Hard Real-time CORBA (HRTC) for process control systems

Manuel Rodríguez*, Santos Galán, Ricardo Sanz, Carlos García, Rafael Chinchilla
Universidad Politécnica of Madrid, José Gutiérrez Abascal, 28006 Madrid, Spain

Abstract
Control systems for process plants are complex applications running in several interacting computers with varying degrees of integration. The construction, deployment and maintenance of the software system is a difficult problem and distributed object oriented technology offers a good way to deal with it. The open standard CORBA provides flexible middleware capable of integrating complex applications in heterogeneous environments. Even with recent advances in the real-time specification for CORBA, it is only suitable for soft real-time applications and do not deal with the tight requirements of closed control loops. In this paper, a process control testbed is used to unveil the hurdles toward the goal of CORBA control systems. The benefits of such technology are discussed.

Keywords: Networked control, Hard Real-Time, CORBA, Process Control, Distributed Object Computing

1. Introduction

Most present-day plant-wide control systems are very complex, constituted by diverse hardware and software components which interact with each other. With the incorporation of intelligent sensors, the computers reach even the lower level of the control hierarchy.

They are also distributed systems, different tasks run on different processors (computers, networks interfaces, PLC's...) and common resources are shared between processors. Distributed systems are designed to improve performance and increase system reliability.

The control system have been traditionally separated into several levels:

- **Field level.** This level is dedicated to the instruments (sensors and actuators) and basic regulatory control. It is communicated via fieldbus.
- **Process control level.** This level takes over the advanced and supervisory control (advanced controller, multivariable controller, model predictive control,...). This level also computes a local optimization. It is communicated via an ethernet based protocol.
- **Business level.** The upper level is dedicated to global optimization, scheduling and planning. It is communicated via ethernet.

* Author to whom correspondence should be addressed: mrod@diquima.upm.es
Although these levels have been always present in the process industry the control implementation has been evolving along the years. From the first direct digital control to the current Distributed Control System and to the future where the control system is totally distributed to field with the loops in individual devices. To implement these coming distributed systems efficiently and with enough flexibility, middleware seems to be the most appropriate tool to simplify the task (Sanz and Alonso, 2001). The construction, deployment and maintenance of the software system is an extremely difficult problem. Even though there are no silver bullets, object oriented technology offers a good way to build complex systems and when they are running in several, networked computers, distributed object technology has been demonstrated as a feasible way to cope with this complexity while keeping costs under control.

2. CORBA and Real-Time CORBA

2.1 What is CORBA?
CORBA (OMG, 2002) is an open standard that allows programmers to specify interfaces as contracts between servers and clients, these interfaces are specified using a language called IDL (Interface Definition Language). CORBA's key entity is called Object Request Broker (ORB). An ORB is a software bus capable of transmitting messages through a network, from clients to servers, in a transparent way. CORBA provides developers of distributed systems with a flexible middleware capable of integrate complex applications in heterogeneous environments. It was designed from the perspective of surpassing heterogeneity barriers and provide support for modularity and reuse. CORBA, however, was originally designed with large business applications in mind and is not perfectly suited for the construction of embedded control applications. This has changed recently with the RT CORBA specification.

2.2 Real Time CORBA
RT CORBA is an extension of the CORBA standard whose intention is aiding the design of real-time distributed applications (Schmidt, 2002). RT CORBA defines CORBA priorities which have corresponding native priorities on each operating system. RT CORBA (v1.0) has a fixed priorities Scheduling Service. A new RT extension (v2.0) specifically addresses dynamic scheduling.

2.3 Hard Real-Time CORBA
Present day CORBA specifications are suitable only for soft real-time applications. CORBA and its extension RT CORBA are not fully suitable to implement these systems because:

- They have only been designed to build systems with soft real time requirements. CORBA lacks of a real-time interoperable protocol, necessary to integrate control and real time systems.
- The Scheduling Service is incomplete, can not be dynamically reconfigured and does not provide a wide range of scheduling algorithms.
- The Interface specification needs to be extended to express temporal issues.
3. Process Control Testbed (PCT)

3.1 Description
The process is the neutralization of acetic acid (0.1M) with sodium hydroxide (0.1M). It has two control loops: one controls the pH and another one controls the temperature.

![Figure 1: Process Control Testbed process.](image)

The process has two feeds (see figure 1), the first one is the acid which is the one to be neutralized and the second one is the base feed. This feed is set by the pH control loop. There is a small reactor for the neutralization process and its output stream goes to a product tank through a weir. There is an additional loop for temperature control. This loop has no special relevance for the process but it is needed for the experiments to be taken.

![Figure 2: Hardware setup implementation of the Process Control Testbed.](image)

3.2 PCT Requirements
The PCT meets the following general requirements. It is *representative* of the basic characteristics of a process plant control system. It is *reconfigurable* meaning that it can change easily and adopt different configurations and topologies in order to comply with...
the representativity requirement. Finally, it allows mechanisms to make some experiments and to measure the results, so it is testable.
The hardware and software implementation of the PCT is shown in figures 2 and 3.

3.3 PCT Test 1: Closing control loops through networks
For distributed control systems where control loops are closed over a communication network or a field bus, the network can be a bottleneck. The following experiments related to CORBA Control loops have been performed under two network topologies. The first one is using an standard Ethernet Hub where collisions are not avoided. The second one is using an standard Ethernet Switch which splits up the collision domains.

