| | | | | | | |
|---|---|---|---|---|---|---|
| **Robotics Domain Task Force Final Agenda ver1.2.1** | | | | | | **robotics/2006-12-01** |
| OMG Technical Meeting - **Washington DC**, USA -- December 4-8, 2006 | | | | | | |
| | | **TF/SIG** | | http://robotics.omg.org/ | | |
| | | **Host** | **Joint (Invited)** | **Agenda Item** | **Purpose** | **Room** |
| **Sunday (Dec. 3)** | | | | | | |
| | | | | No business | | |
| **Monday (Dec. 4) WG activity** | | | | | | |
| 9:00 | 10:00 | Robotics | (SDO) | Robotics Steering Committee | Robotics/SDO Joint Meeting Kick-off | Suite 1811, 18th FL |
| 10:00 | 12:00 | Robotics | | **Infrastructure WG**(2h):<br>10:00-12:00 -- RTC FTF meeting<br>- Rick Warren | discussion | Suite 0408, 4th FL |
| | | | | **Profiles WG(2h):** Discussion on profile standardization<br>- Seung-Ik Lee, Bruce Boyes | discussion | Suite 1808, 18th FL |
| | | | | **Robotic Services WG(2h)**:<br>AM 10:00 - 10:15 : Welcome – Roadmap presentation<br>AM 10:15 - 10:45 : Presentation on HRI component Architecture<br>AM 10:45 – 12:00 : Localization RFP : Discuss and identify the business<br>- Olivier Lemaire and Soo-Young Chi | discussion | Suite 1811, 18th FL |
| 12:00 | 13:00 | | | **LUNCH** | | Independence B, Ind. Lv1 |
| 13:00 | 18:00 | | | Architecture Board Plenary | | Kennedy, 3rd FL |
| 13:00 | 17:00 | Robotics | (SBC) | **Infrastructure WG**(4h):<br>13:00-15:00 -- Jerry Bickle's talk, and joint meeting with SBC<br>15:00-16:00 -- Infrastructure WG follow-up meeting<br>- Saehwa Kim, Noriaki Ando, and Rick Warren | discussion | Suite 0408, 4th FL |
| | | | | **Profiles WG(4h):** Discussion on profile standardization<br>- Seung-Ik Lee, Bruce Boyes | discussion | Suite 1808, 18th FL |
| | | | | **Robotic Services WG(4h)**:<br>PM 13:00 – 17:00 : Business case discussion (cont'd) + articulation of the RFP regarding to business case + address eventual technical issues<br>- Olivier Lemaire and Soo-Young Chi | discussion | Suite 1811, 18th FL |
| **Tuesday (Dec. 5) WG activity and Plenary** | | | | | | |
| 9:45 | 10:00 | Robotics | SDO | Joint Plenary Opening | Robotics/SDO joint plenary kick-off | Kennedy, 3rd FL |
| 10:00 | 12:00 | Robotics | | **Robotic Services WG(2h):**<br>AM 10:00 – 11:30 : Review of the updated RFP<br>AM 11:30 – 12:00 : Roadmap review, Homework, Next meeting WG Schedule<br>- Olivier Lemaire and Soo-Young Ch | discussion | Suite 1851, 18th FL |
| | | | (SDO,SBC) | Heterogeneous Network Middleware (10:00-11:00)<br>- Vitaly Li (Kangwon National Univ.)<br>[Jerry's talk is scheduled on Monday afternoon] | presentation and discussion | Kennedy, 3rd FL |
| | | | | RTM on Java/HORB: Eclipse meets RT Components (11:00-12:00)<br>- Satoshi Hirano, Takeshi Ohkawa, Qu Runtao (AIST) | presentation and discussion | |
| 12:00 | 13:00 | | | **LUNCH** | | Independence B, Ind. Lv1 |
| 13:00 | 14:00 | Robotics | | **Special Talk:** IEEE 1588: An Update on the Standard and Its Application<br>- John Eidson (Agilent) | presentation and discussion | Kennedy, 3rd FL |
| 14:00 | 15:00 | Robotics | | **Special Talk:** IEEE 1451 Smart Sensor Interface Standards<br>- Kang Lee (NIST) | presentation and discussion | |
| | | | | **Break (30min)** | | |
| 15:30 | 16:30 | Robotics | SDO | **Invited Talk:** Examples of SDO Implementation<br>- Shigetoshi Sameshima (Hitachi) | presentation and discussion | |
| 16:30 | 17:30 | Robotics | | Anybot Studio<br>- Soon-Hyuk Hong (Samsung Electronics) | presentation and discussion | |
| **Wednesday (Dec. 6) Robotics Plenary** | | | | | | |
| 9:00 | 12:00 | Robotics | (SDO) | WG Reports and Roadmap Discussion<br>(Infrastructure, Robotic Services, Profiles) | reporting and discussion | Kennedy, 3rd FL |
| 12:00 | 14:00 | | | **LUNCH and OMG Plenary** | | Independence B, Ind. Lv1 |
| 14:00 | 15:00 | Robotics | (SDO) | Demonstration of Microsoft Robotic Studio<br>- Bruce Boyes (Systronix) | Demonstration and Informative | Kennedy, 3rd FL |
| | | | | **Break (30min)** | | |
| 15:30 | 16:30 | Robotics | (SDO) | JAUS Information Day Report<br>- Hung Pham (RTI) | Information Exchange | |
| 16:30 | 17:00 | Robotics | SDO | Contact Reports:<br>- Makoto Mizukawa(Shibaura-IT), and Yun-Koo Chung(ETRI) | Information Exchange | |
| 17:00 | 17:30 | Robotics | (SDO) | Publicity SC Report, Next meeting Agenda Discussion | Robotics/SDO joint plenary closing | |
| 17:30 | | | | Adjourn joint plenary meeting | | |
| 17:30 | 18:00 | Robotics | | Robotics WG Co-chairs Planning Session<br>(Agenda for San Diego, Draft report for Friday) | planning for next meeting | |
| 18:00 | 20:00 | | | **OMG Reception** | | Independence B, Ind. Lv1 |
| **Thursday (Dec. 7)** | | | | | | |
| 9:00 | 12:00 | Robotics | | RTC-FTF meeting | discussion | Suite 1808, 18th FL |
| 12:00 | 13:00 | | | **LUNCH** | | ndependence B, Ind. Lv |
| 13:00 | 18:00 | | | Architecture Board Plenary | | Kennedy, 3rd FL |
| | | | | | | |
| 16:00 | 18:00 | MARS | | Agenda Coordinatging Meeting - San Diego TM | planning for next meeting | Tidewater, 2nd FL |
| **Friday** | | | | | | |
| 8:30 | 12:00 | | | AB, DTC, PTC | | Independence A, Ind. Lv.1 |
| 12:00 | 13:00 | | | **LUNCH** | | Tidewater, 2nd FL |
| **Other Meetings of Interest** | | | | | | |
| **Monday** | | | | | | |
| 8:00 | 8:45 | OMG | | New Attendee Orientation | | Prince William, 3rd FL |
| 9:00 | 17:00 | OMG | | Service Oriented Architecture (SOA) Information Day | | Lincoln/Toosevelt, 3rd FL |
| 18:00 | 19:00 | OMG | | New Attendee Reception (by invitation only) | | Tidewater, 2nd FL |
| **Tuesday** | | | | | | |
| 9:00 | 12:00 | OMG | | OMG Architecture Information Day | | Lincoln/Toosevelt, 3rd FL |
| 13:00 | 18:00 | OMG | | DDS Information Day | | Lincoln/Toosevelt, 3rd FL |
| **Wednesday** | | | | | | |
| 8:30 | 17:00 | OMG | | Emerging Standards for Service -Oriented Architectures | | Jeffersohn, 3rd FL |
| **Please get the up-to-date version from http://staff.aist.go.jp/t.kotoku/omg/RoboticsAgenda.pdf** | | | | | | |

# Minuets of the Robotics DTF Plenary

Sep 26-27, 2006
Anaheim, CA, USA
robotics/2006-12-02

**Meeting Highlights**

The 2nd Submission of Robotic Technology Component (RTC) has been recommended in MARS-PTC, AB board, and Technology Committee. And we chartered RTC Finalization Task Force (FTF).

We had four interesting talks; SunSPOT demo – Bruce Boyes(Systronix) and Eric Arseneau (Sun), SysML brief tutorial - Sanford Friendenthal (Lockheed Martin), Robot Ontology - Minsu Jang (ETRI), and Japanese Agriculture Robot - Yoshisada Nagasaka (AFFRC).

Three WGs have active discussions about the topics of potential RFPs.

**List of generated documents**

- robotics/2006-09-01 Final Agenda (Tetsuo Kotoku)
- robotics/2006-09-02 Boston Meeting Minutes [approved] (Hung Pham)
- robotics/2006-09-03 Kickoff Presentation (Tetsuo Kotoku)
- robotics/2006-09-04 RTC 2nd Revised Submission Review [mars/2006-09-18] (Rick Warren)
- robotics/2006-09-05 Robotic Functional Services WG Meeting Schedule (Olivier Lemaire)
- robotics/2006-09-06 Localization (Yeon-Ho Kim)
- robotics/2006-09-07 Localization RFP - DRAFT (Kyuseo Han)
- robotics/2006-09-08 Localization RFP Presentation (Kyuseo Han)
- robotics/2006-09-09 Robotic Profile Presentation (Bruce Boyes)
- robotics/2006-09-10 Steering Committee (Tetsuo Kotoku)
- robotics/2006-09-11 Publicity Report (Masayoshi Yokomachi)
- robotics/2006-09-12 ~~Amendment Robotics-DTF Charter - DRAFT (Hung Pham)~~
- robotics/2006-09-13 Localization RFP Presentation v2 (Kyusen Han)
- robotics/2006-09-14 Wireless Robot Sensors: SunSPOT (Bruce Boyes and Eric Arseneau )
- robotics/2006-09-15 Space Robotics in Past, Current and Future (Hiroshi Ueno)
- robotics/2006-09-16 OMG System Modeling Language Tutorial (Sanford Friedenthal)
- robotics/2006-09-17 Configuration and Deplyment RFP - DRAFT (Rick Warren)
- robotics/2006-09-18 Plenary Opening (Tetsuo Kotoku)
- robotics/2006-09-19 Robot Ontology and Related Research in RTRI (Minsu Jang)
- robotics/2006-09-20 Infrastructure WG Report (Rick Warren)
- robotics/2006-09-21 Robotic Function Services WG Report (Olivier Lemaire)
- robotics/2006-09-22 Robot Device and Data Profile WG Report (Bruce Boyes)
- robotics/2006-09-23 Introduction to RTC (Rick Warren)
- robotics/2006-09-24 Autonomous Systems for Japanese Agriculture in Paddy Field (Yoshisada Nagasaka)
- robotics/2006-09-25 Contact Report (Makoto Mizukawa)
- robotics/2006-09-26 KIRSF- Contact Report (Yun-Koo Chung)
- robotics/2006-09-27 Closing Presentation (Tetsuo Kotoku)
- robotics/2006-09-28 Roadmap for Robotics Activities (Tetsuo Kotoku)
- robotics/2006-09-29 Next Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)
- robotics/2006-09-30 RTC FTF Charter (Rick Warren)
- robotics/2006-09-31 DTC Report Presentation (Tetsuo Kotoku)
- robotics/2006-09-32 Anaheim Meeting Minutes - DRAFT (Hung Pham)

**MINUTES**

***Tuesday, Coronado Suite***

***Attendees: 18***

Rick Warren (RTI)
Masayoshi Yokomachi (NEDO)
Bruce Boyes (Systronix)
Eric Arseneau (SUN)
Joo Chan Sohn (ETRI)
Yun Koo Chung (ETRI)
Dong Hee Choi (KNU)
Vitaly Li (KNU)
Olivier Lemaire (JARA)
Takeshi Sakamoto (Technologic Arts)
Yoshisada Nagasaka (NARC)
Takashi Suehiro (AIST)
Makoto Mizukawa (SIT)
Seiichi Shin (UEC)
Tetsuo Kotoku (AIST)
Noriaki Ando (AIST)
Hung Pham (RTI)
Sanford Friedenthal (LMC)

**"Introduction to SysML" – Sanford Friendenthal (LMC)**

- presented an overview of SysML and talked about its applicability to robotics
- fielded questions about relationships among various diagrams
- particular interest was expressed in the parametric model diagrams

***Wednesday, Balboa Suite***

***Attendees: 22***

Makoto Mizukawa (SIT)
Seiichi Shin (UEC)
Claude Baudoin (Schlumberger)
Roy Bell (Raytheon)
Rick Warren (RTI)
Joo Chan Sohn (ETRI)
Hung Pham (RTI)
Kyuseo Han (ETRI)
Incheol Jeong (ETRI)
Dong Hee Choi (KNU)
Vitaly Li (KNU)
Yun Koo Chung (ETRI)
Su Young Chi (ETRI)
Olivier Lemaire (JARA)
Masayoshi Yokomachi (NEDO)
Bruce Boyes (Systronix)
Tetsuo Kotoku (AIST)
Takeshi Sakamoto (Technologic Arts)
Minsu Jang (ETRI)
Yoshisada Nagasaka (NARC)
Dave Stringer (Borland)
Noriaki Ando (AIST)

***Proceedings***
Meeting called to order at 8:56am (Toku, AIST)
Review of the Agenda (Toku, AIST)

**"Robot Ontology and Related Research in ETRI" – Minsu Jang (ETRI)**
- Described ontology and its applicability to robotics
  * Using vocabulary to model data, like RDB
  * However, provides constructs for specifying mode complex relationships between data
  * Speaker conclusion: ontology provides well-established mechanism for interoperability in "broad sense"
- Described ontology-related research at ETRI, *e.g.*,
  * service/content selection
  * service/content adaptation
- Considerations
  * how about addressing semantic requirements on profiles? how about relevant use-cases or requirements for RFP?

***WG Reports***
Infrastructure WG Report (Warren, RTI)
- Reviewed existing D&C standards
  * SBC, CCM, XMI
- Moving forward, need to define the scope of a potential D&C RFP
  * delay RFP process until further information can be exchanged with CORBA, SBC, *etc.*

Robotic Functional Services WG report (Lemaire, JARA)
- Roadmap remains on track
- First draft of RFP was written
  * issues regarding scope and perspective, i.e., User's *vs.* Developer's point of view
  * more details of location calculation module is necessary
  * need to define more what is expected for each interface

Robotic Devices and Data Profiles WG report (Boyes, ETRI)
- Demonstration of SunSpot given
- Upcoming presentations proposed
- Work plan
  * review data format of application standards
  * relation to localization sensors specification
  * possible: consider semantic requirements in Profile WG?
  * review draft RFP which has been created through mailing list collaboration
- Draft RFP tentative for Wash DC, perhaps combine these 2 topics
  * typical device abstract interfaces and hierarchies
  * hardware-level resources: define resource profiles

Review of the Boston Minutes (Toku)
- AIST motioned to accept
- JARA seconded
- SIT suggested white ballot

***RTC submission update***
(Warren, RTI)
- Provided overview of the RTC
  * had been recommended for adoption by MARS at previous tech meeting
  * AB had raised issues which needed to be addressed by this technical meeting

**“Autonomous systems for Japanese Agriculture in Paddy Field,” Yoshisada (NARC)**
- Developing autonomous system to transplant rice in paddy fields
- Prototyped rice transplanter with modular sensing package
- What standards do we need?
  * ISO 11783 has not defined protocol for autonomous operations yet
  * Need standard comm protocol to share sensors and controllers among each farm operating robots

***Contact reports***
ISO TC184 – SC2 (Mizukawa, SIT)
- Next meeting in DC Jun 7-8, 2007
  * PT Robots in personal care & Advisory Group (AG) Service robots planning to meet Jun 4-6
  * International Robots and Vision Show will take place in Chicago in Jun 11-15

IROS2006 Workshop (Mizukawa, SIT)
- Organizers: Kotoku (AIST), Chung (ETRI), and Mizukawa (SIT)
- Scheduled Oct 10, 2006.

OS059 Robot Technology System Integration Oct 10-20 (Mizukawa, SIT)
- Organizers: Chung (ETRI) and Mizukawa (SIT)
- Scheduled Oct 18-19, 2006

KIRSF contact report (Chung, ETRI)
- RUPI (Robot Unified Platform Initiative) standardization launched on Jul 4, 06.
- URC (Ubiquitous Robotic Companion) robots (~650) will be distributed in field tests beginning Oct

***Publicity report***
Robotics DTF brochure (Yokomachi NEDO)
- Showed flier to group
- Requested pics of robots to put on flier
- Requested feedback on flier
- Targeting 3 wks before DC

***New Business***
Next meeting agenda for Dec 4-8 in DC (Toku, AIST)
- Monday
  * Steering committee (Mon morning)
  * WG activities (3WG in parallel)
- Tuesday
  * WG activities, joint activity with other SG
- Wednesday
  * Plenary
    - WG reports
    - Guest and member presentation
    - Contact reports
    - DTC report

Meeting was adjourned at 4:52 pm

Prepared and submitted by Hung Pham (RTI).

# Robotics Domain Task Force
## Steering Committee Meeting

December 4, 2006

Arlington, VA, USA

Hyatt Regency Cristal City Hotel





# Anaheim Meeting Summary

- 2nd Submission of Robotic Technology Component (RTC) has been recommended in MARS, AB, and TC. And we chartered RTC Finalization Task Force (FTF)
- We had four interesting talks;
  - SunSPOT demo – Bruce Boyes(Systronix) and Eric Arseneau (Sun)
  - SysML brief tutorial - Sanford Friendenthal (Lockheed Martin)
  - Robot Ontology - Minsu Jang (ETRI)
  - Japanese Agriculture Robot - Yoshisada Nagasaka (AFFRC)
- Three WGs have active discussions about the topics of potential RFPs.

# Agenda

- Agenda Review
- Minutes and Minutes Taker
- Publicity
- Re-Charter
- Roadmap Discussion
- Next meeting Schedule





# Agenda Review

**Mon(Dec.4):**
**Steering Committee**
**Services WG, Profiles WG, Infrastructure WG**
**Tue(Dec.5):**
**Service WG (morning)**
**Task Force Plenary**
**Wed(Dec.6):**
**Task Force Plenary**
**Thu(Dec.7):**
**WG activity follow-up?**

Please check our final agenda.

AIST

# Minutes and Minutes Taker

- 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 Meeting
  - Hung Pham (RTI)
  - Yun-Koo Chung (ETRI)

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



AIST

# Publicity Activities

- Robotics Wiki is available
  http://portals.omg.org/robotics

- 4 page fly sheet

- Information Day: Brussels
  (Meeting with Marketing Staff on Tuesday lunchtime)

4-page fly sheet is authorized in DC

# Re-Charter of Robotics-DTF

Proposal by Hung Pham

– Make attractive expression

– To understand easily by outsiders

Revised version will be proposed at the plenary



# 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)

## Roadmap Discussion

- Confirm the process of working items
- Create new items
  ( we need volunteers)



## Next Meeting Agenda

March 26-30 (San Diego, CA, USA)

**Monday:**

**Steering Committee (Morning)**
**WG activity [3WG in parallel]**

**Tuesday - Wednesday :**

**Robotics-DTF Plenary Meeting**
•Guest and Member Presentation
•Contact reports
•DTC report - Draft

**Thursday:**

**WG activity (optional)**
**Information Exchange with other sub-groups**

# Next Meeting Agenda

- Make a rough agenda at the previous meeting. (rough sketch)
- Agenda planning session on Wednesday after plenary session
- 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!**

# Roadmap for Robotics Activities

robotics/2006-12-04

| Item | Status | Anaheim Sep-2006 | DC Dec-2006 | San Diego Mar-2007 | Brussels Jun-2007 | Jacksonville Sep-2007 | Burlingame Dec-2007 | TBA Mar-2008 | POC / Comment |
|---|---|---|---|---|---|---|---|---|---|
| **RTC Finalization Task Force** | In Process | charter | | | Comment Due 7/2 | | Report Deadline 10/5 | | Rick(RTI) |
| SDO model for xxx Domain | Planned | | discussion | draft RFP | | | | | TBD |
| Robot Technology Components RFP (SDO model for robotics domain) | done Sep-2006 | adoption | Adopted Specification ptc/2006-11-07 | | | | | | Rick(RTI) and Noriaki(AIST) |
| Charter on Robotics WG in SDO | done Oct-2004 | | | | | | | | Kotoku(AIST), Mizukawa(Shibaura-IT) |
| **Flyer of Robotics-DTF [Publicity Sub-Committee]** | In Process | discussion | review Draft | issue ver.1.0 | | | | | Abheek(ADA Software) |
| **Localization Service RFP [Services WG]** | In Process | discussion | draft RFP | review RFP | RFP | | Initial Submittion | Pre-review | |
| **User Identification RFP [Services WG]** | Planned | | discussion | discussion | RFP Outline | review RFP | RFP | | |
| **Programmers API: Typical device abstract interfaces and hierarchies RFP [Profile WG]** | In Process | Topic discussion | RFP Outline | review RFP | RFP | | Initial Submittion | Pre-review | |
| **Hardware-level Resources: define resource profiles RFP [Profile WG]** | In Process | Topic discussion | RFP Outline | review RFP | RFP | | Initial Submittion | Pre-review | |
| **Deplyment and Configuration RFP [Infrastructure WG]** | Planned | discussion | discussion | RFP Outline | review RFP | RFP | | | |
| etc… | Future | | | | | | | | to be discussed |
| **Robotics Information Day [Technology Showcase]** | Planned | | | | Info. Day | | Info. Day | | Yokomachi(NEDO), Kotoku(AIST) |
| Robotic Systems RFI [Robotics: Initial Survey] | done Apr-2006 | | | | | | | | Lemaire, Chung, Lee, Mizukawa, Kotoku |
| Charter on WGs [Service, Profile, Infrastructure] | done Apr-2006 | | | | | | | | Lemaire(JARA), Chi(ETRI), Bruce(Systronix), Lee(ETRI), Rick(RTI), Ando(AIST) |
| Charter on Robotics TF | done Dec-2005 | | | | | | | | Kotoku(AIST), |
| Charter on Robotics SIG | done Feb-2005 | | | | | | | | Kotoku(AIST), Mizukawa(Shibaura-IT) |

**Robotics Domain Task Force Preliminary Agenda -DRAFT- ver0.0.6 robotics/2006-12-05**

OMG Technical Meeting - **San Diego, CA**, USA -- March 26-30, 2007

http://robotics.omg.org/

| | | TF/SIG | | | | |
|---|---|---|---|---|---|---|
| | | Host | Joint (Invited) | Agenda Item | Purpose | Room |
| **Sunday (Mar. 25)** | | | | | | |
| | | | | No business | | |
| **Monday (Mar. 26) WG activity** | | | | | | |
| 9:00 | 10:00 | Robotics | (SDO) | Robotics Steering Committee | Robotics/SDO Joint Meeting Kick-off | |
| 10:00 | 12:00 | Robotics | | **Infrastructure WG**(2h):<br>- Saehwa Kim, Noriaki Ando, and Rick Warren | discussion | |
| | | | | **Profile WG(2h):**<br>- Seung-Ik Lee, Bruce Boyes | discussion | |
| | | | | **Robotic Services WG(2h)**:<br>- Olivier Lemaire and Soo-Young Chi | discussion | |
| 12:00 | 13:00 | **LUNCH** | | | | |
| 13:00 | 18:00 | | | Architecture Board Plenary | | |
| 13:00 | 17:00 | Robotics | | **Infrastructure WG**(4h):<br>- Saehwa Kim, Noriaki Ando, and Rick Warren | discussion | |
| | | | | **Profile WG(4h):** Discussion on profile standardization<br>- Seung-Ik Lee, Bruce Boyes | discussion | |
| | | | | **Robotic Services WG(4h)**:<br>- Olivier Lemaire and Soo-Young Chi | discussion | |
| **Tuesday (Mar. 27) Robotics Plenary** | | | | | | |
| 9:00 | 12:00 | Robotics | | **Infrastructure WG**(3h):<br>- Saehwa Kim, Noriaki Ando, and Rick Warren | discussion | |
| | | | | **Profile WG(3h):**<br>- Seung-Ik Lee, Bruce Boyes | discussion | |
| | | | | **Robotic Services WG(3h)**:<br>- Olivier Lemaire and Soo-Young Chi | discussion | |
| 12:00 | 13:00 | **LUNCH** | | | | |
| 13:00 | 13:15 | Robotics | | Joint Plenary Opening | Robotics/SDO joint plenary kick-off | |
| 13:15 | 14:15 | Robotics | | **Demonstration of Microsoft Robotic Studio**<br>**- Bruce Boyes (Systronix)** | Demonstration and Informative | |
| 14:15 | 15:15 | Robotics | | RFP review? | 1st review | |
| | | | | **Break (15min)** | | |
| 15:30 | 16:30 | Robotics | | RFP review? | 1st review | |
| 16:30 | 17:30 | Robotics | | RFP review? | 1st review | |
| 17:00 | 18:00 | OMG | | The Revision and Finalisation Task Force Chairs' Tutorial | discussion | |
| **Wednesday (Mar. 28) Robotics Plenary** | | | | | | |
| 9:00 | 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** | | | | |
| 14:00 | 15:00 | Robotics | (SDO) | **Special Talk: "Introduction to RoSta activities"**<br>**- Erwin Prassler (GPS Gesellschaft für Produktionssysteme GmbH)** | presentation and discussion | |
| 15:00 | 15:45 | Robotics | (SDO) | **Special Talk: "Open Robot Controller Research and Its Standardization in China"** | presentation and discussion | |
| | | | | **Break (15min)** | | |
| 16:00 | 16:45 | Robotics | | **Introduction of URC Network Robot System in ongoing pilot business with its standardization**<br>**- Hyun Kim (ETRI)** | presentation and discussion | |
| 16:45 | 17:15 | Robotics | SDO | Contact Reports:<br>- Makoto Mizukawa(Shibaura-IT), and Yun-Koo Chung(ETRI) | Information Exchange | |
| 17:15 | 17:30 | Robotics | (SDO) | Publicity SC Report, Next meeting Agenda Discussion | Robotics/SDO joint plenary closing | |
| 17:30 | | | | Adjourn joint plenary meeting | | |
| 17:30 | 18:00 | Robotics | | Robotics WG Co-chairs Planning Session<br>(Agenda for Brussels, Draft report for Friday | planning for next meeting | |
| 18:00 | 20:00 | **OMG Reception** | | | | |
| **Thursday** | | | | | | |
| 9:00 | 12:00 | RTC-FTF | | RTC-FTF meeting (3h) | | |
| 12:00 | 13:00 | **LUNCH** | | | | |
| 13:00 | 18:00 | | | Architecture Board Plenary | | |
| | | | | | | |
| 17:00 | 18:00 | MARS | | Agenda Coordinatging Meeting - Brussels TM | planning for next meeting | |
| **Friday** | | | | | | |
| 8:30 | 12:00 | | | AB, DTC, PTC | | |
| 12:00 | 13:00 | **LUNCH** | | | | |

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

Productivity Tools & Middleware

PRISMTECH

# Software Radio Specification Overview

Jerry Bickle
Chief Scientist
PrismTech

2006 OMG Technical Conference
Washington, D.C.
December 4th , 2006

Spectra Software Defined Radio Products

## Agenda

- Software Radio Spec
  - Profile
  - Platform Specific Technologies
- Component Framework Profile
- Component Framework Flexibility and Optimizations
- Conformance

## Software Radio Specification

- To define a common language for building Software Radios for commercial and military that can be extended and/or constrained, and transformed and implemented in any technology.
- To enable the use of Model Driven Architecture (MDA) and Model Driven Development (MDD) technologies for developing radio systems to address today's problems:
  - General Purpose programming languages and tools have not kept pace with the growth of platform complexity[1]
  - Families of systems are the order of the day
  - Systems and requirements have become more complex
  - Exchange information with other tools in a standard format.

[1] *Model Driven Engineering, Douglas C. Schmidt, IEEE Computer Magazine, February 2006*



## Software Radio Specification

- Software Radio Specification (DTC/2006-04-17 & 18) and Software Radio Profile.xml (DTC/2006-10-02) is broken into multiple volumes and profiles for:
  - Better Understandability of its elements
  - To allow for other domains such as Robotics to use these elements as appropriate

Note: the Information being presented represents the Revision Task Force (RTF) changes.

## Software Radio Specification

- Component Framework Specification Volume (DTC/2006-04-04 & 19)
  - A language to describe and required to model a waveform application, and platform infrastructure and services.
  - Component Framework Profile.xml (DTC/2006-04-09)
    - Application and Device Components
      - *Application Components*
      - *Base Types*
      - *Device Components*
      - *Interface and Port Stereotypes*
      - *Properties*
      - *Resource Components*
    - Infrastructure
      - *Deployment*
        - *Artifacts*
        - *Applications Deployment*
      - *Platform Management*
        - *Device Management*
        - *Domain Event Channels*
        - *Domain Management*
      - *Services*
        - *File Components*
        - *Service Components*



## Software Radio Specification

- Communication Channel and Equipment Specification Volume (DTC/2006-04-05 & 20)
  - A language to describe and required to model a specific hardware platform (Radio Set, Channels, Comm. Equipment) upon which applications execute
  - Communication Channel Profile.xml (DTC-2006-04-10)
    - Communication Channel Stereotypes
    - Communication Channel Facilities
      - *Physical layer Facilities*
        - *IO Facilities*
        - *Modem Facilities*
        - *RF/IF Facilities*
      - *Radio Set Facilities*
    - Communication Equipment
    - Radio Management

# Software Radio Specification

- Software Radio Facilities PIM
  - Component Framework Specification Volume
    - Generic interfaces and types for deploying and managing application and platform service components.
  - Common and Data Link Layer Facilities Specification Volume(DTC/2006-04-08 & 23)
    - Defines related data and control PIM interfaces that can be used to define a waveform or platform component.
    - Based on the “Extended” OSI Model, ISO 7498-1: Open System Interconnection – Basic Reference Model
    - Product of a survey of existing specs such as: 3GPP, DLPI, GLoMo, OBSAI, CPRI, 802.x, X.200e
    - Common and Data Link Layer Facilities.xml (DTC/2006-04-11)
  - Communication Channel and Equipment Specification Volume
    - Physical Layer Facilities PIM
      - *Modem Facilities*
      - *RF Facilities*
      - *IO Facilities: SerialIO and Audio Device Components*



# Common and Data Link Layer Facilities PIM

- Common Radio Facilities
  - Provides common service definitions that are applicable for all applications (waveforms or radio control)
    - OMG Lightweight Services (log, event, naming, etc.)
- Common Layer Facilities
  - Provides interfaces that cross cut through facilities that correlate to layers. These interfaces can be viewed as building blocks for components that realize and/or use multiple interfaces.
  - Protocol Data Unit, Error Control, Flow Control, Measurement, Quality of Service, and Stream Facilities
- Data Link Facilities
  - Link Layer Control (LLC) facilities. LLC layer provides facilities to upper layers, for management of communication links between two or more radio sets.
  - Data Link Layer (Connectionless, Ack ConnectionLess, Connection), and Medium Access Control Facilities

## Software Radio Profile

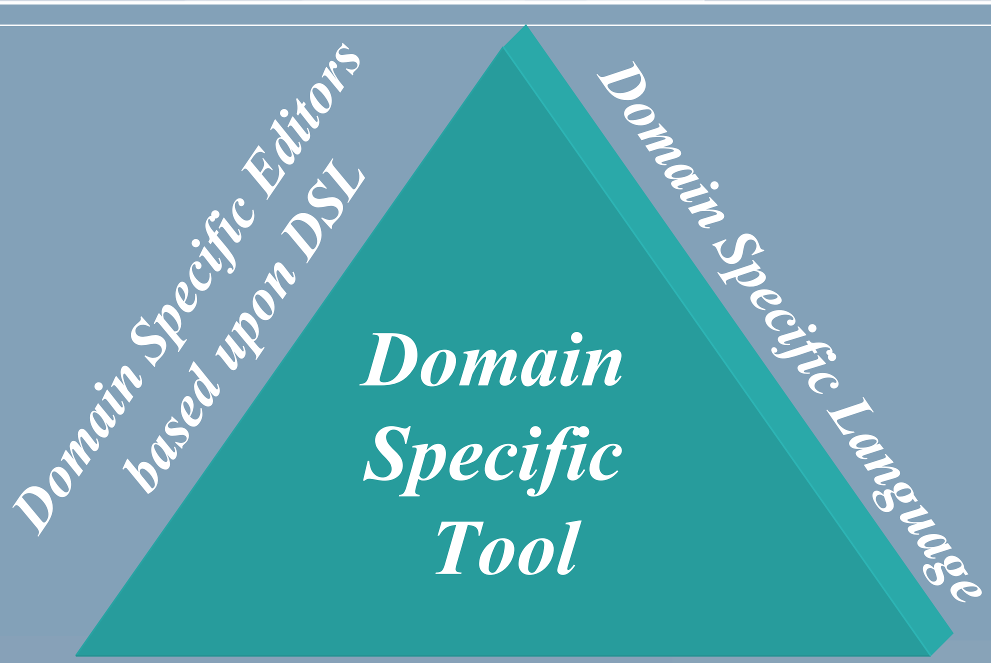
- UML Profile mechanism is used to specific the Software Radio Profile (Domain Specific Language (DSL)) since UML already has existing elements such as:
  - Artifact
  - Component
  - Device
  - Interface
  - Port
  - Property



## Software Radio Profile

- UML Stereotypes were used to extend the UML Elements with specific semantics and constraints.
  - Object Constraint Language is used to express constraints where practical
- The profiles also contains M1 level elements in model library packages
  - These are mostly interfaces that the component definitions are constrained to.
- The profile is captured in XMI, which allows for GDSL tool to exchange information in a standard format.
  - Work that is being completed in the RTF

## Software Radio MDD Tool Features



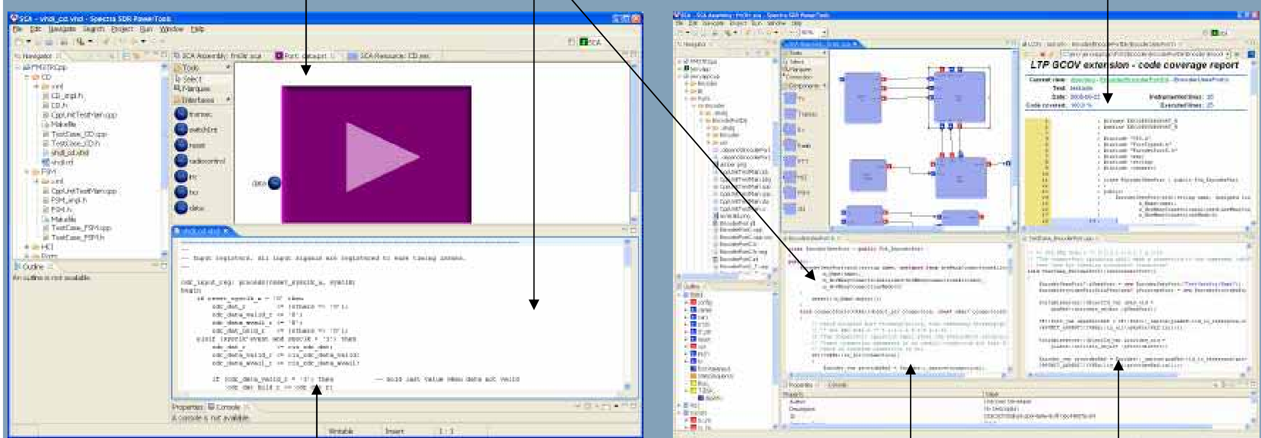

*Domain Specific PSM Generators based upon Transformation Rules*



## Domain Specific Generators

*Translate from declarative to imperative*

*Code Coverage*



*VHDL*

*C++*

*Test Cases*

## Software Radio Normative XMI Files

- Software Radio Profile.xml
  - Component Framework Profile.xml
  - Communication Channel Profile.xml
  - Common and Data Link Layer Facilities.xml

Dependencies

- Communicaiton Channel Profile has dependencies to Component Framework, and Common and Data Link Layer Facilities.
- Common and Data Link Layer Facilities has dependencies to Component Framework



## Software Radio Specification - Platform Specific Technologies

- At this time only CORBA interfaces, XML descriptors and POSIX profiles are defined
  - These definitions are non-normative.
  - Component Document Type Definitions Specification Volume (DTC-2006-04-07 & 22)
    - XML DTD Files (DTC/2006-04-13)
  - POSIX Profiles Specification Volume (DTC-2006-04-06 & 21)
    - Two Profiles are defined which are a subset of the Real-time Controller System Profile (PSE52) Standardized Application Environment Profile - POSIX® Realtime Application Support (AEP), IEEE Std 1003.13-2003
      - *The application environment profile (AEP), which is for constrained embedded general purpose processing, is the preferred profile for embedded processing and its utilization is encouraged for all processing environments.*
      - *The lightweight application environment profile (LwAEP) is more constrained than the AEP and is targeted towards environments with limited computing support such as embedded processors like Digital Signal Processors (DSPs) and micro-controllers.*
- The normative is captured in the XMI SWRadio profiles and Facilities.
- Transformations rules are specified for PIM to PSM in the specification
- Other PSMs could be defined along with there transformation rules

# Software Radio PSM – CORBA Modules

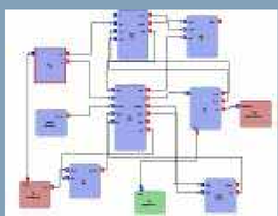
- UML Interfaces map to CORBA IDL
  - The Component Framework Profile interfaces map to the CF CORBA module
    - CF
      - *StandardEvent, PortTypes, FileServices*
  - The Software Radio PIM Facilities map to the DfSWRadio CORBA module.
    - DfSWRadio
      - *CommonLayer, DataLinkLayer, CommonRadio, PhysicalLayer, RadioControl*
  - CF (DTC/2006-04-16) and DfSWRadio (DTC/2006-04-14) IDL files has been broken apart into multiple files to reduce foot print unneeded client and skeleton code.



# SDR MDD

***Model Driven Development!***

*Platform Independent Model created using profile*



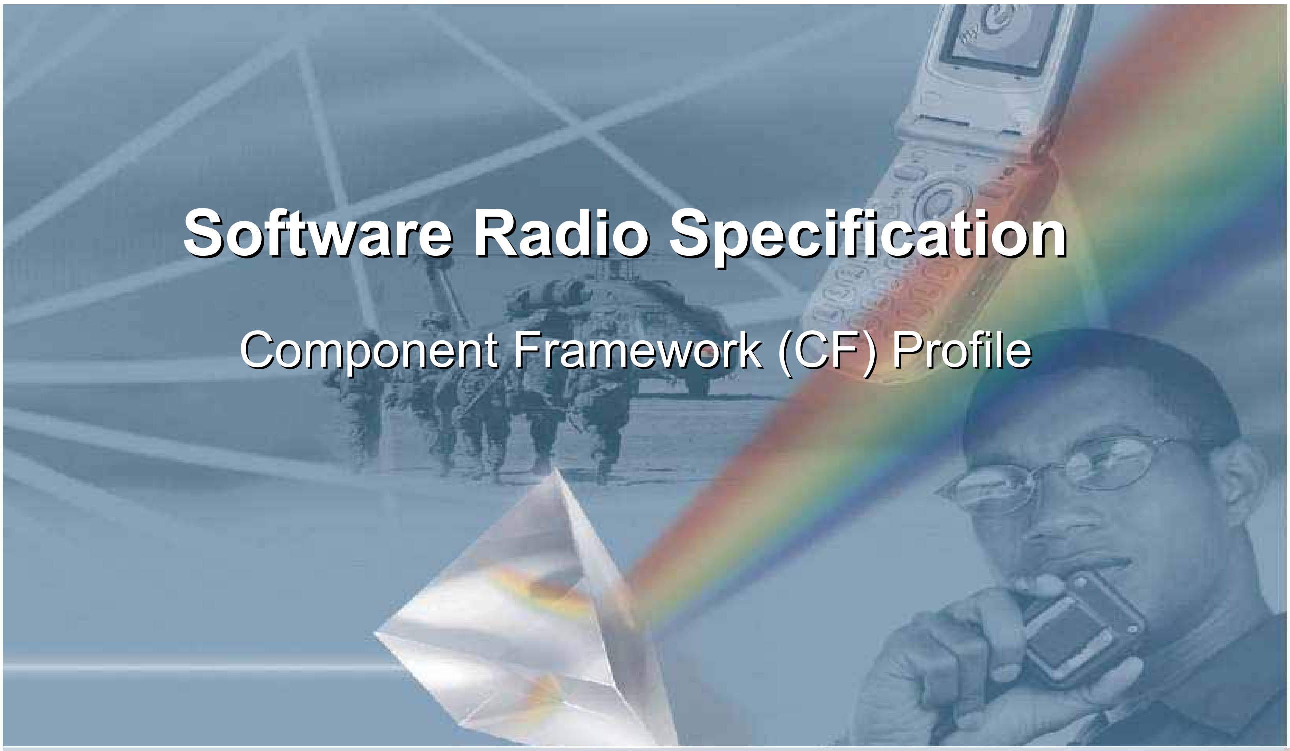
***Transform the Model directly into what you need***

*PSMs*

- SDR Source Code
- SDR Descriptor Files
- SDR Policy Files
- SDR Design Docs
- SDR Compliance Tests

Productivity Tools & Middleware

PRISMTECH



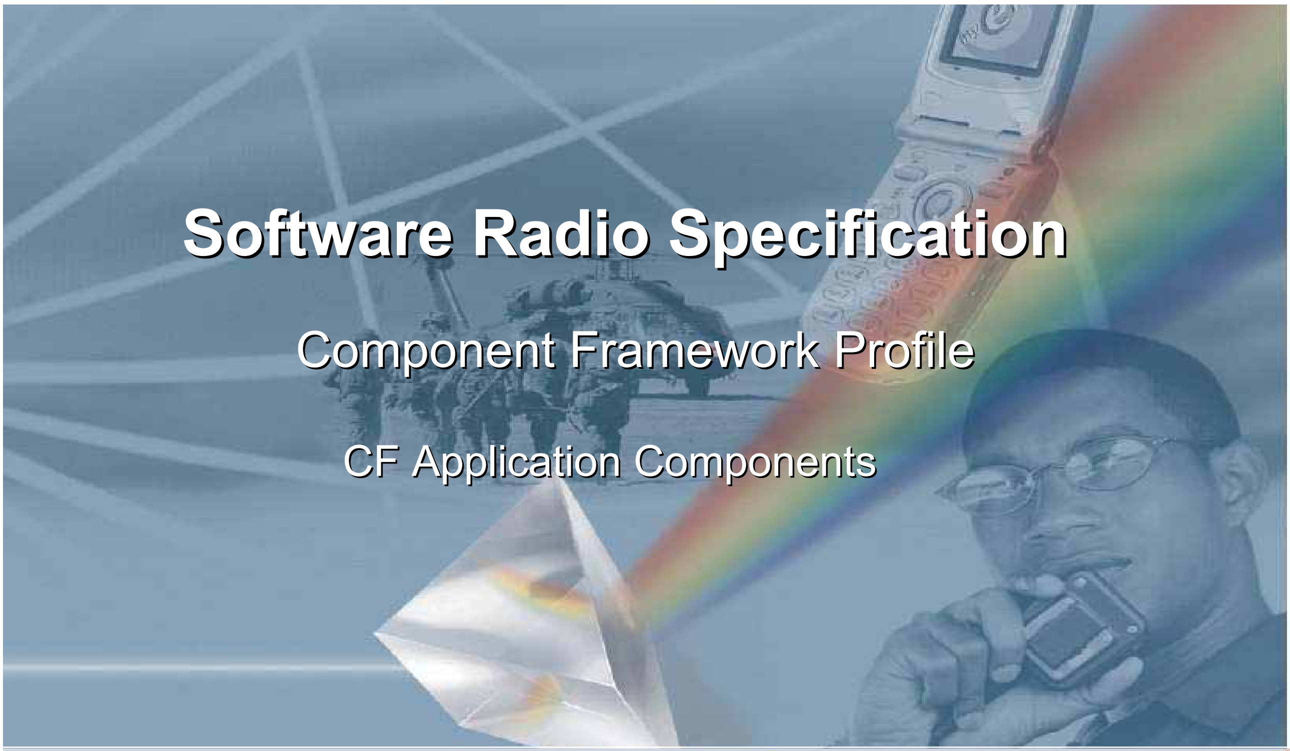
# Software Radio Specification

## Component Framework (CF) Profile

Spectra Software Defined Radio Products

## Component Framework Profile

- CF Application Components
- CF Device Components



# Software Radio Specification

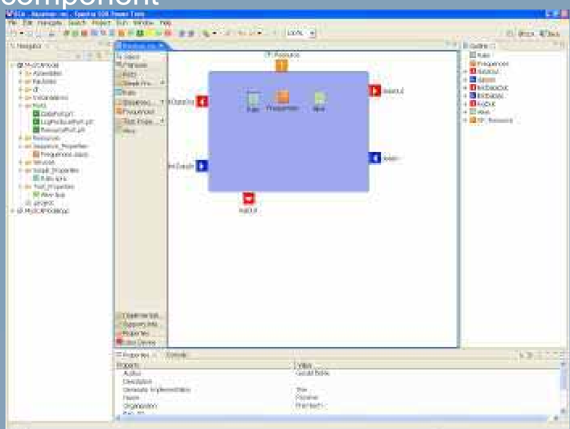
## Component Framework Profile

### CF Application Components

Spectra Software Defined Radio Products

## Component Overview

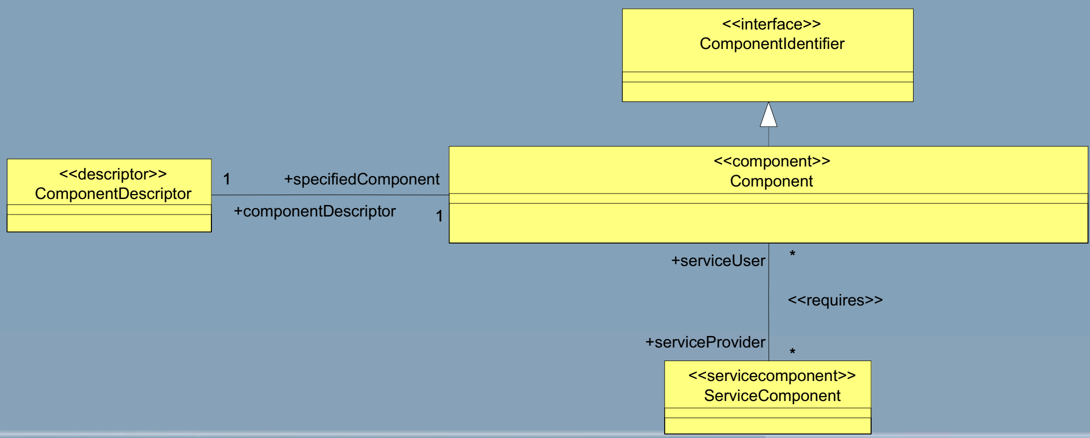
- Component
  - Can realize interfaces, Support Type Interfaces
  - Interface Properties
  - Ports
    - Provides Port - provides an interface implementation
    - Uses (or Requires) Port - uses an interface that is implemented by another component



# CF Component Definition

- CF Component
  - Extends the UML Component by having a unique identifier

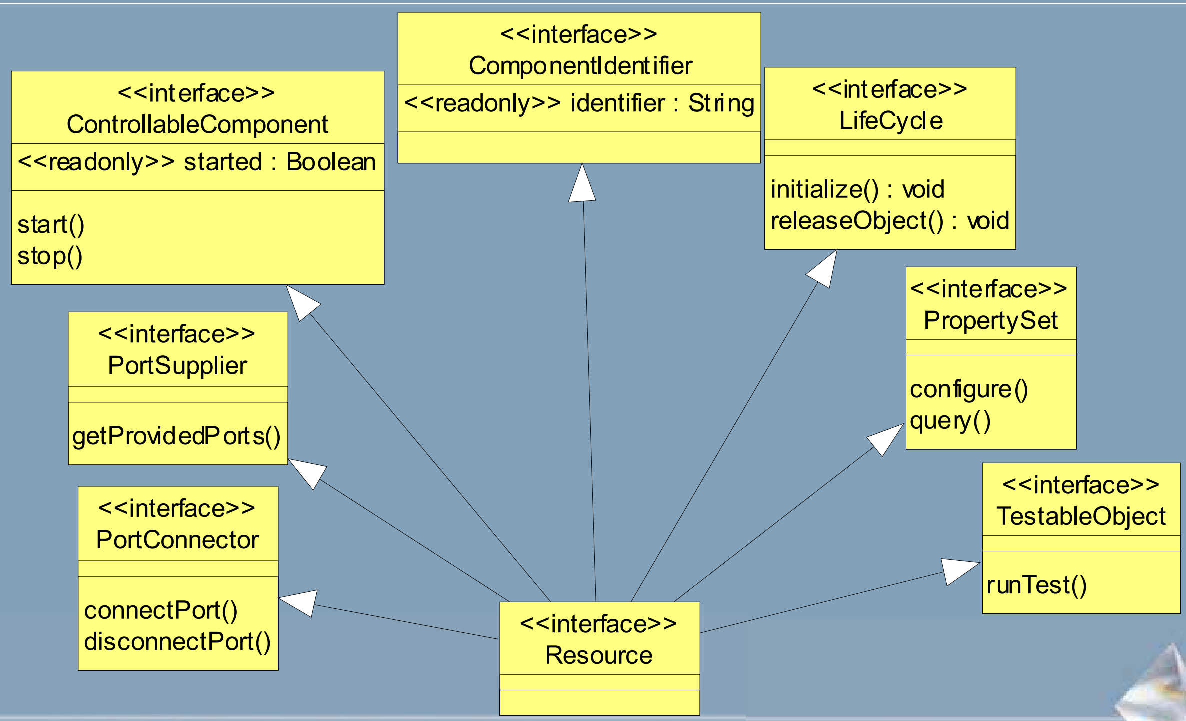





# CF Component Base Interfaces

- A set of ***minimal*** generic interfaces for component management that not only support deployment behavior
  - Lifecycle – life cycle management for a component
  - PropertySet – generic configure and query operations that are based upon primitive types
  - TestableObject – generic interface for testing a component (e.g., BIT)
  - PortSupplier – retrieval of the provided interfaces for a component
  - PortConnector – the management of connection behavior for a component
  - ControllableComponent – overall control of the component
  - ComponentIdentifier – contains the identifier of the component
- The interfaces support
  - Reconfiguration
  - Control
- Resource interface – specialization of all the generic interfaces
  - Provides a complete generic interface for component management and control

# Resource Interface

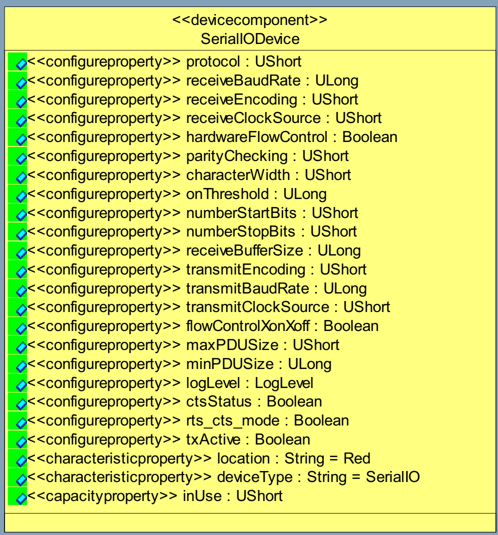


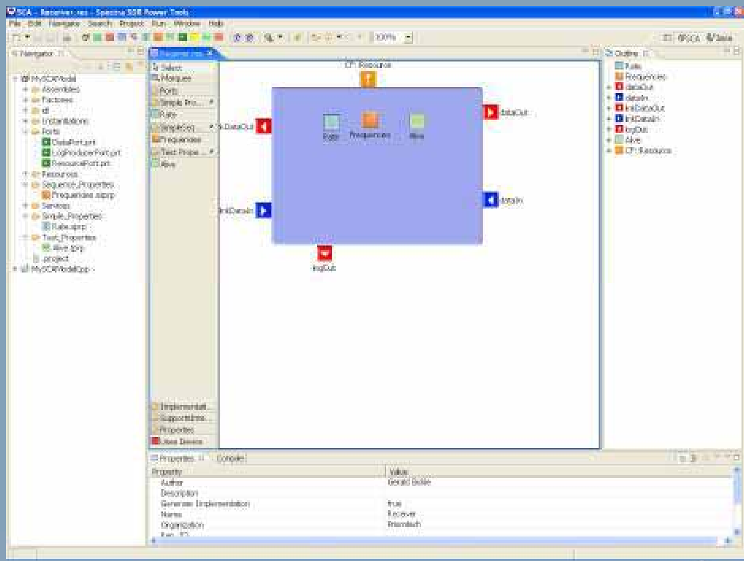



# CF Component Interface Properties

- All properties are based upon primitive types
  - Supplement information beyond UML (UML Tags), identifier, range, units of measure, enumeration literals, etc.
- Property Stereotypes
  - Configure and Query
    - Simple – a property that is a primitive (e.g., octet, boolean, char, string, float, unsigned and signed integers (16, 32, 64 bits), etc.). Profile added more primitive types beyond UML.
    - Simple Sequence – a sequence of simple values of the same primitive type.
    - Struct – a collection of simple properties
    - Struct Sequence – a sequence of struct values of the same struct type.
  - Test
    - Describes a test that contains a set of simple test inputs and simple test outputs
- Each software radio component definition determines its set of properties and interfaces, none mandated by the specification.

