Int. J. Engng Ed. Vol. 19, No. 3, pp. 389–397, 2003
Printed in Great Britain.

0949-149X/91 $3.00+0.00
© 2003 TEMPUS Publications.

The Cockpit: An Effective Metaphor for Web-based Experimentation in Engineering Education*

D. GILLET, F. GEOFFROY, K. ZERAMDINI, A. V. NGUYEN, Y. REKIK and Y. PIGUET
Swiss Federal Institute of Technology in Lausanne, Switzerland. E-mail: denis.gillet@epfl.ch

Web-based experimentation is a key feature in the deployment of eLearning solutions for engineering education. At the Swiss Federal Institute of Technology in Lausanne (EPFL), Web-based experimentation is deployed thanks to a comprehensive environment called the Cockpit. This environment integrates all the components necessary to carry out hands-on practice in a flexible learning context. The Cockpit metaphor has shown to be an effective way to compensate for remoteness and to sustain active learning, interaction, collaboration and knowledge appropriation. Preliminary evaluations have shown both the effectiveness of the chosen solutions and the challenge of deploying a flexible learning approach in a traditional academic institution.

INTRODUCTION

ACADEMIC INSTITUTIONS are increasingly interested in developing Web-based experimentation environments that support virtual and/or remote experimentation [1, 2]. Virtual experimentation corresponds to simulation and remote experimentation refers to manipulation over the Internet of distant laboratory equipments. The Web-based experimentation paradigm is typically deployed by both traditional and open universities for the main purpose of enhancing the offering of didactic resources designed to strengthen hands-on practice in engineering education. The emergence of clusters and networks of universities which have committed to share experimentation facilities [3] becomes an important trend in sustaining the richness of learning resources that are provided to engineering students. In such a distributed laboratory scheme, each partner university can focus on maintaining and enhancing a few high-quality facilities in its own domains of competencies and excellence, which is an appropriate way to spread the best resources. The additional resources that may be required to improve an engineering curriculum can be accessed remotely to the other partners location, hence suppressing the necessity to own such resources locally as well as avoiding the related exploitation costs. Web-based experimentation also offers a tremendous opportunity to add flexibility in traditional curricula by providing students with versatile access to the learning material from both a time and a location perspective. We present the framework in which flexible education is emerging and the benefit that can be gained by integrating Web-based experimentation resources.

Carrying out Web-based experimentation is a matter of observing and acting on a virtual model or on a real equipment using convenient visualisation and control devices. Hence, a student enrolled in such activities can be seen as a pilot sitting in the cockpit of an exploration vehicle to complete a mission. In a Web-based experimentation framework, the given mission is typically a laboratory assignment and the Cockpit is a computer. To take advantage of this similarity, a general cockpit metaphor has been chosen at the Swiss Federal Institute of Technology in Lausanne (EPFL) to design the graphical user interface (GUI) of the environment dedicated to Web-based experimentation. This so-call Cockpit environment includes three main parts: the *experimentation console*, the *toolkit console* and the laboratory journal.

The experimentation console can be regarded as the interaction part that enables the actual realisation of experiments. The toolkit console provides the necessary functionalities to prepare experiments and analyse results. The laboratory journal constitutes the edition and collaboration part that facilitates reporting as well as knowledge integration and sharing. We describe the Cockpit metaphor applied to design the Web-based experimentation interface, as well as its components and their relevance. Then, the authoring tool provided to the people in charge of designing the Cockpit and its pedagogical content is presented. The way state-of-the-art Internet technologies have been deployed to implement the cockpit-like experimentation environment is also mentioned. Finally, the procedures implemented and the experience gained in deploying Web-based experimentation resources at EPFL in the framework of a traditional undergraduate curriculum are detailed along with the results of preliminary user-case

* Accepted 22 February 2003.

validations with students in seeking to introduce the tutoring scheme necessary to realistically sustain hands-on learning activities.

FLEXIBLE EDUCATION

The clear distinctions that could be made in the past between an open university and a traditional university are no longer reflecting the current status of affairs. Open universities are increasingly using the Web to supply on-line material that was previously delivered via postal services. Traditional universities are also progressively making more of their teaching resources available through the Internet [4]. This trend is leading towards the development of a hybrid teaching environment where the lectures and laboratory facilities of traditional curricula are combined with on-line lectures (or only sessions) and on-line accessible laboratory facilities (Fig. 1), providing proper accreditation or partnership agreements are established. Students enrolled in such hybrid educational contexts benefit of accessing pedagogical activities customised to their needs from the contents viewpoint as well as from the scheduling viewpoint. Such a scheme is an interesting form of what is defined as *flexible education*. Laboratory resources play a key role in this paradigm, and they can be effective contributors to flexible learning provided that *Web-based experimentation* alternatives are developed and deployed.

