EVALUATION OF LEARNING IN COMPUTER BASED EDUCATION USING LOG SYSTEMS

Martti Rahkila and Matti Karjalainen

Helsinki University of Technology
Laboratory of Acoustics and Audio Signal Processing
P.O. Box 3000
FIN-02015 HUT, Finland
Martti.Rahkila@hut.fi, Matti.Karjalainen@hut.fi
http://www.acoustics.hut.fi/

ABSTRACT

Evaluation is difficult but important part of learning to both teachers and students. This paper discusses the idea of using log systems to collect information in Computer Based Education (CBE). The log information can be analyzed and used together or even instead of traditional evaluation measures. Furthermore, in terms of self-study CBE material, logging can be beneficially applied to interactively control learning. As a special case of interest, methods for using WWW log system in learning evaluation as well as advanced structures for interactive WWW-based education are discussed.

1. INTRODUCTION

The evaluation of learning is without a doubt of interest to every teacher. However, it is very difficult to develop any objective or technical measures to it. After all, learning evaluation contains also philosophical aspects such as what learning inherently is. Still, teachers have throughout years developed and used some criteria to estimate their work results. The traditional methods include examinations and feedback forms. However, Computer Based Education (CBE) allows another possibility, *logging*, that is, gathering information and analyzing user's actions, to be used together with or even instead of the traditional methods.

This paper discusses the pros and cons of the traditional methods and presents the concept of logging from educational point of view. Three major uses for logging are presented: 1) using logs (files) as a teacher's reference to evaluate learning process, 2) using logging (memory) to interactively guide users in multimedia, self-study CBE applications, and 3) using logs (files and memory) to give users a possibility to examine their own progress and automatically generate reports in multimedia-based, self-study CBE. All three cases are illustrated with examples from CBE applications.

As a special case of interest, WWW and its log system are discussed. All current WWW servers include a logging mechanism and this information could be used for evaluating WWW-based education. A standard WWW server log system is thoroughly discussed and evaluated what can and what can't be analyzed from that data. Methods such as user authentication or HTTP cookie techniques and their use for the identification of users are examined. Furthermore, it is shown how logging can be utilized for interactive user control in an advanced WWW-based education system.

2. EVALUATION OF LEARNING

The evaluation of learning is a difficult but extremely important part of developing courses and teaching methods. The motivating effect is obvious. But equally or even more important than it is to a teacher, it is to a student to evaluate and measure their progress. Only by feedback students are able to understand the quantity or quality of their learning. However, the question of learning evaluation is far from being simple: How to measure if a student has learned something and if so, what would it be? At the very end, the question is not of finding a suitable criteria but a philosophical one: What is learning?

A wide variety of methods for evaluation, assessment, comparison etc. of learning has been presented in the literature, for both course and curriculum level. It is not within the scope of this paper to cover these all, but two common course-level methods are discussed.

Perhaps the most traditional method of evaluating learning at course level is an examination. In this method, the teacher creates a test and students take it. After that, the teacher evaluates answers as correct, incorrect or something in between

0-7803-5643-8/99/$10.00 © 1999 IEEE

and finally describes the results as numbers. The student can then compare his/her results to an absolute scale or to results of others and thus evaluate their performance. The teacher, on the other hand, takes the average results and uses them for an estimate of the overall learning of the students taking the examination. But can we really say that these results describe learning? If the test was cleverly designed, probably yes, at least to some extent, but in general this method is far too dependent of the test itself, the course, the goals, the material etc. Often the situation is that the teacher has to adjust the overall scale leaving nothing but a hunch of true evaluation of learning. Still, there is no need to abandon examinations without carefully considerating it first. Despite their problems, examinations are very often the only practical alternative.

Another commonly used method with courses is a feedback form. Students are given a form and they are expected to use it to evaluate their own learning and/or course with some kind of predefined scale or by short written answers. It is customary that when a form is used together with an examination, they do not affect course grading and usually there is a possibility to fulfill and return the form anonymously. Unfortunately, often the feedback forms have been designed only from the viewpoint of a teacher and the benefit of fulfilling one remains unclear for a student. There are exceptions, of course, and a well designed feedback form can give both students and teachers equally useful information for learning evaluation.

There are also more advanced methods such as course assignments, laboratory experiments etc. that give a student a possibility to apply the things that they have learned. The course assignments certainly give a more accurate estimate of the learning, but for teachers, it is sometimes impossible to organize such in practice. Combinations of the methods described can also be beneficial, see for example [2].