# CF Component Properties using GDSL Tool Illustration



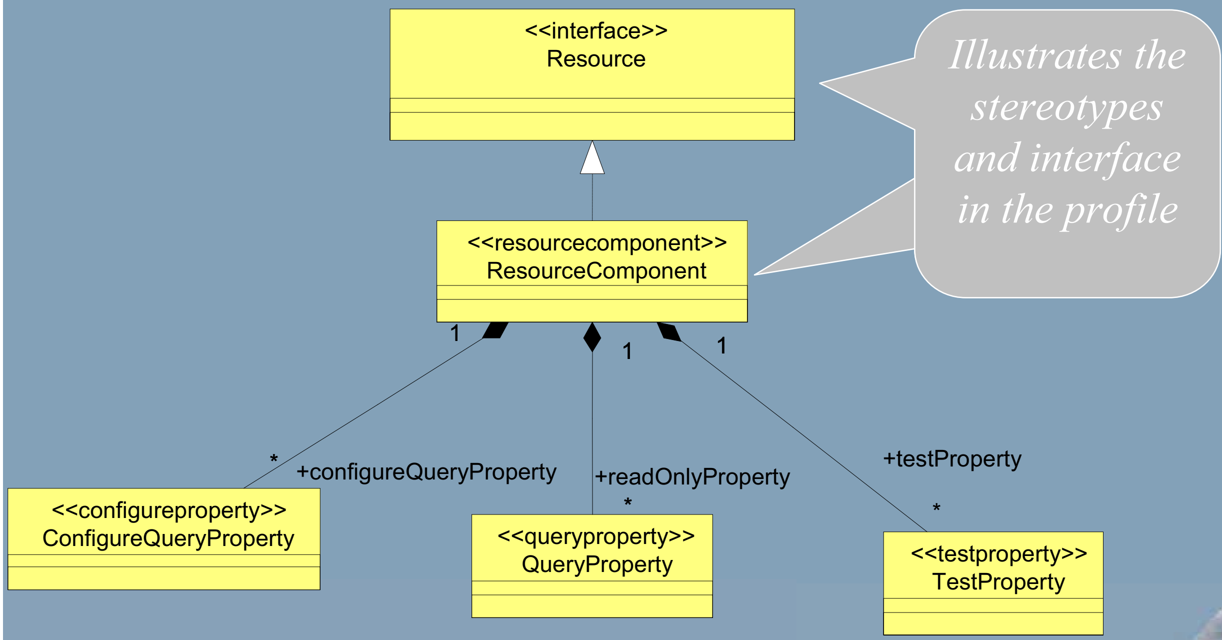





# Resource Component

- A type of CF Component that realizes the Resource interface
  - Supports the Resource interface directly
- Properties
  - Configuration (read and write)
  - Query (read only)
  - Test
- Ports
  - Flow Control
  - Status
  - Data
  - Command and Control

# ResourceComponent Illustration






# Resource Factory Component

- Realizes the ResourceFactory Interface
  - Used to create/tear down Resource Components
- Behavior can be homogeneous (same Resource Component type) or heterogeneous (different Resource Component types)
- Modeled after the Factory Design Pattern
- Used for collocation of Components within the same process space.
- Optional
  - Applications are not required to have these types of components.

## Application Component Definitions

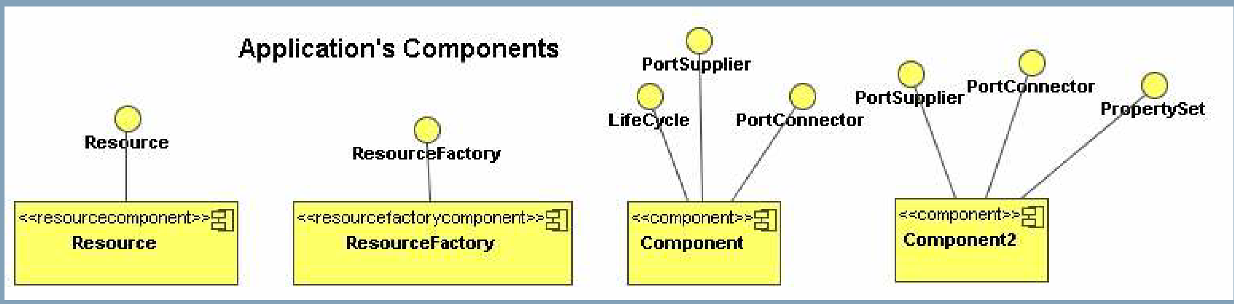
- Component – extension of UML Component that is the base component definition of all CF components
- Resource Component – a specialization of Component that realizes a Resource interface.
  - The Resource Component is extended by the interfaces (e.g., Common and Data Link Layer Facilities) it provides and/or uses.
- Resource Factory Component – a specialization of Component that is a factory that manages Resource Components



## Low Profile Component Definitions

- The OMG UML Profile for CF does not restrict one from building lighter weight components other than a Resource Component.
  - Components that have configuration and/or query properties require the property set interface
  - Components that have uses port require the PortConnector interface
  - Components that have provides port require the PortSupplier interface
  - Components that require initialization and teardown require the LifeCycle interface
  - Components that have testable properties require the TestableObject interface
  - Components that require over all control require the ControllableComponent interface

# Application Components Illustration






# Device Components

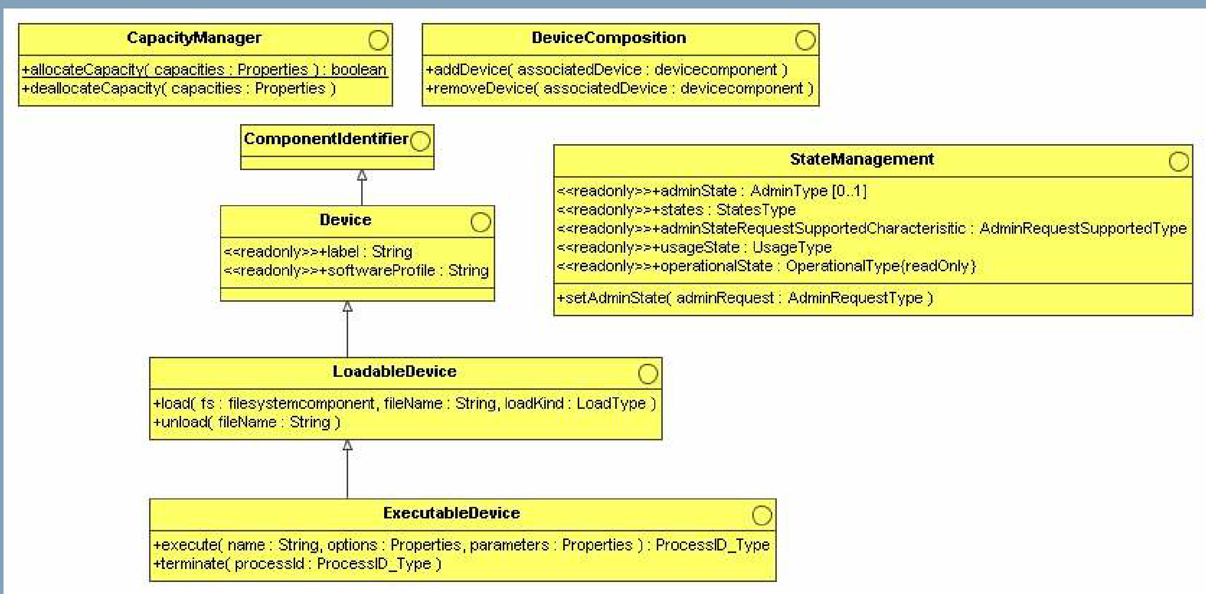
- A Device Component is a specialization of Component and ServiceComponent
- A logical device that manages communication equipment
  - A type of service component that may have states and service properties (capabilities):
    - Characteristics – properties that described qualities and/or type of a device. They are usually static in nature.
    - Capacities – computational property values
    - A Device Component can optional manage its service properties, not mandatory.
  - Provides the mechanism (device driver) to directly interface with the communication equipment
  - Optional states: Admin, Operational, and Usage.
  - Optional Status Properties
  - The device components can be extended by the interfaces it provides and/or uses.

# Device Component Interfaces

- A set of ***minimal*** generic interfaces for device management
  - CapacityManager – supports the management of device's capacities.
  - DeviceComposition –supports management of devices simultaneously
  - Device – extends Resource interface with capacity and state management
  - LoadableDevice – extends Device interface with generic load and unload of software for a communication equipment.
  - ExecutableDevice – extends LoadableDevice interface with generic execution and termination of component containers (e.g., processes, threads, environments).
  - StateManagement – provides statement management for a device.



# Device Component Interfaces

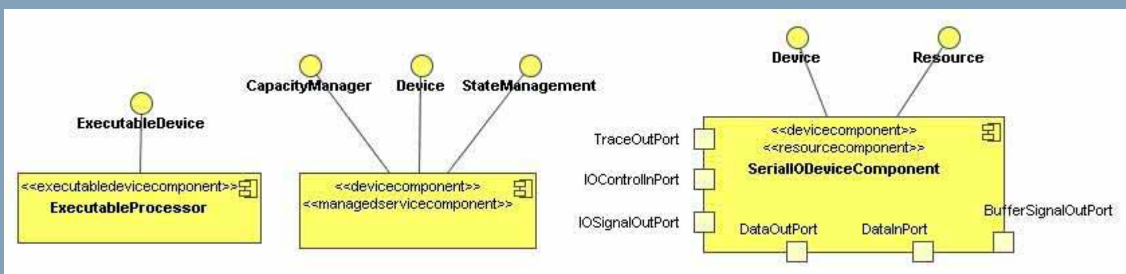


## Low Profile Device Component Definitions

- The OMG UML Profile for CF does not restrict one from building lighter weight Device components since a Device Component does not have to manage its Service Properties.
  - Device Components that manage some or all of their Service Properties would require the capacity operations and StateManagement interfaces.



## Device Component Illustrations



## Component Framework Flexibility and Optimizations

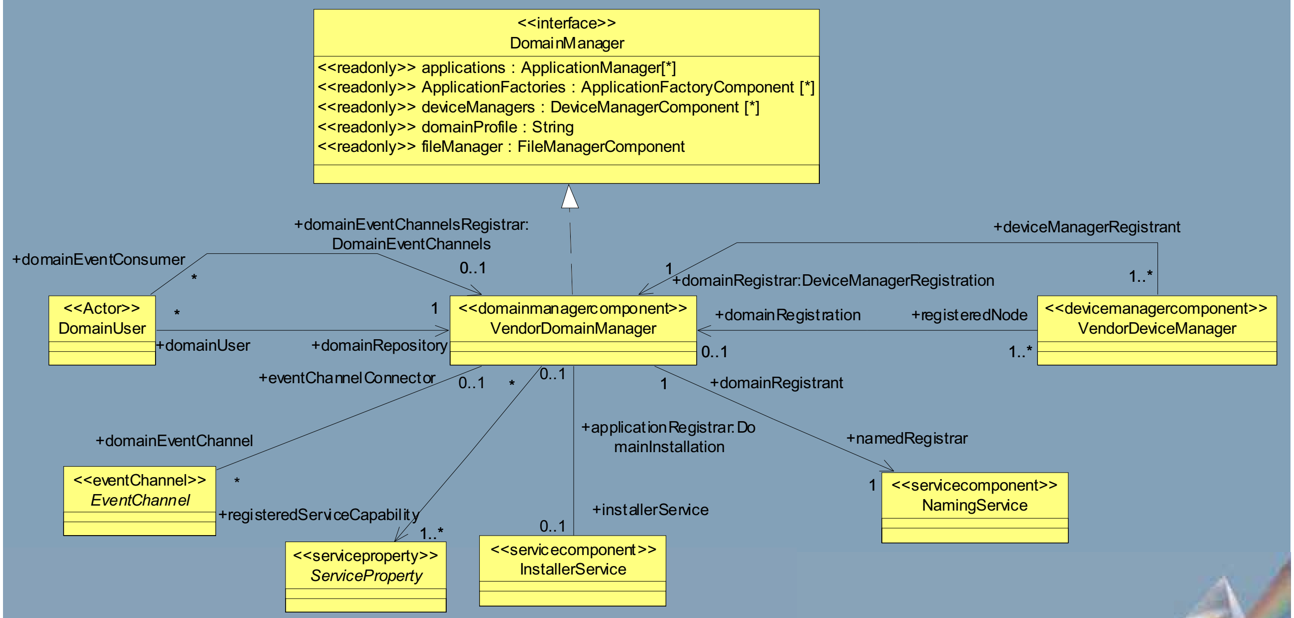
- Component Definitions
- Application Deployment Optimizations



## Component Definitions

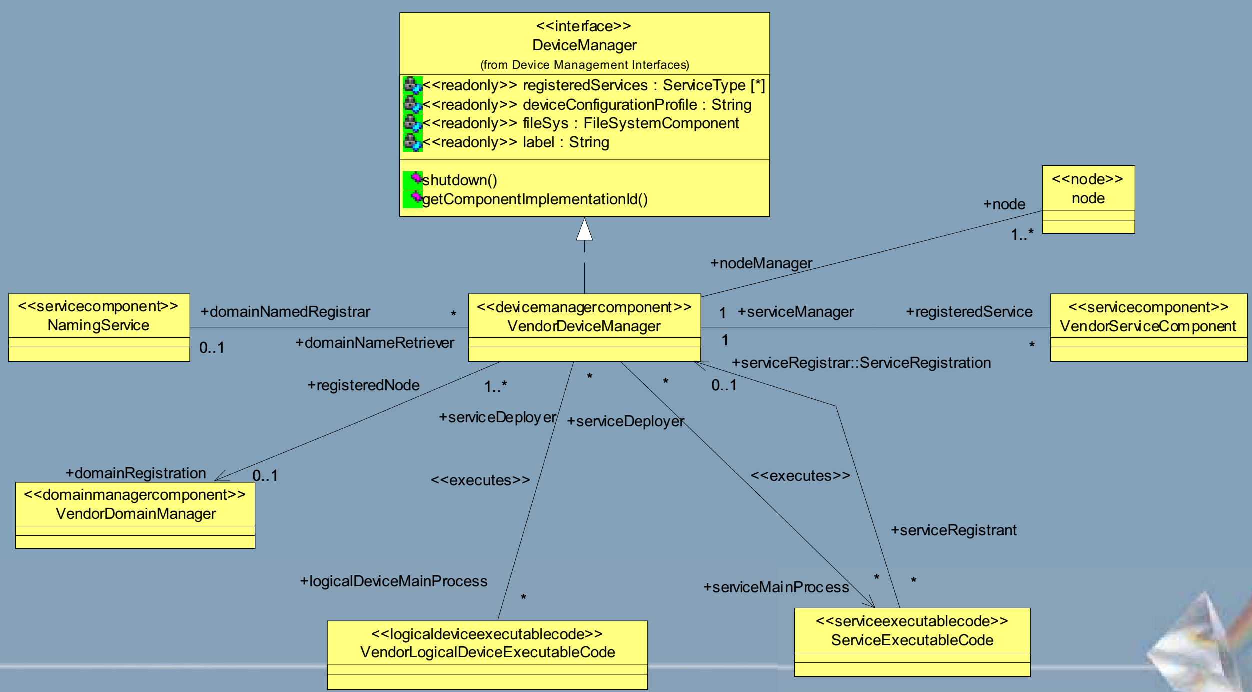
- Flexibility
  - Includes options for PIM level specification of Lightweight component definitions
    - Lighter weight Application Components than Resource Component can be defined. Minimal set of interfaces needed for deployment are: Lifecycle, PropertySet, PortSupplier, PortConnector
      - *If they exist for a component then deployment machinery uses them.*
    - Domain and Device Management allow for port definitions to specific interfaces which are supported
    - Lighter weight Device components support
      - *a configurable combination of states and statuses and*
      - *what capacities, if any, it manages*

# Domain Manager Component Illustration






# Device Manager Component Illustration

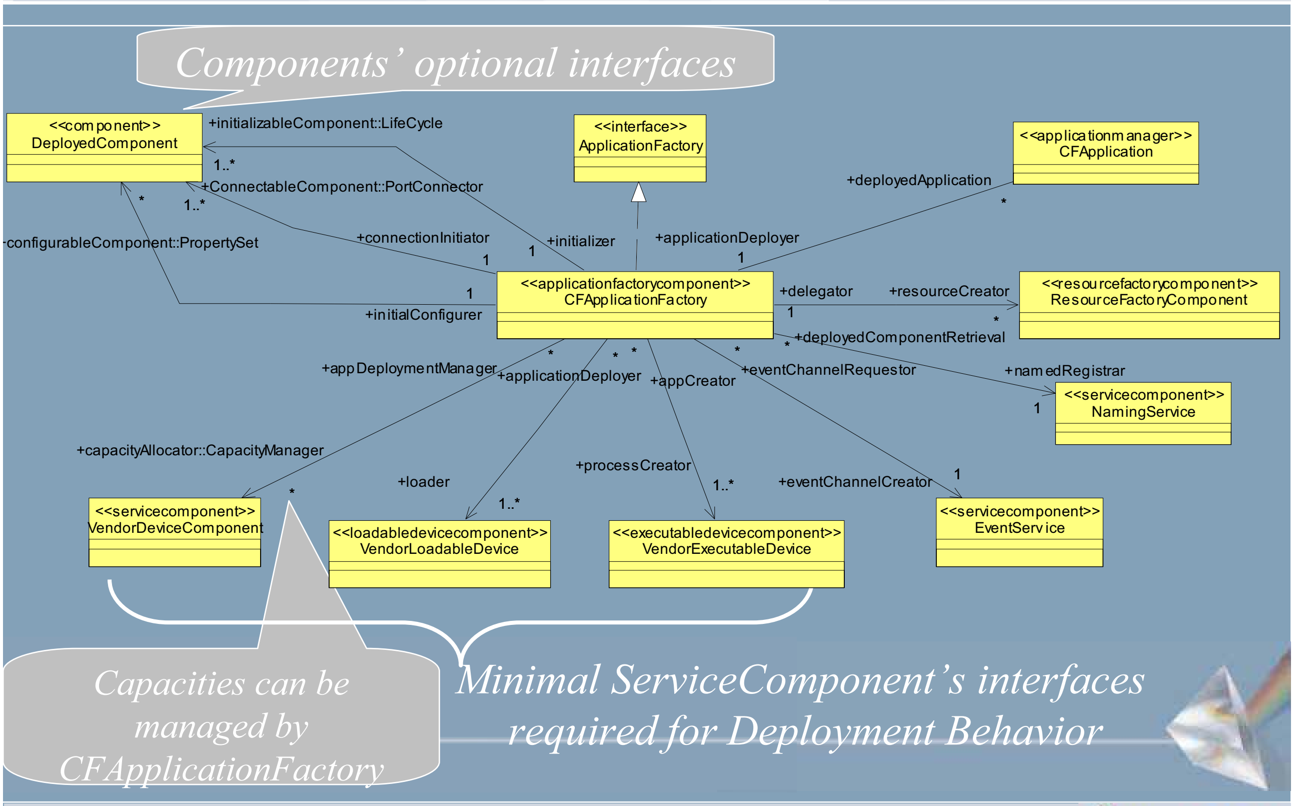


# Application Deployment Optimizations

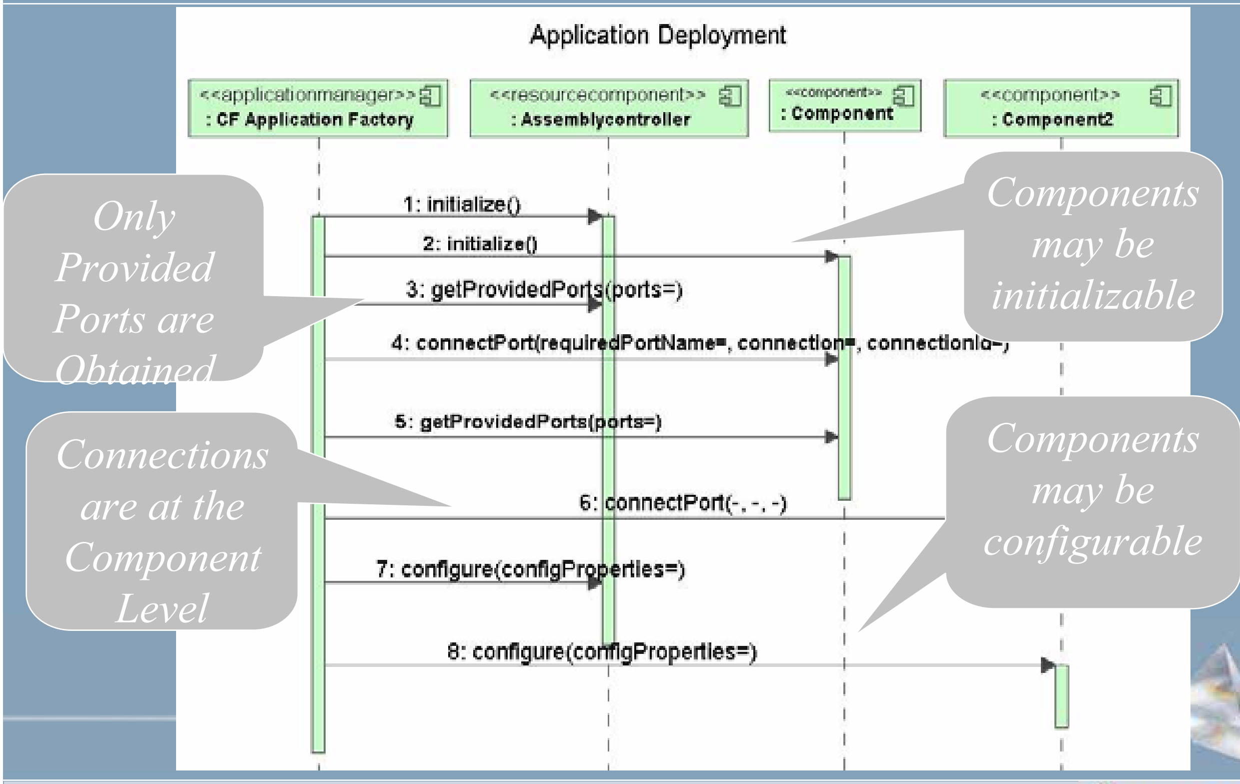
- Application Deployment Optimizations
  - Connection Behavior is simplified
    - Connections are managed at the component level not at uses port level
    - Able to retrieve a list of provided interfaces at one time.
  - Application Factory Component can make all devices' characteristic and capacities decisions
- Teardown Optimization
  - Disconnection Behavior – disconnections are only necessary for radio services (not deployed components)



# Application Deployment Optimizations, cont'd



# Application Deployment Optimizations






# Conformance

- Simply put, conformance is defined at level of component and interface usage.
  - No requirement on what components are required for a radio set/system.
  - One needs to determine what radio components along with their interfaces are required for a software radio being built based upon the radio requirements and level of portability one is striving for.
- The OMG Software Radio specification defines three levels of portability like the SCA, which are at the:
  - Radio domain level,
  - Radio's node level and
  - Application level.

## Summary

- Provides a flexible Software Radio DSL and facilities to model and capture waveform and platform designs independently of technology that can be transformed to any technologies
- The Software Radio DSL can be easily extended or constrained for other domains
- The OMG Software Radio specification maps to the SCA

# Robotics-DTF/SDO-DSIG Joint Meeting

December 5, 2006

Arlington, VA, USA

Hyatt Regency Cristal City Hotel





# Approval of the Anaheim Minutes

Anaheim Minutes review

- **2nd Revised Submission of Robotic Technology Component (RTC) has been recommended in MARS, AB, and TC.**
- **RTC-FTF has been chartered**
- **4 interesting talks**
  **(SunSPOT demo, SysML Tutorial, Robot Ontology, Agriculture Robot)**
- **Reports received from 3 active Technical WGs**
  **(Decided 3 parallel session from next meeting)**
- **Contact reports received for a number of activities**

Anaheim Meeting Quorum : 4

ADA Software, AIST, ETRI, Hitachi, JARA, KAIRA, Kangwon National Univ., OIS, RTI, Samsung, Systronix, Shibaura-IT,
Proxy: NEDO, Technologic Arts

minutes taker(s):

- Yun-Koo Chung (ETRI)
- Bruce Boyes(Systronix)

# Review Agenda

Tuesday (Dec.5)

09:45-10:00 Plenary Opening

10:00-12:00 2 Technical Presentations / Services WG

13:00-17:30 4 Technical Presentations

Wednesday (Dec.6)

09:00-12:00 WG Reports and Roadmap Discussion

14:00-15:00 1 Technical Presentation

15:30-17:00 Contact WG Reports

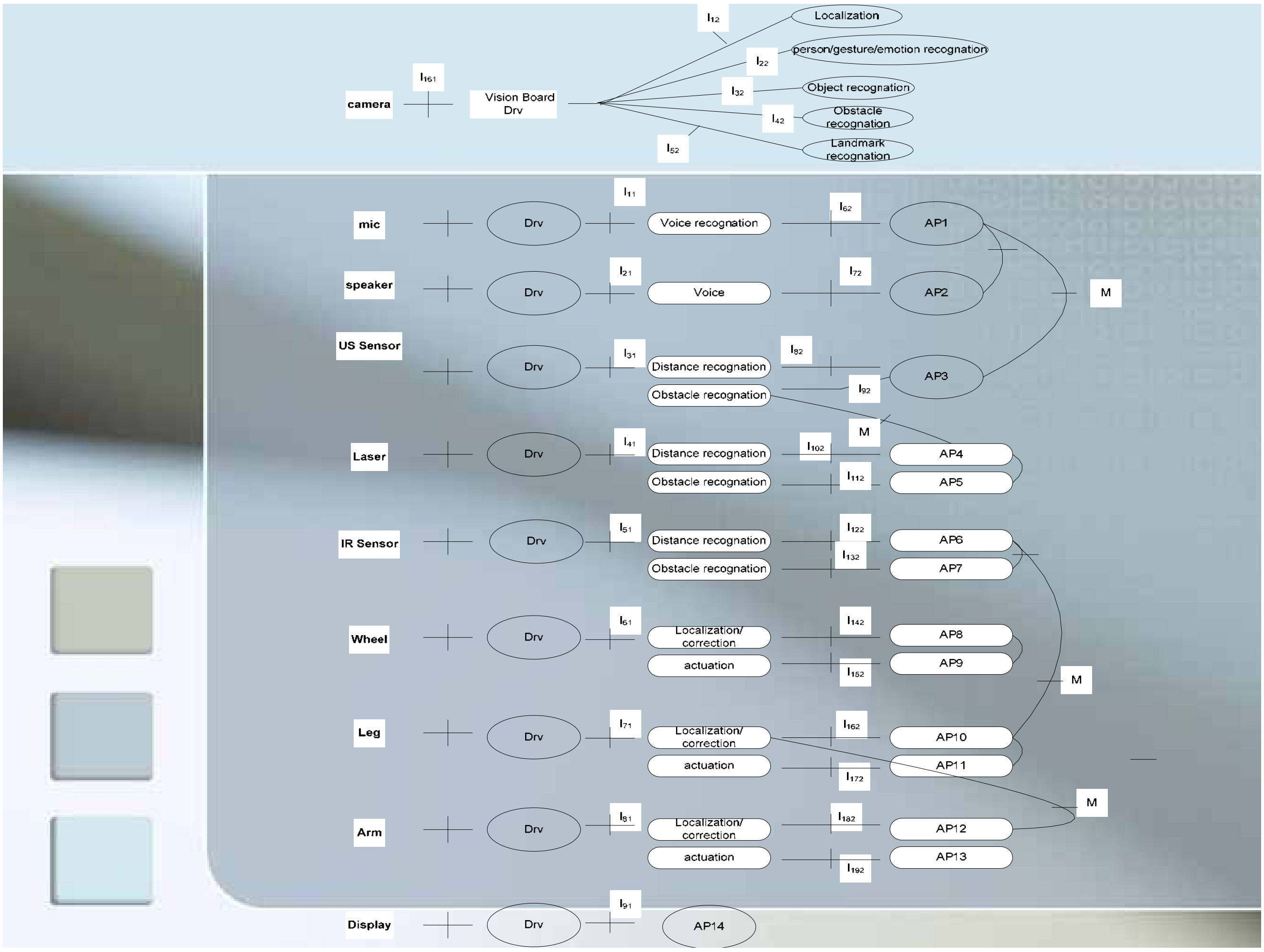
17:00-17:30 Plenary Closing (Next meeting agenda)

17:30-18:00 Robotics WG Co-Chairs Planning Session

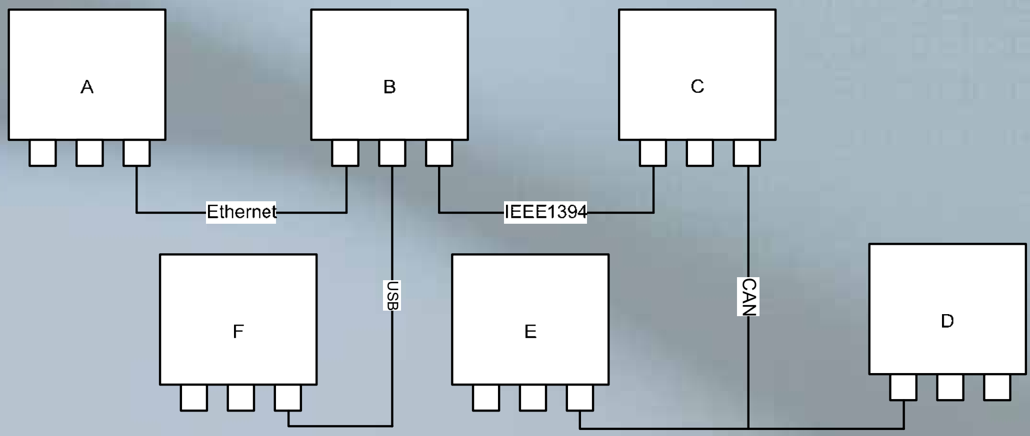
robotics/2006-12-08

# Heterogeneous Network Middleware for a Personal Robot

OMG Technical Meeting, Washington
December 4-8



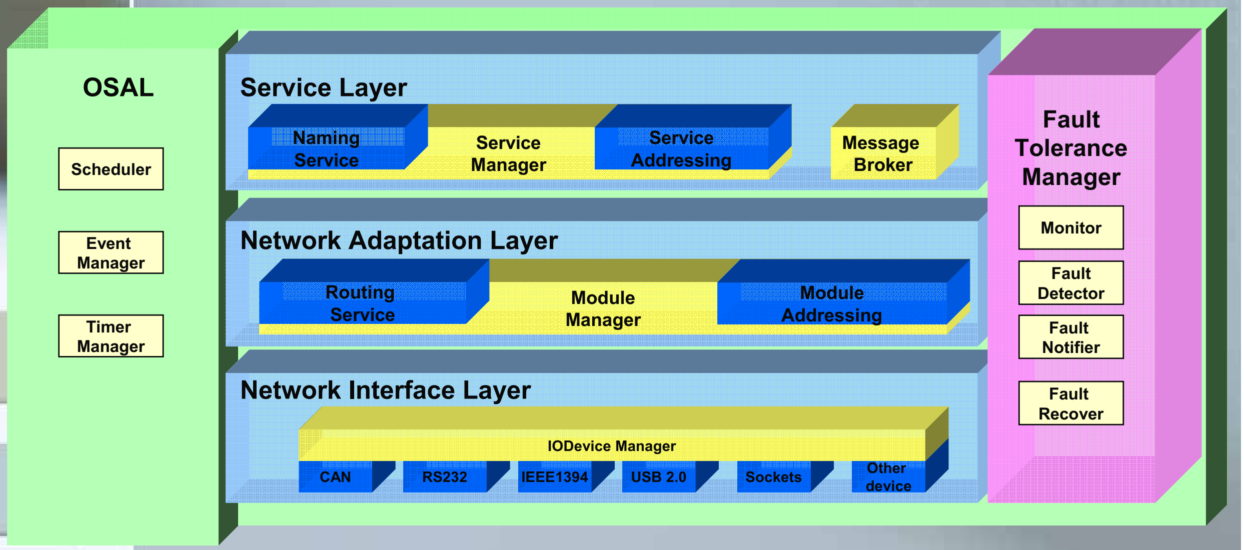
# Modular Robot



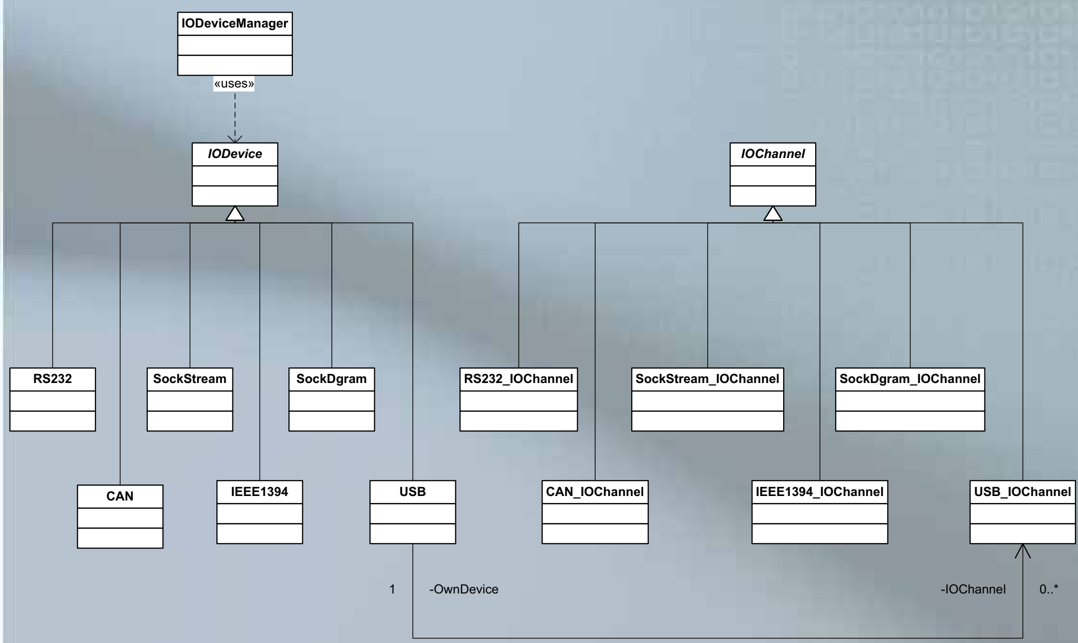

# Requirements

- Support for non-IP (Heterogeneous) networks
- Common Network address space
- Network Fault Tolerance
  - Connection management
  - Routing

# Architecture of HeNeM



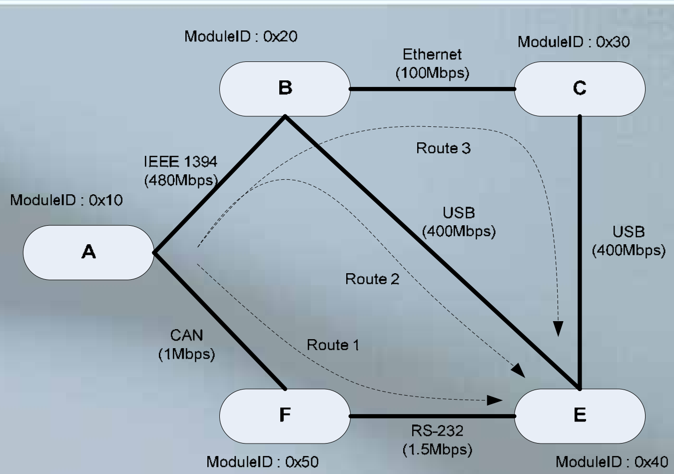

# HetNet Support in a HeNeM: NIL



## HetNet Support in a HeNeM: NAL

- Network Adaptation Layer
  - handles presence of the module in a robot network
  - addressing of the module, routing
  - Negotiated module's address assignment
    - Maximum address in a network + 1
  - Assures that the address of the module is kept in case of absence

## HetNet Support in a HeNeM: NAL




| Module ID | # of Hops | RTT | Physical BW | Allocated BW | Next Hop | Route Status |
|---|---|---|---|---|---|---|
| B | 1 | – | 480 | 50 | B | Active |
| F | 1 | – | 1 | 0 | F | Active |
| C | 2 | – | 100 | 60 | B | Active |
| C | 3 | – | 400 | 18 | B | Active |
| C | 3 | – | 1 | 0.2 | F | Active |
| E | 2 | – | 1 | 0.2 | F | Active |
| E | 3 | – | 400 | 18 | B | Active |
| E | 3 | – | 100 | 60 | B | Active |

# Conclusion

- RTC Compliance
- Binary Size < 500K

robotics/2006-12-09

# RTM on Java/HORB: Eclipse meets RT Components

HIRANO Satoshi, Qu Runtao, KUBOTA Takaya

also with

OHKAWA Takeshi, Mikael Johansson(*)
ANDO Noriaki, KOUTOKU Tetsuo
YASU Yoshiji(**)

*Network Middleware Research Group, AIST*
*Task Intelligence Research Group, AIST*
** KTH Sweden*
*** High Energy Physics Research Institute*

Robotics DTF, Dec 2006
Washington DC

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

# Outline

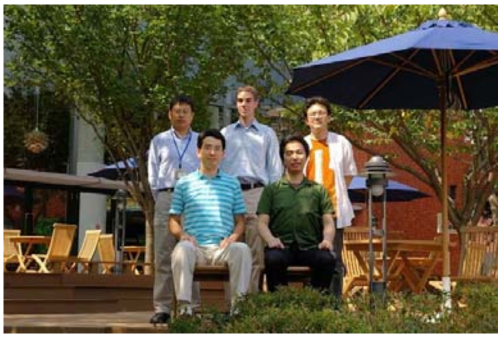
- "RTM on Java/HORB" as an RTC
- Why Java?
- *Demo inverted pendulum*
- Integration of RTC within Eclipse
- Performance Issues
- Deployment
- Consideration on programming model

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

# Who we are?

- A research group on network middleware for embedded systems
- Our technology is available for technology transfer (software, patents)
- HORB
  - lightweight Java middleware
  - e.g. Toshiba ApriAlpha

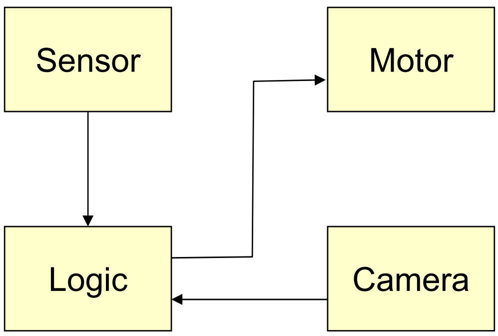
Courtesy of TOSHIBA Corp.





# What is RTC?

- Robotic Technology Component Specification
  - by OMG Robotics Task Force, AIST, RTI
- Software component model for robots



# What's Next for Robotics?

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

- Standards for making markets
- Cost down for affordability
- Safety

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

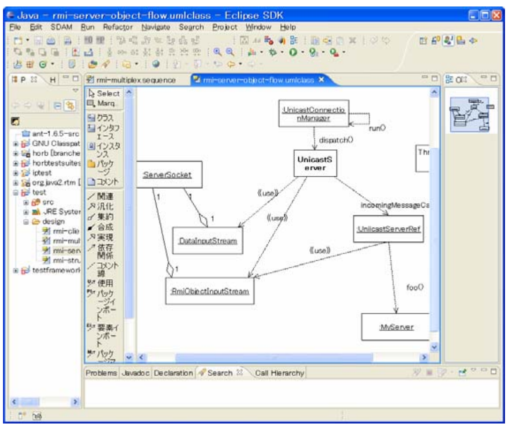
# Why Java? *Pros*

- Productivity
  - to reduce cost
  - 2～10 times better than C++
- Safety
  - no pointer, no security hole with stack overflow
  - (almost) virus free
- Education
  - lots of students



# Why Java? *Pros*

- Eclipse – a standard IDE
  - Visual GUI designer
  - UML editor
  - Plugin architecture
  - RCP (Rich Client Platform)
    - a plugin can run either inside Eclipse or as a standalone application
  - Remote deployment (OSGi)
    - automatic update



# Why Java? *Cons*

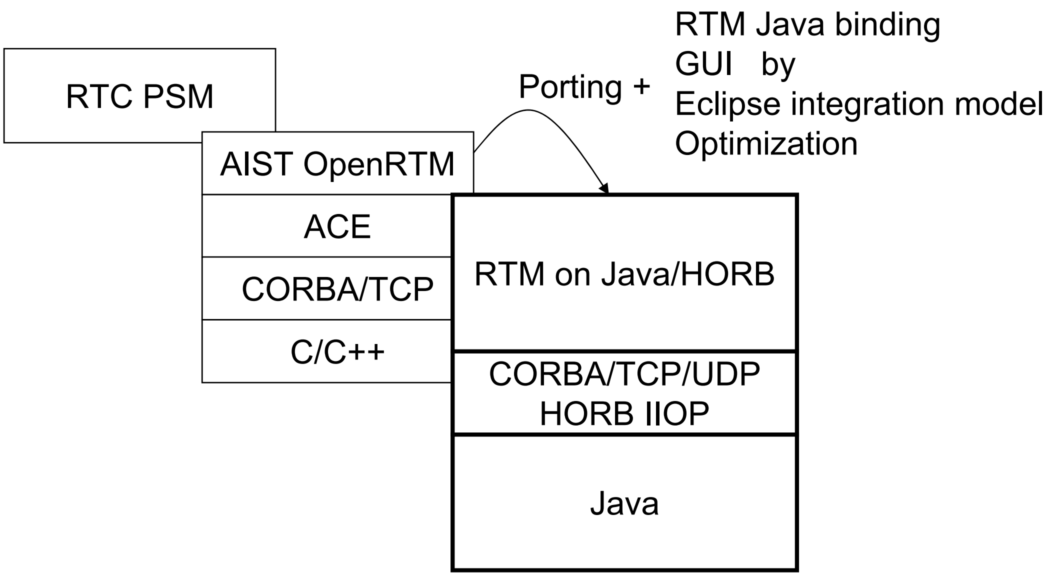
- Lack of real-timeness
  – Garbage collection stops the world

# Work Around

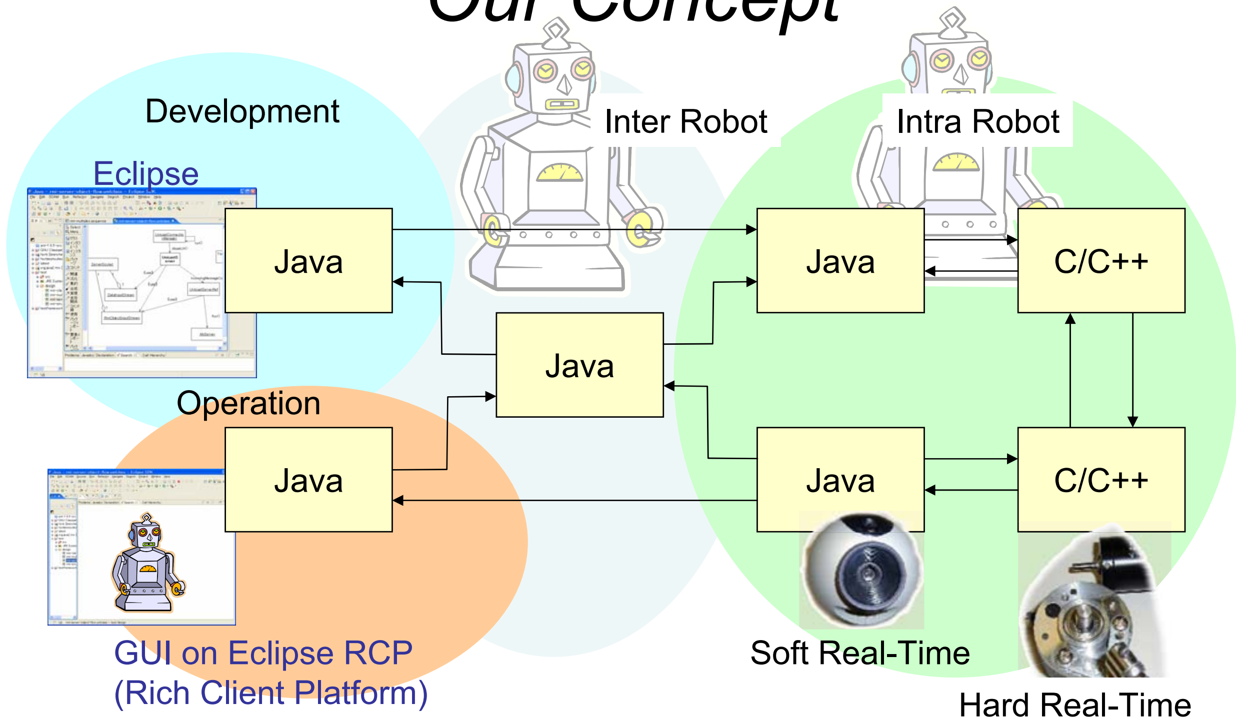
- RTSJ (Real-time Specification for Java)
  – No popular implementaion
- AIST Zero GC ORB Technology
  – Real-time remote method invocation in Java
  – not implemented in this demo.
  – Patent pending



# RTM on Java/HORB

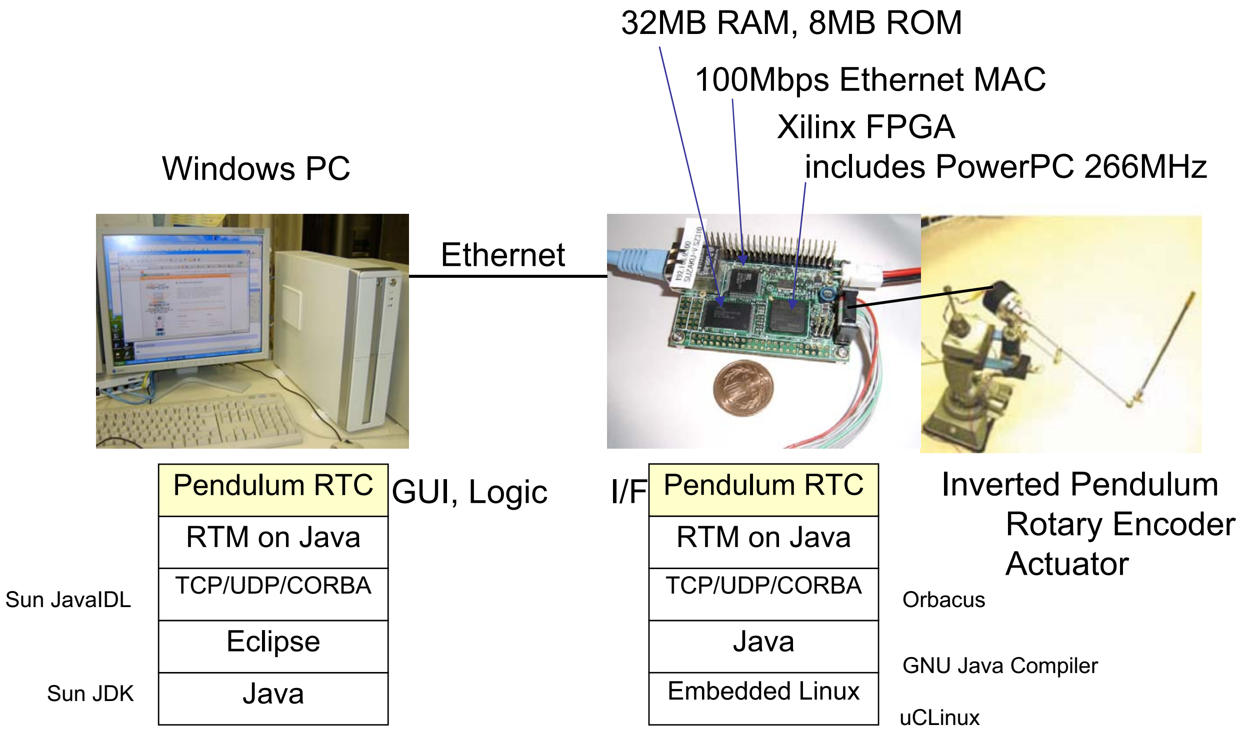


# RTM on Java/HORB
## *Our Concept*

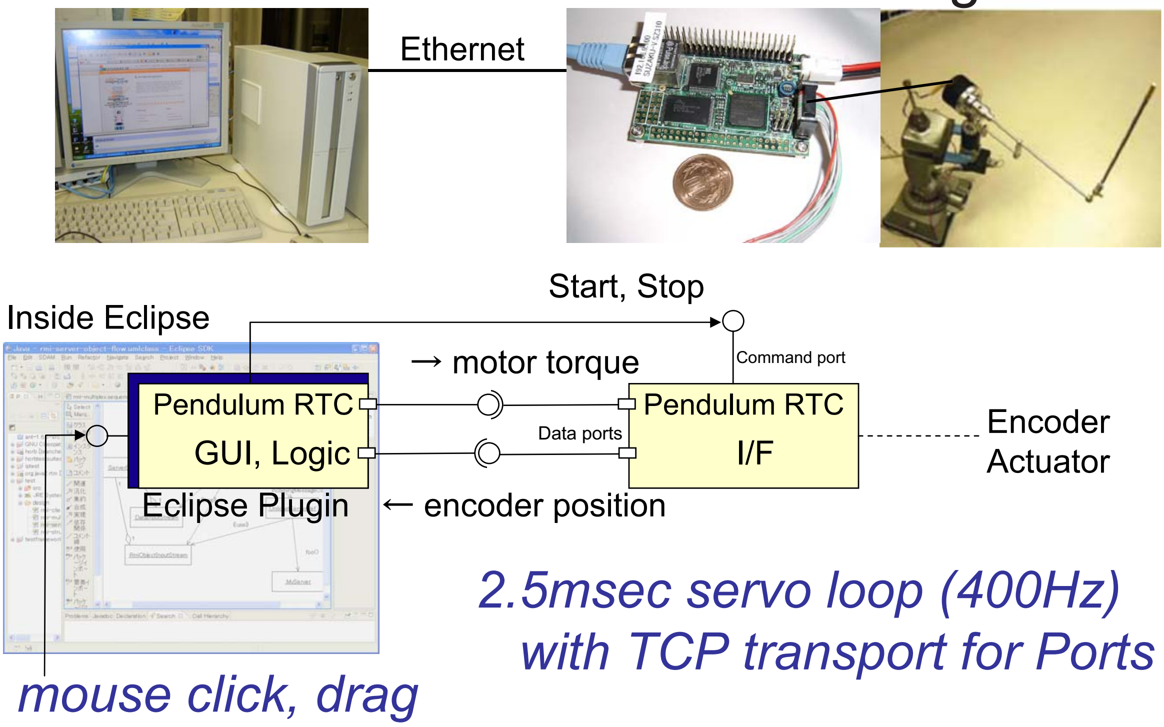





# Demo Configuration



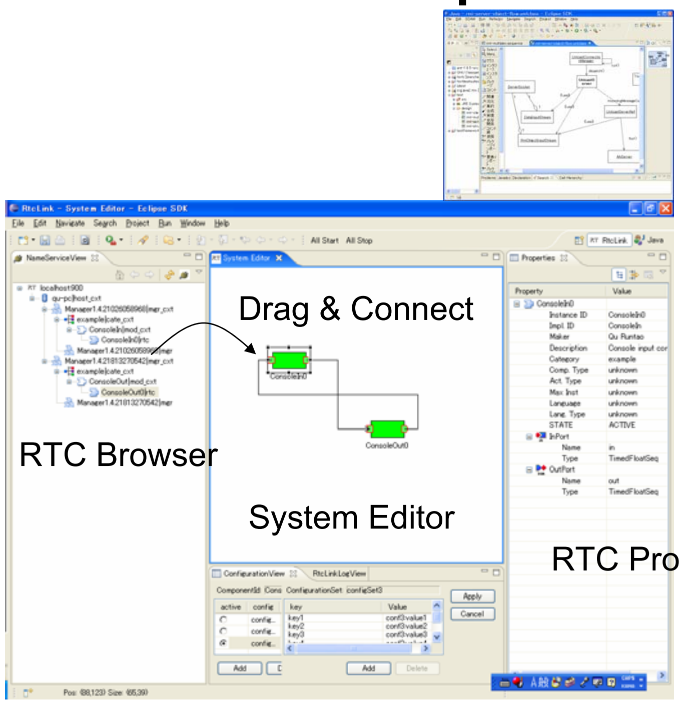
# Demo
# Remote Control for balancing






# Development with Eclipse

1. Visual design (UML, GUI)
2. JDT (Java)
3. CDT (C/C++)
4. Test (DisUnit)
5. RTCLink
6. Debug




(a part of OpenRTM)

