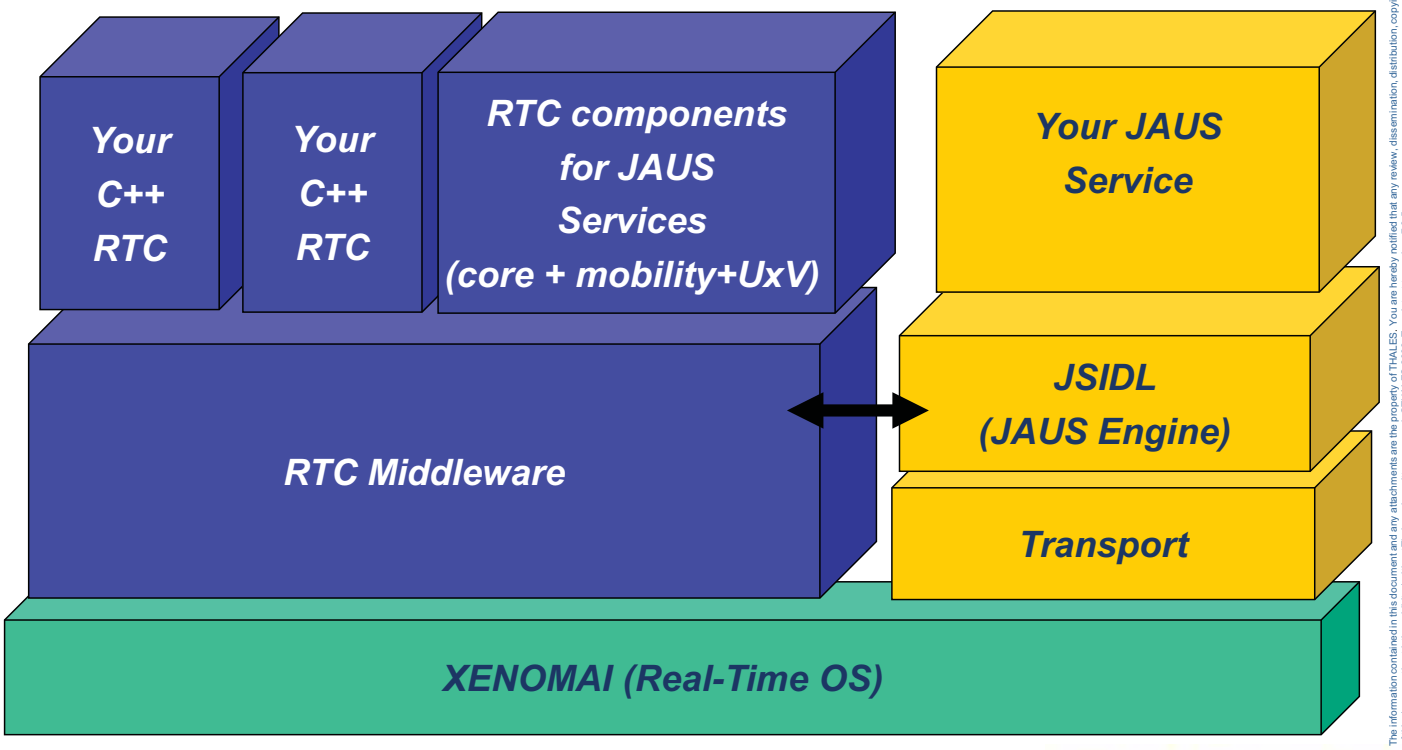
- Capability to address unmanned systems
- Service definitions for unmmaned systems
- Adds ability to build a system of systems

For JAUS

- A consistent implementation of JAUS (may run on different RTC middleware)
- Offer standard component-based approach for implementing JAUS service but also mission and others algorithms.
- Real-Time and embedded performance

THALES

Integrating JAUS



THALES

For RTC:

Define interoperability with JAUS and unmanned systems

- Reuse JAUS (do not define a new one).
- Enable JAUS compliance.
- Extend capabilities over RTE architecture.

For JAUS:

Define standard implementation of JAUS

- Enable RTC compliance.
- Bring real-time and high performance for embedded services /functions.
- Propose standard tools (OMG MDA) for JAUS services design & development.

THALES

Question:

- What about real-time and embedded functions in JAUS/RTC (Sensors, Payload) ?

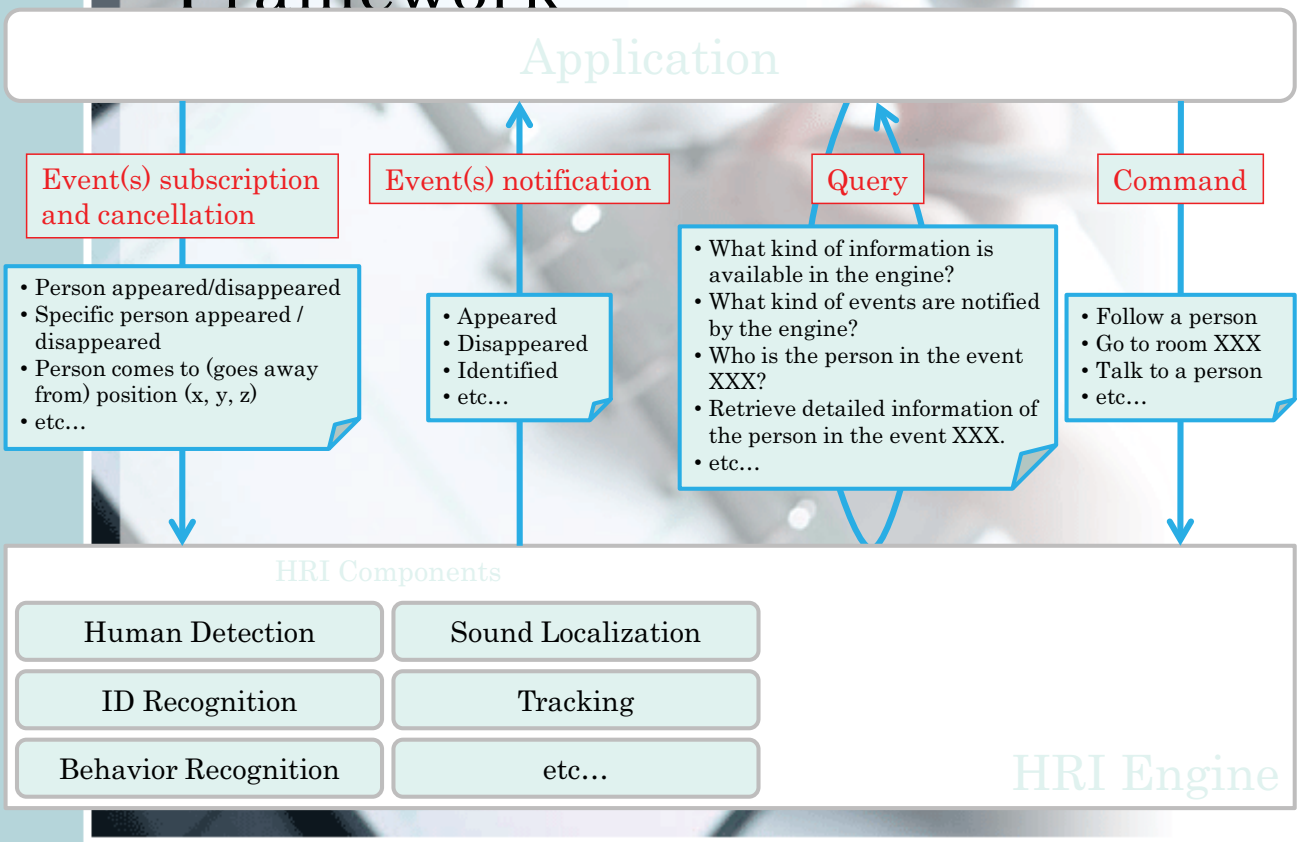
THALES

Robotic Interaction Service (RoIS) Framework

Robotic Functional Services WG,
Robotics DTF

2010/03/23

Schematic Structure of RoIS Framework



Interaction between HRI Engine and Application

- **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

Difference between Query and Command

- Query
  - Retrieve information from HRI engine
  - Status of HRI engine does not change before and after each query
- Command
  - Send motion command for operating robot
  - Set/change configuration parameter(s) of HRI engine component(s) and/or the engine itself
  - Status of robot, HRI engine and/or components in the engine will be changed

What should be discussed in Proposal?

- Data types of human profile
  - For event notifications, queries, etc.
  - Combination with RLS standard

Optional

- Event types of human action
- Query types
  - Details of HRI engine functions
  - HRI engine types
  - etc…
- Command sets for human-robot interaction
- Query interface for accessing detailed data of event source
- etc…

Roadmap

Roadmap

Item	Status	Jacksonville March-2010	Minneapolis June. 21-25, 2010	Boston Sep. 20-14, 2010	Santa Clara December . 6-10, 2010
Robotic Interaction Service Framework	On-going	1st review of RFP	2nd review of RFP and AB	?	Initial submission

Infrastructure WG Progress Report

(Jacksonville meeting)

Co-chairs:

Seung-Woog Jung (ETRI)

Noriaki Ando (AIST)

robotics/2010-03-14

Topics of This Meeting

- RTC DDR RFP 2nd review process
  - Infra. WG meeting (on Sunday)
  - 2nd review in MARS (on Monday morning)
    - mars/10-03-04
  - AB Plenary (on Monday afternoon)
    - ab/10-03-02
  - Infra. WG meeting (on Tuesday morning)
  - MARS meeting for voting (on Thursday)
  - AB Plenary (on Thursday)

Comments for 4-weeks document, and modification

According to the comments from AB members….

- Notation problems
  - Abbreviations, font color, etc…
- 6.5 Mandatory Requirements
- 6.6 Optional Requirements
- 6.7 Issue to be Discussed
  - "Out of focus". No dynamic features are described.
- Schedule
  - The initial-submission was delayed by one meeting.

Mandatory Requirements (1)

- 6.5.1 Proposals shall specify a meta-model for the description of component meta-information necessary to support automatic searching and comparing of RT components in component repositories and in the run-time system.
- 6.5.2 Proposals shall specify a meta-model for the description of RTC interfaces, their compatibility criteria, and deployment requirements.

Mandatory Requirements (2)

- 6.5.3 Proposal shall specify a platform independent model for information service to identify to locate deployed RTCs available for utilization by the requesting robot.
- 6.5.4 Proposals shall specify a data-model for a component information registry using meta-model requested in requirement 6.5.1 and 6.5.2. Proposal shall also specify query mechanism for this repository.

Mandatory Requirements (3)

- 6.5.5 Proposals shall specify a platform independent model for dynamic RTC configuration and deployment, which
  - allows an efficient configuration of RTCs.
  - initiates reconfiguration based on external and/or internal events. A capability for event filtering shall be provided.
  - supports coordinated reconfiguration of multiple robot systems to support performance of coordinated tasks.
  - defines a service interface for the deployment process.
- 6.5.6 Proposals shall reuse or extend the deployment architecture as defined Deployment and Configuration of Component-based Distributed Applications Specification[D&C].

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*

Comments at AB Plenary Meeting

- Mandatory requirements becomes ambiguous
  - The RFP no longer asks for interfaces in 6.5.1.
  - There are many meta-models and data-modes. Relation among them is unclear.
  - Proposed PSM must be clearly specified.
- Other comments
  - Direct-URL should be shown for specification references.
  - At the "Objective" section, RT should be Robotic Technology
  - CORBA IDL should be OMG IDL
  - "identify to locate" in 6.5.3 should be "identify and locate."

Revised document for the next MARS and AB

- According to comments at the AB plenary, mandatory requirements have been revised again.
  - More concrete and more clear
  - Consists of two parts
    - Platform independent deployment and configuration model
    - Platform independent RTC information model
- Other changes
  - The name of specification
    - Old: DDR (Deployment and Dynamic Reconfiguration)
    - New: DDC (Dynamic Deployment and Configuration)
  - 6.6 Optional Requirements
  - 6.7 Issues to be discussed
  - 6.8 Evaluation Criteria

Mandatory Requirements (1)

Platform independent deployment and configuration model

- 6.5.1 Proposals shall specify services and interfaces 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 (2)

Platform independent RTC information model

- 6.5.4 Proposals shall provide a schema describing RTC characteristics.
- 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.

Next

On Thursday

- MARS meeting for review and voting
- AP Plenary Meeting

and

On Friday

- TC approval

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*

Minneapolis Meeting
June 21-25, 2010

Cambridge Meeting
September 20-24, 2010

Santa Clara Meeting
December 6-10, 2010

Contact Report
- IEEE Standards Workshop -

March 23 2010
Young-Jo Cho(ETRI)

Place: Aspen/Spruce Room, Anchorage Hilton Hotel, Anchorage, Alaska
Time: 09:00 ~ 16:00, May 3 (Monday), 2010
Link: http://www.ieee-ras.org/calendar/meetinglist

Purpose of the Workshop

Robotics technology is gaining more importance as a prospective solution to minimize production costs, improve quality of life, and provide safer working conditions. Service robots, in particular, are expected to have a wide influence on various existing and emerging markets, various kinds of which are already working in our daily environment in the form of a cleaning robots, transportation robots, security robots, or unmanned vehicles.

Same as other successful technologies, in order to facilitate the adoption of robots and robotic devices, standardization of key elements of robotics technology is the most effective way. In this workshop, supported by IEEE standards association, we discuss a couple of emerging standardization issues in robotics fields and learn how we can develop new robotics standards through the IEEE standardization procedure.

In the first part of the workshop, we will learn the IEEE guidelines for developing a new standard. In the second part of the workshop, emerging issues in robotics field will be presented to highlight the needs for a new standard. Finally, participants will discuss opportunities and next steps for developing IEEE standards in the field of robotics. Efforts for developing standards which may be introduced to robotics applications will also be presented.

This workshop will serve as a place for sharing knowledge and ideas for developing a new IEEE standard sponsored by Robotics and Automation Society. This workshop is open to anyone who is interested in developing standards, working in the field of robotics or its related technology.

(Draft Agenda)

10:00 - 12:00 Standards Development at IEEE Standards Association: Opportunities and Ideas

10:00 - 10:15	Welcoming Address (Steve Mills, IEEE-SA President elect)
10:15 - 10:45	IEEE-SA Overview
10:45 - 11:00	Break
11:00 - 11:30	Study Group Guidelines
11:30 - 12:00	How to Develop a New Standard: Definition of terms and Q&A

12:00 - 13:00 Lunch (provided) : IEEE and IEEE-SA Orientation

13:00 - 16:00 Technical Project Discussion

13:00 - 13:15	Overview (Steve Mills)
13:15 - 13:40	Emerging Robotics Standard Issue (I)
13:40 - 14:05	Emerging Robotics Standard Issue (II)
14:05 - 14:30	Emerging Robotics Standard Issue (III)
14:30 - 14:50	Break
14:50 - 16:00	Discussion: project opportunities, future steps

Status and Plans for Robotic Localization Service (RLS)

2010.3.23

NISHIO Shuichi
RLS-RTF Chair, JARA/ATR

Status of RLS

- RLS-RTF
  - Approved in December OMG meeting (Long Beach)
- Liaison with ISO/TC211
  - Approved on 5th Jan, 2010
- OMG Specification
  - Published on 16th Feb, 2010
  - http://www.omg.org/spec/RLS/1.0

Date: January 2010

OMG
OBJECT MANAGEMENT GROUP

Robotic Localization Service (RLS)

Version 1.0

OMG Document Number: formal/2010-xx-xx
Standard document URL: http://www.omg.org/spec/RLS/1.0
Associated Schema Files*: http://www.omg.org/spec/RLS/20090901
http://www.omg.org/spec/RLS/20090901/Ability.hpp
http://www.omg.org/spec/RLS/20090901/Architecture.hpp
http://www.omg.org/spec/RLS/20090901/DataFormat.hpp
http://www.omg.org/spec/RLS/20090901/Error.hpp
http://www.omg.org/spec/RLS/20090901/ErrorBase.hpp
http://www.omg.org/spec/RLS/20090901/ErrorType.hpp

Future plans

- Revision at RLS-RTF (OMG) and TC211 work will be held in parallel (MoU planned)
  - Discussion in both fields (Robotics and GIS)
- RLS-RTF meetings will be basically held online (in e-mail basis)
  - If necessary, local meetings may be held
- TC211 meetings are held twice a year
  - 2010: Southampton (May), Sydney (Dec)
- Coordination with Open Geospatial Consortium

Schedule at TC211

- Draft NWIP (1 month circulation)
  - Planned to submit at beginning of Apr.
- Discussion at Programme Maintenance Group
  - Planned to meet at TC211 meeting on May, 2010
    - 24-28 May at Southampton, UK
- NWIP (3 month voting)
- WG assignment, PT starts
  - WG10(Ubiquitous Public Access) planned
  - Start from December TC211 meeting (at Sydney)

ISO International Organization for Standardization

Contact Report: ISO/TC184/SC2

Tetsuo Kotoku
AIST, Japan

Schedule

2010 February Meeting:

- WG1 (Vocabulary on robots and robotic devices) : Feb.11-12
- WG3 (Industrial Safety) : Feb.8-10
- WG7 (Personal care safety) : Feb. 15-17
- WG7/SG on Medical care robots :
- WG8 (Service Robots) : Feb. 12

at Shades of Green (sponsored by RIA)

WG7

Robots and robotic devices
— Safety requirements — Non-medical personal care robot

9 participants **(3rd day)**
(France:1, Germany:3, Japan:5, Korea:3, UK:3, USA:2, OMG:2)

Resolve comments
from the national bodies.

Next Meeting

Future Meetings:
•2011 January meeting will be held in New Zealand.

Paris, France

- WG1: Jun. 24
- WG3: Jun. 21-23
- WG7: Jun.?
- WG8: Jun. 25

Budapest, Hungary

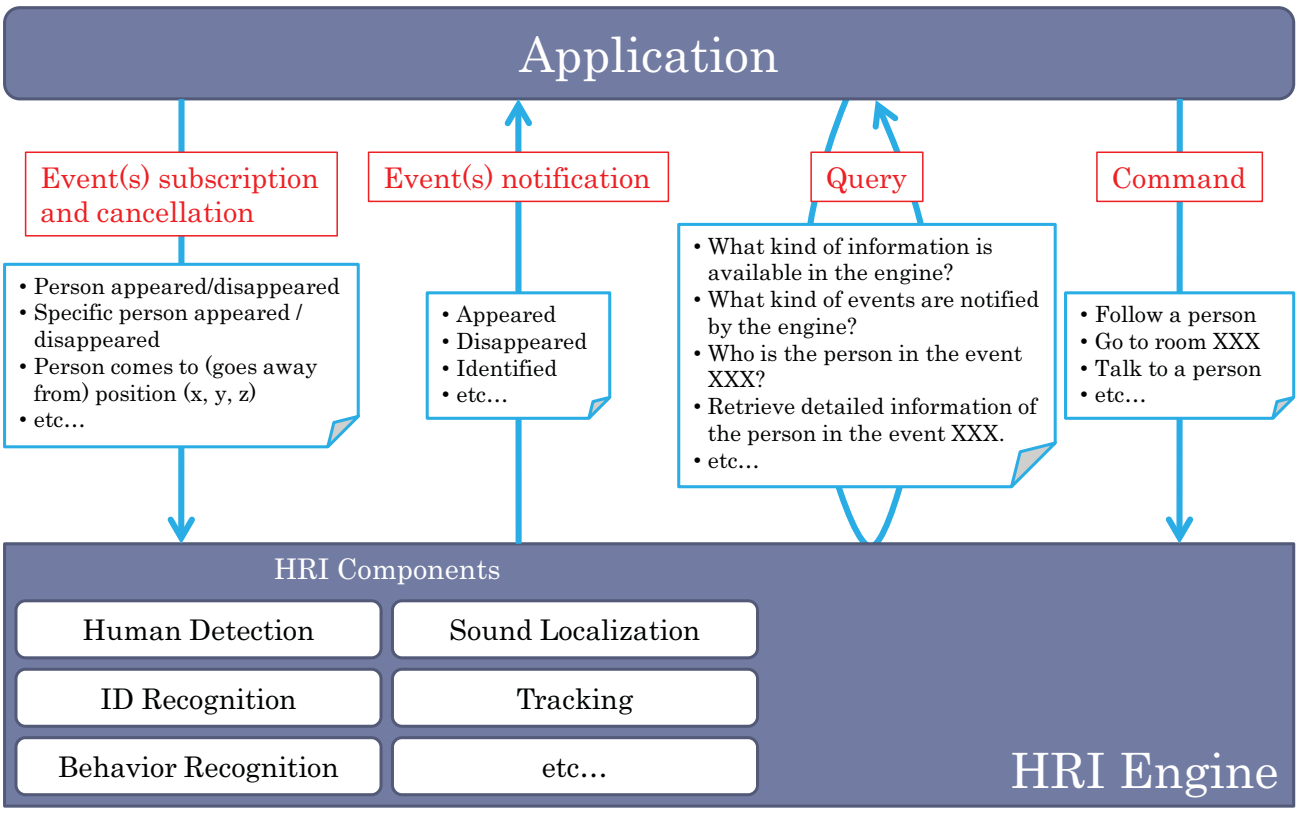
- Oct. 22(Fri) - 23(Sat) for WG 7/Study Group meeting
- Oct. 25(Mon) - 26(Tue) for WG 7
- Oct. 27(Wed) for WG 8
- Oct. 28(Thu) - 29(Fri) for SC 2 plenary meeting

Robotic Interaction Service (RoIS) Framework (rev.2)

Robotic Functional Service WG, Robotics DTF

2010/03/23

Schematic Structure of RoIS Framework



Interaction between HRI Engine and Application

- **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

Difference between Query and Command

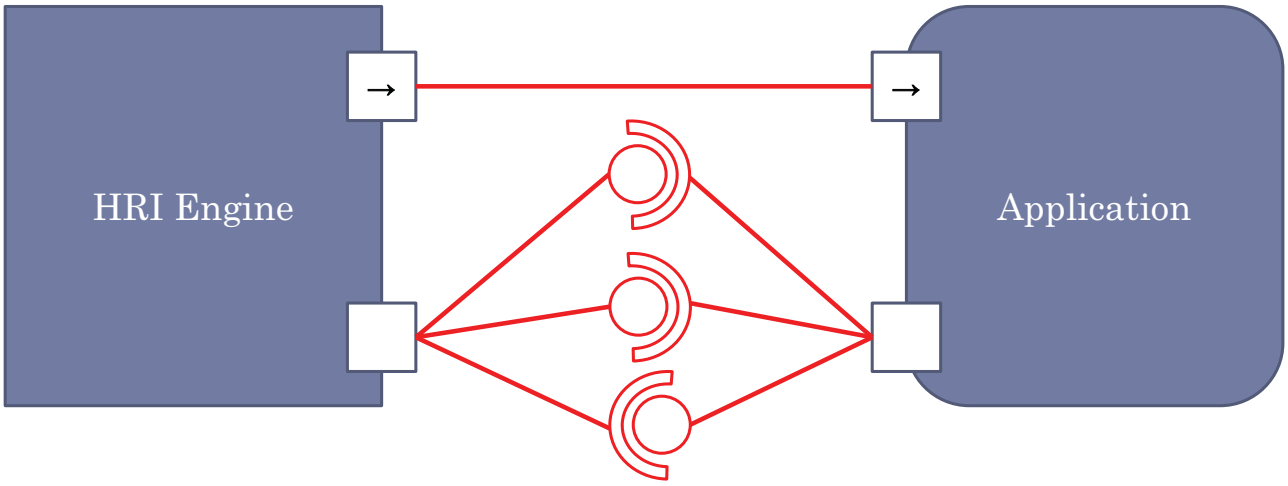
- Query
  - Retrieve information from HRI engine
  - Status of HRI engine does not change before and after each query
- Command
  - Send motion command for operating robot
  - Set/change configuration parameter(s) of HRI engine component(s) and/or the engine itself
  - Status of robot, HRI engine and/or components in the engine will be changed

What should be discussed in Proposal?

