Open-source software in the RT-Middleware project

*Geoffrey Biggs, Noriaki Ando, Tetsuo Kotoku (AIST)

1. Introduction

The RT-Middleware project has now been continuing in Japan for a number of years. It has produced a variety of tools for robot developers, all of them open-source and freely available. In this article, we describe some of the important aspects of this project, and the contributions it has made to open-source software for robotics.

2. Middleware

At the core of the RT-Middleware project is the reference middleware implementation itself, OpenRTM-aist.

OpenRTM-aist [1] is a component-based framework for intelligent systems. It features a component model that governs the life-cycle of each component, managers for controlling deployment of components, a powerful properties system for fine control over component configurations, and an in-depth introspection system.

OpenRTM-aist implements the Object Management Group's Robot Technology Component (RTC) specification [2], which defines the introspection and life cycle management interfaces. This allows OpenRTM-aist to work with any introspection tool compliant with this standard.

OpenRTM-aist builds on the specification with execution context and data transport implementations. Currently it includes a periodic execution context and a composite execution context. A synchronised execution context is available with the OpenHRP tool, described later. For data transport, it provides two CORBA-based transports (one for request-reply and one for data streams), a TCP data stream transport, and a transport using the Data Distribution Service specification to provide a highly-efficient, highly-configurable, real-time transport [3].

2·1 The RTC specification

The OMG's RTC specification defines the interfaces for introspection and life cycle management of software components for robotic systems. This is a freely-available standard, and all the standards it references (such as the OMG's Super-Distributed Object specification) are likewise freely available, making it easy for open-source developers to create compatible middleware and tool implementations.

3. Tools

One of the most important factors in the usability, and therefore the success, of a middleware is the tools support around it. The RT-Middleware project features several tools that make creating and using RT-Components and RT-Systems easier. These tools are available from the project website [4].

An important concept in RT-Middleware is the idea of a complete, unified development environment for robotics. For this reason, most of the tools are implemented as Eclipse plugins. The developer can begin in one tool, such as the robot model editor, and continue on through the entire development process, including stages such as component development, RT-System design, and simulation testing, simply by switching the active perspective in Eclipse.

Some of the tools found in RT-Middleware are described below.

3·1 RTCBuilder

A graphical tool for creating components in a variety of programming languages. Developers can specify information such as the component's interfaces, configuration parameters and even documentation. This information is stored in an instance of the RTCProfile component model, which is then used to generate template source files in the developer's chosen language. The component information can be updated at any time, with the changes merged into the source files even after they have been edited by the developer. Currently, C++, Python, Java and C# are available as target languages.

3·2 RTSystemEditor

RTSystemEditor provides the system-oriented counterpart to RTCBuilder. It is a graphical tool for creating RT-Systems, networks of RT-Components linked together to form a complete application or a composite component. The tool can also be used for managing running RT-Systems, such as adding and removing components from the network and managing component life cycles. This allows the developer to easily test alternative combinations of components, providing rapid application development in robotics.

3·3 OpenHRP

OpenHRP [5] is a comprehensive robot simulator. Although originally a separate project, it has been integrated into RT-Middleware and now forms an important part of the complete tool chain. One of OpenHRP's strengths is accurate dynamics simulation, such as collision detection and gait simulation of humanoid robots. Several sample robot models are included in OpenHRP, including the HRP-4C humanoid robot and a mobile platform.

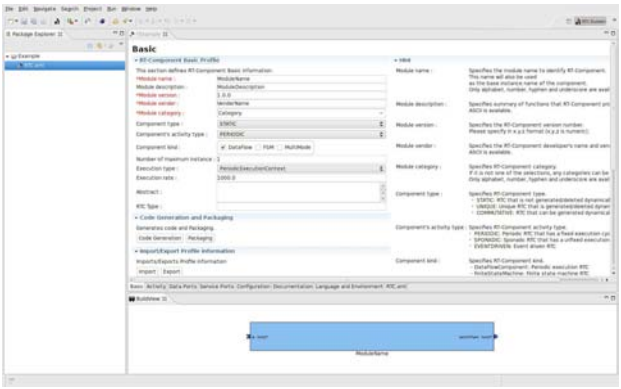

Fig.1: The RTCBuilder tool, used to create and update RT-Component templates.

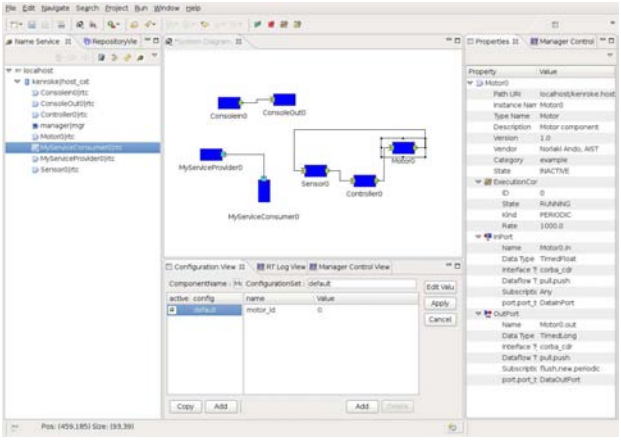

Fig.2: The RTSystemEditor tool, used to create and manage RT-Component networks and composite RT-Components.

