

Using push technology for industrial automation: new ways of conceiving Internet / Intranet for industries

Andrea Molinari and Luigi Colazzo

Università degli Studi di Trento

Dipartimento di Informatica e Studi Aziendali

Via Inama, 5 - 38100 Trento - Italia

Tel.: +39-461-882324 - Fax: +39-461 882124

e-mail: amolinar@cs.unitn.it

e-mail: colazzo@cs.unitn.it

Abstract

The paper presents a prototype that allows shop-floor personnel to perform operations on remote computers using Internet, from controlling intelligent devices to instructing remote personnel about manufacturing processes. The communication protocols and software are typical of a Internet / Intranet environment, but the ubiquitous WWW paradigm is not used. A new approach representing an evolution of the emerging push technology paradigm is presented. This allows new forms of interaction between industrial intelligent devices and remote operators, or among remote operators that must interact in some way. As an example of this approach, an on-line distance learning environment is presented where a hypermedia can be piloted remotely from a master station, in order to instruct shop-floor personnel on specific procedures.

Keywords

push technology, internet, factory automation, distance education

G. Jacucci et al. (eds.), *Globalization of Manufacturing in the Digital Communications Era of the 21st Century*

© Springer Science+Business Media New York 1998

1. INTRODUCTION.

The technical consequences to the explosion of Internet has a tremendous effect on what we can do with computers communicating among each other without being physically in the same place, and the WWW is only one aspect. Most industrial companies are using Internet technologies associating it exclusively with the WWW metaphor. Nevertheless, industry has not been still touched in its core operativity by this phenomenon. Like the major part of production fields, Internet and (especially) the Web represents only something that everyone must have.

The factory automation field has been widely dominated by workstations' technology, where concepts like TCP/IP, client/server, Remote Procedure Call (RPC) etc. has been available for a long time. Now PCs can use the same concepts and tools, having a very favorable cost/performance ratio with respect to Unix-based workstations. Now, from home through a modem we can communicate with every PC in the world, also with a PC that is controlling an industrial equipment. Therefore, the real revolution introduced by Internet is a more subtle and hidden revolution than the Web: *every* computer (not few Unix workstations) can load a communication protocol (TCP/IP) that other millions of computers load. This fact enables the communication with every other PC through the Internet, and therefore allows us to take advantage of the Web, but also of e-mail, newsgroups, mailing lists and so on.

For the specific industrial field, advantages of connecting computers all over the world are trivial though not still completely available. We can imagine a data-field acquisition with a traditional DAQ board in a PC, but with all the data management, the decision of acquisition startup, the processing of results managed by a remote operator through an ordinary Internet connection. Another possible application is the control and change of a production cycle made remotely by a distributed control system (DCS) through Internet. Furthermore, also education in the industry can take advantage of Internet: think, for example to the possibility of instructing operational personnel on a oil platform in the North Sea about a particular procedure of opening a valve, all done using an hypermedia piloted by a remote teacher through the use of Internet.

All these prospects are fascinating, but what is partially wrong in our opinion for these kinds of factory automation is the dominant World Wide Web metaphor, i.e., disseminating information from a dumb server to clients requesting it. If we think at this scenario in an industrial environment, like a shop-floor, we can see that this model of interaction is suitable only for few applications, not directly connected with the core production process. In fact, we can have this relationship in the shop-floor only if the PCs with intelligent devices connected (PLC, CNC, Robot, MHS etc.) act as a WWW servers, collecting the manufacturing process data and making them available to operators' (client) requests. This idea has an evident drawback: all the shop-floor PCs should be, in some way, Internet-based computers with HTTP-server functionalities, to which in turn the intelligent devices are connected. Moreover, all software libraries that interact with the intelligent devices should be written in a Web-aware programming language, i.e. Java, Perl, Vbscript etc..

In this paper we present a different approach in using Internet for factory automation. We refer to a master-slave relationship resembling the emerging push technology, but that constitutes an evolution of it. We have conceived a communication through the Internet in which the slave side (the clients in the WWW relationship) is constituted by a PC with intelligent devices connected, and the master side (the server in the WWW) is managed by an operator, that controls and pilots various remote operations, from data acquisition to process activation to piloting an hypermedia that instructs remote personnel. The paper will explain the theoretical basis of the push technology approach, and will focus on the evolution of this concept that is represented by the prototypes developed. Particular attention will be given to remote training aspects, showing how to implement a distance learning environment in the industrial field.

