Robotics Domain Task Force Final Agenda ver.1.0.1 robotics/2010-12-01

OMG Technical Meeting - Santa Clara, CA, USA -- December 6-10, 2010

http://robotics.omg.org/

		TF/SIG				
		Host	Joint (Invited)	Agenda Item	Purpose	Room
Sunday: WG activites(pm)						
13:00	17:00			Robotics DDC4RTC and RoIS RFP submitters meeting	Arrangement	Napa 1, Lobby Lv
Monday: Robotics Plenary(am) and WG activites(pm)						
8:40	9:00	Robotics		Robotics-DTF Plenary Opening Session	Robotics plenary openning	Winchester, 2nd FL
9:00	10:00	Robotics		Inititial Submission Presentations for RoIS RFP - Su-Young Chi (ETRI)	presentation and discussion	
				Break (15min)		
10:15	11:15	Robotics		Inititial Submission Presentations for RoIS RFP - Miki Sato(JARA/ATR) and Toshio Hori (JARA/AIST)	presentation and discussion	
11:15	12:00	Robotics		**Special Talk**: (how we are) **Building Blocks for Mobile Manipulation** (that you can reuse) **- Brian Garkey (Willow Garage)**	Robotics plenary closing	
12:00	13:00	**LUNCH**				Magnolia, Lobby Lv
13:00	18:00			Robotic Infrastructure WG (5h): - Noriaki Ando (AIST) and Seung-Woog Jung (ETRI)	discussion	Tasman, 2nd FL
				Robotic Functional Services WG(5h): - Su-Young Chi(ETRI), Miki Sato(JARA/ATR) and Toshio Hori(AIST)	discussion	San Tomas, 2nd FL
Tuesday: WG activity						
9:00	12:00			Robotic Infrastructure WG(3h) - Noriaki Ando and Seung-Woog Jung	discussion	Bayshore W., 2nd FL
				Robotic Functional Services WG(3h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	Bayshore E., 2nd FL
12:00	13:00	**LUNCH**				Magnolia, Lobby Lv
13:00	18:00			Robotic Infrastructure WG(5h) - Noriaki Ando and Seung-Woog Jung	discussion	Bayshore W., 2nd FL
				Robotic Functional Services WG(5h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	Bayshore E., 2nd FL
Wednesday: WG activitiy(am) and Robotics-DTF Plenary(pm)						
9:00	12:00			Robotic Functional Services WG(3h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	Bayshore E., 2nd FL
12:00	14:00	**LUNCH and OMG Plenary**				Magnolia, Lobby Lv
14:00	16:00	MARS		Joint Plenary with MARS: Inititial Submission Presentations for DDC4RTC - (ETRI) and (AIST)	presentation and discussion	Laurence, 2nd FL
16:00	16:40	Robotics		WG Reports and Discussion (Service WG, Infrastructure WG, Models in Robotics WG)	presentation and discussion	Bayshore E., 2nd FL
16:40	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	
18:00	20:00	**OMG Evening Reception**				East/West Mezz, 2nd FL
Thursday: WG activitiy follow-up						
9:00	12:00			Robotics WG activity follow-up	discussion	Bayshore W., 2nd FL
12:00	13:00	**LUNCH**				
13:00	18:00			Architecture Board Plenary		Winchester, 2nd FL
Friday						
8:30	12:00			AB, DTC, PTC		Magnolia, Lobby Lv
12:00	13:00	**LUNCH**				

Other Meetings of Interest

Monday						
8:00	8:45	OMG		New Attendee Orientation		Lafayette, 2nd FL
9:00	12:00	OMG		Tutorial - Introduction to OMG's meeting and Middlewere Specifications		Lafayette, 2nd FL
8:00	12:00	OMG		Energy Stnadards Information Day		Cypass, 2nd FL
8:00	17:00	OMG		Telecom Cloud Information Day		Steave Creek, 2nd FL
Tuesday						
7:30	9:00	OMG		Liaison ABSC		Mendocino, Lobby Lv.
8:00	17:00	OMG		Telecom Cluoud Standards & Open Source Workshop		Steve Creek, 2nd FL
8:30	17:00	SE/SysM		System Engineering DSIG/SysML 1.3 RTF Joint Meeting		Napa 1, Lobby Lv.
17:00	18:00	OMG		RTF-FTF Chair's Workshop		Alameda, 2nd FL
Wednesday						
9:00	17:00	OMG		System Assurence PTF		Lafayette, 2nd FL
Thursday						
9:00	17:00	OMG		System Assurance PTF		Lafayette, 2nd FL
13:00	17:00	OMG		SysML-Modelica FTF		Alameda, 2nd FL

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

Robotics-DTF Plenary Meeting
Opening Session

December 6, 2010

Santa Clara, CA, USA

Hyatt Regency Hotel, Cambridge

Approval of Minutes

Meeting Quorum : 3

AIST, ETRI, JARA, Shibaura-IT, Technologic Arts,
Univ. of Electro-Communications, Univ. of Tsukuba,

Minutes taker(s):

Miki Sato
Myung-Eun Kim

Minutes review
=> on Wednesday

Cambridge Meeting Summary

Robotics Plenary: (3 participants)

- Introduction of RTC-CANopen (Makoto Mizukawa) [robotics/2010-09-04]

Agenda Review

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

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

Robotic Interaction Service Framework Initial Submission

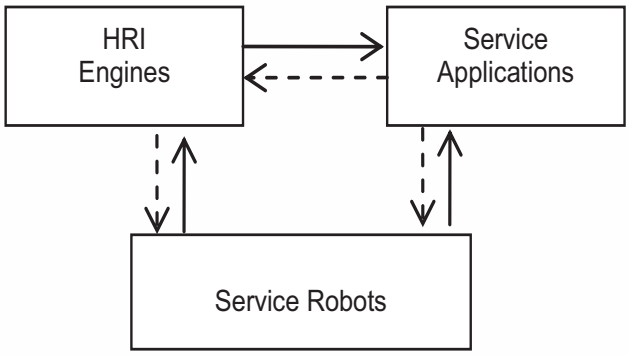
2010. 12. 05.

Functional Service WG
Dr. Su-Young Chi

ETRI
한국전자통신연구원

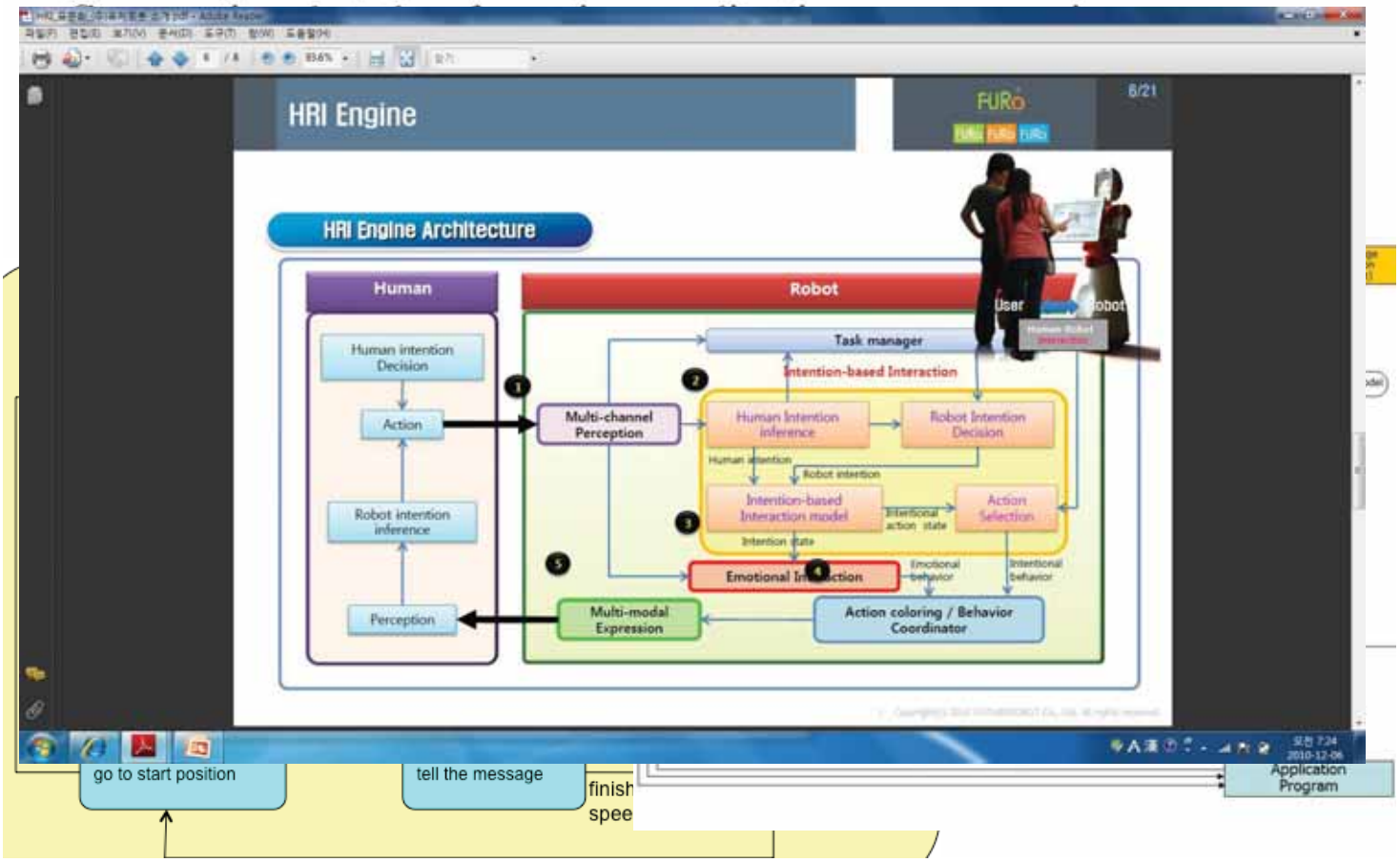
Overview

- Relationship of robotic interaction service framework components



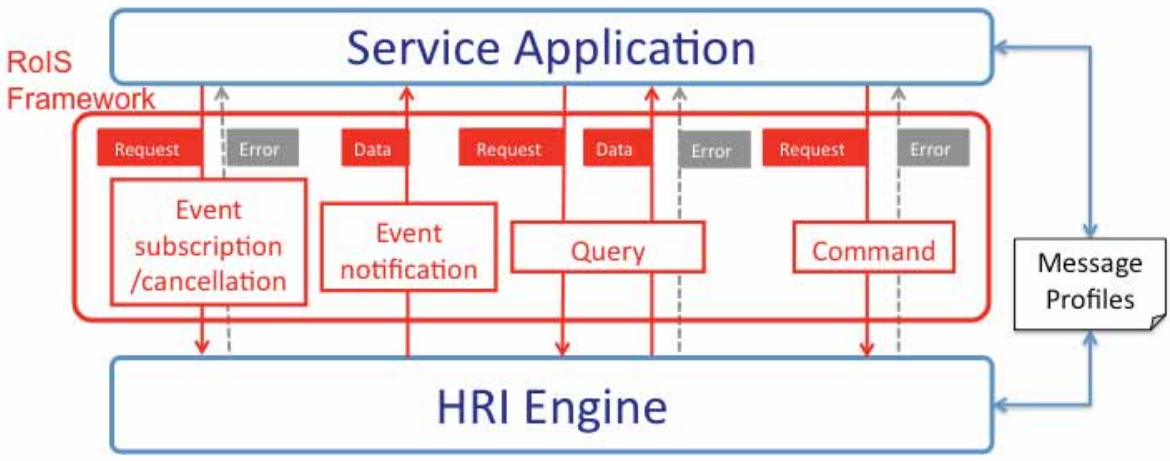
- The robotic interaction service framework should be flexible enough to accommodate a plurality of employed service robots and a wide range of service applications.
- The RoIS achieves the above flexibility by composing its internal structure through three basic components: HRI engines, service applications, service robots.

Overview



RoIS Framework

- In the framework, application communicates with HRI Engine through Event(s) subscription/cancellation, Event(s) notification, Query and Command.



RoIS Framework(on RFP)

- Define Common Messages
  - Event Message
    - ✓ Person detected
    - ✓ Person identified
    - ✓ Command action completed
  - Query Message
    - ✓ Person ID
    - ✓ Position of person/robot
    - ✓ Status of HRI Engine
  - Command Message
    - ✓ Start / stop person detection
    - ✓ Start / stop person identification

RoIS Framework (ETRI)Strategy

- HRI Specified Functions with BioAPI & RLS
  - BioAPI + HRI Specified Functions API
  - RLS + HRI Specified Functions API
  - Human Detection & Tracking
  - Biometric Data Detection (Face, Voice, Gait, etc)
  - Problems on Moving Platform
  - Robotic Applications
  - Conformance Methodology
- Standard Activity
  - ISO / IEC 19784 Amendment Part N : BioAPI-Robot
  - ISO 19111

How can we communicate between RoIS Framework and Service Application?

- Application Service to RoIS Framework
  - ✓ "Identify the person who have just asked to play a music"
  - ✓ "Where is your mother?"
  - ✓ "Who is calling me from the right hand side?"
  - ✓ "Who is that person visible from the camera image?"

- RoIS to Application ServiceFramework
  - "We have found (possible person's ID with likelihood list) from the 60 degree direction"
  - "(possible person's ID with likelihood list) is calling you from 130 degree direction"
  - "(possible person's ID with likelihood list) has disappeared from our camera view"

RoIS Framework

- Event Messages

✓ **Description**

- **Interface to receive information from HRI Engine according to the occurrences of the information in real time.**

✓ **Event List**

Event Name	Argument	Remarks
HD_MOTIONDETECTED	PosInfo Array pInfoArray	Occurrence of moving
HD_FACEDETECTED	PosInfo Array pInfoArray	Detection of face
HD_PERSONFOUND	PosInfo Array pInfoArray	Detection of person
HD_GESTURERECOGNIZED	GestureInfoArray gInfoArray	Recognition of gesture
HD_FACEIDENTIFIED	IDInfoArray iInfoArray	Recognition of face
HD_PERSONIDENTIFIED	IDInfoArray iInfoArray	Recognition of person
HD_SOUNDDETECTED	PosInfo pInfo	Detection of sound
HD_SPECIFICSOUNDDETECTED	SoundInfo sdInfo	Detection of specific sound
HD_SPEAKERRECOGNIZED	IDInfo iInfo	Recognition of speaker
HD_SPEECHRECOGNIZED	SpeechInfo spInfo	Recognition of speech
HD_COMETOMEFINISHED		Finishing of come to me service
HD_FOLLOWMEFINISHED		Finishing of follow me service

RoIS Framework

- Event Messages
  - ✓ RoIS Framework needs events for user disappearance, since it has the user tracking feature.
    - PersonDisappeared
      - A user has disappeared (including multiple user cases)
    - SpecificUserDisappeared
      - Specific user that Application requested, has disappeared.
    - PersonInsideArea
      - When somebody approached within certain distance from the robot.

RoIS Framework

- Events more specific than "PersonFound", may be needed.
  - FaceDetected
    - The user's face is detected, but not identified (including the position information)
  - VoiceDetected
    - The user's voice is detected, but not identified (including the position information) – this is when the speech/non-speech discrimination is possible.
- Auxiliary information of the user recognition
  - UserGenderClassified
    - User is not identified, but the gender is classified.
  - UserAgeClassified
    - User is not identified, but the age is classified.

RoIS Framework

- Event Messages
  - ✓ **Data Structure**

```
typedef int Event;
EnrollInfo {
String m_strUserName;// user name
CDib *m_pMugFace;// user face image
};

RecognizerList {
int nID;// supported recognizer id
CString strDesrciption;// description of recognizer ("face",
"height" ...)
}

GestureList {
int nID;
CString strDesrciption;// description of gesture("waving", "calling" ...)
};

SoundList {
int nID;
CString strDesrciption;// description of sound ("voice", "clap" ...)
};
```


RoIS Framework

- Event Messages
  - ✓ **Data Structure**
    - UAMInfo {
      - Int UAMID;
      - Int Media; (such as image, sound, distance, human sensor)
      - BOOL NeedEnrollment;
      - };

RoIS Framework

✓ API

- int GetNumberOfUAM();
- UAMInfo GetUAMInfo(int nth);
  - Function for UAM Enumeration in the HRI Demon System
- BOOL Initialize()
- BOOL Destroy()
- Void SetProperties(UAMID, Properties p)
  - Property is used when certain information is to be set for a specific UAM
- Properties GetProperties(UAMID)
  - Get property information assigned for a specific UAM.
- BOOL EnrollUser(UAMID, UserInfo);
  - Enrol process is assigned to the UAM, including user interface for enrollment.
  - The result indicates success or fail, using BOOL.
  - The registered data is managed by the UAM itself.
- EnrollInfo GetEnrollmentData(UAMID, UserInfo);
- EnrollInfoArray GetEnrollmentData(UAMID);
  - These two functions are used when the registered data is needed for backup etc.
  - The first one is used for separate data, and the second one is used for all user's data registered at the UAM.
- BOOL DeleteEnrollment(UAMID, UserInfo);
  - This is used to delete the registered data.

RoIS Framework

✓ API

- BOOL AddCandidate(UserInfoArray);
  - To pre-set the specific users as the matching candidate. In this case, the users need to be pre-registered.
- BOOL RemoveCandidate(UserInfoArray);
  - To remove the specific user from the matching candidate list.
- UserInfoArray GetCandidateList();
  - To get the user list, registered as the matching candidate.
- UserInfoArray MatchUser(UAMID);
- UserInfoArray MatchUser();
  - To perform the user identification.
  - This can give command for user identification to a specific UAM.
  - This can also give command to all UAM available to UIC, and get the combined results.
  - When the user is more than one person, the return value is UserInfoArray (User ID with likelihood list and the position information may be transmitted.).

RoIS Framework

✓ API

- PositionInfo FindUser(UserInfo);
  - To find the specific user (if the user can be found, the position of that user can be returned) – even if the user is not found, the system may return the previous history of that user, such as “your mother has moved into the main bedroom five minutes ago”.
- UserInfoArray GetUserMap();
  - This returns the list of visible users, including the position info.

- Void SetEvent(UAMID, EventInfo, CallBack, OnOff);
  - This set or reset a certain Event.
  - This is the self-controlled Event of UIC to Application, without the request of Application.
  - It should be noted that only the pre-set Event may happen (pre-set Event : Events that was set by Application by “SetEvent”.)
- Void RaiseEvent(EventInfo);

RoIS Framework

● Query Messages

✓ Description

- Service application uses this interface to actively solicit information of command results, engine’s status, etc..

✓ API

PersonID:string
GetSpecificUserPos:string
Status of HRI Engine:string
GetUserMap:string
GetImage:string

✓ Data Structure

```
UserInfo {
IDInfo iInfo;
int nGestureType;// waving, calling, raising, stopping
int nSoundType;// clap, voice, whistle ...
CString strSpeech;
}
```

RoIS Framework

- Command Messages
  - ✓ **Description**
    - Service application uses this interface to instruct the HRI Engine to contral its HRI Components
  - ✓ **API**

    ComeToMe:bool
    FollowMe:bool
    StartPerDetection:string
    StopPerDetection:string
    StartPerIdentification:string
    StopPerIdentification:string

