PEARL: A Generic Architecture for Live Experiments in a Remote Laboratory

Tilman Schäfer, Jean Marc Seigneur, Alexis Donelly

Distributed Systems Group, Trinity College Dublin

Abstract. The convenience and widened access offered by computer based learning is already established. However, this pedagogy can be enhanced by the opportunity for students to plan and perform "live" experiments over the Internet (as opposed to simulations). This paper describes the PEARL generic architecture for remote experiments and our initial experience implementing computer vision experiments using it. In developing the PEARL architecture, we have tried to accommodate a wide range of (low-level) equipment interfaces, to accommodate the requirements of users with disabilities and to facilitate the deployment of new remote experiments. Another consideration in our work has been to support, as much as possible, the style of learning identified in Laurillard's Conversational Model [8].
Preliminary feedback indicates that the PEARL architecture facilitates the design and deployment of remote experiments, that such experiments can be embedded in virtual learning environments and that the resulting experiments are well liked by students. This feedback, from both usability and technical viewpoints, has allowed us to improve user interfaces and optimise performance of the system.

1 Introduction

Institutions offering e-learning courses rely heavily on the distribution of educational material with a high pedagogical value. Hence the delivery of course material and educational software such as simulation tools via the WWW is playing an important role in distance learning environments. While there is a need for pedagogically sound courseware [5], current virtual learning environments do not usually address the form of "trial and error" experience that is gained by hands on experiments in a laboratory. PEARL (Practical Experimentation by Accessible Remote Learning) is an EU funded project[1] that is aimed at enabling students to conduct live experiments remotely over the WWW. Such experiments can provide an authenticity and unpredictability that simulations or paper based descriptions can not provide. Live experiments, including those enabled by the PEARL architecture, provide improved access to this "hands on" educational experience, namely:

- access independent of time and location
- access to lab equipment as student numbers increase
- access to expensive equipment otherwise not available
- access for students with disabilities

Students with disabilities, who are rarely considered in the design of laboratory experiments, can now be included by careful design of the user interfaces, since access is no longer physical but now mediated by the internet or WWW. The generic character of the PEARL architecture means that much of this design work can be done once and then re-used in future experiments. Remote access to lab equipment can also play an important role for campus universities where, given increasing number of students and limited resources, access to laboratory facilities, including more expensive items, can be widened or sold on. In addition to the improved pedagogy of live experiments, the generic character of the PEARL architecture facilitates the deployment of (further) remote experiments.

In this paper we describe the generic PEARL architecture for accessible, remote experiments. We describe a prototype implementation of the PEARL system developed at Trinity College Dublin that is used in computer vision courses for visual inspection and digital image processing. We describe the design decisions made and some results from preliminary trials that have enabled us to refine our design and prototype. Similar remote labs are being developed by other consortium members of the PEARL project, including a

[1] funding for the work reported here was provided in part by the Commission of the European Community in project IST-1999-12550 and is gratefully acknowledged.

remote electronics lab for circuit design and microcontroller programming, implemented by the university of Porto [7], a wet chemistry module by the Open University in Milton Keynes, UK, and a biochemistry experiment centred around an electron microscope by the University of Dundee.

The remainder of this paper is structured as follows. In Section 2 we will first present the overall system architecture and describe the various components of our system. Section 3 evaluates our approach under pedagogical and technological aspects. Section 4 finally concludes and presents future work in this area.

2 System overview

A high level view of a typical PEARL system and its use is presented in Figure 1. Clients located at home or within a hosting institution can connect to the remote lab using a standard Java-enabled browser. Applets or Java applications are used to control the equipment and course related material is presented in form of HTML pages.

