Provisioning QoS in Real-Time Distributed Object Architectures for Power Plant Control Applications

L. Bacon[++], E. Becquet[*], E. Gressier-Soudan[*], C. Lizzi[*], C. Logé[$], L. Réveilleau[*]

[*]CEDRIC-CNAM, 292 rue St Martin, 75 141 Paris Cedex 03 France,
{becquet, gressier, lizzi, lreve}@cnam.fr
Tel : 33 1 40 27 22 96, Fax : 33 1 40 27 22 96

[++] EDF, Division Recherche et Développement, Ensemble de Production, Contrôle Commande en Centrales, Groupe Architecture,

6 Quai Watier, 78 401 Chatou Cedex, France
laurent.bacon@edf.fr
Tel : 33 1 30 87 80 43, Fax : 33 1 30 87 82 61

[$] Laboratoire LaRIA ,
5 Rue du Moulin Neuf 80000 Amiens, France
christophe.loge@laria.u-picardie.fr
Tel : 33 3 22 53 40 87, Fax : 33 3 22 45 46 47

Contact author :
gressier@cnam.fr
CEDRIC-CNAM, 292 rue St Martin, 75 141 Paris Cedex 03 France
Tel : 33 1 40 27 22 96

Submission Class : Practice

Key words : QoS, CoS, object oriented middleware, real-time, Internet, Java, CORBA, DCOM, process control application

Provisioning QoS in Real-Time Distributed Object Architectures for Power Plant Control Applications

L. Bacon[++], E. Becquet[*], E. Gressier-Soudan[*], C. Lizzi[*], C. Logé[$], L. Réveilleau[*]
laurent.bacon@edf.fr, becquet@cnam.fr, lizzi@cnam.fr, gressier@cnam.fr,
christophe.loge@laria.u-picardie.fr, lreve@cnam.fr
[++]EDF, DRD/EP/CCC/GA, Chatou, France
[*]CEDRIC-CNAM, Paris, France
[$] Laboratoire LaRIA, Amiens, France

Abstract :

The work in this paper is a case study. This paper describes how we are provisioning QoS in a real-time distributed object platform able to support power plant control applications. A previous experiment with a modified Java-ChorusOS-ATM platform based on RM-ODP/ReTINA binding objects enhanced our knowledge on QoS based distributed object systems. RM-ODP/ReTINA brings a distributed object computing model where, time requirements can be specified in terms of QoS parameters. Our goal is to achieve QoS provisioning with off-the-shelf CoS based products in the Real-Time Distributed Object Architectures to be built. Low level building blocks of the architecture as communication networks and operating systems exist as CoS components. The current step of the project deals with distributed object platform selection. But it is not so easy to choose a QoS-enabled middleware.

Key words : QoS, CoS, object oriented middleware, real-time, Internet, Java, CORBA, DCOM, process control application

1 Introduction

The aim of the project described in this paper, is to study, to propose and to evaluate new system architectures for EDF's power control applications. EDF is the french power utility. It is an ongoing project that started smoothly at the end of 1998. Its current state is to select off-the-shelf products to build a Quality of Service (QoS) aware distributed object platform. Expected results, for the end of 2001, should give design and platform hints in the context of EDF’s power control applications renewal.

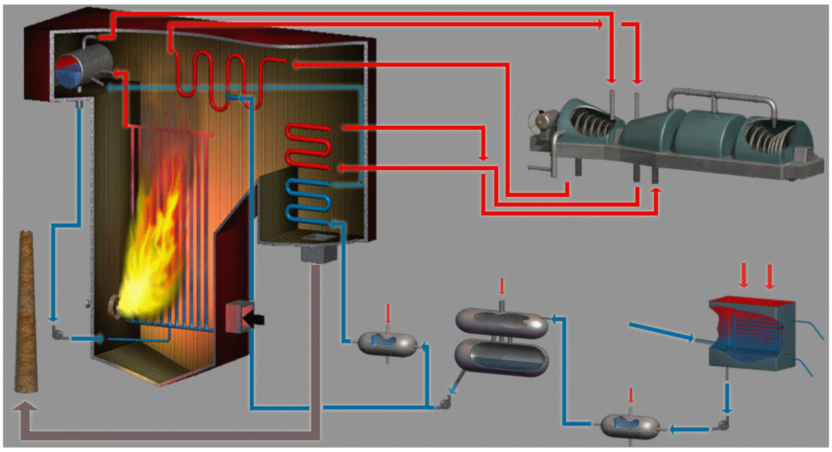
Existing solutions are built as a hierarchy of heterogeneous networks that link heterogeneous hosts running heterogeneous operating systems. This hierarchy is close to the CIM hierarchy [VAL92]: a layered architecture that distributes the power plant control functions over different types of networks. Instrumentation Measurement and Control functions (IMC), real-time functions, are located on bottom networks. Monitoring and management functions are implemented in higher networks.