Feasibility Study – Case#1

- HRI Event Register

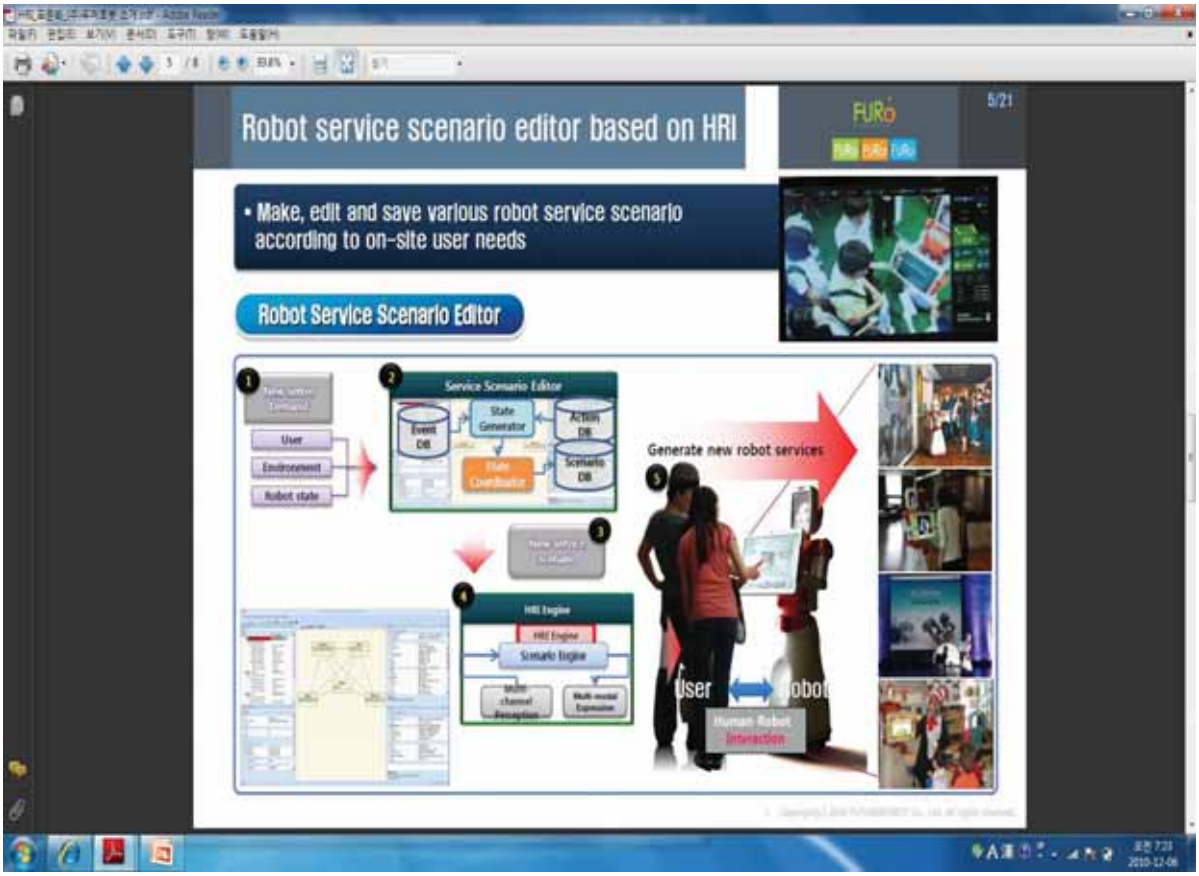
Feasibility Study – Case#1

- HRI Engine

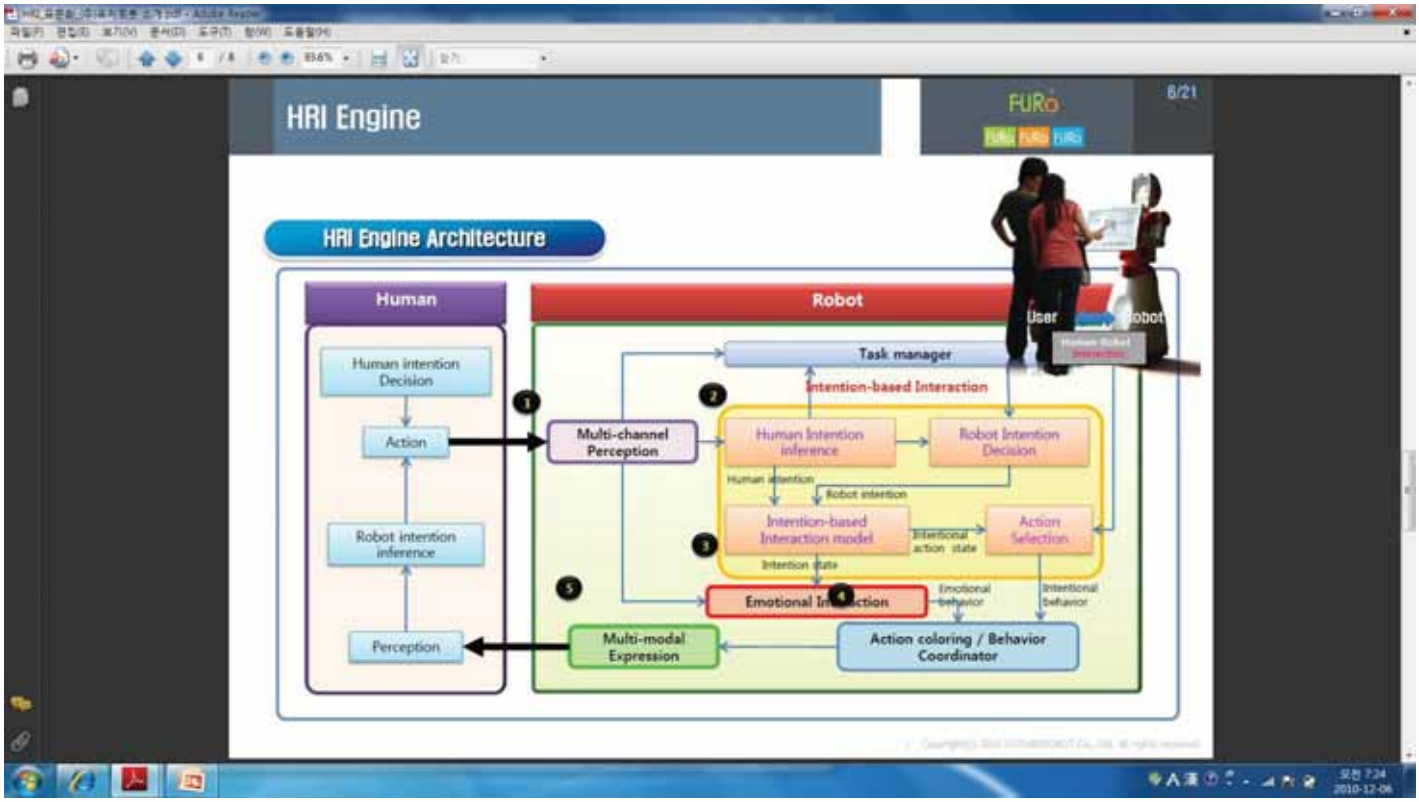
Feasibility Study – Case#2

Feasibility Study – Case#2

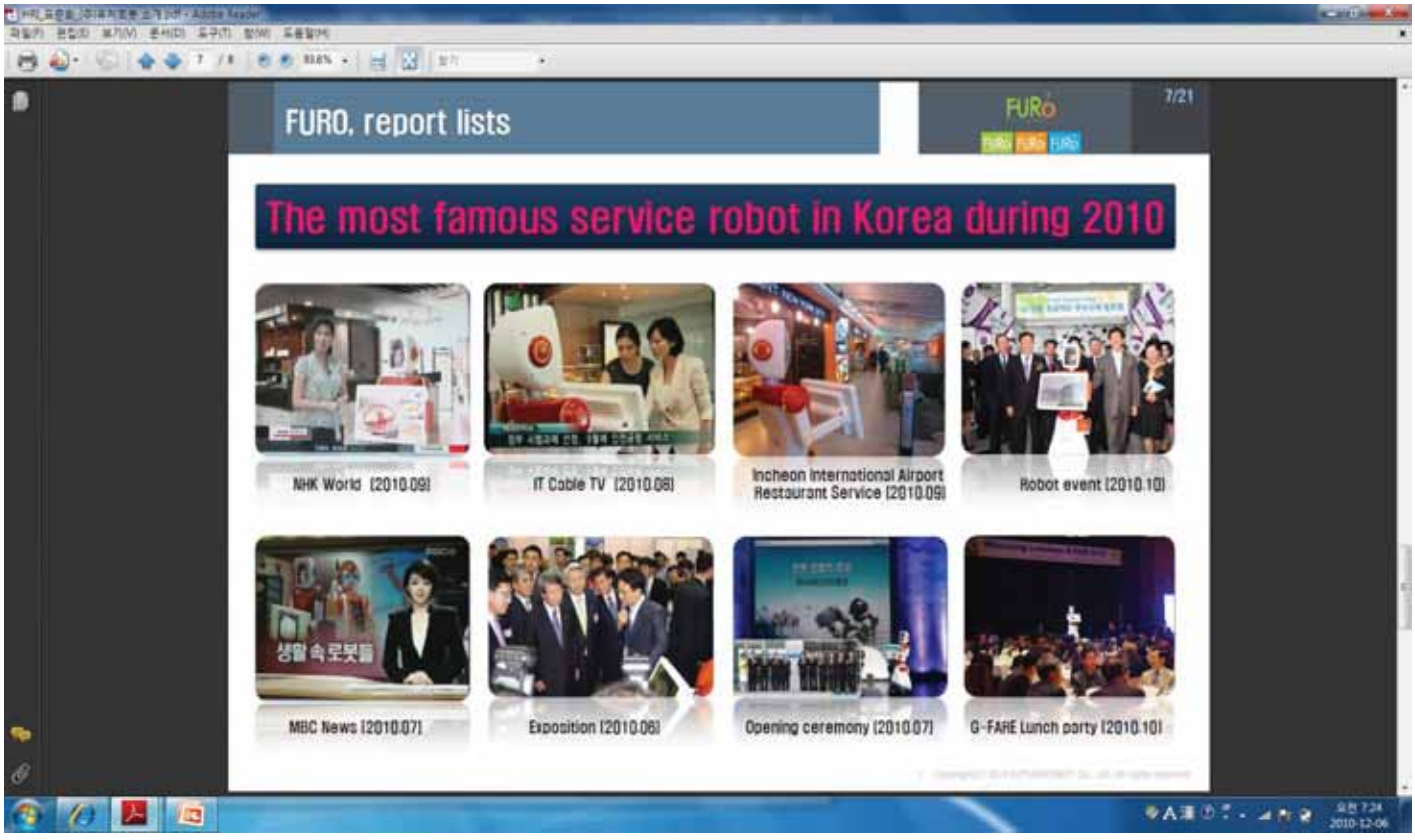
Feasibility Study – Case#2



Feasibility Study – Case#2



Feasibility Study – Case#2

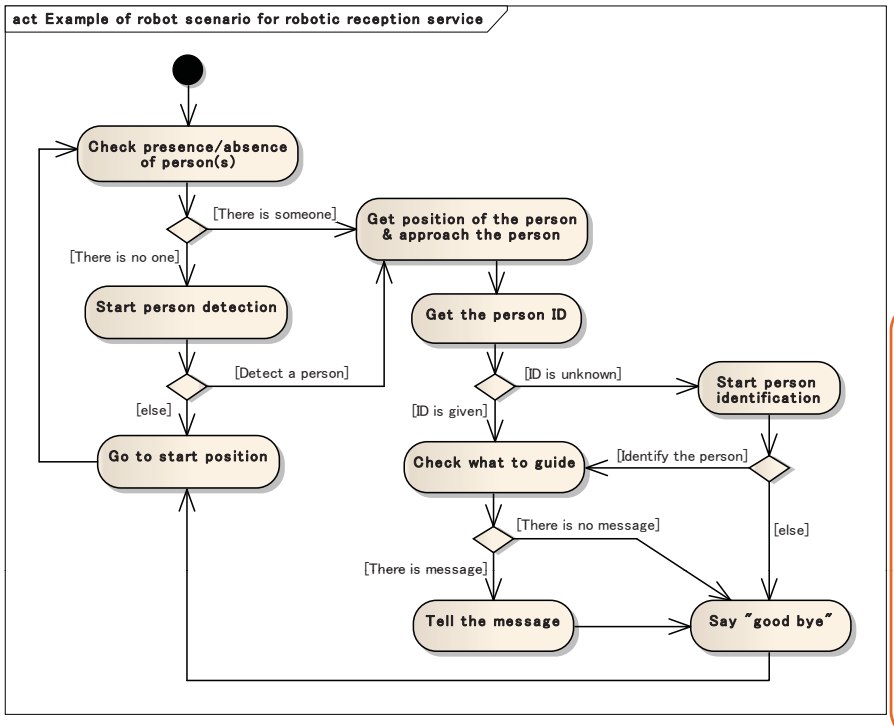


Feasibility Study – Case#2

JARA initial submission to Robotic Interaction Service (RoIS) Framework

Toshio Hori (JARA/AIST), Miki Sato (JARA/ATR)
Dec. 06, 2010

Example Scenario: Robotic Reception Service

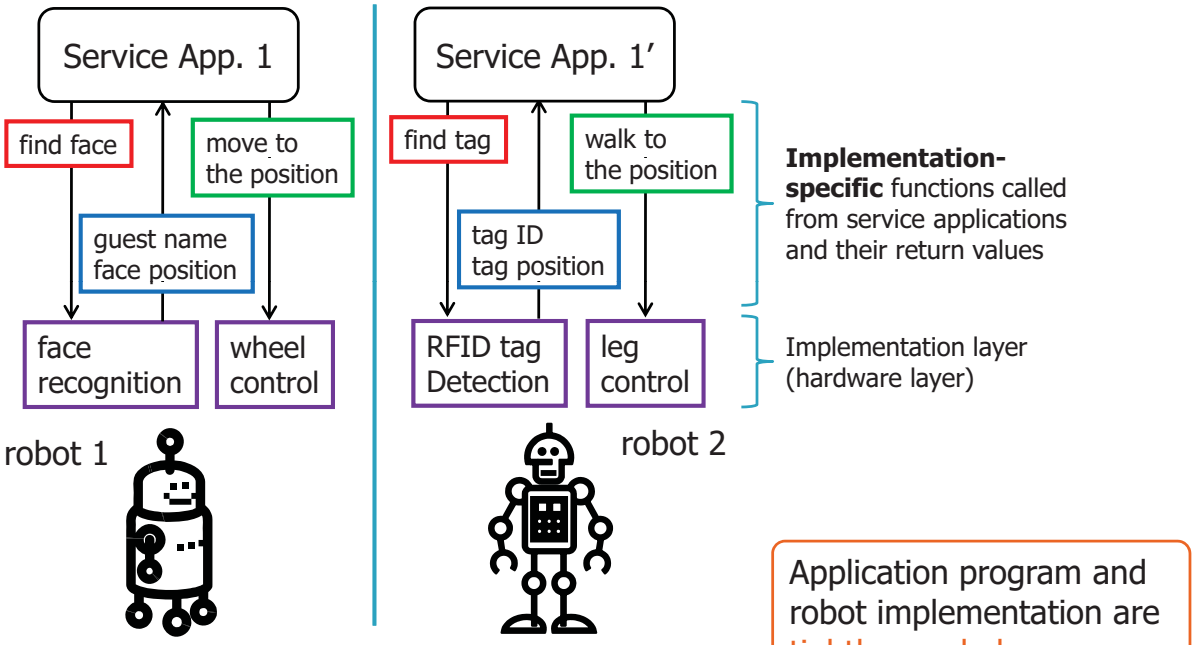


State transitions and robot commands in this scenario are described on the symbolic layer.

Robot applications are usually programmed by directly accessing hardware layer.

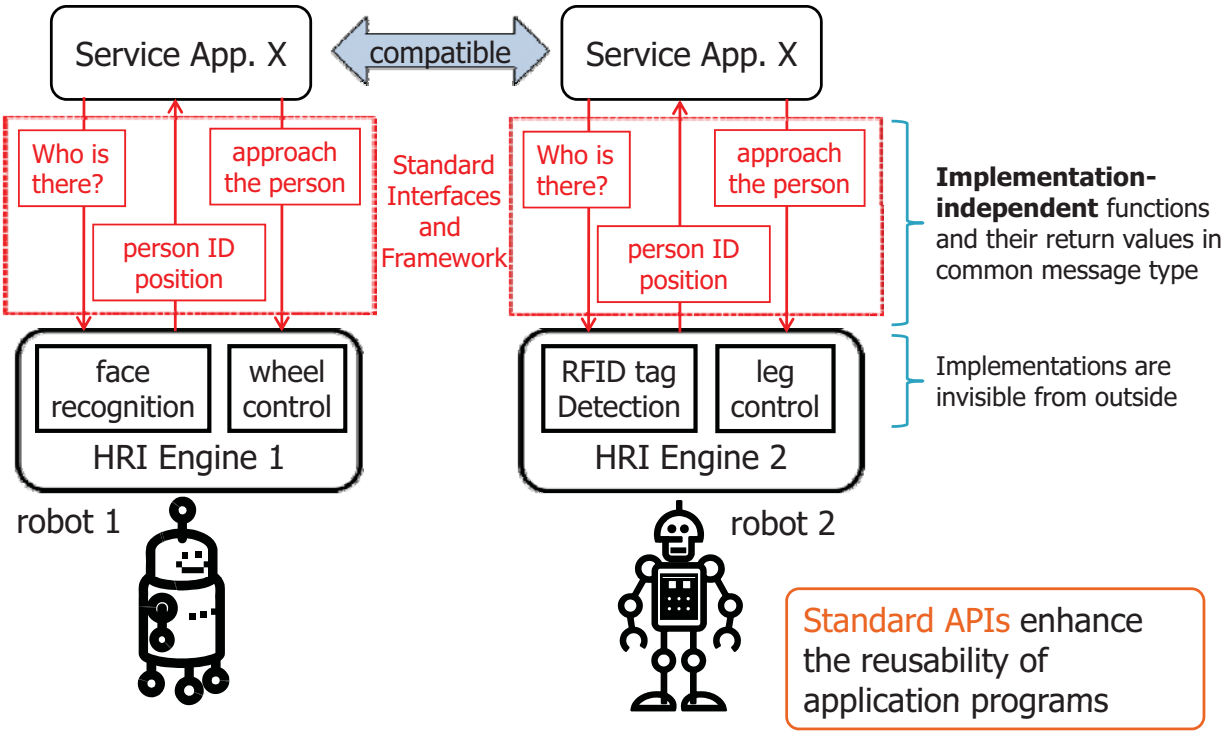
Reusability of service robot applications is low because they use "implementation-specific" functions.

Conventional Style of Service Robot Application Programs



Application program and robot implementation are tightly coupled

RoIS Service Application Programming Style



Standard APIs enhance the reusability of application programs

Problem Statement & Scope of RoIS Framework RFP

- Application program for a specific service robot does not run on other robots because:
  - Different robots may be equipped with a variety of types of sensors and actuators.
  - APIs of robots from different vendors vary even if they use the same sensor types.
- This lack of application portability is an obstacle to the success of the service robot industry.

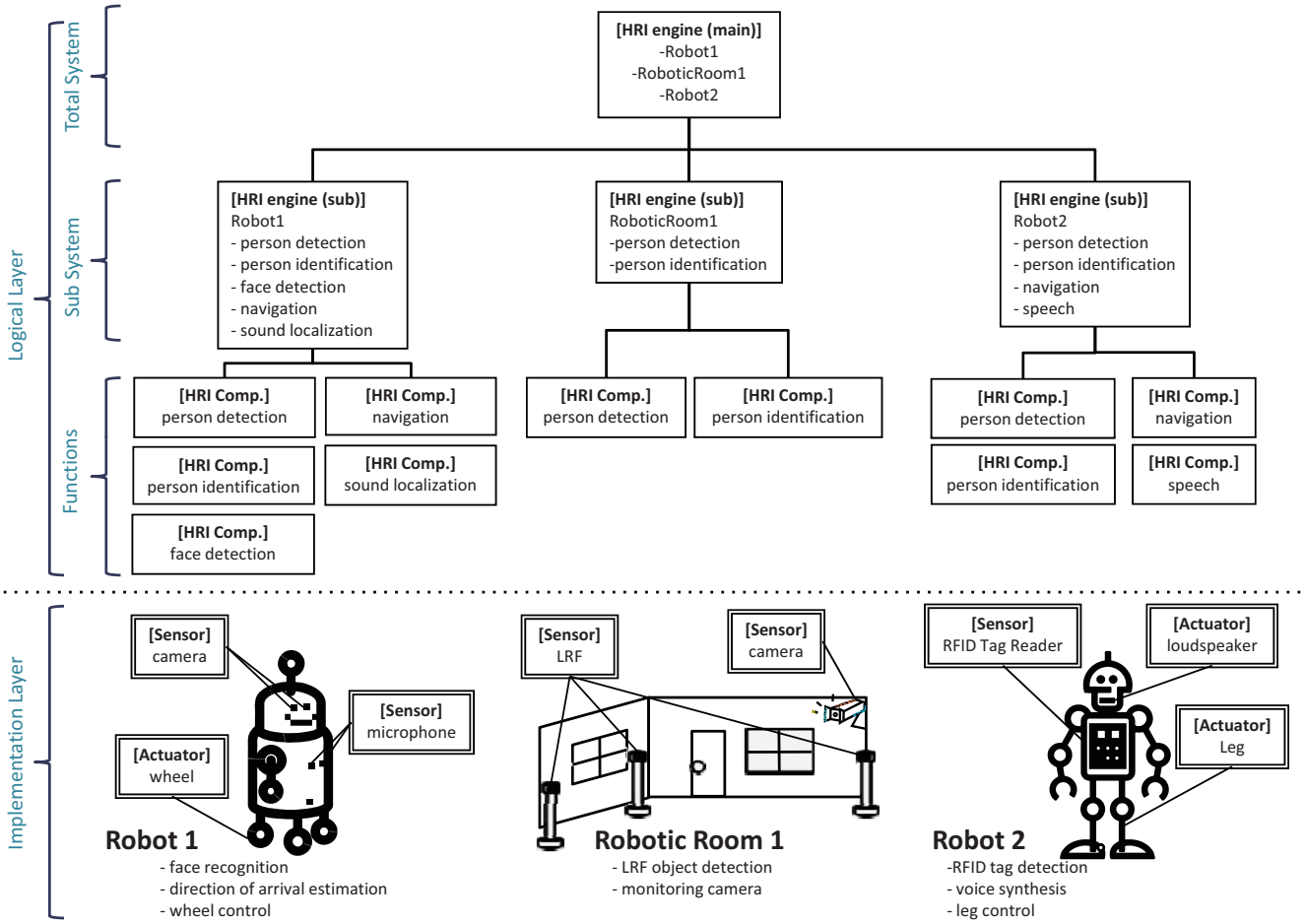
- Establishing a specification of **Robotic Interaction Service (RoIS) Framework** that provides
  - A set of common interfaces between service application and robot control program components,
  - A set of common messages exchanged through the common interfaces.

Introduction of JARA proposal

- JARA proposal is based on the extensive surveys and discussion among robotic researchers and developers from universities, research institutes and companies in Japan.
  - Interface types and message formats in this proposal were collected from actual service robot implementations developed by the participants.

Main Features:

- Implementation-independent interfaces to robotic components
  - 4 interface types: System Interface, Command Interface, Query Interface, and Event Interface.
  - Carefully-examined sequence of interaction
- Message formats extensible by message profiles
  - High interoperability with existing systems
  - Extensive use of existing standards



Terms

- **HRI Component**
  - HRI Component is a logical entity which provides an abstract function of hardware or functional implementations of a robot.
    - For example, a "Person Detection" HRI Component may use camera(s) and a face recognition module (functional implementation), or an RFID tag reader to detect person. Component users don't care its internal structure so long as its output data has the same format.
- **HRI Engine**
  - HRI Engine is an entity which encapsulates HRI Components and works as a logical implementation of a robot system.
  - It may include several HRI Engines. In this case, internal HRI Engines are called sub HRI Engines.

[PIM] Interface types in RoIS Framework

- **System Interface**
  - This interface manages the connection status between the service application and HRI Engine.
- **Command Interface**
  - This interface enables the service application to send commands to the HRI Engine.
- **Query Interface**
  - This interface enables the service application to query the HRI Engine on information it holds.
- **Event Interface**
  - This interface enables the service application to receive notifications on changes in HRI Engine status.

[PIM] Return Code and Returning Operation's Results

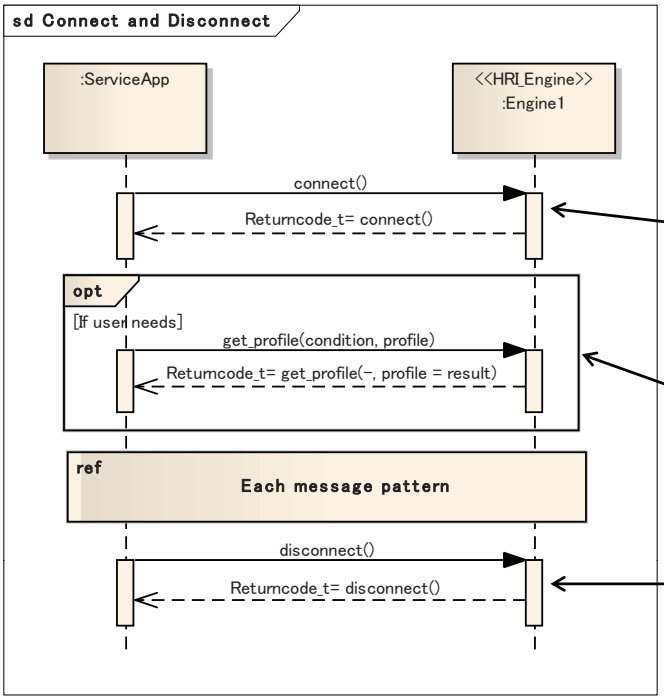
- **Synchronous operations** (Most System Interface & Query Interface)
  - Return Code: Indicates success/failure of operation.
  - Operation's Results: Sent back in parameter list of operation.
    - Ex. "out" parameters for CORBA PSM, references for C++ PSM
- **Asynchronous operations** (Most Command Interface & Event Interface)
  - Return Code: Indicates success/failure of receiving operation request.
  - Operation's Results: Retrieved by explicit query (basically).
    - Using query() [Query Interface], get_command_result() [Command Interface], get_error_detail() [System Interface] and get_event_detail() [Event Interface]

[PIM] Unique ID for each Command & Event

- **Command ID**
  - Assigned by HRI Engine each time execute() is issued.
  - A "completed" event is delivered from the engine upon finishing the command issued by execute(). The event includes the corresponding Command ID as its parameter.
  - Application may use "get_command_result()" by indicating ID to retrieve detailed information of command completed.

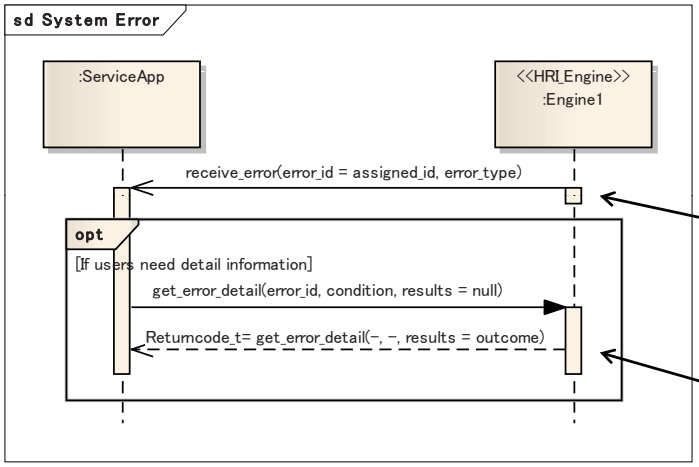
- **Event ID**
  - Assigned by HRI Engine each time the engine delivers event.
  - Application may use "get_event_detail()" by indicating ID to retrieve detailed information of the event.

[PIM] Sequence Diagram of System Interface (1/2)



- Methods of this interface type are called synchronously.
  - *connect()*
    - Interaction between service robot application and HRI Engine starts by this method.
  - *get_profile()*
    - The application may optionally use this method to obtain engine's profile.
  - *disconnect()*
    - Application closes the connection by this method.

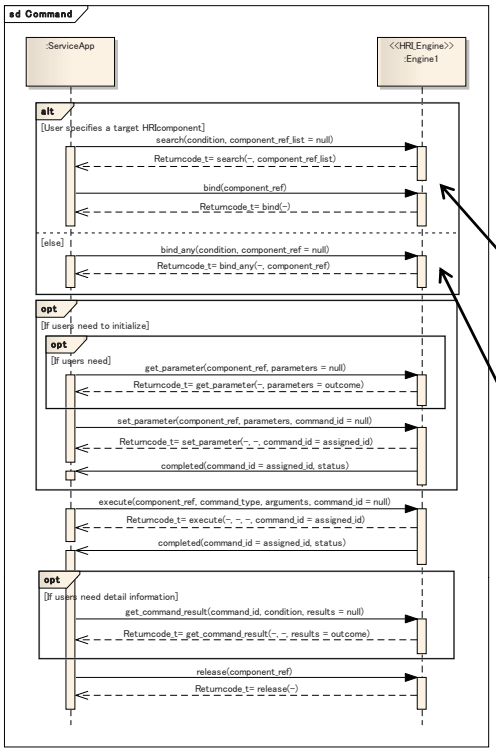
[PIM] Sequence Diagram of System Interface (2/2)



- Application receives system error(s) occurred in the HRI Engine or HRI Component(s).
  - Application cannot unsubscribe system error(s).
- *receive_error()*
  - HRI Engine notifies the occurrence of system error by this method.
  - This is an asynchronous call.
- *get_error_detail()*
  - Application can ask for detail information of the error by this method. (optional)

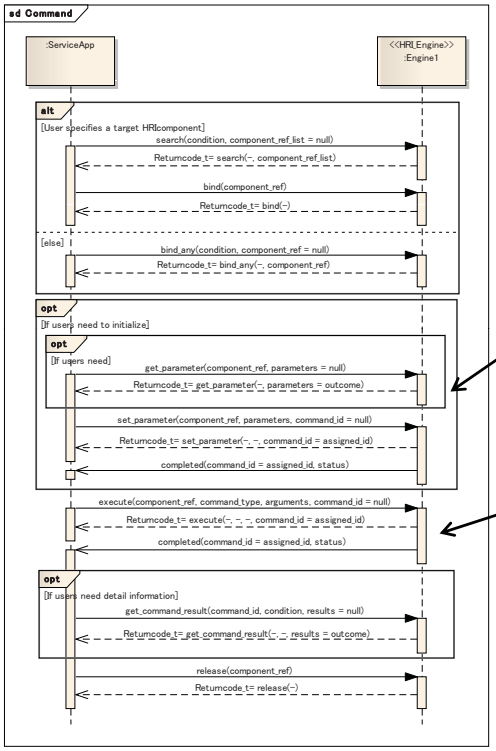
List of System Errors

[PIM] Sequence Diagram of Command Interface (1/3)



- Methods of this interface type are called asynchronously.
- Firstly, application selects or asks HRI Engine to assign an appropriate HRI Component being used in the following command execution.
- *search() / bind()*
  - Application asks HRI Engine to return a list of HRI Components which satisfy "search condition," selects a component from the list, then binds it for further command execution. Otherwise
- *bind_any()*
  - Application asks HRI Engine to assign an appropriate component automatically.

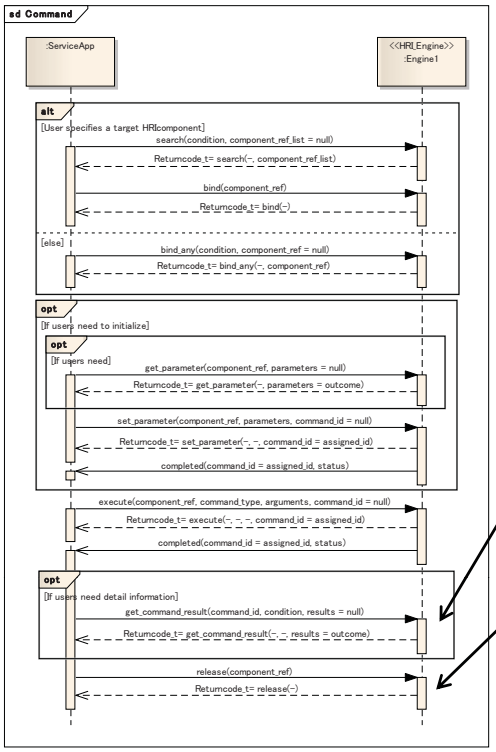
[PIM] Sequence Diagram of Command Interface (2/3)



- Application may get and set parameters of the component before issuing a command.
  - *get_parameter() / set_parameter()*
    - Application may obtain and set HRI Component parameters (optional).
  - *execute()*
    - Application issues a command to the component which was bound beforehand.
    - When the command finishes, a "completed" event is delivered.

