**Robotics Domain Task Force Final Agenda ver.1.0.3** **robotics/2006-06-03**

**OMG Technical Meeting - Boston, MA, USA -- June 26-30, 2006**

| | | TF/SIG | | | | |
|---|---|---|---|---|---|---|
| | | Host | Joint (Invited) | Agenda Item | Purpose | Room |
| **Sunday (June 25)** | | | | | | |
| 13:00 | 17:00 | SDO | Robotics | Robotics Technology Components(RTCs) Submitter's Meeting | discussion | **Congress 313, 3rd FL** |
| **Monday (June 26) WG activity** | | | | | | |
| 8:40 | 9:00 | Robotics | (SDO) | Welcome and review agenda | Robotics/SDO Joint Meeting Kick-off | **Aquitania C, 1st FL** |
| 9:00 | 10:00 | Robotics | (SDO) | **Robotic Services WG**: Presentation of RUPI (tentative)<br>- Soo-Young Chi | discussion | |
| 10:00 | 11:00 | MARS | SDO, Robotics | Robot Technology Components RFP revised submission presentation **[10:00-11:00]** | review, vote-to-vote, voting | **Michelangelo A, 2nd FL** |
| 11:00 | 12:00 | Robotics | (SDO) | **Robotic Services WG**: Definition of Functional Services in Robotic Systems, WG Steering Committee, Roadmap Update<br>- Olivier Lemaire and Soo-Young Chi | discussion | **Aquitania C, 1st FL** |
| 12:00 | 13:00 | | | **LUNCH** | | **Pavilion** |
| 13:00 | 18:00 | | | Architecture Board Plenary | | **Gr. Ballroom A, 2nd FL** |
| 13:00 | 15:00 | Robotics | | **Profile WG:** Discussion on profile standardization<br>- Seung-Ik Lee, Bruce Boyce | Sharing common concept on profile standarization | **Gr. Ballroom, 2nd FL** |
| 15:00 | 17:00 | Robotics | SDO | Steering Committee of Robotics DTF<br>(include Publicity SC discussion) | Volunteer recruit | |
| **Tuesday (June 27) WG activity** | | | | | | |
| 8:30 | 9:00 | Robotics | SDO | **Robotic Service WG**: POEM : an implementation of position estimation module | discussion | **Michelangelo B, 2nd FL** |
| 9:00 | 9:30 | Robotics | SDO | **Robotic Service WG**: SAIT proposal of standards for Robot Localization based on MDA<br>- Yeon-Ho Kim (Samsung) | discussion | |
| 9:30 | 10:00 | Robotics | SDO | **Robotic Service WG**: Discussion on Localization Technology<br>- Wonpil Yu (ETRI) | discussion | |
| 10:00 | 12:00 | Robotics | SDO | **Robotic Service WG**: Discussion on localization technology : Information sharing concerning research on existing technology and standards? What should/could be standardized ? What the business cases are for each items.<br>- Olivier Lemaire and Soo-Young Chi | discussion | |
| 12:00 | 13:00 | | | **LUNCH** | | **Pavilion** |
| 13:00 | 14:00 | Robotics | (SDO) | Spur- a locomotion command system<br>- Eijiro Takeuchi and Takashi Tsubouchi (Tsukuba Univ.) | Infomative | **Michelangelo B, 2nd FL** |
| 14:00 | 15:00 | Robotics | (SDO) | Robot Modeling Framework -- Demonstration --<br>- Abheek Bose (ADASoftware) | Infomative | |
| 15:00 | 16:00 | Robotics | SDO | **Infrastructure WG:** RSCA and the Deployment and Configuration of Components<br>- Prof. Seongsoo Hong (Seoul National Univ.) | presentation | |
| 16:00 | 17:00 | Robotics | SDO | **Infrastructure WG:** Requirements for Deplaymnent and Configuration of RTCs<br>- Saehwa Kim, Noriaki Ando, Rick Warren | discussion | |
| **Wednesday (June 28) Robotics Plenary** | | | | | | |
| 8:50 | 9:00 | Robotics | (SDO) | Plenary Openning | Robotics/SDO joint plenary kick-off | **Michelangelo B, 2nd FL** |
| 9:00 | 9:40 | Robotics | (SDO) | **Special Talk:** RoboBusiness2006 Report<br>- Jon Siegel (OMG) | Infomative | |
| | | | | **Break (20min)** | | |
| 10:00 | 12:00 | Robotics | (SDO) | WG Reports and Roadmap Discussion<br>(Infrastructure, Robotic Service, Profile) | reporting and discussion | |
| 12:00 | 14:00 | | | **LUNCH and OMG Plenary** | | **Pavilion** |
| 14:00 | 15:00 | SDO | Robotics | Robot Technology Components RFP revised submission presentation<br>- Rick Warren and Noriaki Ando | reporting | **Michelangelo B, 2nd FL** |
| | | | | **Break (30min)** | | |
| 15:30 | 16:30 | Robotics | | Contact Reports:<br>- Makoto Mizukawa(ISO), Wataru Inamura(JAUS), and Yun-Koo | | |
| 16:30 | 17:00 | Robotics | | Publicity SC Report, Next meeting Agenda Discussion | Robotics/SDO joint plenary closing | |
| 17:00 | 18:00 | OMG | | Chair's Meeting | | **Britannic AB, 1st FL** |
| **Thursday** | | | | | | |
| 8:30 | 9:00 | Robotics | | Robotics WG Co-chairs Planning Session (Agenda for Anaheim) | planning for next meeting | **Constellation 316, 3rd FL** |
| 9:00 | 10:00 | Robotics | SDO, Ontoloty | **Special Talk:** Introduction to OMG Ontology-PSIG (tentative)<br>- Elisa F. Kandall and Evan Warrance (Ontology-PSIG) | Information Exchange | |
| 10:00 | 11:30 | Robotics | | WG activity (1.5h) | follow up discussion | |
| **11:30** | **12:00** | **MARS** | **SDO, Robotics** | **Robot Technology Components RFP revised submission voting** | voting | **Aquitania AB, 1st FL** |
| 12:00 | 13:00 | | | **LUNCH** | | **Pavilion** |
| 13:00 | 18:00 | | | Architecture Board Plenary | | **Gr. Ballroom A, 2nd FL** |
| 13:00 | 17:00 | Robotics | | WG activity (4h) | follow up discussion | **Constellation 316, 3rd FL** |
| **Friday** | | | | | | |
| 8:30 | 12:00 | | | AB, DTC, PTC | | **Gr. Ballroom AB, 2nd FL** |
| 12:00 | 13:00 | | | **LUNCH** | | **Pavilion** |

**Other Meetings of Interest**

| | | | | | | |
|---|---|---|---|---|---|---|
| **Monday** | | | | | | |
| 8:00 | 8:45 | OMG | | New Attendee Orientation | | Queen Mary, 2nd FL |
| 9:00 | 12:00 | OMG | | Introduction to OMG's meeting and Middlewere Specifications | | Queen Mary, 2nd FL |
| 13:00 | 17:00 | OMG | | Softwere Radio Specification Tutorial | | Queen Mary, 2nd FL |
| 18:00 | 19:00 | OMG | | New Attendee Reception (by invitation only) | | Skyline, 1st FL |
| **Wednesday** | | | | | | |
| 9:00 | 12:00 | OMG | | Introduction to the OMG Systems Modeling Language (OMG SysML) | | Aquitania AB, 1st FL |
| 9:00 | 18:00 | OMG | | Service Oriented Architecture (SOA) Information Day | | Skyline, 1st FL |

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

# Minutes of the Robotics DTF Plenary – St. Louis, MO, USA – Approved (robotics/2006-06-04)

## Overview and votes

We had two special Talks and six RFI response presentations. By the excellent leadership of seven volunteers, we had very active group discussions and three working groups were chartered.

## OMG Documents Generated

robotics/2006-04-04 Robotics-DTF Final Agenda (Tetsuo Kotoku)
robotics/2006-04-05 Steering Committee Presentation (Tetsuo Kotoku)
robotics/2006-04-06 Tampa Meeting Minutes [approved] (Saku Egawa)
robotics/2006-04-07 Opening Presentation (Tetsuo Kotoku)
robotics/2006-04-08 Robotics-DTF Roadmap (Tetsuo Kotoku)
robotics/2006-04-09 "Real-Time ORB Middleware: Standards, Applications, and Variations" (Christopher Gill, Washington Univ.)
robotics/2006-04-10 "Communication protocol for the URC robot and server" (Hyun-Sik Shim, Samsung Electronics)
robotics/2006-04-11 "URBI: a Universal Platform for Personal Robots" (Jean-Christophe Baillie, ENSTA/Gostai)
robotics/2006-04-12 "Fujitsu's robotics research and standardization activites" (Toshihiko Morita, Fujitsu)
robotics/2006-04-13 "Standardization of device interfaces for home service robot" (Ho-Chul Shin, ETRI)
robotics/2006-04-14 "Voice interface standardization items network robot in noisy environments" (Soon-Hyuk Hong, Samsung Electronics)
robotics/2006-04-15 "Home robot navigation in SAIT" (Seok-Won Bang and Yeon-Ho Kim, Samsung Advanced Institute of Technology)
robotics/2006-04-16 "ITR – Internet Renaissance ~ The world's first humanoid robot to be harmonized with the family~" (Hiroyuki Nakamoto, Systems Engineering Consultants)
robotics/2006-04-17 Service WG activity report (Olivier Lemaire, JARA and Chi, ETRI)
robotics/2006-04-18 Profile WG activity report (Bruce Boyes, Systronix and Seung-Ik Lee, ETRI)
robotics/2006-04-19 Infrastructure WG activity report (Noriaki Ando, AIST and Rick Warren, RTI)
robotics/2006-04-20 Contact Report: Korea Intelligent Robot Standardization Forum (KIRSF) (Yun-Koo Chung, ETRI)
robotics/2006-04-21 Contact Report: ORiN (Makoto Mizukawa, Shibaura Institute of Technology)
robotics/2006-04-22 Robot Technology Components RFP Progress Report in MARS (Rick Warren, RTI)
robotics/2006-04-23 Summary Report of Robotic Systems RFI (Tetsuo Kotoku, AIST)
robotics/2006-04-24 DTC Report Presentation (Yun-Koo Chung, ETRI)
robotics/2006-04-25 Meeting Minutes - DRAFT (Hung Pham, RTI)
robotics/2006-04-27 Robotics Profile WG (Seung-Ik Lee)

## Agenda

Monday, April 24, 2006
13:00-14:20 Group discussion for Service WG
14:20-15:40 Group discussion for Profile WG
15:40-17:00 Group discussion for Tool WG
17:00-18:00 Steering Committee of Robotics-DTF

Tuesday, April 25, 2006 [Robotics-DTF Plenary]
08:30-09:00 Progress Report of Robot Technology Components RFP [MARS]
10:05-10:20 Welcome and Review Agenda
10:20-11:20 Special Talk: Prof. Chris Gill (Washington Univ., St. Louis)
11:20-12:00 RFI Presentation
13:00-14:00 Special Talk: Prof. Jean-Cristophe Baillie (ENSTA, France)
14:00-16:20 RFI presentations
16:20-17:40 Group discussion for Infrastructure WG

Wednesday, April 26, 2006 [Robotics-DTF Plenary]
08:30-09:50 RFI presentation
10:00-11:40 WG reports and Roadmap Discussion
11:40-11:50 Contacts Report
11:50-12:00 Next Meeting Agenda Discussion, etc
12:00 Plenary Adjourn
14:00-18:00 Infrastructure WG / Robotic Services WG

Thursday, April 27, 2006
08:30-10:00 RTCs RFP submitter's meeting (open discussion)
10:00-12:00 Robotic Services WG
13:00-14:00 Robotics RFI Summary Report [MARS]

## Minutes

**25 Apr 2006 (Tuesday)**

Meeting called to order at 10:05am

Attendees:
Tetsuo Kotoku (AIST)
Makoto Mizukawa (SIT)
Hideo Shindo (NEDO)
Saku Egawa (Hitachi)
Takashi Suehiro (AIST)
Masayoshi Yokomachi (NEDO)
Eul Gyoon Lim (ETRI)
Ho-Chul Shin (ETRI)
Takashi Tsubouchi (Tsukuba Univ.)
Hung Pham (RTI)
Olivier Lemaire (JARA)
J. Christophe Baillie (GOSTAI)
Dono di Crescento? (SELEX SI)
Chris Gill (Washington Univ.)
Hyun-Sik Shim (Samsung)
Soon-Hyuk Hong (Samsung)
Yun-Koo Chung (ETRI)
Soo-Yung Chi (ETRI)
Hiroyuki Nakamoto (SEC)
Toshihiko Morita (Fujitsu)
Seok-Won Bang (Samsung)
Wonpil Yu (ETRI)
Bruce Boyes (Systronix)
Noriaki Ando (AIST)
Rick Warren (RTI)
Joseph Jacob (OIS)
Hiroyuki Nakamoto (SEC)

Yeon-Ho Kim (Samsung)
Victor Giddings (OIS)

Review of Tampa meeting (Tetsuo Kotoku (TK))

Approval of Tampa meeting minutes
- RTI made motion to approve minutes
- ETRI seconded the motion
- Motion passed

Overview of the meeting agenda (TK)

**Special talk: "Real-Time ORB Middleware: Standards, Applications, and Variations." Prof. Chris Gill (cdgill@cse.wsutl.edu)**
(robotics/2006-04-09)
- "Standards enforce commonality"
- However, "…applications are heterogeneous"
- "What if commonality & heterogeneity don't match"
- "Developing and using standards-based middleware effectively demands attention to the [above] issues"
- Stressed the need for "agile" specification that does not encumber upon the user unnecessary features. These unneeded features are expensive, *e.g.*, computational, bandwidth, memory, *etc.*, and add no value.

**RFI response: "Communication protocol for the URC robot and server." Hyun-Sik Shim (Samsung Electronics)**
(robotics/2006-04-10)
- Goal to distribute 3 functional components through network, *i.e.*, sensing, processing, & action. Sensing → richer set of sensing capabilities can be available from external network of sensors. Processing → more powerful server-side computers.
- Client-server architecture (achieved scale of 16 servers, 64 clients)
- Successfully demonstrated on several platforms

**Special talk: "URBI: a Universal Platform for Personal Robotics." Prof. Jean-Christof Baillie (ENSTA/UEI Lab)**
(robotics/2006-04-11)
- Stressed need for universal platform to program proliferation of robots
- Interface object scripting language that incorporates event-driven and parallel execution semantics, which sets it apart from, *e.g.*, Python, etc.
- Open architecture; can integrate CORBA or other DOM interface objects from many languages (*e.g.*, C++, Java, Matlab, pyton, *etc.*); can adapt to existing standards or push towards their creation

**RFI response: "Fujistu's robotics research and standardization activities." Toshihiko Morita (Fujistu)**
(robotics/2006-04-12)

- Humanoid for Open Architecture Platforms (HOAP); sold over 100 units
- Mobile Agent Robot Of the Next-generation (MARON)
- Service robot, named exciting nova on network (enon)
- Scripts used to describe execution scenario, *e.g.*, escorting, patrolling, *etc*.
- Identified a number of needs for standardization, *e.g.*, protocol for receiving information from network, internal software components
- RSi Service Architecture specifies necessary functions to provide robot services = RSi Common Service + Profiles

**RFI response: "Home service robot hardware component interface." Ho-Chul Shin (ETRI)**
(robotics/2006-04-13)
- developing LEGO-type embedded systems for low cost, popularized home service robot
- robot core chipsets and SoC
- networked thin-client robot, Ubiquitous Robot Companion
- sensor networks, *etc.*
- Physical Media independent Interface (PMI) provides a standard interface to range of physical devices; consists of a Device Data Management Layer and Device Connection Management Layer.

**RFI response: "Voice Interface Standardization Items for Network Robot in Noisy Environments." Soon-Hyuk Hong (Samsung)**
(robotics/2006-04-14)
- Vision of the robot as the interface between the user and the "robot system"
- Speech recognition is performed offline
- Stressed need for standardization of voice interface
  + would reduce the uncertainties of robot's voice recognition performance in noisy environments
  + prevent investment overlap and cut down on development costs, *etc.*
- Points of standardization
  + mic. & array characteristics for network robot
  + speech recognition performance guideline
  + input/output parameter for communication between server and network robot
  + resource portion of network robot

Meeting adjourned at 4:20pm.

**26 Apr 2006 (Wednesday)**

Meeting called to order at 8:30am.

Attendees:
Tetsuo Kotoku(AIST)
Yun-Koo Chung (ETRI)

Hideo Shindo (NEDO)
Hiroyuki Nakamoto (SEC)
Takashi Suehiro (AIST)
Makoto Mizukawa (SIT)
Masayoshi Yokomachi (NEDO)
Takashi Tsubouchi (Tsukuba Univ.)
Olivier Lemaire (JARA)
Hung Pham (RTI)
Soo-Young Chi (ETRI)
Hyun-Sik Shim (Samsung)
Bruce Boyes (Systronix)
Joseph Jacob (OIS)
Saku Egawa (Hitachi)
Soo-Hyuk Hong (Samsung)
Toshihiko Morita (Fujitsu)
Seok-Won Bang (SAIT)
Wonpil Yu (ETRI)
Ho-Chul Shin (ETRI)
Eul Gyoon Lim (ETRI)
Noriaki Ando (AIST)
Roger Burkhart (Deere & Company)

**RFI response: "Home robot navigation in SAIT." Seok Won Bang and Y. H. Kim (Samsung Advanced Institute of Technology)**
(robotics/2006-04-15)
- Presented brief history of Samsung home service robot.
- Developing context-aware technologies for home service robots.

**RFI response: "ITR – The world's first contents downloadable humanoid robot harmonize with the family." Hiroyuki Nakamoto (Systems Engineering Consultants)**
(robotics/2006-04-16)
- ITR = internet renaissance + internet robot (*i.e.*, walking internet radio)
- ITR server connects robot to other services, *e.g.*, contents server, device controller, *etc*.
- proposing a Robot Transaction Markup Language (RTML) as protocol for downloading "Robot Content."
- RTML is described in XML and based on SOAP

WG reports and roadmap discussion (Kotoku).

Robotics services WG report (Lemaire).
- Goal of robotics
- Presented roadmap; 2 RFP on the horizon
  + localization
  + direction
- mission statement was accepted with no dissent.

  + RTI made motion to accept charter
  + Prof. Tanaka seconded
- WG co-chairs elected (Chi and Lemaire) w/ no dissent
  + AIST made motion to elect candidates
  + Systronix seconded
  + Shibara IT proposed white ballot.
  + motion passed

Robotics profile WG report (Boyes)
- presented mission statement
- presented roadmap
- mission statement was accepted w/ no dissent
  + Jara made motion
  + Shibura IT seconded
  + ETRI proposed white ballot
- WG co-chairs elected (Boyes and Lee) w/ no dissent
  + AIST made motion
  + JARA seconded
  + ETRI proposed white ballot

Infrastructure WG report (Ando)
- presented mission statement
- mission statement was accepted w/ no dissent
  + RTI made motion
  + AIST seconded motion
  + Shibura IT seconded
- WG co-chairs elected (Warren, Ando, Kim) w/o dissent
  + AIST made motion
  + NEDO seconded motion
  + Shibura IT proposed white ballot

Contact reports
- ORiN Forum (Mizukawa).
- Korea activities (Chung)
  + KIRSF
  + URC Technology Cooperation Forum Workshop
  + Robot standardization workshop in Jun 16, 2006 on Jeju Island, Korea.

Meeting adjourned at 12:06 pm.

Prepared and submitted by Hung Pham and Yun-Koo Chung.

# Roadmap for Robotics Activities

robotics/2006-06-05

| Item | Status | St. Louis<br>Apr-2006 | Boston<br>Jun-2006 | Anaheim<br>Sep-2006 | DC<br>Dec-2006 | San Diego<br>Mar-2007 | TBA<br>Jun-2007 | Jacksonville<br>Sep-2007 |
|---|---|---|---|---|---|---|---|---|
| **Robot Technology Components RFP (SDO model for robotics domain)** | In Process | Pre-review | Revised Submittion | adoption | | | | |
| SDO model for xxx Domain | Planned | | | discussion | draft RFP | RFP | | |
| Charter on Robotics WG in SDO | done Oct-2004 | | | | | | | |
| **Flyer of Robotics-DTF [Publicity Sub-Committee]** | In Process | discussion | review 1st Draft | issue ver.1.0 | | | | |
| **Localization Service RFP [Services WG]** | In Process | discussion | discussion | draft RFP | review RFP | RFP | | Initial Submittion |
| **User Identification RFP [Services WG]** | Planned | | | discussion | draft RFP | review RFP | RFP | |
| **Programmers API: Typical device abstract interfaces and hierarchies RFP [Profile WG]** | In Process | | Topic discussion | draft RFP | review RFP | RFP | | Initial Submittion |
| **Hardware-level Resources: define resource profiles RFP [Profile WG]** | In Process | | Topic discussion | draft RFP | review RFP | RFP | | Initial Submittion |
| **Deplyment and Configuration RFP [Infrastructure WG]** | In Process | | Outline discussion | rough draft | draft RFP | review RFP | RFP | |
| etc… | Future | | | | | | | |
| Robotic Systems RFI [Robotics: Initial Survey] | done Apr-2006 | Response Presentation | | | | | | |
| Charter on WGs [Service, Profile, Infrastructure] | done Apr-2006 | issued | | | | | | |
| Charter on Robotics TF | done Dec-2005 | | | | | | | |
| Charter on Robotics SIG | done Feb-2005 | | | | | | | |
| Robotics Information Day [Technology Showcase] | done Jan-2005 | | | | | | | |

# RUPI(Robot Unified Platform Initiative) Application Component for URC Intelligence Robot

2006. 06. 26.

Soo-Young Chi Ph.D

Human Robot Interaction Research Team

한국전자통신연구원
Electronics and Telecommunications Research Institute

robotics/2006-06-06

# Contents

I RUPI Definition

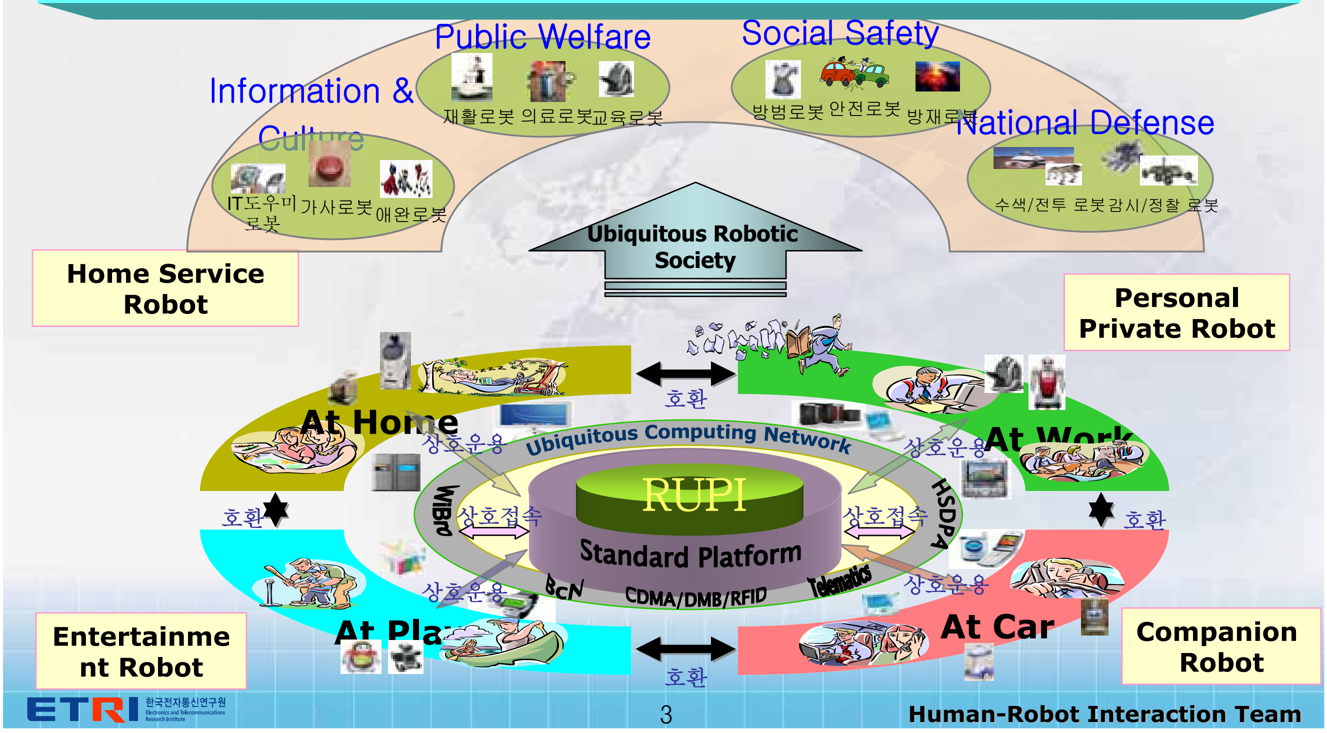
II RUPI Contents

III RUPI Application Component

# RUPI DEFINITION

Robot Unified Platform Initiative

RUPI is a S/W Standard Platform and Standard Specification for Intelligence Robot with Inter-Exchange Inter-Operability and Inter-Connection among , Many Kind of Robot S/W Platform, network and De



Public Welfare

재활로봇 의료로봇교육로봇

Social Safety

방범로봇 안전로봇 방재로봇

Information & Culture

IT도우미 로봇 가사로봇 애완로봇

National Defense

수색/전투 로봇감시/정찰 로봇

Ubiquitous Robotic Society

Home Service Robot

Personal Private Robot

Entertainment Robot

Companion Robot

At Home

At Work

At Play

At Car

호환

상호운용

상호접속

Ubiquitous Computing Network

RUPI

Standard Platform

WiBro

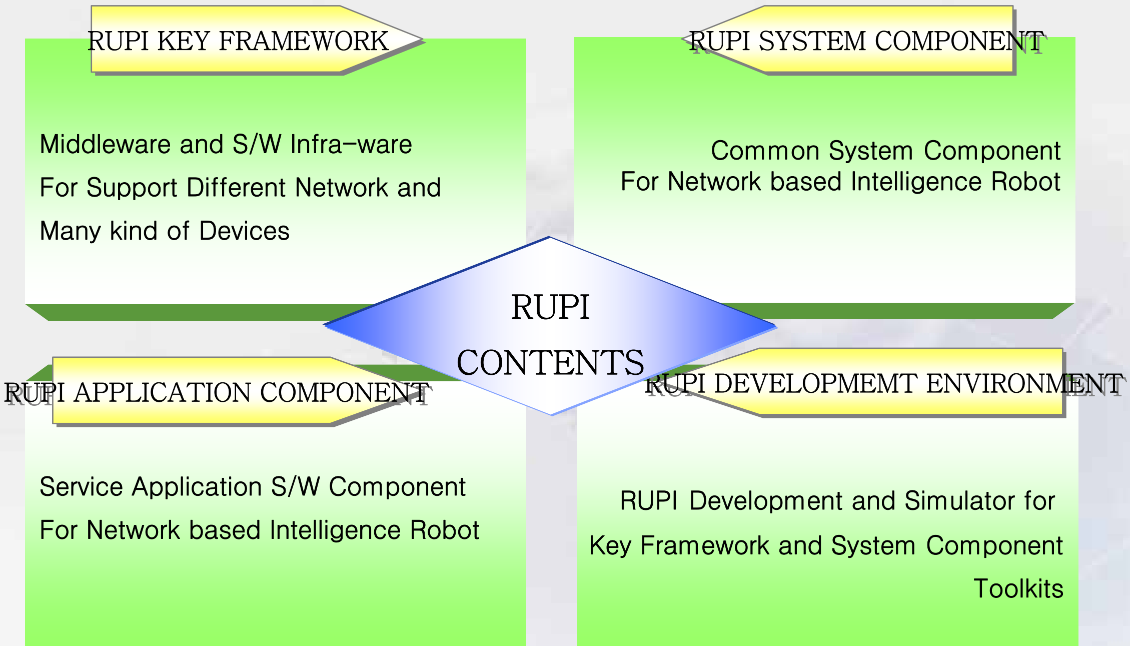
HSDPA

BcN

CDMA/DMB/RFID

Telematics

# RUPI CONTENTS



RUPI CONTENTS

RUPI KEY FRAMEWORK

Middleware and S/W Infra-ware
For Support Different Network and
Many kind of Devices

RUPI SYSTEM COMPONENT

Common System Component
For Network based Intelligence Robot

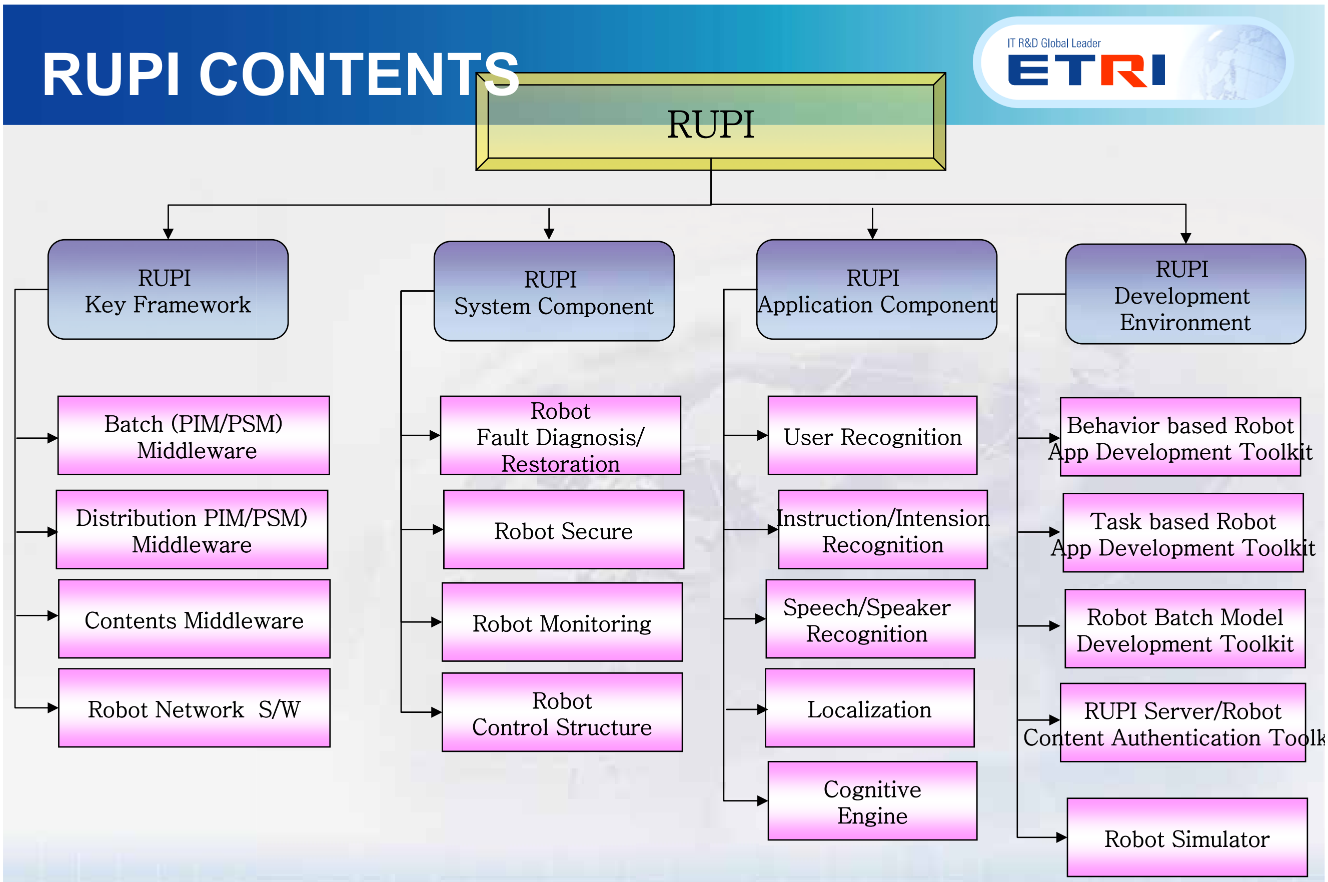
RUPI APPLICATION COMPONENT

Service Application S/W Component
For Network based Intelligence Robot

RUPI DEVELOPMEMT ENVIRONMENT

RUPI Development and Simulator for
Key Framework and System Component
Toolkits

# RUPI CONTENTS



RUPI

RUPI Key Framework
- Batch (PIM/PSM) Middleware
- Distribution PIM/PSM) Middleware
- Contents Middleware
- Robot Network S/W

RUPI System Component
- Robot Fault Diagnosis/ Restoration
- Robot Secure
- Robot Monitoring
- Robot Control Structure

RUPI Application Component
- User Recognition
- Instruction/Intension Recognition
- Speech/Speaker Recognition
- Localization
- Cognitive Engine

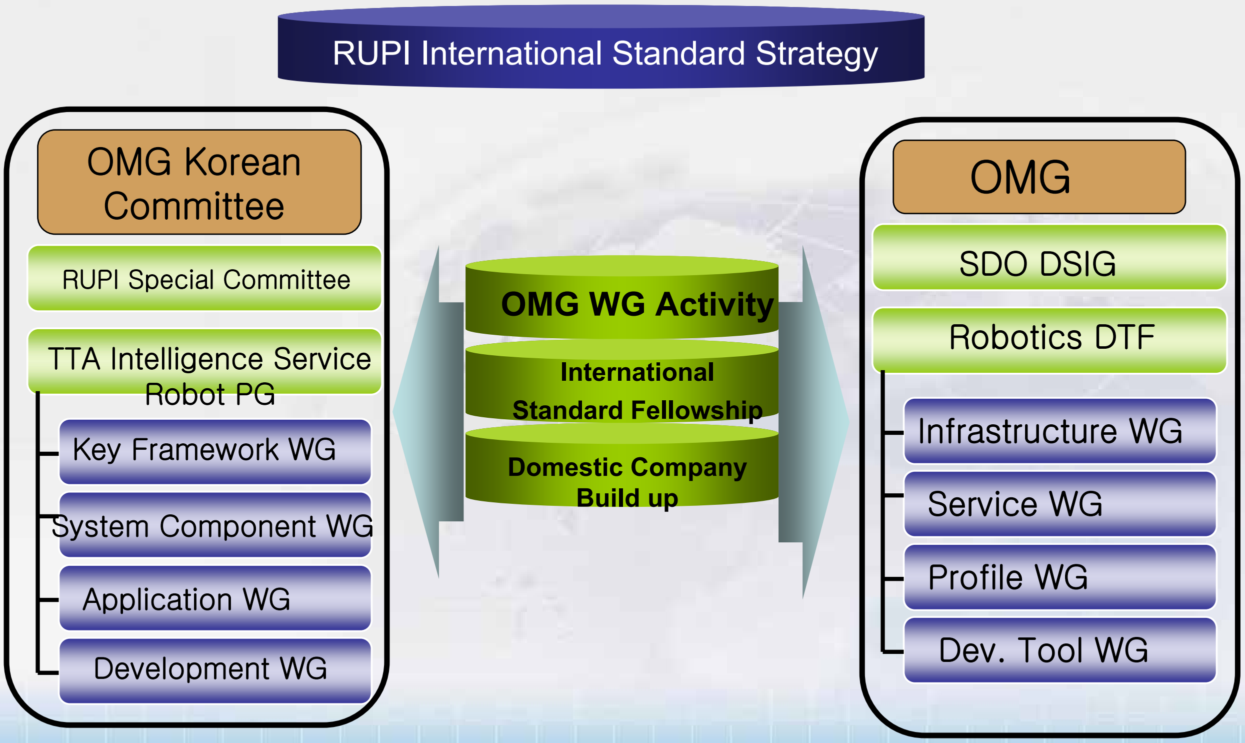
RUPI Development Environment
- Behavior based Robot App Development Toolkit
- Task based Robot App Development Toolkit
- Robot Batch Model Development Toolkit
- RUPI Server/Robot Content Authentication Toolk
- Robot Simulator

# RUPI APPLICATION COMPONENT

## RUPI Application Component

- Development of Application S/W Component Network based Intelligency Robot for Services
- Development of Application Component S/W Standard and Platform

# RUPI International Standard Strategy



RUPI International Standard Strategy

OMG Korean Committee

RUPI Special Committee

TTA Intelligence Service Robot PG

Key Framework WG

System Component WG

Application WG

Development WG

OMG WG Activity

International Standard Fellowship

Domestic Company Build up

OMG

SDO DSIG

Robotics DTF

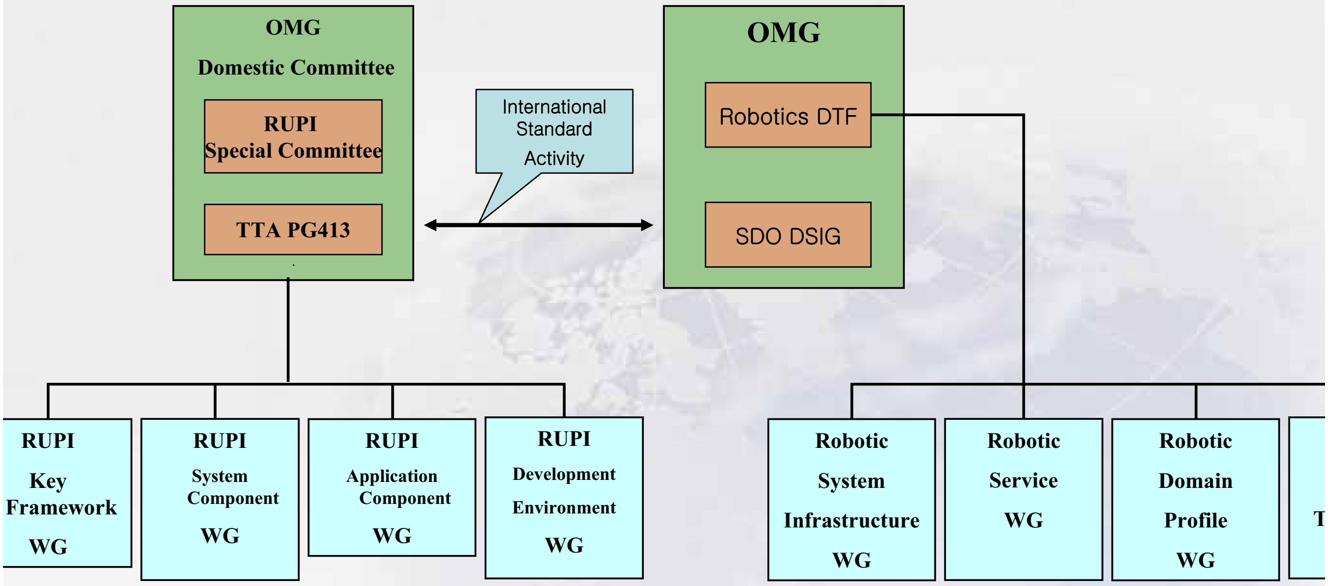
Infrastructure WG

Service WG

Profile WG

Dev. Tool WG

# RUPI International Standard Strategy



OMG

Domestic Committee

RUPI Special Committee

TTA PG413

International Standard Activity

OMG

Robotics DTF

SDO DSIG

RUPI Key Framework WG

RUPI System Component WG

RUPI Application Component WG

RUPI Development Environment WG

Robotic System Infrastructure WG

Robotic Service WG

Robotic Domain Profile WG

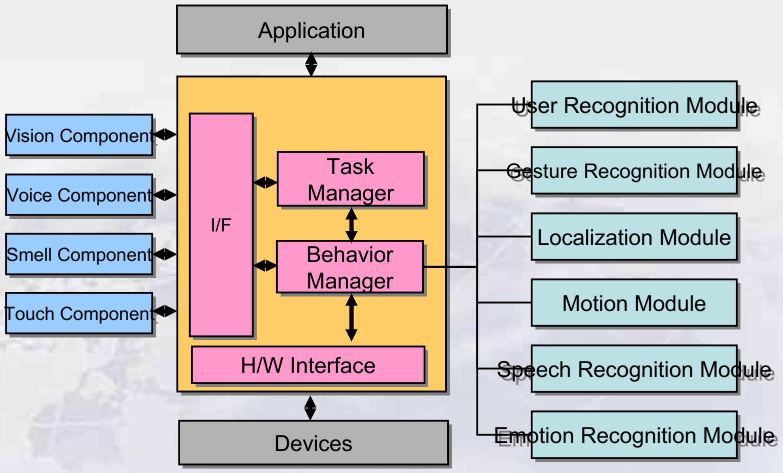
T

# RUPI APPLICATION COMPONENT DIAGRAM

❑Component Interface

❑Task manager
  - Planning and Management of Many Kind of Behavior Module with Application Service

❑Behavior manager



Application

Vision Component

Voice Component

Smell Component

Touch Component

I/F

Task Manager

Behavior Manager

H/W Interface

Devices

User Recognition Module

Gesture Recognition Module

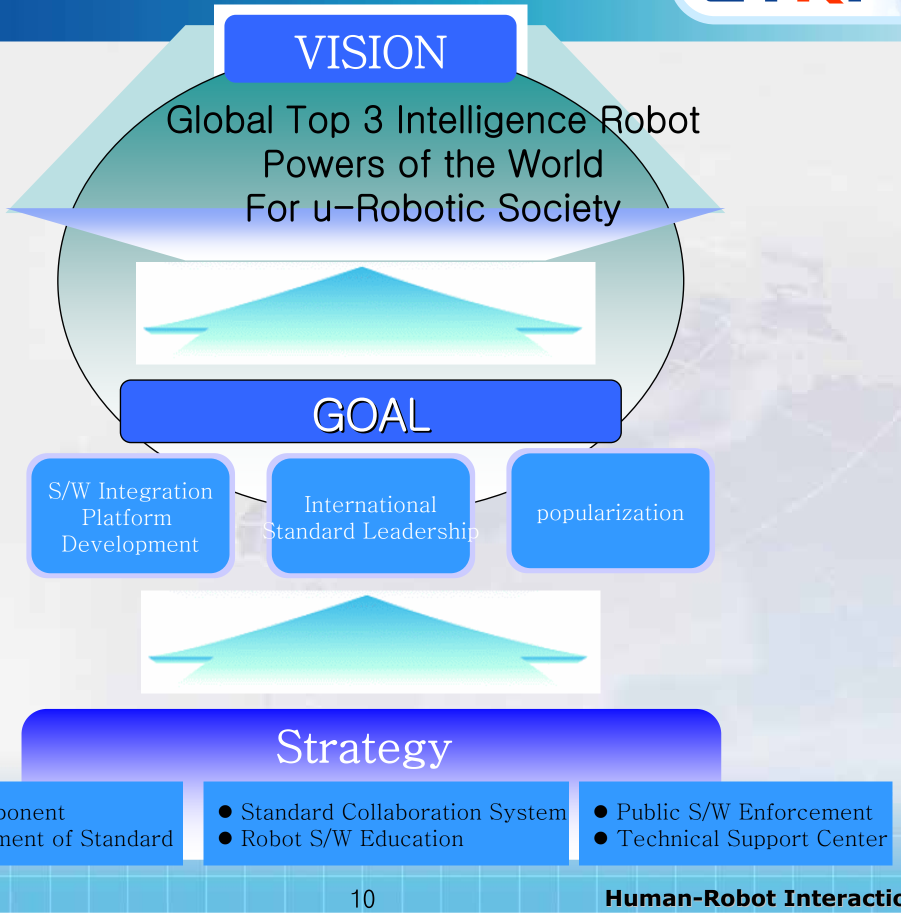
Localization Module

Motion Module

Speech Recognition Module

Emotion Recognition Module

# OUR ROBOT VISION



VISION

Global Top 3 Intelligence Robot Powers of the World For u-Robotic Society

GOAL

S/W Integration Platform Development

International Standard Leadership

popularization

Strategy

- S/W Component
- Establishment of Standard

- Standard Collaboration System
- Robot S/W Education

- Public S/W Enforcement
- Technical Support Center

# Thank You!

robotics/2006-06-08

# Profile WG Meeting

(OMG Boston Meeting)
2006.06.26 (13:00~15:00)

**Seung-Ik Lee, co-chair of Profile WG**

## Topics

- **Review of previous meeting**
- **Discussion on this meeting**
  - Typical devices
  - Top-down or bottom-up
  - Scope
  - Nomenclature and classification
  - Level of granulites
  - Size of specification
  - Integration with other existing standards
- **Roadmap discussion**

# St. Louis Meeting (2006.04.26)

- **Agreed to use mailing list to carry on significant progress between meetings**
- **Chartering**
  - Mission Statement and Road Map

# Perspectives

# Application programmer's view

## Overview

- Application programmers should have an easy-to-use and abstract access to physical resources (like OSI layers 5-7)
- This what we typically think of as the “API”

# Mission Statement

- **Define scope and model of API**
  - devices used in any robotics field
  - starting with office or home domains
- **Define typical devices**
- **Device hierarchies (like class hierarchies)**
- **Define interfaces & Data structures**
  - Consider standards such as JAUS
- **Device Profiles**
  - Enumeration of available resources
  - Resource configuration and capabilities

# Physical transducer and resource view

# Overview

- Details of addressing and hardware interfaces are handled here (like OSI layers 1-4)
- This is what developers of hardware (transducers, communications, and other such physical devices) will use to create an abstracted and easy-to-use interface for application programmers.

# Mission Statement

# Apply relevant standards (IEEE, etc) to robotics

- **Smart sensors IEEE-1451**
- **Precision networked clock IEEE-1588**
- **Arrange presentations on the above at OMG meetings**
  - 1451 in Anaheim?
  - 1588 in Wash DC? (near NIST)

# I/O point tagging, provides

- **Enumeration of available resources**
- **Storage of configuration and capabilities**
  - on the actual device or as close to it as possible

# Boston Meeting (2006.06)

# Issues to be Discussed

## Proposal of typical devices

- Robot itself ( we need to, say, turn it off or on)
- Differential wheel type (more generally speaking, movement devices)
- Head (pan & tilt devices)
- Camera (for getting images and others)
- Proximity sensors ( such as IRs and sonars)
- Bumpers ( collision detection)
- Battery Monitor
- Speaker
- Mic
- RFID

## Proposal of typical devices (2)

- Gyros
- Accelerometers
- Odometry
- Wireless sensor device (e.g., Zigbee-enabled sensing devices)
- Display

# Top-down v.s. bottom-up approach

- **Top-down**
  - First define what a "Device" in general is in an abstract way, what makes a device component (in contrast to a pure software component). Then, derive the definition to more specific devices )
- **Bottom-Up**
  - First define ad-hoc interfaces of devices and hope we will find commonness

# Scope

- **What kind of devices are candidates for standardization**
- **Application area or domains**
  - Not necessary
  - Should we confie the scope to only service robotics ?

# Define the nomenclature and classification

- **a Bumper could be considered as a proximity sensor (distance = 0)**
- **Can a Battery be considered as a device**
- **A Pan-Tilt Camera is a Head**

# Level of granulity

- **Named after the kind of data they treat**
  - Accelerometers
  - Proximity sensors
- **Named after technology**
  - RFID

# One big specification?

- How we manage extensions?
- Definition of all these "devices" will be bundled into one big specification, into several unitary specification

# Integration with other existing standards

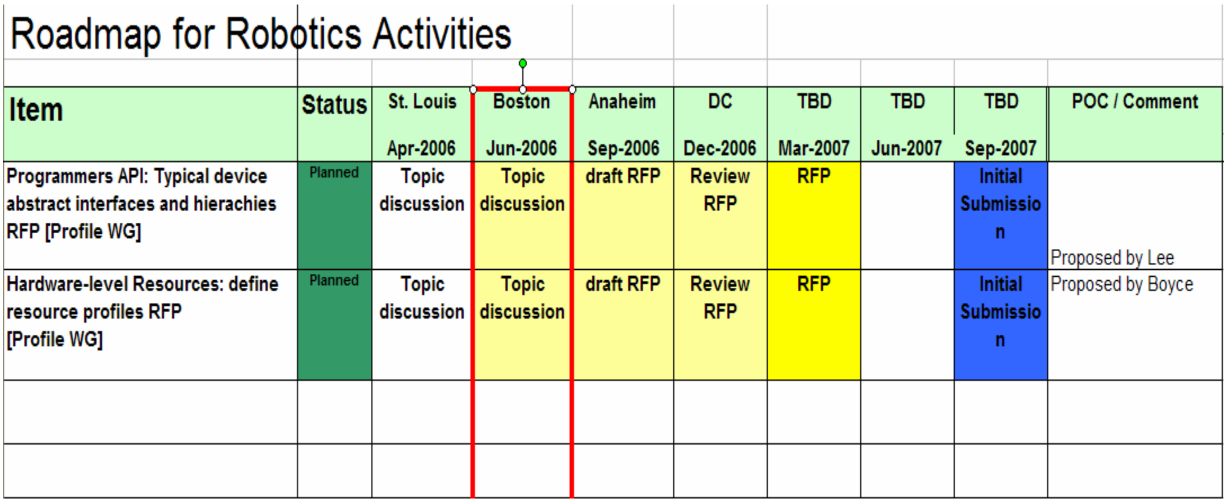
- IEEE-1451
- JAUS

# Naming of our WG

- Robotic Device and data profiles WG !!!!

# Roadmap discussion

- Any need to update or change the roadmap? → no need!!



Roadmap for Robotics Activities

| Item | Status | St. Louis<br>Apr-2006 | Boston<br>Jun-2006 | Anaheim<br>Sep-2006 | DC<br>Dec-2006 | TBD<br>Mar-2007 | TBD<br>Jun-2007 | TBD<br>Sep-2007 | POC / Comment |
|---|---|---|---|---|---|---|---|---|---|
| Programmers API: Typical device abstract interfaces and hierachies RFP [Profile WG] | Planned | Topic discussion | Topic discussion | draft RFP | Review RFP | RFP | | Initial Submission | Proposed by Lee |
| Hardware-level Resources: define resource profiles RFP [Profile WG] | Planned | Topic discussion | Topic discussion | draft RFP | Review RFP | RFP | | Initial Submission | Proposed by Boyce |
| | | | | | | | | | |
| | | | | | | | | | |

## Topics to be included in the RFP

- **How do you define devices?**
  - The difference between devices and services?
- **Several aspect can be considered to define a device :**
  - Data it deals with (input / output)
  - Physical Characteristics
  - Physical action on environment
  - -> How to combine all these aspects
- **For each aspect :**
- **Provide device classification system related to a given aspect**
- **How to manage composite devices? Virtual Devices ? From a given aspect point of view**
- **How can definition apply to any platform?**
- **How to enumerate devices present in a system from a given aspect point of view?**
  - What are requirements for enumeration (what does a query look like)
  - (list up all devices present in a robot)
- **How to manage device configuration and introspection**
- **How to integrate existing standards**
- **Discussion : As a proof of concept, provide definition main devices fond in home/service robot (give definition)**

## Topics to be included in the RFP

- **How do you define devices?**
  - The difference between devices and services?
- **Several aspect can be considered to define a device :**
  - Data it deals with (input / output)
  - Physical Characteristics
  - Physical action on environment
  - -> How to combine all these aspects
- **Provide device classification system**
- **How to manage composite devices? Virtual Devices ?**
- **How can device definition apply to any platform?**
- **How to enumerate devices present in a system?**
  - What are requirements for enumeration (
  - (list up all devices present in a robot)
- **How to manage device configuration and introspection**
- **How to integrate existing standards**
- **Discussion : As a proof of concept, provide definition main devices fond in home/service robot (give definition)**

# Robotics DTF
# Steering Committee Meeting

June 26, 2006

Boston, MA, USA

HYATT Harborside

# Agenda

- Agenda Review
- Minutes
- Publicity
- Roadmap Discussion
- Next meeting Schedule

# Review Agenda

**Mon(Jun.26): Aquitana C, 1st FL**

**Service WG, Profile WG, Steering Committee**
**Joint MARS-PTF(RTC RFP submission)**

**Tue(Jun.27): Michelangelo B, 2nd FL**

**Service WG, Infrastructure WG, Presentations**

**Wed(Jun.28): Michelangelo B, 2nd FL**

**TF Plenary (Robobusiness, Reports)**

**Thu(Jun.29): Constellation 316, 3rd FL**

**Special Talk, Planning Session, WG activity**

**Joint Meeting with MARS/RTESS**
**Monday, June 26, 2006**
**10:00-11:00 (Michelangelo A, 2nd FL)**

# Minutes

- Process:
  - Make a draft with in 5days
  - Send the initial draft to robotics-chairs@omg.org
  - Post the draft to the OMG server within a week
  - Make an announcement to robotics@omg.org
  - Send comments to robotics@omg.org
  - Approve the revised minutes at the Next meeting

- Volunteers for this Boston Meeting
  - Hung Pham (RTI)
  - Olivier Lemaire (AIST)

**We have to post our meeting minutes within a week!**

# Publicity Activities

- 4 page fly sheet
  Draft of Abheek@ADA Software
  **Abheek@ADA Soft,**
  **Olivier@AIST,**
  **Chung@ETRI,**
  **Yokomachi@NEDO**

**Action:**

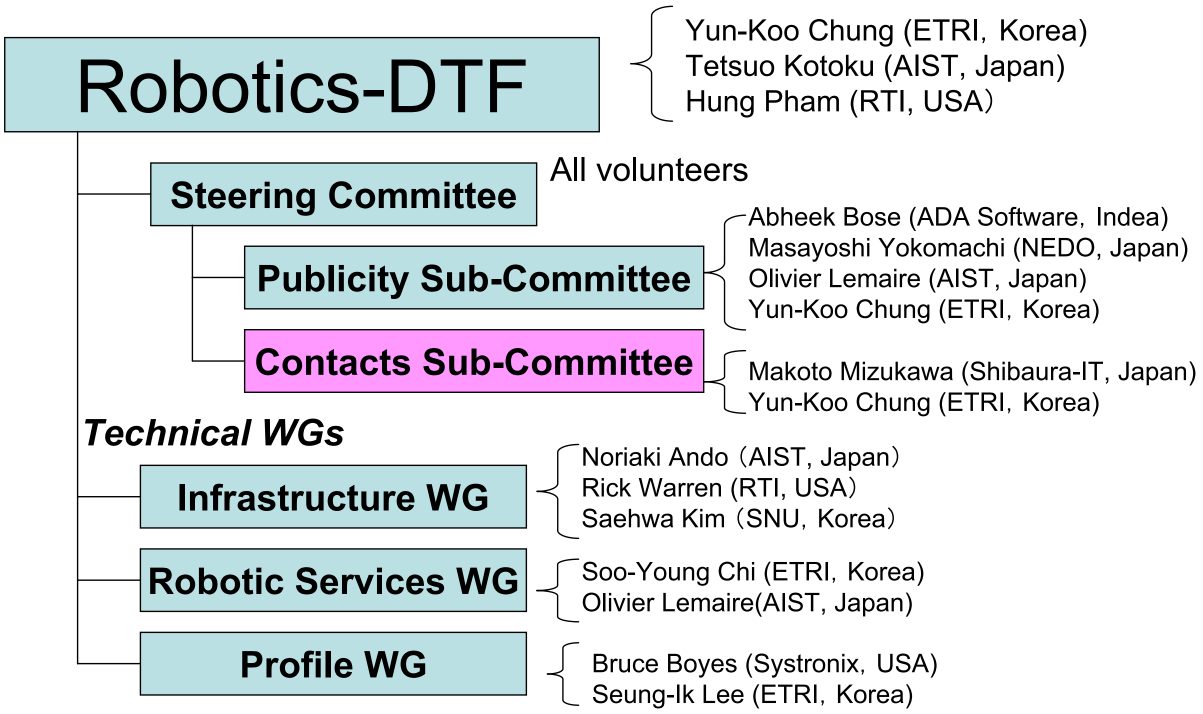
Send each organization logo to Abheek.

4 page fly sheet will be authorized in Boston

# Publicity Activities

- Korea-Japan RSW2006
  Friday, June 16, 2006, Jeju Island, Korea
  Chung@ETRI
- RoboBusiness2006
  June 20-21, 2006, Pittsburgh, PA, USA
  http://www.robobusiness2006.com/
  Jon Sigel and Bruce@Systronix
- IROS2006 Workshop
  October 9-15, Beijing, China
  http://www.iros2006.org/
  Kotoku@AIST, Chung@ETRI, Mizukawa@Sibaura-IT
- SICE-ICASE International Joint Conference
  October 18-21, Pusan, Korea
  http://sice-iccas.org/
  Mizukawa@Sibaura-IT

# Organization



Robotics-DTF
- Yun-Koo Chung (ETRI, Korea)
- Tetsuo Kotoku (AIST, Japan)
- Hung Pham (RTI, USA)

Steering Committee — All volunteers

Publicity Sub-Committee
- Abheek Bose (ADA Software, Indea)
- Masayoshi Yokomachi (NEDO, Japan)
- Olivier Lemaire (AIST, Japan)
- Yun-Koo Chung (ETRI, Korea)

Contacts Sub-Committee
- Makoto Mizukawa (Shibaura-IT, Japan)
- Yun-Koo Chung (ETRI, Korea)

*Technical WGs*

Infrastructure WG
- Noriaki Ando（AIST, Japan）
- Rick Warren (RTI, USA)
- Saehwa Kim（SNU, Korea）

Robotic Services WG
- Soo-Young Chi (ETRI, Korea)
- Olivier Lemaire(AIST, Japan)

Profile WG
- Bruce Boyes (Systronix, USA)
- Seung-Ik Lee (ETRI, Korea)

# Next Meeting Agenda

Sep. 26-30, 2006 (Anaheim, CA, USA)

**Monday :**

**RTCs RFP revised submission review** [MARS]
**Steering Committee**

**Monday-Tuesday, Thursday :**

**WG activities**

**Wednesday :**

**Robotics-DTF Plenary Meeting**
- Guest Presentation(s)
- WG reports & Roadmap discussion
- Contact reports
- DTC report - Draft

# Next Meeting Agenda

- Make a rough agenda at the previous meeting. (rough sketch)
- Agenda planning session on Thursday morning
- Post a preliminary agenda 4weeks before the meeting.
- Print a final agenda at the meeting site.

**We have to post our preliminary Agenda a month before!**

# OMG Robotics DTF

## Description Document

OMG Technical Meeting
Tampa, Florida
Monday, February 13, 2006

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

**robotics/2006-06-10**

# Motivation & Goals

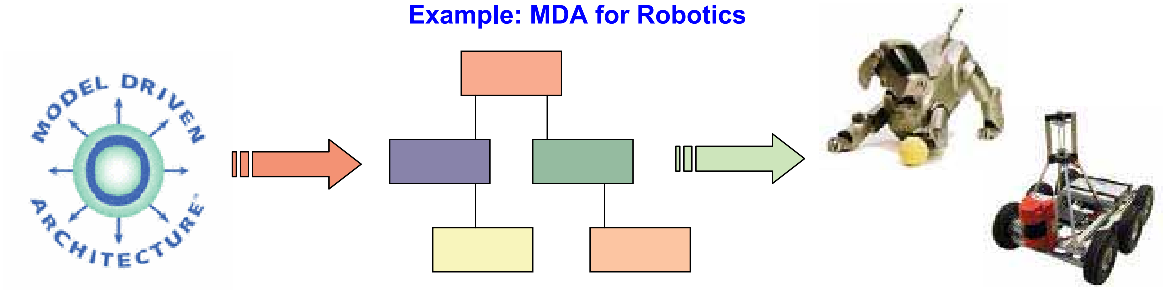
- High diversity of Robotic Systems but….
  - Specialized for accomplishing tasks set by the designer/manufacturer
  - Systems, subsystems are not interoperable with one another
- Engineering Industry – e.g. Automobiles
  - Excellent example of standardized components
  - Great deal of system interoperability and re-use

**The primary reason for successful interoperability and integration is Standards!**

# Proposed Methods

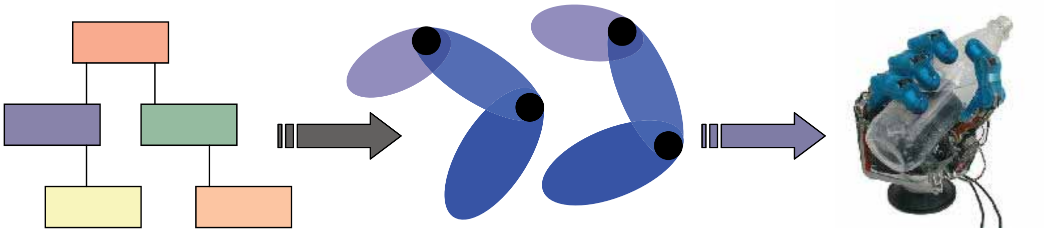
**Adapt & extend existing OMG technologies that apply to robot systems**

Example: MDA for Robotics



**Implement the adapted technologies in both hardware & software**

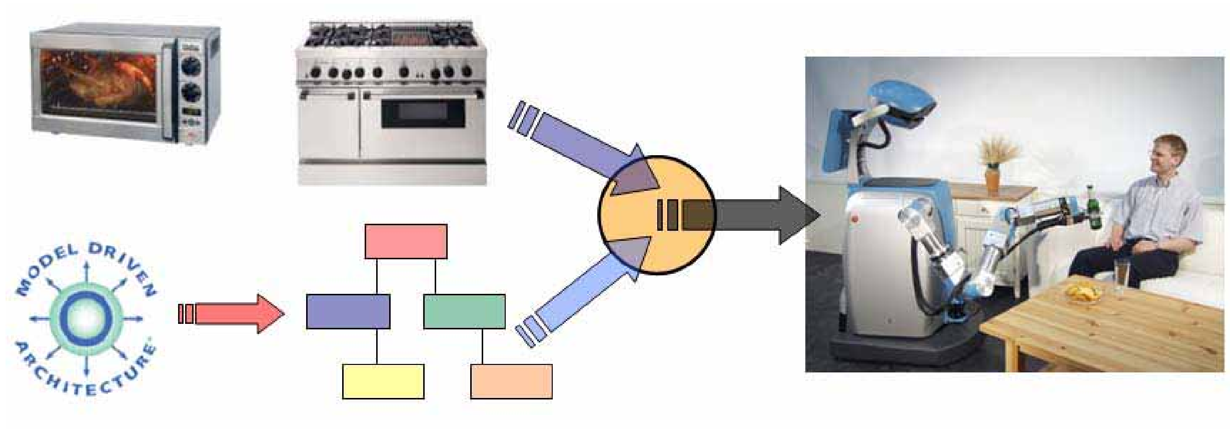
Example: Model driven development of "Finger" components & its controller for a robot hand



# Proposed Methods

**Collaborate with other organizations implementing standards**

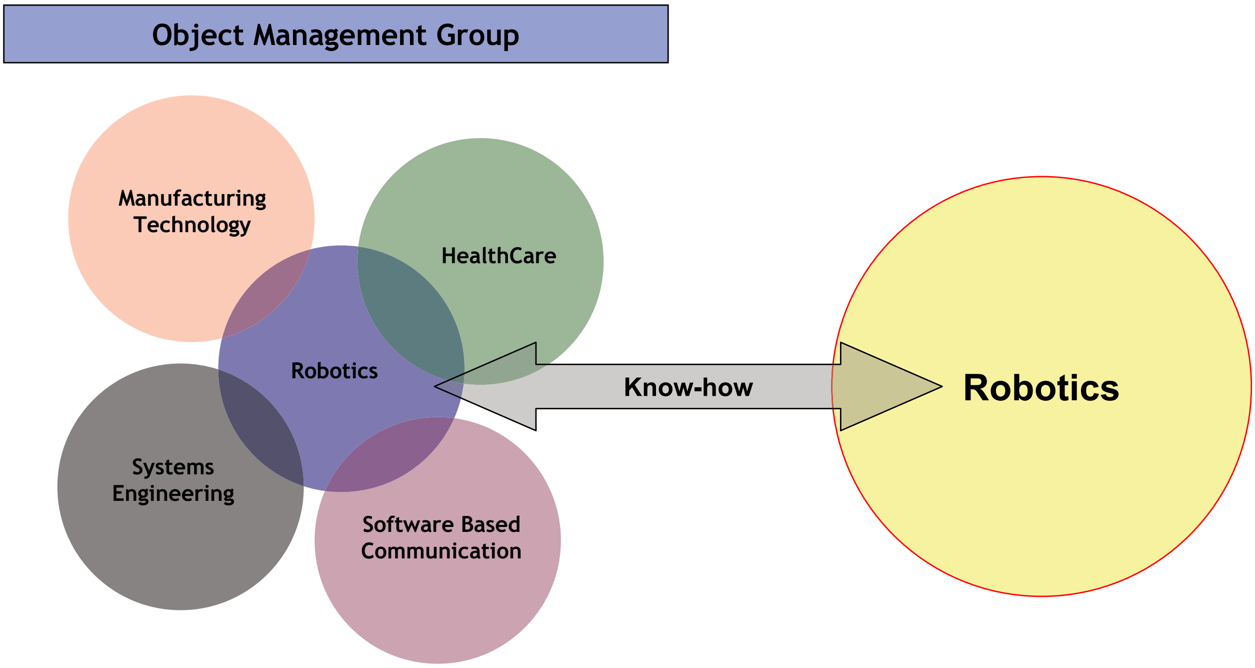
Example: Home Appliances + Robot Technologies = Housekeeping Robot



# Proposed Methods

**Task Distribution for Overlapping Technologies** Coordination with OMG Task Forces and AB

**Mutual Understanding and Information Exchange** between OMG and Robotics Communities Worldwide



Object Management Group

Manufacturing Technology

HealthCare

Robotics

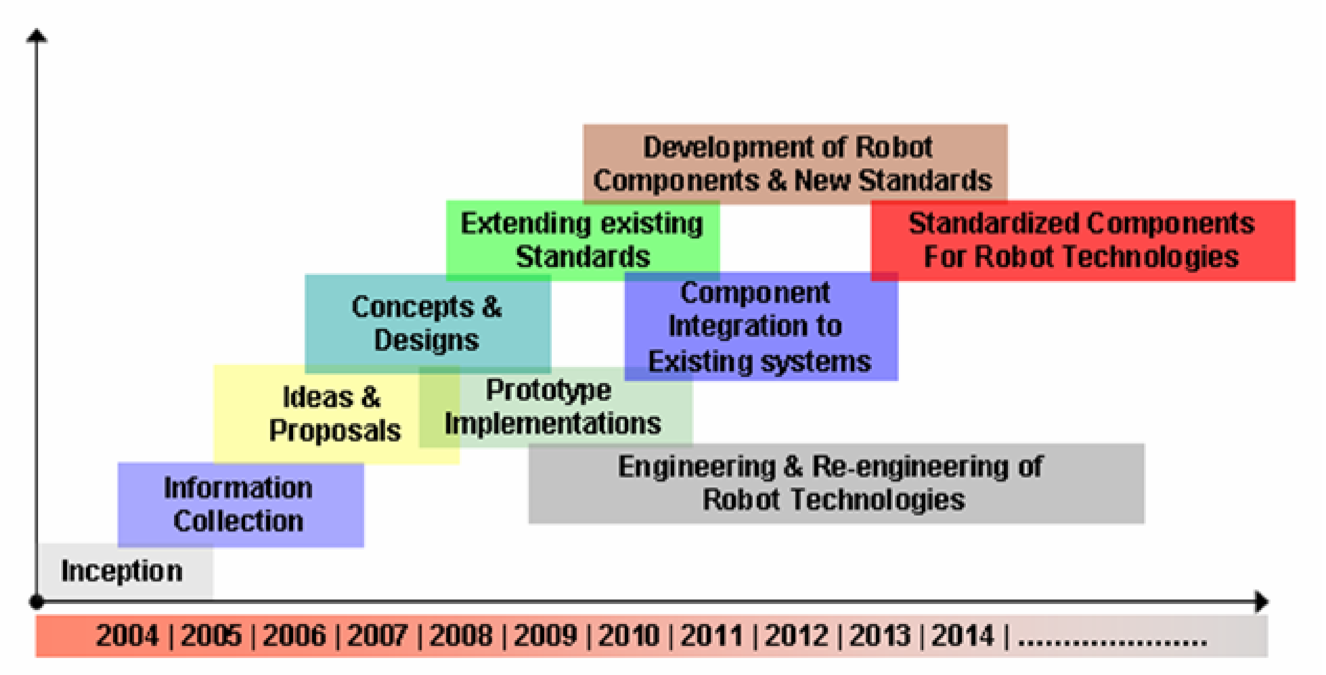
Systems Engineering

Software Based Communication

Know-how

**Robotics**

# Activity Roadmap



Development of Robot Components & New Standards

Extending existing Standards

Standardized Components For Robot Technologies

Concepts & Designs

Component Integration to Existing systems

Ideas & Proposals

Prototype Implementations

Engineering & Re-engineering of Robot Technologies

Information Collection

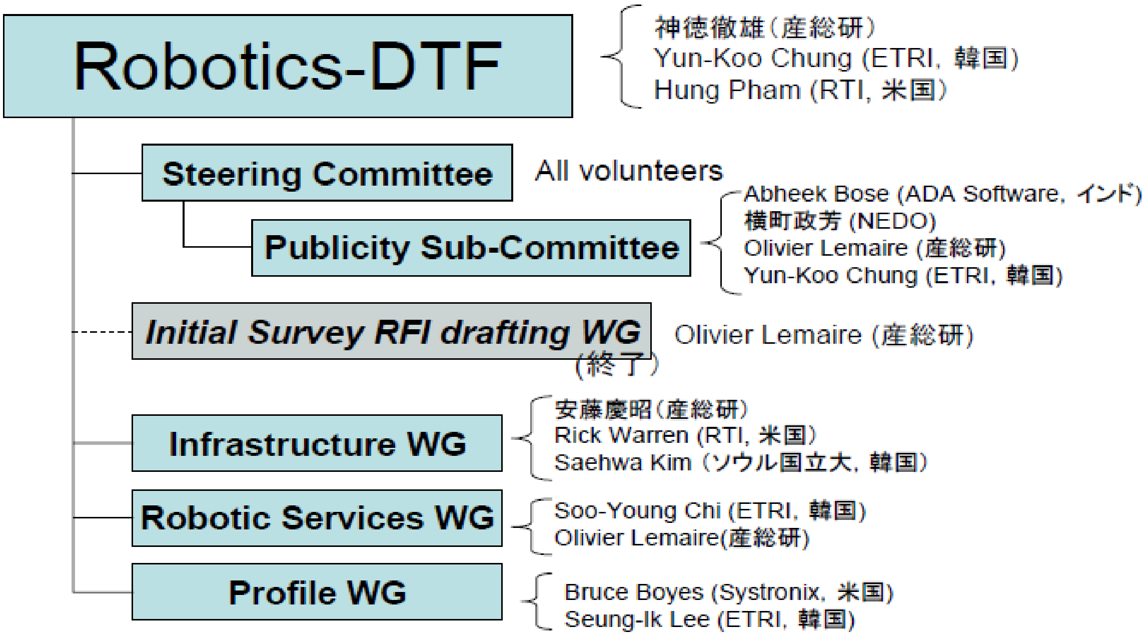
Inception

2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | ......................