The project has been divided in four phases: analysis of EDF's requirements on IMC applications [BAC99a], evaluation of off-the-shelf real-time systems [BAC99b], analysis of object oriented middleware capabilities to provide QoS properties, design and implementation of a representative test application. In our context, QoS parameters are related to time constraints, they do not deal with reliability nor security constraints. We are in the third phase of the project. Taking into account the results at the end of the first phase, we decided to base our design on the Open Distributed Processing Reference Model (RM-ODP) and its extension for real-time multimedia called ReTINA [BLA97]. The RM-ODP/ReTINA framework allows the design of object oriented distributed systems and the specification of real-time constraints between interacting objects in terms of QoS parameters. Encompassing networks, operating systems, middleware and application, the QoS paradigm is relevant to our problem. It offers a uniform way to handle time requirements: delay, deadline, response time, jitter, throughput…. This paper presents our intermediate results. Our work leads to the following conclusion: a trade-off should be made. It seems that no off-the-shelf distributed object platforms are able to deliver true QoS. QoS can be approximated using prioritization (object broker, operating system, protocols, networks). This approach leads to Classification of Service (CoS). It is the reason why we are saying QoS provisioning. The same approximation is encountered in the context of Internet network protocols.

The paper is organized as follow. Section 2 describes application characteristics. Section 3 presents the design of existing EDF's architectures. Section 4 deals with users' requirements. Section 5 presents a preliminary prototyping. Section 6 analyses the design of real-time distributed object architecture (RT-DOA) with off-the-shelf products. Section 8 concludes with our current results.

2 Application characteristics

The main activity of EDF is to produce energy. Thermic and nuclear energy produce electricity through a water/steam cycle. Our study is concerned by thermic energy. Figure 1 represents a typical example of a thermic power plant. Water is heated and transformed to steam. Steam crosses a turbine coupled with an alternator. The alternator, moved by steam, produces electricity. The hot steam is transformed back into water with external fresh water, a new cycle is able to begin.



red = steam, blue = water

Figure 1. Thermic Production Process.

The water/steam cycle is a continuous process. The physical process, depicted in figure 1, can be very complex. It needs different components along the production path: pumps, tanks (water, fuel), valves, electricity elements (transformers, circuit breakers…)… The need for full reliability, the environmental constraints, and the efficiency required lead to a growing complexity with more and more computers involved in the control of the process. It is typically a Supervision and Control Aided Data Acquisition application. There are few commands compared to acquisition. Captors and actuators talk to plant computers that mainly regulate the process and control electricity production. The process state is shown to the crew through a graphical representation based on a friendly user interface on dedicated terminals. Operators get IMC events and state changes through this interface. Other important functions are related to: recording of events, data acquisitions and commands, secure access means for production management or remote control, dedicated printing and archiving… There is one computer per main function. Redundant computers and networks insure fault tolerance.

Power plant objects implement process components (pumps, valves…). They summarize process component behaviour. Variables or measurements are detailed information related to component internals. The two representations are used simultaneously. This aspect is common in the context of industrial applications. It is standardized and generalized in fieldbus systems. States and attributes of power plant objects are computed using variables on a per period basis or on a state change basis.

Information exchanges are mostly related to data acquisition. Data acquisition is an up-stream, from physical process toward control computers and operators (summarized way). Commands go down-streams. Data from the process can be analog, numeric or on-off signals. Data acquisition can be periodic states or event based (state changes). Data streams are characterized by two modes: steady state, and burst. Some exchanges are made in client/server mode (state or variable consultation). Except for the reliability requirement, data exchange patterns are similar to distributed multimedia application streams. Real-time constraints are softer. The application end-to-end delays (from acquisition to operator, or, from command to actuators) are about 500ms, small jitter is allowed, but no information can be lost.

In a first analysis, the object representation associated to power plant application components and data exchange patterns justify the choice of the RM-ODP/ReTINA approach to design power plant control applications. Also, this meta-model offer a powerful and general framework to specify complex distributed applications.

3 Existing system architectures of power plant control applications

We have seen previously that power plant control applications are distributed by nature. Control devices are distributed along the process and control functions are mapped on different computers with replication schemes for fault tolerance purpose. The classical design of such applications is layered by the Computer Integrated Manufacturing model [VAL92]:

- Level 0 deals with captors and actuators.
- Level 1 supports acquisition and control computers, they summarize information from lower levels and support regulation loops.
- Level 2 performs monitoring functions.
- Level 3 deals with plant production control.

Our design focuses on level 1 and 2. Very recently, we had to take into account functions generally supported at level 0 and 4 (distribution management functions).

