DEVELOPMENT AND EVALUATION OF A COMPUTER-BASED LABORATORY TEACHING TOOL

N.M. Avouris*, N. Tselios*, E.C. Tatakis#

University of Patras, Electrical and Computer Engineering Department

* Human-Computer Interaction Group

{N.Avouris, nitse}@ee.upatras.gr

Laboratory of Electromechanic Energy Conversion

E.C.Tatakis@ee.upatras.gr

Abstract **–** Effective evaluation of educational software is a key issue for successful introduction of advanced tools in the curriculum. This paper focuses on our experience of developing and evaluating a tool for computer-assisted teaching of engineering laboratory courses. Various categories of educational software that can be used for laboratory courses are described. An example of such experimental system is also provided. Finally issues and techniques relating with usability evaluation of instructional software are described with examples of application of these techniques during the design and experimental use of electrical-machines laboratory educational software.

Keywords : computer-based teaching, laboratory course, educational software evaluation, software usability, heuristic evaluation

1. INTRODUCTION

During the last years, software tools in various forms have started playing an increasingly important role in educating students of traditionally hard engineering subjects like electrical, mechanical or civil engineering. With computer-based tools becoming more affordable we have the expectation that time and distance factors will have less impact on the way instruction is delivered to students of such subjects. As an example, during recent years a significant increase of use of educational software in electrical engineering laboratories has been observed [10]. While laboratory-based training does not seem to be due for replacement by software model-based training in the foreseeable future, instructional software can be used as a complementary tool during laboratory work. For instance, the theoretical background of the laboratory work can become available to the students through multimedia software, preparation of the laboratory exercises and contacting of the laboratory experiments can be supported by computer systems, used for collecting data, processing measurements, testing wiring and equipment configurations, simulating behavior of equipment etc. It is hoped that computer-assisted teaching can help us deliver a deep and solid education to the specific subject, increasing the interaction between the instructor, student and the concepts under investigation.

Appropriate use of this educational software allows students to build knowledge by giving them opportunities to explore the equipment to be used beforehand in a safe for them and the machinery way, interact with it, experiment, problem-solve, and collaborate. Interactive, multimedia experience cannot replace the real laboratory work but can enhance the learning process of many students, help them find the relation between the theoretical principles and the observed behavior in an easy and intuitive way.

Any instructor or laboratory personnel that decides to build or use educational software for teaching laboratory courses, should go through a process of evaluating the system and deciding on its suitability for the educational process concerned. The evaluation techniques to be used come from a body of knowledge and results produced by the learning technology and software usability disciplines. These results on educational software usability evaluation need to be adapted for the specific needs of Laboratory instruction.

In order to accomplish this task we need to measure, through subjective and objective tests, the quality and reliability of the educational software, the overall usability level and the way that this software is going to be integrated in the instructional process.

In this paper we describe our experience with designing, developing and evaluating educational software for supporting laboratory work. This typical system implements many educational software techniques like use of multimedia components, interactive machine diagrams, graphic formula representations and detailed theory explanation. The software also contains quiz material with comments on possible errors, through which monitoring of the learning process can be achieved. It also provides explanations on the work that the student has to accomplish in the laboratory and tools and guidelines concerning drawing of conclusions by the student and drafting of the Laboratory Report.

Subsequently, the software evaluation process is discussed, following a brief survey of usability and learnability evaluation techniques, considered particularly interesting and applicable in this context. We believe that the reported experience is of great interest to instructors of Engineering Laboratory Courses who often find themselves in the dilemma of selecting among alternative tools to support them in their job.

2. SOFTWARE TOOLS IN LABORATORY COURSES

There are various categories of educational software that can be used in Laboratory Courses, grouped by the specific task, which they are focused on:

- **Multimedia presentations and tutorials.** These tools provide a theory background to the student, enhanced by the use of different media such as sound, video, text and hypertext (topic-linked documents), pictures and animations. These presentations aim at providing the students with a realistic description of the topic and enhance greatly their interest.
- **Problem solving and self-examination systems.** They give the opportunity to the students to evaluate the quality and amount of acquired knowledge relative to the subject studied, and unveil their weaknesses and misunderstandings.
- **Laboratory preparation software.** This class of educational software provides information about the structure of a laboratory exercise, gives the theoretical background, analyzes the physical models underlining the equipment used, discusses the tasks which the student has to accomplish and provides tools or guidelines to collect the data, helps them come to conclusions and explain their observations.