# Domain Members

AIST
ETRI
www.etri.re.kr
RTI
JAUS
evolution robotics.
NIST National Institute of Standards and Technology
NEDO 技術開発機構
Schlumberger
mBus Intelligent Control
FUJITSU
芝浦工業大学 SHIBAURA INSTITUTE OF TECHNOLOGY
Seoul National University RTOS Lab
Georgia Tech
THE UNIVERSITY OF UTAH
ADA Robotics
SYSTRONIX Embedded Java Spoken Here
TOSHIBA

# Robotics DTF Organization



Robotics-DTF
- 神徳徹雄（産総研）
- Yun-Koo Chung (ETRI, 韓国)
- Hung Pham (RTI, 米国）

Steering Committee
- All volunteers

Publicity Sub-Committee
- Abheek Bose (ADA Software, インド)
- 横町政芳 (NEDO)
- Olivier Lemaire (産総研)
- Yun-Koo Chung (ETRI, 韓国)

*Initial Survey RFI drafting WG* （終了）
- Olivier Lemaire (産総研)

Infrastructure WG
- 安藤慶昭（産総研）
- Rick Warren (RTI, 米国)
- Saehwa Kim（ソウル国立大, 韓国）

Robotic Services WG
- Soo-Young Chi (ETRI, 韓国)
- Olivier Lemaire(産総研)

Profile WG
- Bruce Boyes (Systronix, 米国)
- Seung-Ik Lee (ETRI, 韓国)

# Proposed Charter for Robotics DTF

**Mission:**
**The purpose of the Robotics activities is to foster the integration of robotics systems from modular components through the adoption of OMG standards.**
**To realize this purpose, we will:**

- **-Adapt and extend OMG technologies that apply to the specific domain of robotics systems where no current baseline specifications exist, such as MDA for Robotics.**
- **-The object technology is not solely limited to software but is extended to real objects.**
- **-This effort promotes the use of OMG technologies in various markets.**
- **-Promote mutual understanding between the robotics community and the OMG community.**
- **-Endeavor to collaborate with other organizations for standardization, such as the one for home information appliances, and make an open effort to increase interoperability in the field of robotics.**
- **-Coordinate with the appropriate OMG subgroups and the Architecture Board, for technology areas that overlap with other OMG Task Forces, to determine where the work will be accomplished.**

# Robot Infrastructure WG in Robotics DTF

**①Robot Infrastructure Chair:Rick Warren (RTI), Ando(AIST)、Sae-hwa Kim(SNU)**

**Mission:**

**-The purpose of the infrastructure WG of Robotics DTF is to standardize fundamental models, common facilities, and middleware to support the development and integration of a broad range of robotics applications**

**- This WG should collaborate with other groups within OMG.**

**Note: Common facilities: fundamental services general to wide range of robotics applications**

# Robot Service WG in Robotics DTF

**②Robot Service ＜Chair：Chi (ETRI), Lemaire (AIST)＞**

**Mission:**

- **Establish a clear definition of functional services in robotic systems**
- **Identify and categorize services commonly used in robotic application and the technologies involved**
- **Define standard interfaces that expose these technologies to robotic application developments**
- **Coordinate with other groups within the OMG Robotics Task Force to keep specification consistent**

# Robot Profile WG in Robotics DTF

**③Robot Profiles＜Chair：Bruce Boyce (Systronix), Seung Ik (ETRI)＞**
**(e-mail: omg-profile@m.aist.go.jp)**

**Mission**

**Application Programmers' View**

- **Define scope and model of API**
- **Define typical devices**
- **Device Hierarchies (like class hierarchies)**
- **Define Interfaces and Data Structure**
- **Device Profiles (Enumeration of Available Resources, Resource Configuration and Capabilities)**

**Physical Transducer and Resource View**

- **Apply relevant standards (IEEE, etc.) to robotics**
  **(IEEE 1451 smart sensor, IEEE 1588 Precision networked clock, etc.)**
- **I/O point tagging provides**
  **Enumeration of available resources**
  **Storage of Configuration and capabilities on the actual device or as close to it as possible**

# Join Us!

**OMG Account Representatives**

If your company name begins with the letter ***A - K*** or with a number, contact:

**Carol Kelly**
*Manager, Business Development*
Email: ckelly@omg.org
Phone: +1-781-444 0404 ext. 134

If your company name begins with the letter ***L - Z*** or with a number, contact:

**Susan Connolly**
*Manager, Business Development*
Email: susan@omg.org
Phone: +1-781-444 0404 ext. 133

**OMG Robotics DTF Chairs**

| Yun Koo Chung | Hung Pham | Tetsuo Kotoku |
|---|---|---|
| ETRI,<br>Korea | Real Time<br>Innovations, USA | Advanced Institute<br>of Science &<br>Technology, Japan |
| Email<br>ykchung@etri.re.kr | Email<br>hpham@rti.com | Email<br>t.kotoku@aist.go.jp |

robotics/2006-06-11

## Robotics Domain Task Force

**Perhaps some pictures of a few robots**

**(If it can indicate standards, it would be nice!)**

**A SLOGAN!! PUNCHLINE!**
- **TBD – Homework!**

**Start the logos from the 2nd page …. Possible 2 versions!**

**About Object Management Group & the Robotics DTF**

Line 1
Line 2
Line 3
Line 4
Line 5

**Focus and aims of the Robotics DTF**

Line 1
Line 2
Line 3
Line 4
Line 5

**Established Working Groups & their focus**

Line 1
Line 2
Line 3
Line 4
Line 5

**ROADMAP**

Line 1
Line 2
Line 3
Line 4
Line 5

**Feedbacks from Yokomachi-san**
- Motivation and Goals
  - o Why we need standards
- Proposed Methods
- Targets of the DTF
- Structure of the Working Groups
  - o DTF and its Sub TFs
- Mission Statement (must make it short)
  - o From the Robotics POV
- Description of the WGs
  - o Profile
  - o Infrastructure
  - o Services
  - o Tools ??
- News and Events attended
- Related Organisations
  - o ISO, IEEE, OASIS, etc - how to mention them (subtle but clear)
- Weblinks , URL
- Invitation to contribute to the standardisation drive
  - o What can we do!
- Opinion poll report summary (from Tampa)
  - o Bar Graph (Olivier)
- Restrict too much infoh

# POEM - an implementation of position estimation module

Takashi Tsubouchi, Professor

Eijiro Takeuchi, Ph. D candidate

Intelligent Robot Laboratory,

University of Tsukuba, Japan

# POEM – Position Estimation Module

## An EKF based modularization for cumulative error correction of odometry by landmark sensing

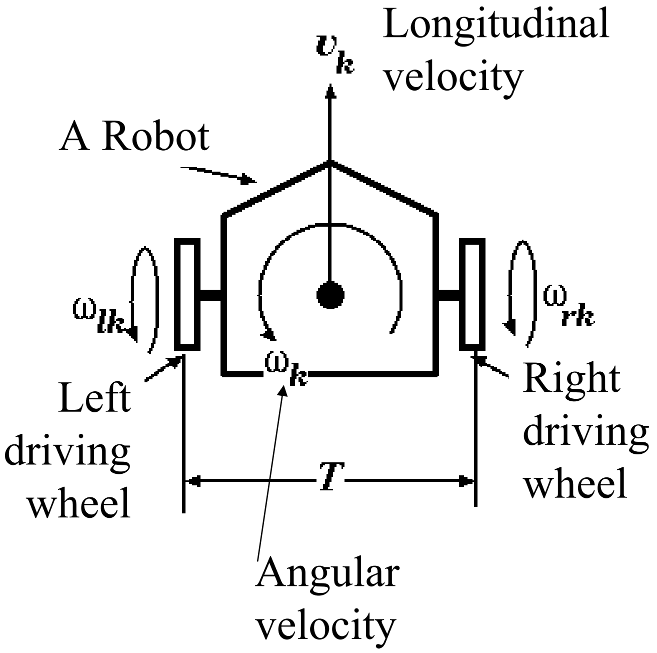
- Y. Watanabe and S. Yuta: “Estimation of Position and its Uncertainty in the Dead Reckoning System for the Wheeled Mobile Robot,” 20th ISIR pp. 205-21, 1989-10, Tokyo.
- Y. Watanabe and S. Yuta: “Position Estimation of Mobile Robots with Internal and External Sensors Using Uncertainty Evolution Technique,” 1990 IEEE International conference on Robotics and Automation pp. 2011-2016, 1990-05, Cincinnati.
- T. Nishizawa, A. Ohya and S. Yuta: “An Implementation of On-board Position Estimation for a Mobile Robot,” IEEE International Conference on Robotics and Automation (ICRA) '95, pp.395-400, 1995-05, Nagoya.
- T. Yamamoto, S. Maeyama, A. Ohya and S. Yuta: “An Implementation of Landmark-Based Position Estimation Function as an Autonomous and Distributed System for a Mobile Robot,” 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems '99 IROS, TP2 1-4, 1999-10, Korea.
- etc.

# Applications of POEM in the Lab.

- Laboratory tour guide robot
- Long distance navigation in corridor environment
- Intelligent powered handcart
- Outdoor navigation by GPS and odometry data fusion
- Outdoor navigation based on natural landmark
- etc. … So many!

# Background
## - Odometry and Position Identification -

# Odometry

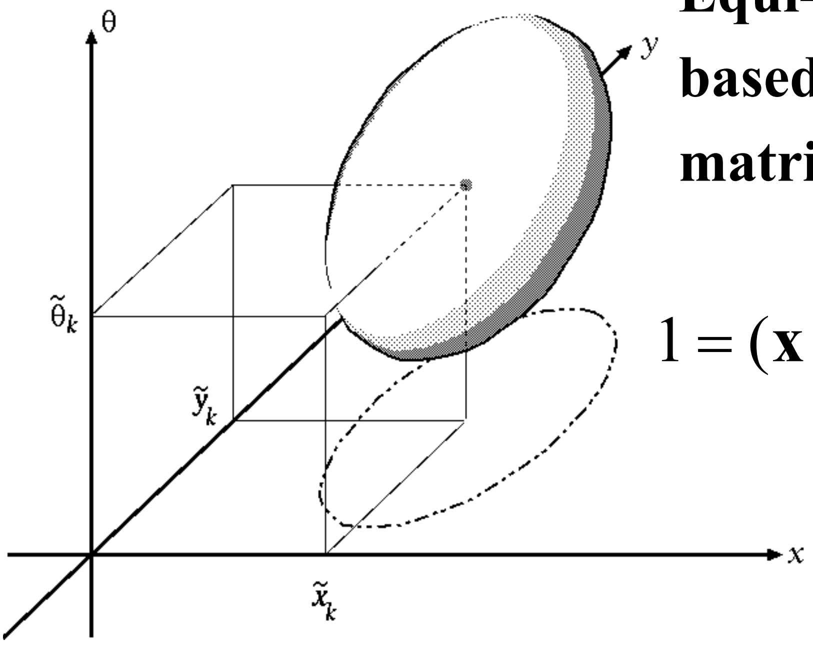


$$\theta = \int \omega_k \ dt$$

$$x = \int v_k \cos\theta \ \ dt$$

$$y = \int v_k \sin\theta \ \ dt$$

# Error Ellipsoid



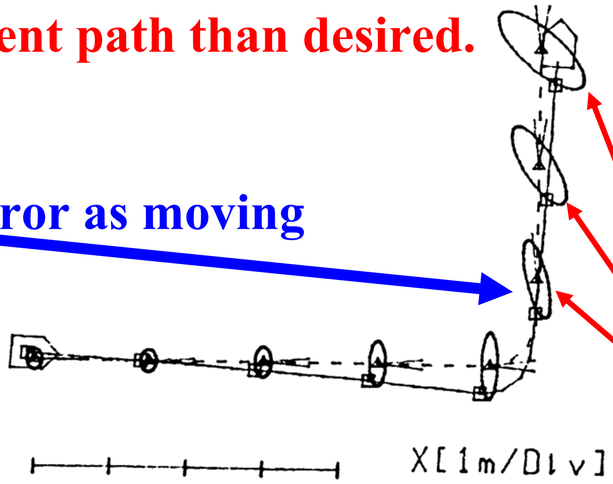
**Equi-probability surface based on covariance matrix** $\Sigma = E(\Delta \mathbf{x} \Delta \mathbf{x}^t)$

$$1 = (\mathbf{x} - \tilde{\mathbf{x}}_k)^t \mathbf{\Sigma}^{-1} (\mathbf{x} - \tilde{\mathbf{x}}_k)$$

$$\left\{ \begin{array}{l} \tilde{\mathbf{x}}_k = (\tilde{x}_k \quad \tilde{y}_k \quad \tilde{\theta}_k)^t \\ \mathbf{x} = (x \quad y \quad \theta)^t \end{array} \right\}$$

# Cumulative Error of Odometry

- Path following is based on odometory measured position and heading direction
  - (Renew every sampling time [ms order])
  - Cumulative error causes the robot running along different path than desired.



Expansion of error as moving

Error Ellipsoid

# Cumulative Error Correction

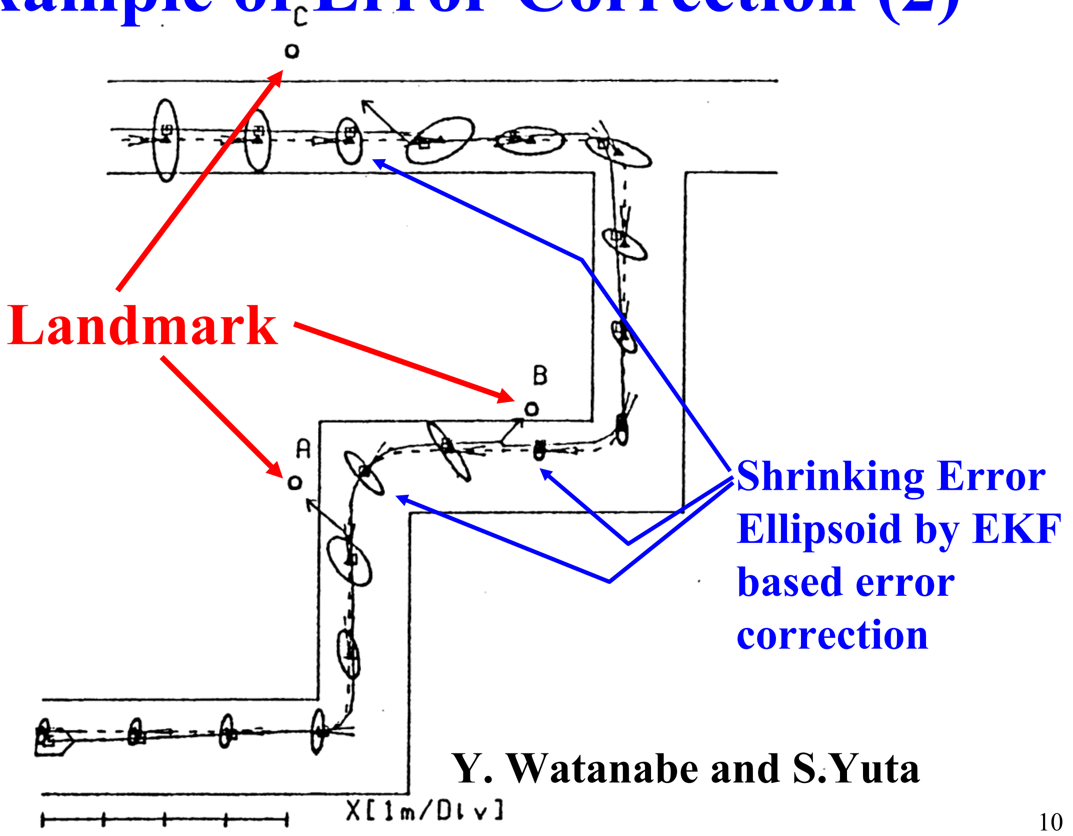
- Correct error if the robot position is identified by observation of landmark with sensors.
- Allow intermittent sensory observation.
  - e.g. When the error ellipsoid becomes large or eigen values of covariance exceed the threshold.

# Error Correction Algorithm

- Utilize EKF or Most Likelihood Estimation framework

  -- Correction of Estimating Position and Covariance Matrix

# Example of Error Correction (2)

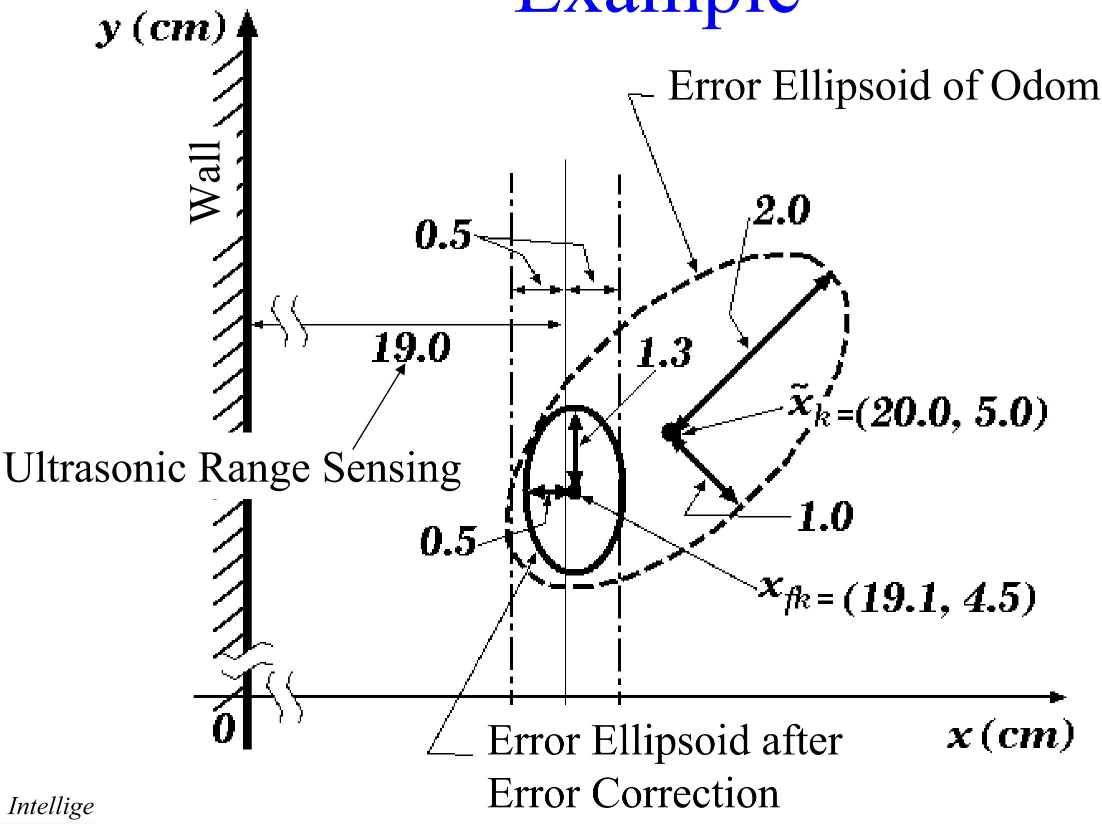


Landmark

Shrinking Error Ellipsoid by EKF based error correction

Y. Watanabe and S.Yuta

# Example

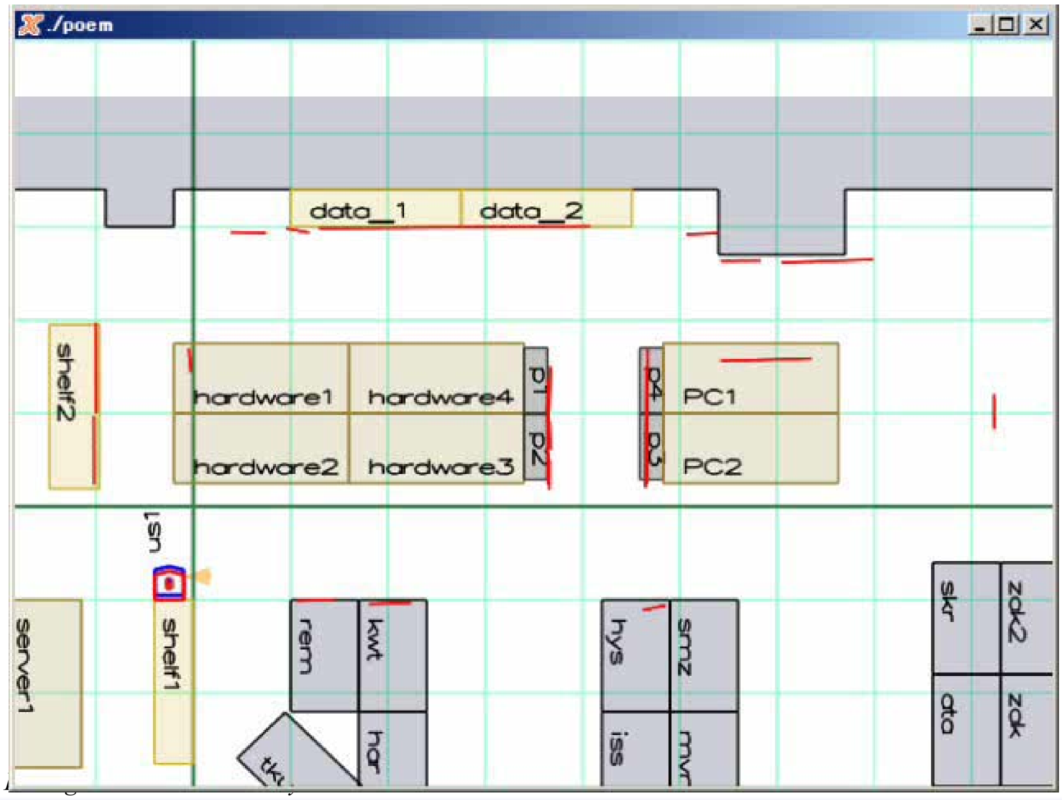


Error Ellipsoid of Odometry

Ultrasonic Range Sensing

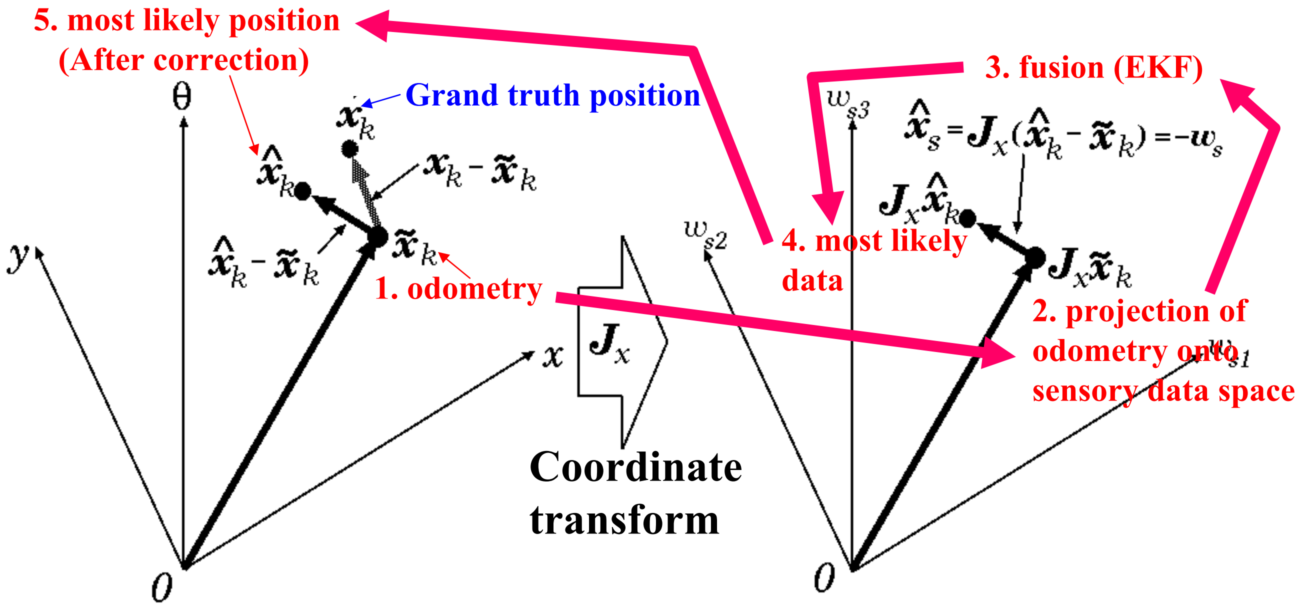
Error Ellipsoid after Error Correction

# Example of Error Correction (2)



E. Takeuchi

# Mathematical Framework Schematic View



5. most likely position (After correction)

Grand truth position

1. odometry

Coordinate transform

3. fusion (EKF)

4. most likely data

2. projection of odometry onto sensory data space

$\hat{\boldsymbol{x}}_s = \boldsymbol{J}_x(\hat{\boldsymbol{x}}_k - \tilde{\boldsymbol{x}}_k) = -\boldsymbol{w}_s$

θ
$y$
$x$
0
$\hat{\boldsymbol{x}}_k$
$\boldsymbol{x}_k$
$\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k$
$\hat{\boldsymbol{x}}_k - \tilde{\boldsymbol{x}}_k$
$\tilde{\boldsymbol{x}}_k$
$\boldsymbol{J}_x$
$w_{s3}$
$w_{s2}$
$w_{s1}$
$\boldsymbol{J}_x\hat{\boldsymbol{x}}_k$
$\boldsymbol{J}_x\tilde{\boldsymbol{x}}_k$
0

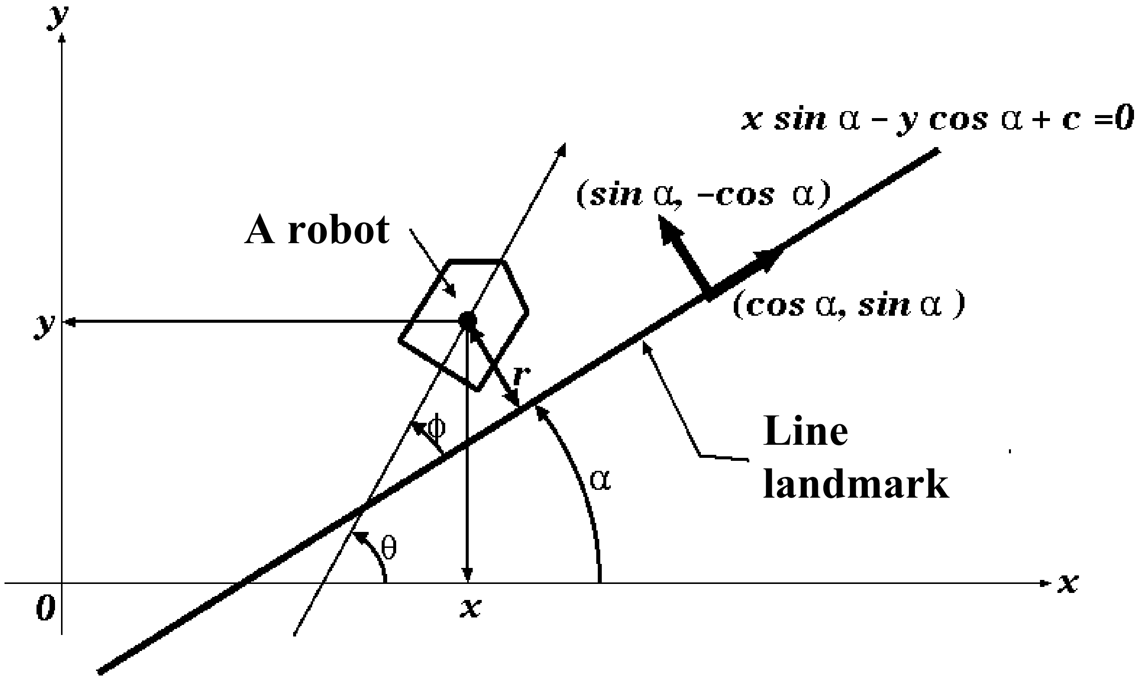
Real world space (*x*, *y*, *th*)

Sensory data space (3D case)

# Implementation of POEM

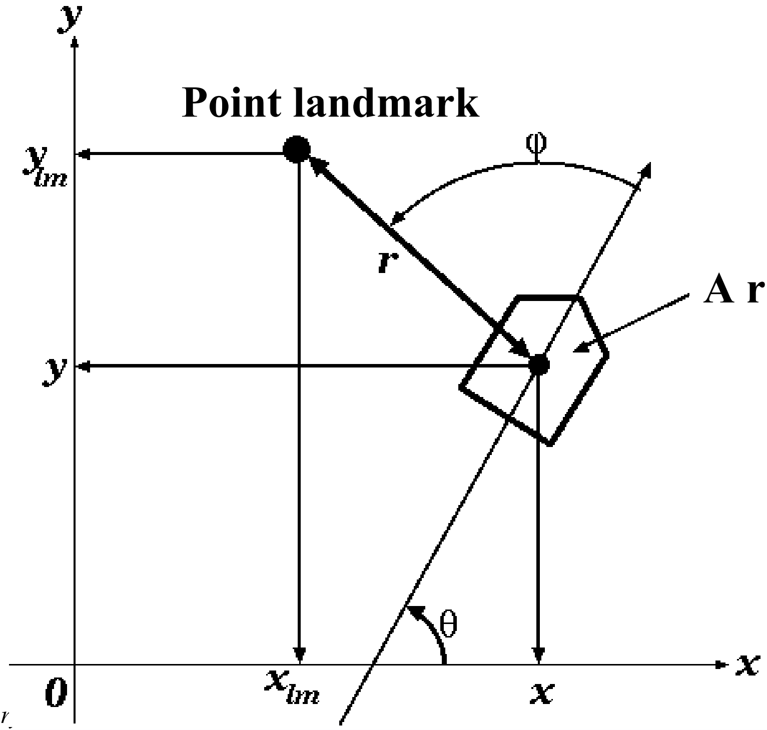
# Case1: Line Landmark

(Komoriya et al. 1993)

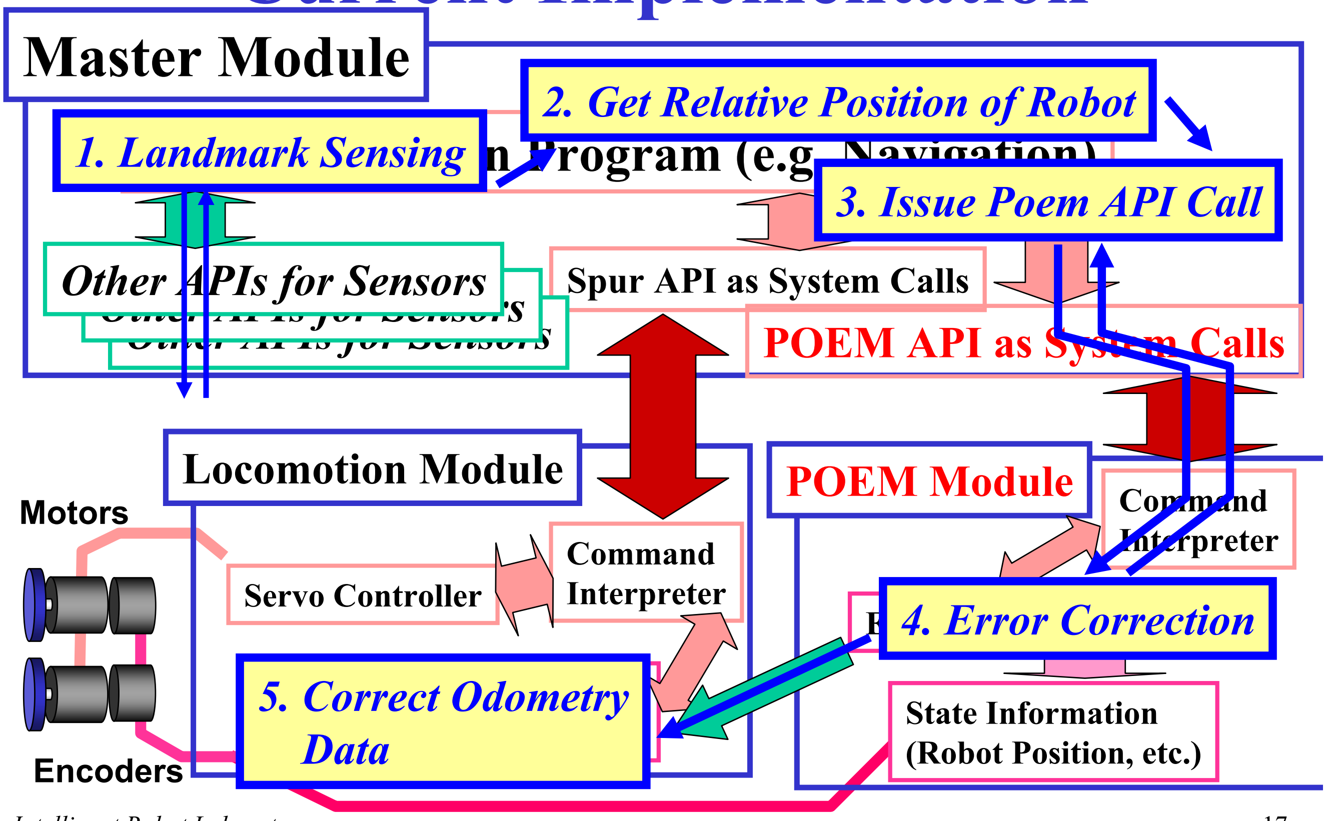


# Case2: Point Landmark

(Komoriya et al. 1993)



# Current Implementation

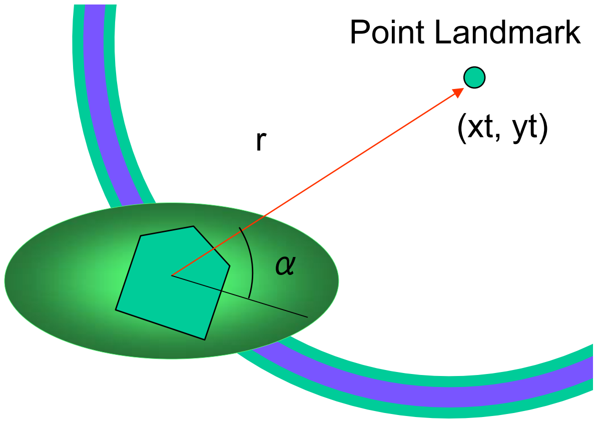


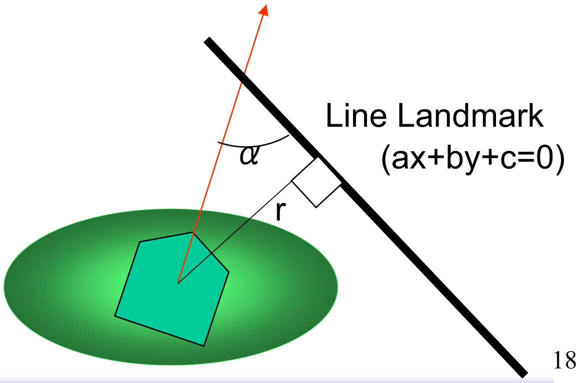
# POEM fuse command (1)

- **Poem_fuse_Pmark_DirDis** $(x_t, y_t, \alpha, r, \sigma_\alpha, \sigma_r)$

- **Poem_fuse_Pmark_Dir** $(x_t, y_t, \alpha, \sigma_\alpha)$

- **Poem_fuse_Lmark_DirDis** $(a, b, c, \alpha, r, \sigma_\alpha, \sigma_r)$



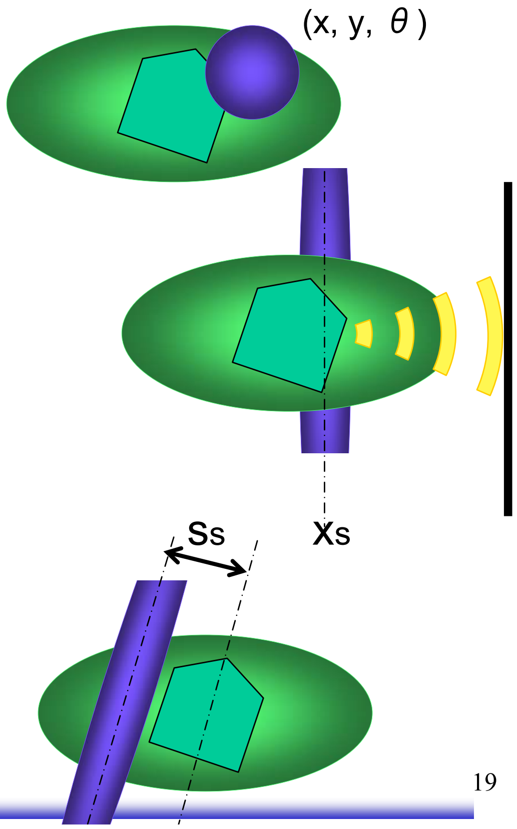


# POEM fuse command (2)

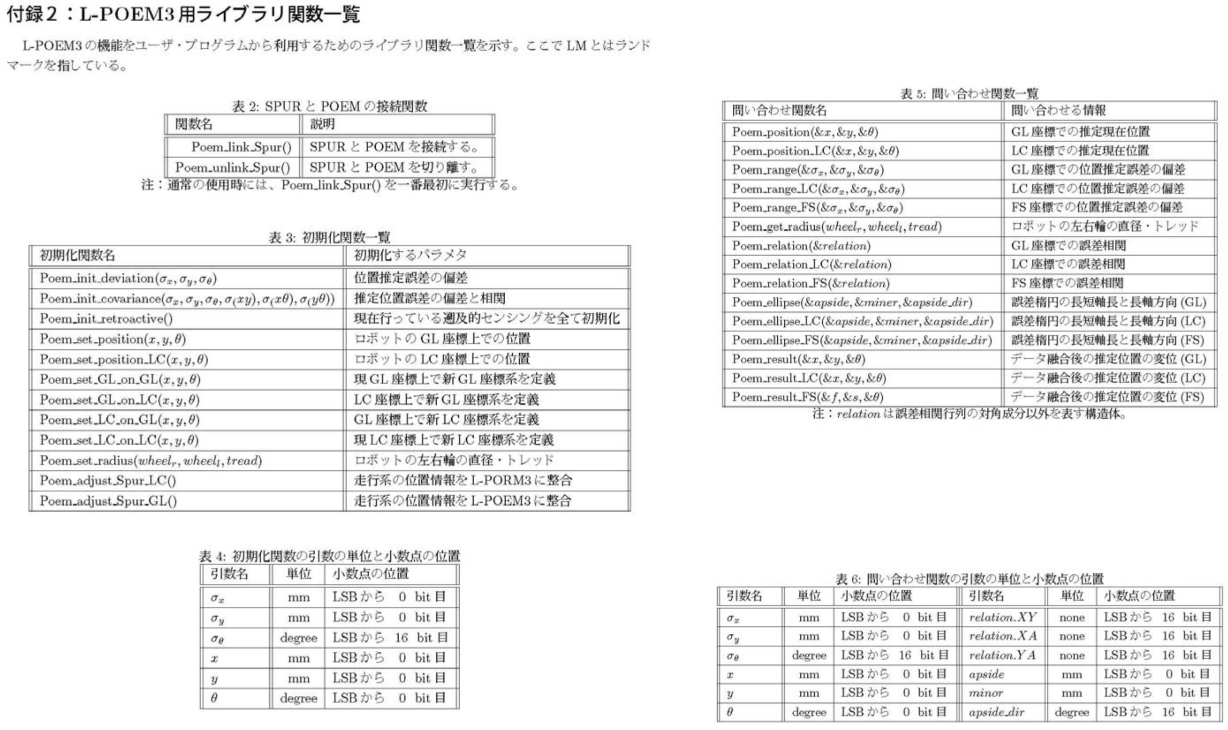
- Poem_fuse_position
  ($x_s, y_s, \theta_s$, &Σ)

- Poem_fuse_x($x_s$, $\sigma_x$)
- Poem_fuse_y($y_s$, $\sigma_y$)
- Poem_fuse_th($\theta_s$, $\sigma_\theta$)

- Poem_fuse_F($f_s$, $\sigma_f$ )
- Poem_fuse_S($s_s$, $\sigma_s$ )
- Poem_fuse_O($o_s$, $\sigma_o$ )



# Other Commands



付録２：L-POEM3 用ライブラリ関数一覧

L-POEM3 の機能をユーザ・プログラムから利用するためのライブラリ関数一覧を示す。ここで LM とはランドマークを指している。

表 2: SPUR と POEM の接続関数

| 関数名 | 説明 |
|---|---|
| Poem_link_Spur() | SPUR と POEM を接続する。 |
| Poem_unlink_Spur() | SPUR と POEM を切り離す。 |

注：通常の使用時には、Poem_link_Spur() を一番最初に実行する。

表 3: 初期化関数一覧

| 初期化関数名 | 初期化するパラメタ |
|---|---|
| Poem_init_deviation($\sigma_x, \sigma_y, \sigma_\theta$) | 位置推定誤差の偏差 |
| Poem_init_covariance($\sigma_x, \sigma_y, \sigma_\theta, \sigma_(xy), \sigma_(x\theta), \sigma_(y\theta)$) | 推定位置誤差の偏差と相関 |
| Poem_init_retroactive() | 現在行っている遡及的センシングを全て初期化 |
| Poem_set_position($x, y, \theta$) | ロボットの GL 座標上での位置 |
| Poem_set_position_LC($x, y, \theta$) | ロボットの LC 座標上での位置 |
| Poem_set_GL_on_GL($x, y, \theta$) | 現 GL 座標上で新 GL 座標系を定義 |
| Poem_set_GL_on_LC($x, y, \theta$) | LC 座標上で新 GL 座標系を定義 |
| Poem_set_LC_on_GL($x, y, \theta$) | GL 座標上で新 LC 座標系を定義 |
| Poem_set_LC_on_LC($x, y, \theta$) | 現 LC 座標上で新 LC 座標系を定義 |
| Poem_set_radius($wheel_r, wheel_l, tread$) | ロボットの左右輪の直径・トレッド |
| Poem_adjust_Spur_LC() | 走行系の位置情報を L-PORM3 に整合 |
| Poem_adjust_Spur_GL() | 走行系の位置情報を L-POEM3 に整合 |

表 4: 初期化関数の引数の単位と小数点の位置

| 引数名 | 単位 | 小数点の位置 |
|---|---|---|
| $\sigma_x$ | mm | LSB から 0 bit 目 |
| $\sigma_y$ | mm | LSB から 0 bit 目 |
| $\sigma_\theta$ | degree | LSB から 16 bit 目 |
| $x$ | mm | LSB から 0 bit 目 |
| $y$ | mm | LSB から 0 bit 目 |
| $\theta$ | degree | LSB から 0 bit 目 |

表 5: 問い合わせ関数一覧

| 問い合わせ関数名 | 問い合わせる情報 |
|---|---|
| Poem_position(&$x$, &$y$, &$\theta$) | GL 座標での推定現在位置 |
| Poem_position_LC(&$x$, &$y$, &$\theta$) | LC 座標での推定現在位置 |
| Poem_range(&$\sigma_x$, &$\sigma_y$, &$\sigma_\theta$) | GL 座標での位置推定誤差の偏差 |
| Poem_range_LC(&$\sigma_x$, &$\sigma_y$, &$\sigma_\theta$) | LC 座標での位置推定誤差の偏差 |
| Poem_range_FS(&$\sigma_x$, &$\sigma_y$, &$\sigma_\theta$) | FS 座標での位置推定誤差の偏差 |
| Poem_get_radius($wheel_r, wheel_l, tread$) | ロボットの左右輪の直径・トレッド |
| Poem_relation(&*relation*) | GL 座標での誤差相関 |
| Poem_relation_LC(&*relation*) | LC 座標での誤差相関 |
| Poem_relation_FS(&*relation*) | FS 座標での誤差相関 |
| Poem_ellipse(&*apside*, &*miner*, &*apside_dir*) | 誤差楕円の長短軸長と長軸方向 (GL) |
| Poem_ellipse_LC(&*apside*, &*miner*, &*apside_dir*) | 誤差楕円の長短軸長と長軸方向 (LC) |
| Poem_ellipse_FS(&*apside*, &*miner*, &*apside_dir*) | 誤差楕円の長短軸長と長軸方向 (FS) |
| Poem_result(&$x$, &$y$, &$\theta$) | データ融合後の推定位置の変位 (GL) |
| Poem_result_LC(&$x$, &$y$, &$\theta$) | データ融合後の推定位置の変位 (LC) |
| Poem_result_FS(&$f$, &$s$, &$\theta$) | データ融合後の推定位置の変位 (FS) |

注：*relation* は誤差相関行列の対角成分以外を表す構造体。

表 6: 問い合わせ関数の引数の単位と小数点の位置

| 引数名 | 単位 | 小数点の位置 | 引数名 | 単位 | 小数点の位置 |
|---|---|---|---|---|---|
| $\sigma_x$ | mm | LSB から 0 bit 目 | *relation.XY* | none | LSB から 16 bit 目 |
| $\sigma_y$ | mm | LSB から 0 bit 目 | *relation.XA* | none | LSB から 16 bit 目 |
| $\sigma_\theta$ | degree | LSB から 16 bit 目 | *relation.YA* | none | LSB から 16 bit 目 |
| $x$ | mm | LSB から 0 bit 目 | *apside* | mm | LSB から 0 bit 目 |
| $y$ | mm | LSB から 0 bit 目 | *minor* | mm | LSB から 0 bit 目 |
| $\theta$ | degree | LSB から 0 bit 目 | *apside_dir* | degree | LSB から 16 bit 目 |

# Retroactive Position Estimation

## Time consuming case in sensory data processing.

## POEM III

S. Maeyama, A. Ohya, S. Yuta: “Non-stop outdoor navigation of a mobile robot”,
International Conference on Intelligent Robots and Systems (IROS) '95,pp.130-135 (1995-08,Pittsburgh) .

# Conclusions

- Philosophy and background of POEM
- Command System of POEM

- Hope to be a reference plan to develop a standard specification

# Acknowledgement

- Special thanks to
  - Prof. Shin’ichi Yuta for presentation permission

robotics/2006-06-13

# SAIT Proposal of Standards for Robot Localization based on MDA

27 June 2006

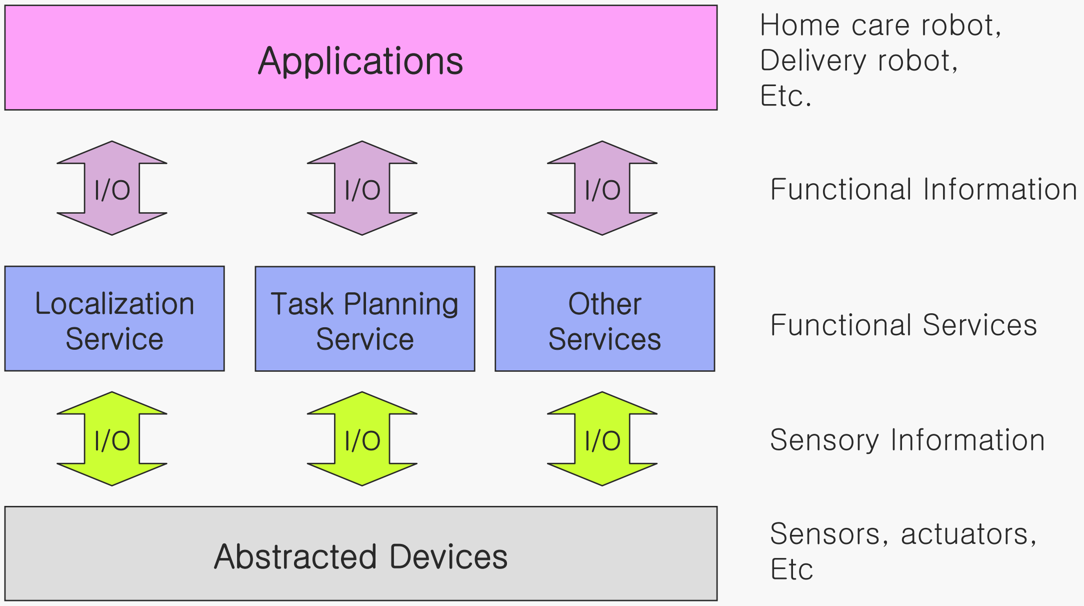
Yeonho Kim

Samsung Advanced Institute of Technology

## What we should standardize for localization services in robotics?

- **Definition of localization service:** Functional description of the localization services in robotics
- **Input & output specification**: Data structure related to localization of robots and other objects in a working space.
- **Internal data specification**: Geometrical representation and its data structure of spatial environment including robots and other objects and relationship between them.
- **Abstraction of algorithm**: Abstracted structure of algorithms to find the position and orientation of robots and other objects in the working area.

# I/O Specification



Applications

Home care robot,
Delivery robot,
Etc.

I/O | I/O | I/O

Functional Information

Localization Service | Task Planning Service | Other Services

Functional Services

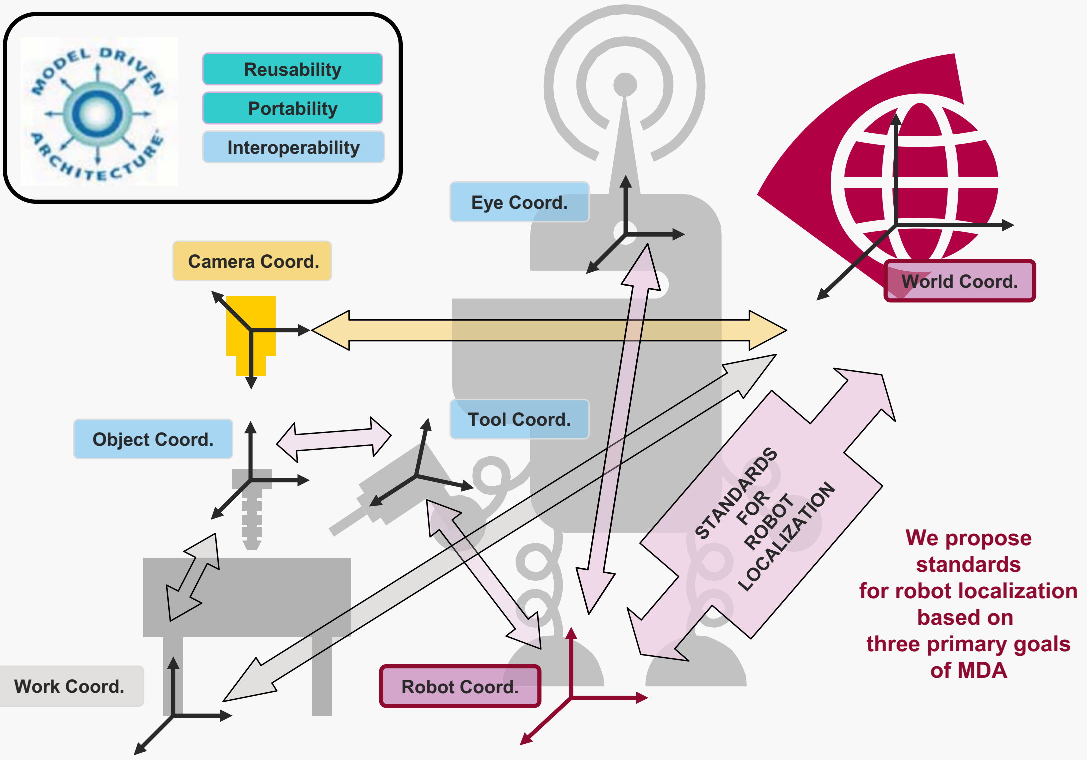
I/O | I/O | I/O

Sensory Information

Abstracted Devices

Sensors, actuators,
Etc

# Internal Data Representation (Coordinate Systems)



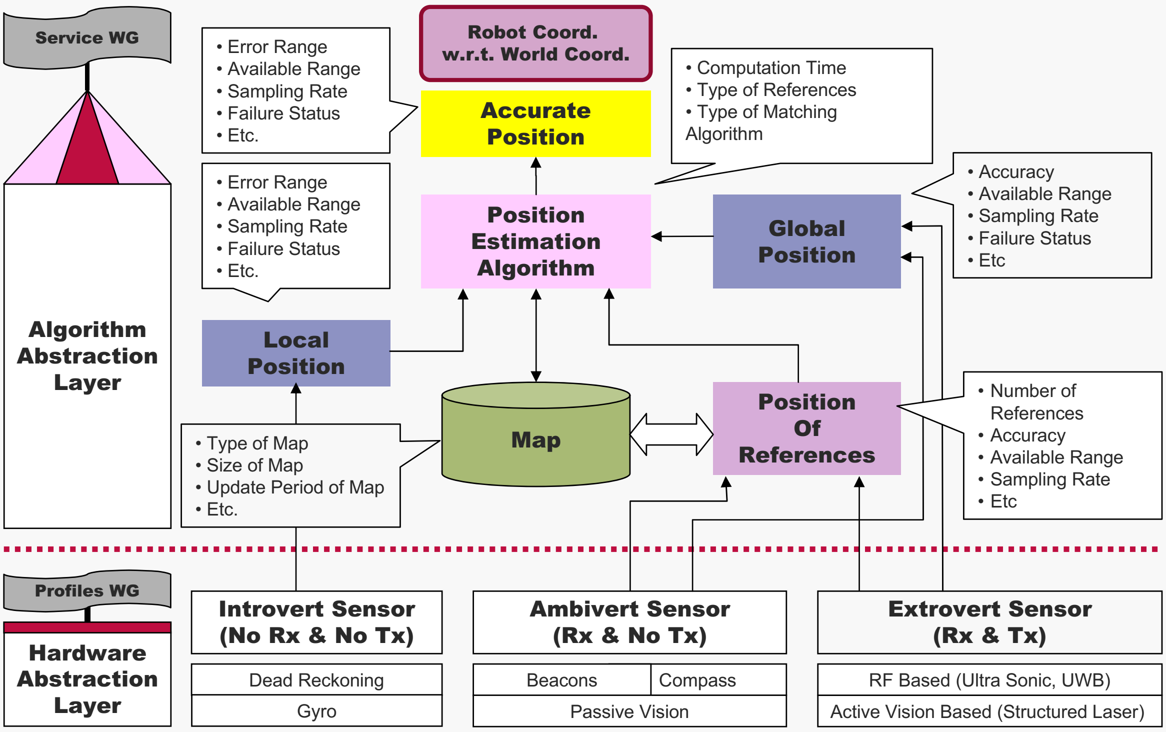
Reusability

Portability

Interoperability

We propose standards for robot localization based on three primary goals of MDA

# Abstraction of Algorithm



Service WG

Algorithm Abstraction Layer

- Error Range
- Available Range
- Sampling Rate
- Failure Status
- Etc.

Robot Coord. w.r.t. World Coord.

Accurate Position

- Computation Time
- Type of References
- Type of Matching Algorithm

- Error Range
- Available Range
- Sampling Rate
- Failure Status
- Etc.

Position Estimation Algorithm

Global Position

- Accuracy
- Available Range
- Sampling Rate
- Failure Status
- Etc

Local Position

Map

Position Of References

- Number of References
- Accuracy
- Available Range
- Sampling Rate
- Etc

- Type of Map
- Size of Map
- Update Period of Map
- Etc.

Profiles WG

Hardware Abstraction Layer

Introvert Sensor (No Rx & No Tx)

Dead Reckoning

Gyro

Ambivert Sensor (Rx & No Tx)

Beacons

Compass

Passive Vision

Extrovert Sensor (Rx & Tx)

RF Based (Ultra Sonic, UWB)

Active Vision Based (Structured Laser)

# Thanks for Your Attention !

Robotics DTF – Robotics Service WG meeting @ Boston, USA

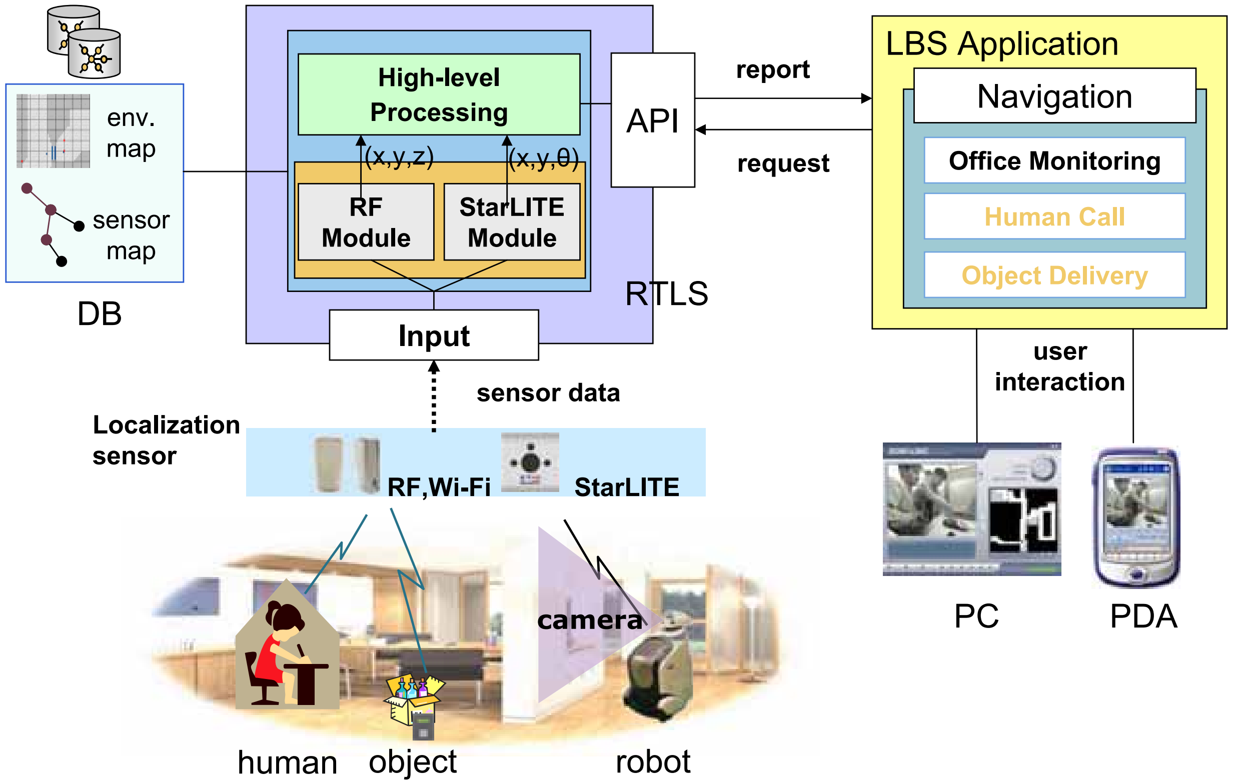
# Issues on Localization Service

2006. 6. 25.
Wonpil Yu
ETRI Intelligent Robot Research Division

## Contents

❑ Localization service architecture
❑ Use case
❑ Requirements on localization service
❑ Where to standardize?
❑ Things to consider

# An architecture using localization service



DB

env. map

sensor map

High-level Processing

(x,y,z)

(x,y,θ)

RF Module

StarLITE Module

API

RTLS

Input

report

request

LBS Application

Navigation

Office Monitoring

Human Call

Object Delivery

user interaction

PC

PDA

sensor data

Localization sensor

RF,Wi-Fi

StarLITE

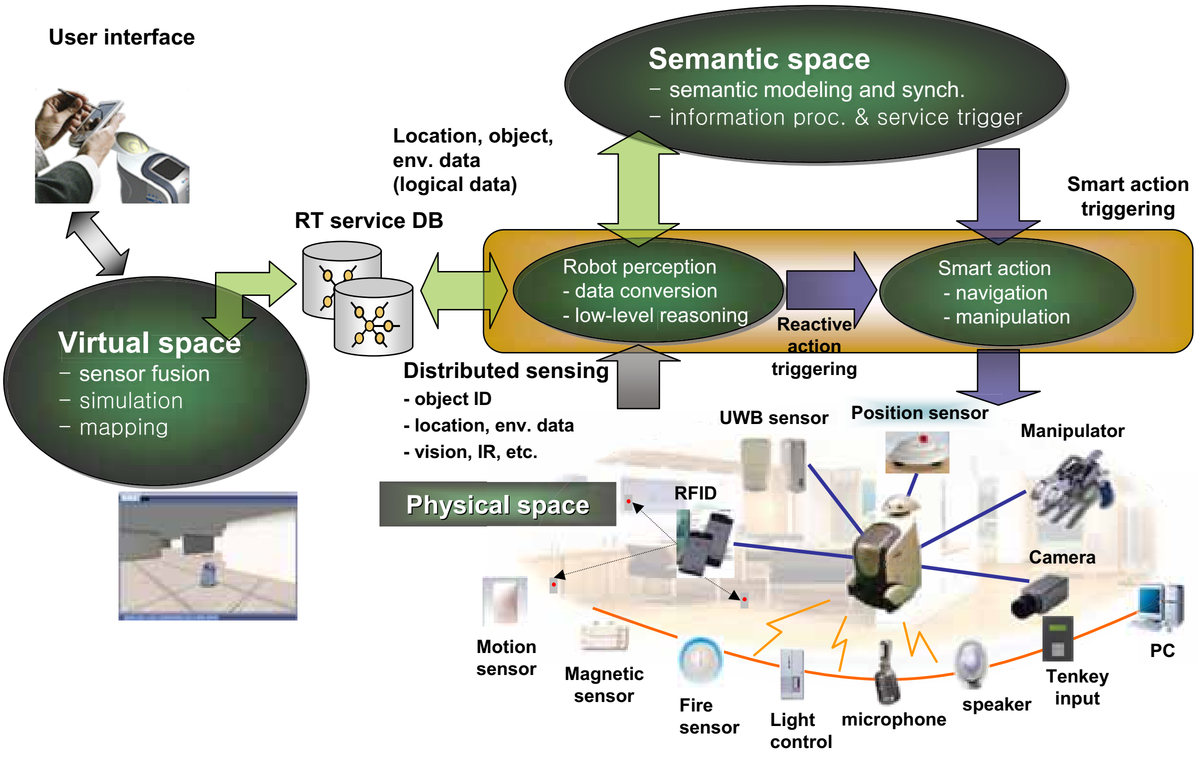
camera

human

object

robot

# Use case of localization service (building an intelligent robotic space 1/2)



User interface

Semantic space
- semantic modeling and synch.
- information proc. & service trigger

Location, object, env. data (logical data)

RT service DB

Smart action triggering

Robot perception
- data conversion
- low-level reasoning

Smart action
- navigation
- manipulation

Reactive action triggering

Virtual space
- sensor fusion
- simulation
- mapping

Distributed sensing
- object ID
- location, env. data
- vision, IR, etc.

Physical space

UWB sensor

Position sensor

Manipulator

RFID

Camera

PC

Motion sensor

Magnetic sensor

Fire sensor

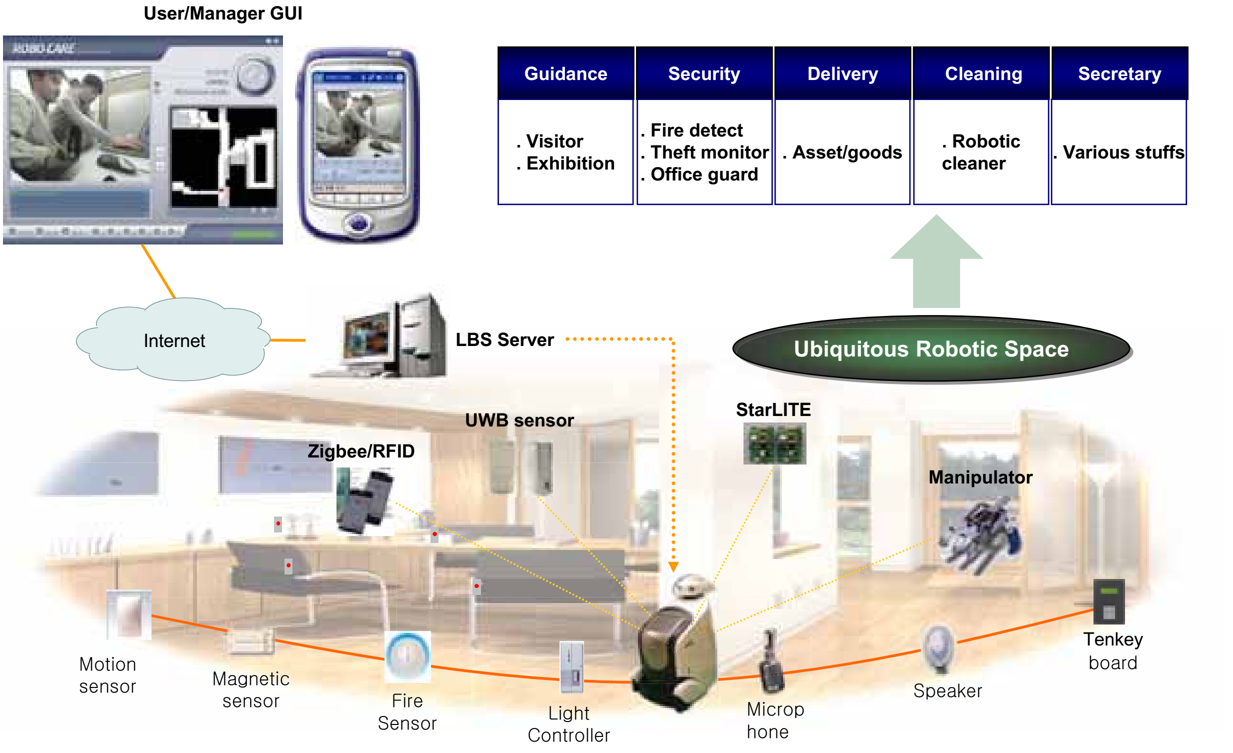
Light control

microphone

speaker

Tenkey input

## Use case of localization service (building an intelligent robotic space 2/2)



| Guidance | Security | Delivery | Cleaning | Secretary |
|---|---|---|---|---|
| . Visitor<br>. Exhibition | . Fire detect<br>. Theft monitor<br>. Office guard | . Asset/goods | . Robotic cleaner | . Various stuffs |

## Requirements on localization service (1/2)

❑ Route determination

- Determine a route and n
- Navigation information: r , etc.

❑ Positioning

- localization: given symbo
- Reverse (x, y, θ), find symbolic (semantic) location
  - Sym ion: room A, next to a book shelf, etc.

❑ Presentatic

- Display g map, route (path), navigation information, etc. on a

# Requirements on localization service (2/2)

- ❑ Location calculation
  - Determine position of a robot by using physical sensor data equipped on top of the robot or from a sensor network
- ❑ Map management
  - Management of update information
  - Management of layered map information
    - Conventional environment map (e.g., binary SONAR map, omni-directional visual map)
    - Information map including (position data, properties of objects, special information related to a particular area)

# Where to standardize? (1/2)

- ❑ Route determination
- ❑ Positioning
  - Modeling language or data type to describe indoor environment consisting of abundant number of objects
- ❑ Presentation
  - How to pack (encode) information in order to deliver the information to a user terminal? (SVG?)

# Where to standardize? (2/2)

❑ Location calculation

- API interface exposing location data to 3rd party application
- Localization middleware needed to handle various sensors and networking
- Unified way of representing location uncertainty for robot navigation

❑ Map management

- Occupancy (grid) map/topological map/semantic map
- How to transfer map data among robot, server, and user terminal

# Things to consider

❑ LIF (Location Interoperability Forum) at OMA (Open Mobile Alliance)

❑ Open Location Services (OpenLS) at OGC (Open GIS Consortium)

# Thank you!
# Any questions?

# Localization/Positioning for Robotics

# 3 Physical Scales

- Global Navigation
  - which is the ability to determine one's position in absolute or map-referenced terms, and to move to a desired destination point.
- Local Navigation
  - the ability to determine one's position relative to objects (stationary or moving) in the environment, and to interact with them correctly.
- Personal Navigation
  - which involves being aware of the positioning of the various parts that make up oneself, in relation to each other and in handling objects.

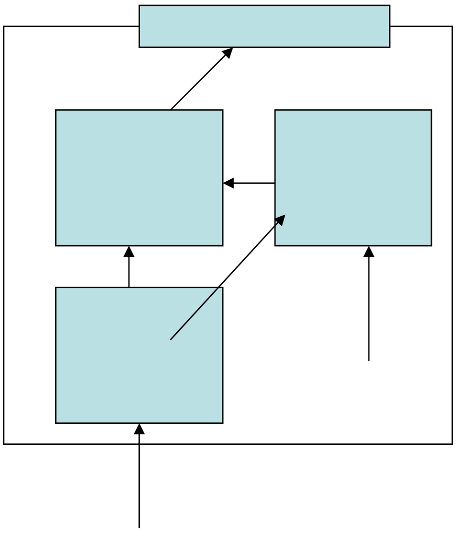
# Existing Standards / Technologies

- **Spatial Ontologies**
  - Ontospace…
- ISO/TS 10303-1131:2005 specifies the application module for Construction geometry.
  - * geometric primitives of one and two dimensions;
  - * geometry used in the construction of two-dimensional and three-dimensional shape representations;
  - * association of a construction geometry to a geometric model or a construction from which it is derived.
- ISO 19107 http://portal.opengeospatial.org/files/?artifact_id=1093
- ISO 19112 Spatial referencing by geographic identifiers
- **Open Geospatial Consortium**, Inc. (OGC) (http://www.opengeospatial.org/specs/?page=abstract)
  - The Open Geospatial Consortium, Inc. (OGC) is a non-profit, international, voluntary consensus standards organization that is leading the development of standards for geospatial and location based services. Through our member-driven consensus programs, OGC works with government, private industry, and academia to create open and extensible software application programming interfaces for geographic information systems (GIS) and other mainstream technologies.
- **Open Mobile Alliance (OMA)**, (http://www.openmobilealliance.org/tech/affiliates/lif/lifindex.html)
  - The OMA Location Working Group (LOC) continues the work originated in the former Location Interoperability Forum (LIF) and Location Drafting Committee of the former WAP Forum.
- **Geography Markup Language (GML)**

---

- Introduction:
  - Define Localization Service (use all cases presented in Dr Yu's presentation) and restrict to certain area
  - Include typical structure and limit scope
  - Purpose of the specification (what is good for user, what is good for developer (why we want to standardize the sub processes (small boxes)..)
- Constraints :
  - 3 levels of positioning way to bridge them
- What is requested :
  - Input / Output Data Specification
  - Abstraction of algorithm (typical building blocks)
  - Programmers interface / User interface
  - PIM and PSM
- → Describe why a particular way was chosen, its limitation and metrics (describe how solutions will be evaluates - /resource /generalization /speed /describe main advantages-limitation of the solution…)
- Not Requested :
  - Specific Algorithm
- Issues to be discussed :
  - how to manage semantic space reasoning for localization
  - Do we need to include map data format in specification ?

- Concentrate on Location service developer or Location data user
- Request identification of  main building blocks of Positioning System (local localization, Global, Position Fusion…)



# Spur - a Locomotion Command System for a Mobile Robot

**Takashi Tsubouchi,** Professor

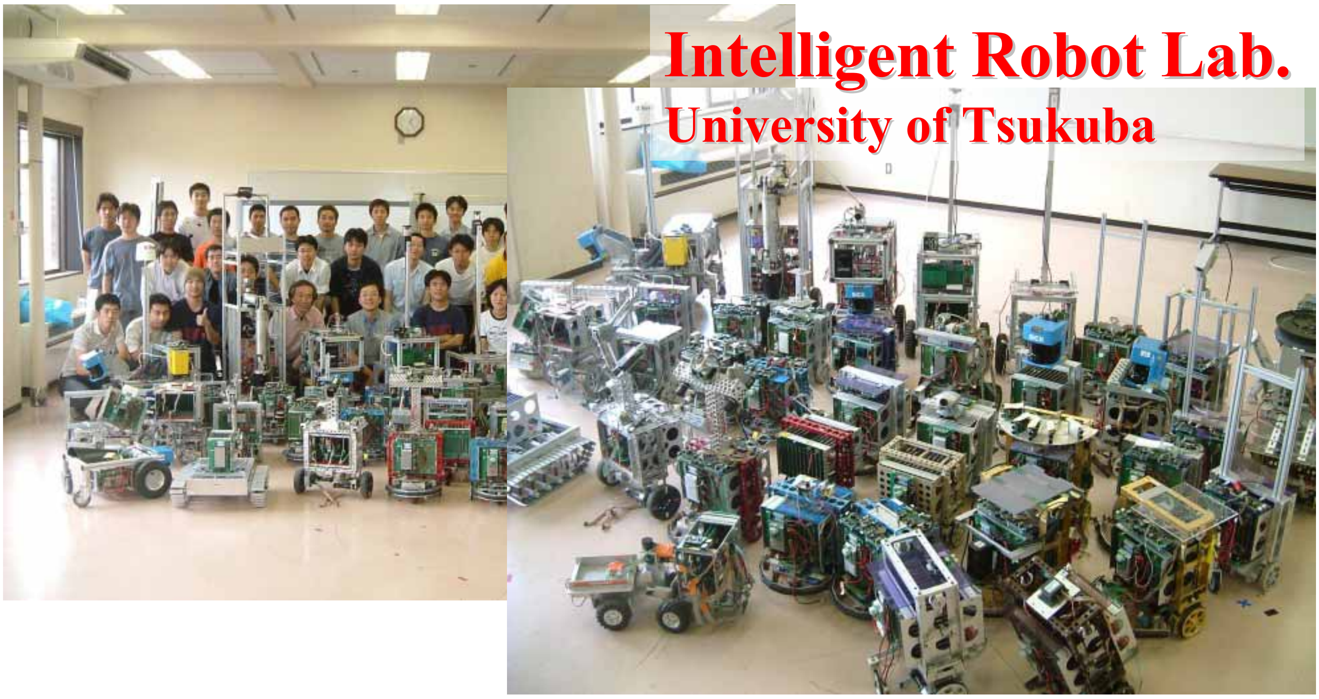
**Eijiro Takeuchi,** Ph. D candidate

**Intelligent Robot Laboratory,**

**University of Tsukuba, Japan**

# Introduction

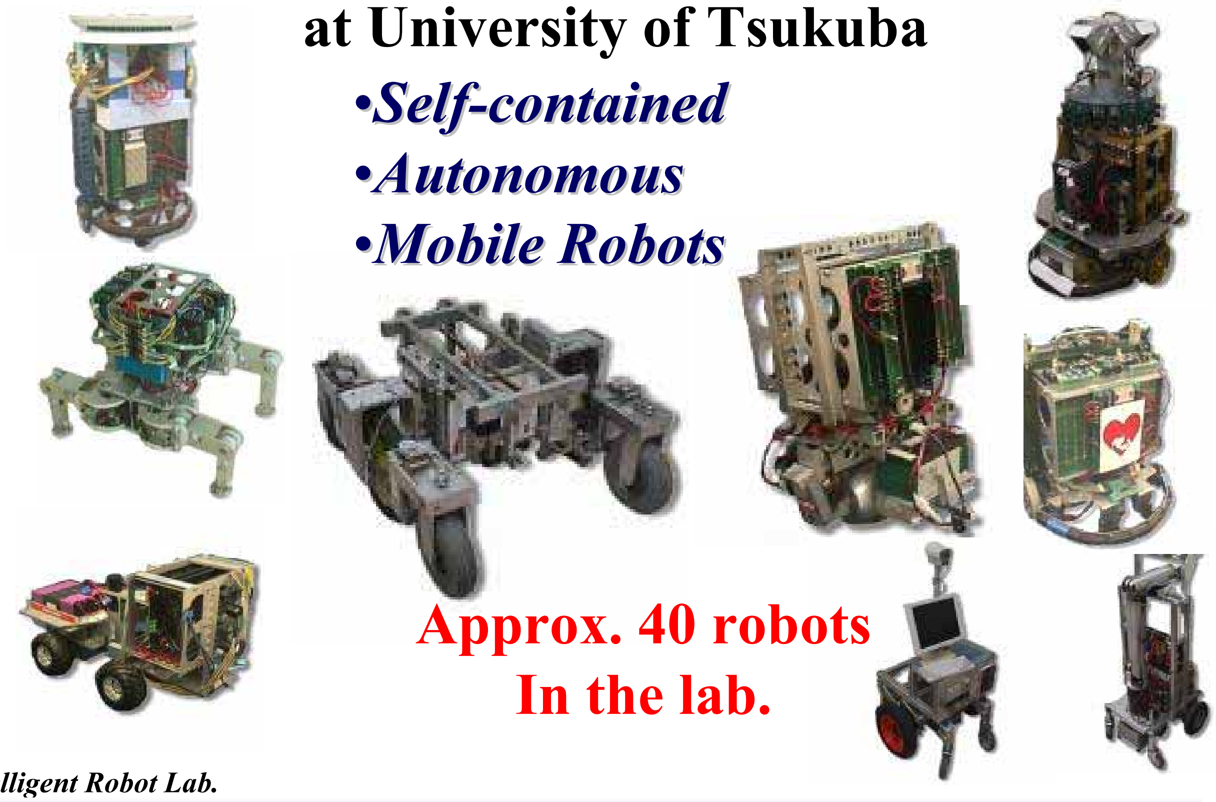
# Lab. Members and Our Robot



Intelligent Robot Lab.
University of Tsukuba

# *Intelligent Robot Laboratory “Robo-ken”*

at University of Tsukuba

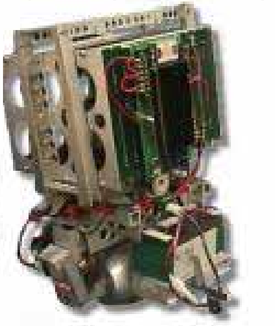


- *Self-contained*
- *Autonomous*
- *Mobile Robots*

Approx. 40 robots
In the lab.