2. INTERNET AND INDUSTRIES

There are many trivial uses of the Web/Internet for industries, not peculiar to this sector but useful for any company:

- to search for information of various kind;
- to disseminate information about the company (company and products presentation, advertisement, price lists etc.);
- to connect directly with people, from competitors to partners, through the use of e-mail, mailing lists, databases available through Web pages, etc.

Another use of Internet/Intranet/Extranet technologies is common to all markets, and are related to the groupware technology. These collaboration tools allow people to interact remotely using computer technology, having meeting but being physically separated, sharing whiteboards through networks, exchanging pieces of software artifacts (documents, program codes, etc), discussing common topics through videoconferencing systems etc. These products usually involve several people from different departments inside the factory (engineering, marketing, service and support, R&D etc), for example in the initial design of a new product. However, due to the cost of the technology and due to the poor network performances today commonly available , they are not so widespread and not available to medium-low scale companies.

From the same area of collaborating through the WWW technology in manufacturing is the system presented in (Andrew et al.,1996). Mediator is an open architecture information and knowledge management system for the concurrent support of the manufacturing life cycle in a distributed enterprise. It has been applied to supporting manufacturing activities throughout the product life cycle (Gaines et al.,1995), and to distributed education through the web (Norrieet al.,1995). As many products, Mediator was originally implemented using custom software on local area networks and then re-engineered on the World Wide Web in order to support international collaboration in distributed concurrent manufacturing. In this system the WWW is used as a knowledge transport mechanism to access distributed sites and data files, and the browser acts as a file and data serving agent on the Web. A client helper allows users to dynamically link concept nodes in a

concept map editor to Internet sites and files.

A particular use of Internet for industries regards Web-based manufacturing simulation. The WWW, with its ability in disseminating information and facilitating distributed business decision-making, has the potential to improve the deployment of simulation technology. For example, a simulation model may reside on a computer at location A, and it can be accessed by a client machine at location B via the Internet. A web browser can enable the user at location B to execute the model and see its results.

In many projects involving industries (see, for example, (Kazlauskas et al.,1997) and (Sheikh et al.,1997)), we can see the use of the Web for distance learning purposes. In this case, we have the publication of educational material on the WWW, specific for industrial topics and their peculiarities. In many cases, however, the use of Web Computer Based Training (CBT) is not specific for industries, but has simply industrial specific contents. It is obviously different having a classic student-teacher relationship, rather than a worker-foreman relationship. This has some implications on how the educational material is built, but have no implication on technology that is used to do this. We will discuss in detail this use of Internet in section 4.

In conclusion, in industries Internet-related technologies are nowadays more devoted to general-purpose uses rather than affecting core-production activities. All these examples have the client/server metaphor typical of WWW in common. In this paper, we want to show a different approach to the Internet resource inside manufacturing and control systems in a manufacturing enterprise. This refers to a new way of interaction among devices and operators using the Internet-related technology. We are developing some prototypes that allow to "push" data on remote industrial machines piloted by other remote PCs or operators, this all done through the use of Internet. This approach can be applied both in the specific production activities and in the educational field.

3. PUSH TECHNOLOGY:

A NEW WAY OF CONCEIVING INTERNET FOR INDUSTRIES.

In the WWW, clients communicates with the Web server through a predefined protocol (HTTP), requesting HTML documents by means of a click on hotwords defined with HTML language. Here the metaphor is more related to a "pull" mechanism, as the initiative of interaction is in the client's hands. When s/he clicks on a hotword, the request is sent to the server, that replies with an HTML document (or other) (Figure 1)

The advent of new Internet-based technologies is changing the way we can conceive interaction between remote intelligent shop-floor equipment and operators. Push technology is surely one of the most important. Push technology can be simply defined as a set of technologies used to send information to a client without the client requesting it.