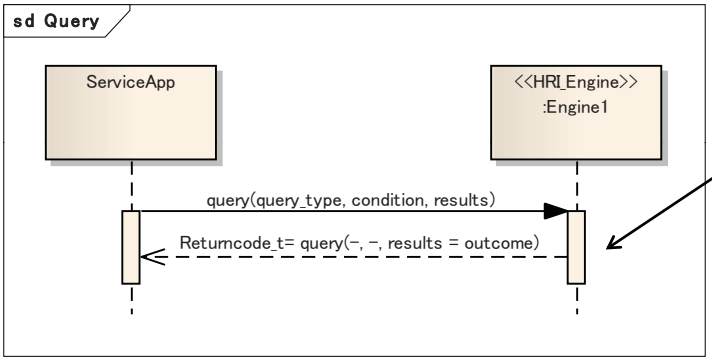
List of Completed Status

[PIM] Sequence Diagram of Command Interface (3/3)



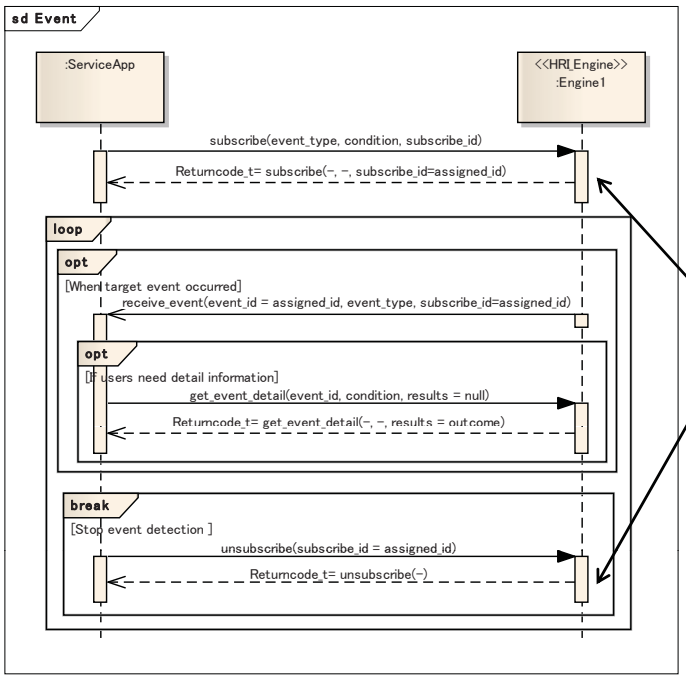
- After the command execution, application may ask the engine to return command result(s), then releases the component.
  - *get_command_result()*
    - When application requires the command result(s), it may optionally use this method to obtain it(them).
  - *release()*
    - This method releases an HRI Component which was bound by bind() or bind_any().

[PIM] Sequence Diagram of Query Interface



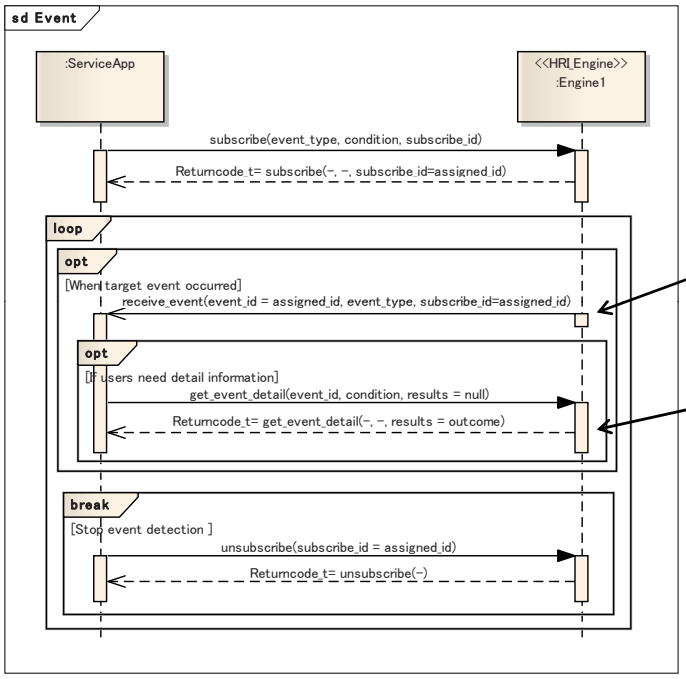
- This is a synchronous method because state transitions in a robot scenario are triggered by this result.
- *query()*
  - Application use this method to obtain current status or stored information of HRI Engine or HRI Components.

[PIM] Sequence Diagram of Event Interface (1/2)



- Application must subscribe to events which are of interest.
  - Without subscription, events are not delivered to application. This is a different point from system errors.
- *subscribe() / unsubscribe()*
  - Application uses these methods to subscribe to or unsubscribe events.

[PIM] Sequence Diagram of Event Interface (2/2)



- Methods of this interface type are basically asynchronous operations.
  - *receive_event()*
    - HRI Engine notifies the occurrence of event by this method.
  - *get_event_detail()*
    - Application retrieves details of events notified when it requires.

[PIM] Common HRI Components

HRI Component Name	Description
person_identification	Identifies person
person_detection	Detects presence and position of person
face_detection	Detects presence and position of person's face
sound_localization	Detects presence of sound and its arrival direction and originating position
self_localization	Detects position of robot (or installed sensor)
reaction	Performs affirmative, negative, or neutral reaction
navigation	Moves to specified location
gesture_recognition	Recognizes person's gesture
speech_recognition	Recognizes person's speech
speech	Generates speech

[PIM] Common Methods for HRI Components

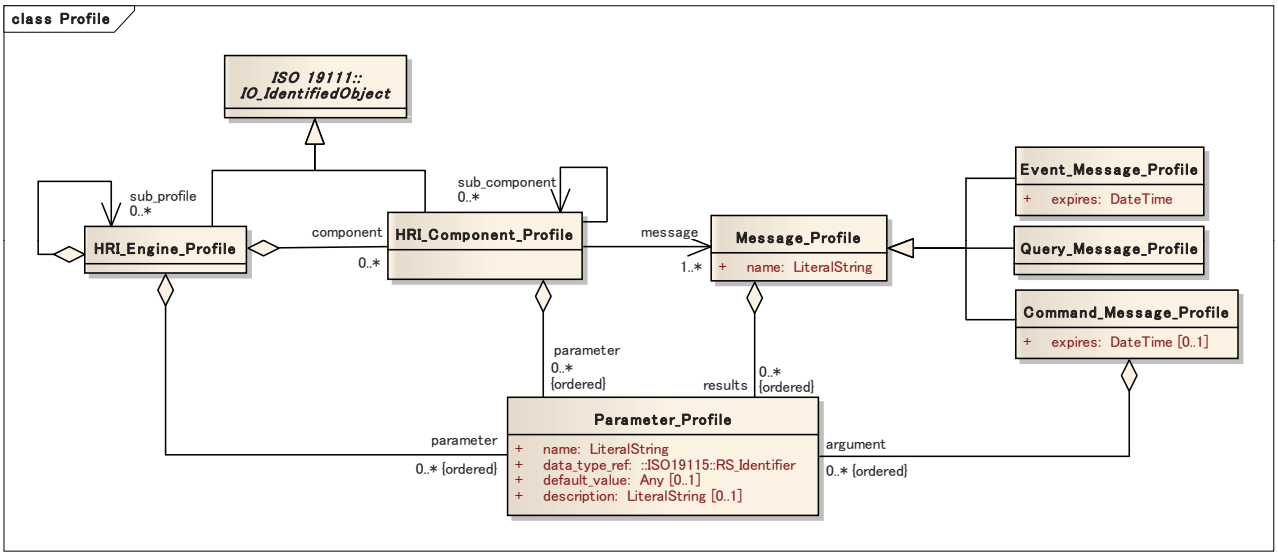
Description : basic methods for all HRI components.	
Command Method	
start	Start the functionality of the HRI component.
stop	Stop the functionality of the HRI component.
pause	Pause the functionality of the HRI component.
resume	Resume the functionality of the HRI component.
Query Method	
status	Obtain status information.

List of Component Status

[PIM] RoIS Profiles

- **HRI Engine Profile**
  - Defines a list of HRI Components and parameters possessed by an HRI Engine unit.
- **HRI Component Profile**
  - Defines a list of messages and parameters possessed by an HRI Component unit.
- **Message Profile**
  - Defines messages to be sent and received between the service application and HRI Engine via the RoIS Framework.
  - A base profile for Event/Query/Command Message Profiles.
- **Parameter Profile**
  - Defines the parameters of message arguments, results, the HRI Engine, and HRI Components.

[PIM] RoIS Profile Metamodel



AIST ATR JARA initial submission 2010/12/06

Related Standards

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

Resolution of RFP Mandatory Requirements (1/4)

1. Proposals shall provide a diagram or description giving an overview of the architecture, including the RoIS framework, robotic service application and HRI Engine (composed of robotic components).
   - The proposed overall architecture for the RoIS Framework is described in Section 7.

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

Note: All section numbers correspond to those in the JARA initial submission proposal (robotics/10-10-01).

Resolution of RFP Mandatory Requirements (2/4)

3. Proposals shall specify data structure for each interface
   - Data structure for each interface is described in Section 7.4.3.

4. Proposals shall provide a specification of the following common messages for all HRI Engines:
   - Event notification (message types are omitted)
   - Query (message types are omitted)
   - Command (message types are omitted)
     - Section 7.6 presents the specification of these common messages.

Resolution of RFP Mandatory Requirements (3/4)

5. The specification shall meet the following criteria:
   - Be general enough to incorporate robotic components for various sensors, actuators and algorithms in HRI Engine
   - Satisfy interoperability and reusability, to allow an HRI Engine to be replaced with another without difficulty
     - Message types proposed in this document are extensible by message profiles, whose structure is described in Section 7.5. In this specification, messages are defined independently of any specific sensors, actuators and algorithms.
     - HRI engine is treated at abstraction level that is independent of implementation. HRI engine will be replaced without difficulty. In addition, messages described in Section 7.6 include not only common messages presented in the Mandatory Requirements in the RFP (see above item No.4) but also messages that are common among several service robot systems which we surveyed. They will be interoperable and reusable when a certain PSM is defined

Resolution of RFP Mandatory Requirements (4/4)

6. Proposals shall specify existing technologies to achieve functions required for RoIS framework.
   - Section 4 refers to the existing technologies that realize RoIS Framework functions available.

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

8. Proposals shall provide a Platform Independent Model (PIM).
   - PIM is described in Section 7.

Resolution of RFP Optional Requirements (1/2)

1. Proposals may specify schemes for the functionalities listed below.
   - Error handling for each interface type
   - Returning command results and status
     - Error handling scheme for each interface type is described in Section 7.4.1.2, 7.4.1.3, and 7.4.1.4, respectively.
     - Schemes for returning command results and status are twofold as described in Section 7.4.1.2. One is return codes defined in Section 7.3 and the other is a parameter "status" in "completed" events defined in Section 7.4.1.2. A Query Interface "get_command_result" is also defined to retrieve command results explicitly.

2. Proposals may provide a schema to describe message profiles.
   - A schema to describe message profiles is described in Section 7.5. Annex A provides a possible description of concrete message profiles.

Resolution of RFP Optional Requirements (2/2)

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

Responses to RFP Issues to be Discussed (1/4)

- Proposals shall demonstrate their feasibility by giving a specific example application based on the proposed model.
  - This proposal is based on current robot implementations in Japan. The proposed model was carefully designed to hold the ability to describe the most complex systems, and at the same time, keep the compactness in implementation so as to be possible to be applied to embedded systems where resources are limited.
- Proposals shall demonstrate their applicability to existing robotic interaction service technologies.
  - As stated above, the proposed model was carefully designed to hold the ability to describe the existing robotic interaction service implementations in Japan.

Responses to RFP Issues to be Discussed (2/4)

- Proposals shall discuss simplicity of implementation.
  - One of the design policies of this proposal is that, while it preserves the ability to describe and realize complex systems, simple systems to be able to be described simply. Here we have paid much attention to keep the core, mandatory part of the model to be compact, and at the same time, to hold easy and versatile extendibility. This would allow the proposed model to describe the most complex systems, and at the same time, to be applied to embedded systems where resources are limited.
- Proposals shall discuss the possibility of applying the proposed model to other fields of interest such as intelligent service robot applications.
  - The proposed specification is designed to have the least dependency on interactive robot service usage, while concerns were made to make it suitable for controlling and handling constraints in the physical real-world environments. Thus, besides interactive service robots, this proposal can be easily applied to other fields of interest that controls physical entities or gathers information from real world, such as intelligent service robots or sensor network-driven applications.

Responses to RFP Issues to be Discussed (3/4)

- Proposals shall show that they are independent of on-the-wire protocol communication technology.
  - The proposed specification is carefully designed to be method-level or message-level definition so to be independent of any on-the-wire protocol communication technology.
- Proposals shall discuss their relationship to and dependency on existing middleware standards, such as CORBA [CORBA], DDS [DDS] or RTC [RTC].
  - This proposal requires several functionalities on its underlying communication protocols or middlewares. However, the requirements, such as event subscription facility, are general enough for modern software systems. Thus, we can say that the proposal holds no explicit dependency onto the underlying communication protocols or middleware. As can be seen in the PSM description in Section 8, mapping the proposed model to other existing communication protocols or middleware standards shall be trivial.

Responses to RFP Issues to be Discussed (4/4)

- Proposals shall discuss their generality with respect to the anticipated range of sensors, actuators and the algorithms in the HRI Engine.
  - The proposed specification is generalized for various sensors, actuators and algorithms by handling HRI engine and its functions independently of any specific hardware and algorithm as described in Section 1.7.1 item No. 5.

AIST ATR JARA initial submission 2010/12/06

[PIM] System Errors

ENGINE_INTERNAL_ERROR	An error internal to the engine.
COMPONENT_INTERNAL_ERROR	An error internal to the component.
COMPONENT_NOT_RESPONDING	No response received from the component.
USER_DEFINED_ERROR	An error defined by the user.
Note: Corresponding situations of these error types should be defined with respect to each HRI engine.	

[PIM] Completed Status

OK	Successful return.
ERROR	Generic, unspecified error.
ABORT	The operation was aborted.
OUT_OF_RESOURCES	Service ran out of the resources needed to complete the operation.
TIMEOUT	The operation timed out.
Note: Corresponding situations of these statuses should be defined with respect to each command message.	

[PIM] Component Status

UNINITIALIZED	The component is not initialized.
READY	The component is ready to use.
BUSY	The component is used by other application(s).
WARNING	Warning against the use of the component
ERROR	Generic, unspecified error.

(how we are) Building Blocks for Mobile Manipulation (that you can re-use)

Brian Gerkey

December 6, 2010

Willow Garage

Building Blocks: Why?

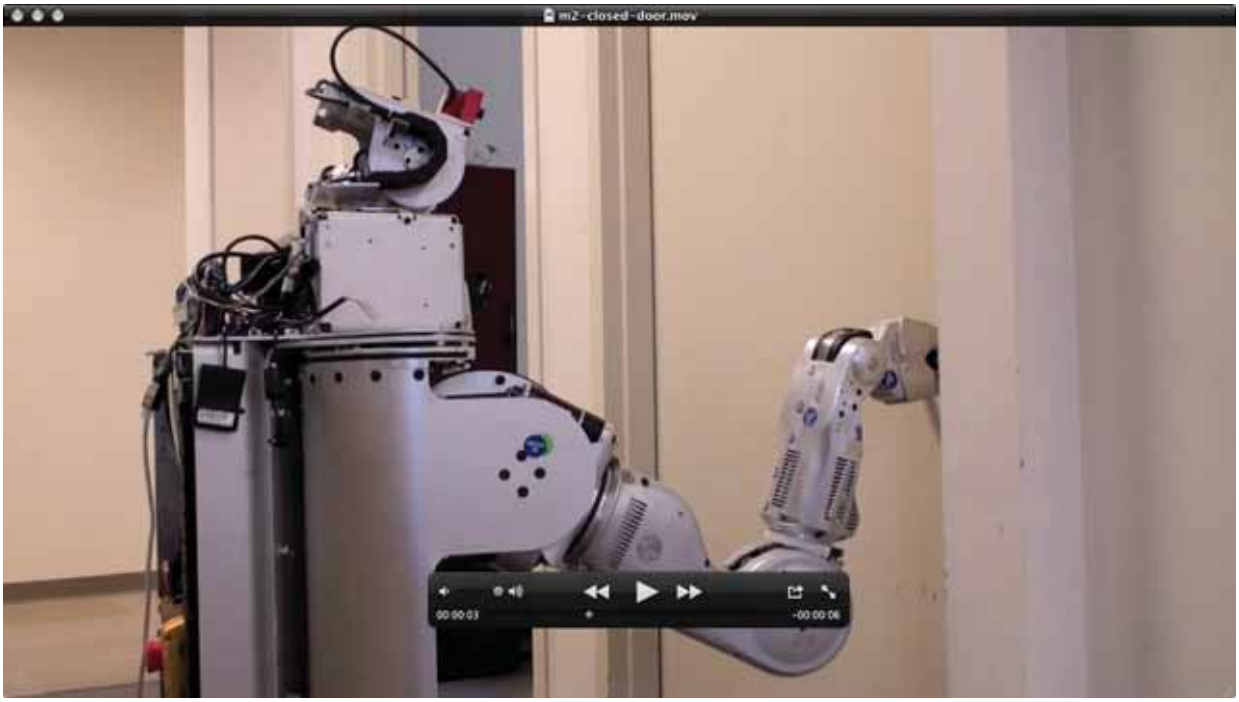
- Robots are more useful and easier to use if they have some basic capabilities
- Even “solved” problems entail significant implementation effort
- Important to know that your building block works
- Important to know that it **continues to work**
- Big benefit of packaging it up for reuse

Building Blocks: How?

Good old-fashioned software engineering principles:

- Document what you build
  - Even during fast-moving development; time spent on documentation is never wasted.
- Test everything you care about
  - **If it's not tested, it's broken**
  - Automate the tests; nobody will run them manually
- Make versioned releases for others to use
  - Insulate others from your ongoing development
  - Allow others to reproduce your experiments

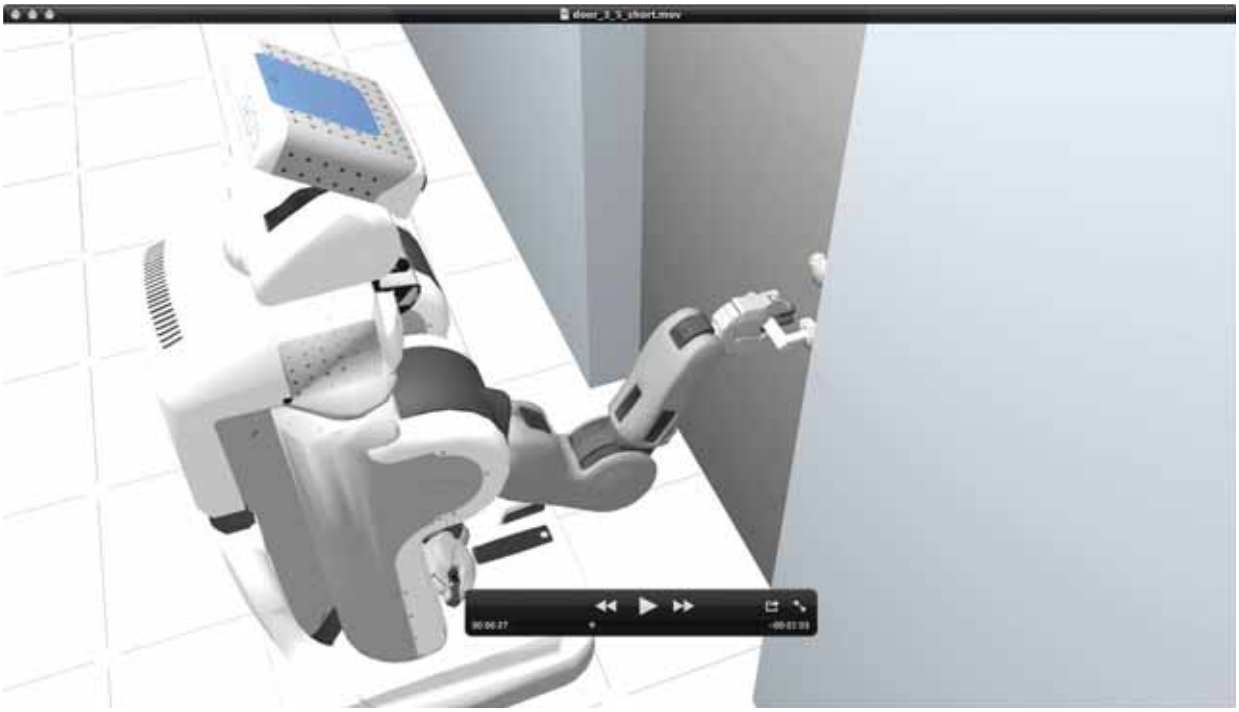
Opening a door



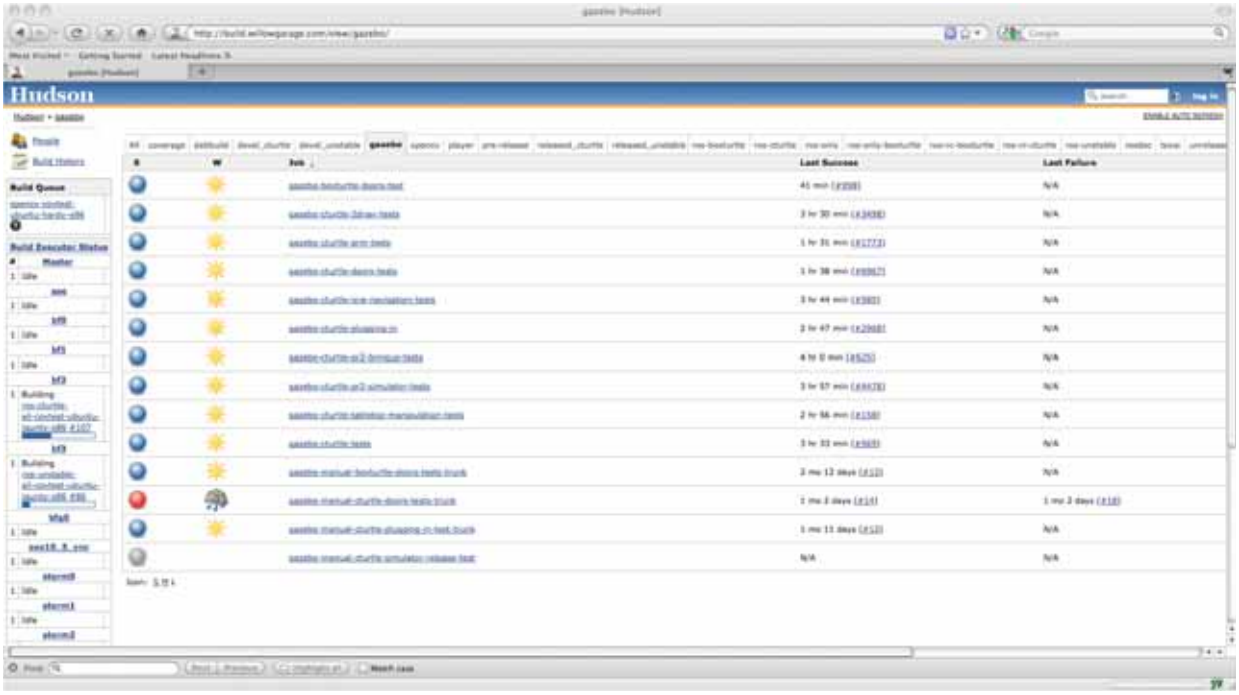
Opening a door

- Okay, the robot opens the door. How do you make sure that it continues to open the door as you make changes?