# A problem to run within Eclipse

for GUI, easy debugging and deployment

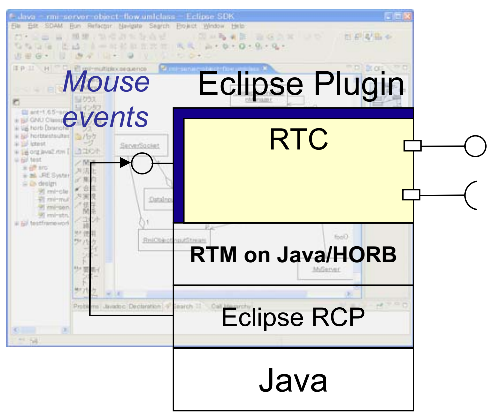
- but… RTM and Eclipse have different programming models
  - RTM   dataflow execution model
  - Eclipse  SWT event driven model



# An integration model of Eclipse and RTC was explored and defined

*Have fancy GUIs!*

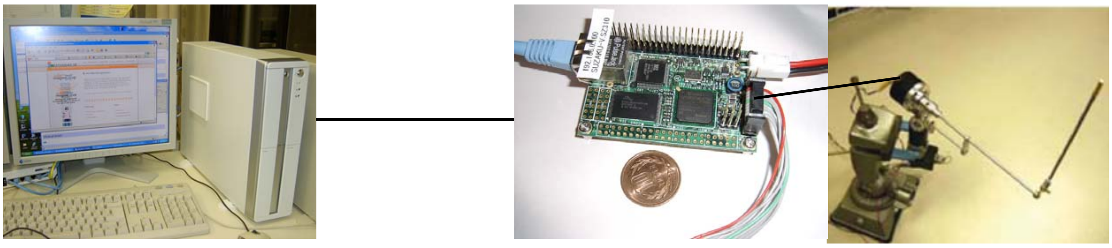
- an RTC can be an Eclipse Plugin
  - RTC Ports and Services
  - SWT events (Mouse click, Window move…)

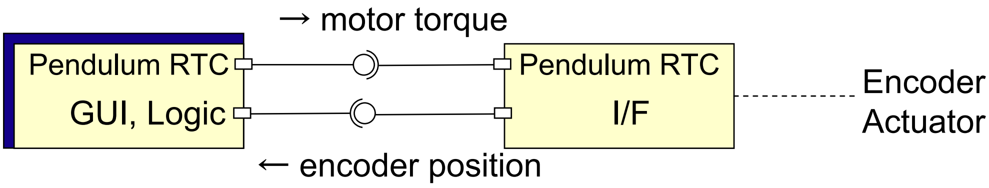



AIST NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

# Performance Issue
for Tight Servo Loop

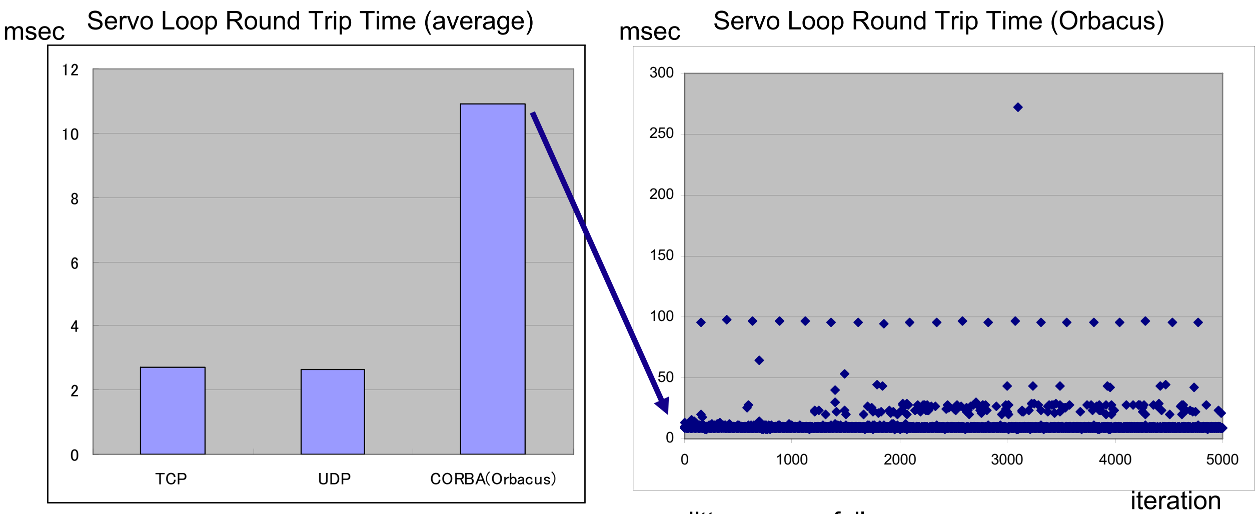
- Demo: three transport types for ports
  - CORBA(Orbacus), TCP, UDP
  - (CORBA for RTC Services)








# Performance Issue for Tight Servo Loop




balance balance drop!

Jitter causes failure.
We'll improve it with Zero GC ORB technology.

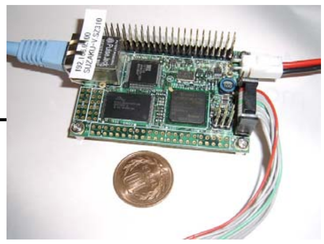
# Performance Issue
for Tight Servo Loop

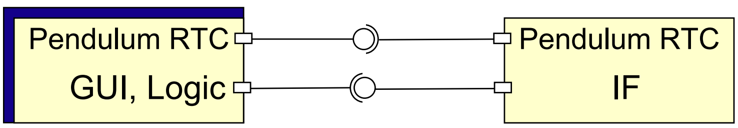
- CORBA(Orbacus) fails to stand up the pendulum.
- In case of Orbacus jitter (GC, other threads, etc.) causes failure. To avoid GC, we'll improve it with our Zero GC ORB technology.
- Alternative transports should be provided for small CPUs and tight servo loop.



# Consideration on RTC
Is local scheduling enough?

- Global scheduling is needed
  - priority shipping to prevent priority inversion
  - distributed global deadline scheduling






AIST NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

# Consideration on RTC from our experiences

- Dataflow execution model?
  - Periodic task activation
  - Pluggable task scheduler is desired to support variety of deadline scheduling algorithms
- Stimulus response execution model with FSM?
  - Asynchronous task activation
  - Why FSM?
- What we needed is just <u>blocking read and write</u> with timeout.
  - how about “Asynchronous messaging execution model”?



# Deployment

- Requirements
- Sample implementation which meets the requirements

AIST NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

# Deployment Requirements

- Console-less operation
  - No more telnet & ftp
- Wide range of robots and non robot apps.
  - 10g sensor node to 10ton construction robot
  - No XML
- Both network deployment and ROM boot
- Flexible and easy configuration change
  - Supports DIP SW
  - Supports any configuration style
  - {Vendor / product / entity } specific configuration



- Neutral to language, network media, OS
  - C, C++, Java, RTOS, Windows, Linux, WIFI, Zigbee…
  - Protocol bridge
- Interoperability
  - One deployment server for RTCs from multiple vendors
- Scalability
  - Capability of handling millions of robots
  - Dynamic load balancing at deployment servers
  - Fault tolerance

AIST NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

# Deployment Requirements

- Security
  - Authentication of both robots and deployment servers
  - Communication encryption during download
- Utilizing of existing technology and infrastructure
- ROM Flash
- Emergency reset
  - Automatic reset after program fault (e.g. SEGV error)
- Logging



# Impossible?

- We should have a baseline solution which can be used in any configuration.

AIST NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

# Possible technologies for the requirement

- HTTP for network download
  - Scalability
  - Security by optional https
- DNS
  - Server lookup
  - Dynamics load balancing (DNS round robin)
  - Fault tolerance
  - (optional IPv6, local name resolution)
- filelist file



# World's simplest implementation which meets the requirements

```
DIPSW=`dipsw_prog`
if [ $DIPSW == “0” ]
then
    startup_rom
else
    ADDR=  “network address”    # `/sbin/ifconfig | grep
    URLBASE=http://deployserver/vendor-$ADDR
    wget "$URLBASE/hosts“
    cp hosts /etc/hosts
    wget "$URLBASE/filelist"
    for file in `cat filelist`
    do
          wget "$URLBASE/$file"
    done
    chmod +x startup
   startup
fi
reboot
```

https: for authentication and encryption

DNS load balancing
Web Server fault tolerance

Flexible confuration

RTC invocation
or ROM flash

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

# Demo

- Inverted Pendulum works fine!
- Real-time remote control from Java RTC (also a Eclipse plugin)
  - (almost) 2.5msec servo loop
  - handles mouse events (start, stop, sliders)

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

# RTM on Java/HORB a bit more detail

- Development
- Integration of Eclipse and RTC
- Performance Issue
- Consideration on RTC from our experiences

# IEEE 1588: An Update on the Standard and Its Application

John C. Eidson
Agilent Technologies, Inc
5301 Stevens Creek Boulevard, MS 54U-SM
Santa Clara, CA 95051-7201
john_eidson@agilent.com

http://ieee1588.nist.gov





## Outline

What is IEEE 1588

History of the Standard

IEEE 1588 Basics

IEEE 1588 Version 2

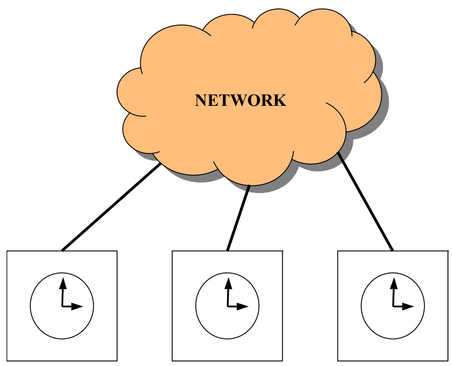
Applications

## What is IEEE 1588

IEEE 1588 synchronizes real-time clocks in the nodes of a distributed networked system.

Enables a new methodology for measurement and control

- BASED ON TIME
- NOT ON TIME-OF-RECEIPT-BASED EVENT NOTIFICATION.







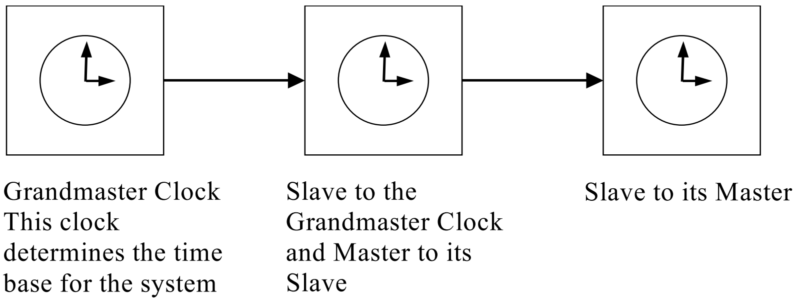

## History and Status of IEEE 1588

1. Published in November 2002
2. IEC 61588 in May 2004
3. Symposia in 2003, 2004, 2005, and 2006. 2007 in Vienna
4. Industrial Automation, T&M, Military, Power Generation and Distribution, Home Entertainment, Semiconductor Manufacturing, Telecommunications
5. Products: Microprocessors, GPS Linked Clocks, Boundary Clocks, NIC Cards, Protocol Stacks, RF Instrumentation, Aircraft Flight Monitoring Instruments
6. Referenced in: IEEE 802.1as, PC37.1, IEEE 1646-2004, LXI Consortium, ODVA
7. Version 2 Due to Ballot in Feb-Mar 2007

## V1 Synchronization Basics

**Step 1: Organize the clocks into a master-slave hierarchy (based on observing the clock property information contained in multicast Sync messages)**

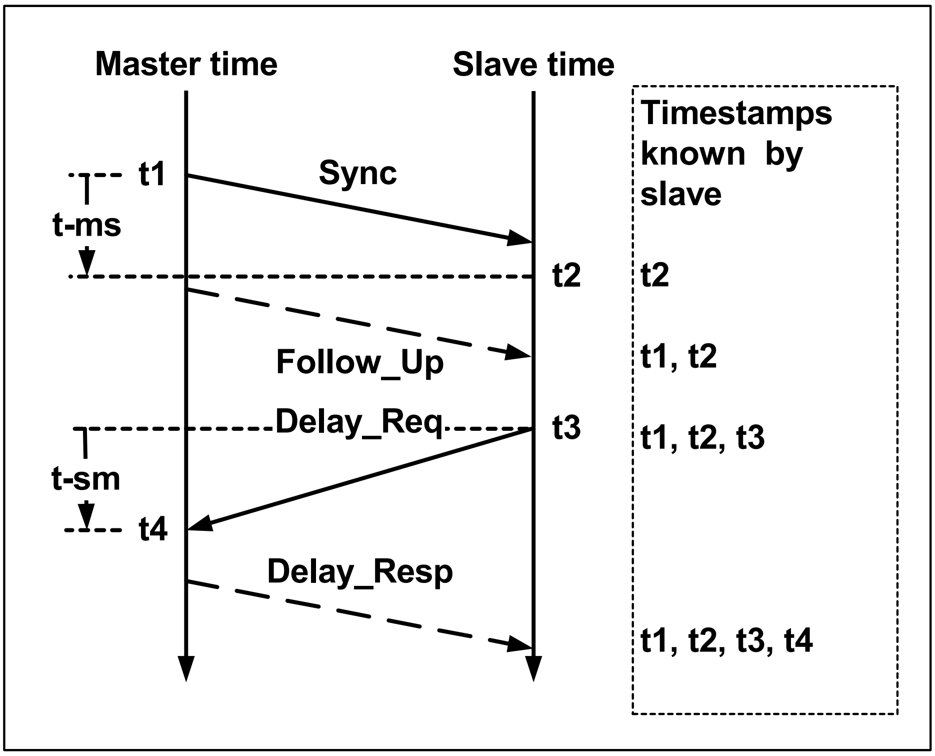
**Step 2: Each slave synchronizes to its master (based on Sync, Delay_Req, Follow_Up, and Delay_Resp messages exchanged between master and its slave)**



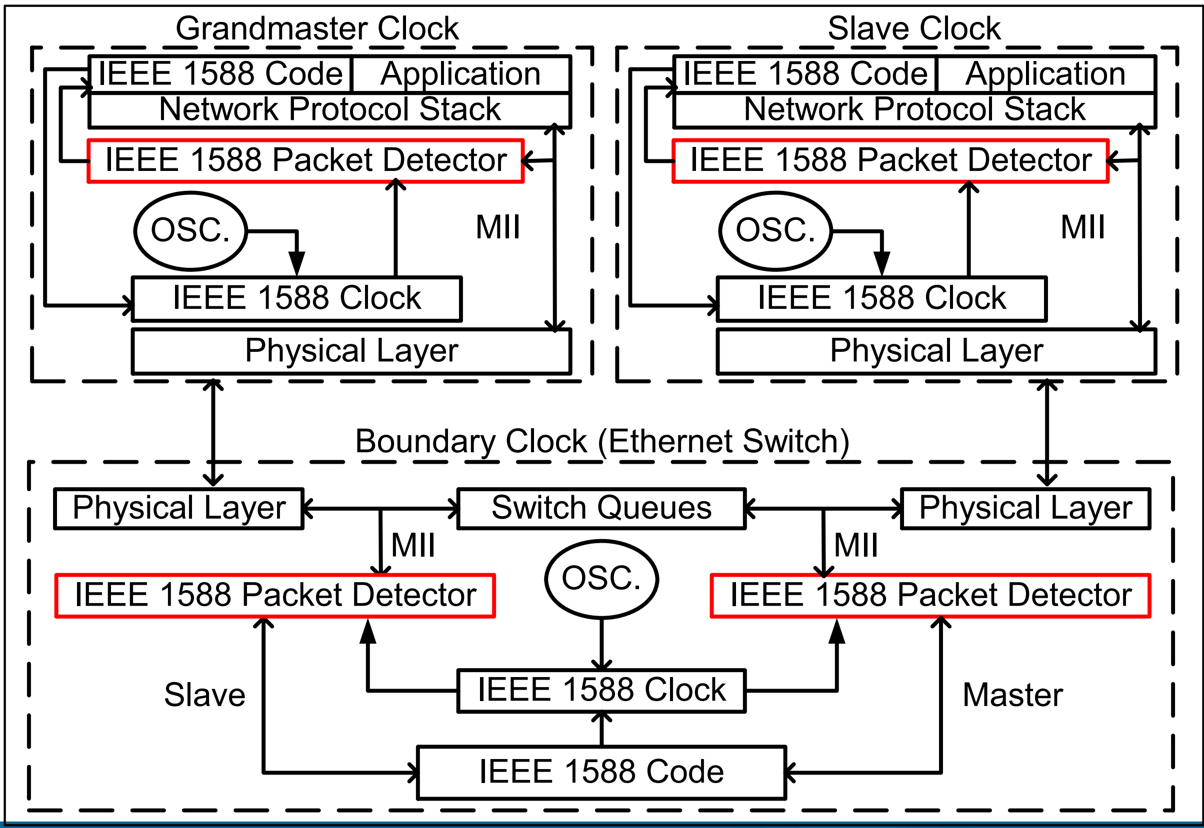





## IEEE 1588 Basics



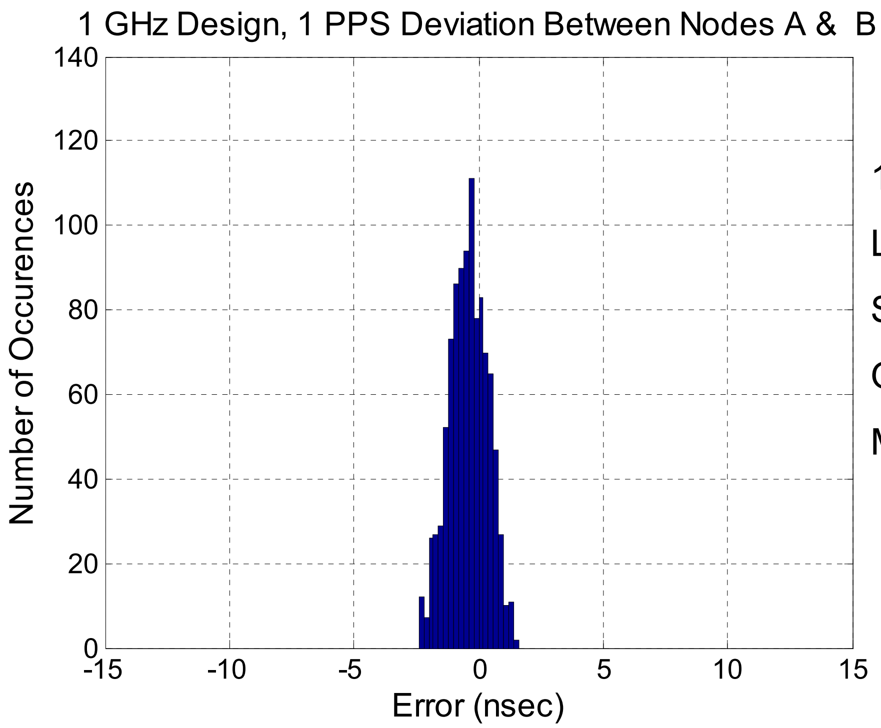
# IEEE1588 Basic Version 1 Components








# Current status: Timing accuracy




100 BT

LSB 1 ns

Std. Dev. 0.7 ns

Outliers: +1.6 to -2.4 ns

Mean -0.3 ns

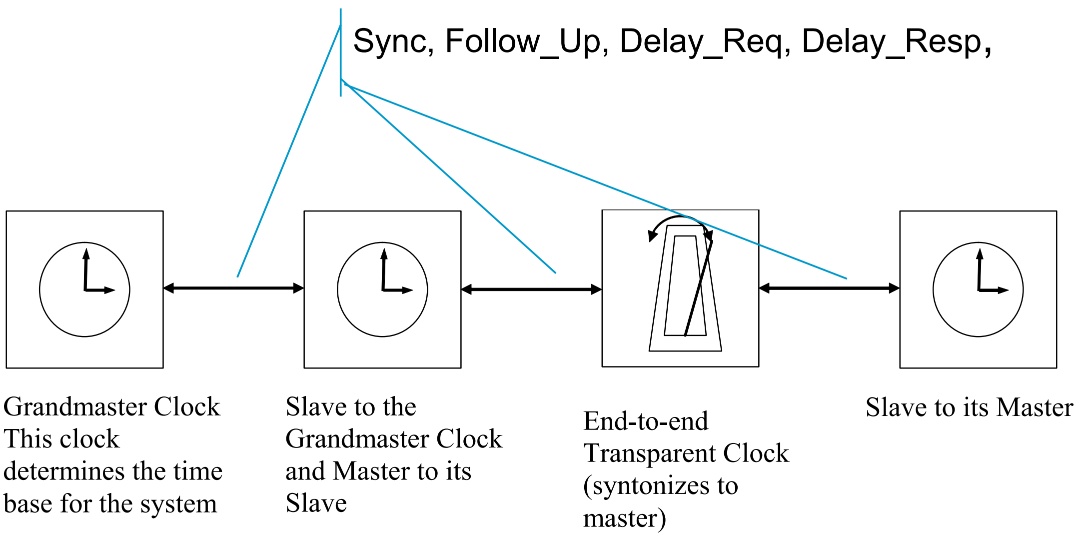
## IEEE 1588 Version 2: Motivation & Goals

1. New application areas and new requirements
2. Goals:
   - Enable sub-microsecond synchronization (military, T&M)
   - Shorten timing messages, equal length (telecom)
   - Long linear topologies (industrial)
   - Fault tolerant features (industrial)
   - Mappings: Layer 2 Ethernet, DeviceNet, ControlNet, ProfiNET, IPv6

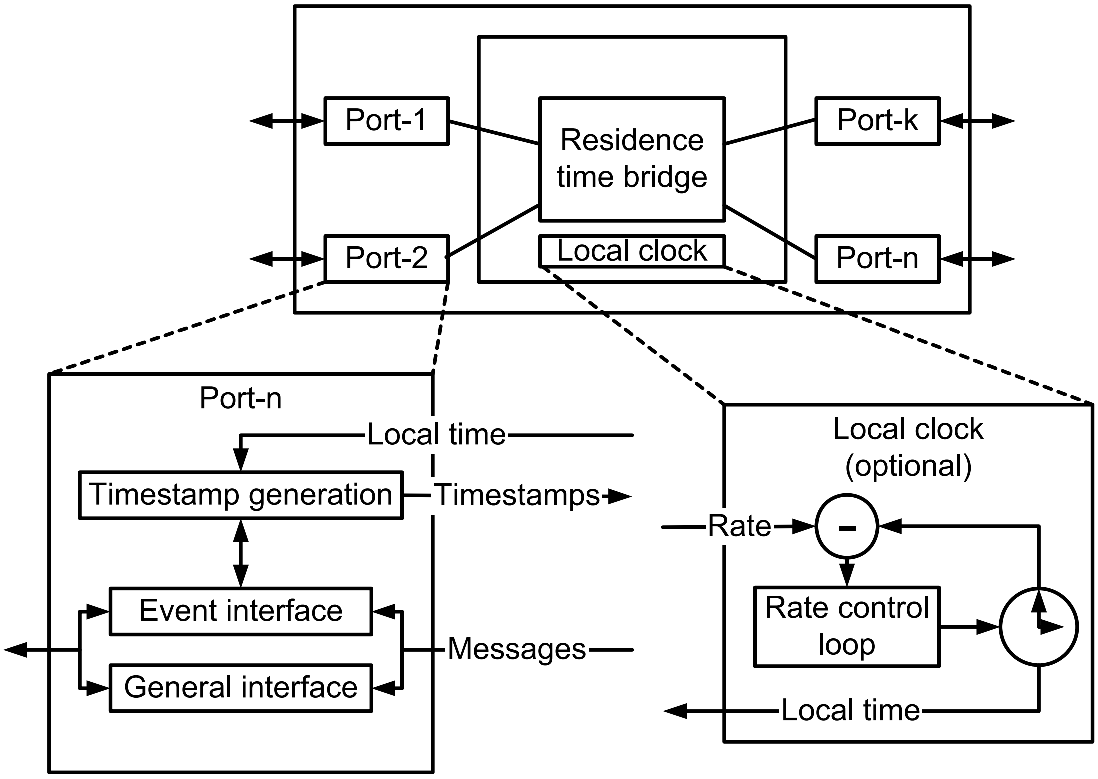




## V2 Synchronization Basics + End-to-end Transparent



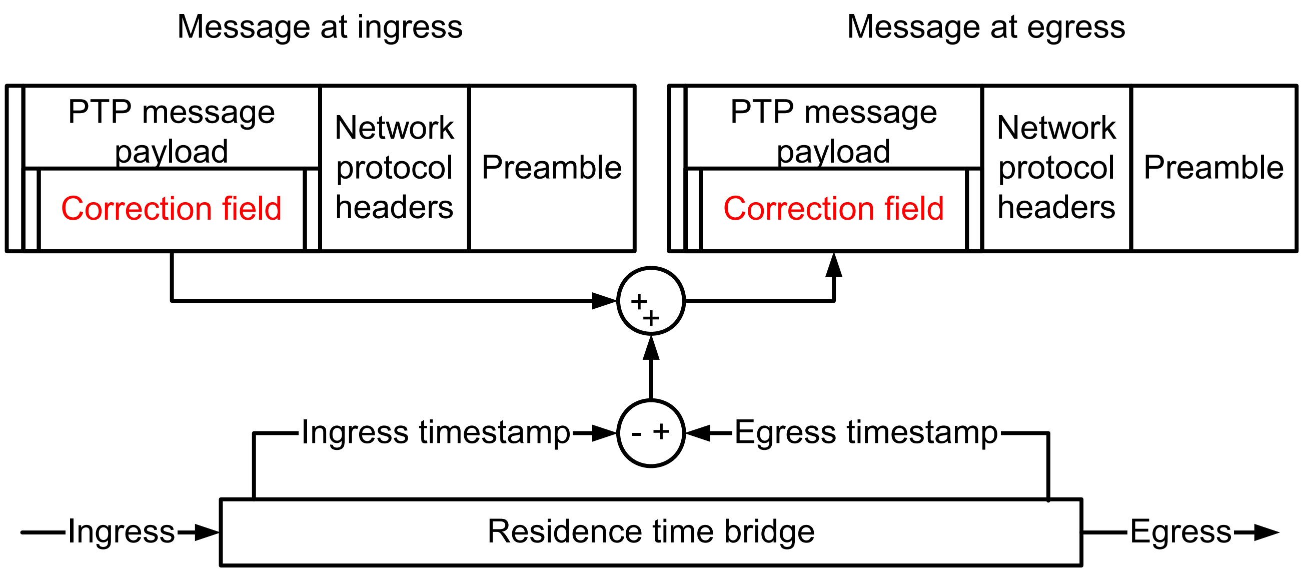
# V2 End-to-end Transparent Clock








# V2 End-to-end Transparent Clock Corrections

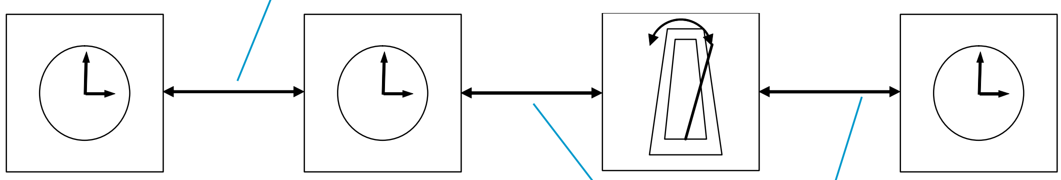




© Copyright 2006 Agilent Technologies, Inc

# V2 Synchronization Basics + Peer-to-peer Transparent

Sync, Follow_Up, Delay_Req, Delay_Resp



Grandmaster Clock This clock determines the time base for the system

Slave to the Grandmaster Clock and Master to its Slave

Peer-to-peer Transparent Clock (syntonizes to master)

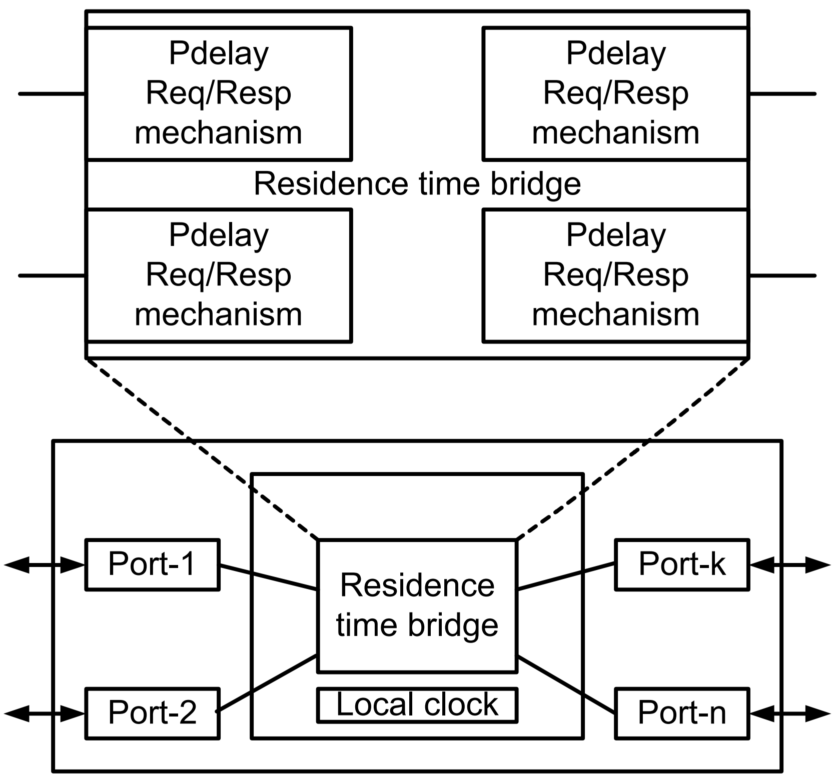
Slave to its Master

Sync, Follow_Up, Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up

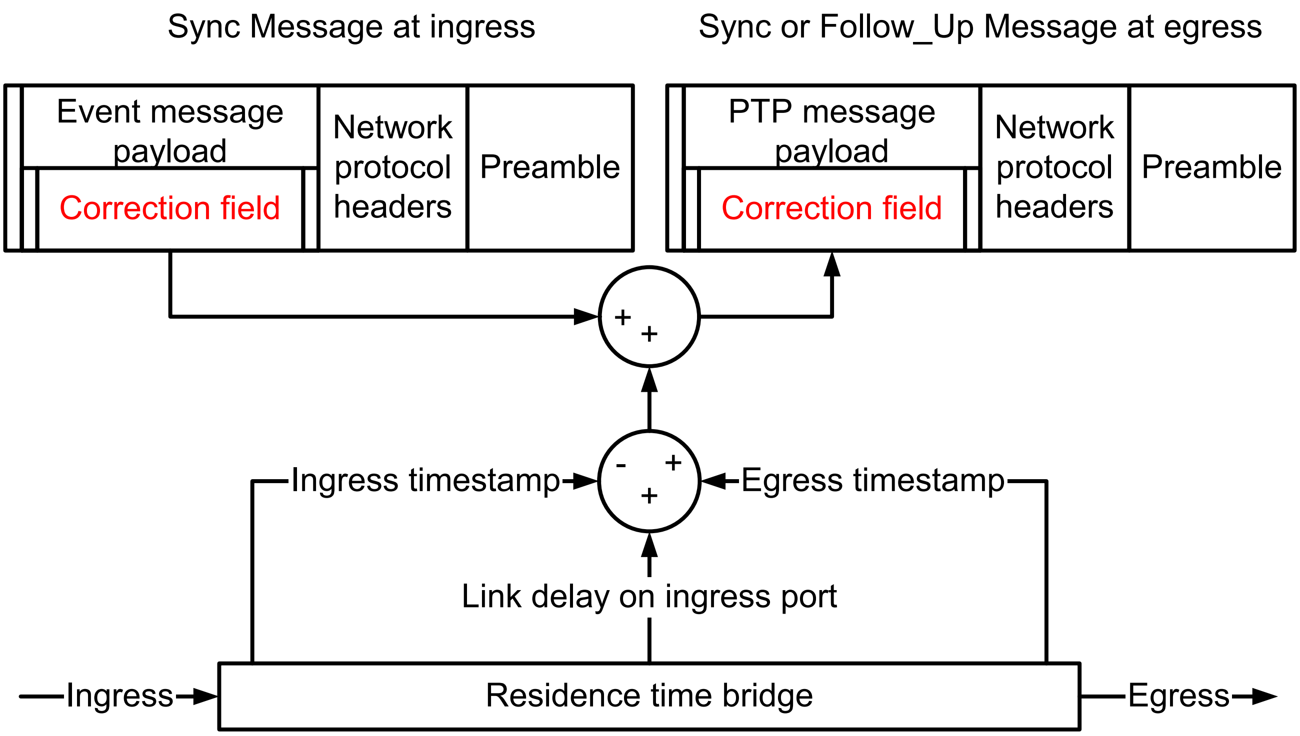




# V2 Peer-to-peer Transparent Clock



# V2 Peer-to-peer Transparent Clock Corrections








# V2 1588 Switches

Boundary clocks

a) **+ Good for hierarchical systems**
b) **+ Scale well with the number of devices**
c) **+ Can translate between different media.**
d) **+ Can resolve 1:N introduced by ordinary switches or end-to-end TC**
e) **— Must maintain state for scaling and 1:N**
f) **— Poor for linear systems (large number of daisy chained servos)**

End-to-end transparent clocks

a) **+ Can be used for hierarchical systems**
b) **+ Good for linear systems (eliminates cascaded servos)**
c) **— Scale poorly with the number of devices (master sees all slaves)**
d) **— Can introduce 1:N topology**

# V2 1588 Switches (con't)

Peer-to-peer transparent clocks

a) **+ Can be used for hierarchical systems**
b) **+ Scale well with the number of devices**
c) **+ Good for linear systems (large number of daisy chained clocks)**
d) **+ Rapid recovery with changes in network topology**
e) **— Cannot resolve 1:N introduced by ordinary switches, wireless, or end-to-end TC**
f) **— Must maintain per port path length state and measuring mechanisms.**
g) **— Only used in homogeneous P2P systems. Requires a boundary clock at the edges.**
h) **— 6 (vs. 4) measurements per link introduces slight degradation in accuracy**





# Split 'timing' and 'master-slave hierarchy determination'

V1 Sync message (165 octets)

- Network headers
- Version, subdomain, type
- Source identification
- Control, flags, update rates
- Origin timestamp
- Grandmaster information
- Local clock information
- Parent clock information

V2 Sync message (46 octets)

- Network headers
- PTP common: type, version, domain, CORRECTION FIELD, source identification, update rates, *control*
- Origin timestamp

## Split ‘timing’ and ‘master-slave hierarchy determination’

V1 Sync message (165 octets)

- Network headers
- Version, subdomain, type
- Source identification
- Control, flags, update rates
- Origin timestamp
- Grandmaster information
- Local clock information
- Parent clock information

V2 Announce message (88 octets)

- Network headers
- PTP common: type, version, domain, CORRECTION FIELD, source identification, update rates, *control*
- *Origin timestamp*
- Grandmaster information
- Local clock information
- Parent clock information





## IEEE 1588 on Wireless

1. No mapping in IEEE 1588 v2 to any wireless protocol
2. IEEE 802.11? Is working on a 1588 specification: WIP
   – Access points will be Boundary Clocks
   – Peer-to-peer 1588 technology
   – Some change in messages to use TCP/IP acks instead of 1588 Pdelay_Resp (or something like this- details not known)

# Applications:

•Industrial Automation and Motion Control

•Test and Measurement

•Military and Aerospace

•Power Distribution

•Home Audio-Visual: IEEE 802.1as

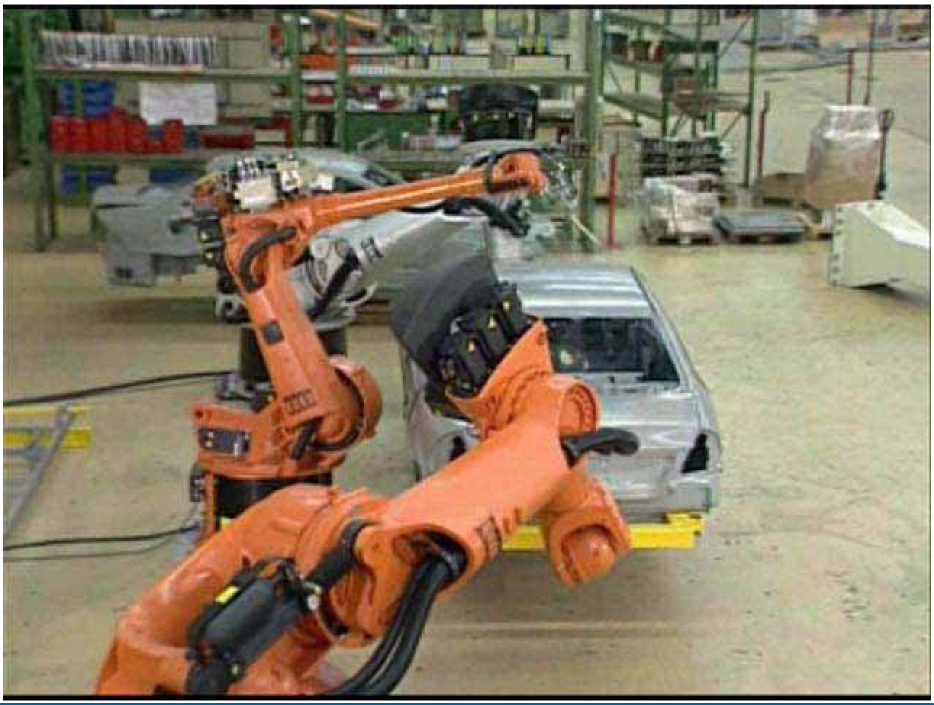
•Telecommunications



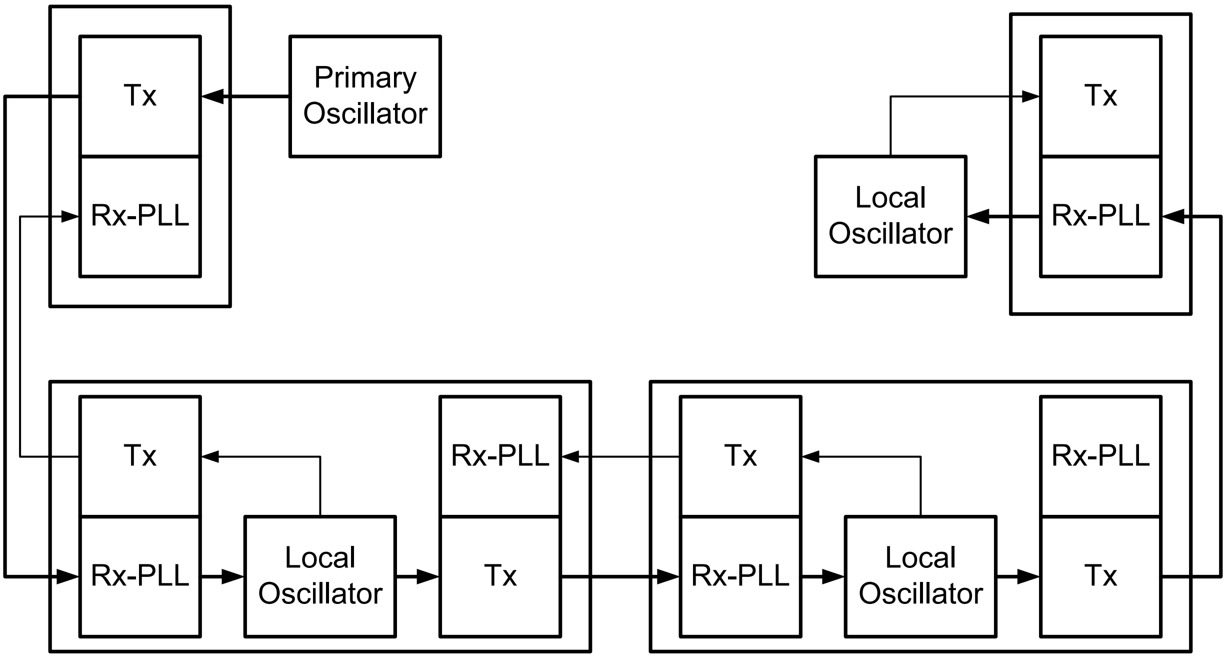


# RoboTeam in Action: Process Relative Motion

**Courtesy of Kuka Robotics Corp.**



# Chain of 1588 synchronous switches: Telecommunications








# Questions?

NIST

# *IEEE 1451*
# *Smart Sensor Interface Standards*

OMG-Robotics Working Group Meeting
December 5, 2006
Crystal city, Virginia

**Kang Lee**
kang.lee@nist.gov

**Manufacturing Engineering Laboratory**
**National Institute of Standards and Technology**

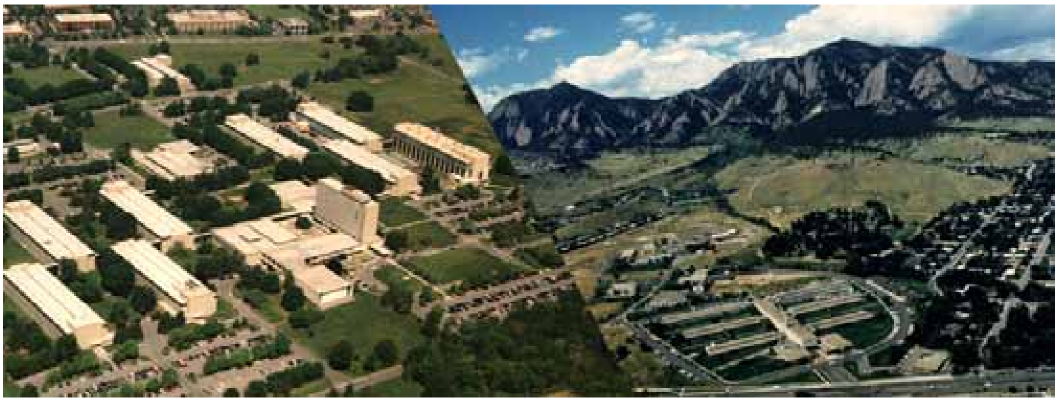
NIST

# *Outline*

- **Background – how IEEE 1451 comes about**
- **IEEE 1451 Smart Transducer Interface Standard**
- **IEEE 1451.0 and IEEE 1451.X**
- **Sensor Standards Harmonization Effort**

## National Institute of Standards and Technology (NIST)

**Mission: Develop and promote measurement, standards, and technology to enhance productivity, facilitate trade, and improve the quality of life.**



**Gaithersburg, Maryland** **Boulder, Colorado**

## Research & Development on Flexible Manufacturing Systems


- Studied interfaces among components of manufacturing systems – robots, machines, and workstation computers.
- In order to support fully automated, unmanned, overnight operation and the ability to make parts in small batches on demand, the manufacturing system grew more complex.
- Many sensors are needed for intelligent autonomous operation.
- Lack of a set of standardized sensor interfaces for system integration.

## Trends

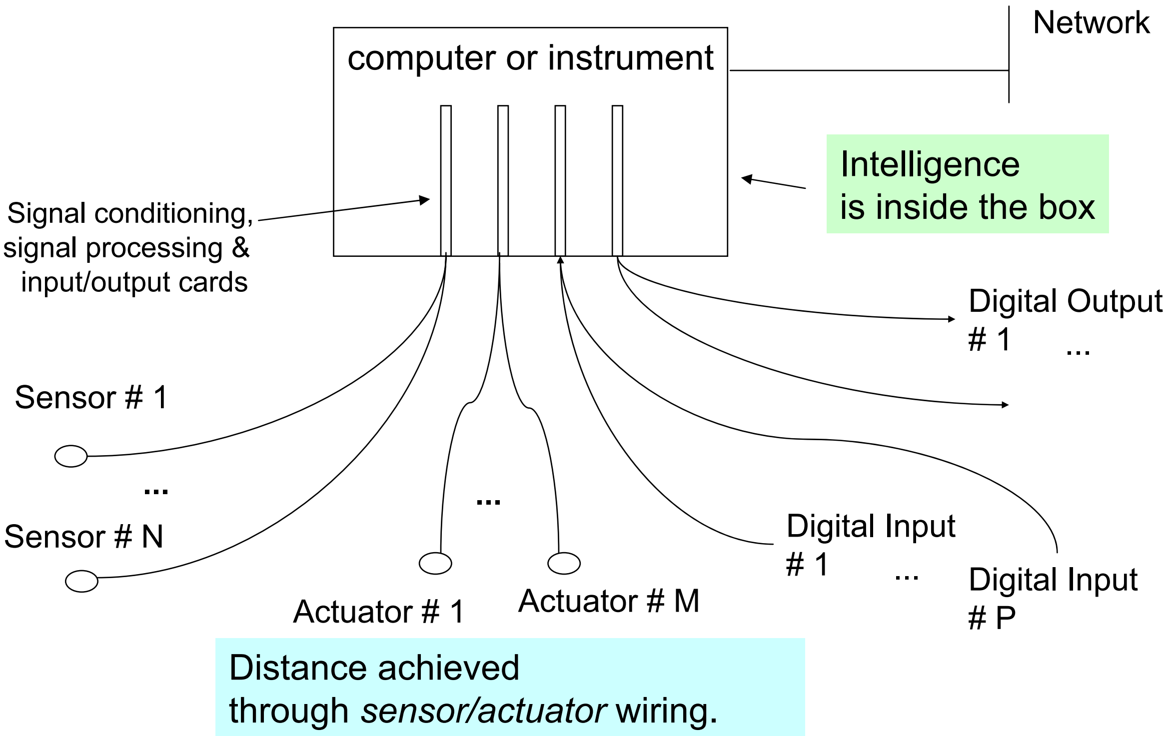
- Fast moving toward using networked, digital, and wireless communications for sensors, e.g.
  - Industry and manufacturers moving to use networked, wireless systems for sensor connectivity to enhance automation and to lower life cycle costs.
  - Military moving to use networked, wireless sensors to enhance automation, and improve condition-based maintenance.
  - Homeland security applications moving to use networked, wireless sensor systems for remote monitoring and situation awareness.
- All these applications are seeking open standard solutions.