![Diagram of network setup](image)

**Figure 3**: Software setup implementation of the CORBA pH control loop.

The experiments performed are:
CCS: The pH control loop was running and some disturbances where applied.
TI: Intensive data traffic. The network was heavily loaded and the performance of the pH loop was recorded.
CA: Concurrent Access. Multiple virtual objects where accessing the pH value at the same time. The performance of the pH loop was recorded.

| Table 1: Experiment results in milliseconds using the Hub |
| --- | --- | --- | --- | --- | --- | --- |
| Signals | CCS 41-25 | CCS 34-21 | TI 41-25 | TI 34-21 | CA 41-25 | CA 34-21 |
| Mean | 1,822 | 3,564 | 9,979 | 3,105 | 5,405 | 14,327 | 8,317 | 8,751 | 16,206 |
| Asymmetry coef. | 0,695 | 1,125 | -2,325 | 3,095 | 2,920 | 3,286 | -1,794 | 4,800 | 0,675 |
| Standard deviation | 0,055 | 0,385 | 0,317 | 2,788 | 3,138 | 13,316 | 1,415 | 11,307 | 6,747 |
| Average deviation | 0,048 | 0,290 | 0,124 | 1,684 | 2,053 | 7,293 | 0,874 | 5,684 | 5,943 |
Table 2: Experiment results in milliseconds using the Switch

<table>
<thead>
<tr>
<th>Signals</th>
<th>CCS</th>
<th>SWITCH</th>
<th>TI</th>
<th>CA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>41-25</td>
<td>34-21</td>
<td>41-41</td>
<td>41-25</td>
</tr>
<tr>
<td>Mean</td>
<td>0.946</td>
<td>1.821</td>
<td>10.009</td>
<td>0.923</td>
</tr>
<tr>
<td>Asymmetry coef.</td>
<td>5.425</td>
<td>1.444</td>
<td>0.093</td>
<td>6.302</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>0.089</td>
<td>0.033</td>
<td>0.288</td>
<td>0.116</td>
</tr>
<tr>
<td>Average deviation</td>
<td>0.045</td>
<td>0.026</td>
<td>0.115</td>
<td>0.046</td>
</tr>
</tbody>
</table>

Signal 41 is when the pH sensor sends the requested data. Signal 25 is the reception of the pH value from the sensor in the controller. Signal 21 is when the controller sends the computed control action to the actuator and Signal 34 is the reception of the control action (base flow) from the controller in the actuator.

All the PCs were synchronized using NTP (Deeths and Brunette, 2001) (Network Transfer Protocol).

Results show how the Switched approach behaves quite well in any case although the concurrent access affects the performance of the loop.

3.4 PCT Test 2: Legacy systems integration

In this experiment the possibility of integration of commercial control systems (in this case the Honeywell TDC 3000 DCS system) under CORBA has been explored. With the available hardware, to integrate the TDC in the CCS, the system has been wrapped (with a PC) via the serial interface or via the GUS. The serial interface has the advantage of directly accessing the controller (HPM) like sensors or actuators do. Other routes to access the controller (PLNM, APP) are mostly equivalent to the GUS, and always have to go through the NIM, introducing latency in the communication.

In the HPM is where the control algorithms are executed and where the input/output cables of the field instruments are connected (I/O subsystem). So, it is clear that the serial interface is the easiest way for integrating the TPS in a CCS if it is intended to communicate at control level. (If the interest is only recording historic data, other vias are more convenient).

3.5 PCT Test 3: Interaction of simulation of objects with control agents

In this experiment the simulator ABACUSS II (Barton, 1994) has been used to make a model of the system (process+control) and to integrate it under CORBA to make two tests:

Operators training. In this case the process and the control system are simulated. The HMI sends and receives data from the simulator. The time of execution of the simulation is slowed down to resemble the actual time of the process.

Hardware in the loop. In this test the system simulated is the process plant with the sensors and actuators. The pH controller is left out of the model. The real pH controller implemented in a separate CORBA node is used to control the simulated plant.
4. Conclusions

In these experiments the actuator and the sensor have been wrapped with the CORBA layer through the use of a PC. In the actual process industry CORBA should go embedded in the instrument itself, taking into advantage that the current trend is towards digital, “intelligent” devices. This means that the footprint should be quite small as the memory of this devices is low. The overhead imposed is not significant for the loop timing properties, it can cope with concurrent requests and it works well with multiple objects (around two hundred objects and 6000 thousand signals were alive in the intensive traffic experiment).

The possibility and characteristics of the integration of legacy systems in CCS are fundamentally determined by the facilities provided by vendors of that system, not CORBA. In conclusion, the integration of the TPS in a CCS system is possible but constrained in capacity and scan period. Additionally, there is uncertainty in the temporal behaviour. This allows some degree of integration in typical process plants but is not the ideal case.

The integration of a simulator and the HMI and the interaction with the actual regulator has been easy using CORBA. This has been due to the availability of ABACUSS II as a library. The CORBA object has wrapped the simulator interface and linked to the library to obtain the final CORBA simulation object. The use with commercial simulators is not so straightforward.

It is clear that a priorities policy is needed for process (and any) control systems. But for large and complex control systems where predictability (or at least a Worst Case) is a must it is advisable to use deadlines instead of priorities (you have to know when –in the worst case- is going to happen). This is something that has to be implemented in CORBA. The use of CORBA with an standard wide used network as Ethernet is appealing for the process control domain as the control layers can flatten, costs can be reduced and information be available to any node in the system. The switched Ethernet approach can be suitable for process control but as the process control layer has to be predictable a limit has to be set, and at least a worst case scenario is needed. Concluding, despite the missing features, CORBA with Real-Time Ethernet seems a good option for process control systems and definitely is more than an alternative to OPC (Ole for Process Control).

References