3. THE CONCEPT OF LOGGING

In computer science, logging is normally considered as saving user or debug information to a file. It is a very much standard technique to 1) calculate the usage of the software and 2) trace back possible problem situations. In client-server applications, logging is a very solid and natural part of the system, because much of the log information has to be exchanged between the client and the server anyway. Therefore, for example all common Internet services like WWW, ftp, email etc. include a log system.

However, in this paper, the concept of logging should be understood in a more general way: as exchanging and storing information. The information can then be saved into a file or only kept in a memory while an application is used, depending on what it is used for. What information is in this sense logged also depends on a particular application even though common parameters definitely exist. Thus the concept of logging should actually be thought of as including user related *metadata* with the original information.

It should be noted that the concept of logging may include restrictions by law. In some countries and in certain type of services or applications, logging is regarded obligatory by laws and in some countries, even the same ones, the laws prohibit any logging that may expose or relate to persons' true identity.

Therefore, whenever a log system that authenticates users is used, the authentication mechanism should not relate to person's true identity but primarily identify and distinguish the users from each other and secondarily identify they as individual persons. In the field of education this in fact could be a problem, think of for example electronic examinations. Still, quite often it is only desired to distinguish the students as individual students rather than true, individual persons in legal sence.

4. LOGS AS TEACHER'S REFERENCE

In 1994-1995 a tutorial level CBE application, "Introduction to Signal Processing" [1] was developed at the Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing. The application is a multimedia self-study package, but it has been used as a part of the course "Fundamentals of Acoustics I" which is the very first acoustics course for students. "Introduction to Signal Processing" is an Apple Macintosh application and students used the program in a dedicated workstation located in the laboratory.

The application consists of "pages" (windows) with either static information, interactive demonstrations or questions, and "chapters", collections of pages that cover a certain topic. The students were allowed to go through these topics in any order they wished, but there was also a default path that could be followed. From the very beginning of the development project it was clear that information was needed to evaluate the success and usage of this CBE application. From teacher's point of view, the questions of interest included:

- How much time would the students spend with it?
- How much material would they go through?
- In which order would the students study subtopics?
- Are there pages/topics that students would spend a lot more time on than others?
- Are there pages/topics that students would skip?
- What is the average time students spend on the application or page?

- Would the students take the default path or would they use the possibility to study the topics in some other order?

In order to get answers to these questions, a log system was implemented into the application. The following information was logged:

- **login:** `course name, username, student number, date, time`
- **page open:** `symbol name of the page, universal time`
- **page close:** `symbol name of the page, universal time`
- **logout:** `time`

The information was saved as text and later imported to a spreadsheet program for detailed analysis. After using the application, students were also encouraged to fill in a feedback form that contained questions related to both the contents and the implementation.

The use of "Introduction to Signal Processing" was essentially only a small part of a regular course that included an examination as the primary course evaluation method. The application was not intended to replace any other course or course material but to provide additional material to the regular acoustics course. The CBE-material was not included in the course examination and that is why the evaluation had to be based on some other method. Therefore, in order to estimate learning with the CBE-application, the logs were analyzed and two criteria chosen as learning measures: the overall time a student spent with the program and the amount of pages student browsed. These criteria were given minimum values based on the contents. Students who spent more time and went through more pages than the minimum values, were given additional points in the examination (5 % of the maximum). Feedback forms (paper) were used as a backup method to ensure log evaluation results especially in possible problem situations.

The CBE-application being a separate, self-study experiment without any relation to previous or parallel material prevents a thorough, comparative analysis to be included. However, the overall results from the first group of students (approx. 80) are as follows: A comparison of the results of the feedback form analysis and the log analysis clearly showed that students who really concentrated and spent time with the CBE-application, actually learned something from it and that the chosen criteria would be suitable for learning evaluation. Additional log analysis results like average time spent on a page showed that students spent more time on pages that contained interactive material but otherwise no major differences between the pages could be found. However, this kind of result was rather expected in the original design goals for this CBE-application.

5. LOGS AS INTERACTIVE GUIDES

The "Introduction to Signal Processing" contained also another logging mechanism: it kept the information of visited pages in memory. Because the pages were divided into topic-specific groups (chapters), it was possible to keep track of the chapters and automatically suggest users with topics that they had not covered yet. Furthermore, the visited pages were also attached to navigation history thus allowing students to refer to any page they had already visited.