Opening a door: regression test

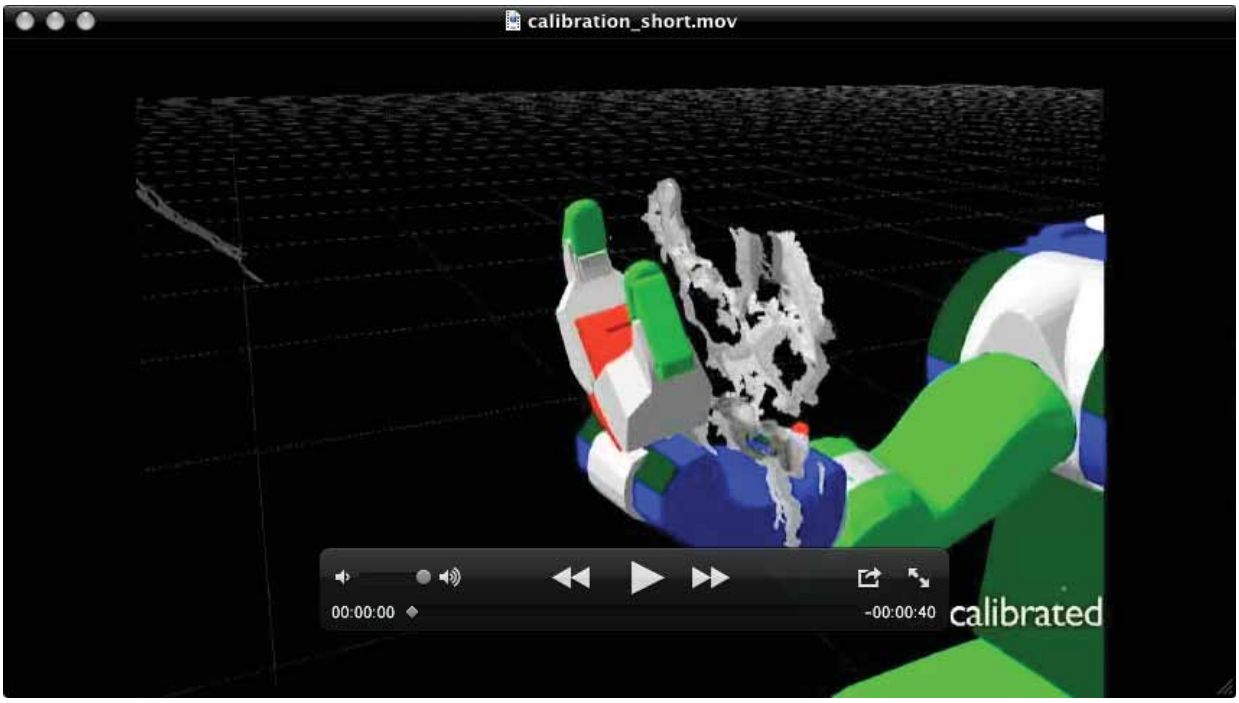


Automated testing: Hudson

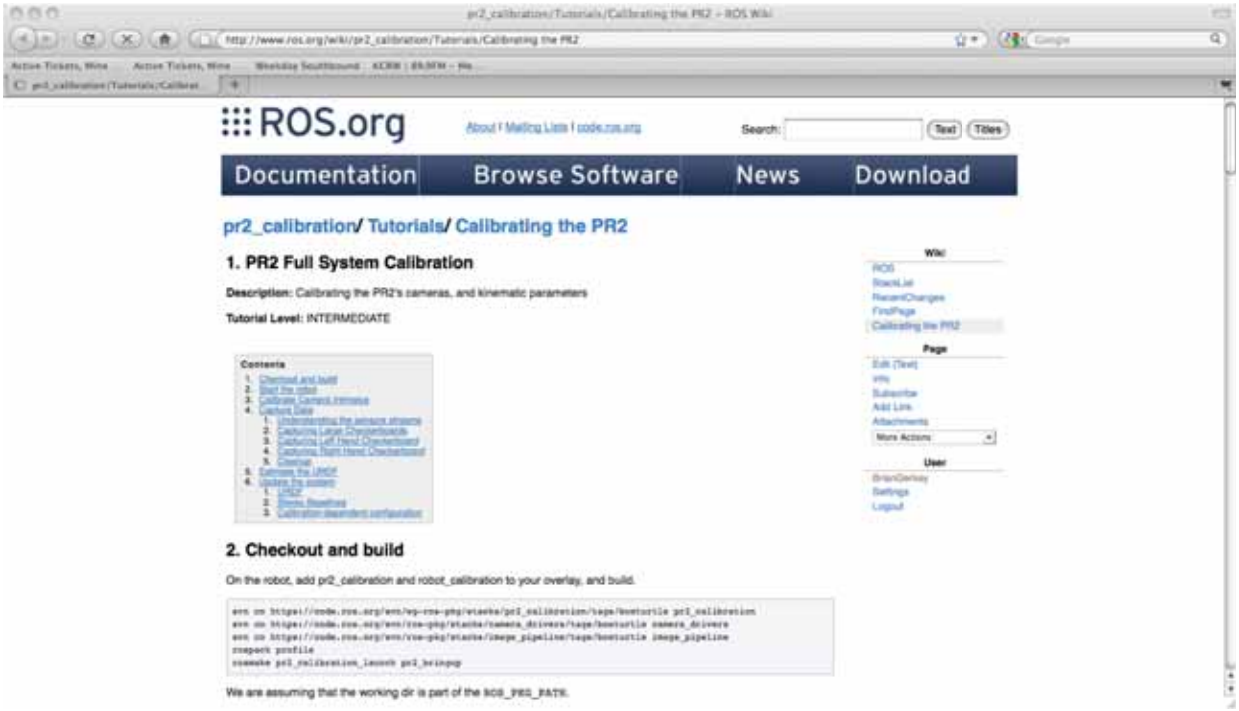


Plugging in

Calibration



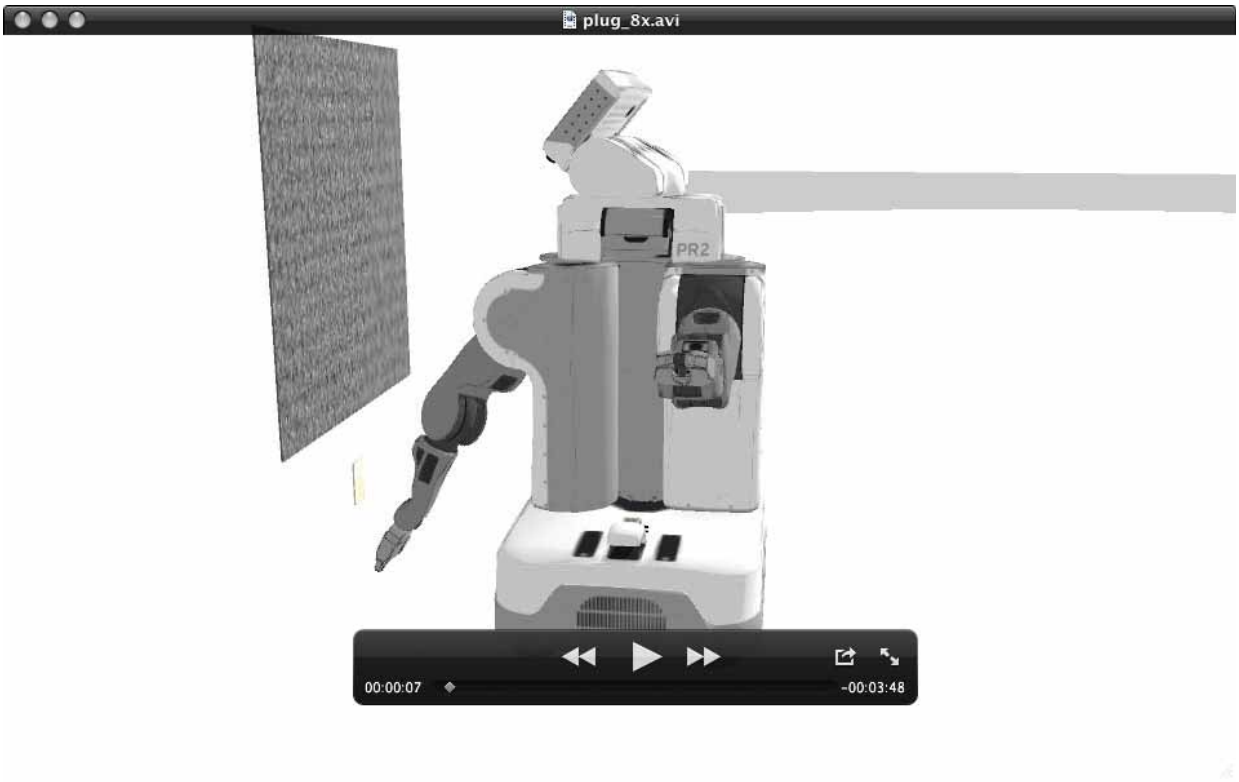
Calibration: documentation



Plugging in: calibrated

Plugging in: putting it all together

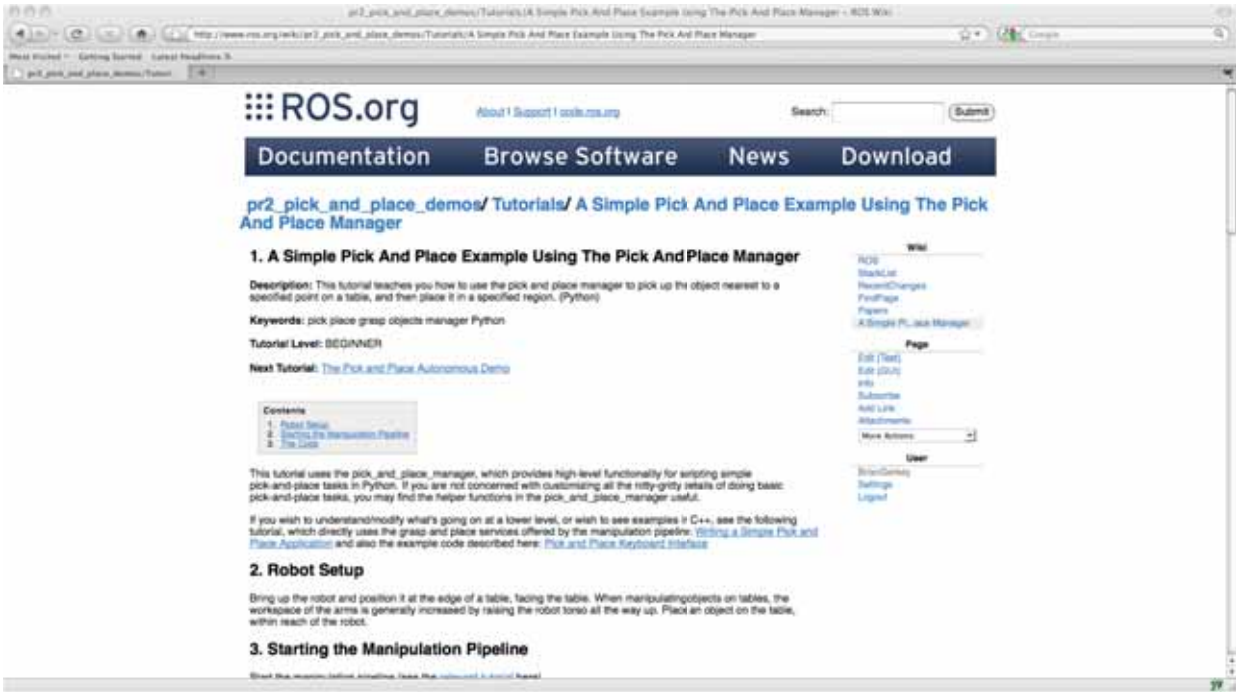
Plugging in: regression test



Tabletop manipulation

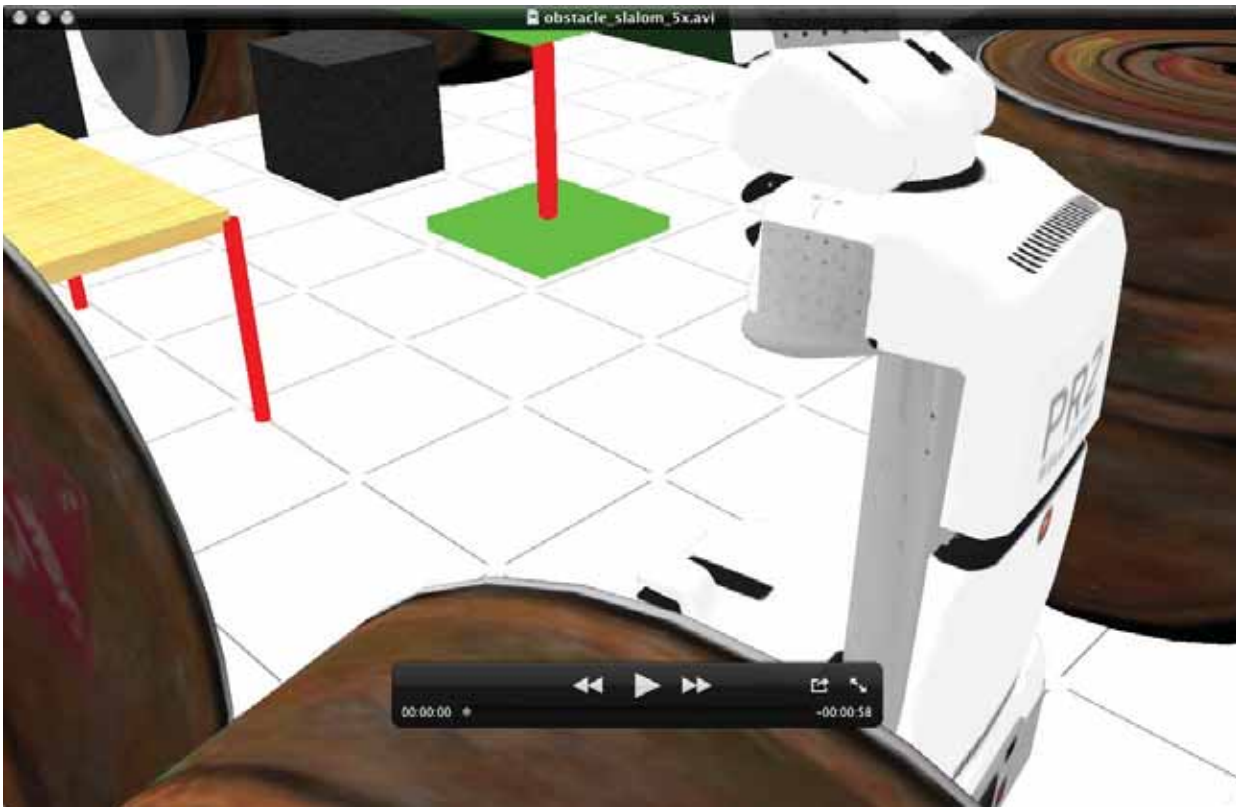
Tabletop manipulation: regression test

Tabletop manipulation: documentation



Navigation

Navigation: regression test



Detour: assembling building blocks

Navigation: experiments

Navigation: how can others compare?

- Okay, your robot navigates successfully, and you've published some results. How can others reproduce what you've done?

Releasing experimental code: why? (*)

- Code used to make experimental claims in papers should be Open
  - key part of experimental design
  - necessary to replicate, refute, or extend results
- *"An article about computational science in a scientific publication is* **not** *the scholarship itself, it is merely* **advertising** *of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures."* – D. Donoho

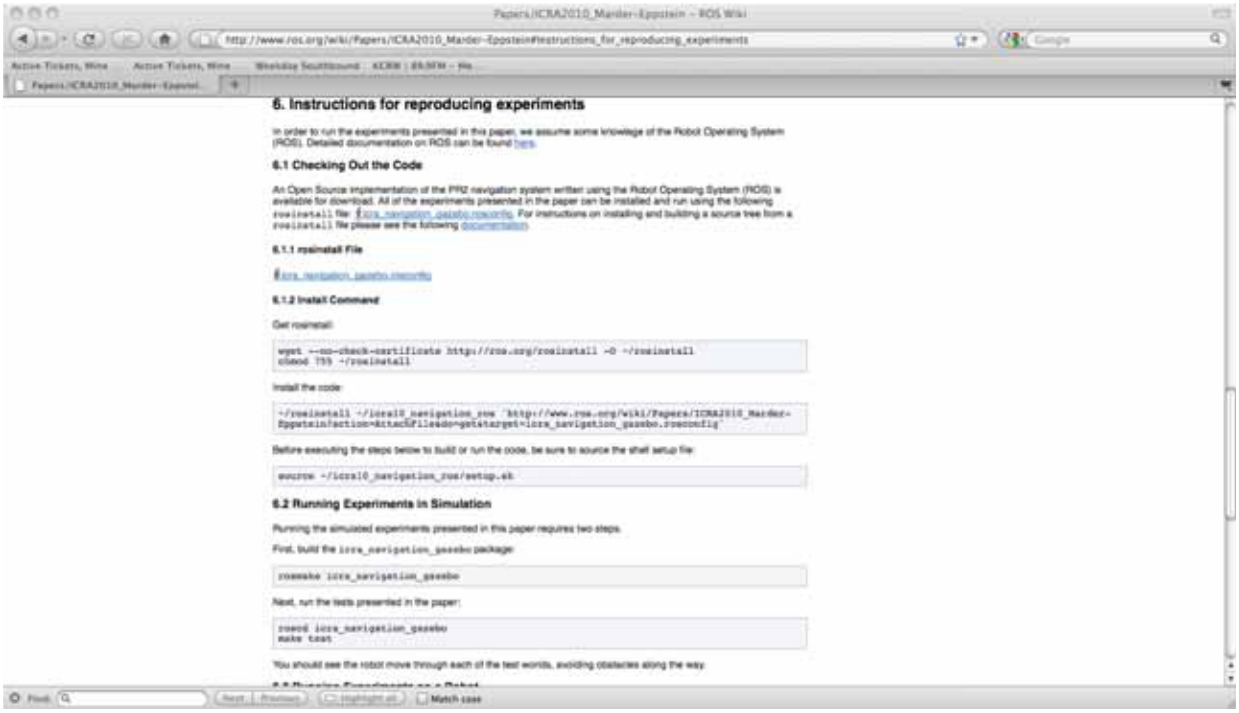
[*] See Buckheit & Dohono, 1995

Releasing experimental code: how? (*)

- Include versioned download details in the paper
  - SVN URL + revision; Git ref + hash
- What about dependencies (libraries, etc.)?
  - A big tarball?
  - A tool like CDE (http://stanford.edu/ pgbovine/cde.html) ?
- Can't share physical state?
  - share configuration info for a well-known simulator
- Goal: publishing a paper without code is like reporting mean without variance

[*] See Wawerla & Vaughan, RSS 2009 workshop on experimental practice

Navigation: reproducible experiments



Navigation: experiments in simulation

Summary

- Reusable building blocks advance the state of the art
- Automated testing is vital
- Simulation is an excellent tool for automated testing
- Sharing experimental code improves scientific practice

Big thanks to many people, but especially **John Hsu** and **Tony Pratkanis**!

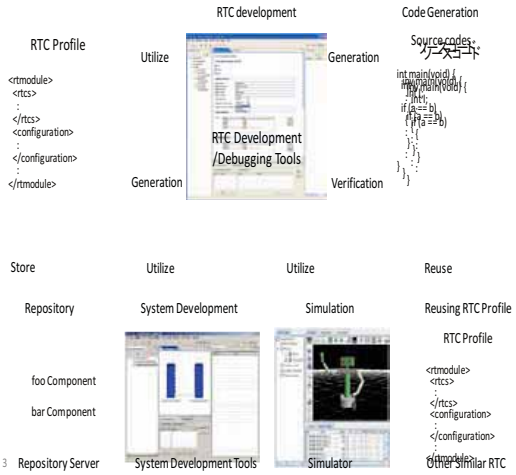
Initial Submission for DDC4RTC RFP

document number of DDC4RTC RFP : mars/10-06-16
document number of this initial submission : mars/10-11-06
presentation: mars/10-12-??

Seung-Woog Jung, Myung-Eun Kim
Infrastructure WG, Robotics DTF
ETRI, KOREA

Terms and Definitions

- RTC 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



Terms and Definitions

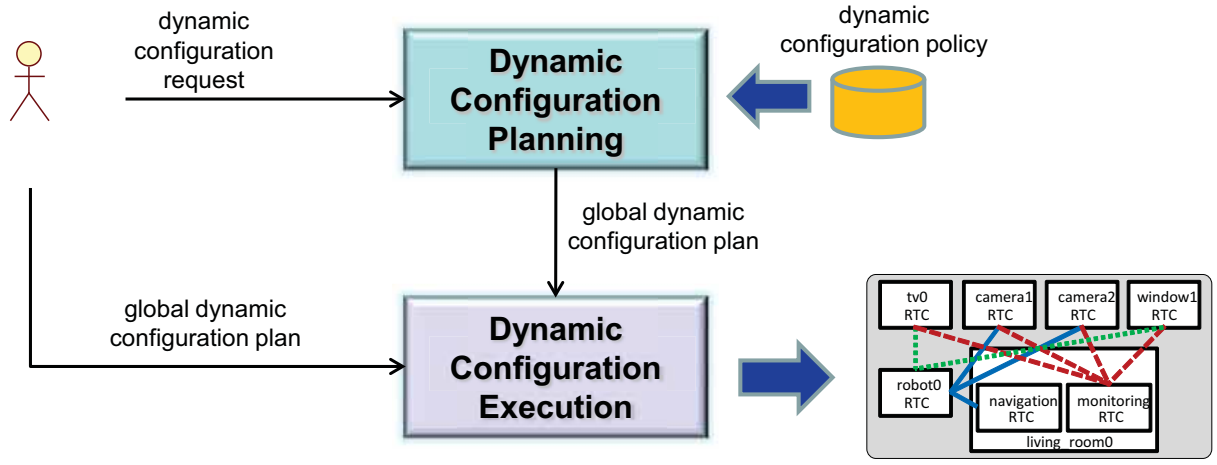
- Configuration
  - A configuration is a process which makes connections between RTCs.
  - An RTC-based system is generally built by composing RTCs or RTC-based subsystems.
  - An RTC-based system or subsystem includes connection information among RTCs.
- Dynamic configuration(or Reconfiguration)
  - A dynamic configuration is a process which changes a configuration(source configuration) to another(target configuration) at run-time.

Terms and Definitions

- Repository
  - The repository manages a set of package information of RTCs and RTC-based systems. The package information includes a profile and a package of RTC/RTC-based system
- Directory
  - A storage that manages the references and the meta-information of running RT component instances

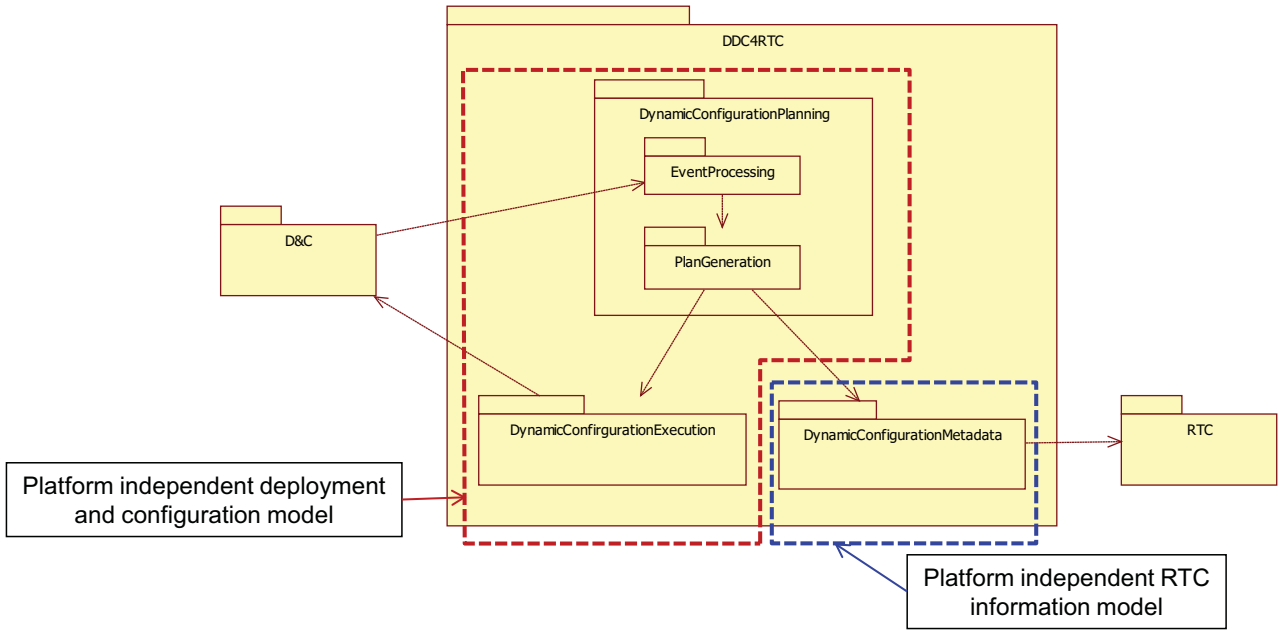
Dynamic Configuration Process

- Dynamic Configuration Planning Process
  - the act of taking a dynamic configuration request and generating a global dynamic configuration plan based on the predefined dynamic configuration policies
- Dynamic Configuration Execution Process
  - a process which takes a global dynamic configuration plan and executes the dynamic configuration based on the global plan



Package Diagram

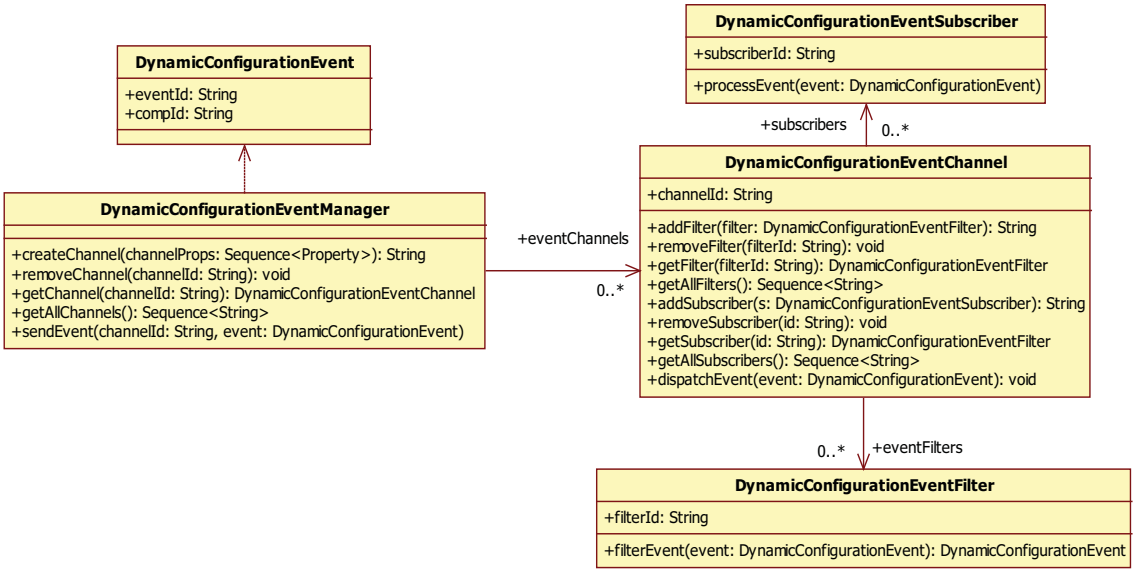
- 3 subpackages : Dynamic Configuration Planning, Dynamic Configuration Execution, and Dynamic Configuration Metadata



Dynamic Configuration Planning Subpackage

Event Processing subpackage

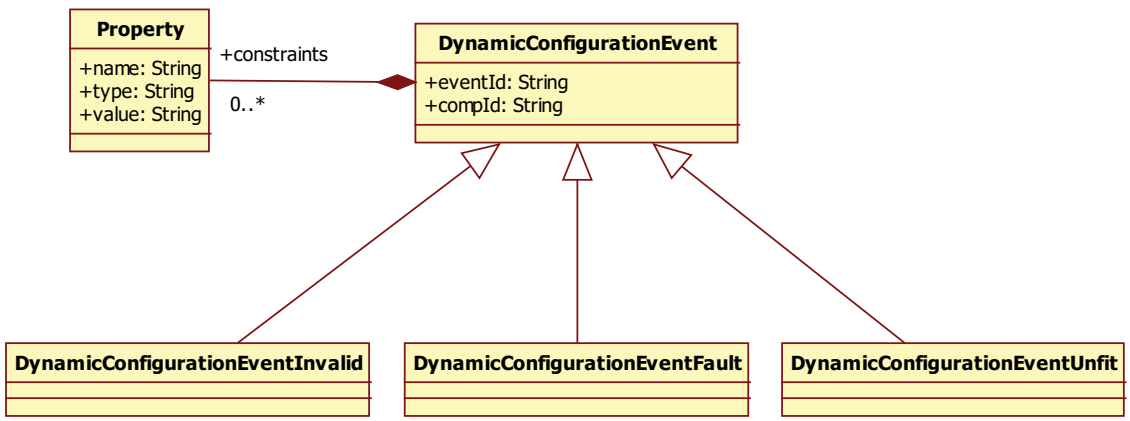
- The event processing model describes how to make dynamic configuration events in order to request or initiate dynamic configuration.



