

Evaluation of Usage Patterns for Web-based Educational Systems using Web Mining

Dave Donnellan,
School of Computer Applications
Dublin City University
Dublin 9
Ireland
daviddonnellan@eircom.net

Claus Pahl
School of Computer Applications
Dublin City University
Dublin 9
Ireland
cpahl@compapp.dcu.ie

Abstract

Virtual courses often separate teacher and student physically from one another, resulting in less direct feedback. The evaluation of virtual courses and other computer-supported educational systems is therefore of major importance in order to monitor student progress, guarantee the quality of the course and enhance the learning experience for the student. We present a technique for the usage evaluation of Web-based educational systems focussing on behavioural analysis, which is based on Web mining technologies. Sequential patterns are extracted from Web access logs and compared to expected behaviour.

1. Motivation

The evaluation of computer-supported educational systems is of major importance (Britain, Liber 1999), (IBM 2001), (Smeaton, Keogh 1999). Often the deployment of these systems replaces the teacher or tutor, thus there is little or no contact between student and teacher. The teacher receives less direct feedback. In order to assess the quality of the course material and monitor the students, evaluation becomes essential (Turk 2000).

We will address the evaluation of a multi-service integrated Web-based virtual course here. The approach followed is server-side evaluation. This will not capture all student activities, but will – as we will see – capture the essential ones. An advantage is that this form of analysis allows a constant monitoring of all students, no additional equipment is needed. We will base the evaluation on the Web access log, which records all Web page accesses by users. Our objective is to evaluate the student behaviour, i.e. to determine the student's navigation behaviour and their use of interactive tools and features integrated into the Web-based course system.

2. Integrated Virtual Courses

Our course is Web-based, i.e. uses an open standard as the basic platform (Smeaton, Crimmins 1997), (Smeaton, Keogh 1999). This guarantees the usability of the course without the need to install any other software at the student's side except a Web browser with standard plug-ins. Our course – an introduction to databases - supports several learning modes – attending lectures, tutorials or labs – through an integration of different educational services. Interactivity is a crucial element in a virtual course, since it allows engaging the student. Figure 1 show an interactive service part of our virtual database course.

