Paper 9:

WWW-based tools for capturing on-line communications as an educational resource

Irene Neilson & Calum Smeaton

Irene Neilson,
Department of Computer Science,
University of Liverpool,
Liverpool

Calum Smeaton,
Orbital Technologies,
Edinburgh

Summary

- The WWW provides a rich communications interface to the Internet. To date, however, this infrastructure has been used primarily as a as a vehicle for the presentation of information. Only recently has the need to provide systems which engage students with Web based learning materials been recognised. In this paper, we argue the case for yet another level of software tool development: tools for the capture, archiving and retrieval of those interactions which are taking place around the web based learning material. We suggest that the communications in which students engage about Web based learning materials are of considerable educational value. They are the embodiment of the feedback that good teachers use to become better teachers in the traditional classroom situation. In the on-line environment, tools can be developed which capture, structure and make such dialogues available for inspection and interrogation thus creating a learning resource which grows naturally over time. Issues critical to the effective design of such systems are explored and one system, Organik, is described in detail.

Introduction

- The WWW provides a rich communications interface to the Internet. To date, however, the WWW has primarily been used as a vehicle for the presentation of information to students. Increasingly it is recognised that, in order to maximise the value of this infrastructure to education, software tools are required that facilitate both the engagement of the student with the educational material and communication between the student and tutor about the presented material (Laurillard 1993)

A substantial body of research is thus focused on increasing student interaction with WWW based learning materials: through the provision of annotation systems; through the utilisation of electronic mail, computer conferencing tools and discussion group tools like Hypernews (McKendree and Mayes 1997) and through the integration of other applications, for example simulations (Neilson et al. 1996), into the hypermedia environment provided by the Web. Many software packages are now "Web aware". Word processing software offers automatic translation of documents into HTML. Web interfaces to libraries and help system databases are commonplace in Higher Education establishments. Previously proprietary information stores, such as that created by Lotus Notes, have become Web accessible through software extensions like the domino server.

In this paper, we suggest that there is yet another level of software tool development that can take place:- tools for the capture, archiving and retrieval of those interactions which are taking place around the learning. It is our argument that the communications that students engage in about learning materials, both with each other and with tutors, are of considerable educational value. They reflect the common misconceptions students have about a domain. They highlight weaknesses in the design of the material being presented to the students. They force the articulation of important points of practical relevance that are often ignored in formally structured teaching materials They are the embodiment of the feedback that good tutors use to become better tutors in the traditional classroom situation. In the on-line environment, however, that feedback can be captured, structured and made available for inspection and interrogation and, in the context of the WWW, made accessible from anywhere.

The educational value of the proposed system

- The potential value of such a system appears considerable. Students could learn vicariously from exploring this on-line resource in the same way that, in a conventional classroom context, they learn from hearing the questions raised by others or from observing other tutor-pupil interactions. Insights would be shared. In a distance learning context, confidence might be boosted by a student realising that they were not alone in their failure to immediately understand learning material. The system would provide a means through which students could become aware of their own level of understanding relative to other students. Students could act as each other's voices by articulating the question someone else was struggling to formulate. Such an on-line resource is also advantageous to the tutor. It could act as a filter of student demands on the tutor's time. The tutor need only address those queries for which an answer cannot be readily found in the system or which require a more extended reply than that already offered electronically. The tutor could also use the system to refine the original expository material that was presented to the students.

Such a system would also open up interesting possibilities of collaboration between different educational institutions. Given the distributed nature of the infrastructure provided by the WWW, the answering of student queries about learning material need not be restricted geographically. Tutors in different educational institutions could take responsibilityfor handling questions which fall within their particular domain of expertise. Communications between student and tutor could be captured regardless of the geographic location of either participant in the interaction. The system can thus be a multi-centre resource which evolves over time.

In sum, our focus in this paper is on the development of software tools that facilitate the reuse of the electronic communications which take place in the on-line learning environment provided by the WWW. As indicated above there is a prima facie case for the potential value of such tools (see also Mayes and Neilson 1995). Work on similar concepts in the field of organisational memory research, (Ackerman and Malone 1990, Ackerman 1994b, Ackerman 1994b, Ackerman and McDonald 1996, Eisenstadt, Shum, and Freeman 1996) and design rationale research, (Buckingham 1996), also offers support. Within the educational field, the importance of the dialogue surrounding learning material has been recognised in the work of Boder, (Boder 1992) and Goodyear and Steeples (Goodyear and Steeples 1992) in which peer based discussions were used to refine a knowledge base.Bulletin board style discussions have been used by Brailsford et al (Brailsford and Davies 1997) in the teaching of biology to refine the concept oriented database generated by in the computer assisted learning package, Knowledge Tree. None of these educational projects are however WWW based.