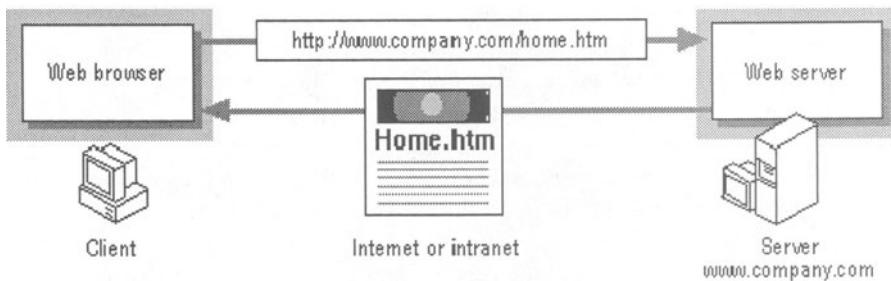


Figure 1. Traditional Web interaction

The server "pushes" information onto the clients, thus reverting the traditional relationship derived from the Web. We have the contrary of what happens on the WWW; the server assumes the control of the "event-action-reaction" mechanism previously in the hands of Web users. It is the server that starts data interchange process, deciding what control data are to be sent to which client computers, and when they will be sent. We already had a early-stage push technology, i.e., E-mail. With this approach we can foresee applications that resemble broadcasting services, like electronic newspapers, software distribution, TV channel - style services. "Active Channels", "Netcaster", "Multicast technology", "Webtops" (instead of desktops), we can find a lot of terms in the IT current "slang" always identifying the same technology.

Push technology has been recently included in the most widespread browsers, where all kind of information can reach the our desktop. Super-software houses like Microsoft and Netscape have implemented their "push technology" tools, mainly devoted to a "TV channel" approach. Netscape has developed Netcaster™, Microsoft has included "channels" inside the latest version of its Internet browser. The concept is the same in both cases: reaching the user's desktop through dedicated Tcp/Ip ports and through the UDP mechanism of broadcasting packets. Every piece of information is packaged in a proprietary datagram that is disseminated through the Internet. A channel is a set of web pages that use push technology to deliver their content automatically to the user's computer in a predetermined manner, rather than being pulled in by an explicit user interaction. Users can customize what, when, and how often channel information gets pushed to their desktop. Summarizing, push technology has the following advantages:

- information can be updated periodically without user intervention;
- content can be personalized, so users get just the information they want;
- information is cached, enabling richer content to be used;

As usually happens when a new interesting technology appears, we have now a wide range of "pushiness", from trivial information dispatchers to extensive user controls over content, amount, delivery time and methods, delivery format of information. There are also some technologies similar to push technology, like off-line Web browsers, private delivering information systems for niche audiences, notification systems of updated content of the site, distribution of software

upgrades. There are even some standards on the run regarding the format of "push packets" delivered to clients: see (Microsoft, 1997) for an example. Generally, the mechanism of interaction between users and push applications is the following:

- the user installs the application that allows to browse push channels;
- the user subscribes to channels he desires;
- the user sets some preferences, like notification preferences, upgrade frequency, path to be followed, way of being warned when new contents is available etc.
- once these steps are done, the servers starts to send information to the user respecting the settings.

Probably push technology (or other technologies resembling the same approach) will be the future of the tangled Web. However, push technology doesn't seem to solve our original perplexity regarding the use of the Web approach in an industrial perspective. Push technology, as it is now, suffers from two great limitations:

- on the other side of the communication channel there is always a dumb server, i.e., a programmed machine and not a human operator supervising processes;
- all PCs with their intelligent devices should act once again as Internet servers, "pushing" information on operators' PCs instead of being queried by them.

However, the major problem regarding push technology is the lack of *interactivity*. The actions performed by the push server are mainly automatic and unattended, thus having limited application to the industrial automation field.

4. REVERTING THE PUSH TECHNOLOGY APPROACH

The approach characterizing push technology at this stage should be therefore changed, using the metaphor of "pushing" but having the shop-floor operator behind the server, and seeing the shop-floor intelligent equipment as clients on which the active server pushes information about, for example, the production cycle. The prototype presented in this paper overtakes these limitations, changing the concept of server into master, and the concept of clients into slaves. What we need in a distributed control system is an *active server*, with a man (not a machine) that controls information flow and decides what to do next. The operator beyond the master station interacts directly with the slaves units through Internet, piloting remote behaviors from a central station.

The prototype has been developed for educational purposes in 1993, and now is available for many different applications, from groupware to remote database interaction to real-time communication among remote operators(Colazzo et al.,1996). At the moment, we are experimenting a prototype that pilots directly industrial tools through an Internet-based network, reaching remote stations, controlling remotely the behavior of attended or unattended stations. Another important application field, described in details in the next section, is the educational application of the prototype, that allows master station to guide remote operators through new processes, showing actions to be done on the remote computer.