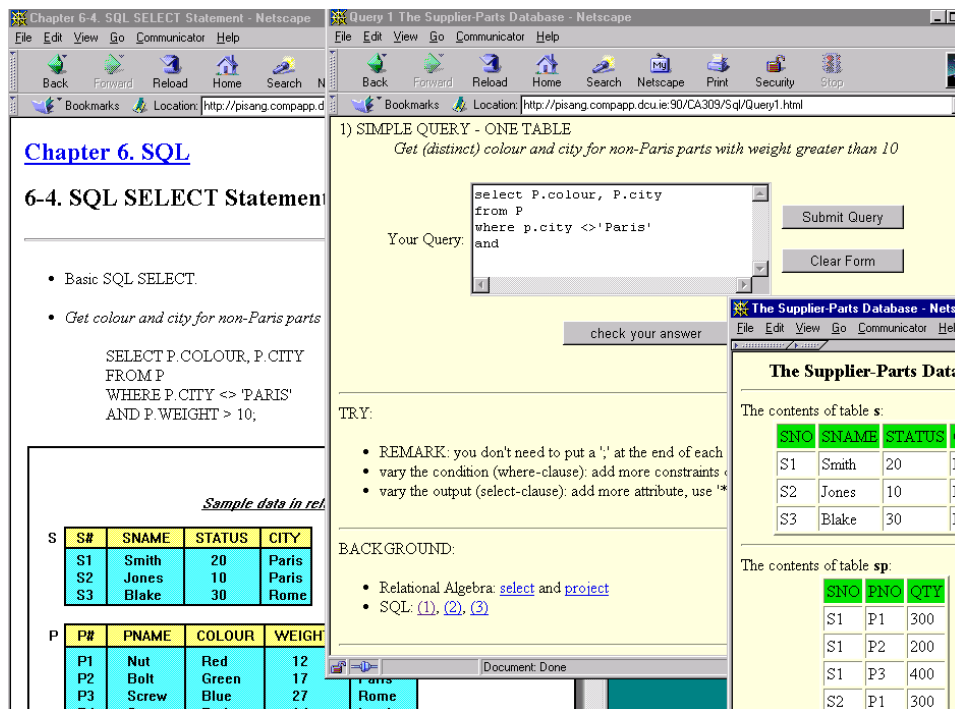


Figure 1. Interactive SQL Service

The student can type a solution attempt into the text field in the window in the middle and submit the attempt to a remote database server, which executes the student query and returns a result – a table containing records in this case, see Figure 2. The window on the left shows part of the lectures. The window on the right shows some tables from the database on which the tutorial service works. This screen shot shows the potential of using Internet-technologies for educational systems – several activities such as lectures, tutorials and accessing (dynamic) background material can be combined at the same time.

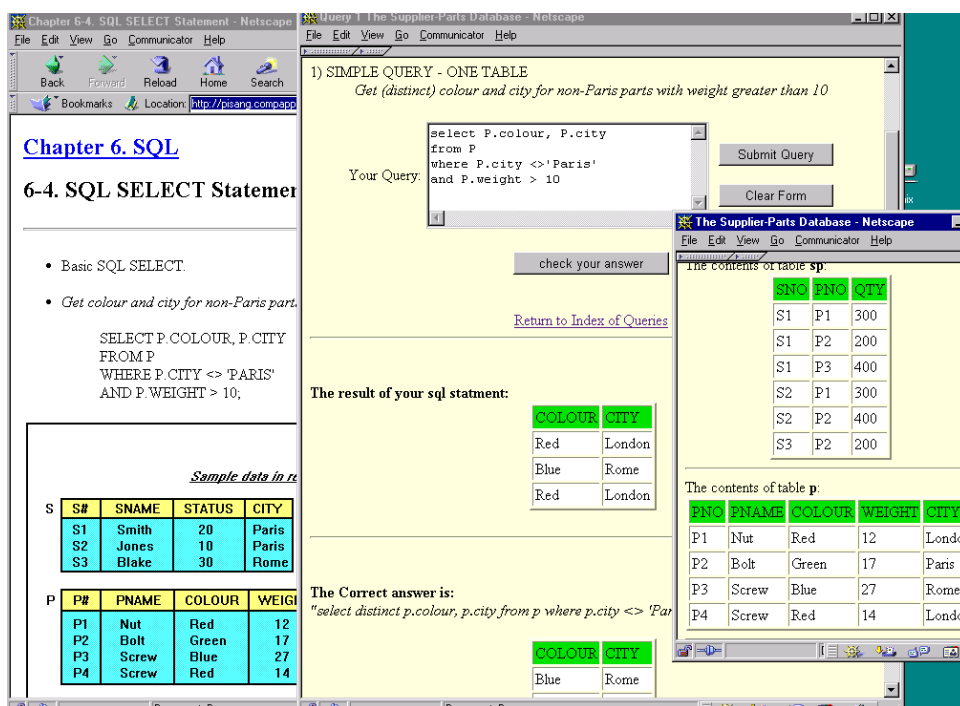


Figure 2. Execution of Interactive SQL Service

Web-based, or other virtual courses, offer a new potential for the design of courses. They allow us to overcome some of the constraints that limit the traditional delivery of courses. Traditionally, lectures, tutorials and labs are separated from each other, happening at different times and at different places. Virtual courses however allow a teacher to design a course with a close integration of these different modes of learning. In this situation, the description of expected student behaviour and the evaluation of the actual behaviour is highly important. We will come back to this issue in Section 4.

We will use Web mining to evaluate student behaviour in virtual courses. This will be based on standard Web mining techniques (Agrawal, Srikant 1995), but would like to point out here that educational systems differ from commercial systems and students differ from visitors of a commercial Web site (Britain, Liber 1999), (Lennon 1997). The student's goal is a long-termed one: learning. Students usually spent a relatively long time in the system, and they will repeatedly visit the site. Adequate mechanisms need to be in place to support the teacher in planning and developing such complex behavioural patterns and to evaluate this behaviour (De Bra, Houben, Kornatzky 1994), (Grønbaek, Trigg 1999), (Lowe, Hall 1999), (Stutt, Motta 1998). The process of learning has to be described and analysed.