## That is what IEEE 1451 is all about ?

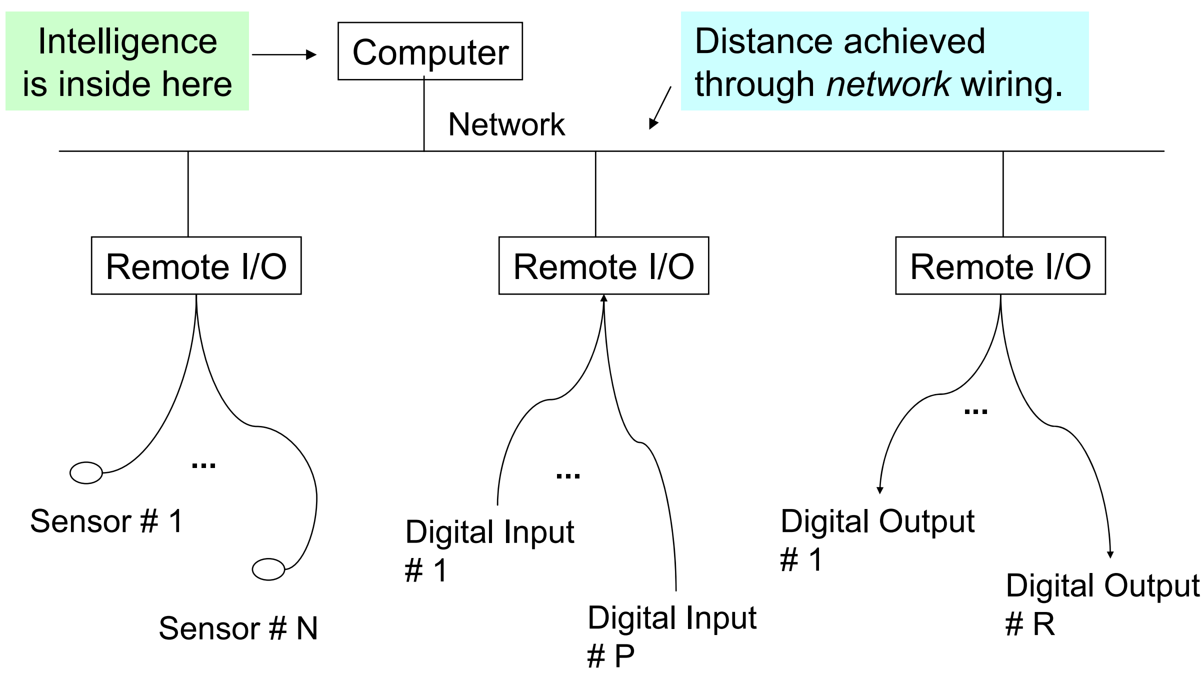
**Standard ways to connect sensors and actuators to networks and systems that facilitate**

***Interoperability***

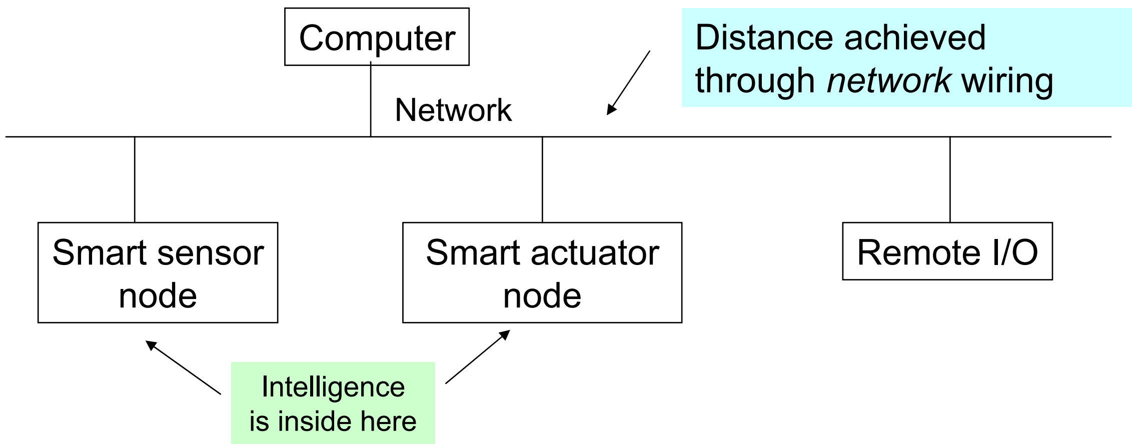
# Measurement & Control (M&C) System




# Distributed Measurement & Control System

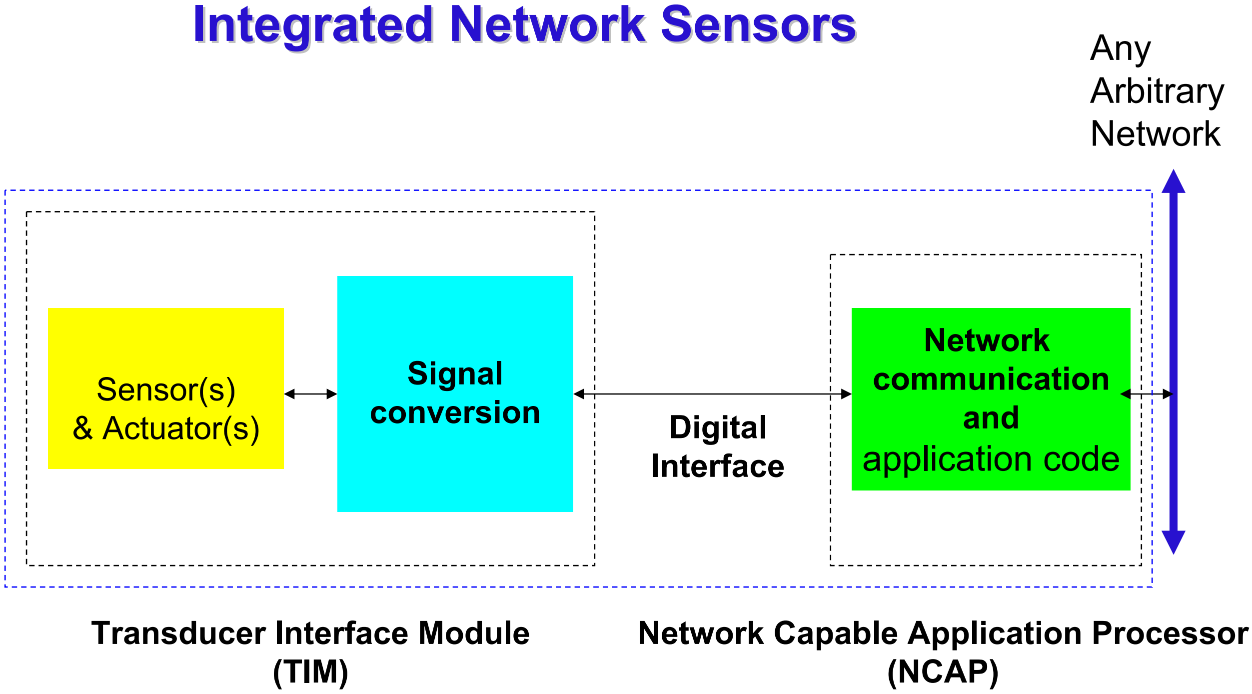


## Distributed Smart Sensor/Actuator System

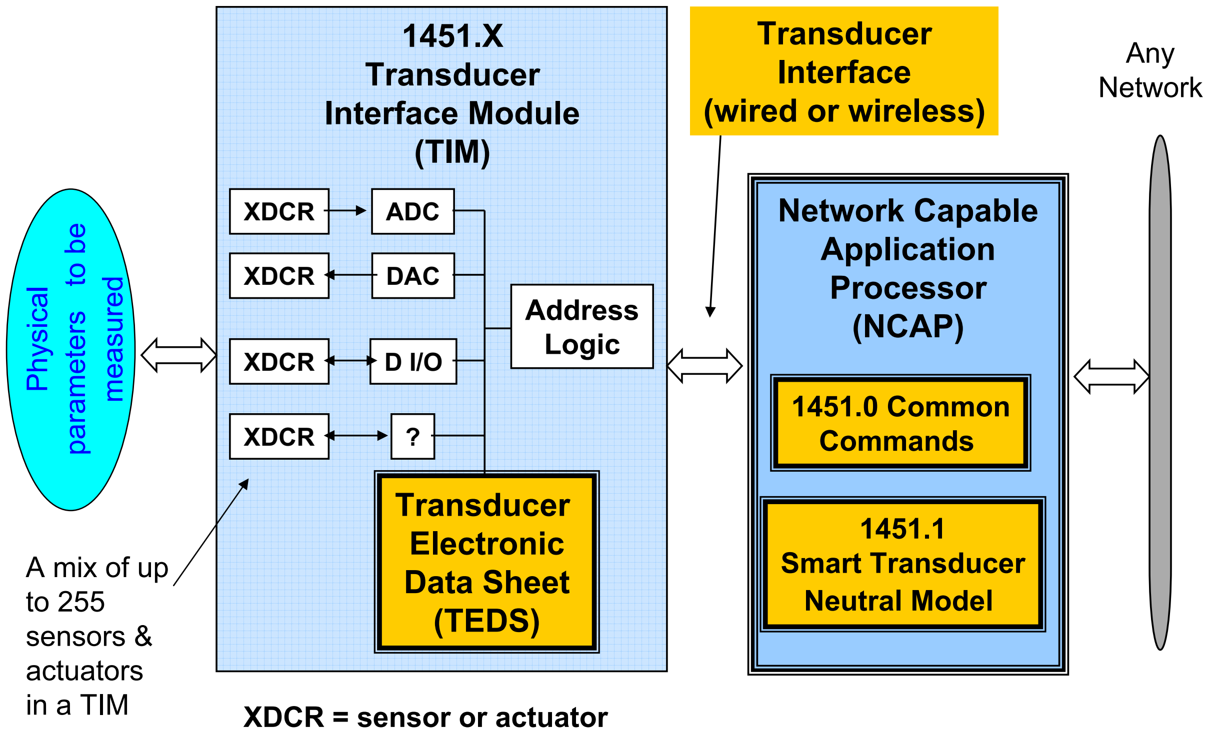



- Smart nodes are still networked.
- Intelligence is distributed.
- Role of computer changes

## Integrated Network Sensors

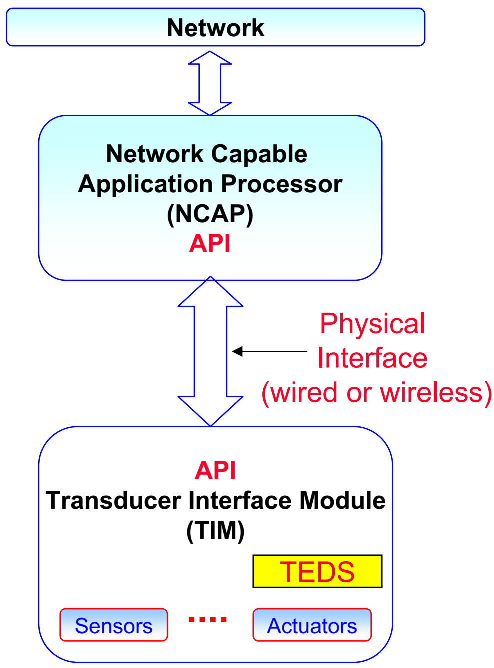


## IEEE 1451 Smart Transducer Interface System Approach

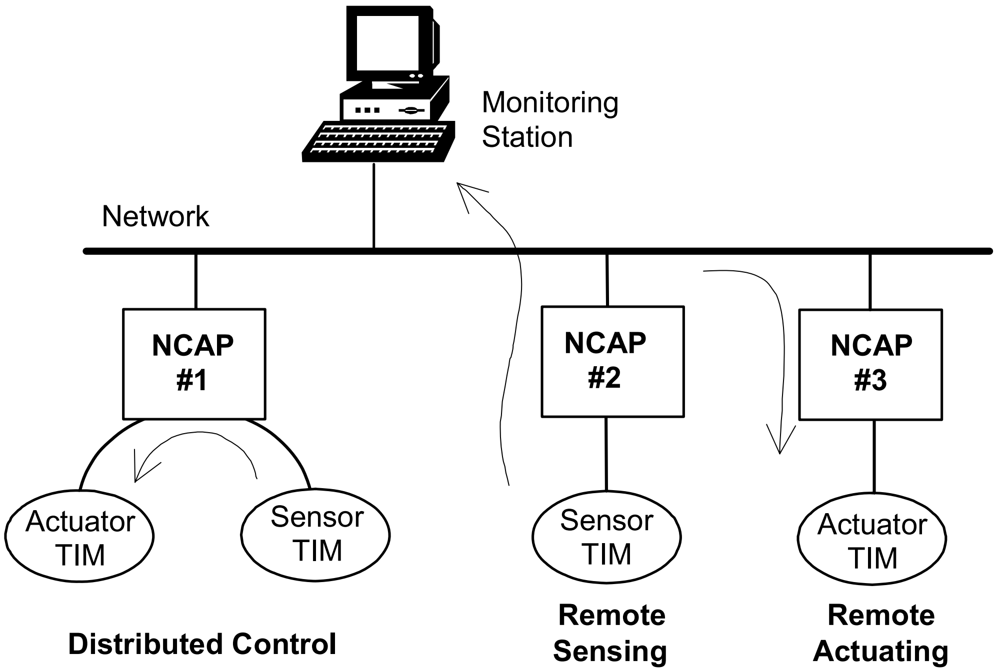



## IEEE 1451 -
## Smart Transducer Interface Standard

- An industry open sensor interface standard
- Objective is to provide a set of common interfaces for connecting transducers (sensors or actuators) to a network
- It supports wired and wireless connections
- The Transducer Electronic Data Sheet (TEDS) allows self-describing of sensors
- Provide a framework for a distributed system architecture



# Application Model of IEEE 1451-based Sensor Network




# Transducer Electronic Data Sheets (TEDS)

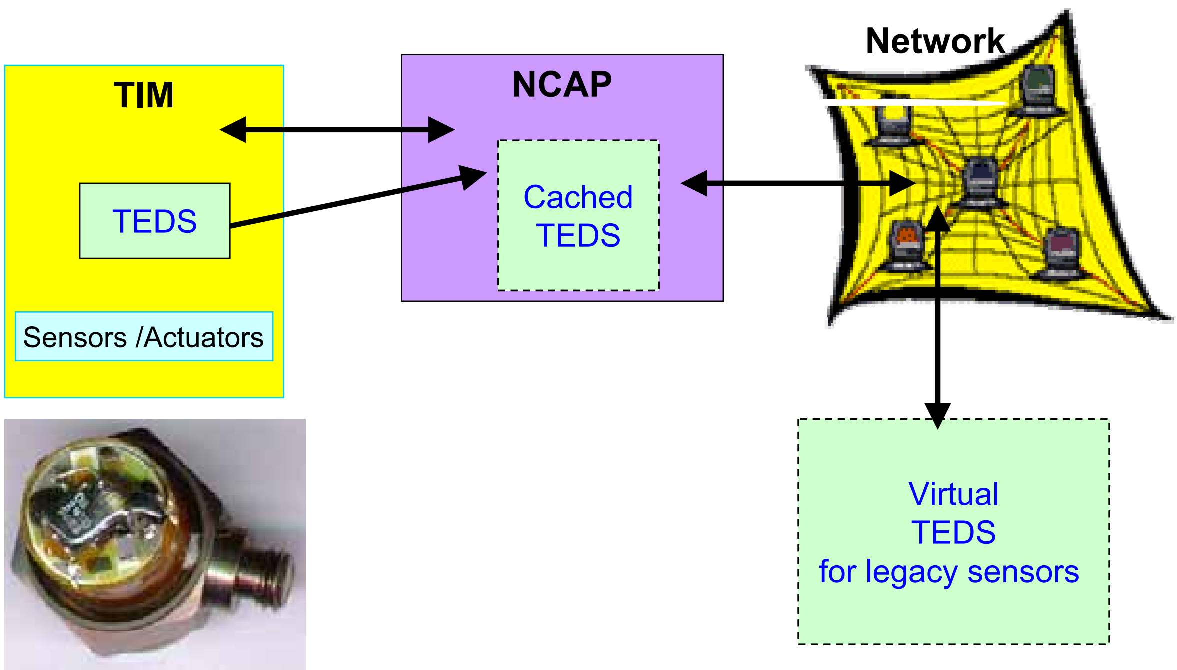
**A TEDS contains manufacture-related information about the sensors, whereas Bar Codes or RFID contains product ID and product item number.**

- **Meta-TEDS**
  - provides the characteristics of the entire Transducer Interface Module (TIM)
  - manufacturer's identification, model number, serial number, revision number, date code, product description, …
- **Transducer Channel TEDS**
  - provides the characteristics of a single transducer channel
  - lower and upper range limits, physical unit,…
- **Calibration TEDS**
  - provides the parameters to convert data to/from engineering units
  - last calibration date-time, multinomial coefficient, …
- **Geographic Location TEDS**
  - location of the sensor according to OGC (Open Geospatial Consortium) specifications

## Transducer Electronic Data Sheet (TEDS) – cont'd

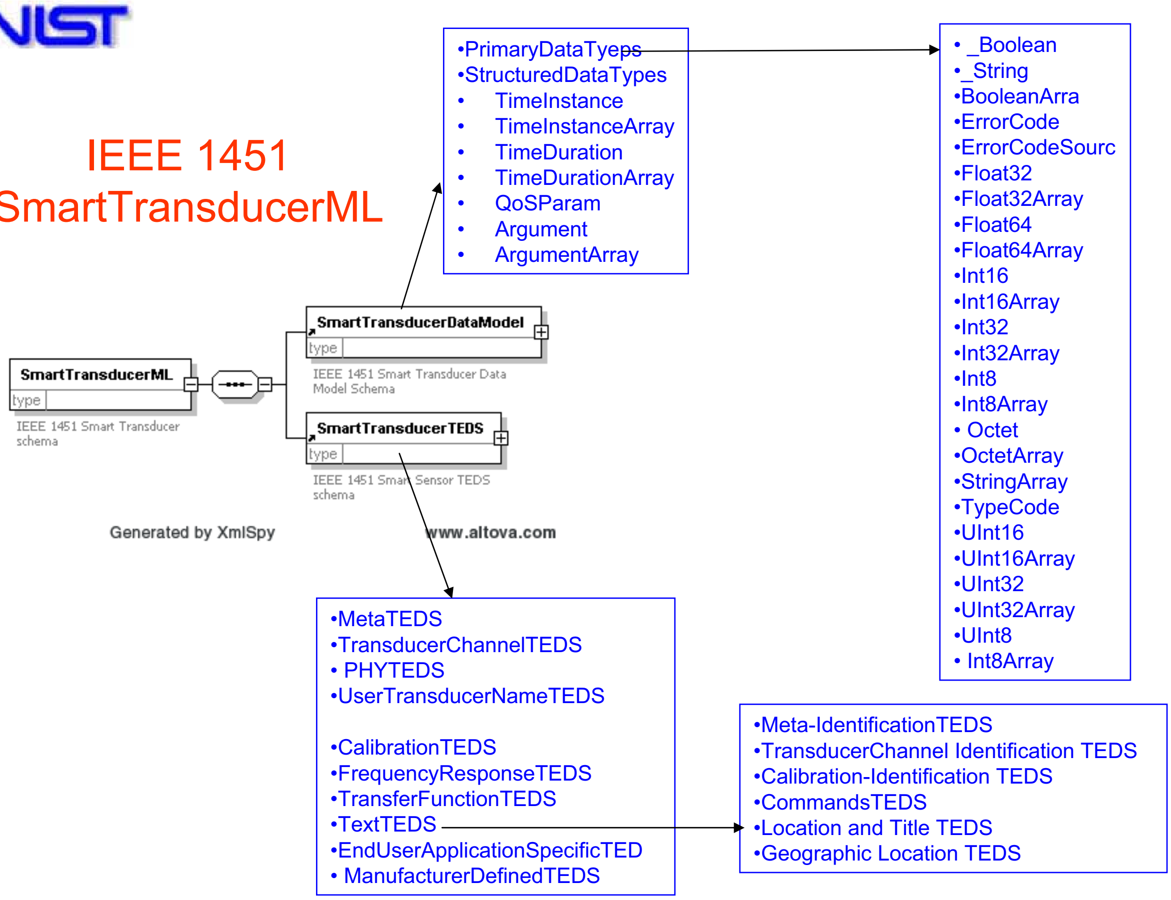
- **PHY TEDS**
  – defines parameters unique to the physical communications media
- **Manufacturer Defined TEDS**
  – allows the manufacturer to define additional features
- **Text-based TEDS**
  – allows manufacturer to provide textual information with the device.
  – written in XML (eXtensible Mark-up Language)
  – binary XML directory
  – followed by text-based XML TEDS data
- **End user application specific TEDS**
  – written by the user with user data.

## Where Do TEDS Live?

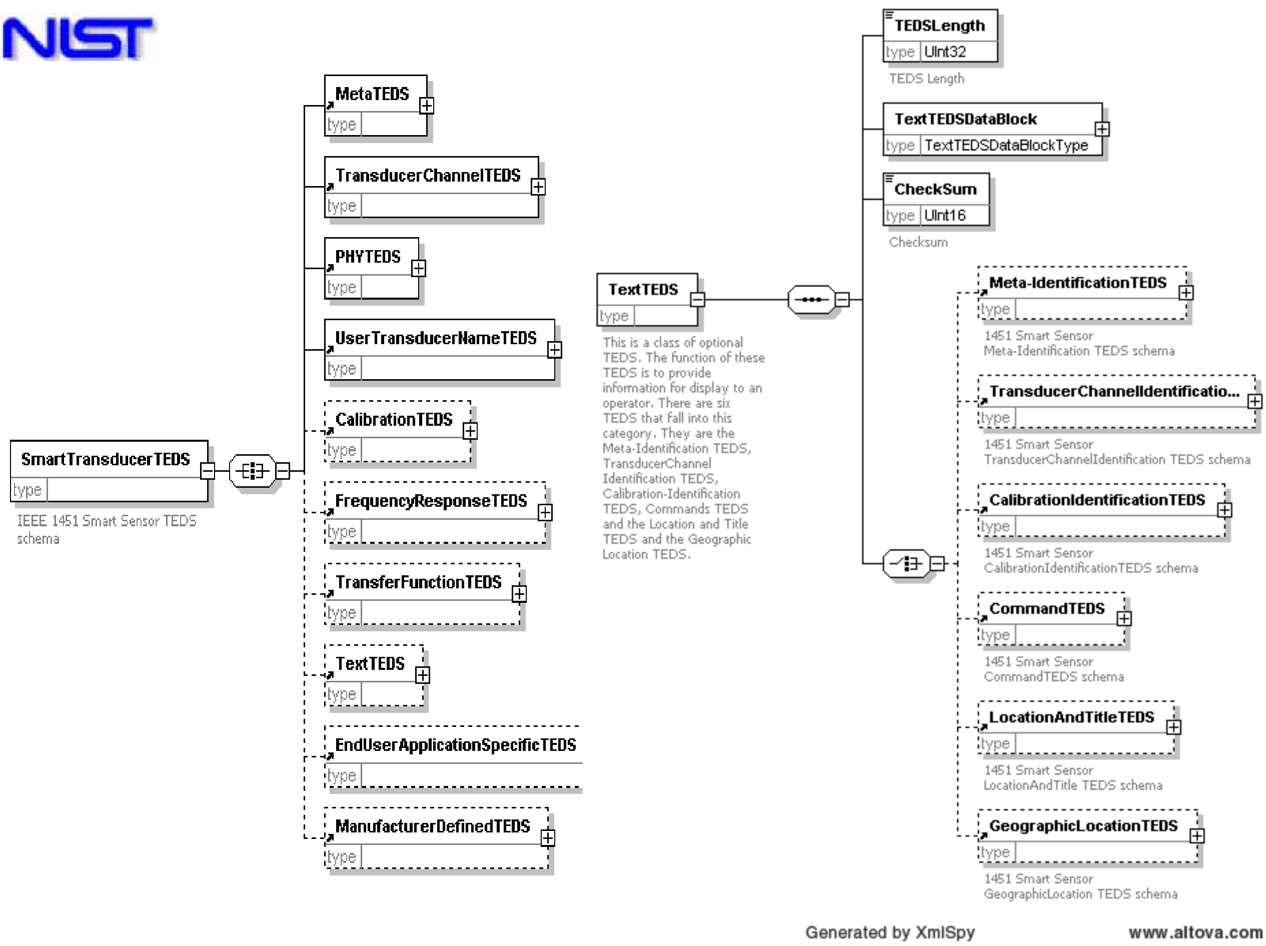


NIST

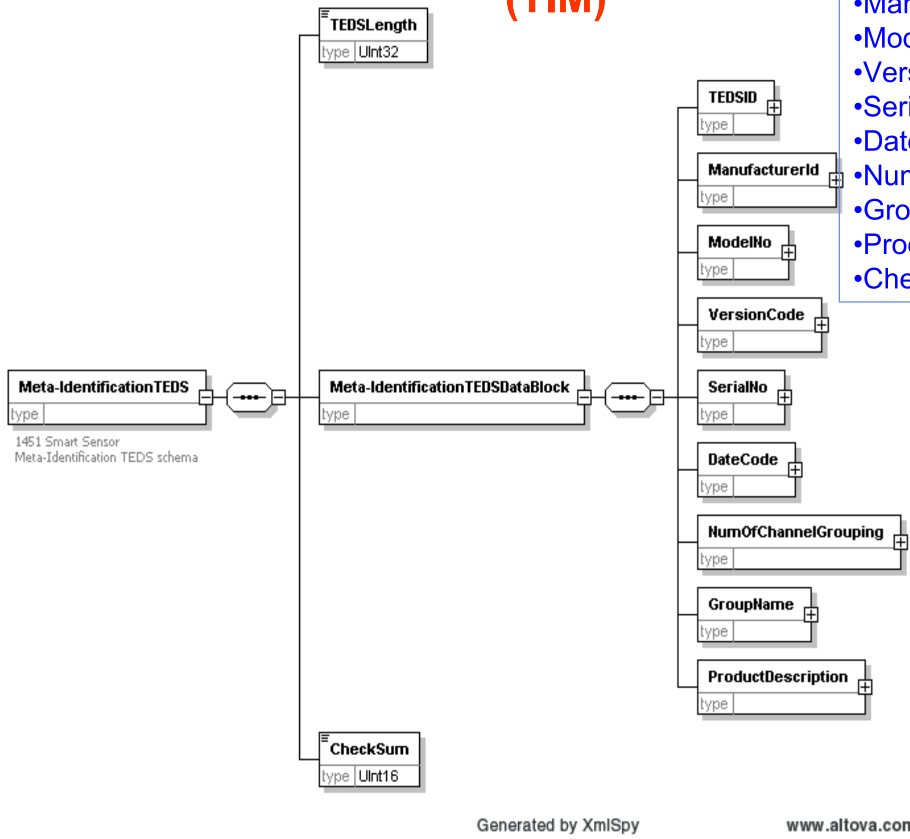
# IEEE 1451 SmartTransducerML




NIST



# Meta-IdentificationTEDS (TIM)

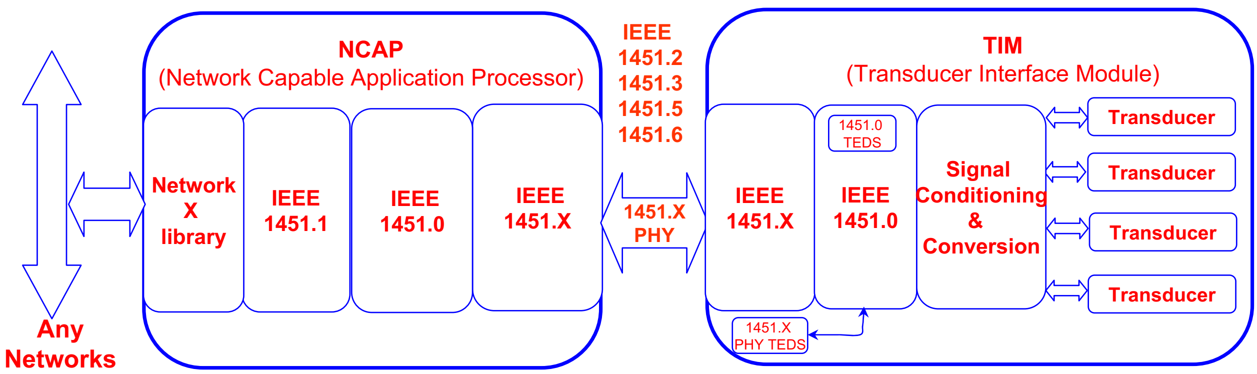



- TEDSLength
- TEDSID
- ManufacturerId
- ModelNo
- VersionCode
- SerialNo
- DateCode
- NumOfChannelGrouping
- GroupName
- ProductDescription
- CheckSum.

# Suite of IEEE 1451 Standards

IEEE 1451, a suite of Smart Transducer Interface Standards, describes a set of open, network-independent communication interfaces for connecting transducers (sensors or actuators) to microprocessors, instrumentation systems, and networks.

The key feature of these standards is the definition of Transducer Electronic Data Sheets (TEDS) that stores transducer identification, calibration, correction data, measurement range, and manufacture-related information, etc. TEDS are paper sensor data sheets in electronic form.



# IEEE 1451.0 Provides Common Functions to IEEE 1451.x

- **Hot Swap Capability**
- **Status Reporting**
- **Self-Test Capability**
- **Service Request Messaging**
- **Synchronous Data Acquisition from Arrays of Sensors**
- **Streaming Data Mode**

# IEEE 1451.0
# Common Transducer Electronic Data Sheets (TEDS)

- The IEEE 1451.0 TEDS are used to describe the entire TIM which includes the transducer (sensor and actuator), signal conditioner, and data converter.

- The TEDS is a memory device attached to the transducer, which stores transducer identification, calibration, correction data, measurement range, and manufacture-related information, etc.

- TEDS includes all kinds of transducer data, can be view as a transducer data model

- Four TEDS are specified as a minimum requirements and others are optional

Meta TEDS
TransducerChannel TEDS
User's transducer name TEDS
PHY TEDS
Calibration TEDS
Frequency Response TEDS
Transfer Function TEDS
Text based TEDS
End user application specific TEDS
Manufacturer-defined TEDS

Meta-Identification TEDS
TransducerChannel Identification TEDS
Calibration-Identification TEDS
Commands TEDS
Location and Title TEDS
Geographic Location TEDS

## IEEE P1451.0 Transducer Services

Transducer Services

The API that contains classes and interfaces for

- discovering registered TIMs,
- accessing transducer channels to make measurements or write actuators,
- managing TIM access,
- reading and writing TEDS.

## Sensor Web Services based on IEEE 1451

Sensor web services based on IEEE 1451.0 are a wrapper of IEEE 1451.0 services:

TimDiscoveryServices:

- reportTims( timIds )
- reportSensors( timId, channelIds, sensorNames )

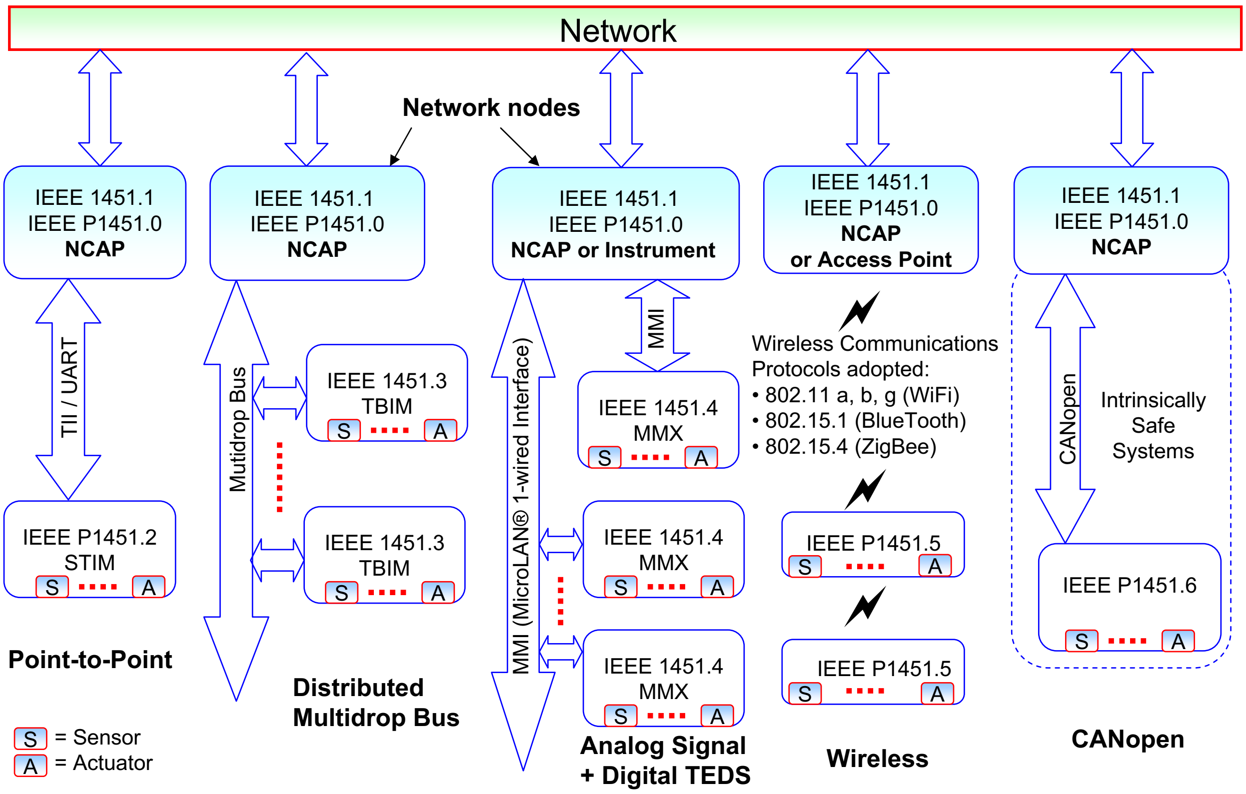
SensorObservationServices:

- readSensorData ( tim_id, channel_id, timeout, sampleMode, ArgumentArray result )
- writeSensorData ( tim_id, channel_id, timeout, sampleMode, value )

SensorTEDSServices:

- readSensorTEDS ( tim_id, tedsType, timeout, sampleMode, ArgumentArray result)
- writeSensorTEDS ( tim_id, tedsType, timeout, sampleMode, ArgumentArray value)

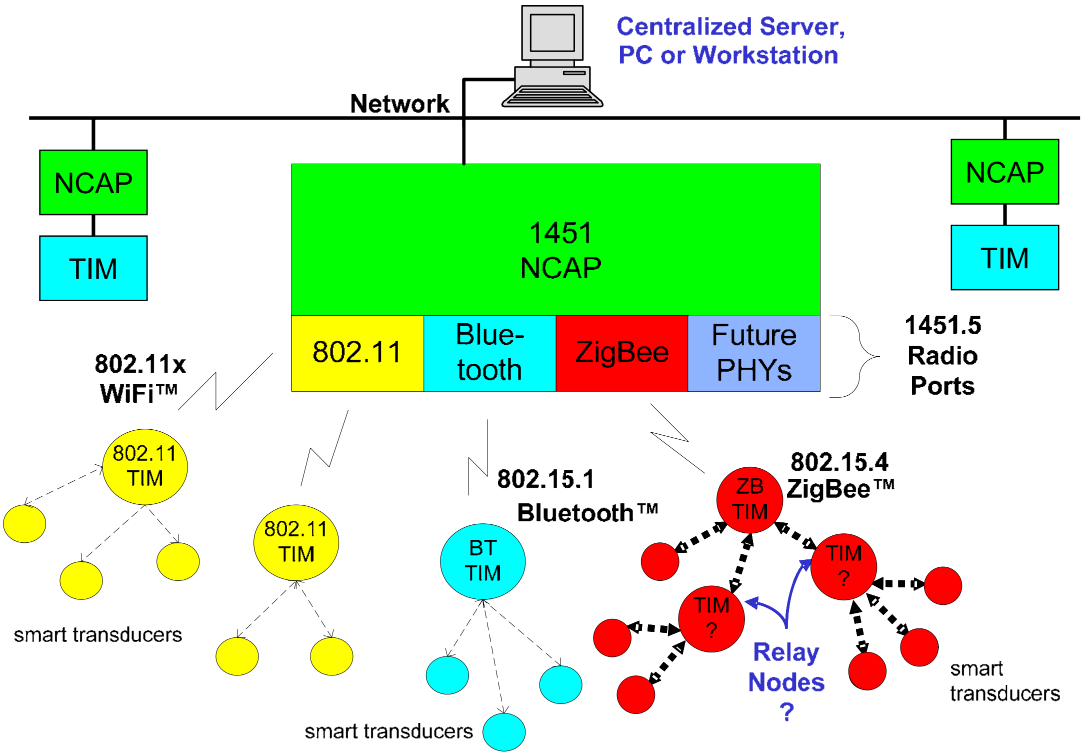
# Suite of IEEE 1451 Standards



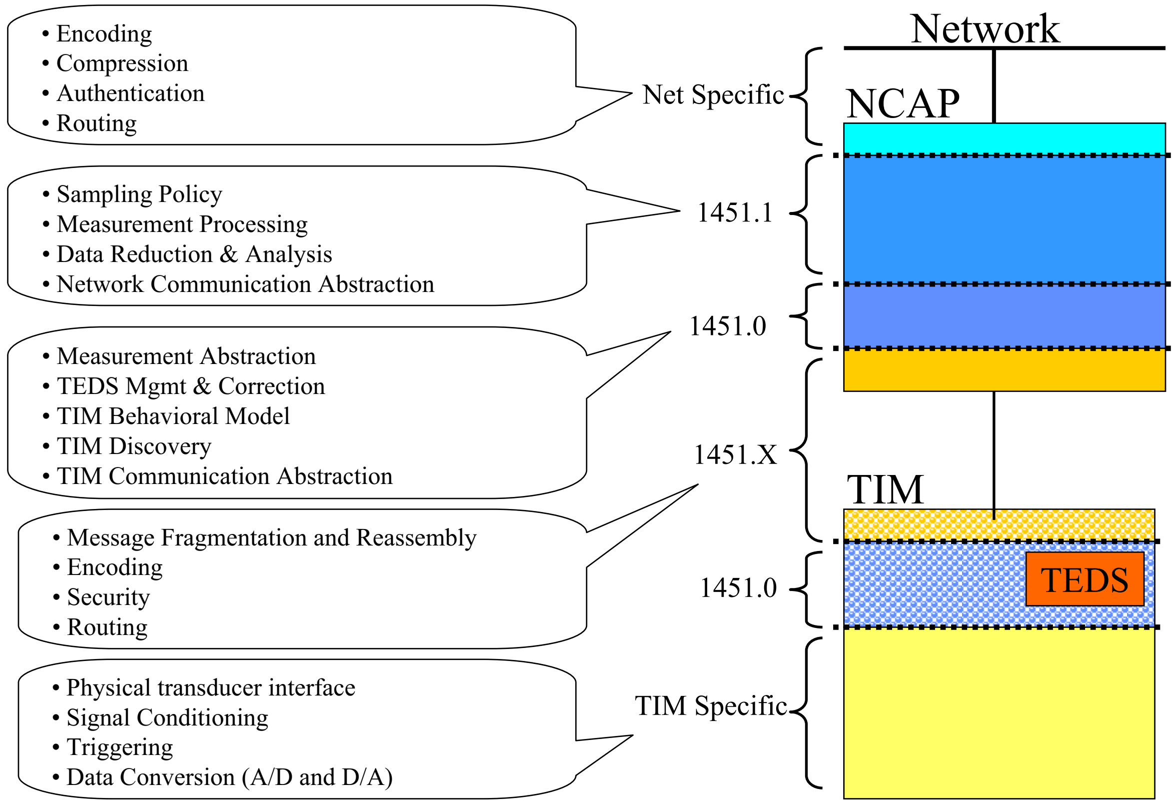

# IEEE 1451.2 Interface

| Line | Logic | Driven by | Function |
|---|---|---|---|
| DIN | Positive logic | NCAP | Address and data transport from NCAP to STIM |
| DOUT | Positive logic | STIM | Data transport from STIM to NCAP |
| DCLK | Positive edge | NCAP | Positive-going edge latches data on both DIN and DOUT |
| NIOE | Active low | NCAP | Signals that the data transport is active and delimits data transport framing |
| NTRIG | Negative edge | NCAP | Performs triggering function |
| NACK | Negative edge | STIM | Serves two functions:<br>—Trigger acknowledge<br>—Data transport acknowledge |
| NINT | Negative edge | STIM | Used by the STIM to request service from the NCAP |
| NSDET | Active low | STIM | Used by the NCAP to detect the presence of a STIM |
| POWER | N/A | NCAP | Nominal 5 V power supply |
| COMMON | N/A | NCAP | Signal common or ground |

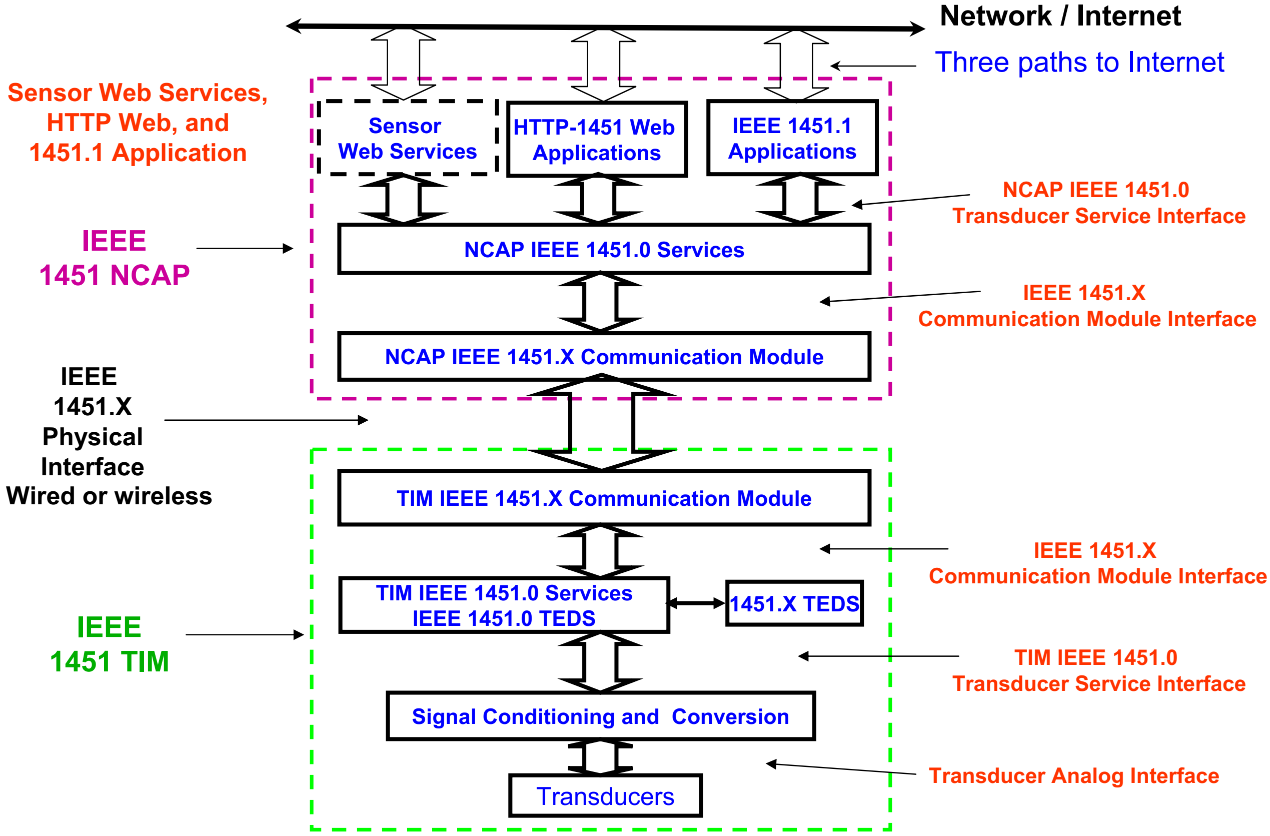
## Wireline and Wireless Sensors in a Network



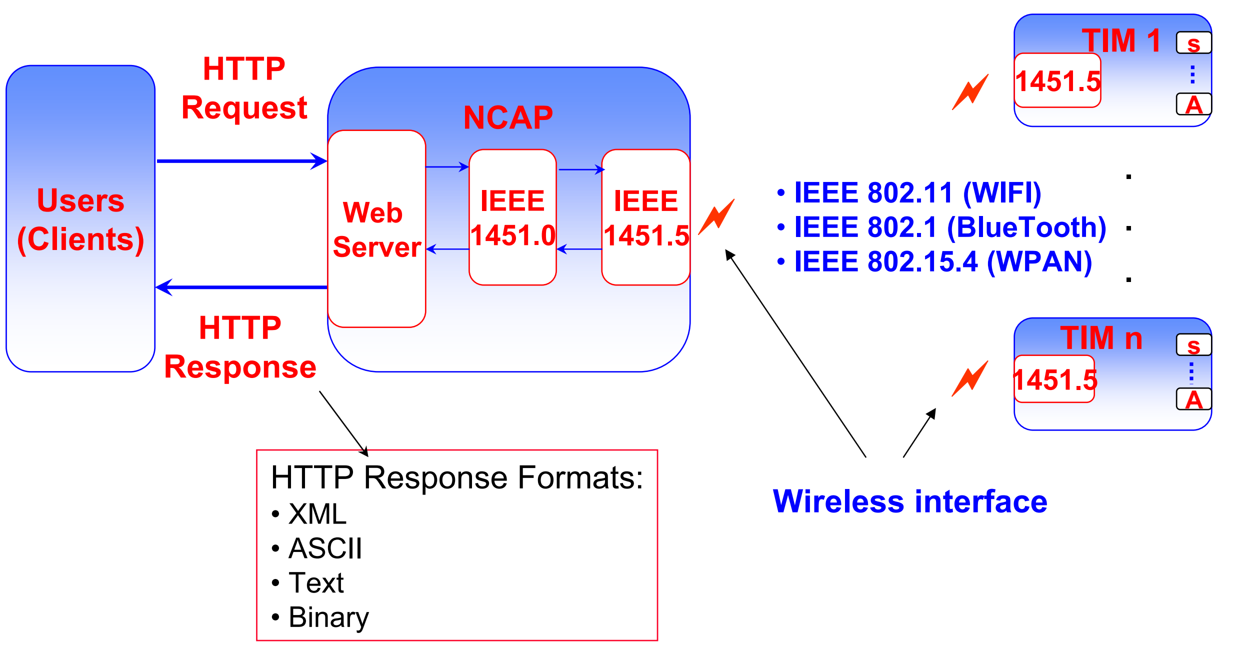

## IEEE 1451 Stack



# IEEE 1451 Stack - continued




# IEEE 1451 HTTP Web Application Integrating IEEE 1451.0 and 1451.5 Standards



# 1451 HTTP Request: Examples

- **ReadSensorData:**

```
GET (timId, transducerId, timeDuration, sampling Code,
outputFormat)
```

- **ReportSensorAlert:**

```
GET (timId, transducerId, descendingThreshhold
ascendingThreshhold, outputFormat)
```

- **ReportSensorAlert:**

```
GET (timId, transducerId, timeDuration, samplingCode,
outputFormat)
```

# IEEE 1451 HTTP API Summary

| | |
|---|---|
| Discovery API | •**TIMDiscovery**<br>•**TransducerDiscovery** |
| Transducer Access API<br>(Read & Write) | •**Read**<br>•**StartRead**<br>•**StartStream**<br>•**Write**<br>•**StartWrite** |
| TEDS Manager API | •**ReadTeds**<br>•**ReadRawTeds**<br>•**WriteTeds**<br>•**WriteRawTeds**<br>•**UpdateTedsCache** |
| Transducer Manager API | •**SendCommand**<br>•**StartCommand**<br>•**Trigger**<br>•**StartTrigger** |

# Status of the IEEE 1451 Standards

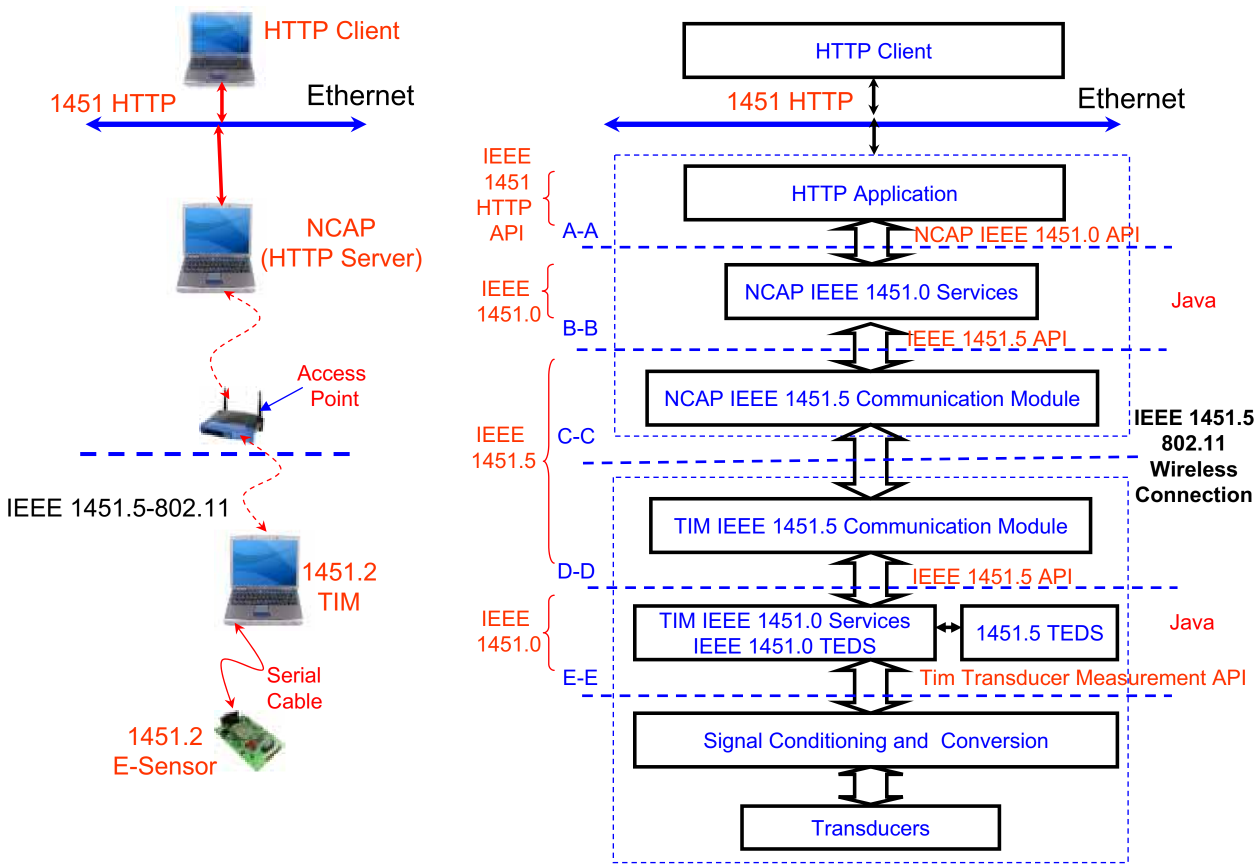
- **IEEE Std 1451.1-1999,** Network Capable Application Processor (NCAP) Information Model for smart transducers -- *Published standard, being revised*

- **IEEE P1451.0,** Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats -- *In balloting*

***Physical layers***

- **IEEE Std 1451.2-1997,** Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats -- *Published standard, being revised*
- **IEEE Std 1451.3-2003,** Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems -- *Published standard*
- **IEEE Std 1451.4-2004,** Mixed-mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats – *Published standard*
- **IEEE P1451.5,** Wireless Sensor Communication and Transducer Electronic Data Sheet (TEDS) Formats – *Submitted to IEEE for decision as a standard*
- **IEEE P1451.6,** A High-speed CANopen-based Transducer Network Interface – *In progress*

## A Reference Implementation of IEEE 1451 HTTP Web Application



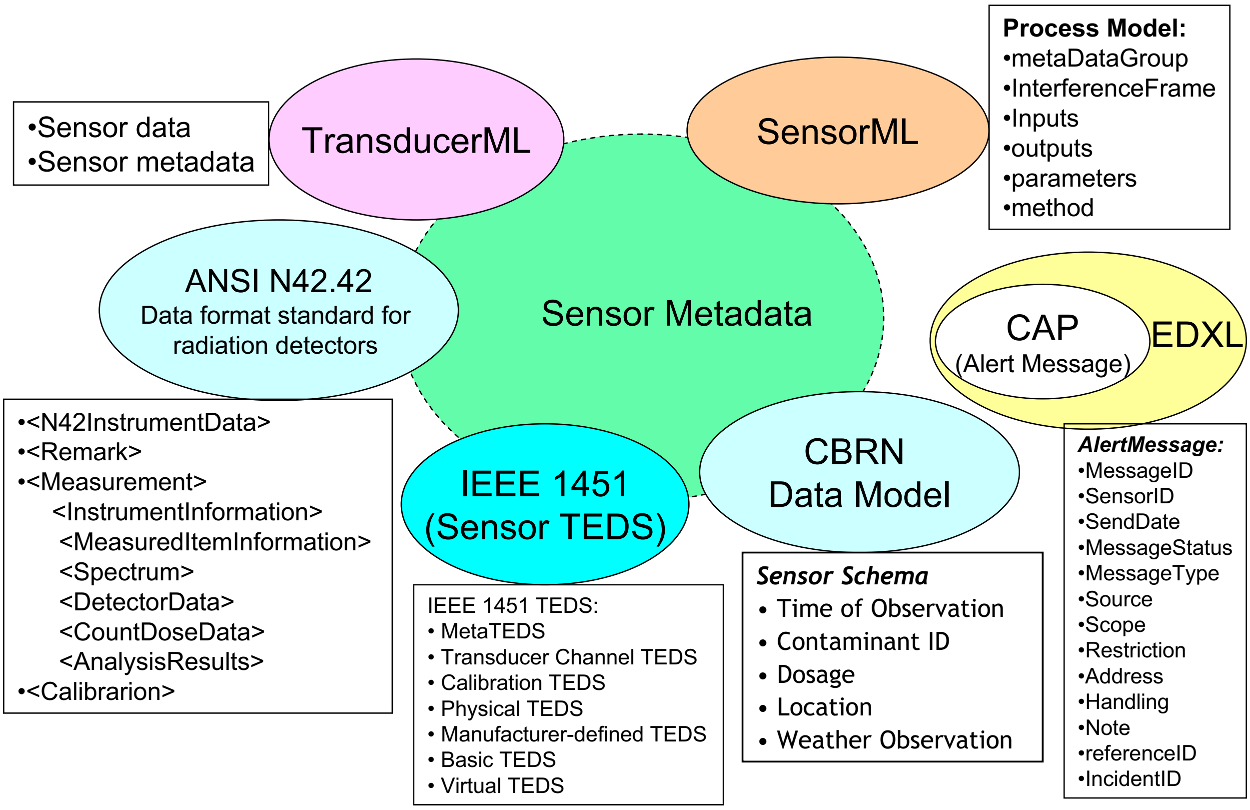
## Sensor Standards Harmonization Working Group (SSHWG)

- **Purposes**
  - Provide a forum for industry, academia, and government to exchange information and improve understanding of the various sensor-related standards programs being advanced by various standards development organizations.
  - Identify opportunities to frame the harmonization of sensor-related standards to meet the need of the community, current discussion – Ontology as a tool for harmonization and use cases.
- NIST agreed to lead the effort and organize quarterly working group meetings started Dec 2005.
- Next meeting is 2/27/07 at NIST.