The prototype is based on a library for the PC that allows remote processes to interact in "one to one", "one to many" and "many to one" relationship. The

approach is interesting also for security purposes, because we have the operator, and not the server, that can supervise all operations. He acts on the master computer, and his actions are broadcasted (or better, multi-casted) on clients connected to it. Clients, if authorized and/or if another operator is behind them, can respond to the master, interacting with it in many ways, from simply sending data to heavily modifying servers' state. This rate of intervention of the clients into master's programs gives the level of "collaborative" process that can be built among server and clients. At the moment, some different prototypes are under experimentation:

- a distance learning prototype, in which server can pilot remote stations in order to show industrial processes, simulations, lessons, etc to unskilled personnel. In this prototype the client side is a "dummy" station;
- a distributed control system, in which the previous interaction model has been used for remote process launching, data acquisition, etc. In this case, the slave station is still "dummy", but can respond to server with control data, sending for example the results of manufacturing processes;
- a system for managing remote databases: the server interacts with clients' local databases, for example data acquired from a CNC, through the Internet/ Intranet and without CGI, Perl or JDBC, but working on user's database;
- a simulation environment in which personnel interact with a software that emulates manufacturing processes under the supervision of a remote "teacher" that can intervene in the simulation showing what is wrong and how to solve it. This environment resembles some "virtual factory" concepts, i.e., a computer environment used to model or simulate manufacturing a product;
- finally, a collaborative design environment, actually limited to the knowledge sharing and representation of industrial processes. This is the most advanced aspect, still at a primitive stage, but the one that deserves more attention on human computer interaction aspects, rather than technical problems. Allowing people to interact on the same data (a design, a sketch, a process idea etc.) is a very complicated task due to human, not technical, issues.

We present the first applicable prototype created with the library, that refers to distance learning and collaboration among remote operators through the Internet.

5. DISTANCE EDUCATION ON INTERNET FOR INDUSTRIES: THE PROTOTYPES

The Web has given a strong impetus to distance learning, but in part it also has distorted its meaning. A web tutorial is surely a cost-effective method for training in the factory as well as in other environments. Many workers can use a Web CBT at different locations, and each person can decide his/her own path through the material. Internet-based courseware can be classified, in our opinion, into four types:

- Courseware composed of a sequence of Html pages, pre-packaged by the teacher and statically connected, in which the student moves at his own pace according to pre-set paths. This is the most frequently used approach, as it is the simplest, but we are skeptical about its educational usefulness. In fact, the

student must construct the didactic path by himself/herself, and there is no real-time control on the sequence and on what the student reads. The supervision of the learning process available to the teacher is very limited;

- Improvements of the previous approach towards greater interaction, where feedback is provided with tools such as E-mail, form-based text or partially synchronous methods (e.g. chat, IRC). These methods are, however, reduced to a solely exchange of texts, and the student is still free to move in the didactic material by his/her own, with no (or few) guidance from the teacher.
- Courseware used during a classroom lesson, with html pages stored on a Intranet Web server, and teacher using them during face-to-face sessions as conventional slides. The advantage is that the teacher can prepare pages with more interactivity and multimedia material, as probably the network is a company's Intranet with better performances. However, in this case the "distance" factor is completely absent. See, for example, (Bos et al.,1996).
- Systems based, at various levels of complexity, on video-conferencing equipment (for example, shared whiteboards), where the Internet is used as the communication channel. This is with no doubt the best approach among the four, as it is the closest to the real "classroom" situation. Unfortunately, companies need a dedicated high speed connection to use these systems, due to the unsatisfactory performances of the "network of all networks". The bandwidth normally available on the Internet certainly does not provide for the broadcasting of full-motion pictures of the teacher and his/her lesson.

The lack of interactivity is common to most of the approaches, and in our opinion is a limiting factor in the education tied to the Web. For an introduction to our conception of interactive didactic material, see (Colazzo, 1995). In the first two categories, interactivity is reduced to an asynchronous communication. The instructor has limited control over the educational process, and this is not advisable if the lesson regards, for example, a manufacturing process. On the other hand, the mechanisms based on video-conferencing pose the issue of the technological equipment required, the availability of high speed networks and the lack of persistency of the lesson.