- **Data types of human profile**
  - For event notifications, queries, etc.
  - Combination with RLS standard

  Optional
- **Event types of human action**
- **Query types**
  - Details of HRI engine functions
  - HRI engine types
  - etc…
- **Command sets for human-robot interaction**
- **Query interface for accessing detailed data of event source**
- etc…

Scope of RoIS Standard



Mandatory items

- Architectures

Roadmap for Robotics Activities

robotics/2010-03-19

Item	Status	Long Beach, CA Dec-2009	Jacksonville FL Mar-2010	Minneapolis MN Jun-2010	Cambridge CANCEL Sep-2010	Santa Clara CA Dec-2010	? Mar-2011	POC / Comment
Flyer of Robotics-DTF [Publicity Sub-Committee]	Suspended							
Robot Interaction Service (RoIS) Framework RFP [Robotic Functional Services WG]	In Process	discussion	1st review RFP	2nd Review & RFP issue		Initial Submittion		
Dynamic Deployment and Configuration for RTC (DDC) RFP [Robotic Infrastructure WG] in MARS	In Process	1st review RFP	2nd Review & RFP issue			Initial Submittion		Sponsor: MARS
UML profile for Archtecture Framework for Robotics/Unmanned Systems [Robotic Data and Profiles WG]	In Process		discussion	White Peper				
Behavioral States and Instructions RFP [Robotic Functional Services WG]	Planned							
Robotic Map Services RFP [Robotic Functional Services WG]	Planned							IEEE R&A?
Hardware-level Resources: define resource profiles RFP [Profile WG]	Future							
etc…	Future							
Robotics Information Day [Technology Showcase]	Future							
RLS Revision Task Force	In Process	Charter						will go to ISO/TC211

Related Events

Chu-suk
(Special Holidays in Korea)

Robotics-DTF Plenary Meeting
Wrap-up Session

March 23rd, 2010

Jacksonville, FL, USA

Hyatt Regency Jacksonville Riverfront

Document Number

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 [copy of ab/2010-03-02] (Noriaki Ando)
robotics/2010-03-07 Robotic 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)

Document Number (cont.)

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)

AIST

Document Number (cont.)

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 Seung-Woog Jung)

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

Call for volunteer

- Robotic Infrastructure WG Co-Chair

  Beom-Su Seo
  -> Seung-Woog Jung(ETRI)

- Robotic Functional Services WG Co-Chair

  Shuichi Nishio
  -> Miki Sato (ATR)

Organization (from 23rd March, 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)

Robotic Data and Profiles WG
- Bruce Boyes (Systronix, USA)
- Laurent Rioux (Thales)

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

Next Meeting Agenda

Plan A

June 21-25 (Minneapolis, MN , USA)

Monday:

Opening Session (morning)
Robotic Interaction Service (RoIS) Framework RFP 2nd Review (am)
WG activity (pm)

Tuesday:

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

Wednesday:

WG activity follow-up [if necessary]

Thursday:

Robotic Interaction Service (RoIS) Framework RFP Voting (am)

AIST

Plenary Attendee (21 participants)

- Claude Baudoin (Cebe)
- David Martin (DeVivo / SAE AS4)
- David Miller (Boeing)
- Geoffrey Biggs (AIST)
- Hugues VINCENT (Thales)
- Itsuki Noda (AIST)
- Jaeyeong 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)

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

RLS Implementation and Issues

Jae-Yeong Lee, Wonpil Yu
ETRI, Korea

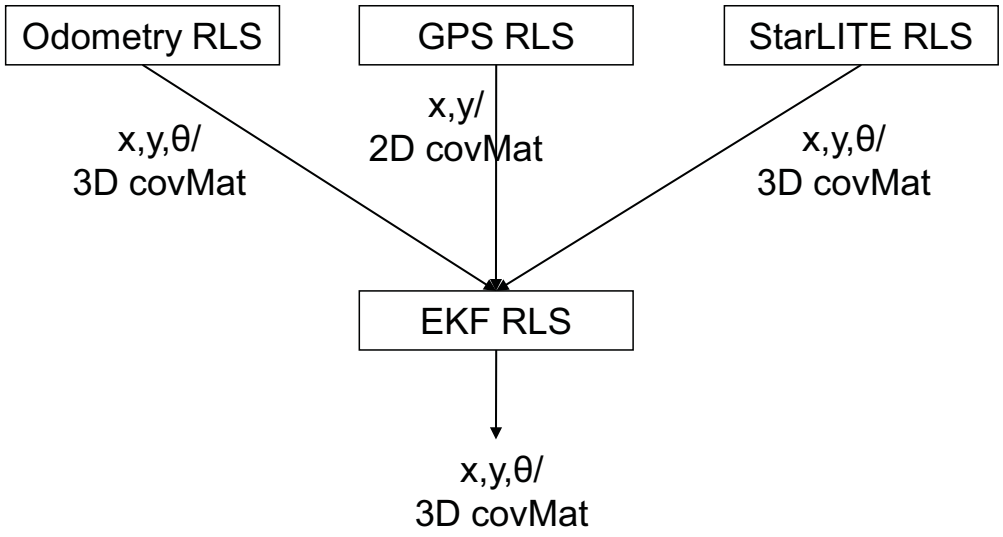
ETRI Electronics and Telecommunications Research Institute

ETRI's Implementation of RLS

- Overview
  - Language: C++
  - Reference Specification: formal/2010-02-03
- Features
  - embed EKF-based sensor fusion algorithm
  - multiple sensors can be connected simultaneously
    - different dimension (2D, 3D)
    - different sensor characteristic (error, update rate)
  - applied & tested on real robot

ETRI Electronics and Telecommunications Research Institute

ETRI's Implementation of RLS



- EKF RLS

```
void CRoLoAppDlg::TestGPS()
{
		ServiceOdometry odometry;
		ServiceGPS gps;
		ServiceEKF service;
		service.connect(&odometry);
		service.connect(&gps);
		service.activate();

		RecordStream record;
		OutStream * stream = service.connect(&record);
		stream->activate();

		FILE * f = fopen("gps.txt", "rt");

		double x,y;    // gps
		double t1,t2,t3; // temp
		double ox, oy, otheta;

		while(fscanf(f, "%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf¥n", &x, &y, &t1, &t2, &t3, &ox, &oy, &otheta) != EOF)
		{
			odometry.SendData(ox, oy, otheta);
			gps.SendData(x, y);
		}

		fclose(f);
}
```

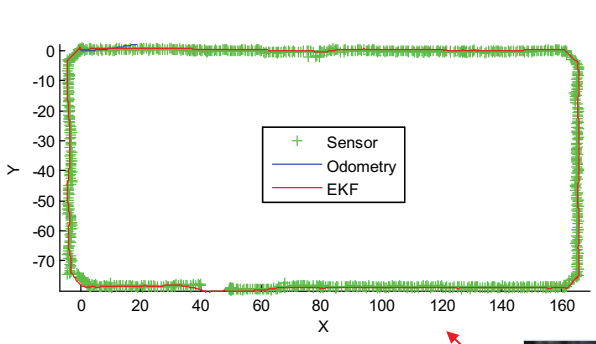
- GPS RLS

```
void ServiceGPS::InitRoLoService()
{
        error_variance = 0.5*0.5;                       // meter^2

        ServiceAbility ability;

        ability.expectedLatency.val = 0.01;             // second
        ability.expectedLatency.def.type.code = "double";
        ability.expectedLatency.def.unit.uomName = "UomTime";

        ability.outStreamAbility.frequency.domain.insert(0);
        ability.outStreamAbility.frequency.domain.insert(1);
        ability.outStreamAbility.frequency.val = 0;

        PositionElementSpecification * posSpec = new PositionElementSpecification();
        posSpec->identifier.code = "GPS";
        posSpec->name.code = "Pos. Elem. Spec. for GPS position";
        posSpec->crs = new StaticRelativeCartesianCRS();
        posSpec->errType.identifier.code = "ET_Gaussian";

        DataSpecification * dataSpec = new DataSpecification();
        dataSpec->identifier.code = "ROMI";
        dataSpec->name.code = "Data Spec. for position of ETRI ROMI mobile robot";
        dataSpec->elemSpecs.push_back(posSpec);

        ability.outStreamAbility.dataSpec.domain.insert(dataSpec);
        ability.outStreamAbility.dataSpec.val = dataSpec;
```

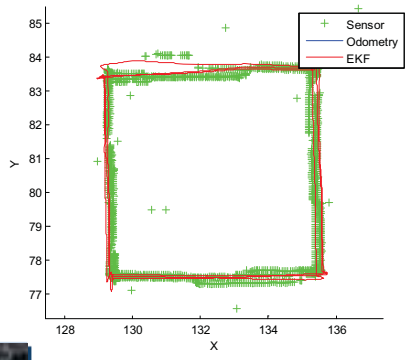

- GPS RLS

```
        setAbility(&ability);

        PositionElement * p = new PositionElement();
        p->pos.numeric = new GM_Position();
        p->pos.numeric->direct.coordinate.push_back(0);
        p->pos.numeric->direct.coordinate.push_back(0);
        p->pos.numeric->direct.dimension = 2;
        Gaussian * err = new Gaussian();
        err->cov.nRow = 2;
        err->cov.nCol = 2;
        err->cov.vals.push_back(error_variance);
        err->cov.vals.push_back(0);
        err->cov.vals.push_back(0);
        err->cov.vals.push_back(error_variance);
        p->err = err;
        data.elems.push_back(p);
}
void ServiceGPS::SendData(double x, double y)
{
        ((PositionElement *)data.elems[0])->pos.numeric->direct.coordinate[0] = x;
        ((PositionElement *)data.elems[0])->pos.numeric->direct.coordinate[1] = y;

        list<OutStream *>::iterator itr;
        for(itr = outStreams.begin(); itr != outStreams.end(); itr++)
        {
                const StreamAbility * ability = (const StreamAbility *)((*itr)->getAbility());
                data.spec = ability->dataSpec.val;
                (*itr)->setData(data);
        }
}
```

- EKF experiment
  - tested on real odometry & location sensor data
  - applied to ROMI which operates 24 hours



GPS

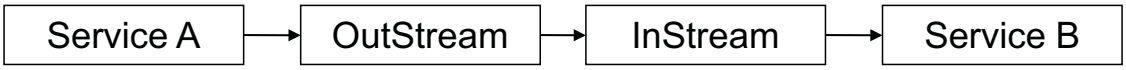


Outlier filtering

ETRI Electronics and Telecommunications Research Institute

Implementation Issues

- InStream & OutStream
  - They are just conceptually different in usage but physically the same
  - The separation makes the implementation difficult



  - InStream may be redundant
  - OutStream also requires setData()
- Stream

ETRI Electronics and Telecommunications Research Institute

Implementation Issues

- Most RoLo classes are hierarchically organized but the interfacing usually is based on the base class type
  - eg) Error PositionElement::err, AttributeBase AttributeSet::attrs

- In C++ Implementation
  - the element types should be a pointer or reference to enable homomorphism
    - InterfaceBase::get/setParameterValues(::std::set<ParameterValueBase *>)
    - AttributeSet::list<AttributeBase *> attrs
    - Data::vector<Element *> elems
    - ...
  - it means that we need dynamic memory allocation/deallocation for data passing, making the implementation very difficult
    - identifying real instance type
    - copying data
  - In ETRI implementation, we introduced clone() method for such classes

- C++ PSM part of RLS Spec. should be corrected into pointer type not to make the developers confusing

Implementation Issues

- Service::connect()
  - difference of two connect() methods are not clearly described
  - developers can be very confused
  - the notations IN service and OUT service is confusing
    - given a relation A->B, which one is IN service?
  - Receiver(IN) / Sender(OUT) would be more clear conceptually

```
InStream * connect(const InStream * target, const OutStream * source);
// establish connection from output stream to input stream
// A->B, B.connect(InStream, A.OutStream)  ?

OutStream * connect(const InStream * source = NULL);
// establish connection from input stream to output stream
// A->B, A.connect(B.InStream) ?
```

Implementation Issues

- In InterfaceBase
  - we can access parameter values in two ways
    - InterfaceBase:getAbility
    - InterfaceBase::getParameterValues
  - InterfaceBase::ability can be StreamAbility or ServiceAbility
    - They have inherited attribute (attrs) from AttributeSet
    - They also have their own class-specific parameters (frequency, expectedLatency, ...)
  - there only is InterfaceBase::getAbility
- Problems
  - we need InterfaceBase::setAbility to initialize Service or Stream
  - It is not clear what the configurable parameters of InterfaceBase::get/setParameterValues are.
    - inherited attribute 'attrs' from AttributeSet
    - class-specific parameters: streamType, frequency, expectedLatency
    - dataFormat, dataSpec, inStreamAbilities, outStreamAbilites
  - Implementation of InterfaceBase::get/setParameterValues is very complex and confusing
    - class-specific parameters also are attribute, but handled separately from 'attrs' of AttributeSet

Implementation Issues

- The usage of ParameterValue class is not clear
  - First, it is hard to access the target Parameter from ParameterValueBase
  - Second, it is not clear whether we should provide a pointer to a real Parameter instance for ParameterValue::param attribute or use a clone.
    - if clone, we should deallocate the memory (who do?)
    - if not clone, we should not deallocate the memory
  - There is a possibility that the values of ParameterValue::val and ParameterValue::param.val are inconsistent. How about ParamterValue class to refer to the ID of Parameter instance not the instance itself?

Implementation Issues

- We defined common data format but not specified how to identify them
- CRS is very important in RLS,
  - but the usage is not clearly described in the Spec.
  - just referred ISO
  - especially, AxisDirection is confusing
  - inconsistent implementation may be possible among developers
- Auto configuration between RoLo Service modules across the network is not supported currently
  - return value of getAbility: variable data length, data type, ...
  - how to interpret the result without sharing of memory address
  - we need to define how to serialize data

Implementation Issues

- In StreamAbility, dataFormat and dataSpec is confusing
  - dataFormat has its own DataSpecification
  - Difference of these two DataSpecifications is not clear. is it just a redundant duplication?
- Minor ones
  - In C++ PSM, Service class should be derived from InterfaceBase class, not from OutStream
  - Argument type of InterfaceBase::get/setParameterValues is 'set' in PIM but 'list' in PSM. It may cause the developers confusing
  - Obligation of AttributeSet::attrs & AttributeSet::def seems not correct (attrs: M, def: O ?)

ETRI Electronics and Telecommunications Research Institute

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:

Letters of Intent due: 2010. Sep.
Submissions due: 2010. Nov

Objective of this RFP

This RFP solicits proposals for the framework and interfaces applied to robotic interaction service.

In particular, the proposal shall provide:

- Description of Robotic Interaction Service (RoIS) architecture.
- Interfaces between robotic service applications and an object which provides functions for human robot interaction.
- Data structure for each interface.

For further details see Chapter 6 of this document.

6.0 Specific Requirements on Proposals

6.1 Problem Statement

Many researchers are developing various robotic interacting service technologies such as,

- speech recognition
- face detection and recognition
- speaker recognition
- user tracking and following
- sound source localization etc.,
-

They agree that RoIS is essential in the breakthrough of robot industry. However, current robots have very limited RoIS capabilities. One of the obvious reasons is low performance of the provided technologies.

Every single algorithm is embodied as a function in a library. Coordination between components is delegated to the application programmers who do not have expertise about the component details.

In a view of application programmers, they do not know the image formats, details in it. Even they do not want to know. Sound formats also the same. They do not understand clearly when and which component should be called. Cooperation between components, it is not an expertise of them.

New model for RoIS technology is needed. Various algorithms cooperate to recognize the environment in an independent process which is called HRI Engine. It is processes that actively gather and manage the information that application programmers need by applying various RoIS components effectively. Application programmers do not need to know the details of the HRI Engine. That is, RoIS capabilities are encapsulated in HRI Engine.

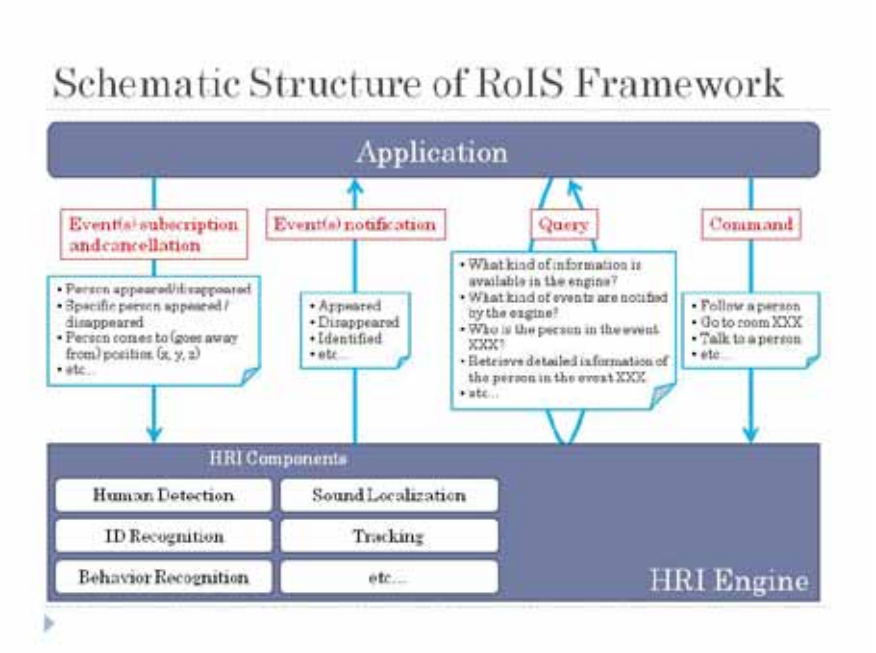


Fig. 1. RoIS Framework

RoIS communicates with HRI engine through EVENT(s) subscription/cancellation, EVENT(s) notification, QUERY and COMMAND.

