THE USE OF CORBA IN PROCESS CONTROL [2]

Santos Galán, Manuel Rodríguez, Ricardo Sanz, [1]
Carlos García, Rafael Chinchilla, Adolfo Yela

ASLAB, Universidad Politécnica de Madrid, Spain

Abstract:
Present-day control systems in a process plant are composed by a heterogeneous collection of hardware and software entities scattered over a set of heterogeneous platforms. This HW/SW heterogeneity is a source of extreme complexity in the control system regarded as a whole. To implement the new complex control systems Distributed Object Computing (DOC) seems the adequate technology. In the global DOC landscape, CORBA is a well known framework for the construction of modularised, object oriented, distributed applications. The use of CORBA-based control systems (CCS) has been investigated recently with promising results demonstrating the use of CORBA for Process Control Systems.

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

1. INTRODUCTION

Control systems have been traditionally separated into several levels: *Field level*, dedicated to the instruments (sensors and actuators) and basic regulatory control, communicated via fieldbus. *Process control level*, taking over the advanced and supervisory control, including local optimization, communicated via an Ethernet based protocol. And *Business level*, dedicated to global optimization, scheduling and planning, communicated via Ethernet too.

Although these levels have been always present in the process industry the control implementation has been evolving along the years. Now we talk of plant-wide integration reaching even the lowest levels in production plants: sensors, actuators and basic controllers. This complete vertical integration means that paths are available from sensors to management information systems and back, eliminating some of the barriers that the underlying information technology poses to the design space for monitoring and control systems.

Present-day control systems in a process plant are composed by a heterogeneous collection of hardware and software entities scattered over a set of heterogeneous platforms (operator stations, remote units, process computers, programmable controllers, intelligent devices) and communication systems (analog cabling, serial lines, fieldbuses, LANs or even satellite communications). This HW/SW heterogeneity is a source of extreme complexity in the control system regarded as a whole. Complexity is due to the increased functionality these systems are required to support as well as to the fact that the components are now supposed to be part of a more general system and well integrated in it.

Besides, they are distributed and hierarchical systems, currently PLC (Programmable Logic System) or DCS (Distributed Control System) but moving to a totally distributed system with loops closed in the field level. Distributed systems are

[1] corresponding author, `sgalan@diquima.upm.es`
[2] The project is funded by the European Commission as IST-37652, "HRTC, Hard Real-Time CORBA".

designed to improve performance and increase system reliability in order to meet timing, resources and concurrency constraints on each node. Although the control loop still is a central element in these systems, they contain support for large amount of additional functionalities, e.g., discrete logic control, operator interface, supervisory control and monitoring applications, database access, production planning and scheduling, etc.

These systems are typically also programmable, as defined by IEC 61131-3. The basic technology used today to implement control systems is software technology, from SCADAs and DCSs to intelligent controllers, based on soft computing. The process of incorporation of information technology (IT) into industrial processes has made profound modifications to production systems, but the incorporation of the new technology at the controllers level is having some difficulties. In most cases the difficulties are mainly due to classical barriers posed to innovation in production systems: lack of predictability, need for non-stop operation, lack of reliability and availability, less than ideal market maturity, exploitation managers resilience, etc.

To implement the new complex control systems distributed object computing seems the adequate technology as it, besides other advantages, enhances systems integrability and simplifies the construction of complex information applications. Because modular development is the only known practical way for complex systems engineering.

In the global distributed object computing landscape, CORBA is a well known framework for the construction of modularised, object oriented, distributed applications. CORBA (Common Object Request Broker Architecture, (OMG, 2000; OMG, 1999; OMG, 1998)) is an open standard which 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 was not perfectly suited for the construction of embedded control applications. This has changed recently because Real-time CORBA has found its place into mainstream CORBA specifications. This makes CORBA a specification that deals with real-time issues from the very core (a real difference from other distributed objects technologies).