Our project tries to eliminate these drawbacks, providing a framework for the construction of hypermedia with which the instructor interacts with the remote students in both synchronous and asynchronous manner. The Web remains a vehicle that is an accessory and is used simply to transmit the voice or picture of the teacher, in parallel and completely independently of the rest of the prototype. The teacher can direct the lesson in a way that guarantees coherence in the exposition of the arguments. This is done in a relatively simple way because the hypermedia on the master's and slave's sides are exactly the same, but activated in a "master" or "slave" state. The hypermedia reacts on messages received through Internet on basis of this activation state. The instructor operates on his/her copy of the hypermedia, and so doing the prototype sends codified messages to the same hypermedia on slaves computers through a pre-defined protocol. The master can hence direct actively the lesson, deciding which information is sent to whom and

when, answering questions, checking the state of navigation etc. Multimedia files, graphics, animations are already present in the connected computers, it is only necessary to activate them appropriately. Actually it is possible to update the slave computer by sending a few tens of bytes, whatever the multimedia content is and wherever the computer is on the Internet. All elements are visible on the teacher/master computer, reducing the disorientation and the distraction induced by the presence on the screen of elements that have nothing to do with the information content. The prototype acts as a simplified multimedia authoring system based on the widely-known Asymetrix Toolbook package, and also provides several tools for the management of the lesson. These tools constitute a sort of "control panel" where the teacher can:

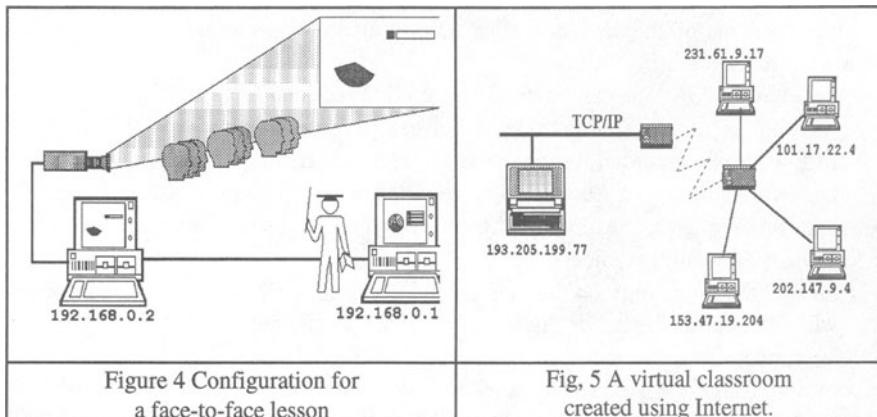
- decide which objects can be displayed on the students' screens (Figure 2);
- see objects that are visible to the slave/student (Figure 3);
- follow a pre-recorded sequence of objects to display, both forwards and backwards: this is very useful when showing how to do a process;
- vary this sequence, by moving objects from their original positions;
- show or hide all the objects;
- display the index only on the master, and access other pages of the hypermedia while deciding whether or not to make these movements visible to the slaves;
- and more.

Figure 2 and Figure 3 also show the simple way by which the teacher sees what is currently visible on student's screen and what is not. Only objects with the hatched border will be shown on the slaves' computers. The complete screen is available to the master, and the slave's screen is instantly updated with a simple click on the wanted object, or choosing it from the control panel.

<p>STARTING CONSIDERATIONS -2</p> <p>■ other "rewriting" solutions:  forse de saltare</p> <p>⌚ html - too simple and poor - no programming -throw away programming already done ⌚</p> <p>⌚ Java - not for end users - not still a RAD for CBT- throw away programming already done ⌚</p> <p>⌚ Why running after Java (\$\$\$)? Tb is a far better, complete, mature environment ⌚</p> <p>⌚ Above all, these (browser-based) solutions miss INTERACTIVITY between teacher and student ⌚</p>	<p>STARTING CONSIDERATIONS -2</p> <p>⌚ Java - not for end users - not still a RAD for CBT- throw away programming already done ⌚</p> <p>⌚ Above all, these (browser-based) solutions miss INTERACTIVITY between teacher and student ⌚</p>
<p>Fig 2 Teacher's screen with parts transmitted to students</p>	<p>Figure 3 The student's screen in correspondence with Figure 3</p>

These tools provide a high degree of interactivity during the lesson. Furthermore, a permanent base is formed for following study: the didactic material, as it is sent to the slave stations participating in the lesson, remains for future uses.