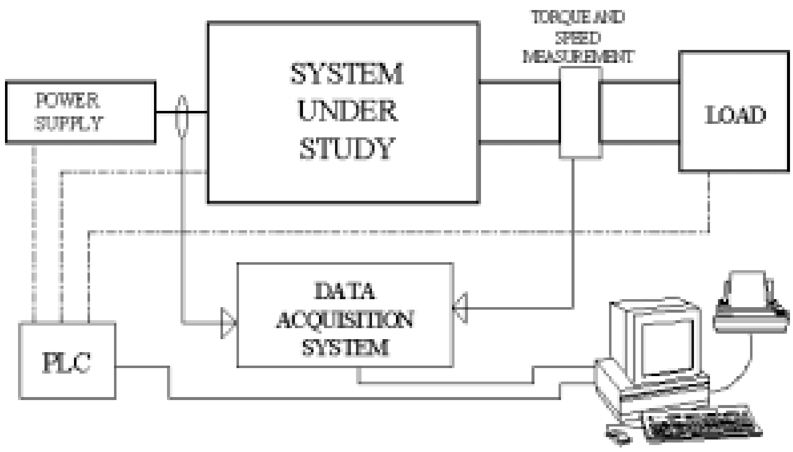

Figure 1. Possible interconnection of support system with laboratory equipment

- **Laboratory work support system.** This is perhaps the most interesting case, since it involves deep integration of software system and the laboratory equipment [5, 11, 20, 21]. The objective is to automate the execution of the laboratory work. An example of such system, relating to an Electrical machines Laboratory, is shown in figure 1. This improves the classical way to perform engineering laboratory work, which presents the following disadvantages:

a) A lot of time is spent by the students for preparatory activities like wiring and by the supervisor for checking the wiring.

b) Long manual measurements result in shifting attraction away from the lab objective.

c) Measuring instruments are protected against accidental damage

d) Lab reports, containing measurement analysis, are prepared by the students after leaving the laboratory. Consequently, there is no possibility for checking or repeating measurements that could have contributed to deeper understanding of the subject.

e) Group work in labs, results in not effective participation of all students during the laboratory work.

- **Modeling and simulation** This type of educational software makes use of computer models in order to simulate the behavior of the system or process under study, substituting the laboratory equipment [1,6,7,12,13,18, 22]. This way it is possible to repeat an experiment many times, comparing the findings with the model-based values.

We can expect that a complete laboratory-educational software can cover most of the above functionality, maximizing the interaction with the student and enhancing various phases of the educational process. The automation of the experimental bench through introduction of a combination of the above systems, improves the quality of education, offering important advantages such as:

a) The students devote their time to fruitful discussion and useful observations, having the possibility to analyze directly the measurements, repeat some of them, compare with theoretical or simulation results etc

b) The students are concentrated on understanding fundamental concepts and not performing tedious wiring and measurements

c) The students can be introduced to interrelated disciplines, such as from electrical machines to power electronics, automation and control, data acquisition etc

d) Drilling can be enabled at any time without supplementary effort by the educational staff

e) Minimization of failures due to wrong wiring

f) Minimization of the effort needed by the laboratory support staff.

g) New possibilities on continuous education, distance-learning, collaboration with industries and training of industrial personnel etc.

3. PEGASUS: A CASE OF LABORATORY EDUCATIONAL SOFTWARE

Pegasus has been designed as an integrated electrical machines laboratory support system. It was the result of collaboration of *Human-Computer Interaction group* and *Laboratory of Electromechanical Energy Conversion* of our Department [3,4]. The first prototype was built to support students conducting laboratory work involving a 3-phase Induction Motor. Pegasus aims at helping the student study the background concepts required for preparation and doing the relevant laboratory exercises. The student can work at her own pace and familiarize herself with the equipment to be used through multimedia presentation some time before the laboratory. The self-examination module gives the opportunity to the student to evaluate the progress, and discover possible weaknesses and misunderstandings. The system can also be used during the laboratory exercise for collecting measurements, preparing graphs and reviewing the progress of the laboratory work at run time.

2.1 Typical system usage

The Pegasus system is made up of 5 modules, described briefly in the following.

1.**Induction Motor background theory exploration** Here the user can review the background theory presented in the same form as the laboratory hand-outs. This module has been incorporated in the system for completeness sake, so that the student will not need to go back to the printed material in order to obtain information about the laboratory and the background theory. An active presentation of the motor equivalent circuit has been included, the parameters of which have to be derived during the laboratory exercise. The student can interact with any part of a graphic representation of the motor by simply clicking or moving the mouse over it in order to get feedback about the particular part and its role.

2. **Self-examination module.** Through this module, Pegasus provides a set of quizzes of various types (multiple choice, direct answer entry and diagram evaluation) divided in three difficulty levels. The user can select any question at any time and attempt to answer it within a specific time slot. In case of incorrect answer, the student receives clarifications and tips in order to understand why the answer is not correct, as shown in figure 2. Overall performance is also measured. The questions have been evaluated for their pedagogical and scientific completeness by a group of experts. The performance of each user is stored into a database in order to compare previous performances or suspend the interaction and continue later the session from the same point.