Some people in the process industry consider that CORBA is an alternative replacement of OPC, but this is based on a lack of understanding of both technologies. CORBA is not a service but a middleware technology that happens to be better than COM (the software under OPC). Particular services like OPC can be built and delivered atop of it. In fact there is an OMG specification that provides a complete replacement with enhancements of OPC servers (it is called HDAIS). CORBA is much wider in scope than OPC and technologically more sound and powerful than COM. For example, CORBA specifies mechanisms for real-time behavior or fault tolerance that is basic for the construction of control applications.

The use of CORBA-based control systems (CCS) has been investigated recently with promising results demonstrating the use of CORBA for Process Control Systems. In the following sections the results are presented as well as the current research directions originated from them.

2. CORBA-BASED CONTROL SYSTEMS

The project HRTC[3] paved the way to CORBA-based implementation of distributed controllers. The project uses a real-time object request broker, with a prototype implementation of a hard real-time network transport, to build two testbeds, the Robot Control Testbed (RCT) and the Process Control Testbed (PCT), addressing issues of hard real-time composability in heterogeneous applications and tight timing requirements.

The long-term objectives of the work performed in the project were focused in the advancement of the CORBA technology for the implementation of distributed complex control systems. The concrete main objective of HRTC was to increase the suitability and acceptance of OMG CORBA specifications for the implementation of distributed control systems in industry. To do this the project focused on the generation of CORBA knowledge to be used by control engineers and the increase of activities related with control systems in the CORBA community.

2.1 Process Control Testbed results

The experiments are instrumental in eliciting the requirements for hard real-time distributed control systems using CORBA technology. The testbed experiments identify some of the benefits that such distributed object computing can bring to the control field.

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

[3] The project was funded by the European Commission as IST-37652 "HRTC, Hard Real-Time CORBA". Find more information at www.hardrealtimecorba.org.

one controls the temperature. This loop has no special relevance for the process but it is needed for the experiments to be taken.

The PCT (figure 1) has been designed to comply with the following general requirements:

- To be representative of the basic characteristics of a process plant control system.
- To be reconfigurable meaning that it can change easily and adopt different configurations and topologies in order to comply with the representativity requirement.
- And, finally, to allow mechanisms to make some experiments and to measure the results, i.e., to be testable.

Figure 2 shows the software configuration to perform the test on a basic CORBA control loop. 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.

All the computers were synchronized using NTP (Network time protocol). The signals corresponds to:

Signal 41 : The pH sensor sends the requested data.
Signal 25 : Reception of the pH value from the sensor in the controller.
Signal 21 : The controller sends the computed control action to the actuator.
Signal 34 : Reception of the control action (base flow) from the controller in the actuator.

Table 1 shows the statistics regarding this experiment using a hub. It shows how a network with a Hub and no external traffic behaves quite well for process control. The complete loop timing around 10 milliseconds is acceptable. And data dispersion is quite low.

Table 1. Performance statistics (ms) for control loop using a hub

Signal	25 - 41	34-21	41-41
Mean	1.822	3.564	9.979
Asymmetry coef.	0.695	1.125	-2.325
Standard deviation	0.055	0.385	0.317
Average deviation	0.048	0.290	0.124

The statistics of the same test using a switch are in table 2. It shows how this network behaves well for process control. The numbers are quite similar to those obtained in the CCS Hub test because of the low collisions level in the previous test. This test presents a good timing response for the Ethernet for process control even with the CORBA middleware overhead.

Table 2. Performance statistics (ms) for control loop using a switch

Signal	25 - 41	34-21	41-41
Mean	0.946	1.821	10.009
Asymmetry coef.	5.425	1.444	0.093
Standard deviation	0.089	0.033	0.288
Average deviation	0.045	0.026	0.115

The intensive data traffic, with a heavy load in the network were performed using both topologies obtaining the results of table 3. This results are quite poor, specially in terms of the high standard deviation. On the other hand, the switched topology performs quite well under the heavy network traffic as it is shown in table 4.

