Designing an Internet based Teleproductics System

João Paulo N. Firmeza, Fernando M. S. Ramos
Electronics and Telecommunications Dep. – University of Aveiro
Campus Universitário de Santiago, 3800 Aveiro, Portugal
Phone: +351 34 370500 Fax: +351 34 370545
E-mail: firmeza@inesca.pt & fmr@inesca.pt

Abstract

The improvement of development tools for the Internet potentates the appearance of new challenging fields of applications to the global network. One new promising field is Teleproductics that comprises the remote access and control of industrial facilities through the Internet.

This paper describes in detail how current Internet development tools can be used for this application, and results from a research project that is focused on the development of virtual learning laboratories for the teaching of mechanical engineering.

Keywords

Teleproductics, Internet, Client-Server architecture, WEB, Distance Learning

1. GENERAL CONCEPTS

Teleproductics is a recent concept that refers to the interaction among industrial facilities, telecommunication networks and computer systems.

Applications of Teleproductics are very wide; examples of applications are virtual laboratories for teaching mechanical engineering (enabling cost reductions due to the overcome of multiplication of similar facilities), and the remote control of self-contained highly automated industrial plants.

A teleproductics system enables the remote operation and control of one (or several) industrial process(es) by using a telecommunications network.

Figure 1 illustrates the concept of Teleproductics.

Figure 1 The Teleproducts concept.

This article describes an implementation approach for such type of system, and includes a discussion on the tools and strategies that will be used and a proposal of architecture.

2. IMPLEMENTATION APPROACH

The implementation foreseen is a prototype of a Teleproducts system, and is intended to demonstrate the capabilities and feasibility of the general concept.

The system to be designed will have the following main modules:

- User interface to the facilities of the industrial process.
- Telecommunications network.
- Industrial site interface.
- Industrial process (viewed in the perspective of the mechanical parts and the electro-mechanical control interfaces).
- Feedback circuits (return video channel and control data transfer).

Figure 2 Simplified diagram of the proposed system.

2.1 The control and management Interface

In this proposal the user interface will be built mostly over an HTTP (*HyperText Transfer Protocol*) client. This client acts like a container for the user graphical interface.

The HTTP client, an ordinary World Wide Web (WEB) browser, will be the support for the display of all the operations related to the direct control of the industrial process. The WEB browser should be upgraded with standard add-ons available as plug-in parts, to enable the implementation of the features required.

The use of a browser as the host for the user interface improves the platform independence of the management module. The system on the operator's side will work under any platform containing an installed browser compatible with the system's specifications.

The target environment for the client will be Microsoft Windows. This option is quite obvious, since it's one of the most common operating environments, with lots of existing commercial applications and development tools. Another strong point of this option is the low cost associated with the solution: Windows is a very cheap operating system that runs on relatively low cost platforms. We must also emphasise its good performance, good graphical user interface and good network component integration.

Resulting from these previous considerations, on the operator's side, we have the architecture represented on figure 3:

Figure 3 The user interface. Implementation over a WEB browser.

2.2. Network

On the proposed system, the network acts like one of the main components, since all the main modules (hardware & software) will rely on the presence of a network.

On the network physical layer, no restrictions will be imposed, allowing the use of any of the common physical topologies, as far as the TCP/IP (Wilderv, 1993) stack could be used.

The TCP/IP protocol is the protocol used on the Internet, and, as a consequence, the protocol used by WEB browsers and servers.

From the implementation point of view, network support will be automatically assured by the use of an operating system with native TCP/IP support.

In addition to that, the real system will have some other network related considerations, which will be focused later on.

2.3. Interface with the industrial unit

In this proposal of implementation, the industrial unit will be based on 4 main modules:

- **Network Input/Output module:** This module is mostly supported by a WEB server. The management interface will be supplied to the operation and management system by this component. This module will be also the one who will receive the data supplied by the operator at the user's interface.
- **Configuration Database management:** This module handles the data archive responsible for the storage of the system's current configuration.
- **Actions generation module:** This module will be dependent of the specific industrial process. Its function will be the generation of actions and commands for controlling the configurable industrial process's components. It will get its input data from the database and, by interpreting the stored data, produces commands to control the physical components of the production unit.
- **Industrial process monitoring system:** This module will be responsible for feedback data transmission to the management and operation units. Depending on the specific system, there may be several types of feedback data produced by several systems' elements, such as video signals, alarms, reports, production ratios, etc.

Figure 4 shows the basic topology of the interaction system with the industrial unit:

Figure 4 System's topology: Interaction with the industrial process.

