CoRoBA, a Framework for Multi-Sensor Robotic Systems Integration

Eric Colon
eric.colon@rma.ac.be

Research in robotics has moved from a limited capability single robot to multi-robot systems equipped with a plethora of sensors, leading to de-facto multi-processor and distributed applications. Because of the lack of standards, ad hoc solutions are generally developed resulting in difficulties to extend and reuse existing software. In order to increase efficiency of future developments, tools for integrating multi-sensor robotic systems are required.

This thesis proposes a solution to this problem with a framework called CoRoBA (Controlling Robot with CORBA). CoRoBA is made up of component based execution units. It comes with a 3D simulation application and utility programs for distributing and managing the live and run cycle of multi-process applications.

The implementation of the framework is based on several Design Patterns that make the design flexible, elegant and ultimately reusable. The execution unit in CoRoBA is a component. Components are independent execution units and have separated interfaces for the configuration and the actual functionality they provide. According to the classical control theory, components are divided in three categories, Sensors, Processors and Actuators. They form a chain along which information is transferred and like in classic control schemes, the data flow is unidirectional. Sensors read data from external devices and transmit them to other components. Processors process received data and forward results to components linked to output devices, which are called Actuators. This division provides a clear view of the functionality of each component and consequently facilitates their reuse in new applications.

In this framework, communication between components relies on the industry standard CORBA. CORBA has been selected because of its language and platform independence. Using such a standard simplifies the development and improves the interoperability with existing software. CORBA is actually a specification of the Object Management Group and the TAO (The ACE ORB) implementation has been chosen among others because it is an open source, efficient and standards-compliant real-time implementation of CORBA. The framework offers two different communication mechanisms; the first one is based on classical synchronous communication while the second relies on Events. In this mode, components exchange data by pushing Events through Event Channels that can be seen as pipes connecting suppliers and consumers of Events. Event based communication increases the flexibility of an application by decreasing the coupling between components.

Components are multi-threaded and posses different running modes for the transmission of events, namely PERIODIC, SYNCHRO and TRIGGER. In the PERIODIC mode, components produce events at regular time intervals. In the SYNCHRO mode, new output events are produced by the component when an input event is received and in the TRIGGER mode, an external signal must be received in order to process or produce an event. The availability of different modes increases the flexibility of the framework, each mode being actually useful in a different context.

Having a simulator offers many advantages. First of all it is tremendously cheaper than real robots and sensors, particularly when experimenting with multi robots systems. It allows focusing on intelligence and control and disposing of other, less interesting problems. It makes possible reducing the development time by trying different scenarios and algorithms before experimenting them in a real environment. A simulator also increases safety when developing and testing new control applications. For these reasons a Java based 3D multi robot simulator has been developed. Java provides comprehensive Application Programming Interfaces to communicate with CORBA objects and a 3D library (Java3D) for the modelling and rendering of virtual worlds. The robot simulator is responsible for the realistic motion of the robot by using geometric, kinematic and dynamic models, and takes care of the collision with fixed and moving obstacles like other robots. The simulated sensors produce measurement data that are injected in the application control loop. The software integrates seamlessly with the components of CoRoBA because all robots and sensors have a CORBA interface. The utilisation philosophy is to develop and tune control algorithms in simulation and to simply replace simulated by real components once satisfying results have been reached, no further modification of the Processor components being required.

Several distributed control applications have been implemented in order to validate the framework. Shared control and autonomous navigation applications involving different robots have been successfully tested in simulation. Development of multi-robots applications, distributed simulation and real robots and sensors has also been addressed. A qualitative and quantitative evaluation of the framework has shown that the proposed solution is efficient, usable and stable.