Table 3. Performance statistics (ms) for heavy network load using a hub

Signal	25 - 41	34-21	41-41
Mean	3.105	5.405	14.327
Asymmetry coef.	3.095	2.920	3.285
Standard deviation	2.788	3.138	13.315
Average deviation	1.684	2.053	7.292

Table 4. Performance statistics (ms) for heavy network load using a switch

Signal	25 - 41	34-21	41-41
Mean	0.923	1.911	9.986
Asymmetry coef.	6.302	2.169	-0.456
Standard deviation	0.116	0.030	0.497
Average deviation	0.046	0.019	0.225

For the concurrency access test virtual (CORBA) objects were generated, all these objects were trying to access to the actual pH sensor node (figure 3). The behaviour of the main control loop under this circumstances has been measured for both topologies obtaining the results of tables 5 and 6.

Table 5. Performance statistics (ms) for concurrent access using a hub

Signal	25 - 41	34-21	41-41
Mean	8.317	8.751	16.206
Asymmetry coef.	-1.794	4.800	0.674
Standard deviation	1.415	11.307	6.746
Average deviation	0.874	5.684	5.942

Table 6. Performance statistics (ms) for concurrent access using a switch

Signal	25 - 41	34-21	41-41
Mean	1.194	1.883	11.760
Asymmetry coef.	1.967	1.318	2.717
Standard deviation	0.165	0.035	5.388
Average deviation	0.121	0.027	3.083

The results for the hub are even worse than those obtained with the intensive traffic test. The results for the switch show that the concurrency access affects the switched ethernet in a significant manner. Although results are quite good it can be noted that the timing is slightly increased and specially that the variability of the loop is quite high.

