Dynamic Deployment and Configuration Standard for Robotic Technology Component: DDC4RTC

Noriaki Ando, AIST (DDC4RTC FTF Co-chair)
Seungwoog Jung, ETRI (DDC4RTC FTF Co-chair)
Overview

- RTC specification and its implementation
- Motivation
- DDC4RTC specification
  - RTC Specific features
  - ApplicationSupervisor
- Conclusion
# OMG RTC Family

<table>
<thead>
<tr>
<th>Name</th>
<th>Vendor</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenRTM-aist</td>
<td>AIST</td>
<td>C++, Python, Java</td>
</tr>
<tr>
<td>OpenRTM.NET</td>
<td>SEC</td>
<td>.NET(C#, VB, C++/CLI, F#, etc..)</td>
</tr>
<tr>
<td>miniRTC, microRTC</td>
<td>SEC</td>
<td>RTC implementation for CAN·ZigBee based systems</td>
</tr>
<tr>
<td>RTMSafety</td>
<td>SEC, AIST</td>
<td>Functional safety standard (IEC61508) capable RTM implementation</td>
</tr>
<tr>
<td>RTC CANOpen</td>
<td>SIT, CiA</td>
<td>Standard for RTC mapping to CANOpen by CiA (Can in automation) and implementation by SIT</td>
</tr>
<tr>
<td>PALRO</td>
<td>Fuji Soft</td>
<td>C++ PSM implementation for small humanoid robot</td>
</tr>
<tr>
<td>OPRoS</td>
<td>ETRI</td>
<td>Developed by Korean national project</td>
</tr>
<tr>
<td>GostaiRTC</td>
<td>GOSTAI, THALES</td>
<td>C++ PSM implementation on URBI</td>
</tr>
<tr>
<td>Honda R&amp;D RTM</td>
<td>Honda R&amp;D</td>
<td>C++, Python. FSM Component.</td>
</tr>
</tbody>
</table>
Background

• Component model standard and implementations
  – OpenRTM-aist and OPRoS
• No deployment standard for RTC

Same component model, but no interoperability between system description
Motivation

• Many RTCs are distributed spatially
• Systems would be constructed as RTCs aggregation
• System structure should be changed according to the environmental changes in run-time
DDC4RTC Specification

- RFP: Minneapolis meeting, Jun. 2010
  - mars/10-06-16 (Deployment and Dynamic Configuration (DDC) of Robotic Technology Components (DDC4RTC) RFP
- Submitters: ETRI, AIST
- Initial Submissions: Santa Clara Meeting Dec. 2010
- Approved by AB and TC: Jun. 2012

DEPL + SupervisorFSM = DDC4RTC

- DEPL: Deployment and Configuration of Component-based Distributed Applications Specification
- RTC: Robotic Technology Component specification
DDC4RTC Packages

- Consists of four packages
  - Component Data Model
  - Component Management Model
  - Execution Data Model
  - Execution Management Model
- Each package inherits same name package of DEPL specification.
Component Data Model

Port in DEPL and Port in RTC

Port models in DEPL and RTC are different

Port and Component in DEPL

Port and Component in RTC

Port itself is a service (RTC::PortService)
Application Supervisor

SupervisorFSM

RTC based Systems

State A

State B

State C

ApplicationSupervisor Service

SupervisorFSM Description
Behavior of ApplicationSupervisor and RTCs.

RTC-based system managed ApplicationSupervisor consists of RTC0 and RTC1. The system activated by an event from Application, and deactivated by an event from SensorRTC.
Conclusion

- A dynamic deployment and configuration standard: DDC4RTC was introduced.
  - Now finalization phase in OMG
  - FTF report will be submitted next June?, and the specification will be in public 2014.

- It is based on DEPL and RTC specifications in OMG.
- SupervisorFSM and ApplicationSupervisor is added for dynamic systems
- By reusing existing standard specification, most of parts could be shared and extension parts could be minimized.