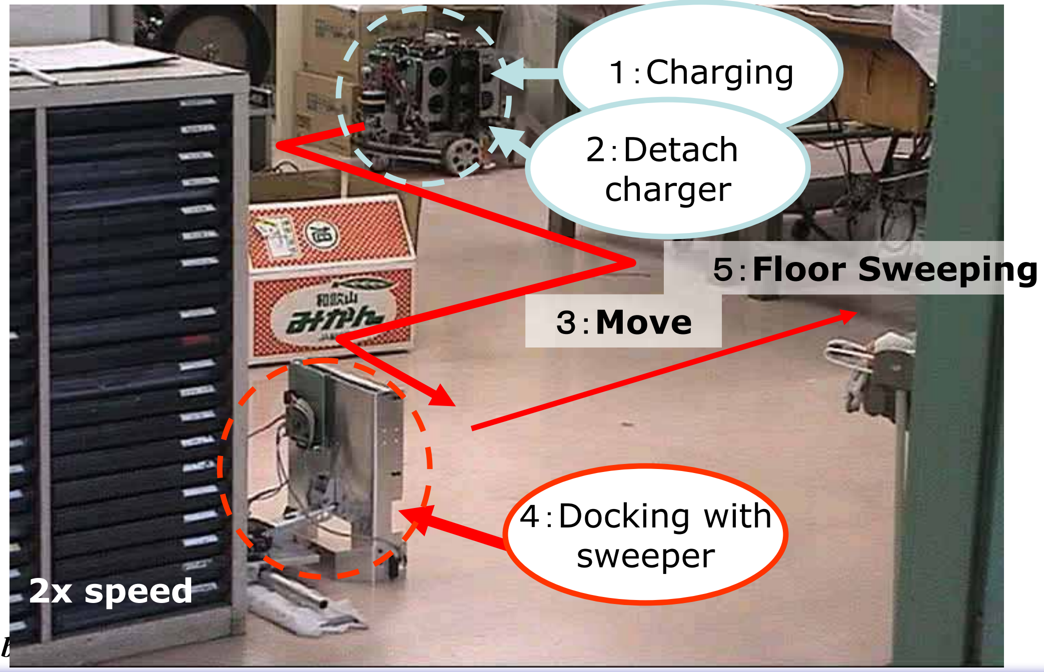
## Autonomous Mobile Robots "*YAMABICO*" family

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



## Demo (Floor Sweeping in Free Space)

### This video was presented at Tampa meeting



Courtesy
of E.Takeuchi

# Demo (Floor Sweeping in Free Space)

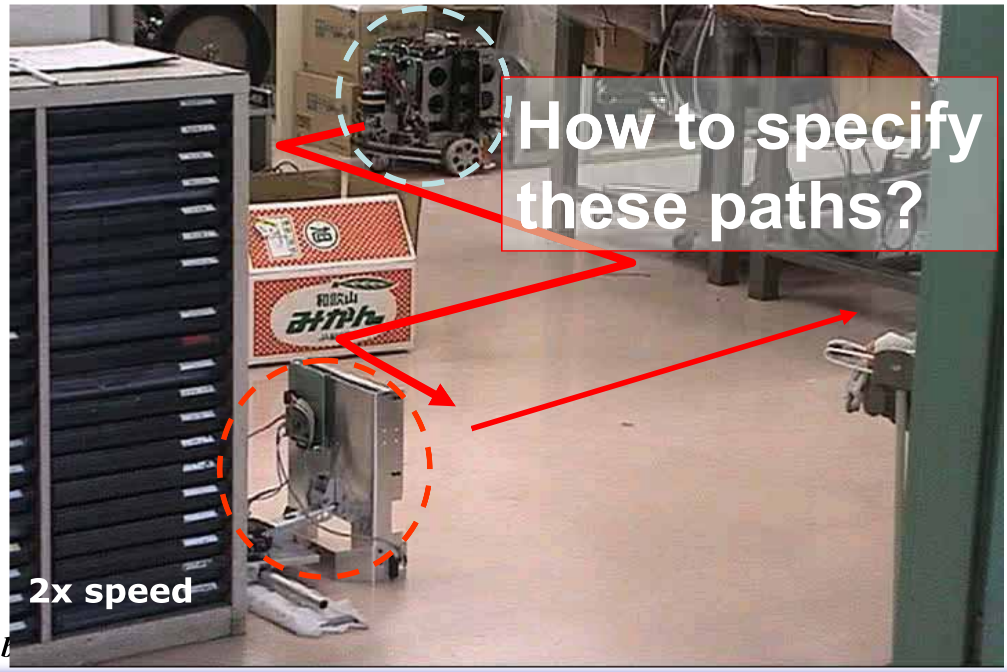
## This video was presented at Tampa meeting

Courtesy
of E.Takeuchi

**2x speed**

# Demo (Floor Sweeping in Free Space)

## This video was presented at Tampa meeting



How to specify these paths?

2x speed

Courtesy
of E.Takeuchi

# A Locomotion Control and Command System - "Spur"

- "Spur" = trajectory of a skier falling hill down (a German word)
- The command system developed by the Intelligent Robot Lab. (U. of Tsukuba)

S. Iida and S. Yuta: "Control of a Vehicle Subsystem for an Autonomous Mobile Robot with Power Wheeled Steering," 1990 IEEE International Workshop on Intelligent Motion Control pp.859-866, 1990-08 Istanbul.

S. Iida and S. Yuta: "Vehicle Command System and Trajectory Control for Autonomous Mobile Robots," 1991 IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS '91), 1991-11 Osaka.

# Users of Spur

- Intelligent Robot Lab. (Approx. 40 robots)
  - Two implementations
  - On the distributed architecture (Yamabico conventional) and the centralized architecture (PC Yamabico – Takauchi)

•"Wakamaru" (Mitsubishi Heavy Industry)

•Guard-Robo Series (ALSOK)

•…

# Background

## Fundamental policy

Separation of path specification from motion control

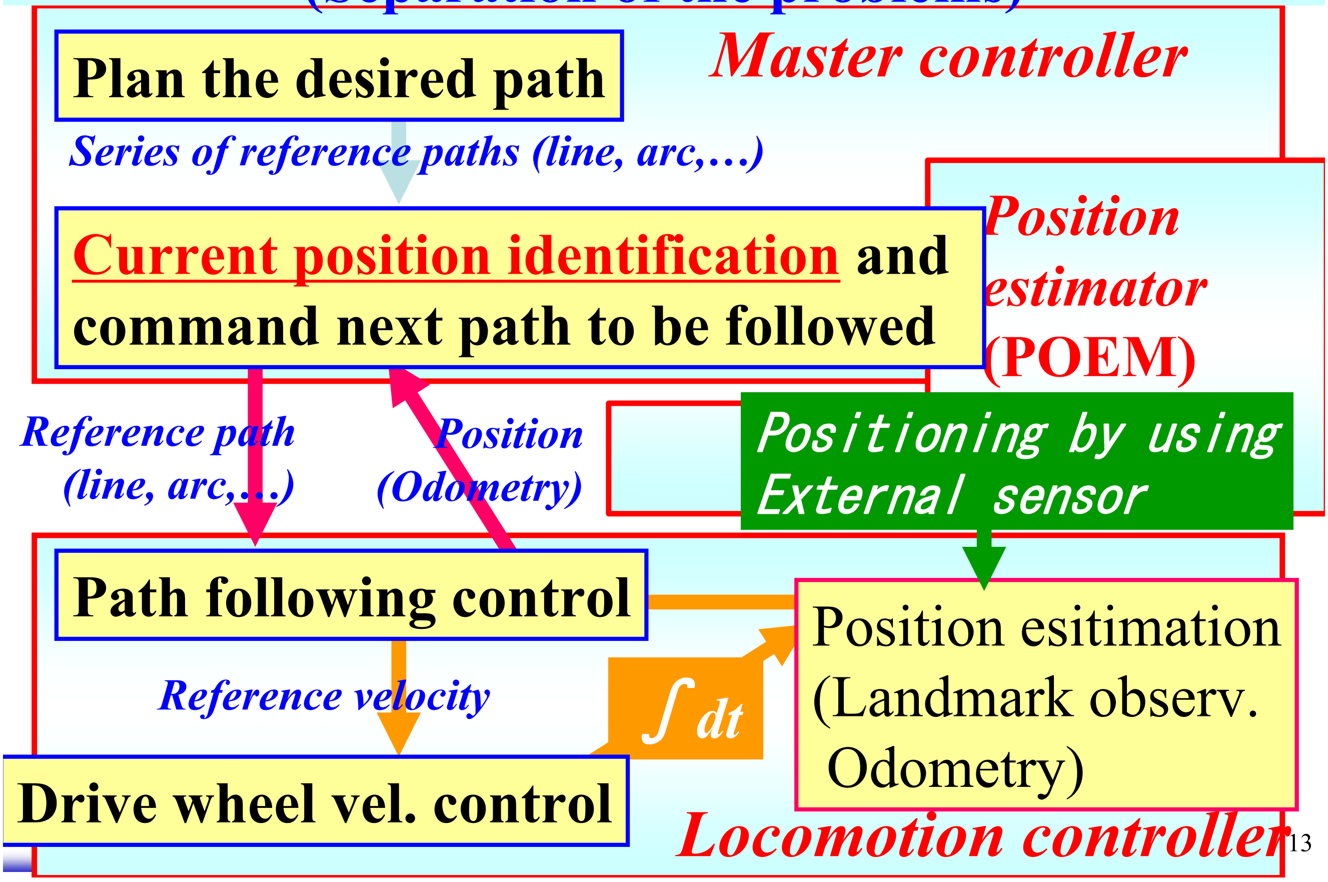
Event-driven behavior description

## How to specify mobile robot behavior?

- An example: a taxi driver and a passenger
- **The passenger asks the driver to do what:**

1. Command how to push throttle and how to steer every time.
   **Actuator control level**
2. Command desired street or turning corner with watching the outside by the passenger.
   **Event-driven commander**
3. Tell the driver the path order at once, rely on the driver and fall into asleep.
   **Already planned path given**
4. Tell the driver the destination, rely on the driver and fall into asleep.
   **Path planning by the driver**

## Behavior description and control
## (Separation of the problems)



Master controller

Plan the desired path

Series of reference paths (line, arc,…)

Current position identification and command next path to be followed

Position estimator (POEM)

Positioning by using External sensor

Reference path (line, arc,…)

Position (Odometry)

Path following control

Position esitimation (Landmark observ. Odometry)

Reference velocity

∫dt

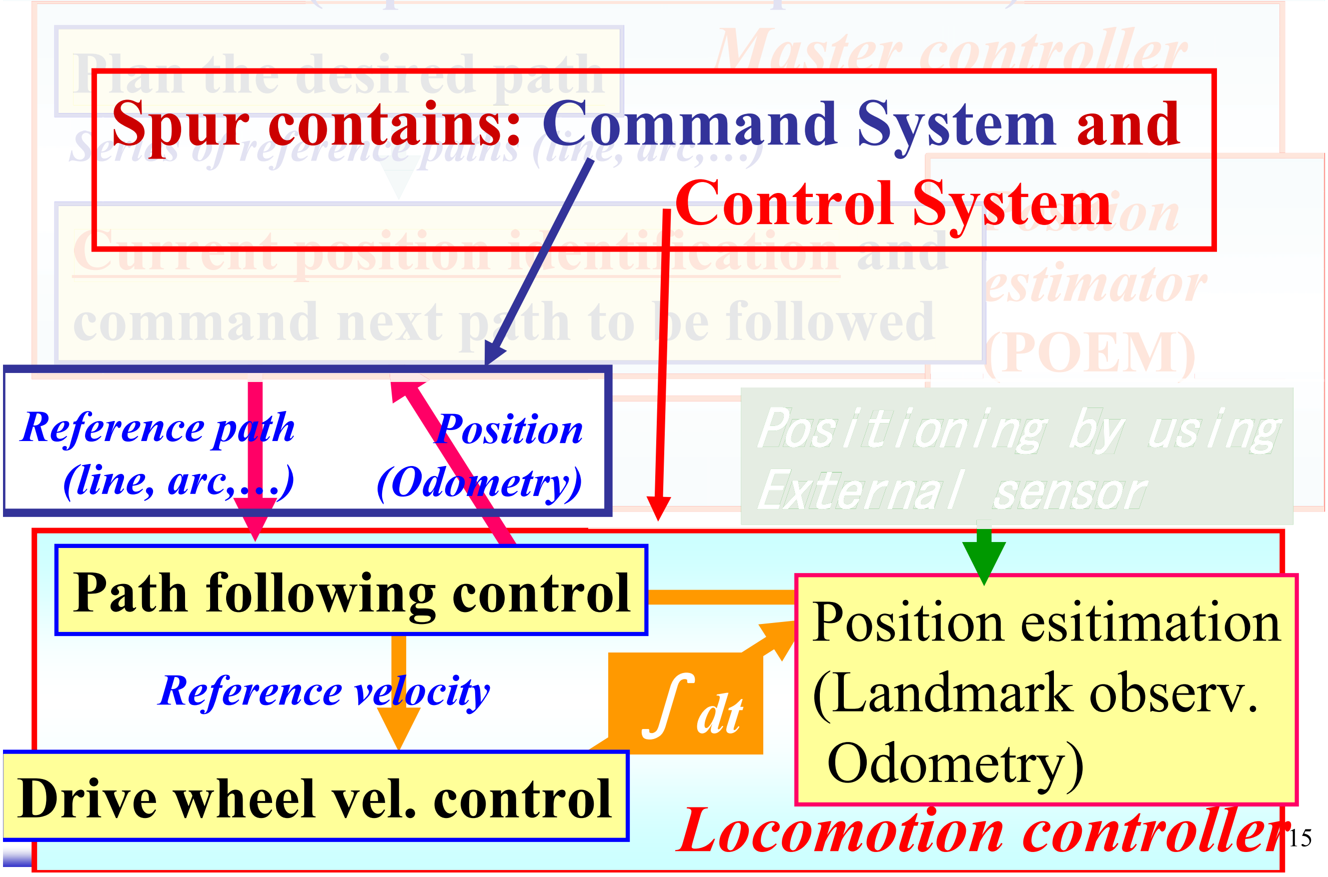
Drive wheel vel. control

Locomotion controller

# Policy

- Implement a “locomotion controller” first to follow the reference paths (line and arc).
- Behavior is described in a manner to command the necessary reference path to locomotion controller when the necessary event happens.
- Position identification and obstacle watch
  - – Description manner of the user program

## Behavior description and control (Separation of the problems)



Plan the desired path

Series of reference paths (line, arc,...)

Master controller

Current position identification and command next path to be followed

Position estimator (POEM)

Positioning by using External sensor

Spur contains: Command System and Control System

*Reference path (line, arc,...)*

*Position (Odometry)*

Path following control

*Reference velocity*

Drive wheel vel. control

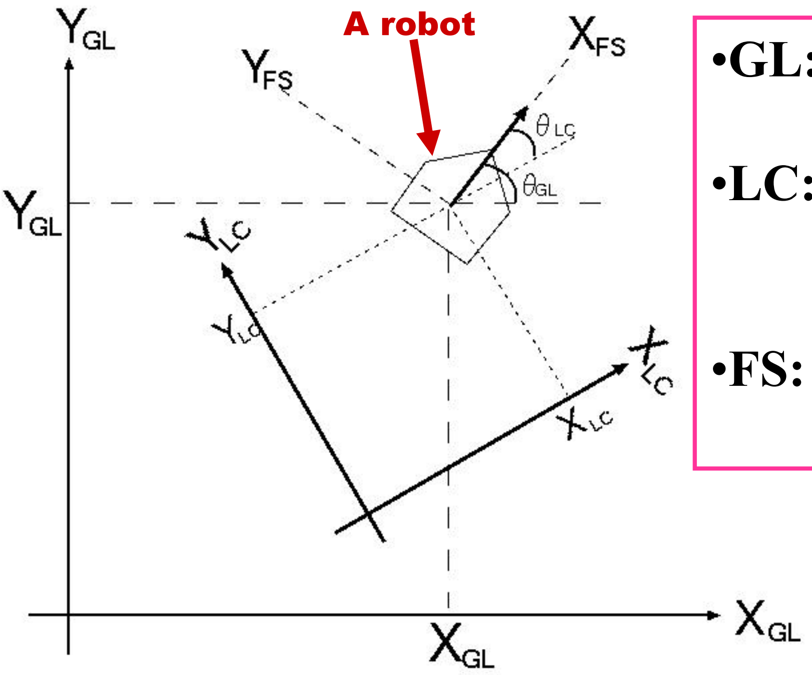
$\int dt$

Position esitimation (Landmark observ. Odometry)

*Locomotion controller*

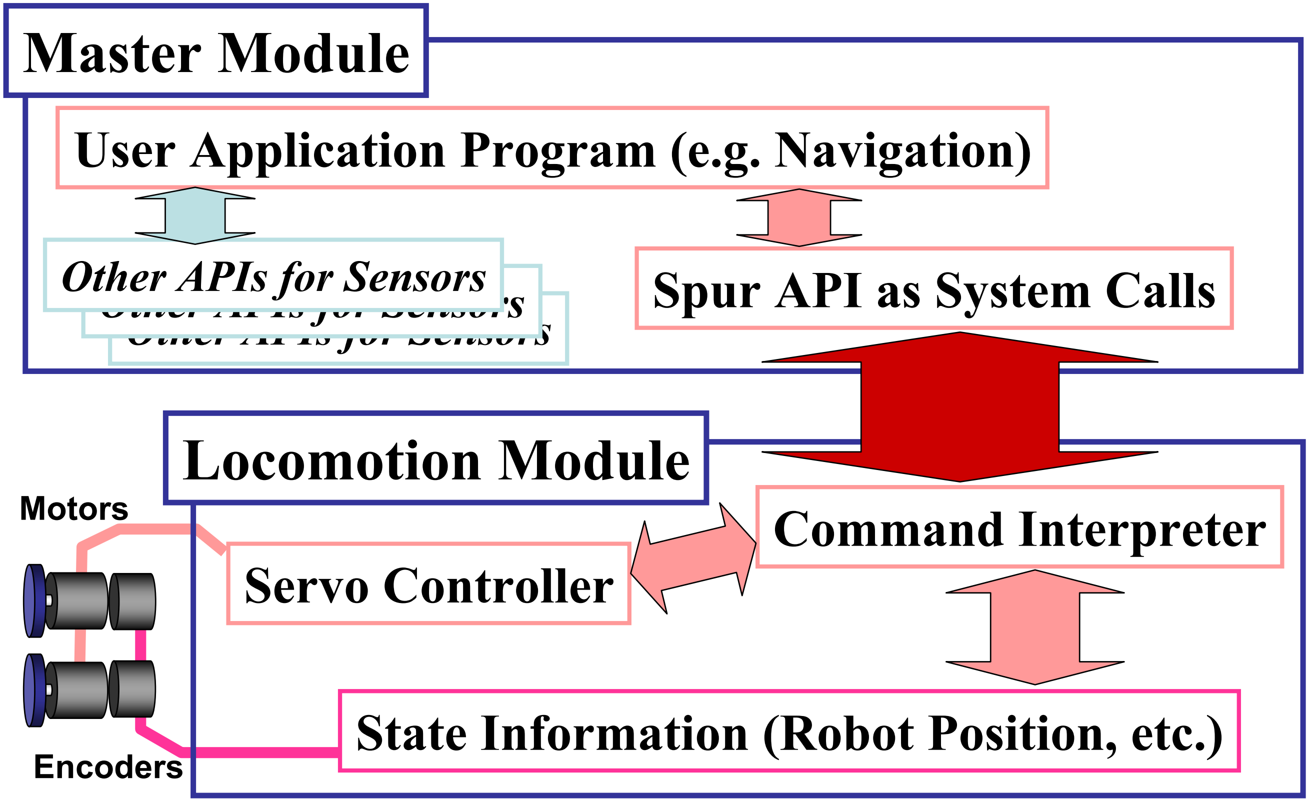
# Command System

# Coordinate Systems



- •GL: Global coordinate
  (Fixed on the ground)
- •LC: Local coordinate
  (Fixed on the ground)
- •FS: Front-Side coordinate
  (moves on the robot)

# Framework



Master Module

User Application Program (e.g. Navigation)

*Other APIs for Sensors*

Spur API as System Calls

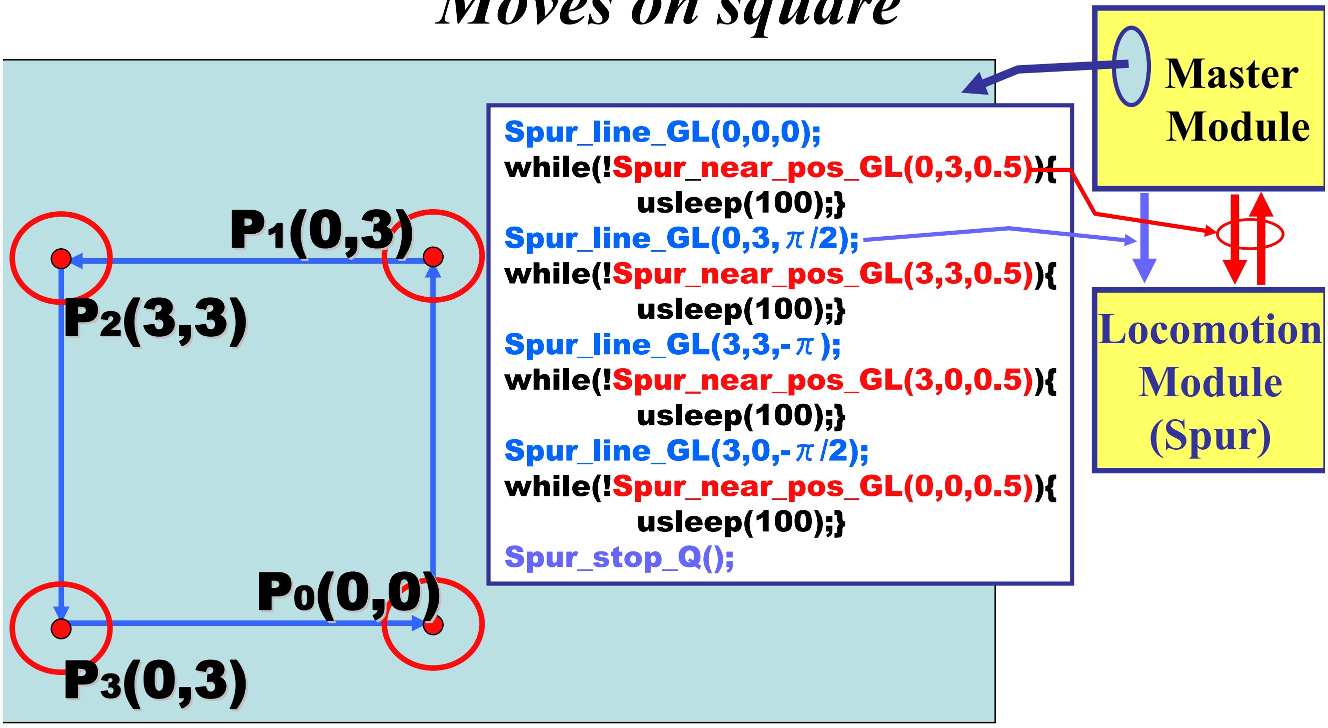
Locomotion Module

Command Interpreter

Servo Controller

State Information (Robot Position, etc.)

# An example of user program *"Moves on square"*



```
Spur_line_GL(0,0,0);
while(!Spur_near_pos_GL(0,3,0.5)){
        usleep(100);}
Spur_line_GL(0,3,π/2);
while(!Spur_near_pos_GL(3,3,0.5)){
        usleep(100);}
Spur_line_GL(3,3,-π);
while(!Spur_near_pos_GL(3,0,0.5)){
        usleep(100);}
Spur_line_GL(3,0,-π/2);
while(!Spur_near_pos_GL(0,0,0.5)){
        usleep(100);}
Spur_stop_Q();
```

# Command Summary

- Motion control commands
  - Go along line / Go along arc / Spin / Stop and servo
- Get state information commands
  - Position / Speed / Acceleration
- Change Parameters commands
  - Velocity setting / Acceleration setting / Coordinate system setting

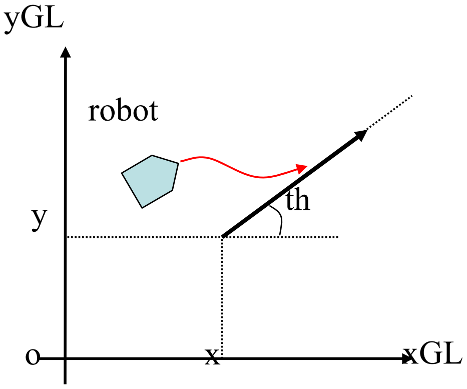
# Motion Control Commands (1)

## Go Along Line

void Spur_line_GL(int x, int y, int th)

void Spur_line_LC(int x, int y, int th)

void Spur_line_FS(int x, int y, int th)



**Go along a line passing through (*x,y*) and angle with *th* in the specified coordinate system. [(*x,y*) in millimeter in unit and *th* in degree ]**

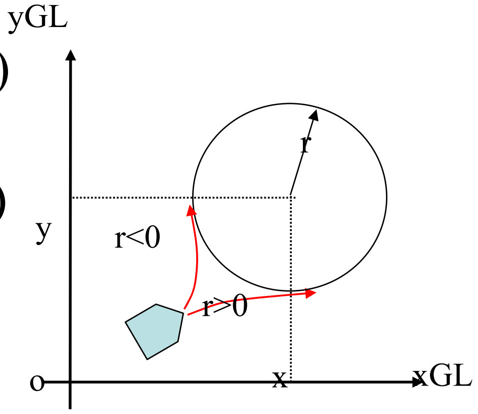
# Motion Control Commands (2)

## Go Along Arc (1)

void Spur_arc_c_GL(int x, int y, int r)

void Spur_arc_c_LC(int x, int y, int r)

void Spur_arc_c_FS(int x, int y, int r)



**Go along an arc of center (*x,y*) and radius *r* (if *r* < 0 then CW, if *r*> 0 then CCW).**

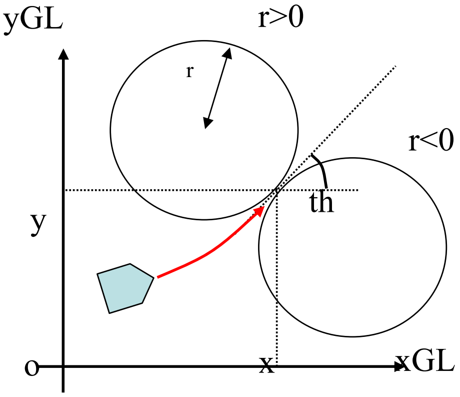
# Motion Control Commands (3)

## Go Along Arc (2)

void Spur_arc_t_GL(int x, int y, int r)

void Spur_arc_t_LC(int x, int y, int r)

void Spur_arc_t_FS(int x, int y, int r)



**Go along an arc of which tangent passes through ( *x, y* ) and has angle *th.* (if r < 0 then CW, or if r > 0 then CCW)**

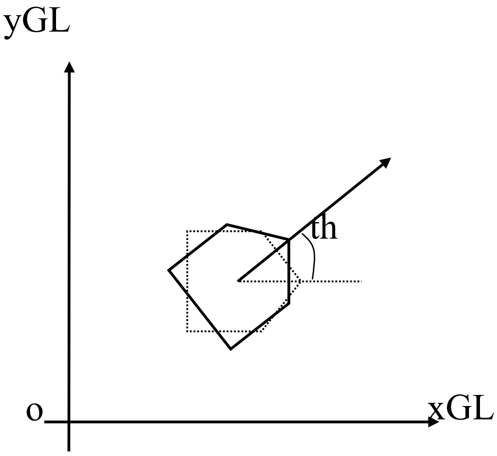
# Motion Control Commands (4)

## Spin

void Spur_spin_GL(int th)

void Spur_spin_LC(int th)

void Spur_spin_FS(int th)



**Spin until heading direction becomes *th* in degree**

# Motion Control Commands (5)

## Stop and servo

void Spur_stop_GL(int x, int y, int th)

void Spur_stop_FS(int x, int y, int th)

void Spur_stop_LC(int x, int y, int th)

**Stop on the line passing through ( *x, y* ) an angle *th* .**

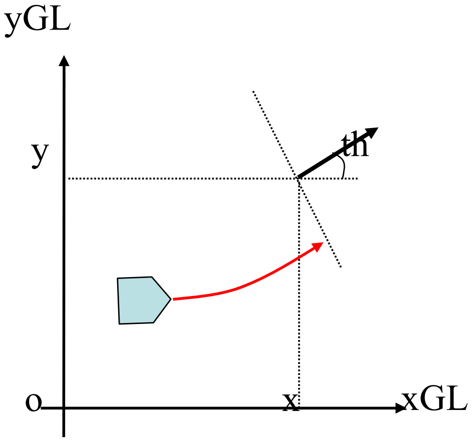
void Spur_stop_q( )
void Spur_stop_Q( )

**Stop with maximum deceleration.**

void Spur_servo_free( )

void Spur_servo( )

**Servo free or on**



# Get State Information Commands

## Position related

int Spur_get_pos_GL(int *x,int *y,int *th)

int Spur_get_pos_LC(int *x,int *y,int *th)

**Get positions (*x,y,th*)**

int Spur_near_pos_GL(int x, int y, int r)

int Spur_near_pos_LC(int x, int y, int r)

**Returns *1* if the robot is within radius *r* whose center is at (*x,y*) otherwise *0*.**

int Spur_near_ang_GL(int th, int error)

int Spur_near_ang_LC(int th, int error)

**Returns *1* if the robot heading direction is angle *th* within tolerance *error*, otherwise *0*.**

int Spur_over_line_GL(int x,int y,int th)

int Spur_over_line_LC(int x,int y,int th)

**Returns *1* if the robot goes over the line passing through ( *x, y* ) and angle *th+90 degs*, otherwise *0*.**

## Speed and acceleration related

int Spur_get_vel(int *v, int *w)

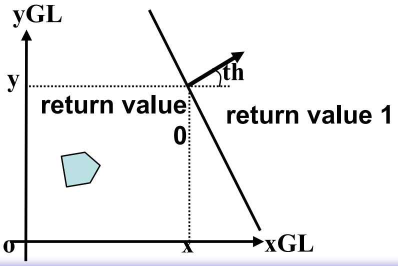
**Get speed and angular speed (*v,w*) .**

int Spur_near_vel(int vel, int error)

**Returns *1* if the speed is *vel* within tolerance of *error*, otherwise *0*.**

int Spur_near_ang_vel(int accel, int error)

**Returns *1* if the acceleration is *accel* within tolerance of *error*, otherwise *0*.**



# Change Parameters Commands

## Velocity setting

void Spur_set_vel(int vel)

**Set cruising speed with *vel.***

void Spur_set_ang_vel(int vel)

**Set angular speed with *vel.***

## Acceleration setting

void Spur_set_accel(int accel)

**Set acceleration with *accel.***

void Spur_set_ang_accel(int alpha)

**Set angular acceleration with *alpha.***

## Coordinate system setting

void Spur_set_pos_GL(int x, int y, int th)

void Spur_set_pos_LC(int x, int y, int th)

**Tell position and heading direction (*x,y,th*) to the robot.**

void Spur_set_LC_on_GL(int x, int y, int th)

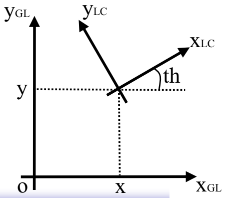
void Spur_set_LC_on_LC(int x, int y, int th)

void Spur_set_GL_on_GL(int x, int y, int th)

**Set coordinate system of the former on the latter coordinate system at (*x,y,th*).**

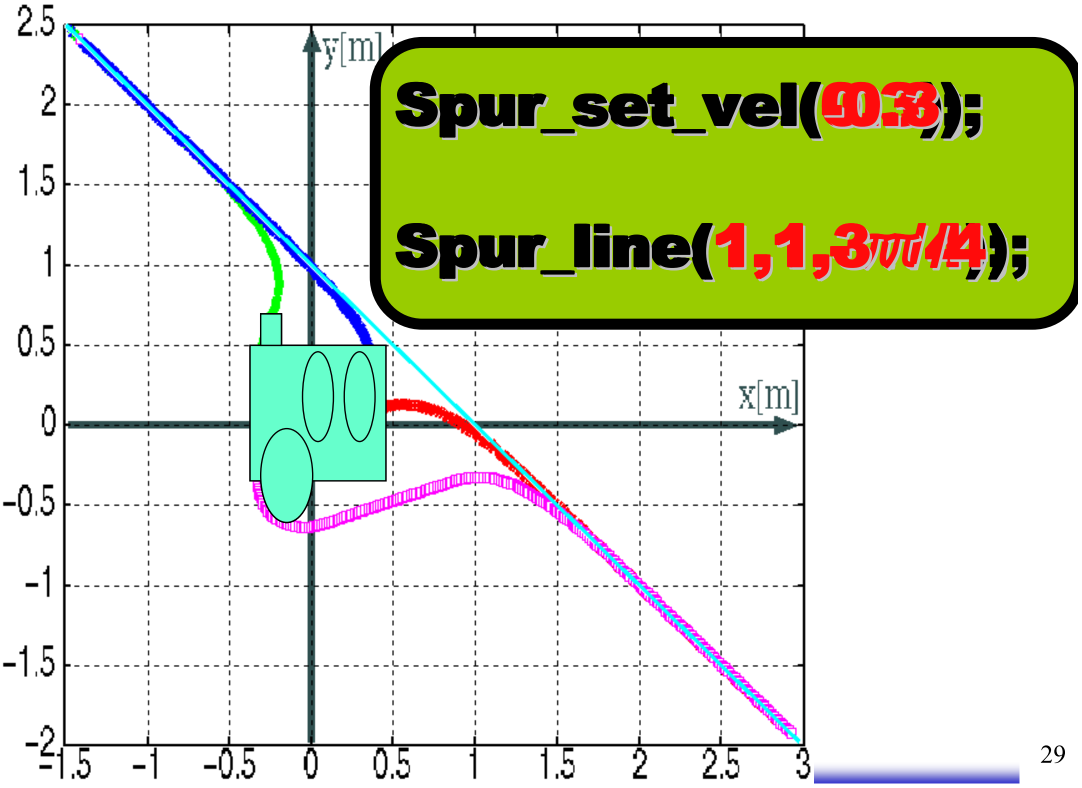
void Spur_adjust_pos_GL(int x, int y, int th)
void Spur_adjust_pos_LC(int x, int y, int th)
void Spur_adjust_pos_FS(int x, int y, int th)

**Move coordinate system to represent the robot position at (*x,y,th*).**

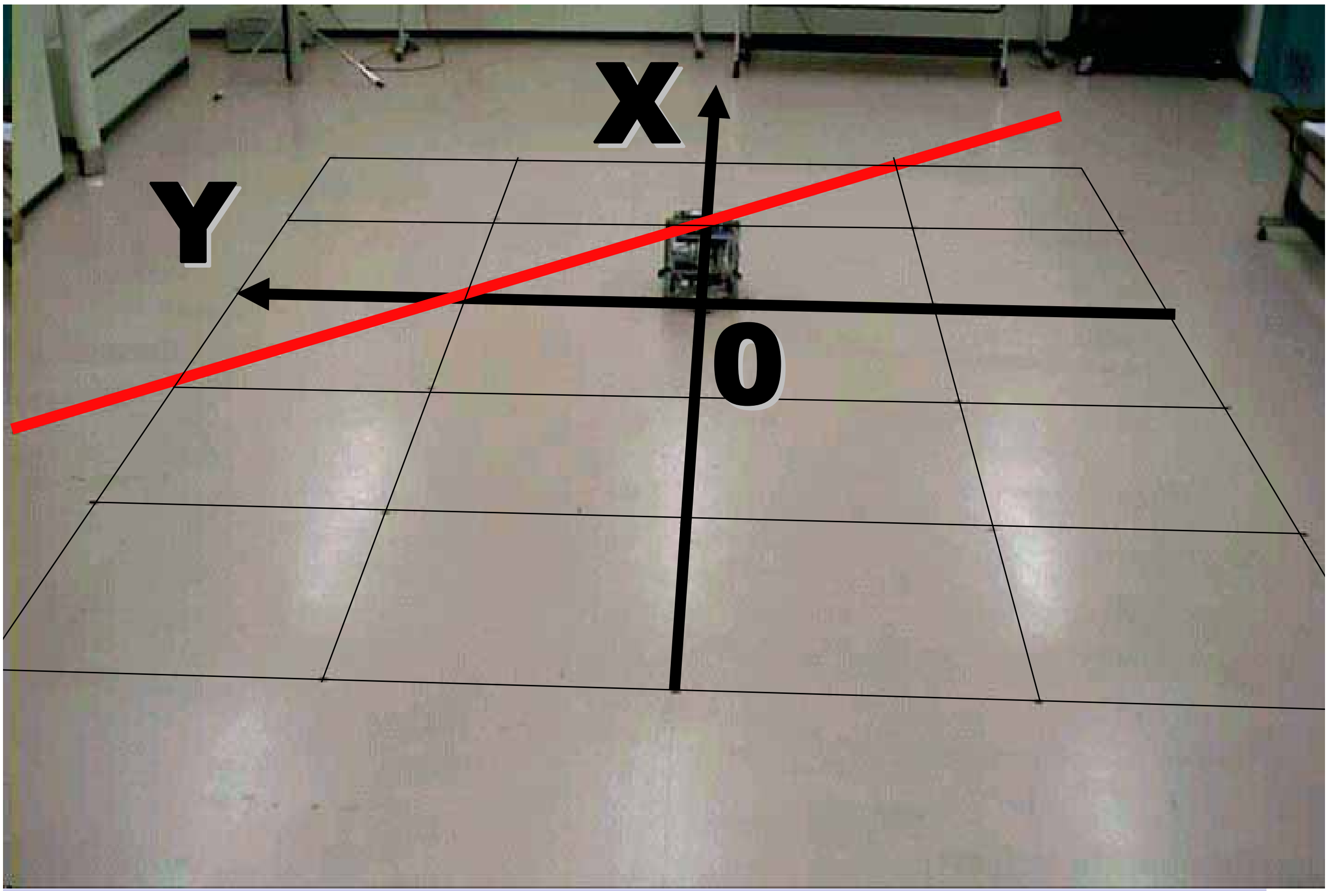


# Examples

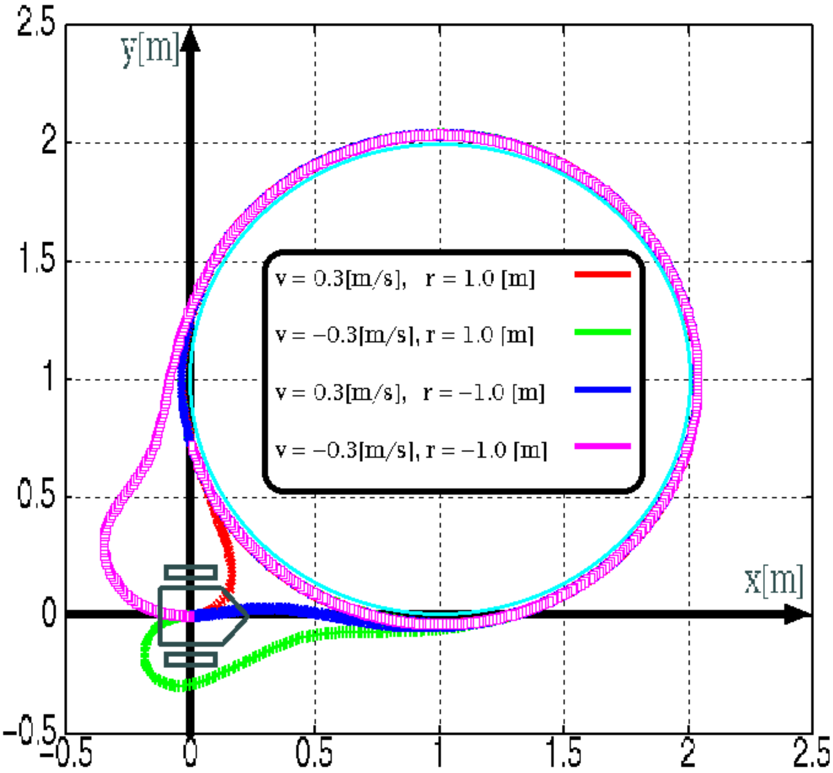
# Go along a line [Spur_line_GL()]

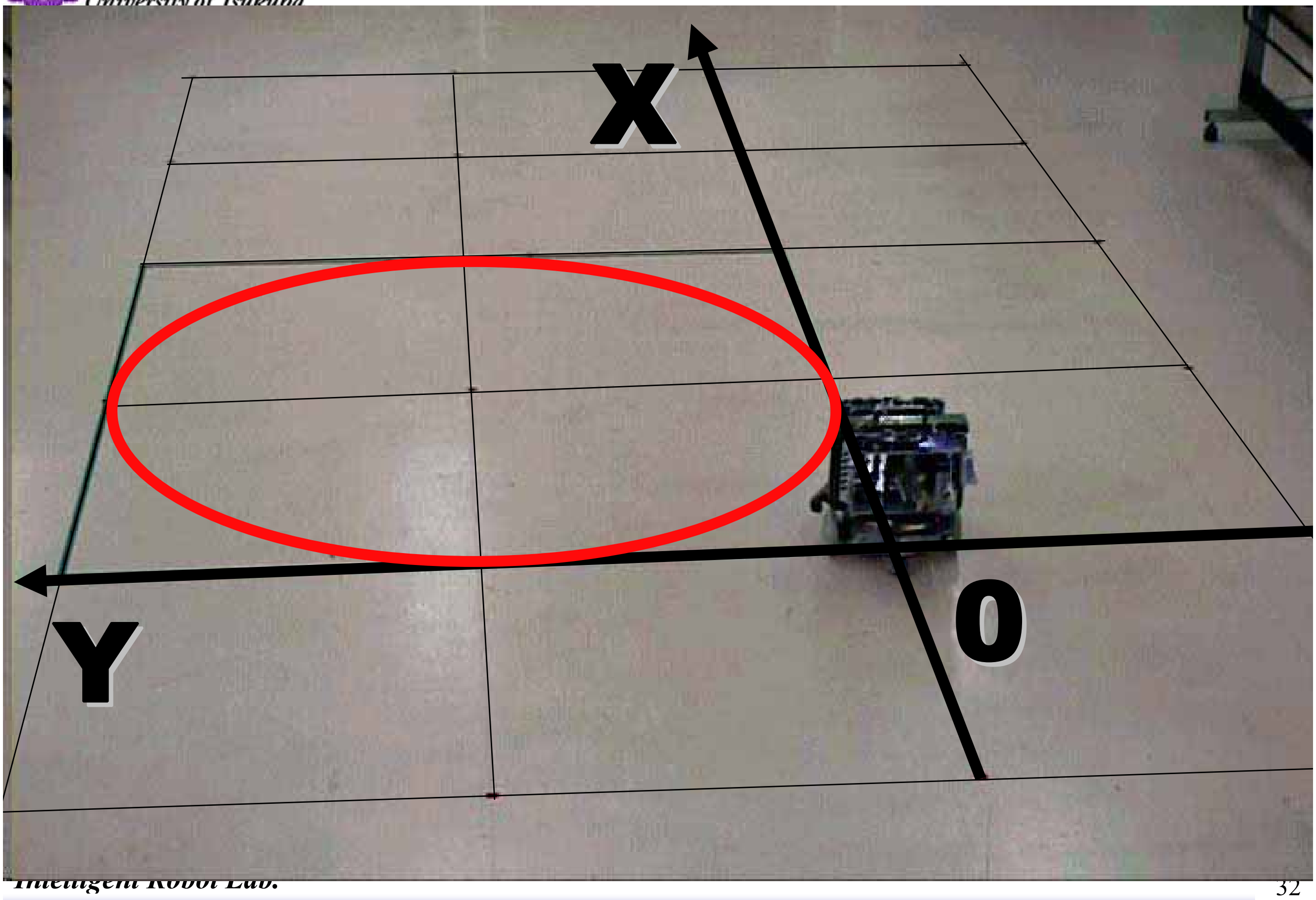


```
Spur_set_vel(0.3);

Spur_line(1,1,3π/4);
```

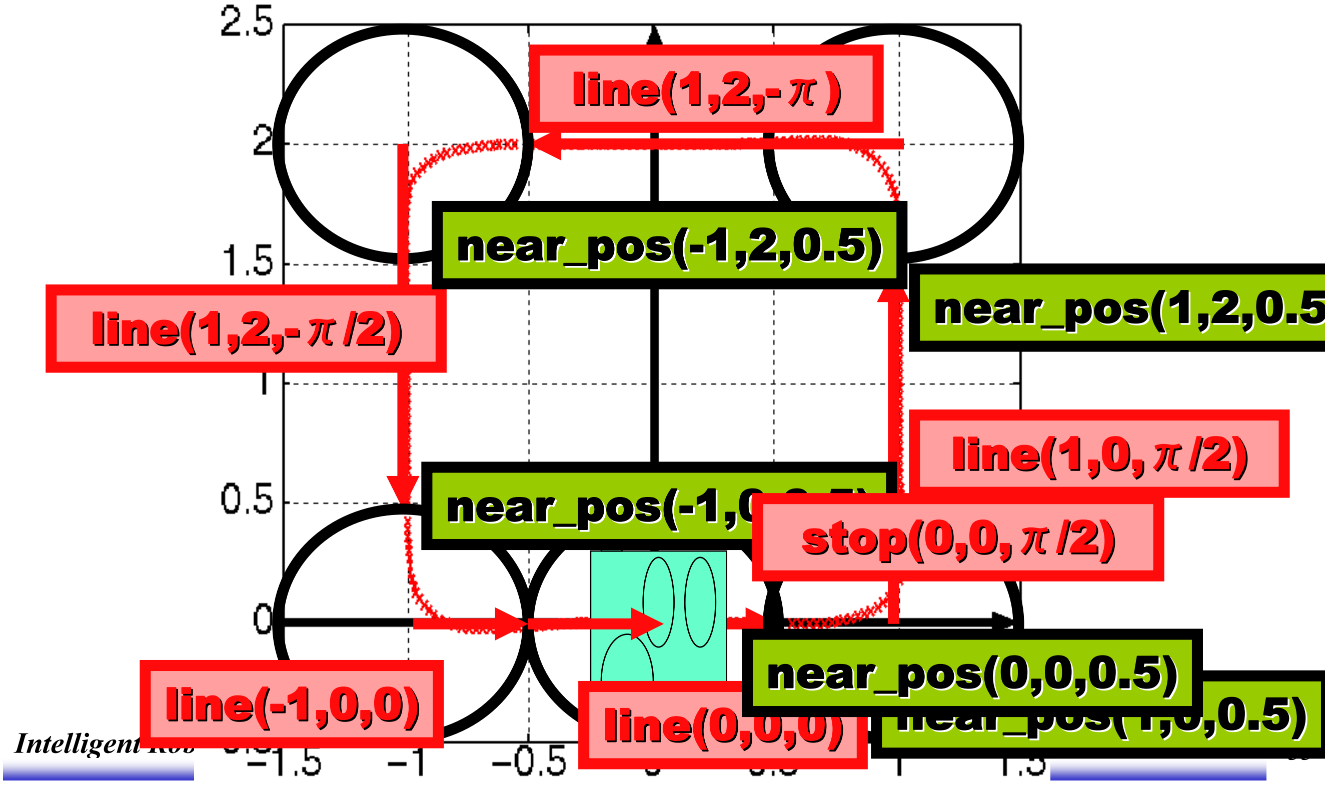


# Go along an Arc [Spur_arc_c_GL()]





# Draw Square Path



line(1,2,-π)

near_pos(-1,2,0.5)

near_pos(1,2,0.5

line(1,2,-π/2)

line(1,0,π/2)

near_pos(-1,0

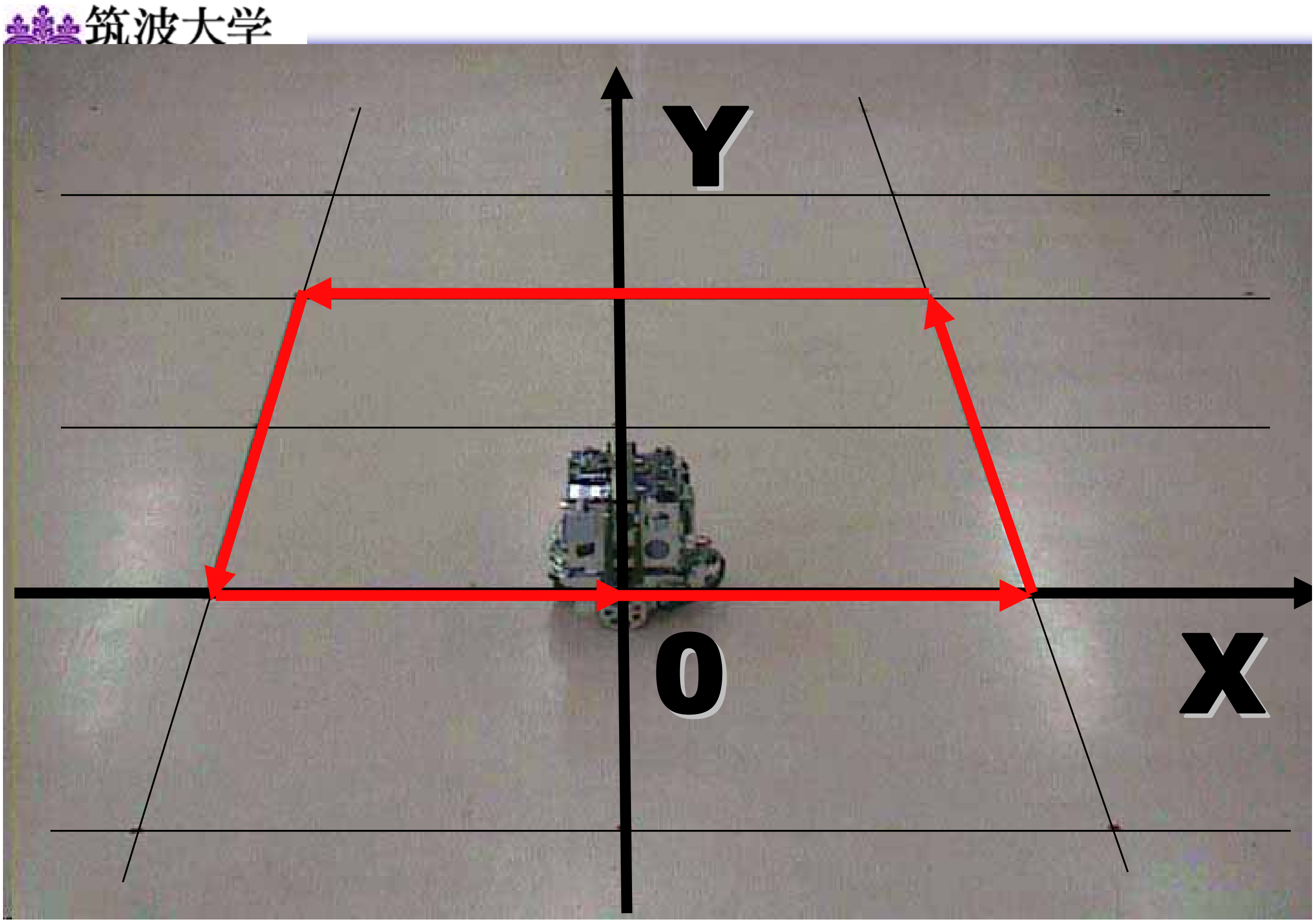
stop(0,0,π/2)

near_pos(0,0,0.5)

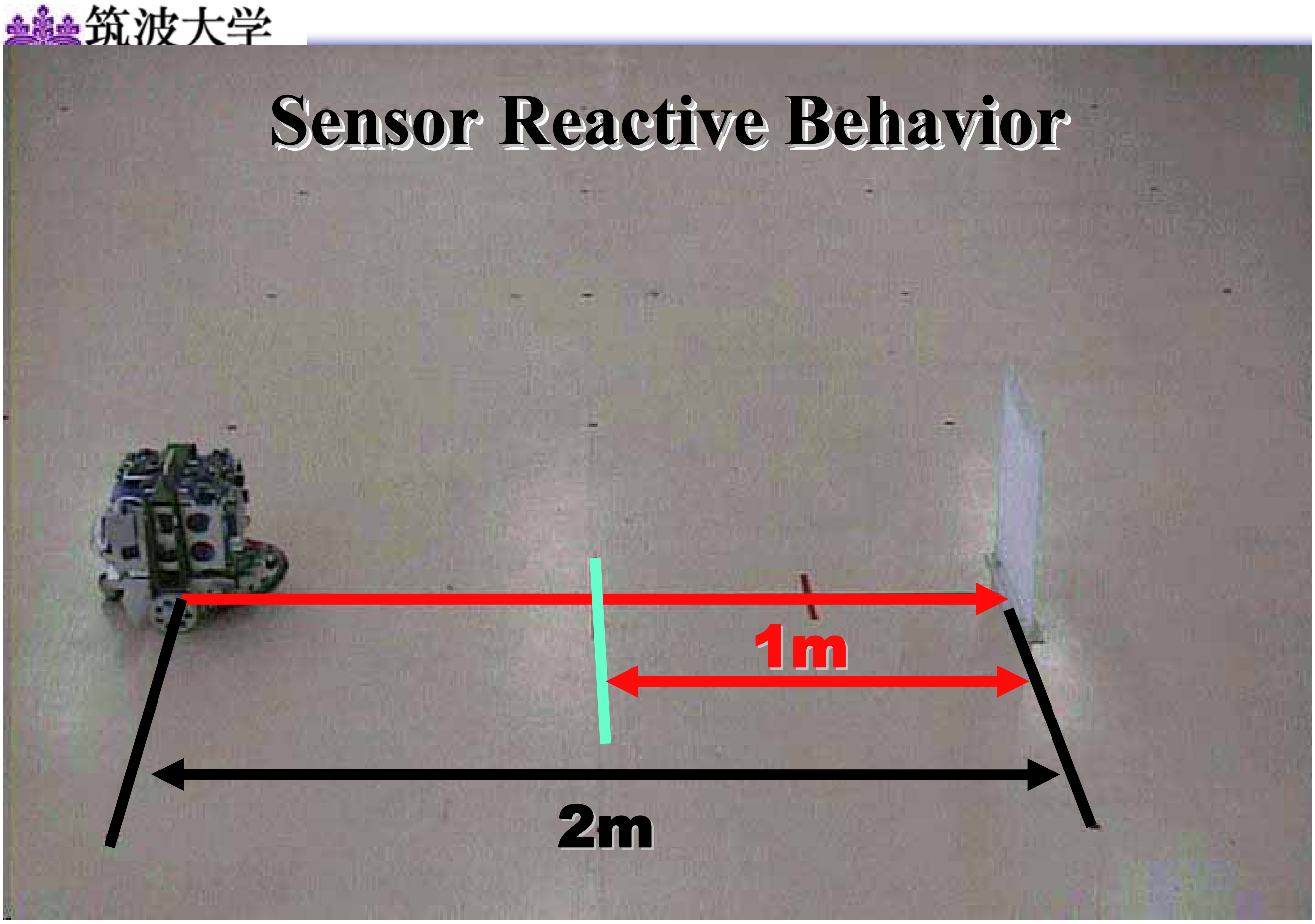
line(-1,0,0)

line(0,0,0)

near_pos(1,0,0.5)



# Sensor Reactive Behavior



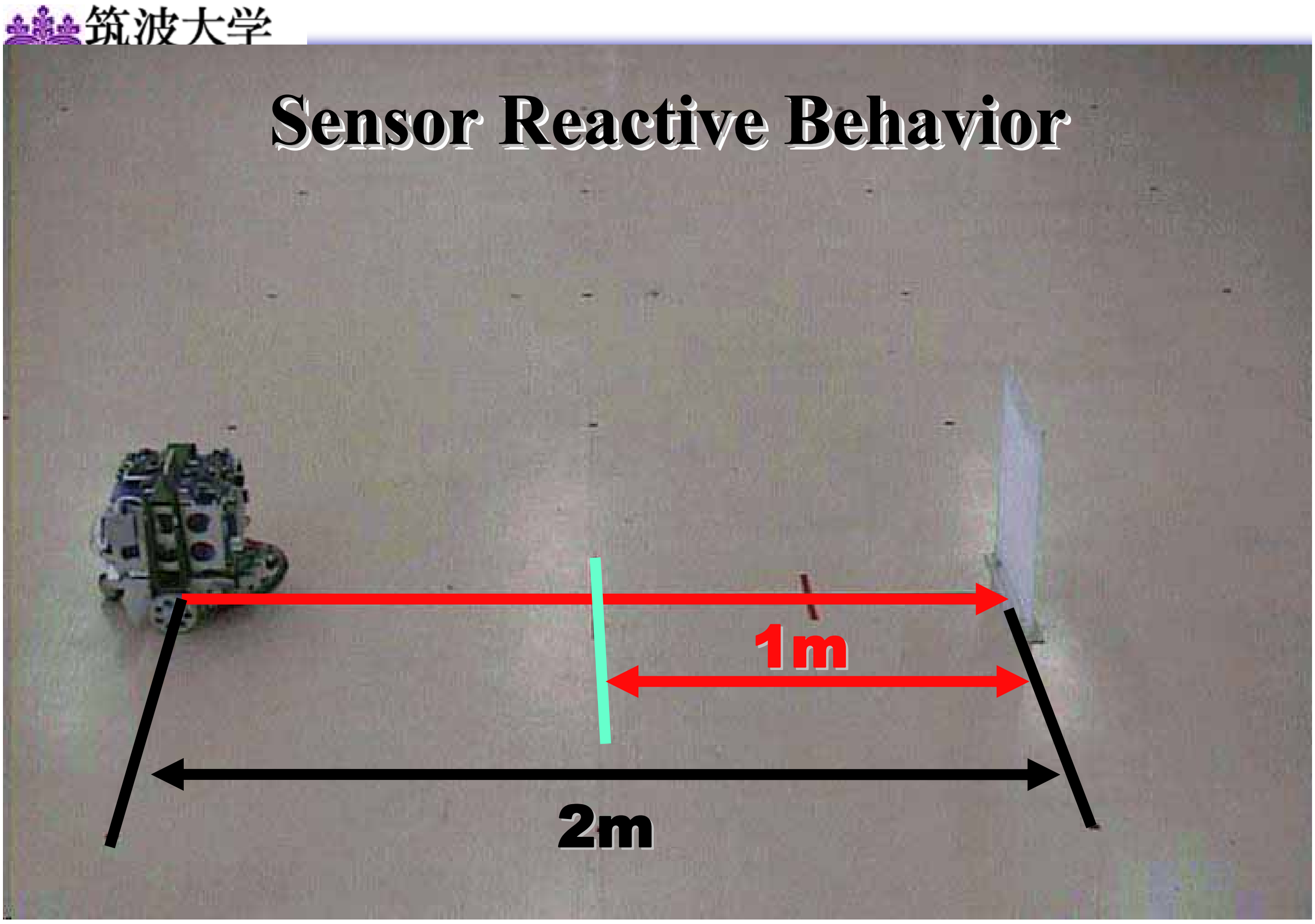
# Sensor Reactive Behavior
## (Example Code in C-like)

```
#define USAlartDist 1000 /* in mm */
#define StopXCoord 2000 /* in mm */
#define StopYCoord 0
#define StopWithinRadius 300 /* in mm */

Spur_line_GL(0,0,0);
for(;;) {
     if(US_dist(FRONT) <= USAlartDist) {
        Spur_Stop_Q();  break;
     }
     if(Spur_near_pos_GL(StopXCoord,StopYCoord,StopWithinRadius))
        Spur_Stop_Q();  break;
     }
     usleep(100); /* sleep 100ms */
}
```

*Go along line!*

*Monitor Event and Do Action*

# Sensor Reactive Behavior



# Servo Controller

- **Simple linear state feedback based on Vehicle Kinematics**
  - **Displacement from desired path (*dx,dy,dth*)**

    **-> Reference body velocity (*v, w*)**
- **Feedforward Dynamics Compensator**
  - **Reference body velocity (*v,w*)**

    **-> dynamics compensator**

    **-> wheel angular velocity ($w_R$,$w_L$)**

# Conclusions

- Philosophy and background of Spur
- Command System of Spur

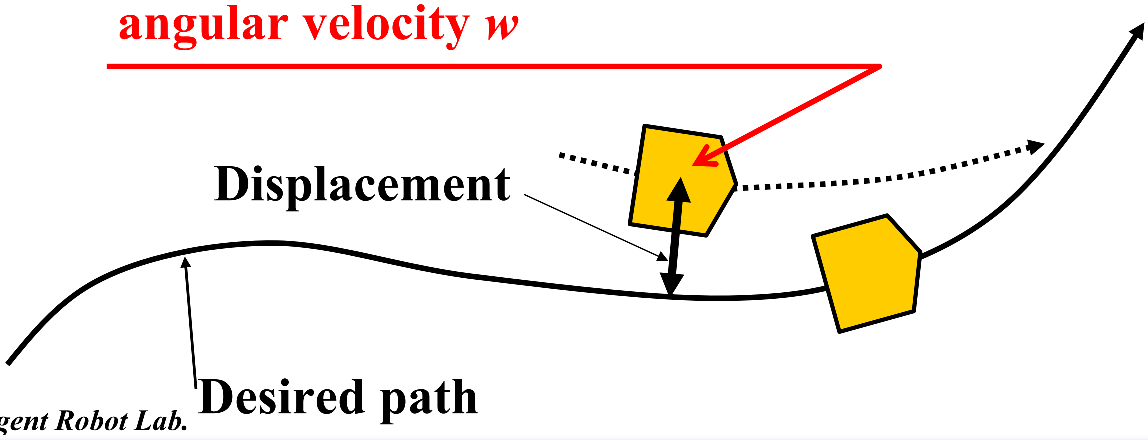
- Hope to be a reference plan to develop a standard specification

# Acknowledgement

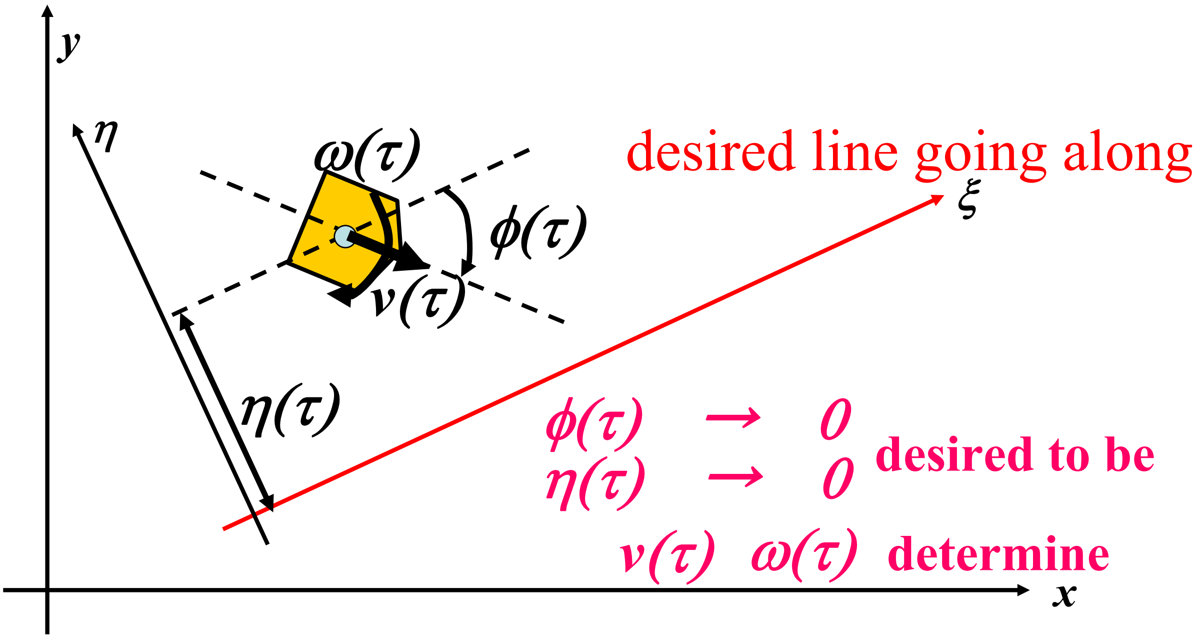
- Special thanks to
  - Prof. Shin’ichi Yuta for presentation permission
  - Mr. Akira Suzukawa for video session.
  - Mr. Kimitoshi Yamazaki for command list

# Steering based on Vehicle Kinematics

- Displacement from desired path
- In order to go along and ride on the path
  - to head toward tangent direction of the path,
  - determine longitudinal velocity *v*, and angular velocity *w*



# Go along a line



desired line going along

$\phi(\tau) \rightarrow 0$
$\eta(\tau) \rightarrow 0$ desired to be

$v(\tau)$ $\omega(\tau)$ determine

# Robot Modeling Framework

## -- Demonstration --

Robotics Domain Task Force
OMG Technical Meeting, Boston
June 2006

Abheek Kumar Bose
Robotics Engineer, ADA Robotics
ADA Software Group

robotics/2006-06-17

# Demonstration Objectives

- Achieve a starting point for the RoboticTools Work Group
- Gather ideas of how tools and robot profiles (Profiles WG) can be closely coupled
- Invite other interested members to join, collaborate & contribute towards the development of such robotic tools
- Feedback from the OMG members to achieve higher standards for robot tools
- Analyze and compare other known modeling tools to obtain a "capability list" and adapt these features to the robotics domain

# RMF Overview

- Based on the following Eclipse Technologies
  - Eclipse Modeling Framework (EMF – www.eclipse.org/emf)
  - Graphical Editing Framework (GEF – www.elipse.org/gef)
  - Eclipse Plugin Architecture – extension and extension points
- Eclipse Modeling Framework
  - Provides a metamodel of any robotic system
  - EMF currently allows model creation through:
    - Rational Rose (*.mdl)
    - XSD
    - Annotated Java
    - Eclipse Ecore Models
  - Code Generation Capabilities
- Graphical Editing Framework
  - Provides a Graphical Editor
  - Model elements are represented visually
  - Automatic visual updates when model is changed
- Extension & Extension Points
  - Each model element is realized as an external plugin (component)
  - Plugins are classified under Sensors, Processors & Actuators
  - Components combine to create plugins – plugins create plugins

# RMF Overview

- Based on the following Eclipse Technologies
  - Eclipse Modeling Framework (EMF – www.eclipse.org/emf)
  - Graphical Editing Framework (GEF – www.elipse.org/gef)
  - Eclipse Plugin Architecture – extension and extension points
- Eclipse Modeling Framework
  - Provides a metamodel of any robotic system
  - EMF currently allows model creation through:
    - Rational Rose (*.mdl)
    - XSD
    - Annotated Java
    - Eclipse Ecore Models
  - Code Generation Capabilities
- Graphical Editing Framework
  - Provides a Graphical Editor
  - Model elements are represented visually
  - Automatic visual updates when model is changed
- Extension & Extension Points
  - Each model element is realized as an external plugin (component)
  - Plugins are classified under Sensors, Processors & Actuators
  - Components combine to create plugins – plugins create plugins

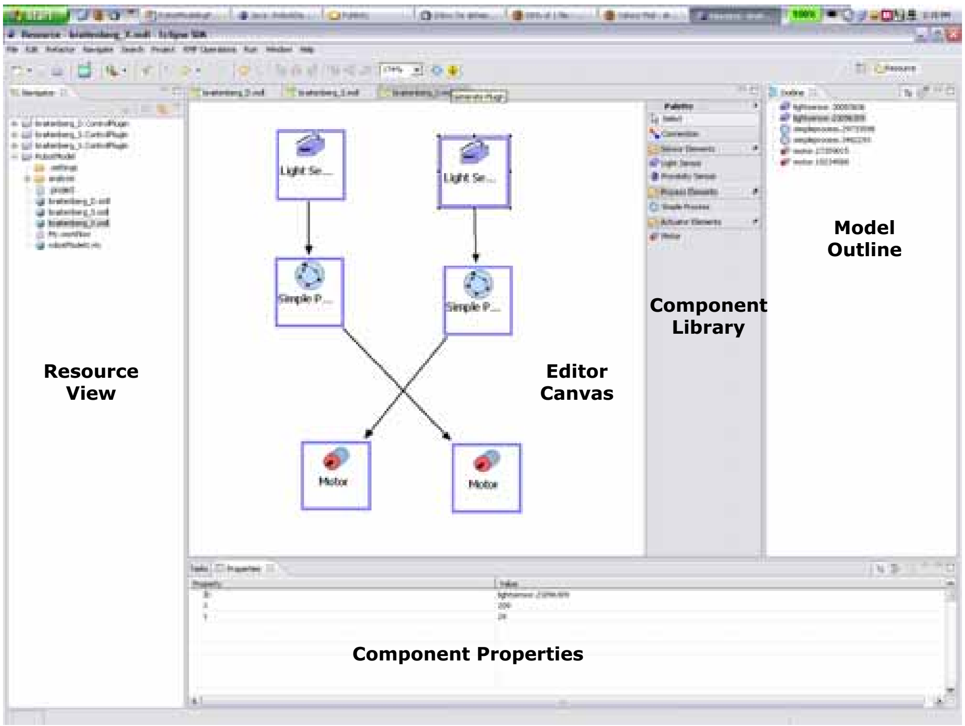
# RMF Overview

- Based on the following Eclipse Technologies
  - Eclipse Modeling Framework (EMF – www.eclipse.org/emf)
  - Graphical Editing Framework (GEF – www.elipse.org/gef)
  - Eclipse Plugin Architecture – extension and extension points
- Eclipse Modeling Framework
  - Provides a metamodel of any robotic system
  - EMF currently allows model creation through:
    - Rational Rose (*.mdl)
    - XSD
    - Annotated Java
    - Eclipse Ecore Models
  - Code Generation Capabilities
- Graphical Editing Framework
  - Provides a Graphical Editor
  - Model elements are represented visually
  - Automatic visual updates when model is changed
- Extension & Extension Points
  - Each model element is realized as an external plugin (component)
  - Plugins are classified under Sensors, Processors & Actuators
  - Components combine to create plugins – plugins create plugins

# RMF Overview

- Based on the following Eclipse Technologies
  - Eclipse Modeling Framework (EMF – www.eclipse.org/emf)
  - Graphical Editing Framework (GEF – www.elipse.org/gef)
  - Eclipse Plugin Architecture – extension and extension points
- Eclipse Modeling Framework
  - Provides a metamodel of any robotic system
  - EMF currently allows model creation through:
    - Rational Rose (*.mdl)
    - XSD
    - Annotated Java
    - Eclipse Ecore Models
  - Code Generation Capabilities
- Graphical Editing Framework
  - Provides a Graphical Editor
  - Model elements are represented visually
  - Automatic visual updates when model is changed
- Extension & Extension Points
  - Each model element is realized as an external plugin (component)
  - Plugins are classified under Sensors, Processors & Actuators
  - Components combine to create plugins – plugins create plugins

# RMF Environment



Resource View

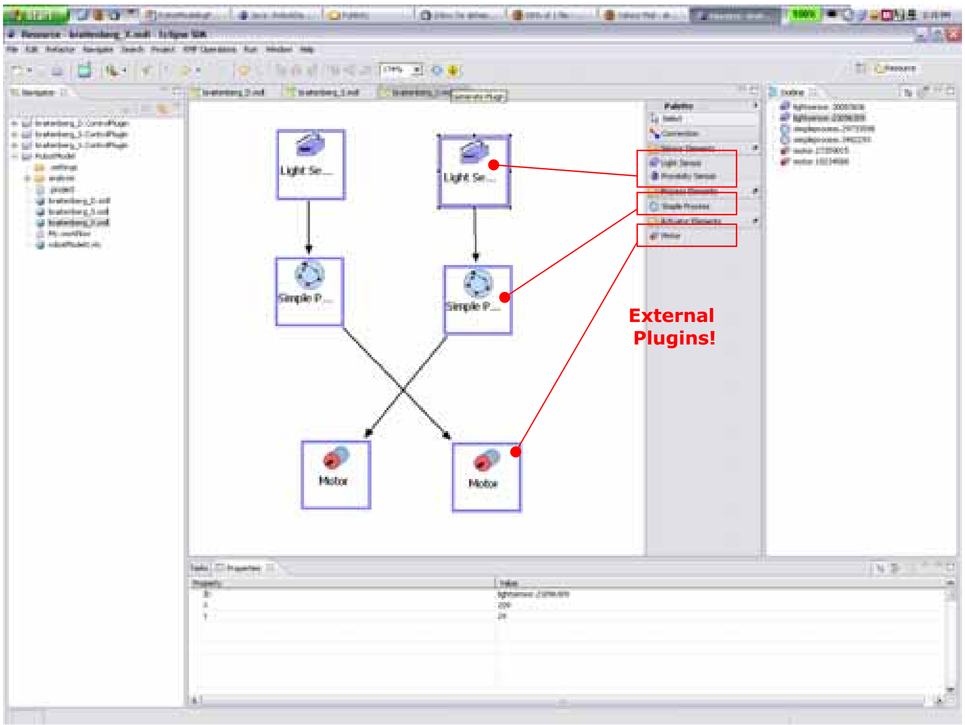
Editor Canvas

Component Library

Model Outline

Component Properties

# RMF Environment



External Plugins!