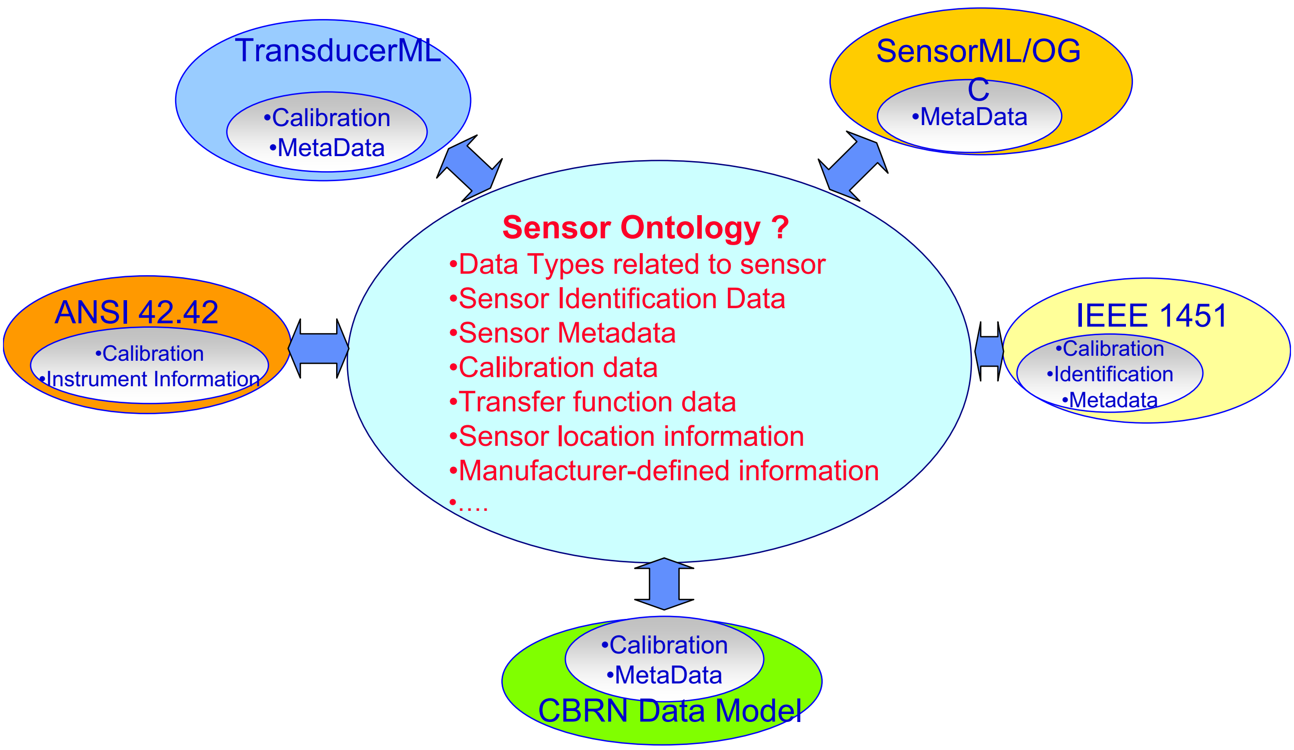
## Harmonization Issues and Opportunities

- Identify gaps and harmonization issues among existing and emerging standards.
- Develop Testbed and implementation demonstrations which are good ways to show that systems are functional and interoperable from lower-level sensor networks to high-level applications.
- Provide cooperating standards organizations with a common set of use cases, along with access to sensor assets for testing and prototyping.
- Ontology can make diverse and distributed data and information sources interoperate. Is ontology a tool for harmonizing sensor and related standards of interest?

# Sensor Standards Harmonization

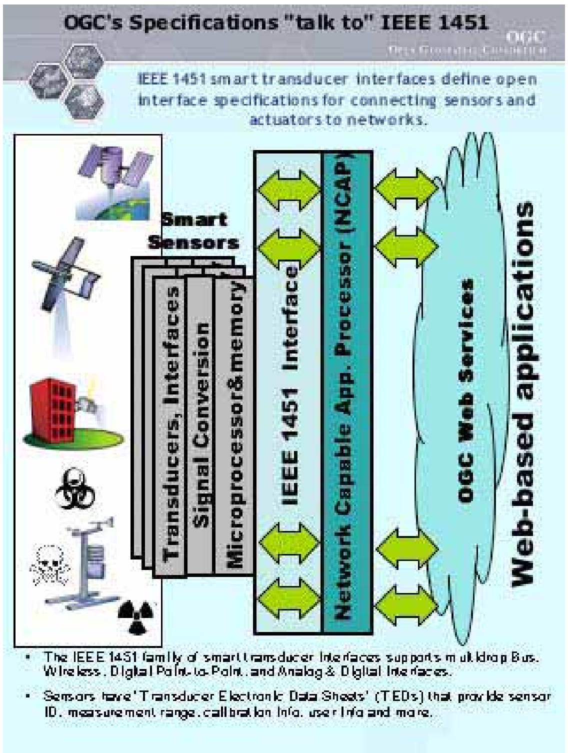



# Sensor Standard Harmonization Using Ontology ?



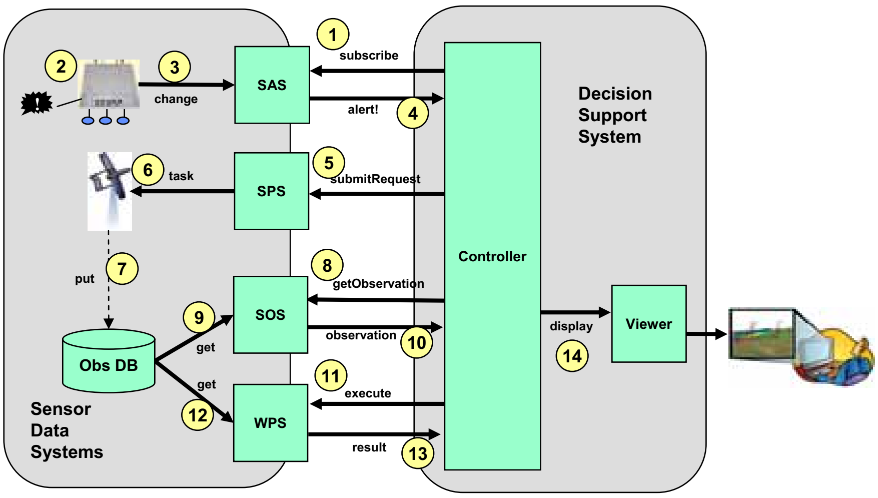
## Collaboration between IEEE and OGC

- Common interest in working together to advance open standards to enable transducer interconnection, discovery, access, integration, and usage within and across systems, networks, and enterprises.




## OGC OWS-4 Test Bed Scenario: Alert-driven Processing of Sensor Data

- Seek to <u>automate and shorten</u> the decision-action loop.
- Focus on:
  - automated <u>sensor management</u> and <u>sensor data processing</u> to produce actionable information for decision makers.

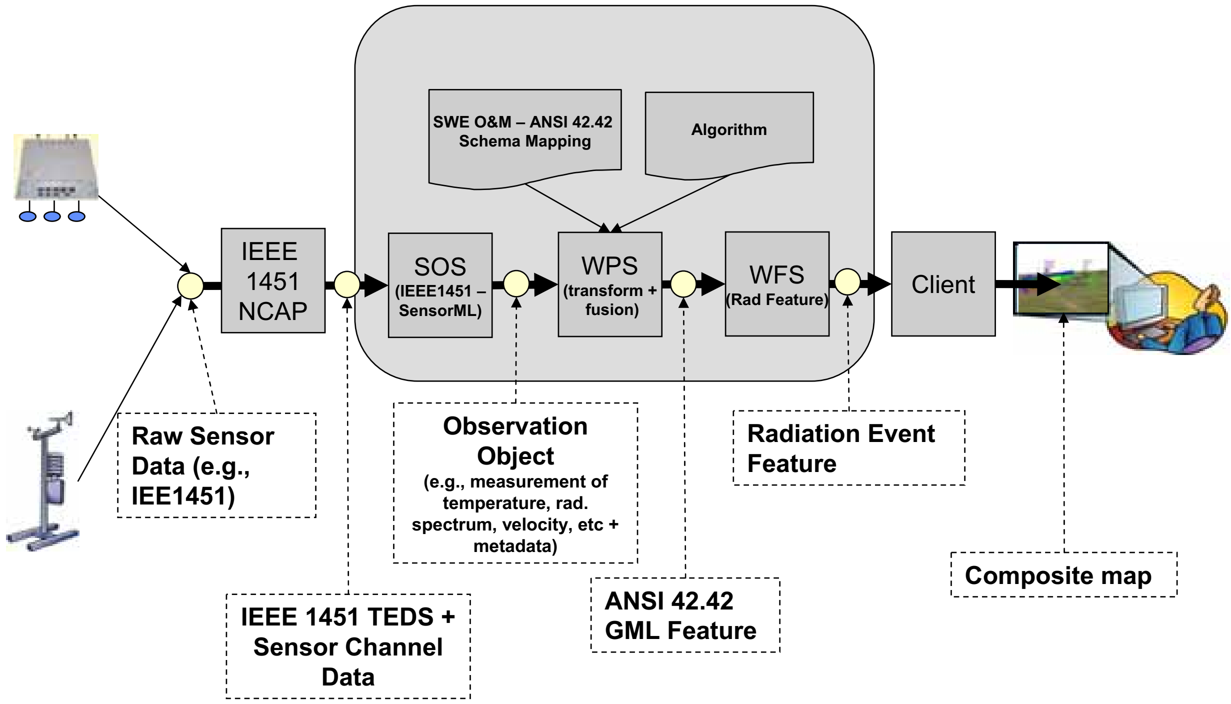



- **The approach must be <u>modular, extensible</u> and <u>standards-based</u> (i.e., not a "point solution")**

OGC – Open Geospatial Consortium

Courtesy of OGC

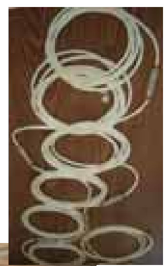
# OGC Integrates IEEE 1451 in its OWS-4 Test Bed Demonstration

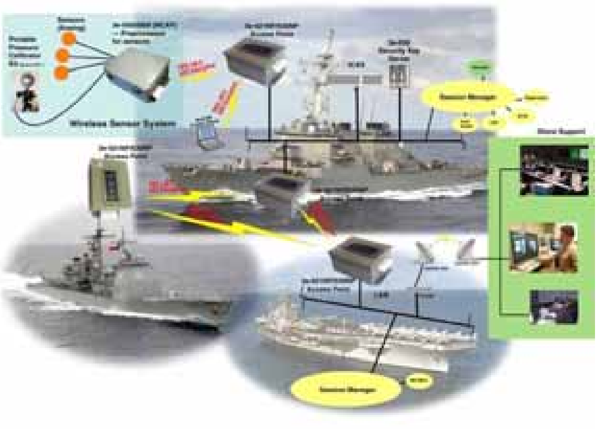


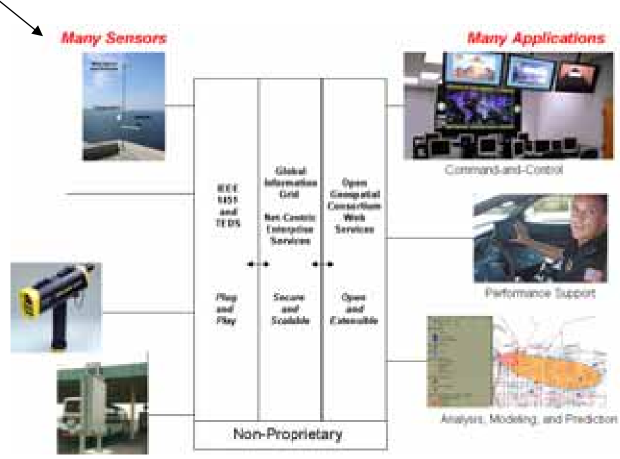

Courtesy of OGC

# Examples of IEEE 1451 Applications

- IEEE 1451.4 interface in LabView.
- IEEE 1451.2 in condition monitoring of casings for oil drilling.
- IEEE 1451 in SensorNet.
- IEEE 1451 sensor network in naval vessels for CBM.







## Benefits from IEEE 1451

### Sensor manufacturers

- Multiple products may be developed just by changing the TEDS.
- Standard physical interfaces
- Standard calibration specification

### System integrators

- Self-documenting hardware and software
- Systems that are easier to maintain
- Rapid transducer replacement
- Mechanism to store installation details

## Benefits from IEEE 1451 - cont'd

### Application software programmers

- Standard transducer model for control and data
- Same model for accessing a wide variety of measurements
- "Hooks" for synchronization, exceptions, simultaneous sampling
- Support for multiple languages

### End users

- Sensors that are easier to use; "you just plug them in".
- Analysis software that can automatically provide:
  - physical units
  - readings with significant digits
  - transducer specifications
  - installation details such as physical location and ID of transducer

## For More Information on IEEE TC9 Activities

- **Contact: Kang Lee at kang.lee@nist.gov**
- **IEEE standards can be purchased at http:// ieee.org**
- **IEEE 1451 websites:**
  **1451: http://ieee1451.nist.gov**
  **1451.0: http://grouper.ieee.org/groups/1451/4**
  **1451.4: http://grouper.ieee.org/groups/1451/4**
  **1451.5: http://grouper.ieee.org/groups/1451/5**
  **1451.6: http://grouper.ieee.org/groups/1451/6**
- **IEEE 1588 website: http://ieee1588.nist.gov,** Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems

# Super Distributed Objects

## - an example of implementation -

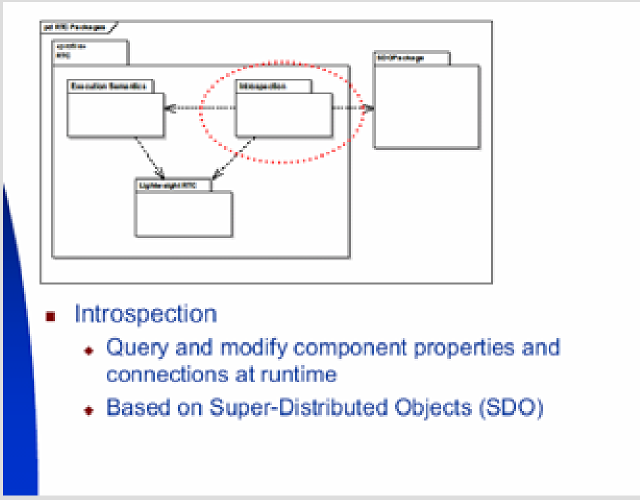
2006.12.5

Hitachi Ltd.



# 1. Background

## SDO in RTC




Ref. http://www.omg.org/docs/robotics/2006-09-01.pdf~2006-09-32.pdf

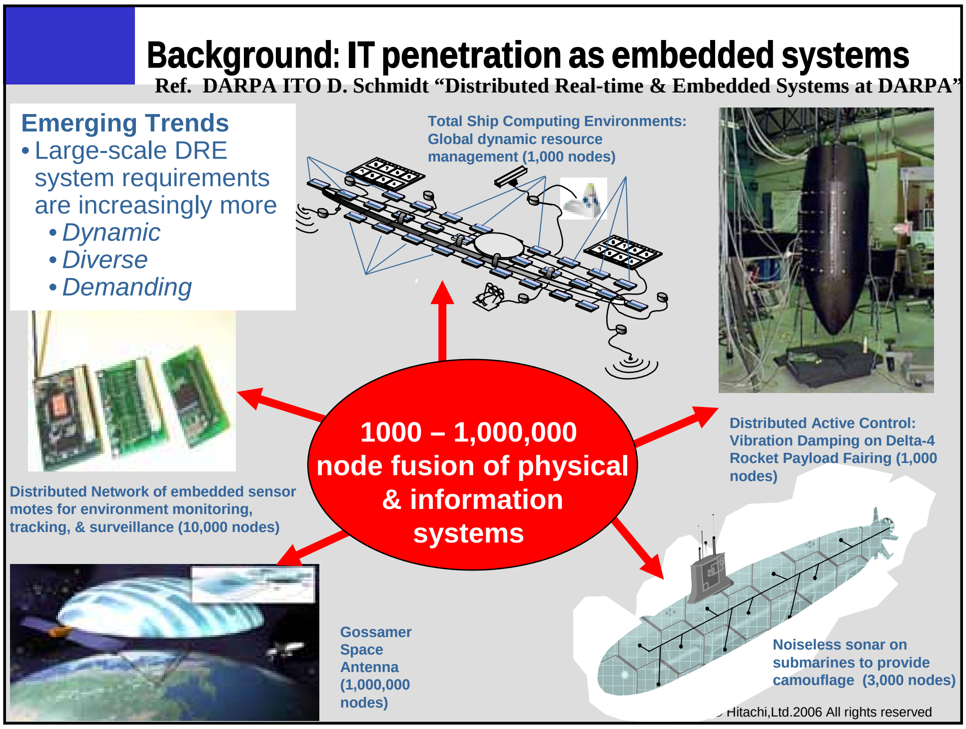


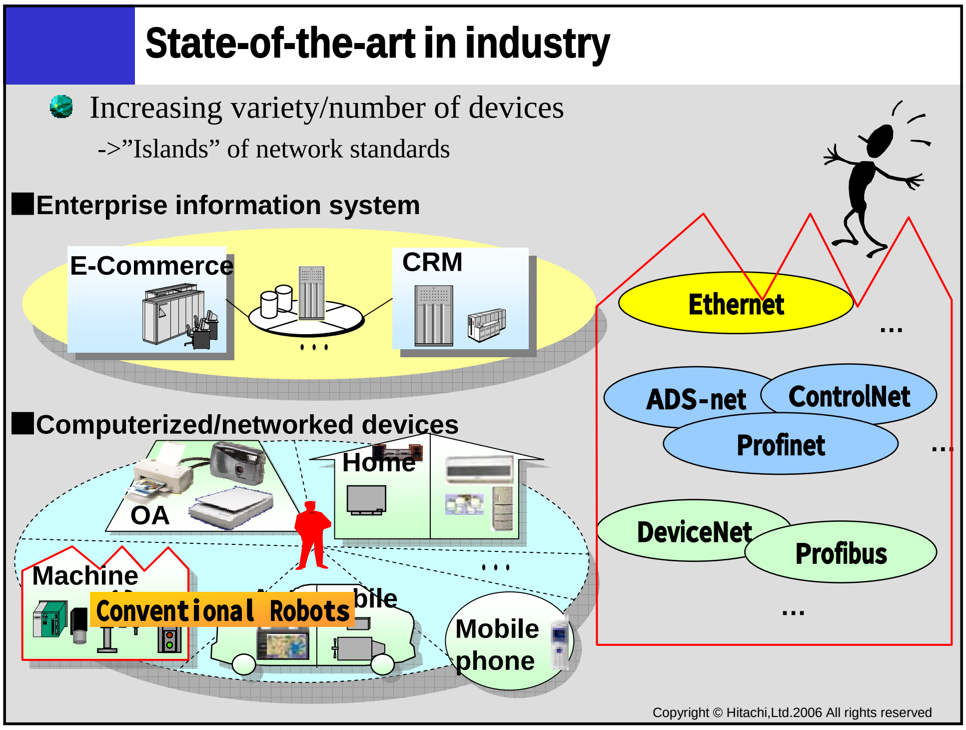
## History of SDO

- **<u>1999/11</u>** **Forum on Super Distributed System at OMG Cambridge meeting**
- **<u>2000/01</u>** **SDO SIG establishment**
- **<u>2000/06</u>** **RFI (Request for Information)**
- **<u>2001/07</u>** **Whitepaper**
- **<u>2002/02</u>** **RFP (Request for Proposal)**
- **<u>2003/03</u>** **Specification adoption**

  \- **PIM* and PSM* for SDO** (Platform Independent/Specific Model)
- **<u>2004/03</u>** **Finalization**
- **~ :Joint with Robotics activities**

SIG: Special Interest Group, RFI: Request for Information, RFP: Request for Proposal


Background: IT penetration as embedded systems
Ref. DARPA ITO D. Schmidt "Distributed Real-time & Embedded Systems at DARPA"
Emerging Trends
• Large-scale DRE system requirements are increasingly more
• Dynamic
• Diverse
• Demanding
Total Ship Computing Environments: Global dynamic resource management (1,000 nodes)
1000 – 1,000,000 node fusion of physical & information systems
Distributed Active Control: Vibration Damping on Delta-4 Rocket Payload Fairing (1,000 nodes)
Distributed Network of embedded sensor motes for environment monitoring, tracking, & surveillance (10,000 nodes)
Gossamer Space Antenna (1,000,000 nodes)
Noiseless sonar on submarines to provide camouflage (3,000 nodes)
Hitachi,Ltd.2006 All rights reserved

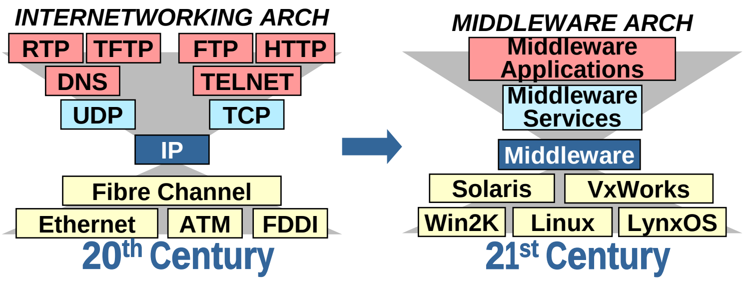

State-of-the-art in industry
Increasing variety/number of devices
->"Islands" of network standards
■Enterprise information system
E-Commerce
CRM
■Computerized/networked devices
OA
Home
Machine
Conventional Robots
bile
Mobile phone
Ethernet
ADS-net
ControlNet
Profinet
DeviceNet
Profibus
Copyright © Hitachi,Ltd.2006 All rights reserved

## Technical focus

# Standards used by middleware

We are creating the new generation of open ***DRE system middleware*** technologies to

1. ***Simultaneously control*** multiple QoS properties &
2. ***Improve software*** development quality, productivity, & assurability



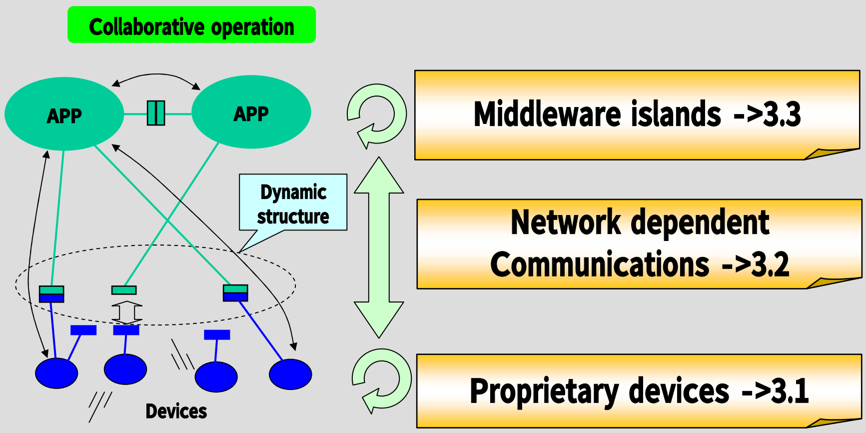

Ref. DARPA ITO D. Schmidt "Distributed Real-time & Embedded Systems at DARPA"



# 2. Overview of SDO and Examples of implementation

# Three levels of heterogeneity

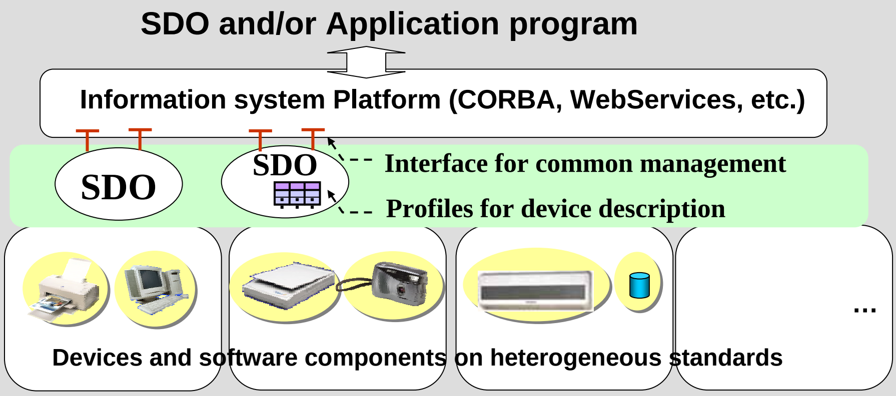
SDO based systems





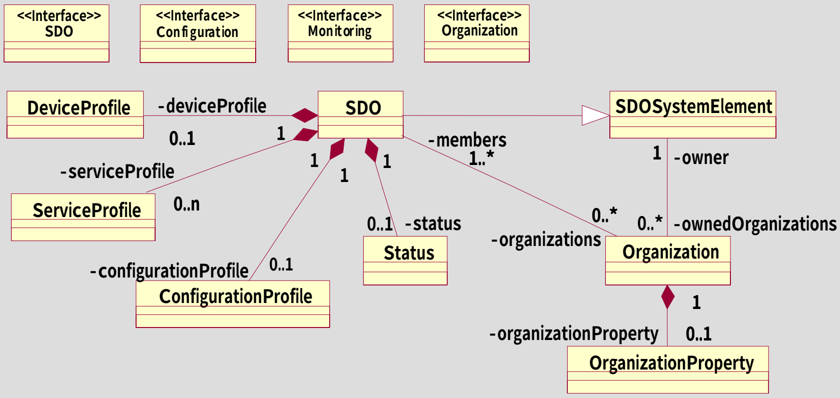

# 3.1 Description and Discovery: SDO model

- Meta model for real-world objects
  - Resource data model
    - Core data structure and named values for extensible properties
  - Interface
    - Common management functions



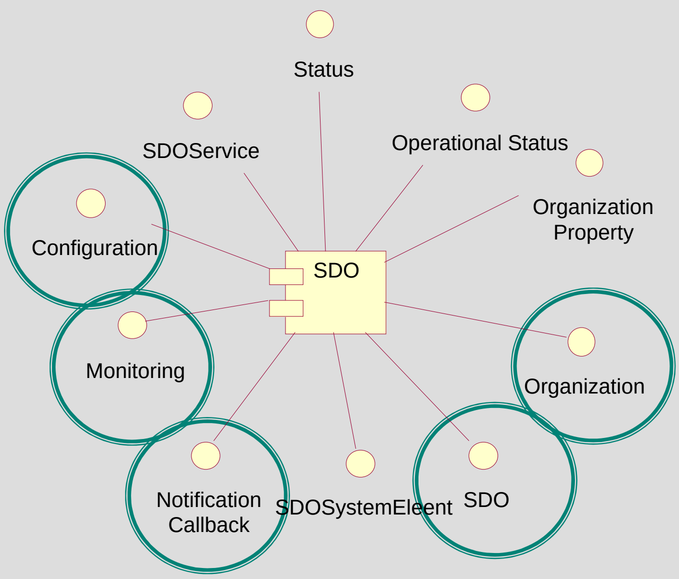
## 3.1 Description and Discovery: SDO model

- Resource data model
  - Profiles, Status, and Organization
- Interface
  - SDO interface
    - reference point to other interfaces
    - mandatory for all SDO
  - Configuration and Monitoring interface



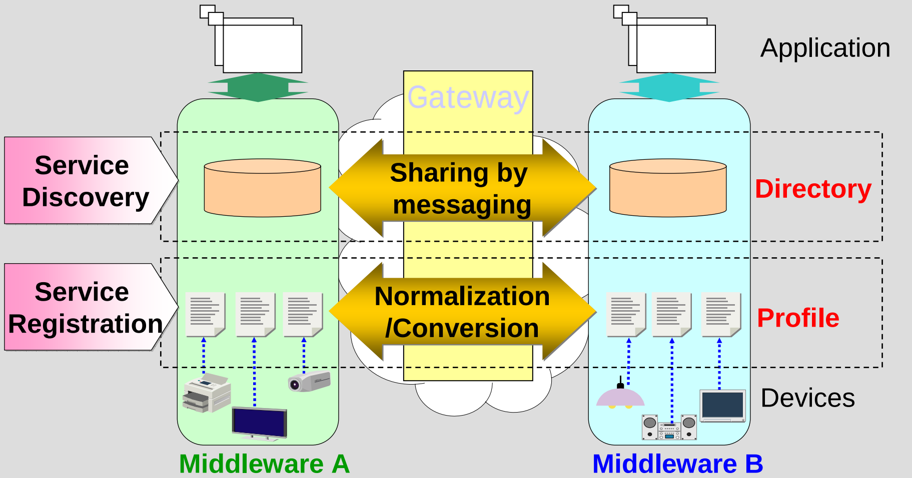



## 3.1 SDO Interfaces



## 3.3 Middleware interoperability: Approach

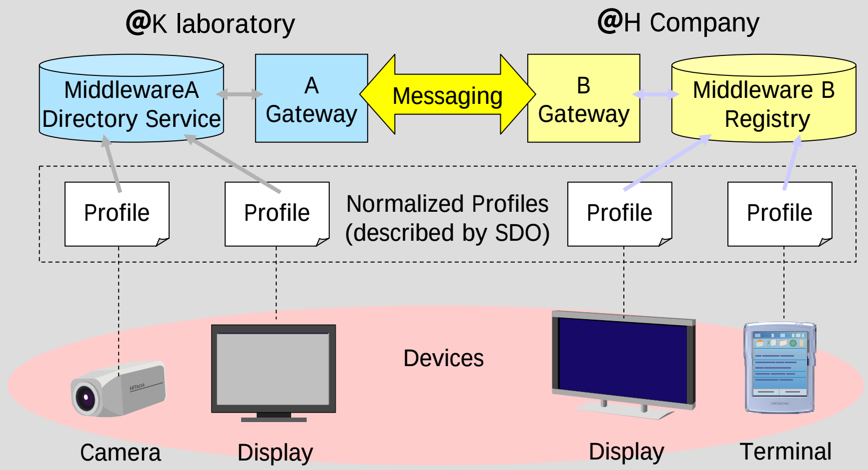
- 2 phased interoperation
  - Normalization of profiles, and Directory sharing.






## 3.3 Middleware interoperability: Profile normalization

- Profiles of each middleware are normalized and shared by using SDO

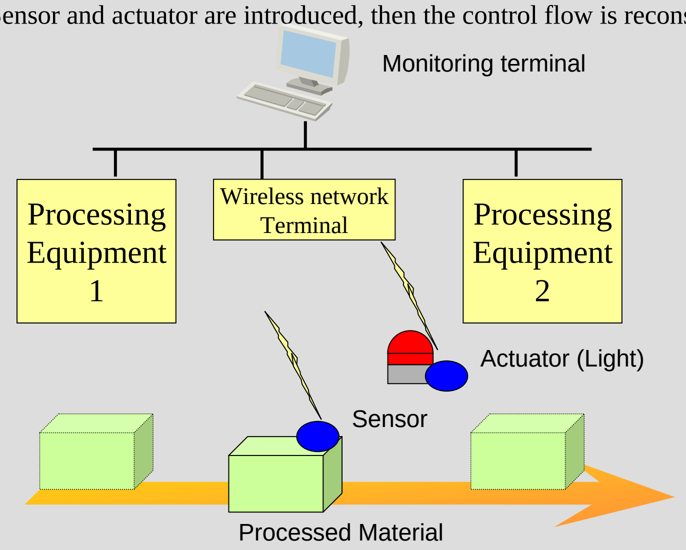


# 3. Demonstration



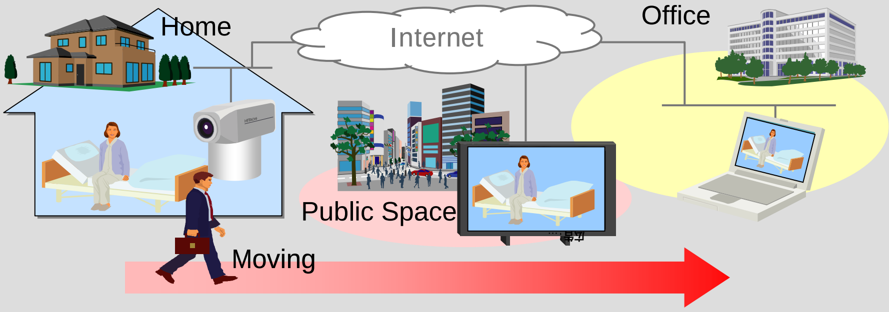
## Demonstration(1): Unified communication

- Demonstration of a control system
  - Beverage factory
- Sensor and actuator are introduced, then the control flow is reconstructed



## Demonstration(2): Middleware interoperability

- Home surveillance service:
  - Keeping a watch to cared people as follow-me service.



# Anybot Studio

2006. 12. 05

SoonHyuk Hong
Telecommunication R&D Center
SAMSUNG ELECTRONICS CO., LTD

## Contents

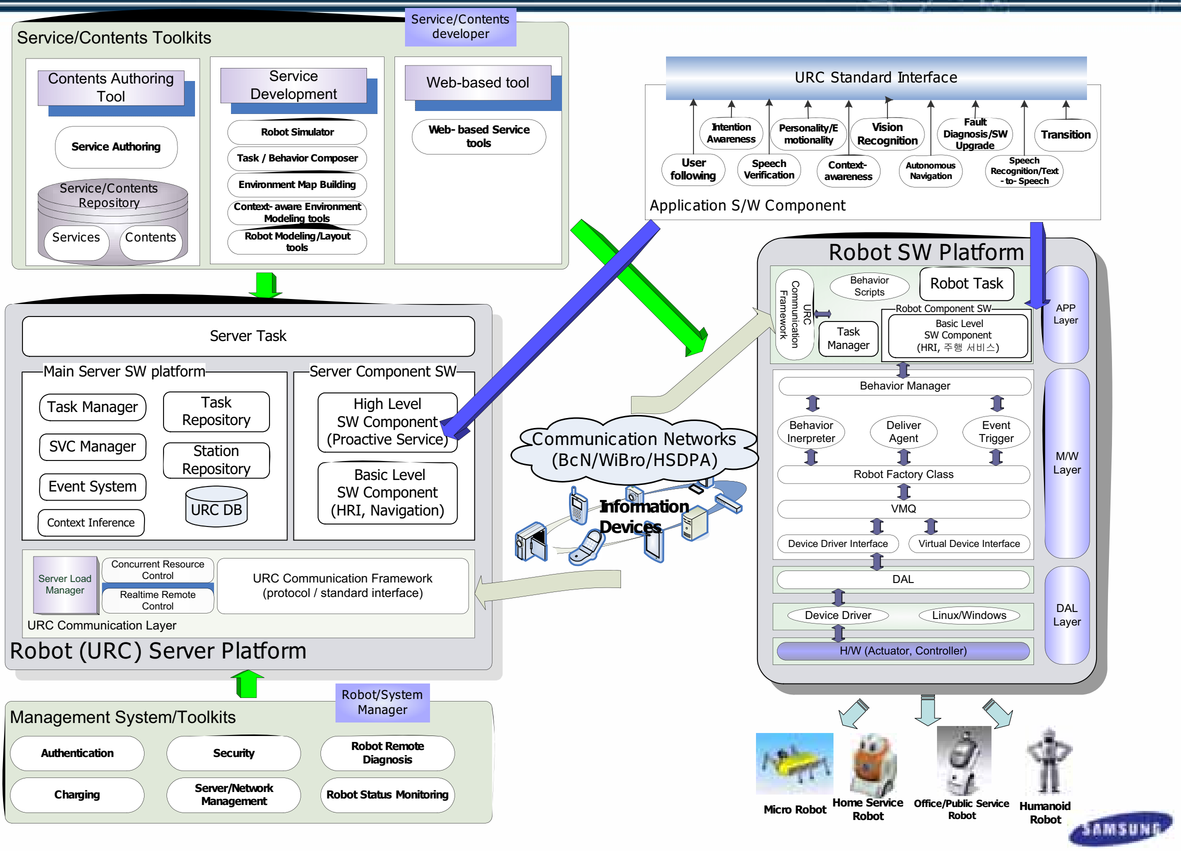
- Overview
  - Anybot Studio
- GRIS (General Robot Infra Structure)
  : Design concept, considerations
  - VMQ (Virtual Message Queue)
  - RFC (Robot Factory Class)
  - GRI (GRIS Remote Interface)
- Robot Simulator

# Anybot Studio

- **Anybot Studio**
  - **Robot S/W Total Solution for Network-based Robot System, URC (Ubiquitous Robotics Companion)**
- **It consists of**
  - ***Robot S/W Platform (GRIS)***
  - ***Robot Simulator***
  - **Server Platform (Middleware, Protocol)**
  - **Remote Management Tools**
  - **Contents Authoring Tools**



# Anybot Studio



# 1. Robot S/W Platform - GRIS

- **GRIS (General Robot Infra Structure)**

- **Design requirements**
  - **Rapid development of Robot Application**
  - **Various Robot H/W platforms**
  - **Easy porting to other platform**

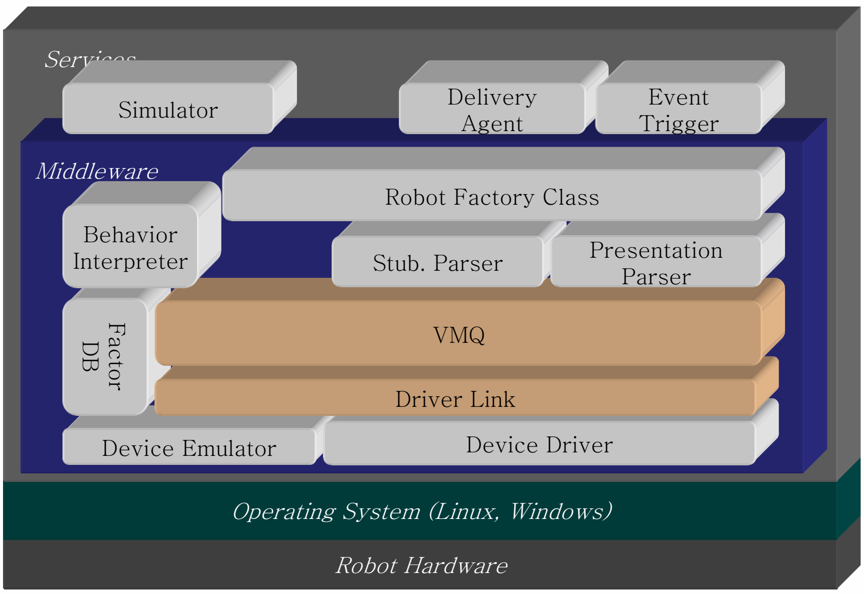
  - **Common robot interface (Robot APIs)**
  - **H/W independency (Middleware)**
  - **Multi OS platform (Linux, Windows)**

- **It provides**
  - ***VMQ (Virtual Message Queue)***
  - ***RFC (Robot Factory Class)***
  - ***GRI (GRIS Remote Interface)***
  - **BI (Behavior Interpreter)**
  - **Multi-OS Wrapper**



# GRIS

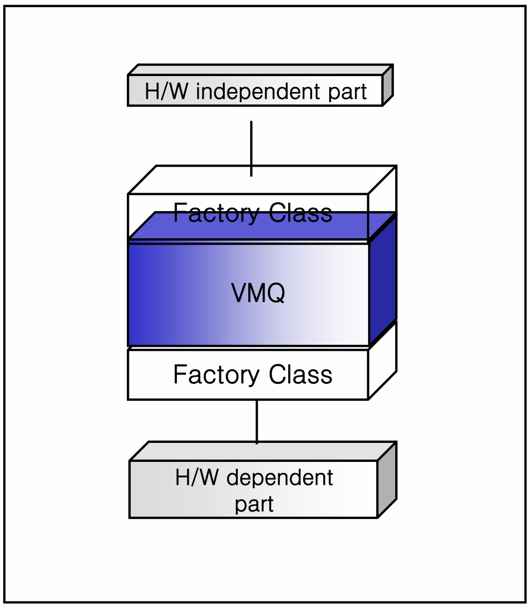
- **Construct a Robot S/W Platform**
  - **VMQ (Virtual Message Queue)**
    - **Robot Middleware for dividing application from H/W dependent Library**
    - **Supports service call and object call**
    - **Exchange data between Front-End and Back-End Interface**
  - **Robot Factory Class & GRI**




- **Design Issues**
  - **Portability**
  - **H/W Independency**
  - **Device Abstraction**
  - **Extensibility**

# GRIS

## GRIS



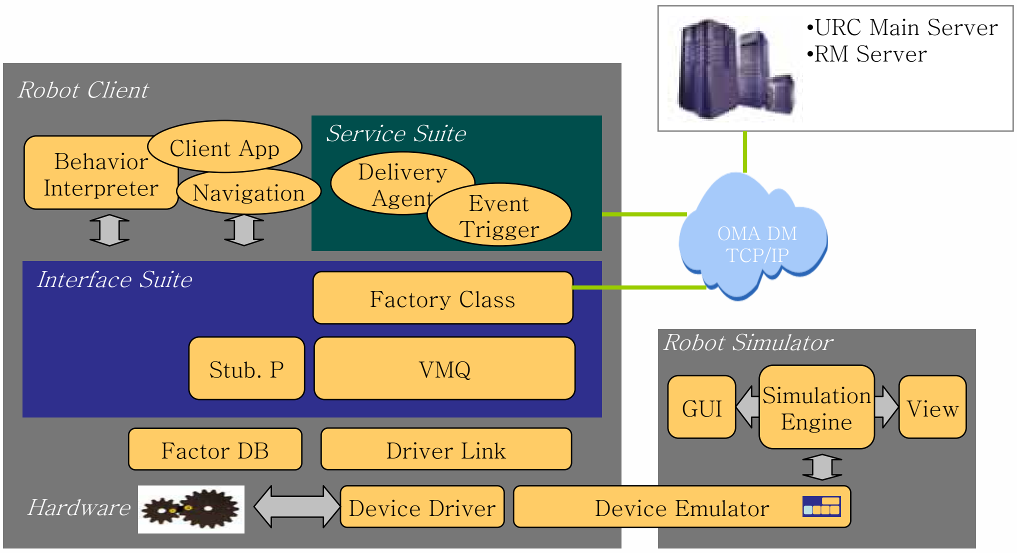

- VMQ manages exchanged data btw application and H/W devices
- VMQ supports function calls through Up/Down Interface
- Define Robot Factory Class for Up/Down Interface
- Provide extensibility
  - Adding new RFC APIs by GRI
- Additional S/W modules for managing robot resources

SAMSUNG



# System Configuration

- Robot S/W Platform, GRIS
- Applications
  - URC Client, Navigation, Service Suite
- Robot Simulator
- Servers
  - URC Main Server, RM (Remote Management) Server




SAMSUNG

# RFC (Robot Factory Class)

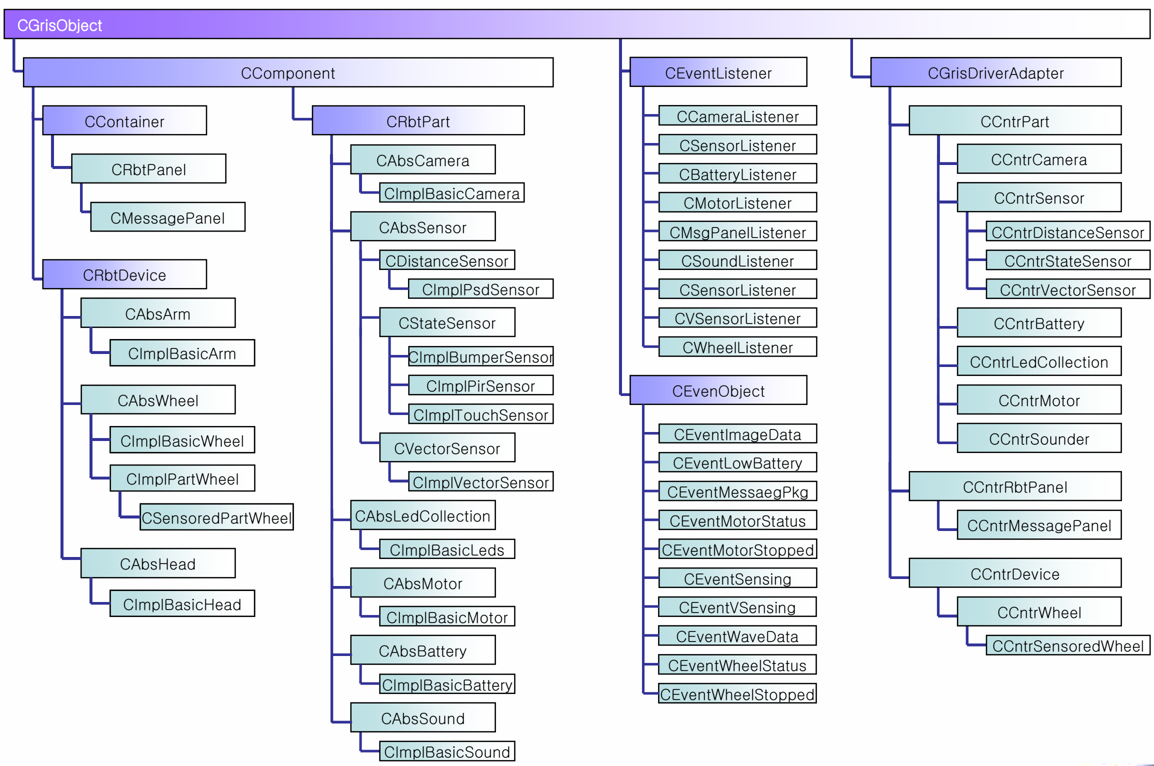
- An generalized/abstracted classes for controlling robot devices and sensing

- Design requirements
  - Provide abstracted class hierarchy considered robot H/W device components
  - Provide common robot functions which can be applied to various robots commonly
    - RFC should be a common model to adopt any kind of robot devices
  - Provide Up/Down Interface for function call and data exchange
  - Provide event-driven architecture for application



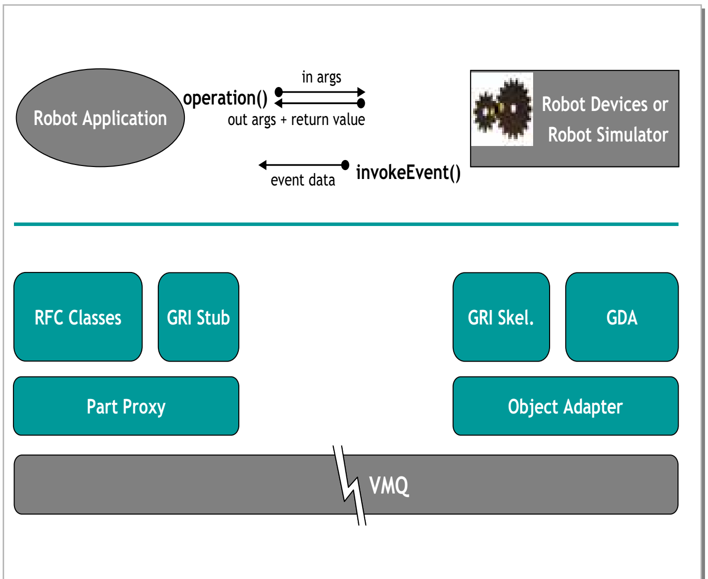
# RFC (Robot Factory Class)

❑ Robot Factory Class Hierarchy



# RFC (Robot Factory Class)

❑ RFC components



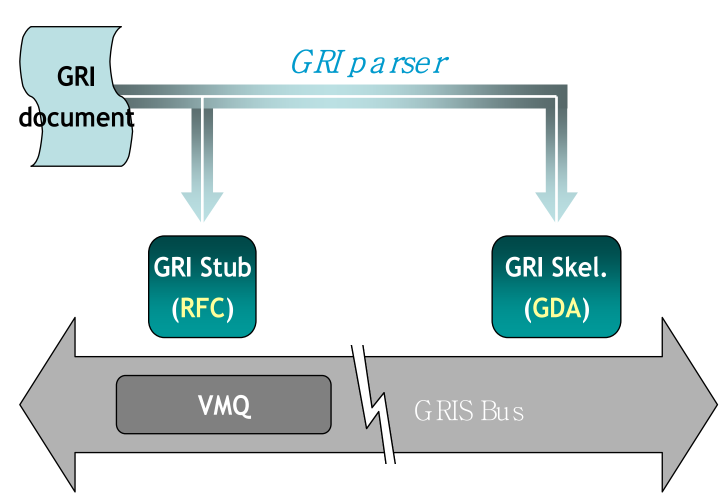

*RFC : Robot Factory Class, GDA : Gris Device Driver Adapter, GRI : Gris Remote Interface*

- **RFC Classes**
  - Abstract robot components (part/device) as generalized classes
  - Event Driven Architecture
  - Support easy way to build various robot system
  - Up/down interface
  - H/W independency
- **Part Proxy**
  - deliver service-invocations
  - marshalling and unmarshalling
  - demultiplex event-invocations
- **Object Adapter**
  - Manage object reference map
  - demultiplex service-requests
  - deliver event-invocations
- **GRI Stub**
  - RFC extension
- **GRI Skeleton**
  - GDA extension



# GRI (GRIS Remote Interface)

- **Automatic tool for generating new RFC**
  - **Adding some specific API functions to RFC for some specific robots**