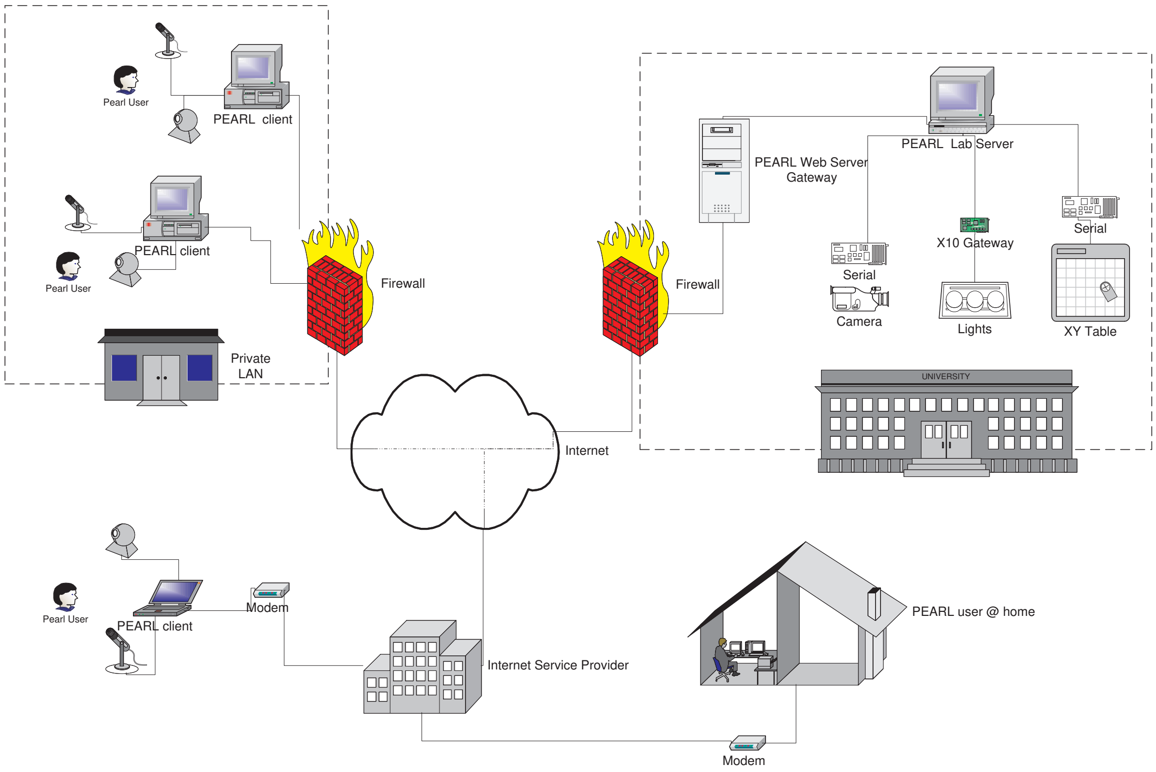

Fig. 1. A possible usage scenario of PEARL within the World Wide Web.

The PEARL system is based on a 3-tier client/server architecture, similar to the ones described in [2] and [6]. Client machines of PEARL users are separated from machines located in the lab. The latter machines (called Lab Servers) are directly connected to the experimental equipment and act as controllers. A gateway machine, acting as a web server, connects PEARL clients (running a Java enabled browser) to the lab servers. In our implementations a CORBA-based [12] connection is used between web server and lab server. This means that once a CORBA interface to the experimental equipment has been constructed or provided, the remainder of the PEARL architecture can be exploited with relative ease to develop a new experiment. Effectively, we have used CORBA to hide and encapsulate a diversity of proprietary interfaces. It is then a relatively straightforward matter to make remote method invocations from the web server to these interfaces (see Figure 2). Other information, such as laboratory scripts, further learning material, user interface descriptions and user preferences are stored on the web server. They can be composed and adapted dynamically on demand to accommodate user preferences using a mechanism similar to style sheets [9]. However, this aspect of the PEARL arhitecture is at early stage of development.

Aspects that have been taken into account while designing the system are the presence of firewalls and low bandwidth connections of users connected via modem. These issues will be discussed more thoroughly in section 2.3. The incorporation of PEARL experiments into collaboration tools such as CUWeb from CuSeeMe Networks. [4] has also been considered. In fact, the pedagogical value of remote experiments can be significantly improved by interactions between students or between students and a tutor. In this way, we hope to support the learning style described in Laurillard's Conversational Model [8]. In this pedagogical model, students develop their own knowledge, based on interactions ("conversations") with each other, with their tutor/demonstrator and with the experimental equipment. One PEARL experiment has already been integrated into a CuSeeMe based learning environment.