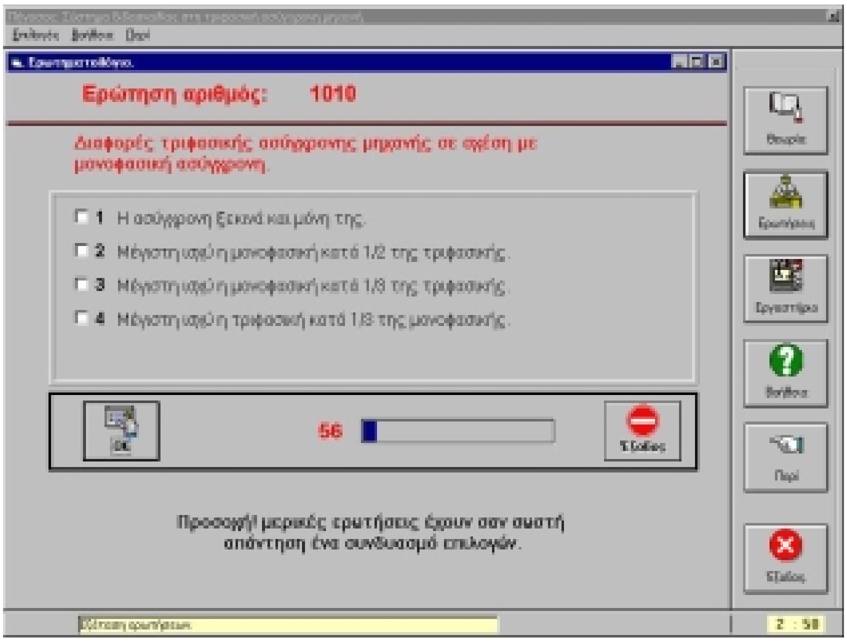

Figure 2. A multiple-choice question from the self-examination module

3. **Laboratory connectivity diagram exploration.** Using this module, the student can become familiar with the equipment connectivity diagram, which she will have to build during the laboratory exercises (selection of measuring instruments from a library and comparison of the diagram with the effected connections). The student can explore and navigate an interactive diagram, shown in figure 3, from which can receive feedback and further information, by simply clicking on each part. The student can also experience different equipment connectivity configurations and load variations.

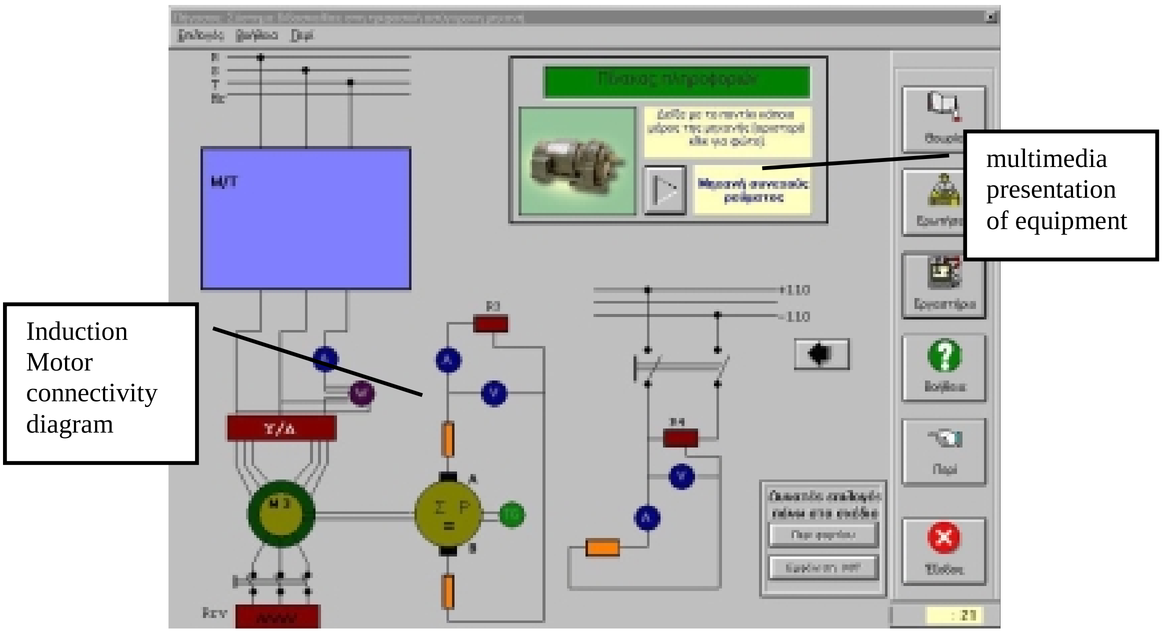

Figure 3. Laboratory connectivity diagram with superimposed multi-media presentation of the equipment

4. **Laboratory equipment multi-media presentation** A multimedia presentation of the equipment to be used and the exercises to be performed is also included. The student can see through this presentation pictures of the equipment concerned, videos of laboratory personnel presenting the scope of the exercises and watch other students perform the requested exercises. This presentation can be related directly with the laboratory connectivity diagram and the background theory discussed earlier.