- **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

Robotic interacting service framework is one of the most fundamental ingredients for truly useful robot systems. Many useful services require the robotic interacting service as a premise. With information regarding users, the robot can provide services customized to specific users, and services can be delivered to appropriate users.

Some examples of useful robotic interacting service are listed as follows;

- Owner recognition
- Intruder detection
- User tracking
- Access control
- Searching watch-list
- Searching missing child
- User preference service
- Human based photo retrieval
- Etc.

6.2 Scope of Proposals Sought

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

It is necessary to consider the followings in the specification of robotic interaction service framework.

(1) The robotic interaction service framework specification shall provide a interface between applications and robotic components.

(2) The robotic interaction service framework specification must be general enough to incorporate various sensors and algorithms.

(3) The robotic interaction service framework specification shall provide the data representation for its external application interface as well as its internal functionalities

- The data representation may includes elements for specifying user such as user identity descriptor format or user identity template, multi-modal data format , input data type, etc.

(4) The robotic interaction service framework specification shall satisfy interoperability and reusability. An robotic interaction service framework implemented by one vendor should be able to be replaced with robotic interaction service framework provided by other vendors with little efforts.

(5) The robotic interaction service framework specification shall provide a minimum set of functionalities to satisfy the following:

- Overall Architecture for robotic interaction service framework shall be defined (diagram or description for overview)
- Interface types between applications and HRI engine

- Event subscription and cancellation
- Event notifications
- Query
- Command

- Data structure for each interface type

6.3 PIM using UML shall be defined.Relationship to Existing OMG Specifications

Submitters shall examine the following OMG specifications for possible benefit:

- Platform Independent Model (PIM) and Platform Specific Model (PSM) for 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]
- Localization 1.0 Beta 1 [dtc/2008-07-01]

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

Proposals shall provide a Platform Independent Model (PIM) and at least one CORBA-specific model of robotic interaction service framework or C++ specific model of robotic interaction service framework. The models shall meet the following requirements.

1. Proposals shall specify a general mechanism for robotic interaction service information of enrollee to be interacting.
   - Overall Architecture for robotic interaction service framework shall be defined (diagram or description for overview)
   - Interface types between applications and HRI engine
     - Event subscription and cancellation
     - Event notifications
     - Query
     - Command
   - Data structure for each interface type
   - PIM using UML shall be defined.

6.6 Optional Requirements

Proposals may specify interfaces for the functionalities listed below.

- Human Profile for robot interaction service
- Error handling for each interface type

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 the way they bring real-time support (cf. section 6.2, point (6)).

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	**23. March. 2010**
RFP placed on OMG document server	**Before 22. May. 2010**
Approval of RFP by Architecture Board *Review by TC*	**23. March. 2010**
TC votes to issue RFP	**23. March. 2010**
LOI to submit to RFP due	**12. Jan. 2009**
Initial Submissions due and placed on OMG document server ("Three week rule")	**By 25. May.2009**
Voter registration closes	
Initial Submission presentations	**22. June.2009**
Preliminary evaluation by TF	
Revised Submissions due and placed on OMG document server ("Three week rule")	**9. Nov.2009**
Revised Submission presentations	**7. Dec.2009**
Final evaluation and selection by TF	**11. Dec.2009**

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

None

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.

Missions for the Modeling in robotics WG

- Bring MDE support for robotics
  - Include:
    - Software engineering
    - Hardware engineering
    - System engineering
- Models interchange
- Ontology for robotics
- To be extend…

IDEAS / SUBJECTS

- MDE / Software:
  - UML profile for RTC
    - Include DDC.
  - UML profile for RLS
- robotics systems modeling language
- Robotics hardware modeling language
  - Include Mechatronics, electronics, …
- Unified Modeling language to design robotics software
- Models for representing the environment
  - Which standards to use to represent the environment, ….
- Robotics specific domain language for dimensions analysis

mars/10-03-05

Errata to revised submission of the RTC Deployment and Dynamic Reconfiguration (DDR) Request For Proposal, ab/10-03-02

This document lists some minor errors and corrections to the revised submission of the RTC Deployment and Dynamic Reconfiguration (DDR) Request For Proposal, mars/10-02-14

Change Overview

1. The name of the specification has been changed to "Dynamic Deployment and Configuration (DDC) of Robotic Technology Components"
2. All the "deployment and dynamic reconfiguration" has been changed to "dynamic deployment and configuration."
3. LOI due and submission due has been corrected according to the revised schedule.
4. Abbreviations mistakes and notation inconsistency have been corrected.
5. In "Objective of this RFP" on p.1, "Methods to …" has been modified to "Ways to …" to avoid misunderstanding for "methods" of specific meaning.
6. In the 6.2, "CORBA IDL" has been corrected to "OMG IDL."
7. In the 6.3.1 Relationship to OMG specifications, all the specifications have been put down with direct URL.
8. In the 6.3.2 Relationship to other OMG Documents and work in progress, references to MARTE have been moved to 6.3.1, because MARTE is already formal specification.
9. In the 6.6 Mandatory Requirements, some expressions of sentences have been modified to make clear the meaning of this RFP's requirements.
10. In the 6.6 Mandatory Requirements, requirements have been divided into two parts. One is "Platform independent deployment and configuration model", other is "Platform independent RTC information model."
11. In the 6.6 Mandatory Requirements, some ambiguous words "meta-model", "data-model" and "platform independent model" have been changed to more clear expression such as "schema" and "services."
12. An optional requirement has been moved from mandatory requirements.
13. 6.8 Evaluation Criteria has been modified.
14. In the 6.10 RFP Timetable, a mistake of the date of initial submission due has been corrected.
15. Some general references in Appendix B have been modified.

Object Management Group

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

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

~~RTC~~ Dynamic Deployment and ~~Dynamic Reconfiguration (DDR~~Configuration (DDC) of Robotic Technology Components~~)~~

~~Initial Draft~~ Request For Proposal Draft

OMG Document: mars/2010-~~02-14~~03-05 (errata)
mars/2010-03-06 (with change bar)
mars/2010-03-07 (convenience document)

Letters of Intent due: ~~XX June~~31 August 2010
Submissions due: ~~23 August~~8 November 2010

Objective of this RFP

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

In particular, the proposal shall provide:

- ~~Descriptions specific to robotics~~Ways to ~~for the deployment of RT components.~~
- ~~Interfaces~~search for and deploy~~ing RT components~~Robotic Technology Components (RTC) into robotic systems at ~~runtime~~run-time.

- WayMethods and interfaces tofor notifying the relevant RT componentRTC instances of environment changes.

- WayMethods and interfaces for to searching for appropriate RT componentRTC instances and dynamically reconfiguringconfigure them.

1.0 Introduction

1.1 Goals of OMG

The Object Management Group (OMG) is the world's largest software consortium with an international membership of vendors, developers, and end users. Established in 1989, its mission is to help computer users solve enterprise integration problems by supplying open, vendor-neutral portability, interoperability and reusability specifications based on Model Driven Architecture (MDA). MDA defines an approach to IT system specification that separates the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform, and provides a set of guidelines for structuring specifications expressed as models. OMG has established numerous widely used standards such as OMG IDL[IDL], CORBA[CORBA], Realtime CORBA [CORBA], GIOP/IIOP[CORBA], UML[UML], MOF[MOF], XMI[XMI] and CWM[CWM] to name a few significant ones.

1.2 Organization of this document

The remainder of this document is organized as follows:

Chapter 2 - *Architectural Context* - background information on OMG's Model Driven Architecture.

Chapter 3 - *Adoption Process* - background information on the OMG specification adoption process.

Chapter 4 - *Instructions for Submitters* - explanation of how to make a submission to this RFP.

Chapter 5 - *General Requirements on Proposals* - requirements and evaluation criteria that apply to all proposals submitted to OMG.

Chapter 6 - *Specific Requirements on Proposals* - problem statement, scope of proposals sought, requirements and optional features, issues to be discussed, evaluation criteria, and timetable that apply specifically to this RFP.

Appendix A – *References and Glossary Specific to this RFP*

Appendix B – General References and Glossary

1.3 Conventions

The key words "**must**", "**must not**", "**required**", "**shall**", "**shall not**", "**should**", "**should not**", "**recommended**", "**may**", and "**optional**" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.4 Contact Information

Questions related to the OMG's technology adoption process may be directed to *omg-process@omg.org*. General questions about this RFP may be sent to *responses@omg.org*.

OMG documents (and information about the OMG in general) can be obtained from the OMG's web site *(http://www.omg.org/)*. OMG documents may also be obtained by contacting OMG at *documents@omg.org*. Templates for RFPs (like this document) and other standard OMG documents can be found at the OMG Template Downloads Page at *http://www.omg.org/technology/template_download.htm*

2.0 Architectural Context

MDA provides a set of guidelines for structuring specifications expressed as models and the mappings between those models. The MDA initiative and the standards that support it allow the same model specifying business system or application functionality and behavior to be realized on multiple platforms. MDA enables different applications to be integrated by explicitly relating their models; this facilitates integration and interoperability and supports system evolution (deployment choices) as platform technologies change. The three primary goals of MDA are portability, interoperability and reusability.

Portability of any subsystem is relative to the subsystems on which it depends. The collection of subsystems that a given subsystem depends upon is often loosely called the *platform*, which supports that subsystem. Portability – and reusability - of such a subsystem is enabled if all the subsystems that it depends upon use standardized interfaces (APIs) and usage patterns.

MDA provides a pattern comprising a portable subsystem that is able to use any one of multiple specific implementations of a platform. This pattern is repeatedly usable in the specification of systems. The five important concepts related to this pattern are:

(1) *Model* – A model is a representation of a part of the function, structure and/or behavior of an application or system. A representation is said to be formal when it is based on a language that has a well-defined form ("syntax"), meaning ("semantics"), and possibly rules of analysis, inference, or proof for its constructs. The syntax may be graphical or textual. The semantics might be defined, more or less formally, in terms of things observed in the world being described (e.g. message sends and replies, object states and state changes, etc.), or by translating higher-level language constructs into other constructs that have a well-defined meaning. The optional rules of inference define what unstated properties you can deduce from the explicit statements in the model. In MDA, a representation that is not formal in this sense is not a model. Thus, a diagram with boxes and lines and arrows that is not supported by a definition of the meaning of a box, and the meaning of a line and of an arrow is not a model—it is just an informal diagram.

(2) *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.

(3) *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.

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

(5) *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. A mapping may be expressed as associations, constraints, rules, templates with parameters that must be assigned during the mapping, or other forms yet to be determined.

For example, in case of CORBA the platform is specified by a set of interfaces and usage patterns that constitute the CORBA Core Specification [CORBA]. The CORBA platform is independent of operating systems and programming languages. The OMG Trading Object Service specification [TOS] (consisting of interface specifications in OMG Interface Definition Language (OMG IDL)) can be considered to be a PIM from the viewpoint of CORBA, because it is independent of operating systems and programming languages. When the IDL to C++ Language Mapping specification is applied to the Trading Service PIM, the

C++-specific result can be considered to be a PSM for the Trading Service, where the platform is the C++ language and the C++ ORB implementation. Thus the IDL to C++ Language Mapping specification [IDLC++] determines the mapping from the Trading Service PIM to the Trading Service PSM.

Note that the Trading Service model expressed in IDL is a PSM relative to the CORBA platform too. This highlights the fact that platform-independence and platform-specificity are relative concepts.

The UML Profile for EDOC specification [EDOC] is another example of the application of various aspects of MDA. It defines a set of modeling constructs that are independent of middleware platforms such as EJB [EJB], CCM [CCM], MQSeries [MQS], etc. A PIM based on the EDOC profile uses the middleware-independent constructs defined by the profile and thus is middleware-independent. In addition, the specification defines formal metamodels for some specific middleware platforms such as EJB, supplementing the already-existing OMG metamodel of CCM (CORBA Component Model). The specification also defines mappings from the EDOC profile to the middleware metamodels. For example, it defines a mapping from the EDOC profile to EJB. The mapping specifications facilitate the transformation of any EDOC-based PIM into a corresponding PSM for any of the specific platforms for which a mapping is specified.

Continuing with this example, one of the PSMs corresponding to the EDOC PIM could be for the CORBA platform. This PSM then potentially constitutes a PIM, corresponding to which there would be implementation language specific PSMs derived via the CORBA language mappings, thus illustrating recursive use of the Platform-PIM-PSM-Mapping pattern.

Note that the EDOC profile can also be considered to be a platform in its own right. Thus, a model expressed via the profile is a PSM relative to the EDOC platform.

An analogous set of concepts apply to Interoperability Protocols wherein there is a PIM of the payload data and a PIM of the interactions that cause the data to find its way from one place to another. These then are realized in specific ways for specific platforms in the corresponding PSMs.

Analogously, in case of databases there could be a PIM of the data (say using the Relational Data Model), and corresponding PSMs specifying how the data is actually represented on a storage medium based on some particular data storage paradigm etc., and a mapping from the PIM to each PSM.

OMG adopts standard specifications of models that exploit the MDA pattern to facilitate portability, interoperability and reusability, either through ab initio

development of standards or by reference to existing standards. Some examples of OMG adopted specifications are:

1. *Languages* – e.g. IDL for interface specification, UML for model specification, OCL for constraint specification, etc.

(6) Mappings – e.g. Mapping of OMG IDL to specific implementation languages (CORBA PIM to Implementation Language PSMs), UML Profile for EDOC (PIM) to CCM (CORBA PSM) and EJB (Java PSM), CORBA (PSM) to COM (PSM) etc.

(7) *Services* – e.g. Naming Service [NS], Transaction Service [OTS], Security Service [SEC], Trading Object Service [TOS] etc.

(8) *Platforms* – e.g. CORBA [CORBA].

(9) *Protocols* – e.g. GIOP/IIOP [CORBA] (both structure and exchange protocol), XML Metadata Interchange [XMI] (structure specification usable as payload on multiple exchange protocols).

(10) *Domain Specific Standards* – e.g. Data Acquisition from Industrial Systems (Manufacturing) [DAIS], General Ledger Specification (Finance) [GLS], Air Traffic Control (Transportation) [ATC], Gene Expression (Life Science Research) [GE], Personal Identification Service (Healthcare) [PIDS], etc.

For an introduction to MDA, see [MDAa]. For a discourse on the details of MDA please refer to [MDAc]. To see an example of the application of MDA see [MDAb]. For general information on MDA, see [MDAd].

Object Management Architecture (OMA) is a distributed object computing platform architecture within MDA that is related to ISO's Reference Model of Open Distributed Processing RM-ODP[RM-ODP]. CORBA and any extensions to it are based on OMA. For information on OMA see [OMA].

3.0 Adoption Process

3.1 Introduction

OMG adopts specifications by explicit vote on a technology-by-technology basis. The specifications selected each satisfy the architectural vision of MDA. OMG bases its decisions on both business and technical considerations. Once a specification adoption is finalized by OMG, it is made available for use by both OMG members and non-members alike.

Request for Proposals (RFP) are issued by a *Technology Committee* (TC), typically upon the recommendation of a *Task Force* (TF) and duly endorsed by the *Architecture Board* (AB).

Submissions to RFPs are evaluated by the TF that initiated the RFP. Selected specifications are *recommended* to the parent TC after being *reviewed* for technical merit and consistency with MDA and other adopted specifications and *endorsed* by the AB. The parent TC of the initiating TF then votes to *recommend adoption* to the OMG Board of Directors (BoD). The BoD acts on the recommendation to complete the adoption process.

For more detailed information on the adoption process see the *Policies and Procedures of the OMG Technical Process* [P&P] and the *OMG Hitchhiker's Guide* [Guide]. In case of any inconsistency between this document and the [P&P] in all cases the [P&P] shall prevail.

3.2 Steps in the Adoption Process

A TF, its parent TC, the AB and the Board of Directors participate in a collaborative process, which typically takes the following form:

- *Development and Issuance of RFP*

  RFPs are drafted by one or more OMG members who are interested in the adoption of a standard in some specific area. The draft RFP is presented to an appropriate TF, based on its subject area, for approval and recommendation to issue. The TF and the AB provide guidance to the drafters of the RFP. When the TF and the AB are satisfied that the RFP is appropriate and ready for issuance, the TF recommends issuance to its parent TC, and the AB endorses the recommendation. The TC then acts on the recommendation and issues the RFP.

- *Letter of Intent (LOI)*

  A Letter of Intent (LOI) must be submitted to the OMG signed by an officer of the member organization which intends to respond to the RFP, confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements. (See section 4.3 for more information.). In order to respond to an RFP the organization must be a member of the TC that issued the RFP.

- *Voter Registration*

  Interested OMG members, other than Trial, Press and Analyst members, may participate in specification selection votes in the TF for an RFP. They may need to register to do so, if so stated in the RFP. Registration ends on a

specified date, 6 or more weeks after the announcement of the registration period. The registration closure date is typically around the time of initial submissions. Member organizations that have submitted an LOI are automatically registered to vote.

- *Initial Submissions*

  Initial Submissions are due by a specified deadline. Submitters normally present their proposals at the first meeting of the TF after the deadline. Initial Submissions are expected to be complete enough to provide insight on the technical directions and content of the proposals.

- *Revision Phase*

  During this time submitters have the opportunity to revise their Submissions, if they so choose.

- *Revised Submissions*

  Revised Submissions are due by a specified deadline. Submitters again normally present their proposals at the next meeting of the TF after the deadline. (Note that there may be more than one Revised Submission deadline. The decision to set new Revised Submission deadlines is made by the registered voters for that RFP.)

- *Selection Votes*

  When the registered voters for the RFP believe that they sufficiently understand the relative merits of the Revised Submissions, a selection vote is taken. The result of this selection vote is a recommendation for adoption to the TC. The AB reviews the proposal for MDA compliance and technical merit. An endorsement from the AB moves the voting process into the issuing Technology Committee. An eight-week voting period ensues in which the TC votes to recommend adoption to the OMG Board of Directors (BoD). The final vote, the vote to adopt, is taken by the BoD and is based on technical merit as well as business qualifications. The resulting draft standard is called the *Alpha Specification*.

- Business Committee Questionnaire

  The submitting members whose proposal is recommended for adoption need to submit their response to the BoD Business Committee Questionnaire [BCQ] detailing how they plan to make use of and/or make the resulting standard available in products. If no organization commits to make use of the standard, then the BoD will typically not act on the recommendation to adopt the standard - so it is very important to fulfill this requirement.

- Finalization

  A Finalization Task Force (FTF) is chartered by the TC that issued the RFP, to prepare an Alpha submission for publishing as a Formal (i.e. publicly available) specification, by fixing any problems that are reported by early users of the specification. Upon completion of its activity the FTF recommends adoption of the resulting Beta (draft) specification. The parent TC acts on the recommendation and recommends adoption to the BoD. OMG Technical Editors produce the Formal Specification document based on this Beta Specification.

- Revision

  A Revision Task Force (RTF) is normally chartered by a TC, after the FTF completes its work, to manage issues filed against the Formal Specification by implementers and users. The output of the RTF is a Beta specification reflecting minor technical changes, which the TC and Board will usually approve for adoption as the next version of the Formal Specification.

3.3 Goals of the evaluation

The primary goals of the TF evaluation are to:

- Provide a fair and open process
- Facilitate critical review of the submissions by members of OMG
- Provide feedback to submitters enabling them to address concerns in their revised submissions
- Build consensus on acceptable solutions
- Enable voting members to make an informed selection decision

Submitters are expected to actively contribute to the evaluation process.

4.0 Instructions for Submitters

4.1 OMG Membership

To submit to an RFP issued by the Platform Technology Committee the submitter or submitters must be either Platform or Contributing members on the date of the submission deadline, while for Domain Technology RFPs the submitter or submitters must be either Contributing or Domain members. Submitters sometimes choose to name other organizations that support a submission in some way; however, this has no formal status within the OMG

process, and for OMG's purposes confers neither duties nor privileges on the organizations thus named.

4.2 Submission Effort

An RFP submission may require significant effort in terms of document preparation, presentations to the issuing TF, and participation in the TF evaluation process. Several staff months of effort might be necessary. OMG is unable to reimburse submitters for any costs in conjunction with their submissions to this RFP.

4.3 Letter of Intent

A Letter of Intent (LOI) must be submitted to the OMG Business Committee signed by an officer of the submitting 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. These terms, conditions, and requirements are defined in the *Business Committee RFP Attachment* and are reproduced verbatim in section 4.4 below.

The LOI should designate a single contact point within the submitting organization for receipt of all subsequent information regarding this RFP and the submission. The name of this contact will be made available to all OMG members. The LOI is typically due 60 days before the deadline for initial submissions. LOIs must be sent by fax or paper mail to the "RFP Submissions Desk" at the main OMG address shown on the first page of this RFP.

Here is a suggested template for the Letter of Intent:

This letter confirms the intent of <organization required> (the organization) to submit a response to the OMG <RFP name required> RFP. We will grant OMG and its members the right to copy our response for review purposes as specified in section 4.7 of the RFP. Should our response be adopted by OMG we will comply with the OMG Business Committee terms set out in section 4.4 of the RFP and in document omg/06-03-02.