In self-study, multimedia CBE, such behavior is very useful because it allows the application itself to suggest the student what topics he/she should study next and, furthermore, relationships between topics to be in control of the CBE developer, that is, the teacher.

"Introduction to Signal Processing" utilized this behavior by containing an interactive experiment that students were allowed to explore only after all topics had been covered. The experiment itself combined information from the other topics. In general, this kind of control is somewhat essential in self-study CBE to make sure students have the necessary background information before moving on to more advanced material. It is also useful in case of electronic examinations.

6. LOGS AS STUDENT'S REFERENCE

If a CBE-application contains lots of material or the material consists of different topics related to each other, it is necessary to give student a possibility to estimate how much they have done already and how much they still have to do in order to cover all or certain amount of the material. One useful tool for that is the previously mentioned navigation history, but "Introduction to Signal Processing" contained also another tool for this: the status window.

After logging in, the student could, at any stage, choose "Status" from a menu. The status window then presented information such as the topics already covered, the topics to do and total time spent so far. It could have also included complex estimates like a time estimate how long it would take for a student to study everything (based on the average "speed") etc. but such were not implemented in "Introduction to Signal Processing".

From the point of view of self-evaluation of learning, the status tool is very useful and at its best quite accurate, depending of course on the particular CBE-application.

Another interesting idea is to use logging for automatically generating reports. An example of this type of use was included in a "Psychoacoustics" CBE-package [1]. The application contained theory and demonstrations of psychoacoustical phenomena in the form of listening tests. The students could for instance measure their own frequency masking curves by performing a series of listening tests. The automatically generated report in this case was student's own frequency masking curve, which he/she could easily compare with theoretical results.

The automatic report generating is a useful concept in every CBE application that involves measurements (real or simulated) or test/question-like interactivity.

The students can also be given access to their own log files afterwards. Analyzing the logs by themselves would easily point out issues and topics that were not thoroughly examined and estimate what topics should be given further notice. Basically, the analysis could be similar to the one teachers perform, but in this case, it would only evaluate the learning process of an individual student.

7. WWW LOG SYSTEM

In the last few years, the WWW (World Wide Web) has become perhaps the most important tool for CBE. From the very beginning, WWW servers (*httpd*) have included a logging mechanism as well. Like many other Internet services, the httpd log system has been designed mainly for "hit-rate" type statistical analysis or debugging purposes, but in fact it is one of the most flexible and well-designed log systems around.

To be precise, the WWW log system is a sum of three components: 1) the server software, 2) the client software and 3) the HTTP protocol. The information that is written to the log depends on the server software and also the parameters that are sent by the browser. The information exchange is done within the capabilities of the HTTP protocol [4].

A typical httpd log contains (example taken from the documentation of one of the most popular WWW-servers, *apache* (`http://www.apache.org/`)):

host ident authuser date request status bytes
where the tokens are

- **host** The fully-qualified domain name of the client, or its IP number if the name is not available.
- **ident** If IdentityCheck is enabled and the client machine runs identd, then this is the identity information reported by the client.
- **authuser** If the request was for a password protected document, then this is the userid used in the request.
- **date** The date and time of the request, in format day/month/year:hour:minute:second zone
- **request** The request line (url) from the client, enclosed in double quotes (").
- **status** The three digit status code returned to the client.
- **bytes** The number of bytes in the object returned to the client, not including any headers.

If a token does not have a value then it is represented by a hyphen (-).

The above is called Common Log Format (CLF) and the information is written to so called TransferLog. Apache server also has a possibility to use customized log formats with a number of additional keywords. However, the CLF as such is adequate for most purposes except for two particular tokens, Referer and User-Agent. The Referer is the url to the page from where the user came from (via HTML-link) and User-Agent, an id-string of the browser software sending the request. Apache server can be configured to store these directly to TransferLog with custom log commands "%{Referer}i" and "%{User-Agent}i". Errors are usually written to a separate ErrorLog with similar information + possibly some details of the error itself.

The log systems in other WWW-servers vary both in format and in customisability but the differences are quite small.

On a busy server the log files easily become very big and thus difficult to handle. Hence there is a number of specialized WWW log analysis tools available, for instance *analog* (`http://www.statslab.cam.ac.uk/~sret1/analog/`) or *WUsage* (`http://www.boutell.com/wusage/`). The most advanced ones are very fast and customizable in terms of analysis parameters and input or output format, but usually they provide only basic statistical measures such as hit rates etc.