2.4. The production process

The production process may be composed by a set of electro-mechanical elements controlled by one or several central control units (CCUs). The central control units must be able to receive the configuration data transmitted from the server for each of the process's elements, and keep track of its current states.

Figure 5 illustrates the concept of a process (a production line) controlled by a central control unit.

Figure 5 Diagram of a production line controlled by a central control unit.

3. THE WEB BASED CLIENT/SERVER SYSTEM

All the implementation issues related to the operation, control and management of the remote industrial process would be built using an HTTP client/server topology (Elbert/Martyana, 1994) (figure 6).

Using this approach there will be an HTTP client (a WEB browser) on the operator's side, and an HTTP server on the industrial process side.

From this point on, we'll refer to the system on the operator's side as the *Management and Operation Unit* and to the system on the production side as the *Production Unit*.

Figure 6 Client-server approach.

3.1. The management and operation unit: HTML¹ Interface

The implementation of the user's interface will be built over a WEB browser. This enables the use of any platform on which the chosen browser is supported. This is one of the greatest advantages of using this approach, since it allows this important system's element to be architectural independent. If we choose, for instance, the *Netscape Navigator*² as the browser to be used we can install management clients on any of its supported platforms (Windows 3.1, 95 and NT, Macintosh, SunOS, HP-UX, Solaris, Linux, BSD and AIX).

In the proposed implementation, some browser add-ons or plug-ins will be used and, as a consequence, the browser should support them on each of the used platforms.

To minimize the effect of such a limitation and in order to be able to support less powerful platforms, the system will always offer, as a complementary option, standard HTML (December/Ginsburg, 1995) versions of the user interface.

The user approach to the management and operation interface will be built following a top-down perspective using this logical strategy:

1. As a first approach to the management and operation system, the operator will access a tridimensional model of the whole production facility, allowing him to "walk" inside it just by using a pointing device (usually the mouse). This is done using a plug-in for the browser, which enables it to render VRML (*Virtual Reality Modelling Language*) (Pesce, 1995) objects. If Netscape Navigator is 3.0 or Microsoft Internet Explorer³ 3.0, the VRML is used, VRML extension is a standard. The use of this tool enables the representation of models of the reality just through interpreting simple text script files. Just navigating inside the 3D space of the process model, the operator easily reaches each one of its components (figure 7).

Figure 7 Example of layout of a production line implemented in VRML.

¹ HTML: Hypertext Markup Language.

² *Netscape Navigator* is a registered trademark of *Netscape Communications Inc.*

³ *Internet Explorer* is a registered trademark of *Microsoft Corporation*.

2. By clicking on the representation of the process's components, an HTML page will be displayed on the browser, allowing the user to get information about the corresponding component; furthermore forms to input the data necessary to configure the component may be displayed (figure 8). This data will be stored on the server, located on the production unit, as will be discussed later.

The screenshot shows a Netscape browser window titled "[Production Line N° 1]". The address bar shows "http://www.1000.com/1000.html". The main content area displays the following information:

- Production Line N° 1**
- Machine #23**
- Configuration Data**
- STATE**: ☐ ON ☒ OFF
- Parameters**: A text input field with a scrollable list of parameters below it.
- Operation Speed**: A text input field with the value "15".
- Buttons: "Accept Configuration" and "Restart".

The status bar at the bottom indicates "Document Done".

Figure 8 Example of an HTML form for data entry.

3. After the completion of the data acceptance and storing processes, the operator returns to VRML mode, enabling the configuration of other components (figure 9).

Figure 9 Top-Down diagram of the operator's actions.

3.1.1. The use of the Java language as a tool to enhance the user interface

The Java⁴ (Thomas, 1996) language allows writing small applications (applets) which are executed locally by the browser (as far as the Java specifications are supported on it).

The main advantages of using Java are its object orientation, which makes it strongly modular, and its multi-platform characteristic: one same Java application can be executed on any machine independently of its platform, because is the browser that translates the pseudo-code files that arrive via HTTP to the machine's specific code.

In addition, Java has an open API (*application programming interface*) (Yellin, 1996), which easily permits the enhancement of its basic possibilities.

Focusing on the system under discussion, the use of the Java language can be very interesting to solve issues related to the enhancement and increase of functionality of the user's interface.

As examples of this, Java will enable the implementation of an on-line help system embedded on the pages consisting on small mouse sensitive applets, or the integration of some multimedia gadgets.

3.1.2. Additional modules to the management and operation unit

Coexisting with the WEB browser, the operator needs some additional tools. On such a system two other modules are quite important: the alarm monitoring module and the videoconference client module

The function of the alarm monitoring module is to receive the alarm messages generated on the industrial unit's server whenever abnormal or error situations occur on the industrial process.