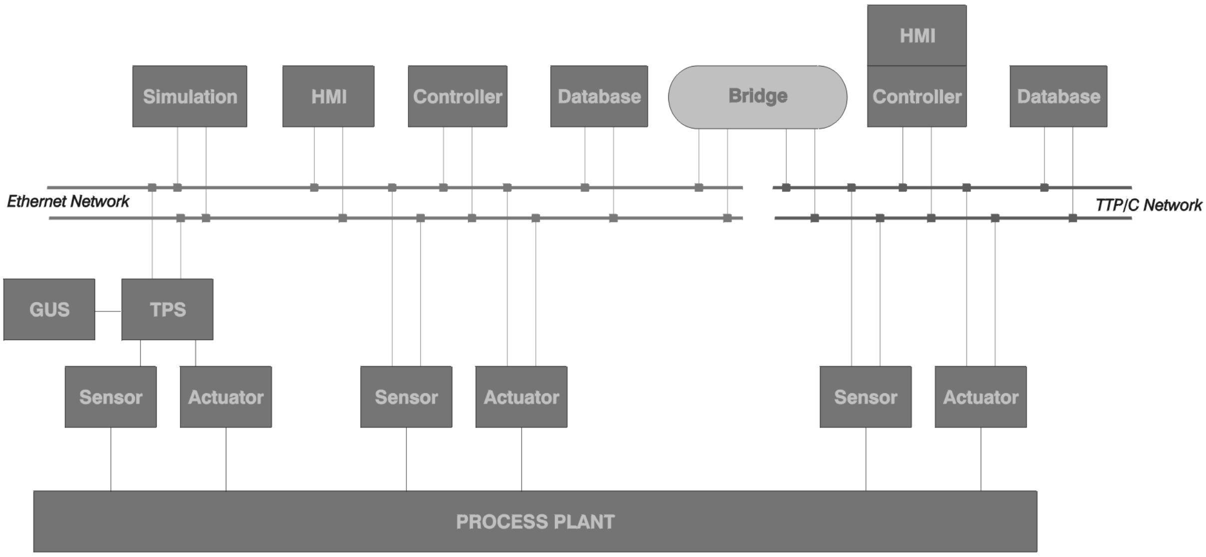

Fig. 1. Scheme of the complete Process Testbed

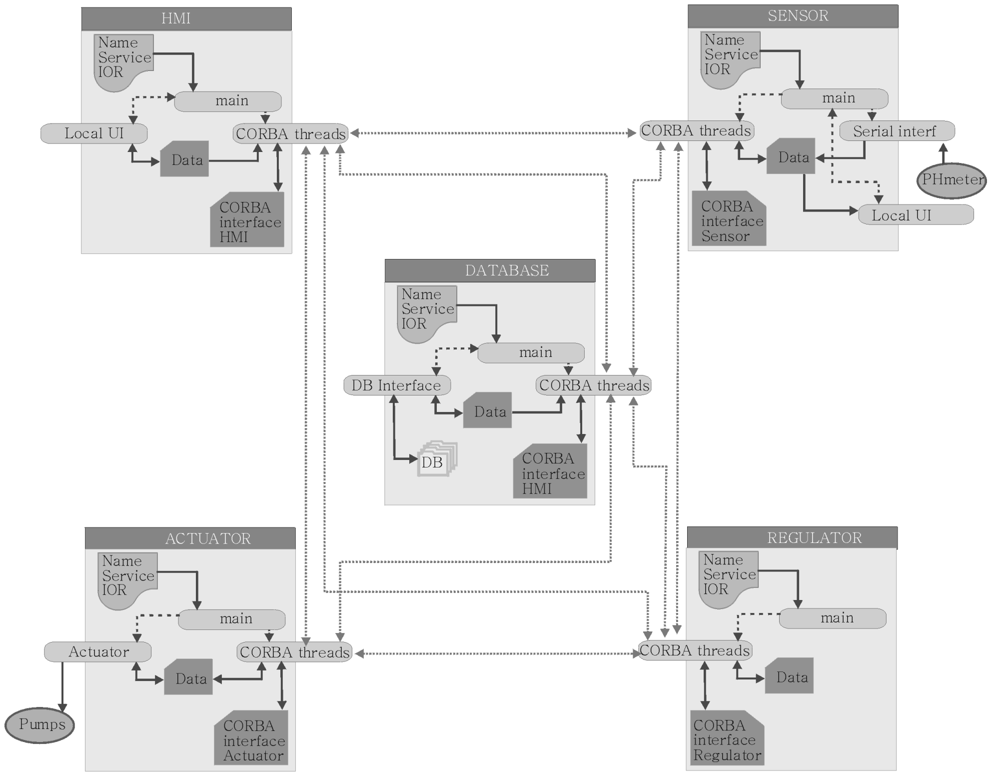

Fig. 2. Software setup implementation of the CORBA pH control loop

2.2 HRTC results analysis

Project activities have reached quite good results in all the main five objectives:

- Know-how in distributed real-time object-oriented control systems: Domain analysis, architecting and engineering information collection have been performed. The resulting documents are not definitive but should be considered as a first step in the production of engineering material to be used by control engineering practitioners. These documents will be made public through fue CORBA Control Systems Website.
- A pluggable real-time ORB protocol prototype: Not one but two pluggable protocols have been developed; one over TTP and another one over switched Ethernet. The TTP protocol demonstrates the possibility of jitter reduction sacrificing flexibility of the application object interfaces. The Ethernet protocol demonstrates that existing off-the-shelf hardware technologies can meet the needs of CORBA control systems when properly managed by software.