<contact name and details required> will be responsible for liaison with OMG regarding this RFP response.

The signatory below is an officer of the organization and has the approval and authority to make this commitment on behalf of the organization.

<signature required>

4.4 Business Committee RFP Attachment

This section contains the text of the Business Committee RFP attachment concerning commercial availability requirements placed on submissions. This attachment is available separately as an OMG document omg/06-03-02.

Commercial considerations in OMG technology adoption

A1 Introduction

OMG wishes to encourage rapid commercial adoption of the specifications it publishes. To this end, there must be neither technical, legal nor commercial obstacles to their implementation. Freedom from the first is largely judged through technical review by the relevant OMG Technology Committees; the second two are the responsibility of the OMG Business Committee. The BC also looks for evidence of a commitment by a submitter to the commercial success of products based on the submission.

A2 Business Committee evaluation criteria

A2.1 Viable to implement across platforms

While it is understood that final candidate OMG submissions often combine technologies before they have all been implemented in one system, the Business Committee nevertheless wishes to see evidence that each major feature has been implemented, preferably more than once, and by separate organisations. Pre-product implementations are acceptable. Since use of OMG specifications should not be dependant on any one platform, cross-platform availability and interoperability of implementations should be also be demonstrated.

A2.2 Commercial availability

In addition to demonstrating the existence of implementations of the specification, the submitter must also show that products based on the specification are commercially available, or will be within 12 months of the date when the specification was recommended for adoption by the appropriate Task Force. Proof of intent to ship product within 12 months might include:

- *A public product announcement with a shipping date within the time limit.*
- *Demonstration of a prototype implementation and accompanying draft user documentation.*

Alternatively, and at the Business Committee's discretion, submissions may be adopted where the submitter is not a commercial software provider, and therefore will not make implementations commercially available. However, in this case the BC will require concrete evidence of two or more independent implementations of the specification being used by end- user organisations as part of their businesses. Regardless of which requirement is in use, the submitter must inform the OMG of completion of the implementations when commercially available.

A2.3 Access to Intellectual Property Rights

OMG will not adopt a specification if OMG is aware of any submitter, member or third party which holds a patent, copyright or other intellectual property right (collectively referred to in this policy statement as "IPR") which might be infringed by implementation or recommendation of such specification, unless OMG believes that such IPR owner will grant a license to organisations (whether OMG members or not) on non-discriminatory and commercially reasonable terms which wish to make use of the specification. Accordingly, the submitter must certify that it is not aware of any claim that the specification infringes any IPR of a third party or that it is aware and believes that an appropriate non-discriminatory license is available from that third party. Except for this certification, the submitter will not be required to make any other warranty, and specifications will be offered by OMG for use "as is". If the submitter owns IPR to which an use of a specification based upon its submission would necessarily be subject, it must certify to the Business Committee that it will make a suitable license available to any user on non- discriminatory and commercially reasonable terms, to permit development and commercialisation of an implementation that includes such IPR.

It is the goal of the OMG to make all of its technology available with as few impediments and disincentives to adoption as possible, and therefore OMG strongly encourages the submission of technology as to which royalty-free licenses will be available. However, in all events, the submitter shall also certify that any necessary licence will be made available on commercially reasonable, non-discriminatory terms. The submitter is responsible for disclosing in detail all known restrictions, placed either by the submitter or, if known, others, on technology necessary for any use of the specification.

A2.4 Publication of the specification

Should the submission be adopted, the submitter must grant OMG (and its sublicensees) a world- wide, royalty-free licence to edit, store, duplicate and distribute both the specification and works derived from it (such as revisions and teaching materials). This requirement applies only to the written specification, not to any implementation of it.

A2.5 Continuing support

The submitter must show a commitment to continue supporting the technology underlying the specification after OMG adoption, for instance by showing the BC development plans for future revisions, enhancement or maintenance.

4.5 Responding to RFP items

4.5.1 Complete proposals

A submission must propose full specifications for all of the relevant requirements detailed in Chapter 6 of this RFP. Submissions that do not present complete proposals may be at a disadvantage.

Submitters are highly encouraged to propose solutions to any optional requirements enumerated in Chapter 6.

4.5.2 Additional specifications

Submissions may include additional specifications for items not covered by the RFP that they believe to be necessary and integral to their proposal. Information on these additional items should be clearly distinguished.

Submitters must give a detailed rationale as to why these specifications should also be considered for adoption. However submitters should note that a TF is unlikely to consider additional items that are already on the roadmap of an OMG TF, since this would pre-empt the normal adoption process.

4.5.3 Alternative approaches

Submitters may provide alternative RFP item definitions, categorizations, and groupings so long as the rationale for doing so is clearly stated. Equally, submitters may provide alternative models for how items are provided if there are compelling technological reasons for a different approach.

4.6 Confidential and Proprietary Information

The OMG specification adoption process is an open process. Responses to this RFP become public documents of the OMG and are available to members and non-members alike for perusal. No confidential or proprietary information of any kind will be accepted in a submission to this RFP.

4.7 Copyright Waiver

Every submission document must contain: (i) a waiver of copyright for unlimited duplication by the OMG, and (ii) a limited waiver of copyright that allows each OMG member to make up to fifty (50) copies of the document for review purposes only. See Section 4.9.2 for recommended language.

4.8 Proof of Concept

Submissions must include a "proof of concept" statement, explaining how the submitted specifications have been demonstrated to be technically viable. The technical viability has to do with the state of development and maturity of the technology on which a submission is based. This is not the same as commercial availability. Proof of concept statements can contain any information deemed relevant by the submitter; for example:

> "This specification has completed the design phase and is in the process of being prototyped."
>
> "An implementation of this specification has been in beta-test for 4 months."
>
> "A named product (with a specified customer base) is a realization of this specification."

It is incumbent upon submitters to demonstrate the technical viability of their proposal to the satisfaction of the TF managing the evaluation process. OMG will favor proposals based on technology for which sufficient relevant experience has been gained.

4.9 Format of RFP Submissions

This section presents the structure of a submission in response to an RFP. *All submissions* must contain the elements itemized in section 4.9.2 below before they can be accepted as a valid response for evaluation or a vote can be taken to recommend for adoption.

4.9.1 General

- Submissions that are concise and easy to read will inevitably receive more consideration.
- Submitted documentation should be confined to that directly relevant to the items requested in the RFP. If this is not practical, submitters must make clear what portion of the documentation pertains directly to the RFP and what portion does not.

- The key words "**must**", "**must not**", "**required**", "**shall**", "**shall not**", "**should**", "**should not**", "**recommended**", "**may**", and "**optional**" shall be used in the submissions with the meanings as described in RFC 2119 [RFC2119].

4.9.2 Required Outline

A three-part structure for submissions is required. Part I is non-normative, providing information relevant to the evaluation of the proposed specification. Part II is normative, representing the proposed specification. Specific sections like Appendices may be explicitly identified as non-normative in Part II. Part III is normative specifying changes that must be made to previously adopted specifications in order to be able to implement the specification proposed in Part II.

PART I

- •A cover page carrying the following information (a template for this is available [Inventory]):
  - The full name of the submission
  - The primary contact for the submission
  - The acronym proposed for the specification (e.g. UML, CORBA)
  - The name and document number of the RFP to which this is a response
  - The document number of the main submission document
  - An inventory of all accompanying documents, with OMG document number, short description, a URL where appropriate, and whether they are normative.
- List of OMG members making the submission (see 4.1) listing exactly which members are making the submission, so that submitters can be matched with LOI responders and their current eligibility can be verified.
- Copyright waiver (see 4.7), in a form acceptable to the OMG.

  One acceptable form is:

  "Each of the entities listed above: (i) grants to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version, and (ii) grants to each member of the OMG a nonexclusive, royalty-free, paid up, worldwide license to make up to fifty (50) copies of this document for internal review purposes only and not

for distribution, and (iii) has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used any OMG specification that may be based hereon or having conformed any computer software to such specification."

If you wish to use some other form you must get it approved by the OMG legal counsel before using it in a submission.

- For each member making the submission, an individual contact point who is authorized by the member to officially state the member's position relative to the submission, including matters related to copyright ownership, etc. (see 4.3)
- Overview or guide to the material in the submission
- Overall design rationale (if appropriate)
- Statement of proof of concept (see 4.8)
- Resolution of RFP requirements and requests

  Explain how the proposal satisfies the specific requirements and (if applicable) requests stated in Chapter 6. References to supporting material in Part II should be given.

  In addition, if the proposal does not satisfy any of the general requirements stated in Chapter 5, provide a detailed rationale.

- Responses to RFP issues to be discussed

  Discuss each of the "Issues To Be Discussed" identified in Chapter 6.

PART II

The contents of this part should be structured based on the template found in [FORMS] and should contain the following elements as per the instructions in the template document cited above:

- Scope of the proposed specification
- Proposed conformance criteria

  Submissions should propose appropriate conformance criteria for implementations.

- Proposed normative references

  Submissions should provide a list of the normative references that are used by the proposed specification

- Proposed list of terms and definitions

  Submissions should provide a list of terms that are used in the proposed specification with their definitions.

- Proposed list of symbols

  Submissions should provide a list of special symbols that are used in the proposed specification together with their significance

- Proposed specification

PART III

- Changes or extensions required to existing OMG specifications

  Submissions must include a full specification of any changes or extensions required to existing OMG specifications. This should be in a form that enables "mechanical" section-by-section revision of the existing specification.

4.10 How to Submit

Submitters should send an electronic version of their submission to the *RFP Submissions Desk* (*omg-documents@omg.org*) at OMG Headquarters by 5:00 PM U.S. Eastern Standard Time (22:00 GMT) on the day of the Initial and Revised Submission deadlines. Acceptable formats are Adobe FrameMaker source, ODF (ISO/IEC 26300), OASIS Darwin Information Typing Architecture (DITA) or OASIS DocBook 4.x (or later).

Submitters should make sure they receive electronic or voice confirmation of the successful receipt of their submission. Submitters should be prepared to send a single hardcopy version of their submission, if requested by OMG staff, to the attention of the "RFP Submissions Desk" at the main OMG address shown on the first page of this RFP.

5.0 General Requirements on Proposals

5.1 Requirements

5.1.1 Submitters are encouraged to express models using OMG modeling languages such as UML, MOF, CWM and SPEM (subject to any further constraints on the types of the models and modeling technologies specified in Chapter 6 of this RFP). Submissions containing models expressed via OMG modeling languages shall be accompanied by an OMG XMI [XMI] representation of the models

(including a machine-readable copy). A best effort should be made to provide an OMG XMI representation even in those cases where models are expressed via non-OMG modeling languages.

5.1.2 Chapter 6 of this RFP specifies whether PIM(s), PSM(s), or both are being solicited. If proposals specify a PIM and corresponding PSM(s), then the rules specifying the mapping(s) between the PIM and PSM(s) shall either be identified by reference to a standard mapping or specified in the proposal. In order to allow possible inconsistencies in a proposal to be resolved later, proposals shall identify whether the mapping technique or the resulting PSM(s) are to be considered normative.

5.1.3 Proposals shall be *precise* and *functionally complete*. All relevant assumptions and context required for implementing the specification shall be provided.

5.1.4 Proposals shall specify *conformance criteria* that clearly state what features all implementations must support and which features (if any) may *optionally* be supported.

5.1.5 Proposals shall *reuse* existing OMG and other standard specifications in preference to defining new models to specify similar functionality.

5.1.6 Proposals shall justify and fully specify any *changes or extensions* required to existing OMG specifications. In general, OMG favors proposals that are *upwards compatible* with existing standards and that minimize changes and extensions to existing specifications.

5.1.7 Proposals shall factor out functionality that could be used in different contexts and specify their models, interfaces, etc. separately. Such *minimalism* fosters re-use and avoids functional duplication.

5.1.8 Proposals shall use or depend on other specifications only where it is actually necessary. While re-use of existing specifications to avoid duplication will be encouraged, proposals should avoid gratuitous use.

5.1.9 Proposals shall be *compatible* with and *usable* with existing specifications from OMG and other standards bodies, as appropriate. Separate specifications offering distinct functionality should be usable together where it makes sense to do so.

5.1.10 Proposals shall preserve maximum *implementation flexibility*. Implementation descriptions should not be included and proposals shall not constrain implementations any more than is necessary to promote interoperability.

5.1.11 Proposals shall allow *independent implementations* that are *substitutable* and *interoperable*. An implementation should be replaceable by an alternative implementation without requiring changes to any client.

5.1.12 Proposals shall be compatible with the architecture for system distribution defined in ISO's Reference Model of Open Distributed Processing [RM-ODP]. Where such compatibility is not achieved, or is not appropriate, the response to the RFP must include reasons why compatibility is not appropriate and an outline of any plans to achieve such compatibility in the future.

5.1.13 In order to demonstrate that the specification proposed in response to this RFP can be made secure in environments requiring security, answers to the following questions shall be provided:

- What, if any, are the security sensitive elements that are introduced by the proposal?
- Which accesses to security-sensitive elements must be subject to security policy control?
- Does the proposed service or facility need to be security aware?
- What default policies (e.g., for authentication, audit, authorization, message protection etc.) should be applied to the security sensitive elements introduced by the proposal? Of what security considerations must the implementers of your proposal be aware?

The OMG has adopted several specifications, which cover different aspects of security and provide useful resources in formulating responses. [CSIV2] [SEC] [RAD].

5.1.14 Proposals shall specify the degree of internationalization support that they provide. The degrees of support are as follows:

a) Uncategorized: Internationalization has not been considered.

b) Specific to <region name>: The proposal supports the customs of the specified region only, and is not guaranteed to support the customs of any other region. Any fault or error caused by requesting the services outside of a context in which the customs of the specified region are being consistently followed is the responsibility of the requester.

c) Specific to <multiple region names>: The proposal supports the customs of the specified regions only, and is not guaranteed to support the customs of any other regions. Any fault or error caused by requesting the services

outside of a context in which the customs of at least one of the specified regions are being consistently followed is the responsibility of the requester.

d) Explicitly not specific to <region(s) name>: The proposal does not support the customs of the specified region(s). Any fault or error caused by requesting the services in a context in which the customs of the specified region(s) are being followed is the responsibility of the requester.

5.2 Evaluation criteria

Although the OMG adopts model-based specifications and not implementations of those specifications, the technical viability of implementations will be taken into account during the evaluation process. The following criteria will be used:

5.2.1 Performance

Potential implementation trade-offs for performance will be considered.

5.2.2 Portability

The ease of implementation on a variety of systems and software platforms will be considered.

5.2.3 Securability

The answer to questions in section 5.1.13 shall be taken into consideration to ascertain that an implementation of the proposal is securable in an environment requiring security.

5.2.4 Conformance: Inspectability and Testability

The adequacy of proposed specifications for the purposes of conformance inspection and testing will be considered. Specifications should provide sufficient constraints on interfaces and implementation characteristics to ensure that conformance can be unambiguously assessed through both manual inspection and automated testing.

5.2.5 Standardized Metadata

Where proposals incorporate metadata specifications, usage of OMG standard XMI metadata [XMI] representations must be provided as this allows specifications to be easily interchanged between XMI compliant tools and applications. Since use of XML (including XMI and XML/Value [XML/Value])

is evolving rapidly, the use of industry specific XML vocabularies (which may not be XMI compliant) is acceptable where justified.

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 ~~dynamic reconfiguration~~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.

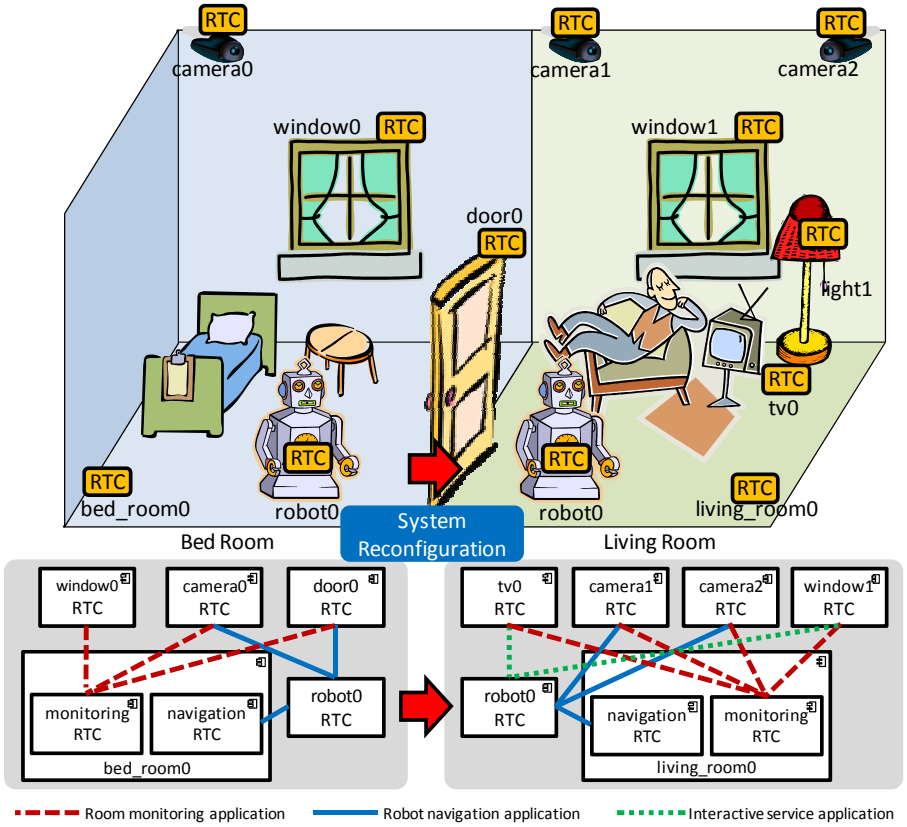


Figure 1 Typical robotic application environment

In order to address functionality of dynamic deployment and ~~dynamic reconfiguration~~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.

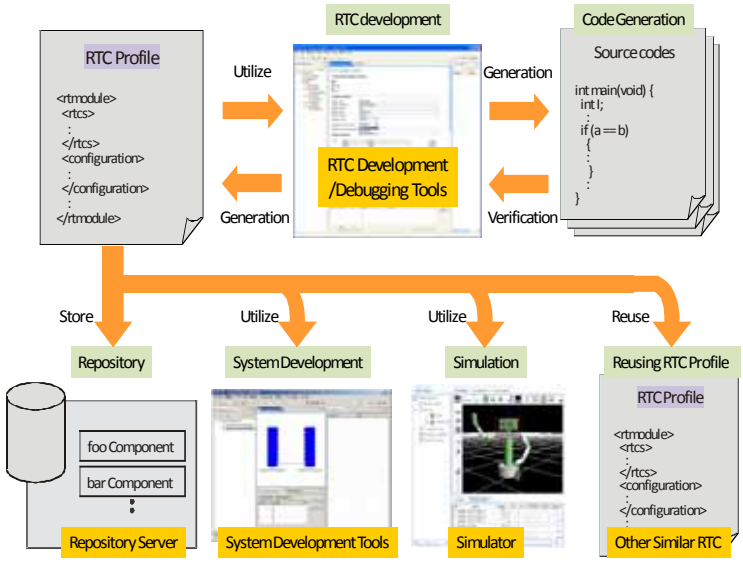


Figure 2 Use of the RTC Profile

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.

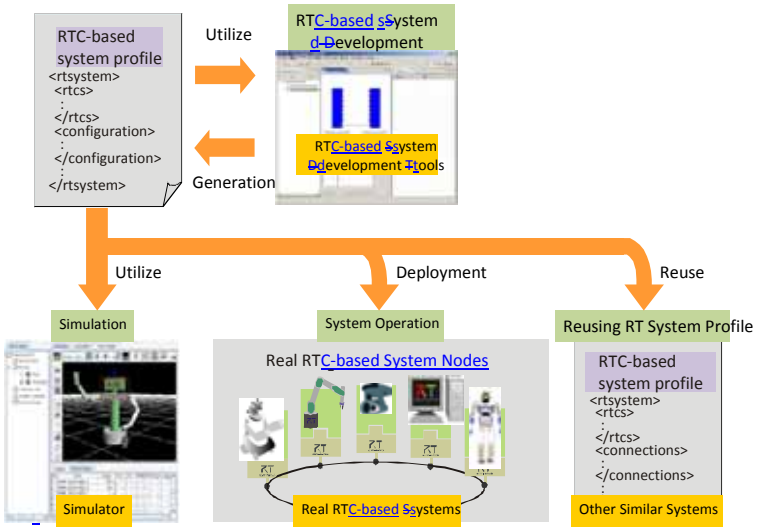


Figure 3 Use of RTC-based system profile

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 ~~reconfiguration~~configuration commonly results from the impairment of the participating component instances and/or changes in the robot location. To support such ~~reconfiguration~~configuration, the robot application (or system) needs to be notified whenever the situation changes. Since not all changes require ~~reconfiguration~~configuration, it must be possible to specify the specific environment changes that trigger ~~reconfiguration.~~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 (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 ~~reconfigure~~configure RTCs at runtime.