Dynamic Configuration Planning Subpackage

DynamicConfigurationEvent

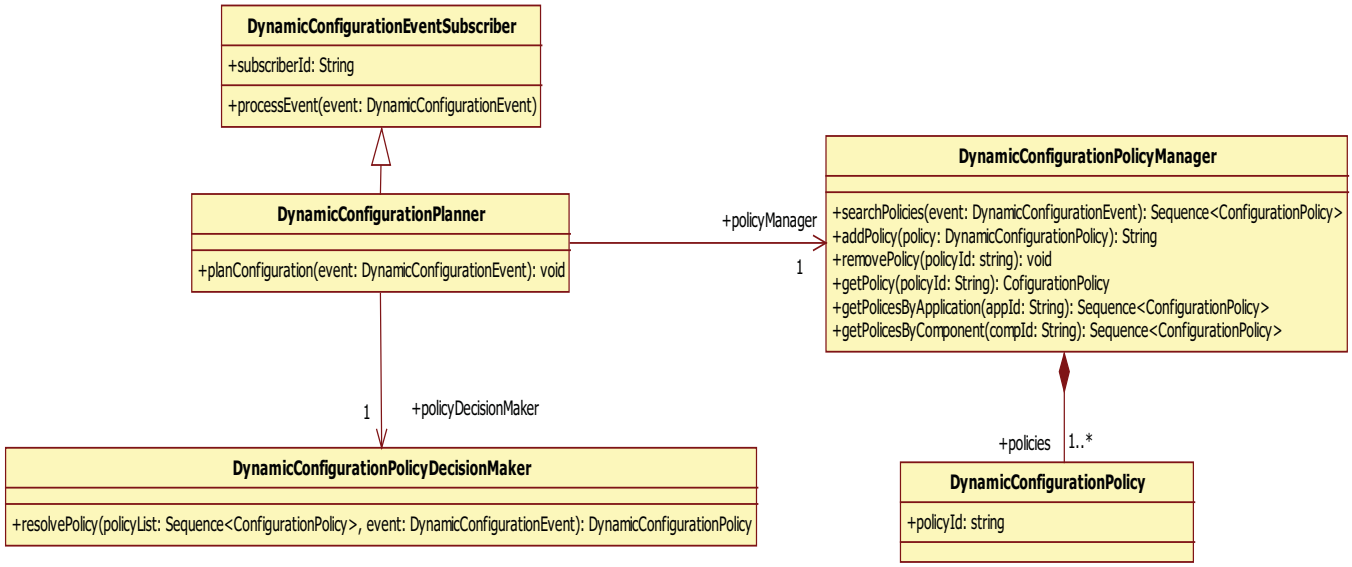
- The DynamicConfigurationEvent class contains information of components which should be configured dynamically and configuration constraints.



Dynamic Configuration Planning Subpackage

Plan Generation subpackage

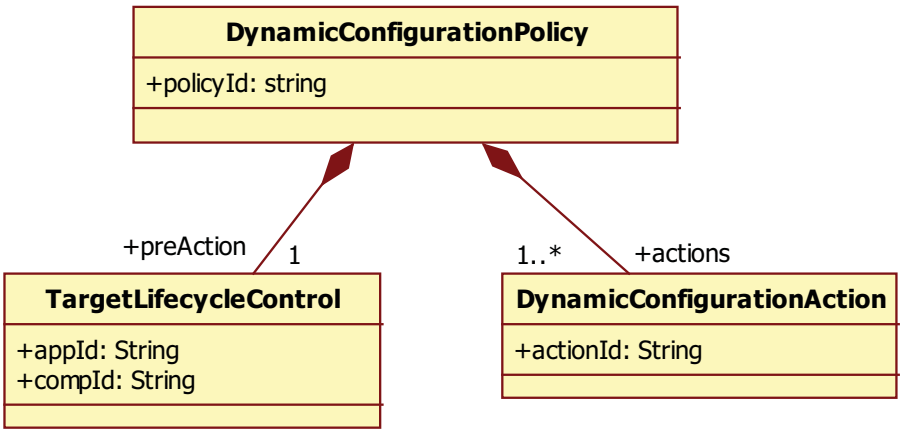
- The plan generation model describes how to generate a global dynamic configuration plan based on dynamic configuration policies in response to a dynamic configuration event.



Dynamic Configuration Planning Subpackage

Dynamic Configuration Policy

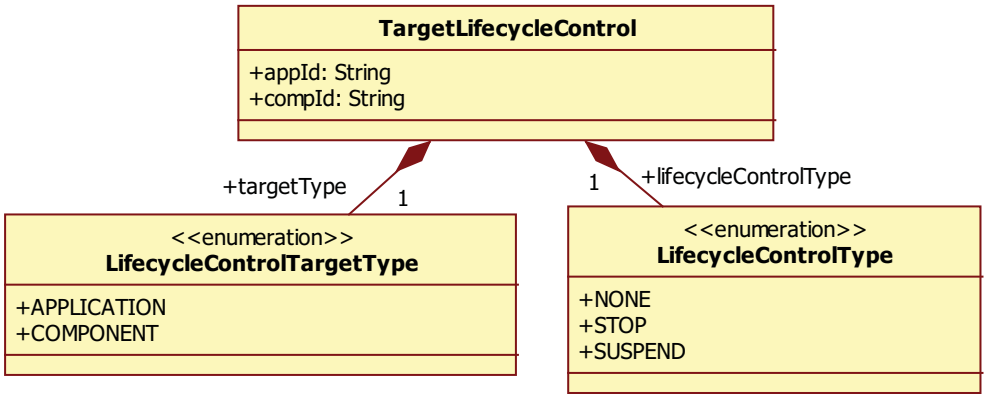
- A DynamicConfigurationPolicy describes the rule of how to do dynamic configuration for a dynamic configuration event.



Dynamic Configuration Planning Subpackage

Target Lifecycle Control

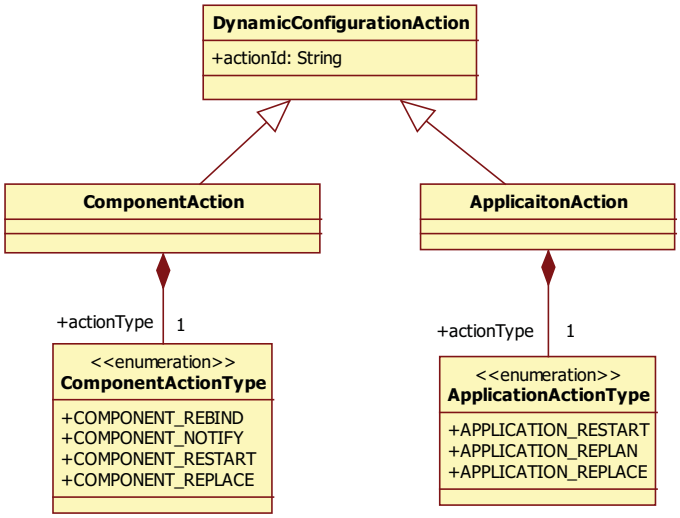
- Before dynamic configuration, the component or the application which is required to be reconfigured can be suspended or stopped.
- The TargetLifecycleControl describes what action should be done to the component or the application before dynamic configuration.
- The target of the lifecycle control can be a component or an application and the lifecycle control can be one of stopping or suspending the component or the application.



Dynamic Configuration Planning Subpackage

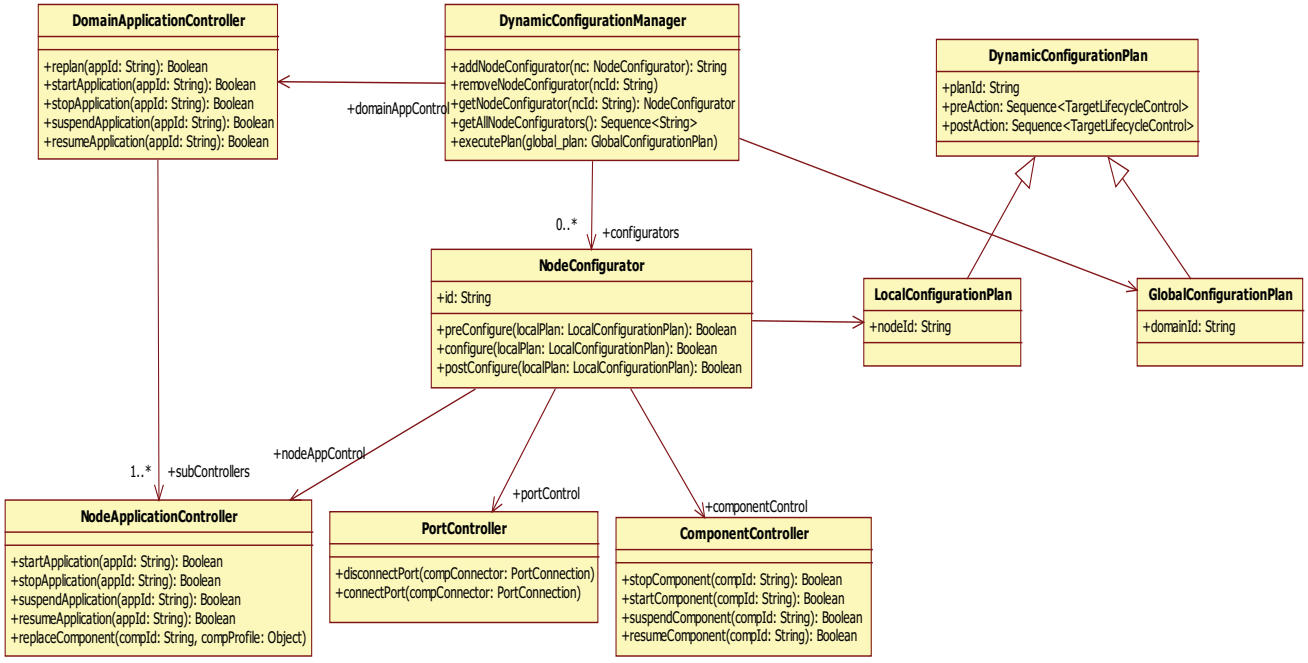
DynamicConfigurationAction

- There are several ways to configure a system dynamically.
- The DynamicConfigurationAction describes how to configure the system. A dynamic configuration action can be applied to a component or an application. Therefore it has two child classes, ComponentAction and ApplicationAction.



Dynamic Configuration Execution Subpackage

- The dynamic configuration execution model describes a process of how to reconfigure a domain application based on a GlobalConfigurationPlan which is generated by DynamicConfigurationPlanner.



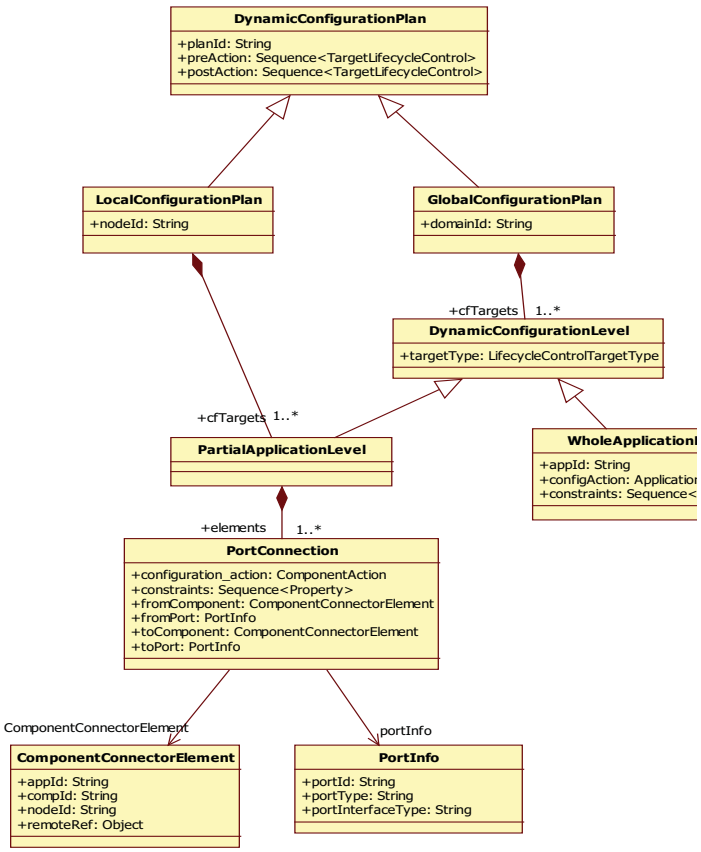
Dynamic Configuration Execution Subpackage

DynamicConfigurationPlan

- **DynamicConfigurationPlan** describes information to be used in reconfigureing the target application.

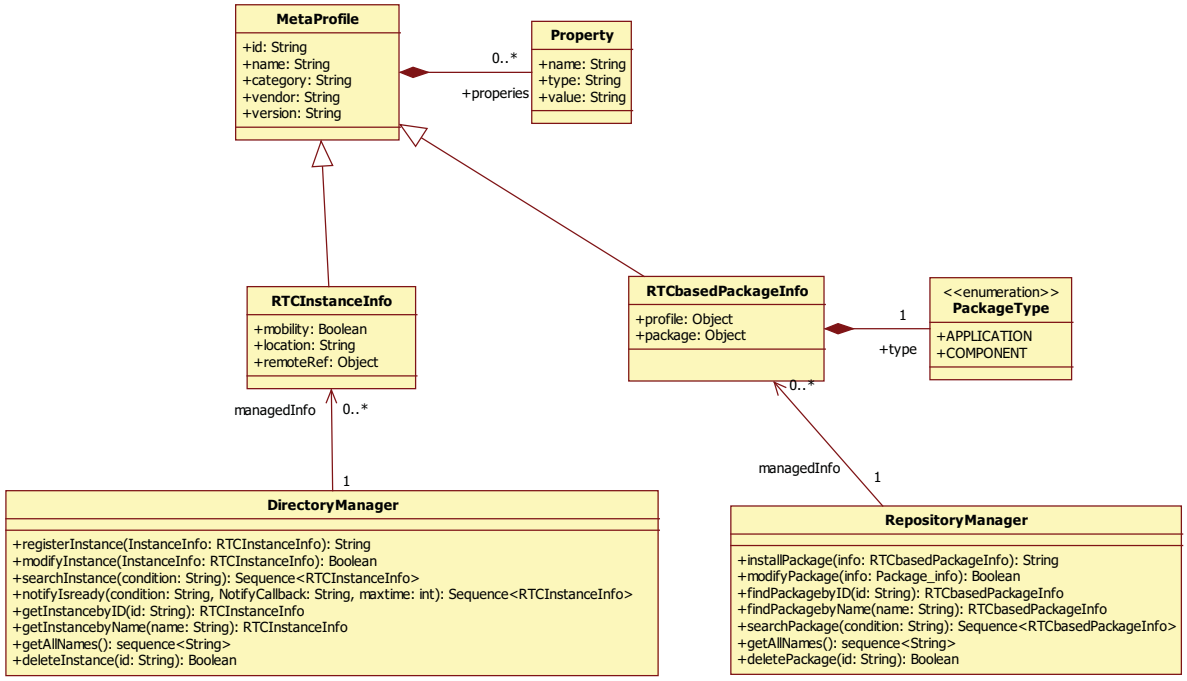
- A **GlobalConfigurationPlan** describes information to be used for the dynamic configuration of a domain application which is executed over several nodes.

- A **LocalConfigurationPlan** describes information to be used for component level dynamic configuration.



Dynamic Configuration Metadata Subpackage

- Dynamic configuration metadata describes information of RTCs, RTC-based systems, and RTC instances to be used for dynamic configuration.



Dynamic Configuration Planning Subpackage

Query

- The **condition** to be used to search RTCbasedPackageInfos or RTCInstanceInfos must abide by the following rules.

<Condition> ::== <Comparison Exp>[Boolean Operator><Condition>]
<Comparison Exp> ::== <Argument 1><Comparison Operator><Argument2>
<Argument1> ::== <String>
<Comparison Operator> ::== { = | !=}
<Argument 2> ::== <String>
<Boolean Operator> ::== { && | ||}

ex) Camera.id = “camera1” && Camera.location = “room1”

[Note]
The <Argument1> must be one of attribute names of an RTCbasedPackageInfo and the <Argument2> must be a value of an attribute of an RTCbasedPackageInfo.

Conclusion

Mandatory Requirements		Our Submission
Platform independent deployment and configuration model	6.5.1 Proposals shall specify **interfaces to services for dynamic configuration and deployment of RTCs**	-DynamicConfigurationPlanner -DynamicConfigurationPolicyManager -DynamicConfigurationPolicyDecision Maker -DynamicConfigurationManager
	6.5.2 Proposal shall specify interfaces **to initiate RTC configuration based on external and/or internal events**. A capability for **event filtering** shall be provided	-DynamicConfigurationEventManager -DynamicConfigurationEventChannel -DynamicConfigurationEventFilter
	6.5.3 Proposals shall reuse or extend at least the PIM, including terms and definitions, of the deployment architecture as defined by the Deployment and Configuration of Component-based Distributed Applications Specification [DEPL]	-DomainApplicationController -NodeApplicationController -ComponentController -PortController

Conclusion

Mandatory Requirements		Our Submission
Platform independent RTC information model	6.5.4 Proposals shall provide a schema, **the RTC Profile**, describing RTC characteristics such as basic RTC information, ports information and so on, based on the RTC specification	-RTCInstanceInfo
	6.5.5 Proposals shall provide a schema, the RTC-based System Profile, describing RTC-based systems characteristics such as port connection information, configuration information, deployment conditions and so on	-RTCbasedPackageInfo
	6.5.6 Proposals shall specify query services to discover and interrogate characteristics of RTCs and RTC-based systems	-RepositoryManager
	6.5.7 Proposal shall specify query services to discover characteristics and location information of deployed RTCs and RTC-based systems	-DirectoryManager

DDC4RTC Initial Submission

Geoffrey Biggs et al
AIST, Japan

Outline

- Two central concepts:
  - Supervision tree
  - State-based application life-cycle management.
- Close fit to D&C concepts
  - Reuse D&C for deployment planning
  - DDC4RTC must supply the “when”

Goals

- Allow robot applications to deployed into available hardware and software resources:
  - On application initialisation
  - At run-time in response to changing conditions
- Be scalable, from the smallest sub-system to the largest collection of cooperating robots.
- Accurately specify the state of an application at any point in time.
  - Allow for deterministic implementations.

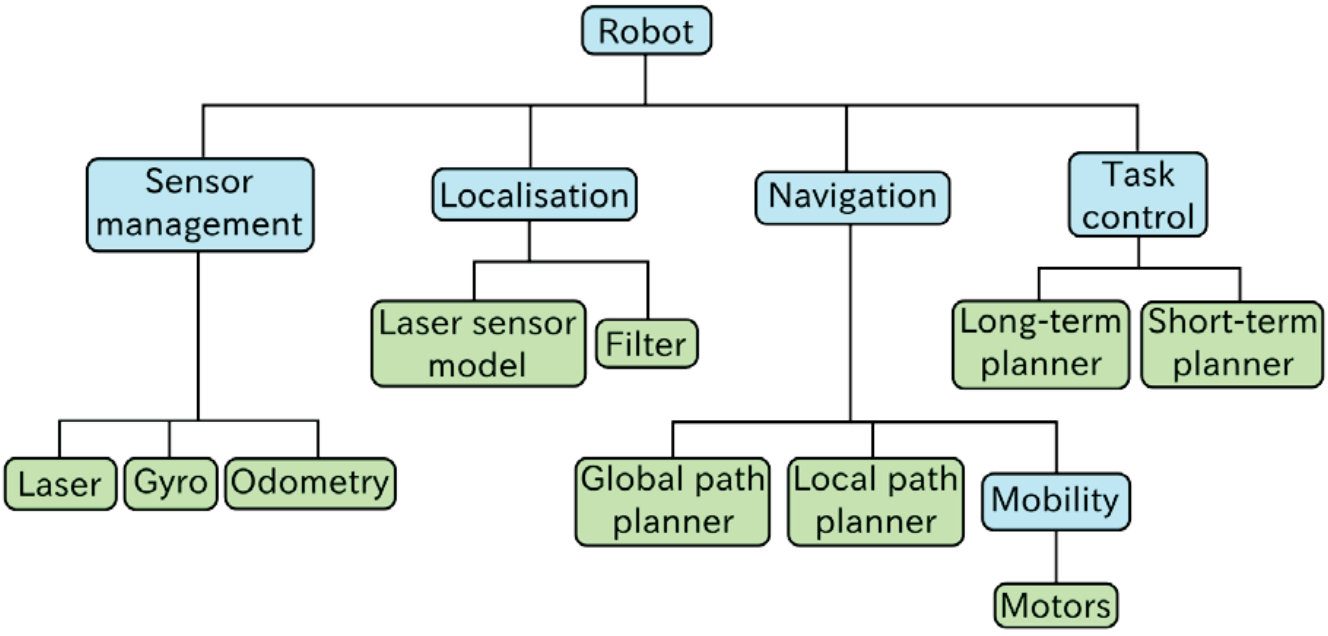
Supervision tree

- A key concept in component deployment, configuration and run-time management.
  - Allows encapsulation of sub-systems
  - Ensures scalability of deployment and configuration infrastructure
- Not a new concept.
  - Used by Ericsson in industry for over 25 years.
  - D&C specification uses the same concept with a different name.

Supervision tree

- Important for allowing propagation of errors up to a level where they can be handled.
  - e.g. A sensor failure in the localisation system should be handled internally.
- If it cannot be handled internally, the entire localisation system is determined to have failed.
  - Handle this at the next level up the tree.

Supervision tree example



Supervisors

- Core part of supervision trees.
- Responsible for managing their application.
  - One supervisor per application.
  - Even if the application is a single monolithic RTC.
- Monitors the condition of all components and child applications.
- Monitors other events relevant to the supervisor.
  - Events tell it when to perform a supervisory task.

AIST

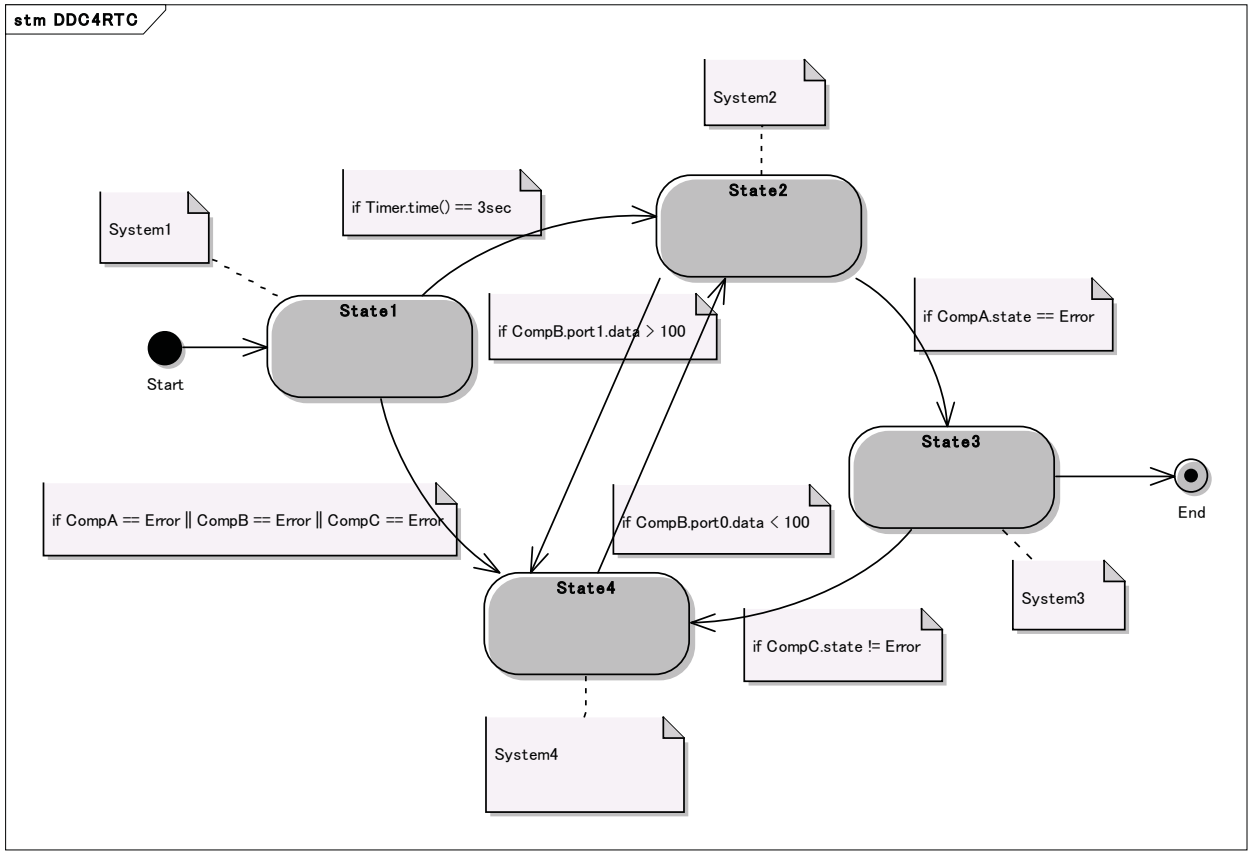
Supervisors

- Must be able to prevent “bouncing.”
  - This is when a supervisor bounces rapidly between two states (e.g. an error condition and repair).
- Prevent bouncing by allowing maximum rates.
  - e.g. if repair of an error condition fails more than 5 times in 10 seconds.
- Bouncing indicates a failure to handle a condition and the supervisor terminates.
  - Handling a failed application is the responsibility of the parent supervisor.

