

# Future Directions of Internet-based Control Systems\*

M. Coccoli and A. Boccalatte

University of Genova — DIST, Department of Communication, Computer, and Systems Science

Recent advances in object-oriented, client/server technologies, and the Internet are supplying the technology enablers needed to provide a uniform information architecture that can be used to build software architecture allowing the inter-operation and integration of a wide set of diverse applications. Moreover the emerging standards start playing a significant role in the shaping of automation architectures in enterprises. The inclusion within a classical control system of Internet-related technologies and open distributed application concepts would give the present system compliance with current and future technological trends. At the present time, in the field of the industrial automation, real-time embedded control systems more and more need Internet connectivity for operations of remote plant administration, training, and supervisory activities. In this paper the state of the art in embedded control systems is presented within the field of industrial automation applications, and the technological scenario is discussed, followed by the trend for the evolution in process control systems. Guidelines for the design of innovative, thus competitive control systems are suggested. A case study is presented, outcome of an EC project in which one of the authors is involved, where a remote maintenance system is realized.

**Keywords:** networking, real-time, control systems, embedded systems.

## 1. Introduction

In this paper, the state of the art and future directions of novel Internet-based control systems will be presented together with some guidelines, which should drive software and hardware architecture design for the next generation of process control systems. The proposed target will be achieved by describing a set of components which, joined together, can enable the development of a really open and flexible

software system with full network capabilities specifically designed for embedded real-time control systems (Open Control System). Such a system will exploit the most advanced and promising concepts and technologies, aiming to provide a real-time process control engine running on standard platforms both hardware and software (commercial off-the-shelf, COTS) thus avoiding the disadvantageous use of proprietary ones. The entire system will strongly leverage on Web related technology, field-bus (device networks), control networks, and standard communication/interfacing methods, thus enabling the system to be used within Component Based Solution Assembly (CBSA) ar-

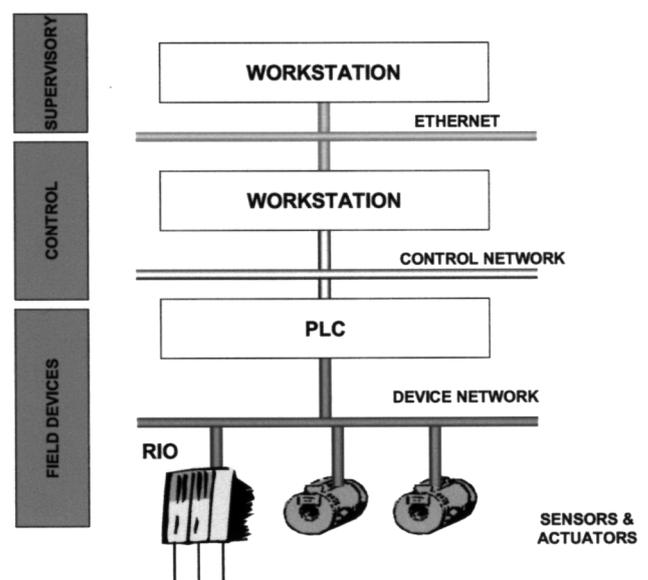


Fig. 1. Control systems' layers.

\*This work has been supported by the European Commission Directorate-General Information Society, contract No. IST-2000-28318 NEOXITE.