5. **Run-time lab module.** In this module, which can be used immediately before the exercise execution and during the exercises in a computer next to the equipment or the student laptop, the student can receive specific directions about the steps required to carry out successively the lab and extensive explanation of what each step requires in order to be accomplished. Also the students can store measurements taken and process the measured values in a custom spreadsheet provided. Direct comparison of the experimental measurements with the model values can be done, in order to ensure that the process is correct. Pegasus can also support graphs drawing and generally help the students arrive to justifiable conclusions and explain their observations.

4. USABILITY EVALUATION

Development of effective interactive software involves substantial use of **evaluation experiments** throughout the process [8,14,15,17]. During this evaluation the system is measured against pre-determined goals and objectives. Evaluation of laboratory educational software of the complexity described in the previous sections, requires new techniques. The instructional process involved in engineering laboratories, comprises many phases. Some of them impose time constraints (e.g. the actual experiment is usually performed within a predefined period of time), while others, like the experiment preparatory phase, necessitate high interconnection with a body of background theoretical knowledge, which should be related with practical aspects of the experiment and the process. These characteristics indicate that an evaluation process, with multiple objectives, suitable for this class of software should be derived and used. For this reason, the evaluation method proposed in this paper comprises a number of distinct phases:

- **Usability evaluation by interaction experts,** based on well-established heuristics and design guidelines on presentation, user interaction etc.
- **Laboratory instructors evaluation,** which ensures the quality of content and conformance with the requirements of the curriculum and the established procedures.

- **Pedagogical experts evaluation,** relating with the instructional aspects of the software, adaptability to specific students needs, pedagogical strategy etc.
- **In field evaluation** testing through observation and interviews with students after they have used the software following typical educational scenarios including relevant laboratory work. This evaluation can result in subjective measurements of user satisfaction as well as objective measurements of system performance.

Various specific techniques have been proposed in the literature to be applied during the above phases. Questionnaires, interviews, logging performance, timing and keystroke level measurements, cognitive walkthrough, subjects' observation through adequate equipment etc.

In the following, after a short survey of existing techniques, we present our experience with evaluating Pegasus usability, using suitable alternative techniques and discussing the results of the evaluation process.

4.1 Usability definition

Usability is described in ISO 9241 draft standard as the "extend to which a product can be used with effectiveness, efficiency and satisfaction in a specified context of use" [9]. In [14] the term is further analyzed in: Easiness and speed of learning of system use, efficiency to use, easiness to remember system use after certain period of time, reduced number of user errors and easy recovery from them, subjective satisfaction of users.

Many attempts have been made to measure usability according to these dimensions and evaluate interactive software systems [14,16]. Also many attempts have been made to relate these aspects with system performance. In our case measure of performance can be considered the measure of improved learning, e.g. better understanding by the students of the laboratory setting and the task in hands and better relation of the task to the underlining theoretical model of the experiments. The evaluation techniques that are suitable for Laboratory software evaluation are described in the following.

4.2 Inspection methods

Usability inspection methods are evaluation methods involving usability experts examining the software user interface. Many inspection methods can be based on specifications that have not necessarily been implemented yet, so they can be performed early in the software lifecycle, though some methods also address issues like the overall system usability concerning the final prototype.

The most suitable methods of this category for our case are:

Heuristic evaluation involves usability specialists who judge whether each dialogue element follows established usability principles (the "heuristics").

Cognitive walkthrough uses a detailed procedure to simulate a student problem-solving process at each step through the dialogue, determining if the simulated user's goals and memory content can be assumed to lead to the next correct action.

Pluralistic walkthrough uses group meetings where students, developers, and usability experts step through a learning scenario, discussing each dialogue element.

Feature inspection lists sequences of features used to accomplish typical tasks, checks for long sequences, cumbersome steps, steps that would not be natural for students to try, and steps that require extensive knowledge/experience in order to assess a proposed feature set.

Standards inspection, during which experts inspect the interface for compliance with certain standards. This can involve user interface standards as well as educational software standards, departmental standards if they exist, etc.

Guidelines checklist help ensure that usability principles will be considered in a design. Usually, checklists are used in conjunction with a usability inspection method. The checklist gives the inspectors a basis by which to compare the product.

It should be observed that inspection techniques are also suitable to be used by other experts like the domain experts and pedagogical experts. These will perform the **Laboratory instructors evaluation,** which ensures the quality of content provided and conformance with the requirements of the curriculum and the established procedures and the **Pedagogical experts evaluation,** relating with the pedagogical aspects of the software, adaptability to specific students needs and pedagogical method as described earlier.

4.3 Testing methods

These are tests, measuring system performance against pre-defined criteria. Typically we observe individual users performing specific tasks with the system. Data are collected on measured performance. For example, time required to complete the task or number of errors made. Selection of appropriate users and representative tasks is essential. Also a properly designed and organized usability laboratory is important. [15, 8, 16].