3. Web Mining

Data mining is defined as the discovery and extraction of information from a database. Web mining is data mining for the Web, i.e. data available in Web-based systems is analysed. The database here is the access log generated by a Web server. It records each single access request for a document, which is denoted by a URL. These URLs can denote classical HTML-pages, but can also be images or executable documents such as scripts (e.g. Perl) or programs (e.g. Java servlets). Each entry usually contains the following fields:

- Client: IP address
- Ident: requestor ID (rarely used)
- User: (authenticated) user name
- Date: date of request
- Method: HTTP GET or POST
- Request: URL of requested document
- Protocol: HTTP version
- Status: success indicator (200 is success)
- Bytes: bytes requested/transferred

Not all fields might be available, e.g. the Ident or User information are often not available. The following is an example of three requests:

```
136.206.18.130 - rkyne.ca3 [08/Nov/2000:11:38:15 +0000]
"GET /CA309/ch5-ov.html HTTP/1.0" 200 43
136.206.18.14 - lgavin.ca3 [08/Nov/2000:11:38:18 +0000] "GET
/CA309/ch3-2c.html HTTP/1.1" 200 2048
136.206.18.16 - bahern.ca3 [08/Nov/2000:11:38:25 +0000] "GET
/CA309/Asgn.html HTTP/1.1" 200 2018
```

The objective is to extract sequential patterns from the log file. We divide the log into sessions. A *session* is defined as a sequence $P = \langle P_1, \dots, P_n \rangle$ of requests P_i from one user for a period of time in which the user is active. A request P_i is in our case a page request, i.e. a URL. Inactivity for a period of about 30 minutes indicates the end of a session. A session reflects a period of active usage of a particular student. A sequence $P = \langle P_1, \dots, P_n \rangle$ is contained in a sequence $Q = \langle Q_1, \dots, Q_m \rangle$ if $P_1 = Q_{i_1}, \dots, P_n = Q_{i_n}$ such that $i_1 < \dots < i_n$. This means essentially that each element of the P-sequence can be found in the Q-sequence, and additionally that the P-elements appear in the Q-sequence in the same order in which they appear in the P-sequence. The idea of containment is necessary to filter out irrelevant activities – students might go back to previous pages, lookup other pages, even leave the system temporarily. The P-sequence is a candidate pattern. Elements of Q that are not in P are those irrelevant URL-requests. A sequence is called maximal if it is not contained in any other sequence. Maximality allows us to get rid of shorter sequences that are contained in others. These would not provide any additional information.

In order to find out what patterns students follow, we need to look at the number of students that follow a particular sequence in a session. A student *supports* a sequence if the sequence can be found in any of that student's sessions. The *support for a sequence* is defined as the fraction of total students that support this sequence. A *sequential pattern* is a maximal sequence that has a certain minimum support. The choice of the minimum depends on the system and the objectives of the analysis. It needs to be determined heuristically. A high minimal support will only reveal patterns that are supported by a vast majority of students. A low minimum will show more patterns, which in extreme cases reflect more the behaviour of individual students than that of the whole group. The site structure, and in particular the degree of choice has an influence on the best choice of the threshold. For systems with a high degree of choice, the threshold should be low in order to detect common behaviour.

The following shows the support for a short sequence of URLs – leading from the course home page via the table of contents to an overview page of Chapter 6. The high support can be explained by the fact that this chapter covers the main practical elements (which are relevant for the continuous assessment and the final exam).

```

/CA309/home.html
/CA309/toc.html
/CA309/ch6-ov.html    Support = 11.8%

```

The sequential patterns describe the actual behaviour of students on an abstract level. These will later on be compared with the expected behavioural patterns specified by the teacher or course developer – see Section 4.

Before we look at the implementation of these techniques and results obtained from an evaluation, we shall briefly address principle problems connected with this technique. Some techniques in Web- and Web browser-technologies cause problems here. One problem is that most browsers use caching to avoid the repeated download of documents. This is a client-side technology, and thus a request for a page in the cache is not logged at the server side - and will therefore not be part of the evaluation and might lead to erroneous results. This problem can be avoided by generating pages dynamically – a technique which is usually deployed for adaptive or XML-based systems. URLs will be expanded by a time stamp or a similar data item. Unfortunately, this idea solves one problem, but creates another. We are interested in

sequential patterns based on the original page URL's, but not on the extended ones. We have to introduce equivalence classes of URLs here on which the pattern analysis can take place.