Finite state machine

- Supervisor's internal state is an FSM.
- Start state's deployment plan indicates the initial deployment of the application.
- Re-deployment occurs when the FSM shifts state.
- FSM shifts state in response to events.
- Supervisor (application) terminates when the FSM reaches a terminal state.
  - Termination of a supervisor causes termination of all child components and applications.

Finite state machine example



Management model

- ApplicationSupervisor is the central entity in an RTC Application.
  - Unique within an application.
- Contents is specified in the Dynamic Deployment Data Model.
  - Contained components specified in RTCProfiles.
  - Available deployments specified in RTSProfiles.
- Entry point for deploying an application.

SupervisorFSM

- Part of the Dynamic Deployment Data Model.
- Specifies the FSM of an ApplicationSupervisor.
  - Includes initial and final states used for starting and stopping applications.

Constraints

- The constraints package defines methods and structures for specifying constraints.
- Used to filter events that transition the supervisor FSM.

RTCProfile

- Data model for an RT-Component.
- Contains metadata about the component.
  - Author, vendor, etc.
  - State information
  - Interfaces (ports)
  - Documentation
  - …
- XML PSM.

Compatibility with D&C

- Monolithic RTCs and Composite RTCs equivalent to D&C's monolithic components and assemblies.
- Use D&C data structures for deployment plans.
- Use D&C concepts of domain and node applications.
  - Supervisors are equivalent to ApplicationManagers – with dynamic features added.
  - Supervisors use NodeApplicationManagers to execute their deployment plans.

Supporting live component replacement

- Re-deployment may replace components providing similar functionality.
- Need to preserve state across instances.
- Supervisors allow state to be preserved and passed on to replacement components.
  - Specify "lineages" of components, e.g. the path planner.
  - State transferred between instances of components in a lineage during re-deployment.

Summary

- Supervision tree concept.
- Finite state machine.
- Lots of reuse from D&C.
- RTCProfile and RTSProfile
  - XML specifications

robotics/2010-12-09

Robotic Functional Service WG Meeting Minutes

Chair: S. Chi (ETRI), T. Hori (AIST/JARA) and M. Sato (ATR/JARA)

==========

Monday, December 6, 2010

13:00-17:00 Functional Service WG, San Tomas

Joined organization:

AIST, ETRI, KAR, JARA, ATR, Technologic Arts, SIT, Univ. of Tsukuba, UEC, Toshiba

- Discussion to merge two submissions for RoIS RFP.
- Planning to revise based on JARA's submission.
- Exchanging views through review of JARA's submission.

  [about Structure of the RoIS Framework: Section 7.2]

  - "Common" and "Specific" component are confusing. These terms should be renamed "Basic" and "User-defined".
  - The roles of "main HRI engine" and "sub HRI engine" were discussed.

  [about Return Codes: Section 7.3]

  - Return code enumeration is reasonable.

  [about RoIS Interface: Section 7.4.1 Interaction]

  - Sequences of each interface type are fine.

  [about RoIS Interface: Section 7.4.2 Interfaces]

  - The role of "condition" was discussed.
  - Relationship to existing standards, especially ISO standards, should be studied.
- Scheduled private meeting before next TM (in Arlington, March 2011)
  - Data: end of January or February 2011.
  - Place: in Japan (eg. Fukuoka).

==========

Tuesday, December 7, 2010

09:30-17:00 Functional Service WG, Bayshore W

Joined organization:

AIST, ETRI, KAR, JARA, ATR, Technologic Arts, SIT, Univ. of Tsukuba, UEC, Toshiba

- Discussion about comment from Ingo (Bielefeld University, Germany)
  - Their system seems similar to our submission and interesting, but it is difficult to review their document in a short time.
  - We should ask him to read our submissions and clear up the difference first. (Dr. Hori's homework)
- Exchanging views through review of JARA's submission.

[about RoIS Interface: Section 7.4.3 Message Data]

- The role of "condition" should be discussed in more detail. (We need to study "QueryExpression[ISO19143]")

[about RoIS Interface: Section 7.5 Profiles]

- Types of profiles are reasonable.
- It seems difficult to have a fixed "expires" of event message because it may vary according to the situation and the status of HRI engine, such as occurrence frequency of events and an amount of available memories.
- So "expires" should not be included in the event message profile as fixed values.
- "receive_event()" should have an "expires" parameter to notify the time limit for each event occurrence.
- On the other hand, "expires" of command message can be fixed in the profile because it can be designed without caring the situation or the status of HRI engine. (When HRI engine is busy, it should be interrupted.)
- In the case of command expiration, "completed()" is called with "TIMEOUT" status.
- “expires” for event should be renamed to “expire”.
- “expires” for command should be renamed to “timeout”.
- The need to set "priority" for the event message was discussed.
- "priority" should be mentioned, but its specification should not be included in the (revised) proposal because "priority" varies on the types of robotic systems.

[about Common Messages: Section 7.6]

- Types of common HRI components were discussed. (Components name

was changed to “Basic HRI Components.” See above.)

- "reaction" should be renamed more appropriately if we can find the name.
- "navigation" should be renamed more appropriately if we can find the name.
- "object tracking" should be added to the basic component list.
- The meaning of "MOTIONDETECTED" event should be cleared to judge the need of "motion detection". (Dr. Chi's homework)
- The meaning of "USER GENDER CLASSIFIED" event should be cleared to judge the need of "user gender classification (or user identification)". It may be included in "person identification". (Dr. Chi's homework)
- The meaning of ""USER AGE CLASSIFIED"" event should be cleared to judge the need of "user age classification (or user identification)". It may be included in "person identification". (Dr. Chi's homework)

[about each Common Messages: Section 7.6.1-7.6.10]

- Results of "Person identification" may not include "position" and "pose" of the person because these results can be obtained by "person detection".
- "ID" in the each result is confusing. For example, ID in "person detection" may be misunderstood as "person ID". Each ID should be renamed such as "temporal ID" or “serial number”, ”index number” or something.
- We need more discussion whether “detection” and “identification” must be separated, and also the meaning of these functions.
- Results of "Person detection" should not include "behavior" because recognizing person behavior should be separated from functionality of detection.
- Results of "Sound localization" should include both "position" and "direction".
- Enumeration of "Reaction Types" should be discussed in more detail.
- "OOS" in "Reaction Types" should be renamed appropriately.
- Definition method for a predefined codebook in "gesture recognition" should be discussed in more detail.
  (We need to study "RLS" and "BioAPI[ISO19784-1]")
- "speech recognition" and "speech" need to be reviewed by the expert in speech recognition and speech synthesis at ETRI. (Dr. Chi's homework)

Initial Submission for DDC4RTC RFP

document number of DDC4RTC RFP : mars/10-06-16
document number of this initial submission : mars/10-11-06
presentation: mars/2010-12-24

Seung-Woog Jung, Myung-Eun Kim
Infrastructure WG, Robotics DTF
ETRI, KOREA

Terms and Definitions

- RTC Profile
  - A description that represents the static state of an RT component.
- RTC-based system profile
  - An RTC-based system is generally built by composing RTCs or RTC-based subsystems.
  - An RTC-based system or subsystem includes connection information among RTCs.
  - An RTC-based system profile is a description of how RT-components are connected and interact with each other, and RT-component configuration parameters

Terms and Definitions

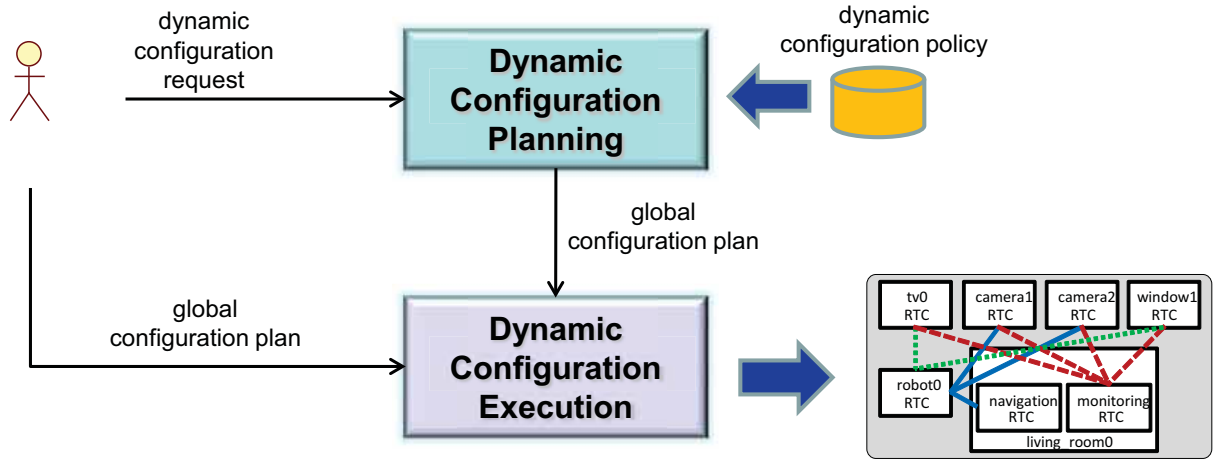
- Configuration
  - A configuration is a process which makes connections between RTCs.
- Dynamic configuration(or Reconfiguration)
  - A dynamic configuration is a process which changes a configuration(source configuration) to another(target configuration) at run-time.

Terms and Definitions

- Repository
  - Repository manages a set of package information of RTCs and RTC-based systems. The package information includes a profile and a package of RTC/RTC-based system
- Directory
  - A storage that manages the references and the meta-information of running RT component instances

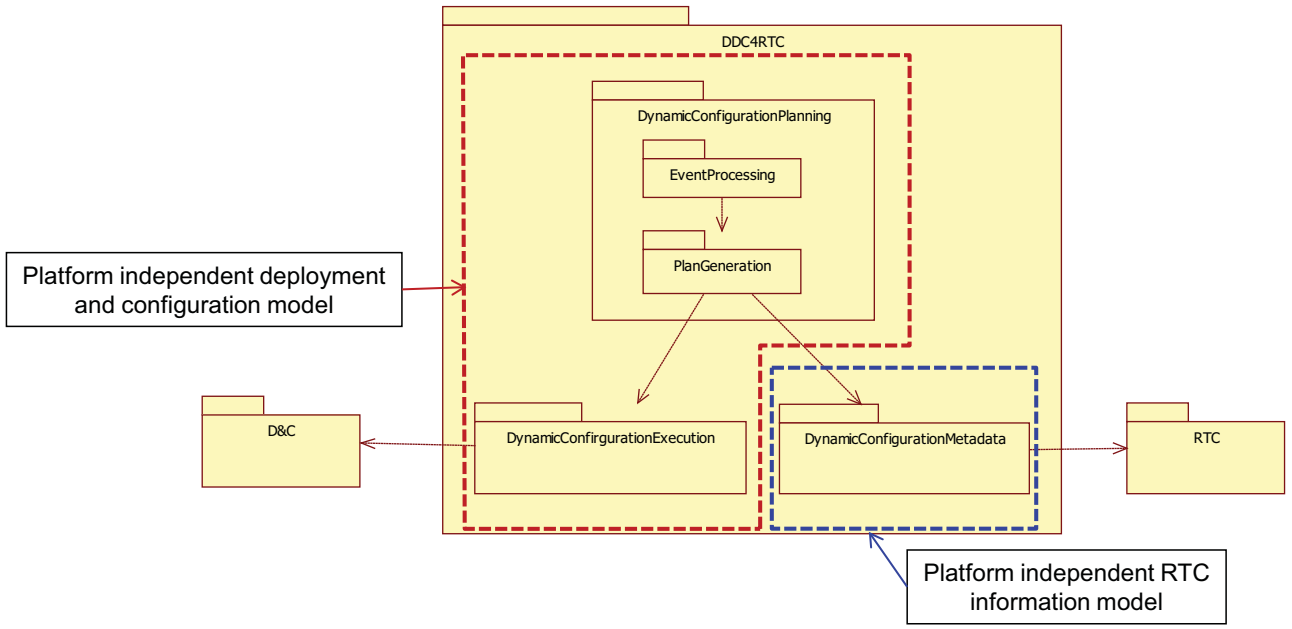
Dynamic Configuration Process

- Dynamic Configuration Planning Process
  - the act of taking a dynamic configuration request and generating a global configuration plan based on the predefined dynamic configuration policies
- Dynamic Configuration Execution Process
  - a process which takes a global configuration plan and executes the dynamic configuration based on the global plan



Package Diagram

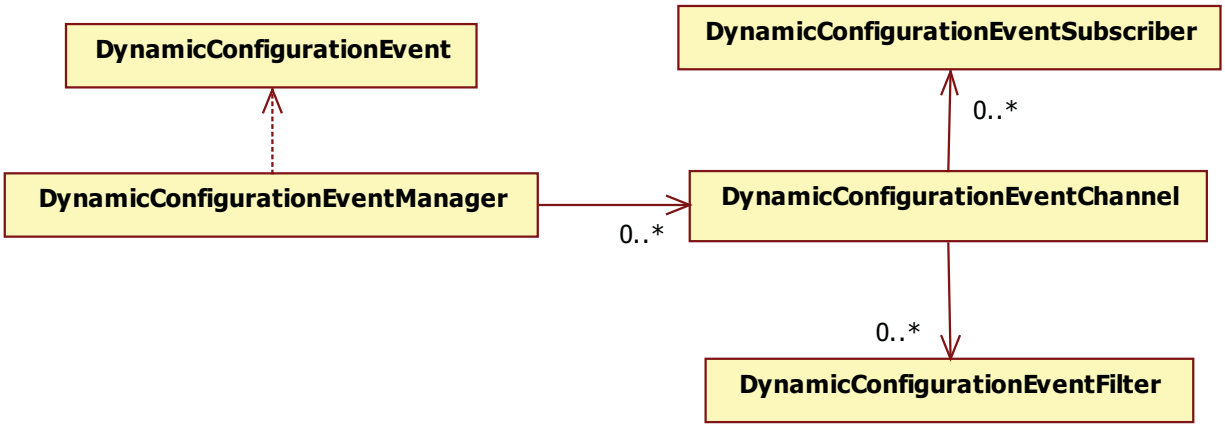
- 3 subpackages : Dynamic Configuration Planning, Dynamic Configuration Execution, and Dynamic Configuration Metadata



Dynamic Configuration Planning Subpackage

Event Processing subpackage

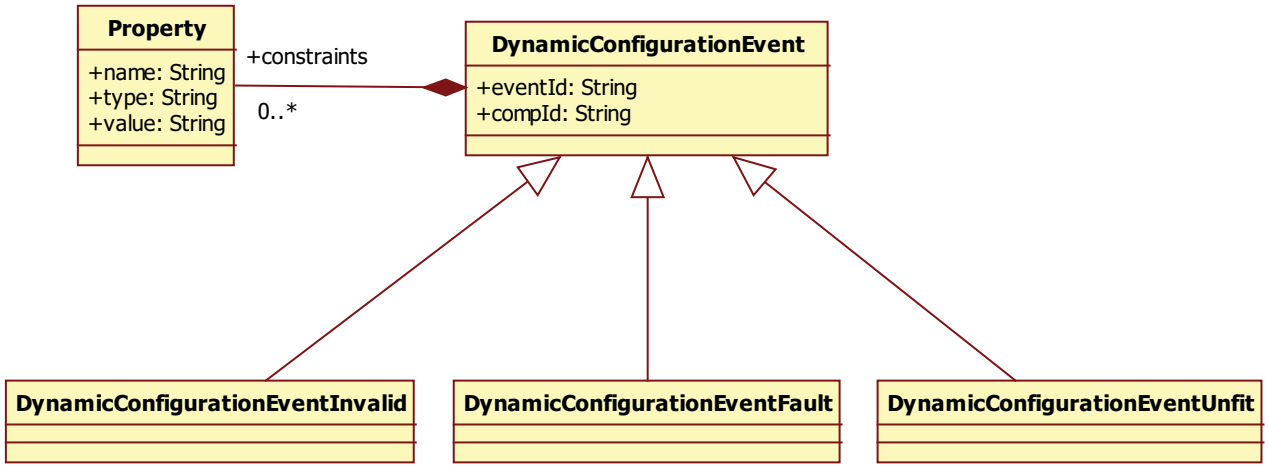
- The event processing model describes how to make dynamic configuration events in order to request or initiate dynamic configuration.



Dynamic Configuration Planning Subpackage

DynamicConfigurationEvent

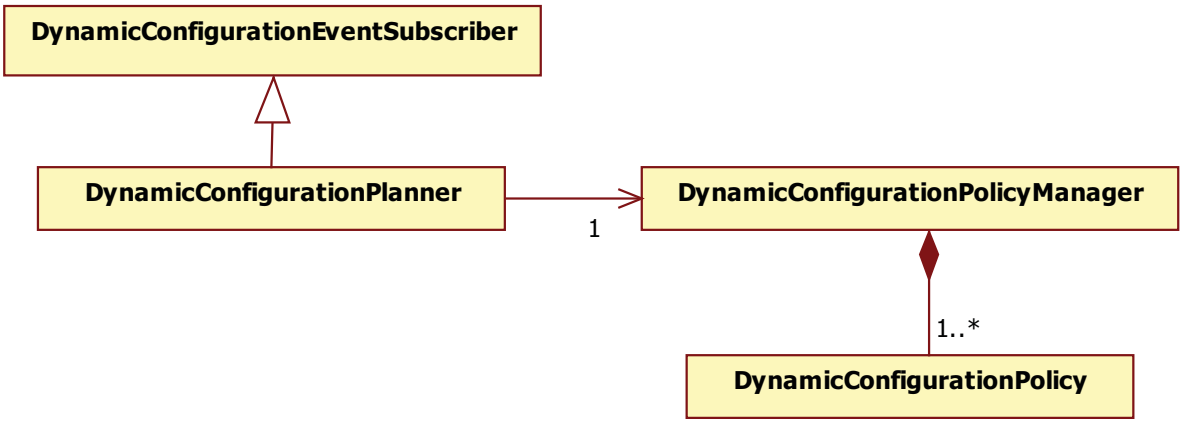
- The DynamicConfigurationEvent class contains information of a component which should be configured dynamically and configuration constraints.



Dynamic Configuration Planning Subpackage

Plan Generation subpackage

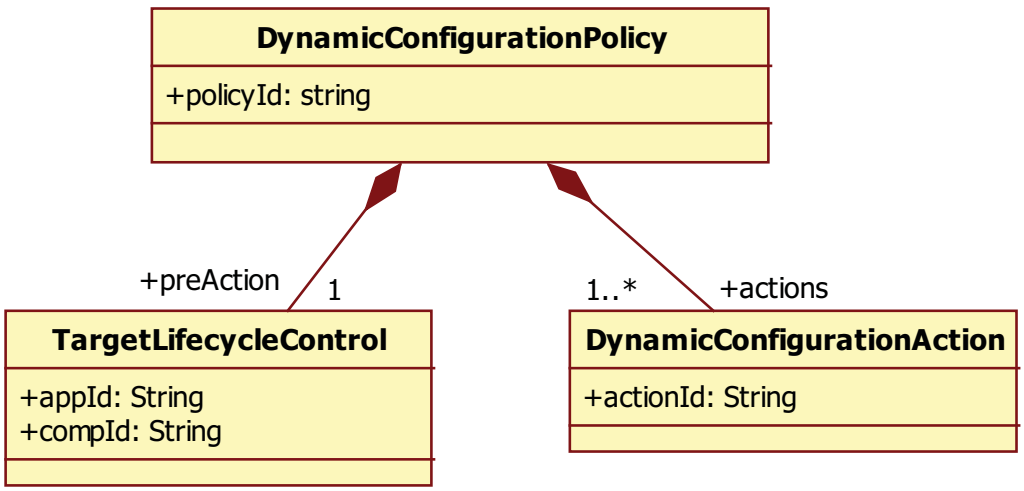
- The plan generation model describes how to generate a global configuration plan based on dynamic configuration policies in response to a dynamic configuration event.



Dynamic Configuration Planning Subpackage

Dynamic Configuration Policy

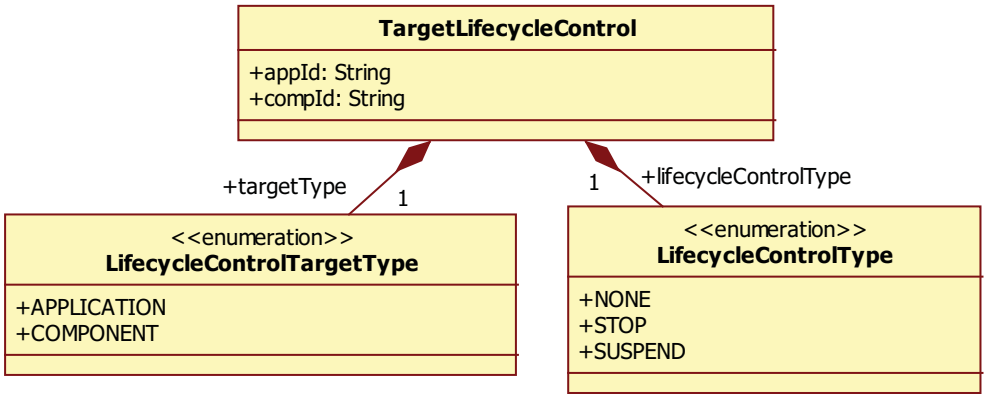
- A DynamicConfigurationPolicy describes the rule of how to do dynamic configuration.



Dynamic Configuration Planning Subpackage

Target Lifecycle Control

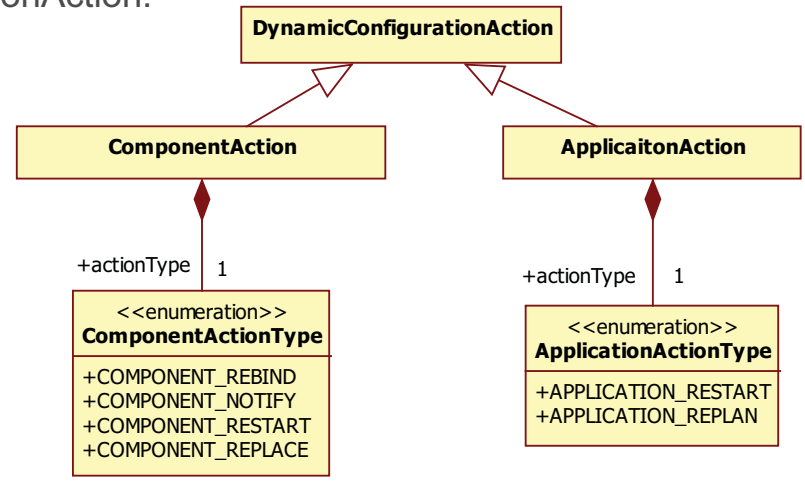
- Before dynamic configuration, the component or the application which is required to be reconfigured can be suspended or stopped.
- The TargetLifecycleControl describes what action should be done to the component or the application before dynamic configuration.
- The target of the lifecycle control can be a component or an application
- The lifecycle control can be one of stopping or suspending the component or the application.