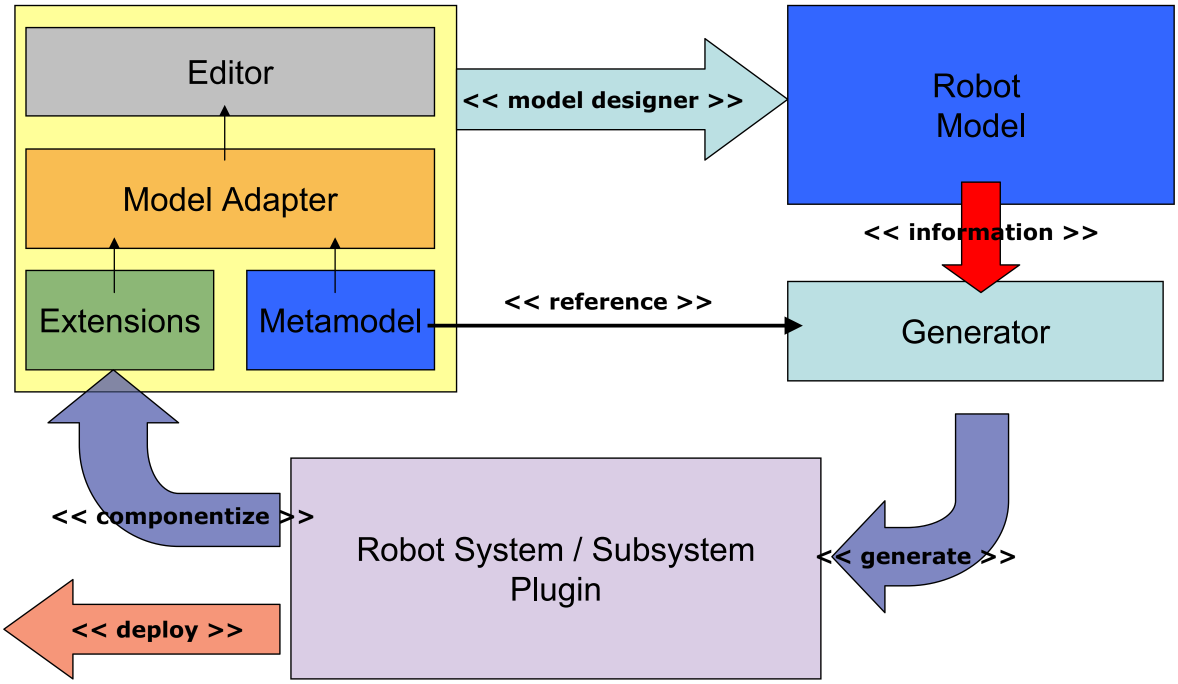
# RMF Generated Model

```
<com.adasoftware.rmf.robotmetamodel.models:RobotDiagram xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:com.adasoftware.rmf.robotmetamodel.models="http:///com/adasoftware/rmf/robotmetamodel/models.ecore"
zoom="1.7440633245382586">
<robotElements xsi:type="com.adasoftware.rmf.robotmetamodel.models:SensorElement" x="55" y="18"
iD="lightsensor.30083606" smallImageURL="URLImageDescriptor(bundleentry://255/icons/lightSensor16.gif)"
largeImageURL="URLImageDescriptor(bundleentry://255/icons/lightSensor24.gif)" label="Light Sensor"
className="com.adasoftware.rmf.sensors.lightsensor.LightSensorExtension"
referenceID="com.adasoftware.rmf.sensors.lightsensor">
  <sourceConnections source="lightsensor.30083606" target="simpleprocess.29733598" />
</robotElements>
<robotElements xsi:type="com.adasoftware.rmf.robotmetamodel.models:SensorElement" x="209" y="29"
iD="lightsensor.23096309" smallImageURL="URLImageDescriptor(bundleentry://255/icons/lightSensor16.gif)"
largeImageURL="URLImageDescriptor(bundleentry://255/icons/lightSensor24.gif)" label="Light Sensor"
className="com.adasoftware.rmf.sensors.lightsensor.LightSensorExtension"
referenceID="com.adasoftware.rmf.sensors.lightsensor">
  <sourceConnections source="lightsensor.23096309" target="simpleprocess.3482293" />
</robotElements>
<robotElements xsi:type="com.adasoftware.rmf.robotmetamodel.models:ProcessElement" x="71" y="129"
targetConnections="//@robotElements.0/@sourceConnections.0" iD="simpleprocess.29733598"
smallImageURL="URLImageDescriptor(bundleentry://252/icons/simpleProcess16.gif)"
largeImageURL="URLImageDescriptor(bundleentry://252/icons/simpleProcess24.gif)" label="Simple Process"
className="com.adasoftware.rmf.process.simpleprocess.SimpleProcessExtension" connectionMode="2"
referenceID="com.adasoftware.rmf.process.simpleprocess">
  <sourceConnections source="simpleprocess.29733598" target="motor.18234588" />
</robotElements>
…
…
…
…
```

# RMF Generated Model

```
<com.adasoftware.rmf.robotmetamodel.models:RobotDiagram xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:com.adasoftware.rmf.robotmetamodel.models="http:///com/adasoftware/rmf/robotmetamodel/models.ecore"
zoom="1.7440633245382586">
<robotElements xsi:type="com.adasoftware.rmf.robotmetamodel.models:SensorElement" x="55" y="18"
iD="lightsensor.30083606" smallImageURL="URLImageDescriptor(bundleentry://255/icons/lightSensor16.gif)"
largeImageURL="URLImageDescriptor(bundleentry://255/icons/lightSensor24.gif)" label="Light Sensor"
className="com.adasoftware.rmf.sensors.lightsensor.LightSensorExtension"
referenceID="com.adasoftware.rmf.sensors.lightsensor">
  <sourceConnections source="lightsensor.30083606" target="simpleprocess.29733598" />
</robotElements>
<robotElements xsi:type="com.adasoftware.rmf.robotmetamodel.models:SensorElement" x="209" y="29"
iD="lightsensor.23096309" smallImageURL="URLImageDescriptor(bundleentry://255/icons/lightSensor16.gif)"
largeImageURL="URLImageDescriptor(bundleentry://255/icons/lightSensor24.gif)" label="Light Sensor"
className="com.adasoftware.rmf.sensors.lightsensor.LightSensorExtension"
referenceID="com.adasoftware.rmf.sensors.lightsensor">
  <sourceConnections source="lightsensor.23096309" target="simpleprocess.3482293" />
</robotElements>
<robotElements xsi:type="com.adasoftware.rmf.robotmetamodel.models:ProcessElement" x="71" y="129"
targetConnections="//@robotElements.0/@sourceConnections.0" iD="simpleprocess.29733598"
smallImageURL="URLImageDescriptor(bundleentry://252/icons/simpleProcess16.gif)"
largeImageURL="URLImageDescriptor(bundleentry://252/icons/simpleProcess24.gif)" label="Simple Process"
className="com.adasoftware.rmf.process.simpleprocess.SimpleProcessExtension" connectionMode="2"
referenceID="com.adasoftware.rmf.process.simpleprocess">
  <sourceConnections source="simpleprocess.29733598" target="motor.18234588" />
</robotElements>
…
…
…
…
```

# RMF Concept



# RMF Features

- Graphical Editor
  - Ease of use and better model representation
- Extendible model library via external plugins
  - Internal features are encapsulated
  - Any plugin contributes to the modeling framework
- Model Serialization using XMI
  - Possible model interchange with other modeling tools
- Recursive modeling
  - Plugins create plugins
  - Models create models
- Code Generation
  - Automatic Control Code Generation
  - Complexity of control code realized through components and their connections

# RMF Wish-list

- Editor Generation
  - Generate the Graphical Editor depending on the component metamodel
- Integrate to Eclipse Graphical Modeling Framework
  - Eclipse GMF acts as a bridge between GEF and EMF
  - Generic Modeling Possible
- Algorithm and Intelligence Modeling
  - Currently RMF models only the component
  - Algorithms and component intelligence is realized via code
- Your contributions…
  - Look into the infrastructure group – there may NOT be a need of a Tools WG
  - Mission Statement

# RMF Demo

- Robot Modeling Framework
  - Modeling
  - Code Generation
- CORBA Communication Framework
  - Control Client
  - Executes Controller Plugins
  - Communication via CORBA
- Robot Simulator
  - Java based Simulation
  - Uses Open Dynamics Engine
  - Integrated CORBA Server
  - Supports XML based robot models

eclipse

Robot Modeling Framework

Thank You!

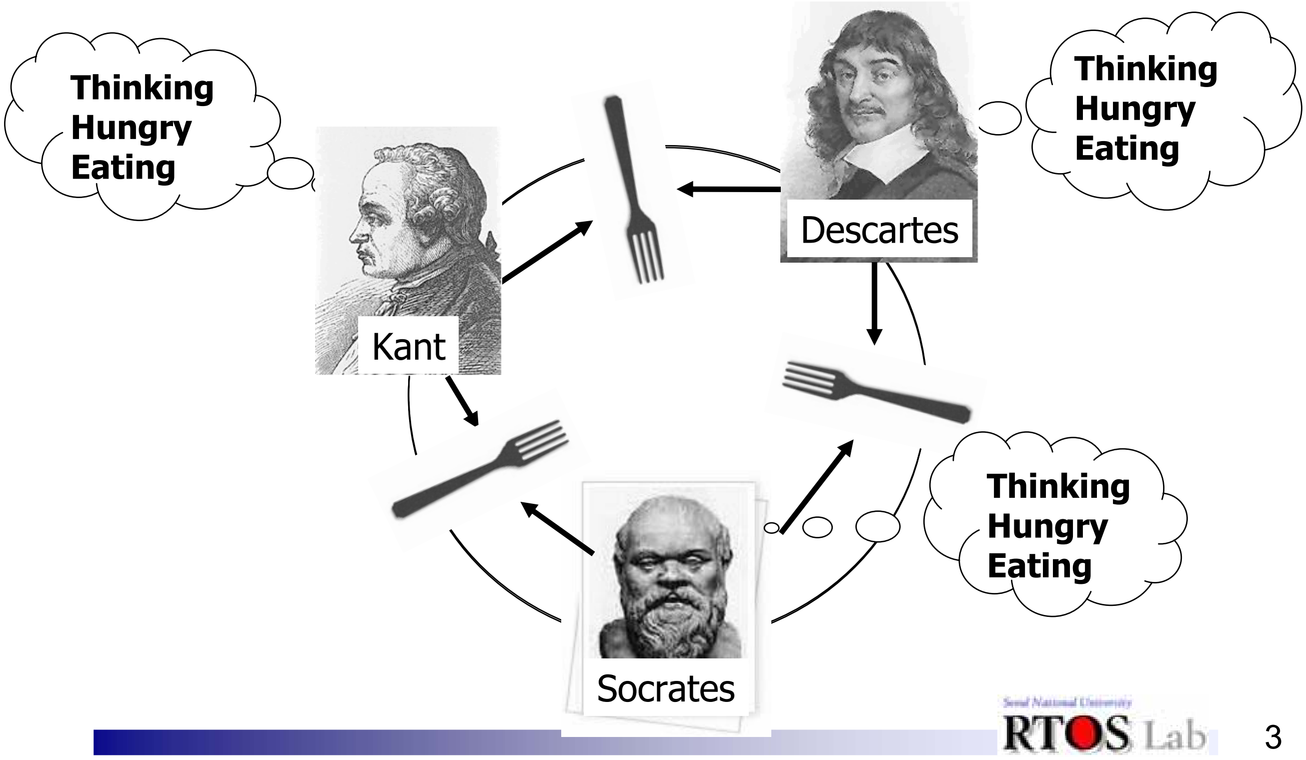
# Understanding RSCA with Example

*May 2. 2006*

Prof. Seongsoo Hong
Real-Time Operating Systems Lab
Seoul National University
http://redwood.snu.ac.kr

# Dining Philosopher Problem

# The Dining Philosopher Problem



Thinking
Hungry
Eating

Kant

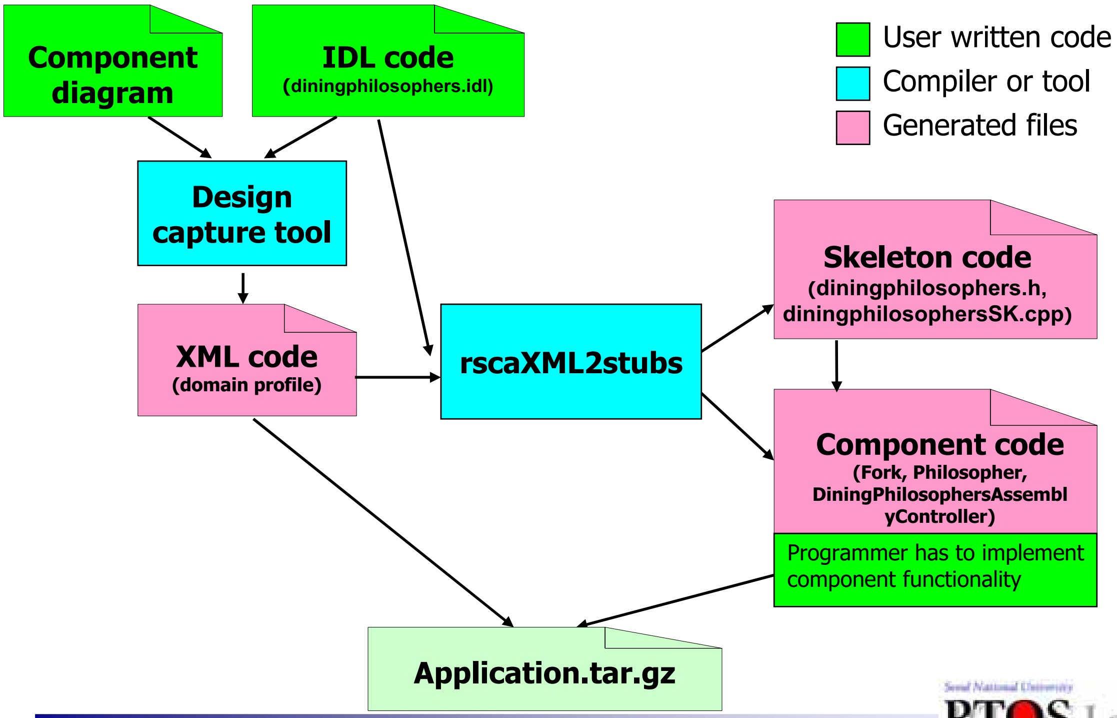
Descartes

Thinking
Hungry
Eating

Socrates

Thinking
Hungry
Eating

# Application Programming Process



Component diagram

IDL code (diningphilosophers.idl)

🟩 User written code
🟦 Compiler or tool
🟪 Generated files

Design capture tool

XML code (domain profile)

rscaXML2stubs

Skeleton code (diningphilosophers.h, diningphilosophersSK.cpp)

Component code (Fork, Philosopher, DiningPhilosophersAssemblyController)

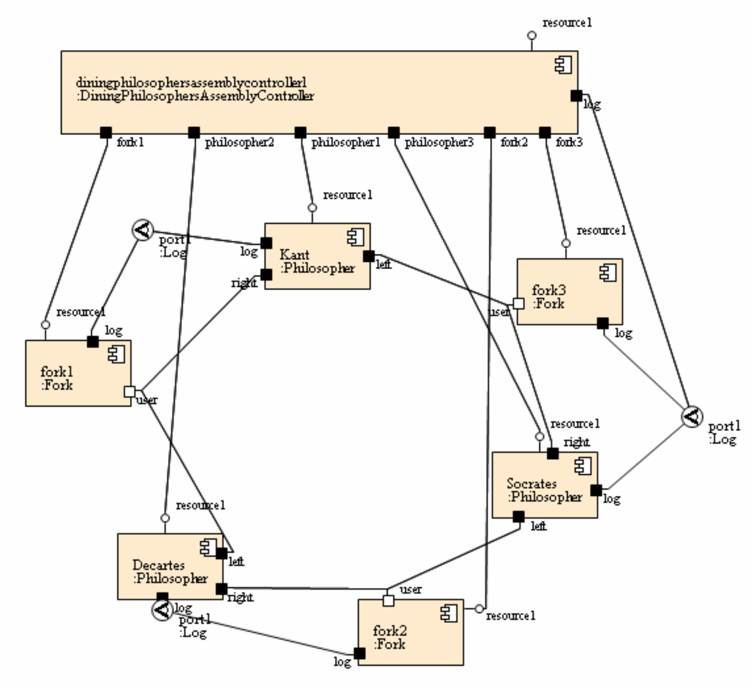
Programmer has to implement component functionality

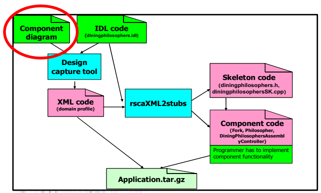
Application.tar.gz

# Key Observation: Application Programmers Focus only on Business Logic!

# 1. Design Component Diagram and Write IDL Code

# Component Diagram





Component diagram

IDL code (diningphilosophers.idl)

Design capture tool

XML code (domain profile)

rscaXML2stubs

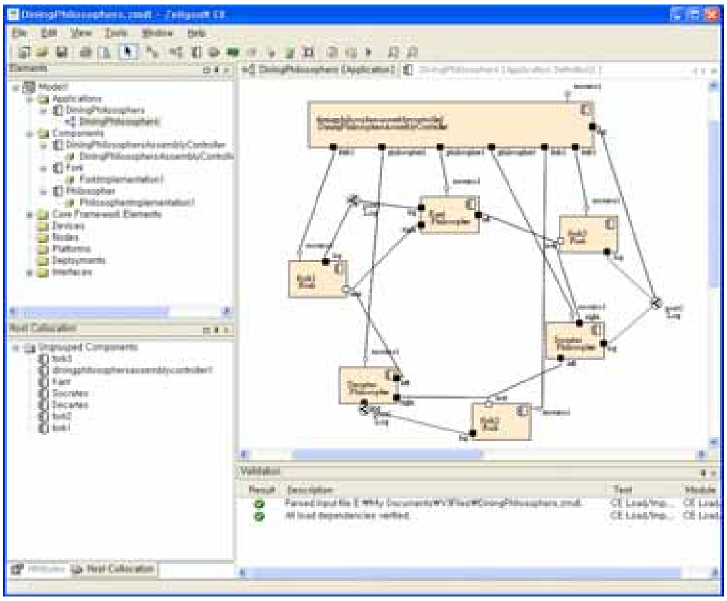
Skeleton code (diningphilosophers.h, diningphilosophersSK.cpp)

Component code (Fork, Philosopher, DiningPhilosophersAssemblyController)

Programmer has to implement component functionality

Application.tar.gz

# Design Capture by Zeligsoft Tool

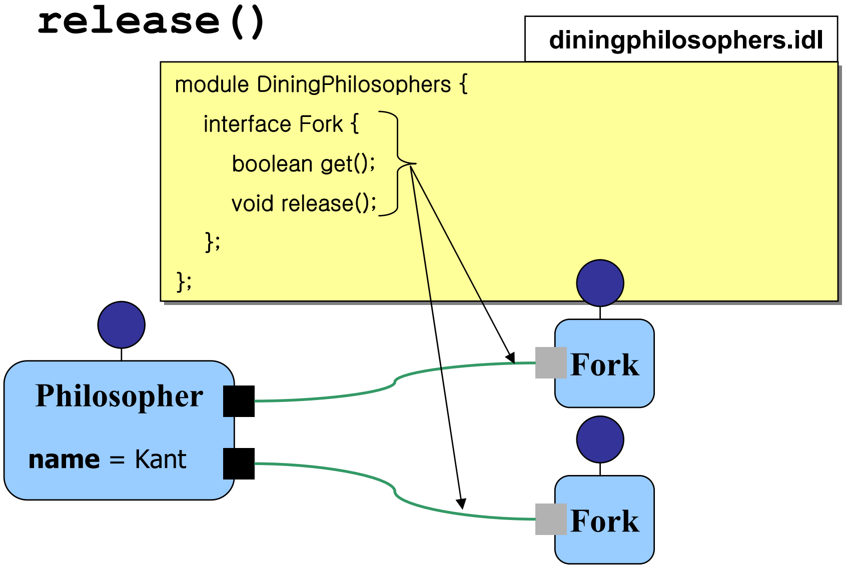


Parsed input file E:₩My Documents₩VFiles₩DiningPhilosophers.zmdl. CE Load/Imp... CE Load

All load dependencies verified. CE Load/Imp... CE Load

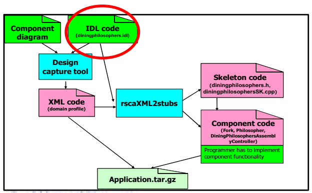
# IDL Code

- Philosophers get or release forks through the "Fork" interface which defines two operations **`get()`** and **`release()`**



diningphilosophers.idl

```
module DiningPhilosophers {
    interface Fork {
        boolean get();
        void release();
    };
};
```

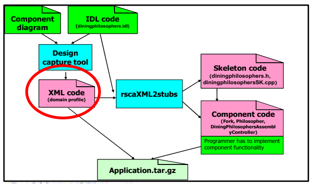


Component diagram
IDL code
(diningphilosophers.idl)
Design capture tool
XML code
(domain profile)
rscaXML2stubs
Skeleton code
(diningphilosophers.h, diningphilosophersSK.cpp)
Component code
(Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# 2. Generate Domain Profiles

# Domain Profile (1)

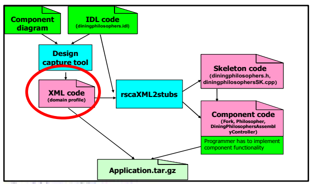
- Domain profile (XML code) is generated by Zeligsoft tool
  - Software Assembly Descriptors (SAD)
    - DiningPhilosophers.sad.xml
  - Software Package Descriptors (SPD)
    - DiningPhilosopherAssemblyController.spd.xml
    - Philosopher.spd.xml
    - Fork.spd.xml
  - Software Component Descriptors (SCD)
    - DiningPhilosopherAssemblyController.scd.xml
    - Philosopher.scd.xml
    - Fork.scd.xml


Component diagram
IDL code
(diningphilosophers.idl)
Design capture tool
XML code
(domain profile)
rscaXML2stubs
Skeleton code
(diningphilosophers.h, diningphilosophersSK.cpp)
Component code
(Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Domain Profile (2)

- Software Assembly Descriptors (DiningPhilosophers.sad.xml)
  - Instantiate three philosophers Kant, Socrates, and Decartes

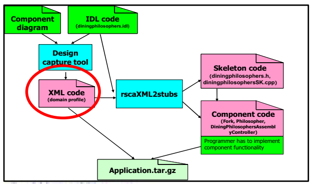
```
<componentplacement>
  <componentfileref refid="PhilosopherFile_cea2f677-547f-4789-b638-309eb8ad094a" />
  <!--[Component Kant]-->
  <componentinstantiation id="DCE:40c473c4-e06e-480d-af08-3db05bf765d1">
    <usagename>Kant</usagename>
  </componentinstantiation>
  <!--[Component Socrates]-->
  <componentinstantiation id="DCE:6e94e9c1-1cef-40a1-8495-2860266e1e68">
    <usagename>Socrates</usagename>
  </componentinstantiation>
  <!--[Component Decartes]-->
  <componentinstantiation id="DCE:7120a7cd-1297-4797-9753-2458f1c91e0b">
    <usagename>Decartes</usagename>
  </componentinstantiation>
</componentplacement>
```


Component diagram
IDL code
(diningphilosophers.idl)
Design capture tool
XML code
(domain profile)
rscaXML2stubs
Skeleton code
(diningphilosophers.h, diningphilosophersSK.cpp)
Component code
(Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Domain Profile (3)

- Software Assembly Descriptors (DiningPhilosophers.sad.xml)
  - Instantiate three forks fork1, fork2, and fork3

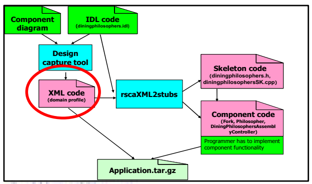
```
<componentplacement>
   <componentfileref refid="ForkFile_6fd64c73-d2e0-4cad-8ef1-503b7489e7b3" />
   <!--[Component fork3]-->
   <componentinstantiation id="DCE:24f44cb7-cea5-40aa-a017-48f6f16d4d96">
      <usagename>fork3</usagename>
   </componentinstantiation>
   <!--[Component fork2]-->
   <componentinstantiation id="DCE:96f5ad0e-e3f4-4834-ab8c-51e15522456b">
      <usagename>fork2</usagename>
   </componentinstantiation>
   <!--[Component fork1]-->
   <componentinstantiation id="DCE:c25c096c-0242-4766-adcd-dad0dd235e3c">
      <usagename>fork1</usagename>
   </componentinstantiation>
</componentplacement>
```


Component diagram
IDL code
(diningphilosophers.idl)
Design capture tool
XML code
(domain profile)
rscaXML2stubs
Skeleton code
(diningphilosophers.h, diningphilosophersSK.cpp)
Component code
(Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Domain Profile (4)

- Software Assembly Descriptors (DiningPhilosophers.sad.xml)
  - Philosophers try to get forks laid on left and right side
    - E.g.) The connection between Kant and fork3 enables Kant to get the fork3 with his left hand

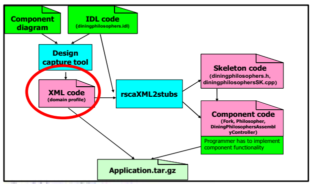
```
<connectinterface id="DCE:755c92bb-72ec-4310-a317-91193a27d566">
   <usesport>
      <usesidentifier>left</usesidentifier>
      <findby>
         <namingservice name="Kant_DCE:40c473c4-e06e-480d-af08-3db05bf765d1"/>
      </findby>
   </usesport>
   <providesport>
      <providesidentifier>user</providesidentifier>
       <findby>
         <namingservice name="fork3_DCE:24f44cb7-cea5-40aa-a017-48f6f16d4d96"/>
      </findby>
   </providesport>
</connectinterface>
```


Component diagram
IDL code
(diningphilosophers.idl)
Design capture tool
XML code
(domain profile)
rscaXML2stubs
Skeleton code
(diningphilosophers.h, diningphilosophersSK.cpp)
Component code
(Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Domain Profile (5)

- Software Package Descriptors
  - Each SPD describes various implementations of any given component
    - E.g.) Philosopher.spd.xml

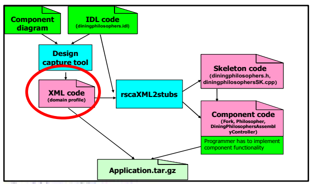
```
<!--[Implementation PhilosopherImplementation1]-->
<implementation id="DCE:fad01baf-cfff-4162-a027-42e2f3f3e9e5"
aepcompliance="aep_compliant">
  <code type="Executable">
    <localfile name="via-bin/PhilosopherServer" />
    <entrypoint>main</entrypoint>
  </code>
  <compiler name="gcc" version="3.3" />
  <os name="linux" version="2.4" />
  <processor name="x86" />
</implementation>
```


Component diagram
IDL code (diningphilosophers.idl)
Design capture tool
XML code (domain profile)
rscaXML2stubs
Skeleton code (diningphilosophers.h, diningphilosophersSK.cpp)
Component code (Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

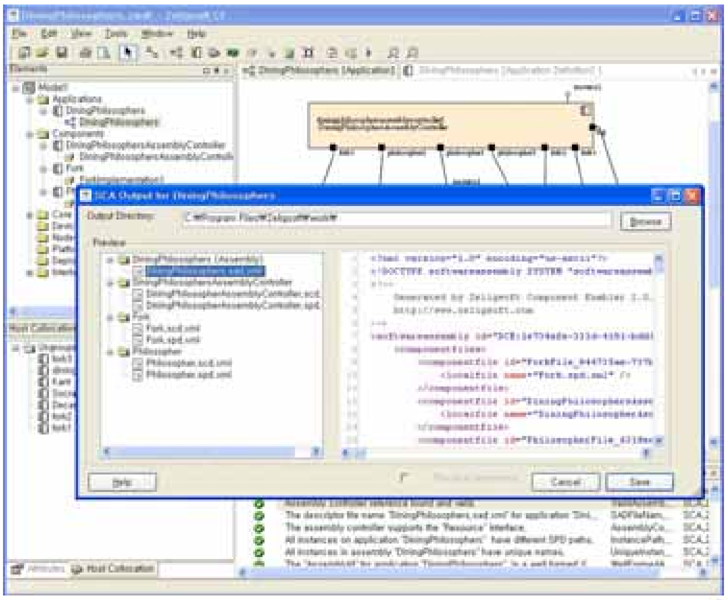
# Domain Profile (6)

- Software Component Descriptors
  - Each SCD describes component information such as CORBA version, component interfaces, and so on
    - E.g.) Philosopher.scd.xml

```
<componentfeatures>
  <supportsinterface repid="IDL:CF/Resource:1.0" supportsname="Resource" />
  <supportsinterface repid="IDL:CF/LifeCycle:1.0" supportsname="LifeCycle" />
  <supportsinterface repid="IDL:CF/PortSupplier:1.0" supportsname="PortSupplier" />
  <supportsinterface repid="IDL:CF/PropertySet:1.0" supportsname="PropertySet" />
  <supportsinterface repid="IDL:CF/TestableObject:1.0" supportsname="TestableObject" />
  <ports>
    <uses repid="IDL:LogService/Log:1.0" usesname="log">
      <porttype type="data" />
    </uses>
    <uses repid="IDL:DiningPhilosophers/Fork:1.0" usesname="left">
      <porttype type="data" />
    </uses>
    <uses repid="IDL:DiningPhilosophers/Fork:1.0" usesname="right">
      <porttype type="data" />
    </uses>
  </ports>
</componentfeatures>
```


Component diagram
IDL code (diningphilosophers.idl)
Design capture tool
XML code (domain profile)
rscaXML2stubs
Skeleton code (diningphilosophers.h, diningphilosophersSK.cpp)
Component code (Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Domain Profiles Generation by Zeligsoft Tool


SCA Output for DiningPhilosophers
Output Directory:
Browse
Preview
DiningPhilosophers (Assembly)
DiningPhilosophers.sad.xml
DiningPhilosophersAssemblyController
DiningPhilosopherAssemblyController.scd.
DiningPhilosopherAssemblyController.spd.
Fork
Fork.scd.xml
Fork.spd.xml
Philosopher
Philosopher.scd.xml
Philosopher.spd.xml
Help
Cancel
Save
Model
Applications
DiningPhilosophers
Components
DiningPhilosophersAssemblyController
Fork
Host Collocation

```
<?xml version="1.0" encoding="us-ascii"?>
<!DOCTYPE softwareassembly SYSTEM "softwareassemb
<!--
    Generated by Zeligsoft Component Enabler 2.0.
    http://www.zeligsoft.com
-->
<softwareassembly id="DCE:1e7...
    <componentfiles>
        <componentfile id="ForkFile_...
            <localfile name="Fork.spd.xml" />
        </componentfile>
        <componentfile id="DiningPhilosophersAsse
            <localfile name="DiningPhilosopherAs
        </componentfile>
        <componentfile id="PhilosopherFile_43...
```

Assembly controller reference found and valid.
The descriptor file name 'DiningPhilosophers.sad.xml' for application 'Dini...
The assembly controller supports the "Resource" interface.
All instances on application 'DiningPhilosophers' have different SPD paths.
All instances in assembly 'DiningPhilosophers' have unique names.
The "AssemblyId" for application "DiningPhilosophers" is a well formed ...

# 3. Generate Skeleton and Component Code with rscaXML2stubs

# Skeleton Code

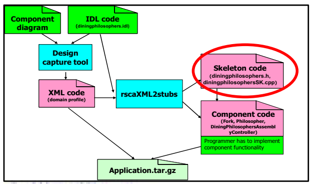
- Skeleton code is generated by rscaXML2stubs
  (rscaXML2stubs calls IDL complier)

**diningphilosophers.h**

```
#ifndef __diningphilosophers_hh__
#define __diningphilosophers_hh__

#ifndef USE_omniORB_logStream
#define USE_omniORB_logStream
#endif
…
_CORBA_MODULE DiningPhilosophers

_CORBA_MODULE_BEG

#ifndef __DiningPhilosophers_mFork__
#define __DiningPhilosophers_mFork__

  class Fork;
  class _objref_Fork;
  class _impl_Fork;
…
```

**diningphilosophersSK.cpp**

```
#include "diningphilosophers.h"
#include <omniORB4/IOP_S.h>
#include <omniORB4/IOP_C.h>
#include <omniORB4/callDescriptor.h>
#include <omniORB4/callHandle.h>
#include <omniORB4/objTracker.h>

OMNI_USING_NAMESPACE(omni)

DiningPhilosophers::Fork_ptr
DiningPhilosophers::Fork_Helper::_nil() {
  return ::DiningPhilosophers::Fork::_nil();
}
…
DiningPhilosophers::Fork_ptr
DiningPhilosophers::Fork::_narrow
(CORBA::Object_ptr obj)
…
```


Component diagram
IDL code (diningphilosophers.idl)
Design capture tool
XML code (domain profile)
rscaXML2stubs
Skeleton code (diningphilosophers.h, diningphilosophersSK.cpp)
Component code (Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Component Code (1)

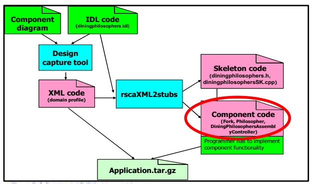
- Components code is generated by rscaXML2stubs

**Fork.h**

```
..
#include "diningphilosophers.h"

class user_DiningPhilosophers_Fork_i:
  public POA_DiningPhilosophers::Fork,
  public PortableServer::RefCountServantBase {

private:
    bool m_isRunning;
    bool m_isUsed;

public:
    user_DiningPhilosophers_Fork_i();

    virtual void startProcessing();
    virtual void stopProcessing(
    //methods corresponding to defined IDL
    CORBA::Boolean get();
    void release();
};
…
```

**Fork.cpp**

```
#include "Fork.h"

user_DiningPhilosophers_Fork_i
::user_DiningPhilosophers_Fork_i(){
    m_isRunning = false;
    m_isUsed = false;
}
Void user_DiningPhilosopher_Fork_i
::startProcessing(){
    m_isRunning = true;
}
Void user_DiningPhilosopher_Fork_i
::stopProcessing(){
    m_isRunning = false;
}

CORBA::Boolean
user_DiningPhilosophers_Fork_i::get(){
    // implement component functionality here
}

void
user_DiningPhilosophers_Fork_i::release(){
    // implement component functionality here
}
```


Component diagram
IDL code (diningphilosophers.idl)
Design capture tool
XML code (domain profile)
rscaXML2stubs
Skeleton code (diningphilosophers.h, diningphilosophersSK.cpp)
Component code (Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Component Code (2)

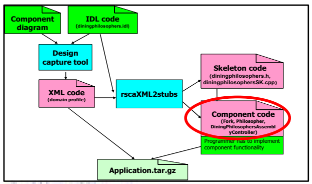
- Components code is generated by rscaXML2stubs

**Fork.h**

```
class Fork: public CF_Resource_impl{
public:
    bool m_isRunning;
    bool m_initStatus;
    Fork (..);
    virtual ~Fork ();
    virtual void initialize();
    virtual CORBA::Object_prt getPort(char *);
    virtual void runTest(...);
    virtual void start();
    virtual void stop();
     ...

private:
    user_DiningPhilosophers_Fork_i * m_user;
    CORBA::Object_var m_user_Object;
    ...
};
```

**Fork.cpp**

```
Fork:: Fork(..){
    m_isRunning = false;
    m_initStatus = false;
}
Void Fork::initialize(){
 //add code to initialize the resource
}
CORBA::Object_prt Fork::getPort(char *name){
    CORBA::Object_ptr portObject;
    if(strcmp(“user”,name)==0){
       portObject=
       CORBA::Object::_duplicate(m_user_Object.in());
    }else { ... }
    return portObject;
}
Void Fork::runTest(...){
    //add code to test
}
void Fork::start() throw  ...{
  m_isRunning = true;
  m_user_DiningPhilosophers_Fork_i->startProcessing();
  // add code here
}
void Fork ::stop() throw..{
  m_isRunning= false;
  m_user_DiningPhilosophers_Fork_i->stopProcessing();
  // add code here
}
```


Component diagram
IDL code
(diningphilosophers.idl)
Design capture tool
XML code
(domain profile)
rscaXML2stubs
Skeleton code
(diningphilosophers.h, diningphilosophersSK.cpp)
Component code
(Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Component Code (3)

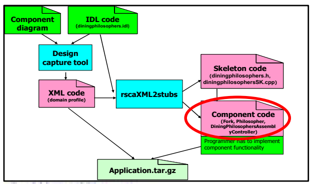
- Components code is generated by rscaXML2stubs

**Philosophers.h**

```
..
#include "diningphilosophers.h“

class PhilosopherLeftOutPortPOA : public
 POA_CF::Port(){
    PhilosopherLeftOutPortPOA();
    virtual void connectPort(CORBA::Object ,
     char * id);
    virtual void disconnectPort(char * id);
    ...
    DiningPhilosophers::Fork_prt m_object;
};

class PhilosopherRightOutPortPOA  : public
 POA_CF::Port(){
 PhilosopherRightOutPortPOA();
    virtual void connectPort(...);
    virtual void disconnectPort(char * id)
    ...
    DiningPhilosophers::Fork_prt m_object;
};
```

**Philosophers.cpp**

```
#include “Philosopher.h”

PhilosopherLeftOutPortPOA
::PhilosopherLeftOutPortPOA(){
    m_object = DiningPhilosphers::Fork::_nil();
}

void PhilosopherLeftOutPortPOA::
connectPort(CORBA::Object_prt connet, char * id){
  ...
  m_object =
  DiningPhilosophers::Fork::_narrow( connect );
}

void PhilosopherLeftOutPortPOA::
disconnectPort(CORBA::Object_prt connet){
  m_object = DiningPhilosphers::Fork::_nil();
}

PhilosopherRightOutPortPOA
::PhilosopherRightOutPortPOA(){
    m_object = DiningPhilosphers::Fork::_nil();
}
...
```


Component diagram
IDL code
(diningphilosophers.idl)
Design capture tool
XML code
(domain profile)
rscaXML2stubs
Skeleton code
(diningphilosophers.h, diningphilosophersSK.cpp)
Component code
(Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Component Code (4)

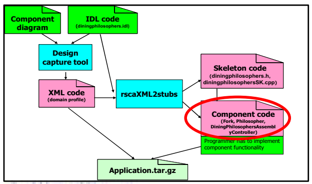
- Components code is generated by rscaXML2stubs

**Philosophers.h**

```
class Philosopher: public CF_Resource_impl {
public:
    bool m_isRunning;
    bool m_initStatus;

    Philosopher(..);
    virtual ~Philosopher();
    virtual void initialize();
    virtual CORBA::Object_prt getPort(char *);
    virtual void runTest(...);
    virtual void start();
    virtual void stop();
..
private:
   PhilosopherLeftOutPortPOA *
                   m_leftOutPortPOA;
   CF::Port_var m_leftOutPort;

   PhilosopherRightOutPortPOA *
                   m_righyOutPortPOA;
   CF::Port_var m_rightOutPort;
};
```

**Philosophers.cpp**

```
Philosopher::Philosopher(..){
    m_isRunning = false;
    m_initStatus = false;
}
Void Philosopher::initialize(){
    //add code to initialize the resource
}

CORBA::Object_prt Philosopher
::getPort(char *name){
    CORBA::Object_ptr portObject;
    if(strcmp(“left”,name)==0){
       portObject=
       CORBA::Object::_duplicate(m_leftOutPort.in());
    }else { ... }
    return portObject;
}
Void Philosopher::runTest(...){
    //add code to test
}
void Philosopher::start() throw  ...{
    m_isRunning = true;
    // add code here
}
void Philosopher::stop() throw..{
    m_isRunning = false;
    // add code here
}
```


Component diagram
IDL code (diningphilosophers.idl)
Design capture tool
XML code (domain profile)
rscaXML2stubs
Skeleton code (diningphilosophers.h, diningphilosophersSK.cpp)
Component code (Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Component Code (5)

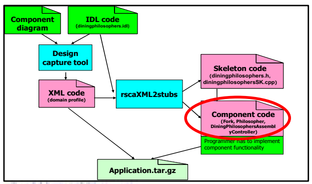
- Components code is generated by rscaXML2stubs

**DiningPhilosophersAssemblyController.h**

```
class DiningPhilosophersAssemblyController:
public CF_Resource_impl{
public:
    bool m_isRunning;
    bool m_initStatus;
    Fork (..);
    virtual ~Fork ();
    virtual void initialize();
    virtual CORBA::Object_prt getPort(char *);
    virtual void runTest(...);
    virtual void start();
    virtual void stop();
    ...
private:
    CF::Resource_ptr m_Kant;
    CF::Resource_ptr m_Decartes;
    CF::Resource_ptr m_Socrates;
    CF::Resource_ptr m_fork1;
    CF::Resource_ptr m_fork2;
    CF::Resource_ptr m_fork3;
    CF::Application_ptr m_application;
    ...
};
```

**DiningPhilosophersAssemblyController.cpp**

```
#include “DiningPhilosophersAssemblyController.h"
DiningPhilosophersAssemblyController
:: DiningPhilosophersAssemblyController (..){
    m_isRunning = false;
    m_initStatus = false;   ...
}
CORBA::Object_prt
DiningPhilosophersAssemblyController
::getPort(char *name){
    CORBA::Object_ptr portObject;
    if(strcmp(“log”,name)==0){
       portObject=
       CORBA::Object::_duplicate(m_leftOutPort.in());
    }else { ... }
    return portObject;
}
Void DiningPhilosophersAssemblyController::runTest{
    //add code to test
}
void DiningPhilosophersAssemblyController::start() {
    m_isRunning = true;
    m_Kant->start();
    m_Decartes->start();
    m_Socrates->start();
    m_fork1->start();
    m_fork2->start();
    m_fork3->start();
}
```

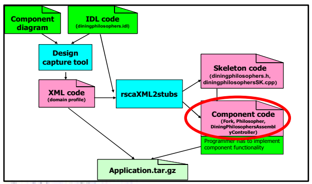

Component diagram
IDL code (diningphilosophers.idl)
Design capture tool
XML code (domain profile)
rscaXML2stubs
Skeleton code (diningphilosophers.h, diningphilosophersSK.cpp)
Component code (Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Component Code (6)

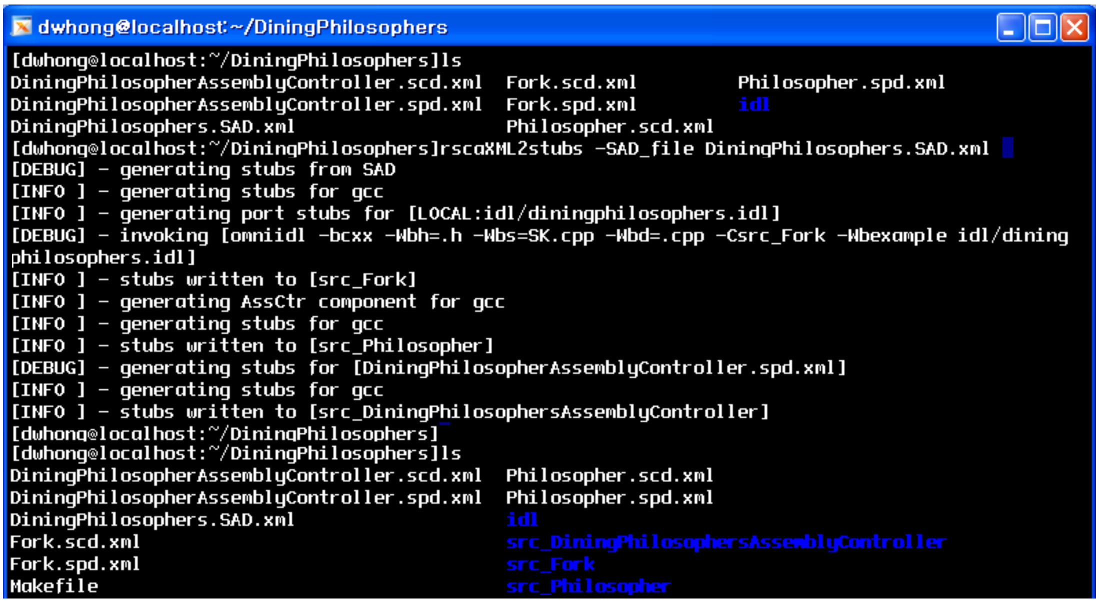
- Components code is generated by rscaXML2stubs

DiningPhilosophersAssemblyController.cpp

```
void Fork ::stop(){
    m_isRunning = false;
    m_Kant->stop();
    m_Decartes->stop();
    ….
}
void DiningPhilosopherAssemblyController::initialize(){
    CF::Application::ComponentElementSequence_var componentNamingContexts
                    = m_application->componentNamingContexts();
    CORBANameServiceRegistrar *registrar = new CORBANameServiceRegistrar();
    CF::Resource_ptr component;
    for(unsigned int i=0; i<c_NamingContexts->length(); i++)
    {
        string elementID = (const char*) (componentNamingContexts[i]).elementID;
        CORBA::Object_var oo = registrar->getObjectReference(elementID);
        component = CF::Resource::_narrow(oo.in());
        if( elementID.find("Kant") != elementID.npos)
            m_Kant = component;
        else if(elementID.find("Decartes") != elementID.npos)
            m_Decartes = component;
        else if(elementID.find("Socrates") != elementID.npos)
            m_Socrates = component;
        ….
    }
}
```


Component diagram
IDL code (diningphilosophers.idl)
Design capture tool
XML code (domain profile)
rscaXML2stubs
Skeleton code (diningphilosophers.h, diningphilosophersSK.cpp)
Component code (Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Code Generation by rscaXML2subs



dwhong@localhost:~/DiningPhilosophers

```
[dwhong@localhost:~/DiningPhilosophers]ls
DiningPhilosopherAssemblyController.scd.xml  Fork.scd.xml          Philosopher.spd.xml
DiningPhilosopherAssemblyController.spd.xml  Fork.spd.xml          idl
DiningPhilosophers.SAD.xml                   Philosopher.scd.xml
[dwhong@localhost:~/DiningPhilosophers]rscaXML2stubs -SAD_file DiningPhilosophers.SAD.xml
[DEBUG] - generating stubs from SAD
[INFO ] - generating stubs for gcc
[INFO ] - generating port stubs for [LOCAL:idl/diningphilosophers.idl]
[DEBUG] - invoking [omniidl -bcxx -Wbh=.h -Wbs=SK.cpp -Wbd=.cpp -Csrc_Fork -Wbexample idl/dining
philosophers.idl]
[INFO ] - stubs written to [src_Fork]
[INFO ] - generating AssCtr component for gcc
[INFO ] - generating stubs for gcc
[INFO ] - stubs written to [src_Philosopher]
[DEBUG] - generating stubs for [DiningPhilosopherAssemblyController.spd.xml]
[INFO ] - generating stubs for gcc
[INFO ] - stubs written to [src_DiningPhilosophersAssemblyController]
[dwhong@localhost:~/DiningPhilosophers]
[dwhong@localhost:~/DiningPhilosophers]ls
DiningPhilosopherAssemblyController.scd.xml  Philosopher.scd.xml
DiningPhilosopherAssemblyController.spd.xml  Philosopher.spd.xml
DiningPhilosophers.SAD.xml                   idl
Fork.scd.xml                                 src_DiningPhilosophersAssemblyController
Fork.spd.xml                                 src_Fork
Makefile                                     src_Philosopher
```

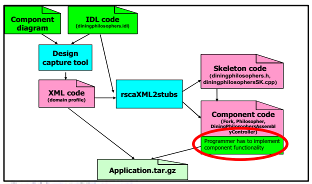
# Type in Component Functionality

# Component Functionality (1)

**Fork.cpp**

```
CORBA::Boolean user_DiningPhilosophers_Fork_i::get() {
    pthread_mutex_lock(&m_accessMutex);
    if (m_isUsed == true) {
        pthread_mutex_unlock(&m_accessMutex);
        return CORBA::Boolean(false);
    } else {
        m_isUsed = true;
        pthread_mutex_unlock(&m_accessMutex);
        return CORBA::Boolean(true);
    }
}
```

```
void user_DiningPhilosophers_Fork_i::release()  {
    pthread_mutex_lock(&m_accessMutex);
    m_isUsed = false;
    pthread_mutex_unlock(&m_accessMutex);
}
```

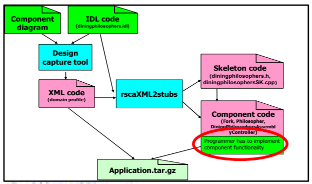

Component diagram
IDL code
(diningphilosophers.idl)
Design capture tool
XML code
(domain profile)
rscaXML2stubs
Skeleton code
(diningphilosophers.h, diningphilosophersSK.cpp)
Component code
(Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Component Functionality (2)

**Philosophers.cpp**

```
void Philosopher::start()
 throw (CORBA::SystemException, CF::Resource::StartError) {
    m_isRunning = true;
    pthread_create(&m_thread, NULL, philosopherStart, (void*)(this));
}
```

```
void *philosopherStart(void* arg) {
    DPRINT("[Philosopher::philosopherStart()] " << endl);
    Philosopher *philosopher = (Philosopher *)(arg);
    while (philosopher->m_isRunning == true) {
        philosopher->thinking();
        philosopher->hungry();
        philosopher->eating();
    }
}
```

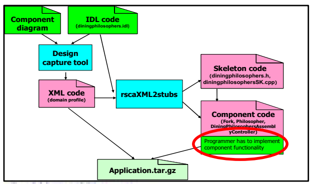

Component diagram
IDL code (diningphilosophers.idl)
Design capture tool
XML code (domain profile)
rscaXML2stubs
Skeleton code (diningphilosophers.h, diningphilosophersSK.cpp)
Component code (Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Component Functionality (3)

**Philosophers.cpp**

```
void  Philosopher::thinking() {
    DPRINT(m_Name << " Philosopher::thinking() "<< endl);
    sleep(m_thinkingTime);
}
```

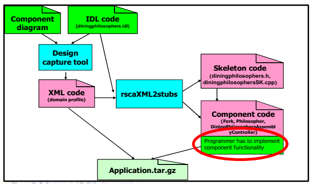
```
void  Philosopher::eating() {
        sleep(m_eatingTime);
        m_leftOutPortPOA->m_object->release();
        m_rightOutPortPOA->m_object->release();
}
```


Component diagram
IDL code (diningphilosophers.idl)
Design capture tool
XML code (domain profile)
rscaXML2stubs
Skeleton code (diningphilosophers.h, diningphilosophersSK.cpp)
Component code (Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

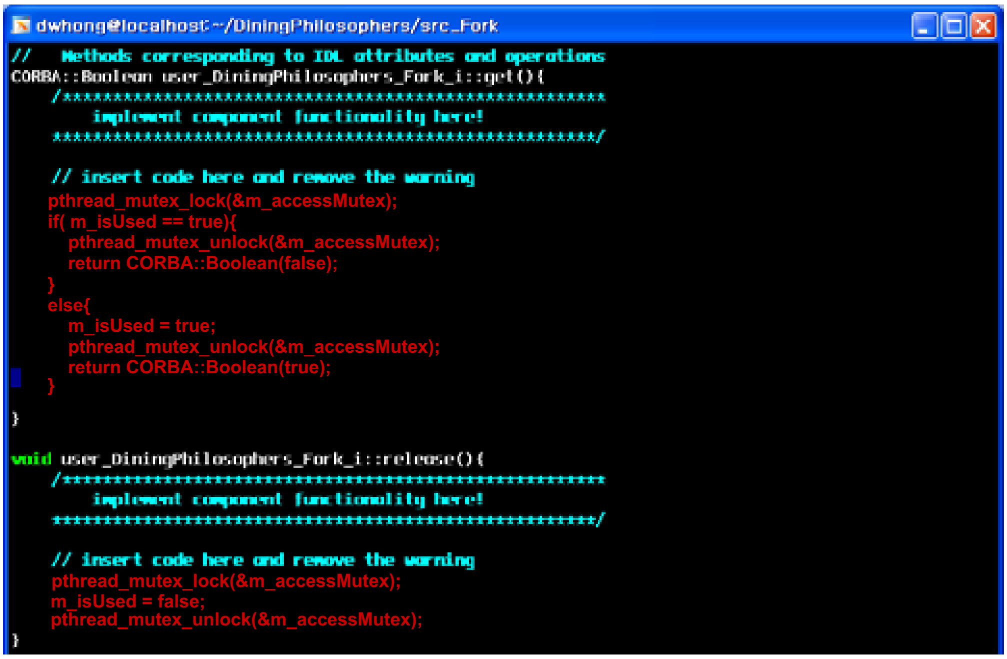
# Component Functionality (4)

**Philosophers.cpp**

```
void  Philosopher::hungry() {
    while (true) {
        // Try to get the right fork first
        if (m_rightOutPortPOA->m_object->get() == true) {
            if (m_leftOutPortPOA->m_object->get() == true){
                break;
            }
            else{
                m_rightOutPortPOA->m_object->release();
            }
        }
        sleep(m_hungryTime);
    } // end of while
}
```


Component diagram
IDL code (diningphilosophers.idl)
Design capture tool
XML code (domain profile)
rscaXML2stubs
Skeleton code (diningphilosophers.h, diningphilosophersSK.cpp)
Component code (Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Type in Component Code



```
//   Methods corresponding to IDL attributes and operations
CORBA::Boolean user_DiningPhilosophers_Fork_i::get(){
    /******************************************************
        implement component functionality here!
    ******************************************************/

    // insert code here and remove the warning
    pthread_mutex_lock(&m_accessMutex);
    if( m_isUsed == true){
      pthread_mutex_unlock(&m_accessMutex);
      return CORBA::Boolean(false);
    }
    else{
      m_isUsed = true;
      pthread_mutex_unlock(&m_accessMutex);
      return CORBA::Boolean(true);
    }

}

void user_DiningPhilosophers_Fork_i::release(){
    /******************************************************
        implement component functionality here!
    ******************************************************/

    // insert code here and remove the warning
    pthread_mutex_lock(&m_accessMutex);
    m_isUsed = false;
    pthread_mutex_unlock(&m_accessMutex);
}
```

# 4. Compile and Build Package

# Entire Source Tree Structure

DiningPhilosophers/

- src_DiningPhilosophersAssemblyController/

| | | |
|---|---|---|
| DiningPhilosophersAssemblyController.h | // 50 lines | Code for the component |
| DiningPhilosophersAssemblyController.cpp | // 165 lines | |
| DiningPhilosophersAssemblyControllerServer.h | // 58 lines | Launcher code |
| DiningPhilosophersAssemblyControllerServer.h | // 295 lines | |

- src_Fork/

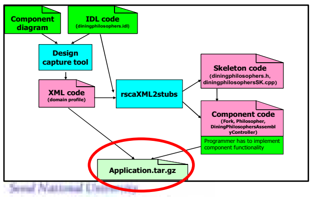
| | | |
|---|---|---|
| Fork.h | // 79 lines | Code for the Fork component |
| Fork.cpp | // 236 lines | |
| ForkServer.h | // 58 lines | Launcher code |
| ForkServer.cpp | // 295 lines | |
| diningphilosophers.h | // 215 lines | Skeleton code |
| diningphilosophersSK.cpp | // 280 lines | |

- src_Philosopher/

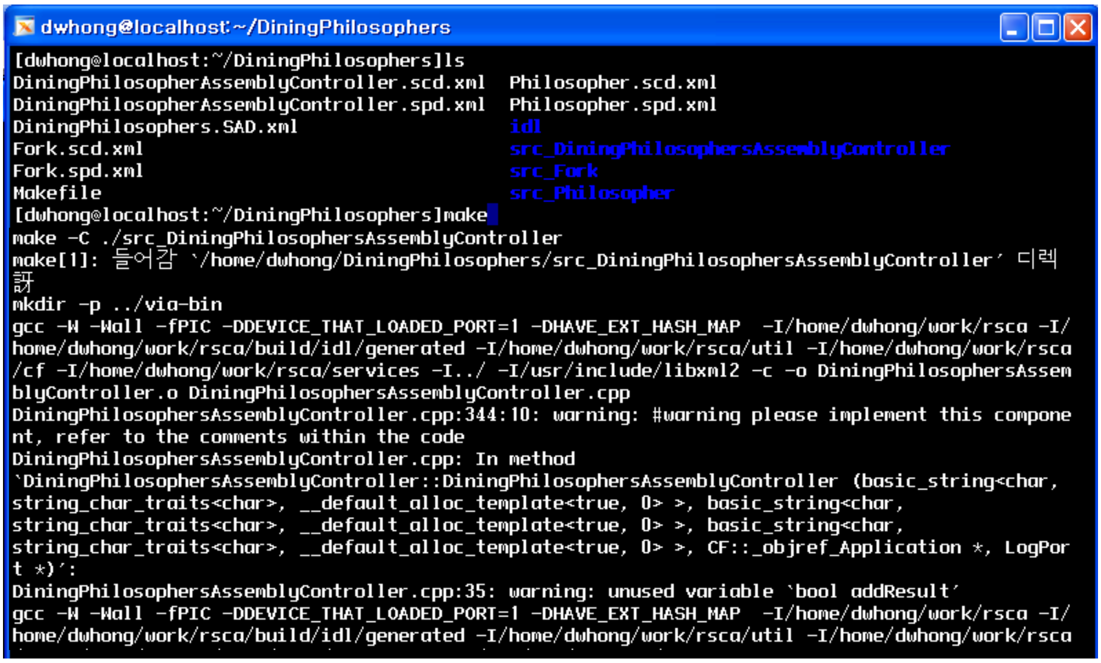
| | | |
|---|---|---|
| Philosopher.h | // 50 lines | Code for the Philosopher component |
| Philosopher.cpp | // 165 lines | |
| PhilosopherServer.h | // 58 lines | Launcher code |
| PhilosopherServer.cpp | // 295 lines | |
| diningphilosophers.h | // 215 lines | |
| diningphilosophersSK.cpp | // 280 lines | Skeleton code |

# Application Package

- Application is packaged as a tar.gz type
  - DiningPhilosophers-generic-x86.tar.gz
    - DiningPhilosophers.SAD.xml
    - DiningPhilosopherAssemblyController.scd.xml
    - DiningPhilosopherAssemblyController.spd.xml
    - Fork.scd.xml
    - Fork.spd.xml
    - Philosopher.scd.xml
    - Philosopher.spd.xml
    - bin/
      - DiningPhilosophersAssemblyControllerServer
      - ForkServer
      - PhilosopherServer

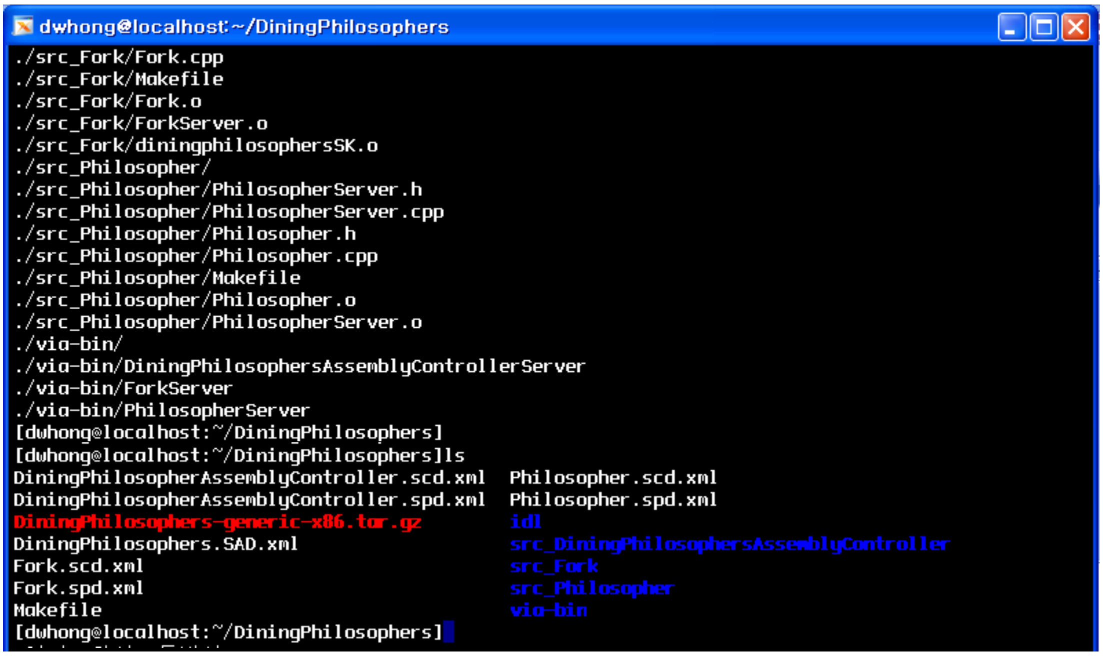

Component diagram
IDL code
(diningphilosophers.idl)
Design capture tool
XML code
(domain profile)
rscaXML2stubs
Skeleton code
(diningphilosophers.h, diningphilosophersSK.cpp)
Component code
(Fork, Philosopher, DiningPhilosophersAssemblyController)
Programmer has to implement component functionality
Application.tar.gz

# Compile



```
[dwhong@localhost:~/DiningPhilosophers]ls
DiningPhilosopherAssemblyController.scd.xml  Philosopher.scd.xml
DiningPhilosopherAssemblyController.spd.xml  Philosopher.spd.xml
DiningPhilosophers.SAD.xml                   idl
Fork.scd.xml                                 src_DiningPhilosophersAssemblyController
Fork.spd.xml                                 src_Fork
Makefile                                     src_Philosopher
[dwhong@localhost:~/DiningPhilosophers]make
make -C ./src_DiningPhilosophersAssemblyController
make[1]: 들어감 `/home/dwhong/DiningPhilosophers/src_DiningPhilosophersAssemblyController' 디렉
訝
mkdir -p ../via-bin
gcc -W -Wall -fPIC -DDEVICE_THAT_LOADED_PORT=1 -DHAVE_EXT_HASH_MAP  -I/home/dwhong/work/rsca -I/
home/dwhong/work/rsca/build/idl/generated -I/home/dwhong/work/rsca/util -I/home/dwhong/work/rsca
/cf -I/home/dwhong/work/rsca/services -I../ -I/usr/include/libxml2 -c -o DiningPhilosophersAssem
blyController.o DiningPhilosophersAssemblyController.cpp
DiningPhilosophersAssemblyController.cpp:344:10: warning: #warning please implement this compone
nt, refer to the comments within the code
DiningPhilosophersAssemblyController.cpp: In method
`DiningPhilosophersAssemblyController::DiningPhilosophersAssemblyController (basic_string<char,
string_char_traits<char>, __default_alloc_template<true, 0> >, basic_string<char,
string_char_traits<char>, __default_alloc_template<true, 0> >, basic_string<char,
string_char_traits<char>, __default_alloc_template<true, 0> >, CF::_objref_Application *, LogPor
t *)':
DiningPhilosophersAssemblyController.cpp:35: warning: unused variable `bool addResult'
gcc -W -Wall -fPIC -DDEVICE_THAT_LOADED_PORT=1 -DHAVE_EXT_HASH_MAP  -I/home/dwhong/work/rsca -I/
home/dwhong/work/rsca/build/idl/generated -I/home/dwhong/work/rsca/util -I/home/dwhong/work/rsca
```

# Compile

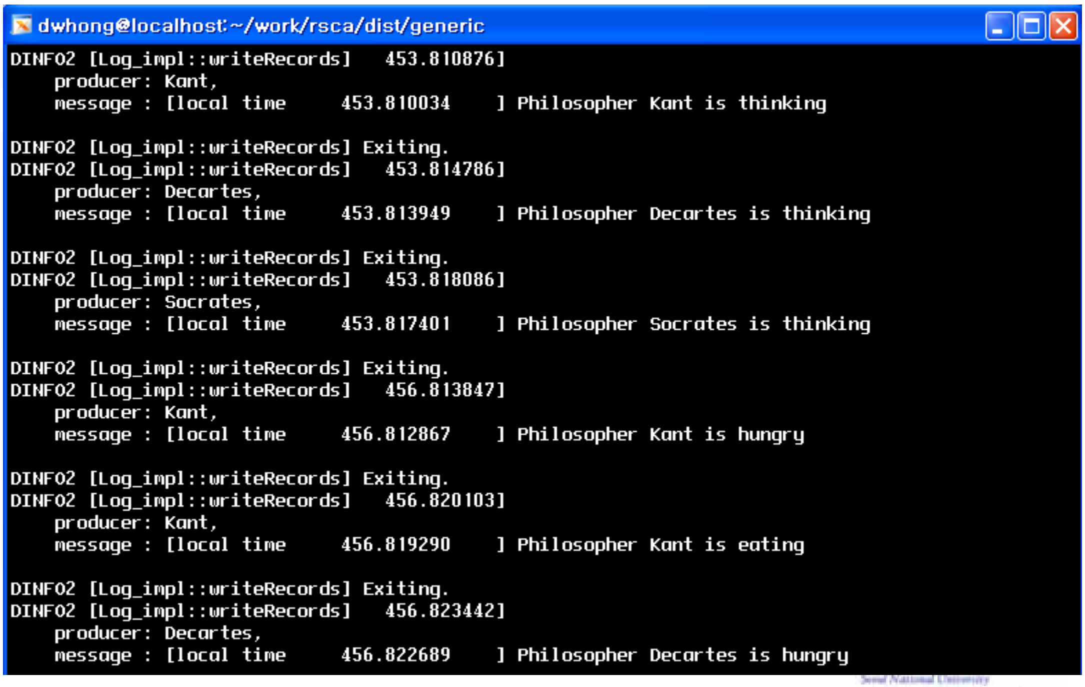


dwhong@localhost:~/DiningPhilosophers

```
./src_Fork/Fork.cpp
./src_Fork/Makefile
./src_Fork/Fork.o
./src_Fork/ForkServer.o
./src_Fork/diningphilosophersSK.o
./src_Philosopher/
./src_Philosopher/PhilosopherServer.h
./src_Philosopher/PhilosopherServer.cpp
./src_Philosopher/Philosopher.h
./src_Philosopher/Philosopher.cpp
./src_Philosopher/Makefile
./src_Philosopher/Philosopher.o
./src_Philosopher/PhilosopherServer.o
./via-bin/
./via-bin/DiningPhilosophersAssemblyControllerServer
./via-bin/ForkServer
./via-bin/PhilosopherServer
[dwhong@localhost:~/DiningPhilosophers]
[dwhong@localhost:~/DiningPhilosophers]ls
DiningPhilosopherAssemblyController.scd.xml  Philosopher.scd.xml
DiningPhilosopherAssemblyController.spd.xml  Philosopher.spd.xml
DiningPhilosophers-generic-x86.tar.gz        idl
DiningPhilosophers.SAD.xml                   src_DiningPhilosophersAssemblyController
Fork.scd.xml                                 src_Fork
Fork.spd.xml                                 src_Philosopher
Makefile                                     via-bin
[dwhong@localhost:~/DiningPhilosophers]
```

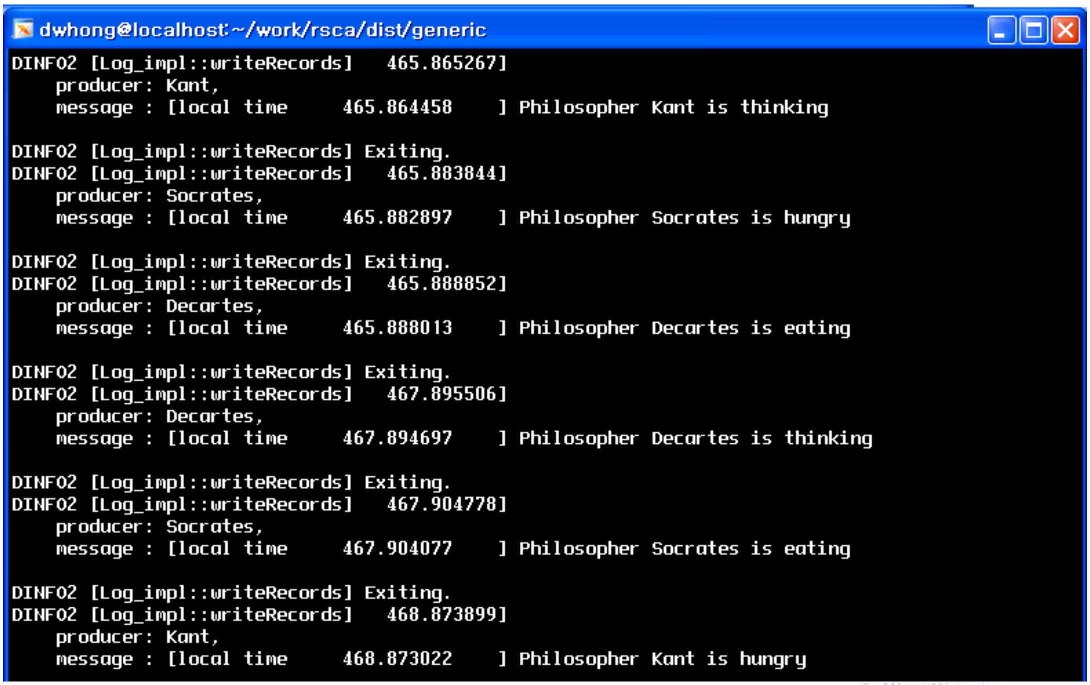
# Execute the System

# Execution (1)



```
DINFO2 [Log_impl::writeRecords]   453.810876]
    producer: Kant,
    message : [local time     453.810034    ] Philosopher Kant is thinking

DINFO2 [Log_impl::writeRecords] Exiting.
DINFO2 [Log_impl::writeRecords]   453.814786]
    producer: Decartes,
    message : [local time     453.813949    ] Philosopher Decartes is thinking

DINFO2 [Log_impl::writeRecords] Exiting.
DINFO2 [Log_impl::writeRecords]   453.818086]
    producer: Socrates,
    message : [local time     453.817401    ] Philosopher Socrates is thinking

DINFO2 [Log_impl::writeRecords] Exiting.
DINFO2 [Log_impl::writeRecords]   456.813847]
    producer: Kant,
    message : [local time     456.812867    ] Philosopher Kant is hungry

DINFO2 [Log_impl::writeRecords] Exiting.
DINFO2 [Log_impl::writeRecords]   456.820103]
    producer: Kant,
    message : [local time     456.819290    ] Philosopher Kant is eating

DINFO2 [Log_impl::writeRecords] Exiting.
DINFO2 [Log_impl::writeRecords]   456.823442]
    producer: Decartes,
    message : [local time     456.822689    ] Philosopher Decartes is hungry
```

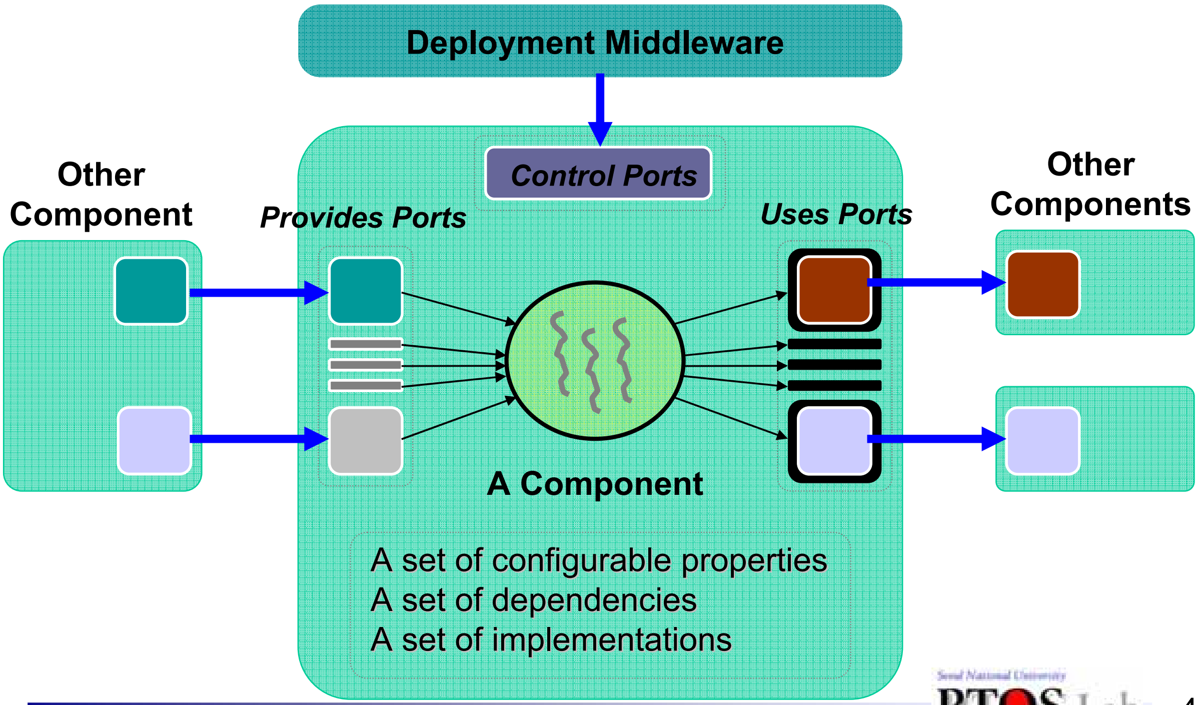
# Execution (2)



```
DINFO2 [Log_impl::writeRecords]   465.865267]
    producer: Kant,
    message : [local time     465.864458    ] Philosopher Kant is thinking

DINFO2 [Log_impl::writeRecords] Exiting.
DINFO2 [Log_impl::writeRecords]   465.883844]
    producer: Socrates,
    message : [local time     465.882897    ] Philosopher Socrates is hungry

DINFO2 [Log_impl::writeRecords] Exiting.
DINFO2 [Log_impl::writeRecords]   465.888852]
    producer: Decartes,
    message : [local time     465.888013    ] Philosopher Decartes is eating

DINFO2 [Log_impl::writeRecords] Exiting.
DINFO2 [Log_impl::writeRecords]   467.895506]
    producer: Decartes,
    message : [local time     467.894697    ] Philosopher Decartes is thinking

DINFO2 [Log_impl::writeRecords] Exiting.
DINFO2 [Log_impl::writeRecords]   467.904778]
    producer: Socrates,
    message : [local time     467.904077    ] Philosopher Socrates is eating

DINFO2 [Log_impl::writeRecords] Exiting.
DINFO2 [Log_impl::writeRecords]   468.873899]
    producer: Kant,
    message : [local time     468.873022    ] Philosopher Kant is hungry
```

