The TELEMEC project, developed by the Grupo de Automática y Robótica of the Pontificia Universidad Javeriana - Cali, included the design and development of a set of mechanical, electrical, electronic pieces and software modules that allows the construction of four different physical systems: the electrical elevator, the conveyor belt with multiple stations, the servo mechanism and the simple pendulum. These mechanisms constitute scale models of plants of real systems that can be used in different areas and courses of several engineering academic programs. This paper is focused specifically in the description, design and development of the software tool implemented for the management, control and use of TELEMEC system as a remote laboratory operated via Internet.

1 Introduction

The TELEMEC Project [1] arises from the desire of the members of the Grupo de Automática y Robótica of the Pontificia Universidad Javeriana - Cali to provide a Hardware/Software platform, so that it allows engineering students to make remote experiments with mechatronic systems without incurring in additional costs for building the plants; thus becoming a remote laboratory [2-3].

At the present time, on the Web there are multiple platforms to make experiments via Internet (like those provided by The Remote Laboratory of the Technological Institute of Stevens, see URL: http://dynamics.soe.stevens-tech.edu) or to manipulate objects remotely by guiding a robot (see URL: http://www.gse.uci.edu/mriel/webtour2/2/future1.html). TELEMEC goal is to provide a platform to make control experiments remotely. Here the student can interact with a remote system by sending to it the control parameters, and later on, he/she receives feedback by means of a series of graphics showing the controlled and sensed variables and videos captured by a couple of cameras located near the plant. These graphics and videos are displayed in a synchronous way: the state of the variables and its corresponding effect in the plant are observed at the same time. This characteristic becomes the main advantage of the system implemented in this project. Additionally, the master user can share their experiments with all the connected student users [1].

The software tool [4] offers the interface with the end user, and allows the system management. It was designed following the UML [5] notation and implemented under a distributed object environment with aid of the CORBA standard [6]. This tool should be operating system independent and allowing make experiments remotely via Internet; in addition, the user feedback should be as friendly as possible.

The next section will describe the TELEMEC components and later sections will deal with the design and development of the implemented software tool.

![Figure 1. General components of the system](image-url)
2 General System Architecture

The general system components are shown in Figure 1 and are explained in the following.

Plants

The idea is to provide a set of pieces that can be assembled in order to build typical plants that emulate basic industrial processes without having to think every time about how they can be designed from a mechanical point of view. The chosen plants are: the DC servomechanism for position and speed control, that allows to study the essential mechanism of electromechanical machines; the conveyor belt with multiple stations and the electrical elevator of several floors, in which the concepts apprehended in the first model are applied; and finally, the simple one degree-of-freedom pendulum, in which the speed control is fundamental.

PC1 (RT Linux)

It is a PC running the Linux operating system with RT Linux patches applied. It supports the control operations of the physical plant and hosts a distributed component of the software tool as will be explained later. To establish communication with the physical plant, PC1 has a data acquisition card (developed within the project [1]) connected to its parallel port. Its purpose is to provide the interchange of analog and digital data without putting any extra boards inside the computer. Besides that, additional electronic systems are required to provide the sensory circuits, conditioning the signals and bring the necessary power to the plant motor.

Video Cameras

An essential feature of the system philosophy is that the students can observe what happens with the plant while the experiment is running. Hence, two Axis 2100 digital cameras [7] are located near the plants; each one takes 10 pictures per second with a 320 x 240 dpi resolution. The great advantage of using those cameras is that they are stand-alone and have an embedded Web server, easing the development of video drivers and its mobility across the working space.

PC2 Server

This equipment is another PC, which is in charge of the system administration functions. It has access to the Internet by means of a University router and provides the necessary components for accessing and using the system. It also hosts the system database, some distributed components to support system operation and the Web server for system administration.

PC Clients

TELEMEC users can use any kind of computer provided that they have at least 64 MB of RAM and 10 MB of free disk space. The client application was implemented in Java [8] and therefore it is independent of the hardware platform and the operating system.

3 Distributed Object Architecture

Analyzing the conditions of use and the required infrastructure to provide and implement the TELEMEC system, it was found that the following characteristics endorse the implementation of a distributed object environment [9] with aid of the CORBA standard:

The data are distributed: The system data resides mainly in two nodes: PC1 with the controller data and PC2 with the images captured by the cameras and the database tables (administration, users, etc).
The processes are distributed: Several nodes are involved in the system and each one is in charge of specific tasks. For example, PC1 is in charge of the control tasks, PC2 of the image processing, administration and user request flow, and the end user computer is in charge of receiving user requests and visualizing the experiment results.

The users are distributed: TELEMEC users could be in any place of the world and can access the TELEMEC server by establishing a connection via Internet. Additionally, if the user log in as a master user, he/she could share his/her experiment results with other users, connected as student users.

Given the aforementioned reasons, well-defined Business Objects [10] were developed by means of IDL interfaces [6]. Thus, making future developments and/or upgrades of the system should be less complex by following the guidelines of the CORBA standard. The system components were distributed in the different nodes as shown in Figure 2.