Web-based experimentation can include virtual and/or remote laboratory resources. A typical *virtual laboratory resource* is an interactive experiment that relies on a simulation engine and a virtual-reality user interface. The experimental system of engineering interest is simply simulated using software. Although using standalone simulation packages on a personal computer is a common and effective approach, there are three main reasons to provide instead Web-based access to the simulation resources. Web-based access:

- enables platform independence if conscientiously implemented;
- greatly facilitates the distribution and upgrade of the didactic resources which are kept up-to-date on the server;
- enables simplified management of the software licences when using Web-compliant commercial products.

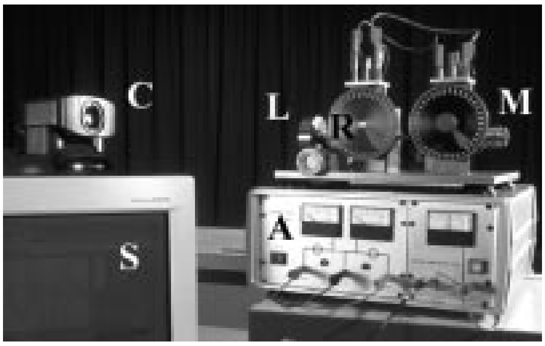


Fig. 1. An electrical drive that can be accessed either on campus in the laboratory premise or remotely using a Web browser. (**C**: Web camera; **M**: DC motor; **L**: inertial load; **R**: rotating disk; **A**: power amplifier; **S**: real-time control system and Internet communication server.)

A typical *remote laboratory resource* is a physical experimental system that is equipped with the necessary facilities to enable Web-based monitoring, measurement, and manipulation. Efficient monitoring relies on the use of live video and augmented-reality tools. Although virtual instruments, such as software-representation of an oscilloscope, for example, are commonly introduced as user interfaces to operate real laboratory resources over the Internet, the current tendency is to still refer to such activities as *remote experimentation.* The *virtual experimentation* designation is reserved for activities involving only simulated equipment. Most of the educational applications of *Web-based experimentation* carefully integrate both virtual and remote experimentation.

In contrast to video-on-demand lectures, virtual laboratory resources constitute more versatile components that can be integrated in a wide variety of curricula. This is due to the fact that lectures are oriented towards specific-content delivery [5], while Web-based experimentation resources can be seen as rather generic components integrated in versatile environments which support both know-how and knowledge acquisition and reinforcement. In fact, a very wide range of topics can be illustrated using the same laboratory resource.

In a successful flexible education framework, a Web-based experimentation facility must be presented as an attractive alternative to working in direct contact with the real laboratory equipment. To meet this challenge, the user-interface must enable a sustainable level of interaction with the experiment, be attractive, intuitive, and of an overall high quality. In addition, the environment must provide collaborative work facilities to reproduce the teamwork typically implemented in a traditional laboratory framework. In fact, the quality of the interface, the availability of collaborative tools and the value added by the possibility of accessing the resources at any time and from anywhere must be perceived as adequate trade-off for the remoteness of the activity. These advantages provide a non-coercive motivation for students to participate and enrol in remote-laboratory activities.

ENVIRONMENT DEDICATED TO WEB-BASED EXPERIMENTATION

The EPFL is currently developing and evaluating a generic environment for Web-based experimentation [6]. The idea is to provide a flexible environment that promotes active learning [7], interaction, collaboration and knowledge appropriation through hands-on practice.

A comparative study has been carried out between automatic control, biomechanics and fluid mechanics to determine the most common features required to complete typical experimentation assignments by students enrolled in the third and fourth year of the engineering curricula at EPFL. Students have also been observed in real laboratory conditions, by pedagogues, to understand their needs and interaction modes. In addition to dramatically improving effectiveness and reducing the development time, this concerted approach has led to a generic solution that can easily be deployed for educational purposes in other engineering domains.

As mentioned in the introduction, the Cockpit environment that has resulted from the comparative study and the observations of the students contain all the components necessary to successfully complete laboratory assignments (Fig. 2). The three main parts of the Cockpit are the *Experimentation Console*, the *Toolkit Console* and the *Laboratory Journal*. In addition, a *Navigation Bar* is provided that integrates a clear statement of the assignment *objective* and the *status* of the task currently processed in order to sustain awareness. The navigation bar also gives access to a number of supplementary pieces of information that have to be integrated for completeness, including relevant *reminders* or *links* presenting the underlying theory, an experimental *protocol* which describes the step-by-step procedures necessary to perform the laboratory assignment, and finally, a *description* of the environment, including the *laboratory resources* (virtual or real) and the specific *Cockpit features* that are used in the experiment.

The *Experimentation Console* can be regarded as the interactive part of the environment. It enables the actual realisation of experiments. The interactions that can be sustained are mainly in the form of changes that the students can make to parameters or algorithms that effect the behaviours of the virtual model or of the real piece of equipment. The responses to the changes are displayed graphically in real time, recognising that the timeliness of the posting of signals is critical to provide a remotely located student with an adequate level of visual perception [8]. The experimentation console in use at EPFL for hands-on practice in automatic control is designed in a modular fashion that allows incorporating the latest developments in sustainable interaction systems [9], thus ensuring a level of quality-of-service that is consistent with the underlying pedagogical objectives. Two different versions of the experimentation console are available. The first one which provides the best performance from an interaction point of view is a standalone application built using LabVIEW. The second one is a Java applet developed to enable operations by students using computers on which they have no right to installed files (typically in computer rooms located on campus). A detailed comparison between these two implementation alternatives can be found in [8].