Current solutions are built as a hierarchy of networks. They are dedicated cost effective architectures, based on message oriented interactions, built on top of OSI or proprietary protocol stacks. Generally, networks are heterogeneous : fieldbuses or point to point links at the lower level, LANs elsewhere. Encountered operating systems are: real-time executives at level 1, general purpose reliable kernels at higher levels. Heterogeneity is a key feature of the environment that we should address. Different companies are involved to supply an architecture: interoperability and integration problems are common.

Current solutions are cost effective, and, generally monolithic. The component unit of engineering is the host: an host can be added or removed with all its functions. If a function is mandatory, fault tolerance mechanisms avoid any deny of service. Today solutions are not enough modular, nor flexible. They are not built on a distributed system basis. Object orientation is not considered in existing solutions. They have been designed 15 years ago or more. Any solution has to run at least thirty years.

Figure 2 depicts representative hardware and software architectures. Figure 3 shows data streams between level 1 and level 2 calculators. In the two figures level 1 and level 2 components share the same network, an expected result of our study.

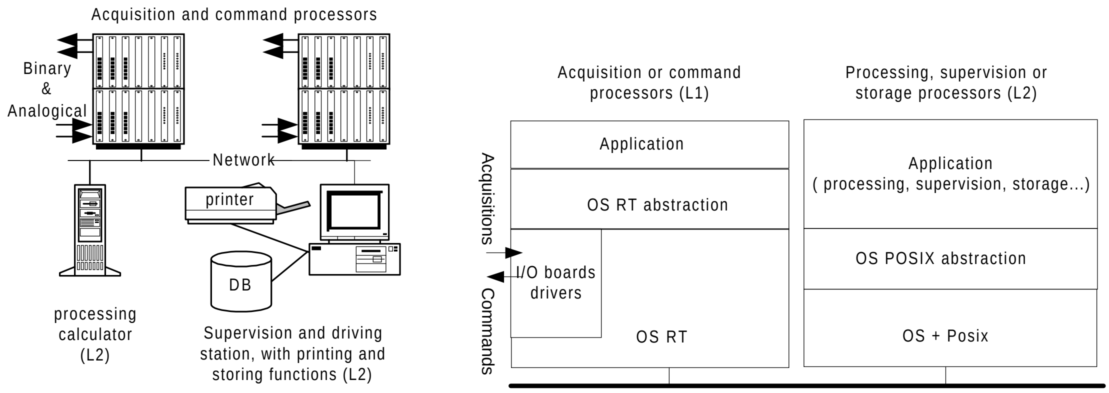


Figure 2. Hardware and software common architectures

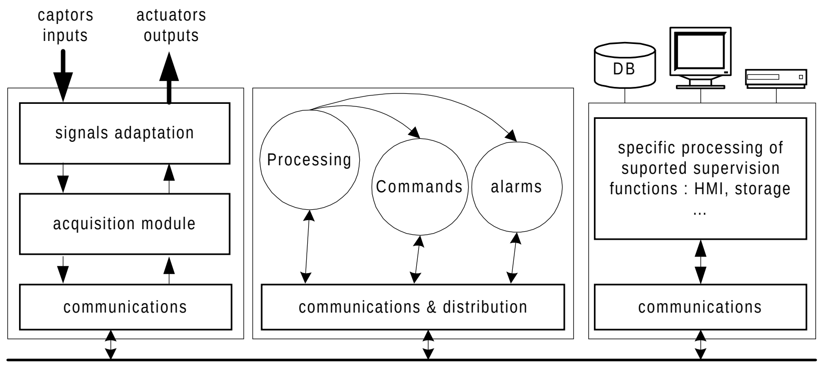


Figure 3. Control Command data streams

4 User's Requirements

User's requirements about EDF power plant control applications emerge from two sources : inquiries [BAC99a] and strategic considerations.

When the project started, inquiries have been made to characterize power plant control applications, to know architecture design properties, to list engineering requirements, and to define new functionalities to be supported [BAC99a]. The main results are summarized hereafter:

- Standards should be used everywhere: hardware components, networks, operating systems, application programming interfaces (API).
- Ethernet is the basic LAN technology. Few fieldbus systems are used. Level 0 communications are specific.
- OSI and Internet protocols are preferred. There are few proprietary networks, mostly dedicated to field communications. Internet becomes the mandatory protocol family.

Protocols over the Transport layer can be OSI with RFC1006 [ROS87] or RFC 2126 [POU97] (obsoletes RFC1006). The choice depends on available products. In fact protocols as ISO-MMS [VAL92] are still required for backward compatibility.