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

To use D&C in the robotics domain and expand RTC, the RFP proposes the specifications for the dynamic deployment and ~~dynamic reconfiguration~~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 ~~dynamic reconfiguration~~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 ~~CORBA~~OMG IDL (Interface Definition Language) and XML (eXtensible Mark-up Language).

The proposed specification shall provide functionality for component deployment and dynamic system ~~reconfiguration~~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.~~2~~3 [formal/2009-02-04] http://www.omg.org/spec/UML/2.3/Infrastructure/PDF/
- Unified Modeling Language: Superstructure Version 2.~~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

- ~~UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded systems, beta 3 - convenience document with change bars [ptc/09-05-13]~~

- ~~MARTE model library XMI file [ptc/09-05-16]~~

- ~~MARTE Profile XMI file [ptc/09-05-15]~~

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

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

~~6.5.1 Proposals shall specify a meta-model for the description of component meta-information necessary to support automatic searching and~~

~~comparing of RT components in component repositories and in the run-time system.~~

Platform independent deployment and configuration model

~~6.5.2~~ Proposals shall specify ~~a meta-model for the description of RTC services~~interfaces~~, their compatibility criteria, and deployment requirements.~~

~~6.5.3 Proposal shall specify a platform independent model for information service to identify to locate deployed RTCs available for utilization by the requesting robot.~~

~~6.5.4 Proposals shall specify a data-model for a component information registry using the meta-model requested in requirements 6.5.1 and 6.5.2. The proposal shall also specify a query mechanism for this repository.~~

~~6.5.5~~6.5.1 ~~Proposals shall specify a platform independent model for~~ for dynamic ~~RTC~~ configuration and deployment~~, which~~ of RTCs.

~~a) allows an efficient~~Proposal shall specify means to initiate RTC configuration ~~of RTCs.~~

~~6.5.6~~6.5.2 ~~initiates reconfiguration~~ based on external and/or internal events. A capability for event filtering shall be provided.

~~b) supports coordinated reconfiguration of multiple robot systems to allow performing coordinated tasks.~~

~~c) defines a service interface for the deployment process.~~

~~6.5.7~~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].

Platform independent RTC information model

6.5.4 Proposals shall provide a schema describing RTC characteristics.

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.

6.6 Optional Requirements

~~None~~

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 demonstrate its feasibility by using a specific application based on the proposed model.~~

~~6.7.2 Proposals shall demonstrate its applicability to existing technology such as the RTC specification [RTC].~~

~~6.7.3 Proposals shall discuss simplicity of implementation.~~

~~6.7.4~~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 ~~EJB,~~ CCM [CCM]~~-~~, SDRP ~~SCA~~[SDRP], D&C [D&C] ~~-~~and other well-known component models.

~~6.7.5 Proposals shall discuss the possibility of providing a standard mechanism for advertising and querying component instances and receiving change notifications~~

~~6.7.6~~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].

~~6.7.7~~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

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

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, 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, 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 ~~6th~~ 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] 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.

Object Management Group

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

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

~~RTC~~ Dynamic Deployment and ~~Dynamic Reconfiguration (DDR~~Configuration (DDC) of Robotic Technology Components~~)~~

~~Initial Draft~~ Request For Proposal Draft

OMG Document: mars/2010-~~02-14~~03-05 (errata)
mars/2010-03-06 (with change bar)
mars/2010-03-07 (convenience document)

Letters of Intent due: ~~XX June~~31 August 2010
Submissions due: ~~23 August~~8 November 2010

Objective of this RFP

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

In particular, the proposal shall provide:

- ~~Descriptions specific to robotics~~Ways to ~~for the deployment of RT components.~~
- ~~Interfaces~~search for and deploy~~ing RT components~~Robotic Technology Components (RTC) into robotic systems at ~~runtime~~run-time.

- WayMethods and interfaces tofor notifying the relevant RT componentRTC instances of environment changes.

- WayMethods and interfaces for to searching for appropriate RT componentRTC instances and dynamically reconfiguringconfigure them.

1.0 Introduction

1.1 Goals of OMG

The Object Management Group (OMG) is the world's largest software consortium with an international membership of vendors, developers, and end users. Established in 1989, its mission is to help computer users solve enterprise integration problems by supplying open, vendor-neutral portability, interoperability and reusability specifications based on Model Driven Architecture (MDA). MDA defines an approach to IT system specification that separates the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform, and provides a set of guidelines for structuring specifications expressed as models. OMG has established numerous widely used standards such as OMG IDL[IDL], CORBA[CORBA], Realtime CORBA [CORBA], GIOP/IIOP[CORBA], UML[UML], MOF[MOF], XMI[XMI] and CWM[CWM] to name a few significant ones.

1.2 Organization of this document

The remainder of this document is organized as follows:

Chapter 2 - *Architectural Context* - background information on OMG's Model Driven Architecture.

Chapter 3 - *Adoption Process* - background information on the OMG specification adoption process.

Chapter 4 - *Instructions for Submitters* - explanation of how to make a submission to this RFP.

Chapter 5 - *General Requirements on Proposals* - requirements and evaluation criteria that apply to all proposals submitted to OMG.

Chapter 6 - *Specific Requirements on Proposals* - problem statement, scope of proposals sought, requirements and optional features, issues to be discussed, evaluation criteria, and timetable that apply specifically to this RFP.

Appendix A – *References and Glossary Specific to this RFP*

Appendix B – General References and Glossary

1.3 Conventions

The key words "**must**", "**must not**", "**required**", "**shall**", "**shall not**", "**should**", "**should not**", "**recommended**", "**may**", and "**optional**" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.4 Contact Information

Questions related to the OMG's technology adoption process may be directed to *omg-process@omg.org*. General questions about this RFP may be sent to *responses@omg.org*.

OMG documents (and information about the OMG in general) can be obtained from the OMG's web site *(http://www.omg.org/)*. OMG documents may also be obtained by contacting OMG at *documents@omg.org*. Templates for RFPs (like this document) and other standard OMG documents can be found at the OMG Template Downloads Page at *http://www.omg.org/technology/template_download.htm*

2.0 Architectural Context

MDA provides a set of guidelines for structuring specifications expressed as models and the mappings between those models. The MDA initiative and the standards that support it allow the same model specifying business system or application functionality and behavior to be realized on multiple platforms. MDA enables different applications to be integrated by explicitly relating their models; this facilitates integration and interoperability and supports system evolution (deployment choices) as platform technologies change. The three primary goals of MDA are portability, interoperability and reusability.

Portability of any subsystem is relative to the subsystems on which it depends. The collection of subsystems that a given subsystem depends upon is often loosely called the *platform*, which supports that subsystem. Portability – and reusability - of such a subsystem is enabled if all the subsystems that it depends upon use standardized interfaces (APIs) and usage patterns.

MDA provides a pattern comprising a portable subsystem that is able to use any one of multiple specific implementations of a platform. This pattern is repeatedly usable in the specification of systems. The five important concepts related to this pattern are:

(1) *Model* – A model is a representation of a part of the function, structure and/or behavior of an application or system. A representation is said to be formal when it is based on a language that has a well-defined form ("syntax"), meaning ("semantics"), and possibly rules of analysis, inference, or proof for its constructs. The syntax may be graphical or textual. The semantics might be defined, more or less formally, in terms of things observed in the world being described (e.g. message sends and replies, object states and state changes, etc.), or by translating higher-level language constructs into other constructs that have a well-defined meaning. The optional rules of inference define what unstated properties you can deduce from the explicit statements in the model. In MDA, a representation that is not formal in this sense is not a model. Thus, a diagram with boxes and lines and arrows that is not supported by a definition of the meaning of a box, and the meaning of a line and of an arrow is not a model—it is just an informal diagram.

(2) *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.

(3) *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.

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

(5) *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. A mapping may be expressed as associations, constraints, rules, templates with parameters that must be assigned during the mapping, or other forms yet to be determined.

For example, in case of CORBA the platform is specified by a set of interfaces and usage patterns that constitute the CORBA Core Specification [CORBA]. The CORBA platform is independent of operating systems and programming languages. The OMG Trading Object Service specification [TOS] (consisting of interface specifications in OMG Interface Definition Language (OMG IDL)) can be considered to be a PIM from the viewpoint of CORBA, because it is independent of operating systems and programming languages. When the IDL to C++ Language Mapping specification is applied to the Trading Service PIM, the

C++-specific result can be considered to be a PSM for the Trading Service, where the platform is the C++ language and the C++ ORB implementation. Thus the IDL to C++ Language Mapping specification [IDLC++] determines the mapping from the Trading Service PIM to the Trading Service PSM.

Note that the Trading Service model expressed in IDL is a PSM relative to the CORBA platform too. This highlights the fact that platform-independence and platform-specificity are relative concepts.

The UML Profile for EDOC specification [EDOC] is another example of the application of various aspects of MDA. It defines a set of modeling constructs that are independent of middleware platforms such as EJB [EJB], CCM [CCM], MQSeries [MQS], etc. A PIM based on the EDOC profile uses the middleware-independent constructs defined by the profile and thus is middleware-independent. In addition, the specification defines formal metamodels for some specific middleware platforms such as EJB, supplementing the already-existing OMG metamodel of CCM (CORBA Component Model). The specification also defines mappings from the EDOC profile to the middleware metamodels. For example, it defines a mapping from the EDOC profile to EJB. The mapping specifications facilitate the transformation of any EDOC-based PIM into a corresponding PSM for any of the specific platforms for which a mapping is specified.

Continuing with this example, one of the PSMs corresponding to the EDOC PIM could be for the CORBA platform. This PSM then potentially constitutes a PIM, corresponding to which there would be implementation language specific PSMs derived via the CORBA language mappings, thus illustrating recursive use of the Platform-PIM-PSM-Mapping pattern.

Note that the EDOC profile can also be considered to be a platform in its own right. Thus, a model expressed via the profile is a PSM relative to the EDOC platform.

An analogous set of concepts apply to Interoperability Protocols wherein there is a PIM of the payload data and a PIM of the interactions that cause the data to find its way from one place to another. These then are realized in specific ways for specific platforms in the corresponding PSMs.

Analogously, in case of databases there could be a PIM of the data (say using the Relational Data Model), and corresponding PSMs specifying how the data is actually represented on a storage medium based on some particular data storage paradigm etc., and a mapping from the PIM to each PSM.

OMG adopts standard specifications of models that exploit the MDA pattern to facilitate portability, interoperability and reusability, either through ab initio

development of standards or by reference to existing standards. Some examples of OMG adopted specifications are:

1. *Languages* – e.g. IDL for interface specification, UML for model specification, OCL for constraint specification, etc.

(6) Mappings – e.g. Mapping of OMG IDL to specific implementation languages (CORBA PIM to Implementation Language PSMs), UML Profile for EDOC (PIM) to CCM (CORBA PSM) and EJB (Java PSM), CORBA (PSM) to COM (PSM) etc.

(7) *Services* – e.g. Naming Service [NS], Transaction Service [OTS], Security Service [SEC], Trading Object Service [TOS] etc.

(8) *Platforms* – e.g. CORBA [CORBA].

(9) *Protocols* – e.g. GIOP/IIOP [CORBA] (both structure and exchange protocol), XML Metadata Interchange [XMI] (structure specification usable as payload on multiple exchange protocols).

(10) *Domain Specific Standards* – e.g. Data Acquisition from Industrial Systems (Manufacturing) [DAIS], General Ledger Specification (Finance) [GLS], Air Traffic Control (Transportation) [ATC], Gene Expression (Life Science Research) [GE], Personal Identification Service (Healthcare) [PIDS], etc.

For an introduction to MDA, see [MDAa]. For a discourse on the details of MDA please refer to [MDAc]. To see an example of the application of MDA see [MDAb]. For general information on MDA, see [MDAd].

Object Management Architecture (OMA) is a distributed object computing platform architecture within MDA that is related to ISO's Reference Model of Open Distributed Processing RM-ODP[RM-ODP]. CORBA and any extensions to it are based on OMA. For information on OMA see [OMA].

3.0 Adoption Process

3.1 Introduction

OMG adopts specifications by explicit vote on a technology-by-technology basis. The specifications selected each satisfy the architectural vision of MDA. OMG bases its decisions on both business and technical considerations. Once a specification adoption is finalized by OMG, it is made available for use by both OMG members and non-members alike.

Request for Proposals (RFP) are issued by a *Technology Committee* (TC), typically upon the recommendation of a *Task Force* (TF) and duly endorsed by the *Architecture Board* (AB).

Submissions to RFPs are evaluated by the TF that initiated the RFP. Selected specifications are *recommended* to the parent TC after being *reviewed* for technical merit and consistency with MDA and other adopted specifications and *endorsed* by the AB. The parent TC of the initiating TF then votes to *recommend adoption* to the OMG Board of Directors (BoD). The BoD acts on the recommendation to complete the adoption process.

For more detailed information on the adoption process see the *Policies and Procedures of the OMG Technical Process* [P&P] and the *OMG Hitchhiker's Guide* [Guide]. In case of any inconsistency between this document and the [P&P] in all cases the [P&P] shall prevail.

3.2 Steps in the Adoption Process

A TF, its parent TC, the AB and the Board of Directors participate in a collaborative process, which typically takes the following form:

- *Development and Issuance of RFP*

  RFPs are drafted by one or more OMG members who are interested in the adoption of a standard in some specific area. The draft RFP is presented to an appropriate TF, based on its subject area, for approval and recommendation to issue. The TF and the AB provide guidance to the drafters of the RFP. When the TF and the AB are satisfied that the RFP is appropriate and ready for issuance, the TF recommends issuance to its parent TC, and the AB endorses the recommendation. The TC then acts on the recommendation and issues the RFP.

- *Letter of Intent (LOI)*

  A Letter of Intent (LOI) must be submitted to the OMG signed by an officer of the member organization which intends to respond to the RFP, confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements. (See section 4.3 for more information.). In order to respond to an RFP the organization must be a member of the TC that issued the RFP.

- *Voter Registration*

  Interested OMG members, other than Trial, Press and Analyst members, may participate in specification selection votes in the TF for an RFP. They may need to register to do so, if so stated in the RFP. Registration ends on a

specified date, 6 or more weeks after the announcement of the registration period. The registration closure date is typically around the time of initial submissions. Member organizations that have submitted an LOI are automatically registered to vote.

- *Initial Submissions*

  Initial Submissions are due by a specified deadline. Submitters normally present their proposals at the first meeting of the TF after the deadline. Initial Submissions are expected to be complete enough to provide insight on the technical directions and content of the proposals.

- *Revision Phase*

  During this time submitters have the opportunity to revise their Submissions, if they so choose.

- *Revised Submissions*

  Revised Submissions are due by a specified deadline. Submitters again normally present their proposals at the next meeting of the TF after the deadline. (Note that there may be more than one Revised Submission deadline. The decision to set new Revised Submission deadlines is made by the registered voters for that RFP.)

- *Selection Votes*

  When the registered voters for the RFP believe that they sufficiently understand the relative merits of the Revised Submissions, a selection vote is taken. The result of this selection vote is a recommendation for adoption to the TC. The AB reviews the proposal for MDA compliance and technical merit. An endorsement from the AB moves the voting process into the issuing Technology Committee. An eight-week voting period ensues in which the TC votes to recommend adoption to the OMG Board of Directors (BoD). The final vote, the vote to adopt, is taken by the BoD and is based on technical merit as well as business qualifications. The resulting draft standard is called the *Alpha Specification*.

- Business Committee Questionnaire

  The submitting members whose proposal is recommended for adoption need to submit their response to the BoD Business Committee Questionnaire [BCQ] detailing how they plan to make use of and/or make the resulting standard available in products. If no organization commits to make use of the standard, then the BoD will typically not act on the recommendation to adopt the standard - so it is very important to fulfill this requirement.

- Finalization

  A Finalization Task Force (FTF) is chartered by the TC that issued the RFP, to prepare an Alpha submission for publishing as a Formal (i.e. publicly available) specification, by fixing any problems that are reported by early users of the specification. Upon completion of its activity the FTF recommends adoption of the resulting Beta (draft) specification. The parent TC acts on the recommendation and recommends adoption to the BoD. OMG Technical Editors produce the Formal Specification document based on this Beta Specification.

- Revision

  A Revision Task Force (RTF) is normally chartered by a TC, after the FTF completes its work, to manage issues filed against the Formal Specification by implementers and users. The output of the RTF is a Beta specification reflecting minor technical changes, which the TC and Board will usually approve for adoption as the next version of the Formal Specification.

3.3 Goals of the evaluation

The primary goals of the TF evaluation are to:

- Provide a fair and open process
- Facilitate critical review of the submissions by members of OMG
- Provide feedback to submitters enabling them to address concerns in their revised submissions
- Build consensus on acceptable solutions
- Enable voting members to make an informed selection decision

Submitters are expected to actively contribute to the evaluation process.

4.0 Instructions for Submitters

4.1 OMG Membership

To submit to an RFP issued by the Platform Technology Committee the submitter or submitters must be either Platform or Contributing members on the date of the submission deadline, while for Domain Technology RFPs the submitter or submitters must be either Contributing or Domain members. Submitters sometimes choose to name other organizations that support a submission in some way; however, this has no formal status within the OMG

process, and for OMG's purposes confers neither duties nor privileges on the organizations thus named.

4.2 Submission Effort

An RFP submission may require significant effort in terms of document preparation, presentations to the issuing TF, and participation in the TF evaluation process. Several staff months of effort might be necessary. OMG is unable to reimburse submitters for any costs in conjunction with their submissions to this RFP.

4.3 Letter of Intent

A Letter of Intent (LOI) must be submitted to the OMG Business Committee signed by an officer of the submitting 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. These terms, conditions, and requirements are defined in the *Business Committee RFP Attachment* and are reproduced verbatim in section 4.4 below.

The LOI should designate a single contact point within the submitting organization for receipt of all subsequent information regarding this RFP and the submission. The name of this contact will be made available to all OMG members. The LOI is typically due 60 days before the deadline for initial submissions. LOIs must be sent by fax or paper mail to the "RFP Submissions Desk" at the main OMG address shown on the first page of this RFP.

Here is a suggested template for the Letter of Intent:

This letter confirms the intent of <organization required> (the organization) to submit a response to the OMG <RFP name required> RFP. We will grant OMG and its members the right to copy our response for review purposes as specified in section 4.7 of the RFP. Should our response be adopted by OMG we will comply with the OMG Business Committee terms set out in section 4.4 of the RFP and in document omg/06-03-02.

<contact name and details required> will be responsible for liaison with OMG regarding this RFP response.

The signatory below is an officer of the organization and has the approval and authority to make this commitment on behalf of the organization.

<signature required>

4.4 Business Committee RFP Attachment

This section contains the text of the Business Committee RFP attachment concerning commercial availability requirements placed on submissions. This attachment is available separately as an OMG document omg/06-03-02.

Commercial considerations in OMG technology adoption

A1 Introduction

OMG wishes to encourage rapid commercial adoption of the specifications it publishes. To this end, there must be neither technical, legal nor commercial obstacles to their implementation. Freedom from the first is largely judged through technical review by the relevant OMG Technology Committees; the second two are the responsibility of the OMG Business Committee. The BC also looks for evidence of a commitment by a submitter to the commercial success of products based on the submission.

A2 Business Committee evaluation criteria

A2.1 Viable to implement across platforms

While it is understood that final candidate OMG submissions often combine technologies before they have all been implemented in one system, the Business Committee nevertheless wishes to see evidence that each major feature has been implemented, preferably more than once, and by separate organisations. Pre-product implementations are acceptable. Since use of OMG specifications should not be dependant on any one platform, cross-platform availability and interoperability of implementations should be also be demonstrated.

A2.2 Commercial availability

In addition to demonstrating the existence of implementations of the specification, the submitter must also show that products based on the specification are commercially available, or will be within 12 months of the date when the specification was recommended for adoption by the appropriate Task Force. Proof of intent to ship product within 12 months might include:

- *A public product announcement with a shipping date within the time limit.*
- *Demonstration of a prototype implementation and accompanying draft user documentation.*

Alternatively, and at the Business Committee's discretion, submissions may be adopted where the submitter is not a commercial software provider, and therefore will not make implementations commercially available. However, in this case the BC will require concrete evidence of two or more independent implementations of the specification being used by end- user organisations as part of their businesses. Regardless of which requirement is in use, the submitter must inform the OMG of completion of the implementations when commercially available.

A2.3 Access to Intellectual Property Rights

OMG will not adopt a specification if OMG is aware of any submitter, member or third party which holds a patent, copyright or other intellectual property right (collectively referred to in this policy statement as "IPR") which might be infringed by implementation or recommendation of such specification, unless OMG believes that such IPR owner will grant a license to organisations (whether OMG members or not) on non-discriminatory and commercially reasonable terms which wish to make use of the specification. Accordingly, the submitter must certify that it is not aware of any claim that the specification infringes any IPR of a third party or that it is aware and believes that an appropriate non-discriminatory license is available from that third party. Except for this certification, the submitter will not be required to make any other warranty, and specifications will be offered by OMG for use "as is". If the submitter owns IPR to which an use of a specification based upon its submission would necessarily be subject, it must certify to the Business Committee that it will make a suitable license available to any user on non- discriminatory and commercially reasonable terms, to permit development and commercialisation of an implementation that includes such IPR.