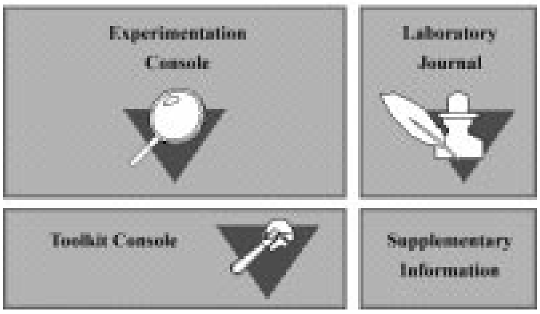


Fig. 2. Main components of the Cockpit environment.

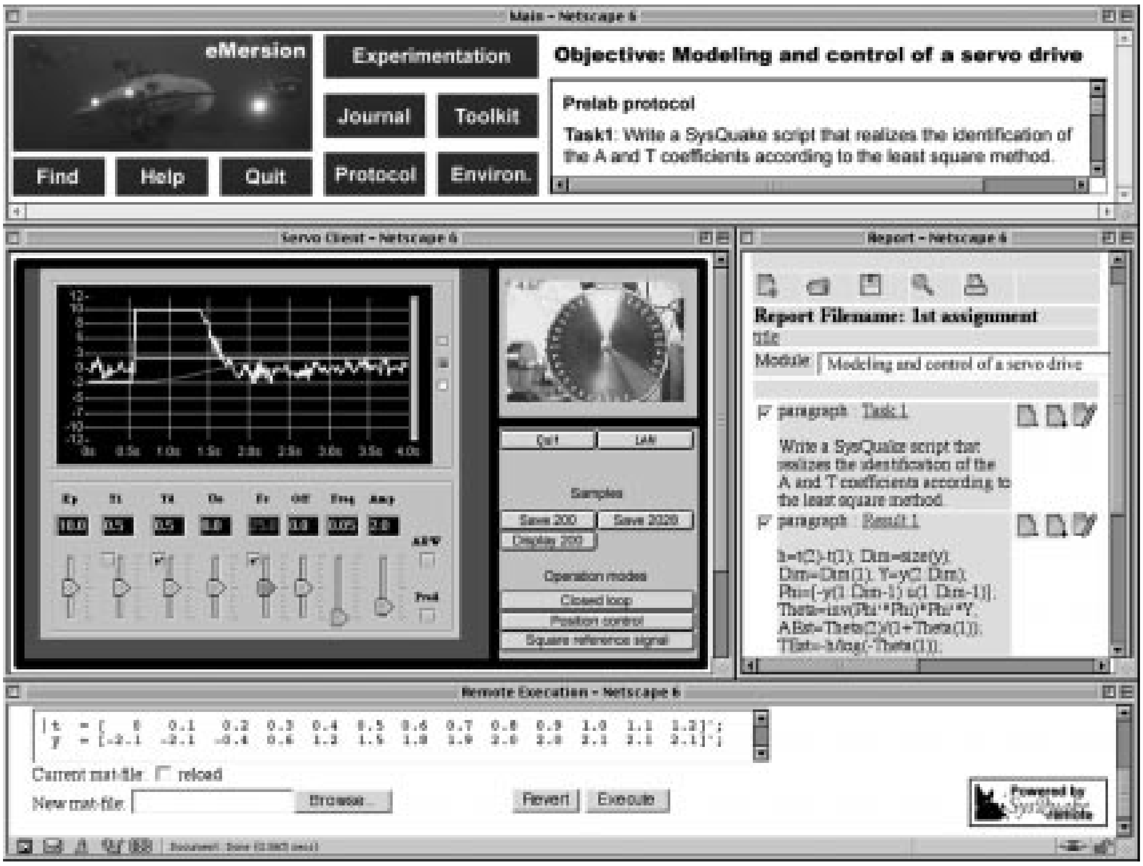
The *Toolkit Console* is dedicated to provide the students with tools to carry out interactive design and analysis activities related with the experiments. A classical solution used in automatic control for such a purpose is the standalone Matlab software package from Mathworks. For Web-based pre and/or post processing activities there are however additional requirements to fulfil that require alternative solutions.

- Among these requirements, one can mentioned: access to server processing should be made available via two kinds of user interfaces, with forms to lead the student to the solution while allowing to change parameters or experimental data, or by entering arbitrary code fragments to perform any kind of computation and creating any graphics.
- Students should have access to the software from any computer without the necessity to install additional software; a Web browser should be sufficient.
- Security should be enforced in both ways: to protect the server against attempts (voluntary or not) by the remote user to break into it, to crash it, or to consume excessive resources; and to protect the client against a server he or she does not necessarily trust.
- Integration with other Web resources such as the Cockpit should be as good as possible, to benefit from online documentation, merging between exercises and the software tools to solve them, to integrate dynamic figures, etc.
- The work overhead for the teachers and webmasters should be as low as possible; code reuse should be maximised.