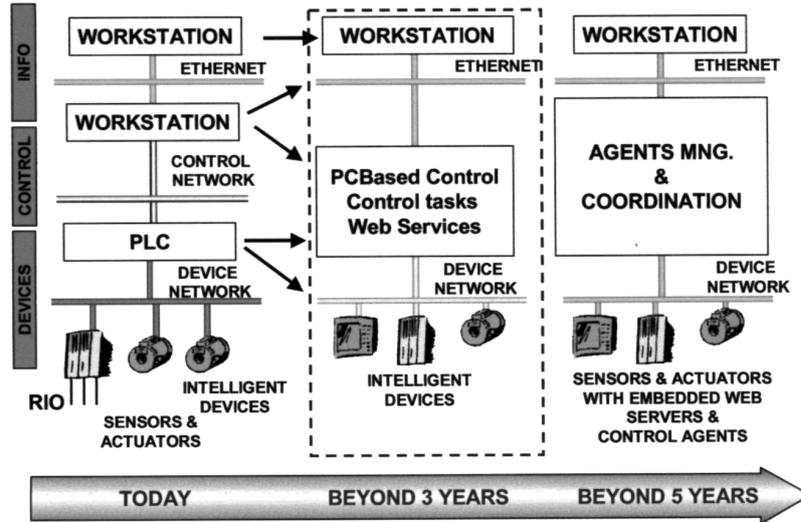


Fig. 2. The evolution of process control systems.

chitectures or together with Laboratory Information Management Systems (LIMS). Indeed, adoption of field-bus technology will provide a flexible and open communication layer for the link between the control level and the devices' one as it is sketched in Fig. 1. Such a system, devoted to an effective plant control, will also embed and demonstrate optimisation by means of advanced mathematical tools, thus providing a modular, flexible, customisable, and self-sufficient control system for even complex processes and applications, outlining the main features for the future genera-

tion of truly advanced Flexible Manufacturing Systems (FMS). Moreover, the inclusion of Internet technology will ensure full compliance of the system with current trends of information technology (IT) within process industry towards future control systems architectures, where devices will more likely become smart and autonomous, also embedding web servers as well as connection services.

The proposed technical and conceptual scenario can be graphically summarised as it is represented in Fig. 2, in which an interesting trend is

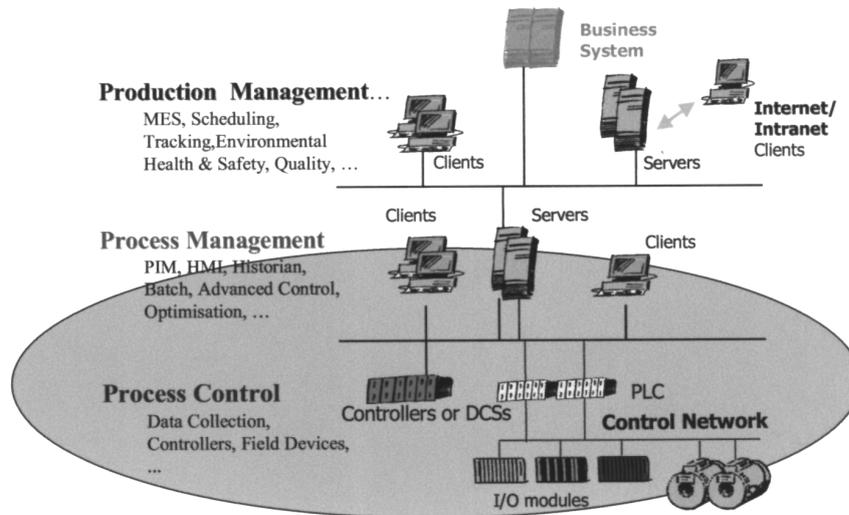


Fig. 3. The area of process control.

presented, within process control systems, that, at first glance, could be considered quite “visionary”, but, on the other hand, IT researchers believe it is more likely achievable within five to six years. In Fig. 3, the project context is highlighted, that is the automation level which will be the main target for the proposed concepts. Process industry community is increasingly looking for easier and seamless integration between these layers, but minimising, even if not trying to completely skip, the development of custom code. On the other hand, end users are also wishing and expecting modern integrated plant systems embedding complementary, and usually customer/process customisable, tools for optimisation such as statistical process control and advanced control algorithms (e.g., simulation tools, on-line parameter tuning). At present, in most cases, such tasks must be performed by external systems (e.g., mathematical models and observers running on dedicated PCs) as it is not possible for them to be executed on PLCs or even on some DCSs.