- Operating systems are heterogeneous, real-time kernels address low level computing, general purpose operating systems address higher level functions. Real-time kernels are expected to meet the real-time constraint requirements from the application. Real-time POSIX extensions should be supported. Easy to use engineering tools for system development are mandatory. These aspects has been evaluated for LynxOS [BAC99c], pSOS+ [BAC99d], VxWorks [BAC99e], and during the second phase of the project.
- Products must be supported by well-known long-life suppliers. They should be use widely in industry. A power plant control application lives 30 years at least.
- New functions deal with multimedia. For maintenance purpose and fault detection, audio (golden ears) and video (fault localization, process recognition) streams are needed. IMC and multimedia data should share the same communication network.

Strategic considerations deal with power plant control application renewal. System architectures have to evolve. They have to take care of emerging technologies: high speed Ethernet switches (100Mb/s or 1Gb/s), Internet networks, CoS/QoS aware/enabled protocols, object oriented software buses, web facilities, real-time modular micro-kernels, intelligent captors and actuators, more flexible easy to use human machine interfaces… Such technical requirements are brand new in the context of power plant control. Also, a very important goal is to lower the cost of applications.

5 Preliminary Prototyping : An ATM based Real-Time Distributed Object Platform

Prior to the project with EDF, we were involved in the experiment of a real-time distributed platform to address process control and manufacturing applications. ANTARA, given in figure 4, provided initial design constraints: PowerPC processors, real-time micro-kernel ChorusOS, and a 155Mb/s ATM network. ATM has been chosen for its native capability to handle QoS constrained streams. Micro-kernels are supposed to drive actuators and captors, or programmable logic controllers, figure 4 gives an overview. The prototype, developed, by C. Lizzi, a PhD student became a QoS based distributed object system [LIZ99c]. It uses a modified ChorusOS micro-kernel [LIZ99c] with a real-time inter-process communication facility [LIZ98b] based on an ATM network [LIZ98a] and offers a specific real-time java virtual machine to run object oriented application entities [LIZ99b]. More details are given in [LIZ00].

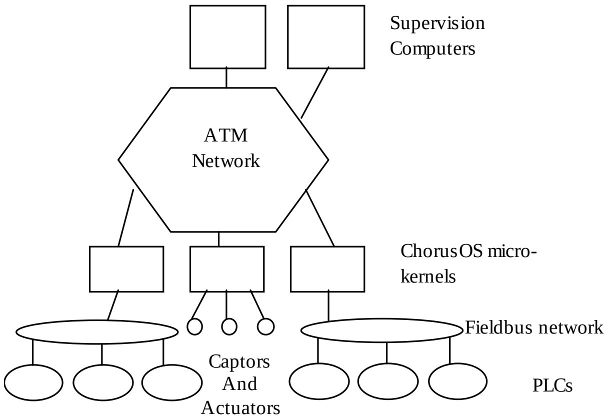


Figure 4. ANTARA platform for Process Control [CST96]

To meet hard real-time constraint requirements, the ChorusOS micro-kernel has been enhanced in the following ways:

- The native scheduler has been modified to enable deadline and criticity scheduling.
- A zero-copy mechanism allows an efficient buffer management for I/O operations, this feature is mandatory for network communications and distributed inter-process communication (D-IPC).
- QoS based D-IPC extends native IPC, its QoS characteristics are directly mapped on the QoS capabilities of ATM networks. The way QoS is implemented conforms the binding object model issued from the computational and engineering viewpoints of RM-ODP.
- A clock synchronization protocol is supported.
- Local and distributed deadline inheritance are implemented. Distributed deadline inheritance means end-to-end inheritance across the network.
- On top of previous services, a java virtual (JVM) has been implemented to support distributed objects, it is based on the Japhar open source package [REI98].

From a real-time distributed object platform design point of view, the features added to the JVM are an interesting contribution. This JVM can benefit from the micro-kernel enhancements and can be said real-time. The new kernel features are available through the way of new classes bundled in a specific Java package. Java threads are mapped on kernel threads. New classes allow periodic and aperiodic tasks scheduled according to a deadline based scheduling policy (Earliest Deadline First). Other extensions have been integrated to the JVM. It is able to use the previous real-time communication system through the Java Native Interface (JNI). The deadline concept is extended to Java threads communications, and uses the end-to-end deadline inheritance over ATM networks. The JNI interface has been optimized to allow direct sharing of memory between Java and native code. This feature grants Java to access efficiently memory areas mapped by data acquisition and control devices.

The current implementation of this RT-JVM seems to be enough efficient for power plant control applications. Java threads are able to exchange data at a throughput of 97Mb/s, with 0.3% cell loss (ATM layer) and 15 ms delay. Similar experiments in the context of MMS like services on top of a Java ORB environment directly over an ATM network with Linux and a regular JVM give a throughput of 82Mb/s, with a 14.1-45% cell loss, and 90 ms delay [SEI99b].

From an industrial point of view, today, it is not realistic to build a process control application with the same system platform: heterogeneity is a key issue. The ATM technology is attractive for its native QoS capabilities, but it is still too expensive and too specific.