The videoconference client module will receive and display the video signals generated by the monitoring cameras, if installed on the industrial unit.

3.2. The production unit: A WEB Server based system

At the production unit side the interface to the Internet will be implemented through a server (figure 10). This server, that we'll call WEB server, will provide access to text files containing HTML and VRML scripts. Additionally the server will be used to store specific applications required to manage the processing of data received from the configuration forms, and other applications such as monitoring and communication tasks. Some of those applications are CGI⁵ applications, which are launched by the HTTP service.

Coexisting with the services and associated applications, this machine will be the host for a database. The function of this database is the storage and management of the configuration data of the components of the production unit. On the proposed implementation this database will be based on a SQL Server (ANSI X3.135, 1989), for increased functionality and modularity of the system.

To enable the coexistence of all those services, the operating system to be used on this server must be a multitask operating system, with integrated network support.

The proposed solution is based on *Windows NT Server* (Solomon, 1996), which includes the entire network functionalities required embedded. Furthermore its version 4.0 includes also the IIS⁶ which includes an HTTP server with support for CGI's, direct access to databases and

⁴ Java is a registered trademark of *Sun Microsystems*

⁵ CGI: Common Gateway Interface

⁶ IIS: *Internet Information Server*, from *Microsoft Corporation*

an API called ISAPI⁷, that allows to write applications for the WEB service, with some advantage over CGI applications as we'll discuss later.

Using the *Microsoft SQL Server for Windows NT* the connectivity and access to the database problem is also solved.

On this machine there will be also a service to process and transmit/receive the control data from/to the central control unit, and also a service to handle the feedback video signals, including its transmission to all the connected management and operation units.

Figure 10 The server of the production unit.

3.2.1 The HTTP service on the WEB Server

This service has the responsibility to deliver text files to the clients, some of which containing HTML pages and data entry forms and others containing VRML scripts.

This service can also launch small applications at client's request. These applications are called CGI applications, since they follow the CGI norms. In the specific case of the adopted server (MS IIS), is also possible to have another type of applications: ISAPI's which are DLLs⁸ loaded by the HTTP service itself, having some advantages over CGI applications, as we'll discuss later.

Figure 11 illustrates data flow on an HTTP server, emphasising its request/response orientation.

⁷ ISAPI: Internet Server Application Programming Interface

⁸ DLL: Dynamic Link Library

Figure 11 HTTP server functional diagram.

3.2.2 Data received from the management and operation units: acceptance and validation

The processes involved in data acceptance and validation are executed whenever a form containing configuration data is submitted to the HTTP server.

The main actions taken after data reception are the following:

1. Testing data for validation: When a data set is submitted to the server, sometimes it isn't considered as a valid set, because its originating form fields weren't correctly filled, or just because that data combination is not valid on the current system's state. (some actions over an element may be dependent of the state of other elements)

When such situation occurs, the server must generate an HTML page containing explicitly the error description and the situation which generated it.

This validation process is done by a CGI application (or an ISA) launched by the server after the form reception process. As result a response page will be generated and sent to the user.

2. Database updating: When a valid set of data is received on the server, the database will be updated with these data (on the table corresponding to the element which owns the data).

On a similar way, this is also done using a CGI or ISA, capable of receiving the data set as input parameters, and write it to the correct database table.

This application will interact with the database engine (the SQL Server on our implementation) using the ODBC (Open DataBase Connectivity) from *Microsoft* (figure 12).

Figure 12 Data Base interaction using ODBC

Figure 13 illustrates the data validation and acceptance processes.

Figure 13 Flow diagram of the data validation and update process.

3.2.3 The Database

3.2.3.1 DATABASE ARCHITECTURE

The database will be used to store the actual status of all the configurable components of the industrial processes. It is composed by several tables each one representing one type of component, and include fields to represent all the element's possible states and properties.

The several tables should contain relationships between them, since there will be several cases where a change on a table's field implies the change of other fields from other tables.

3.2.3.2 THE SQL SERVER

The use of SQL Server provides to the system the power and efficiency of a database engine, with high performance and giving the possibility of distributing the stored data to its clients.

This functionality can bring great benefits to the system, essentially in terms of performance and flexibility. By enabling the independence between the database and the other services on the server, it will be possible to minimize the overhead associated with the database management, because the database may be implemented on a dedicated machine if the number of accesses per time unit is considered to be high (figure 14).