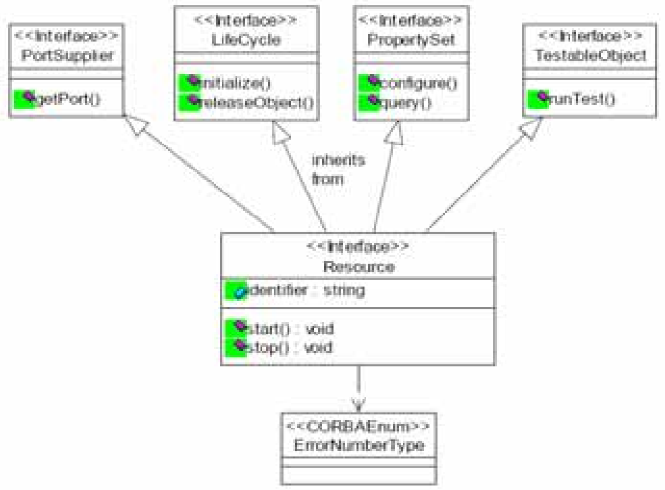
# RSCA Component Model

# RSCA Component Model



A set of configurable properties
A set of dependencies
A set of implementations

# RSCA Interfaces (1)



*Resource* Interface UML

- Every component must implement *Resource* interface

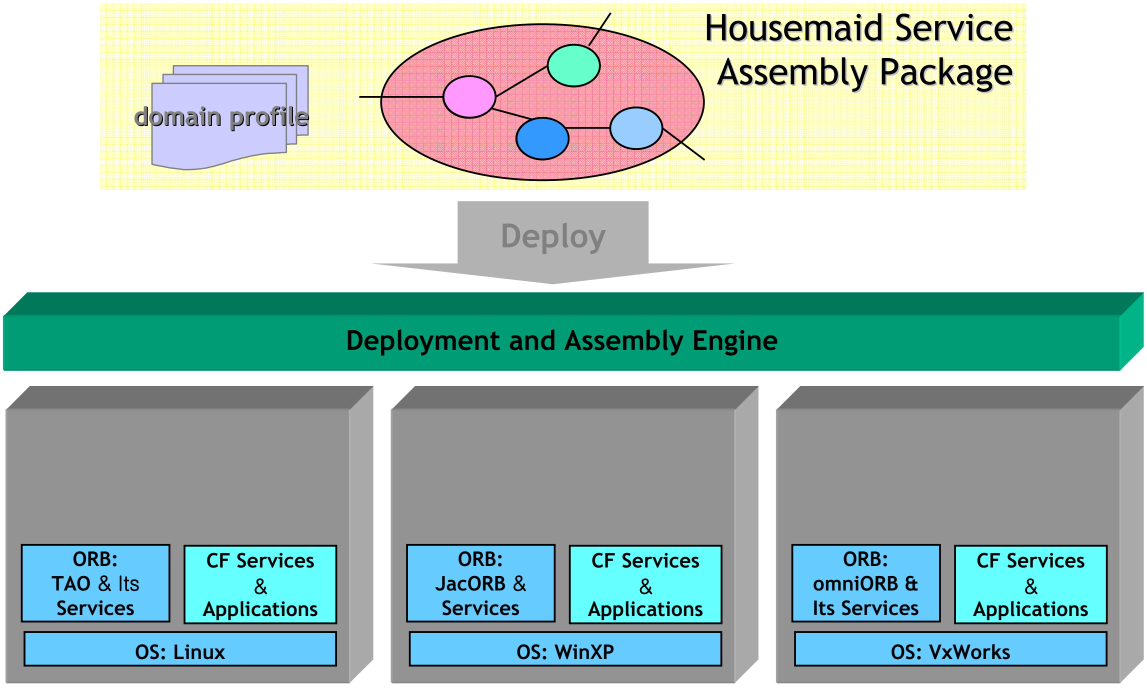
# RSCA Interfaces (2)

- LifeCycle interface
  - defines the generic operations for initializing or releasing
    - `initialize(),releaseObject()`
- PortSupplier interface
  - provides the `getPort` operation for those components that provide ports
    - `getPort()`
- PropertySet interface
  - defines the operations to access component properties
    - `configure(), query()`
- TestableObject interface
  - defines the operations used to test component implementations
    - `runTest()`
- Resource interface
  - provides a common API for the control and configuration of a software component
    - `start(),stop()`

# RSCA Port

- What is a port ?
  - Used to provide (use) services to (from) other components
  - A port implements a set of port-specific interfaces for other components
- RSCA's component at least
  - Should provide control ports implementing a set of interfaces for management purpose
  - Should be described in a set of XML descriptors
    - Called domain profile

# Dynamic Deployment of RSCA CF (1)



Housemaid Service Assembly Package

domain profile

Deploy

Deployment and Assembly Engine

ORB: TAO & Its Services | CF Services & Applications | OS: Linux

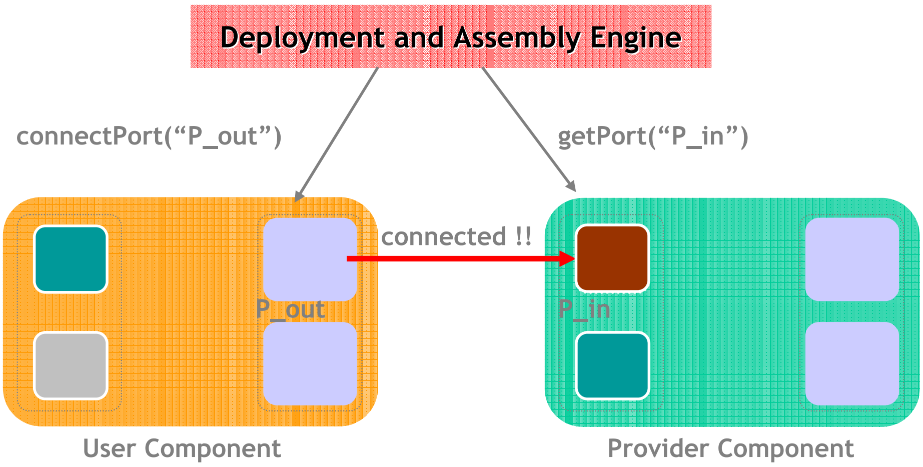
ORB: JacORB & Services | CF Services & Applications | OS: WinXP

ORB: omniORB & Its Services | CF Services & Applications | OS: VxWorks

Based on the information described in the domain profile

# Dynamic Deployment of RSCA CF (2)

- Dynamic deployment is based on the port (CORBA object) and IOR



robotics/2005-06-19

# Agenda for OMG Robotics DTF Infrastructure WG

OMG Boston Meeting
June 27, 2006

Saehwa Kim
Real-Time Operating Systems Lab.
Seoul National University
http://redwood.snu.ac.kr

# Contents

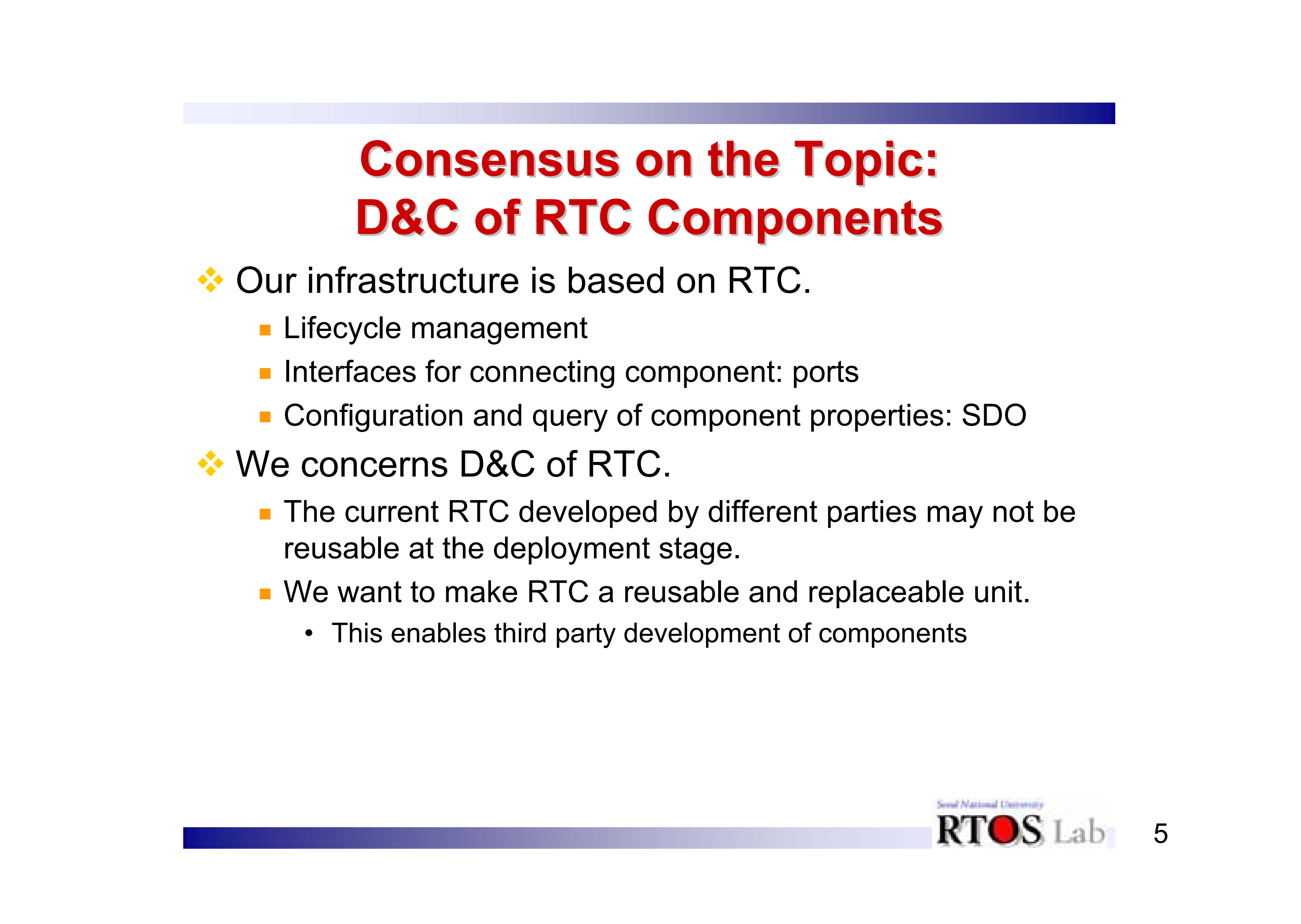
- Concerns of Prioritization (from the last St. Louis Meeting)
- Roadmap for Deployment and Configuration (D&C) RFP
- Consensus on the Topic: D&C of RTCs
- RFP Outline
- Gathering requirements
- Gathering volunteers for draft RFP

# Concerns of Prioritization (from the Last St. Louis Meeting)

- Top priority
  - Deployment & configuration (D&C)
- Others concerned
  - Resource management
  - Event management
  - Data distribution
  - Behavior of control systems

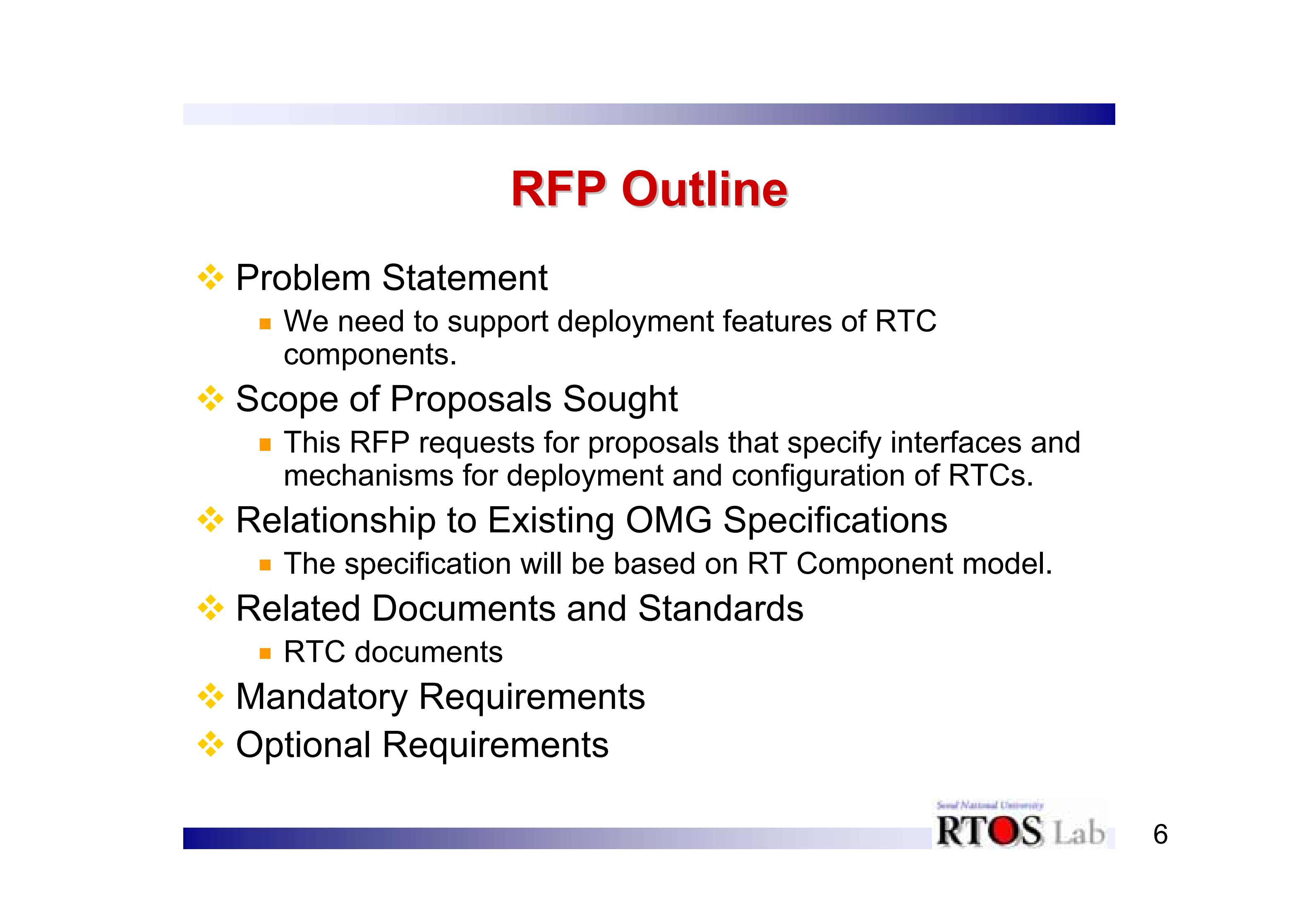
# Roadmap for D&C RFP

- Original roadmap (from the last St. Louis meeting)
  - Outline/framework RFP in Boston (June)
  - Draft RFP in Anaheim (Sep.)
  - Review RFP in Washington D.C. (Dec.)
  - Second review RFP (Mar.)
  - Issue RFP (Mar.)
- Proposed revised roadmap
  - Outline/framework RFP in Boston (June)
  - Review internal draft RFP in Anaheim (Sep.)
  - Issue draft RFP in Washington D.C. (Dec.)
  - Review official RFP in San Diego (Mar.)
  - Issue official RFP (June)



# Consensus on the Topic: D&C of RTC Components

- Our infrastructure is based on RTC.
  - Lifecycle management
  - Interfaces for connecting component: ports
  - Configuration and query of component properties: SDO
- We concerns D&C of RTC.
  - The current RTC developed by different parties may not be reusable at the deployment stage.
  - We want to make RTC a reusable and replaceable unit.
    - This enables third party development of components



# RFP Outline

- Problem Statement
  - We need to support deployment features of RTC components.
- Scope of Proposals Sought
  - This RFP requests for proposals that specify interfaces and mechanisms for deployment and configuration of RTCs.
- Relationship to Existing OMG Specifications
  - The specification will be based on RT Component model.
- Related Documents and Standards
  - RTC documents
- Mandatory Requirements
- Optional Requirements

## Candidate Requirements

- Requirements for deployment facility
  - To place, install, load, connect, execute components
- Requirements for component (and node) description facility
  - To maintain persistent information needed for component deployment
- Requirements for packaging facility

## Requirements for Deployment Facility

- Responses shall build upon interfaces for connection components (ports) in RTC component model.
- Responses shall specify concrete elements for

# Requirements for Component Description Facility

- Responses shall specify an information model that describes components.
- Responses shall specify a PSM that specifies describes interfaces for a programmatic representation of this information model and

## 1 What is Deployment and Configuration (D&C)? Why is it important?

- Makes interoperability possible
  - RTC submissions define what a component is, but not how to find, load, or create one
  - What is persistence format of component configuration files?
  - How are binaries and descriptors packaged together?
  - Can a particular component run on a particular platform?
  - How are available components discovered?
- Improves reuse
  - Not just model reuse, but also PSM-level (source and/or binary) reuse
  - Component defined using compliant tool can be run on any compliant middleware

## 2 Volunteers to draft document(s)

- SNU
- ADA
- RTI
- AIST

## 3 Deployment use cases

- Static deployment
  - Relationships defined at design time
  - Binaries statically linked at compile time
  - Benefits
    - Necessary for deployment to embedded devices
      - Platform may not support process model or dynamic library loading
      - Less resource intensive than dynamic deployment
    - Most deterministic kind of deployment
  - Limitations
    - Where each component runs must be decided before application launch
    - Many types of changes require rebuilding whole application
  - Users
    - RTI
    - Automobile and aerospace industries
  - Feedback
    - Necessary for price-sensitive applications, limited-resource platforms
    - Guaranteeing that deployment matches design
    - Difficult to achieve with distributed system: some dynamic configuration (e.g. location) usually important
- Semi-dynamic deployment
  - Relationships defined at design time (by middleware)
  - Allocation of components to nodes can take place at application launch
  - Components connected during application initialization
    - After that, relationships are static
    - Middleware determines "appropriateness" of relationships
      - Based on comparing capabilities/characteristics of hardware and applications
    - Components loaded/started dynamically by middleware
  - Existing standards
    - (Lw)CCM
    - SWRadio
    - Either requires extensions to handle extra information from RTC
  - Benefits
    - Most of determinism of design-time relationships
    - Possibility to package components once for multiple platforms
      - Middleware can choose implementation/configuration based on deployment platform
  - Limitations
    - Changes to component definitions require redeploying that component (but not whole application)
    - Changes to inter-component relationships require restarting application
  - Users

- SNU
- Hitachi
- Feedback
  - Important for in-the-field upgrading of applications
  - Deployment/integration of third-party applications
  - Some benefit over fully dynamic: security (don't let anyone talk to anyone else)
    - Some security provision must be made in D&C, esp. authentication (authorization as well?)
  - What about failures (e.g. connectivity)?
    - Tools should validate configuration at design time
    - Errors starting up/deploying
      - How to log errors (and successes, etc.)?
      - Is log/test API part of this specification?
      - RFP could require respondents to specify how events will be logged
      - Request logging best practices/conventions as part of response
    - Runtime errors should be handled at component middleware layer or above
      - Wait for industry demand for error handling in middleware itself
      - Look for cues in related specifications

- Fully dynamic deployment
  - Application defines connections
  - Components discover one another at runtime
    - Relationships chosen dynamically based on which components discover each other
    - Components can come and go while application is running
    - Relationships can change at any time
  - Existing standards
    - SDO allows introspection of discovered components
    - CORBA defines naming service components can use to discover each other
    - SCA Core Framework supports looking up components by provided interfaces
    - SCA Core Framework allows components to find out when other components go away and fail over, but not directly supported by framework
  - Benefits
    - Requires least work up-front
    - Easy to change your mind at runtime based on observed behavior without shutting down application
  - Limitations
    - Least deterministic kind of deployment
    - Potentially difficult to persist/repeat configuration changes made at runtime
      - Working around this is desirable. AIST, for example, doesn't avoid this yet.
  - User: AIST
  - Feedback
    - Useful when application broadly distributed
    - No a priori knowledge of collaborators necessary
- RFP scope
  - Q: Should RFP require all use cases or ask respondents to pick their use cases (possibly defining additional) and explain why?
    - A: Let respondents choose, incl. identifying new use cases

## 4 Required information

- Component definitions
  - Binary implementation(s)
  - Ports
  - Properties
  - Descriptor must indicate whether binaries support static, semi-dynamic, or fully dynamic deployment (or some combination)
  - Execution semantics
    - Execution contexts
    - Order of periodic execution
    - States and transitions
    - Modes of operation
  - Should we consider source code-only components? (i.e. no compiled binary)
- Component configurations
  - Connections/assemblies
  - Property values
  - Which node component is deployed on

    - Component itself just describes requirements…
    - …and someone else can map requirements to available nodes
    - Resulting node choice may be different for different implementations of same logical component
  - Execution rate(s)
- To support static deployment:
  - Depending on implementation, descriptor file(s) may not be needed at runtime
- To support dynamic deployment:
  - Capability/characteristic model to validate deployment
    - What resources does component "A" require?
    - What resources does platform "B" provide?
    - "Resource" may include some timeliness contracts
    - If B provides >= A's requirements, A can be deployed on B
  - Clarification:
    - Configuration document may be static (probably text/XML file), …
    - …but (in the case of fully dynamic deployment) relationships may change at runtime
- Address two models of hardware/software interaction
  - Platform model
    - Hardware hosts application(s)
  - Hardware in the loop model
    - Components with hardware and software-only components are peers
    - Real hardware may be replaced with software simulation (or visa versa) transparently

# Robotics-DTF/SDO-DSIG Joint Meeting

June 28, 2006

Boston, MA, USA

HYATT Harborside

# Approval of St. Louis Minutes

- Ask for a volunteer (minutes taker)
  - Hung Pham (RTI)
  - Olivier Lemaire (AIST)
- St.Louis Minutes review

[Robotics] **We had two special Talks (Christopher Gill, Washington Univ. , Jean-Christophe Baillie, ENSTA/Gostai ) and six RFI response presentations. By the excellent leadership of seven volunteers, we had very active group discussions and three working groups were chartered.**

[SDO] **Make a progress report of revised submission of Robot Technology Components RFP**.

# Review Agenda

**Wed(Jun.28): Michelangelo B, 2nd FL**

**TF Plenary (Robobusiness, WG Reports, Contact Reports)**

**Thu(Jun.29): Constellation 316, 3rd FL**

**Special Talk, Planning Session, WG activity**

**Joint Meeting with MARS/RTESS**
**Thursday, June 29, 2006**
**11:30-12:00 (Michelangelo A, 2nd FL)**

# Document Number

- robotics/2006-06-03 Final Agenda (Tetsuo Kotoku)
  robotics/2006-06-04 Boston Meeting Minutes [approved] (Hung Pham)
- robotics/2006-06-05 Roadmap (Tetsuo Kotoku)
- robotics/2006-06-06 Robotics Services WG: Introduction to RUPI (Soo-Young Chi)
  robotics/2006-06-07 Robotics Services WG: Definition of Functional Services (Olivier Lemaire)
- robotics/2006-06-08 Profile WG: Discussion on profile standardization (Seung-Ik Lee)
- robotics/2006-06-09 Steering Committee Presentation (Tetsuo Kotoku)
- robotics/2006-06-10 Publicity SC: Flyer discussion (Masayoshi Yokomachi)
- robotics/2006-06-11 Publicity SC: Flyer rough draft (Abheek Bose)
- robotics/2006-06-12 Robotic Services WG: POEM - an implementation of position estimation module (Takashi Tsubouchi)
- robotics/2006-06-13 Robotic Services WG: SAIT Proposal of Standards for Localization based on MDA (Yeon-Ho Kim)
- robotics/2006-06-14 Robotic Services WG: Issues on Localization Services (Wonpil Yu)
- robotics/2006-06-15 Robotic Services WG: Discussion on Localization Technology (Olivier Lemaire)
- robotics/2006-06-16 Spur - a Locomotion Command System for Mobile Robot (Takashi Tsubouchi)
- robotics/2006-06-17 Robot Modeling Framework (Abheek Bose)
- robotics/2006-06-18 Infrastructure WG: Understanding RSCA with Example (Seongsoo Hong)
- robotics/2006-06-19 Infrastructure WG: Agenda for Infrastructure WG (Saehwa Kim)
- robotics/2006-06-20 Infrastructure WG: Infrastructure WG minutes

# Document Number

robotics/2006-06-21 Plenary Opening / Ending Presentation (Tetsuo Kotoku)

robotics/2006-06-22 Robobusiness2006 Presentation (Jon Siegel)

robotics/2006-06-23 Robotics Device and Data Profile WG Report (Seung-Ik Lee)

robotics/2006-06-24 Robotic Services WG Report (Olivier Lemaire)

robotics/2006-06-25 Infrastructure WG Report (Rick Warren)

robotics/2006-06-26 Introduction to RTC (Rick Warren)

robotics/2006-06-27 Contact Report: ISO184 / SC2 (Makoto Mizukawa)

robotics/2006-06-28 Contact Report: Introduction of JAUS for the benefit of Robotic Standardization (Wataru Inamura)

robotics/2006-06-29 Contact Report: KIRSF (Yun-Koo Chung)

robotics/2006-06-30 Flyer first Draft candidate #1 (Abheek Bose)

robotics/2006-06-31 Flyer first Draft candidate #2 (Abheek Bose)

robotics/2006-06-32 Agenda Coordination for Anaheim Meeting (Tetsuo Kotoku)

robotics/2006-06-33 Introduction to OMG Ontology PSIG (Elisa Kandall)

robotics/2006-06-34 DTC Report Presentation (Hung Pham)

robotics/2006-06-35 Meeting Minutes - DRAFT (Hung Pham and Olivier Lemaire)

# Publicity Activities

- 4 page fly sheet

  Draft of Abheek@ADA Software

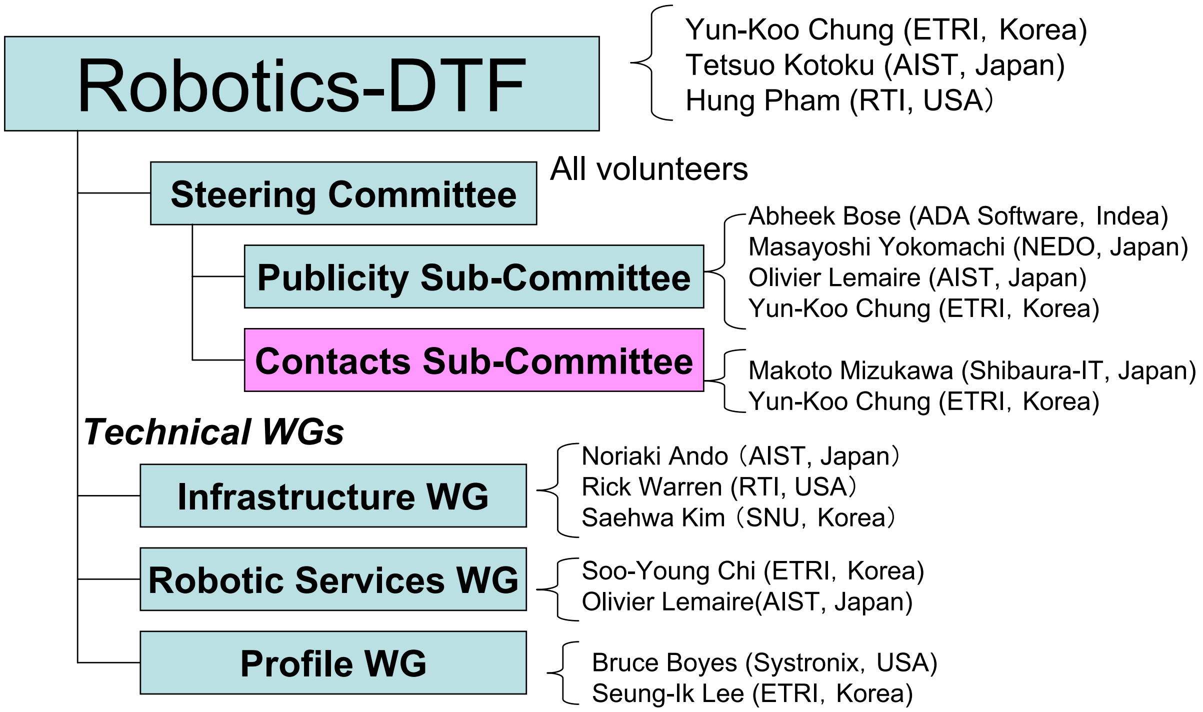
  **Abheek@ADA Soft,**
  **Olivier@AIST,**
  **Chung@ETRI,**
  **Yokomachi@NEDO**

**Action:**

Send each organization logo to Abheek.

4 page fly sheet will be authorized in Boston

# Organization



Robotics-DTF

Yun-Koo Chung (ETRI, Korea)
Tetsuo Kotoku (AIST, Japan)
Hung Pham (RTI, USA)

Steering Committee

All volunteers

Publicity Sub-Committee

Abheek Bose (ADA Software, Indea)
Masayoshi Yokomachi (NEDO, Japan)
Olivier Lemaire (AIST, Japan)
Yun-Koo Chung (ETRI, Korea)

Contacts Sub-Committee

Makoto Mizukawa (Shibaura-IT, Japan)
Yun-Koo Chung (ETRI, Korea)

*Technical WGs*

Infrastructure WG

Noriaki Ando（AIST, Japan）
Rick Warren (RTI, USA)
Saehwa Kim（SNU, Korea）

Robotic Services WG

Soo-Young Chi (ETRI, Korea)
Olivier Lemaire(AIST, Japan)

Profile WG

Bruce Boyes (Systronix, USA)
Seung-Ik Lee (ETRI, Korea)

# Publicity Activities

- IROS2006 Workshop
  October 9-15, Beijing, China
  http://www.iros2006.org/
  Kotoku@AIST, Chung@ETRI, Mizukawa@Sibaura-IT

**• Call for volunteer (Program Committee)**
**• Call for paper (submission due: Aug. 1st)**

- SICE-ICASE International Joint Conference
  October 18-21, Pusan, Korea
  http://sice-iccas.org/
  Mizukawa@Sibaura-IT

**• Call for Participation (Organized Session)**

# Next Meeting Agenda

Sep. 26-30, 2006 (Anaheim, CA, USA)

**Monday :**

**Steering Committee**

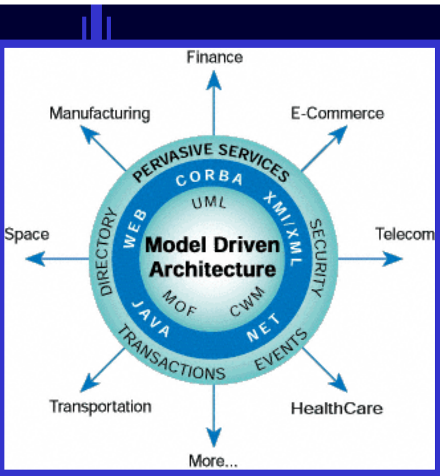
**Monday-Tuesday, Thursday :**

**WG activities**

**Wednesday :**

**Robotics-DTF Plenary Meeting**

- Guest and Member Presentation
- WG reports & Roadmap discussion
- Contact reports
- DTC report - Draft



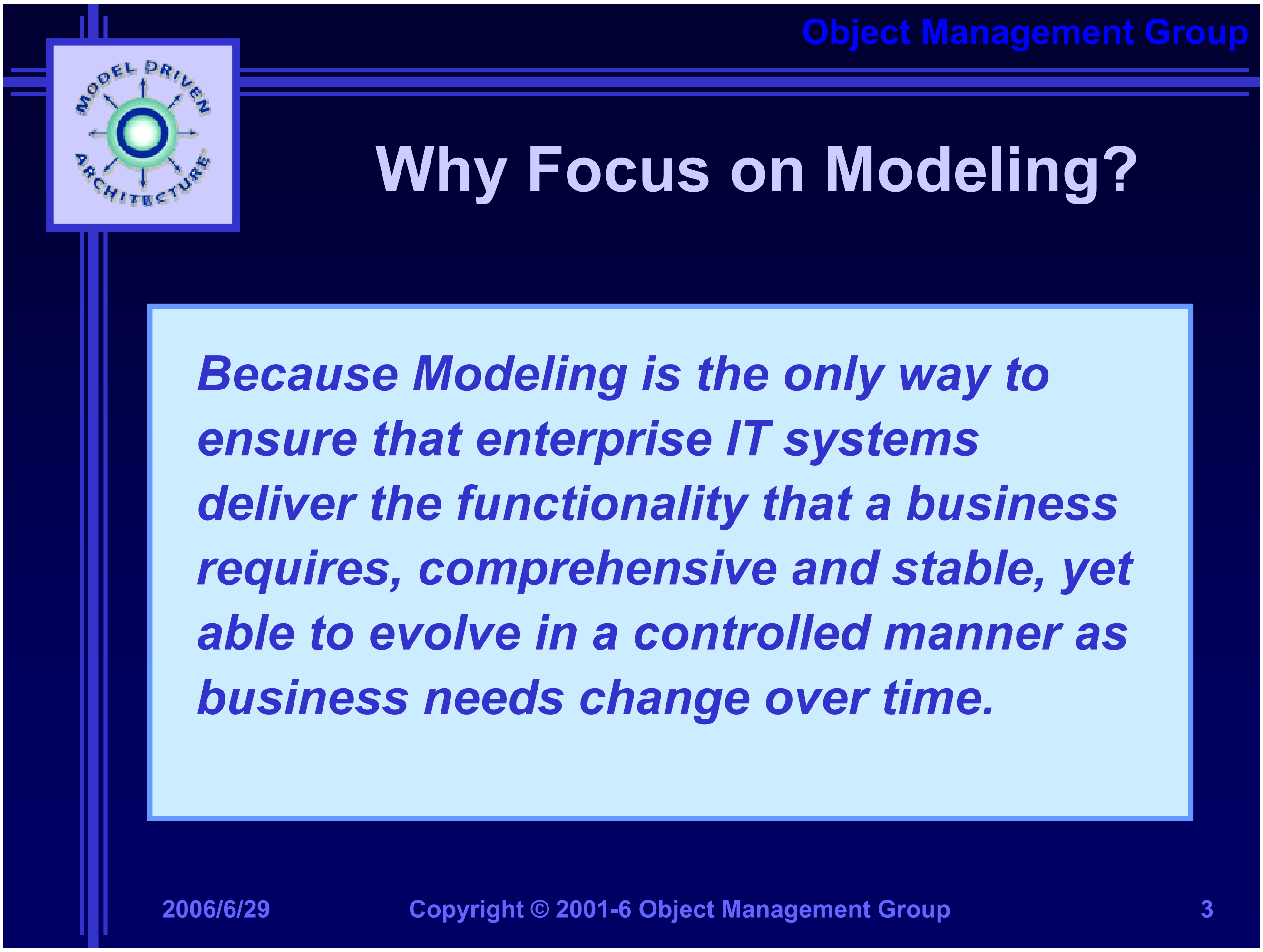
# Model Driven Architecture (MDA) Software Development in Robotics

October, 2005

Jon Siegel, Ph.D.
Vice President, Technology Transfer
Object Management Group
siegel@omg.org
www.omg.org
+1 781-444-0404

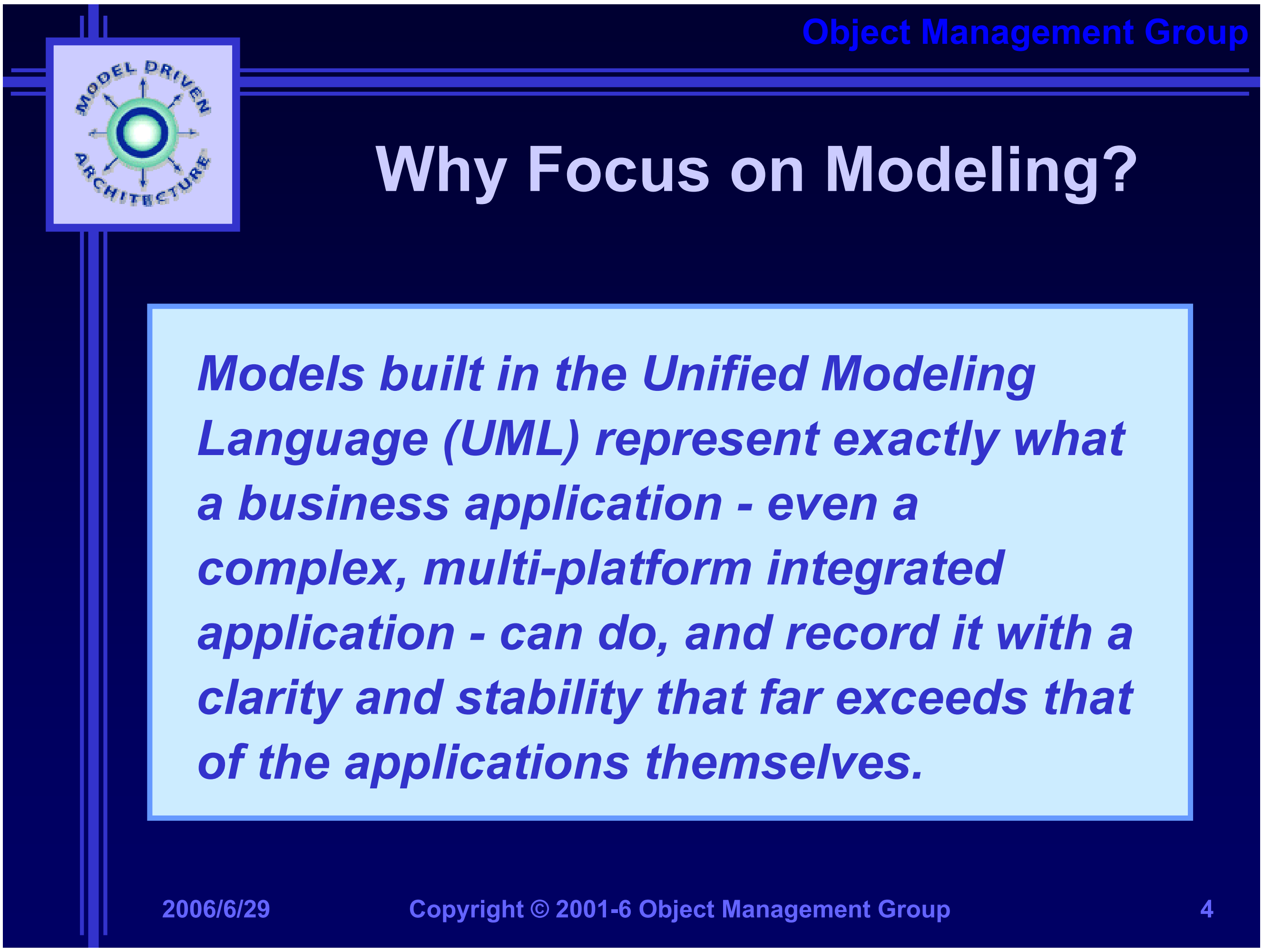
# What is OMG?

- Object Management Group - 16-year-old not-for-profit Computer Industry Standards Consortium
- Home of UML, the Industry's Modeling Standard
- and the Model Driven Architecture (MDA)
- Open Membership and Adoption Process
  - One-member, One-vote
- Specifications Available Free on our Website
- Buy Implementing Products from Vendors
  - Vendors may be OMG members, or may not
- Over 500 members including Companies, Government Agencies, Universities



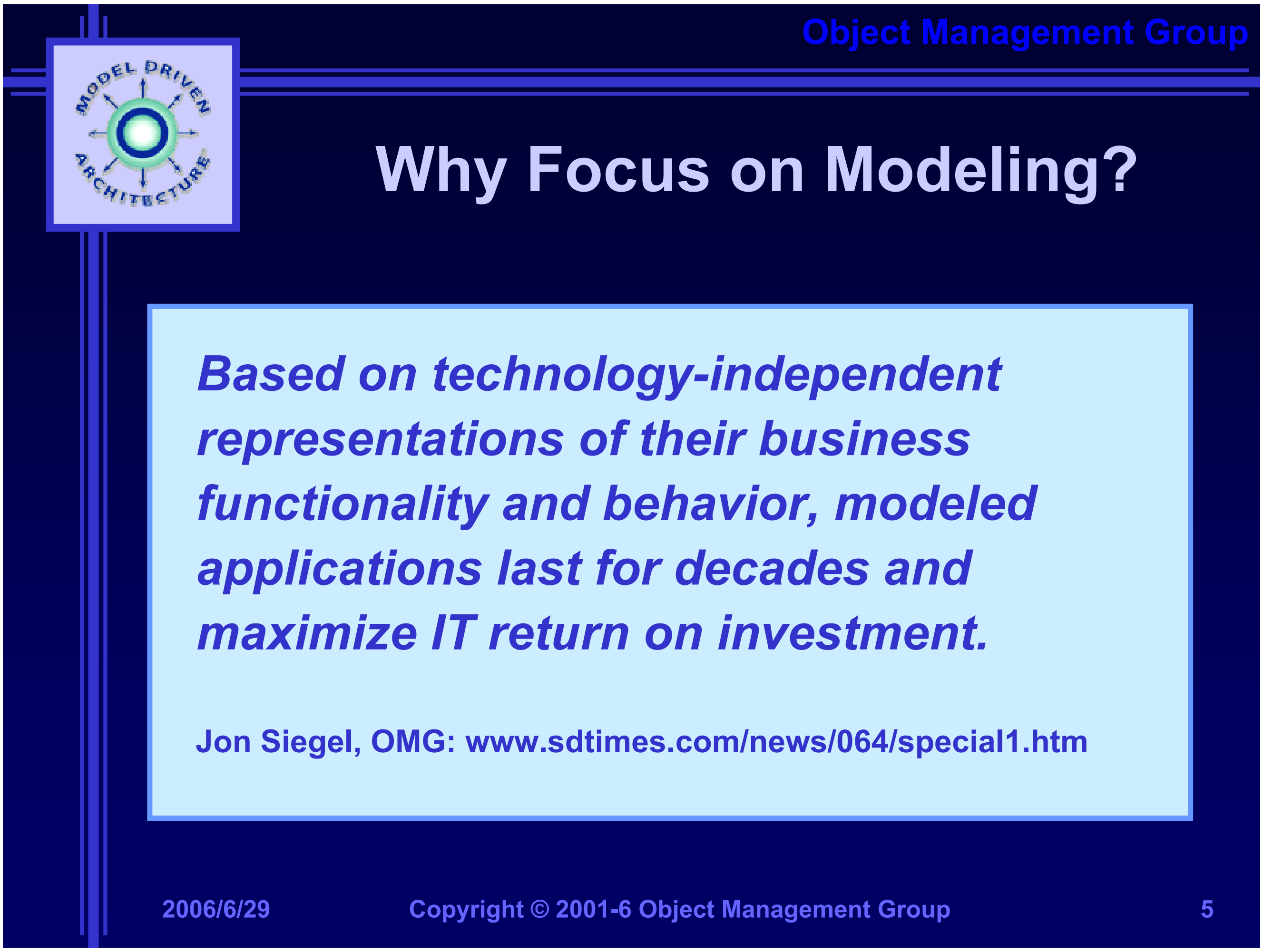
## Why Focus on Modeling?

***Because Modeling is the only way to ensure that enterprise IT systems deliver the functionality that a business requires, comprehensive and stable, yet able to evolve in a controlled manner as business needs change over time.***



## Why Focus on Modeling?

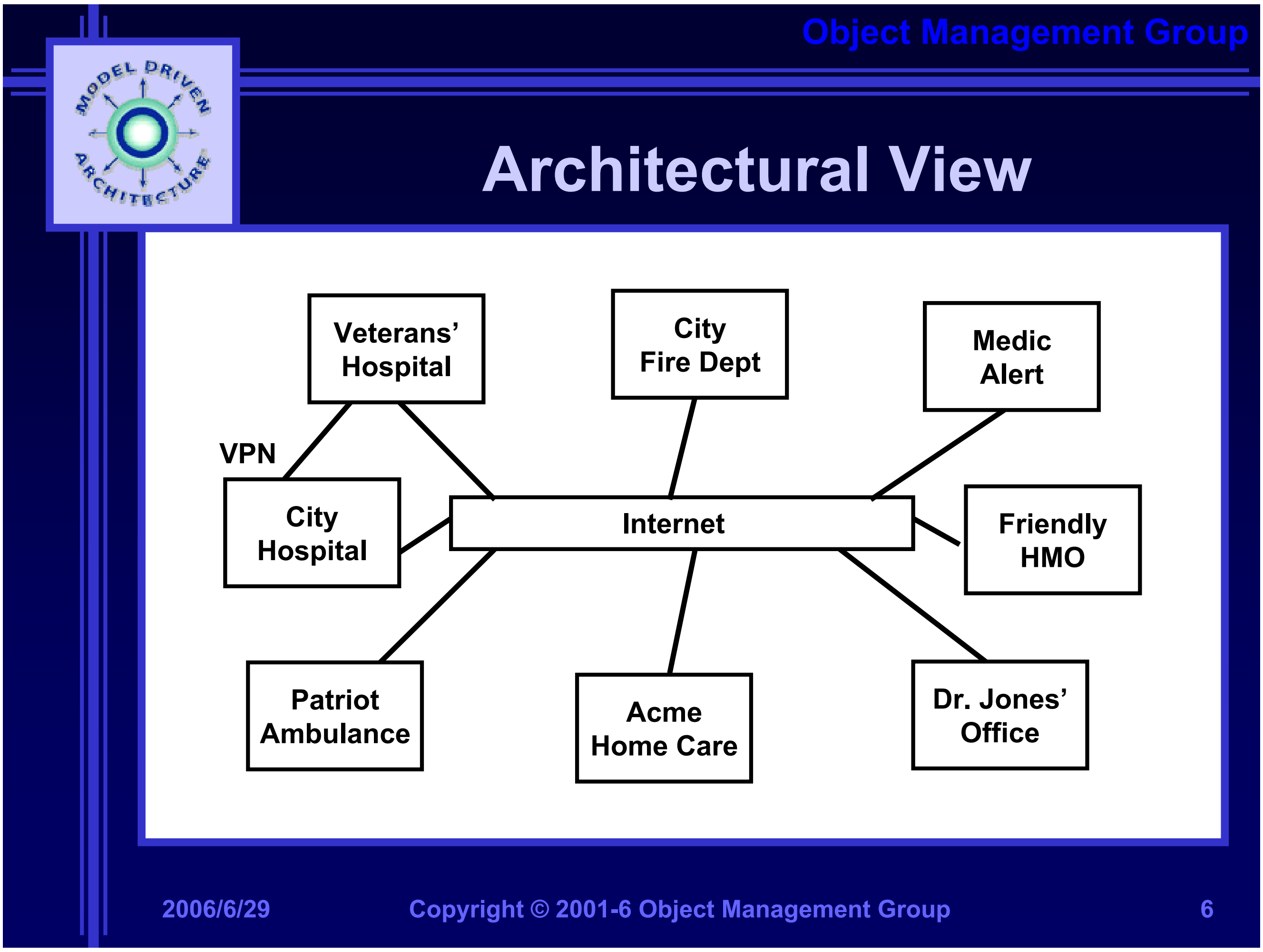
***Models built in the Unified Modeling Language (UML) represent exactly what a business application - even a complex, multi-platform integrated application - can do, and record it with a clarity and stability that far exceeds that of the applications themselves.***



# Why Focus on Modeling?

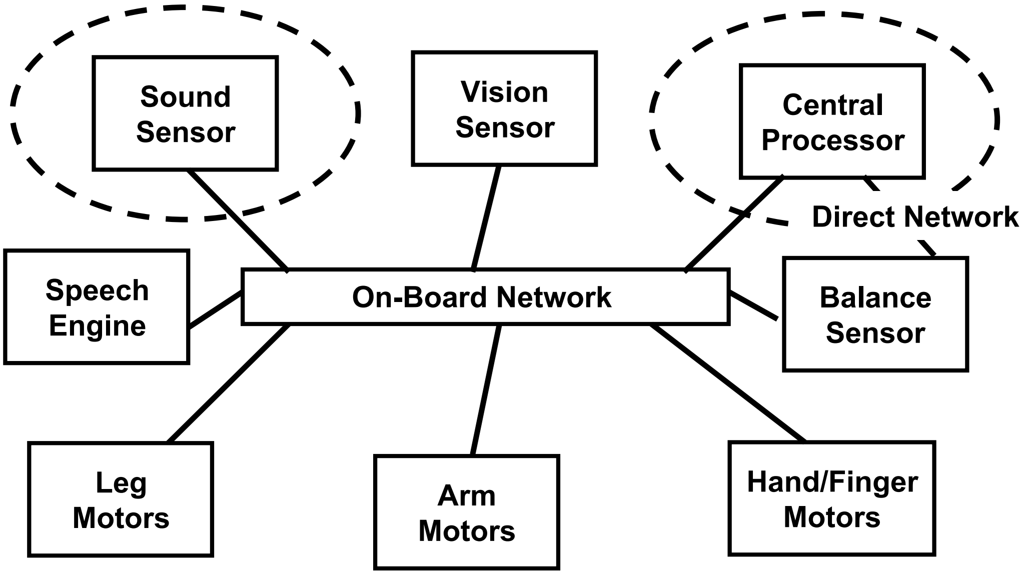
*Based on technology-independent representations of their business functionality and behavior, modeled applications last for decades and maximize IT return on investment.*

Jon Siegel, OMG: www.sdtimes.com/news/064/special1.htm



# Architectural View

# Robotics Version!



Sound Sensor

Vision Sensor

Central Processor

Direct Network

Speech Engine

On-Board Network

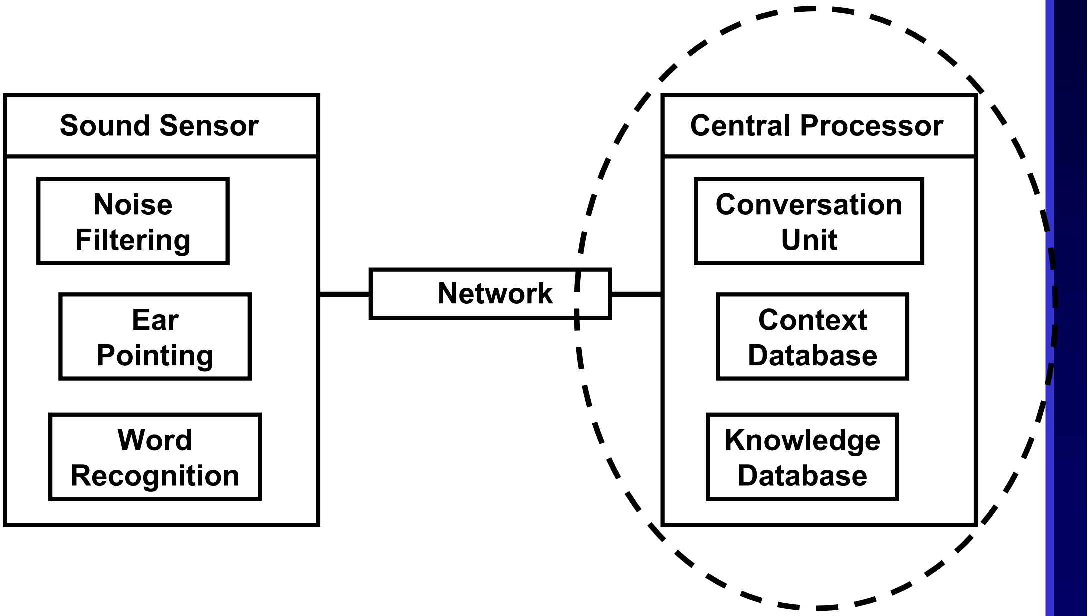
Balance Sensor

Leg Motors

Arm Motors

Hand/Finger Motors

# Zoomed In, Still Architectural



Sound Sensor

Noise Filtering

Ear Pointing

Word Recognition

Network

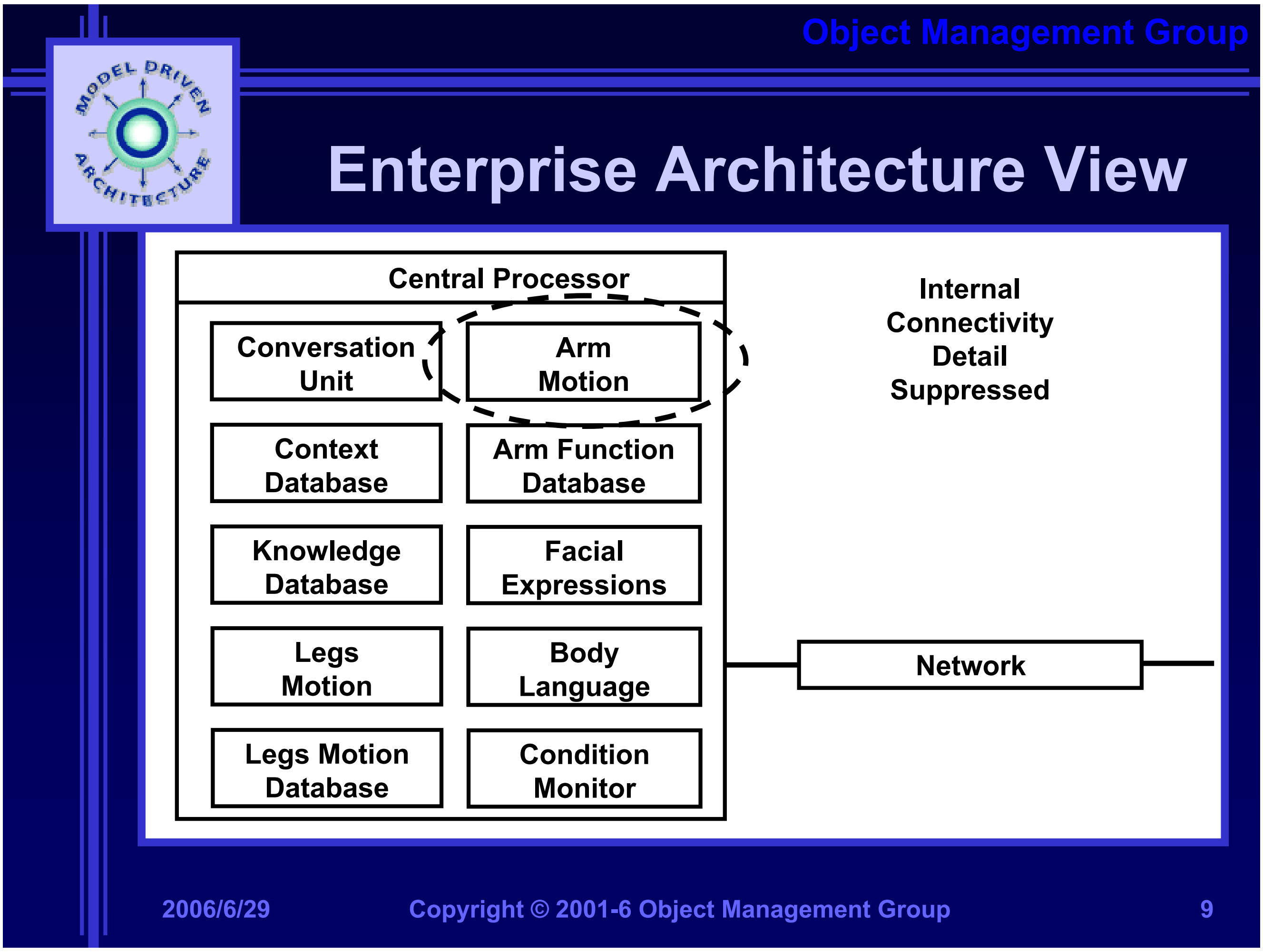
Central Processor

Conversation Unit

Context Database

Knowledge Database

# Enterprise Architecture View



Central Processor

Conversation Unit

Arm Motion

Context Database

Arm Function Database

Knowledge Database

Facial Expressions

Legs Motion

Body Language

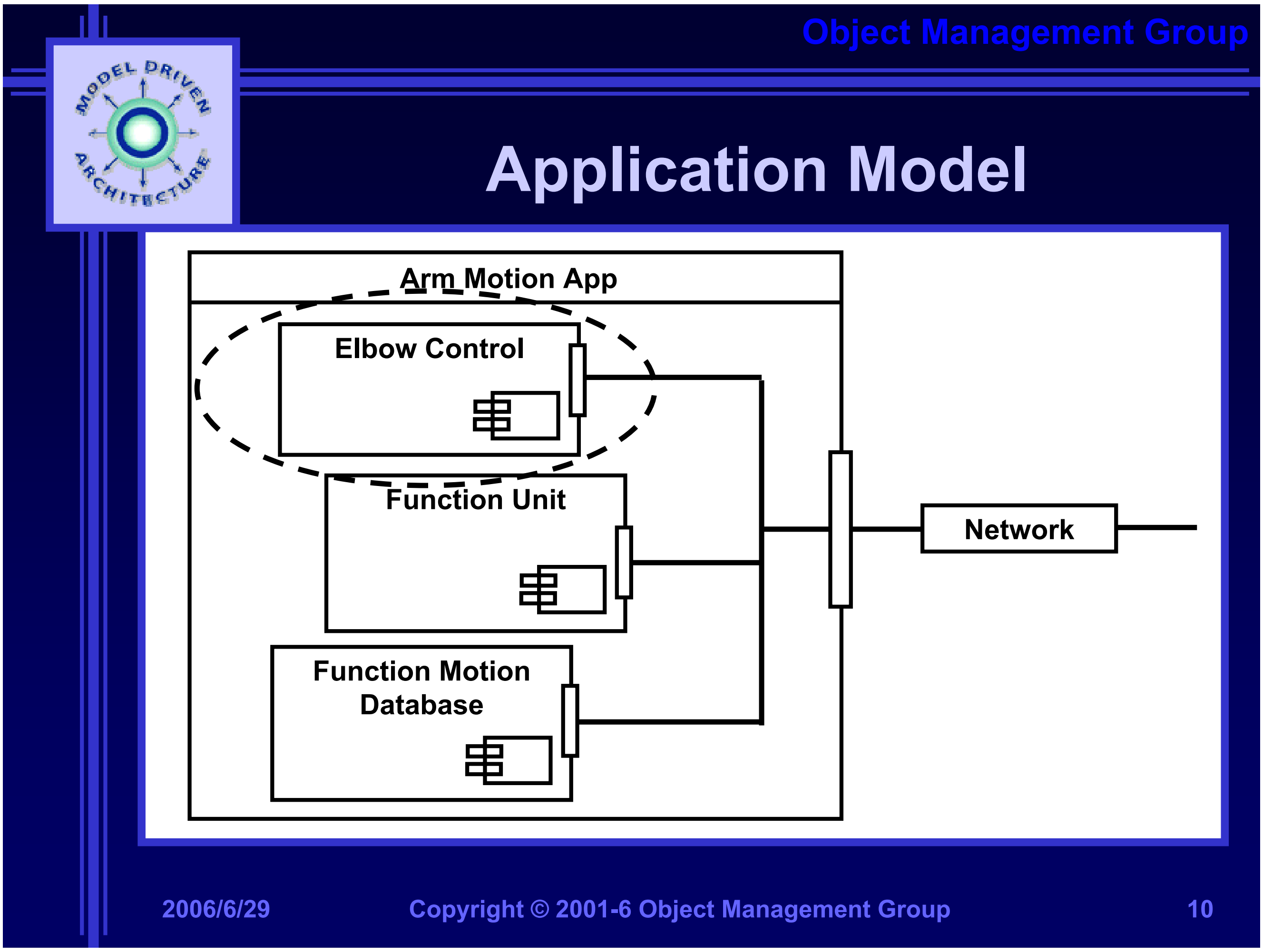
Legs Motion Database

Condition Monitor

Internal Connectivity Detail Suppressed

Network

# Application Model



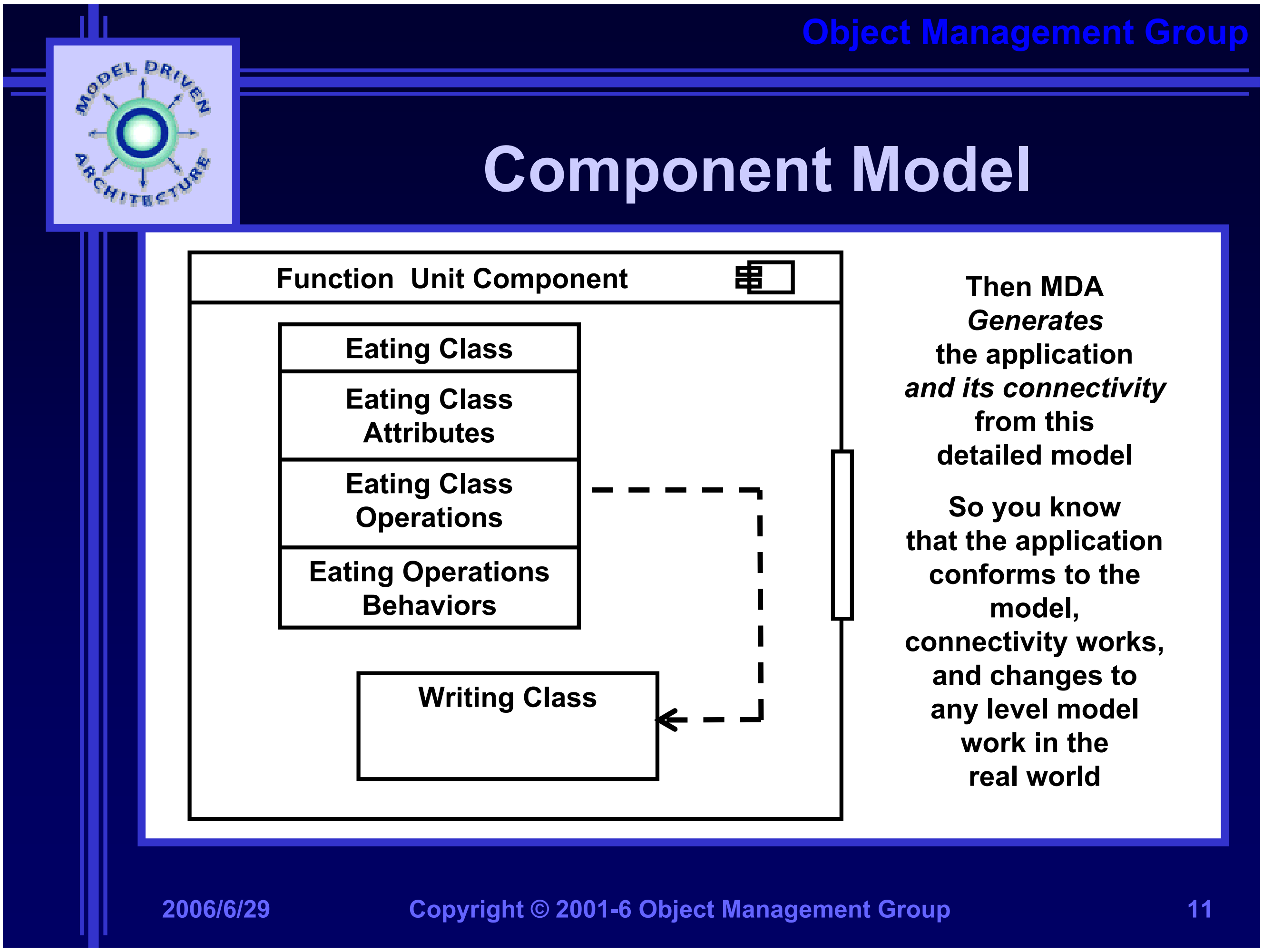
Arm Motion App

Elbow Control

Function Unit

Function Motion Database

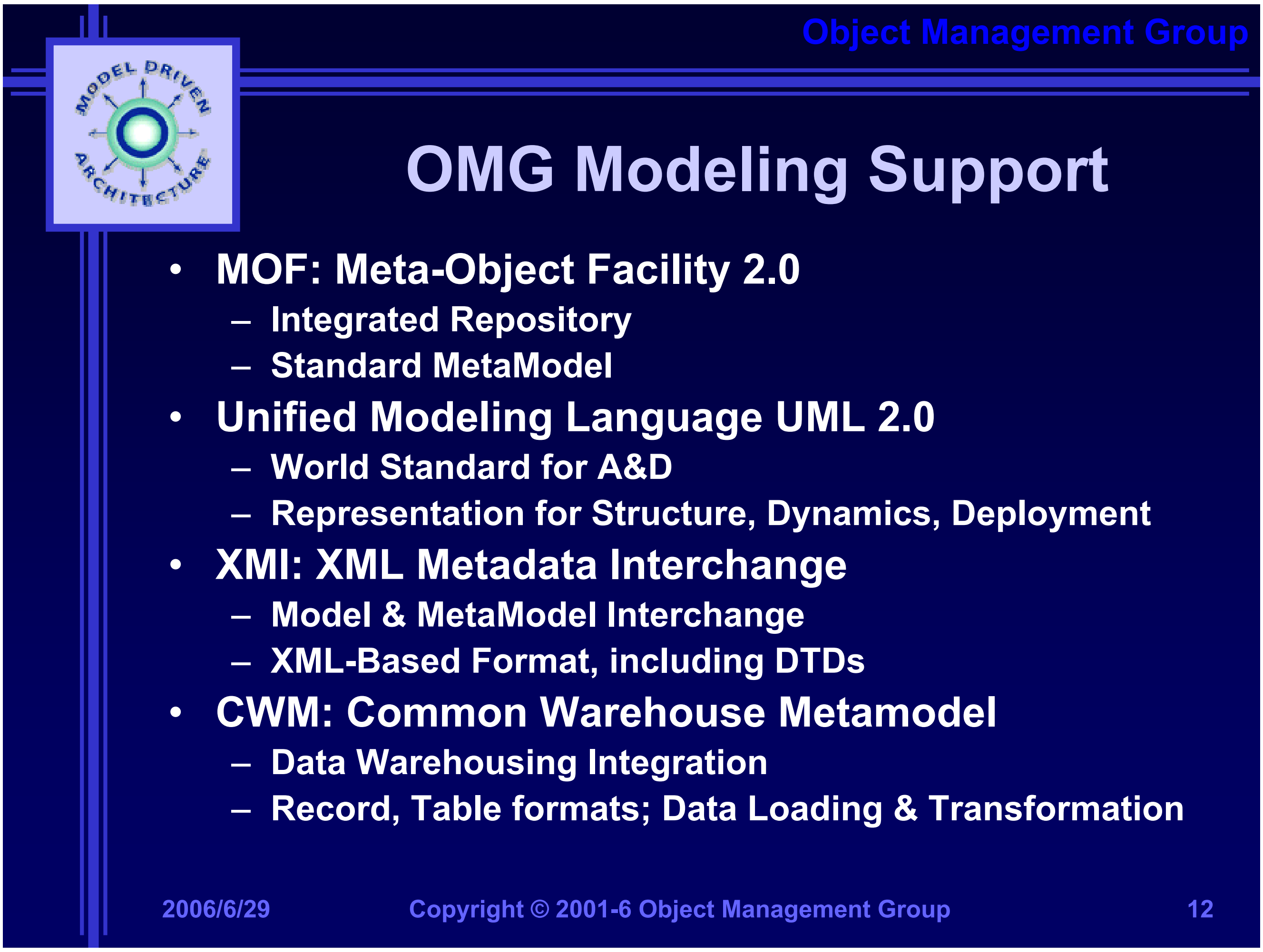
Network



# Component Model

Then MDA *Generates* the application *and its connectivity* from this detailed model

So you know that the application conforms to the model, connectivity works, and changes to any level model work in the real world

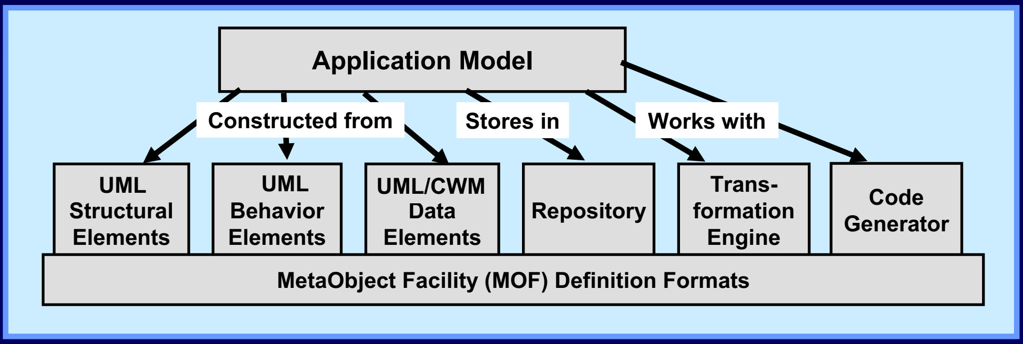


# OMG Modeling Support

- MOF: Meta-Object Facility 2.0
  - Integrated Repository
  - Standard MetaModel
- Unified Modeling Language UML 2.0
  - World Standard for A&D
  - Representation for Structure, Dynamics, Deployment
- XMI: XML Metadata Interchange
  - Model & MetaModel Interchange
  - XML-Based Format, including DTDs
- CWM: Common Warehouse Metamodel
  - Data Warehousing Integration
  - Record, Table formats; Data Loading & Transformation

# MOF - Foundation for Modeling

- MOF standardizes the basis for the elements that modeling languages define for you to model with
- Based on MOF, all of these diverse model elements can share repositories and interchange models among compliant tools:
  - Interchange of models and metamodels among toolsets
  - UML, MOF Itself, CWM, SPEM, XMI, UML Profiles
- And Especially, MOF supports the MDA!