Dynamic Configuration Planning Subpackage

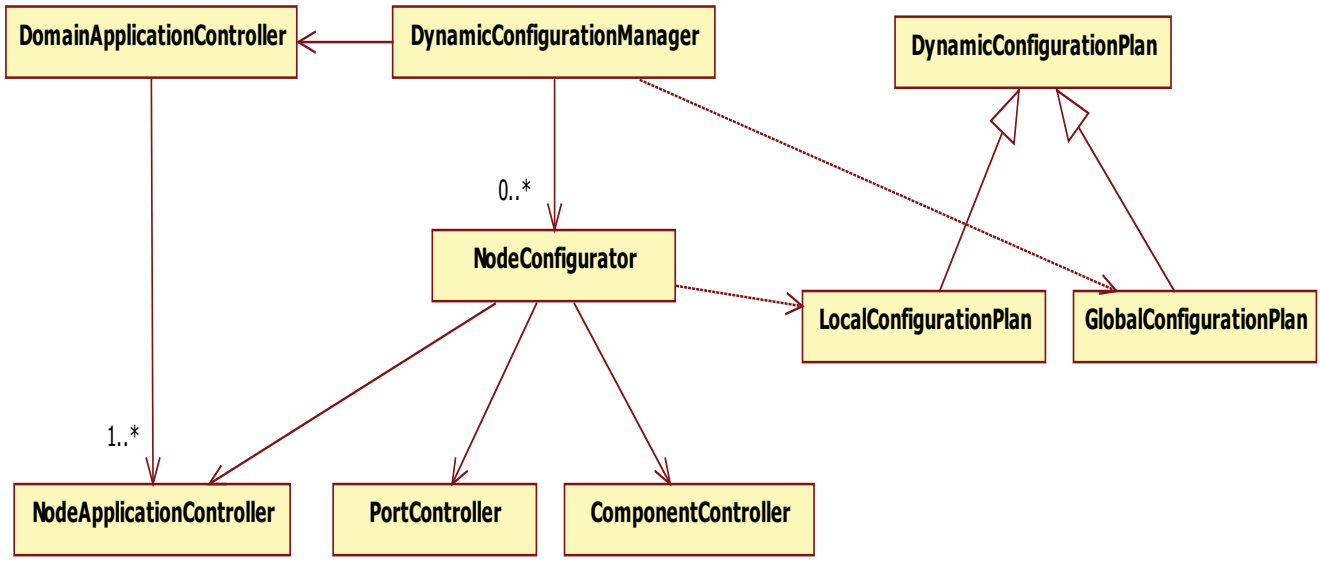
DynamicConfigurationAction

- There are several ways to configure a system dynamically.
- The DynamicConfigurationAction describes information of how to configure the system.
- A dynamic configuration action can be applied to a component or an application. Therefore it has two child classes, ComponentAction and ApplicationAction.



Dynamic Configuration Execution Subpackage

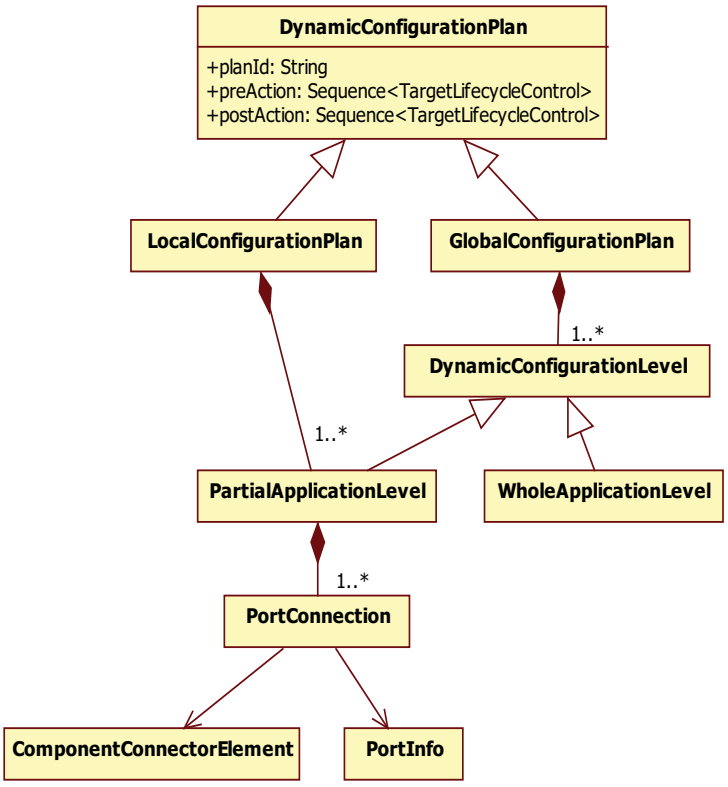
- The dynamic configuration execution model describes a process of how to reconfigure a domain application based on a GlobalConfigurationPlan which is generated by DynamicConfigurationPlanner.



Dynamic Configuration Execution Subpackage

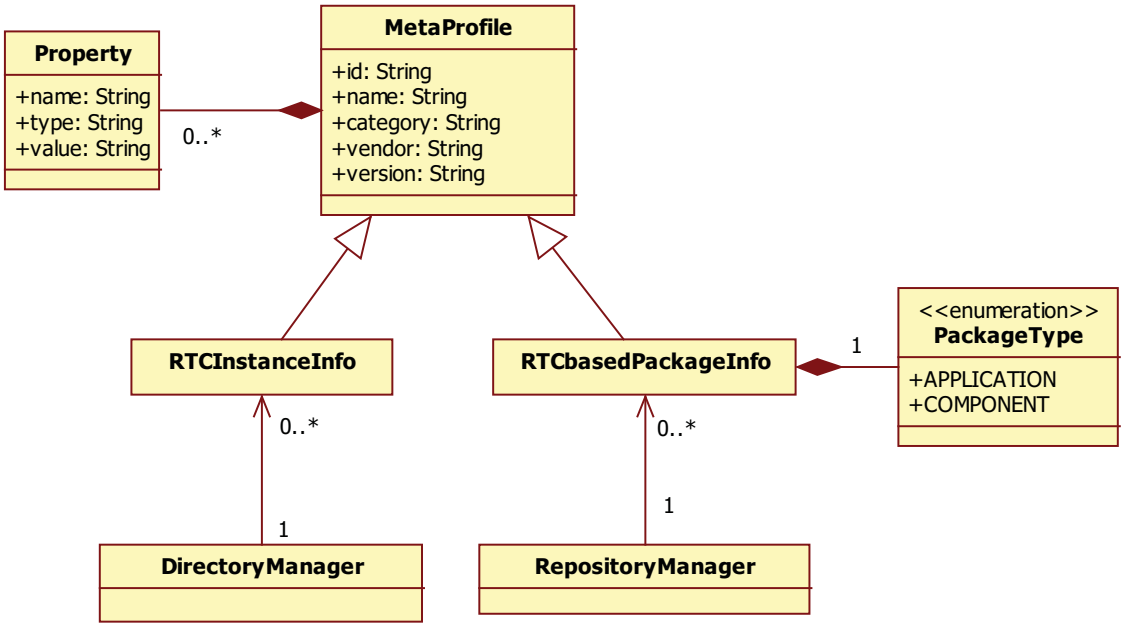
DynamicConfigurationPlan

- **DynamicConfigurationPlan** describes information to be used in reconfiguring the target application.
- A **GlobalConfigurationPlan** describes information to be used for the domain level dynamic configuration which is executed over several nodes.
- A **LocalConfigurationPlan** describes information to be used for component level dynamic configuration.



Dynamic Configuration Metadata Subpackage

- Dynamic configuration metadata describes information of RTCs, RTC-based systems, and RTC instances to be used for dynamic configuration.



Dynamic Configuration Planning Subpackage

Query

- The **condition** to be used to search RTCbasedPackageInfos or RTCInstanceInfos must abide by the following rules.

<Condition> ::== <Comparison Exp>[Boolean Operator><Condition>]
<Comparison Exp> ::== <Argument 1><Comparison Operator><Argument2>
<Argument1> ::== <String>
<Comparison Operator> ::== { = | !=}
<Argument 2> ::== <String>
<Boolean Operator> ::== { && | ||}

ex) Camera.type = “color-camera” && Camera.location = “room1”

[Note]
The <Argument1> must be one of attribute names of an RTCbasedPackageInfo and the <Argument2> must be a value of an attribute of an RTCbasedPackageInfo.

Conclusion

Mandatory Requirements		Our Submission
Platform independent deployment and configuration model	6.5.1 Proposals shall specify **interfaces to services for dynamic configuration and deployment of RTCs**	-DynamicConfigurationPlanner -DynamicConfigurationPolicyManager -DynamicConfigurationPolicyDecision Maker -DynamicConfigurationManager
	6.5.2 Proposal shall specify interfaces **to initiate RTC configuration based on external and/or internal events**. A capability for **event filtering** shall be provided	-DynamicConfigurationEventManager -DynamicConfigurationEventChannel -DynamicConfigurationEventFilter
	6.5.3 Proposals shall reuse or extend at least the PIM, including terms and definitions, of the deployment architecture as defined by the Deployment and Configuration of Component-based Distributed Applications Specification [DEPL]	-DomainApplicationController -NodeApplicationController

Conclusion

Mandatory Requirements		Our Submission
Platform independent RTC information model	6.5.4 Proposals shall provide a schema, **the RTC Profile**, describing RTC characteristics such as basic RTC information, ports information and so on, based on the RTC specification	-RTCInstanceInfo
	6.5.5 Proposals shall provide a schema, the RTC-based System Profile, describing RTC-based systems characteristics such as port connection information, configuration information, deployment conditions and so on	-RTCbasedPackageInfo
	6.5.6 Proposals shall specify query services to discover and interrogate characteristics of RTCs and RTC-based systems	-RepositoryManager
	6.5.7 Proposal shall specify query services to discover characteristics and location information of deployed RTCs and RTC-based systems	-DirectoryManager

DDC4RTC Initial Submission

MARS/2010-12-25

Geoffrey Biggs *et al*
AIST, Japan

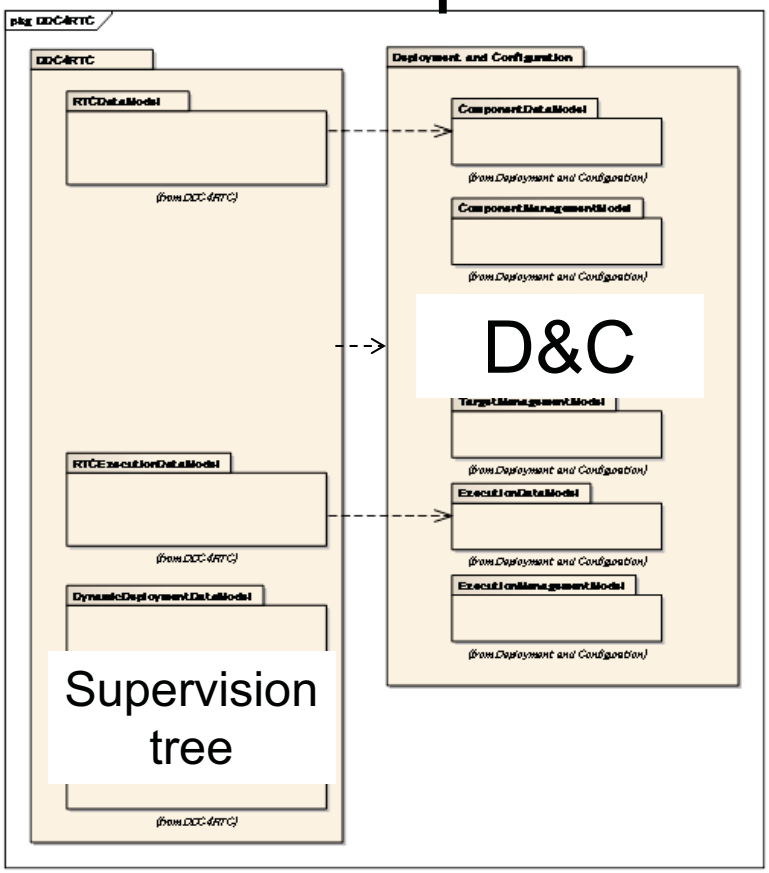
Outline

- Two central concepts:
  - State-based assembly life-cycle management.
  - Supervision tree.
- Close fit to D&C concepts
  - Reuse D&C for deployment planning
  - DDC4RTC must supply the “when”

Goals

- Allow component assemblies to be deployed into available resources:
  - On application initialisation
  - At run-time in response to changing conditions
- Be scalable, from the smallest sub-system to the largest collection of cooperating robots.
- Accurately specify the state of an assembly at any point in time.
  - Allow for deterministic implementations.

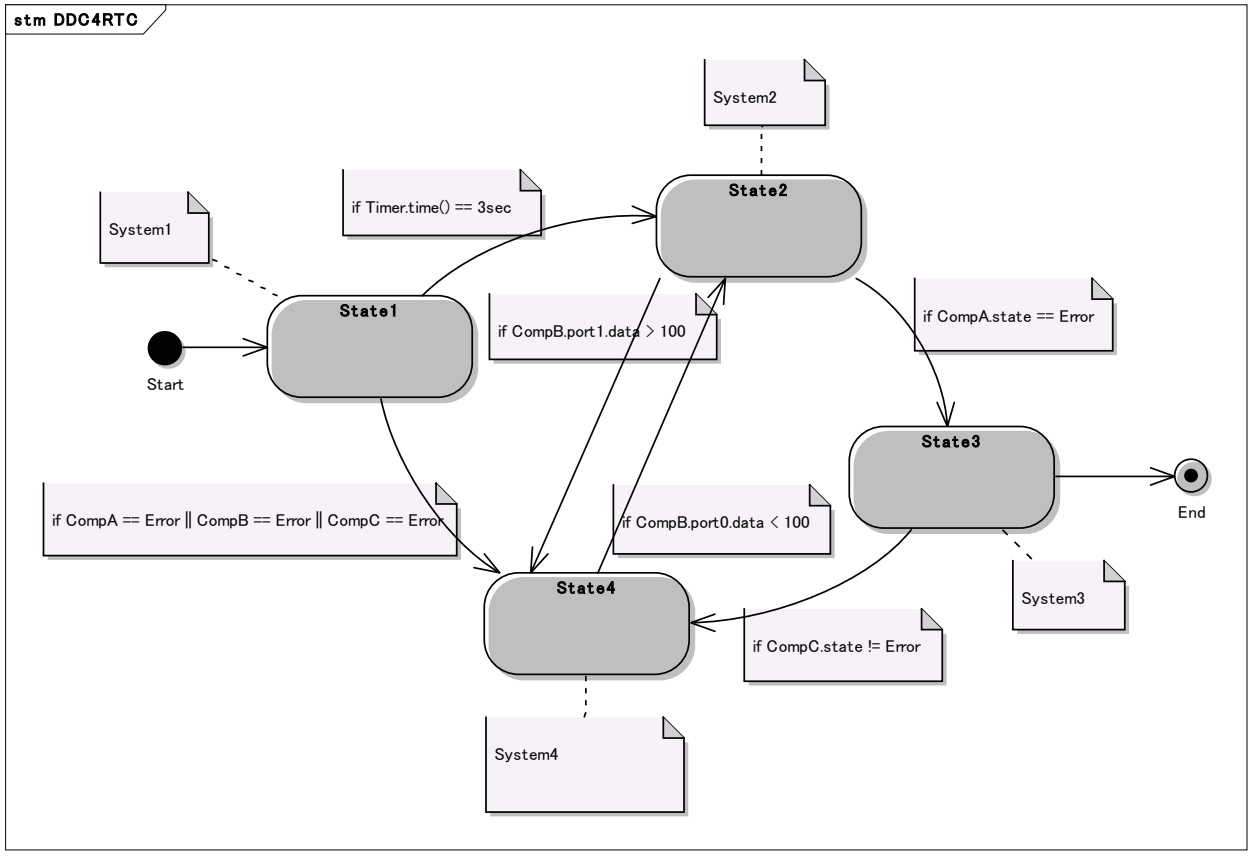
Relationship to D&C



Finite state machine

- Supervisor's internal state is an FSM.
- Start state's deployment plan indicates the initial deployment of the assembly.
- Re-deployment occurs when the FSM shifts state.
- FSM shifts state in response to events.
- Supervisor (assembly) terminates when the FSM reaches a terminal state.
  - Termination of a supervisor causes termination of all child components and assemblies.

Finite state machine example



Supervision tree

- A key concept in component deployment, configuration and run-time management.
  - Allows encapsulation of sub-systems
  - Ensures scalability of deployment and configuration infrastructure
- Not a new concept.
  - Used by Ericsson in industry for over 25 years.
  - D&C specification uses a very similar with a different name.

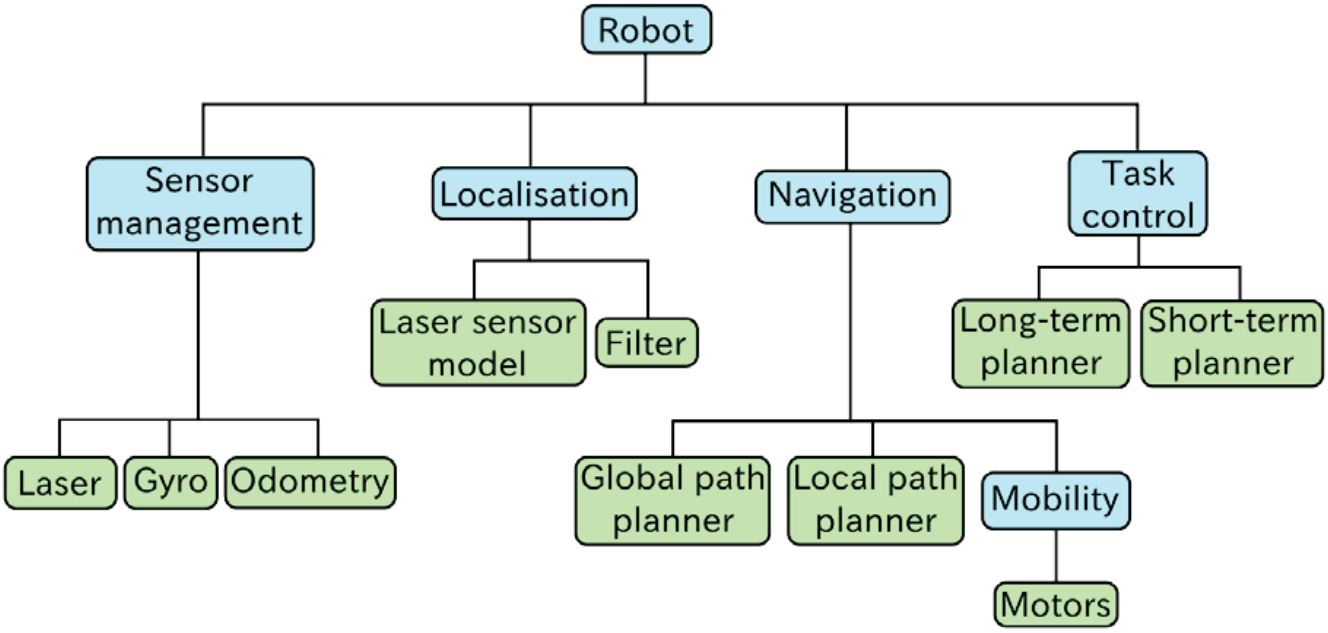
AIST

Supervision tree

- Important for allowing propagation of errors up to a level where they can be handled.
  - e.g. A sensor failure in the localisation system should be handled internally.
- If it cannot be handled internally, the entire localisation system is determined to have failed.
  - Handle this at the next level up the tree.

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

Supervision tree example



Supervisors

- Core part of supervision trees.
- Responsible for managing their assemblies.
  - One supervisor per assembly.
  - Even if the application is a single monolithic RTC.
- Monitors the condition of all components and child assemblies.
- Monitors other events relevant to the supervisor.
  - Events tell it when to perform a supervisory task.

Supervisors

- Must be able to prevent “bouncing.”
  - This is when a supervisor bounces rapidly between two states (e.g. an error condition and repair).
- Prevent bouncing by allowing maximum rates.
  - e.g. if repair of an error condition fails more than 5 times in 10 seconds.
- Bouncing indicates a failure to handle a condition and the supervisor terminates.
  - Handling a failed application is the responsibility of the parent supervisor.

Management model

- ApplicationSupervisor is the central entity in an RTC assembly.
  - Unique within an assembly.
- Contents is specified in the Dynamic Deployment Data Model.
  - Contained components specified in RTCProfiles.
  - Available deployments specified in RTSProfiles.
- Entry point for deploying an assembly.

SupervisorFSM

- Part of the Dynamic Deployment Data Model.
- Specifies the FSM of an ApplicationSupervisor.
  - Includes initial and final states used for starting and stopping applications.

Constraints

- The constraints package defines methods and structures for specifying constraints.
- Used to filter events that transition the supervisor FSM.

RTCProfile

- Data model for an RT-Component.
- Contains metadata about the component.
  - Author, vendor, etc.
  - State information
  - Interfaces (ports)
  - Documentation
  - …
- XML PSM.

Compatibility with D&C

- Monolithic RTCs and Composite RTCs equivalent to D&C's monolithic components and assemblies.
- Use D&C data structures for deployment plans.
- Use D&C concepts of domain and node applications.
  - Supervisors are equivalent to ApplicationManagers – with dynamic features added.
  - Supervisors use NodeApplicationManagers to execute their deployment plans.

Summary

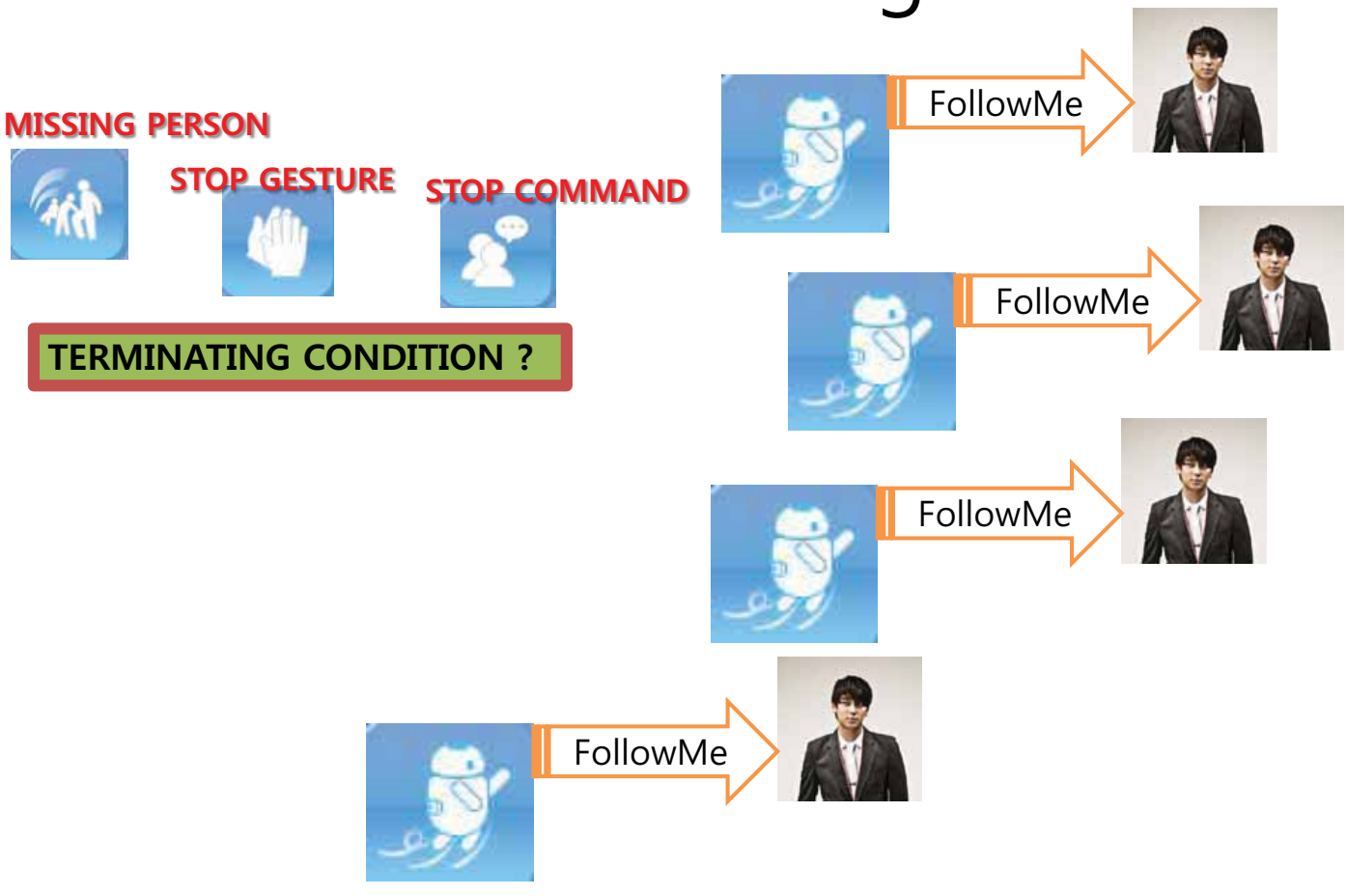
- Supervision tree concept.
- Finite state machine.
- Lots of reuse from D&C.
- RTCProfile and RTSProfile
  - XML specifications

RoIS Framework Tracking Issue

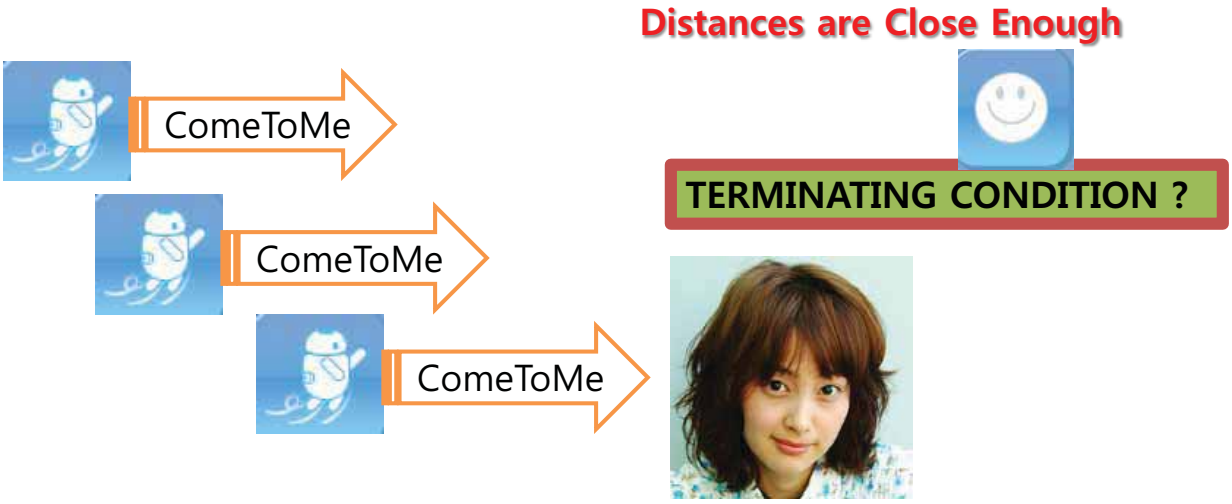
2010-12-08

Ph.D. Suyoung Chi

Human Tracking API



Human Tracking API



Human Tracking API

- Event

Event Name	Argument	Remarks
HD_TRACKING	IDInfo iInfo*	
HD_TRACKFINISHED	FinishState	Finished Tracking

```
Enum FinishState {
FinishByLosingUserTrack,
FinishByUserCommand,
FinishByPredeterminedCondition (Distance are close enough, Stop Motion or Gesture)
};
```

Request

Function N	Return V	Argument	Remarks
StopComeToMe	bool	PosInfo pInfo	
StopFollowMe	bool	IDInfo iInfo	

Roadmap for Robotics Activities

robotics/2010-12-13

Item	Status	Cambridge MA Sep-2010	Santa Clara CA Dec-2010	Washington DC Mar-2011	Salt Lake City UT Jun-2011	Orlando FL Sep-2011	Santa Clara CA Dec-2011	POC / Comment
Flyer of Robotics-DTF [Publicity Sub-Committee]	Suspended							
Robot Interaction Service (RoIS) Framework RFP [Robotic Functional Services WG]	In Process		Initial Submittion		Revised Submittion & Voting			
Dynamic Deployment and Configuration for RTC (DDC4RTC) RFP [Robotic Infrastructure WG] in MARS	In Process		Initial Submittion		Revised Submittion & Voting			Sponsor: MARS
JAUS-RTC White Paper [Modelling for Robotics WG]	Planned			2nd Review & issue				
Models for Robotics RFI [Modelling for Robotics WG]	Planned			2nd Review & issue				
Robotic Map Services RFP [Robotic Functional Services WG]	Planned							IEEE R&A?
etc…	Future							
Robotics Information Day [Technology Showcase]	Future							
RTC Revision Task Force	Planned		Charter					
RLS Revision Task Force	In Process							will go to ISO/TC211

Related Events

Chu-suk
(Special Holidays in Korea)

Infrastructure WG Report

Seung-Woog Jung
Infrastructure WG, Robotics DTF
ETRI, KOREA

Infra WG meeting on Monday

- ETRI and AIST presented their own initial submissions.
- ETRI
  - Basic building blocks
    - Dynamic Configuration Planning
    - Dynamic Configuration Policy
    - Dynamic Configuration Execution
  - Comments
    - global plan should be scalable.
    - use D&C features if possible
    - there are some ambiguous definitions
    - …

Infra WG meeting on Monday

- AIST
  - Basic building blocks
    - Supervision Tree
    - Supervision FSM
    - RTC, RTS profiles
  - Comments
    - ambiguous terms : assembly, application, RTC-based system
    - supervision tree should cover the case that there are two parents
    - Supervision FSM is deterministic. So it should be extended to support some dynamic features.