Figure 14 Data base distribution using the *SQL Server*

3.2.4. CGI Applications and ISAPIs

The WEB server to be used should be able to execute CGI applications and ISAPIs. In the described system will be used *Microsoft IIS* and the server applications will be ISAs⁹. The advantages of this type of WEB server applications over CGI applications are, basically, the following:

- Good language support for writing ISAs: it's possible to write an ISA using C++ language. The most recent C++ compiler - *Microsoft Visual C++ 4.2*, has even built in classes to easily use the ISAPI (figure 15).

Figure 15 ISA related classes of the *Microsoft Foundation Classes*

- ISAs run on the same address space as the HTTP server, and they can access all its resources (what does not happen with CGI applications).
- ISAs have less overhead than CGI applications, since new processes don't need to be created, as with CGIs. This doesn't waste processing time necessary to do interprocess communication, since everything runs on the same process.

⁹ ISA: Internet Server Application

A WEB client calls an ISA using the HTTP server the same way as with CGI applications. The following example shows how this invocation is performed:

```
CGI:
http://plant.corporation.com/CGIexample.exe?Param1
?Param2?.....?ParamN

ISA:
http://plant.corporation.com/ISAexample.dll?Param1
?Param2?.....?ParamN
```

3.2.5. Complementary services to HTTP Service

3.2.5.1 MONITORING DATA ACQUISITION SERVICE

The monitoring data acquisition service receives the data from the monitor devices placed over the industrial process and sends it to the connected management and operation units. This service can easily be divided into two sub-services: one responsible for the acquisition, treatment and transmission of video signals from the cameras placed on the line, and other service responsible for processing the alarms generated when an abnormal event occurs on the industrial process.

Analysing these two cases separately, we have:

Video acquisition service: This service receives the coded video signals from the hardware acquisition boards installed on the server (or on other machines placed over the LAN). and sends this data to the connected management and operation units.

Two different implementation approaches rise when it comes to the implementation strategy of this service:

On the first one (the most elegant one), the service, when requested by a client on the network, starts sending back the acquired video coded using a stream format. Under these conditions, whenever a client requests the visualisation of the data from one of the cameras, using its video-conferencing client style software module, the service will start the transmission of real time data to the client¹⁰.

On the second case, the service stores data using stream format multimedia files. This format must allow the direct display of the video image using the client's browser, with real time download and visualisation from the network.

Using this strategy, one file will be created and used for each of the cameras installed on the industrial site, and with a cyclical re-write mechanism each of the files are substituted with more recent ones when their sizes reach a limit value. This process doesn't allow really real-time visualisation but, on the other side, it allows the inclusion of video frames within HTML pages, enabling the visualisation of a video window in the operator's browser client area.

As a tool to implement this solution, *Microsoft* already provides a native stream format and a SDK to develop applications for it. This format is called *Active Movie* and enables the encapsulation of most of the common multimedia formats in a single stream format¹¹. This is a data type independent of its contents and allows the transport over several network physical topologies and logical protocols. It supports the video formats .AVI, QuickTime¹² and

¹⁰ This process will be similar to several video-conferencing systems used on the Internet on a generalized way, as for instance, the system CU-See Me

¹¹ .asf format: *ActiveMovie Stream Format*

¹² *QuickTime* is a registered trademark of *Apple Computer*

MPEG, repackaging them to optimise its transport. With the .asf format, data reproduction will start immediately after the first packet of bits arrival to the client.

To correctly use this format it's necessary to install the correct plug-in for the browser in use on the management and operation units.

Alarm processing service: This service keeps listening to one of the serial ports of the server, where are supposed to arrive the alarm signals from the central control unit (figure 16).

When an alarm signal is received, a process is initialised that updates an alarm log file and also broadcasts the alarm message to all the connected clients.

The clients, which should have their own alarm monitors active anytime, will receive a message describing the error situation and the state of the element who produced it. This enables an immediate reaction to the alarm event.

Figure 16 The alarm processing service

3.2.5.2. COMMAND GENERATION AND CENTRAL CONTROL UNIT (CCU) INTERFACE SERVICE

This process interacts with the CCU. It is a Windows NT service that runs on the server and will be always listening to changes on the configuration's database.

Whenever a database modification is performed by one of the WEB server applications (ISAs), a message will be sent to this service to notify that change. This message will have a specific format, containing a field with the element's identification affected by the modification. With this information, the service will request a database query to the SQL Server to read the corresponding element's information. Using that data, the system will construct a frame and will transmit it to the CCU, using a server's RS-232 communication port.