Application Model

Constructed from

Stores in

Works with

UML Structural Elements

UML Behavior Elements

UML/CWM Data Elements

Repository

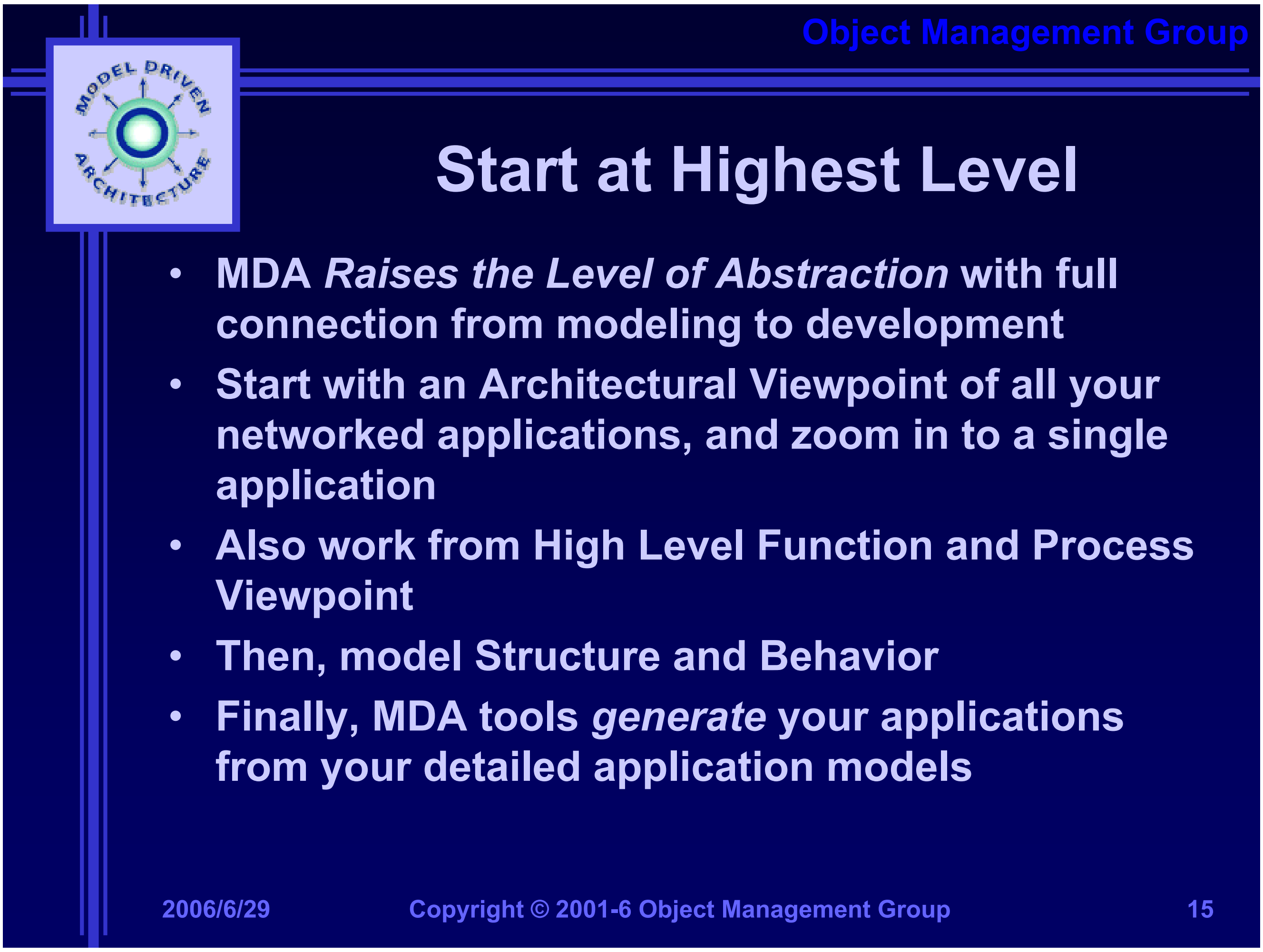
Trans-formation Engine

Code Generator

MetaObject Facility (MOF) Definition Formats

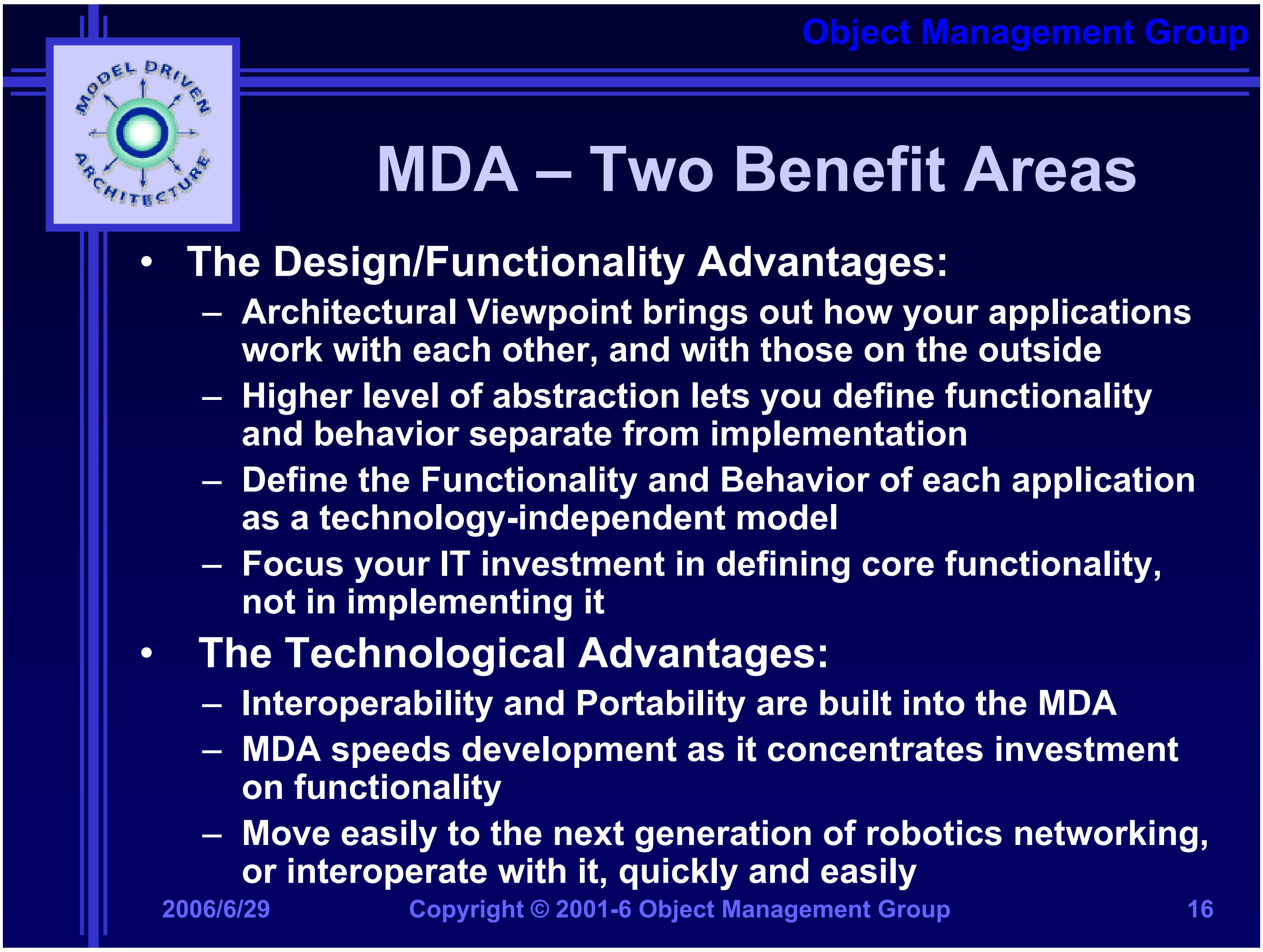
# UML – The Modeling Standard

- Integrates all the modeling you need to do
  - Functional and Business Modeling
  - Architectural/Deployment Modeling
  - Application Structure and Behavior
  - Component-Based Applications
  - Classes and Objects
  - Data Structures
  - Behavior, as State Machines, Data and Control Flow, Use Cases, more
  - The Industry Standard for Modeling



# Start at Highest Level

- MDA *Raises the Level of Abstraction* with full connection from modeling to development
- Start with an Architectural Viewpoint of all your networked applications, and zoom in to a single application
- Also work from High Level Function and Process Viewpoint
- Then, model Structure and Behavior
- Finally, MDA tools *generate* your applications from your detailed application models

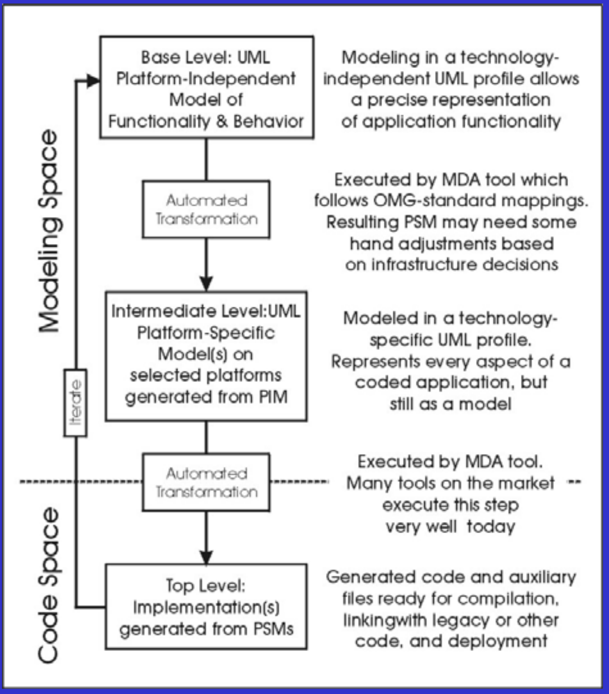


# MDA – Two Benefit Areas

- The Design/Functionality Advantages:
  - Architectural Viewpoint brings out how your applications work with each other, and with those on the outside
  - Higher level of abstraction lets you define functionality and behavior separate from implementation
  - Define the Functionality and Behavior of each application as a technology-independent model
  - Focus your IT investment in defining core functionality, not in implementing it
- The Technological Advantages:
  - Interoperability and Portability are built into the MDA
  - MDA speeds development as it concentrates investment on functionality
  - Move easily to the next generation of robotics networking, or interoperate with it, quickly and easily

# MDA: Designed for Efficiency

- Structure is a Spectrum progressing from Modeling at the Top to Code development at the bottom



Modeling Space
Code Space
Iterate
Base Level: UML Platform-Independent Model of Functionality & Behavior
Automated Transformation
Intermediate Level:UML Platform-Specific Model(s) on selected platforms generated from PIM
Automated Transformation
Top Level: Implementation(s) generated from PSMs

Modeling in a technology-independent UML profile allows a precise representation of application functionality

Executed by MDA tool which follows OMG-standard mappings. Resulting PSM may need some hand adjustments based on infrastructure decisions

Modeled in a technology-specific UML profile. Represents every aspect of a coded application, but still as a model

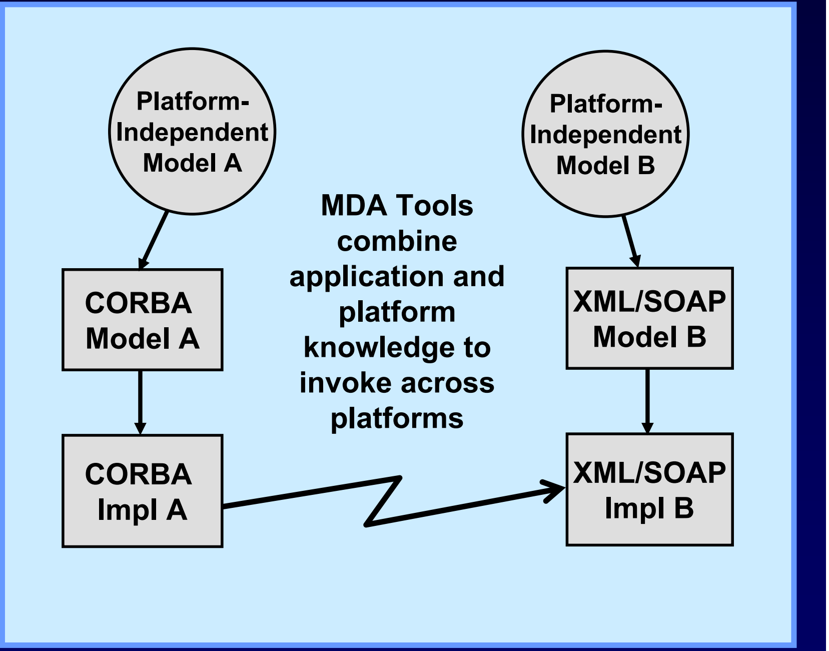
Executed by MDA tool. Many tools on the market -- execute this step very well today

Generated code and auxiliary files ready for compilation, linkingwith legacy or other code, and deployment

# MDA Applications Interoperate

MDA Tools will generate cross-platform invocations connecting either instances of a single MDA application, or one application to another.

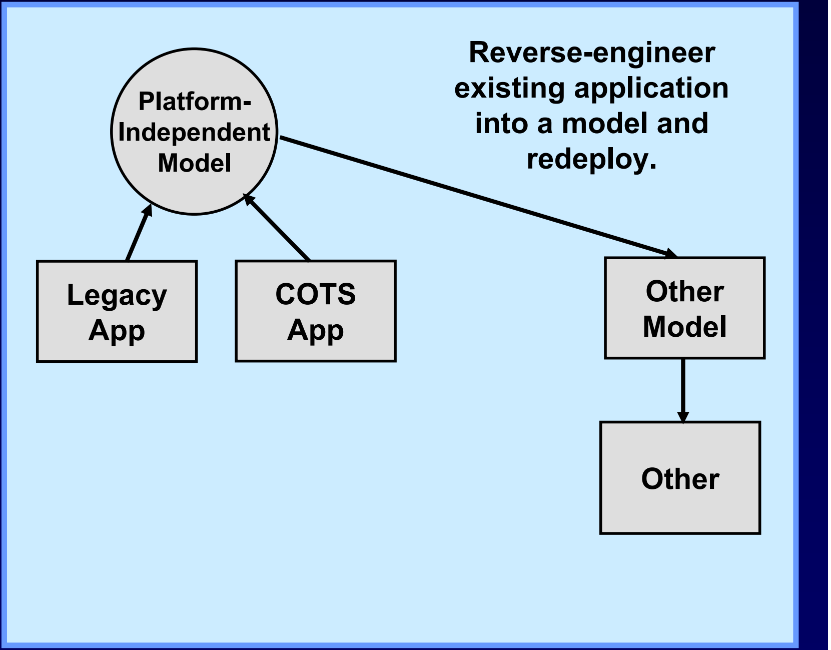
Standard *Pervasive Services* – directory, security, more – will also be accessed through cross-platform invocations where necessary.



MDA Tools combine application and platform knowledge to invoke across platforms

# Integrating Legacy & COTS

Tools for Reverse Engineering automate creation of models for re-integration on new platforms



Reverse-engineer existing application into a model and redeploy.

# MDA Specifications

- MDA Architecture (June 2003)
- UML 2.0 (Complete)
- UML Profiles:
  - Profile for EDOC (complete)
  - Profile for EAI (complete)
  - Profile for CORBA (complete)
  - Profile for EJB (JCP, complete)
- Support from XMI, CWM (complete)
- Pervasive Services (coming)
- Domain Specifications

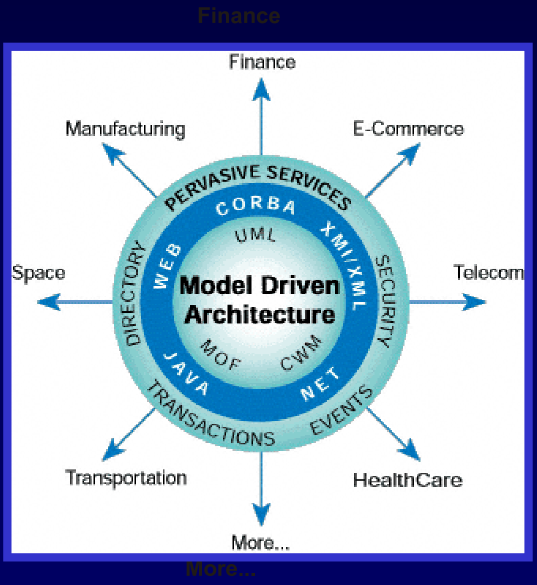
# MDA in Industry Standards

OMG (and other) Task Forces standardize Domain (Industry-Specific) Facilities as PIMs.

With implementations on multiple platforms, no technology or platform barriers prevent widespread adoption and use.

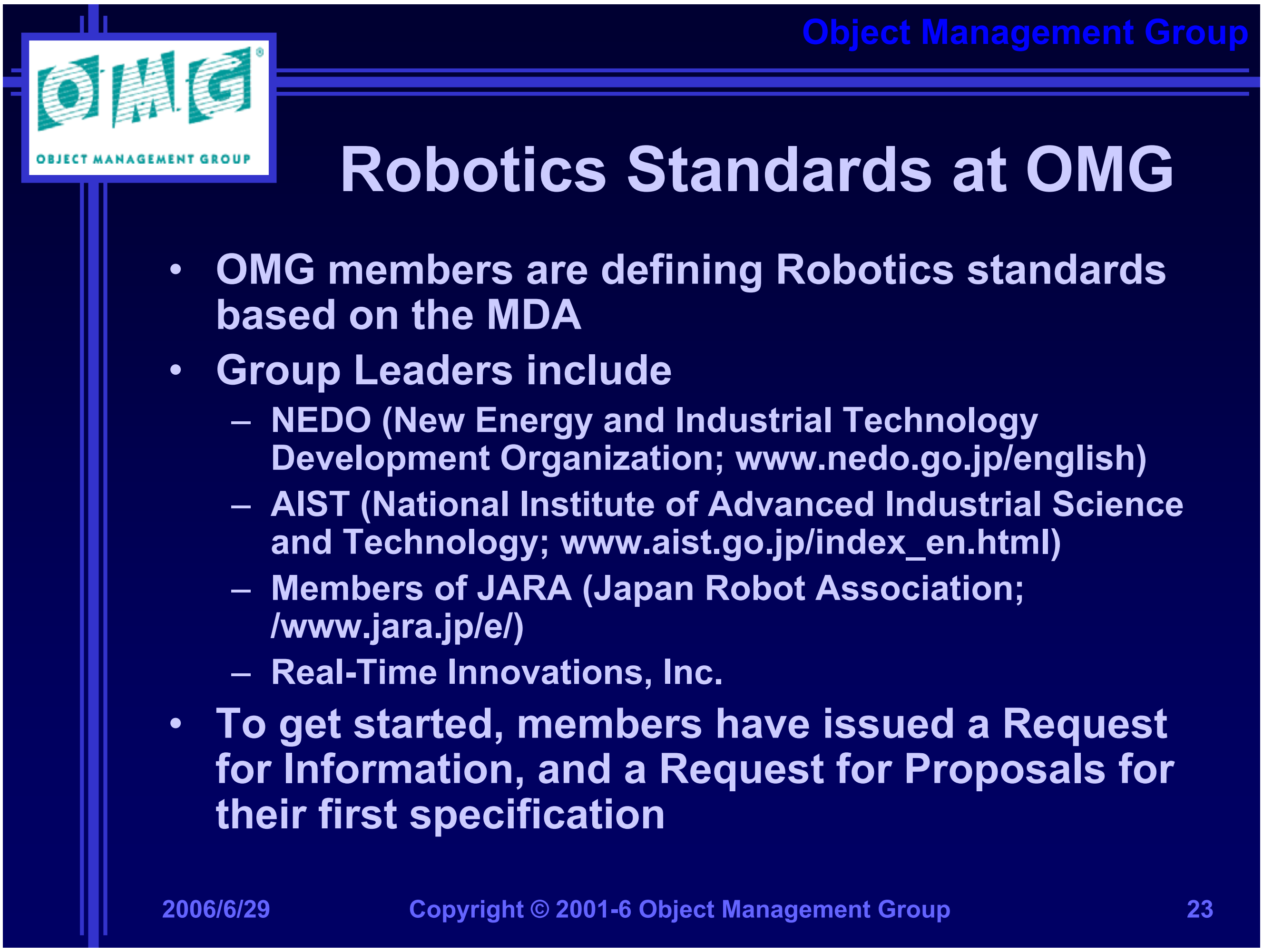
Interoperate cross-platform with other standard applications.

Both PIM and set of PSMs and interface code – on *every* mapped platform – become OMG standards.



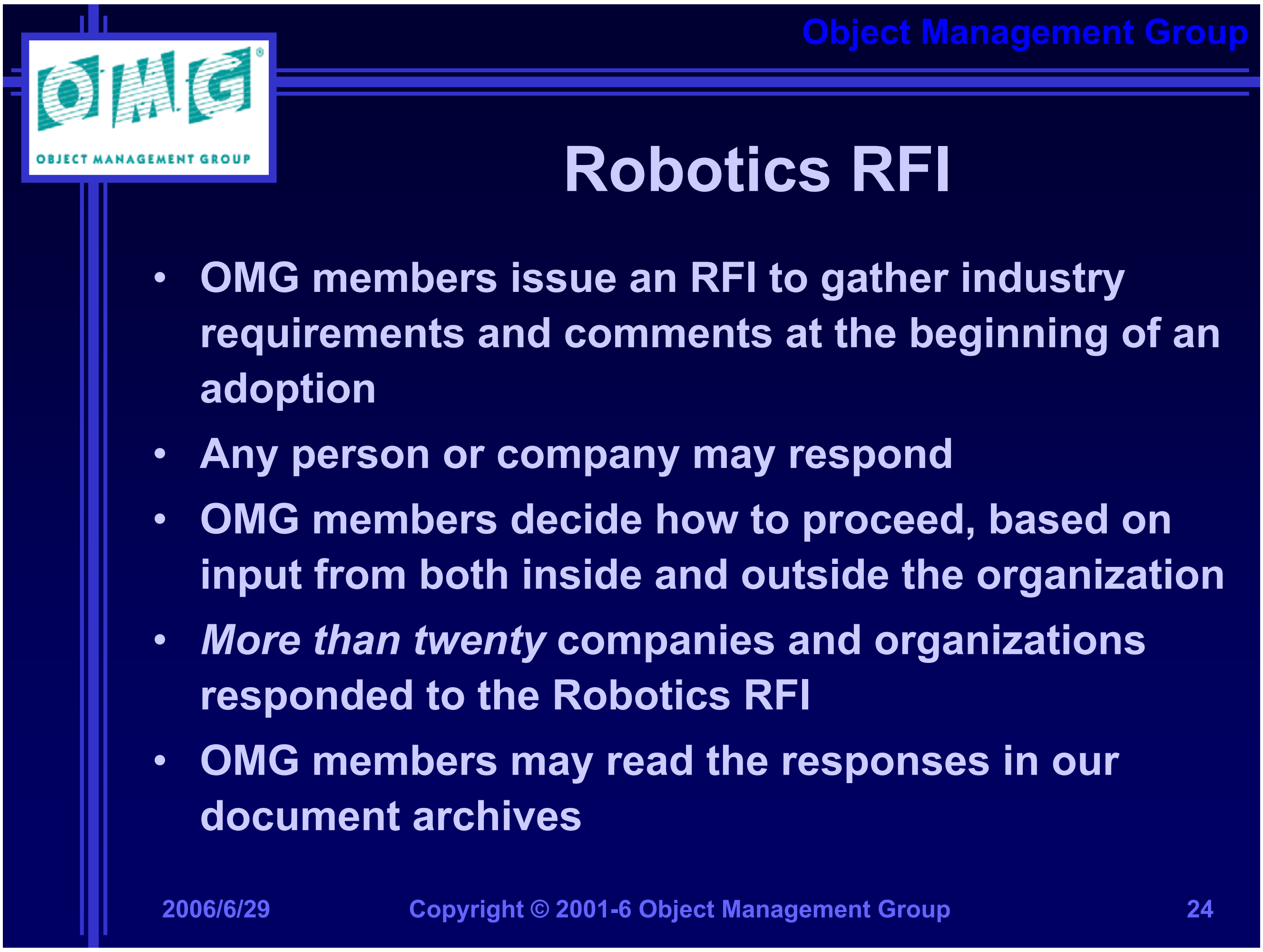
# Real-Time & Embedded Specs

- Middleware Specifications
  - Realtime CORBA (Static Scheduling)
  - Realtime CORBA (Dynamic Scheduling)
  - Extensible Transport Frameworks
  - CORBA/e (CORBA *for Embedded*)
- Components and Services Specifications
  - Lightweight CORBA Component Model
  - Lightweight Services
  - Lightweight Logging Service
  - Data Distribution Service
  - Online Upgrades
  - Unreliable Multicast
- DRE Modeling Specifications
  - UML Profile for Schedulability, Performance, and Time
  - UML Profile for Quality of Service and Fault Tolerance



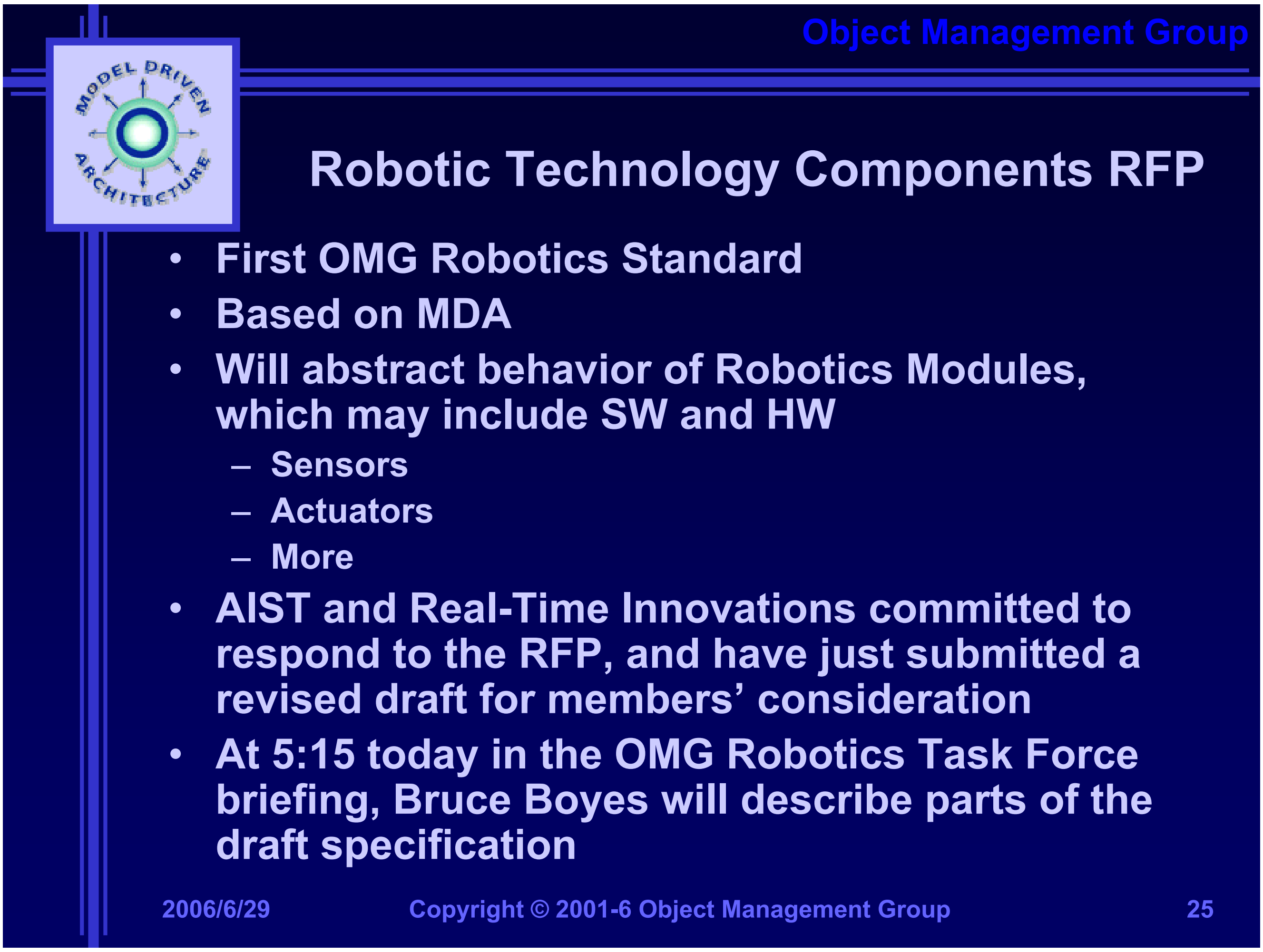
# Robotics Standards at OMG

- OMG members are defining Robotics standards based on the MDA
- Group Leaders include
  - NEDO (New Energy and Industrial Technology Development Organization; www.nedo.go.jp/english)
  - AIST (National Institute of Advanced Industrial Science and Technology; www.aist.go.jp/index_en.html)
  - Members of JARA (Japan Robot Association; /www.jara.jp/e/)
  - Real-Time Innovations, Inc.
- To get started, members have issued a Request for Information, and a Request for Proposals for their first specification



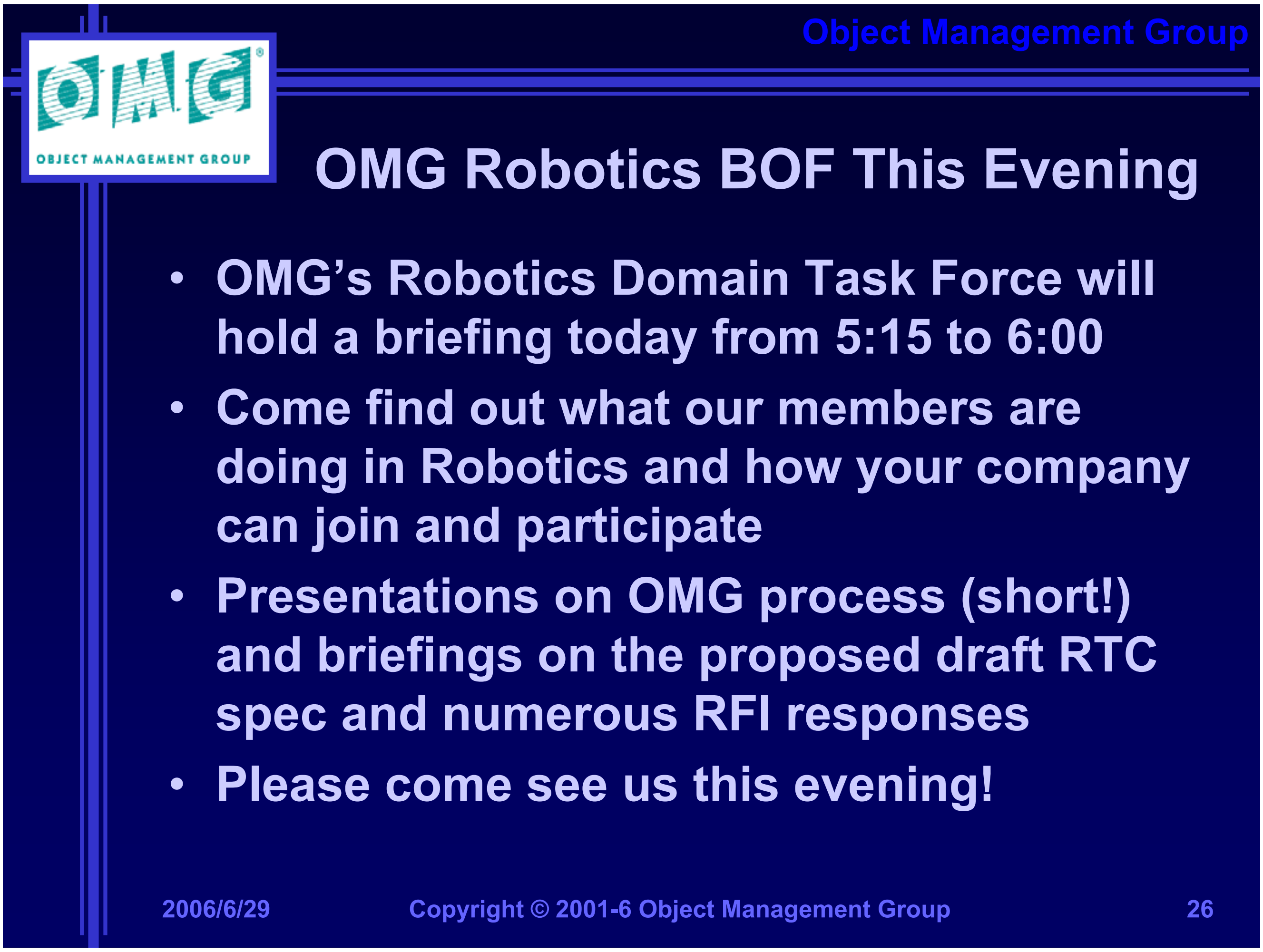
# Robotics RFI

- OMG members issue an RFI to gather industry requirements and comments at the beginning of an adoption
- Any person or company may respond
- OMG members decide how to proceed, based on input from both inside and outside the organization
- *More than twenty* companies and organizations responded to the Robotics RFI
- OMG members may read the responses in our document archives



## Robotic Technology Components RFP

- First OMG Robotics Standard
- Based on MDA
- Will abstract behavior of Robotics Modules, which may include SW and HW
  - Sensors
  - Actuators
  - More
- AIST and Real-Time Innovations committed to respond to the RFP, and have just submitted a revised draft for members’ consideration
- At 5:15 today in the OMG Robotics Task Force briefing, Bruce Boyes will describe parts of the draft specification



## OMG Robotics BOF This Evening

- OMG’s Robotics Domain Task Force will hold a briefing today from 5:15 to 6:00
- Come find out what our members are doing in Robotics and how your company can join and participate
- Presentations on OMG process (short!) and briefings on the proposed draft RTC spec and numerous RFI responses
- Please come see us this evening!

# OMG: Background

- About 500 member companies; Not-for-profit open-membership specifications consortium.
- Founded April 1989 - Seventeen Years Old
- Small staff (22 full time); no internal development. Representatives in Germany, Japan.
- Home of the Model Driven Architecture and MDA-Based Standards, Maximizing IT ROI by extending software and infrastructure lifetime across technology transitions

# Worldwide Scope

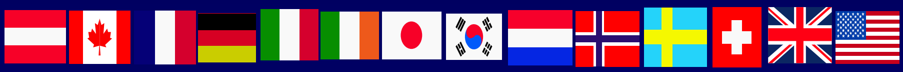
Alcatel
Artisan
BEA Systems
Bank of America
Boeing Corp.
Borland
BAE Systems
CBOE
Charles Schwab

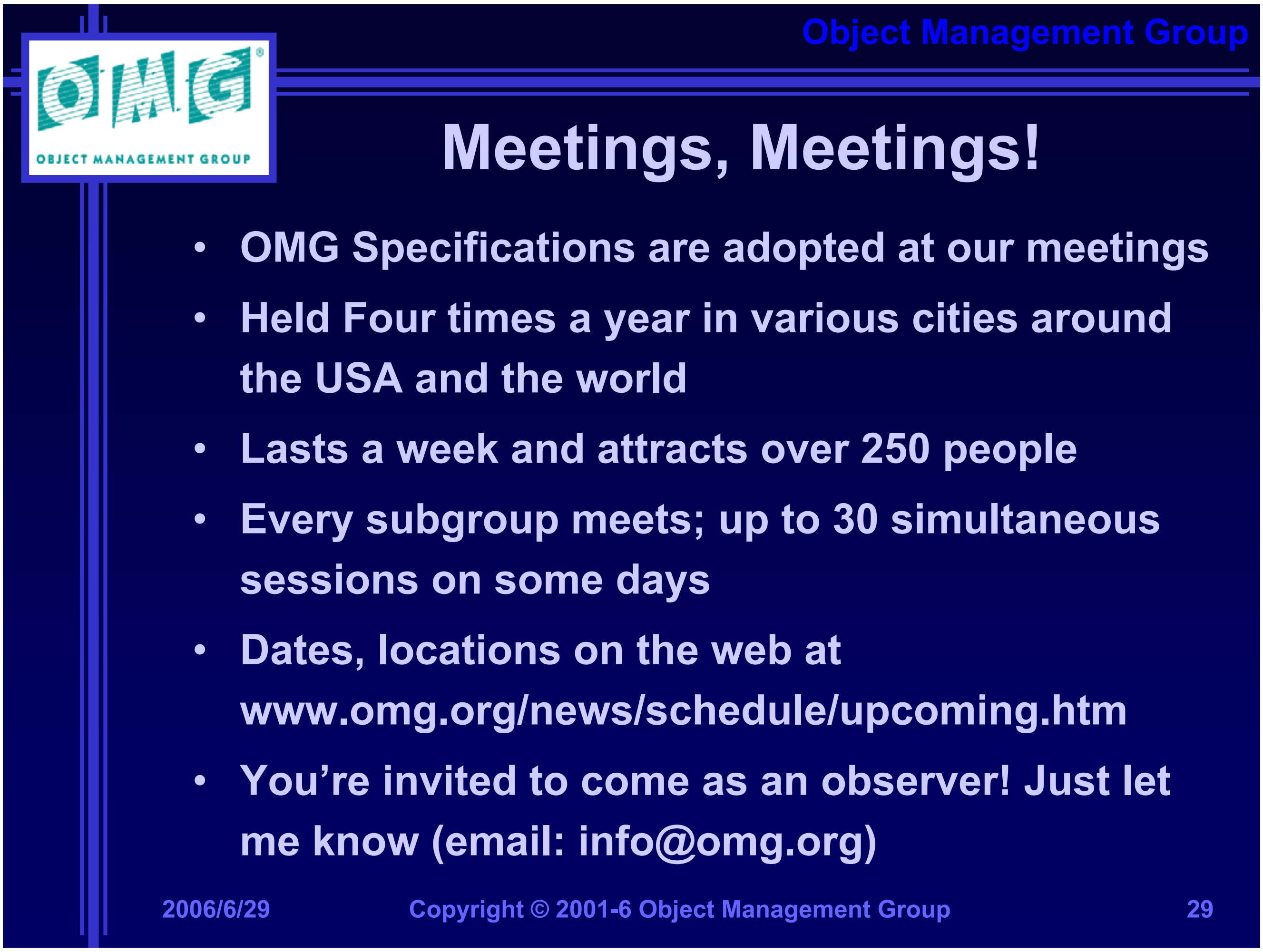
Computer Assocs
Compuware
Daimler-Benz AG
Deere & Co.
EDS
Ericsson
Fair Isaac
Fujitsu
GCHQ

Fraunhofer Fokus
HP
Hitachi
IBM
IONA
Lockheed
MetaMatrix
Mitre
Motorola

NEC
NIST
Nokia
Northrup
Oracle
PrismTech
Raytheon
Sandia
SAP AG

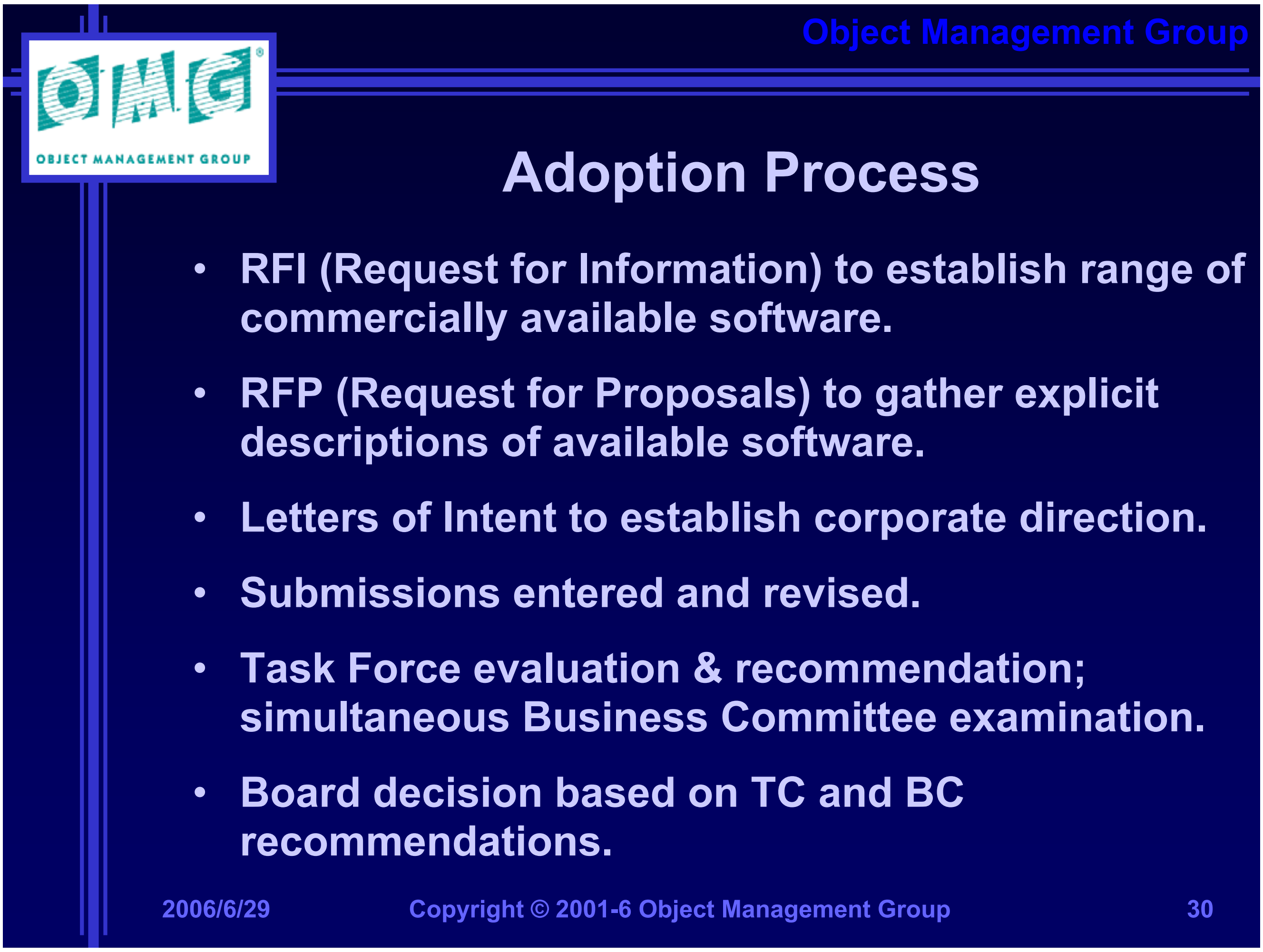
Siemens
Software AG
Sony
Sun
Telelogic
Thales
Unisys
W3C
Workflow Mgmt





# Meetings, Meetings!

- OMG Specifications are adopted at our meetings
- Held Four times a year in various cities around the USA and the world
- Lasts a week and attracts over 250 people
- Every subgroup meets; up to 30 simultaneous sessions on some days
- Dates, locations on the web at www.omg.org/news/schedule/upcoming.htm
- You’re invited to come as an observer! Just let me know (email: info@omg.org)



# Adoption Process

- RFI (Request for Information) to establish range of commercially available software.
- RFP (Request for Proposals) to gather explicit descriptions of available software.
- Letters of Intent to establish corporate direction.
- Submissions entered and revised.
- Task Force evaluation & recommendation; simultaneous Business Committee examination.
- Board decision based on TC and BC recommendations.

# Availability

Innovative approach for selection of standard interfaces to adopt:

1. OMG adopts & publishes MDA PIMs and PSMs, and Implementation Interface Specifications.
2. Implementations of the Interface Specifications must be available commercially from OMG Platform, Domain, or Contributing member.
3. MDA PIMs and PSMs, and Interface Specifications, are freely available to members and non-members alike.
4. MDA PIMs and PSMs, and Interface Specifications chosen from existing products or prototypes in a competitive selection process.

# OMG Links & Contacts

- OMG Homepage:
  - http://www.omg.org
- Download our specifications:
  - http://www.omg.org/specifications
- MDA Central:
  - http://www.omg.org/mda
- MDA Executive overview:
  - http://www.omg.org/mda/executive_overview.htm
- Find out about UML:
  - http://www/omg.org/uml
- Find out about CWM:
  - http://www.omg.org/cwm
- Contact OMG:
  - Email info@omg.org or siegel@omg.org

# OMG Robotics DTF Robotic Devices and Data Profile Working Group Progress Report

**Boston, Jun 28 2006**

**Seung-Ik Lee and Bruce Boyce, co-chairs**

## Introduction to Robotic Device and Data Profile Working Group

- **The Working Group met Jun 26, 2006**
- **Discussions**
  - **Typical devices (List of typical devices)**
  - **Approach (Top-down or bottom-up)**
  - **Scope**
  - **Nomenclature and classification**
  - **Level of granulites**
  - **Integration with other existing standards**
- **Results**
  - **Items to be requested in the RFP to be issued**
  - **Roadmap discussion (unchanged)**

## Proposed Items to be requested in the RFP (tentative)

- Aspects to be considered to classify devices
  - Data it deals with (input / output)
  - Characteristics
  - Action on environment
  - Other
- Device classification system
  - Ex. Device Hierarchy

## Proposed Items to be requested in the RFP (tentative)

- Types of devices
  - Primitive devices (correspond to physical devices)
  - Composite devices (combination of primitive devices)
  - Virtual (logical, abstract) devices
    - Do not necessarily have physical device correspondents
    - Fusion of primitive devices and/or composite devices
- Interfaces (or APIs)
  - Define [(abstract) interfaces] for various types of devices

# Proposed Items to be requested in the RFP (tentative)

- **Management and enumeration of devices**
  - **How to enumerate devices present in a robot**
  - **What are requirements for enumeration (what does a query look like)**
  - **How to manage those devices**

# Proposed Items to be requested in the RFP (tentative)

- **Device configuration**
  - **How to describe device properties**

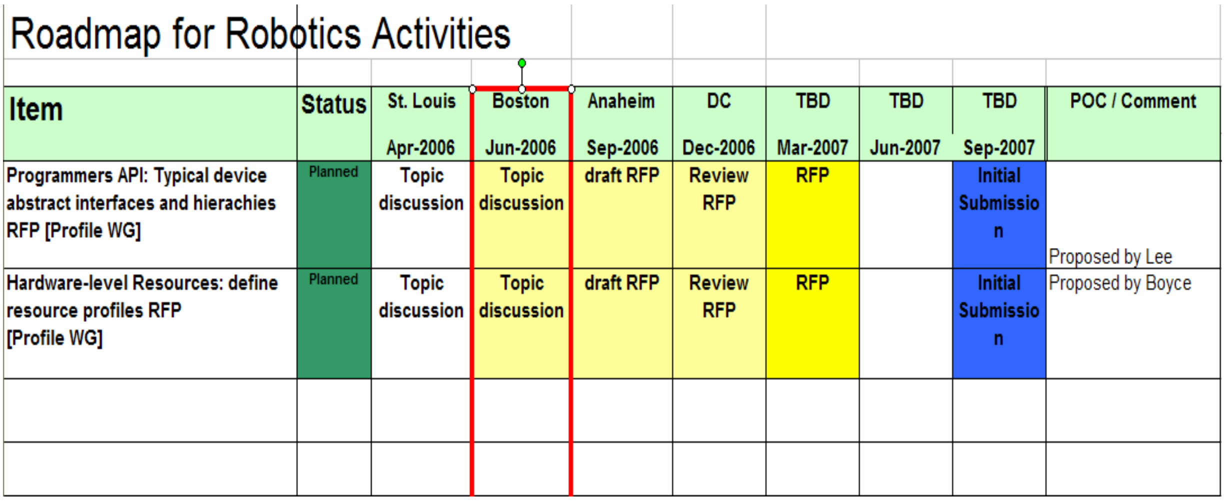
- **How to integrate existing standards**
  - **Relations with other existing standards**
  - **Adoption, revision, etc.**

# Proposed Items to be requested in the RFP (tentative)

- **Discussion**
  - **1. As a proof of concept, provide definition main devices fond in home/service robot (give definition)**
  - **The difference between devices (especially virtual devices) and functional services?**

# Roadmap

- **Unchanged**



Roadmap for Robotics Activities

| Item | Status | St. Louis | Boston | Anaheim | DC | TBD | TBD | TBD | POC / Comment |
|---|---|---|---|---|---|---|---|---|---|
| | | Apr-2006 | Jun-2006 | Sep-2006 | Dec-2006 | Mar-2007 | Jun-2007 | Sep-2007 | |
| Programmers API: Typical device abstract interfaces and hierachies RFP [Profile WG] | Planned | Topic discussion | Topic discussion | draft RFP | Review RFP | RFP | | Initial Submission | Proposed by Lee |
| Hardware-level Resources: define resource profiles RFP [Profile WG] | Planned | Topic discussion | Topic discussion | draft RFP | Review RFP | RFP | | Initial Submission | Proposed by Boyce |
| | | | | | | | | | |
| | | | | | | | | | |

# Near Future Plan

- Discussion on other existing standards
  - IEEE 1451 (at Anaheim)
  - JAUS (volunteers ?)
- Presentation on existing implementation
  - Example implementation of device classification, device interfaces (APIs), management, types of devices (primitive, composite, virtual), and any other related topics
  - Requesting Volunteers !!

# Working Group Mailing List

- Please use the WG mail list for all profile communication, by sending to:
- omg-profile@m.aist.go.jp
- First: to join, send a message from your email with the subject “subscribe {your name}” and be sure to always post to the list with that same email address.

robotics/2006-06-24

## - OMG Robotics DTF-

## - Robotic Services Working Group -

# Progress Report

## - Boston TC Meeting -

Boston (Massachusetts, USA) – June 28, 2006

**Co-chairs :** Olivier Lemaire (olivier.lemaire@aist.go.jp) / Soo-Yong Chi (chisy@etri.re.kr)

# Schedule

- Monday

  9:00 – 10:00 : Presentation

  *RUPI – Robot Unified Platform Initiative* (Soo-Young Chi)

  11:00 – 12:00 : WG Steering Committee

- Tuesday

  8:30 – 9:00 : Presentation

  *POEM, An implementation of position estimation module* (Takashi Tsubouchi)

  9:00 – 9:30 : Presentation

  *SAIT Proposal of Standards for Robot Localization based on MDA* (Yeon-Ho Kim)

  9:30 – 10:00 : Presentation

  *Issues on Localization Service* (Wonpil Yu)

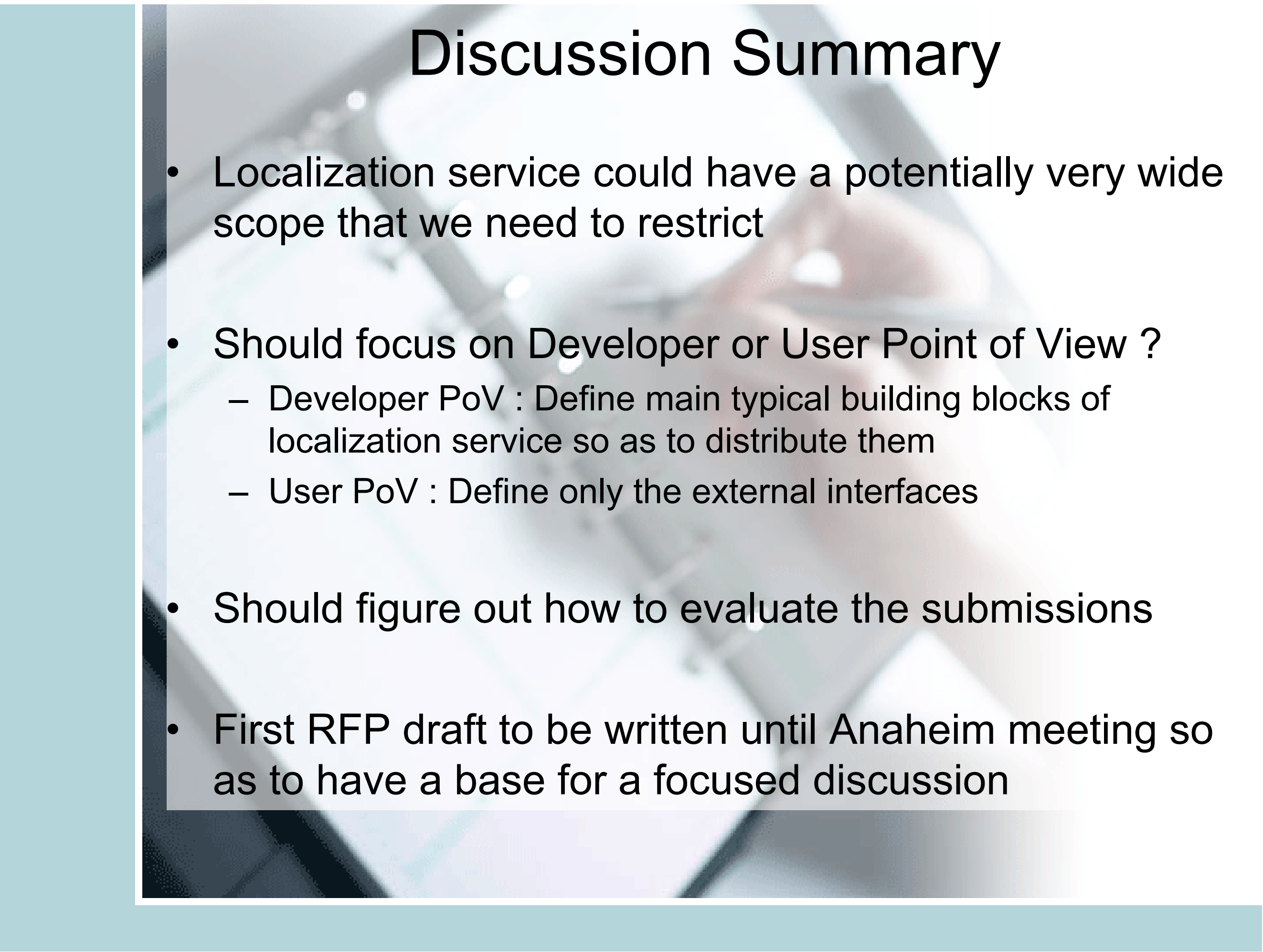
  10:00 – 12:00 : Discussion

# Steering Committee

- Roadmap Update
  - “User Identification” activity need to gather members
    - Set to Stand-by
  - “Localization Service” RFP issuance postponed to San Diego meeting
- Working Group Renaming
  - The terms “Service WG” and “Profile WG” are confusing. We proposed a renaming to :
    - “Robotic Capability WG”
    - “Robotic Functional Services WG”

# Roadmap

| Item | Status | St. Louis<br>Apr-2006 | Boston<br>Jun-2006 | Anaheim<br>Sep-2006 | Was. DC<br>Dec-2006 | San Diego<br>Mar-2007 | TBD<br>Jun-2007 |
|---|---|---|---|---|---|---|---|
| **Localization Service** | On-going | Topic Discussion | Topic Discussion | Draft RFP | Draft RFP | RFP | |
| **User Identification Service** | Stand-by | Proposed | -- | ? | ? | ? | ? |



# Discussion Summary

- Localization service could have a potentially very wide scope that we need to restrict
- Should focus on Developer or User Point of View ?
  - Developer PoV : Define main typical building blocks of localization service so as to distribute them
  - User PoV : Define only the external interfaces
- Should figure out how to evaluate the submissions
- First RFP draft to be written until Anaheim meeting so as to have a base for a focused discussion

robotics/2006-06-25

# Infrastructure Working Group

Progress Report

Boston, MA
June 2006

## Summary: Presentation

- RSCA and an approach to Deployment & Configuration of components
  - Dr. Seongsoo Hong, Seoul National Univ.

# Summary: D&C

- Discussion of RFP for D&C of RTC
  - Review of minutes and notes from St. Louis meeting
  - Decision to delay issuance by 1 meeting
    - Boston (June 2006): Discussion of RFP requirements
    - Anaheim (Sep. 2006): Review of internal draft document(s)
    - Washington D.C. (Dec. 2006): Issue draft RFP
    - San Diego (Mar. 2007): Review draft RFP
    - June 2007: Issue RFP

# Summary: Action Items

- Several organizations agreed to write up opinions/contents for draft RFP
  - Either in actual RFP form or just specific concerns
  - Volunteers:
    - ADA Software
    - AIST
    - RTI
    - Seoul National University

robotics/2006-06-26

# Introduction to RTC

Robotic Technology Component Specification

Robotics DTF

June 28 2006

Boston, MA, United States

Dr. Noriaki Ando
n-ando@aist.go.jp
National Institute of Advanced
Science & Technology (AIST)

Rick Warren
rick.warren@rti.com
Real-Time Innovations
(RTI)

## Timeline Overview

- September 2005: RFP issued
  - ptc/2005-09-01
- February 2006: Initial submissions
  - National Institute of Advanced Industrial Science and Technology (AIST)
    - mars/2006-01-05
    - Japan Robot Association (JARA) and Technologic Arts Incorporated join as supporters
  - Real-Time Innovations (RTI)
    - mars/2006-01-06
- June 2006: Revised submission
  - Joint submission by AIST and RTI
    - mars/2006-06-11
  - Seoul National University joins as supporter
- Adoption until +1 year: Finalization

# Timeline of this Meeting

- Presentation to MARS Monday
  - Some questions and discussion
    - Error/exception handling
    - Basic data types
    - PSM(s) mandatory?
    - XMI now available
  - Vote-to-vote
    - Passed
- Vote to adopt delayed until Thursday
  - 11:30 am
  - Voters: Please attend or give proxies
  - If passed, go to AB in afternoon

# What is RTC?

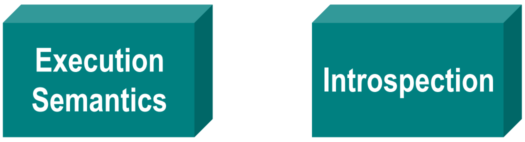
- Robotic Technology Component (RTC) Specification
- Component model for robotics
  - Basis for software modularization and integration at infrastructure/middleware level

# Benefits of RTC

- General benefits of component-orientation

+

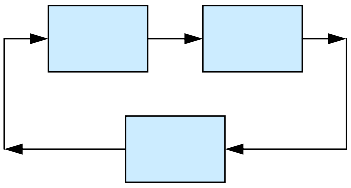
- Power of domain-specific extensions



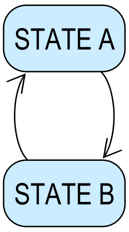
- Choice of platforms: CORBA/CCM or Local connectors

# Benefits of RTC

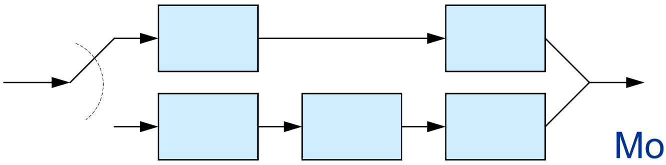
- Execution Semantics package standardizes common design patterns



Data flow / periodic, synchronous processing



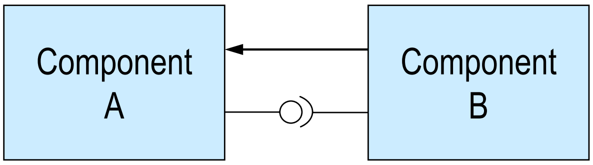
Stimulus-response / discrete-event processing



Modes of operation

## Benefits of RTC

- Introspection package provides the information you need about…
  - components, ports, and connections



  - which components are working together, and at what rate

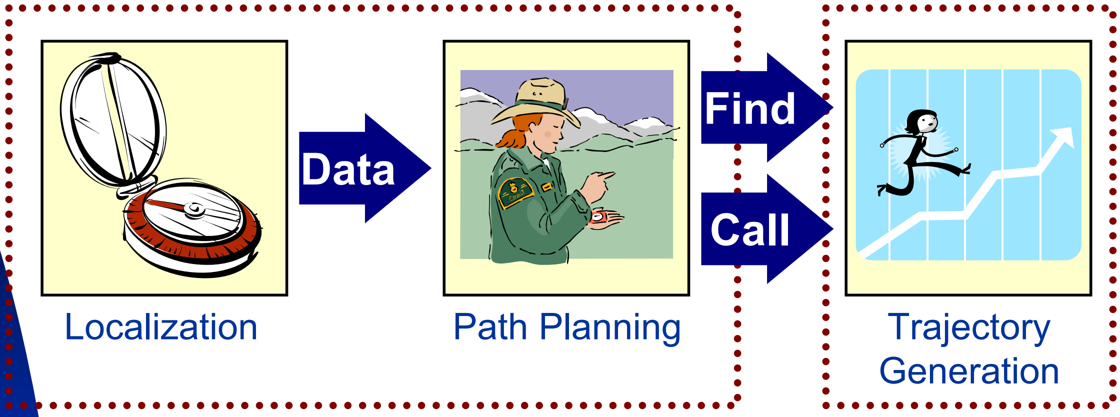
## Summary: Features of RTC

- Provides rich component lifecycle to enforce state coherency among components (LwRTC)
- Defines data structures for describing components and other elements (Introspection)
- Supports fundamental design patterns
  - Collaboration of fine-grained components tightly coupled in time (*e.g*. Simulink) (LwRTC, Local PSM)
  - Local or distributed components (PSMs)
  - Stimulus response with finite state machines (Exec. Sem.)
  - Dynamic composition of components collaborating synchronously or asynchronously (Exec. Sem., Introspection)

# Relation to Existing Standards

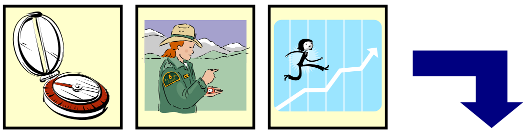
- UML
  - Domain-specific profile for UML components
- Super Distributed Objects (SDO)
  - Introspection of distributed components
  - Ports exposed as SDO services
- May optionally be combined or implemented with another model
  - *e.g.* Lightweight CORBA Component Model
  - *e.g.* Software Radio components

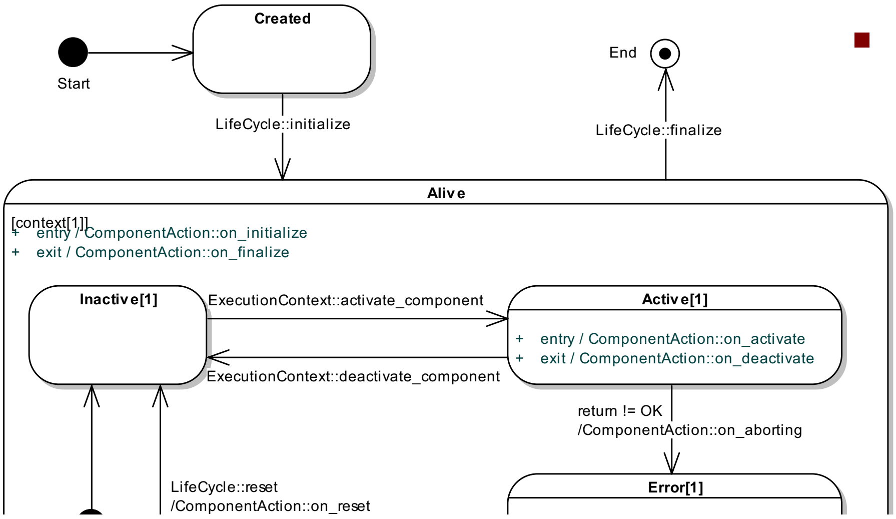
# Example: Path Planning



- Localization component streams coordinates to path planning component
- Path planning component chooses trajectory generator component dynamically
- Path planner invokes trajectory generator

# Component Lifecycle (2.2.2.3)




Created
Start
LifeCycle::initialize
End
LifeCycle::finalize
Alive
[context[1]]
+ entry / ComponentAction::on_initialize
+ exit / ComponentAction::on_finalize
Inactive[1]
ExecutionContext::activate_component
Active[1]
+ entry / ComponentAction::on_activate
+ exit / ComponentAction::on_deactivate
ExecutionContext::deactivate_component
return != OK
/ComponentAction::on_aborting
LifeCycle::reset
/ComponentAction::on_reset
Error[1]

- Every component has standard lifecycle
- “Alive” component can be in 1 of 3 states:
  - Active: switched on
  - Inactive: switch off
  - Error: error handling
- Each lifecycle transition has callback

# Life Cycle Callbacks (2.2.2.5)

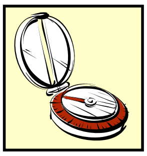
- Callback operations defined by *ComponentAction* interface
  - Realized by every RT component

| Callback | Called when component… |
|---|---|
| on_initialize | initialized (once) |
| on_finalize | destroyed (only once) |
| on_activate | switched on (any # times) |
| on_deactivate | switched off (any # times) |
| on_aborting | encounters error |
| on_error | is in error steady state |

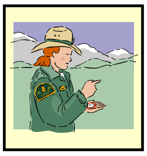
# Execution Context (2.2.2.6)

- Components that work together to accomplish same task *participate* in same execution context
  - Context corresponds to logical thread
- Behavioral pattern of participating components determined by context's *execution kind*
  - PERIODIC
    - Periodic ordered execution at a given rate
  - EVENT_DRIVEN
    - Asynchronous stimulus response
  - OTHER
    - Ad hoc collaboration or vendor extension

# Execution Kinds



- Localization component runs in PERIODIC context



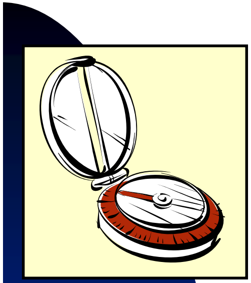
- Path planning component runs in EVENT_DRIVEN context

- Trajectory generator runs in OTHER context

* One component per context in this example

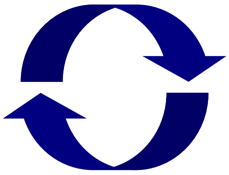
- In general, relationship is many-to-many

# Periodic Execution (2.3.1)



- Localization component needs to execute periodically
  - In every period, it outputs the current location
- Declare ability to participate in PERIODIC context with *dataFlowParticipant* stereotype
  - Realize additional callbacks from *DataFlowComponentAction* interface
    - on_execute invoked periodically
- Component doesn't have to manage timing, blocking, looping itself

# Order of Operations

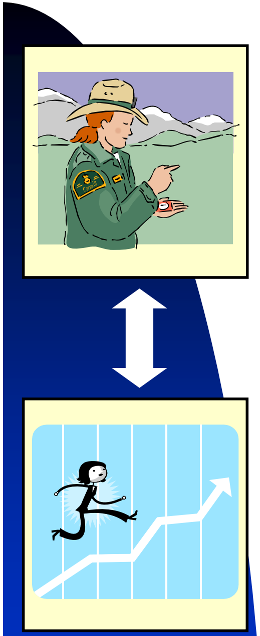


- on_initialize
  - One-time initialization
- on_activate
  - Prepare for execution
- on_execute
  - Get and output location
- on_deactivate
  - Cleanup from execution
- on_finalize
  - Final destruction

## Stimulus Response (2.3.2)

- Path planning component re-invokes trajectory generator whenever location reaches waypoint
- Stimulus response provided with UML finite state machines
- Declare ability to participate in EVENT_DRIVEN context with *fsmParticipant* stereotype
  - Realize additional callbacks from *FsmComponentAction* interface
    - on_transition invoked on state transition
- Component doesn’t have to manage event queuing and dispatch itself

## Dynamic Connection (2.4)



- Path planner dynamically discovers available trajectory generators at runtime
- Introspection API tells path planner about trajectory generators’ properties and configuration
- Dynamic port connect()/disconnect() allows path planner to decide when/if to connect to trajectory generator(s)

## Getting Involved

- Evaluate RTC for your application
- Post to the newsgroup
  - omg-infrastructure@m.aist.go.jp
- If it’s broken, help fix it
  - Give feedback to implementers
    - AIST and RTI are maintaining list of open issues
  - Participate in Finalization Task Force (FTF)

## Coming Up Next

- Vote to adopt
  - MARS meeting Thursday 11:30am
  - Proxies, please
- Finalization Task Force (FTF)
  - Starts post-adoption
  - Membership not limited to submitters
  - Process described at http://www.omg.org/gettingstarted/process4-Finalize.htm

Open Resource Interface for the Network /
Open Robot Interface for the Network

# Contact Report

ISO TC184/SC2 Plenary
15-16 June 2006, Saint Denis, France

ORiN Forum
http://www.orin.jp/

Chair: Makoto Mizukawa
Shibaura Institute of Technology

# ISO TC184：Scope

- □ Standardization in the field of industrial automation and integration concerning discrete part manufacturing and encompassing the application of multiple technologies, i.e. information systems, machines and equipment, and telecommunications.
- □ Excluded:
  - electrical and electronic equipment as dealt with by IEC / TC 44;
  - programmable logical controllers for general application dealt with by IEC / TC 65.

# TC184 Committee Title

- □ TC 184/AG Advisory group
- □ TC 184/SC 1
  - Physical device control
- □ TC 184/SC 2
  - Robots for industrial environments
- □ TC 184/SC 4
  - Industrial data
- □ TC 184/SC 5
  - Architecture, communications and integration frameworks

# ISO/TC 184/SC 2 ***Long term direction***

- □ Present title of ISO/TC 184/SC 2
  - Robots for industrial environments
- □ The recommendation from the Study group
  - ***Robots and robotic devices***
  - Widening of the SC 2 scope to include robot applications in any environment, not only industrial, is the most appropriate way to meet the standardization needs of new emerging robot environments.

# New Title of ISO/TC 184/SC 2

- ***Robots and robotic devices***
  - Standardization in the field of automatically controlled, reprogrammable, manipulating robots and robotic devices, programmable in more than one axis and either fixed in place or mobile.
  - Excluded: toys and military applications.

- Proposing to be accomplished at the next ISO/TC 184 plenary meeting.

# ISO/TC 184
# Action to widen the scope

- *Automation systems and integration in industrial environments and robotics*
  - Standardization in the field of industrial automation and integration concerning discrete part manufacturing and encompassing the application of multiple technologies, i.e. information systems, machines and equipment, and telecommunications.
  - **Standardization in the field of robotics, including fixed and mobile robots, in industrial and non-industrial environments.**
    - Excluded:
      - electrical and electronic equipment as dealt with by IEC / TC 44;
      - programmable logical controllers for general application dealt with by IEC / TC 65

## ISO 10218-1: *Robots for industrial environments - Safety requirements -*

- The publication
  - ISO 10218-1: *Part 1: Robot*
- The progress
  - ISO 10218-2 *Part 2: Robot system and integration*→A CD (Committee Draft).
  - The proposed development track : 36 months.

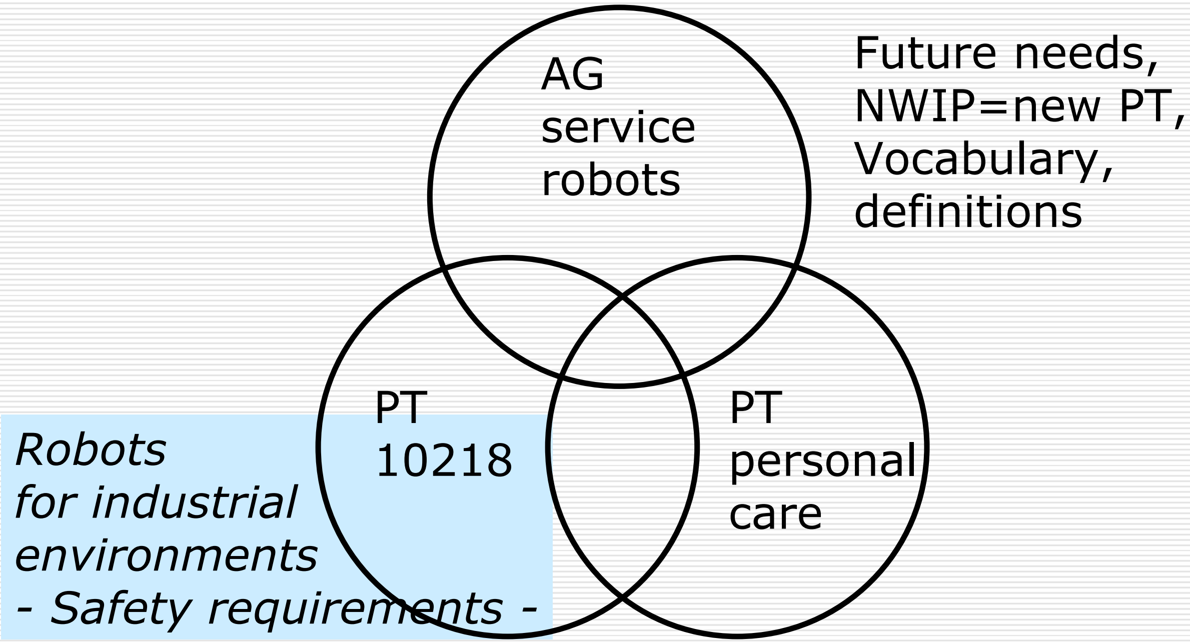
## Starting project team ***Robots in personal care***

- Develop a safety standard in the field of robots in personal care applications, including healthcare applications and excluding entertainment applications.
- Ensure consistency with ISO 10218-1 and the planned ISO 10218-2.
- Present a new work item proposal for SC 2 at appropriate time.
- Leader : Gurvinder Virk

# Starting Advisory group
# ***Service robots***

- □ To further explore needs for standardization in the field of service robots and to report its findings at SC 2 plenary meetings.
- □ Leader: Seungbin Moon.

# ISO TC 184/SC 2
# Advisory Group and Project Teams



AG service robots

Future needs, NWIP=new PT, Vocabulary, definitions

PT 10218

PT personal care

*Robots for industrial environments - Safety requirements -*

# ORiN and RAPI

- □ ORiN (Open Resource interface for the Network)
- □ RAPI (Robot communication framework and Application Program Interface)
  - to distribute a new work item proposal based on RAPI, including distribution to ISO/TC 184 and ISO/TC 184/SC 5 for possible comments.
  - In order to better inform the other subcommittees, SC 2 recommends ORiN to make the same presentation on RAPI at the next ISO/TC 184 plenary meeting in Madrid 9-10 October 2006.

# Relationship between ORiN and RAPI

ORiN: Open Resource interface for the Network

- ➢ Based on the advanced PC software technologies.
  - ✓ [CAO] Distributed Object Technology for API: DCOM, CORBA.
  - ✓ [CRD] Data Schema: XML-Schema.
  - ✓ [CAP] Communication Protocol: SOAP, etc.
- ➢ ORiN targets on various FA devices.

RAPI: Robot communication framework and Application Program Interface

- ➢ An abstraction of ORiN.
- ➢ A subset of ORiN in functionality.
  - ✓ Limit functions to the Robots (excluding other FA devices)
  - ✓ API and Data Schema (excluding Communication Protocol*)

# Schedule (tentative)

| 2006 | | 2007 | | | | 2008 | | | | 2009 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3rd Q | 4th Q | 1st Q | 2nd Q | 3rd Q | 4th Q | 1st Q | 2nd Q | 3rd Q | 4th Q | 1st Q | 2nd Q |
| Creating the 1st draft | | | | | Updating | | | | | Finalizing | |

◆ 6/15 ISO meeting in Paris → ◇ RAPI 1st revision → Commit the NWIP ◇

◇ RAPI draft by the ORiN forum

◇ Start the discussion by the PT

*Objective*: To create a standard framework for Robot communications.
*Term of Activity*: June 2006 to June 2009, 3 Years

*Project Leader*: Dr. Makoto MIZUKAWA, Shibaura Institute of Technology, Chair of ORiN Forum.
*Project Sub-leader*: Satoshi SAKAKIBARA, DENSO WAVE INC.
*Project Secretariat*: Japan Robot Association

# Future plan, etc

- **Next meeting**
  - Washington D.C.
  - 14-15 June 2007
- **New liaisons**

  ISO/TC 184/SC 2 requires the secretariat to propose and explore establishing liaisons with

  - IEEE "Robotics and Automation society", liaison observer to be confirmed
  - OMG, liaison observer to be confirmed

# ORiN

*Open Resource Interface for the Network /*
*Open Robot Interface for the Network*

Japan Robot Association

ORiN forum

http://www.orin.jp/

robotics/2006-06-28

OMG Technical Meeting in Boston Robotics-DTF
Jun 27, 2006

# Introduction of JAUS for the benefit of Robotic Standardization

Wataru Inamura

Explore the Engineering Edge

IHI

Ishikawajima-Harima Heavy Industries Co., Ltd.

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

## 0. Agenda

Explore the Engineering Edge

- Introduction
- JAUS Outline
- JAUS Components
- JAUS Messages
- Localization Components & Messages
- Conclusion

# 1. Introduction

I made presentation "Development Framework for Mobile Robot based on JAUS and RT-Middleware" as response to RFI at Tampa meeting.

It showed how to use JAUS with RTC.

So now, I would like to introduce JAUS Components as Devices or Services.