2.1 PEARL Web Server

We now describe in more detail the components of the PEARL web server as depicted in Figure 2. The PEARL web server (Jakarta Tomcat 3.2) acts as a gateway between clients and the lab equipment. It serves course material and applets to control the equipment. In the TCD experiment, all commands controlling the equipment are issued via HTTP. The HTTP requests are parsed and handled by servlets in the web server and translated into CORBA remote invocations on the lab server. Return values, such as status or error messages, are finally passed back to the client in form of simple HTML documents. Authentication and logging of user sessions is also performed by the web server: clients can only connect to the remote lab during scheduled lab sessions and all user actions are logged in a database. This log can be used to diagnose faults or assess student performance. CORBA (Orbacus 4.0) has been our technology of choice for connecting the web server to the machine controlling the lab equipment. This is mainly because most driver software for the lab equipment comes in form of C/C++ code and CORBA allows a seamless integration of client and server programs written both in Java and C++.

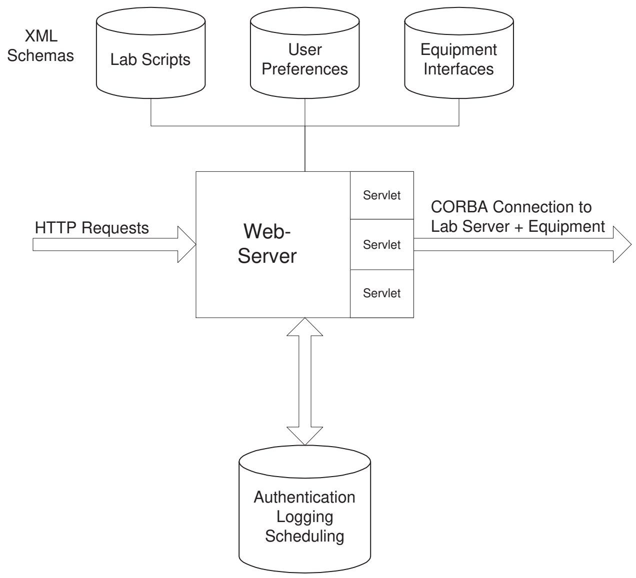

Fig. 2. Overview of the PEARL web server. Servlets parse incoming HTTP requests and forward them to the lab server in form of CORBA remote invocations.

A generic architecture is being elaborated to store and develop both learning material and descriptions of user interfaces to the experimental equipment (see Figure 2). An XML schema is being developed to describe the CORBA interfaces to equipment that can be used in the preparation of servlets in the back end of the web server. Another schema under development will describe the user interface the experiment equipment. Other XML schemas are being developed to describe user preferences and the laboratory script. These may be combined dynamically on demand, using a style-sheet like mechanism [9] to produce user interfaces tailored to specified user preferences. In this way it should be possible to accommodate the requirements of users with disabilities. While interfaces based on HTML pages are straightforward, if a little tedious, to make accessible [13], the same is not always true of Java based interfaces. See section 4 for further discussion.

2.2 Experiments

We now describe in more detail the visual inspection rig developed at Trinity College Dublin and the type of experiments it supports. Students have full control over a workbench consisting of a camera, XY-table and a number of independent light sources, as depicted in Figure 3.

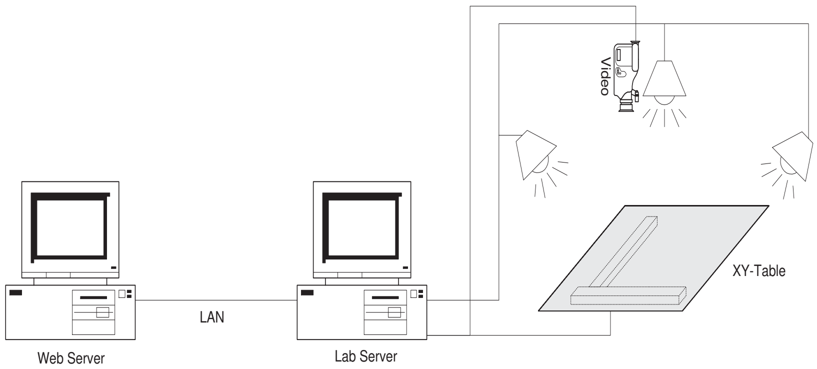

Fig. 3. A remote rig for visual inspection and digital image processing. The camera, XY-table and three independent light sources can be controlled remotely.