A control system designed and developed according to the guidelines outlined within this paper will be able to perform standard as well as advanced control mechanisms by means of a main control engine, including both local and remote human-machine interfaces (HMIs), standard networking capabilities, and web services. The resulting system will definitely represent a step ahead towards next generation of process control systems. After this short introduction, the state of the art for real-time control systems will be outlined in Sec. 2. Needed innovation will be enumerated in Sec. 3 while the migration from PLC to standard PC architecture will be described in Sec. 4. Some technical highlights will be discussed in Sec. 5, and conclusions will follow thereafter.

## 2. State of the Art in Process Control Systems

A traditional control system can be split into three main layers: the field, related to the input/output devices, the controller itself, which is in charge of executing the control logic, and a monitoring, programming, and supervision level as sketched in Fig. 1. As already mentioned, at present, due to many reasons, the

integration between these layers cannot be so smooth. A variety of communication protocols, proprietary technologies, and non-standard interfaces exist and are widely used. For example, current systems use very different networks typology for the interconnection of their three main layers. Such networks, usually proprietary, may show very different characteristics, from both functional (protocols) and physical (electric signals) points of view, therefore impeding the desired smooth integration and information flow throughout the above layers. Today, only the network systems in the upper level can be considered close to the standard, because of the increasing use of Ethernet and the success of TCP/IP. The other two networks, specifically the device network and the control network, are still based on more or less proprietary protocols, even though during the past few years several different field-busses have been adopted, some of them internationally standardised, some other being “de facto” standards.

Currently available technology provides great potentiality, which is confirmed by the increasing importance of IT within modern and competitive enterprise systems. As regards plant systems, over the past years the focus has moved from the hardware to the software, mainly due to the achievable system flexibility. Migration of control capabilities from dedicated devices and systems (e.g., PLCs and DCSs) towards standard architectures is fully in line with the mentioned technological trend. Moreover, increasing reliability and performance offered by standard PC platforms are gaining ground. Therefore, what could be considered visionary some years ago, will be more likely the state of the art within three to four years. Convergence of different functionality towards a single piece of hardware is, as a concept, increasingly accepted, even if, from a very practical point of view, the era where PLCs and DCSs, or any process control-dedicated architectures, will be completely replaced with the new ones, is still far to come. In fact, the industrial community is somehow reluctant with respect to discontinuous innovation.

Here, an influential theory is worth mentioning. Well known within evolutionary biology community, it states that evolution does not gradually occur as small steps over long periods of time, but rather as long periods of stability interspersed with rapid changes. These periods of

rapid changes are often the result of severe environmental changes. This theory is called punctuated equilibrium (Harrold, 2000). Analysing the story of process control systems, there is evidence of the punctuated equilibrium behaviour in this area and over the last few years, Internet-related technology (when fully and profitably exploited) has become the cornerstone for the evolution towards a definitely new generation of process control systems. Recently, due to economical reasons too, no sensible evolution has been observed within these systems, in the sense that most of the steps ahead have always been strictly related to smooth adaptations and slight modifications of systems rather than to competitive evolutions/revolutions. On the other hand, it is also true that “control is control”, in the sense that behaviour and general strategies which can be adopted in controlling a valve are quite well defined concepts. Therefore the innovation should be considered embedded in where the behaviour is “generated” and managed.

Finally, marketing investigations show that from a commercial point of view, many companies and software houses state their ability to provide competitive PC-based control systems. Some other companies claim to have such products, but further investigation shows that most of them are still based on proprietary solutions. As a general consideration, integration of very different functionalities that are not present within mentioned products will be an innovation itself.

### 3. Needed Innovation

As pointed out in the previous section, over the past few years, industrial automation community, the end-users in particular, have been very careful in adopting new technology, both hardware and software, due to the small amount in terms of performance enhancement that they could get compared to the drawbacks that an upgrade of the whole production system could introduce. Indeed, recently the degree of acceptance concerning the most innovative IT criteria and approaches has been fast growing. This is so, because confidence in emerging techniques has increased, and, above all, these new technologies can provide an immediate and effective answer to most of the pressing demands from the production that a modern and competitive industrial company feels urgent, (e.g.,