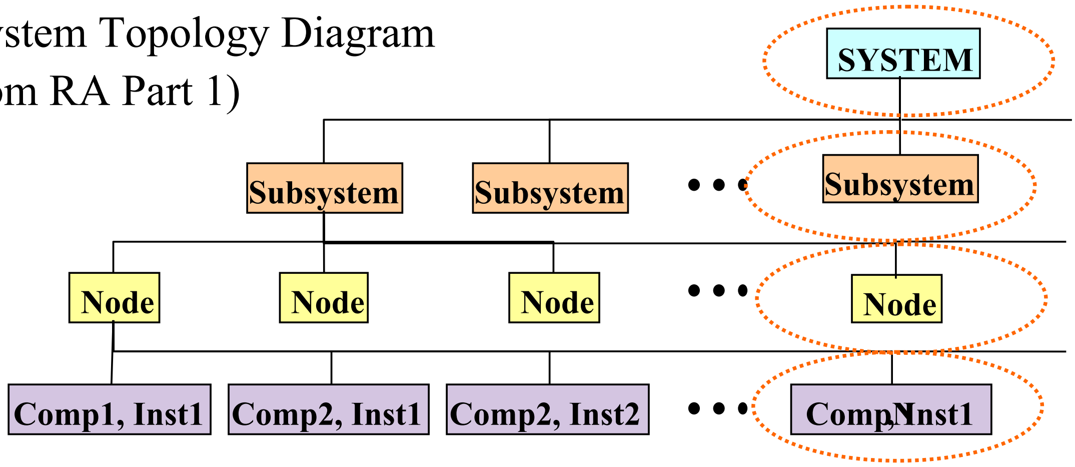
# 2.1. JAUS Outline

## JAUS（Joint Architecture for Unmanned Systems）

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

## 2.2. JAUS System Topology

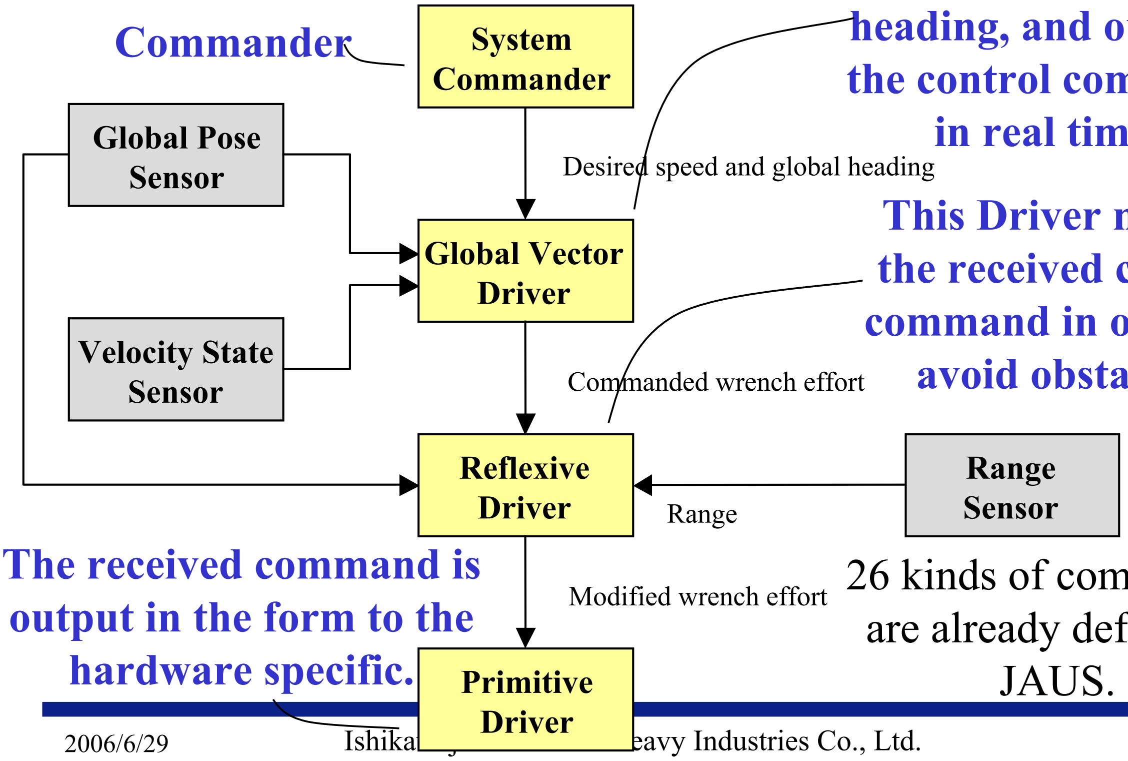
System Topology Diagram
(From RA Part 1)



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

## 2.3. JAUS Components Combination

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



**This Driver receives the desired speed and heading, and outputs the control command in real time.**

**This Driver modify the received control command in order to avoid obstacles.**

**The received command is output in the form to the hardware specific.**

26 kinds of components are already defined in JAUS.

## 2.4. JAUS Messages Outline

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

■ The messages for unmanned systems are already defined.

■ The messages are classified into six categories. (Command、Query、Inform、Event Setup、Event Notification、Node Management)

■ Message format ＝ header(16bytes) + option data

■ The option data is specific to each kind of message.

## 3. JAUS Components

5 Categories

- COMMAND AND CONTROL COMPONENTS
- COMMUNICATIONS COMPONENTS
- PLATFORM COMPONENTS
- MANIPULATOR COMPONENTS
- ENVIRONMENT SENSOR COMPONENTS

26 Components

Useful components are already defined.

Include Localization or other devices.

# 4. JAUS Messages

6 Categories

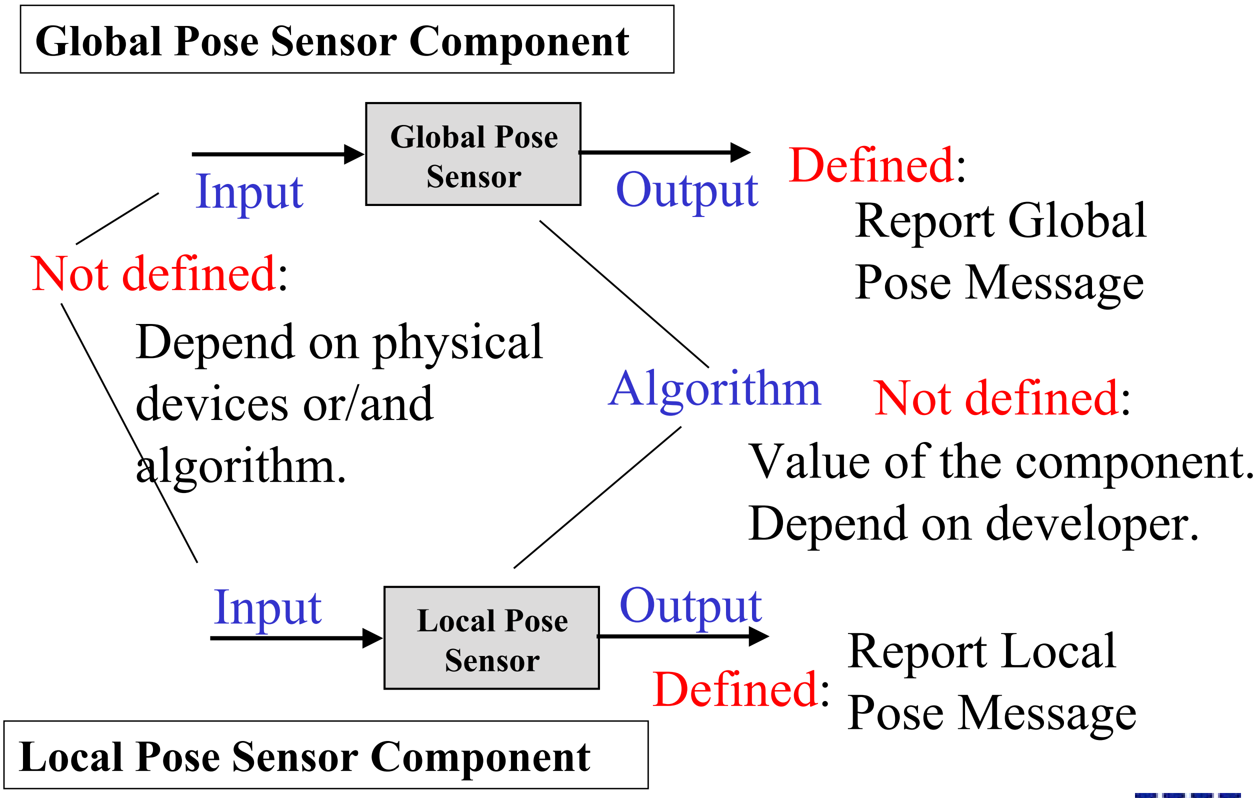
- COMMAND CLASS
- QUERY CLASS
- INFORM CLASS
- EVENT SETUP CLASS
- EVENT NOTIFICATION CLASS
- NODE MANAGEMENT CLASS

110 Messages

Useful messages are already defined.

Include Localization or other devices.

# 5.1 Components for Localization



**Global Pose Sensor Component**

**Global Pose Sensor**

Input

Output

Defined:
Report Global
Pose Message

Not defined:
Depend on physical
devices or/and
algorithm.

Algorithm

Not defined:
Value of the component.
Depend on developer.

Input

**Local Pose Sensor**

Output

Defined:
Report Local
Pose Message

**Local Pose Sensor Component**

# 5.2 Message for Localization – Global

## Report Global Pose Message

| Field | Name | Type | Units | Interpretation |
|---|---|---|---|---|
| 1 | Presence Vector | Unsigned Short | N/A | See mapping table that follows |
| 2 | Latitude (WGS 84) | Integer | Degrees | Scaled Integer<br>Lower Limit = -90<br>Upper Limit = 90 |
| 3 | Longitude (WGS 84) | Integer | Degrees | Scaled Integer<br>Lower Limit = -180<br>Upper Limit = 180 |
| 4 | Elevation | Integer | Meters | Scaled Integer<br>Lower Limit = -10,000<br>Upper Limit = 35,000 |
| 5 | Position RMS | Unsigned Integer | Meters | A RMS value indicating the validity of the position data.<br>Scaled integer<br>Lower Limit = 0<br>Upper Limit = 100 |

# 5.3 Message for Localization - Global

## Report Global Pose Message

| Field | Name | Type | Units | Interpretation |
|---|---|---|---|---|
| 6 | $\phi$ (Roll) | Short Integer | Radians | Scaled Integer<br>Lower Limit = - $\pi$<br>Upper Limit = $\pi$ |
| 7 | $\theta$ (Pitch) | | | |
| 8 | $\psi$ (Yaw) | | | |
| 9 | Attitude RMS | Unsigned Short Integer | Radians | A RMS value indicating the validity of the Roll/Pitch/Yaw data<br>Scaled integer<br>Lower Limit = 0<br>Upper Limit = $\pi$ |
| 10 | Time Stamp | Unsigned Integer | | Bits 0-9: milliseconds, range 0...999<br>Bits 10-15: Seconds, range 0...59<br>Bits 16 – 21: Minutes, range 0...59<br>Bits 22-26: Hour (24 hour clock), range 0..23<br>Bits 27-31: Day, range 1...31 |

## 5.4 Message for Localization - Local

### Report Local Pose Message

| Field | Name | Type | Units | Interpretation |
|---|---|---|---|---|
| 1 | Presence Vector | Unsigned Short | N/A | See mapping table that follows |
| 2 | X | Integer | Meters | Scaled Integer<br>Lower Limit = -100,000<br>Upper Limit = 100,000 |
| 3 | Y | Integer | Meters | Scaled Integer<br>Lower Limit = -100,000<br>Upper Limit = 100,000 |
| 4 | Z | Integer | Meters | Scaled Integer<br>Lower Limit = -10,000<br>Upper Limit = 35,000 |
| 5 | Position RMS | Unsigned Integer |  | A RMS value indicating the validity of the position data.<br>Scaled Integer<br>Lower Limit = 0<br>Upper Limit = 100 |

## 5.5 Message for Localization - Local

### Report Local Pose Message

| Field | Name | Type | Units | Interpretation |
|---|---|---|---|---|
| 6 | $\phi$ (Roll) | Short Integer | Radians | Scaled Integer<br>Lower Limit = - $\pi$<br>Upper Limit = $\pi$ |
| 7 | $\theta$ (Pitch) |  |  |  |
| 8 | $\psi$ (Yaw) |  |  |  |
| 9 | Attitude RMS | Unsigned Short Integer | Radians | A RMS value indicating the validity of the Roll/Pitch/Yaw data<br>Scaled integer<br>Lower Limit = 0<br>Upper Limit = $\pi$ |
| 10 | Time Stamp | Unsigned Integer |  | Bits 0-9: milliseconds, range 0...999<br>Bits 10-15: Seconds, range 0...59<br>Bits 16 – 21: Minutes, range 0...59<br>Bits 22-26: Hour (24 hour clock), range 0..23<br>Bits 27-31: Day, range 1...31 |

# 6. Conclusion

- There are many useful component and message ( = data structure) definitions .

- We can use JAUS Specification as an starting point for Robotics-DTF Standards.

- I would like to send more information and my opinion to mailing list of Robotic Device and Data Profile WG & Robotic Functional Service WG.

Explore the Engineering Edge

Thank you for your attention.

## 3.1. COMMAND AND CONTROL COMPONENTS

All Components are shown.

### COMMAND AND CONTROL COMPONENTS

- System Commander (ID 40)
- Subsystem Commander (ID 32)

## 3.2. COMMUNICATIONS COMPONENTS

ng Edge

### COMMUNICATIONS COMPONENTS

- Communicator (ID 35)

## PLATFORM COMPONENTS

- **Global Pose Sensor (ID 38)**
- **Local Pose Sensor (ID 41)**

} Localization Service Components

- Velocity State Sensor (ID 42)
- Primitive Driver (ID 33)
- Reflexive Driver (ID 43)
- Global Vector Driver (ID 34)
- Local Vector Driver (ID 44)
- Global Waypoint Driver (ID 45)
- Local Waypoint Driver (ID 46)
- Global Path Segment Driver (ID 47)
- Local Path Segment Driver (ID 48)

## MANIPULATOR COMPONENTS

- Primitive Manipulator (ID 49)
- Manipulator Joint Position Sensor Component (ID 51)
- Manipulator Joint Velocity Sensor Component (ID 52)
- Manipulator Joint Force/Torque Sensor Component (ID 53)
- Manipulator Joint Positions Driver Component (ID 54)
- Manipulator End-Effector Pose Driver Component (ID 55)
- Manipulator Joint Velocities Driver Component (ID 56)
- Manipulator End-Effector Velocity State Driver Component (ID 57)
- Manipulator Joint Move Driver Component (ID 58)
- Manipulator End-Effector Discrete Pose Driver Component (ID 59)

# 3.5 ENVIRONMENT SENSOR COMPONENTS

## ENVIRONMENT SENSOR COMPONENTS

- Visual Sensor (ID 37)
- Range Sensor (ID 50)

# 4.1.1. COMMAND CLASS Messages

All Messages are shown.

## COMMAND CLASS

### Core Subgroup - Codes 0001-01FF

- Code 0001h: Set Component Authority
- Code 0002h: Shutdown
- Code 0003h: Standby
- Code 0004h: Resume
- Code 0005h: Reset
- Code 0006h: Set Emergency
- Code 0007h: Clear Emergency

# 4.1.2. COMMAND CLASS Messages

## COMMAND CLASS

## Core Subgroup - Codes 0001-01FF

- Code 0008h: Create Service Connection
- Code 0009h: Confirm Service Connection
- Code 000Ah: Activate Service Connection
- Code 000Bh: Suspend Service Connection
- Code 000Ch: Terminate Service Connection
- Code 000Dh: Request Component Control
- Code 000Eh: Release Component Control
- Code 000Fh: Confirm Component Control
- Code 0010h: Reject Component Control
- Code 0011h: Set Time

# 4.1.2. COMMAND CLASS Messages

## COMMAND CLASS

## Communications Subgroup - Codes 0200-03FF

- Code 0200h: Set Data Link Status
- Code 0201h: Set Data Link Select

## 4.1.3. COMMAND CLASS Messages

COMMAND CLASS

Platform Subgroup - Codes 0400-05FF

- Code 0405h: Set Wrench Effort
- Code 0406h: Set Discrete Devices
- Code 0407h: Set Global Vector
- Code 0408h: Set Local Vector (DEPRECATE at Version 4.0)
- Code 040Ah: Set Travel Speed
- Code 040Ch: Set Global Waypoint
- Code 040Dh: Set Local Waypoint (DEPRECATE at Version 4.0)
- Code 040Fh: Set Global Path Segment
- Code 0410h: Set Local Path Segment (DEPRECATE at Version 4.0)

## 4.1.4. COMMAND CLASS Messages

COMMAND CLASS

Manipulator Subgroup - Codes 0600-07FF

- Code 0601h: Set Joint Efforts
- Code 0602h: Set Joint Positions
- Code 0603h: Set Joint Velocities
- Code 0604h: Set Tool Point
- Code 0605h: Set End-Effector Pose
- Code 0606h: Set End-Effector Velocity State
- Code 0607h: Set Joint Motion
- Code 0608h: Set End-Effector Path Motion

# 4.1.5. COMMAND CLASS Messages

## COMMAND CLASS

### Environment Sensor Subgroup - Codes 0800-09FF

- Code 0801h: Set Camera Pose
- Code 0802h: Select Camera
- Code 0805h: Set Camera Capabilities
- Code 0806h: Set Camera Format Options

# 4.2.1. QUERY CLASS Messages

## QUERY CLASS

### Core Subgroup - Codes 2000-21FF

- Code 2001h: Query Component Authority
- Code 2002h: Query Component Status
- Code 2011h: Query Time

### Communications Subgroup - Codes 2200-23FF

- Code 2200h: Query Data Link Status
- Code 2201h: Query Selected Data Link Status
- Code 2202h: Query Heartbeat Pulse

# 4.2.2. QUERY CLASS Messages

## QUERY CLASS

## Platform Subgroup - Codes 2400-25FF

- Code 2400h: Query Platform Specifications
- Code 2401h: Query Platform Operational Data
- Code 2402h: Query Global Pose
- Code 2403h: Query Local Pose (DEPRECATE at Version 4.0)
- Code 2404h: Query Velocity State
- Code 2405h: Query Wrench Effort
- Code 2406h: Query Discrete Devices
- Code 2407h: Query Global Vector
- Code 2408h: Query Local Vector (DEPRECATE at Version 4.0)

# 4.2.3. QUERY CLASS Messages

## QUERY CLASS

## Platform Subgroup - Codes 2400-25FF

- Code 240Ah: Query Travel Speed
- Code 240Bh: Query Waypoint Count
- Code 240Ch: Query Global Waypoint
- Code 240Dh: Query Local Waypoint (DEPRECATE at Version 4.0)
- Code 240Eh: Query Path Segment Count
- Code 240Fh: Query Global Path Segment
- Code 2410h: Query Local Path Segment (DEPRECATE at Version 4.0)

# 4.2.4. QUERY CLASS Messages

## QUERY CLASS

### Manipulator Subgroup - Codes 2600-27FF

- Code 2600h: Query Manipulator Specifications
- Code 2601h: Query Joint Efforts
- Code 2602h:  Query Joint Positions
- Code 2603h: Query Joint Velocities
- Code 2604h: Query Tool Point
- Code 2605h: Query Joint Force/Torques

# 4.2.5. QUERY CLASS Messages

## QUERY CLASS

### Environment Sensor Subgroup - Codes 2800-29FF

- Code 2800h: Query Camera Pose
- Code 2801h: Query Camera Count
- Code 2802h: Query Relative Object Position
- Code 2804h: Query Selected Camera
- Code 2805h: Query Camera Capabilities
- Code 2806h: Query Camera Format Options
- Code 2807h: Query Image

# 4.3.1. INFORM CLASS Messages

INFORM CLASS

Core Subgroup - Codes 4001-41FF

- Code 4001h: Report Component Authority
- Code 4002h: Report Component Status
- Code 4011h: Report Time

Communications Subgroup - Codes 4200-43FF

- Code 4200h: Report Data Link Status
- Code 4201h: Report Selected Data Link Status
- Code 4202h: Report Heartbeat Pulse

# 4.3.2. INFORM CLASS Messages

INFORM CLASS

Platform Subgroup - Codes 4400-45FF

- Code 4400h: Report Platform Specifications
- Code 4401h: Report Platform Operational Data
- Code 4402h: Report Global Pose
- Code 4403h: Report Local Pose (DEPRECATE at Version 4.0)
- Code 4404h: Report Velocity State
- Code 4405h: Report Wrench Effort
- Code 4406h: Report Discrete Devices
- Code 4407h: Report Global Vector
- Code 4408h: Report Local Vector (DEPRECATE at Version 4.0)

Localization Service Messages (I/O)

## 4.3.3. INFORM CLASS Messages

### INFORM CLASS

### Platform Subgroup - Codes 4400-45FF

- Code 440Ah: Report Travel Speed
- Code 440Bh: Report Waypoint Count
- Code 440Ch: Report Global Waypoint
- Code 440Dh: Report Local Waypoint (DEPRECATE at Version 4.0)
- Code 440Eh: Report Path Segment Count
- Code 440Fh: Report Global Path Segment
- Code 4410h: Report Local Path Segment (DEPRECATE at Version 4.0)

## 4.3.4. INFORM CLASS Messages

### INFORM CLASS

### Manipulator Subgroup - Codes 4600-47FF

- Code 4600h: Report Manipulator Specifications
- Code 4601h: Report Joint Efforts
- Code 4602h: Report Joint Positions
- Code 4603h: Report Joint Velocities
- Code 4604h: Report Tool Point
- Code 4605h: Report Joint Force/Torques

## 4.3.5. INFORM CLASS Messages

INFORM CLASS

Environment Sensor Subgroup - Codes 4800-49FF

- Code 4800h: Report Camera Pose
- Code 4801h: Report Camera Count
- Code 4802h: Report Relative Object Position
- Code 4804h: Report Selected Camera
- Code 4805h: Report Camera Capabilities
- Code 4806h: Report Camera Format Options
- Code 4807h: Report Image

## 4.4. Other Class Messages

EVENT SETUP CLASS

EVENT NOTIFICATION CLASS

NODE MANAGEMENT CLASS

Currently no those Class messages are defined.

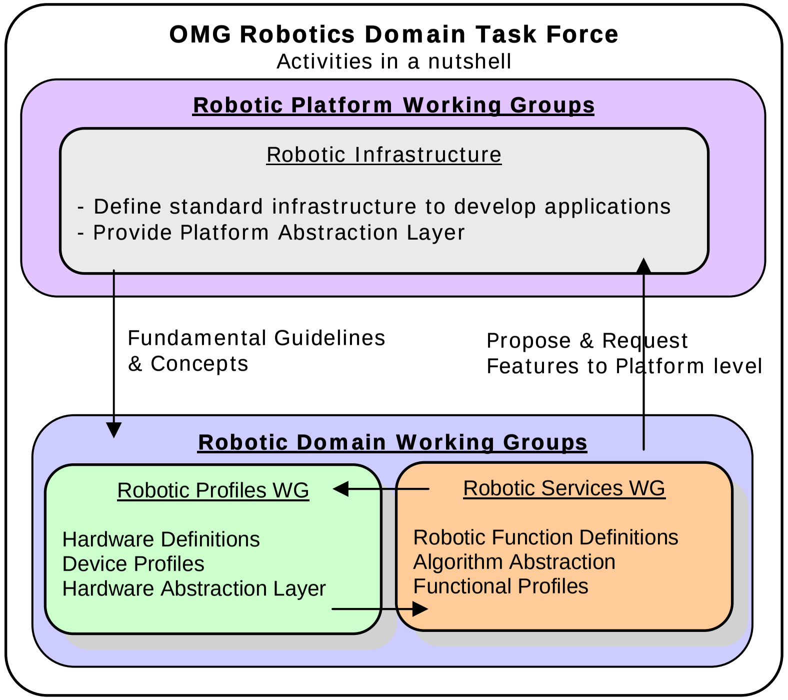
# KIRSF – Contact Report

Robotics DTF (Boston Meeting)
Date: June 28th, 2006
Reporter: Yun Koo Chung

- RUPI (Robot Unified Platform Initiative) standardization planning will be launched on July 4th :
  - Specifications for testing and performance evaluation for commercial home service robots
  - Communication protocols for URC robots.
  - Supports OMG and ISO standardization activities

- Korea-Japan Robot Standardization Workshop 2006 was held.
  - June 16th (Friday), 2006
  - Jeju, Korea – 6 speakers presented

**robotics/2006-06-29**



## OMG Robotics Domain Task Force

Activities in a nutshell

**Robotic Platform Working Groups**

Robotic Infrastructure

- Define standard infrastructure to develop applications
- Provide Platform Abstraction Layer

Fundamental Guidelines & Concepts

Propose & Request Features to Platform level

**Robotic Domain Working Groups**

Robotic Profiles WG

Hardware Definitions
Device Profiles
Hardware Abstraction Layer

Robotic Services WG

Robotic Function Definitions
Algorithm Abstraction
Functional Profiles

More info abt WGs??
News and Events
Timeline and Roadmap (for each WG??)

**Feedbacks from Yokomachi-san**

- Motivation and Goals
  - Why we need standards
- Proposed Methods
- Targets of the DTF
- Structure of the Working Groups
  - DTF and its Sub TFs
- Mission Statement (must make it short)
  - From the Robotics POV
- Description of the WGs
  - Profile
  - Infrastructure
  - Services
  - Tools ??
- News and Events attended
- Related Organisations
  - ISO, IEEE, OASIS, etc - how to mention them (subtle but clear)
- Weblinks , URL
- Invitation to contribute to the standardisation drive
  - What can we do!
- Opinion poll report summary (from Tampa)

# Robotics Domain Task Force

**Perhaps some pictures of a few robots**

**(If it can indicate standards, it would be nice!)**

**A SLOGAN!! PUNCHLINE!**

- **TBD – Homework!**

**Join Us!**

**Membership Information**

**Websites**

**Etc etc**

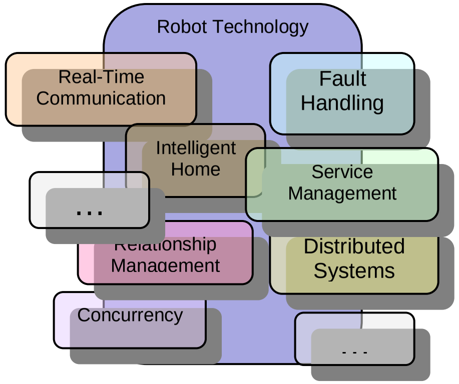
**Robotics DTF Contacts**

**Workgroup Contacts**

## About Object Management Group & the Robotics DTF

The Object Management Group (OMG), home to UML, CORBA and other such technologies, is a not for profit consortium targeting the production and maintenance of the computer industry specifications for interoperable enterprise applications. The Robotics Domain Task force aims to establish such standards for the Robotics Industry.

## Focus and targets



Robot Technology is a complicated mix of various domain technologies and as a result very complicated to form a common standard. To make things more challenging, key players in the industry today develop their own unique standards which act against inter–operability

The Robotics DTF in OMG addresses these issues by:

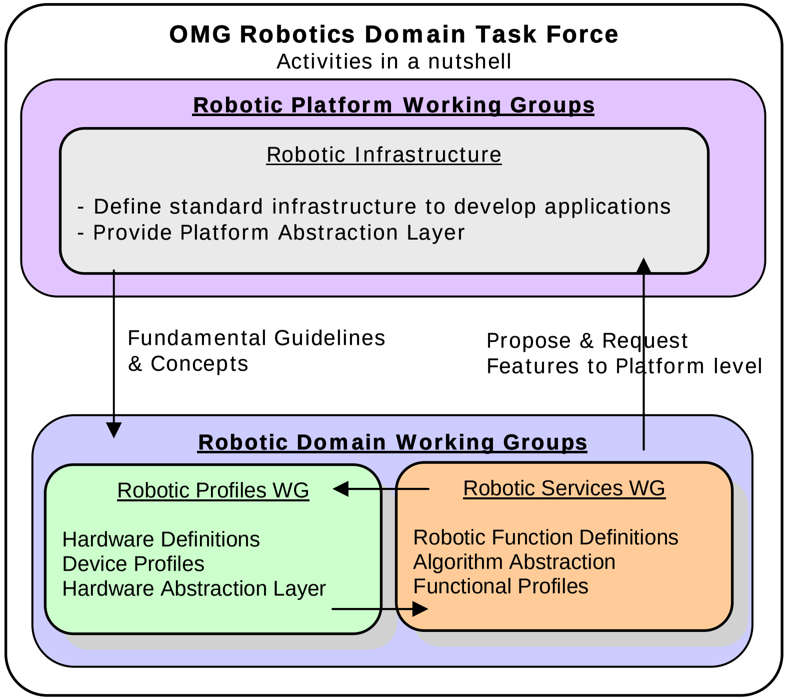
- adoption of existing OMG standards to the domain
- extending OMG technologies to robotic applications
- form a bridge between OMG and Robotic communities
- collaborate with other similar organisations like ISO, IEEE, OASIS to encourage interoperability
- coordinate with other task forces in OMG to develop common standards

## Established Working Groups & their focus

A survey was conducted amongst the DTF members to determine the important aspects within Robotics to be standardised. Resulting from the survey, three workgroups were formed:

- Infrastructure
- Robotic Functional Services
- Robotic Devices and Data Profile

Working Group Diagram Pending



OMG Robotics Domain Task Force
Activities in a nutshell

**Robotic Platform Working Groups**

Robotic Infrastructure

- Define standard infrastructure to develop applications
- Provide Platform Abstraction Layer

Fundamental Guidelines & Concepts

Propose & Request Features to Platform level

**Robotic Domain Working Groups**

Robotic Profiles WG

Hardware Definitions
Device Profiles
Hardware Abstraction Layer

Robotic Services WG

Robotic Function Definitions
Algorithm Abstraction
Functional Profiles

More info abt WGs??
News and Events
Timeline and Roadmap (for each WG??)

**Feedbacks from Yokomachi-san**

- Motivation and Goals
    - o Why we need standards
- Proposed Methods
- Targets of the DTF
- Structure of the Working Groups
    - o DTF and its Sub TFs
- Mission Statement (must make it short)
    - o From the Robotics POV
- Description of the WGs
    - o Profile
    - o Infrastructure
    - o Services
    - o Tools ??
- News and Events attended
- Related Organisations
    - o ISO, IEEE, OASIS, etc - how to mention them (subtle but clear)
- Weblinks , URL
- Invitation to contribute to the standardisation drive
    - o What can we do!
- Opinion poll report summary (from Tampa)

# Robotics Domain Task Force

**Perhaps some pictures of a few robots**

**(If it can indicate standards, it would be nice!)**

**A SLOGAN!! PUNCHLINE!**

- **TBD – Homework!**

b

**Join Us!**

**Membership Information**

**Websites**

**Etc etc**

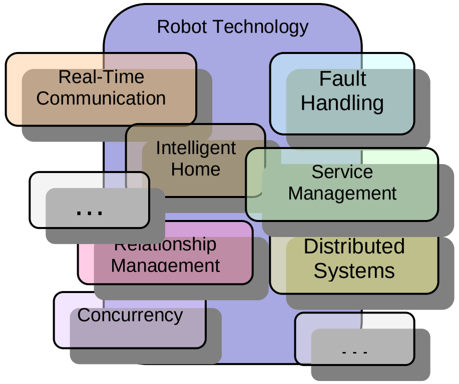
**Robotics DTF Contacts**

**Workgroup Contacts**

## About Object Management Group & the Robotics DTF

The Object Management Group (OMG), home to UML, CORBA and other such technologies, is a not for profit consortium targeting the production and maintenance of the computer industry specifications for interoperable enterprise applications. The Robotics Domain Task force aims to establish such standards for the Robotics Industry.

## Focus and targets



Robot Technology is a complicated mix of various domain technologies and as a result very complicated to form a common standard. To make things more challenging, key players in the industry today develop their own unique standards which act against inter–operability

The Robotics DTF in OMG addresses these issues by:

- adoption of existing OMG standards to the domain
- extending OMG technologies to robotic applications
- form a bridge between OMG and Robotic communities
- collaborate with other similar organisations like ISO, IEEE, OASIS to encourage interoperability
- coordinate with other task forces in OMG to develop common standards

## Established Working Groups & their focus

A survey was conducted amongst the DTF members to determine the important aspects within Robotics to be standardised. Resulting from the survey, three workgroups were formed:

- Infrastructure
- Robotic Functional Services
- Robotic Devices and Data Profile

Working Group Diagram Pending

# Anaheim Agenda Coordination

Sun: No business

Mon: (WG activity)

- AM: Infrastructure WG(2h) drafting RFP
- AM: Service WG(2h) discussion
- PM1: Profile WG(2h) presentation(IEEE1451) + discussion
- PM2: Steering Committee

Tue: (WG activity)

- AM1: Service WG (2h) presentation (User Identification) + discussion
- PM1: Profile WG (1.5h) discussion
- PM2: Infrastructure WG (2h) discussion

Wed: (Plenary)

- SP1: John Hogg “Introduction to Zeligsoft Component Enabler 2.4?”
- SP2: Bruce Boyes “Microsoft Robotics Studio?”
- SP3: SysML (ManTIS is postpone to the Washington DC (Tue. or Thu.))

Thu: (WG activity)

- AM:
- PM1:
- PM2:

# Ontology Definition Metamodel (ODM)

## Background & Overview

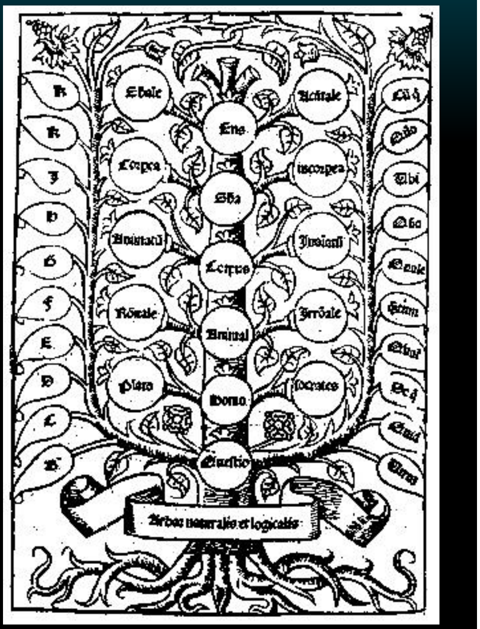
Elisa Kendall

Sandpiper Software

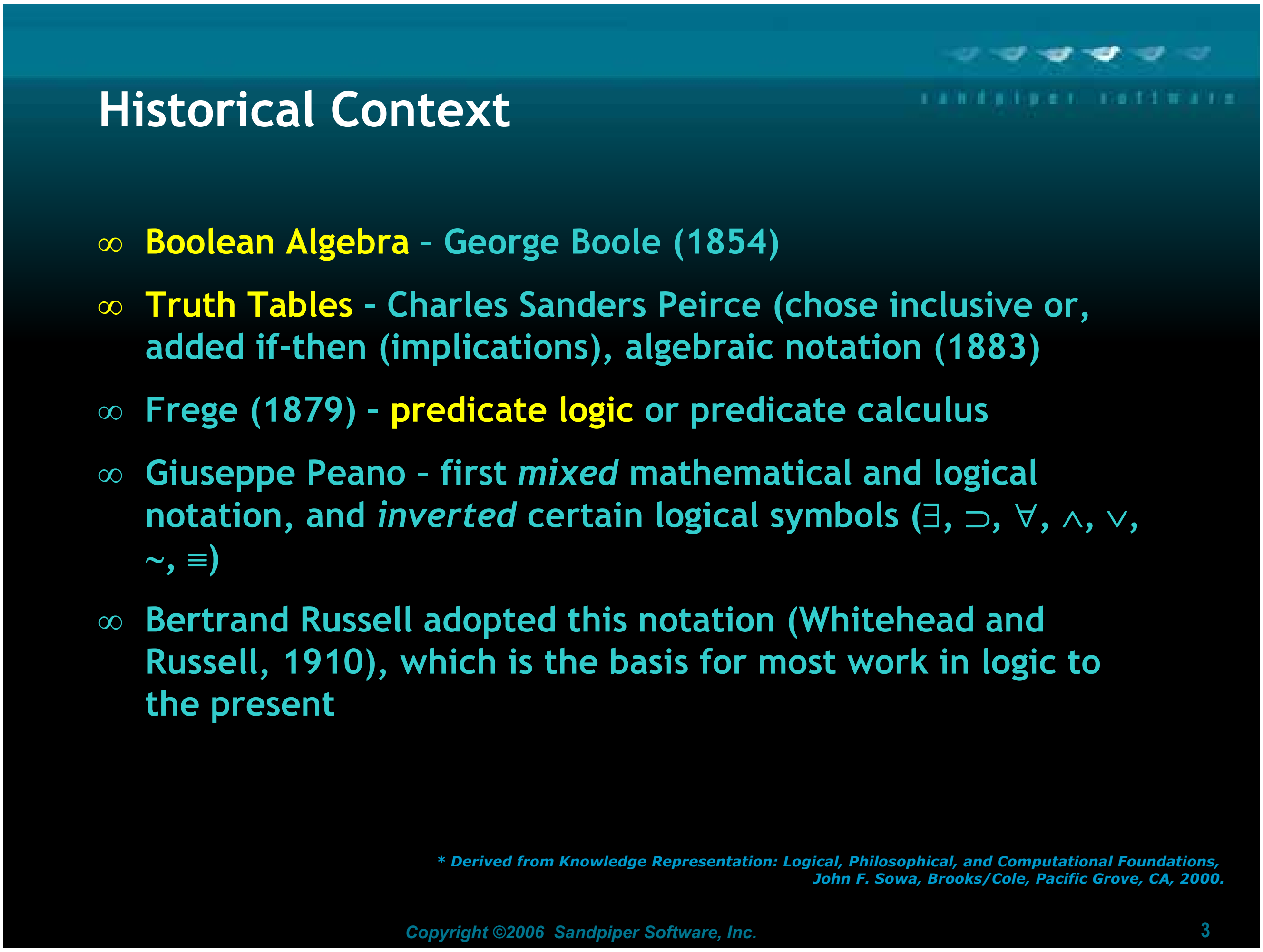
June 29, 2006

## Historical Context

- Semantic Networks
  - Developed from research in artificial intelligence using graph notation
  - First used by Porphyry in the 3rd century to describe Aristotle's hierarchy of species (including inheritance and differentiae)
- Automated Reasoning
  - First invented by 13th century (Lull)
  - Tree of nature and logic - disks with rotating circles, the largest had 14 concentric circles with 16 sectors each, $16^{14}$ possibilities
- Mathematical Logic
  - Leibniz' encodings, 17th century
  - Definition of the notions of T-Box and A-Box – for modern AI systems, Ron Brachman et al, 1983



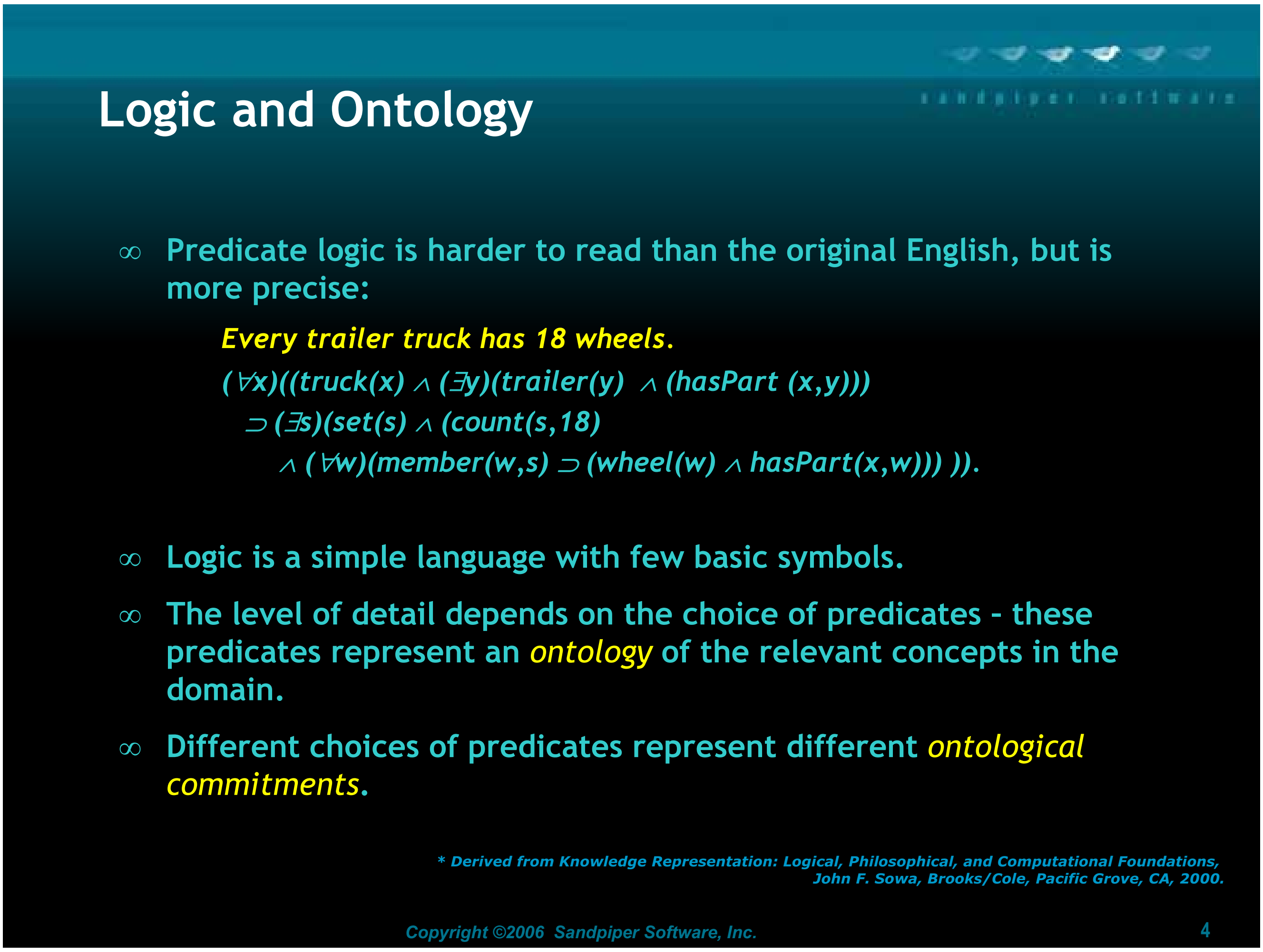
** Derived from Knowledge Representation: Logical, Philosophical, and Computational Foundations, John F. Sowa, Brooks/Cole, Pacific Grove, CA, 2000.*



# Historical Context

- **Boolean Algebra** - George Boole (1854)
- **Truth Tables** - Charles Sanders Peirce (chose inclusive or, added if-then (implications), algebraic notation (1883)
- Frege (1879) - **predicate logic** or predicate calculus
- Giuseppe Peano - first *mixed* mathematical and logical notation, and *inverted* certain logical symbols (∃, ⊃, ∀, ∧, ∨, ~, ≡)
- Bertrand Russell adopted this notation (Whitehead and Russell, 1910), which is the basis for most work in logic to the present

** Derived from Knowledge Representation: Logical, Philosophical, and Computational Foundations, John F. Sowa, Brooks/Cole, Pacific Grove, CA, 2000.*



# Logic and Ontology

- Predicate logic is harder to read than the original English, but is more precise:

  *Every trailer truck has 18 wheels.*

  $$(\forall x)((truck(x) \wedge (\exists y)(trailer(y) \;\wedge (hasPart\ (x,y)))$$
  $$\supset (\exists s)(set(s) \wedge (count(s,18)$$
  $$\wedge\ (\forall w)(member(w,s) \supset (wheel(w) \wedge hasPart(x,w)))\ )).$$

- Logic is a simple language with few basic symbols.
- The level of detail depends on the choice of predicates - these predicates represent an *ontology* of the relevant concepts in the domain.
- Different choices of predicates represent different *ontological commitments*.

** Derived from Knowledge Representation: Logical, Philosophical, and Computational Foundations, John F. Sowa, Brooks/Cole, Pacific Grove, CA, 2000.*

# Definitions

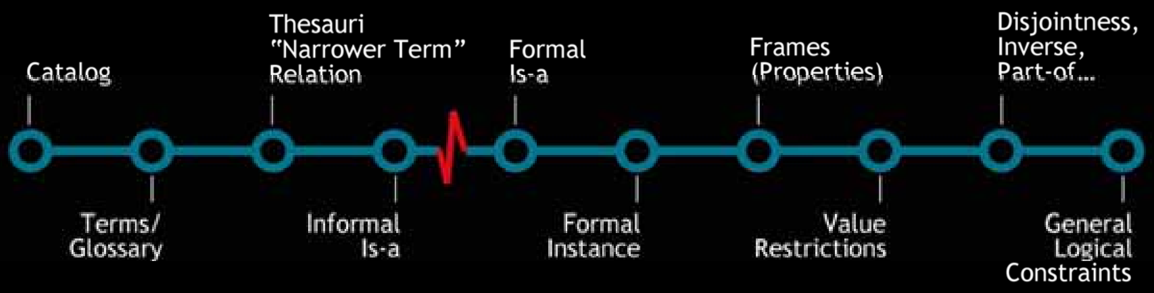
- **An ontology** is a specification of a conceptualization.
  - *Tom Gruber*
- **Knowledge engineering** is the application of logic and ontology to the task of building computable models of some domain for some purpose. - *John Sowa*
- **Artificial Intelligence** can be viewed as the study of intelligent behavior achieved through computational means. Knowledge Representation then is the part of AI that is concerned with how an agent uses what it knows in deciding what to do -*Brachman and Levesque - KR&R*
- **Knowledge representation** means that knowledge is formalized in a symbolic form, that is, to find a symbolic expression that can be interpreted. - *Klein and Methlie*
- The task of **classifying** all the words of language, or what's the same thing, all the ideas that seek expression, is the most stupendous of logical tasks. Anybody but the most accomplished logician must break down in it utterly; and even for the strongest man, it is the severest possible tax on the logical equipment and faculty. - *Charles Sanders Peirce*, letter to editor B. E. Smith of the *Century Dictionary*

# Level Setting

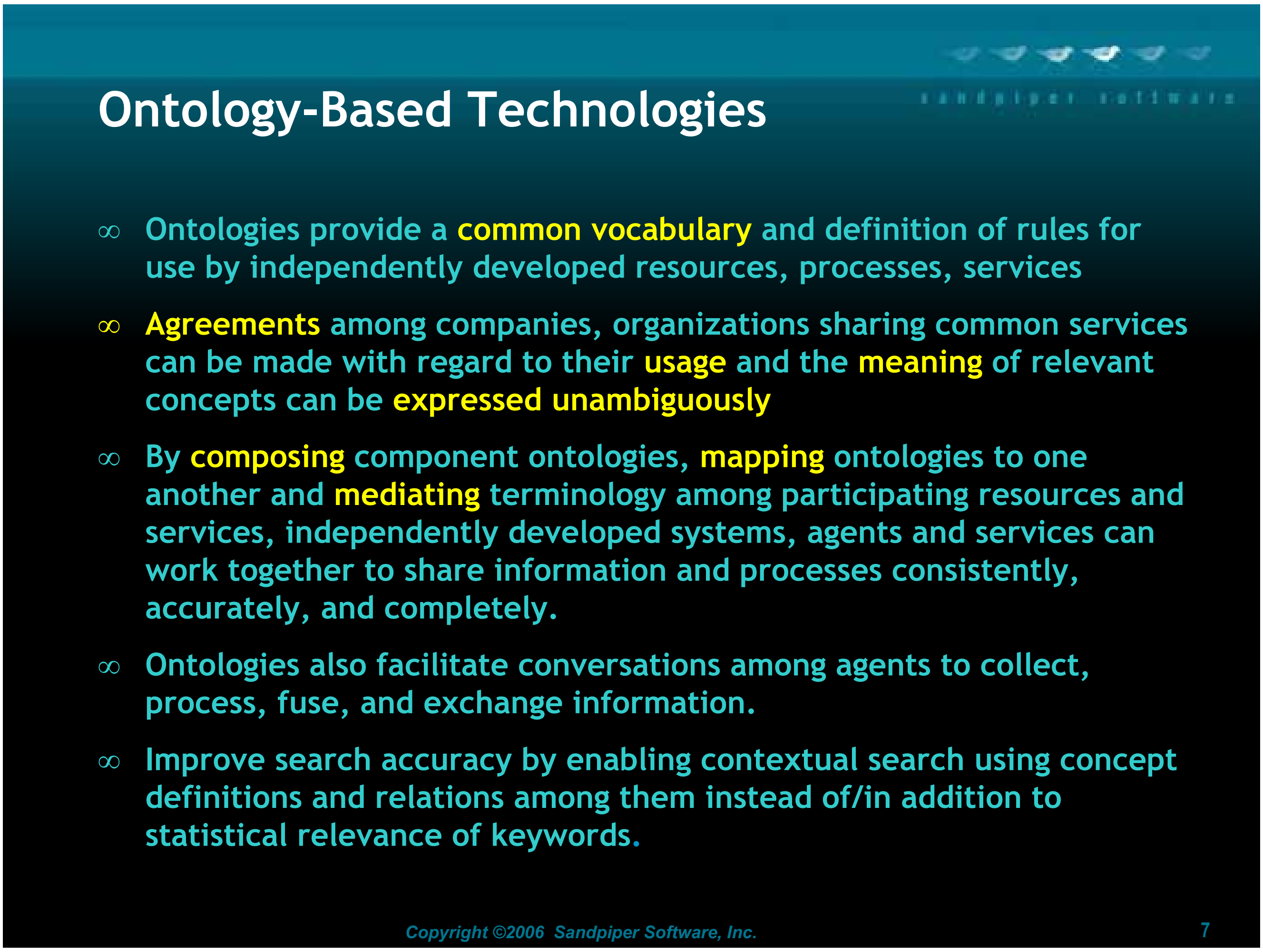
**An ontology** specifies a rich description of the

- Terminology, concepts, nomenclature
- Properties explicitly defining concepts
- Relations among concepts (hierarchical and lattice)
- Rules distinguishing concepts, refining definitions and relations (constraints, restrictions, regular expressions)

relevant to a particular domain or area of interest.

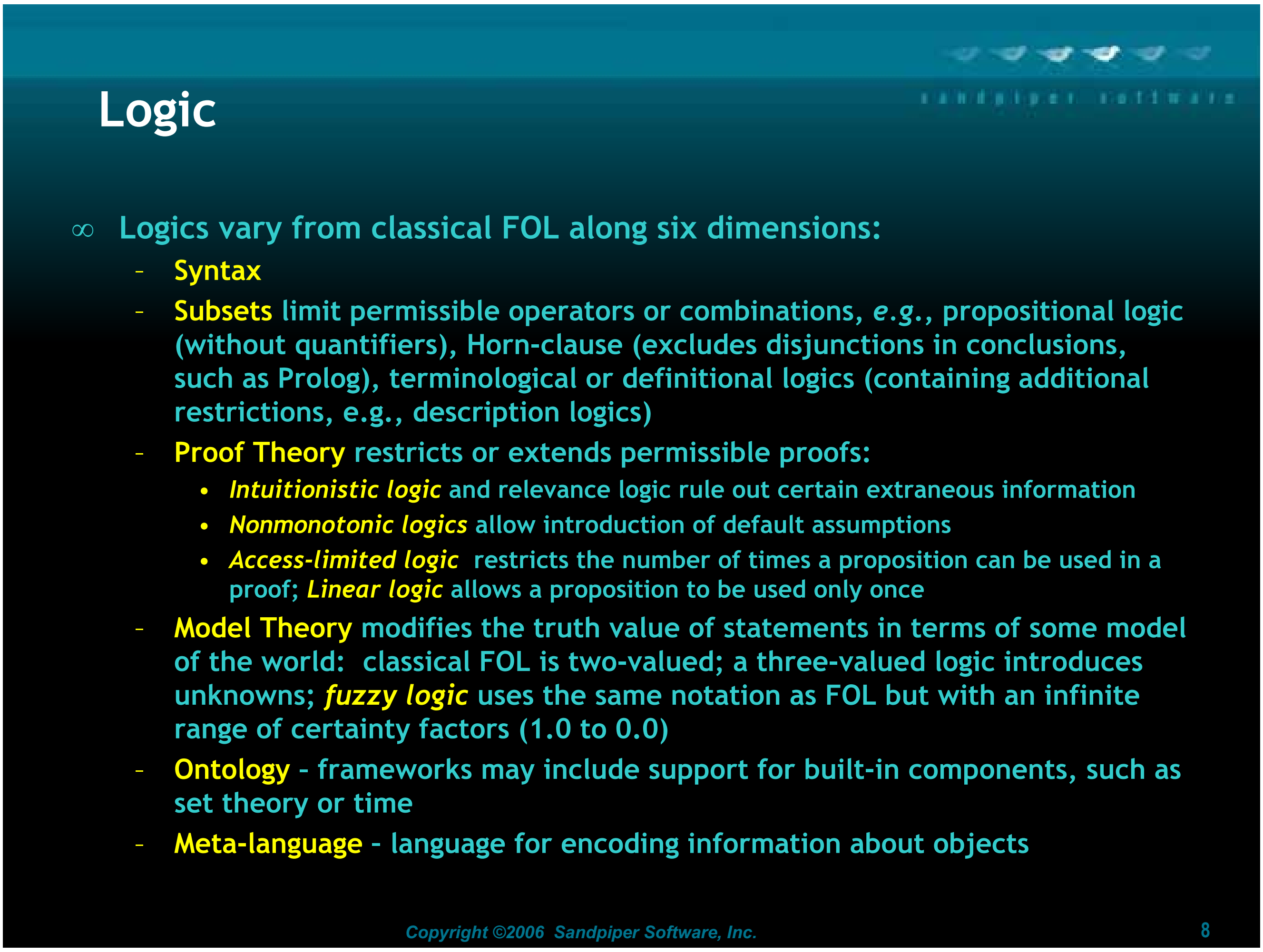


**Based On Aaai '99 Ontologies Panel - Mcguinness, Welty, Ushold, Gruninger, Lehmann*



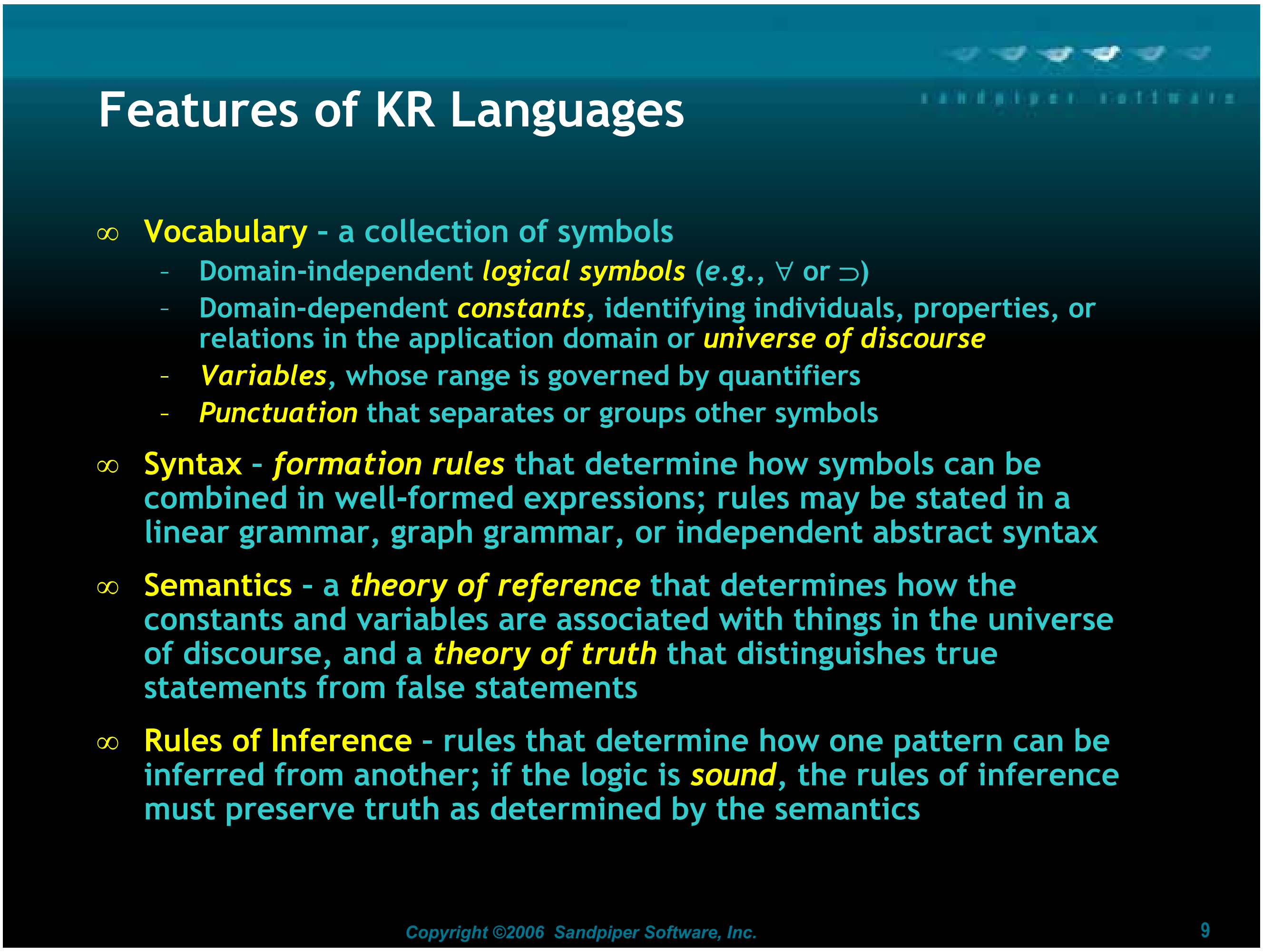
## Ontology-Based Technologies

- Ontologies provide a **common vocabulary** and definition of rules for use by independently developed resources, processes, services
- **Agreements** among companies, organizations sharing common services can be made with regard to their **usage** and the **meaning** of relevant concepts can be **expressed unambiguously**
- By **composing** component ontologies, **mapping** ontologies to one another and **mediating** terminology among participating resources and services, independently developed systems, agents and services can work together to share information and processes consistently, accurately, and completely.
- Ontologies also facilitate conversations among agents to collect, process, fuse, and exchange information.
- Improve search accuracy by enabling contextual search using concept definitions and relations among them instead of/in addition to statistical relevance of keywords.



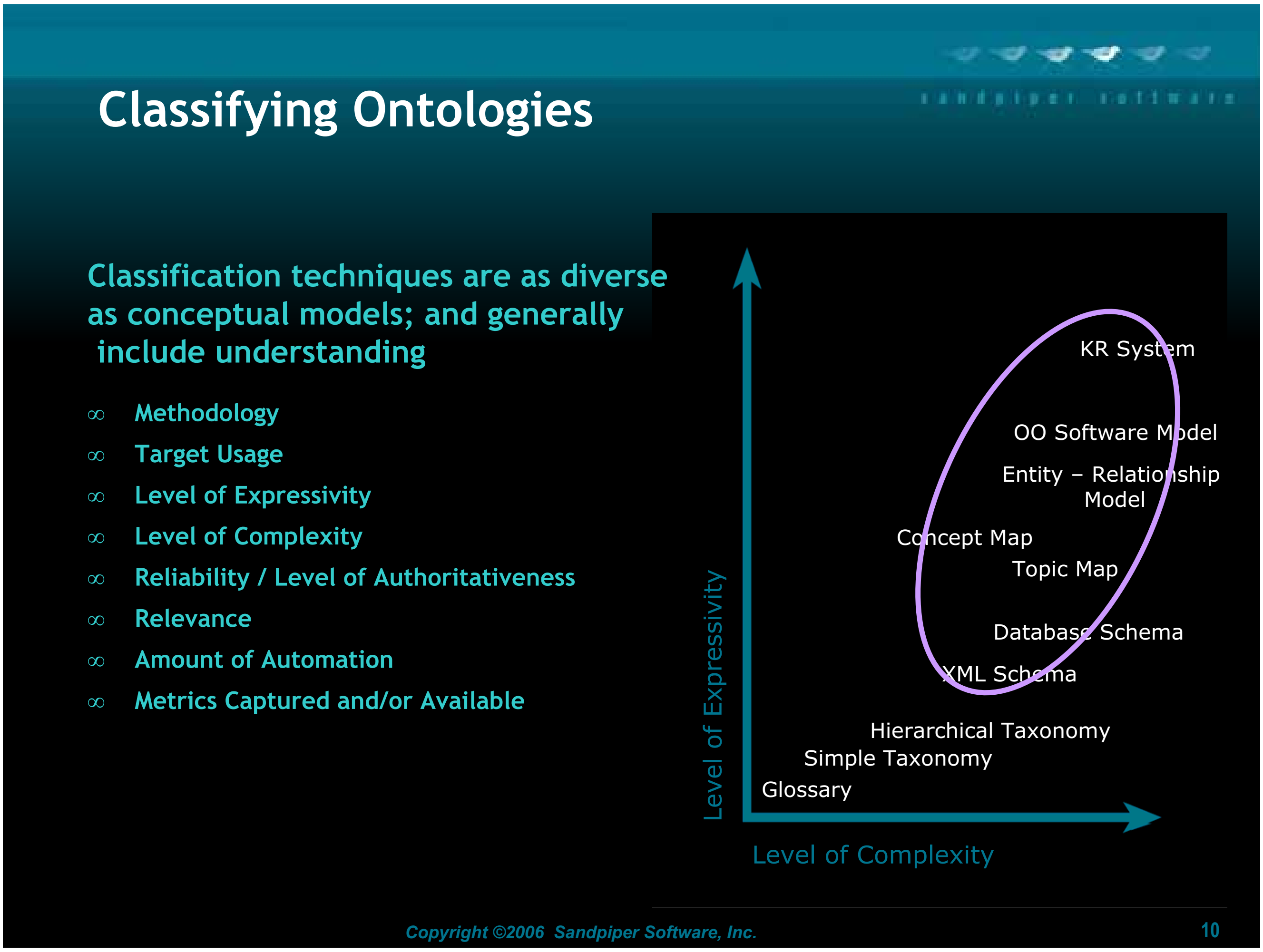
## Logic

- Logics vary from classical FOL along six dimensions:
  - **Syntax**
  - **Subsets** limit permissible operators or combinations, *e.g.*, propositional logic (without quantifiers), Horn-clause (excludes disjunctions in conclusions, such as Prolog), terminological or definitional logics (containing additional restrictions, e.g., description logics)
  - **Proof Theory** restricts or extends permissible proofs:
    - *Intuitionistic logic* and relevance logic rule out certain extraneous information
    - *Nonmonotonic logics* allow introduction of default assumptions
    - *Access-limited logic* restricts the number of times a proposition can be used in a proof; *Linear logic* allows a proposition to be used only once
  - **Model Theory** modifies the truth value of statements in terms of some model of the world: classical FOL is two-valued; a three-valued logic introduces unknowns; *fuzzy logic* uses the same notation as FOL but with an infinite range of certainty factors (1.0 to 0.0)
  - **Ontology** - frameworks may include support for built-in components, such as set theory or time
  - **Meta-language** - language for encoding information about objects



# Features of KR Languages

- Vocabulary – a collection of symbols
  - Domain-independent *logical symbols* (*e.g.*, ∀ or ⊃)
  - Domain-dependent *constants*, identifying individuals, properties, or relations in the application domain or *universe of discourse*
  - *Variables*, whose range is governed by quantifiers
  - *Punctuation* that separates or groups other symbols
- Syntax – *formation rules* that determine how symbols can be combined in well-formed expressions; rules may be stated in a linear grammar, graph grammar, or independent abstract syntax
- Semantics – a *theory of reference* that determines how the constants and variables are associated with things in the universe of discourse, and a *theory of truth* that distinguishes true statements from false statements
- Rules of Inference – rules that determine how one pattern can be inferred from another; if the logic is *sound*, the rules of inference must preserve truth as determined by the semantics



# Classifying Ontologies

Classification techniques are as diverse as conceptual models; and generally include understanding

- Methodology
- Target Usage
- Level of Expressivity
- Level of Complexity
- Reliability / Level of Authoritativeness
- Relevance
- Amount of Automation
- Metrics Captured and/or Available

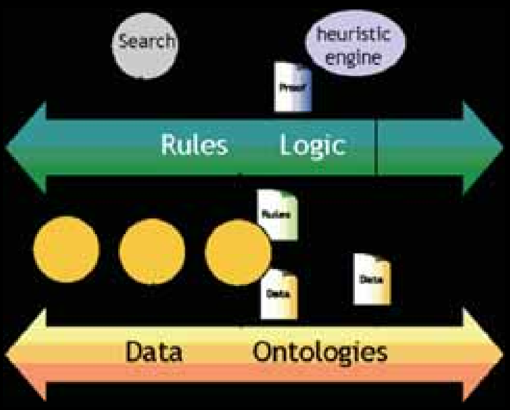
# Critical Questions

- ∞ Intended use of ontologies, including domain requirements (*e.g.*, scientific and engineering apps require formulas, units of measure, computations that are difficult to represent in DAML+OIL/OWL)
- ∞ Intended use of KRSs that implement them, including reasoning requirements, questions to be answered
- ∞ For distributed environments, the number and kinds of resources, processes, services requiring ontologies - how distributed, how unique, developed collaboratively or independently, dynamic community participation or static
- ∞ What kinds of transformations are required among processes, resources, services to support semantic mediation
- ∞ Ontology and KRS alignment / de-confliction / ambiguity resolution requirements
- ∞ Ontology and KRS composition requirements, dynamic vs. static composition, in what environment and under what constraints
- ∞ Performance, sizing, timing requirements of target environment

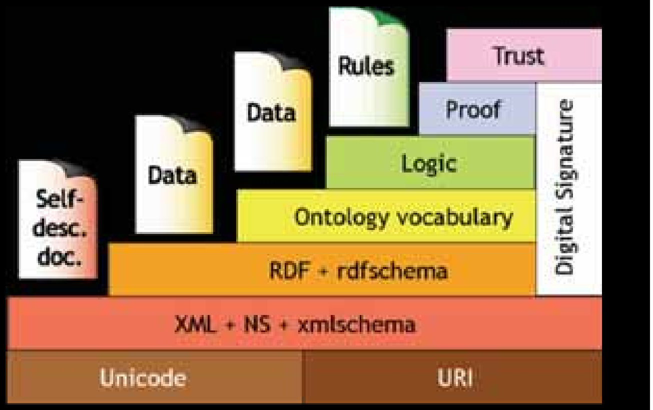
# The Semantic Web

*"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."*

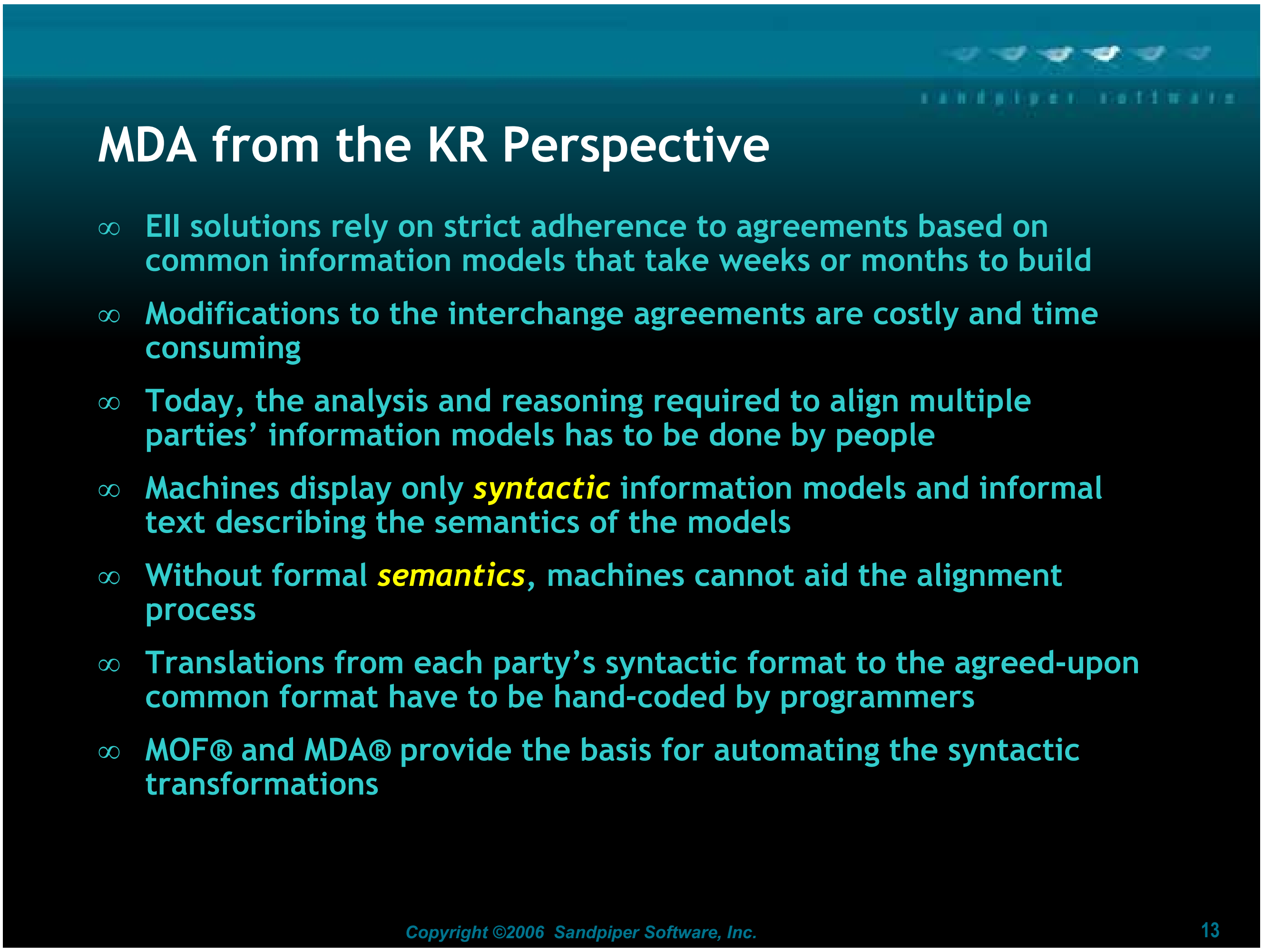
*-- Tim Berners-Lee*



*"the bus"*

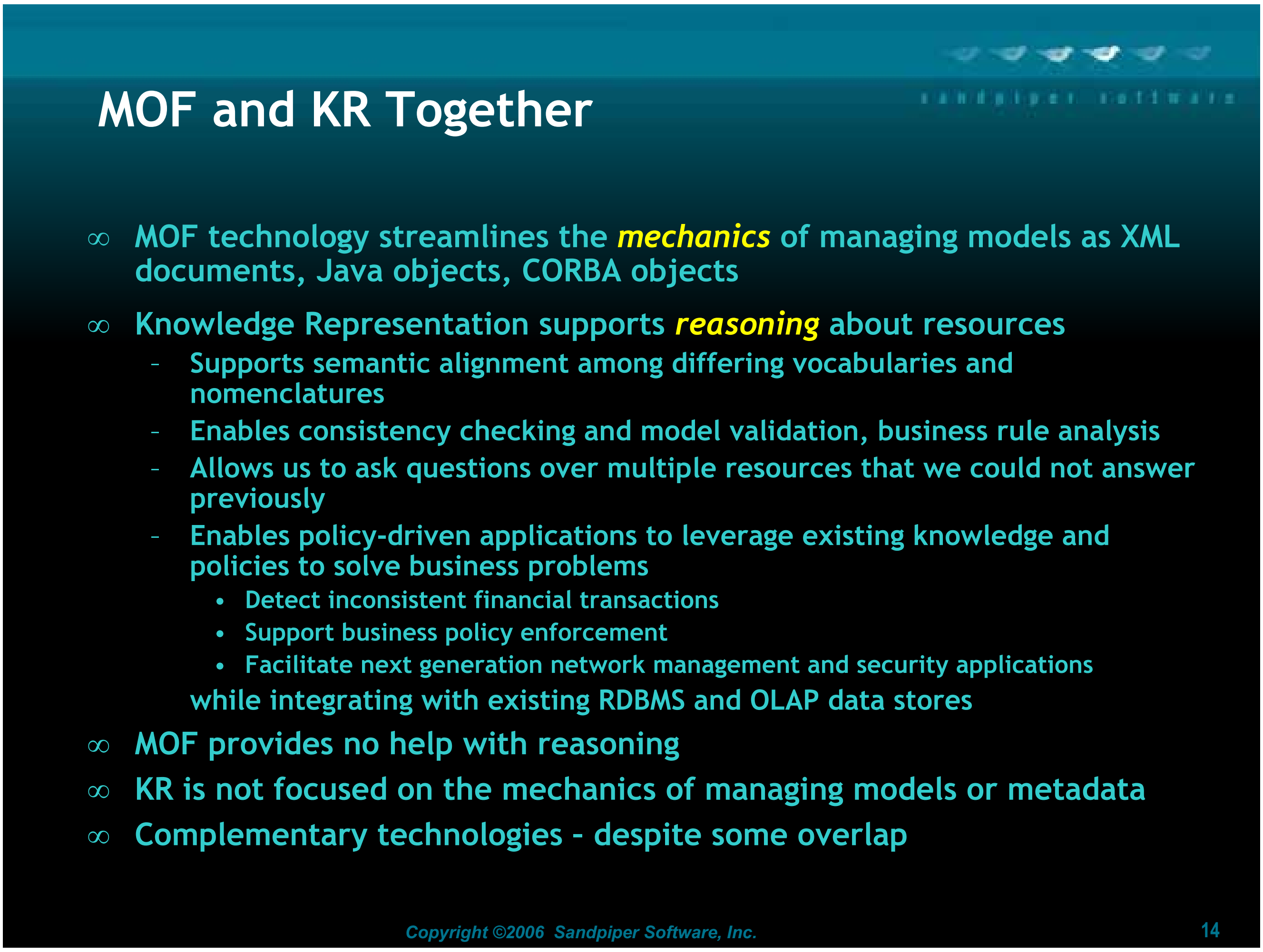


*"the wedding cake"*



## MDA from the KR Perspective

- EII solutions rely on strict adherence to agreements based on common information models that take weeks or months to build
- Modifications to the interchange agreements are costly and time consuming
- Today, the analysis and reasoning required to align multiple parties' information models has to be done by people
- Machines display only ***syntactic*** information models and informal text describing the semantics of the models
- Without formal ***semantics***, machines cannot aid the alignment process
- Translations from each party's syntactic format to the agreed-upon common format have to be hand-coded by programmers
- MOF® and MDA® provide the basis for automating the syntactic transformations



