Development of RTC–CANopen for Seamless RT Middleware Expansion

Makoto Mizukawa
Dept. Electrical Engineering, College of Engineering, Shibaura Institute of Technology, Tokyo, Japan
Mizukawa@sic.shibaura-it.ac.jp

The component based approach for robotics development has been proposed. In this approach, robots are divided into various robotics elements as components and those elements are combined to be a complete robotics system. RT middleware is also developing as a platform on which RT functional elements (RT components) run.

Since RT middleware enables reuse and switch of RT components, it becomes possible to respond to various requirements from users by combination of RT components. And it also becomes possible to develop parts individually and make development projects smaller. In the result, each vendor can concentrate on parts and fields which they are specialized (Fig.1).

It is defined that a RT component is robotics component which provides some meaningful functionality, so devices like servomotors, various sensors, cameras, and so on are naturally called as RT components, more than that, combinations of these devices such as motor driven carts and arms are also called as RT components. In addition, they are not limited only for modules embedded in hardware, but software modules like different control algorithms also become RT components.

What RT components should provide, for example interface specifications and component models, is defined in the standard specification (Robotic Technology Component Specification) at OMG (Object Management Group) 1. However, this standard specification defines only functionality and state transitions RT components should provide as shown in Fig.2, and details about communication techniques are not described.

CANopen specified and maintained by CiA (Can in Automation) 2. CANopen supports not only CAN bus but also various native buses such as Ethernet, Flex Ray etc.

RTC–CANopen is a device profile for RT components, which consists of modularized robotics functional elements, to run on a CANopen environment. This profile describes necessary information to make existing
devices for CANopen work as RT components.

RTC–CANopen enables mutual collaboration between various devices connected to native buses set up for CANopen, and RT components running on general purpose networks. It becomes easy to combine them and build a robotic system.

A system built on RTC–CANopen consists of DeviceRTCs running on an embedded MPU, ProxyRTCs managing DeviceRTCs on RT middleware, and RTC–CANopen Servers managing the whole system as shown in Figure 3. ProxyRTCs and DeviceRTCs have correspondence relations. Using such relations, when a ProxyRTC connects to another ProxyRTC, their corresponding DeviceRTCs transmit data directly via a native bus.

RTC–CANopen has following key features:

◆ Lightweight
  Implement light RT middleware providing only necessary functions for embedded devices.

◆ Robustness
  Improve system robustness introducing monitoring methods for operations on native buses.

◆ Real–time
  Implement fast and reliable communication between components with the aid of advantages of native buses.

◆ Reusability
  Improve reusability of each element, since both hardware and software can be dealt as components.

◆ Flexibility
  It is possible to change system configurations flexibly, switching parts or changing combinations.

RTC–CANopen profile is defined by adding necessary information for devices to run as RT components, to the existing CANopen profile. Therefore, it is compatible with existing CANopen compliant products, and makes extension for robots easier. A set of tools are also provided, which generates robotics system definition files from definition files of CANopen (DCF files) automatically.

RTC–CANopen is a profile for using existing CANopen compliant products as components which are parts of robots. It enables improvement of robotic system development efficiency and realization of various requirements from users.

References
2) CAN in Automation, http://www.can–cia.org/