cost reduction, time-to-market, enterprise-wide integration). Up to now, the problem has been the lack of technology infrastructure needed to build a uniform and consistent information architecture that could work for both low-level process control systems and high level plant automation and business systems. In the process control area, traditional, closed loop regulation techniques are thoroughly known, well established, and operational. Still, there is a lot of room for innovation and optimisation of control algorithms (actually most of them are simple P.I.D. controllers tuned by trial-and-error procedures); in particular with novel low cost processors easily allowing large number crunching and very fast data processing, and exploiting fast and reliable network communications, a real-time control system can be driven by data from the sensors, together with the results of optimisation algorithms, identification tools, observers on non accessible variables, that could be performed on-line, in order to get an effective adaptive real-time control of the plant. Moreover, classic theory on adaptive control can be profitably used on production plants. In this field, there is a lot of room for innovation regarding the adopted control algorithms. With the aim of improving a plant's performance, it should also be considered the possibility of having an estimate on some of the non-accessible variables via identification procedures or even through the definition of observers. Control strategies based on optimal predictive control can be considered too. Such a technique will take into account innovative topics as multi-objective functions and will produce multivariable model predictive control techniques and tools that can be effectively used in embedded systems. Within this context, a promising and interesting aspect of innovation is the possibility of integrating the control part with a simulation environment in such a way that tests on different control strategies or huge fine parameter tuning actions can be performed off-line, always with the aim to improve the plant performance, while, through the exploitation of suited numeric integration algorithms, small memory-foot print simulation environments can be obtained to act on-line with the plant control. Actually, there are many math tools which are widely used in most research laboratories and they are becoming more and more popular in industries too; the possibility of integrating a real-time control system

with the design phase of the control strategy should appear a very attractive item to the end-users, because, in this way, one step less would be necessary between the design phase and the implementation. Moreover, the use of a simulation environment will allow decision making founded on the simulated effect of control actions, based upon a what-if basis. That is, the simulator will be used not only in the design phase but also in real-time. It is possible the realisation of interfaces with the available today's simulators as well as the realisation of a small, embedded self-contained own simulator based on fast computing rather than advanced graphical functions.

On the side of information technology, continuous improvement of traditional programming techniques or adoption of a platform-independent language should be performed for the sake of the source code portability. But, specifically, the reader's attention should be focused on the ongoing innovation of operating systems dedicated to real-time applications (real-time operating systems - RTOS), on the use of graphical user interfaces, and on the network connectivity. Given embedded computers' unique and varied requirements, a graphic user interface (GUI) solution must be flexible and the integration of a GUI with the control part of the plant should be done always keeping deterministic real-time characteristics. Another interesting point is the development of networked systems where, within client-server architecture or distributed systems, office applications can share data with any real-time database running with the control system. In such a high-end networked real-time embedded controller, Internet applications can be foreseen as well as distributed programming techniques use will be enabled since the possibility of remotely monitoring a plant is very attractive especially thinking of control systems working in hostile environments (e.g., gas, oil, emergency conditions). Such a system will be also suitable for the implementation of virtual laboratories and it will be able to offer easy connectivity towards Laboratory Information Management Systems (LIMS). In this respect, recent agent-based programming techniques could be exploited and new paradigms for control systems architectures could be defined (Ivezic et al., 1999, Jennings and Woolridge, 1996). A competitive and effective exploitation of these general-interest

concepts can only be achieved through the re-definition of what should be the model for an embedded real-time control system in the direction of network services, openness, flexibility, and scalability.