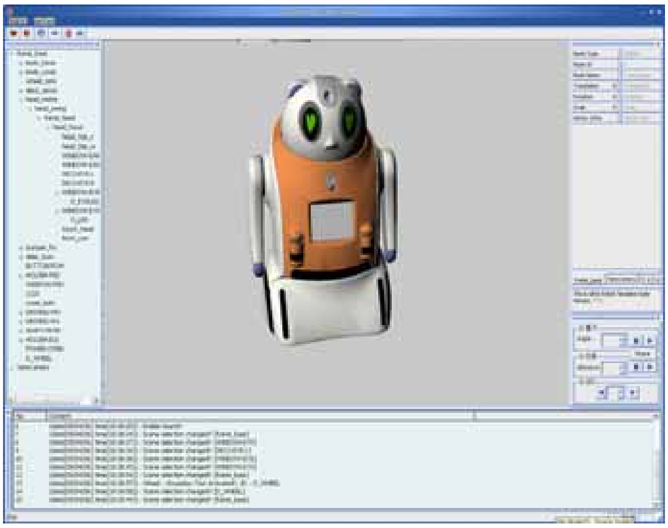



*RFC : Robot Factory Class, GDA : Gris Device driver Adapter*

- **GRI benefits**
  - **Define only interface for exchanged data and operation**
  - **GRI parser generates Up/down interfaces for newly added API**
  - **Developers can add their own APIs easily without knowing detailed RFC and VMQ operations**
  - **Reduce developer's mistake in their code**

# Robot Simulator

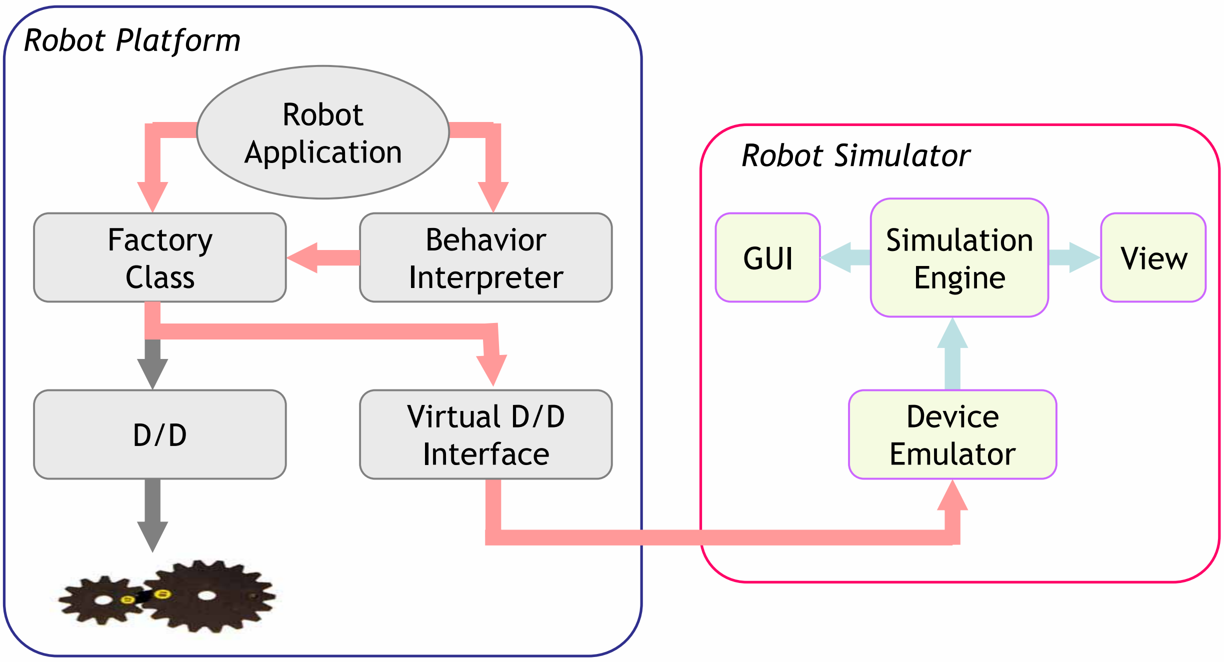
- Tools for testing real robot program without real robot platform
  - Simulates robot motion, sensor data
  - Supports Behavior scripts for testing simple motion, composite motion
  - Provides virtual device driver interface
    - Virtual camera, distance sensors, touch sensors, PIR sensor, sound, actuator etc.
  - Simulation can be executed by real robot program
    - Virtual Device Driver should be implemented in the way of real device driver interface of RFC



- Environment Configuration
  - Obstacle
  - Environment structures
- Components Modeling
  - Actuator
  - Camera
  - Sensors : PIR,PSD,Touch, …
  - Robot appearance
- Simulation
- Result

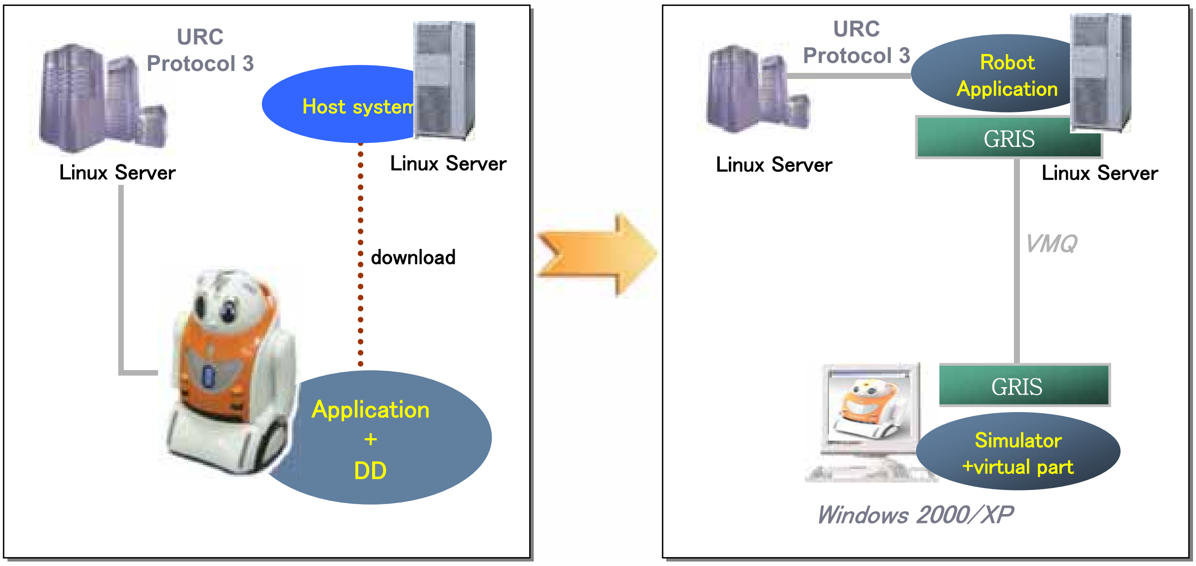


# Robot Simulator



# Robot Simulator

❑ Development Environment




**with real robot only**

**With GRIS platform & Simulator**



# Application – fairy tale

**A Robot Simulator with GRIS**

**A Real Robot with GRIS**

# Discussion

- Brief introduction of Anybot Studio
  - I t covers all the S/W suites of Network-based robot system, especially for URC system
- Here, our discussion is focused on Device Abstraction in robot S/W platform
- Our Implementations for this purpose are
  - GRIS (Robot S/W platform)
    - VMQ (Middleware)
    - RFC (Robot Factory Class)
    - GRI (GRIS Remote Interface)
- Our Implementations shows an example of applying device abstraction for developing robot S/W
- It provides an convenient way to construct an robot application for application/contents/service developers
  - Because it support only common robot interface



# Thank You!

## Washington DC 2006 Dec 4-6

# OMG Robotics DTF Robotics Devices and Data Profiles Working Group Progress Report

Seung-Ik Lee and Bruce Boyes, co-chairs

## 2006 Anaheim Summary

- The Working Group met 2006 Sep 25-27
  - Changed name of WG
  - Reviewed Boston meeting minutes
  - Presentation on Localization which might affect Profile WG
  - Semantics/ontology presentation – might also affect Profile WG
  - No work on Draft RFP
  - Presentation and demonstration of wireless sensors (SunSPOTs)
- Plans for next meeting Dec 2006

# 2006 Washington Summary

1. Presentation on IEEE-1588
2. Presentation on IEEE-1451
   1. Plan to support TEDs in our specification
      1. Store our tagging data in XML form
      2. Support 1451-compliant access methods to our tag data
3. Monday we worked on a draft RFP
   1. Draft of programmer API view is mostly complete
   2. We had some discussion of whether we need a separate RFP for the hardware-view, or if it can be included.
   3. We will present this RFP draft now...

# WG Actions prior to next meeting (1)

1. Do we need one RFP for hardware/bottom up and another for software API/top down, or just one?
2. Review OMG SmartSensor and SDO specs
   1. Volunteers needed...
3. Add use of existing specifications (1451, 1588, JAUS, OMG SmartSensor and SDO) to RFP
4. Invite Crossbow to our meeting
5. Create draft RFP for publication ~March 05.

# WG Actions prior to next meeting (2)

6. Work with Services WG, e.g., our transducers work with their localization abstractions
7. Conduct a lot of work on the mail list before next meeting. Yes! We can really do this!
8. Work with publicity committee and OMG publicity people to plan Robotics Information Day at Brussels meeting

# Robotics Devices and Data Profiles WG Road Map

| Item | Anaheim Sep-2006 | Wash DC Dec-2006 | San Diego Mar-2007 | Brussels Jun-2007 | Florida Sep-2007 |
|---|---|---|---|---|---|
| Programmers API: Typical device abstract interfaces and hierarchies | Discuss | Outline draft RFP | Review draft RFP | Review & Issue RFP | Response |
| Hardware-level Resources: define resource profiles | Discuss | Outline draft RFP | Review draft RFP | Review & Issue RFP | Response |

Note1: Florida meeting is the same time as RoboDev conference in California. OMG and some WG members will be at RoboDev instead of the OMG technical meeting

Note2: We can fulfill the OMG process by presenting the draft RFP at the start of the Brussels meeting and publishing it near the end of the meeting. However this may be optimistic due to the amount of work remaining to write the RFP

# Profile WG Mail 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/2006-12-15

# Robotic Device Interface Draft RFP Discussion

**Seung-Ik Lee and Bruce Boyes**

**OMG Washington Meeting, Dec. 4, 2006**

## Agenda

- Time tables of this meeting
- Roadmap presentation
- RFP discussion: Programmer’s view on robotic devices
- Review of Roadmap

# Time tables of this meeting

| Day | Where | Topic |
|---|---|---|
| **Monday (Dec. 4)**<br>**10:00~12:00** | **WG** | **- Welcome**<br>**- roadmap discussion**<br>**- RFP discussion on Programmer's view on devices (draft of draft RFP ?)** |
| **Monday (Dec. 4)**<br>**10:00~12:00**<br>**13:00~17:00** | **WG** | **-Anybot presentation**<br>**(Mr. Hong, Samsung Electronics)**<br>**- RFP discussion con't** |
| **Tuesday**<br>**13:00~14:00** | **Plenary** | **Special Talk: IEEE 1588** precision networked time reference<br>- John Eidson (Agilent) |
| **Tuesday**<br>**14:00~15:00** | **Plenary** | **Special Talk: IEEE 1451**<br>- Kang Lee (NIST) |
| **Tuesday**<br>**16:30~17:30** | **Plenary** | **-Anybot introduction**<br>**- (Mr. Hong, Samsung Electronics)** |



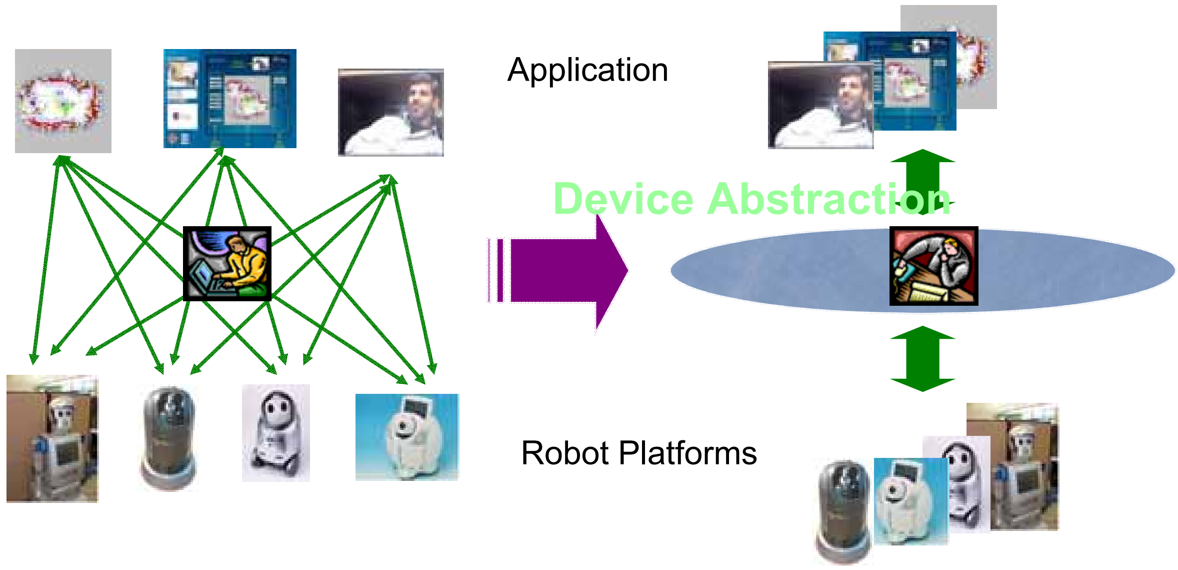
# Review of roadmap

| Item | Status | St. Louis Apr-2006 | Boston Jun-2006 | Anaheim Sep-2006 | DC Dec-2006 | San Diego Mar-2007 | Brussels, Belgium Jun-2007 | Jacksonville, FL USA Sep-2007 | Burlingame Dec-2007 | POC / Comment |
|---|---|---|---|---|---|---|---|---|---|---|
| Programmers API: Typical device abstract interfaces and hierachies RFP | Planned | Topic discussion | Topic discussion | draft RFP | Review RFP | RFP | | Initial Submission | | Proposed by Lee |
| Hardware-level Resources: define resource profiles RFP | Planned | Topic discussion | Topic discussion | draft RFP | Review RFP | RFP | | Initial Submission | | Proposed by Boyes |

- **Discussion**

# Why we need device abstraction and profiles?

From Programmers' view






# Items of generic RFPs

- **Objective**
- **Problem statement**
- **Scope of proposals sought**
- **Relationship to existing OMG specification**
- **Related activities, documents and standards**
- **Requirements**
  - **Mandatory requirements**
  - **Optional requirements**
- **Issues to be discussed**
- **Evaluation criteria**

# Objectives

- abstract interfaces for robots and their devices including remote transducers which interact with robots
- profiles which describe capabilities and properties of the devices
- a device classification method for abstracting the structure of robots and robotic devices
- Enumeration and management of robots and robotic devices

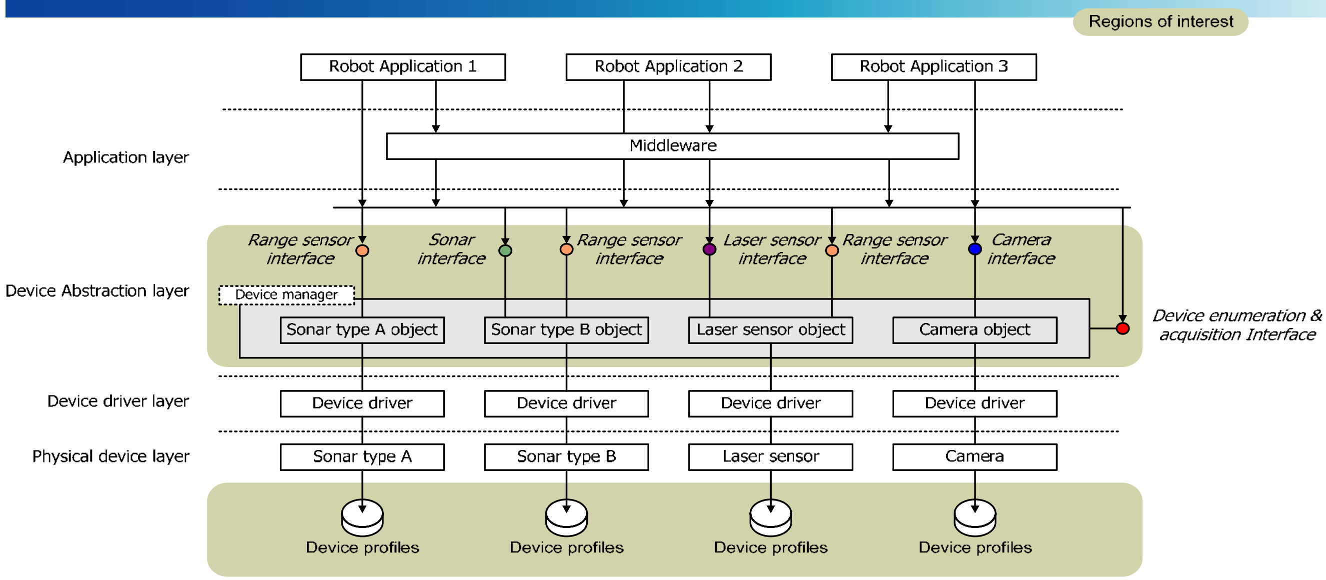
*Properties – fixed, intrinsic properties…*
*Capabilities –functionality… abilities, features, potential….*



# Problem statements (1)

- Current status
  - Developers use proprietary APIs to access devices and there maybe be many different interfaces to a single device
  - Because there is no standard interfaces to robotic devices, developers have a hard time in porting robotic applications to robot platforms other than the one for which the applications were originally developed
  - There is a great need for an open simulator which will execute robotic applications on simulations of heterogeneous hardware
  - Currently, it is impossible to run an application on several, heterogeneous robotic platforms
  - Somehow there should be clear benefits to vendors to support emerging standards
- Purpose
  - To improve robot application's portability
  - To provide consistent interfaces to robotic devices
  - To provide application developers with structured view on devices
  - We want to run an application on several, heterogeneous robotic platforms, and on simulators

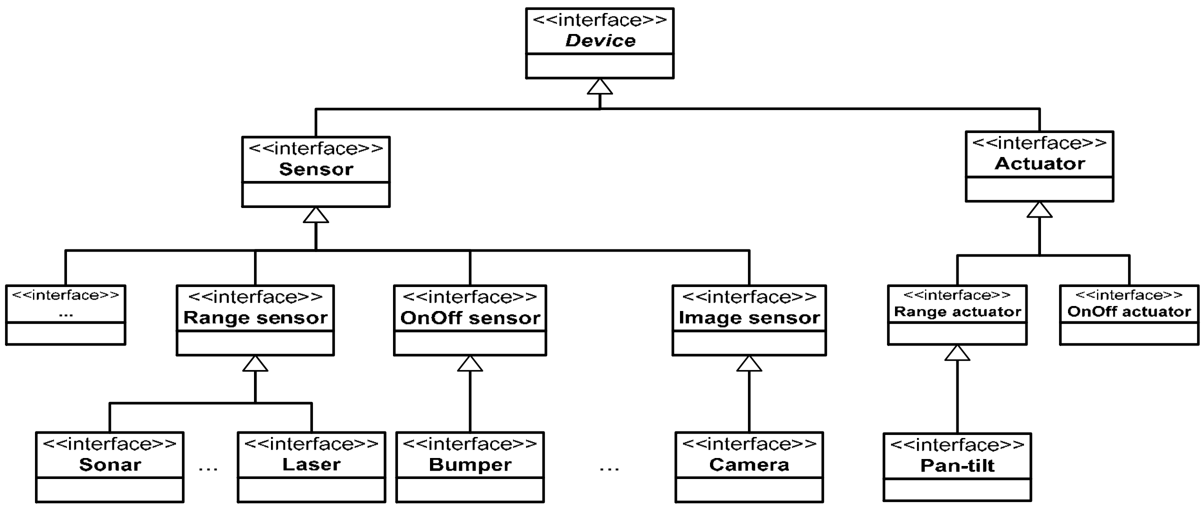
# Problem statements (2)




- **Where are the needs for standardization**
  - **Interfaces to robot and *robotic* devices**
  - **Interfaces to runtime device enumeration, and *device management***
  - **Interfaces should be domain-independent, application-independent, and *support arbitrary robotic devices***



# Problem statements (3)




- **A device classification or categorization hierarchy**
  - **An example is class hierarchy. But, it does not necessary to be class hierarchy**
- **The hierarchy should be closely related with device interfaces and profiles**
- **Interfaces should be applicable at least two or more specific devices of a same type or different types**

# Requirements

- **Proposals shall provide a platform independent model (PIM) and at least one platform specific model of abstract interfaces and profiles for devices of robots.**
- **Mandatory requirements: Proposals shall specify**
  - **abstract interfaces for robots and their devices including remote transducers which interact with robots**
  - **profiles which describe capabilities and properties of the devices**
  - **a device classification method for abstracting the structure of robots and robotic devices**
  - ***Enumeration and management of robots and robotic* devices**
- **Optional requirements**
  - **Synchronization, real-time capability**



# Relationship to existing OMG specifications

- **Platform Independent Model (PIM) and Platform Specific Model (PSM) for Super Distributed Objects (SDO) Specification version 1.0 [formal/2004-11-01]**
- **Robot Technology Component (RTCs) RFP [ptc/2005-09-01]**
- **Meta-Object Facility(MOF) 2.0 Core Available Specification [ptc/2004-10-15]**
- **Unified Modeling Language: Infrastructure version 2.0 [ptc/2004-10-14]**
- **Unified Modeling Language: Superstructure version 2.0 [formal/2005-07-04]**

## Relationship to existing OMG specifications (2)

- OCL 2.0 Specification version 2.0 [ptc/2005-06-06]
- Smart Transducers Interface Specification version 1.0 [formal/2003-01-01]
- Something related to synchronization and real-time capability
- DIAS…



## Related activities, documents and standards

- IEEE 1451
- IEEE 1588
- JAUS
- CANOpen
- OPC-UA (from Microsoft, ), OLE for process control

# Issues to be discussed

- None



# Evaluation criteria

- consistency in their specification
- feasibility and versatility across a wide range of different devices
- demonstration

robotics/2006-12-16

# - OMG Robotics DTF-

## - Robotic Functional Services Working Group -

# Meeting Report

## - Washington TC Meeting -

Washington (Virginia, USA) – December 06, 2006

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

# Schedule

- **Monday**
- AM 10:00 - 10:15 : Welcome – Roadmap presentation
- AM 10:15 - 10:45 : Presentation on HRI component Architecture
  (Soo-Young Chi – ETRI) + Discussion
- AM 10:45 – 12:00 : Localization RFP : Discuss the business case of the RFP.
  - AM 10:45 – 11:15 : Business Case of the Location Calculation Interface
    (Yeonho Kim – Samsung)
  - AM 11:15– 11:45 : Business Case of the Localization Service
    (Olivier Lemaire – AIST)
- PM 13:00 – 17:00 : Business case discussion (cont'd) + articulation of the RFP regarding to business case + address eventual technical issues
- **Tuesday**
- AM 10:00 – 11:30 : Review of the updated RFP
- AM 11:30 – 12:00 : Roadmap review, Homework, Next meeting WG Schedule
- PM 13:00 – 15:00 : Joint with Robotic Device and Data Profile WG : IEEE standard presentations
- **Wednesday**
- AM 9:00 – 12:00 : WG Reports and Roadmap Discussion

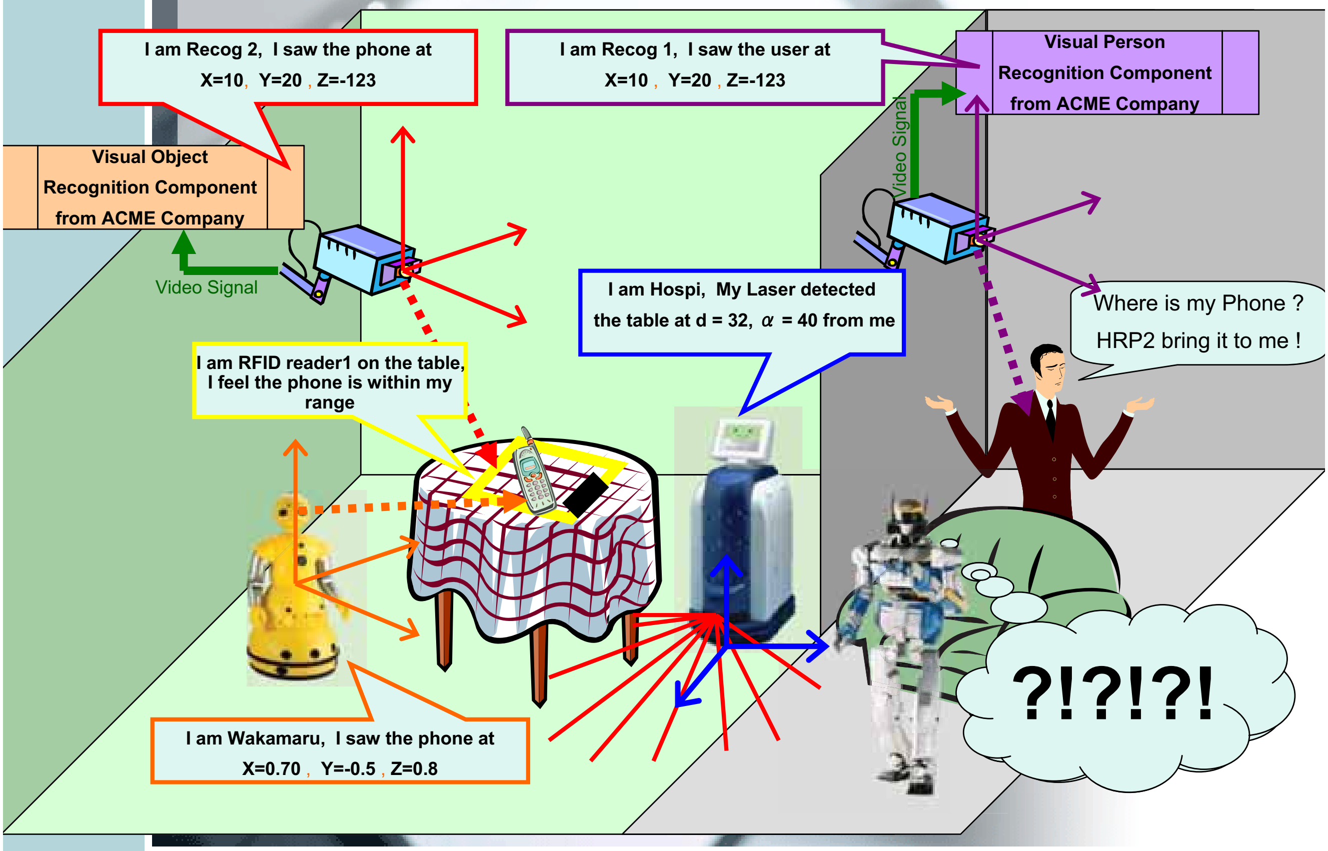
# Problem Statement

- One of the main difference between robotic systems and traditional information system is the fact of dealing with physical objects, real or virtual (in case of simulation) including :
  - The robot itself
  - The objects it manipulates
  - The environment in which it evolves
  - The people with who it interacts
- One of the most important information inherent to physical objects is their shape and location in space. This information is fundamental for :
  - The robot to be able to go from a point A to a point B (self-location, robot ⇔ obstacle relative location)
  - The robot to manipulate object (gripper ⇔ object relative location)
  - The robot to locate physical events and understand its environment (sensor location)
  - The robot to interact with the user (robot ⇔ user relative location)
- The location of an object is usually referred as the combination of it position and orientation both with regard to a coordinate system, a reference frame and a unit of measurement.

# Problem Statement

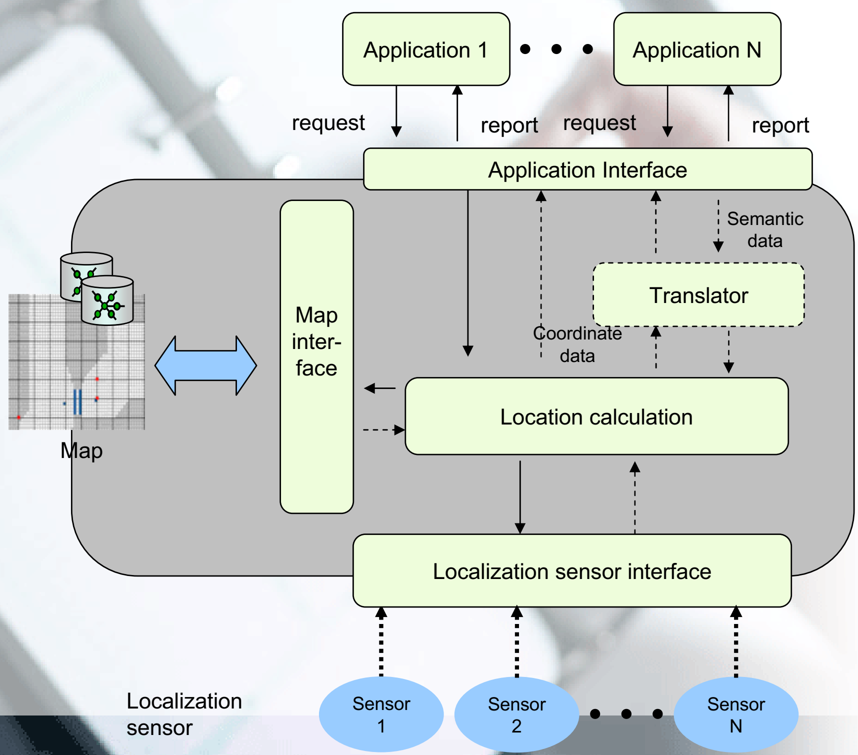
- In Mobile Robotic Systems, Localization refers to the technique through which a robot can determine or update its location through analysis of sensor data.
- In Ubiquitous Robotic Systems, Localization refers more generally to the technique through which the system can determine or update the location of an object through analysis of sensor data.
- The fact of using sensor data makes the localization process subject to noise and uncertainty of sensor
- A wide range of localization solutions are or will become available, making use of various types of sensors and algorithms.
  - Each will provide a different level of accuracy
  - Each will return a location depending on the sensor used
  - Each component is developed independently

# Typical Ubiquitous Robotic Scene




# Previous System Description

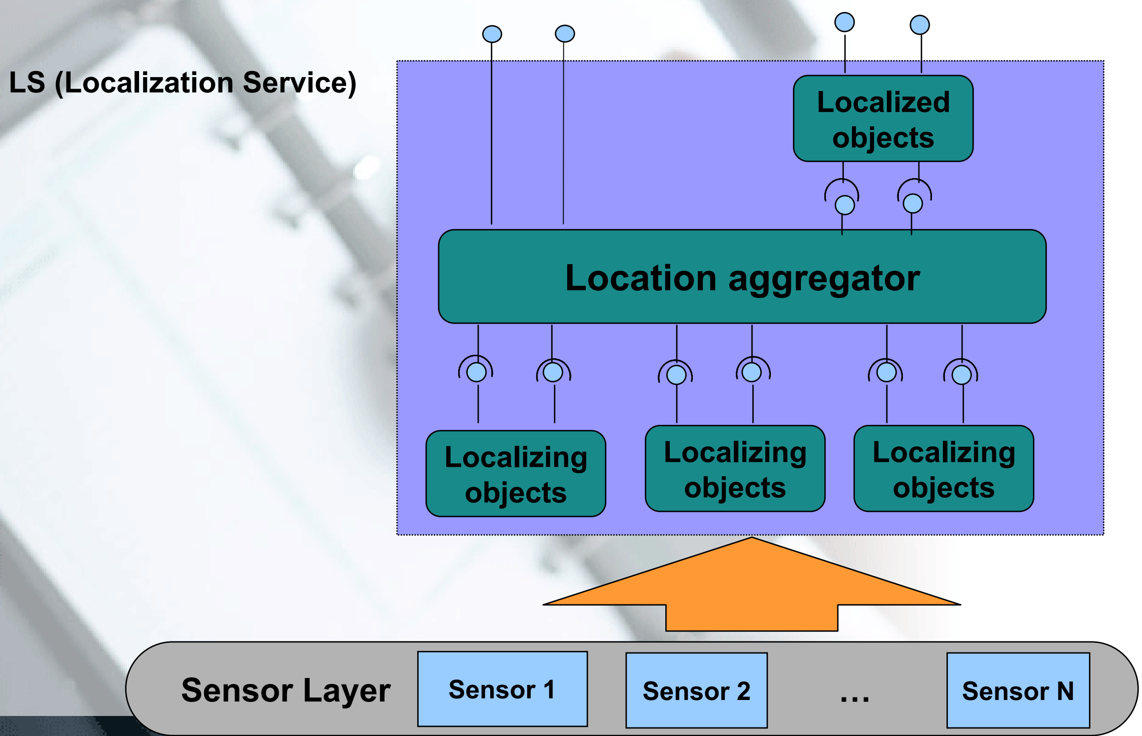
- Comments on RFP -



# Conclusion from Monday's Presentations

- Main focus should be set on the Application Interfaces as they can help both Robot Developer (to develop cheaper Hardware) and System Integrator (to ensure consistency in location related information)
- Location Calculation Interface should be merged with Application Interface (as they provide same output). However, some specialization (such as for dead-reckoning may be considered)
- Map Interfaces is technically necessary for multi robot system but a standard in this area will have to be too general and will certainly be never used.
- Sensor Interface represent an input to the process that, if standardized would hinder innovation, would be ignored or would add an unnecessary abstraction layer (redundant with the work of the Robotic Device and Data Profile WG)

# An Example of New LS Structure



## Mandatory requirements

- **Provide PIM and at least one RTC-PSM of LS**
- **Proposals shall specify general interfaces for accessing the location of objects.**
  - Proposals shall specify a minimum set of parameters to represent the location of objects including position and associated entries.
  - Proposals shall specify the format of the structures used to present:
    - Location data, Coordinate Systems, Reference Frame
- **Proposals shall specify generic interface for modules that perform location calculation.**
  - Each module shall provide interfaces to supply its generated location data to other modules
  - Advertising what type of object / what object can be localized
  - Registering new objects
  - Accepting localization request
  - Publishing the localization process result
  - Advertising what kind of sensor data can be used, what sensors are used
- **Proposals shall specify the interface of a facility that provides functionalities related to:**
  - Managing the different coordinate systems and frames defined in a robotic system, as well as their physical relationship
  - Managing the different localizing objects available in the robotic system
  - Managing the different localized objects present in the system
  - Providing a conversion of a location from one Coordinate System/ reference Frame/ Unit System tuple to another
  - Aggregate multiple location sources into one final position, using pluggable location fusion algorithms.

## Issues to be discussed

- A proposal shall
  - Demonstrate its feasibility by using a specific application based on the proposed model
  - Demonstrate its applicability to existing technology such as RTLS (Real-Time Location System)
  - Discuss simplicity of implementation
  - Discuss the possibility to apply the proposed model to other fields of interest such as Sensor Network
  - Proposals shall discuss the possibility of providing standard mechanism to access map data.

# Roadmap

| Item | Status | Anaheim Sep-2006 | Was. DC Dec-2006 | San Diego Mar-2007 | Brussels Jun-2007 | Jackson Ville Sept-2007 | Burlingame Jun-2007 |
|---|---|---|---|---|---|---|---|
| **Localization Service** | On-going | Topic Discussion | Draft RFP | Draft RFP 1st Review | 2nd Review RFP Issue | | RFP |
| **User Identification Service** | Stand-by | ? | ? | ? | ? | | ? |

robotics/2006-12-17

## - OMG Robotics DTF-
## - Robotic Functional Services Working Group -

# Meeting Schedule

## - Washington TC Meeting -

Arlington (Virginia, USA) – December 4, 2006

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

# Schedule

- **Monday**
- AM 10:00 - 10:15 : Welcome – Roadmap presentation
- AM 10:15 - 10:45 : Presentation on HRI component Architecture (Soo-Young Chi – ETRI) + Discussion
- AM 10:45 – 12:00 : Localization RFP : Discuss the business case of the RFP.
  - AM 10:45 – 11:15 : Business Case of the Location Calculation Interface (Yeonho Kim – Samsung)
  - AM 11:15– 11:45 : Business Case of the Localization Service (Olivier Lemaire – AIST)
- PM 13:00 – 17:00 : Business case discussion (cont'd) + articulation of the RFP regarding to business case + address eventual technical issues

- **Tuesday**
- AM 10:00 – 11:30 : Review of the updated RFP
- AM 11:30 – 12:00 : Roadmap review, Homework, Next meeting WG Schedule
- PM 13:00 – 15:00 : Joint with Robotic Device and Data Profile WG : IEEE standard presentations

- **Wednesday**
- AM 9:00 – 12:00 : WG Reports and Roadmap Discussion

# Roadmap

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

# 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"

# 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

AIST

robotics/2006-12-18

# Business case for the RFP on Robotic Localization Service

OMG Technical Committee Meeting
Arlington (Virginia, USA) – December 04th , 2006

Olivier LEMAIRE

AIST

National Institute of Advanced Industrial Science and Technology (AIST)

Ubiquitous Functions Research Group

AIST

# Concept of Robotic Localization

# Goal / Rational

- One of the main difference between robotic systems and traditional information system is the fact of dealing with physical objects, real or virtual (in case of simulation) including :
  – The robot itself
  – The objects it manipulates
  – The environment in which it evolves
  – The people with who it interacts
- One of the most important information inherent to physical objects is their shape and location in space. This information is fundamental for :
  – The robot to be able to go from a point A to a point B (self-location, robot ⇔ obstacle relative location)
  – The robot to manipulate object (gripper ⇔ object relative location)
  – The robot to locate physical events and understand its environment (sensor location)
  – The robot to interact with the user (robot ⇔ user relative location)



# 4 Physical Scales

| Location Scale | | Usual Range | Accuracy | Coordinate System | Sensors |
|---|---|---|---|---|---|
| Geo-centric Navigation | ability to determine one's position in absolute or map-referenced terms, and to move to a desired destination point, usually outdoor. | 100m ~ 10000km | Location :<br>>= 10 meters<br>Orientation : | Geodesic | GPS,<br>Camera,<br>RF,<br>Laser Range Finder |
| Global Navigation | ability to determine one's position in absolute or map-referenced terms, and to move to a desired destination point within a limited space (like within a building). | 50 cm ~ 200 m | Location :<br>From a few centimeters to a few meters (depends on sensor)<br>Orientation :<br><5 deg | Cartesian | Encoder,<br>Gyroscope,<br>Laser Range Finder,<br>Camera,<br>Indoor GPS,<br>RFID |
| Local Navigation | the ability to determine one's position relative to objects (stationary or moving) in the environment, and to interact with them correctly. | 1cm ~ 10 m | Location :<br>< 5 mm<br>Orientation :<br>< 1deg | Cartesian,<br>Polar,<br>(Spherical) | Encoder,<br>Gyroscope,<br>Laser Range Finder,<br>Camera,<br>Indoor GPS,<br>RFID |
| 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 | 1mm ~ 1m | Location :<br>< 1 mm<br>Orientation :<br>< 0.1deg | Cartesian,<br>Polar,<br>Spherical | Encoder,<br>Gyroscope,<br>Camera,<br>Contact switch |

# Some Related Definitions (from Wikipedia)

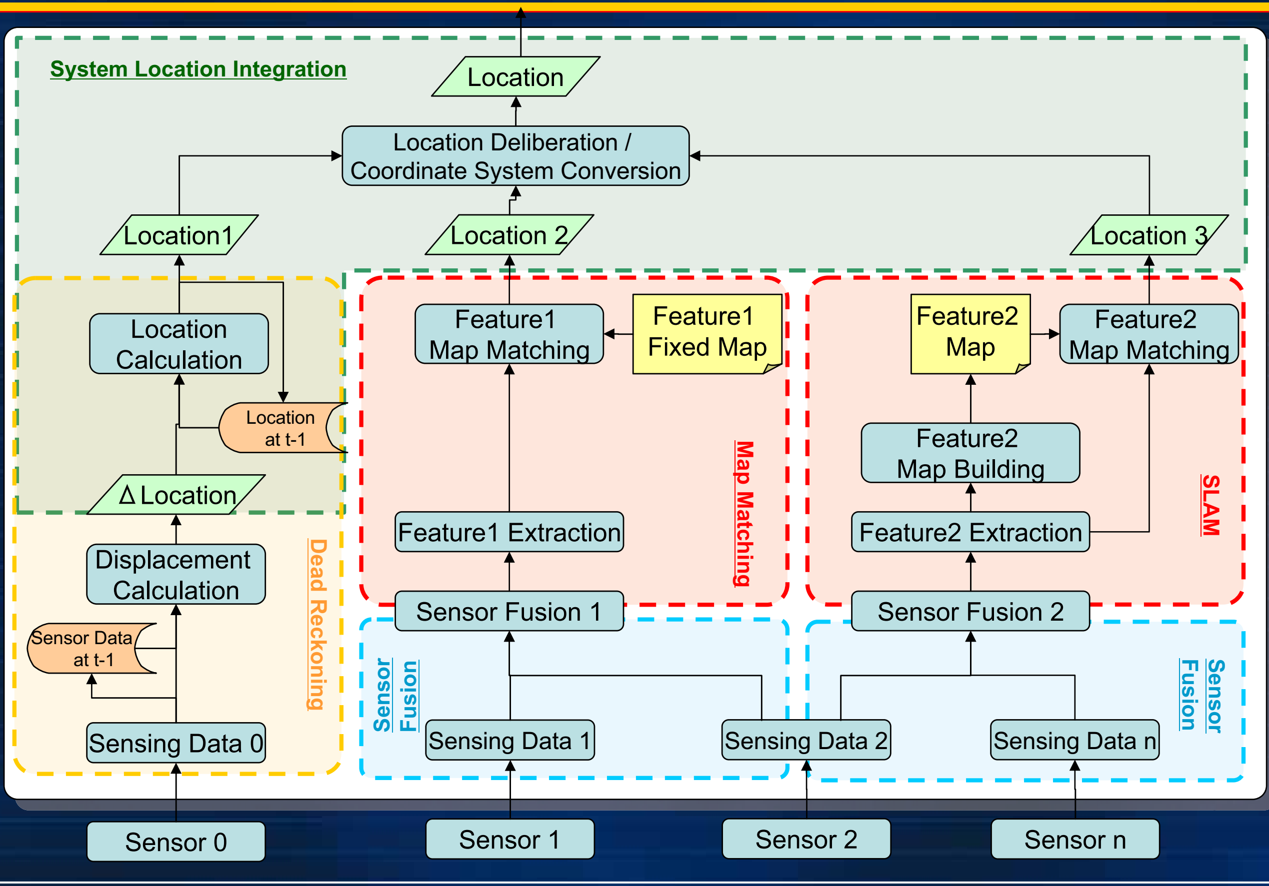
- **Sensor Fusion**
  - **Sensor fusion** is the combining of sensory data or data derived from sensory data from disparate sources such that the resulting information is in some sense *better* than would be possible when these sources were used individually. The term *better* in that case can mean more accurate, more complete, or more dependable, or refer to the result of an emerging view, such as stereoscopic vision (calculation of depth information by combining two-dimensional images from two cameras at slightly different viewpoints).
  - The data sources for a fusion process are not specified to originate from identical sensors. One can distinguish *direct fusion*, *indirect fusion* and fusion of the outputs of the former two. Direct fusion is the fusion of sensor data from a set of heterogeneous or homogeneous sensors, soft sensors, and history values of sensor data, while indirect fusion uses information sources like a priori knowledge about the environment and human input.
  - Sensor fusion is also known as *(multi-sensor) data fusion* and is a subset of *information fusion*.
  - Transducer Markup Language (TML) is a XML based markup language which enables sensor fusion.
  - Sensor fusion is an overarching term for a number of methods and algorithms. Just to mention a few:
    - Kalman filter
    - Bayesian networks
    - Dempster-Shafer

- **Dead Reckoning (ie Odometry)**
  - **Dead reckoning** (**DR**) is the process of estimating a global position of a navigating agent by advancing a known position using direction, speed, time and distance of travel. This method of navigation is used in ships, aircraft, automobiles, rail engines, construction site engines (e.g., in tunnels) and, more recently, mobile robots.
  - **Odometry** is the study of position estimation during wheeled vehicle navigation. The term is also sometimes used to describe the distance traveled by a wheeled vehicle. Odometry is used by some track or wheeled robots to estimate (not determine) their position relative to a starting location. Odometry is the use of data from the rotation of wheels or tracks to estimate change in position over time. This method is often very sensitive to error. Rapid and accurate data collection, equipment calibration, and processing are required in most cases for odometry to be used effectively.

- **SLAM / Map Matching**
  - **Simultaneous localization and mapping** (**SLAM**) is a technique used by robots and autonomous vehicles to build up a map within an unknown environment while at the same time keeping track of their current position. This is not as straightforward as it might sound due to inherent uncertainties in discerning the robot's relative movement from its various sensors.
  - If at the next iteration of map building the measured distance and direction travelled has a slight inaccuracy, then any features being added to the map will contain corresponding errors. If unchecked, these positional errors build cumulatively, grossly distorting the map and therefore the robot's ability to know its precise location. There are various techniques to compensate for this such as recognising features that it has come across previously and re-skewing recent parts of the map to make sure the two instances of that feature become one. Some of the statistical techniques used in SLAM include Kalman filters, particle filters (aka. Monte Carlo methods) and scan matching of range data.
  - A seminal work in SLAM is the research of R.C. Smith and P. Cheesman on the representation and estimation of spatial uncertainty in the mid 1980s. Other pioneering work in this field was conducted by the research group of Hugh F. Durrant-Whyte in the early 1990s.
  - SLAM in the mobile robotics community generally refers to the process of creating geometrically accurate maps of the environment. Topological maps is another method of environment representation which capture the connectivity (i.e., topology) of the environment rather than creating a geometrically accurate map. As a result, algorithms that create topological maps are not referred to as SLAM.
  - SLAM has not yet been fully perfected, but it is starting to be employed in unmanned aerial vehicles, autonomous underwater vehicles, planetary rovers and newly emerging domestic robots.
  - SLAM can use many different types of sensor to acquire data used in building the map such as laser range finders, sonar sensors and cameras.



# Localization Process Typical Data Flow



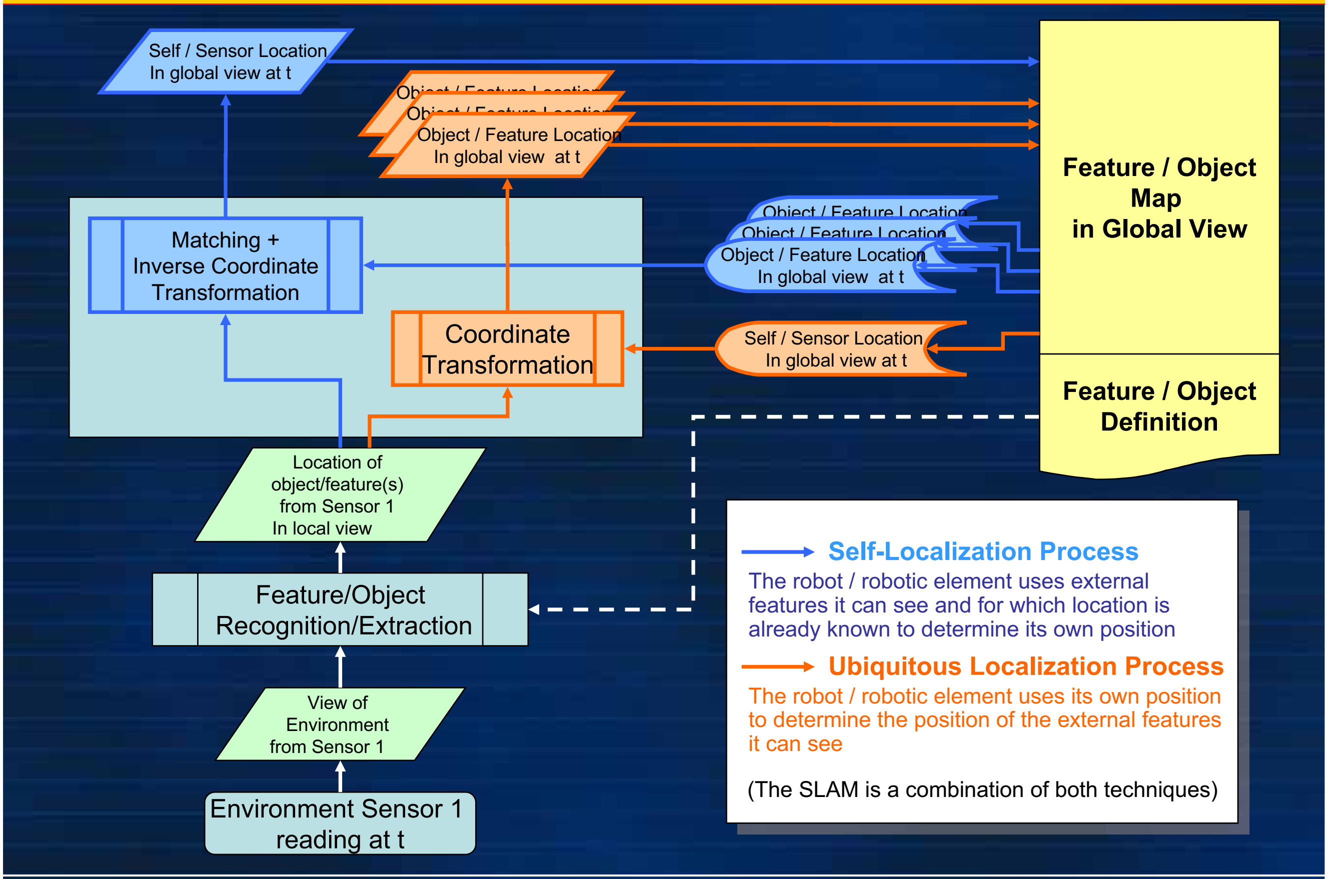
# Many Features…

- **Common factor for matching techniques**
  - **Notion of feature associated to location**

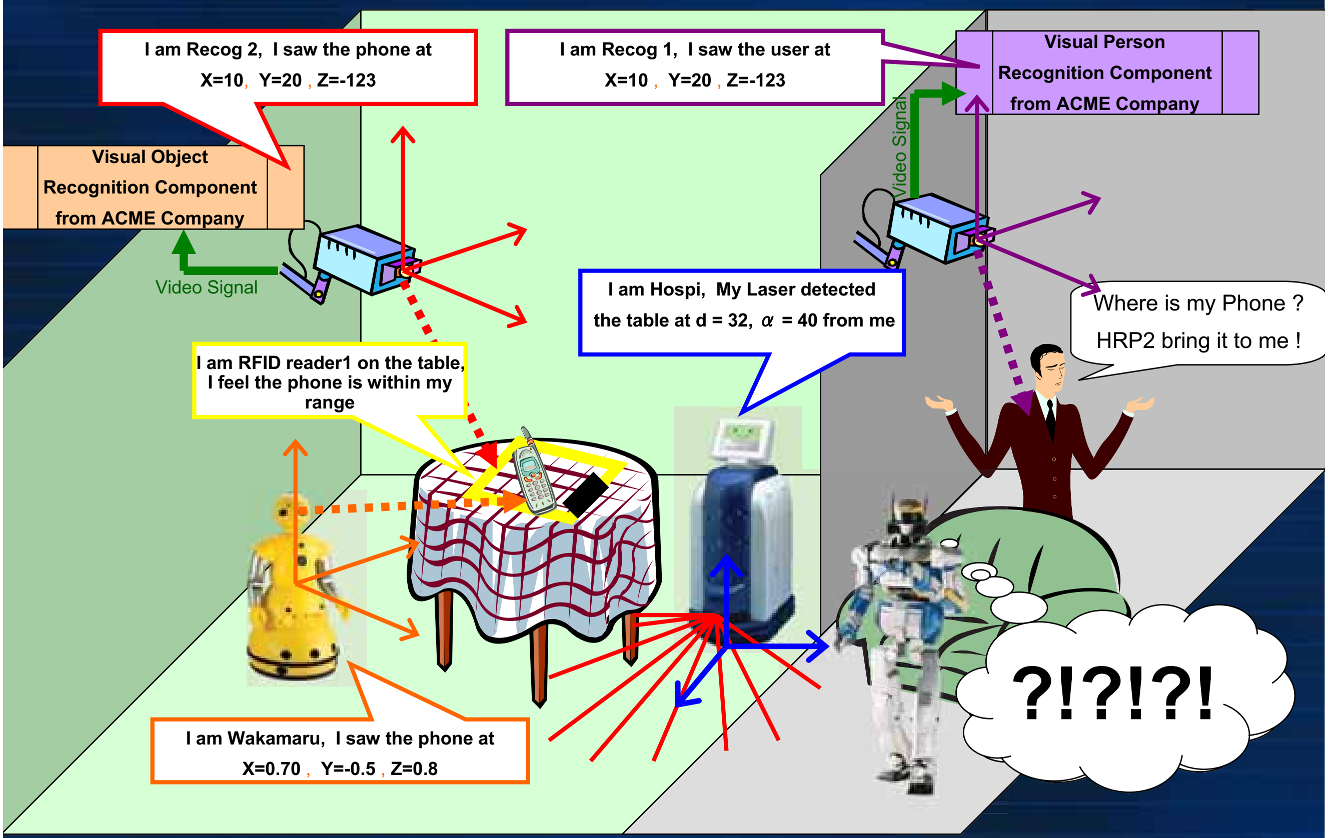
- **Many possible kinds of features !**
  - Geometrical features (straight lines, corners, polygons…)
  - Topological features (intersection, road, slopes…)
  - Cell of grid map
  - Symbolic features
  - Visual features (snapshot) / Visually derived (SIF)
  - Sound features (audio print)
  - … And more to come !!