4. Evaluation

Behavioural patterns are a design tool for teachers or course developers for Web-based virtual courses. A model of the course topology – the navigation infrastructure and the interactive elements integrated into dynamic pages – underlies the specification of behavioural patterns. A *behavioural pattern* is a path expression on the course topology. The following is an example:

$$\text{DBQuery1}^+ ; [\text{Check1}] ; \text{DBQuery2}^+ ; [\text{Check2}]$$

DBQuery1, Check1, etc, shall be URLs. This expression specifies that the student can repeatedly access page DBQuery1, (the $^+$ -operator) then might access Check1 (an option denoted by $[\dots]$), then repeatedly access DBQuery2 and finally might access Check2 (again an option). The semicolon denotes sequential composition. Overall, the control flow combinators for our path expression language are:

- iteration P^+ : the page P can be access any number of times, but at least once.
- option $[P]$: the page P might or might not be accessed.
- sequence $P ; Q$: the page Q will be accessed after page P.

It is important to note that we can see sequential patterns as path expressions.

The notation might be extended to include a parallel composition $P || Q$ which says that pages P and Q can be accessed concurrently – e.g. using two Web browser windows. We shall ignore this possibility here. However, we would like to point out that logging and evaluating multi-window activity is important, and will help to obtain a more accurate analysis of student behaviour.

We now need to compare a specification of expected behaviour in terms of path expressions and actual sequential patterns. An ordering relation shall indicate whether an actual sequential pattern satisfies a constraint formulated by a behavioural pattern. An *ordering* $S \leq T$ on path expressions compares actual and intended use and decides whether the actual use conforms with the intended use. So, $S \leq T$ means that pattern expression S satisfies T. We now present some rules that define this relation. Typically S will be a sequential pattern and T a behavioural pattern. The rules allow us to decide whether the sequential pattern satisfies the behavioural one. In the following, the letters S, T, U, X, and Y stand for path expressions. The expression ST means that S and T are concatenated, i.e. sequentially composed.

- $T^+ \leq T$ means that actual repetitions are allowed,
- $S \leq [S]$ means that the user can choose to access S, and
- $SU \leq S[T]U$ means that optional pages can be left out.

A mathematical property shall be noted: the relation \leq shall be reflexive, antisymmetric, and transitive, i.e. should form a partial ordering. A weaker variant of \leq can also be introduced: $STU \subseteq XY$ if $S \leq X$ and $U \leq Y$ which allows students to deviate for a while from the pattern. Deviation, choice and repetition in actual navigation sequences are important patterns for understanding the way students work with the system

The final calculation is the determination of the *support* for a *behavioural pattern*. The support is defined as the fraction of sequential patterns that support the behavioural pattern. A

few results of this evaluation shall be mentioned. Firstly, a large number of teacher-specified behavioural patterns are supported by sequential patterns. Examples are longer sequences through the lecture material (e.g. chapters of the material) or the repeated use of the interactive services. Secondly, some erratic behaviour is found in the sequential patterns. Being lost in the Web site could be an explanation, although students rarely mention the structure of the site and being lost when asked about the quality of the virtual course. Thirdly, organisational behaviour such as downloading notes, looking up news, results, etc. have more support than expected.

5. Implementation

A tool for pattern analysis is currently being implemented. The implementation of the analysis is divided into different phases:

1. The first phase cleans the log file. The log file contains all requests, including those for images contained in the pages. These are not relevant and are removed in order to allow a more efficient implementation of further phases.
2. The second phase deals with session extraction. The log file is reorganised into sessions. The file is now ordered by user IDs with sessions for each user ordered chronologically.
3. The next phase calculates the support for the sequences. Sequences with a minimum support are stored.
4. The maximal sequences in the set of sequences with minimum support have to be determined in the next step. These maximal sequences are the sequential patterns.
5. The sequential patterns are then compared with the teacher-specified behavioural patterns and the support of behavioural patterns is determined.