#### 4. Moving Process Control Tasks from Dedicated Platforms Towards Standard PCs

It should be clear, at this point, that the central role standard hardware platforms will play in the near future as well as the fact that the proposed migration implies a lot of issues, ranging from the provision of real-time and deterministic performances in run-time mode, up to control tasks configuration environments and debugging tools. Also the hardware part of the system should meet specific criteria, such as using silicon disks instead of mechanical ones, ability to correctly boot and run even in non-standard configurations (e.g. keyboard, mouse or graphical cards not present). Moving control tasks from PLC to PC leads to the adoption of field-busses for the connection of the plant devices to the soft-control engine. Since flexibility is one of the most important features characterizing such systems, a strong flexibility should be guaranteed as far as the usable field-busses are concerned. Integration of features other than just the control ones, such as local human machine interfaces, archiving capabilities within standard common repositories, open interfaces to third party applications and legacy systems, and standard networking capabilities are the winning factors of these architectures, allowing the users to profitably include PC-based control within a larger context and since modern computers allow number crunching and very fast data processing, it is more likely foreseeable that advanced control algorithms (e.g., optimisation, multiphase loop controllers, accurate mathematical models) could be executed on the same machine, therefore eliminating the need of specific PCs dedicated to the control task, that is money saving. Diverse technical solutions should be evaluated, such as performing all the tasks on a single CPU or even distributing the workload between the two CPUs on the same bus. The results would strongly depend on the future evolution of CPUs, their reliability

and their cost reduction trend (as projection for future commercial exploitation of the system).

On the other hand, recent advances in object-oriented, client/server technologies have provided the technology enablers needed to provide a uniform information architecture that can be used to build software architecture allowing the inter-operation and integration of a diverse set of applications. Moreover, the emergence of standards in the domains of operating environments, networking, GUIs, and repositories starts playing a significant role in the shaping of automation architectures in enterprises. The inclusion within control systems of standard Internet-related technologies and, more generally speaking, open distributed application concepts, will guarantee compliance of the system with current and future technological trends, i.e. standard networking protocols, full adoption of multi-tier architectures strategy, web-based interfaces, etc.

## 5. Technical Highlights

In this section, the most innovative aspects of the proposed architecture are presented. Such features are related to the development of network services, flexibility, scalability, openness, a set of new functionality incorporated into the control system in order to enhance plants efficiency and control process quality, a business wide access to information, possible integration with commercial and trading systems, and, last but not least, global reduction of costs. Future control systems will show their unique ability to unify plant-level control and other systems with enterprise resources planning (ERP) or business systems. Such a theme is becoming increasingly important to manufacturers around the world. So, it should be offered a completely integrated set of applications in a fully networked environment with a reliable underlying infrastructure that permits easy linkage with other systems, and also has graphical human-machine-interface (HMI) programs for monitoring and configuration activities.

Flexibility mainly resides in the ability of integrating input/output devices and in the adoption and integration of standard computing architectures, both from hardware and software points of view. From the economical and functional

aspects, choice of the right I/Os is one of the key points in the design of any control system architectures: such a choice can even be driven by efficiency criteria. By exploiting and further enhancing the field-bus technology that can provide the end-users with a very high degree of flexibility in selecting the desired I/O devices, the users will be enabled to easily compose their own customized I/O system by selecting what they consider the best suited products (e.g., with the most convenient performance index and/or prices) from the wider and wider range today's market is able to offer. So the end-user will be definitely enabled to tailor its I/O system to the real needs of the plant, even according to a cost-based analysis. Another key point is always the choice of the computing part since the adoption of standard computing architectures can provide manifest benefits, guaranteeing the best flexibility in the selection of the characteristics of the system and in the choice of the components' suppliers for the entire control system.

Industry standard protocols such as Asynchronous Transfer Mode (ATM), Transmission Control Protocol/Internet Protocol (TCP/IP), and the Hypertext Transfer Protocol (HTTP) have been enabling the development and deployment of standard networking architectures so that standard network communication protocols can be adopted for data exchange. Moreover, emerging distributed technologies for the linkage of diverse software applications can be regarded as powerful means towards the development of a real-time control system fully integrated in a wide office automation and business process context. According to all of the above considerations, it is clearly emerging that an innovative embedded real-time control system should be independent on specific solutions, and should rely on standard communication protocols; moreover, it should be easily integrated with other standard products, and it should be easily customisable to the user specific requests.