![Deployment Diagram](image)

**Figure 2.** Deployment Diagram

From a functional viewpoint, the system can be partitioned as follows: An Administrative Application, a Server Application and a Client Application. They will be described in the following paragraphs.

**Administrative Application**

This application (see Figure 3) was implemented in Java and uses a Tomcat server [11] to generate dynamic HTML pages. Managing the system via the Web allows the system administrator to make his/her duties from any place as long as his/her computer is connected to PC1 by using a standard Web navigator. As shown in the Deployment Diagram, the navigator hosted in the PC client access the administrative application with the aid of the Tomcat server, which interacts with a PostgreSQL [12] database that stores the system configuration and the user accounts via a JDBC [13] connection.

Through the application it is possible to: Add plants to the system as well as setup the experiments that can be made and the parameters to use for each one; configure the available video cameras (IP address, port, etc.); generate valid plant usage schedules; manage user accounts, etc.

**Server Application**

The Server Application is in charge of synchronizing each one of the components that constitute the architecture. It is accessed by the application client through the IIOP protocol [6] for communicating
remote objects and it uses the Tnameserv (an implementation of the COSNamingService service [14] provided in the JDK1.3 [8]). The application makes use of video drivers, implemented with sockets connected to the Web server of each camera. To transfer the experiment result files from the server to the client, the application uses the Linux ProFTP server. Finally, in order to start the experiments in the physical plants and to get the sensed variables, it uses the distributed object PlantControl [4], hosted in PC1.

**Front End Application (Client Application)**

Finally, we have the Client Application (see Figure 4), through which the end user interacts with the system. Via this application, the student, master or singular user can be connected to the TELEMEC system. The application allows sending the controller parameters, select the active video cameras, download the experiment results and visualize them with the help of a VCR-like Control Panel (to advance, rewind, slow down, etc.), synchronizing the graphic data with the videos. Also, the experiments can be stored in the local disk in order to play them offline.

![Figure 3. Administrative Application](image1.png)

![Figure 4. Client Application](image2.png)

**4 Results**

**Tools and Implementation Language**

Every component of the application was implemented in Java, since this language offers many APIs for video and image handling (JMF [15]), generation of dynamic Web pages (Servlets and JSPs [16]), generation of stand-alone applications (Swing [17]) and implements part of the CORBA standard. In addition, in order to lower the implementation costs, the text editors (vi, xemacs), the DBMS (PostgreSQL), the FTP server (ProFTP), the operating system (Linux), etc., are free software. Furthermore, Linux distributions offer various tools to support the software development: the Concurrent Version control System (cvs), maintenance of projects (make), search of patterns and regular expressions (grep), etc.

**Some experiments and space-temporary considerations**

In the Elevator plant it is possible to make two different practices: A position test, which consists in moving the mobile mechanism to the station(s) set by the user; or, a speed test, that consists of moving the mobile until the upper end of the rod and then to the lower end at a constant speed, set by the user.

Table 1 shows some experiment results as well as the time required to download them (using a Dial-up connection with a 56Kbps Modem). The first column determines the type of experiment (Position or Speed); the second one, the number of video cameras used; the third one, the duration of the experiment (time the mobile is moving); next is the time the application has to wait for the experiment to end and the time required to transfer the file to the client. The last column shows the size of the generated files.
Table 1. Duration and size of some experiments.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Cameras</th>
<th>Duration (sec)</th>
<th>Time Elapsed</th>
<th>FTP</th>
<th>File (Kb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pos (P=3 – 2)</td>
<td>1</td>
<td>2.95</td>
<td>21''</td>
<td>5'55''</td>
<td>430.8</td>
</tr>
<tr>
<td>Vel (v=0.2)</td>
<td>1</td>
<td>6.39</td>
<td>22''</td>
<td>5'39''</td>
<td>531.4</td>
</tr>
<tr>
<td>Pos (P=3 – 2 - 2.5)</td>
<td>2</td>
<td>4.43</td>
<td>2'30''</td>
<td>15'20''</td>
<td>1063</td>
</tr>
<tr>
<td>Vel (v=0.9)</td>
<td>2</td>
<td>1.97</td>
<td>27''</td>
<td>4'02''</td>
<td>384.3</td>
</tr>
</tbody>
</table>

5 Conclusions

Given the distributed nature of the processes, the data and the users in TELEMEC, using the CORBA standard and tools like JDK1.3, which implements part of it, it was possible to develop the application as it was specified from the beginning in the design derived from the classes diagram. Therefore, there was no need to take into account the software layer necessary to implement communication between the components (i.e. sockets, files or any other mechanisms to implement RPCs [10]), getting well defined and easily re-usable Business Objects [10].

Since TELEMEC was implemented as a basic framework, by adding another experiments aside from the mechatronic plants defined originally the system can be extended, bringing to the users a wider range of practices. Furthermore, it is possible to use the system with non-mechatronic systems by following the standard controller-server application exchange format as defined in [4].

6 References

12 PostgreSQL, Inc., PostgreSQL. IN URL: http://www.postgresql.org/
17 Sun Microsystems, Inc., Project Swing (JavaTM Foundation Classes) Software. IN URL: http://java.sun.com/products/jdk/1.2/docs/guide/swing/