Efficiency is a key issue here. Log files for our virtual course system contain usually more than 200000 entries per course delivery. Inefficient implementations of in particular the support calculation will result in unacceptably long execution times. We refer to (Agrawal, Srikant 1995) for more details on efficient implementation of mining algorithms.

6. Conclusions

Pattern analysis based on Web mining technologies provides a useful evaluation tool. Student behaviour in educational Web sites can be determined and compared to expected behaviour. Insufficiencies of the technique, such as a not complete account of all student activities or technological problems such as caching, are compensated for by the possibility of monitoring course delivery at any time including all students. Since in our case all the functionality is located on the server side and therefore all activities involving these functions are logged, a server-side evaluation is a suitable approach.

The usage evaluation of Web-based systems can be classified into two dimensions: time and space. Usage in time addresses the frequency/regularity of usage, number of accesses, etc. Usage in space is concerned with usage patterns based on the course topology. Our evaluation is an evaluation in space. The combination with an evaluation in time can provide additional valuable information. Various tools that provide statistics on numbers of accesses, frequencies, etc. are available – see e.g. (Analog 2001).

We have looked at using Web mining for the purpose of behavioural analysis. However, the technology can be used to obtain a wider range of information. This could include monitoring individuals or groups of students, or the identification of weak students.

The technique presented here is limited to activities in one browser window. If several windows are used concurrently, then this behaviour would have to be recognised as a concurrent one. A corresponding operator for the path expression notation has been suggested. The importance of analysing multi-window activity is also stated by other authors, see e.g. (Badii, Murphy 2000). The extension of the analysis toward concurrent activities is planned for the future. In a first step, the notation for behavioural patterns should be extended to encompass parallel activities and corresponding rules to determine satisfaction. In a second step, the pattern analysis should be extended from sequential to parallel patterns.

References

Agrawal, R. and R. Srikant, R. (1995) *Mining Sequential Patterns*. In Proc. 11th International Conference on Data Engineering ICDE, Taipei, Taiwan.

Analog (2001) *Analog Logfile Analyser*. Web site: <http://www.analog.cx>.

Britain, S. and O. Liber, O. (1999): *A Framework for Pedagogical Evaluation of Virtual Learning Environments*, Report JTAP Programme, UK.

Badii, A. and Murphy, A. (2000): *Point-of-Click: Managed Mix of Explicit & Implicit Usability Evaluation with PopEval_MB & WebEval_AB*. Proc. 2nd EnCKompass Workshop, Dublin, Ireland.

De Bra, P. and Houben, G.-J. and Kornatzky, Y. (1994) *A Formal Approach to Analysing the Browsing Semantics of Hypertext*". *Proceedings CSN-94*, Utrecht, NL.

Grønbaek, K. and R.H. Trigg, R.H. (1999) *From Web to Workplace: Designing Open Hypermedia Systems*. MIT Press.

IBM (2001) *Education – Online Courses*. Web page: <http://www2.software.ibm.com/developer/education.nsf/java-onlinecourse-bytitle>.

Lowe, D. and Hall, W. (1999) *Hypermedia & and the Web - an Engineering Approach*. John Wiley & Sons.

Lennon, J.A. (1997) *Hypermedia Systems and Applications*. Springer-Verlag.

Smeaton, A.S. and Crimmins, F. (1997) *Virtual Lectures for Undergraduate Teaching*. In Proceedings ED-MEDIA'97 World Conference on Educational Multimedia and Hypermedia.

Smeaton, A.S. and Keogh, G (1999) *An Analysis of the Use of Virtual Delivery of Undergraduate Lectures*. *Computers & Education*, 32(1):83-94.

Stutt, A. and Motta, E. (1998) *Knowledge Modelling: an Organic Technology for the Knowledge Age*. In M. Eisenstadt and T. Vincent. *The Knowledge Web*. Kogan Page.

Turk, A. (2000) *A Contingency Approach to Designing Usability Evaluation Procedures for WWW Sites*. Proc. 2nd EnCKompass Workshop, Dublin, Ireland.