Matching the needs of end users, device suppliers, and software developers too can be obtained through a specific tool able to guarantee interaction with the many different parts of the whole control system through a complete set of interfaces, the so called abstraction layers. Throughout the adoption of abstraction layers technique, the core engine of the control system becomes insensitive with respect to

particular implementations you are faced with. Moreover, a strong interaction with the plant is guaranteed through the field-bus, and the connections with other office applications happen through TCP/IP communication layer. Interfaces should be provided to enable data exchange between different software applications, and they should permit control strategy design and implementation via mathematical models and tools possibly based on the use of emerging graphical programming languages or specifically designed mathematical algorithms (e.g., adaptive control, sliding mode control, model predictive control) depending on the context the plant is working in.

The embedded market has recently changed such that the basic operating system is just one in a long list of requirements to be considered when beginning the design cycle of a new project. The onslaught of cheaper processing power coupled with new powerful programming languages and development tools and environments, has led to very complex, sophisticated software and to a variety of new products, including information and Internet appliances, and other consumer-oriented embedded devices. Today, beyond the core operating system technology, any developers require a complete package of products and services to successfully build their applications within a strict timeframe. In response to these changes, real time operating systems too are evolving and from the monolithic entities of embedded systems' early days, they have been growing towards today's multileveled hierarchies, in which the lowest level is a real-time kernel. Reliability, connectivity and scalability, together with the building blocks for flexibility and customisation such as portability and GUIs support, are mandatory requirements for any innovative embedded systems. Due to important price sensitivity, functionality requirements, and resource limitations, an embedded system should also offer the possibility to be the most scalable and configurable such that only the applications' necessary components will be used. In fact, unlike desktop systems, embedded systems may want to run with a limited hardware budget, for low cost solutions, and thus must make do only with what is needed. Operating system portability will give the developers essential flexibility in application design. Since embedded computers are a complete custom in-

tegration of hardware and software components, an operating system should support a wide variety of processors and be easily portable to both custom and standard hardware.

Within such a new philosophy, advanced functionalities should be located into single controller boxes. The "box" model could be an off-the-shelf Personal Computer industrialised to meet reliability and environmental specifications such as incorporating some HMI functionality and implementing an advanced software DCS. New functionality to be added to the embedded control system should mainly cover the graphical user interface needs and control capabilities enhancement in fact one of the key components of the flexibility concept is the ability to build brand differentiation through the user interface. Due to the decreasing hardware costs, new functionality and the Internet phenomenon, graphics support for an operating system can become a key building for the embedded device. Interactive and intuitive graphical user interfaces (GUIs) enable the user to more easily control any device and access the features previously buried in static "manual" interfaces with buttons and alpha numeric displays. Given embedded computers' unique and varied requirements, a GUI solution should obviously be flexible.

Finally the proposed design philosophy for an innovative control system can be resumed in the following key-points:

- Portability of the core engine on different platforms.
- Scalability
- Fulfilment of OEM market needs
- Flexibility in selecting the field-bus and the I/O system
- Control and supervisory capability
- Integrated functionality
- Standard networking capabilities
- Openness to other software products available on the market

The architecture could be the one shown in Fig. 4. Each box of the above can be regarded as an independent component of the embedded real-time control system. The assembly of the above

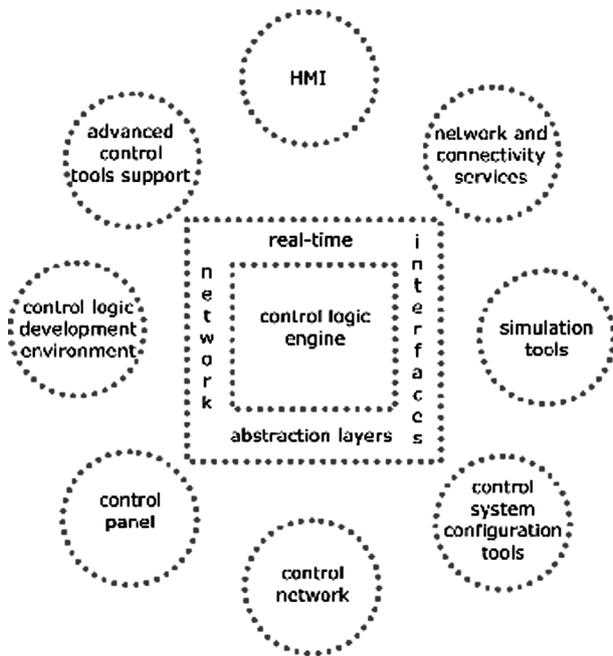


Fig. 4. The proposed control system architecture.

components gives rise to a complex and complete real-time control system whose kernel is integrated by configuration and network facilities, graphical programming environments for easy systems' configuration and development, in addition to advanced tools devoted to process control and simulation for the definition of the best suited control strategies.