The interaction in this sort of virtual classroom can take various forms. In figure 6 we show the simplest configuration, suitable for a traditional lesson in which the workers are collected in a room. This configuration evolves towards situations that are conceptually identical, especially with regards the operations that the teacher must carry out. The real classroom in figure 4 becomes virtual by connecting slave computers to the instructor's computer through Internet addresses (Figure 5), with workers attending by their computer. The master's control over the connections is provided by the simple panel, shown in Figure 6.



With the same technology and few modifications, the prototype can produce software packages installable on PCs controlling intelligent devices, and these situations can be controlled remotely by an operator in the same way the teacher controls the lesson. As the teacher can manage the lessons' connections by sending different messages or by operating differently with each slave computer, the same can be done for process control applications. The activation mode (master or slave) determining the behavior of the hypermedia is an interesting tool also for settings with educational activities towards trainers, or where the students must have the chance to consult the material in other situations than the lesson. For example, we can think of factory training where several people are initially trained to, in turn, instruct other employees.

At the current state of experimentation many advantages have been encountered, especially for company instruction, mainly regarding the coherence of the lesson guaranteed by the teacher's guidance and the distribution of slave workstations in different locations. As the only things visible to the student are those which the teacher desires, this latter can freely operate on the master computer, consult other material, open another application, consult personal notes etc. without influencing the lesson and without having these movements visible on the connected computers. Another advantage is provided by the fact that the system integrates transparently with a hypermedia development product that is well known and easy to use, without requiring any programming knowledge on the part of the master.

<p style="text-align: center;">AGENDA</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <ul style="list-style-type: none"> ■ Present a new vision of Tb as an Internet SDK </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <ul style="list-style-type: none"> ■ Present some functions to build distributed Tb apps </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <ul style="list-style-type: none"> ■ Discuss relations among Tb, the Net, the Web, Html, Java </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <ul style="list-style-type: none"> ■ Clarify relations among Tb server, crossware </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <ul style="list-style-type: none"> ■ Present some prototypes </div> <p>A. Melisari - ON-Line Learning '97</p>	
<p>Figure 6 Controlling participants during an interactive lesson</p>	<p>Figure 7 A control panel in its real aspect</p>

Using appropriate commands it is possible to directly intervene on multimedia pages or other remote hypermedia. This allows an instructor to modify/extend dynamically the contents of the application during the lesson itself. It is possible, for example, to change the color or position of an object, the text of a label, the aspect of an animation etc. It is also possible to change the most important properties of remote objects, i.e. the code that conditions their behavior, and this gives great dynamism to the lesson. This fact has obvious and important implications. For example, the control panel in Figure 7 is not so simple as the one in Figure 8, where the labels are added "on the fly" on request of a novel worker that has never seen the panel. In a "pre-cooked" tutorial on the control panel, this is not possible because the teacher is not there while the novel worker browses the lesson. Here we have the teacher that browses the lesson coherently, and reacts to possible questions adding new elements to the hypermedia.

As shown in Figure 8, the instructor has all the elements at disposal, and decide what to be displayed on workers' computers and when. When preparing the lesson, the instructor decides to create animations that will be piloted by him/her, isolating some elements of the control panel that are important for a certain context (Figures 9 and 10), interacting with personnel but guiding them in the comprehension of the procedures, all this through the Internet. The arrows and labels of Figure 9 and 10 could have been added also during the lesson, thus better explaining the content, for example, in response to a question coming from the students. An interesting experimentation regards the co-operation between teacher and students/workers during a lesson. In the current version, the lesson's "rudder" is firmly in the teacher's hands. A student intervenes in the lesson by electronically "putting his hand up". If enabled by the teacher, currently the student can:

- send a message to the teacher. This can be broadcasted to all the participants: the teacher therefore acts as a "relay" for student-teacher messages;
- use a marker to highlight parts that are not clear concepts;

- change attributes of the objects on the current slide, such as the position, the text, the color etc., in a way that allows direct communication with the teacher regarding the lesson (for example, a simulation of physical movement carried out by the student by moving objects, and checked by the teacher in real-time).