The most widely accepted usability testing techniques, suitable for laboratory software, are:

Thinking Aloud Protocol is a technique widely used during usability testing. During the course of a test, the participant is asked to vocalize his or her thoughts, feelings, and opinions while interacting with the software, performing a task - part of a user scenario. While this technique is particularly difficult to use with young students, who are distracted by the process, it can easily be applied at College level and Higher Education Laboratory courses.

Co-discovery is a type of usability testing where a group of students attempt to perform tasks together while being observed, simulating typical educational process, where most people have someone else available for help. This can be particularly suitable in our case if the laboratory is done by groups of students.

Questioners and Interviews based protocols These techniques instead of waiting for the students to vocalize their thoughts, prompt them by asking direct questions about the system. Their ability (or lack of) to answer questions can help evaluators decide about parts of the system interface that present difficulties for the users.

Performance measurement. Some usability tests are targeted at determining hard, quantitative data. Most of the time this data is in the form of performance metrics. E.g. required time to execute specific tasks.

Finally **in-field studies** concern observation of the users performing their tasks in their usual environment of study/work. These techniques have the advantage of the natural user performance and group interaction however present limitations in terms of measuring performance, since the necessary testing equipment cannot be used in a typical workplace.

5. EXPERIMENTS ON PEGASUS USABILITY EVALUATION

During design and experimental deployment of the Pegasus software, a number of usability evaluation experiments have been contacted, based on adapted techniques according to the principles described above. Three experiments are outlined here. It should be mentioned however, that the described experiments are not prescriptive, since one can select alternative techniques than the ones used in our case, out of the list of section 4, in order to suite the specific characteristics of the software under evaluation best.

5.1 Adapted Heuristic evaluation

This evaluation phase was interleaved with the prototype development. During heuristic evaluation the interface was carefully examined by a group of system design experts against a list of commonly accepted principles (heuristics). The experts studied the interface and made their observations in written form. They were also provided with an evaluation sheet in which they could quantify their judgement by assigning marks indicating degree of conformance with each particular rule. The range of assigned values was [-2 ..+2]. An importance weight (w_i) was also given to each rule, indicating the relevance of the general principle to the system according to the experts opinion. A quantitative evaluation of the system can by obtained by the formula: $e= \sum w_i .r_i$, where r_i the average score of rule i and w_i the relative weight of this rule according to all experts opinions. This approach is particularly useful in case of comparative evaluation of alternative systems.

The heuristic rules of interface design that the experts tested, are proposed by [14]. The rules used where the following:

(1) Use of simple and natural language

(2) Minimization of required memory load from the user

(3) Consistency in terminology and symbols used throughout the interface and conformance with the domain norms and standards

(4) Degree and quality of system feedback

(5) Provision of clearly marked exits and undo

(6) Provision of shortcuts for experienced users

(7) Informative error messages.

(8) Prevention of errors

	termi-nology	memory load	consist-ency	feed-back	undo	short-cuts	error messag	prevent error
mean values	1.6	1.1	-0.5	1.4	0.7	-0.6	0.5	0.3
weight	0.94	0.84	0.74	0.72	0.74	0.54	0.76	0.68

Figure 4. Adapted heuristic evaluation of Pegasus. Synoptic results table

A group of ten evaluators was engaged in this experiment. This follows empirical studies, which have shown that six to eight evaluators are able to detect about 75% of the usability problems.

Heuristic evaluation is performed by letting each individual evaluator inspect the interface alone. This procedure is important in order to ensure independent and unbiased views. The evaluators received instructions to go through the interface at least twice following a typical scenario of use, looking at each element of the interface (for example, each menu item, button, control, etc.) After the evaluation session, a meeting with the participation of all the experts helped them exchange their views and discuss their individual findings.

The mean values of the quantitative judgement of the experts are shown in figure 3. According to this graph, negative mark was given to application of rules 3 (terminology consistency) and 6 (provision of shortcuts). The importance of all rules, according to the experts was in the range [0.5.. 1.0]. Through this graph areas of problems are clearly shown. The weight factors also help identify the severity of the problems, so combination of high importance weight and low mark necessitates corrective actions. This view is also accompanied by the written suggestions of the experts. Suggestions like "The audio presentation should be made optional-with clearly identified on/off selection" have been codified and taken into consideration in the next design iteration. In particular suggestions relating to rules of high importance or suggestions with high frequency were given high priority in the list of modifications.

It has been proven that heuristic evaluation can be used at almost any time during the development cycle, although it is probably best suited to earlier stages, when the design is not firm enough. Experts can be provided with paper mockups, or even just design specifications, and still get a good amount of usability problems discovered before actual production work begins. However in our case the experiment was based on a late stage prototype, and was used as an overall system quality index, according to the experts’ judgement.

5.2 Performance Measurement

The next experiment concerned measurement of system performance in relation to a planned objective and a predefined scenario of typical use. This test usually involves typical users of the system. In our case, our experts (graduate students of the ECE Department) could play also this role, since they had all done similar laboratory work in the past and they were familiar with the underlying theory of the particular laboratory. For this reason we decided to integrate this experiment with that of Heuristic Evaluation, described above. So the users involved in this experiment were also ten.