Only if based on standard components and architectures, such an embedded real-time control system will be able to offer the needed degree of granularity, in order to fulfil the most diverse implementations. Development of the Control Logic Engine will have to be oriented to produce not just another soft-logic software but an effective execution engine for a DCS-oriented system. By the full integration with all the described modules, users will be able to develop and implement any customised control strategies and easily interact with them in a profitable way.

From the side of the control logic engine development, it is obvious that real-time characteristics are needed: so, the software development platform will provide a real-time multitasking kernel which will have to support basic actions for real-time programming, such as create, delete, suspend, resume on typical real-time multitasking objects like semaphores, tasks or threads, tickers, shared memory, mutex, events.

The software platform will also have to provide a TCP/IP stack enabled for UDP protocol with BSD sockets interface. Moreover, for the sake of generality, the hardware platform will have to support 8, 16, and 32 bit (signed and unsigned) types of variables. IEEE754 compliance should be enough for floating-point variables.

The control logic engine software architecture should be designed by using abstraction layers. An abstraction layer is a portion of code allowing customising of the control logic engine over different hardware and software platforms. In this respect, the abstraction layers shall be modified each time a new configuration is foreseen, that is the porting activity to any particular platform/architecture. It can be foreseen that drivers for most common configurations and widely diffused field-busses will be available from hardware suppliers or system integrators. Typically, hardware, network and kernel abstraction layers are necessary but it is up to the control logic engine designer the realisation of a different project which could even be more articulated and complex depending on the targeted application.

Based on network technologies, it appears very important to develop a standard communication layer in order to allow the embedded controller to interact with any other application at different levels of business automation. In order to connect the control system with the rest of the world, a set of connectivity services will have to be considered.

### 5.1. Graphical Applications

With the aim of supplying the embedded control system with a graphical user interface, in a client/server-based networked architecture, there should be no constraints on the choice of clients typology, in particular for remote monitoring applications, where the portability to different platforms has to be considered mandatory requirement for a system to be open. In order to have full portability and compatibility in an Internet environment, modern object-oriented platform independent languages such as Java or JavaScript combined with W3C standards HTML, XML and XSL, and cgi-bin or asp technology seem to be more adequate, and should be provided. For example, for the development

of a control panel, it is important to point out that a web-based application could run on any computers only having a www browser without any additional costs for software licenses or any particular royalty agreements with software companies. Developing a web-based control panel would be a very low cost solution for the realisation of an advanced graphical user interface which could run in parallel and by-pass, or even replace the traditional industrial controllers' control panel.

For the development of the programming and configuration environment too, the graphical interface will have to cover a fundamental role and the same considerations as before are still valid. In fact the configuration environment as well as the control logic development environment aims to give a user-friendly interface for making the control system description and definition easy, and for defining and programming the control strategy. In the phase of description of the control system, the platform and the I/Os are chosen (possibly among standard ones) and the needed device drivers' code will be linked to the control logic engine that will run on the controller. The control logic development should be done within a user-friendly environment for the programming of the control logic engine functionality that is the choice of the best control algorithm depending on the present application and according to the plant necessity too.

## 5.2. Advanced Control Support and Simulation Tools

Based on both classical and non-standard control theory, different algorithms for on-line parameter tuning and system identification will be developed by means of standard tools and with the aim to be easily interfaced with the other components of the system as well as with other third party or commercial tools. By means of simulations in a suited mathematical environment, it will be possible to predict the behaviour of the system in response to particular inputs or any changes in its control parameters. In order to keep the system open to any possible solutions, it will be provided with an interface to external simulators and mathematical tools so that the control system designer will be able to choose the degree of depth of his analysis de-

pending on the tool used and its characteristics in terms of accuracy of mathematical models.