# Self Localization vs. Ubiquitous Localization



# Typical Ubiquitous Robotic Scene






# Issues and Requirements

AIST

# Facts from Previous Scene

- What information HRP2 request and how did?
  - Position Relative to itself ? Its gripper ? Relative to the house ?
  - How can it find components that will reply to its request ?
  - How can it decide what information to use ?
- According to Recog2 and Wakamaru, the phone is a 2 totally different positions
  - What is the base/frame used to express the position information from Recog1 / Hospi ?
  - What is the unit used to express the position information from Recog1 ?
- According to Recog1 and Recog2, the phone and the user are exactly at the same position
  - Images were 2 different cameras placed at different position in the space
  - What is the relative position of each camera within the space ?
- The RFID Reader detects the proximity of the phone but cannot tell exactly where it is, nor where it itself is (just knows it is attached to the table). Hospi cannot see the phone but can locate the table.
  - Hospi knows only 2D coordinates (like most wheeled robots)
  - Hospi does not only provide a position relative to itself in polar coordinates
  - Could combining these information allow for precisely locating the phone ?



AIST

- **Object/Robot location information provided by most localization solution are only partially explicit , while the rest of the information remains implicit**
  - **The system integrator must know these implications to build a system and can support only specific configuration : the system cannot adapt dynamically**

## Scope of Robotic Systems (for this RFP)

- Mobile Indoor Service Robots
  - Including Communication Robots, Delivery Robots, Surgery Robots…
  - Home space, Office space, Factory space
- Intelligent Space
- Global Navigation, Local Navigation and Personal Navigation
- Robot-self Localization and Ubiquitous Object Localization



# Interfaces Expected in the RFP and their Business Application

## Actors to Consider in Localization Use-Case

- As of today, 2 main actors and their point of view should be considered .
- In the (hopefully near) future, 2 main intermediate actors should be considered.

(Mobile) Robot Manufacturer / Solution Provider

(Mobile) Robot Component Developer

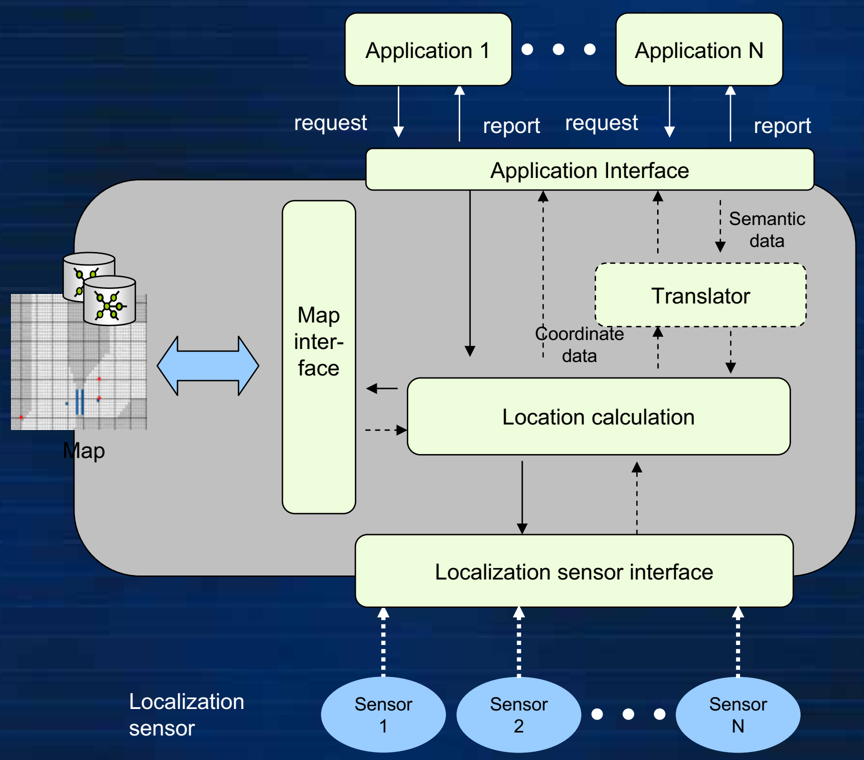
(Mobile) Robot Integrator

Robotic System Integrator

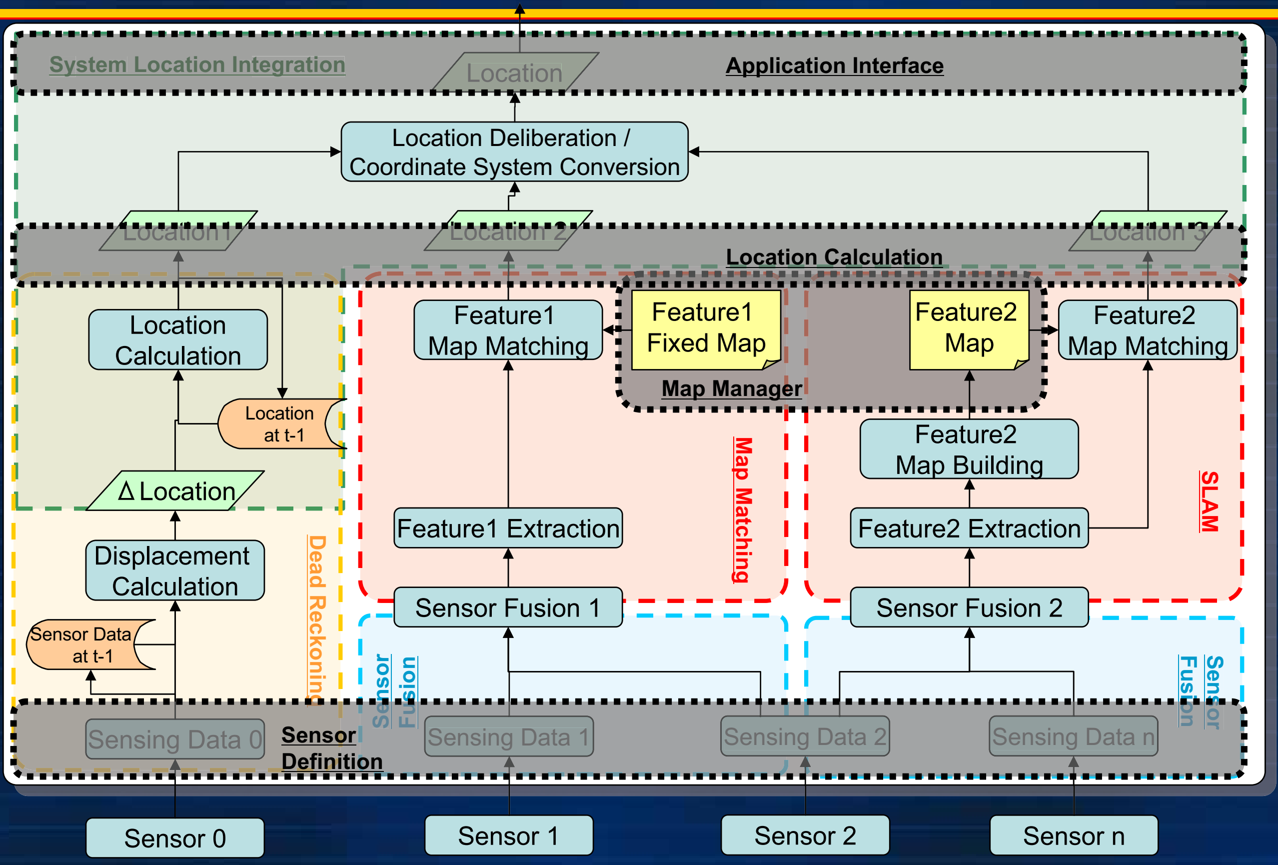
When deciding what interface should be proposed in the standard, we must ask our selves **how and why each actor will use the interface** and **how his business will benefit** for the standard interface.



## Interfaces proposed at the last Anaheim meeting



## Interfaces within the Localization Data Flow






## Comments

- In Map Matching and SLAM, all elements are tightly coupled by the **nature of the feature**.
  - Difficult to split them in separate functional blocks
  - Multi vendor system is very improbable at this level.
- The Application Interface and Location Calculation basically have the same output
  - Couldn't we merge them into a single interface or package
- The Sensor Data Interface is an Input Interface
  - Should define a contract on what is provided (output) rather than what will be used to provide it (input)
  - Adds one non functional layer (performance issues)
  - Work will be redundant with Profile WG

# Application Interface

# Application Interfaces

- The Robot Application Localization Interfaces (RALI) intend to expose location information of the physical entities present in the system in a consistent and explicit way, mostly to the system integrator

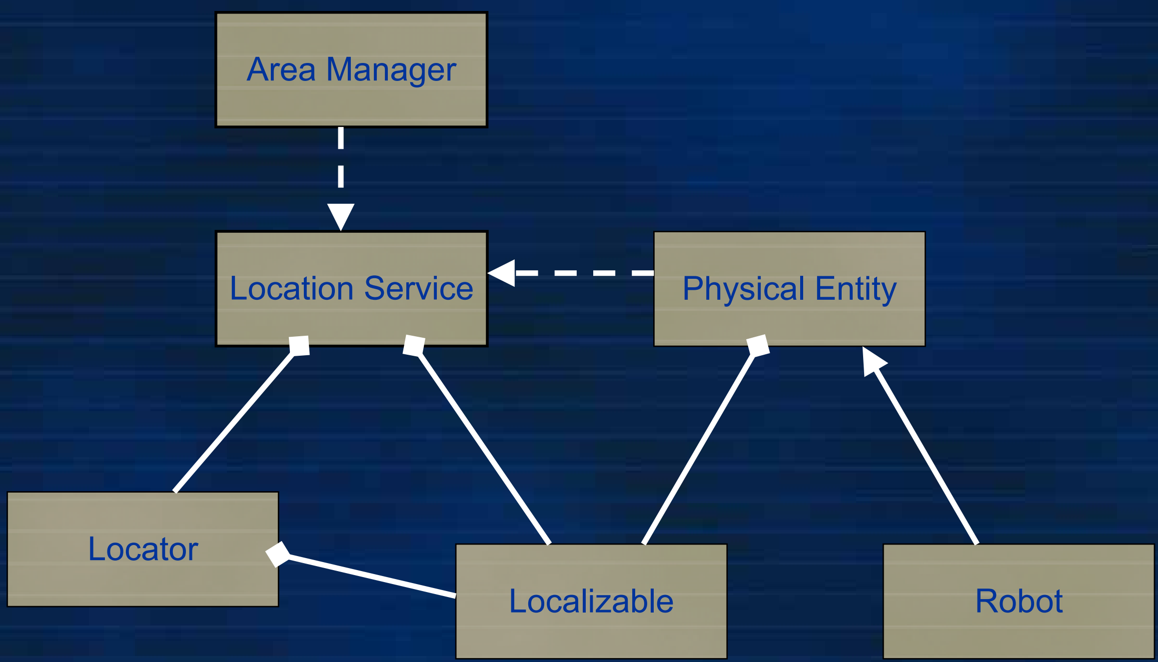
- 4 Types of interfaces / profiles can be proposed :
  - Localizable
    - Interface supported by all components representing an active physical entity and which ensure the accessibility of information related to the self-location into any arbitrary Coordinate System (usually chosen by the component developer).
    - Facet of a component, in charge of
      - Publishing the component's physical entity / parts self location information
  - Locator
    - Interface supported by all components capable of determining the location of a physical entity or not into any arbitrary Coordinate System (usually chosen by the component developer).
    - Individual component, in charge of
      - Advertising what type of object / what object can be localized
      - Registering new objects
      - Accepting localization request
      - Publishing the localization process result
      - (Optionally) Advertising what kind of sensor data can be used, what sensors are used
      - (Optionally) Dynamically its sensor connection configuration

# Application Interfaces

- 4 Types of interfaces / profiles can be proposed (cont'd):
  - RoS Location Service
    - Central service, in charge of
      - Managing the different coordinate systems and frames defined in a robotic system, as well as their physical relationship.
      - Managing the different locators available in the robotic system.
      - Managing the different localizable present in the system.
      - Acting as locator proxy when  the expected CS is different from the provided one.
      - Aggregate multiple location sources into one final position, using pluggable location fusion algorithms
  - Area Manager (optional)
    - Central Service in charge of
      - Managing the definition of physical areas and their coordinates
      - Managing relation between physical entities and the areas
      - Managing predefined actions to execute when an entity enters into a certain area

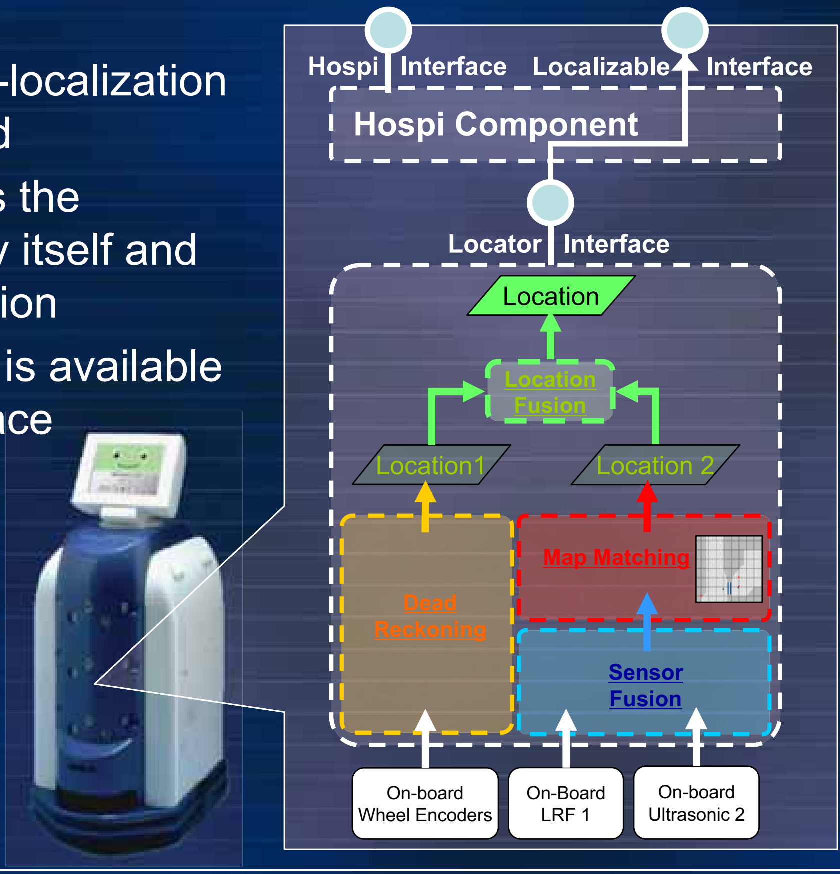


# Relation between the 4 Interfaces



# Typical Robot Maker Use of Application Interfaces

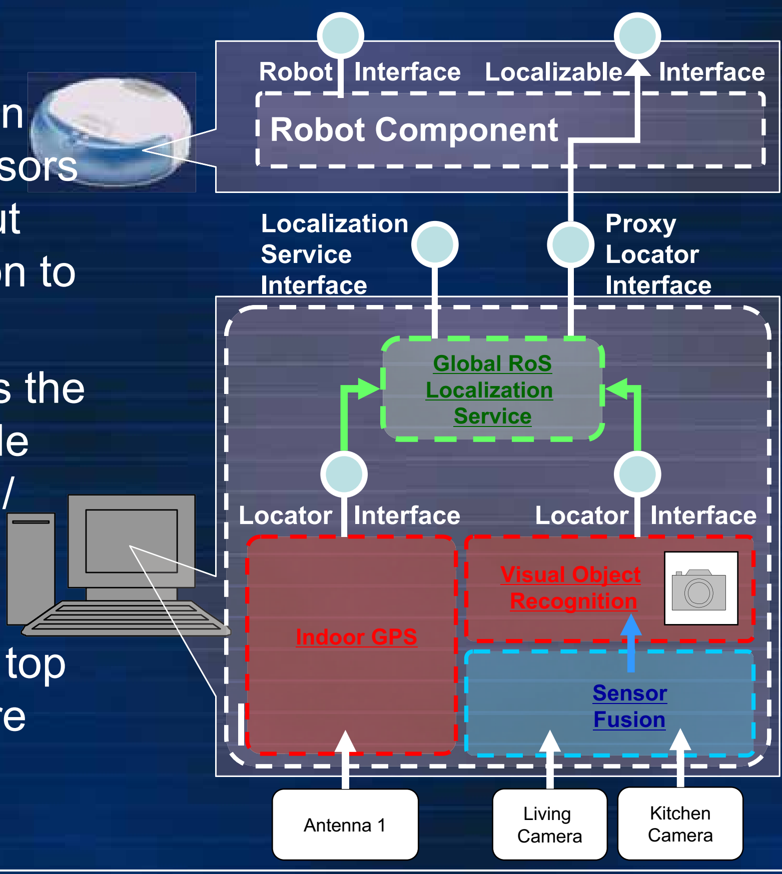
- The robot has it self-localization capabilities on board
- The maker develops the complete solution by itself and can protect his solution
- The Robot Location is available from top level interface






# Low Cost Robot Maker Use of Application Interfaces

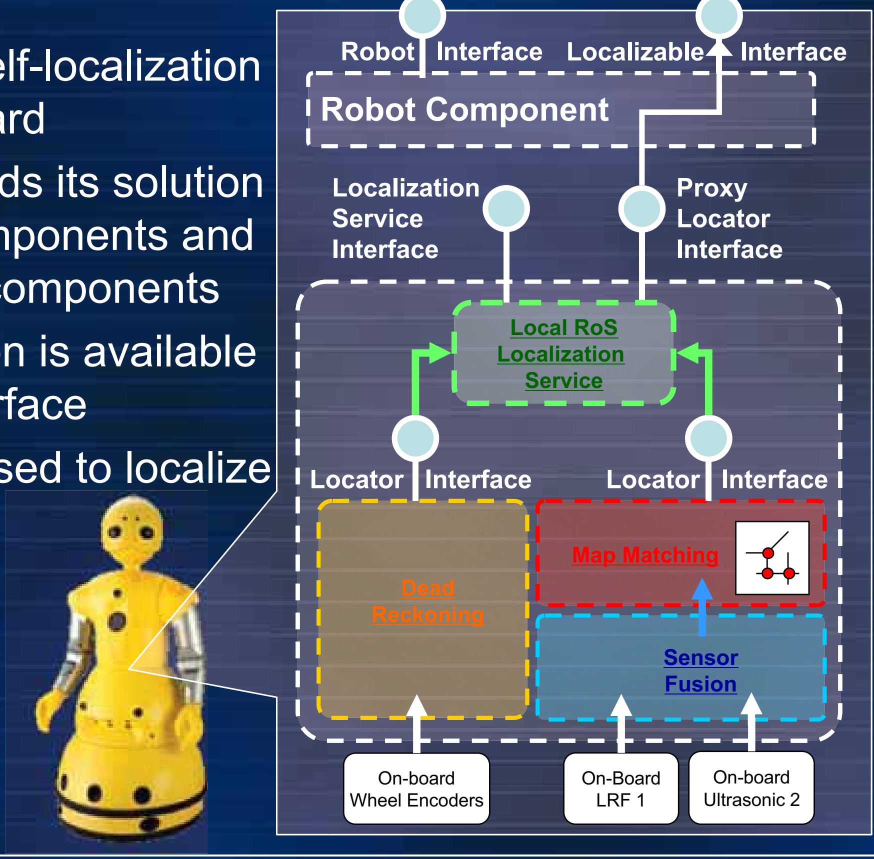
- The robot has no localization capability on board (no need for sensors or CPU -> Low cost) but delegates its localization to an external entity
- The robot maker leaves the system integrator decide which locator(s) to use / environment
- The Robot Location remains available from top level interface as if there was self-localization




Intelligent Systems Research Institute – Ubiquitous Functions Research Group

# Mobile Robot Integrator Use of Application Interfaces

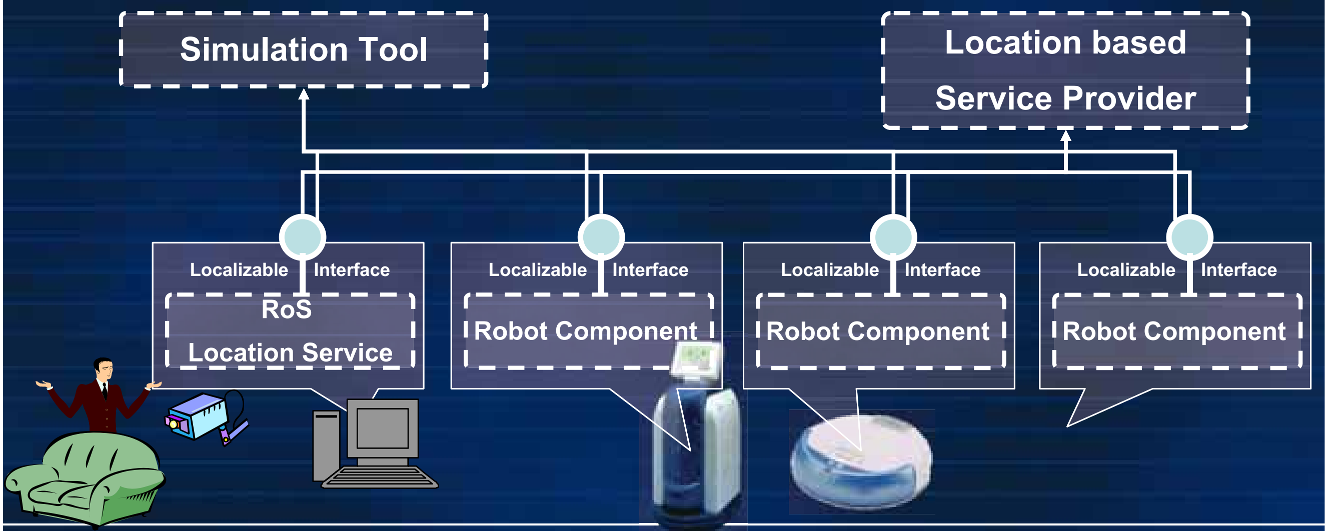
- The robot has it self-localization capabilities on board
- The integrator builds its solution from available components and can freely switch components
- The Robot Location is available from top level interface
- Locators can be used to localize different features






# Robot System Integrator Use of Application Interfaces

- The Integrator can use consistent tools to track robot position and send position related/dependent requests
- The Integrator can see all objects in the same way (active with self-loc AND passive)
- Integrator can chose the locators according to the environment, locator are dynamically pluggable
- Locators can be used to localize different features



# Application Interface Standard Benefit Grid

| Interface (Overall benefit of a Standard) | Business Actor (Overall Satisfaction if Standard) | Loc. Solution Provider / Robot Manufacturer P.o.V. | Robot Integrator / System Integrator P.o.V. |
|---|---|---|---|
| | Expectation, Acceptance / Specialization, Difficulty | Want to reduce cost of robot (HW & SW)<br>What to sell same robot for all environment<br>Want to develop innovative localization solution without being locked<br>Want to sell a suite rather than simple components/tools | Have access to location information of all physical entities in a consistent way<br>Be able to seamlessly combine location related component developed separately<br>Have consistent tools (visualization, simulation) |
| **Localizable Interface** | Provides consistent access to self-location related information independently of the calculation algorithm. Easy to define, Light | A robot maker does not need this interface but may find it convenient when dealing with location of objects other than robot<br>Will help some developers to shape component | Have access to location information of all physical entities in a consistent way |
| **General Locator Interface** | Provides consistent access to component able to locate physical entities independently of the calculation algorithm. Easy to define, Light | Allow Robot Integrator to use self-localization component according to the sensors available in the Robot – or the location service in none available | Allow for reusing localization / detection component to localize several different entities<br>Allow to seamlessly add extra localization capabilities to suit environment |
| **Location Service Interface** | Provide standard way to access location related information of all entities in space and manage locators<br>Rather easy to define | Low cost robot can rely on Service to provide location based service without having self localization<br>Let maker use its own coordinate system internally for performance<br>No obligation to use or support | Allow to have access to location information of all physical entities in a consistent way<br>Be able to seamlessly combine location related component developed separately<br>Allow to have consistent tools (visualization, simulation) |
| **Area Manager Interface** | Provide a standard way to define areas in space and register action to perform when a predefine entity-area relation become true or evaluate is a service is available in given area.<br>Requires a deep study De Facto standard is more likely | | |



# Location Calculation Interface

AIST

# Location Calculation Interface Standard Benefit Grid

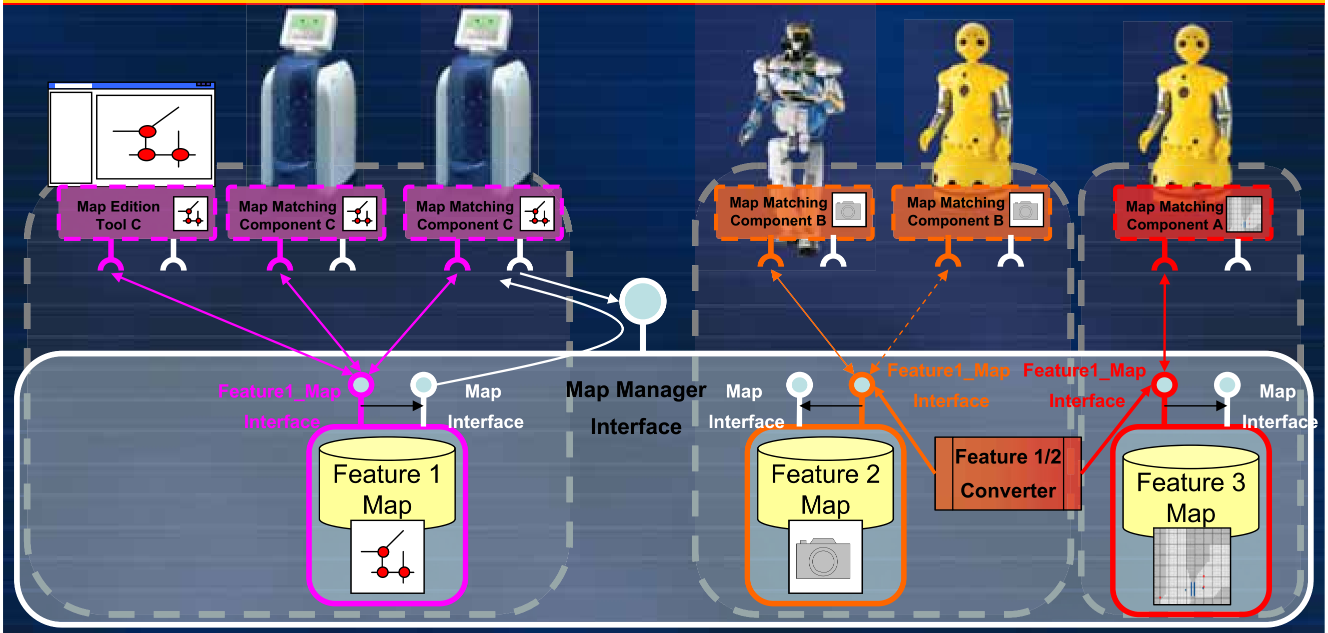
| | Business Actor (Overall Satisfaction if Standard) | Loc. Solution Provider / Robot Manufacturer P.o.V. | Robot Integrator / System Integrator P.o.V. |
|---|---|---|---|
| **Interface** (Overall benefit of a Standard) | Expectation, Acceptance / Specialization, Difficulty | Want easy development process<br>Want to develop innovative localization solution without being locked<br>Want to sell a suite rather than simple components/tools | Be able to select the localization components that best suit his environment<br>Build systems as cheap as possible by sharing / reusing functionalities and resources |
| **Locator Specialized Interface** | Provide a standard specialized view of the most commonly used localization algorithms.<br>Hard to define, Requires a deep study of all existing system to find commonalities, Cannot be exhaustive, Can become outdated.<br>Breaks the process/outcome decoupling goal of component approach | Would help some developers to implement their solution faster.<br>Most developer will need to extend/modify the interface to fit the need of their algorithm.<br>Most developer will want to extend/modify the interface to ensure a customer will buy the complete suite rather than a simple component. | Would provide some more tuning capabilities at run time (fine tuning of algorithms) but this is of limited application and makes the system more complex to manage |
| **Dead-Reckoning Interface** | Provides consistent access to dead reckoning related information independently of the calculation algorithm. Easy to define, Light<br>Limited use | Would help some developers to implement their solution faster.<br>Most developer will need to extend/modify the interface to fit the need of their algorithm.<br>Most developer will want to extend/modify the interface to ensure a customer will buy the complete suite rather than a simple component. | Dead Reckoning is not usable as is and is of limited use |



AIST

# Map Manager Interface

# Use-Case of the Map Interface




- The Map Manager / Map Interfaces acts in the same way as a general component discovery service -> No specialization
- The map matching component can only deal with the feature map for which it has been developed so the Component Developer will also be responsible for implementing the storage solution, as well as the map visualization and editing tool -> The developer is in control of all parts of the solution, no standard is necessary
- Even different robots using the same type of localization solution may not be able to share their map as the hardware configuration will be different and so will be their internal map (even for the same type of robot, the calibration may be different)
- A map converter can be provided but it needs only the specialized interface.



# Map Interface Standard Benefit Grid

| | **Business Actor** (Overall Satisfaction if Standard) | Loc. Solution Provider / Robot Manufacturer P.o.V. | Robot Integrator / System Integrator P.o.V. |
|---|---|---|---|
| **Interface** (Overall benefit of a Standard) | Expectation, Acceptance<br>Specialization, Difficulty | Want easy development process<br>Want to develop innovative localization solution without being locked<br>Want to sell a suite rather than simple components/tools | Want to be able to select Map Storage, Map Matching components and Map editing Tools independently and easily interchange them.<br>Want all robots to take advantage of other robots maps |
| **Map Manager Interface** | Centrally manage the different maps available in the system.<br>Easy to define, Simple collection management, nothing really new | Standardized map manager is not the main concern of the solution provider.<br>More interested in the maps themselves as way differentiation. | A map manager would allow for central management and easy retrieval of maps.<br>However, the functionality provided is not different from a general component discovery service so it would only make the system more complex for nothing |
| **Map General Interface** | Provide a unified view of a map as a "collection of features" and allow for the management of these features.<br>Easy to define, Collection management with some very general attributes related to map feature definition | Would help some developers to implement their solution faster in a limited way.<br>The general concept of feature may not suite all needs (harder to adapt to grid maps) | Only a facility for the map manager.<br>A general map interface alone will not enable efficient map editing (only text based)<br>A general map interface will not enable map sharing between robots |
| **Map Specialization Interfaces** | Provide a standard specialized view of the most commonly used maps.<br>Hard to define, Requires a deep study of all existing system to find commonalities, Cannot be exhaustive, Can become outdated.<br>De Facto standard is more likely | Would help some developers to implement their solution faster.<br>Most developer will need to extend/modify the interface to fit the need of their algorithm.<br>Most developer will want to extend/modify the interface to ensure a customer will buy the complete suite rather than a simple component. | Would allow the integrator to choose the map storage solution, matching component and map edition tool independently.<br>However, as most developers will extend the interface, the benefit is limited.<br>Even this interface will not ensure that different robots can share the same map<br>Will still be in charge to develop custom map converters |

# Conclusion

## Temporary Conclusion

- Main focus should be set on the Application Interfaces as they can help both Robot Developer (to develop cheaper Hardware) and System Integrator (to ensure consistency in location related information)
- Location Calculation Interface should be merged with Application Interface (as they provide same output). However, some specialization (such as for dead-reckoning may be considered)
- Map Interfaces is technically necessary for multi robot system but a standard in this area will have to be too general and will certainly be never used.
- Sensor Interface represent an input to the process that, if standardized would hinder innovation, would be ignored or would add an unnecessary abstraction layer (redundant with the work of the Robotic Device and Data Profile WG)

# That’s it !

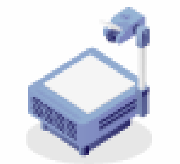
# Propose the Interaction Model Architecture for HRI Components Standardization

2006. 10. 10


---

IT R&D Global Leader ETRI

## Contents



# STATE-OF-THE-ART

✓In the last years, robotics research has focused more and more intensively on applications oriented towards service to humans, medical assistance, and human-friendliness in general.

✓This led to a number of impressive results in academic and industrial contexts, in the development of robots in diverse shapes, as humanoids, pets, medical tools, or appliances.

✓While the application range is wide, and so is the possible shaping of this kind of robots, a common element is that they are increasingly introduced into the Society of Humans.

✓Crucial aspects of robot design are therefore the modalities, mechanisms and tools of human-robot interaction, of communication with human beings, with the environment and possibly with other robots, and of interaction of the robotic systems with human environments, modelled on persons' needs and habits, and no more arranged according to the robot functions.





# Robot as Team Member

✓Toward Human-Robot Collaboration, highlights the importance of creating robot capabilities and interfaces that address human concerns such as social appropriateness, safety, and quality of service.

✓Robots are, or soon will be, used in such critical domains as search and rescue, military battle, mine and bomb detection, scientific exploration, law enforcement, and hospital care.

✓Such robots must coordinate their behaviors with the requirements and expectations of human team members; they are more than mere tools but rather quasi-team members whose tasks have to be integrated with those of humans.

✓Robot as Team Member, highlights the importance of building core science and understanding the social and technical issues in human-robot interaction in the context of teams and groups

# Special Issue on Human-Robot Interaction

- ✓ The recent trend toward developing a new generation of robots that can participate in our lives and exist in human environments has introduced the need for investigating the paradigms, techniques, and technologies for the interaction between people and robots.
- ✓ An important goal for the field of *Human-Robot Interaction* is to develop autonomous and semi-autonomous robots that operate within human spaces and play a beneficial role in the daily lives of ordinary people.
- ✓ Interaction between people and robots may potentially span physical, cognitive, task-based, social, or emotional dimensions.
- ✓ Human-robot interaction poses multi-faceted problems, requiring not only technical but also cultural, sociological, psychological, philosophical, and even ethical considerations.



# Special Issue on Human-Robot Interaction

- ✓ How to model the interaction of a human being with a robot?
- ✓ How to manage the physical, intellectual, and emotional exchange between human beings and robots?
- ✓ How to realize an effective communication of robots with the humans and their environments?
- ✓ These questions and many others are the stimulus for this Special Issue of the human-Robot interaction, aimed at gathering the latest results by robotics researchers facing the diverse problems related to human-Robot interaction.

## Getting to Know Socially Intelligent Robots

- ✓ Human-Robot Interaction and Communication is a quickly growing research area at the intersection of research fields such as robotics, engineering, psychology, ethology and cognitive science.
- ✓ Significant initiatives are currently underway funded by public, academic, governmental as well as industrial initiatives, exploring and aiming at advancing this research field and opening up novel and challenging applications.
- ✓ Robots moving out of laboratory and manufacturing environments face hard problems of perception, action and cognition.
- ✓ For robots to be accepted as assistants or companions in people's private homes and everyday environments technological solutions do not suffice: The human in the loop', as the potential customer and user will decide on the ultimate success of a home robot' as a product.
- ✓ Application areas that heavily involve human contact are a particularly challenging domain.



## Getting to Know Socially Intelligent Robots

- ✓ Human societies have easily assimilated new technologies, such as mobile phones, but it is less clear in which application areas robots will be accepted.
- ✓ Robots as embodied beings, physical, possibly humanoid or android entities that share our living environments and accompany our lives will have a certain degree of autonomy, initiative, cognitive skills and will communicate and interact with people in ways inspired by human-human contact.
- ✓ Interaction and communication of embodied physical robots with humans is multi-modal, and involves deep issues of social intelligence, communication and interaction that have traditionally been studied primarily in psychology and other areas.
- ✓ The design of a robot's behaviour, appearance, and cognitive and social skills is highly challenging, and requires interdisciplinary collaborations across the traditional boundaries of established disciplines.

# III. Propose the standardization for HRI Components



# What do users want?

| Requirements from users | We may need .. for standards |
|---|---|
| Simple and natural interfaces | Voice or<br>Vision(Gesture, face, etc.) interfaces |
| Simple and natural commands<br>(don't want remember voice or gesture commands) | Common or natural words interfaces<br>Natural gesture interfaces |
| If they have to remember keyword,<br>They want to remember same word of their language for the same service of different robot makers.<br>(If you want to know the weather, you just use "weather") | Keyword standards |
| If they have to use keyword and If they are Korean,<br>They want to use Korean keywords.<br>(If they want to know the weather, they want to use "날씨" means weather) | Keyword Internationalization Standard |

# What do robot makers want?

| *Requirements from robot makers* | *We may need .. for standards* |
| --- | --- |
| They want to know interface guide line For HRI functions (speech, vision, etc.) | → *Standard interfaces* |
| They want to use existing functions about HRI technologies | → *Standard components* |
| They cannot have enough machine power to use HRI functions and they want to use just HRI services from remote server. | → *Remote HRI Service Component Standards* |
| They want to upgrade functions or algorithms easily with low maintenance costs. | → *Algorithm Independent architecture* |



# What do App. Developers want?

| *Requirements from App. Or function developers* | *We may need .. for standards* |
| --- | --- |
| They want same interface independent on robot types and makers to develop applications. | → *Standard Interfaces* |
| They do not want to develop new applications for the same services or functions again for each different robot | → *Standard Components Or remote service Components standard* |

# Standard Items

| From users | From makers | From App. developers |
|---|---|---|
| *Voice or Vision(Gesture, face, etc.) interfaces* | *Standard interfaces* | *Standard Interfaces* |
| *Common or natural words interfaces*<br>*Natural gesture interfaces* | *Standard components* | |
| *Keyword standards* | *Remote HRI Service Component Standards* | *Standard Components Or remote service Components standard* |
| *Keyword Internationalization Standard* | *Algorithm Independent architecture* | |



# Standard Items

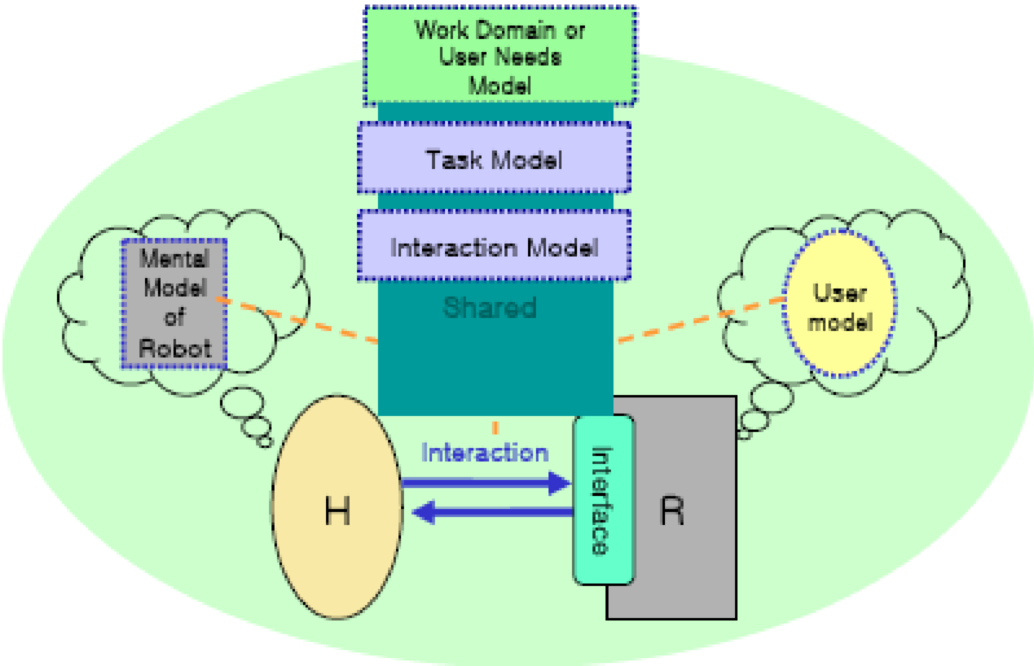
| From users | From makers | From App. developers |
|---|---|---|
| *Voice or Vision(Gesture, face, etc.) interfaces* | *Standard interfaces* | *Standard Interfaces* |
| *Common or natural words interfaces*<br>*Natural gesture interfaces* | *Standard components* | |
| *Keyword standards* | *Remote HRI Service Component Standards* | *Standard Components Or remote service Components standard* |
| *Keyword Internationalization Standard* | *Algorithm Independent architecture* | |

Sense/Expression Interfaces

Sense/Expression Information model

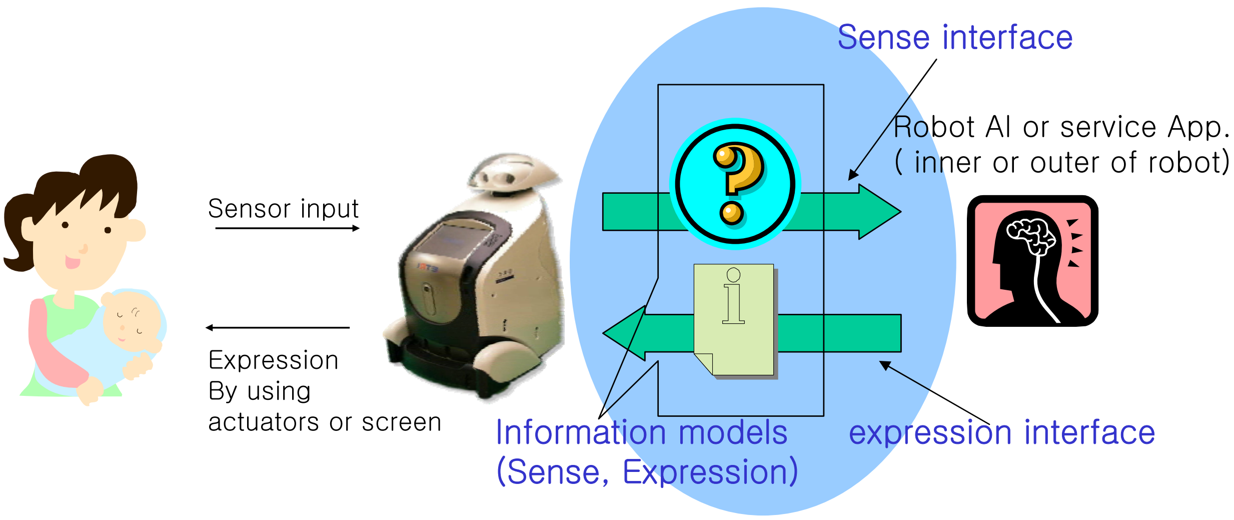
Sense/Expression Architecture

# Model of HRI Component Function






# Interaction Model Architecture for HRI
## What do we suggest for the standard



# Information Model

- Our proposed standard for HRI consists of one model and two interfaces
  – information model and recognition and expression interfaces
- Information model designs the types and structures of information objects as well as their management methods.
- The objects are two types
  ✓ recognition and expression.
- First of all, information model consists of the types structures and protocols of information from sensors and applications of robot.



# Sense Information Model

- Sense information model says When, Who say, What, Where from Human to Robot information
- Sense Information model consists of 4W Model
  ✓ Who information : face recognition, speaker recognition, etc.
  ✓ Where information : vision based location information, landmark based location info., etc.
  ✓ What information : speech recognition, gesture recognition, etc.
  ✓ When information: scheduler, clock, etc.

# Sense Information Model Detailed

Continue…

- Who information model
  - ✓ Generally it can be obtained by face recognition , speaker recognition, etc.
  - ✓ We define whoinfo as path + userId information
    - Ex) “//korea/south/taejon/etri/mrsong” : “//korea/south/taejon/etri/” Is path, “mrsong” is id
  - ✓ If the robot do not know the correct user id, they send WhoHint(face image or voice sound) to server
    - Ex) WhoHint = { Face Image or Voice sound }
    - if server do not know location
      - WhoHint = { (Image or Sound), Location path}



# Sense Information Model Detailed

Continue…

- Where information model
  - ✓ Generally it can be obtained by location finding algorithm
  - – We define whereinfo as address(or path) + position information
    - Ex) Can be direct type or near type
      - direct type: “//korea/south/taejon/etri/7thbuilding/L864/x=10;y=13”
      - near type : →have to translate direct position.
      - “//korea/south/taejon/etri/robotdivion/hriteam/song”

## Sense Information Model Detailed

Continue…

- What information model
  - Generally it can be obtained by speech recognition, gesture recognition, any other command recognition methods.
  - We define whatinfo as natural string or path command
    - Ex)
      - Natural string
        - "change the TV channel", or "show today schedule"
      - Command format with path information
        - TV ("//korea/south/taejon/etri/7thbuilding/L864/38TV"). channel.changeUp();
  - If the robot do not understand a correct command, the robot send WhatHint( gesture image, voice or any other command data) to server
    - Ex) WhatHint = { Image, voice or other command data(IR remote controller info, etc.) }



## Sense Information Model Detailed

Continue…

- When information model
  - Generally it can be obtained by clock and scheduler, etc.
  - WhenInfo show current time
  - We define wheninfo as time information
    - Ex)
      - Date Time : from clock
        - "2006-09-13-14:33:23"
      - If command use when info( in case users say "response until tommorow"), it can be processed robot AI with Whatinfo and Wheninfo.
  - If the robot do not send wheninfo, servers use their internal clock.

## Sense Interfaces

- Sense Interfaces use sense information model as their arguments and results.
- include interfaces for 4W info model and Hint model
- include sensor type interfaces and processor type interfaces
  - Sensor type: active, event style
  - Processor type: passive, it can be used in a application.



## Sense Interfaces Detailed

- Sensor(Event) type Interface:
  - On… interfaces
    - OnWho(WhoInfo), OnWho(WhoHint)
    - OnWhere(WhereInfo), OnWhere(WhereHint)
    - OnWhat(WhatInfo), OnWhat(WhatHint)
    - OnWhen(WhenInfo)
    - OnSee(ViewImage), OnSound(SoundData)
      - If robot cannot sense anything, In some case, robot send all images or sounds to be processed to server.

## Sense Interfaces Detailed

- Processor(Query) type Interface :
  - Get… interfaces
    - GetWho(), GetWhoHint()
    - GetWhere(), GetWhereHint()
    - GetWhat(), GetWhatHint()
    - GetWhen()
    - GetSee(), GetSound()
  - Get… interfaces have block mode and non-block mode



## Expression Information Model

- Expression Information model consists of
  - How information
    - Just say abstract expression commands: first step
    - It can be translated each methods for each robots: second step
    - Therefore, Expression information model require some special architecture

Expression model (abstract)
ExpressionHint model (detailed actions)
ExpressionRender architecture

## Expression Information Model Detailed

- Expression Model
  - It can be produced from Robot AI or service application in server
  - It can be processed on robot side or server side.
    - In case of robot side, robots process expression words
    - In case of server side, servers process words by calling robot control interfaces.





## Expression Information Model Detailed

- Expression Model
  - It have to be defined with standard word→ we have to define some robot expression word.
    - Ex) “say(<text>)”, “approach(<user>)”, etc.
    - It may match to same interface
      - “say(hello)” can match to robot.say(“hello”)
  - Some case, an expression model may not matched to an expression interface
    - Ex) expression can be “greeting” but a robot may not do “greeting” because they do not know the meaning of “greeting”
      - In this case, a robot use expression hint.

## Expression Information Model Detailed

- Expression Hint Model
  - If a robot cannot process expression, they want expression hint.
  - ExpressionHint shows action flow instead of abstract words.



## Expression Information Model Detailed

- Expression Hint Model
  - It can be differently implemented by each robot makers or robot service application developer .
    - ExpressionModel: Greeting
      - A maker:
        - Expression(Greeting) = ExpressionHint( TTS("Hello"))
      - B maker
        - Expression(Greeting) = ExpressionHint( robot.arm.shakeHands());

## Expression Interfaces

- Expression Interfaces use expression information model or expression hint model as their arguments and results.
  - Expression Interface…
    - ExpressionRenderer.Expression("some expression")
    - ExpressionRenderer.Expression("expression",ExpressionHint hint)



## Expression Renderer

- Expression Renderer process Expression, ExpressionHint
- If Renderer cannot process Expression, it try to get ExpressionHint from robot makers.
- Renderer is a process engine or task engine.
- It can be placed on application server or robot side.