3·4 rtshell

While most of the tools in RT-Middleware form part of a graphical tool chain, the project also provides resource-efficient command-line-based tools. These are contained in the rtshell toolkit. rtshell provides individual commands to perform nearly every function available in the RTSystemEditor tool, as well as additional functionality, such as debugging component inputs and outputs, and recording and replaying logs of transmitted data.

3·5 Libraries

Alongside the tools, there are open-source libraries available that allow developers to programmatically perform many of the same tasks:

rtctree A Python library for interacting with running RT-Components. This library can be used to inspect the RT-Components via the introspection interfaces (from the RTC specification), to manage aspects such as connections between components, and even manage the lifetime of components through the component manager's interfaces. The rtshell tools use this library for access to components' introspection and management interfaces.

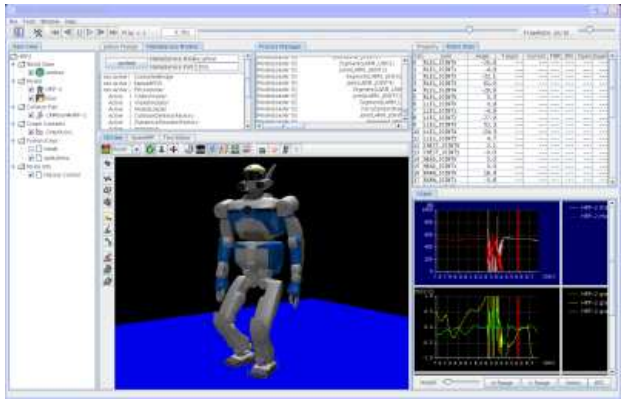

Fig.3: OpenHRP 3 provides RT-Middleware with a comprehensive simulator, including dynamics simulation.

rtsprofile A library that provides an execution-time model of a complete RT-System. The model is loaded from and saved to XML or YAML files. Through this library, and particularly when combined with the rtctree library, developers can perform processing on complete RT-Systems. The rtshell tools use this library for deployment management of RT-Systems.

4. Projects

As the RT-Middleware project has grown, so has the number of open-source RT-Components and tools provided by users of the project. A long-standing issue has been the difficulty in finding them. Searching the Internet for an RT-Component that meets your needs is difficult. This difficulty is compounded by out-dated web pages and documentation, as well as difficult-to-use or even broken software. When it comes to tools, finding available tools is so difficult that few even try beyond searching the official website.

The RT-Middleware project aims to make the job of finding and using existing software resources easier. To achieve this, it has transforming the official OpenRTM-aist web site into a project portal. This portal allows anyone to register a new project related to RT-Middleware.

To date, the project portal has attracted 44 RT-Component projects (of which several are component collections consisting of multiple components themselves, such as a collection of components wrapping OpenCV functions), 7 implementations of OpenRTM-aist or transports for them, and 10 tools (including complex tools such as OpenHRP). Included amongst

these are the implementations of OpenRTM-aist and tools produced by AIST. Software developed by other users includes a UML modeling tool for RT-Components, a tool for connecting with Scilab, and components to use the popular OpenRAVE planning library.

Because RT-Middleware focuses on usable software for robotics, an important consideration is that users are able to immediately begin using the components and tools that are available. As compiling something from source is often one of the headaches of using downloaded software, the portal encourages project submitters to provide binary installers alongside the source for their components and tools.

5. Conclusions

The RT-Middleware project has produced many open-source tools and components for robot development. Central to the project is OpenRTM-aist, an open-source reference implementation of the OMG's RTC specification. Around this have developed a number of tools for the steps involved in developing a robot, such as designing the robot model, developing components and complete component-based applications, and testing through a simulator. These tools are integrated into a complete tool chain as Eclipse plugins. In addition, the project has attracted the attention of many developers willing to produce new components and new tools as open-source software, making them freely available to other users of the project's software.

RT-Middleware has benefited greatly from releasing its products as open-source. The free and open nature of the software has encouraged its users to release their own, related software freely to others, growing the number of available software tools significantly beyond what a closed-source project would have attracted. The extra attention received has also provided greater opportunities for feedback and improvement.

References

[1] Ando, N., Suehiro, T., Kotoku, T.: A software platform for component based rt-system development: Openrtm-aist. In: SIMPAR '08: Proceedings of the 1st International Conference on Simulation, Modeling, and Programming for Autonomous Robots. pp. 87–98. Springer-Verlag, Berlin, Heidelberg (2008)

[2] Robotic Technology Component Specification (RTC). `http://www.omg.org/spec/RTC/` (2011)

[3] Data Distribution Service Specification (DDS). `http://www.omg.org/spec/DDS/` (2011)

[4] OpenRTM-aist official website. `http://www.openrtm.org/` (2011)

[5] OpenHRP3 Official Site. `http://www.openrtp.jp/openhrp3/jp/` (2011)