6 Real-Time Distributed Object Architecture with Off-The-Shelf Products

6.1 Principles

The previous work helped us in many ways. It asserted the relevance of the RM-ODP/ReTINA framework to specify and implement QoS features. This experience gave us means to analyse platform design of real-time distributed object systems, and, enhanced our knowledge on their properties.

The QoS aware Java-ChorusOS-ATM platform is efficient but has been built from scratch. This approach is no more suitable with off-the-shelf products. We need some guidelines to go further. Consequently, we are also using the RM-ODP/ReTINA engineering viewpoint to help the design of an efficient RT-DOA. Figure 5, issued from [BLA97], recalls the mapping of binding objects between the computational and the engineering viewpoints.

One can observe that the implementation of current off-the-shelf object oriented middleware like: MS-DCOM, OMG-CORBA based platforms, SUN-JAVA/RMI, corresponds in many points to the design illustrated in figure 5. All of them include transportation, naming/localization, and marshalling/unmarshalling mechanisms [RAJ00],[CHU00].

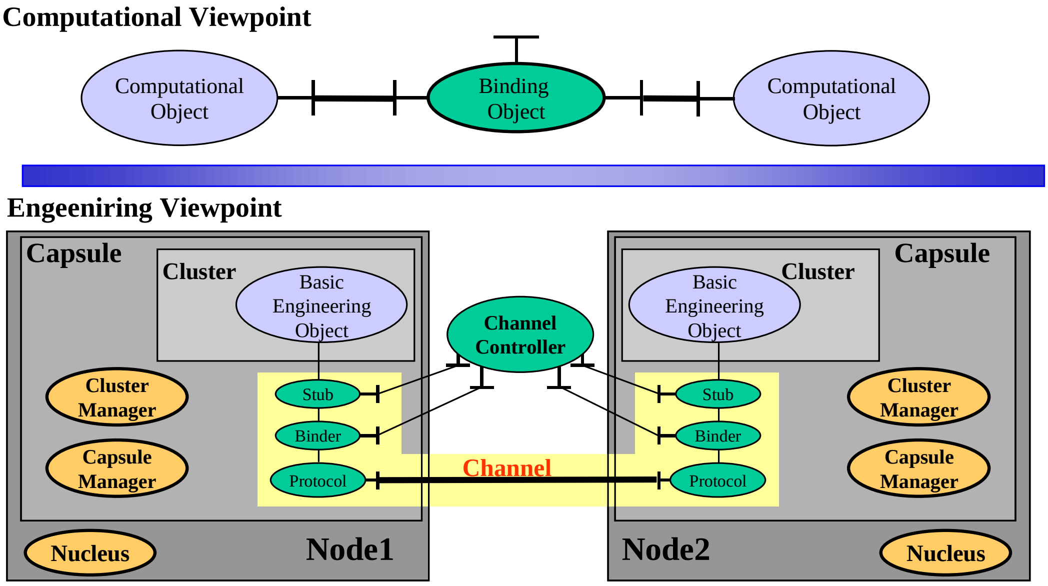


Figure 5. Mapping of binding objects between the computational and the engineering viewpoints from RM-ODP/ReTINA

From an engineering point of view, problems come with the way QoS is implemented in the execution platform: networks, operating systems and middlewares. Strictly speaking, in the real-time context, QoS is related to time parameters, for examples deadlines in operating systems and latency in communication networks. Usually "CoS" is dedicated to DiffServ[IET00b] Internet networks, but this concept can be applied to priority based operating systems, distributed systems and applications. The same terminology is used for real-time CORBA[OM99a], it appears in the errata where it is said "RTScheduling changed to RTCosScheduling" [OMG99b]. Off-the-shelf products are CoS based as far as they rely on priorities. To fill the gap between the QoS paradigm and CoS based available products, we need some tools that are able to map QoS constraints to CoS parameters like [NAH95] [NAH96].

6.2 Network Building Block

Network, including communication protocols, and operating systems are the basis of the RT-DOA. From a QoS management point of view, the first experience with ATM was quite easy. ATM is built to deliver native QoS properties. Equivalent properties are not easy to find in the technology required by EDF: Ethernet, and Internet protocols.

From a network point of view, QoS should be supported from an end to the other end, across all the intermediate nodes along the data path. The power plant network (level 0 through 3) is built as a sum of LANs but is only one intranet, this means that no routing (layer 3 switching) is needed today within the plant. Level 0 through 3 networks eliminate QoS/Cos routing considerations. As a consequence, we are not focusing on MPLS (Multi-Protocol Label Switching) or QoS-routing proposals from IETF. As we are concerned by high speed Ethernet, layer 2 switching is a key level to support network QoS/CoS. The IEEE 802.1p [IEE97] devoted to priority management and associated to the IEEE 802.1Q [IEE98] standard (VLAN), brings some help and defines our first building block. But 802.1p compliant hardware should be available and handled by operating systems and upper layers protocols and.