# Conclusion

**We need to many opinions**

**and join from others**

**Thank You**

robotics/2006-12-20

# Infrastructure Working Group

Meeting Summary

December 6, 2006

Washington, DC, USA

*Chairs*:

Noriaki Ando

Saehwa Kim

Rick Warren

# Overview & Goals

- *Primary activity*: RFP for deployment and configuration (D&C) of RT components
- *Last meeting*: RFP schedule postponed pending more information about existing standards
- *This meeting's goals*
  - Learn more about software radio components
  - Restart schedule for RTC D&C RFP

# Joint Meeting with SBC

- Meeting attended by 2 SBC co-chairs
  - Jerry Bickle of PrismTech
  - Keith Richardson of Mitre
- Presentation about SW radio component framework from Jerry
  - Slides (PDF): robotics/2006-12-06
- We gave them RTC introduction
- Q&A



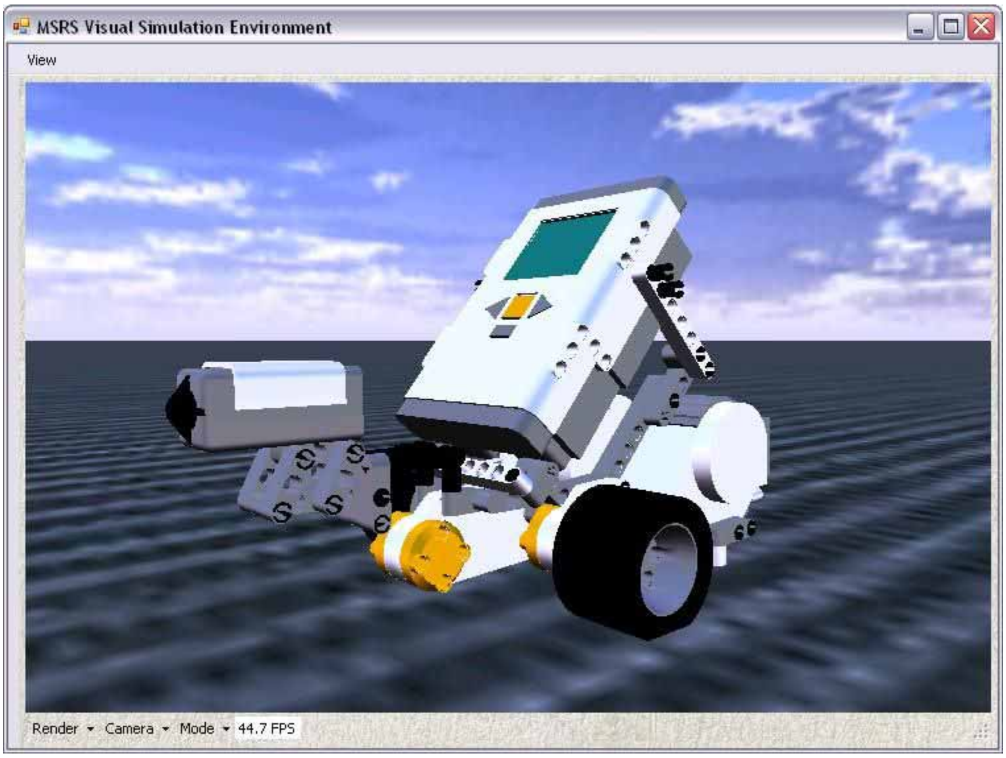
# Roadmap: San Diego (*March*)

- Talks (may be plenary or WG activities)
  - Zeligsoft SW radio tool demonstration
  - MS Robotic Studio presentation (Bruce)
- Have D&C RFP requirements ready to discuss
  - Discuss on wiki between meetings
  - Several organizations have homework to supply their requirements
  - *Requirements will be trimmed down*
    - Don't try to solve every problem in 1 RFP
    - *Open question*: what degree of interoperability should be targeted initially?

# Roadmap: Brussels (*June*)

- Issue draft RFP (3 wks before)
- Open question: issue RFP through our DTF or through MARS?
  - Domain members cannot respond to MARS (platform) RFP
  - Platform members cannot respond to Robotics (domain) RFP
  - Contributing members can respond to either
  - *Potential responders: which membership do you have?*

# Lego® NXT® and Microsoft Robotics Studio User Report






# Goals and Overview – OMG Dec '06

Demonstrate *interoperability* between *Microsoft's Robotics Studio and Lego NXT robots*.

MRS has proven to be much harder to use and program than we expected, and other team priorities limited our resources. So we have retrenched to explore the simplest available existing tutorials and demonstrations.

We expected this "off-the-shelf" tutorial to be trivial, but even this has proven more difficult than planned.

Robots have a long way to go to be truly accessible and easy to use.

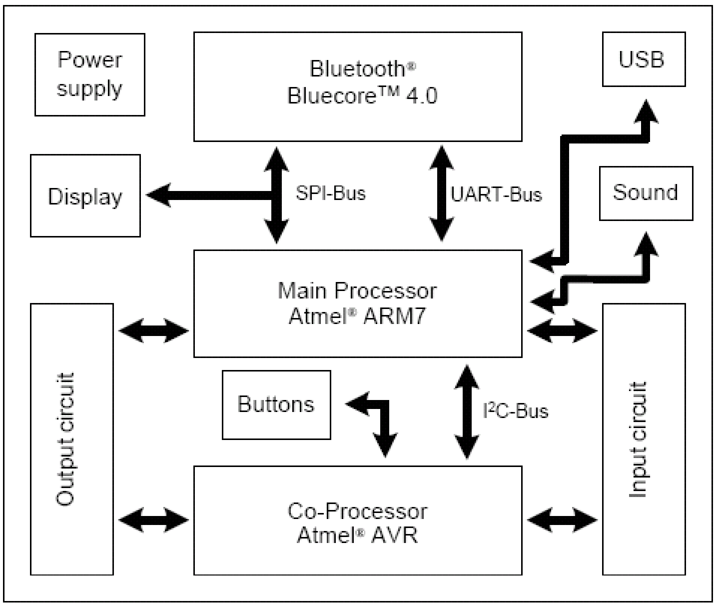
# Lego NXT – what's new

Lego NXT is a considerable departure from RIS/RCX:

- "studless" construction: stronger & more flexible
- Arm7 32-bit controller, I2C for smart motor and sensor control (yeah!) with easy expansion.
- 3 motor outputs which drive new servo motors with feedback and high torque
- 4 sensor inputs, and sonar and sound sensors
- Bluetooth serial port and USB 2.0
- Graphical LCD, 60x100 pixels
- SW and HW is much more open (thanks, Lego!)



# NXT Brick Block Diagram




Note the AVR co-processor which handles ADC and PWM

# Lego NXT – details (1)

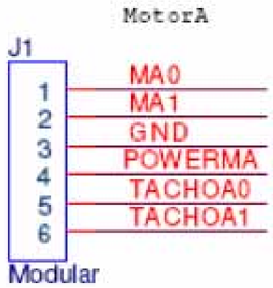
- Atmel AT91SAM7S256, 256 KB Flash, 64 KB RAM, 48 MHz, plus ATmega48, 4 KB Flash, 512 Byte RAM, 8 MHz. ARM and AVR communicate with I2C at 380 Kbits. JTAG is possible to both (but voids the NXT warranty).
- AVR does motor PWM, ADC and power management.
- Bluetooth: CSR BlueCoreTM 4 v2.0 +EDR System, Serial Port Profile (SPP), 47 KByte RAM, Ext 8 MBit FLASH, 26 MHz. BT Class 2 (~10 m) and 4 NXT can connect at <= 220 Kbps.
- USB 2.0 at 12 Mbps



# Lego NXT – details (2)

- Four input ports, I2C interface @ 9600 bps (master-only, with 16-byte input and output buffers) plus analog input. Port 4 has an RS485 transceiver, IEC 61158 Type 4/EN 50170 compliant, “for future expansion” at 921.6 Kbps. Sonar is only current I2C sensor
- Three output ports, with 2 x PWM out plus 2 x digital input.
- LCD uses SPI @ 2MHz to UltraChip 1601 LCD controller.
- Sound output using PWM from ARM7, and Sunplus SPY0030A amplifier chip driving a 16-ohm speaker

# Output Signals

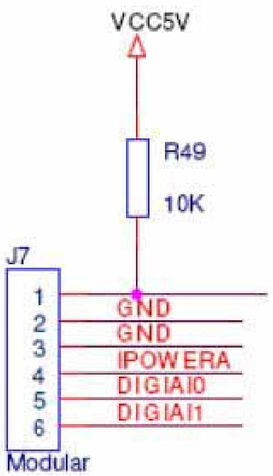



| | |
|---|---|
| Pin 1, MA0 | PWM output signal for the actuators |
| Pin 2, MA1 | PWM output signal for the actuators |
| Pin 3, GND | Ground signal related to the output supply |
| Pin 4, POWERMA | 4.3 Volt output supply |
| Pin 5, TACHOA0 | Input value that includes Schmitt trigger functionality |
| Pin 6, TACHOA1 | Input value that includes Schmitt trigger functionality |

Three motor outputs, all identical



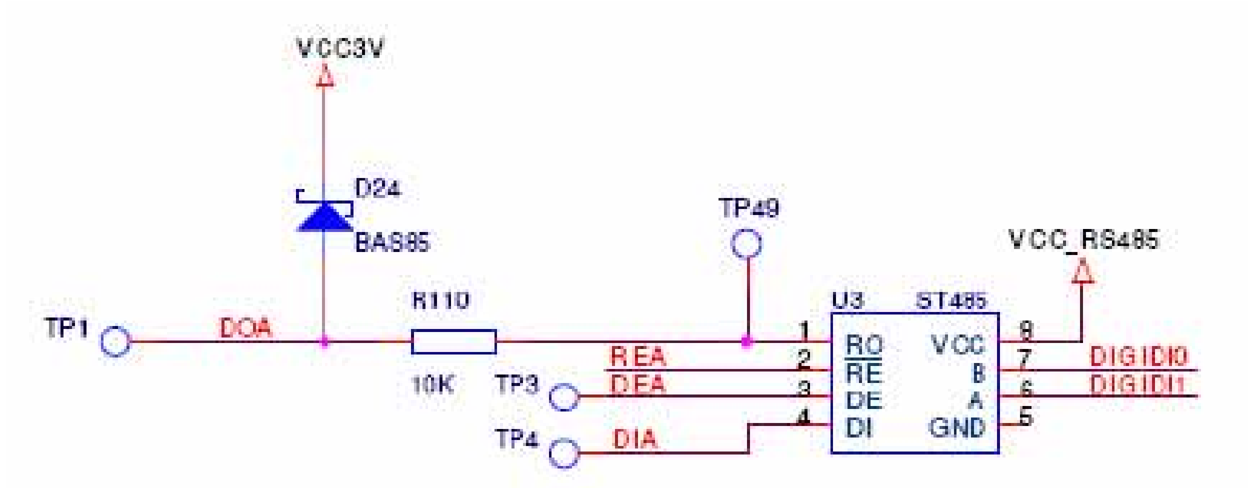
# Input Signals




- Pin 1, ANA — Analog input and possible current output signal
- Pin 2, GND — Ground signal
- Pin 3, GND — Ground signal
- Pin 4, IPOWERA — 4.3 Volt output supply
- Pin 5, DIGIAI0 — Digital I/O pin connected to the ARM7 processor
- Pin 6, DIGIAI1 — Digital I/O pin connected to the ARM7 processor

4 inputs, all have this structure, but Port 4 is special

# Input Signals on Port 4 only




Port 4 also has this 922 kbit high speed circuitry for "future expansion"



# NXT Power Consumption

| Supply voltage | Current | | Effect (Battery = 9 Volts) | |
|---|---|---|---|---|
| | Max [mA] | Normal [mA] | Max [mW] | Normal [mW] |
| No load on motors | | | | |
| 9 Volt | 339 | 114 | 5184 | 1422 |
| 5 Volt | 271 | 112 | 1744 | 448 |
| 3.3 Volt | 72 | 38 | 410 | 216 |
| Load on motors | | | | |
| 9 Volt | 2901 | 848 | 26109 | 7632 |
| 5 Volt | 271 | 112 | 1142 | 307 |
| 3.3 Volt | 72 | 38 | 410 | 137 |
| | | | | |
| Standby | 46 uA assumed standby current due to brown out detection | | | |

# Architecture Coincidence?

- By sheer coincidence, this ARM- and AVR- architecture is also used in the SunSPOT wireless sensor. SunSPOT uses different chips but the general idea is very similar.
- Perhaps this proves that a good idea can occur to a lot of people at the same time.
- Expect to see a small-controller/big-controller architecture in a lot of new embedded and robotic systems
  - Small controllers are now cheaper than CPLD, FPGA or even simple “glue logic”
  - Small controllers are more flexible (ADC, memory, UART)
  - Small controllers can be easily updated and debugged *in situ*



# Lego NXT – software

NXT software runs on PC and Mac. It's based on National Instruments Labview. Educational and consumer versions (discussed here) are available.

Graphical programming environment easy for children

USB or BT deployment

Other languages (C, Java) are available as third-party products (some open-source). Not currently mature

ARM7 and AVR gives Lego a capable controller

Little/no implementation of *future* features yet

Robot-Fusion on java.net
OMG Robotics DTF Washington DC 2006 Dec 06
Bruce Boyes, Robotic Devices and Data Profiles WG co-chair

# Microsoft Robotics Studio

MRS, Microsoft's entry into the *robotics market* (whatever that is), fills a middleware void by providing:

- Service-oriented abstraction between hardware and high level behaviors. This architecture has both good and bad points.
- Re-usable components, once created
- Excellent, scaleable, real-world simulator engine (gravity, variable surface friction, HW accel)
- Browser interface
- Runs only under Windows but supports arbitrary remote robot hardware

FYI: Is this a a data-centric, vs O-O system?



# Have a bar of SOAP…

SOAP (originally Simple Object Access Protocol) is a protocol for exchanging XML-based messages over computer network, normally using HTTP.

SOAP forms the foundation layer of the Web services stack, providing a basic messaging framework that more abstract layers can build on.

The original acronym was dropped with Version 1.2 of the standard, which became a W3C Recommendation on June 24, 2003, as it was considered to be misleading.

# MRS Architecture - DSSP

- DSSP is a simple SOAP-based application protocol that defines a lightweight service model with a common notion of service identity, state, and relationships between services.
- DSSP defines a set of state-oriented message operations that provide support for structured data retrieval, manipulation, and event notification.
- DSSP defines applications as compositions of services interacting in a decentralized environment. The functionality provided by DSSP is an extension of the application model provided by HTTP and is expected to be used as an addition to existing HTTP infrastructure.
- Services are lightweight entities that can be created, manipulated, monitored, and destroyed repeatedly over the lifetime of an application



# DSSP Terminology (1)

- Service - a computational unit that has identity, state, behavior, and context
- Application - a composition of services that can be harnessed to achieve a desired task through orchestration
- Services have a globally unique identifier
- Service state - data representing a service at a specific time
- Service behavior/contract - The combination of the content model describing the state and the messages exchanges that a service defines for communicating with other services
- Contract identifier - The globally unique identifier for the behavior of a service.

# DSSP Terminology (2)

- Partner - A labeled reference representing a behavioral relationship between services
- Service context - contains information about a service instance including which its contract and partners
- Message - a one-way message, request, or response participating in a DSSP operation
- Request - a message participating in a request-response message exchange pattern.
- Response - a message participating in a request-response message exchange pattern.



# DSSP Services

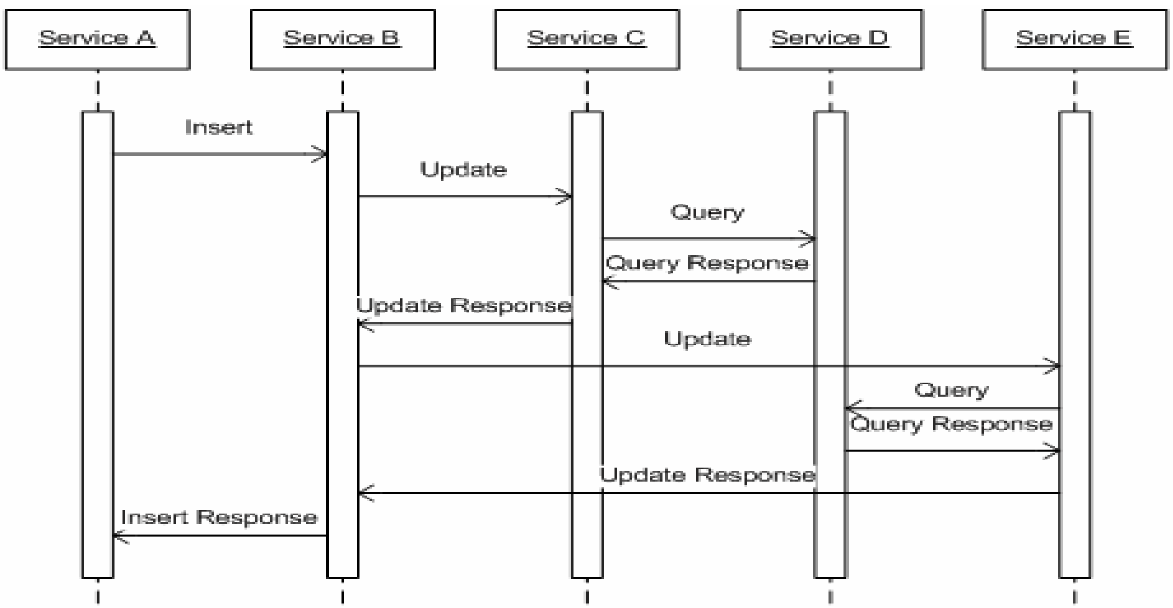
- A service consists of:
  - Identity – The globally unique reference of the service.
  - Behavior – The definition of the service functionality.
  - Service State – The current state of the service.
  - Service Context – The relationships the service has to other services
- Services can expose their state and hide their behavior, and also the other way round?
- DSSP enables but does not require a shared data model across all services.

# DSSP Service State

- The state of a service representing a motor may consist of rotations per minute, temperature, oil pressure, and fuel consumption.
- A service representing a work queue may contain a list of all queued work items and their current status. The work items themselves may be services allowing the work queue to simply refer to them using their identity.
- A service representing a keyboard may contain information about which keys have been pressed.
- *Any information that is to be retrieved, modified, or monitored using DSSP must be expressed as part of the service state.*



# DSSP Composite Services




Service A has partner B. Service B has partner C and E. Service C has partner D.

## Some DSSP Service Operations

- Services are CREATED and DROPped
- Initialization by creation state passing by reference or later by REPLACE
- Service state can be accessed via QUERY (structured) or GET (unstructured, gets entire state).
- LOOKUP returns service ID, contract ID and context
- Service state can be modified by INSERT, UPDATE, UPSERT, DELETE, and REPLACE operations



## DSSP Events

- DSSP defines an event as a state change in a service.
  - Event notifications are one-way messages; there is no provision for an acknowledgement
  - Request-response messages are supported, but there is no time constraint, and any timeouts must be provided by the programmer
  - SOAP faults are generated
- A service receives events by the SUBSCRIBE operation
- Subscriptions can be filtered
- Subscriptions can't be “empty” but a query can have an empty result

# DSSP Service Pro and Con

- Pro
  - No class diagram or hierarchy
  - Everything must be a service
  - On the surface this appears simple, perhaps elegant
- Con
  - Every device needs a service proxy, so this means you might need a lot of proxies
  - Simple devices suffer some latency and overhead due to the proxy
  - Everything must be a service, even “devices”, and so far, most robotic sensors are presented as devices.
  - In practice is it simple and elegant? Not sure yet...



# DSSP Questions

- Service vs Device
  - Simple devices might be best as devices?
  - Composite devices could offer a service, e.g., a camera data stream with a pan/tilt device, or a localization service which relies on multiple devices for its data.
- Concurrency and threading are claimed to be handled by DSSP but the details are not clear to me
- Security of data and control messages and access?

# Why MRS is Compelling

Ability to realistically simulate robots and environment without having actual hardware.

Re-usable device services. XML data.

Debugger and monitor, accessed via a local or remote web browser. Actual device data can be captured and then played back in the simulator.

Support for C# and VB and J# (in theory anyway)

Device runtime can be small and simple, scalable – $200 PIC to X86, video, $1M DARPA vehicle



# Evolutionary, not Revolutionary

All of the MRS capabilities have been shown previously, but AFAIK not in one open, affordable, and commercially viable package.

Microsoft seems committed to grow and maintain MRS and has the resources to make it great.

Microsoft may have the clout/pull to induce vendors to support this middleware which ultimately will simplify the work of robot users and researchers. Leading vendors are already supporting it.

These reasons are why it's interesting to OMG Robotics DTF.

# Where things started to bog down

MRS requires every transducer to be supported with a DSSP service DLL. These are most easily written in C# (fine if you already know C#). If you have a lot of transducers, this means you must write a lot of DLLs. You only need to do this once... well, also at release of new MRS or Windows versions.

MRS documentation is still in a preliminary state.

Building some custom hardware for Vex, working out 802.15.4 radio protocols, etc is taking more time...

Team member's main jobs kept intruding on resources for this project (typical volunteer problem...)



# So, take the simplest approach...

MRS has Lego NXT tutorials with prebuilt code. What could be simpler?

But first, you need a compatible PC BT adapter, and the NXT BT setup instructions are minimal.

NXT claims “your first robot in 30 minutes”. It took me an hour, with help from a 4- and 8- year old.

Upgrading the NXT firmware cause the NXT brick to fall victim to the dreaded “clicking brick syndrome”. The Lego website was no help. Two hours of Googling and rebooting finally got it working again.

## Some findings…

- Web browser UI
  - May have to be IE, in my limited experience? What other remote access, even limited, is available on non-Windows remote terminals?
  - Multiple access?
- User applications execute on .Net Runtime
- DSSP proxies services
  - Each transducer or service requires a DLL
- Transport to remote robots is media-agnostic
  - 802.11 and BT are supported now, though BT has issues
  - We plan to support 802.15.4



## More of what we learned so far…

- Everything takes longer than planned.
  - I guess we already knew this, but wished otherwise.
  - Learning new tools takes even longer…
  - Documentation is important!
- If your product has know problems, isn't it better to admit them and publish the solutions yourself?
- I seem to have a knack for breaking things and experiencing every possible problem.
  - This is good for debugging, and entertaining for others, but frustrating for me
- As much “openness” as possible is a Good Thing

# Why there is no demo today

- Starting the Lego service(s) is unreliable
  - BT adapter specifics are not standard
  - Instructions are hard for me to fully grasp
  - Multiple tedious manual command line steps
- NXT goes to sleep just as we manage to connect
  - There is probably a way to keep alive, but batteries...
  - AC power is available in the educational version
- Exception handling (or not) leaves us wondering where and why things aren't working
- All this software is still new and maybe we just aren't smart enough to use it yet...



# Goals for OMG Mar '07 (San Diego)

Robot-Fusion will show *interoperability* between Java-based robots and smart sensors within a service-oriented architecture. We hope to integrate *Microsoft's Robotics Studio*, *Java VEX/JCX robots* and *SunSPOTs* using *802.15.4 wireless* as well as the Phase 1 Lego NXT bots and Bluetooth.

In addition we want to demonstrate Java-programmed behavior, simulated in MRS and deployed on the JVex robots.

# Goals for OMG Jun '07 (Brussels)

Integrate mobile and stationary devices in a cooperative real-time task. We hope to expand the scope  to include other platforms and additional 802.15.4 wireless sensors such as the Crossbow and Telos motes.



# References

Lego NXT Mindstorms Hardware Developer kit manual, available online.

A simple blog with photos and links to numerous NXT and related resources is at www.mrs.systronix.com

Robot Fusion project at https://robotics.dev.java.net/ - you need to join to get code, but it's free

robotics/2006-12-22

# SAE AS-4 for Unmanned Systems

A Report of JAUS "Information Day"

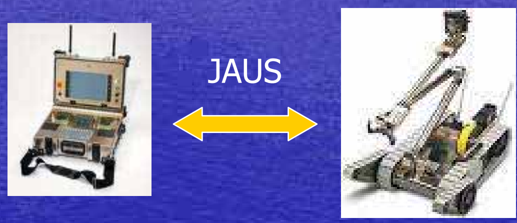
held on Nov 15, 2006

## Talk Overview

- Outline
  - Background of JAUS
  - Membership
  - JAUS moving to SAE Avionics System
  - Structure of SAE AS-4
    - Activities
    - Charters
  - Example of future direction
  - Example of existing specification

# JAUS Background

- JAUS defines components and standard messages for unmanned systems, *i.e.*, teleoperated, semi-autonomous
  - Dec 97
    - Working Group chartered
    - Scope was ground vehicles




  - Sep 02
    - JAUS re-scoped to include all unmanned systems, *e.g.*, ground, air, water, underwater
  - Apr 04
    - JAUS WG officially adopted by SAE
  - Future
    - Moving toward "service oriented" specifications

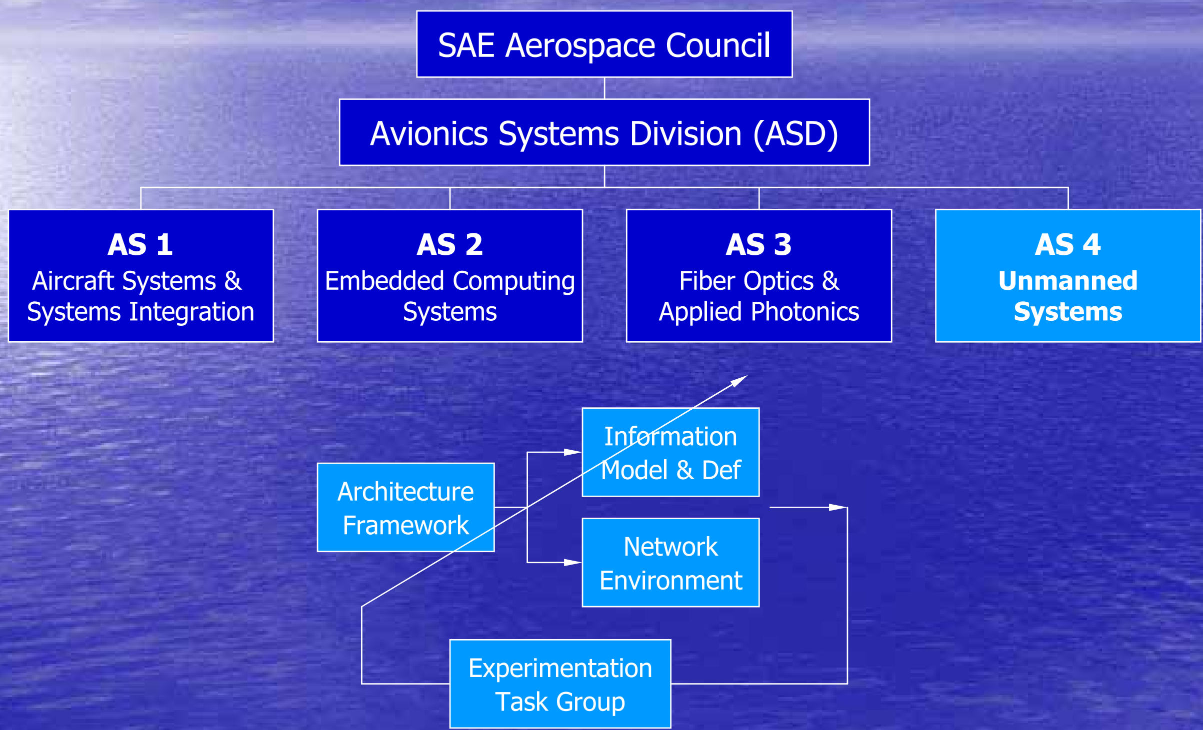
# Membership summary

- By the numbers
  - 119 members to date
  - 52 organizations
  - 71 voting members
  - 16 associate members
- Distribution
  - Users ~35%
  - Suppliers ~55%
  - Academia & National Labs ~10%

# Organization list

- Lockheed Martin
- L-3 Com
- Boeing
- Cybernet
- Autonomous Solutions
- DDC
- Applied Perception
- John Deere
- FAA
- Interoptek
- ARA
- Torch
- Navcom
- Caterpillar
- Radiance Tech
- Virginia Tech
- Foster Miller
- Applied Systems Intelligence
- Harris
- Re2
- AeroVironment
- OSD
- Battelle
- DOC NIST
- AAI
- General Dynamics
- Kairos Autonomi
- BAE
- Rockwell
- Northrop Grumman
- European Aeronautic Defence
- PSU
- Carnegie Mellon
- EG&G
- Univ.of Florida
- QinetiQ
- Jacobs
- Augusta Systems
- iRobot
- U.S.Army
- Defense Technologies
- Interoptek
- U.S. Navy
- Nomadio
- Textron
- U.S. Air Force
- Draper Labs
- Coroware
- EADS
- WINTEC
- Intelligent Innovations
- Azimuth
- SAIC
- DeVivo AST

# JAUS Has Moved!



# AS-4A Architecture Framework

- Activities
  - Establishes interoperability requirements
  - Defines capabilities, *e.g.*, mission planning, world models, mobility
  - Defines relationships between capabilities
- Charter (Abridged)
  - Applications and systems requirements
  - Functional architectures and frameworks
  - Reference architecture components
  - Other issues relating to Architecture Framework of Unmanned Systems

# AS-4B Network Environment

- Activities
  - Defines transport interface
  - Specifies message packaging
  - Optimize transport for efficiency, *e.g.*, header compression
- Charter (Abridged)
  - Network protocol adoption and specification
  - Ad-Hoc network discovery, registration, and configuration
  - Transport and addressing protocols
  - Other issues relating, *etc.*

# AS-4C Information Modeling & Definition

- Activities
  - Defines unmanned system capabilities and their interface (messages)
  - Defines system topology
  - Defines methods for message exchange
  - Defines coordinate systems
- Charter (Abridged)
  - Application- and domain-specific message composition
  - Information transfer protocols and control
  - Models and schemas
  - Other issues relating, *etc.*

# AS-4D Experimentation Task Group

- Activities
  - Performs experiments to test and evaluate proposed specification (changes) prior to publication
- Charter (Abridged)
  - Plan, coordinate and execute interoperability experiments
  - Support advancement of Architecture Framework, Network Environment, & Information Modeling subcommittees

# AS-4C In More Detail

- Information Modeling and Definition currently preparing two documents
  - AS5684, *i.e.*, JAUS Service Interface Definition Language (JISDL)
    - Defines language for specifying service interfaces
    - Uses a schema language for XML
      - Defines a formal structure to create messages
      - Uses finite state machines to define procedure rules
  - AS5710, *i.e.*, JAUS Service Set (JSS)
    - Defines a set of services within domain of unmanned systems, *e.g.*,
      - Communications
      - Navigation
      - Manipulation
    - Complies with JISDL

# JSIDL Example

```
#JAUS Service Interface Definition Language (JISDL)
#
default namespace = "urn:jaus:jssl:1.1"
datatypes xsd = "http://www.w3.org/2001/XMLSchema-datatypes"
start =service_spec
service_spec =
  element service_spec {
    attribute name { text },
    attribute id {xsd:anyURI },
    attribute version {
      xsd:string {
        pattern ="[0-9]{1,2}¥.[0-9]{1,2}(¥.[0-9]{1,2}¥.[0-9]{1,2})*“
      }
    },
  references?,
  element description {external "xhtml1_0-strict.rnc"},
  element assumptions {external "xhtml1_0-strict.rnc"},
  external "message_set.rnc",
  external "behavior.rnc"
…
```

# JAUS Service Specification in Future

**JAUS Service Set** ≈ Reference Architecture (message set) + JAUS Service Interface Definition Language [ Machine readable + Formal methods ]

# Latest Version of JAUS is v3.2

- Volume 1 – JAUS Domain Model (DM)
  - Written in language of "Acquirer" to model all requirements, known and anticipated, of Unmanned Systems
- Volume 2 – JAUS Reference Architecture (RA)
  - Architecture Framework – Maps DM requirements to JAUS message set
  - Message Definitions – Defines general types of messages and their headers
  - Message Sets – Defines the specific messages and their exact contents
- Volume 3 – JAUS Document Control Plan

# A Survey of JAUS v3.2

- JAUS currently defines 5 component groupings
  - Command and control
  - Communications
  - Platform
    - Related to control of vehicle
  - Manipulator
    - Related to control of arm
  - Environmental
    - Related to sensors that detect the environment

# JAUS v3.2 Platform Components

- Vehicle control components
  - Global (& Local) Pose Sensor, *i.e.,* global (local) position & orientation of platform
  - Velocity State Sensor
  - Primitive Driver, *i.e.*, open-loop teleoperated control
  - Reflexive Driver, *i.e.*, teleoperated control *w/ e.g.*, collision avoidance
  - Global (& Local) Vector Drivers, *i.e.*, closed-loop control of vehicle's heading, altitude, (pitch), (roll) and velocity
  - Global (& Local) Waypoint Drivers
  - Global (& Local) Path Segment Drivers

# JAUS v3.2 Manipulator Components

- Manipulator control
  - Primitive Manipulator, *i.e.*, open-loop joint-space control
  - Manipulator Joint Position (Velocity, Force & Torque) Sensors
  - Manipulator Joint Position (Velocity) Drivers, *i.e.*, closed-loop joint-space control
  - Manipulator End-Effector Pose (Velocity) Driver, *i.e.*, closed-loop Cart. space control
  - Manipulator Joint Move Driver, *i.e.*, closed-loop along sequence of points $\{\theta(t_1), \theta(t_2), \ldots\}$ under specified constraints
  - Manipulator End-Effector Discrete Pose Driver, *i.e.*, closed loop control along set of Cartesian points

# JAUS v3.2 Environmental Sensor

- Sensors
  - Range Sensor, *i.e.*, reports range data
  - Visual Sensor, *i.e.*, controls cameras
    - ID = 37
    - Function
      - has the responsibility of controlling the camera(s) of a subsystem
    - Messages
      - Query Camera Pose
      - Query Camera Count
      - Query Camera Capabilities
      - Query Camera Format Options
      - Query Image

# Contact Report

Contact of ISO/TC 184/SC 2
Makoto Mizukawa
Shibaura Institute of Technology



# ORiN and RAPI
## (Middleware for Industrial Applications)

- ORiN (Open Resource interface for the Network)
- RAPI (Robot communication framework and Application Program Interface) [subset of ORiN]
  - the presentation on RAPI was made at the ISO/TC 184 plenary meeting in Madrid 9-10 October 2006.
  - New Work Item Proposal was submitted on 31 Oct 2006.
    - Voting due: 31 Jan 2007
    - Liaison TC184 SC1/WG7,SC4/WG3/T24,SC5/WG6

## the next ISO/TC 184/SC 2 meeting

- 7 and 8 June, 2007
- Washington DC
- The dates 4-6 June are reserved for PT (Project Team) 10218, the new Project team *PT Robots in personal care* and Advisory Group *AG Service robots*, but these meetings are to be confirmed
- The following week, 11-15 June 2007, the International Robots and Vision Show will take place in Chicago, including the ISR and IFR meetings.



## ISO/TC184/SC2 AG1 and PT2 COEX, Seoul, Korea

- AG1: Advisory Group in *Service Robots* [Dr. Kotoku and Dr. Chung]
  - Oct. 17(Tue) – 18(Wed), 2006
- PT2: Project Team for *Robots in Personal Care* [Dr. Chi]
  - Oct. 19(Thu) – 20(Fri)
- Next ISO TC 184/AG meeting
  - La Plaine Saint-Denis (AFNOR's building), 2007-01-19

# IROS2006 Workshop
## Robotic Standardization

- Technically Sponsored with OMG Robotics Domain Task Force
- Contact (Organaizers):
  - Tetsuo KOTOKU (AIST)
  - YunKoo CHUNG (ETRI)
  - Makoto MIZUKAWA (Shibaura Inst. Tech.)
- Tuesday, October 10, 2006 Beijing, China



## OS059 RT (Robot Technology) System Integration Oct 19(Thu), 20(Fri)

- Organizers:
  - Makoto MIZUKAWA (Shibaura Inst. Tech.)
  - Yun Koo Chung (ETRI)
- 20 papers
  - TA12(6)
  - TP12(4)
  - TE12(5)
  - FA12(5)

SICE-ICCAS 2006 SICE-ICASE International Joint Conference 2006
October 18(Wed.)-21(Sat.), 2006 in BEXCO, Busan, KOREA

# KIRSF – Contact Report

**Robotics DTF (Washington DC Meeting)**
**Date: December 6, 2006**
**Reporter: Yun Koo Chung**

1. KIRSF Standardization Activities

– KIRSF adopted 24 standards in the following areas:

- Intelligent robot terminology definition
- Testing and evaluation
- Safety guide of robots
- Language Modeling of URC robots
- Communication protocols and message format for URC
- Robot interface framework for Device Abstraction
- APIs for Robot User identification
- Hardware interface

※ KIRSF: Korean Intelligent Robot Standardization Forum

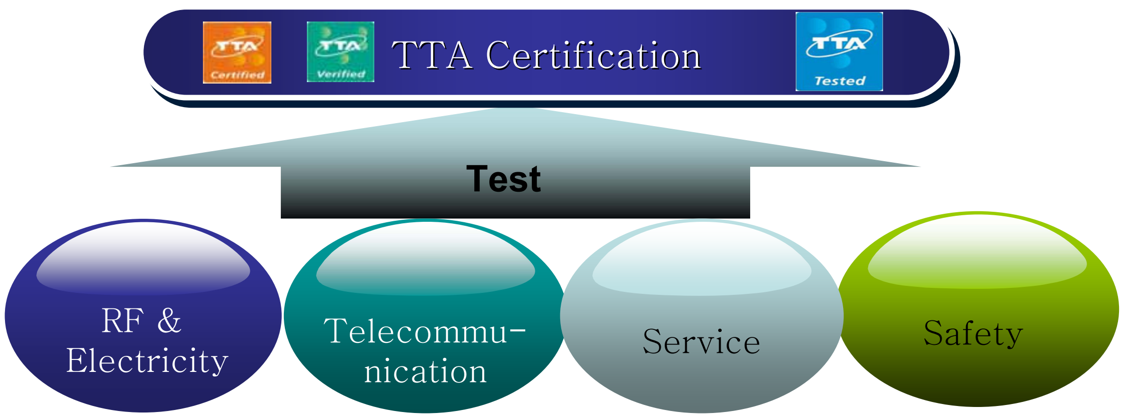


KIRSF – Contact Report

Robotics DTF (Washington DC Meeting)
Date: December 6, 2006
Reporter: Yun Koo Chung

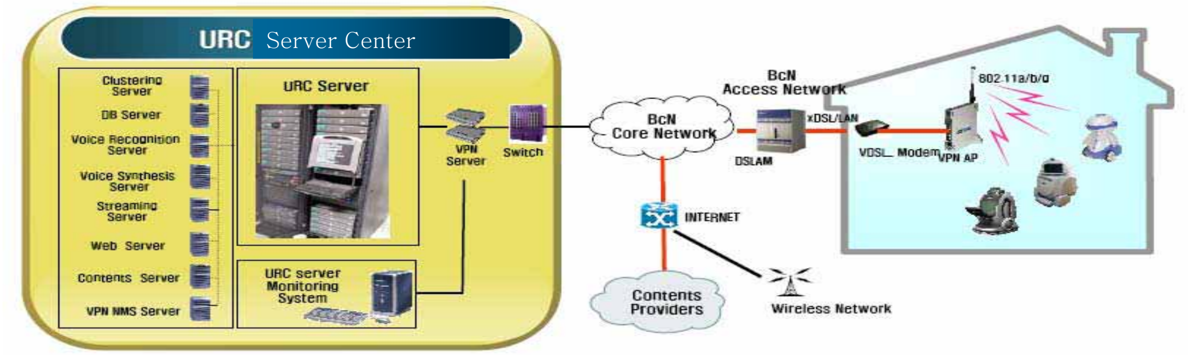
2. TTA Certification Test

- URC network robot products in pilot business were tested for authentification of robot quality assurance. Some test is based on the TTA standards of protocols and profiles.
- Test result: It was not satisfied for certification mark.
  "TTA tested" mark were given with test data sheets..



# KIRSF – Contact Report

**Robotics DTF (Washington DC Meeting)**
**Date: December 6, 2006**
**Reporter: Yun Koo Chung**




3. URC robots in Pilot business

- URC home robots were tested by TTA.
- 1000 home robots and several public service robots have been delivered to KT company from October, 2006.
- All robots are being tested and evaluated for quality satisfaction by KT. Currently about 300 robots have been distributed and serviced in apartment complexes of several cities. KT is monitoring responses from users.

Robotics/2006-12-XX

# KIRSF – Contact Report

**Robotics DTF (Washington DC Meeting)**
**Date: December 6, 2006**
**Reporter: Yun Koo Chung**

3. URC robots in Pilot business (continued)

< Home part example service>

<Post office example service>

Jupiter

Nettoro

Roboid

# KIRSF – Contact Report

Robotics DTF (Washington DC Meeting)
Date: December 6, 2006
Reporter: Yun Koo Chung

## 3. URC robots in Pilot business (continued)

- Perspectiveness of Network robot business:
  . Communication services, robots/parts/contents and software become foundations of a service business model for new IT service robot industry
  - ➔ Robot and IT infra based 'network robot ' improves the robot business.
  - ➔ Service model for various new business creation will be expected and should be created.

## 4. RUPI (Robot Unified Platform Initiative)

- All standards together from URC technology are called "RUPI".
- Currently seeking a project fund for RUPI project is on going.

## Join Us!

OMG's **Robotics Domain Task Force** is leading the work that will lead to standards in robotics software design and development. In the past, most robotics software initiatives had been developed independently. The Robotics DTF has begun a dialog with vendors, end users, researchers, robotics organizations and other interested parties to lay the groundwork for a common platform-independent model of robotics software development.

The first step in this dialog was to issue a Request for Information (RFI) on available products, projects, theories, models and requirements to support development of Service Robotic Systems based on distributed objects.

The Robotics DTF received a number of responses to the RFI and currently has several RFIs and RFPs in the works, so ***now is the ideal time*** to get involved! The Robotics DTF will meets at each OMG Technical Meeting.

### Robotics DTF Participants

The following organizations are already involved!

| | |
|---|---|
| ADA Software | AIST |
| ETRI | Fujitsu |
| Hitachi | IHI, Japan |
| Japan Robot Association | John Deere & Co |
| KAIRA | NEDO |
| NIST | Objective Interface Systems |
| PrismTech | Raytheon |
| Real Time Innovations | Schlumberger |
| SEC | Seoul National University |
| Shibaura Institute of Technology | Systronix Inc. |
| Technologic Arts Inc. | Thales |
| Toshiba | UEC, Japan |
| Universidad Politecnica, Madrid | Zeligsoft Inc. |
| Kangwon National University | |

### Contact Information

For more information about joining OMG and participating in the Robotics DTF or to be added to the Robotics email list, visit http://www.omg.org or contact us at robotics@omg.org or info@omg.org

# Robotics Domain Task Force

http://robotics.omg.org/

**"Robotics standardization is now emerging and becoming ready to use!!"**

## About Object Management Group & the Robotics DTF

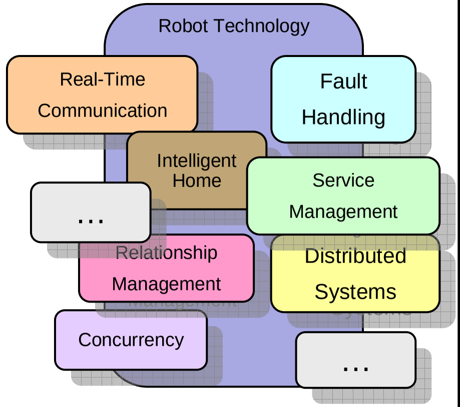
The Object Management Group (OMG), home to **UML**, **CORBA** and other such technologies, is a not for profit consortium targeting the production and maintenance of the computer industry specifications for interoperable enterprise applications.

OMG's **Robotics Domain Task Force** is leading the work that will lead to standards in robotics software design and development. In the past, most robotics software initiatives had been developed independently.

The Robotics DTF has begun a dialog with vendors, end users, researchers, robotics organizations and other interested parties to lay the groundwork for a common platform-independent model of robotics software development.

## 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 organizations like ISO, IEEE, OASIS to encourage interoperability
- coordinate with other task forces in OMG to develop common standards

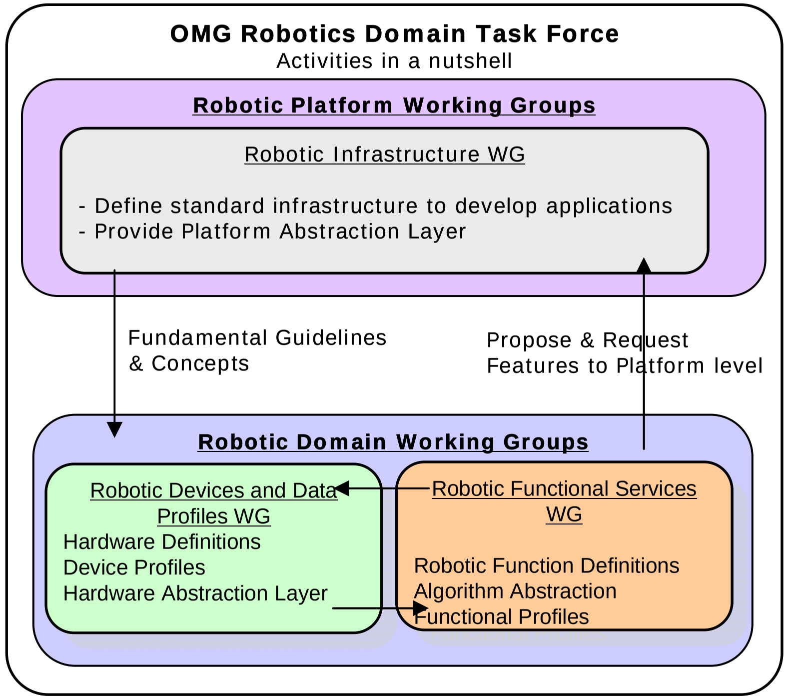
## Established Working Groups & their focus

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

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

## Robotics DTF

- Steering Committee
  - Publicity Sub Committee
  - Contacts Sub Committee
- Technical Working Groups
  - Infrastructure
  - Functional Services
  - Devices and Data Profiles




## The need for Standards

The difference between a robotic vacuum cleaner and a Mars rover may seem vast, but they have a lot in common. For example, both require a method of sensing their environment and relaying information to the servos that control their movement. This commonality of design and function has created both an opportunity and a need for standards.

The goal of robotics standards is to increase interoperability, compatibility and reusability.

# Robotics-DTF/SDO-DSIG

## Joint Meeting
## Closing Session

## December 6, 2006

## Arlington, VA, USA

Hyatt Regency Cristal City Hotel





# Document Number

- robotics/2006-12-01 Final Agenda (Tetsuo Kotoku)
- robotics/2006-12-02 Anaheim Meeting Minutes [approved] (Hung Pham)
- robotics/2006-12-03 Steering Committee Presentation (Tetsuo Kotoku)
- robotics/2006-12-04 Roadmap for Robotics Activities (Tetsuo Kotoku)
- robotics/2006-12-05 Next Meeting Preliminary Agenda - DRAFT (Tetsuo Kotoku)
- robotics/2006-12-06 Software Radio Specification Overview (Jerry Bickle)
- robotics/2006-12-07 Plenary Openning Presentation (Tetsuo Kotoku)
- robotics/2006-12-08 Heterogeneous Network Middleware for a Personal Robot (Vitaly Li)
- robotics/2006-12-09 RTM on Java/HORB: Eclipse meets RT Components (Satoshi Hirano)
- robotics/2006-12-10 IEEE 1588: An Update on the Standard and Its Application (John C. Edison)
- robotics/2006-12-11 IEEE 1451 Smart Sensor Interface Standards (Kang Lee)
- robotics/2006-12-12 Super Distributed Objects - an example of implementation - (Shigetoshi Sameshima)
- robotics/2006-12-13 Anybot Studio (Soon-Hyuk Hong)

# Document Number

- robotics/2006-12-14 Robotic Devices and Data Profiles WG Report (Bruce Boyes)
- robotics/2006-12-15 Robotic Devices and Data Profiles WG Discussion (Seung-Ik Lee and Bruce Boyes)
- robotics/2006-12-16 Robotic Functional Services WG Report (Olivier Lemaire)
- robotics/2006-12-17 Robotic Functional Services WG Meeting Schedule (Olivier Lemaire)
- robotics/2006-12-18 Business case for the RFP on Robotic Localization Service (Olivier Lemaire)
- robotics/2006-12-19 Propose the Interaction Model Architecture for HRI Components Standardization (Su-Young Chi)
- robotics/2006-12-20 Infrastructure WG Report (Rick Warren)
- robotics/2006-12-21 Lego NXT and Microsoft Robotics Studio User Report (Bruce Boyes)
- robotics/2006-12-22 A Report of JAUS "Information Day" (Hung Pham)
- robotics/2006-12-23 Contact Report - ISO/TC184/SC2 (Makoto Mizukawa)
- robotics/2006-12-24 KIRSF - Contact Report (Yun-Koo Chung)
- robotics/2006-12-25 Robotics-DTF Flyer - DRAFT- (Abheek Bose)
- robotics/2006-12-26 Closing Presentation (Tetsuo Kotoku)
- robotics/2006-12-27 DTC Report Presentation (Yun-Koo Chung)
- robotics/2006-12-28 Washington Meeting Minutes - DRAFT (Yun-Koo Chung and Bruce Boyes)





# Publicity Activities

- Robotics Wiki is available
  http://portals.omg.org/robotics
- 4 page fly sheet
  Consult with OMG Marketing Stuff
  issue ver.1.0 in San Diego
- Robotics Information Day
  (Brussels Meeting)
  Set-up Program Committee

# Next Meeting Agenda

March 26-30 (San Diego, CA, USA)

**Monday – Tuesday (AM):**

**Steering Committee (Mon morning)**
**WG activity [3WG in parallel]**

**Tuesday (PM)-Wednesday :**

**Robotics-DTF Plenary Meeting**
•WG Reports, Guest and Member Presentation
•Contact reports
•Next Meeting Agenda

**Thursday:**

**WG activity (optional)**
**RTC-FTF Meeting**



# Potential WG Speaker

## Services WG:

- Toshio Hori (AIST) " Ultrasonic Tag System?"

## Infrastructure WG:

- Jerry Bickle (PrismTech) " Introduction to PrismTech Tool? "
- John Hogg (Zeligsoft), " Introduction to Zeligsoft Component Enabler 2.4? “

## Profiles WG:

- (Crossbow) “Product Introduction "

# Potential Plenary Speaker

## Technical Topics:

- Bruce Boyes (Systronix), " Microsoft Robotics Studio? “

## Standardization Activities:

- Hyun Kim (ETRI) " Introduction of URC Network Robot System in ongoing pilot business with its standardization "
- Erwin Prassler (IEEE/RAS/IAB) "Introduction to RoSta activities"
- Xu Hua (Tsinghua University) "Open Robot Controller Research and Its Standardization in China"

# Robotics-DTF

TC Meeting Date: 8 December 2006
Reporting Chair: Tetsuo Kotoku, YunKoo Chung, Hung Pham
Group page (URL): http:// robotics.omg.org
Group email: robotics@omg.org

## ➢Highlights from this Meeting:

- RTC Finalization
- Discussion of Drafting RFPs in three WGs in Robotics DTF

## ➢6 Special Talks (Tue, Wed Plenary Meeting)

- Heterogeneous Network Middleware by Vitaly Li (KangWon National Univ)
- RTM on Jaba/HORB: Eclipse meets RT Components by Satoshi Hirano (AIST)
- IEEE 1588 Precision networked time reference by John Edison(Agilent)
- IEEE 1451 Smart Sensor Interface Standards by Kang Lee ( NIST)
- SDO: An example of SDO by Shingetoshi Sameshima(Hitachi)
- Introduction to Anybot studio by Soon-Hyuk Hong (Samsung)

## ➢ 3 Working Group activities (Mon~Tue) & Reports:

- Four RFP and 4 Roadmaps discussed:
- Topics: APIs for Device Abstraction, Hardware resource profiles, localization service, Deployment & configuration for components
- WIKI home page was made to activate more reports for WGs.


# Robotics-DTF

•TC Meeting Date: 8 December 2006
•Reporting Chair: Tetsuo Kotoku, YunKoo Chung, Hung Pham
•Group page (URL): http:// robotics.omg.org
•Group email: robotics@omg.org


## ➢Other Issues, Reports and discussion

- ✓ 2 Talks
  - Lego and Microsoft Robotics Studio User Report by Bruce Boyes (Systronix).
  - A report of JAUS Information Day: SAE AS-4 for Unmanned Systems By Hung Pham ( RTI)
- ✓ Contact and Publicity SC report
  - ISO/IEC/TC184/SC2 by Mitzukawa (Shibaura I.T.)
  - KIRSF contact report by Yun Koo Chung (ETRI)
  - Publicity SC report by Abheek Bose (ADA Software): 4 page flyer finalized.
  - Information day in Brussels OMG TC Meeting approved.

## ➢Future deliverables (In-Process):

- RFP drafting for four topics in 3 WGs
- RTC finalization and revision adaptable to most of components

## ➢Next Meeting Agenda Discussion

- WG & Plenary meeting schedule
- RFP drafts and Robotics Information day