The XY-table holds various objects that students can move under the camera's field of view, so for example a printed circuit board and a colour calibration chart. The components on the printed circuit board can for example be inspected with respect to faults. The colour calibration chart contains three patches of the primary colours red, green and blue, which is used in a type of experiment outlined below. The camera can be adjusted in focus, zoom and pan/tilt settings.

In one type of experiment students calculate the camera's responsiveness of the red, green and blue channel by carrying out the following sequence of steps:

- switch on all light sources
- center the colour calibration chart under the camera
- adjust the camera's focus and zoom
- acquire an image
- repeatedly switch off one light source and acquire an image
- calculate the responsiveness of the red, green and blue channels for the different illumination levels

Evaluation and presentation of the acquired data is currently accomplished separately with the MATLAB [10] data analysis and visualisation tool. Other experiments include calculating the camera's focal length and use of a template-matching algorithm to locate components of a desired shape and size on a PC board.

2.3 Discussion

The infrastructure for conducting real-time experiments remotely has to meet a number of requirements in terms of utilising existing bandwidth and prioritising information streams, as described in [1]. To meet these requirements we considered alternative designs for the PEARL system. In this section we will briefly discuss these alternatives with respect to their advantages and disadvantages.

One alternative approach would be to have clients to establish a direct connection to the lab server, using for example Java RMI or CORBA. This would allow the direct streaming of video to the client without indirection via the web server, resulting in higher framerates and responsiveness. Due to Java security restrictions it is however difficult for applets to connect to a machine other than the one where the applet originated from. The presence of firewalls can furthermore prevent direct connections from being established. A workaround to this problem would be to modify the Java policy file on the client machine, although this requires a couple of configurational steps to be carried out by the client who cannot always be assumed to have the necessary computer configuration skills. The priority should be to make it as easy as possible to access the experiment as expert assistance may not always be available.

Streaming of video, however desirable, has in practical terms proven to be hindered by a number of administrative hurdles, both intra- and inter-institutional. It requires both institutions involved to open up specific ports that allow the video streaming to pass through their firewalls and routers. Initiation and coordination of the setting up of experiments can therefore lead to quite a substantial administrative overhead. Moreover, the type of experiments envisaged require high resolution, high definition images. Experiments with live video streaming using the Java Media architecture (JMF) to users with low bandwidth connections have resulted in very poor performance.

With these considerations in mind we adopted a HTTP–only strategy whereby all communication between the client and the server is accomplished by HTTP requests. Using this approach has the advantage that communication is not blocked by possible firewalls and the administrative overhead for the client is kept to a minimum. It also allows to conveniently log and schedule lab sessions using the Java session tracking mechanism. Another advantage is that we could easily provide different user interfaces for users with different impairments, so for example we could choose plain HTML to issue commands and display results. HTML is known to interface better with assistive technologies such as screenreaders and users can customise their display settings such as font size in the browser. On the downside, it requires that images captured by the video camera are first transferred to the web server and are then periodically downloaded by the client, resulting in a slow frame rate of approximately 1 frame/sec.

3 Evaluation

The above system has been used as part of an interim validation phase in order to demonstrate its usability and to further enhance the design of the user interfaces based on the feedback provided by students. The students were filmed during the trial and were interviewed afterwards. In this section we summarise our experience with the PEARL system under two headings:

- **Usability:** is the system usable? How effective and useful did students find the system?
- **Performance:** how did the system perform? What are the system requirements in terms of network traffic?

3.1 Usability

The attitude of student's towards working with the system was overwhelmingly positive. The majority (80%) reported that it was easy to learn how to use the system and 90% felt that it was designed with the student's need in mind. Major points of criticism included low response time (50%) and the lack of sufficient feedback.