It is the goal of the OMG to make all of its technology available with as few impediments and disincentives to adoption as possible, and therefore OMG strongly encourages the submission of technology as to which royalty-free licenses will be available. However, in all events, the submitter shall also certify that any necessary licence will be made available on commercially reasonable, non-discriminatory terms. The submitter is responsible for disclosing in detail all known restrictions, placed either by the submitter or, if known, others, on technology necessary for any use of the specification.

A2.4 Publication of the specification

Should the submission be adopted, the submitter must grant OMG (and its sublicensees) a world- wide, royalty-free licence to edit, store, duplicate and distribute both the specification and works derived from it (such as revisions and teaching materials). This requirement applies only to the written specification, not to any implementation of it.

A2.5 Continuing support

The submitter must show a commitment to continue supporting the technology underlying the specification after OMG adoption, for instance by showing the BC development plans for future revisions, enhancement or maintenance.

4.5 Responding to RFP items

4.5.1 Complete proposals

A submission must propose full specifications for all of the relevant requirements detailed in Chapter 6 of this RFP. Submissions that do not present complete proposals may be at a disadvantage.

Submitters are highly encouraged to propose solutions to any optional requirements enumerated in Chapter 6.

4.5.2 Additional specifications

Submissions may include additional specifications for items not covered by the RFP that they believe to be necessary and integral to their proposal. Information on these additional items should be clearly distinguished.

Submitters must give a detailed rationale as to why these specifications should also be considered for adoption. However submitters should note that a TF is unlikely to consider additional items that are already on the roadmap of an OMG TF, since this would pre-empt the normal adoption process.

4.5.3 Alternative approaches

Submitters may provide alternative RFP item definitions, categorizations, and groupings so long as the rationale for doing so is clearly stated. Equally, submitters may provide alternative models for how items are provided if there are compelling technological reasons for a different approach.

4.6 Confidential and Proprietary Information

The OMG specification adoption process is an open process. Responses to this RFP become public documents of the OMG and are available to members and non-members alike for perusal. No confidential or proprietary information of any kind will be accepted in a submission to this RFP.

4.7 Copyright Waiver

Every submission document must contain: (i) a waiver of copyright for unlimited duplication by the OMG, and (ii) a limited waiver of copyright that allows each OMG member to make up to fifty (50) copies of the document for review purposes only. See Section 4.9.2 for recommended language.

4.8 Proof of Concept

Submissions must include a "proof of concept" statement, explaining how the submitted specifications have been demonstrated to be technically viable. The technical viability has to do with the state of development and maturity of the technology on which a submission is based. This is not the same as commercial availability. Proof of concept statements can contain any information deemed relevant by the submitter; for example:

> "This specification has completed the design phase and is in the process of being prototyped."
>
> "An implementation of this specification has been in beta-test for 4 months."
>
> "A named product (with a specified customer base) is a realization of this specification."

It is incumbent upon submitters to demonstrate the technical viability of their proposal to the satisfaction of the TF managing the evaluation process. OMG will favor proposals based on technology for which sufficient relevant experience has been gained.

4.9 Format of RFP Submissions

This section presents the structure of a submission in response to an RFP. *All submissions* must contain the elements itemized in section 4.9.2 below before they can be accepted as a valid response for evaluation or a vote can be taken to recommend for adoption.

4.9.1 General

- Submissions that are concise and easy to read will inevitably receive more consideration.
- Submitted documentation should be confined to that directly relevant to the items requested in the RFP. If this is not practical, submitters must make clear what portion of the documentation pertains directly to the RFP and what portion does not.

- The key words "**must**", "**must not**", "**required**", "**shall**", "**shall not**", "**should**", "**should not**", "**recommended**", "**may**", and "**optional**" shall be used in the submissions with the meanings as described in RFC 2119 [RFC2119].

4.9.2 Required Outline

A three-part structure for submissions is required. Part I is non-normative, providing information relevant to the evaluation of the proposed specification. Part II is normative, representing the proposed specification. Specific sections like Appendices may be explicitly identified as non-normative in Part II. Part III is normative specifying changes that must be made to previously adopted specifications in order to be able to implement the specification proposed in Part II.

PART I

- •A cover page carrying the following information (a template for this is available [Inventory]):
  - The full name of the submission
  - The primary contact for the submission
  - The acronym proposed for the specification (e.g. UML, CORBA)
  - The name and document number of the RFP to which this is a response
  - The document number of the main submission document
  - An inventory of all accompanying documents, with OMG document number, short description, a URL where appropriate, and whether they are normative.
- List of OMG members making the submission (see 4.1) listing exactly which members are making the submission, so that submitters can be matched with LOI responders and their current eligibility can be verified.
- Copyright waiver (see 4.7), in a form acceptable to the OMG.

  One acceptable form is:

  "Each of the entities listed above: (i) grants to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version, and (ii) grants to each member of the OMG a nonexclusive, royalty-free, paid up, worldwide license to make up to fifty (50) copies of this document for internal review purposes only and not

for distribution, and (iii) has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used any OMG specification that may be based hereon or having conformed any computer software to such specification."

If you wish to use some other form you must get it approved by the OMG legal counsel before using it in a submission.

- For each member making the submission, an individual contact point who is authorized by the member to officially state the member's position relative to the submission, including matters related to copyright ownership, etc. (see 4.3)
- Overview or guide to the material in the submission
- Overall design rationale (if appropriate)
- Statement of proof of concept (see 4.8)
- Resolution of RFP requirements and requests

  Explain how the proposal satisfies the specific requirements and (if applicable) requests stated in Chapter 6. References to supporting material in Part II should be given.

  In addition, if the proposal does not satisfy any of the general requirements stated in Chapter 5, provide a detailed rationale.

- Responses to RFP issues to be discussed

  Discuss each of the "Issues To Be Discussed" identified in Chapter 6.

PART II

The contents of this part should be structured based on the template found in [FORMS] and should contain the following elements as per the instructions in the template document cited above:

- Scope of the proposed specification
- Proposed conformance criteria

  Submissions should propose appropriate conformance criteria for implementations.

- Proposed normative references

  Submissions should provide a list of the normative references that are used by the proposed specification

- Proposed list of terms and definitions

  Submissions should provide a list of terms that are used in the proposed specification with their definitions.

- Proposed list of symbols

  Submissions should provide a list of special symbols that are used in the proposed specification together with their significance

- Proposed specification

PART III

- Changes or extensions required to existing OMG specifications

  Submissions must include a full specification of any changes or extensions required to existing OMG specifications. This should be in a form that enables "mechanical" section-by-section revision of the existing specification.

4.10 How to Submit

Submitters should send an electronic version of their submission to the *RFP Submissions Desk* (*omg-documents@omg.org*) at OMG Headquarters by 5:00 PM U.S. Eastern Standard Time (22:00 GMT) on the day of the Initial and Revised Submission deadlines. Acceptable formats are Adobe FrameMaker source, ODF (ISO/IEC 26300), OASIS Darwin Information Typing Architecture (DITA) or OASIS DocBook 4.x (or later).

Submitters should make sure they receive electronic or voice confirmation of the successful receipt of their submission. Submitters should be prepared to send a single hardcopy version of their submission, if requested by OMG staff, to the attention of the "RFP Submissions Desk" at the main OMG address shown on the first page of this RFP.

5.0 General Requirements on Proposals

5.1 Requirements

5.1.1 Submitters are encouraged to express models using OMG modeling languages such as UML, MOF, CWM and SPEM (subject to any further constraints on the types of the models and modeling technologies specified in Chapter 6 of this RFP). Submissions containing models expressed via OMG modeling languages shall be accompanied by an OMG XMI [XMI] representation of the models

(including a machine-readable copy). A best effort should be made to provide an OMG XMI representation even in those cases where models are expressed via non-OMG modeling languages.

5.1.2 Chapter 6 of this RFP specifies whether PIM(s), PSM(s), or both are being solicited. If proposals specify a PIM and corresponding PSM(s), then the rules specifying the mapping(s) between the PIM and PSM(s) shall either be identified by reference to a standard mapping or specified in the proposal. In order to allow possible inconsistencies in a proposal to be resolved later, proposals shall identify whether the mapping technique or the resulting PSM(s) are to be considered normative.

5.1.3 Proposals shall be *precise* and *functionally complete*. All relevant assumptions and context required for implementing the specification shall be provided.

5.1.4 Proposals shall specify *conformance criteria* that clearly state what features all implementations must support and which features (if any) may *optionally* be supported.

5.1.5 Proposals shall *reuse* existing OMG and other standard specifications in preference to defining new models to specify similar functionality.

5.1.6 Proposals shall justify and fully specify any *changes or extensions* required to existing OMG specifications. In general, OMG favors proposals that are *upwards compatible* with existing standards and that minimize changes and extensions to existing specifications.

5.1.7 Proposals shall factor out functionality that could be used in different contexts and specify their models, interfaces, etc. separately. Such *minimalism* fosters re-use and avoids functional duplication.

5.1.8 Proposals shall use or depend on other specifications only where it is actually necessary. While re-use of existing specifications to avoid duplication will be encouraged, proposals should avoid gratuitous use.

5.1.9 Proposals shall be *compatible* with and *usable* with existing specifications from OMG and other standards bodies, as appropriate. Separate specifications offering distinct functionality should be usable together where it makes sense to do so.

5.1.10 Proposals shall preserve maximum *implementation flexibility*. Implementation descriptions should not be included and proposals shall not constrain implementations any more than is necessary to promote interoperability.

5.1.11 Proposals shall allow *independent implementations* that are *substitutable* and *interoperable*. An implementation should be replaceable by an alternative implementation without requiring changes to any client.

5.1.12 Proposals shall be compatible with the architecture for system distribution defined in ISO's Reference Model of Open Distributed Processing [RM-ODP]. Where such compatibility is not achieved, or is not appropriate, the response to the RFP must include reasons why compatibility is not appropriate and an outline of any plans to achieve such compatibility in the future.

5.1.13 In order to demonstrate that the specification proposed in response to this RFP can be made secure in environments requiring security, answers to the following questions shall be provided:

- What, if any, are the security sensitive elements that are introduced by the proposal?
- Which accesses to security-sensitive elements must be subject to security policy control?
- Does the proposed service or facility need to be security aware?
- What default policies (e.g., for authentication, audit, authorization, message protection etc.) should be applied to the security sensitive elements introduced by the proposal? Of what security considerations must the implementers of your proposal be aware?

The OMG has adopted several specifications, which cover different aspects of security and provide useful resources in formulating responses. [CSIV2] [SEC] [RAD].

5.1.14 Proposals shall specify the degree of internationalization support that they provide. The degrees of support are as follows:

a) Uncategorized: Internationalization has not been considered.

b) Specific to <region name>: The proposal supports the customs of the specified region only, and is not guaranteed to support the customs of any other region. Any fault or error caused by requesting the services outside of a context in which the customs of the specified region are being consistently followed is the responsibility of the requester.

c) Specific to <multiple region names>: The proposal supports the customs of the specified regions only, and is not guaranteed to support the customs of any other regions. Any fault or error caused by requesting the services

outside of a context in which the customs of at least one of the specified regions are being consistently followed is the responsibility of the requester.

d) Explicitly not specific to <region(s) name>: The proposal does not support the customs of the specified region(s). Any fault or error caused by requesting the services in a context in which the customs of the specified region(s) are being followed is the responsibility of the requester.

5.2 Evaluation criteria

Although the OMG adopts model-based specifications and not implementations of those specifications, the technical viability of implementations will be taken into account during the evaluation process. The following criteria will be used:

5.2.1 Performance

Potential implementation trade-offs for performance will be considered.

5.2.2 Portability

The ease of implementation on a variety of systems and software platforms will be considered.

5.2.3 Securability

The answer to questions in section 5.1.13 shall be taken into consideration to ascertain that an implementation of the proposal is securable in an environment requiring security.

5.2.4 Conformance: Inspectability and Testability

The adequacy of proposed specifications for the purposes of conformance inspection and testing will be considered. Specifications should provide sufficient constraints on interfaces and implementation characteristics to ensure that conformance can be unambiguously assessed through both manual inspection and automated testing.

5.2.5 Standardized Metadata

Where proposals incorporate metadata specifications, usage of OMG standard XMI metadata [XMI] representations must be provided as this allows specifications to be easily interchanged between XMI compliant tools and applications. Since use of XML (including XMI and XML/Value [XML/Value])

is evolving rapidly, the use of industry specific XML vocabularies (which may not be XMI compliant) is acceptable where justified.

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 ~~dynamic reconfiguration~~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.

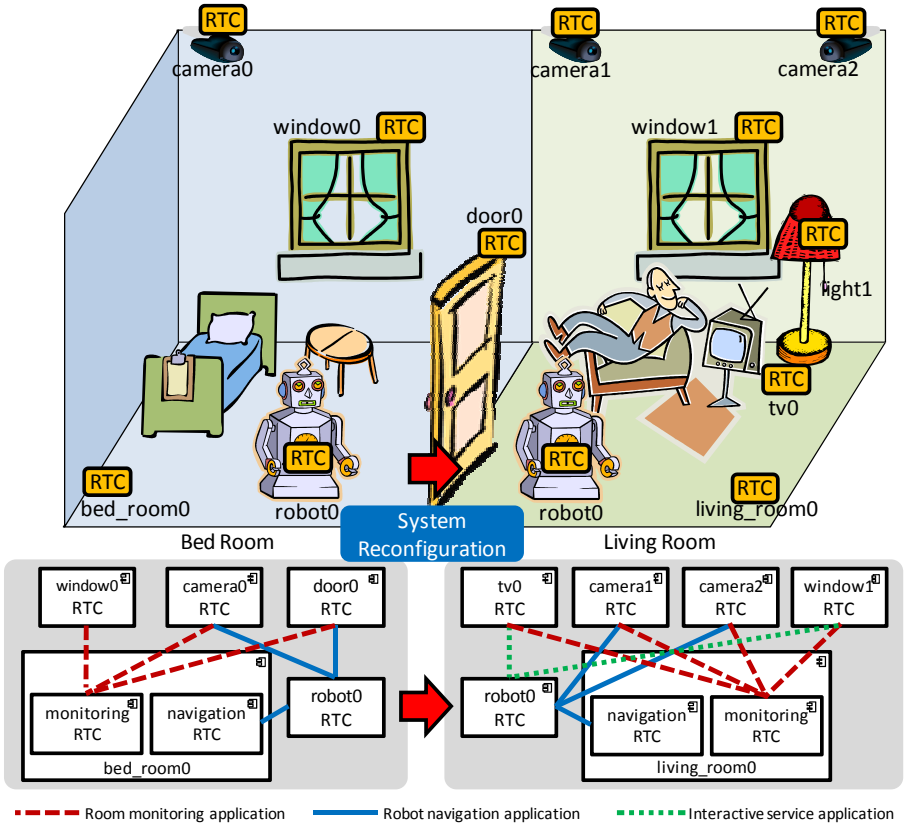


Figure 1 Typical robotic application environment

In order to address functionality of dynamic deployment and ~~dynamic reconfiguration~~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.

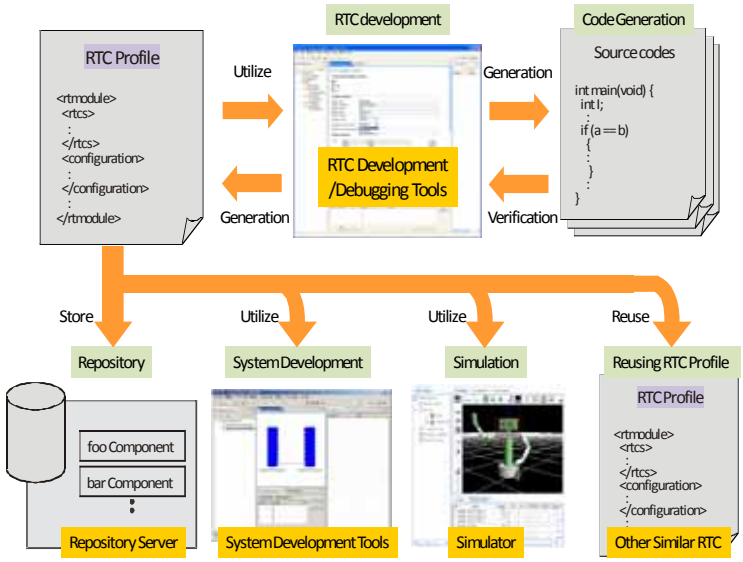


Figure 2 Use of the RTC Profile

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.

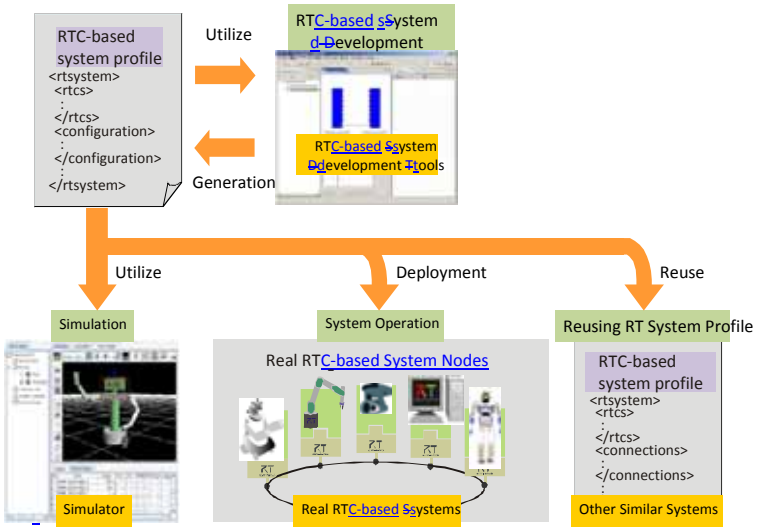


Figure 3 Use of RTC-based system profile

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 ~~reconfiguration~~configuration commonly results from the impairment of the participating component instances and/or changes in the robot location. To support such ~~reconfiguration~~configuration, the robot application (or system) needs to be notified whenever the situation changes. Since not all changes require ~~reconfiguration~~configuration, it must be possible to specify the specific environment changes that trigger ~~reconfiguration.~~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 (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 ~~reconfigure~~configure RTCs at runtime.

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

To use D&C in the robotics domain and expand RTC, the RFP proposes the specifications for the dynamic deployment and ~~dynamic reconfiguration~~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 ~~dynamic reconfiguration~~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 ~~CORBA~~OMG IDL (Interface Definition Language) and XML (eXtensible Mark-up Language).

The proposed specification shall provide functionality for component deployment and dynamic system ~~reconfiguration~~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.~~2~~3 [formal/2009-02-04] http://www.omg.org/spec/UML/2.3/Infrastructure/PDF/
- Unified Modeling Language: Superstructure Version 2.~~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

- ~~UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded systems, beta 3 - convenience document with change bars [ptc/09-05-13]~~

- ~~MARTE model library XMI file [ptc/09-05-16]~~

- ~~MARTE Profile XMI file [ptc/09-05-15]~~

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

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

~~6.5.1 Proposals shall specify a meta-model for the description of component meta-information necessary to support automatic searching and~~

~~comparing of RT components in component repositories and in the run-time system.~~

Platform independent deployment and configuration model

~~6.5.2~~ Proposals shall specify ~~a meta-model for the description of RTC services~~interfaces~~, their compatibility criteria, and deployment requirements.~~

~~6.5.3 Proposal shall specify a platform independent model for information service to identify to locate deployed RTCs available for utilization by the requesting robot.~~

~~6.5.4 Proposals shall specify a data-model for a component information registry using the meta-model requested in requirements 6.5.1 and 6.5.2. The proposal shall also specify a query mechanism for this repository.~~

~~6.5.5~~6.5.1 ~~Proposals shall specify a platform independent model for~~ for dynamic ~~RTC~~ configuration and deployment~~, which~~ of RTCs.

~~a) allows an efficient~~Proposal shall specify means to initiate RTC configuration ~~of RTCs.~~

~~6.5.6~~6.5.2 ~~initiates reconfiguration~~ based on external and/or internal events. A capability for event filtering shall be provided.

~~b) supports coordinated reconfiguration of multiple robot systems to allow performing coordinated tasks.~~

~~c) defines a service interface for the deployment process.~~

~~6.5.7~~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].

Platform independent RTC information model

6.5.4 Proposals shall provide a schema describing RTC characteristics.

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.

6.6 Optional Requirements

~~None~~

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 demonstrate its feasibility by using a specific application based on the proposed model.~~

~~6.7.2 Proposals shall demonstrate its applicability to existing technology such as the RTC specification [RTC].~~

~~6.7.3 Proposals shall discuss simplicity of implementation.~~

~~6.7.4~~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 ~~EJB,~~ CCM [CCM]~~-~~, SDRP ~~SCA~~[SDRP], D&C [D&C] ~~-~~and other well-known component models.