An architecture for the system

- There are various ways in which the infrastructure provided by the WWW might be used to realise a system for the capture, archiving and retrieval of those interactions which are taking place around the learning material.

  One method we explored was use of the Common Gateway Interface. The Common Gateway Interface (CGI) can be used to extend an HTTP server to allow browsers access to programs running on that server, such as programs for accessing a database (Gundavaram 1996). An interactive database of frequently asked questions about an educational topic can thus be readily created through the use of cgi-scripts and the database explored through a conventional WWW browser from anywhere in the world. A prototype system, the Answer Web (Smeaton and Neilson, 1995, Thomas et al, 1996) was successfully implemented using this architecture and field tested[1].

  However, CGI was designed to allow access to programs running on the server via a web client. While it does this very well, there is no infrastructure for supporting programs that are constantly running on the server or for easily passing information between scripts. Each interaction with a database is treated as an independent entity. To pass state information hidden fields have to be used. This makes the design of an interface for the easy administration of a database of questions and answers difficult to implement.

  The many problems presented by the current CGI architecture are widely recognised [http://www.fastcgi.com/kit/doc/fastcgi-whitepaper/]. Several solutions to these problems have been proposed, but these solutions are often dependent on particular platforms or vendors, e.g. Netscape NSAPI & Microsoft ISAPI.

  New developments in Web technologies such Servlets™, Java™, XML offer a more radical alternative to the CGI paradigm for the implementation of systems of this type. They permit the design of a system which is not simply the extension of a WWW server. This has the benefit of allowing other interfaces, such as e-mail and other graphical user interfaces (GUI) to drive the application, something that is not possible if the application is implemented as a web server extension. Accordingly, we have redesigned the proposed system to take advantage of these new developments. The new system is called Organik.

Organik.

Organik is a Java based application server. The core application can be accessed by a number of different protocols and Organik can contact other information sources and applications using a variety of protocols. Figure 1 show the overall architecture of the server.

1. Operational systems exist at Newcastle University Library and at Princeton University where the system is an integral aspect of the Computer Services on-line help system.

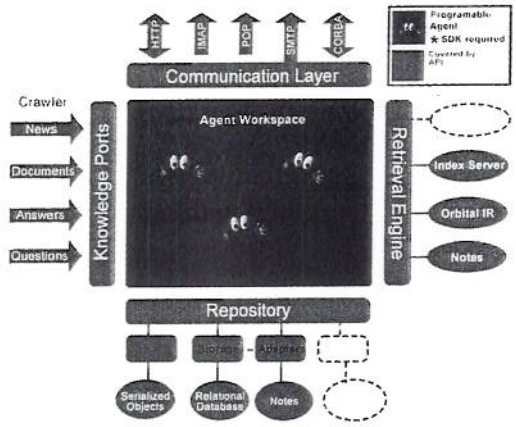

Figure 1: The architecture of Organik

Communications Layer

The communications layer provides the framework for allowing Organik to talk to other systems and other systems to talk to Organik using industry standard protocols. The integration of the communications protocols enables the software to offer greater assistance to tutors in the formulation of an answer to student' queries. Links to relevant, related information can be automatically provided. For example :-

HTTP

This allows the application to be interfaced with any standard Web browser. Thus when a tutor receives a query they can, if they so wish,readily search pre-defined databases (local or Internet based) with the query. Relevant content on these external resources can be pointed to when the question is answered.

NNTP

This allows the application to receive news feeds from any standard newsgroup server. Thus the database of questions and answers can be seeded from existing sources of Frequently Asked Questions about a topic such as Newsgroups. Problems that remain unanswered can also be sent to the newsgroups.

E-MAIL

This allows e-mail to be sent and received using standard protocols. Questions posed to tutors can thus be received and replied to through e-mail. Documents sent as attachments to these queries are automatically inserted and indexed in the database.

RMI/CORBA

This allows Organik servers to talk to each other and also achieves more efficient data transfer between the server and Java applets. In an educational context, this enables different specialist courses running the Organik software to chain their separate databases of question-answer dialogues together.