The design of user interfaces plays a crucial part in the effectiveness of any Distance learning environment. It is through the user interface how students perceive the environment. Inadvertedly, the interface chosen only represents a small, selected part of the real environment. The challenge is to provide a sufficient sense of presence that enables students to distinguish remote from virtual and to ensure that students are not distracted from their learning [3]. Flaws in the design of the user interface can therefore lead to misinterpretation and inhibit the effectiveness of the learning environment.

For example, in a first implementation the position of the XY-table was indicated by a crosshair that could be positioned in a rectangular area in the controlling applet, as shown in Figure 5. The center of the crosshair represented the current position of the XY-table. A number of students misinterpreted the crosshair as being the position of the camera over the table and thus expected, when moving the table to the right, to change the camera's field of view accordingly to a further right position. In fact, moving the table to the right results in objects located on the left hand side of the table to appear in the camera's field of view. In a revised implementation the user is now presented with an outline of the shape of the XY-table that can be dragged and dropped to a new position. By doing so we hope to emphasise the fact that it is the actual XY-table that is controlled rather than the position of the camera.

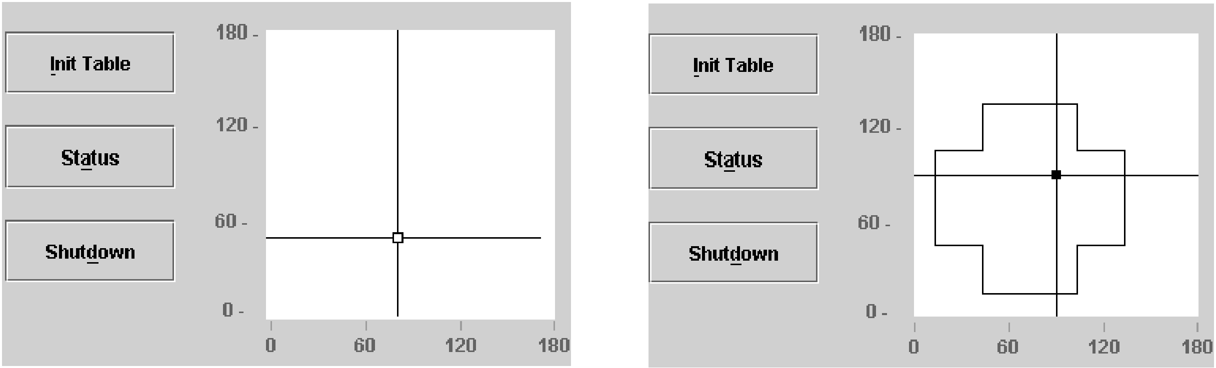

Fig. 4. Two different designs of an user interface controlling the XY-table. In the first design on the left hand side, students confused the position of the XY-table (as indicated by the crosshair) with the camera. In the redesigned version on the right hand side the crosshair (i.e. position of the camera) is fixed and the shape representing the XY-table can be dragged to a new position.

3.2 Performance

We monitored the system with respect to the generated network traffic during the experiments. The highest amount of network traffic is generated between the lab server and the PEARL web server, approximately 1.2 MBits/second. This is due to the video streaming of the camera which films the rig in the lab room. The additional traffic generated by the CORBA remote invocations when controlling the equipment was negligible.

Since a high bandwidth connection between the client and the PEARL server cannot be assumed, clients refresh an image which is periodically sampled from the streaming and served by the web server. In order to further decrease the bandwidth requirements, users have the possibility to switch the resolution and the JPEG compression rate of images, the idea being that in a first step they roughly locate the image portion of interest by means of small size images. In a second step they can then switch to higher resolution in order to get an adequate image for image processing. This reduces the generated traffic to approximately 120 kBits/second between the client of a PEARL user and the PEARL web server gateway.

4 Conclusions and future work

Supporting real-world experiments through the WWW has been the motivation for the work described in this paper. We have implemented a system that allows students to connect to a remote workbench through a standard, Java-enabled browser.