~~6.7.5 Proposals shall discuss the possibility of providing a standard mechanism for advertising and querying component instances and receiving change notifications~~

~~6.7.6~~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].

~~6.7.7~~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

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

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, 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, 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 ~~6th~~ 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] 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.

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/2010-03-05 (errata)
mars/2010-03-06 (with change bar)
mars/2010-03-07 (convenience document)

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.

1.0 Introduction

1.1 Goals of OMG

The Object Management Group (OMG) is the world's largest software consortium with an international membership of vendors, developers, and end users. Established in 1989, its mission is to help computer users solve enterprise integration problems by supplying open, vendor-neutral portability, interoperability and reusability specifications based on Model Driven Architecture (MDA). MDA defines an approach to IT system specification that separates the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform, and provides a set of guidelines for structuring specifications expressed as models. OMG has established numerous widely used standards such as OMG IDL[IDL], CORBA[CORBA], Realtime CORBA [CORBA], GIOP/IIOP[CORBA], UML[UML], MOF[MOF], XMI[XMI] and CWM[CWM] to name a few significant ones.

1.2 Organization of this document

The remainder of this document is organized as follows:

Chapter 2 - *Architectural Context* - background information on OMG's Model Driven Architecture.

Chapter 3 - *Adoption Process* - background information on the OMG specification adoption process.

Chapter 4 - *Instructions for Submitters* - explanation of how to make a submission to this RFP.

Chapter 5 - *General Requirements on Proposals* - requirements and evaluation criteria that apply to all proposals submitted to OMG.

Chapter 6 - *Specific Requirements on Proposals* - problem statement, scope of proposals sought, requirements and optional features, issues to be discussed, evaluation criteria, and timetable that apply specifically to this RFP.

Appendix A – *References and Glossary Specific to this RFP*

Appendix B – General References and Glossary

1.3 Conventions

The key words "**must**", "**must not**", "**required**", "**shall**", "**shall not**", "**should**", "**should not**", "**recommended**", "**may**", and "**optional**" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.4 Contact Information

Questions related to the OMG's technology adoption process may be directed to *omg-process@omg.org*. General questions about this RFP may be sent to *responses@omg.org*.

OMG documents (and information about the OMG in general) can be obtained from the OMG's web site *(http://www.omg.org/)*. OMG documents may also be obtained by contacting OMG at *documents@omg.org*. Templates for RFPs (like this document) and other standard OMG documents can be found at the OMG Template Downloads Page at
http://www.omg.org/technology/template_download.htm

2.0 Architectural Context

MDA provides a set of guidelines for structuring specifications expressed as models and the mappings between those models. The MDA initiative and the standards that support it allow the same model specifying business system or application functionality and behavior to be realized on multiple platforms. MDA enables different applications to be integrated by explicitly relating their models; this facilitates integration and interoperability and supports system evolution (deployment choices) as platform technologies change. The three primary goals of MDA are portability, interoperability and reusability.

Portability of any subsystem is relative to the subsystems on which it depends. The collection of subsystems that a given subsystem depends upon is often loosely called the *platform*, which supports that subsystem. Portability – and reusability - of such a subsystem is enabled if all the subsystems that it depends upon use standardized interfaces (APIs) and usage patterns.

MDA provides a pattern comprising a portable subsystem that is able to use any one of multiple specific implementations of a platform. This pattern is repeatedly usable in the specification of systems. The five important concepts related to this pattern are:

(1) *Model* – A model is a representation of a part of the function, structure and/or behavior of an application or system. A representation is said to be formal when it is based on a language that has a well-defined form

("syntax"), meaning ("semantics"), and possibly rules of analysis, inference, or proof for its constructs. The syntax may be graphical or textual. The semantics might be defined, more or less formally, in terms of things observed in the world being described (e.g. message sends and replies, object states and state changes, etc.), or by translating higher-level language constructs into other constructs that have a well-defined meaning. The optional rules of inference define what unstated properties you can deduce from the explicit statements in the model. In MDA, a representation that is not formal in this sense is not a model. Thus, a diagram with boxes and lines and arrows that is not supported by a definition of the meaning of a box, and the meaning of a line and of an arrow is not a model—it is just an informal diagram.

(2) *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.

(3) *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.

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

(5) *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. A mapping may be expressed as associations, constraints, rules, templates with parameters that must be assigned during the mapping, or other forms yet to be determined.

For example, in case of CORBA the platform is specified by a set of interfaces and usage patterns that constitute the CORBA Core Specification [CORBA]. The CORBA platform is independent of operating systems and programming languages. The OMG Trading Object Service specification [TOS] (consisting of interface specifications in OMG Interface Definition Language (OMG IDL)) can be considered to be a PIM from the viewpoint of CORBA, because it is independent of operating systems and programming languages. When the IDL to C++ Language Mapping specification is applied to the Trading Service PIM, the C++-specific result can be considered to be a PSM for the Trading Service, where the platform is the C++ language and the C++ ORB implementation.

Thus the IDL to C++ Language Mapping specification [IDLC++] determines the mapping from the Trading Service PIM to the Trading Service PSM.

Note that the Trading Service model expressed in IDL is a PSM relative to the CORBA platform too. This highlights the fact that platform-independence and platform-specificity are relative concepts.

The UML Profile for EDOC specification [EDOC] is another example of the application of various aspects of MDA. It defines a set of modeling constructs that are independent of middleware platforms such as EJB [EJB], CCM [CCM], MQSeries [MQS], etc. A PIM based on the EDOC profile uses the middleware-independent constructs defined by the profile and thus is middleware-independent. In addition, the specification defines formal metamodels for some specific middleware platforms such as EJB, supplementing the already-existing OMG metamodel of CCM (CORBA Component Model). The specification also defines mappings from the EDOC profile to the middleware metamodels. For example, it defines a mapping from the EDOC profile to EJB. The mapping specifications facilitate the transformation of any EDOC-based PIM into a corresponding PSM for any of the specific platforms for which a mapping is specified.

Continuing with this example, one of the PSMs corresponding to the EDOC PIM could be for the CORBA platform. This PSM then potentially constitutes a PIM, corresponding to which there would be implementation language specific PSMs derived via the CORBA language mappings, thus illustrating recursive use of the Platform-PIM-PSM-Mapping pattern.

Note that the EDOC profile can also be considered to be a platform in its own right. Thus, a model expressed via the profile is a PSM relative to the EDOC platform.

An analogous set of concepts apply to Interoperability Protocols wherein there is a PIM of the payload data and a PIM of the interactions that cause the data to find its way from one place to another. These then are realized in specific ways for specific platforms in the corresponding PSMs.

Analogously, in case of databases there could be a PIM of the data (say using the Relational Data Model), and corresponding PSMs specifying how the data is actually represented on a storage medium based on some particular data storage paradigm etc., and a mapping from the PIM to each PSM.

OMG adopts standard specifications of models that exploit the MDA pattern to facilitate portability, interoperability and reusability, either through ab initio development of standards or by reference to existing standards. Some examples of OMG adopted specifications are:

1. *Languages* – e.g. IDL for interface specification, UML for model specification, OCL for constraint specification, etc.

(6) Mappings – e.g. Mapping of OMG IDL to specific implementation languages (CORBA PIM to Implementation Language PSMs), UML Profile for EDOC (PIM) to CCM (CORBA PSM) and EJB (Java PSM), CORBA (PSM) to COM (PSM) etc.

(7) *Services* – e.g. Naming Service [NS], Transaction Service [OTS], Security Service [SEC], Trading Object Service [TOS] etc.

(8) *Platforms* – e.g. CORBA [CORBA].

(9) *Protocols* – e.g. GIOP/IIOP [CORBA] (both structure and exchange protocol), XML Metadata Interchange [XMI] (structure specification usable as payload on multiple exchange protocols).

(10) *Domain Specific Standards* – e.g. Data Acquisition from Industrial Systems (Manufacturing) [DAIS], General Ledger Specification (Finance) [GLS], Air Traffic Control (Transportation) [ATC], Gene Expression (Life Science Research) [GE], Personal Identification Service (Healthcare) [PIDS], etc.

For an introduction to MDA, see [MDAa]. For a discourse on the details of MDA please refer to [MDAc]. To see an example of the application of MDA see [MDAb]. For general information on MDA, see [MDAd].

Object Management Architecture (OMA) is a distributed object computing platform architecture within MDA that is related to ISO's Reference Model of Open Distributed Processing RM-ODP[RM-ODP]. CORBA and any extensions to it are based on OMA. For information on OMA see [OMA].

3.0 Adoption Process

3.1 Introduction

OMG adopts specifications by explicit vote on a technology-by-technology basis. The specifications selected each satisfy the architectural vision of MDA. OMG bases its decisions on both business and technical considerations. Once a

specification adoption is finalized by OMG, it is made available for use by both OMG members and non-members alike.

Request for Proposals (RFP) are issued by a *Technology Committee* (TC), typically upon the recommendation of a *Task Force* (TF) and duly endorsed by the *Architecture Board* (AB).

Submissions to RFPs are evaluated by the TF that initiated the RFP. Selected specifications are *recommended* to the parent TC after being *reviewed* for technical merit and consistency with MDA and other adopted specifications and *endorsed* by the AB. The parent TC of the initiating TF then votes to *recommend adoption* to the OMG Board of Directors (BoD). The BoD acts on the recommendation to complete the adoption process.

For more detailed information on the adoption process see the *Policies and Procedures of the OMG Technical Process* [P&P] and the *OMG Hitchhiker's Guide* [Guide]. In case of any inconsistency between this document and the [P&P] in all cases the [P&P] shall prevail.

3.2 Steps in the Adoption Process

A TF, its parent TC, the AB and the Board of Directors participate in a collaborative process, which typically takes the following form:

- *Development and Issuance of RFP*

  RFPs are drafted by one or more OMG members who are interested in the adoption of a standard in some specific area. The draft RFP is presented to an appropriate TF, based on its subject area, for approval and recommendation to issue. The TF and the AB provide guidance to the drafters of the RFP. When the TF and the AB are satisfied that the RFP is appropriate and ready for issuance, the TF recommends issuance to its parent TC, and the AB endorses the recommendation. The TC then acts on the recommendation and issues the RFP.

- *Letter of Intent (LOI)*

  A Letter of Intent (LOI) must be submitted to the OMG signed by an officer of the member organization which intends to respond to the RFP, confirming the organization's willingness to comply with OMG's terms and conditions, and commercial availability requirements. (See section 4.3 for more information.). In order to respond to an RFP the organization must be a member of the TC that issued the RFP.

- *Voter Registration*

Interested OMG members, other than Trial, Press and Analyst members, may participate in specification selection votes in the TF for an RFP. They may need to register to do so, if so stated in the RFP. Registration ends on a specified date, 6 or more weeks after the announcement of the registration period. The registration closure date is typically around the time of initial submissions. Member organizations that have submitted an LOI are automatically registered to vote.

- *Initial Submissions*

  Initial Submissions are due by a specified deadline. Submitters normally present their proposals at the first meeting of the TF after the deadline. Initial Submissions are expected to be complete enough to provide insight on the technical directions and content of the proposals.

- *Revision Phase*

  During this time submitters have the opportunity to revise their Submissions, if they so choose.

- *Revised Submissions*

  Revised Submissions are due by a specified deadline. Submitters again normally present their proposals at the next meeting of the TF after the deadline. (Note that there may be more than one Revised Submission deadline. The decision to set new Revised Submission deadlines is made by the registered voters for that RFP.)

- *Selection Votes*

  When the registered voters for the RFP believe that they sufficiently understand the relative merits of the Revised Submissions, a selection vote is taken. The result of this selection vote is a recommendation for adoption to the TC. The AB reviews the proposal for MDA compliance and technical merit. An endorsement from the AB moves the voting process into the issuing Technology Committee. An eight-week voting period ensues in which the TC votes to recommend adoption to the OMG Board of Directors (BoD). The final vote, the vote to adopt, is taken by the BoD and is based on technical merit as well as business qualifications. The resulting draft standard is called the *Alpha Specification*.

- Business Committee Questionnaire

  The submitting members whose proposal is recommended for adoption need to submit their response to the BoD Business Committee Questionnaire [BCQ] detailing how they plan to make use of and/or make the resulting standard available in products. If no organization commits to make use of the

standard, then the BoD will typically not act on the recommendation to adopt the standard - so it is very important to fulfill this requirement.

- Finalization

  A Finalization Task Force (FTF) is chartered by the TC that issued the RFP, to prepare an Alpha submission for publishing as a Formal (i.e. publicly available) specification, by fixing any problems that are reported by early users of the specification. Upon completion of its activity the FTF recommends adoption of the resulting Beta (draft) specification. The parent TC acts on the recommendation and recommends adoption to the BoD. OMG Technical Editors produce the Formal Specification document based on this Beta Specification.

- Revision

  A Revision Task Force (RTF) is normally chartered by a TC, after the FTF completes its work, to manage issues filed against the Formal Specification by implementers and users. The output of the RTF is a Beta specification reflecting minor technical changes, which the TC and Board will usually approve for adoption as the next version of the Formal Specification.

3.3 Goals of the evaluation

The primary goals of the TF evaluation are to:

- Provide a fair and open process
- Facilitate critical review of the submissions by members of OMG
- Provide feedback to submitters enabling them to address concerns in their revised submissions
- Build consensus on acceptable solutions
- Enable voting members to make an informed selection decision

Submitters are expected to actively contribute to the evaluation process.

4.0 Instructions for Submitters

4.1 OMG Membership

To submit to an RFP issued by the Platform Technology Committee the submitter or submitters must be either Platform or Contributing members on the date of the submission deadline, while for Domain Technology RFPs the

submitter or submitters must be either Contributing or Domain members. Submitters sometimes choose to name other organizations that support a submission in some way; however, this has no formal status within the OMG process, and for OMG's purposes confers neither duties nor privileges on the organizations thus named.

4.2 Submission Effort

An RFP submission may require significant effort in terms of document preparation, presentations to the issuing TF, and participation in the TF evaluation process. Several staff months of effort might be necessary. OMG is unable to reimburse submitters for any costs in conjunction with their submissions to this RFP.

4.3 Letter of Intent

A Letter of Intent (LOI) must be submitted to the OMG Business Committee signed by an officer of the submitting 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. These terms, conditions, and requirements are defined in the *Business Committee RFP Attachment* and are reproduced verbatim in section 4.4 below.

The LOI should designate a single contact point within the submitting organization for receipt of all subsequent information regarding this RFP and the submission. The name of this contact will be made available to all OMG members. The LOI is typically due 60 days before the deadline for initial submissions. LOIs must be sent by fax or paper mail to the "RFP Submissions Desk" at the main OMG address shown on the first page of this RFP.

Here is a suggested template for the Letter of Intent:

This letter confirms the intent of <organization required> (the organization) to submit a response to the OMG <RFP name required> RFP. We will grant OMG and its members the right to copy our response for review purposes as specified in section 4.7 of the RFP. Should our response be adopted by OMG we will comply with the OMG Business Committee terms set out in section 4.4 of the RFP and in document omg/06-03-02.

<contact name and details required> will be responsible for liaison with OMG regarding this RFP response.

The signatory below is an officer of the organization and has the approval and authority to make this commitment on behalf of the organization.

<signature required>

4.4 Business Committee RFP Attachment

This section contains the text of the Business Committee RFP attachment concerning commercial availability requirements placed on submissions. This attachment is available separately as an OMG document omg/06-03-02.

Commercial considerations in OMG technology adoption

A1 Introduction

OMG wishes to encourage rapid commercial adoption of the specifications it publishes. To this end, there must be neither technical, legal nor commercial obstacles to their implementation. Freedom from the first is largely judged through technical review by the relevant OMG Technology Committees; the second two are the responsibility of the OMG Business Committee. The BC also looks for evidence of a commitment by a submitter to the commercial success of products based on the submission.

A2 Business Committee evaluation criteria

A2.1 Viable to implement across platforms

While it is understood that final candidate OMG submissions often combine technologies before they have all been implemented in one system, the Business Committee nevertheless wishes to see evidence that each major feature has been implemented, preferably more than once, and by separate organisations. Pre-product implementations are acceptable. Since use of OMG specifications should not be dependant on any one platform, cross-platform availability and interoperability of implementations should be also be demonstrated.

A2.2 Commercial availability

In addition to demonstrating the existence of implementations of the specification, the submitter must also show that products based on the specification are commercially available, or will be within 12 months of the date when the specification was recommended for adoption by the appropriate Task Force. Proof of intent to ship product within 12 months might include:

- *A public product announcement with a shipping date within the time limit.*

- *Demonstration of a prototype implementation and accompanying draft user documentation.*

Alternatively, and at the Business Committee's discretion, submissions may be adopted where the submitter is not a commercial software provider, and therefore will not make implementations commercially available. However, in this case the BC will require concrete evidence of two or more independent implementations of the specification being used by end- user organisations as part of their businesses. Regardless of which requirement is in use, the submitter must inform the OMG of completion of the implementations when commercially available.

A2.3 Access to Intellectual Property Rights

OMG will not adopt a specification if OMG is aware of any submitter, member or third party which holds a patent, copyright or other intellectual property right (collectively referred to in this policy statement as "IPR") which might be infringed by implementation or recommendation of such specification, unless OMG believes that such IPR owner will grant a license to organisations (whether OMG members or not) on non-discriminatory and commercially reasonable terms which wish to make use of the specification. Accordingly, the submitter must certify that it is not aware of any claim that the specification infringes any IPR of a third party or that it is aware and believes that an appropriate non-discriminatory license is available from that third party. Except for this certification, the submitter will not be required to make any other warranty, and specifications will be offered by OMG for use "as is". If the submitter owns IPR to which an use of a specification based upon its submission would necessarily be subject, it must certify to the Business Committee that it will make a suitable license available to any user on non- discriminatory and commercially reasonable terms, to permit development and commercialisation of an implementation that includes such IPR.

It is the goal of the OMG to make all of its technology available with as few impediments and disincentives to adoption as possible, and therefore OMG strongly encourages the submission of technology as to which royalty-free licenses will be available. However, in all events, the submitter shall also certify that any necessary licence will be made available on commercially reasonable, non-discriminatory terms. The submitter is responsible for disclosing in detail all known restrictions, placed either by the submitter or, if known, others, on technology necessary for any use of the specification.

A2.4 Publication of the specification

Should the submission be adopted, the submitter must grant OMG (and its sublicensees) a world- wide, royalty-free licence to edit, store, duplicate and

distribute both the specification and works derived from it (such as revisions and teaching materials). This requirement applies only to the written specification, not to any implementation of it.

A2.5 Continuing support

The submitter must show a commitment to continue supporting the technology underlying the specification after OMG adoption, for instance by showing the BC development plans for future revisions, enhancement or maintenance.

4.5 Responding to RFP items

4.5.1 Complete proposals

A submission must propose full specifications for all of the relevant requirements detailed in Chapter 6 of this RFP. Submissions that do not present complete proposals may be at a disadvantage.

Submitters are highly encouraged to propose solutions to any optional requirements enumerated in Chapter 6.

4.5.2 Additional specifications

Submissions may include additional specifications for items not covered by the RFP that they believe to be necessary and integral to their proposal. Information on these additional items should be clearly distinguished.

Submitters must give a detailed rationale as to why these specifications should also be considered for adoption. However submitters should note that a TF is unlikely to consider additional items that are already on the roadmap of an OMG TF, since this would pre-empt the normal adoption process.

4.5.3 Alternative approaches

Submitters may provide alternative RFP item definitions, categorizations, and groupings so long as the rationale for doing so is clearly stated. Equally, submitters may provide alternative models for how items are provided if there are compelling technological reasons for a different approach.

4.6 Confidential and Proprietary Information

The OMG specification adoption process is an open process. Responses to this RFP become public documents of the OMG and are available to members and non-members alike for perusal. No confidential or proprietary information of any kind will be accepted in a submission to this RFP.

4.7 Copyright Waiver

Every submission document must contain: (i) a waiver of copyright for unlimited duplication by the OMG, and (ii) a limited waiver of copyright that allows each OMG member to make up to fifty (50) copies of the document for review purposes only. See Section 4.9.2 for recommended language.

4.8 Proof of Concept

Submissions must include a "proof of concept" statement, explaining how the submitted specifications have been demonstrated to be technically viable. The technical viability has to do with the state of development and maturity of the technology on which a submission is based. This is not the same as commercial availability. Proof of concept statements can contain any information deemed relevant by the submitter; for example:

> "This specification has completed the design phase and is in the process of being prototyped."
>
> "An implementation of this specification has been in beta-test for 4 months."
>
> "A named product (with a specified customer base) is a realization of this specification."

It is incumbent upon submitters to demonstrate the technical viability of their proposal to the satisfaction of the TF managing the evaluation process. OMG will favor proposals based on technology for which sufficient relevant experience has been gained.