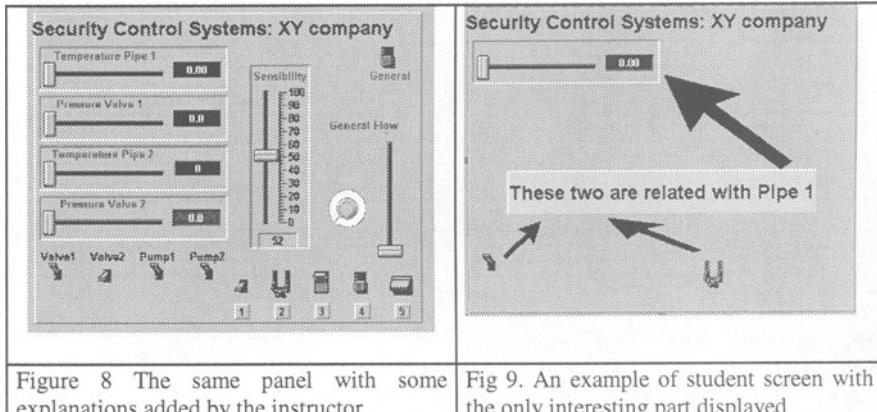


Figure 8 The same panel with some explanations added by the instructor

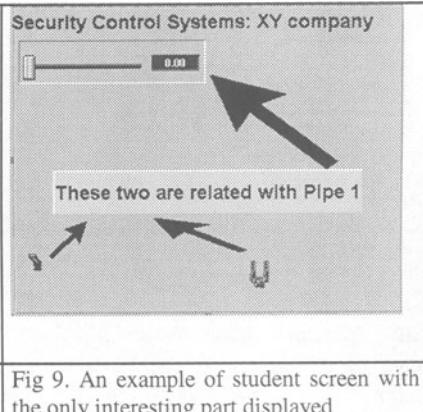


Fig 9. An example of student screen with the only interesting part displayed

The student can also read back through the sequence that the teacher did during the interactive part. Feedback management can be organized in various ways: the system also provides a method of recording the moves made by the students for later evaluation by the teacher. This operation is extremely simple, as the teacher's and student's hypermedia are always "aligned". Reviewing what the student has done represents a sort of "slow-motion" play-back obtained by duplicating the student's moves on the teacher's hypermedia. The current version of the prototype supplies the following basic functions to the developer:

- Actions regarding opening / closure / connection state / forced closure / timeout etc. among the participants, with priority assigned to the teacher;
- Transmission of codified identifications of the objects to be displayed or hidden on the remote stations.
- Remote activation of scripts using a particular protocol, function now supported also by Java through the RMI (Remote Method Invocation).
- Delivery of predefined or "on the fly" scripts to remote sites with their consequent remote execution.
- Creation of objects that were not present previously, modification of their properties and duplication on connected machines.
- Creation of remote objects that were not present and which will not be created on the teacher's computer (for example, the reply to a request of a student).

The protocol also encapsulates service commands sent to the remote computers. The most important include

- Acknowledgement commands regarding certain operations (for example, the request for connection or the launch of a remote command);
- Levels of operation synchrony, in order to create a defined level of synchrony

- or a classical transactional mechanism with Commit and Rollback operations;
 - Levels of security required by the operation underway, especially in the presence of the transmission of commands to be executed remotely.