8. LEARNING EVALUATION WITH WWW LOG SYSTEM

From the point of view of learning evaluation, it is important to notice that the basic logging makes it somewhat impossible to trace back individual users' requests. A standard solution is to use authentication to restrict access to the contents. Most WWW servers and browsers support authentication and in its simplest form, user authentication is easy to implement. Usually authentication involves also account management, preferably as automatic as possible, which is not simple to implement (not being very difficult either, though). A good management system would allow students (or anyone) to register on-line and create the necessary account immediately as well as methods for taking care of changing password or removing account. If user authentication is used, it is possible to do off-line evaluation and analysis similar to one presented in section 4.

Another disadvantage of the basic WWW log system is that the information is written only on a request basis. To put it in other words: it doesn't remember anything. In order to use logging for purposes such as interactive guidance or on-line evaluation, more advanced techniques are needed. Two basic solutions are available: 1) producing the content dynamically (on the server) so that appropriate metadata can be included as for example hidden input fields etc. and 2) HTTP Cookies [5] that are small information fragments traveling with HTTP headers.

The former method, producing contents dynamically supersedes the latter, cookies, in various aspects. Perhaps the most important one is reliability: not all browsers support cookies at all and, even if they do, the user can deliberately deny their use. Cookies are also very limited in their capabilities of carrying information, for instance, the size of a cookie can not exceed 1024 bytes. Still, cookies are very common and used especially for tracking timing or users' paths when browsing through a WWW site.

Producing the contents dynamically allows the developer (teacher) to have all the log information available anytime and furthermore, to specify what metadata and how it is included in the real contents. There are drawbacks too: it is much more difficult for a developer to create dynamic contents than static WWW pages and it always takes more computational resources from the server.

A wide variety of implementation techniques can be used: CGI (Common Gateway Interface), apache module-techniques (mod_perl, mod_php, etc.), Java Servlets and so on. Also, in order to manage large amount of data, that is CBE-material as WWW pages, a database of some kind is needed too. SQL or text databases seem to be the most popular choices at the moment.

It should be noticed that, in principle, client-side interactivity techniques such as Java Applets or scripting languages **do not provide log information** from the processing itself. This is due to the fact that the WWW log system is based on interaction between client and server and client-side processing basically does not involve server for other than providing the code itself.

9. EXAMPLE ANALYSIS

Consider the following (simplified) log data:

```
IP1 user1 time1 url1 code size ref0 brwsr
IP1 user1 time2 url2 code size ref1 brwsr
IP2 user2 time3 url1 code size ref0 brwsr
IP1 user1 time4 url3 code size ref2 brwsr
IP2 user2 time5 url2 code size ref1 brwsr
IP1 user1 time6 url4 code size ref3 brwsr
IP1 user1 time7 url5 code size ref4 brwsr
IP2 user2 time8 url7 code size ref2 brwsr
IP1 user1 time9 url6 code size ref5 brwsr
```

From that data, immidiately can be calculated:

- What pages have `user1` and `user2` visited?
- What path have they followed?
- What is the amount of time they have spent on page/pages? (see note below)
- Have they connected from local network, i.e., used for example computers provided by school/university (IP-address)?
- What WWW-browser software have they used?

As themselves, the results are not very descriptive, but if these measures and urls are mapped with respect to the CBE material they present, they can be of great value in learning evaluation. For example, we can observe that `user1` has studied 6 pages (topics), spent certain amount of time on a page/pages, has followed a linear path, has used a certain WWW-browser and computer connected to university local network. `User2` on the other hand has studied 3 pages in certain time, has jumped to a new topic from page 2 and has used university modem connections.

Furthermore, if there is an interactive task on page 4 that requires applying information from pages 1-3, it would indicate that `user1` has indeed managed to overcome the task and thus *learned* that topic.

It should be noticed that because the WWW log-system is based on request processing, the timing is not accurate in certain cases. For instance, users may jump to some other website by typing in the address directly, later returning back to the original page and the jump would not be seen in the logs. However, in many cases it is reasonable to assume that users study educational material with concentration.

Another important feature of the WWW log-system is that the logging is always performed on the particular WWW server containing the material. This means that if material is distributed over a range of servers, the log data needs to be combined before analysis or advanced solutions like redirecting needs to be used.

The final notice regrading this example is that if the CBE material only contains basic or reference material, but not any interactivity or tasks that require applying information, the log-based evaluation gives metadata type information that is most useful when combined with other evaluation methods. On the other hand, in case of self-study, interactive Web-based education, the log analysis can provide a very good picture of the student's learning process and thus be used as an evaluation measure by itself.