4.9 Format of RFP Submissions

This section presents the structure of a submission in response to an RFP. *All submissions* must contain the elements itemized in section 4.9.2 below before they can be accepted as a valid response for evaluation or a vote can be taken to recommend for adoption.

4.9.1 General

- Submissions that are concise and easy to read will inevitably receive more consideration.
- Submitted documentation should be confined to that directly relevant to the items requested in the RFP. If this is not practical, submitters must make clear what portion of the documentation pertains directly to the RFP and what portion does not.
- The key words "**must**", "**must not**", "**required**", "**shall**", "**shall not**", "**should**", "**should not**", "**recommended**", "**may**", and "**optional**" shall be used in the submissions with the meanings as described in RFC 2119 [RFC2119].

4.9.2 Required Outline

A three-part structure for submissions is required. Part I is non-normative, providing information relevant to the evaluation of the proposed specification. Part II is normative, representing the proposed specification. Specific sections like Appendices may be explicitly identified as non-normative in Part II. Part III is normative specifying changes that must be made to previously adopted specifications in order to be able to implement the specification proposed in Part II.

PART I

- •A cover page carrying the following information (a template for this is available [Inventory]):
  - The full name of the submission
  - The primary contact for the submission
  - The acronym proposed for the specification (e.g. UML, CORBA)
  - The name and document number of the RFP to which this is a response
  - The document number of the main submission document
  - An inventory of all accompanying documents, with OMG document number, short description, a URL where appropriate, and whether they are normative.
- List of OMG members making the submission (see 4.1) listing exactly which members are making the submission, so that submitters can be matched with LOI responders and their current eligibility can be verified.

- Copyright waiver (see 4.7), in a form acceptable to the OMG.

  One acceptable form is:

  "Each of the entities listed above: (i) grants to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version, and (ii) grants to each member of the OMG a nonexclusive, royalty-free, paid up, worldwide license to make up to fifty (50) copies of this document for internal review purposes only and not for distribution, and (iii) has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used any OMG specification that may be based hereon or having conformed any computer software to such specification."

  If you wish to use some other form you must get it approved by the OMG legal counsel before using it in a submission.

- For each member making the submission, an individual contact point who is authorized by the member to officially state the member's position relative to the submission, including matters related to copyright ownership, etc. (see 4.3)

- Overview or guide to the material in the submission

- Overall design rationale (if appropriate)

- Statement of proof of concept (see 4.8)

- Resolution of RFP requirements and requests

  Explain how the proposal satisfies the specific requirements and (if applicable) requests stated in Chapter 6. References to supporting material in Part II should be given.

  In addition, if the proposal does not satisfy any of the general requirements stated in Chapter 5, provide a detailed rationale.

- Responses to RFP issues to be discussed

  Discuss each of the "Issues To Be Discussed" identified in Chapter 6.

PART II

The contents of this part should be structured based on the template found in [FORMS] and should contain the following elements as per the instructions in the template document cited above:

- Scope of the proposed specification
- Proposed conformance criteria

  Submissions should propose appropriate conformance criteria for implementations.

- Proposed normative references

  Submissions should provide a list of the normative references that are used by the proposed specification

- Proposed list of terms and definitions

  Submissions should provide a list of terms that are used in the proposed specification with their definitions.

- Proposed list of symbols

  Submissions should provide a list of special symbols that are used in the proposed specification together with their significance

- Proposed specification

PART III

- Changes or extensions required to existing OMG specifications

  Submissions must include a full specification of any changes or extensions required to existing OMG specifications. This should be in a form that enables "mechanical" section-by-section revision of the existing specification.

4.10 How to Submit

Submitters should send an electronic version of their submission to the *RFP Submissions Desk* (*omg-documents@omg.org*) at OMG Headquarters by 5:00 PM U.S. Eastern Standard Time (22:00 GMT) on the day of the Initial and Revised Submission deadlines. Acceptable formats are Adobe FrameMaker source, ODF (ISO/IEC 26300), OASIS Darwin Information Typing Architecture (DITA) or OASIS DocBook 4.x (or later).

Submitters should make sure they receive electronic or voice confirmation of the successful receipt of their submission. Submitters should be prepared to send a single hardcopy version of their submission, if requested by OMG staff, to the attention of the "RFP Submissions Desk" at the main OMG address shown on the first page of this RFP.

5.0 General Requirements on Proposals

5.1 Requirements

5.1.1 Submitters are encouraged to express models using OMG modeling languages such as UML, MOF, CWM and SPEM (subject to any further constraints on the types of the models and modeling technologies specified in Chapter 6 of this RFP). Submissions containing models expressed via OMG modeling languages shall be accompanied by an OMG XMI [XMI] representation of the models (including a machine-readable copy). A best effort should be made to provide an OMG XMI representation even in those cases where models are expressed via non-OMG modeling languages.

5.1.2 Chapter 6 of this RFP specifies whether PIM(s), PSM(s), or both are being solicited. If proposals specify a PIM and corresponding PSM(s), then the rules specifying the mapping(s) between the PIM and PSM(s) shall either be identified by reference to a standard mapping or specified in the proposal. In order to allow possible inconsistencies in a proposal to be resolved later, proposals shall identify whether the mapping technique or the resulting PSM(s) are to be considered normative.

5.1.3 Proposals shall be *precise* and *functionally complete*. All relevant assumptions and context required for implementing the specification shall be provided.

5.1.4 Proposals shall specify *conformance criteria* that clearly state what features all implementations must support and which features (if any) may *optionally* be supported.

5.1.5 Proposals shall *reuse* existing OMG and other standard specifications in preference to defining new models to specify similar functionality.

5.1.6 Proposals shall justify and fully specify any *changes or extensions* required to existing OMG specifications. In general, OMG favors proposals that are *upwards compatible* with existing standards and that minimize changes and extensions to existing specifications.

5.1.7 Proposals shall factor out functionality that could be used in different contexts and specify their models, interfaces, etc. separately. Such *minimalism* fosters re-use and avoids functional duplication.

5.1.8 Proposals shall use or depend on other specifications only where it is actually necessary. While re-use of existing specifications to avoid duplication will be encouraged, proposals should avoid gratuitous use.

5.1.9 Proposals shall be *compatible* with and *usable* with existing specifications from OMG and other standards bodies, as appropriate. Separate specifications offering distinct functionality should be usable together where it makes sense to do so.

5.1.10 Proposals shall preserve maximum *implementation flexibility*. Implementation descriptions should not be included and proposals shall not constrain implementations any more than is necessary to promote interoperability.

5.1.11 Proposals shall allow *independent implementations* that are *substitutable* and *interoperable*. An implementation should be replaceable by an alternative implementation without requiring changes to any client.

5.1.12 Proposals shall be compatible with the architecture for system distribution defined in ISO's Reference Model of Open Distributed Processing [RM-ODP]. Where such compatibility is not achieved, or is not appropriate, the response to the RFP must include reasons why compatibility is not appropriate and an outline of any plans to achieve such compatibility in the future.

5.1.13 In order to demonstrate that the specification proposed in response to this RFP can be made secure in environments requiring security, answers to the following questions shall be provided:

- What, if any, are the security sensitive elements that are introduced by the proposal?
- Which accesses to security-sensitive elements must be subject to security policy control?
- Does the proposed service or facility need to be security aware?
- What default policies (e.g., for authentication, audit, authorization, message protection etc.) should be applied to the security sensitive elements introduced by the proposal? Of what security considerations must the implementers of your proposal be aware?

The OMG has adopted several specifications, which cover different aspects of security and provide useful resources in formulating responses. [CSIV2] [SEC] [RAD].

5.1.14 Proposals shall specify the degree of internationalization support that they provide. The degrees of support are as follows:

a) Uncategorized: Internationalization has not been considered.

b) Specific to <region name>: The proposal supports the customs of the specified region only, and is not guaranteed to support the customs of any other region. Any fault or error caused by requesting the services outside of a context in which the customs of the specified region are being consistently followed is the responsibility of the requester.

c) Specific to <multiple region names>: The proposal supports the customs of the specified regions only, and is not guaranteed to support the customs of any other regions. Any fault or error caused by requesting the services outside of a context in which the customs of at least one of the specified regions are being consistently followed is the responsibility of the requester.

d) Explicitly not specific to <region(s) name>: The proposal does not support the customs of the specified region(s). Any fault or error caused by requesting the services in a context in which the customs of the specified region(s) are being followed is the responsibility of the requester.

5.2 Evaluation criteria

Although the OMG adopts model-based specifications and not implementations of those specifications, the technical viability of implementations will be taken into account during the evaluation process. The following criteria will be used:

5.2.1 Performance

Potential implementation trade-offs for performance will be considered.

5.2.2 Portability

The ease of implementation on a variety of systems and software platforms will be considered.

5.2.3 Securability

The answer to questions in section 5.1.13 shall be taken into consideration to ascertain that an implementation of the proposal is securable in an environment requiring security.

5.2.4 Conformance: Inspectability and Testability

The adequacy of proposed specifications for the purposes of conformance inspection and testing will be considered. Specifications should provide sufficient constraints on interfaces and implementation characteristics to ensure

that conformance can be unambiguously assessed through both manual inspection and automated testing.

5.2.5 Standardized Metadata

Where proposals incorporate metadata specifications, usage of OMG standard XMI metadata [XMI] representations must be provided as this allows specifications to be easily interchanged between XMI compliant tools and applications. Since use of XML (including XMI and XML/Value [XML/Value]) is evolving rapidly, the use of industry specific XML vocabularies (which may not be XMI compliant) is acceptable where justified.

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.

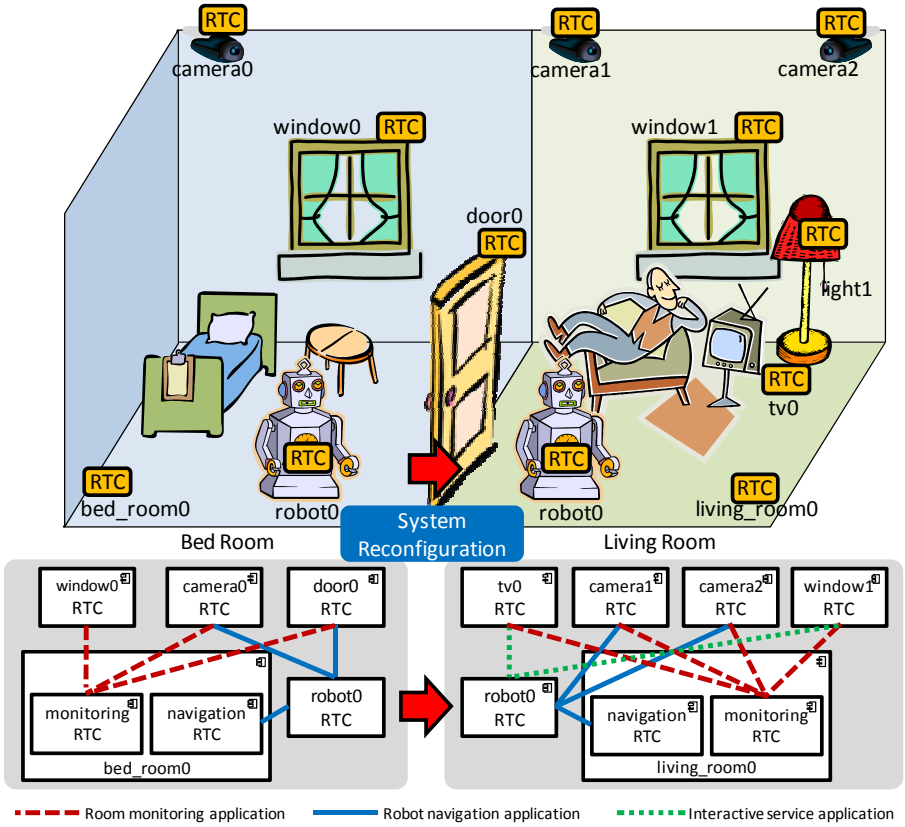


Figure 1 Typical robotic application environment

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.

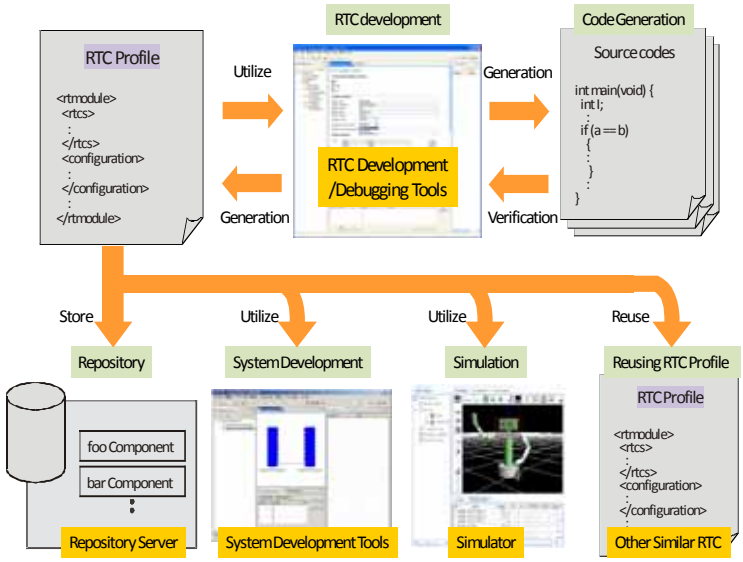


Figure 2 Use of the RTC Profile

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.

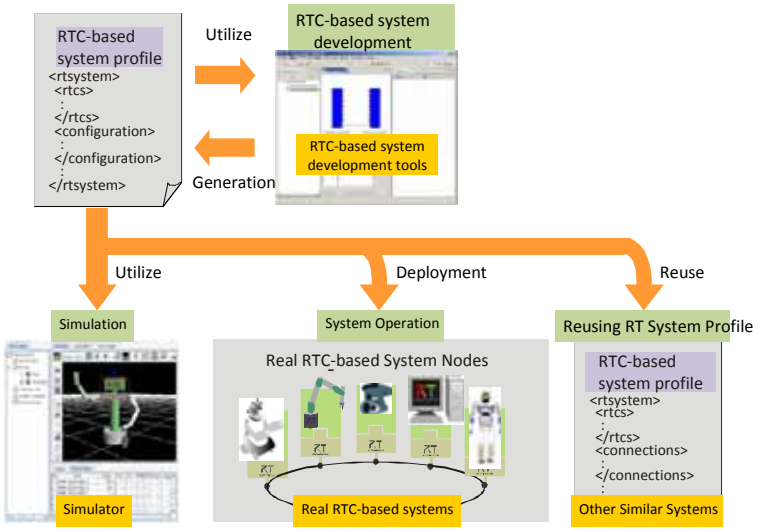


Figure 3 Use of RTC-based system profile

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 (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 defines installation, configuration, planning, preparation, and launch process for component-based applications. D&C 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 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 or 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 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].

Platform independent RTC information model

6.5.4 Proposals shall provide a schema describing RTC characteristics.

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.

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 [D&C] 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, 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, 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*

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] 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 (DDC) for RTC RFP draft

document number:
- mars/2010-03-05 (errata)
- mars/2010-03-06 (with change bar)
- mars/2010-03-07 (convenience)

presentation: mars/2010-03-21

Noriaki Ando
Infrastructure WG, Robotics DTF
National Institute of Advanced Industrial Science and Technology

Comments for 4-weeks document, and modification

According to the comments from AB members….

- Notation problems
  - Abbreviations, font color, etc…
- 6.5 Mandatory Requirements
- 6.6 Optional Requirements
- 6.7 Issue to be Discussed
  - “Out of focus”. No dynamic features are described.
- Schedule
  - The initial-submission was delayed by one meeting.

Comments at AB Plenary Meeting

- Mandatory requirements becomes ambiguous
  - The RFP no longer asks for interfaces in 6.5.1.
  - There are many meta-models and data-modes. Relation among them is unclear.
  - Proposed PSM must be clearly specified.
- Other comments
  - Direct-URL should be shown for specification references.
  - At the “Objective” section, RT should be Robotic Technology
  - CORBA IDL should be OMG IDL
  - "identify to locate" in 6.5.3 should be "identify and locate."

Modifications

- The name of the specification has been changed.
  - Old: DDR (Deployment and Dynamic Reconfiguration)
  - New: DDC (Dynamic Deployment and Configuration)
- According to comments at the AB plenary, mandatory requirements have been revised again.
  - More concrete and more clear
  - Consists of two parts
    - Platform independent deployment and configuration model
    - Platform independent RTC information model
- 6.6 Optional Requirements
  - An optional requirement has been moved from mandatory requirements.
- 6.7 Issues to be discussed
- 6.8 Evaluation Criteria

Mandatory Requirements (1)

Platform independent deployment and configuration model

- 6.5.1 Proposals shall specify services and interfaces 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 (2)

Platform independent RTC information model

- 6.5.4 Proposals shall provide a schema describing RTC characteristics.
- 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.

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*

Robotics Domain Task Force Preliminary Agenda ver0.0.2 **robotics/2010-03-28**

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

http://robotics.omg.org/

		TF/SIG				
		Host	Joint (Invited)	Agenda Item	Purpose	Room
Monday: Robotics Plenary(am) and WG activites(pm)						
9:30	10:00	Robotics		Robotics-DTF Plenary Opening Session	Robotics plenary openning	
10:00	11:00	MARS		Dynamic Deployment and Configuration (DDC) for RTC RFP 3rdReview - Noriaki Ando (AIST) and Jae-Yeong Jung(ETRI)	3rd Review Joint with **MARS**	
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					
13:00	18:00			Architecture Board Plenary		
13:00	17:00			Robotic Infrastructure WG (4h) - Noriaki Ando(AIST) and Seung-Woog Jung(ETRI)	discussion	
				Robotic Functional Services WG(4h): - Su-Young Chi , Miki Sato and Toshio Hori	discussion	
Tuesday: WG activities (am) and Robotics Plenary (pm)						
9:00	12:00			Robotic Infrastructure WG (3h) - Noriaki Ando(AIST) and Seung-Woog Jung(ETRI)	discussion	
				Robotic Functional Services WG(3h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	
12:00	13:00	**LUNCH**				
13:00	15:00	Robotics		Modeling in Robotics - Laurent Rioux (Thales)	presentation and discussion	
				Break (30min)		
15:30	16:30	Robotics		WG Reports and Discussion (Service WG, Profile WG, Robotic Localization Service WG)	presentation and discussion	
16:30	17:10	Robotics		Contact Reports: - Makoto Mizukawa(Shibaura-IT), and Young-Jo Cho(ETRI)	Information Exchange	
17:10	17:30	Robotics		Robotics-DTF Plenary Wrap-up Session (Roadmap and Next meeting Agenda)	Robotics plenary closing	
17:30				Adjourn joint plenary meeting		
17:30	18:00			Robotics WG Co-chairs Planning Session (Preliminary Agenda for next TM, Draft report for Friday)	planning for next meeting	
Wednesday WG activity follow-up						
9:00	12:00			Robotic Functional Services WG(3h): - Su-Young Chi, Hyunsoo Kim and Toshio Hori	discussion	
12:00	14:00	**LUNCH and OMG Plenary**				
14:00	17:00			Robotic Functional Services WG(3h): - Su-Young Chi, Hyunsoo Kim and Toshio Hori	discussion	
18:00	20:00	**OMG Reception**				
Thursday Robotics-DTF Plenary						
9:00	10:00	MARS		Dynamic Deployment and Configuration (DDC) for RTC RFP 3rd Review - Noriaki Ando and Jae-Yeong Jung	Vote to Issue Joint with **MARS**	
10:00	10:30	Robotics		Robotic Interaction Service (RoIS) Framework RFP 2nd Review - Su-Young Chi, Miki Sato and Toshio Hori	Vote to Issue	
10:30	12:00			Robotics WG activity follow-up	discussion	
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)		

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

Robotics-DTF

Date: Friday, 26th March, 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: (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]
- 3 Contact Reports [robotics/2010-03-15,-16,-17]
- Preliminary agenda for upcoming meeting [robotics/2010-03-28]

Robotics-DTF

Date: Friday, 26th March, 2010
Chair:, T. Kotoku, Y. –J. Cho and L. Rioux
URL: http://robotics.omg.org/
email: robotics@omg.org

➢Deliverables from this Meeting:

- Nothing Special

➢Future deliverables (In-Process):

- Dynamic Deployment and Configuration (DDC) for RTC RFP thru MARS-PTF
- Robotic Interaction Service (RoIS) Framework RFP

➢Next Meeting (Minneapolis, MN, USA):

- 3rd Review of DDC for RTC RFP
- 2nd Review of RoIS Framework RFP
- Guest presentations
- Roadmap discussion
- Contact reports

Minutes of the Robotics DTF Meeting - DRAFT

March 22-26, 2010

Jacksonville, FL, USA

(robotics/2010-03-30)

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 And o)
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 And o)
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 m ars/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 A ndo) [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 revie wed 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(A TR)
- 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 consortiu m.

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).