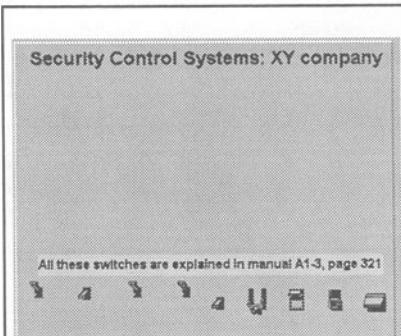


Figure 10. Same screen for teacher, but different result for the student

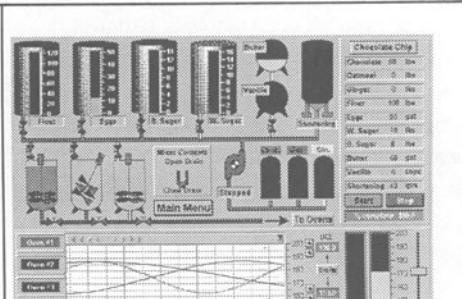


Figure 11 An example of process control application

6 CONCLUSIONS AND FUTURE DEVELOPMENTS

We presented an alternative approach to the use of Internet inside factory automation, with particular regard to the Web-based distance learning. The prototype allows the creation of multimedia lessons in which students and instructor interact in real-time through Internet. The coherence of the lesson is managed by the teacher, and this aspect represents a leap forward in quality in respect to the impossibility for many teachers to create an interactive system via Internet. The development environment originates from a widespread multimedia authoring system that does not require any programming knowledge for the teacher. The instructor prepares the didactic material, decides the sequence of actions to carry out and enables the students at the moment of connection. An on-going development of the system is the creation of a more co-operative didactic environment than currently available, where teacher and students can interact in a non hierarchical manner on didactic material not necessarily pre-constructed.

As regard as specific manufacturing-related implications of the prototype, the advent of new Internet technologies is changing the way we can conceive interaction between industrial intelligent devices and control personnel. The prototype presents a model of interaction that resembles emerging push technology, but with some peculiarities. In fact, control stations and intelligent devices are viewed in a master-slave relationship:

- the slave side is constituted by the intelligent devices connected to a PC;
 - the master side is the computer on which the operator controls and pilots various remote operations, from data acquisition to process activation.

With the prototype it is possible to create applications conceptually similar to the educational ones, but devoted to the control of manufacturing processes or personnel operations on devices. The two copies of the same program are activated

in two different way, one in "master" state and the other in "slave" state. The slave copy will receive the commands (for example, the closing of a valve) that correspond to the action performed by the remote control personnel on the remote copy of the program, all this done through Internet. At the moment, the prototypes in this area of research are still in an early stage, with some experimentation on low-rate remote data acquisition and on remote processes activation through commands sent to PLCs. The perspective is still wider, though not still implemented. Internet-based Distributed Control Systems using Tcp/ip, allow us to imagine a sort of "outsourcing" for the shop floor management. Remote personnel, not necessary employed in the company, can in this way monitor the production process. As an example of what we are foreseeing, imagine the process control system of Figure 11 managed by a remote operator . This picture is a taken from a control system built with National Instrument Lookout™ package.

These applications are showing a bright future for the prototype, but further experiments are necessary in order to evaluate the system in processes where performances are critical. A second aspect of future testing regards security. The correctness of packets received by the remote computers connected through Internet is generally assured by Tcp/Ip, and for educational implementations of the prototype this is not a crucial issue. However, in the case of piloting remote industrial processes, this issue becomes much more important.

7. REFERENCES

- Andrew Z. Lapsley, Brian R. Gaines and Douglas H. Norrie (1996), Concurrent Manufacturing on the Web, WebNet96; San Francisco, CA, Oct. 16-19, 1996
- Colazzo L, Molinari A.(1995), To see or not to see: tools for teaching with hypertext slides, ED-MEDIA 95, World Conference on Educational Multimedia & Hypermedia Graz, Austria, 1995, pp.157-162
- Colazzo L., Molinari A. (1996), Using Hypertext Projection to Increase Teaching Effectiveness, Journal of Educational Multimedia and Hypermedia, Association for the Advancement of Computing in Education, USA
- E.S. Bos, A.Kikstra, C. Morgan (1996), Multiple Levels of Use of the Web as a Learning Tool, ED-Telecom 96, Boston, USA, 1996, Association for the Advancement of Computing in Education, pp. 31-36
- Edward Kazlauskas & I. Soner Yildirim (1997), Integrating Web-based Systems into Teaching Instructional Technology, Short Paper, WebNet97, Toronto, Canada , Oct. 30-Nov. 5, 1997
- Eman El-Sheikh, Fran Bakowska, Chris Penney, Rong Liu & Jon Stickle (1997), Utilizing the WWW for Industrial Training, Poster/demonstration, WebNet97, Toronto, Canada , Oct. 30-Nov. 5, 1997
- Gaines, B.R. and Norrie, D.H. (1995). Knowledge systematization in the international IMS research program. Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics. pp.958-963. New York, IEEE.
- Microsoft Corporation (1997), CDF: Channel Definition Format, Internet document, <http://www.microsoft.com/standards/cdf.htm>
- Norrie, D.H. and Gaines, B.R. (1995). The learning web: A system view and an agent-oriented model. International Journal of Educational Telecommunications 1(1) 23-41.