Introduction to Robotic Technology
Component (RTC-1.1) Specification

Geoffrey Biggs
Intelligent Systems Research Institute
National Institute of Advanced Industrial Science and Technology
Japan
The Robot Technology Component standard

• OMG specification for component-based robot software.
• Aims for greater compatibility and reusability amongst vendors of robot software.
  – Not just the software itself but also the tools.
• An open specification that anyone can implement.
Benefits of the OMG RTC standard

• Manage the lifecycle of all components in a uniform way.

• Provides the data structures necessary to specify components and the related objects for a complete system.

• Separates functional specification and execution control.
  – Supports a variety of execution patterns.

• Static and dynamic component networks.
OMG RTC standardisation history

- September, 2005
  RFP: Robot Technology Components (RTCs) publication.
- February, 2006
  Initial Response: PIM and PSM for RTComponent submissions from AIST (Japan), RTI (America)
- April, 2006
  Merged submission
- September, 2009
  Accepted by the architecture board. Finalisation begins.
- August, 2007
  Finalisation completed
- September, 2007
  Finalisation result passed by the AB.
- April, 2008
  OMG RTC standard 1.0 published
- September, 2012
  OMG RTC standard 1.1 published
Package 1: Lightweight RTC

- Lightweight RTC
  - Stereotypes for components, ports, connectors, etc.
  - Component lifecycle
  - Execution contexts
  - Does not include the introspection functionality
  - For static component networks
Package 2: Execution Semantics

• Execution Semantics
  – Provides execution patterns commonly used in robotics
  1. Synchronous execution processing data (data-flow type)
  2. Stimulus-response or event-driven (FSM type)
  3. Mixture of execution methods (multi-modal type)
Package 3: Introspection

- Introspection
  - Interfaces for acquiring component information
  - Based on the OMG Super Distributed Object standard
  - Dynamic component networks
## 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>CAN·ZigBee</td>
</tr>
<tr>
<td>Dependable RTM</td>
<td>SEC/AIST</td>
<td>Functional safety (IEC61508) capable and certified</td>
</tr>
<tr>
<td>RTC CANOpen</td>
<td>SIT, CiA</td>
<td>CANOpen CiA (Can in automation) based RTC standard</td>
</tr>
<tr>
<td>PALRO</td>
<td>Fuji Soft</td>
<td>C++ implementation for small humanoids</td>
</tr>
<tr>
<td>OPRoS</td>
<td>ETRI</td>
<td>Korean national project</td>
</tr>
<tr>
<td>GostaiRTC</td>
<td>GOSTAI, THALES</td>
<td>C++ implementation for the URBI robot language</td>
</tr>
<tr>
<td>Honda RTM</td>
<td>Honda</td>
<td>C++ implementation</td>
</tr>
</tbody>
</table>
RTC RTF 1.1 overview

• Published September, 2012
• Fixed 8 issues and deferred 9
• Resolved changes were correcting the specification
  – Diagram fixes
  – Grammatical corrections
• Minor comments received from AB review
  – Missing #pragmas in IDL, XMI bugs