Infra WG meeting on Tuesday

- Discussion how to merge ETRI and AIST submissions.

ETRI	AIST
•Policies •Event channels/Event filters •Directory/Repository/Query/Constraints •Execution •Planner •Global/local plans	•Supervision tree •FSM •RTC/RTS Profiles •Repository •Constraints specification
• Merge AIST repository into ETRI repository. **• Use ETRI directory.** **• Drop ETRI Execution (use D&C).** **• Drop ETRI Planner (use D&C).** **• Drop ETRI Global/local plans (use D&C).** **• Merge ETRI policies into AIST supervisors/RTX Profile.** **• Merge Event channels/Event filters into AIST FSM, AIST RTX Profile.** **• Merge AIST constraints and ETRI constraints.**	

MARS Plenary on Wednesday

- ETRI and AIST presented their initial submissions on MARS plenary.
- Comments
  - merge two submissions.

Future Plan

- Infra WG meeting on Thursday morning
  - integration of two submissions
- We hope to submit the merging document in the next OMG meeting

Robotic Functional Service WG
WG Report

WG Co-Chairs: SuYoung Chi, Miki Sato, Toshio Hori
2010/12/08

robotics/2010-12-15

Proposals Presentation (Mon., AM)

- 2 Proposals submitted by the deadline; From ETRI (Korea) and JARA (Japan).
- Presented at the Robotics-DTF plenary meeting in Monday morning.
  - Remote participant: Ingo Lütkebohle (Bielefelt University, Germany)

Discussion (Mon., PM/Tue./Wed.)

- Discussion for merging two proposals
  - Based on JARA's proposal
  - Surveyed and confirmed every item:
    - description
    - interfaces
    - sequence diagrams
    - profiles
    - messages

Discussion [cont.]

- Comment from Ingo (Bielefelt Univ.)
  - We didn't have enough time to discuss his comments and a document so made it our homework to review them.
- Private meeting before the March OMG TM.
  - Date: 2011/01/20(Thu.) – 22(Sat.)
  - Place: in Japan
- WG Meeting minutes (robotics/2010-12-09) will be posted to OMG server soon.

The 5th Korea-Japan-China Joint Workshop on Robotics

- Theme: Asian pioneering robotics -

1. Date:

October 29, 2010 (Friday)

<Related Events>

- ▶ Robot World 2010, October 28-31, KINTEX, http://www.robotworld.or.kr/
- ▶ International Conference on Control, Automation and Systems (ICCAS) 2010, October 27-30, KINTEX, http://2010.iccas.org/

2. Venue

KINTEX, Koyang-City, Kyeonggi-do

3. Organizing Committee

General Chair: Sang-ROk Oh
(Vice President, KIST)

Organizing Committee Co-Chairs:

Tomomasa Sato(University of Tokyo)

Ping Zhou (Ministry of Science and Technology of the PRC)

Sang Moo Lee(Robot Program Director, Ministry of Knowledge Economy)

Program Co-Chairs:

Tetsuo Kotoku(AIST, Japan)

Tianmiao Wang (Beihang University, China)

Young-Jo Cho(ETRI, Korea)

4. Program (draft)

8:30 ~ 9:00 Opening Ceremony

9:00 ~ 10:30 Robotics R&BD policies in 3 countries

- Takanori Shibata(AIST, Japan)
- Tianmiao Wang (Beihang University, China)
- Hong-Seong Park(Kangwon National University, Korea)

10:30 ~ 10:45 Coffee break

10:45 ~12:15 Pioneering efforts for service robot industrialization in 3 countries

- Norihiro Hagita(ATR, Japan)
- Caihua Xiong (Huazhong University Science and Technology, China)
- Il Han Bae(Electronics Times/Korea, Hawaii Univ./USA)

12:15 ~ 14:00 Lunch

14:00 ~ 15:00 Experiences and challenges in robotics technologies (3 talks, 20 minutes/talk)

- Shinsuke Sakakibara(FANUC LTD)
- Jianda Han (Shenyang Institute of Automation Chinese Academy of Sciences, China)
- Myeong Ho Yoo(Samsung Techwin Co. LTD)

15:00 ~ 16:00 Status and perspective of robotics standardization

- Tetsuo Kotoku(AIST, Japan)
- Hongxing Wei (Beihang University, China)
- Young-Jo Cho(ETRI, Korea)

16:00 ~ 16:15 Coffee break

16:15 ~ 17:00 Panel Discussion: Asian challenges for revitalizing robot industry

Moderator: Young-Jo Cho(ETRI)

Tomomasa Sato(University of Tokyo)

Shinsuke Sakakibara (FANUC LTD)

Tianmiao Wang (Beihang University, China)

Fang Xu (SIASUN Robot & Automation Co., Ltd., China)

Sang-Rok Oh(KIST), Sang Moo Lee(MKE)

17:00 ~ 17:30 Closing Discussion

- Next year workshop announcement

Tsutomu Hasegawa(Kyushu Univ.)

18:00 ~ Banquet

RTC Standard Revision Task Force Proposal

Geoffrey Biggs

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

December 14, 2010

5.2.2.6.8 Should either note the pre-condition for the component to be in the inactive state before activation, or clarify that the `on_activate` callback will only be called when changing from inactive to active.

- State diagram in Figure 5.5 *implies* this, but description in section 5.2.2.6.8 makes it ambiguous by stating that `on_activated` is called as a result of calling `activate_component`, not as a result of the state transition.

5.2.2.6.9 Same problem for `deactivate_component`.

5.2.2.6.11 `get_component_state` has a constraint that the RTC must be alive, but provides no specification for error codes when it is not. Implementations must choose their own response (OpenRTM does not fail for RTCs in the created state). Will lead to incompatible implementations.

5.2.2.2.9 Ambiguous about whether *owned* ECs should be in the *participating* ECs list.

- Not specified if it is possible to detach an owned EC or not - should it be an error or should the RTC stop participating in the EC?

- Exit semantics are not clearly defined. If an error occurs during `exit()` or `finalize()`, what state should the RTC be in?

5.2.2.2.7 Return code from `get_context()` (and similar) is not specified for the case of an invalid EC handle. Will lead to incompatible implementations.

5.2.2.2 Ambiguity about whether a passive RTC is alive or created.

Figure 5.3 ECs must be owned by an RTC. This is inflexible; it is not possible to supply an EC from an external source (e.g. a node manager in deployment) without creating an RTC to own it.

- Logic for exiting an RTC requires re-entrant RTCs. This is difficult in some CORBA implementations.

Proposed Charter for Robotic Technology Component (RTC) RTF

TC Meeting Date: 10 December, 2010
Presenter: Young-Jo Cho
Group email: {obtain from Juergen}
WIP page (URL): {obtain from Juergen}

- **Available Specification:**
  - formal/08-04-04 – Specification document
  - ptc/07-09-10 - XMI
  - ptc/07-08-21 - IDL
  - ptc/07-08-22 – C++ Header
- **Members:**
  - Geoffrey Biggs (chair) (AIST)
  - Takeshi Sakamoto (Technologic Arts)
  - Seung Woog Jung (ETRI)
  - Chul Jong Hwang (KAR)
  - Takashi Suehiro (UEC)
  - Makoto Mizukawa (Shibaura IT)
  - Saku Egawa (Hitachi)
  - Noriaki Ando (JARA/AIST)

- **Deadlines:**
  - Comments Due: June 1, 2011
    - Should be set 10 weeks before the report is due* to allow sufficient time for issue resolution.
  - Report Due Date: August 22, 2011
  - Report Deadline: September 30, 2011

Roadmap for Robotics Activities

robotics/2010-12-19

Item	Status	Cambridge MA Sep-2010	Santa Clara CA Dec-2010	Washington DC Mar-2011	Salt Lake City UT Jun-2011	Orlando FL Sep-2011	Santa Clara CA Dec-2011	POC / Comment
Flyer of Robotics-DTF [Publicity Sub-Committee]	Suspended							
Robot Interaction Service (RoIS) Framework RFP [Robotic Functional Services WG]	In Process		Initial Submittion		Revised Submittion & Voting			
Dynamic Deployment and Configuration for RTC (DDC4RTC) RFP [Robotic Infrastructure WG] in MARS	In Process		Initial Submittion		Revised Submittion & Voting			Sponsor: MARS
JAUS-RTC White Paper [Modelling for Robotics WG]	Suspended			2nd Review & issue				
Models for Robotics RFI [Modelling for Robotics WG]	Suspended			2nd Review & issue				
Robotic Map Services RFP [Robotic Functional Services WG]	Planned							IEEE R&A?
etc…	Future							
Robotics Information Day [Technology Showcase]	Future							
RTC Revision Task Force	Planned		Charter					
RLS Revision Task Force	In Process							will go to ISO/TC211

Related Events

Chu-suk
(Special Holidays in Korea)

Robotics Domain Task Force Preliminary Agenda ver0.0.3 **robotics/2010-12-20**

OMG Technical Meeting - Washington DC, USA -- March 21-25, 2011

http://robotics.omg.org/

		TF/SIG				
		Host	Joint (Invited)	Agenda Item	Purpose	Room
Monday:						
12:00	13:00			**LUNCH**		
13:00	18:00			Architecture Board Plenary		
Tuesday: WG activity						
9:00	12:00			**DDC4RTC** (Robotic Infrastructure) WG(3h) - Noriaki Ando (AIST) and Seung-Woog Jung (ETRI)	discussion	
				RoIS (Robotic Functional Services) WG(3h): - Su-Young Chi (ETRI), Miki Sato (JARA/ATR) and Toshio Hori (AIST)	discussion	
12:00	13:00			**LUNCH**		
13:00	18:00			**DDC4RTC** (Robotic Infrastructure) WG(5h) - Noriaki Ando and Seung-Woog Jung	discussion	
				RoIS (Robotic Functional Services) WG(5h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	
Wednesday: WG activitiy						
9:00	12:00			**DDC4RTC** (Robotic Infrastructure) WG(3h) - Noriaki Ando and Seung-Woog Jung	discussion	
				RoIS (Robotic Functional Services) WG(3h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	
12:00	14:00			**LUNCH and OMG Plenary**		
14:00	18:00			**DDC4RTC** (Robotic Infrastructure) WG(4h) - Noriaki Ando and Seung-Woog Jung	discussion	
				RoIS (Robotic Functional Services) WG(4h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	
18:00	20:00			**OMG Reception**		
Thursday: WG activitiy(am) and Robotics-DTF Plenary(pm)						
9:00	12:00			**DD4RTC** (Robotic Infrastructure) WG(3h) - Noriaki Ando and Seung-Woog Jung	discussion	
				RoIS (Robotic Functional Services) WG(3h): - Su-Young Chi, Miki Sato and Toshio Hori	discussion	
12:00	13:00			**LUNCH**		
13:00	18:00			Architecture Board Plenary		
13:00	14:00	Robotics		Special Talk: - **TBA**	presentation and discussion	
14:00	15:00	Robotics		Special Talk: - **TBA**	presentation and discussion	
				Break (30min)		
15:30	16:30	Robotics		WG Reports and Discussion (Service WG, Infrastructure WG, Models in Robotics WG)	presentation and discussion	
16:30	17:00	Robotics		Contact Reports: - Makoto Mizukawa(Shibaura-IT), and Young-Jo Cho(ETRI)	Information Exchange	
17:00	17:30	Robotics		Robotics-DTF Plenary Wrap-up Session (**DTF Co-Chair Election**, Roadmap and Next meeting Agenda)	Robotics plenary closing, **voting**	
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	
Friday						
8:30	12:00			AB, DTC, PTC		
12:00	13:00			**LUNCH**		

Other Meetings of Interest

Monday						
8:00	8:45	OMG		New Attendee Orientation		
18:00	19:00	OMG		New Attendee Reception (by invitation only)		
Tuesday						
7:30	9:00	OMG		Liaison ABSC		

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

Robotics-DTF

Date: Friday, 10th December, 2010
Chair:, T. Kotoku and Y. –J. Cho
URL: http://robotics.omg.org/
email: robotics@omg.org

➢Highlights from this Meeting:

Robotics Plenary: (18 participants)

- 1 Special Talk
  - "(How we are) Building Blocks for Mobile Manipulation (that you can reuse)", Brian Gerkey, Willow Garage [robotics/2010-12-06]
- 2 WG Reports [robotics/2010-12-14,-15]
- 1 Contact Report [robotics/2009-12-16]
- RTC(Robot Technology Component) RTF proposal [robotics/2009-12-17]
- Preliminary agenda for upcoming meeting [robotics/2012-12-20]

Robotics-DTF

Date: Friday, 10th December, 2010
Chair:, T. Kotoku and Y. –J. Cho
URL: http://robotics.omg.org/
email: robotics@omg.org

➢Future deliverables (In-Process):

- Dynamic Deployment and Configuration for RTC (DDC4RTC) revised submission
- Robotic Interaction Service(RoIS) Framework revised submission

➢Next Meeting (Washington DC):

- Election of a Robotics DTF Co-Chair
- Discussion for revised submission of DDC4RTC
- Discussion for revised submission of RoIS
- Guest presentations
- Roadmap discussion
- Contact reports

Minutes of the Robotics DTF Meeting - DRAFT

December 6-10, 2010

Santa Clara, CA, USA

(robotics/2010-12-22)

Meeting Highlights

- Two initial submissions for the Dynamic Deployment and Configuration for Robotic Technology Component (DDC4RTC) RFP were presented in the joint plenary with MARS-PTF.
- Two initial submissions for the Robotic Interaction Service (RoIS) Framework RFP were presented.
- As a special talk, "(How we are) Building Blocks for Mobile Manipulation (that you can reuse)" was presented by Dr. Brian Gerkey (Willow Garage) [robotics/2010-12-06].
- Robotic Technology Component (RTC) 1.1 RTF was chartered [robotics/2009-12-17,18].
- Additional DTF Co-Chair election was announced.

List of Generated Documents

robotics/2010-12-01 Final Agenda (Tetsuo Kotoku)
robotics/2010-12-02 Cambridge Meeting Minutes [approved] (Takashi Tsubouchi and Tetsuo Kotoku)
robotics/2010-12-03 Opening Presentation (Tetsuo Kotoku)
robotics/2010-12-04 ETRI Presentation of Robotic Interaction Service Framework (RoIS) Initial Submission (Su-Young Chi)
robotics/2010-12-05 JARA Presentation of Robotic Interaction Service Framework (RoIS) Initial Submission (Toshio Hori)
robotics/2010-12-06 Special Talk: (how we are) Building Blocks for Mobile Manipulation (that you can reuse) (Brian Gerkey)
robotics/2010-12-07ETRI Presentation of Dynamic Deployment and Configuration for RTC (DDC4RTC) Initial Submission (Seung-Woog Jung)
robotics/2010-12-08 AIST Presentation of Dynamic Deployment and Configuration for RTC (DDC4RTC) Initial Submission (Geoffrey Biggs)
robotics/2010-12-09 Robotic Functional Services WG Minutes (Dec. 6-7, 2010) (Miki Sato)
robotics/2010-12-10 ETRI Presentation of Dynamic Deployment and Configuration for RTC (DDC4RTC) Initial Submission [mars/2010-12-24] (Seung-Woog Jung)
robotics/2010-12-11 AIST Presentation of Dynamic Deployment and Configuration for RTC (DDC4RTC) Initial Submission [mars/2010-12-25] (Geoffrey Biggs)
robotics/2010-12-12 RoIS Framework Tracking Issue (Su-Young Chi)
robotics/2010-12-13 Wrap-up Presentation (Tetsuo Kotoku)
robotics/2010-12-14 Robotic Infrastructure WG Report (Seung-Woog Jung)
robotics/2010-12-15 Robotic Functional Services WG Report (Toshio Hori)
robotics/2010-12-16 The 5th Korea-Japan-China Joint Workshop on Robotics (Young-Jo Cho)
robotics/2010-12-17 RTC-RTF Proposal (Geoffrey Biggs)
robotics/2010-12-18 Charter for RTC-RTF (Geoffrey Biggs)
robotics/2010-12-19 Roadmap for Robotics Activities (Tetsuo Kotoku)
robotics/2010-12-20 Next Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)
robotics/2010-12-21 DTC Report Presentation (Young-Jo Cho)
robotics/2010-12-22 SantaClara Meeting Minutes - DRAFT (Miki Sato and Myung-Eun Kim)

Minutes

Monday, Dec 06, 2010, Winchester, 2nd FL
Robotics DTF Plenary Meeting

8:40-9:00 Robotics DTF Opening Session, Chair: Dr. Kotoku, Quorums: 3
AIST, ETRI, JARA, KAR, Shibaura IT, Univ. of Tsukuba, UEC, Toshiba, Technologic Arts, ATR

- Minutes takers: Miki Sato (ATR) and Myung-Eun Kim (ETRI)
- Cambridge Meeting Review will be on Wednesday
- Agenda Review
  . Monday:
    Initial Submission Presentation for RoIS RFP (Su Young Chi/ETRI)
    Initial Submission Presentation for RoIS RFP (Toshio Hori/AIST)
    Special Talk (Brian Gerkey/Willow Garage)
  . Tuesday:
    Robotic Infrastructure WG/Robotic Functional Services WG
  . Wednesday:
    Joint Plenary with MARS
      .. Initial Submission Presentation for DDC4RTC RFP (Seung-Woog Jung/ETRI)
      .. Initial Submission Presentation for RoIS RFP (Geoffrey Biggs/AIST)
    WG and Contact Reports, Wrap up

09:00-10:00 Initial Submission Presentation for RoIS RFP, Winchester, 2nd FL
- Initial Submission Presentation for RoIS RFP, Su Young Chi, ETRI, Korea
- Proposed lists of messages for 3 interface in RoIS Framework (Event, Command, Query).
- HRI specified functions with BioAPI & RLS are included in APIs of HRI engine.
- Presentation of experimental demonstration for human tracking

10:00-11:00 Initial Submission Presentation for RoIS RFP, Winchester, 2nd FL
- Initial Submission Presentation for RoIS RFP, Toshio Hori, AIST, Japan
- Proposed sequences and messages for 4 interface types (System, Event, Command, Query).
- Profiles of the messages are defined for functionality of HRI engine and components.
- Some comments for consideration of robot mobility, e.g. human tracking
- Some comments for specific scenario of Dialog Action

11:00-12:00 Special Talk: Brian Gerkey (Willow Garage),
- Title: (How we are) Building blocks for mobile manipulation (that you can re-use)
- Introduced how to build blocks: document, test, release.
- The status of building blocks for mobile manipulation in Willow Garage.
- Introduced some field experiments for building blocks for PR2 e.g. opening a door, plugging-in, fetching beer.
- Automated regression tests are conducted on a simulator for each task.
- Wiki pages and packaged codes are prepared for each experiment to improve reproducibility.

Wednesday, Dec 08, 2010, Bayshore E, 2nd FL
Robotics DTF Plenary Meeting

14:00-16:00 Joint Plenary with MARS, Laurence, 2nd FL
- Initial submission presentation for DDC4RTC, Seung-Woog Jung, ETRI, Korea
- Initial submission presentation for DDC4RTC, Geoffrey Biggs, AIST, Japan
- Future plan for the revised submission
  . AIST and ETRI have discussed how to merge two initial submissions and they will make the revised submission by merging them.

16:00-16:10 Infrastructure WG, Seung-Woog Jung (ETRI)
- ETRI and AIST presented their own initial submissions on Monday.

. Exchanged some comments for each initial submission.
. Discussed how to merge ETRI and AIST initial submissions.
- ETRI and AIST got some comments from MARS plenary on Wednesday.
. It is required to merge two initial submissions.
- Future schedule will be fixed on Thursday.

16:10-16:20 Functional Service WG, Toshio Hori (AIST)
- ETRI and JARA presented their initial submissions on Monday.
- ETRI and JARA reached an agreement to merge two initial submissions based on JARA's proposal.
- ETRI and JARA reviewed JARA's proposal in detail and clarified the problems.
- Future plan of private meeting for proposal.
. 2011.01.20 – 2011.01.22 in Japan

16:20-16:25 Functional Service WG Minutes, Miki Sato (ATR)
- Presented the review of JARA's proposal for RoIS RFP.
- Presented the problems for revising the proposal based on discussion.

16:25-16:35 Contact Report, Young-Jo Cho (ETRI)
- The 5th Korea-Japan-China Joint Workshop on Robotics was held in KINTEX, Korea, Oct. 29, 2010
- Presented Robotics R&BD policies in three countries.
- Presented pioneering efforts for service robot industrialization in three countries.
- Presented experiences and challenges in robotics technologies.
- Presented stats and perspective of robotics standardization.

16:35-17:15 RTC 1.1 RTF Proposal, Geoffrey Biggs (AIST)
- Proposed to make a RTC standard revision Task Force.
- Presented the revision parts of the RTC specification.
- The Revision scope is restricted to fix trivial problems.
- ETRI suggested joining the revision task force to modify the RTC specification.
The suggestion of the modification will be posted as comments.
- Agreement to make a RTC RTF
. Members:
Geoffrey Biggs(chair)(AIST)
Takeshi Sakamoto(Technologic Arts)
Seung-Woog Jung(ETRI)
Chul-Jong Hwang(KAR)
Takashi Suehiro(UEC)
Makoto Mizukawa(Shibaura IT)
Saku Egawa(Hitachi)
Noriaki Ando(JARA/AIST)
. Comments Due Date: June 01, 2011
. Report Due Date: Aug. 22, 2011
. Report Deadline: Sep. 30, 2011
- The motion of this charter will be made on MARS.

17:15-17:30 Wrap-up Session, Chair: Dr. Kotoku (AIST)
- Laurent Rioux (Thales) won't attend Robotics-DTF as a DTF/WG chair.
- We would like to thank Laurent for his contribution.
- It is required to elect a new co-chair of Robotics-DTF in next meeting.
- Suspend Modeling for Robotics WG activity.
- Next Meeting Agenda (03/22(Tue.) – 03/24(Thur.) in Washington D.C)

ATTENDEE (18 participants)

- Brian Gerkey (Willow Garage)
- Chul-Jong Hwang (KAR)
- Geoffrey Biggs (AIST)
- Katsuhiro Mayama (Shibaura IT)
- Makoto Mizukawa (Shibaura IT)
- Miki Sato (ATR)
- Miwako Doi (Toshiba)
- Myung-Eun KIM (ETRI)
- Noriaki Ando (AIST)
- Seung-Woog Jung (ETRI)
- Su-Young Chi (ETRI)
- Takashi Suehiro (Univ. of Electro-Communication)
- Takashi Tsubouchi (Univ. of Tsukuba)
- Takeshi Sakamoto (Technologic Arts)
- Tetsuo Kotoku (AIST)
- Toshio Hori (AIST)
- Young-Jo Cho (ETRI)
- Yusuke Zama (Shibaura IT)

Remote ATTENDEE through the GoToMeeting (one participant)

- Ingo Lütkebohle (Bielefeld Univ.)

Prepared and submitted by Miki Sato (ATR) and Myung-Eun Kim (ETRI)