Now, we are able to examine QoS provisioning related to internet protocols at end nodes. IntServ [IET00a] and DiffServ [IET00b] frameworks can be analysed. IntServ offers a reservation framework based on the resource ReSerVation Protocol (RSVP) [IET00a]. RSVP has a drawback: it loads the network with periodic messages, in case of burst, this property could not be enough reactive to be used properly, and it can load the network. Therefore, we decided to eliminate the IntServ approach. DiffServ [IET00b] is not based on resource reservation but on prioritization. Classification of message streams has to be made. It can be done in our context. The Diffserv approach is well-suited to our application domain. The priority of IP datagrams can be used (3 bits in the TOS field) and has to be mapped into the IEEE 802.1p priority of Ethernet frames. The TOS field has been re-designed [IET00b], and called DSCP (DiffServ CodePoint). In the DSCP field, 6 bits are dedicated to priority (64 levels) against 3 bits for TOS (8 levels).

To summarize this section, one can observe that the best way to enable QoS in power plant networks is to adopt a CoS approach for LAN and Internet protocols network building blocks. As far as we know, it seems that the off-the-shelf products that we need are ready on the market place for switches but not for end system hardwares.

6.3 Operating Systems Building Block

Close to the physical process, real-time micro-kernels are generally deployed. We have already evaluated real-time operating systems during 1999 [BAC99b]. Selected products were LynxOS (Lynx Real-Time Systems), pSOS+ (Wind River, formerly Integrated Systems Inc.) and VxWorks (Wind River), because of the users requirements [BAC99a] (real-time aspects, perennity, widely-used in industry, compliance to POSIX standards...). Our conclusions are that all of them are sufficient about performances. API functions were equivalent. The way operating system abstractions are implemented gives some advantage to pSOS+ and VxWorks: they offer a zero-copy version of buffer management for TCP/IP communications, they support priority inheritance for thread synchronization. Also, development environment, installation and configuration flexibility are key features for the choice of an operating system. These aspects are generally never evaluated. Wind River, with Tornado, offers the easiest development environment tool.

General purpose operating systems are used in upper layers of the plant control application. No particular requirements are made except that they satisfy time constraints. Means to classify treatments are needed. It means that they offer access to real-time priorities. One can notice the important use of Microsoft products in IMC applications, then Windows 2000 has to be evaluated.

In both cases, micro-kernels and general purpose operating systems, a special attention is devoted to QoS management API. QoS is supportede in Windows 2000 through the Winsock2 API [ISE99].

To summarize this section, one can observe that the best way to enable QoS in power plant system platforms is to adopt, here too, a CoS approach. As far as we know, it seems that needed off-the-shelf products are emerging on the market place.

6.4 Emerging features of Object Oriented middlewares

Three object oriented middlewares are candidate technologies: Sun-Java/RMI, OMG-CORBA, Microsoft-DCOM. All of them are concerned with real-time properties when a real-time micro-kernel is used, and, as far as we know, it seems that corresponding off-the-shelf products are not ready on the market place. Hereafter, we summarize briefly the key real-time features of each one:

-
- Microsoft-DCOM: DCOM is a proprietary object oriented middleware concurrent to OMG-CORBA. It is an extension of Component Object Model, widely used by Windows based sofwares. DCOM is a multi-language platform. It relies on DCE Remote Procedure

Call over TCP/IP. As far as we know, there is no specification dealing with real-time constraints. Few off-the-shelf implementations of DCOM run over a real-time kernel. We know, VxDCOM from Wind River [WIN00]. It is optimized to use efficiently operating system services. There is not a global approach to plug together real-time properties of the software bus, operating systems, and network services. However this solution could be used in conjunction with Network Data Delivery Service for Tornado, another product from WindRiver that provides QoS control. Anyway, Microsoft world cannot be circumvented as far as it is widely deployed in our context (at least at the upper levels of power plant). Also, no documentation describes the integration of Windows2000 QoS capabilities [ISE99] to DCOM. This feature has to be fully evaluated.