Currently Matlab cannot fulfil all these requirements. While there exists a server for Web applications, named Matlab Web Server, it is virtually impossible to let the remote users type commands directly and create graphics in an easy way. The function *eval*, which evaluates arbitrary code, does not disable potentially dangerous functions such as those which give access to the file system or to the shell. WebMathematica (from Wolfram Research), based on the symbolic math software Mathematica, while it makes easier the integration of graphics, suffers from the same problem; in addition, giving access to the command-line interface of Mathematica to remote clients is explicitly prohibited by the license. For these reasons, the toolkit console of the Cockpit relies on SysQuake Remote, based on SysQuake (from Calerga) [10]. SysQuake has a mathematical language compatible with Matlab. It offers interactive graphics, i.e. the capability to update the graphics in real-time when the user manipulates them with the mouse to change the initial conditions of a simulation or the parameters of a controller design. SysQuake Remote is a module for the Apache Web server which interprets Matlab-compatible code embedded in HTML pages. Graphical commands create images stored temporarily on the server which are included in the document, in a transparent way for the author as well as for the remote user. On the client side, the browser receives plain HTML documents with static PNG images.

The *Laboratory Journal* is the supporting part of the Cockpit environment. It has been designed as an extended electronic version of the traditional notebook used by students to document their laboratory work. The laboratory journal lets students access their preparatory material, store and post-process their experimental results. It also permits documenting and reporting the observations, and it facilitates the key activities of knowledge integration and knowledge sharing. Since the laboratory assignments are usually completed by a team of students, the laboratory journal is designed as a global workspace in which students can work either on private or shared spaces. The design requires that the students first agree on the document structure and on tasks allocation. The students then make individual contributions to different parts of the document. Doing so, they assemble a set of notes, images and experimental results. Finally they work jointly to integrate the components to meet specific deliverables which are submitted as reports. During document co-authoring processes, students are aware of each other using mechanisms as annotation or message exchange. The edited notes or the laboratory journal can be made available, partially or entirely, to others (peers, tutors or educators) for feedback or evaluation purposes. The functionalities provided thanks to the laboratory journal can also be of interest for students who chose to carry out the experiment on campus in the laboratory premises.

The students use a personal login to access a specific list of assignments according to their curriculum. Once an assignment is chosen in the list, the corresponding Cockpit is launched. When using it, the students first consult the protocol (Fig. 3). Then, according to the assigned tasks and their personal work method, they exploit freely the different consoles. They especially work with the experimentation console and the



Objective: Modeling and control of a servo drive

Prelab protocol

Task1: Write a SysQuake script that realizes the identification of the A and T coefficients according to the least square method.

Report Filename: 1st assignment

title

Module: Modeling and control of a servo drive

☑ paragraph Task 1

Write a SysQuake script that realizes the identification of the A and T coefficients according to the least square method.

☑ paragraph Result 1

```
h=t(2)-t(1); Dim=size(y);
Dim=Dim(1); Y=y(2:Dim);
Phi=[-y(1:Dim-1) u(1:Dim-1)];
Theta=inv(Phi'*Phi)*Phi'*Y;
AEst=Theta(2)/(1+Theta(1));
TEst=-h/log(-Theta(1));
```

```
t  = [   0    0.1   0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.0  1.1  1.2]';
y  = [-2.1  -2.1  -0.4  0.6  1.2  1.5  1.8  1.9  2.0  2.0  2.1  2.1  2.1]';
```

Current mat-file: □ reload

New mat-file:

Fig. 3. Cockpit-like Web-based experimentation interface.

laboratory journal, which are the most interactive components of the Cockpit.

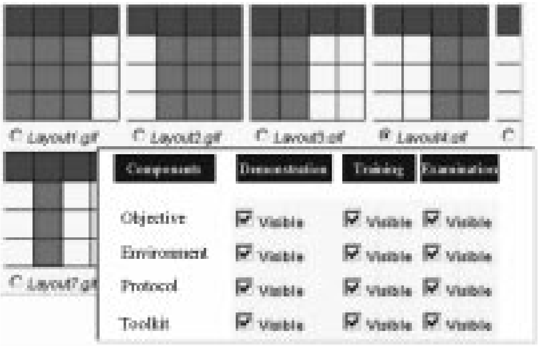
The Cockpits are dynamically generated using XML (Extended Markup Language) [11] and Java [12] technologies. These technologies ease the maintenance of the Web-based experimentation environment, provide a high level of component reusability [13] and lower the development cost. The experimentation console is implemented using a series of Java applets or PHP scripts [14], and the personalised laboratory journal is based on Java and JSP (Java Server Pages) technologies. One important functionality to notice is that the different parts of the Cockpit are connected so that materials such as files, experimental results and configurations, can be automatically transferred from one part to another. This versatility is intended to enable students to mimic as efficiently as possible, using Web-based technologies, the way in which they would carry out traditional hands-on laboratory experiments on campus.