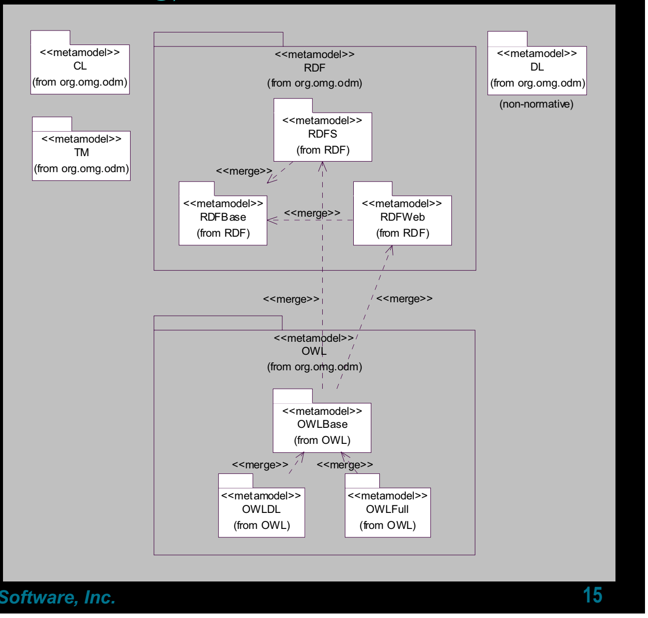
## MOF and KR Together

- MOF technology streamlines the ***mechanics*** of managing models as XML documents, Java objects, CORBA objects
- Knowledge Representation supports ***reasoning*** about resources
  - Supports semantic alignment among differing vocabularies and nomenclatures
  - Enables consistency checking and model validation, business rule analysis
  - Allows us to ask questions over multiple resources that we could not answer previously
  - Enables policy-driven applications to leverage existing knowledge and policies to solve business problems
    - Detect inconsistent financial transactions
    - Support business policy enforcement
    - Facilitate next generation network management and security applications

    while integrating with existing RDBMS and OLAP data stores
- MOF provides no help with reasoning
- KR is not focused on the mechanics of managing models or metadata
- Complementary technologies - despite some overlap

## Ontology Definition Metamodel

- Five EMOF platform independent metamodels, (PIMs), four are normative
- Mappings (MOF QVT for UML ▸ OWL DL & TM ▸ OWL, abstract syntax for CL embedding)
- UML2 Profiles
  - RDF & OWL
  - Topic Maps
- Generated collateral
  - Metamodels/XMI (provided)
  - Proof of concepts
  - Java APIs
- Conformance
  - RDF/S & OWL – several combinations
  - All else optional

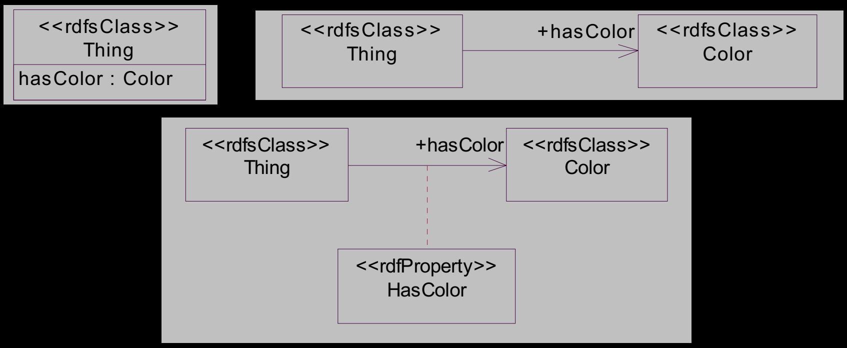


## The UML Profile for RDF & OWL

- Intended to be highly intuitive for UML users
- Reuses UML constructs when they have the same semantics as OWL
- When this is not possible, stereotypes UML constructs that are consistent and as close as possible to OWL semantics
- Uses standard UML 2 notation
- In the few cases where this is not possible, follows the clarifications and elaborations of stereotype notation defined in UML 2.1
- Leverages the model library included in Appendix A for a number of constructs, for example statements, rdf:value, container and list elements, as well as built-in properties

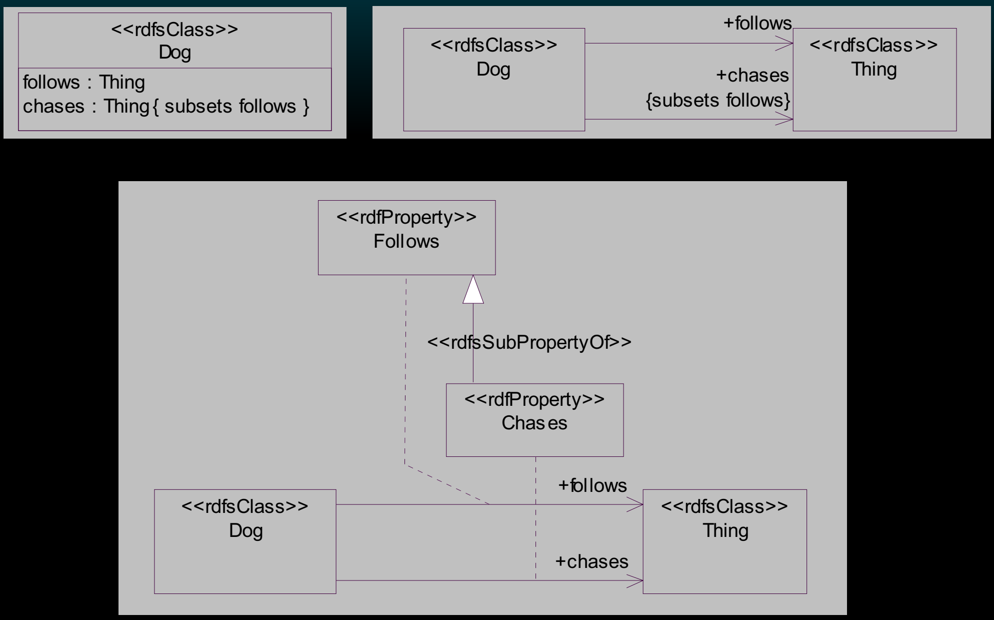
# Key Features of the RDF Profile

- rdfs:Resource is modeled as UML::InstanceSpecification
- Introduction of <<reifies>> stereotype of UML::Dependency to allow such instance specifications to reify classes, properties, individuals, statements, etc.
- rdf:Property is modeled as UML::AssociationClass *and* UML::Property, to provide greatest possible flexibility
- Several possible representations of various aspects of rdf:Property:



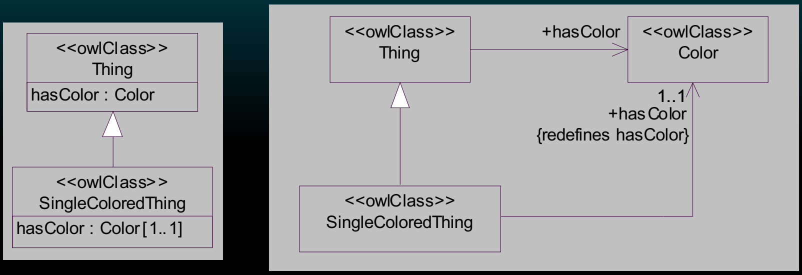
*Alternate forms for rdf:Property, without a specified domain*

# RDF Property Subsetting Options

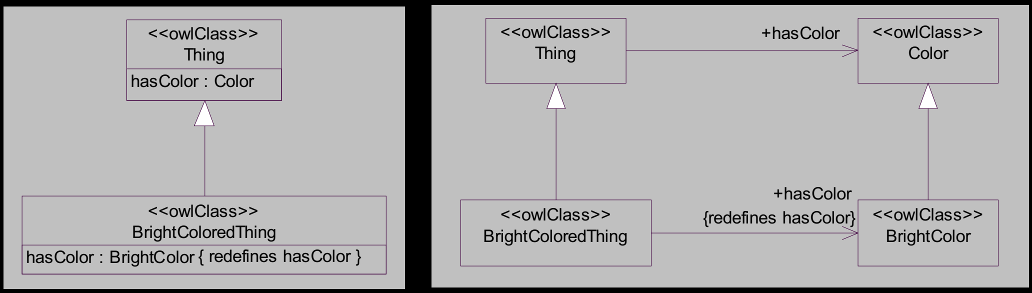


*Alternate forms for rdf:Property, without a specified range*

# Example OWL Number, Value Constraints

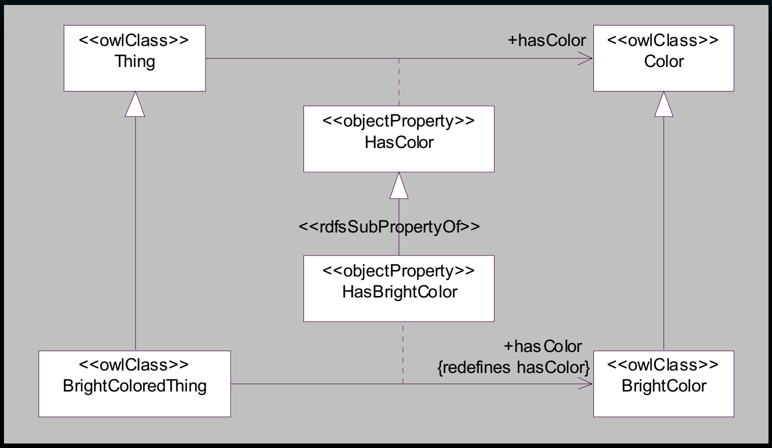


*OWL Cardinality – Restricted Mulitplicity in Subtype*

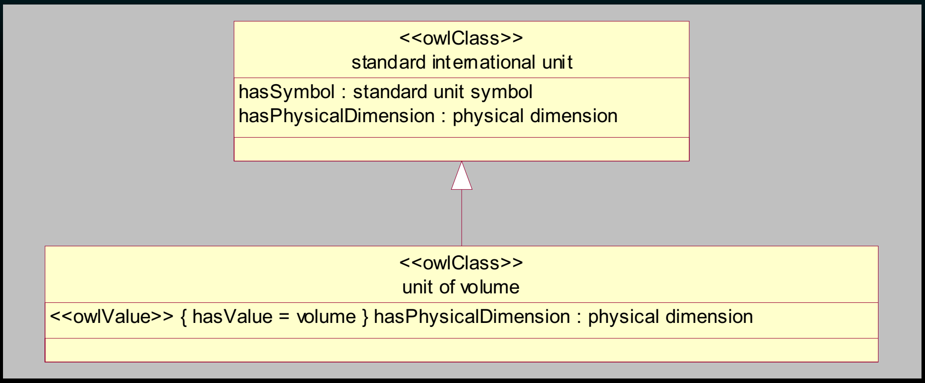


*OWL allValuesFrom – Property Redefinition*

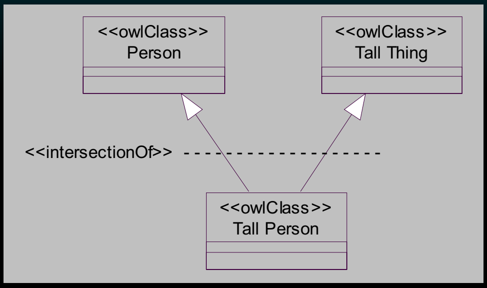
# OWL Property Redefinition (allValuesFrom) Using Association Classes

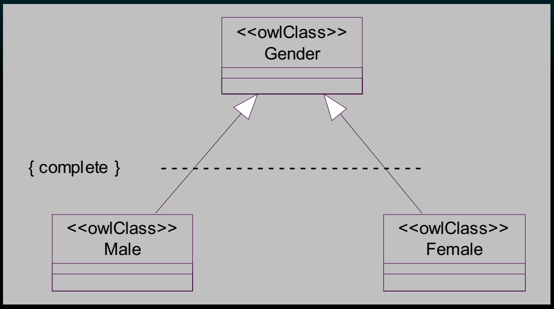


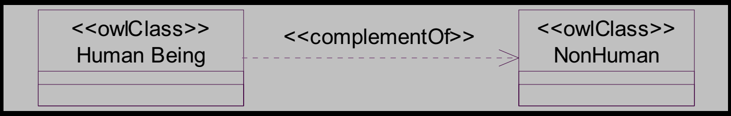
# OWL Property Redefinition (hasValue)


<<owlClass>>
standard international unit
hasSymbol : standard unit symbol
hasPhysicalDimension : physical dimension
<<owlClass>>
unit of volume
<<owlValue>> { hasValue = volume } hasPhysicalDimension : physical dimension

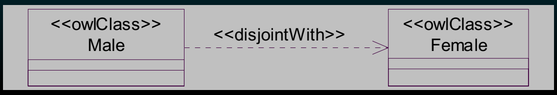
# OWL Intersection, Union, Complement



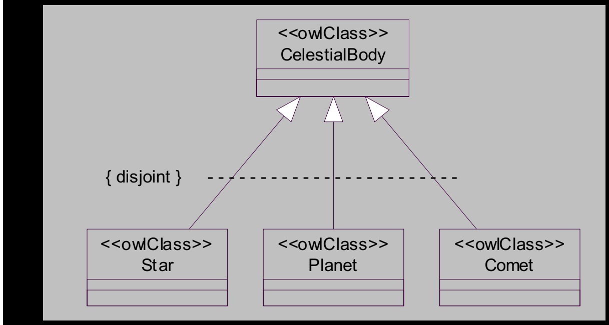




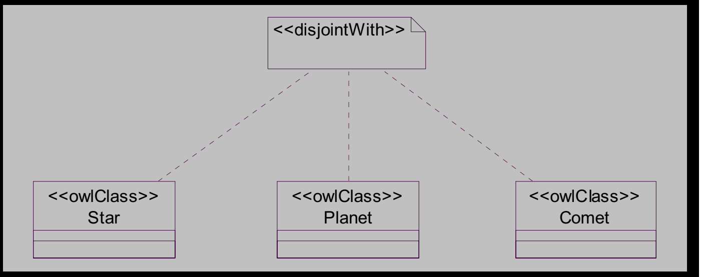
# OWL Disjointness Options



*Simple binary disjoint relationship*

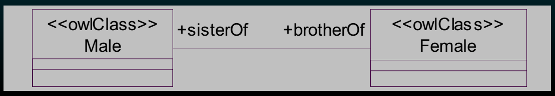


*Disjointness, multiple participants, common parent*

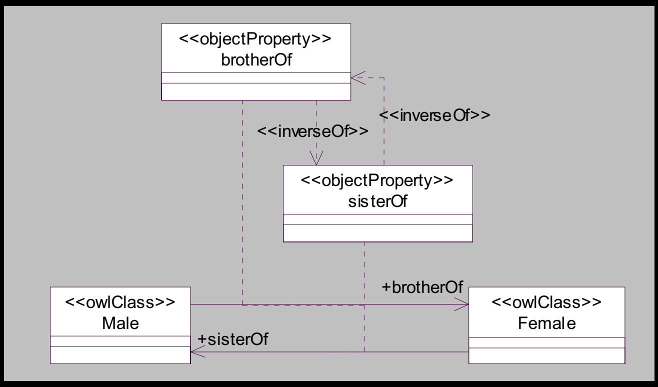


*Disjointness, multiple participants, no common parent*

# OWL Inverse Options



*Simple inverse relationship*

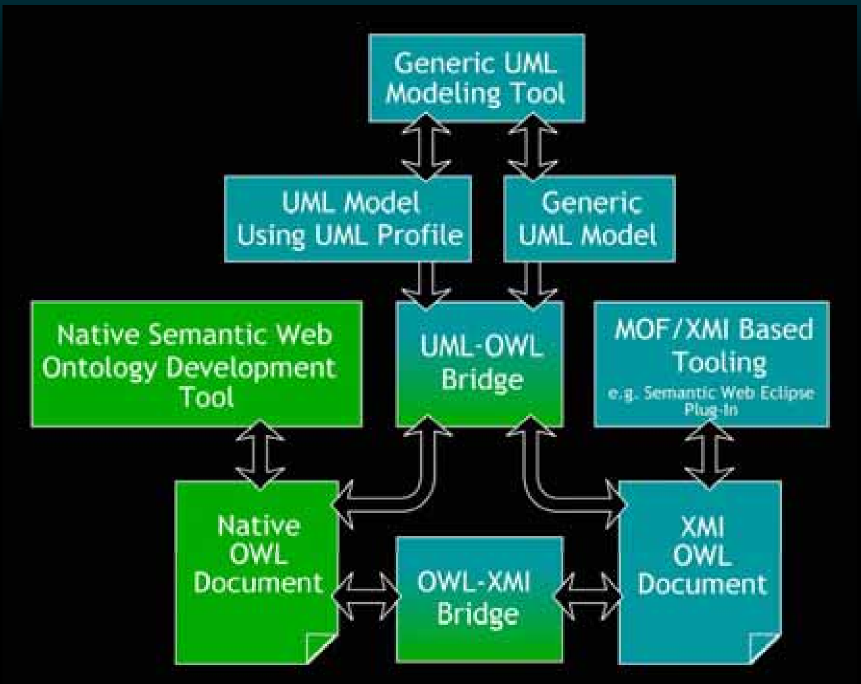


*Inverse relationship among association classes*

## ODM Status

- Platform Independent (Normative) Metamodels (PIMs) include
  - RDF & OWL - abstract syntax, constraints for OWL DL & OWL Full, several compliance options
  - ISO Common Logic (CL)
  - ISO Topic Maps (TM)
- Informative Models
  - DL Core - high-level, relatively unconstrained Description Logics based metamodel (non-normative, informational)
  - Identifier (keys) model extension to UML for ER
- Latest revised submission posted 6/5 to the OMG web site *(http://www.omg.org/docs/ad/06-05-01.pdf)*
- Recommended for Adoption in ADTF 06/28/06
- Next steps include presentation to AB 06/29/06, vote for adoption in PTC beginning 06/30/06

## Bridging KR and MDA



Generic UML Modeling Tool

UML Model Using UML Profile

Generic UML Model

Native Semantic Web Ontology Development Tool

UML-OWL Bridge

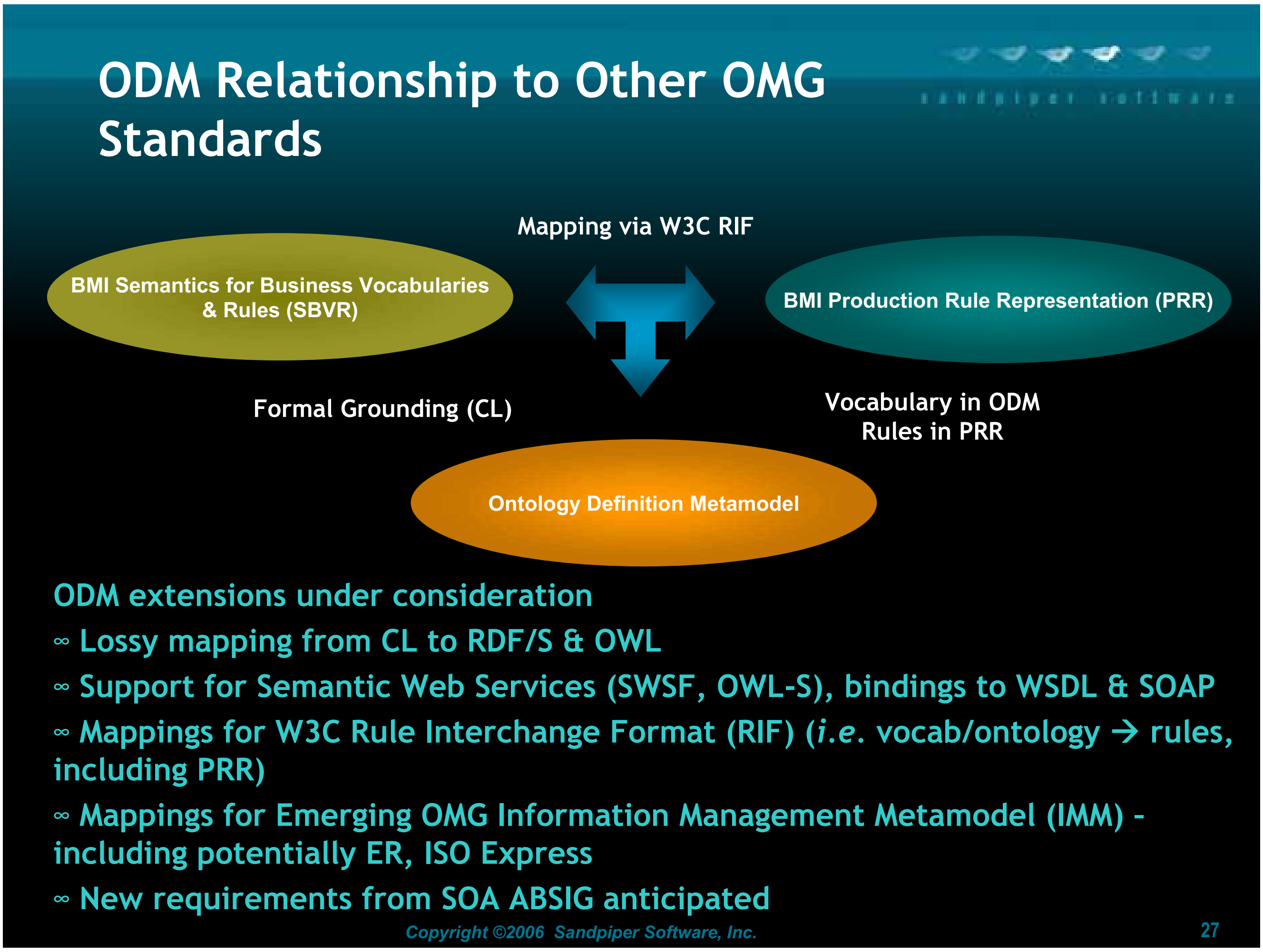
MOF/XMI Based Tooling

e.g. Semantic Web Eclipse Plug-In

Native OWL Document

OWL-XMI Bridge

XMI OWL Document



# ODM Relationship to Other OMG Standards

Mapping via W3C RIF

BMI Semantics for Business Vocabularies & Rules (SBVR)

BMI Production Rule Representation (PRR)

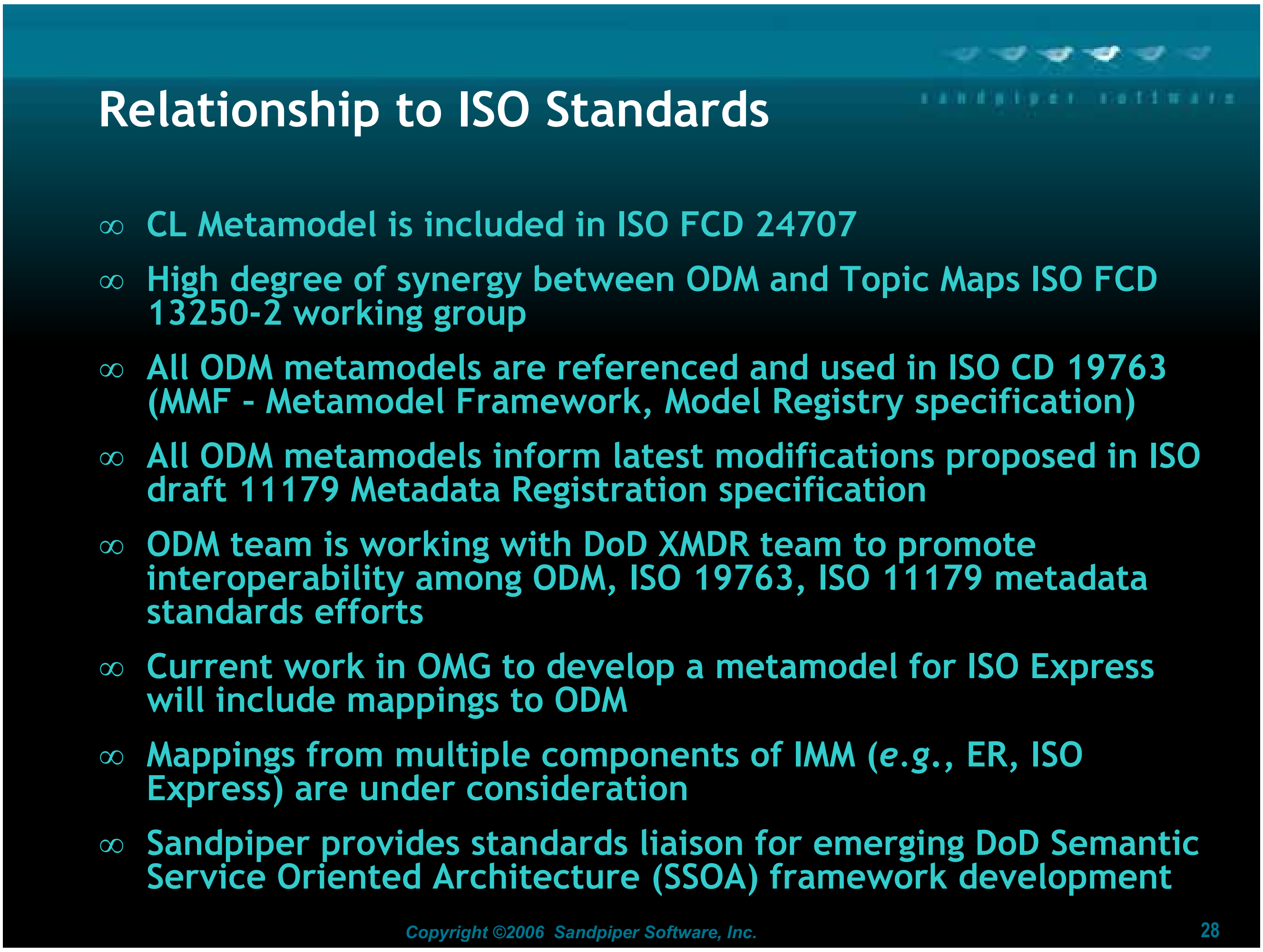
Formal Grounding (CL)

Vocabulary in ODM
Rules in PRR

Ontology Definition Metamodel

ODM extensions under consideration

- ∞ Lossy mapping from CL to RDF/S & OWL
- ∞ Support for Semantic Web Services (SWSF, OWL-S), bindings to WSDL & SOAP
- ∞ Mappings for W3C Rule Interchange Format (RIF) (*i.e.* vocab/ontology → rules, including PRR)
- ∞ Mappings for Emerging OMG Information Management Metamodel (IMM) - including potentially ER, ISO Express
- ∞ New requirements from SOA ABSIG anticipated



# Relationship to ISO Standards

- ∞ CL Metamodel is included in ISO FCD 24707
- ∞ High degree of synergy between ODM and Topic Maps ISO FCD 13250-2 working group
- ∞ All ODM metamodels are referenced and used in ISO CD 19763 (MMF - Metamodel Framework, Model Registry specification)
- ∞ All ODM metamodels inform latest modifications proposed in ISO draft 11179 Metadata Registration specification
- ∞ ODM team is working with DoD XMDR team to promote interoperability among ODM, ISO 19763, ISO 11179 metadata standards efforts
- ∞ Current work in OMG to develop a metamodel for ISO Express will include mappings to ODM
- ∞ Mappings from multiple components of IMM (*e.g.*, ER, ISO Express) are under consideration
- ∞ Sandpiper provides standards liaison for emerging DoD Semantic Service Oriented Architecture (SSOA) framework development

# Summary of the Robotics DTF Plenary – DRAFT –

Jun 28-29, 2006<br>
Boston, MA, USA<br>
(robotics/2006-06-35)

**Meeting Highlights**<br>
RTC submission recommended for adoption by MARS<br>
1 special talk was given by the Ontology PSIG<br>
Reports received from 3 active Technical WGs<br>
Contact reports received for a number of activities<br>
Contacts Sub-Committee formed

Prof. Makoto Mizukawa was authorized as a contact between ISO TC184/SC2.

**List of generated documents**<br>
robotics/2006-06-03 Final Agenda (Tetsuo Kotoku)<br>
robotics/2006-06-04 St. Louis Meeting Minutes [approved] (Hung Pham)<br>
robotics/2006-06-05 Roadmap for Robotics Activities (Tetsuo Kotoku)<br>
robotics/2006-06-05 Roadmap fot Robotics Activities (Tetsuo Kotoku)<br>
robotics/2006-06-06 Robotics Services WG: Introduction to RUPI (Soo-Young Chi)<br>
robotics/2006-06-07 Robotics Services WG: Definition of Functional Services (Olivier Lemaire)<br>
robotics/2006-06-08 Profile WG: Discussion on profile standardization (Seung-Ik Lee)<br>
robotics/2006-06-09 Steering Committee Presentation (Tetsuo Kotoku)<br>
robotics/2006-06-10 Publicity SC: Flyer discussion (Masayoshi Yokomachi)<br>
robotics/2006-06-11 Publicity SC: Flyer rough draft (Abheek Bose)<br>
robotics/2006-06-12 Robotic Services WG: POEM - an implementation of position estimation module (Takashi Tsubouchi)<br>
robotics/2006-06-13 Robotic Services WG: SAIT Proposal of Standards for Localization based on MDA (Yeon-Ho Kim)<br>
robotics/2006-06-14 Robotic Services WG: Issues on Localization Services (Wonpil Yu)<br>
robotics/2006-06-15 Robotic Services WG: Discussion on Localization Technology (Olivier Lemaire)<br>
robotics/2006-06-16 Spur - a Locomotion Command System for Mobile Robot (Takashi Tsubouchi)<br>
robotics/2006-06-17 Robot Modeling Framework (Abheek Bose)<br>
robotics/2006-06-18 Infrastructure WG: Understanding RSCA with Example (Seongsoo Hong)<br>
robotics/2006-06-19 Infrastructure WG: Agenda for Infrastracture WG (Saehwa Kim)<br>
robotics/2006-06-20 Infrastructure WG: Infrastructure WG minutes (Saehwa Kim)<br>
robotics/2006-06-21 Plenary Opening / Ending Presentation (Tetsuo Kotoku)<br>
robotics/2006-06-22 Robobusiness2006 Presentation (Jon Siegel)<br>
robotics/2006-06-23 Robotic Device and Data Profile WG Report (Seung-Ik Lee)<br>
robotics/2006-06-24 Robotic Functional Services WG Report (Olivier Lemaire)<br>
robotics/2006-06-25 Infrastructure WG Report (Rick Warren)<br>
robotics/2006-06-26 Introduction to RTC (Rick Warren)<br>
robotics/2006-06-27 Contact Report: ISO TC184 / SC2 (Makoto Mizukawa)<br>
robotics/2006-06-28 Contact Report: Introduction of JAUS for the benefit of Robotic Standardization (Wataru Inamura)<br>
robotics/2006-06-29 Contact Report: KIRSF (Yun-Koo Chung)<br>
robotics/2006-06-30 Flyer first Draft candidate #1 (Abheek Bose)

robotics/2006-06-31 Flyer first Draft candidate #2 (Abheek Bose)
robotics/2006-06-32 Agenda Coordination for Anaheim Meeting (Tetsuo Kotoku)
robotics/2006-06-33 Ontology Definition Metamodel (ODM) Background and Overview (Elisa Kendall)
robotics/2006-06-34 DTC Report Presentation (Hung Pham)
robotics/2006-06-35 Boston Meeting Minutes - DRAFT (Hung Pham and Olivier Lemaire)
robotics/2006-06-36 MARS RTC Presentation [same as mars/2006-06-22] (Rick Warren)

**Plenary proceedings**

*Wednesday, Michelangelo Suite, 2nd FL*
Meeting called to order at 8:58am (Toku, AIST)

Review of the St. Louis Minutes (Toku)
(robotics/2006-06-04)
- Lee (ETRI) requested that the report document from the Profile WF be given a document number
- Mizukawa pointed out a name misspelling

***Action:***
- Toku (AIST) motioned to accept minutes
- Mizukawa (Shibaura-IT) seconded motion
- Chung (ETRI) suggested white ballot
- Motion passed w/o dissent.

Review of Agenda (Toku)
(robotics/2006-06-21)
- Voting list members of RTC encouraged to join MARS for vote of RTC on Thursday

**"RoboBusiness2006 Report" – Jon Seigel**
(robotics/2006-06-22)
- RoboNexus for consumer robots
- RoboBusiness for military robots and commercial robots
- Keynote speech stressed role of standards in FCS
- Microsoft announced Robot Development Kit
- Provided a brief review of the talk presented at RoboBusiness

***WG Reports***
Profile WG report (Lee, ETRI)
(robotics/2006-06-23)
- Following issues were discussed within WG
  * survey of typical devices
  * general approach
  * scope of WG
  * nomenclature and classification
- Potential issues to be addressed in RFP (tentative)
  * characteristics used to classify devices
  * device hierarchy
  * types of devices
  * interfaces
  * management and enumeration of devices
  * device configuration

  * how to integrate with existing standards
  * proof-of-concept, *i.e.*, provide definition
  * differences between devices and functional services?
- Roadmap discussion
  * RFP to be issued in Mar 2007
- Future informational presentations
  * IEEE 1451
  * JAUS
  * stressed need for presentations showing current implementation of RFP-related topics

Q: where is the divide between Functional Services and Robotic Devices?
A: the answer is not clear

Robotic Services WG report (Lemaire, JARA)
(robotics/2006-06-24)
- Review of presentations given to WG
  * RUPI presentation on Monday
  * Pose estimation module presentation (Tue)
  * Proposal for standards for robot localization (Tue)
- Roadmap update
  * Localization service RFP issue postponed to San Diego
- Discussion of naming

***Action:***
  * AIST motioned to rename WG to ***Robotic Functional Services WG***
  * SIT seconded motion
  * ETRI suggested white ballot
  * vote passed w/o dissent
  * ETRI motioned to rename Profile WG to ***Robotic Devices and Data Profiles WG***
  * JARA seconded
  * RTI suggested white ballot
  * vote passed w/o dissent

- Discussion summary
  * localization services could have a potentially very wide scope that we need to restrict
  * should focus on developer or user's point o fview
    + developer POV: define main typical building blocks of localization service so as to distribute them
    + user's POV: define only the external interfaces
  * should figure out how to evaluate the submissions
  * first RFP draft to be written prior to next (Anaheim) mtg

Infrastructure WG report (Warren, RTI)
(robotics/2006-06-25)
- Review of presentation given to WG
  * RSCA and an approach to deployment and configuration
- Discussion summary
  * review of minutes and notes from St Louis mtg
  * decided to delay issuing RFP by 1 mtg
  * draft written prior to Anaheim
  * volunteers to draft some or part of RFP
    + ADA software

+ AIST
+ RTI
+ SNU

Q (JARA): have you talked about this RFP to other OMG members?
A (RTI): we should do this
Q (JARA): you should present a clear motivation for why we need a new D&C.
A (RTI): it may be that we can leverage CORBA D&C or SDR D&C and extend it, but it's up to the respondents of the RFP to stress that.

Roadmap for DTF presented (Toku)
(robotics/2006-06-05)
- Updates were made based upon member input.
- ADA software suggested that supplement mailing list with WiKi.  Has taken AI to look into coordinating logistics with OMG and presenting to group about WiKi.

Progress report on RTC submission (Warren, RTI)
(robotics/2006-06-26)
- Presented current status; noted that vote-to-vote was passed
- formal vote for adoption to take place on Thu
- provided use-case example of navigation using RTC
- encouraged feedback on the RTC

***Contact reports***
Makoto Mizukawa (SIT) ---------
(robotics/2006-06-27)
ISO TC184/SC2 Plenary held in St Denis, France
- Scope: standardization in the filed of industrial automation and integration concerning discrete part manugfacturing and encomasssping the application of mult technologies, i.e. IT, machines and equip, etc
- 4 subcommittees:
  * physical device control
  * robots for industrial environments (PT10218)
  * industrial data architecture,
  * comms and integration frameworks
- long term direction
  * widened scope of "robots for industrial environments" to *robots and robotic devices*
    + standardization in the field of automatically controlled … robots and robotic devices;
    + excluding toys and military apps
  * started new initiative in *robotics in personal care*
  * exploring needs for standardization in the field of *service robots*

ISO TC 184/SC2 shall send liaison to IEEE "Robotics and Automation Society" and to OMG

ORiN and RAPI
- ORiN (Open Resource interface for the network)
- RAPI (Robot communication framework and Application Program Interface)
  + to distribute a new work item proposal based on RAPI to ISO TC 184/SC2
  + abstraction of ORiN
  + a subset of ORiN functionality

Wataru Inamura (IHI) ------------
(robotics/2006-06-28)
"Introduction of JAUS for the benefit of Robotic Standardization"
- Presented an overview of JAUS
- Recalled how to implement a JAUS-compliant component based upon the emerging RTC specification
- Showed an example of a JAUS-compliant subsystem
  + component definitions
  + message definitions
  + message formats
- suggested that we may be able to use JAUS as a starting point for our activities

Yun Koo Chung (ETRI) ------------
(robotics/2006-06-29)
Korean Intelligent Robot Standardization Forum (KIRSF) contact report
- Robot Unified Platform Initiative (RUPI)
  + initiative to be launched on Jul 4
  + specifications for testing and performance evaluation of commercial home service robots
  + communication protocols for URC robots
  + supports OMG and ISO standardization activities
- Held Korea-Japan Robot Standardization Workshop 2006 on Jun 16th in Jeju, Korea

***Action:***
AIST made motion to ask Prof Mizukawa to be contact person to ISO TC 182/SC2
JARA seconded motion
ETRI suggested white ballot
Motion passed without dissent

***Publicity Committee Report***
Abheek Bose (ADA Software)
(robotics/2006-06-30, -31)
- Presented two possible formats for the brochure
- Will send draft of brochure to mailing list to solicit feedback

***New Business***
(robotics/2006-06-21)
On forming the Contacts Sub-Committee consisting of Chung (ETRI) and Mizukawa (SIT)
- should there be Contacts Sub-committee to be formed (Toku)?
- why do we need this sub-committee (Lemaire)?
- we need to have people be in charge exchanging information back and forth with other organizations (Toku)

***Action:***
- RTI made motion to form Contact Sub-Committee
- AIST seconded motion
- motion passed w/o dissent

Upcoming publicity activities (Toku)
- IROS 2006 Workshop
  + Oct 9-15, Beijing, China
- SICE-ICASE International Joint Conference

Next meeting agenda was tentatively discussed / proposed (Toku)

Meeting was adjourned at 5:15 pm

*Thursday, Michelangelo Suite, 3rd$^{nd}$ FL*
09:45 am
Next Meeting Agenda Coordination – Tetsuo Kotoku
(robotics/2006-06-32)
Mon: (WG activity)
  AM: Infrastructure WG(2h)  drafting RFP
  AM: Service WG(2h) discussion
  PM1: Profile WG(2h) presentation(IEEE1451) + discussion
  PM2: Steering Committee
Tue: (WG activity)
  AM1: Service WG (2h) presentation (User Identification) + discussion
  PM1: Profile WG (1.5h) discussion
  PM2: Infrastructure WG (2h) discussion
Wed: (Plenary)
  SP1: John Hogg "Introduction to Zeligsoft Component Enabler 2.4?"
  SP2: Bruce Boyes "Microsoft Robotics Studio?"
  SP3: SysML  (ManTIS is postpone to the Washington DC (Tue. or Thu.))

10:00 am
**Special Talk : "Introduction to OMG Ontology-PSIG" – Elisa F. Kendall**
(robotics/2006-06-33)
- Ontology is vocabulary with formal set of rules to "dis-ambiguate" communications
  + Necessary for "reasoning engine"
- Ontology Definition Metamodel
  + Five EMOF platform independent metamodels (PIM)
  + Mappings
  + UML2 Profiles
    * RDF & OWL
    * Topic Maps
  + Generate collateral
  + Conformance

Meeting was adjourned at 11:00 am

## Participants:

Hideo Shindo (NEDO-DC)
Yun Koo Chung (ETRI)
Noriaki Ando (AIST)
Seiichi Shin (UEC)
Hung Pham (RTI)
Rick Warren (RTI)
Roger Burkhart (Deere & Company)
Eul Gyoon Lim (ETRI)
Soo Young Chi (ETRI)
Seung-Ik Lee (ETRI)
Fumio Ozaki (Toshiba)
Tomoki Yamashita (Maekawa MFG)
Eijiro Takeuchi (Tsukuba Univ.)
Takashi Tsubouchi (Tsukuba Univ.)

Takashi Suehiro (AIST)
Yeon Ho Kim (Samsung)
Olivier Lemaire (JARA)
Wataru Inamura (IHI)
Tetsuo Kotoku (AIST)
Wonpil Yu (ETRI)
Makoto Mizukawa (SIT)
Hiroyuki Nakamoto (SEC)
Saku Egawa (Hitachi)
Jim Kulp (Mercury)
Glenn Bakecki (Motorola)
John Hogg (Zeligsoft)
Robbin Teegarden (No Magic)
Elisa Kendall (Sandpiper Software)

Prepared and submitted by Hung Pham (RTI) and Olivier Lemaire (JARA).

mars/2006-06-22 and
robotics/2006-06-36

# Introduction to RTC

Robotic Technology Component Specification
Revised Submission

MARS, June 2006
Boston, MA

National Institute of Advanced
Science & Technology (AIST)

Real-Time Innovations
(RTI)

# Timeline

- September 2005: RFP issued
  - ptc/2005-09-01
- February 2006: Initial submissions
  - National Institute of Advanced Industrial Science and Technology (AIST)
    - mars/2006-01-05
    - Japan Robot Association (JARA) and Technologic Arts Incorporated join as supporters
  - Real-Time Innovations (RTI)
    - mars/2006-01-06
- June 2006: Revised submission
  - Joint submission by AIST and RTI
    - mars/2006-06-11
  - Seoul National University (SNU) joins as third supporter

# Problem Statement

- Domain: Distributed robotic systems
- Process problem characteristics
  - Lack of common programming practice(s) makes collaboration among developers difficult
- Technical problem characteristics
  - Complexity + state coherence
  - Behavioral design patterns
    - Autonomous components loosely coupled in time
    - Periodic ordered execution
    - Stimulus response

# Solution Requirements

- Address problem characteristics
- Provide well-defined interfaces
  - Improve reusability
  - Enable interoperability
    - Among components
    - Tool-to-component
- Promote component composition
  - Range of component granularities
  - Solution-driven reusability
  - Problem-driven decomposition

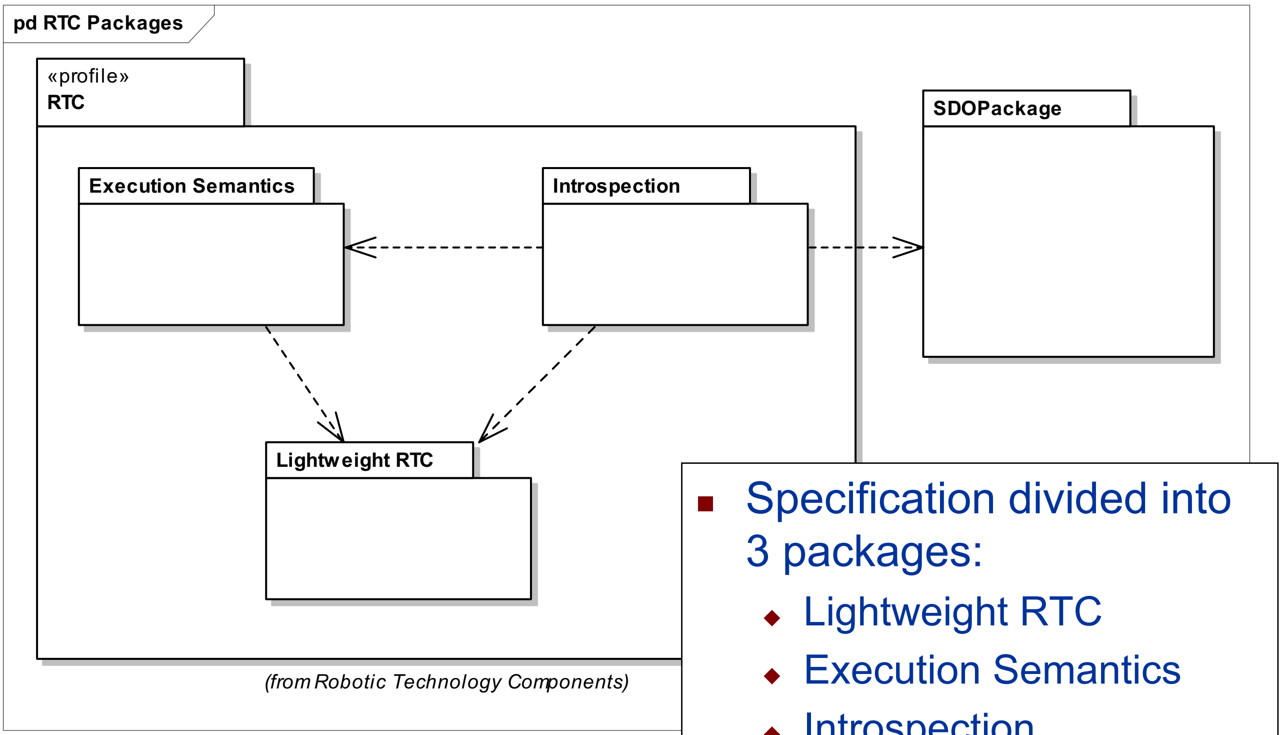
# Features of RTC

- Provides rich component lifecycle to enforce state coherency among components
- Defines data structures for describing components and other elements
- Supports fundamental design patterns
  - Dynamic composition of components collaborating periodically or asynchronously
  - Collaboration of fine-grained components tightly coupled in time (*e.g*. Simulink)
  - Stimulus response with finite state machines

# Relation to Existing Standards

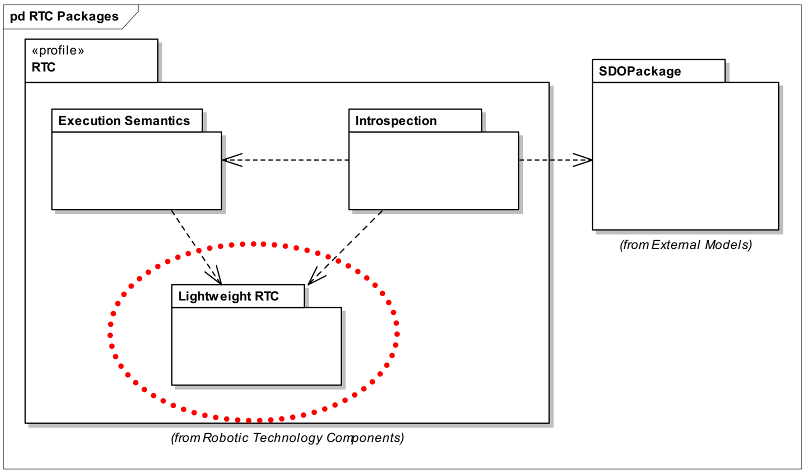
- UML
  - Domain-specific profile for UML components
- Super Distributed Objects (SDO)
  - Introspection of distributed components
  - Ports exposed as SDO services
- May be combined or implemented with another model
  - *e.g.* Lightweight CORBA Component Model
  - *e.g.* Software Radio components

# PIM Overview



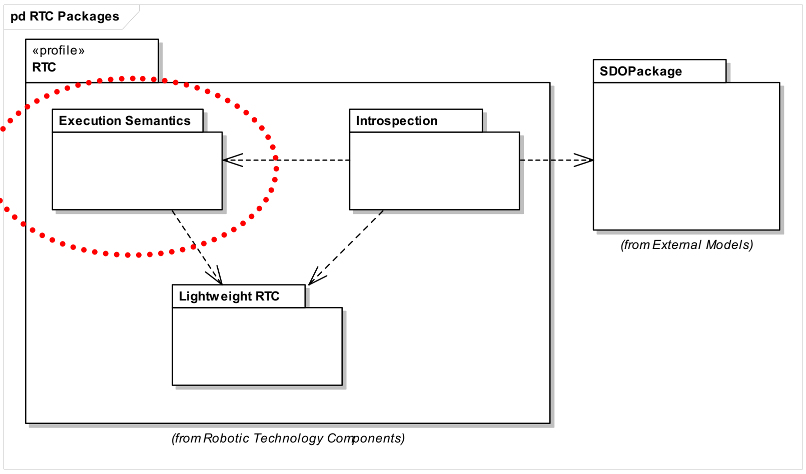
- Specification divided into 3 packages:
  - Lightweight RTC
  - Execution Semantics
  - Introspection

# PIM Overview



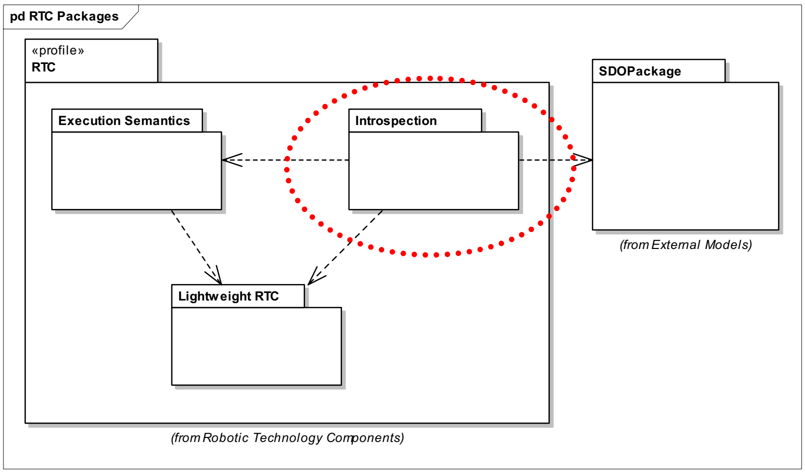
- Lightweight RTC
  - Minimum compliance point
  - Stereotypes and constraints for components, ports, and connectors
  - Component lifecycle
  - Baseline support for component execution

# PIM Overview



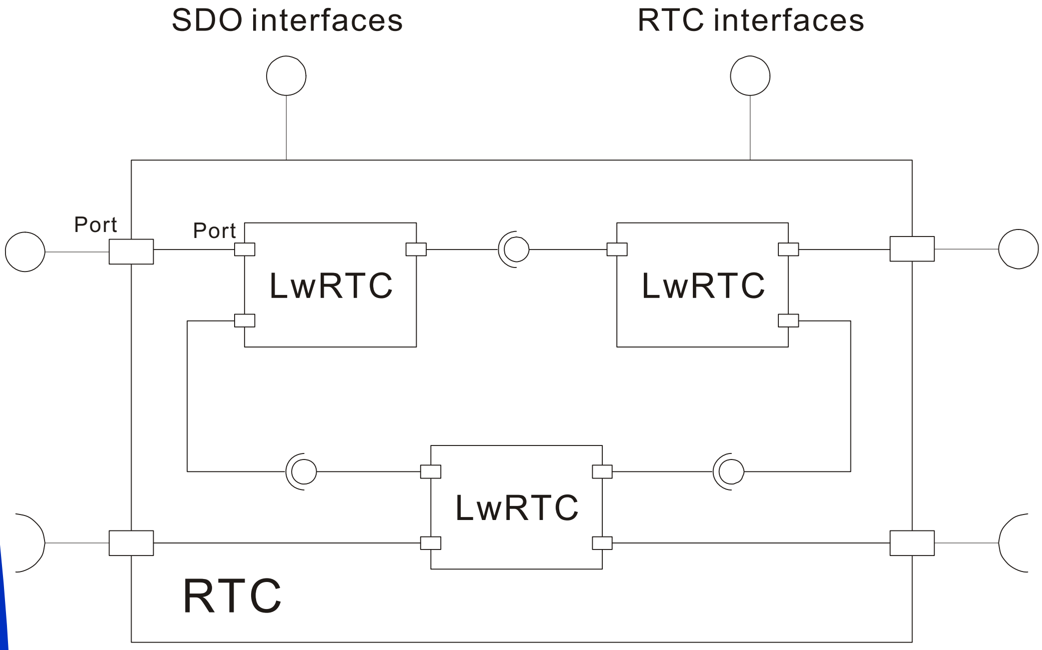
- Execution Semantics
  - Common behavioral design patterns

# PIM Overview



- Introspection
  - Query and modify component properties and connections at runtime
  - Based on Super-Distributed Objects (SDO)

# RT Component Example



# Platform-Specific Models

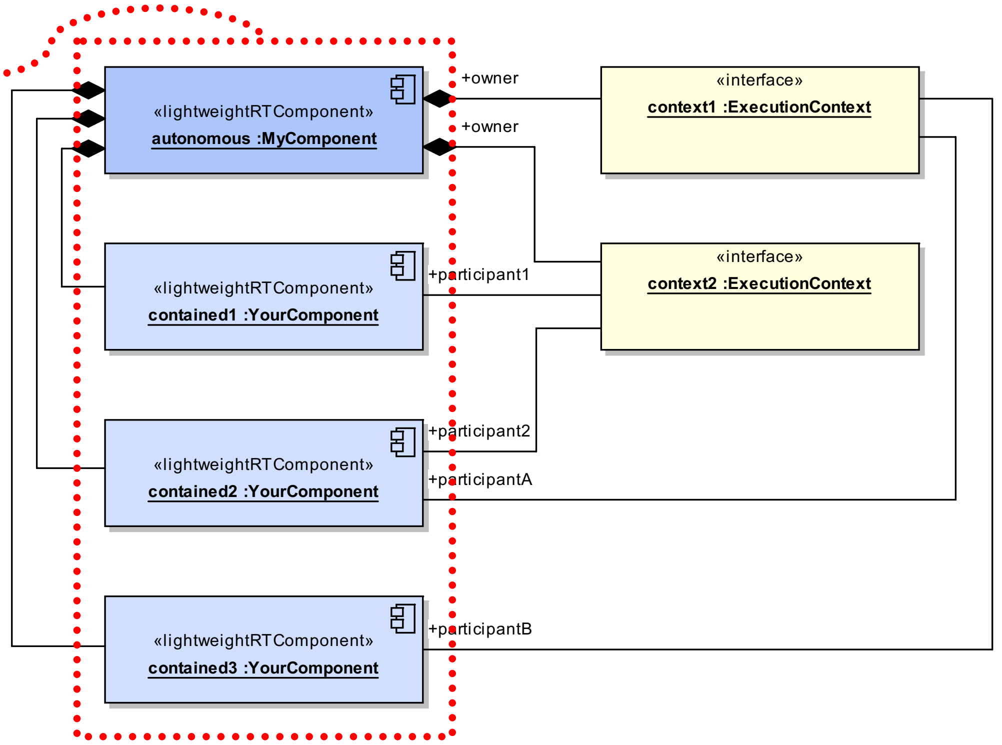
- CORBA IDL
- Lightweight CORBA Component Model
  - Distributed CORBA-based components
- Local components
  - Low-overhead communication in a single process

# Lightweight RTC Overview

- Conforming components extended by *lightweightRTComponent* stereotype
  - Component lifecycle: initialization, activation/deactivation, and finalization
  - Callbacks for responding to lifecycle changes
- ExecutionContext
  - Logical representation of thread
  - Collaboration pattern (“execution kind”)
  - “Participating” components can be started and stopped as a group
    - Components can participate in any number of execution contexts

# Components & Contexts

*Composite* component contains instances of other components



«lightweightRTComponent»
autonomous :MyComponent

«lightweightRTComponent»
contained1 :YourComponent

«lightweightRTComponent»
contained2 :YourComponent

«lightweightRTComponent»
contained3 :YourComponent

«interface»
context1 :ExecutionContext

«interface»
context2 :ExecutionContext

+owner

+owner

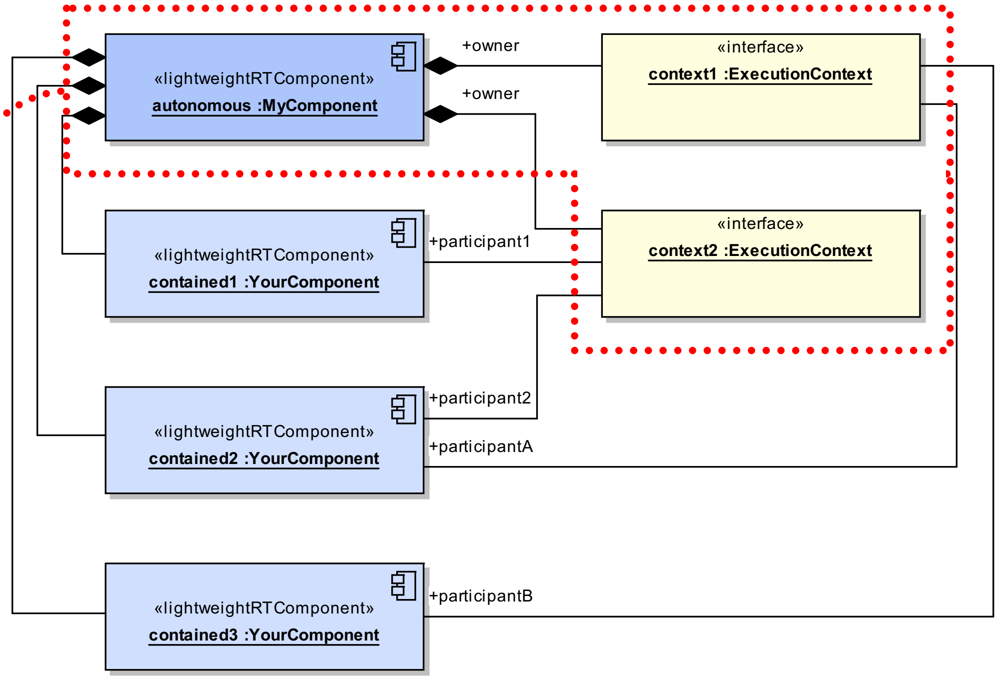
+participant1

+participant2

+participantA

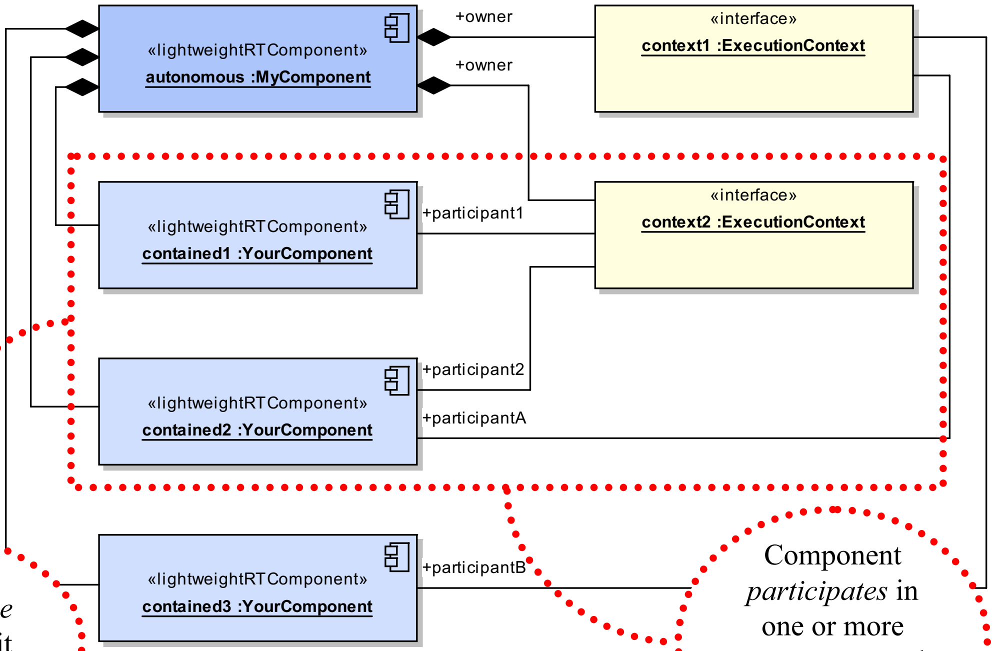
+participantB

# Components & Contexts



*Autonomous* component *owns* execution context(s)

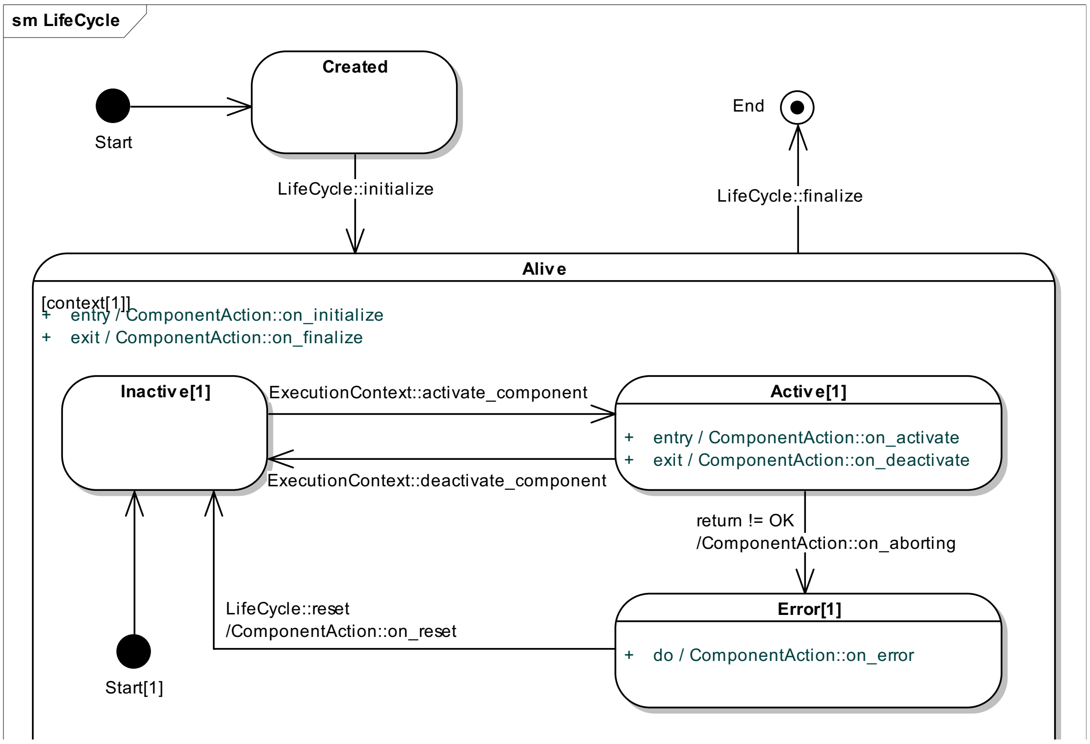
# Components & Contexts



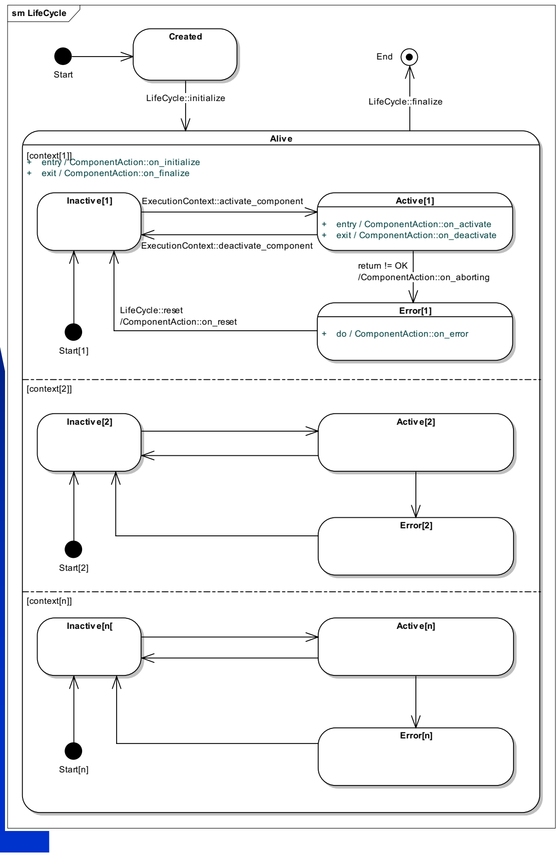
Component is *active* or *inactive* in each context it in which it participates

Component *participates* in one or more contexts owned by containing component(s)

# Component Lifecycle


sm LifeCycle
Created
Start
End
LifeCycle::initialize
LifeCycle::finalize
Alive
[context[1]]
+ entry / ComponentAction::on_initialize
+ exit / ComponentAction::on_finalize
Inactive[1]
ExecutionContext::activate_component
Active[1]
+ entry / ComponentAction::on_activate
+ exit / ComponentAction::on_deactivate
ExecutionContext::deactivate_component
return != OK
/ComponentAction::on_aborting
LifeCycle::reset
/ComponentAction::on_reset
Error[1]
+ do / ComponentAction::on_error
Start[1]

# Component Lifecycle

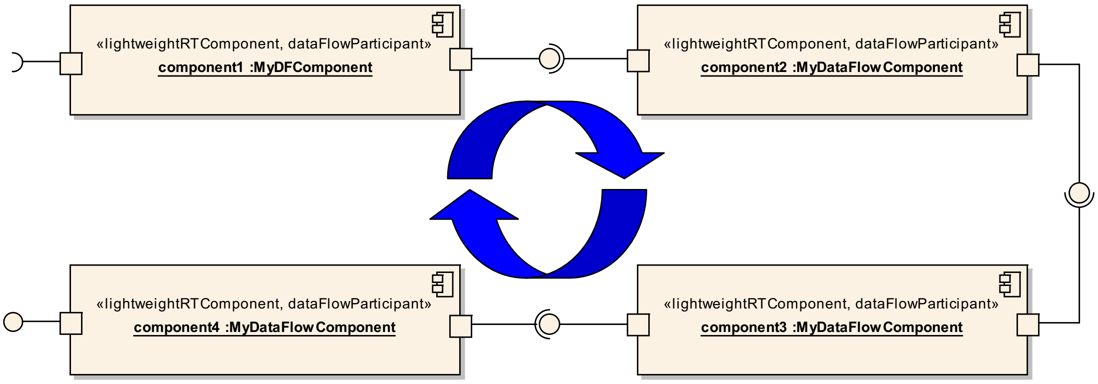

sm LifeCycle
Created
Start
End
LifeCycle::initialize
LifeCycle::finalize
Alive
[context[1]]
+ entry / ComponentAction::on_initialize
+ exit / ComponentAction::on_finalize
Inactive[1]
ExecutionContext::activate_component
Active[1]
+ entry / ComponentAction::on_activate
+ exit / ComponentAction::on_deactivate
ExecutionContext::deactivate_component
return != OK
/ComponentAction::on_aborting
LifeCycle::reset
/ComponentAction::on_reset
Error[1]
+ do / ComponentAction::on_error
Start[1]
[context[2]]
Inactive[2]
Active[2]
Error[2]
Start[2]
[context[n]]
Inactive[n[
Active[n]
Error[n]
Start[n]

- Each execution context represented as parallel region in component lifecycle
- Component can be in different state relative to each execution context
  - *e.g.* active in one, inactive in another

# Execution Semantics Overview

- Common design patterns of robotics applications that components plug into
  - Kinds of execution
    - Data flow
    - Discrete event/stimulus response
  - Multiple modes of operation
- Component declares participation in pattern(s) using stereotype(s)
  - Each execution context has an execution kind
  - Well-known interfaces of participating components are invoked

# Data Flow Execution



«lightweightRTComponent, dataFlowParticipant»
component1 :MyDFComponent

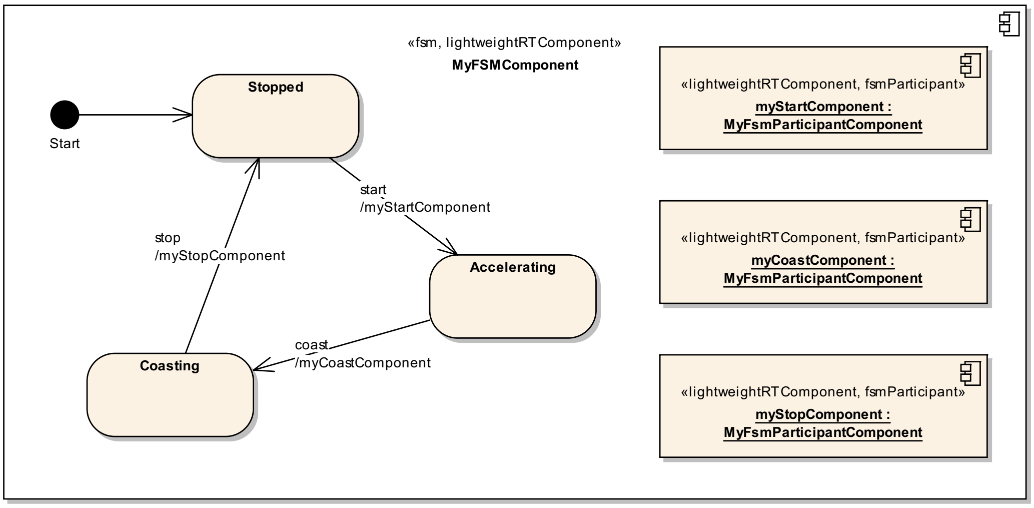
«lightweightRTComponent, dataFlowParticipant»
component2 :MyDataFlowComponent

«lightweightRTComponent, dataFlowParticipant»
component4 :MyDataFlowComponent

«lightweightRTComponent, dataFlowParticipant»
component3 :MyDataFlowComponent

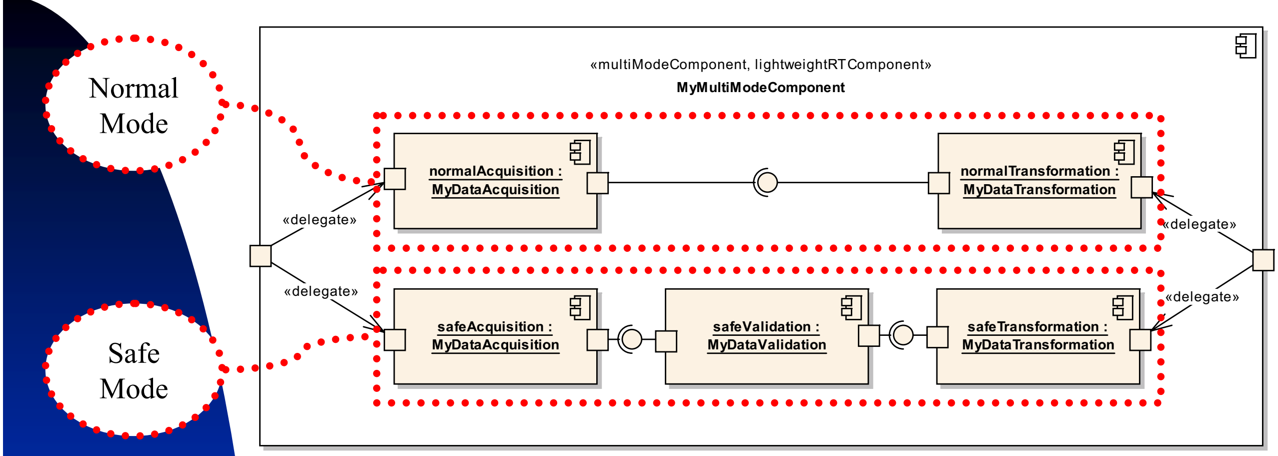
- Periodic, ordered, synchronous invocation of operations on *DataFlowComponentAction* interface
- Execution context of *PERIODIC* kind
- Components marked for participation using *dataFlowParticipant* stereotype
- Execution order determined by *sorting* component instances according to interconnections

# Stimulus Response Execution


«fsm, lightweightRTComponent»
MyFSMComponent
Start
Stopped
start
/myStartComponent
Accelerating
stop
/myStopComponent
coast
/myCoastComponent
Coasting
«lightweightRTComponent, fsmParticipant»
myStartComponent :
MyFsmParticipantComponent
«lightweightRTComponent, fsmParticipant»
myCoastComponent :
MyFsmParticipantComponent
«lightweightRTComponent, fsmParticipant»
myStopComponent :
MyFsmParticipantComponent

- Asynchronous event-driven invocation of operations on *FsmComponentAction* interface
- Execution context of *EVENT_DRIVEN* kind
- Components marked for participation using *fsmParticipant* stereotype

# Modes of Operation



- Each mode is a different implementation of the same functionality
  - *e.g.* Normal mode and Safe mode
  - Only one implementation is active at a time
- Components marked for participation using *multiModeComponent* stereotype

# Introspection Overview

- Basis for interoperability of dynamically distributed components
- Useful interface for analysis tools
- Extends Lightweight RTC with SDO-based API for query and configuration
  - Connect/disconnect components
  - Start/stop execution contexts remotely
  - Get/set properties
  - Query properties of device that component runs on

# Introspection & SDO

- Conforming components extended by *rtComponent* stereotype
  - Components realize all Lightweight RTC interfaces
  - Components realize *SDOPackage::SDO* interface

# Interfaces & Profiles

- Types separated into interfaces and “profiles”
  - Interfaces expose behavioral capabilities of Lightweight RTC elements, *e.g.*:
    - Components
    - Ports
    - ExecutionContexts
  - Profiles are data-only descriptor classes
    - Expose state of underlying element
    - IDL structs
    - Separation allows for more efficient data distribution than remote accessor methods

# Conclusion

- RTC defines domain-specific extensions to a general-purpose component model
  - Behavioral design patterns
  - Introspection of distributed components
- RTC is founded on proven technologies
  - Existing standards
    - UML
    - SDO
    - Lightweight CORBA Component Model
  - Existing proprietary middlewares
    - OpenRTM from AIST
    - Constellation from RTI