- OMG-CORBA: CORBA (currently version 2.3) is a multi-language-multi-platform standard. It has not been designed to deal with real-time environments. CORBA Messaging [OMG98a], MinimumCORBA [OMG98b], and Real-Time CORBA [OMG99a] are new specifications to address this lack. They will be part of the next revision of the standard (3.0). The main features of real-time extensions are the definition of a schedulable entity. Real-time CORBA defines a scheduler (to be mapped on the underlying operating system scheduler), and introduces CORBA priorities, which do not depend on platform or operating system. These priorities can be sent over a network. They can be used to schedule object requests at the server side (on a priority basis rather than a FIFO basis which is the normal way). Moreover, the well-known priority inversion problem is solved via priority inheritance. Real-Time CORBA uses also the concept of QoS and let developpers describe their bindings characteristics (for example if requests must be served on a priority or on a temporal basis). TAO [TAO00] is the most popular ORB that conforms the real-time CORBA profile. We expect to evaluate TAO in the context of power plant control applications.
- Sun-Java/RMI: The Java runtime is platform independent. It is very interesting to alleviate the problem of heterogeneity. There have been different initiatives to promote the use of Java for real-time [NIS99], [JCO00], [RTJ00]. PERC [NIL96] is an advanced product, from NewMonics Inc. PERC provides a JVM that offers rate monotonic scheduling, priority inheritance, real-time garbage collection and support for runtime execution analysis. Interactions between the real-time Java program and its environment

uses proprietary Java classes that enable access to I/O ports and interrupt handling. None of the RT-JAVA projects deals with Remote Method Invocation. This is an important lack that can be handled by specific development, approach that we want to avoid now! An alternate solution implies the use of java sockets, a low level abstraction. By the way, access to QoS facilities could use a native QoS API. Also, companion tools associated to Java, like JINI (for configuration/re-configuration management), JavaSpace (for distributed data management), Enterprise Java Beans (for component oriented engineering) could be helpful. One can notice emerging technologies related to Java processors. TINI from iButton [IBU00] or aJPC104 from aJile [AJI00], for example, could be used as intelligent captors and actuators.

The three object oriented platforms offer their own advantages. Considering QoS provisioning, they all rely on CoS principles.

In conclusion of this section, choosing a middleware able to provision QoS is a hard task. Available technologies are promising, but stable off-the-shelf versions have to be fully evaluated. This implies a carefull examination of the relationship between kernels, network protocols and middleware functions to address EDF's power plant application needs.

7 Conclusion

The work described in this paper is a case study. Our project to build a real-time distributed object platform able to support power plant control applications is at the middle step of its roadmap. The RM-ODP/ReTINA framework is usefull in our context. It brings a computing model expressed in terms of objects. Time requirements of interactions between objects can be specified in terms of QoS parameters. Also, we use RM-ODP's engineering viewpoint to help our design of a QoS aware architecture. Our goal is to achieve QoS provisioning with CoS based off-the-shelf products in the Real-Time Distributed Object Architectures to be built. Low level building blocks as high speed Ethernet switches, DiffServ conformant internet protocol implementations, real-time micro-kernels are available or quite as CoS components on the market place. The current step of the project deals with distributed object platform selection and evaluation. It is not so easy to choose a middleware ready to provision QoS. Three main object oriented middleware families exist: Sun-Java/RMI, OMG-CORBA, Microsoft-DCOM. Their real-time versions are emerging technologies that have to be fully experimented. It is the goal of the next phase of the project.

References

[AJI00] aJile. "aJ-PC104 Datasheet". http://www.ajile.com/aJ-pc104_datasheet.htm. April 29th 2000.

[BAC99a] L. Bacon, E. Gressier-Soudan. "SETR : Systèmes d'Exploitation Temps Réel - Etude des contraintes utilisateurs – Synthèse entretiens". HP-32/98/079/A. 1999.

[BAC99b] L. Bacon, E. Becquet, E. Gressier-Soudan. "Etude Comparative de micronoyaux du commerce". 7ème Séminaire du Laboratoire d'Ingénierie de la Sûreté de Fonctionnement. 16-17 Mars 1999. Domaine de Villepreux. Bordeaux. France.

[BAC99c] L. Bacon, P. Decogné. "SETR : systèmes d'exploitation temps réel – LynxOS – Architecture". HP-32/99/099/A. Juillet 1999.

[BAC99d] L. Bacon, E. Becquet. "SETR : systèmes d'exploitation temps réel – pSOS – Architecture". HP-32/98/098/A. Mars 1999.

[BAC99e] L. Bacon, E. Becquet. "SETR : systèmes d'exploitation temps réel – VxWorks – Architecture". HP-32/98/100/A. Septembre 1999.

[BLA97] G. Blair, J-B. Stefani. "Open Distributed Processing and Multimedia". Addison-Wesley. 1997.

[CST96] CS Technologies Informatiques. "ANTARA System Functional Specification". Document 101-00061, CSTI, April 1996.

[CHU00] P.E. Chung, Y. Huang, S. Yajnik, D. Liang, C.-Y. Wang, Y.-M. Wang. "DCOM and CORBA Side by Side, Step by Step, and Layer by Layer". http://www.cs.wustl.edu/~schmidt/submit/Paper.html. April 30th 2000.

[IBU00] iButton. "Introducing TINI : Tiny InterNet Interface". http://www.ibutton.com/index.html. April 29th 2000.