Figure 17 shows the processing flow of this service:

Figure 17 Flow diagram of the command generation and UCC communication service.

4. THE CENTRAL CONTROL UNIT AND THE ELECTRO-MECHANICAL ELEMENTS

The central control unit (CCU) will be a block based on a rack of communication ports (RS-232) and a message delivery module (figure 18).

On the proposed implementation two RS-232 ports will be present to communicate with the industrial unit’s server, and a number of other ports to communicate with each of the controllable electro-mechanical elements of the production site (as many as the number of these elements).

The first two ports are used, respectively, to receive the control frames produced on the server and to transmit alarms generated by the industrial process to the server.

Figure 18 The central control unit

As a straight forward approach to the CCU's implementation it will be used an off-the-shelf PC equipped with a rack of serial ports, that will run a service dedicated to distribute the arriving control frames to the appropriate communication ports and transmitting all the received alarm signals to the alarms port. Such a system is easily implemented over a machine running the MS-DOS operating system.

A more elaborated solution will be the implementation of the CCU with dedicated hardware, by building a black box encapsulating all the unit's functionality.

5. NETWORK CONSIDERATIONS

As discussed above, a system like this uses the protocols and tools usually used on the Internet (browsers, WEB servers, the TCP/IP stack, etc). This means that such a system could be used in pure Internet environment, being possible to have several management and operation units distributed all over the world. Although this can be conceivable, this scenario presents some limitations: security (Karanjit, 1996) becomes a very important matter (new mechanisms should be taken in consideration), and performance, because bandwidth is limited and uncertain. This can cause us some new problems when it comes to real-time processing events on the system. The proposed implementation takes these constrain into account, and, because of that all the system features remain functional under Internet conditions, although system's responsiveness may be affected.

These disadvantages may be minimized if an Intranet topology is used: the support network is kept isolated from the rest of the Internet. In this scenario it will be easier to manage the available bandwidth, and restrict the security issues.

6. CONCLUSION

Current information and communications technologies enable the development of new applications that may contribute to substantial cost reductions: staff costs through the possibility to remotely control processes and investment costs through the capability to increase resource's sharing.

Teleproductics is a potential example of this, because it will enable remote access to industrial processes.

This paper discussed how current technologies could be used to implement a Teleproductics system based on the Internet. The system under development is focused to implement virtual laboratories for teaching mechanical engineering (Minoli, 1996).

7. REFERENCES

- ANSI X3.135 (1989) Database Language SQL. American National Standards Institute, ANSI X3.135 (ISO 9075)
- December, John and Ginsburg, Mark (1995) HTML and CGI unleashed. SAMS.net Publishing corp, Indianapolis.
- Elbert, Bruce and Martyana, Bobby (1994) Client/Server computing: Architecture, Applications and Distributed Systems Management. Artech House, Boston-London.
- Gesling, James; Yellin, Frank and the Java Team (1996) The JAVA application programming interface. Addison-Wesley.
- Minoli, Daniel (1996) Distance Learning Technology and Applications. Artech House, Boston-London.
- Pesce, Mark (1995) VRML: Browsing and building cyberspace. New Riders, Indianapolis.
- Siyani, Karanjit and Hare, Chris (1996) Internet Firewalls and Network Security, 2nd Ed. New Riders Publishing.
- Solomon (1996) Microsoft SQL Server 6.5 Unleashed, 2nd Ed. SAMS Publishing.
- Thomas, Michael D. (1996) JAVA Programming for the Internet: A guide to creating dynamic, interactive Internet applications. Research Triangle Park, Ventana.
- Wildev, Floyd (1993) A Guide to the TCP/IP protocol suite. Artech House, Boston-London.

8. BIOGRAPHY

João Paulo N. Firmeza, is currently working on his MCs degree, at the Electronics and Telecommunications Department, of the University of Aveiro. His work for the last two years has been related to telematics, with special focus on teleproductics, covering research, and development of a teleproductics prototype system. This work is being done with the collaboration of the telecommunications systems and services section of INESC and supported by JNICT's (*Junta Nacional de Investigação Científica e Tecnológica*) Praxis XXI program.

As an electronics and telecommunications engineering student on Aveiro's university, three years ago, he developed a Windows based ISDN videophone, and since then his work has been focused on computer based telematic systems.

Prof. Fernando Ramos, holds a PhD on Telecommunications from the University of Aveiro, Portugal, where he is currently Professor and head of the Department of Electronics and Telecommunications. He is also leading an R&D group at INESC in the area of telecommunications systems and service. His current research interests include open distance learning and telesurveillance systems based on ISDN, GSM and VSAT.