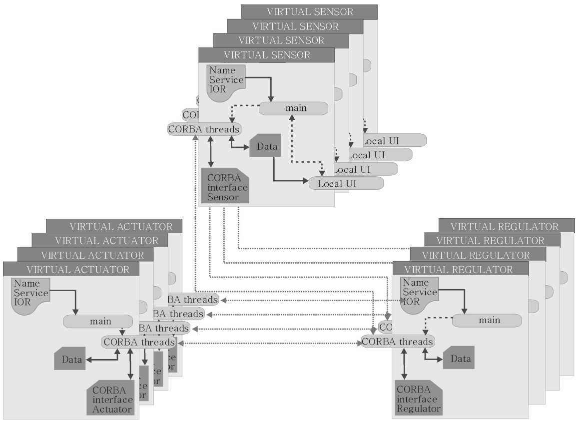

Fig. 3. Software setup implementation of the CORBA heavy network load test

- A robot control testbed: The RCT implements a CORBA control system of a robot. This is a two level controller with visual servoing based on the Ethernet transport.
- A process control testbed: The PCT demonstrates the possibility of using CORBA across the whole plant for process control systems. This system demonstrates true networked control, sensor and actuator wrapping, legacy DCS integration, simulation integration and performance under intensive traffic conditions.
- A specification process for CORBA-based control systems: The OMG has chartered a working group in control systems and the specification process for CORBA technologies in control applications has started with the preparation of a white paper and an RFI.

While not all the expectations have been fulfilled, the final result of the project mostly meets the initial objectives. The results are considered very valuable and they serve the original purpose of enhancing applicability and perceived value of CORBA technology for industrial control systems.

3. CONCLUSIONS AND FUTURE DIRECTIONS

Perhaps the main question is: Why do we need the integration provided by HRTC in process control systems? Beyond many obvious answers (simplicity of flat network, use of heterogeneous components like optimization or simulation, vendor independence, reduction in cost, etc.) we would like to stress one: The modular approach fostered by CORBA will let us develop true modular control systems.

The second point we want to mention is design freedom. Design freedom is necessary in the complex control systems domain to explore alternative controller designs. Excessively restrictive technologies will collapse - unnecessarily - dimensions of the controller design space (Shaw and Garlan, 1996). This is, for example, the case of some fieldbus technologies that support several slaves but only one master. While design restrictions (in the form of prerequisite design decisions) simplify development, they sacrifice flexibility. Can we get both, simple development and flexibility? The key are frameworks where design dimensions are still open even when pre-built designs are available. To continue the example of the fieldbus, the one-master/several-slaves approach is one type of pre-built, directly usable, design; but the underlying fieldbus mechanism should allow for alternative, multi-master designs. This can be done by means of the development of agent libraries that provide predefined partial designs in the form of design patterns (Sanz *et al.*, 1999), and a transparent object-oriented real-time middleware like the one proposed in HRTC. This approach will let developers construct their own agencies to support their own designs.

Another line, that is being explored in the IST COMPARE[4] and ITEA MERCED[5] projects, is the Component/Container Model. Besides its current realizations, the Component/Container Model provides two fundamental features:

- A component not only explicitly describes the services it provides (as an object does) but also the ones it requests to be provided by other components: this allows an easier deployment, achievable by an external tool apart from the application itself.
- A component is meant to be hosted by a container that takes into account the management of the technical properties provided by the underlying infrastructure, on the behalf on its components: this allows a separation of concerns between the business logic (to be hosted by the component) and the technical logic (to be hosted by the container). Example of technical logic are for instance how persistence is managed (in a plain file, in a relational database...) or support for access control to the business logic.

These features are key concepts to master complexity and to allow an effective reuse, hardly achievable with only object orientation.

REFERENCES

OMG (1998). A discussion of the object management architecture. Technical report. Object Management Group. Falls Church, USA.

---

[4] The project was funded by the European Commission as IST-004669 “Component Approach for Real-time and Embedded”.

[5] MERCED, “Market Enabler for Retargetable COTS Components in Embedded Domain”.

OMG (1999). Real-time CORBA adopted specification. Technical Report orbos/99-06-02. Object Management Group. Falls Church, USA.

OMG (2000). Common object request broker architecture and specification. Release 2.4. Technical report. Object Management Group. Falls Church, USA.

Sanz, Ricardo, Miguel J. Segarra, Angel de Antonio, Fernando Matía, Agustín Jiménez and Ramón Galán (1999). Design patterns in intelligent control systems. In: *Proceedings of IFAC 14th World Congress*. Beijing, China.

Shaw, Mary and David Garlan (1996). *Software Architecture. An Emerging Discipline*. Prentice-Hall. Upper Saddle River, NJ.