They were given a typical scenario of use, which comprised some of the following steps:

(Task A) - *register with the system* - *study the equivalent diagram for the machine, follow the presentation of the equipment to be used and the alternative presentation of the 3phase equivalent motor.* - *answer at least two quiz questions for each group of questions and then check your overall performance* - *Go to the Lab Execution part and fill-in the spreadsheet with imaginary measurements, while doing this, try to go back to the machine diagram without using the menu options* - *use the help screens when necessary.*

The students had to perform task A, under observation, but with no intervention of the Usability Lab personnel, who timed their performance and took notes. Before the experiment the objectives were set as shown in the first row of the following Usability Objective Table. Finally after the experiment, the last row of the table was filled-in, where one can see that Pegasus met the objectives.

An issue for discussion in relation to Performance Measure evaluation experiments, is to identify typical performance objectives. It has been suggested that performance is not an issue in educational software, while measuring learning effect is more essential. For laboratory work, however, when typically the exercises have to be done within a certain period of time, performance can have many components, including time-related ones like the one discussed here. Other performance measures can be the number of user errors during execution of a specific task, percentage of improvement in accuracy of measurements, quality of laboratory reports in relation to traditional techniques, number of errors in setting up and preparing the experiment etc.

Time to complete successfully task A	*Best case*	*average*	*Worst case*
Objective	*14-18 min*	*20 min*	*26-30 min*
Measured performance	**12 min**	**17.10 min**	**23 min**

5.3 User Questionnaires based on guidelines checklists

The last evaluation experiment concerned use of the system by a group of actual users (9 undergraduate students of the ECE Department), who had to perform a typical lab preparation task and subsequently fill in a check-list questionnaire. This questionnaire is based on the QUIS [17] usability questionnaire, as it has been adapted by [2] for evaluation of educational material. The questionnaire contains 75 questions grouped in 10 separate categories. These categories, indicative of the objective of the evaluation are: (1) General System Performance, (2) Software Installation, (3) Manuals and On-line Help, (4) On-line Tutorials, (5) Multimedia quality, (6) Information Presentation, (7) Navigation, (8) Terminology and error messages, (9) Learnability, (10) Overall system evaluation.

Each question can receive 5 possible answers in the range [-2 .. +2], while a relevance weight factor is assigned to each question with values in the range [1 (irrelevant question) to 5 (very relevant)]. A free remarks section is included at the end of each group of questions and a general one at the end of the questionnaire. The purpose of the questions is to evaluate specific features of educational system, performance, quality and user satisfaction. This experiment is complementary in nature to the other two. The users test mainly for usability, however this is done through a clear **system utility** perspective, as demonstrated in the students comments. It should be mentioned that unlike the previous experiment where the experts had a clear objective to identify design limitations and measure system performance, in this case the students had as their objective to do the laboratory work in a satisfactory way. Overall user comments expressed their satisfaction with the Laboratory Equipment section, concern about the level of background theoretical presentations and suggestions for more integration with the laboratory equipment at run time, for instance through incorporation of hardware interface to the measuring equipment. Remarks concerning specific terminology used, dialogue flow, multimedia material quality were also made. Some of the later remarks coincided with the comments of the Heuristic Evaluation experiments performed earlier, as discussed in the following section.

This final test, due to the large number of specific questions, was much more tedious to perform, however it has been proven very useful. It can be used as a *formative evaluation* tool, with the purpose to improve design or as a *product evaluation* tool, in order to decide for suitability and quality before adoption by laboratory personnel.

At the end of the three experiments, the final conclusion was drawn by the Evaluation team, which comprised usability experts and instruction personnel of the Laboratory concerned. The Final Report contained a short presentation of the used methodology, an overall judgement on the suitability of the Pegasus software for the particular task, suggestions on integration procedure of the software in the lab routine and a list of suggestions to the Development Team for further improvement, together with a specific list of identified problems.

6. DISCUSSION ON EVALUATION STUDIES.

The evaluation studies described in the previous section identified a number of shortcomings of the educational system, which could not be found by the design team. Additionally the two main experiments had a different focus and for this reason they produce complementary results. It is argued here that a thorough evaluation of any educational software and in particular complex Laboratory-instruction tool, should comprise at least a test involving usability experts and a test involving perspective users. It should be mentioned that a complete evaluation methodology should also comprise evaluation by domain experts and pedagogy experts, as described in section 4. These tests in our case were incorporated in the software design cycle and are not discussed in this paper.