Figure 3 shows the Cockpit designed as Web-based experimentation environment involving the control of a mechatronic system (more specifically, an electrical drive). The motion of the drive is visible in the central video frame of the screen shot.

THE COCKPIT AUTHORING TOOL

An authoring tool has been developed to ease the design of Cockpits. Before using this tool, the content of a dedicated Cockpit is prepared and structured as an XML document. This unique document follows the generic Document Type Definition (DTD) [11] that has been defined using the most common factor encountered in typical laboratory assignments. Then, the authoring tool is used to generate all the Cockpit components from the XML document. This tool consists of a series of three forms to be filled in a browser. First, the necessary resources (such as an XML document and URLs to the experimentation applets) are simply loaded using the form. Then, the choice of the information to be displayed for each Cockpit type is specified (foreground of Fig. 4). Three different Cockpit types have been introduced, which correspond to the different situations in which a same experiment can be introduced, namely demonstration, training or examination. Finally, the layout for each console of the Cockpit is chosen (background of Fig. 4) and all the corresponding components are automatically built (see Fig. 3 for the resulting Cockpit). The chosen layout corresponds to the one initially proposed by the designer. Students can then modify and save it to fit their preferences.

The same pedagogical content can be delivered in different contexts. For example, part of an XML document can be used to produce Cockpits either for demonstration or examination purposes. It is also possible to build from a unique XML document high quality PDF versions for printing.



○ Layout1.gif ○ Layout2.gif ○ Layout3.gif ● Layout4.gif ○

○ Layout7.gif

Components	Demonstration	Training	Examinatio
Objective	☑ Visible	☑ Visible	☑ Visible
Environment	☑ Visible	☑ Visible	☑ Visible
Protocol	☑ Visible	☑ Visible	☑ Visible
Toolkit	☑ Visible	☑ Visible	☑ Visible

Fig. 4. Authoring tool for the Cockpit generation.

To face with the reusability and structure constraints, the XML and associated technologies are the best suitable solutions.

PEDAGOGICAL SCENARIOS

In engineering education, the potential value-added that can be introduced in learning activities comes not only from the availability of resources for carrying out hands-on experimentation, but also from innovative pedagogical scenarios that promote the acquisition of know-how and from the availability of tutors for supporting students. These requirements, well recognised in traditional education paradigms, gain even more importance in distance learning and in flexible education because of the need to compensate for the students' remoteness. However, to date only a few institutions have integrated this human dimension in the deployment of their Web-based experimentation solutions [15, 16].

In the spirit of flexible learning, the students have the possibility to carry out an experiment at a time and from a location of their choosing, hence benefiting from a more effective cognitive experience. It is worthwhile to note that for the regular students at EPFL, i.e., those located directly on campus, there is no formal requirement that they use the resources from locations away from the campus. The students are allowed to carry out the experimental work directly on the campus premises; however, access to the facilities is restricted to a number of specific time slots. Consequently, the experimentation assignments and the support scheme have been designed to ensure that they are equally helpful to local and to remote students.

In teaching automatic control, the challenge has always been for students to link the highly mathematical analysis and design methods introduced during the lectures with the actual implementation of feedback loops. To facilitate this cognitive association process, an inverted pendulum and an electrical servo drive (Fig. 1) are introduced as physical experimentation setups. The former is used by the teacher for demonstration purposes, and the latter by the students for hands-on experimentation.

Assignments have been designed carefully into interactive sessions shorter than the typical 4-hour periods used for classical hands-on practice in the real laboratory premises. It is clearly more difficult for students to face a computer screen for several hours compared to working in a real laboratory facility. The assignments are also more focused in terms of the objectives, while being kept open in terms of methodologies. Care is taken to ensure that the proposed experimental protocol guides the students to learn one topic at a time, rather than being overwhelmed with a large number of concepts during a lengthy session.

As an example, one of the assignments that any student attending the automatic control course at EPFL must complete is the modelling of a mechatronics system. The pedagogical objective is to visualise and characterise the dynamic behaviour of an electrical drive by studying various typical responses in the time domain. The time required to achieve this experimentation is about 3 hours. During this period, which includes a combination of preparatory work, measurement and analysis stages, only a few and short on-line connection periods to the real laboratory resources of about 10 minutes each are necessary. The studies done during this session are limited to the analysis of the behaviour of the drive controlled in open loop. The goal is to validate the theoretical models obtained to describe the speed and position evolution. In addition to the theory reminder and the description of the experimental environment, the Cockpit composed for this Web-based experimentation assignment contains an experimentation protocol as supplementary information. The protocol is divided into two parts, namely, a *prelab* that does not require an access to the real laboratory resources, and the *labwork* itself.