By the design of suited numeric integration algorithms, a small memory-foot print simulation environment can be obtained. Otherwise many math tools exist and are widely used in most of the research laboratories, even in the industries.

## 6. Conclusions

In the presented scenario, a strong need for developing new technologies and tools supporting the fast and reliable implementation of real-time applications, in particular control systems, has been observed. For consumer applications, the main benefit from the adoption of the proposed design paradigm can be in cost and effort reduction for the design and the development of their embedded real-time control systems, which can be built by means of the integration between off-the-shelf hardware and some suited software components. For professional applications a sensible reduction of the complexity of a real-time control system can be obtained. Industries, end-users, and control systems vendors are more and more perceiving networked technologies, embedded solutions, and distributed control as the right way to meet the above mentioned requirements. Moreover technologies and performances are rushing, and we can expect that virtual plant's concept will be effective in the future and this fact can be seen as the first step in the direction of the development of virtual laboratory-based industrial research areas. Such a heterogeneous collection of emerging technologies, ranging from distributed data handling and distributed computing to the underlying multicast network infrastructure, has the potential to make it possible to create distributed scientific and industrial laboratory environments that provide complete location-independent access to instruments and analysis resources, thus enabling remote human collaboration. This possibility can give the ability of experimenting with pilot plants by remote control from different laboratories through the Internet in such a way that both scientific community and industry will benefit from information infrastructures.

The aim of this paper has been the description of future directions and the definition of guide-

lines for the development of a complete set of software modules (components) oriented to embedded applications development and deployment, specifically targeting OEMs (embedded control). In few words the proposed system is going to promote the convergence of information processing, networking technologies and infrastructures. Based on current strength in access network technologies and embedded systems, it can bring together a wide range of essential technologies development related to a really open platform for control systems, that is one of the more interesting topics in this research area. The adoption from the developers' side of a component-based approach will promote software re-usability. The use of off-the-shelf components will also be strongly considered in the definition of the system architecture which will be based on novel standard hardware components.

## References

- [1] D. HARROLD, Here we go again — Where control systems have been, and where they are going, *Control Engineering*, (January 2000).
- [2] N. IVEZIC, T.E. POTOK, L. POUCHARD, Multi-Agent Framework for Lean Manufacturing, *IEEE Internet Computing*, Vol. 3, No. 5 (Sept/Oct 1999).
- [3] N. R. JENNINGS AND M. WOOLDRIDGE, Software Agents, *IEE Review*, (January 1996).

*Received:* November, 2000

*Accepted:* February, 2002

*Contact address:*

Dr. M. Coccoli  
 University of Genova — DIST  
 Department of Communication, Computer, and Systems Science  
 via Opera Pia, 13  
 16145 GENOVA, Italy  
 Phone: +39.010353.2284  
 Fax: +39.010353.2154  
 e-mail: coccoli@dist.unige.it

Professor A. Boccalatte  
 University of Genova — DIST  
 Department of Communication, Computer, and Systems Science  
 via Opera Pia, 13  
 16145 GENOVA, Italy  
 Phone: +39.010353.2284  
 Fax: +39.010353.2154  
 e-mail: nino@dist.unige.it

---

ANTONIO BOCCALATTE graduated in Electronic Engineering (1976) at the University of Genova. In 1987 won a chair at the University of Genova, and presently is Professor of "Data Base Management Systems" at the Faculty of Engineering. He is involved in research activity on System Architecture and Artificial Intelligence.

---



---

MAURO COCCOLI received the Ph.D. degree in Electronic and Computer Science Engineering from the University of Genova, 2000. His research activity covers complex non-linear uncertain systems modeling and control, discontinuous control techniques, simulations activity of industrial plants, software agents, information technology in industrial automation and web-related activity.

---