In order to demonstrate the correlation between the two experiments performed, a comparative study of the results was undertaken. This involved both the quantitative and the qualitative data collected. According to the quantitative results of the experiments, an attempt was made to correlate the scores obtained by the two groups. The first results, concerning the experts judgement on the eight heuristic usability rules, were shown in figure 4. The second experiment contained a different set of questions. These were analyzed in terms of the original heuristics. As a result we found out that 41 out of the 75 questions concerned these heuristics and they could be used as a basis of a comparative study. The remaining 34 questions of test B concerned other aspects, like Help and documentation, Navigation, Efficiency, Modularity, Performance and Media quality.

The relevant questions of test B were related to the heuristics of the first experiment according to the following table:

Heuristic Rule	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***Sum***
Number of questions of Test B	8	5	6,5	4,5	3	7	5	2	41

For the group of these common questions, a comparison of the average scores for the two experiments was performed. The result of this study is shown in figure 5.

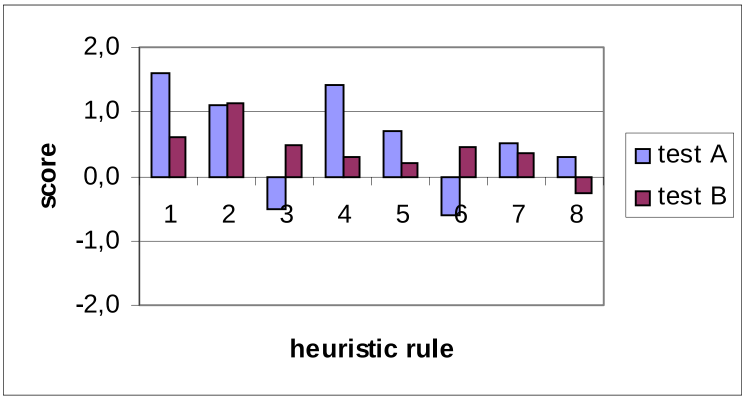

Figure 5. Combined results of evaluation tests A (usability experts) and B (students) of Pegasus, according to the heuristic rules of section 5.1.

As seen in figure 5 the judgement of the two groups of evaluators is not identical. The students were more reluctant to express negative judgement. There are a number of heuristic rules for which the scores are in the same range, like rule 2 (memory load)= satisfactory, 3 (consistency)=average, 5 (clear exits, undo) =average, 6 (shortcuts) =average, 7 (error messages) =average and 8 (prevention of errors) =average. For the heuristic rules 1(use of simple language) and 4 (feedback) there seems to be a clear disagreement, since in both cases the usability experts judged the system as “satisfactory” while the students as “average”. A possible explanation for this disagreement should be sought in the different perspective of the two groups. For instance the use of simple language in the dialogues seen overall by the experts, was considered satisfactory. On the other hand, the students discovered flaws, since they had to perform more detailed evaluation of the “flow of dialogue” in which they where more deeply involved, as they undertook a complete task. Similar observation is valid for the heuristic rule concerning quality of system feedback. For instance, the users had to navigate more thoroughly the multimedia presentation of the laboratory environment and felt more strongly than the usability experts that better system feedback was needed, as shown also in their suggestions discussed bellow.

The comparison of the qualitative judgements involved comparison of the lists of specific suggestions, which were included in the evaluation reports during the two tests. These lists where sorted in terms of frequency of each comment in the corresponding Evaluators Reports. The top entries of the two lists are presented in the following table.

Usability Experts Suggestions	*Frequency*	*Students Suggestions*	*Frequency*
1. Lack of adequate control over audio channel	30%	1. The on-line help was inadequate and not context-sensitive	100%
2. In multiple-choice questions screen space is not managed properly	20%	2. Lack of adequate control over audio channel	78%
3. The “about” sub-dialogue does not terminate	20%	3. Certain inconsistencies in terminology used where identified.	44%

4. Certain inconsistencies in terminology where identified.	20%	4. Not adequate navigation support in multimedia material	33%
5. The color-code scheme used to identify tasks was not consistent.	20%	5. Not adequate user control over presentation of induction motor diagram.	33%

By inspection of this Table, we observe that the student group had higher agreement rate over their suggestions (shown by the high frequency of the suggestions), something reflecting the fact that an after-evaluation session followed the test, where the subjects exchanged their views over the software. Another observation was that while the two lists contained some common comments, like observations 1 and 4 of the test A, however the different perspectives of the two groups resulted overall in more rich set of suggestions.

7. CONCLUSIONS

Our experience with development and evaluation of educational software to support an engineering laboratory course has been described in this paper. The main features of the Software *Pegasus*, developed to support students of our Department during preparation and execution of an Electrical Machines laboratory were presented. Special emphasis was given in the usability evaluation process of the software. A methodology for evaluation of software of this nature was outlined. The three usability evaluation experiments, conducted in the frame of this evaluation process, have been presented. The three experiments where shown to be complementary in nature and can be used by engineers involved with instructional material development during system design and by laboratory instructors during product acquisition and adoption. The described experiments do not necessitate use of special equipment. Each experiment covers a different aspect. *Heuristic Evaluation* tests conformance with well-known usability heuristics, *Performance measurement* tests the system performance in a typical usage and *User questionnaires* tests the reaction of typical users to the system. The results of the experiments were impressive. Each test unveiled different aspects of the software that the developers could not discover. The first one identified usability flaws, the second tested system performance and the third one presented the view of the users. The three combined techniques which have been adapted from well known usability engineering research, form an integrated usability methodology to be implemented by any engineering laboratory as a valuable tool in the effort to make the correct decisions concerning introduction of computer technology in the educational process.