The pedagogical impact and educational value of the PEARL system have been a major concern of the consortium, but is at the same time the most difficult to assess. The remote nature of the experiments not only imposes a number of restrictions on their authenticity, for example, materials cannot be touched and chemicals cannot be smelt, it also makes it difficult for tutors to monitor the student's progress and to provide assistance should the need arise. However, we believe that PEARL can add sufficient value to Distance learning environments in that it allows the acquisition and evaluation of data. The data acquired is subject to stochastic variations and user actions, resulting in an unpredictability that simulations can not produce.

Providing sufficient feedback to the user has been identified to be a major requirement for the successful employment of remote experiments. Video provides the preferred feedback for users to validate their actions and constitutes the perceived responsiveness of the application. Initial time delays of approximately 4–5 seconds between issuing a command and viewing the response have therefore been the major cause of confusion. However, after an initial familiarisation phase most students have been able to get accustomed to the time delays. Within the foreseeable future we believe that the restrictions imposed by firewalls and low bandwidth can be overcome.

Accessibility for users with disabilities has been a focus of the PEARL project and is currently being explored in more detail. The applets controlling the lab equipment have been implemented using the Java

Swing classes that provide some accessibility features and could be interfaced with assistive technology such as screen readers. However, interoperability between Java applets and assistive technologies is still under development and also relies on the provider of the assistive technology. Other technologies such Mozilla's XUL [11] that appear to show promise in this area are not sufficiently mature. In the meantime we envisage to provide different user interfaces for users with different disabilities, so for example we could easily provide an (accessible) HTML [13] only version where commands and parameters are issued using forms. HTML pages can be customised according to the user's preferences in the browser, thus providing better accessibility to users with visual impairments.

Finally, the XML-based stuctures we are developing for the storage of laboratory scripts and information related to equipment interfaces should facilitate the development of future remote experiments. Rather than developing each experiment and equipment interfaces from scratch, course staff should eventually be able to fill in XML descriptions of user interfaces to equipment, CORBA interfaces to equipment and laboratory scripts. The server architecture should combine these dynamically, with specified user preferences, to produce experiment interfaces and lab scripts more efficiently.

We believe that the PEARL generic architecture provides a promising platform from which to develop future remote experiments.

References

1. S. Ch., H. A. Latchman, D. Gillet, and O. D. Crisalle. Requirements for Real-Time Experimentation over the Internet. In *International Conference on Engineering Education, ICEE '98*, 1998.
2. S. H. Chen, R. Chen, V. Ramakrishnan, S. Y. Hu, Y. Zhuang, C. C. Ko, and B. M. Chen. Development of remote laboratory experimentation through internet. In *Proceedings of the 1999 IEEE Hong Kong Symposium on Robotics and Control*, pages 765–760, July 1999.
3. M. Cooper. The challenge of practical work in an eUniversity – real, virtual and remote experiments. http://kmi.open.ac.uk/projects/pearl/publications/index.htm, 2000.
4. Cuseeme networks. http://www.cuseeme.com.
5. B. David, D. Stone, and M. Woodroffe. Experience with developing multimedia courseware for the World Wide Web: the need for better tools and clear pedagogy. *International Journal of Human-Computer Studies*, 1997.
6. H. Ewald and G. Page. Performing Experiments by Remote Control Using the Internet. *Global Journal of Engineering Education*, 4, 2001.
7. J. M. M. Ferreira, R. J. Costa, G. R. Alves, and M. Cooper. The PEARL Digital Electronics Lab: Full Access to the Workbench via the Web. In *Proceedings of the 13th Annual Conference on Innovations in Education - European Association for Education in Electrical and Information Engineering (EAEEIE02)*, 2002.
8. D. Laurillard. Rethinking university teaching. Routledge, London, 1993.
9. H. W. Lie and B. Bos. Cascading style sheets: Designing for the web. Addison-Wesley, 1999.
10. The MathWorks Inc., 21 Eliot St., South Natick, MA 01760.
11. Netscape/Mozilla. XUL. http://www.mozilla.org/xpfe/xptoolkit/xulintro.html.
12. OMG Corba Document. *CORBA Components - Volume I, II, III*, 1999.
13. WAI-WCAG. Web Content Accessibility Guidelines. http://www.w3.org/TR/WCAG10.