Currently, no grades are given for the timely completion of the assignments during the term. The only constraint for the students is to successfully fill the *prelab* form because it poses technical questions that must be answered to gain permission to access the given Web-based laboratory resource necessary to carry out the *labwork*. This requirement has been introduced to ensure that students have the prior knowledge necessary to benefit from the experience, and to motivate them to do preparatory work on their own. A grade is given only at the end of the term, when every student has to complete a randomly selected test-assignment within a pre-specified time. A grade is then given based on the degree of completion of the deliverables accomplished by the student. The existence of a test assignment at the end of the semester is, in our experience, a strong motivating factor to induce the students to learn as much as possible during the term, and to develop a thorough understanding of all the subject matter as well as a methodology to implement automatic control solutions.

Web-based experimentation resources are also introduced for illustration purposes in the context of lectures on advanced control techniques. Students carry out a case study as a written exercise, and then have the opportunity to validate their results on the real remote experiment. Such an approach has proven to be effective in contributing to maintain among the students a high level of motivation for studying the subject matter.

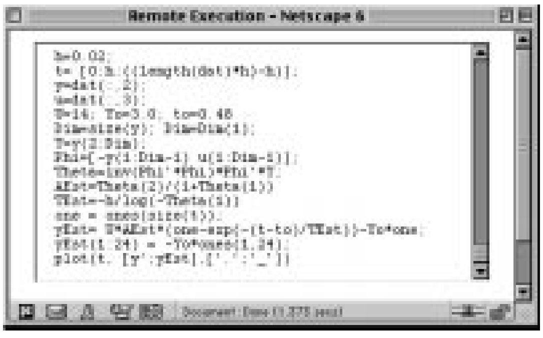
USER-CASE VALIDATION

Since October 2001, a group of 25 volunteers chosen among EPFL students has used the Cockpit environment to carry out laboratory assignments in automatic control. These volunteers had to realise three assignments related with the modelling and digital control of the servo drive (Fig. 1). The main motivations for the students to become volunteers have appeared to be the potential increase in support and the added flexibility to complete the laboratory assignments (especially the test) before the end of the term.

The progress of the experiment occurred as follows. Students were asked to answer some preliminary questions listed in the protocol. They answered them by editing notes or attaching documents to the laboratory journal. When this work was completed (see as example the required script displayed in Fig. 5), the students marked the laboratory journal visible to the tutor as a submitted *prelab*.

These answers were corrected and annotated by the tutor within the laboratory journal. Then, they were authorised to perform the practical part of the experiment assignment (*labwork*) using the experimentation console (Fig. 6).

The volunteers have been observed and interviewed by pedagogues to evaluate their reactions regarding the Web-based environment and the flexible learning approach. Although the environment has been easily integrated by students, their rather poor autonomy level has become an obstacle to really capitalise on the added learning flexibility. As a matter of fact, it has been noticed that most of the students asked for more synchronous feedback when they work alone on their experimental setup to figure out whether or not they are on the right track. This observation has



```
h=0.02;
t= [0:h:((length(dat)*h)-h)];
y=dat(:,2);
u=dat(:,3);
U=14; To=3.0; to=0.48
Dim=size(y); Dim=Dim(1);
Y=y(2:Dim);
Phi=[-y(1:Dim-1) u(1:Dim-1)];
Theta=inv(Phi'*Phi)*Phi'*Y;
AEst=Theta(2)/(1+Theta(1))
TEst=-h/log(-Theta(1))
one = ones(size(t));
yEst= U*AEst*(one-exp(-(t-to)/TEst))-To*one;
yEst(1:24) = -To*ones(1,24);
plot(t, [y';yEst],['.';'_'])
```

Fig. 5. SysQuake Remote script edited as a prelab task to carry out later during the labwork the actual model identification.

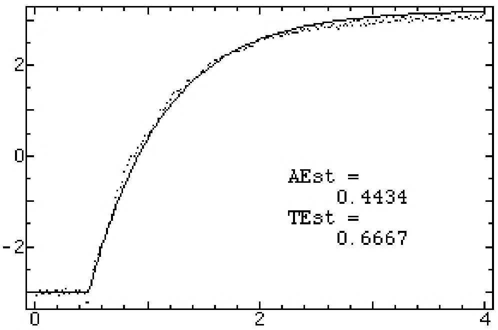


Fig. 6. Open-loop step response measured (dots) using the experimentation applet and fitted data (line) displayed using SysQuake Remote.

been the principal motivation to adapt the tutoring scheme implemented to support students involved in Web-based experimentation activities, as described in the next Section.

TUTORING SCHEMES

Tutors play three different roles in the context of Web-based experimentation assistance. They are involved in providing *kick-off support*, *on-demand support*, and *contractual feedback*.

Kick-off support is the necessary help that has to be offered on campus at the beginning of the Web-based experimentation program to introduce the students to the pedagogical objectives, the learning approach, the evaluation scheme, as well as to the Cockpit functionalities and usage. Such support is essential for ensuring the students' success in the learning experience. The students also receive advice on best practices and hints to avoid typical problems encountered in distance education, including work planing and media usage (such as an electronic forum, FAQ, or e-mail resources) according to their specific support needs.