Acknowledgements Special thanks are due to the Personnel of the Laboratory of Electromechanical Energy Conversion of the ECE Dept of the University of Patras who supported this work and in particular the Head of the laboratory Prof. A Safacas and Lecturer J. Kapatou. We also like to acknowledge the support of the group of Usability Experts from the Human-Computer Interaction Group of the ECE Dept and our undergraduate students who evaluated the Pegasus software.

7. REFERENCES

1. Angelidis V.G.: "Incorporating Instructional Feedback in Elec. Eng. Laboratory Experiments - An Example", IEEE Trans. on Education, Vol. 40,(1), February 1997.
2. Avouris N. et al. Educational Software Evaluation Guidelines, Technical Report, Working Group on Educational Software Evaluation Procedure , Pedagogical Institute, Athens, 1999.
3. Avouris N., Tselios N., Conclusions of Heuristic Evaluation of Educational Software: Application in Laboratory Instructional System, Proceedings Infoworld98, EPY Publication, Peireas, 1998.
4. Avouris N.M., Tselios N., Tatakis E.C., Computer-Assisted Teaching of Electrical Machines: Study and Implementation of Usability Evaluation Techniques, Proc, Electromotion 1999, vol 2, pp. Patras July 1999
5. Belmans R., Reekmans R., Verdyck D., Geysen W.: "Computer Aided Testing of Electrical Machines in the Teaching Laboratory", Int. Conference on Electrical Machines (ICEM'88), Pisa (Italy), 1988, pp. 355-359.

6. Benbouzid M.E.H., Capolino G.A.: "Project-Oriented Power Engineering Laboratory", IEEE Transactions on Power Systems, Vol. 11, No 4, November 1996.
7. Daniels M., Shaffer R A., Re-Inventing the Electrical Machines curriculum, IEEE Transactions on education Vol. 41 No2, May 1998
8. Dix A., Finley J., Abowd G., Beale A., Human-Computer Interaction, Prentice Hall, 1998.
9. Fitzpatrick R, Higgins C., Usability Software and its attributes: A synthesis of software quality, European Community Law and HCI, People and Computers XIII, pp. 3-22, Spriger-Verlag, London, 1998.
10. IEEE Proceedings, F. T. Ulaby, S.W.Director (guest editors), Special Issue on Electrical and Computer Engineering Education, vol 88 (1), January 2000.
11. Kasten D., Kent J.C., Maco T.J., Scott S.C.: "Modernization of Machine Laboratory", Proc. 54th Annual Meeting of the American Power Conference, Chicago (USA), 1992, April, pp. 130-135.
12. Kyranastassis G.J.: NEAPOLIS: A Special Tool for Electrical Power Engineering Education, Proc. 31st Univ. Power Engineering Conf., Iraklio, 1996
13. Nehrir M.H, Fatehi F, Gerez V, Computer modeling for enhancing instruction of electric machinery, IEEE Transactions on education Vol. 38 No2, May1995.
14. Nielsen J., Usability Engineering, Academic Press, London, 1993.
15. Preece J. et al., Human Computer Interaction, Addison-Wesley, 1994.
16. Rubin, J., Handbook of Usability Testing, J. Wiley & Sons, New York, 1994.
17. Schneiderman B., Designing the User Interface, Addison Wesley, 1998.
18. Sebastien R.Y., Mohammed O.A.: "Real-time Electromagnetic Field Analysis System for the Energy Conversion Laboratory", Proceedings of the 1998 IEEE SoutheastCon Conference, Orlando (USA), 1998, April 24-26, pp. 208-211.
19. Squires, D., Preece, J., Predicting quality in educational software: Evaluating for learning, usability and the synergy between them, Interacting with computers, 11, pp.467-483, 1999.

20. Staton D.A., Mc Gilp M.I., Miller T.J.E.: "DC Machine Teaching Experiments", European Conf. on Power Electronics and Applications (EPE'93), Brighton (England), 1993, pp. 35-40.
21. Walsh P.J., Hancock G.C., Zeng Y., Marik O.P.: Measurement Setup for a Power Laboratory Motor Generation Station, IEEE Trans. on Education, Vol. 38, 2, May 1995, pp. 150-157.
22. O. Montero-Hernandez, A. R. De La Rosa, D. Baez-Lopez, R. Alejos, E. Enriquez, Power Lab: A tool to learn electrical machines and power electronics, Computer Applications in Engineering Education, Vol. 7, (4), pp. 213-220 1999.