[IET00a] IETF - The IETF IntServ Working Group. http://www.ietf.org/html.charters/intserv-charter.html. May 1st 2000.

[IET00a] IETF - The IETF DiffServ Working Group. http://www.ietf.org/html.charters/diffserv-charter.html. May 1st 2000.

[IEE97] IEEE. "Traffic Class Expediting and Dynamic Multicast Filtering". 802.1p/D6. April 1997.

[IEE98] IEEE. "Draft Standard P802.1Q/D11, IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks". July 1998.

[ISE99] D. Iseminger; "Windows 2000 Quality of Service". MTP. 1999.

[JCO00] J consortium. "Real-time core extensions specifications for the Java platform". February 2000.

[LIZ98a] C. Lizzi, E. Gressier-Soudan, J. Montiel. "A Real-Time Communication Service for ATM-based Distributed Systems". ICATM'98. IEEE International Conference on ATM. June 98. Colmar France.

[LIZ98b] C. Lizzi, E. Gressier-Soudan. "A Real-Time IPC Service over ATM networks for the Chorus Distributed System". Euromicro'98. 24th Euromicro Conference. Vasteras. Sweden. August 1998.

[LIZ99a] C. Lizzi. "Java Real-Time Distributed Processing over ATM Networks with ChorusOS". ETFA'99. Barcelona, Catalogna. Spain. October 1999.

[LIZ99b] C. Lizzi. "Enabling Deadline Scheduling for Java Real-Time Computing". 20th IEEE Real Time Systems Symposium (RTSS'99). Phoenix, Arizona, USA. December 1999.

[LIZ99c] C. Lizzi. "Design of an ATM based Real-Time Distributed System". PhD Thesis. Conservatoire National des Arts et Métiers. 14 Décembre 1999. Paris. France. (text in french).

[LIZ00] C. Lizzi, L. Bacon, E. Becquet, E. Gressier-Soudan. "Prototyping QoS based Architecture for Power Plant Control Applications". submitted to the IEEE Workshop on Factory Communication Systems (WFCS'2000). October 2000. Portugal.

[NAH95] K. Nahrstedt, J. M. Smith. "The QoS Broker, IEEE Multimedia, Spring 1995, Vol.2, No.1. http://cairo.cs.uiuc.edu/papers/QoSbroker.ps

[NAH96] K. Nahrstedt, J. M. Smith. "Design, Implementation and Experiences with the OMEGA End-point Architercture". IEEE JSAC. Sep. 1996. http://cairo.cs.uiuc.edu/papers/OMEGA.ps

[NIL96] K. Nilsen, "Issues in the Design and Implementation of Real-Time Java", Technical Report, NewMonics Inc., 1996.

[NIS99] National Institute Of Standards and Technology (NIST). "Requirements For Real-time Extensions for the Java platform". September 1999.

[OMG98a] Object Management Group. "CORBA Messaging". Joint Revised Submission. OMG TC Document orbos/98-05-05. May 1998.

[OMG98b] Object Management Group. "minimumCORBA". Joint Revised Submission. OMG TC Document orbos/98-08-04. August 1998.

[OMG99a] Object Management Group. "Real-Time CORBA". Joint Revised Submission. OMG TC Document orbos/99-02-12. March 1999.

[OMG99b] Object Management Group. "Errata for Real-Time CORBA". Joint Revised Submission. OMG TC Document orbos/99-03-29. 1999.

[POU97] Y. Pouffary, A. Young. "RFC 2126 : ISO Transport Service on top of TCP (ITOT)". March 1997.

[RAJ00] G. S. Raj. "A Detailed Comparison of CORBA, DCOM and Java/RMI". http://www.execpc.com/~gopalan/misc/compare.html. April 30th 2000.

[REI98] P. Reinholdtsen. "Japhar : the Hungry Programmers Open Source Java". Norwegian Unix Users Group. November 1998.

[ROS87] M.T. Rose, D.E. Cass. "RFC 1006 : ISO transport services on top of the TCP: Version 3". May 1987.

[RTJ00] The Real-time Java Experts Group. "Real-time Specifications for Java". February 2000.

[SEI99] L. Seinturier, A. Laurent, B. Dumant, E. Gressier-Soudan, F. Horn. "A Framework for Real-Time Communication Based Object Oriented Industrial Messaging Services". ETFA'99. Barcelona, Catalogna. Spain. October 1999.

[TAO00] TAO. "Real-time CORBA with TAO (The ACE ORB), Real-Time Solutions for Real-Time Systems". http://www.cs.wustl.edu/~schmidt/TAO-intro.html. April 30th 2000.

[VAL92] Valenzano. Demartini. Ciminiera. "MAP and TOP Communications". Addison Wesley. 1992.

[WIN00] Wind River. "VxDCOM Datasheet". http://www.windriver.com/products/html/vxdcom.html. April 30th 2000.