On-demand support meets the need of the students for auto-evaluation. It as been observed during classical on-campus laboratory sessions that students solicit immediate feedback from their peers or from any closely-located teaching assistant at the completion of a specific experimental task. This also happens when they face a problem that can block their progress. These problems may be related to either the subject matter studied or the use of the experimentation equipment. In a Web-based experimentation context, additional support is needed to handle any question related with the Cockpit usage. The reaction time to on-demand support has to be close to synchronous in order to avoid a quick loss in the students motivation. To foster the development of autonomy and self-confidence, tutors are typically encouraged to provide methodology advice leading the students towards their own answers rather than providing direct and immediate answers. In such a way, the number of requests for on-demand support decreases from one assignment to the next, after an initial adaptation period. With a classical question-and-answer approach, the decrease in the level of requested support is marginal because students easily get used to receiving and using somebody else's views, causing only the illusion that actual learning is taking place.

To handle on-demand support related with the environment usage, a FAQ (Frequently Asked Questions) database and forum consultation should be the first line of response. Alternatively, telephone and electronic-chat communications are acceptable during working hours. When the tutor deems it necessary, he or she can take control of the student's Cockpit from a distant location using a service such as the Timbuktu [17] or VNC [18] remote display system.

Contractual feedback corresponds to the evaluation of a submitted prelab. This feedback is provided once selected parts of the dedicated laboratory journal corresponding to the assignments have been made accessible to the tutor by a group of students. The tutor can then add notes to the journal, or annotate some parts inside the journal, and if appropriate, activate a flag in the environment to enable the access of the team to the real laboratory resource.

A final and very important issue in implementing successful Web-based experimentation tutoring is to entirely decouple the formative support (assist in providing knowledge) from the normative support (assist in evaluating the extent of knowledge). The teaching assistants should act strictly as formative tutors, while the instructor is responsible for the normative activities, holding exclusively the responsibility for grading students. In our experience, this clear distinction increases the level of interaction between the students and the teaching assistants, by removing the students' fear of being poorly evaluated should they formulate their questions or remarks in a naive fashion.

CONCLUDING REMARKS

A laboratory has traditionally been an essential ingredient to the learning activities involved in engineering education. Laboratory work is recognised as an efficient approach for students to effectively assimilate knowledge and to develop the skills and intuition necessary for solving real-world problems. Laboratory experiences are also valuable for developing both the students' autonomy and their teamwork skills because the assignments are often a combination of individual preparatory activities followed by hands-on practice and analysis stages carried out in a team.

This paper proposes an integrated environment called the Cockpit dedicated to Web-based experimentation that includes remote manipulation facilities as well as collaboration features. It also describes the fashion in which Web-based experimentation resources have been deployed at EPFL. It is argued that this Web tool can be used to increase the flexibility of engineering curricula and to enhance knowledge acquisition and retention. For contextual purposes, the early experiences at EPFL in introducing Web-based experimentation resources are detailed and commented, identifying pitfalls and advantages.

The Cockpit deployment has been facilitated, thanks to an authoring tool developed to be used directly by the educators, without the intervention of IT specialists being required. The authoring tool is simple enough to allow the authors to concentrate on the pedagogical relevance of the integrated resources. Furthermore, by relying on structured resources and components, the authoring tool guarantees that the designed environment is satisfactory regarding the consistency, the functionalities, the ergonomics and the user-friendliness of the environment.

We also present a revised tutoring scheme that is compatible with the deployment of Web-based experimentation in a flexible learning framework.

The difficulty when implementing Web-based experimentation in a traditional university is twofold. First, the renewal of the education scheme is usually conducted concurrently with the introduction of new learning technologies, which can be disorienting for both the students and the educators. Second, online resources deployment has typically to cohabit with traditional lectures delivered at fixed schedules, which leaves less flexibility in time for the students to access Web-based learning material.

Further work is carried out to integrate synchronous feedback facilities through the Cockpit in order to sustain the acquisition of autonomy and auto-evaluation skills among students. This is a required step towards getting benefit from the introduction of flexible learning.

Acknowledgments—This work was supported by the board of the Swiss federal institutes of technology in the framework of its new learning technologies (NLT) program.