10. ADVANCED WWW STRUCTURE

To use logging for true interactive guidance, basically every single web page has to be turned into a call for a program in the server. One solution (see Figure 1) for this is to use a **User Agent** to filter traffic between client and server [3]. (Note: User Agent here is completely different from the one mentioned in section 7).

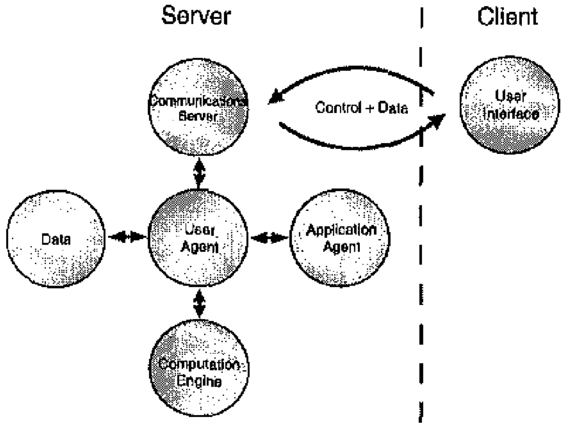

Figure 1: *An advanced WWW architecture*

In this structure, a User Agent represents the real user and takes care of all the requests made to the server. It responds to these requests with appropriate data according to specifications made by the developer. Thus it represents the teacher as well and the appropriate term in this case would be a *Double Agent.*

There are a number of ways to implement such architecture, but these are very sophisticated and far beyond the scope of this paper. In principle, this type of structure allows logging to be applied in both off-line evaluation and on-line interactive control.

11. CONCLUSIONS

Compared to traditional methods, log-based evaluation has certain advantages. Particularly, logs will include real information of the learning process itself, which is quite impossible to expose with traditional methods like written examinations or assessment forms. The concept of logging also supports the *cognitive-constructive* theory of learning [6], provided that the CBE material itself was constructed in this manner.

Logging also adds new possibilities to student self-evaluation, which in university level studies and student-oriented learning is very important. Furthermore, log-based evaluation includes a strong dependency on the contents, same as examinations and feedback forms. When used in combination with examinations or feedback forms, logging can, however, provide a more accurate evaluation of learning.

Furthermore, logging does not have to be restricted to user and transaction related data but more content oriented parameters could be used as well. For example content descriptions or content level would be parameters of interest in learning evaluation. Naturally, these would be case-dependent parameters and certainly more difficult to implement.

Besides evaluation of learning, logging is a very useful concept in other related areas as well. An example could be psychological testing, where it is sometimes even more important to know *how* a person takes the test than *what* he/she answered. Creating such test in an electronic form rather than traditional paper version, and including logging in the system, would result in an extended set of material and, perhaps, increased accuracy for psychological evaluation.

12. REFERENCES

[1] Karjalainen M., Rahkila M., "Learning signal processing concepts and psychoacoustics in the QuickSig DSP environment", in *Proc. 1995 IEEE Int. Conf. Acoust., Speech, and Sig. Proc. (ICASSP'95),* Detroit, Michigan, USA, vol 2, pp. 1125-1128, May 9-12, 1995. `http://www.acoustics.hut.fi/~mara/publications/icassp95/`

[2] Wiezel, A., "Measuring the success of virtual tutoring", in *Proc. 1998 IEEE Frontiers in Education (FIE'98),* Tempe, Arizona, USA, Nov 12-15, 1998. `http://fairway.ecn.purdue.edu/~fie/fie98/papers/1357.pdf`

[3] Rahkila M., Karjalainen M., "An Experimental Architecture for Interactive Web-based DSP education," in *Proc. 1998 IEEE Int. Conf. Acoust., Speech, and Sig. Proc. (ICASSP'98),* Seattle, Washington, USA, May 12-15, 1998. pp. 1857-1860. `http://www.acoustics.hut.fi/~mara/publications/icassp98/`

[4] Fielding, R., et al., "Hypertext Transfer Protocol – HTTP/1.1", RFC 2068, January 1997. `ftp://ftp.isi.edu/in-notes/rfc2068.txt`

[5] Kristol, D., Montulli, L., "HTTP State Management Mechanism", RFC 2109, February 1997. `ftp://ftp.isi.edu/in-notes/rfc2109.txt`

[6] Bruner, J. S., "Toward a Theory of Instruction", Harvard University Press, Cambridge, MA, USA, 1966.