REFERENCES

1. D. Gillet, Ch. Salzmann, R. Longchamp, and D. Bonvin, Telepresence: an opportunity to develop practical experimentation in automatic control education, *European Control Conference*, Brussels, Belgium, July 1997.
2. T. F. Junge, and Ch. Schmid, Web-based remote experimentation using a laboratory-scale optical tracker, *American Control Conference*, Chicago, USA, June 28–30, 2000, pp. 2951–2954.
3. Ch. Schmid, T. I. Eikaas, B. Foss and D. Gillet, A remote laboratory experimentation network, *1st IFAC Conf. Telematics Applications in Automation and Robotics*, Weingarten, July 24–26, 2001.
4. D. Gillet, H. A. Latchman, Ch. Salzmann and O. D. Crisalle, Hands-on laboratory experiments in flexible and distance learning, *J. Eng. Educ.*, April 2001, pp. 187–191.
5. H. A. Latchman, Ch. Salzmann, S. Thottapilly, and H. Bouzekri, Hybrid asynchronous and synchronous learning networks in distance education, *Int. Conf. Eng. Educ.*, Rio de Janeiro, Brazil, 1998.
6. D. Gillet, and G. Fakas, eMersion: a new paradigm for Web-based training in engineering education, *Int. Conf. Engineering Education, ICEE 2001*, Oslo/Bergen, Norway, August 6–10, 2001.
7. B. Armstrong, and R. Perez, Control laboratory program with an accent on discovery learning, *IEEE Control Systems Magazine*, February 2001.
8. X. Vilalta, D. Gillet, and Ch. Salzmann, Contribution to the definition of best practices for the implementation of remote experimentation solutions, *IFAC Workshop on Internet Based Control Education, IBCE'01*, Madrid, Spain, December 12–14, 2001.
9. D. Gillet, Ch. Salzmann, H. A. Latchman, and O. D. Crisalle, Advances in remote experimentation, *Proc. 19th American Control Conference*, Chicago, Illinois, USA, June 28–30, 2000, pp. 2955–2956.
10. Calerga Sarl. http://www.calerga.com.
11. E. Harold, *XML Bible*, IDG Books, USA, (1999).
12. M. Morrison, *Java 1.1 Unleashed*, Third Edition, MacMillan Computer Publishing (1997).
13. J. Roschelle *et al.*, Developing educational software components, *IEEE Computer Magazine*, September 1999, pp. 50–58.
14. L. Atkinson, *Core PHP Programming*, Prentice-Hall (1999).
15. M. Cooper, E. Scanlon, and S. Freake, Remote controlled teaching experiments, in science and engineering subjects, accessible over the World-Wide-Web: the PEARL project, *Proc. ED-Media Conf., Montreal*, Canada, June 26–July 1, 2000.
16. C. Levert, and S. Pierre, Towards a methodology for designing distributed virtual laboratories, *Proc. ED-Media Conf., Montreal*, vol. 2, Canada, June 26–July 1, 2000, pp. 57–62.
17. Timbuktu, http://www.netopia.com/.
18. Virtual Network Computing. http://www.uk.research.att.com/vnc/howitworks.html.

Denis Gillet is MER (Associate Professor) at the Swiss Federal Institute of Technology in Lausanne (EPFL). He received the Ph.D. degree in Control Systems in 1995 from EPFL. His research interests include optimization of dynamic systems, new learning technologies, augmented reality for sustainable interaction, and real-time Internet services. Dr. Gillet received the 2001 iNEER (International Network for Engineering Education and Research) Recognition Award for Innovations and Accomplishments in Distance and Flexible Learning Methodologies for Engineering Education.

Frédéric Geoffroy is engineer in new learning technologies at the Swiss Federal Institute of Technology in Lausanne (EPFL). He received a diploma in mechanical engineering and a DEA (French post grade diploma) from the University of Technology of Compiègne (UTC) in 1996, France. During the last four years, he participated to the development and management of different distance learning projects. His research interests include document engineering and new educational technologies.

Karim Zeramdini is an assistant and a Ph.D. candidate at the Swiss Federal Institute of Technology (EPFL), Lausanne. He received his diploma in Computer Science in 1999 from the University of Sfax, Tunisia. His research interests include document engineering and new educational technologies.

Anh Vu Nguyen earned his B.Sc. Degree in Computer Science from Viet Nam National University, Ho Chi Minh City in 1997. In 1999, he was awarded a two-year Swiss federal scholarship to carry out research activities and to take part in Pre-doctoral School, Swiss Federal Institute of Technology in Lausanne (EPFL). In 07/2001, he obtained a Pre-doctoral School Degree in Communication Systems, EPFL. Since 08/ 2001, he is an assistant and a Ph. D. candidate at the School of Engineering, EPFL. His research interests include collaboration systems, computer supported collaborative learning, document engineering, Web-based technologies and software engineering.

Yassine Rekik, born in 1973 in Belgium, graduated in Computer Sciences Engineering in 1997. He received his Ph.D. in Computer Sciences in the Department on Computer Sciences at the Federal Institute of Technologies in Lausanne (EPFL), Switzerland, in april 2001. He is currently holding a post-doctoral position at the Theoretical Computer Sciences Lab at the EPFL and working on Structured Documents Engineering (XML technology) and their applications in Distant Learning Environments. Since June 2001, he is coordinating the eMersion project's team.

Yves Piguet is CEO of Calerga Sarl, a company he co-founded in 2001 to develop and commercialize scientific software such as SysQuake whose he is the main developer. He is also lecturer at the Swiss Federal Institute of Technology in Lausanne (EPFL). He received a diploma in Microtechnics in 1991 and a Ph.D. degree in Control Systems in 1997, both from EPFL. His research interests include robust control, real-time controllers and their use in mobile robotics, and interactive software and its applications to get an intuitive understanding of complicated theories.