Repository

An object oriented database is provided by the system for the storage of question-answer dialogues between tutors and students. Alternatively, if a database of such queries already exists then this can be connected to the system instead. In this context, the system uses Java Database Connectivity (JDBC) to connect the database to the system and transfer the data. As JDBC is an industry standard which has been supported by virtually all major database manufacturers. This means that Organik can talk to a wide variety of commercial and public databases such as Sybase, Msql etc. It can also be integrated within an Intranet simply by adding it as another server on the network. As indicated, information on any accessible WWW server can be integrated into questions and answers within the system.

Retrieval Engine

Organik provides the ability to search disparate information sources such as Lotus Notes or Microsoft Index Server. A user can perform a query on one or more of these external information sources; the results are then combined using data fusion techniques.

Knowledge Ports

The knowledge ports are used to capture information from a variety of different resources whether it be questions asked by a user, documents retrieved by a web crawler or FAQ files. By using the knowledge ports, Organik can be pre-seeded with documents, discussions, course notes etc.

Agent Workspace

Organik uses server side agents to control the process of asking and answering questions. The behavior of the agents can modified to change the workflow of the system or the actions taken when a questions is asked. The agents are used to support knowledge discovery over an extended period of time. For example, if new information comes into Organik, the agents may notify a user that information relevant to a question or discussion has been discovered.

Making Organik an effective assistant

- To have value to the tutor, the Organik has to reduce the incidence of questions directed to the tutor or at least change the nature of these questions. Answers to simple, routine questions should be readily available from the system. Only when the student has a more complicated query or cannot find an answer to a query in the system, should a question be directed to the tutor. Ideally, with Organik's help, the tutor should discover that s/he has more time to engage in intellectually rich conversation with the student.

  This ideal scenario will only arise if students consistently check if an answer to a query is available in the database before posting it to the tutor. Students will only engage in such behaviour if they are rewarded for doing so ie if the system quickly and reliably retrieves information of relevance to their query or rapidly and reliably indicates that that information is not currently available. Thus, to be successful as an assistant, Organik needs to readily support student information retrieval strategies. It has to facilitate the rapid and accurate retrieval of information relevant to an information need. To support the incidental learning that can occur when browsing, Organik has also to provide the student with an attractive environment for browsing the database of question-answer dialogues.

  The design problem is how to achieve these two goals simultaneously.

  Classically any information retrieval problem is solved in one of two ways :- through data structuring mechanisms or through the employment of powerful search techniques. The relative merits of these approaches as applied to an educational context is described below. The solution adopted within Organik, the augmentation of traditional information retrieval techniques with relevance feedback from the user is a hybrid which we argue combines the best of each of these approaches.

Option 1: Data structuring

Data structuring mechanisms could be used to provide the student with an overview of the information within the database. Alternative views of the database could be provided: - a Tree Structure whereby questions are grouped by category, a listing of Questions chronologically or alphabetically,a listing of questions associated with a given keyword or indeed in the form of a traditional FAQ listing. Keyword searching can be provided through a form interface. The categories for the Tree Structure could be anything the tutor chose, including such models of the problem context as snapshots of simulation states.

Use of data structuring mechanisms to facilitate information retrieval however places ahigh administrative load on the tutor. As the size of the database increases, the effort expended by the tutor to maintain an appropriately structured front-end to the system grows considerably. Our field tests (and that of others, Ackerman 1994a) indicate that this administrative load is at best tolerated and at worst unacceptable, leading to the abandonment of the system. Such structured front-ends to a database of question-answer dialogues are also the product of the tutor rather than the student's mind. Before the student can find an answer to a question, s/he has to discover within what part of the tree structure

the question might lie i.e. to internalise the taxonomy created by the tutor. Yet the learning literature constantly emphasises the value of the student structuring information themselves to their own purpose (Mayes 1993). From a constructivist viewpoint, providing the student with a tutor defined interface to the resource of question-answer dialogues might inhibit rather than promote further learning.

Option 2: Augmented Information Retrieval

An alternative approach is to avoid data structuring altogether and to focus on the provision of effective information retrieval through use of good search algorithms. Pure search based systems tend to require a trade between recall and relevancy and as such can generate an unacceptably high incidence of false i.e. irrelevant returns. This problem can be reduced if the operation of the search tool is refined by relevance feedback from the user (Campbell 1996). This is the solution adopted within Organik.

The Hybrid Solution

In Organik, any query to the system, whether phrased in natural language sentence form or not, is treated as a sequence of keywords. The database is searched for items (question and answer dialogues, documents, etc.) which match the query terms and the latter returned to the student. The student then marks those items of particular relevance to the initial query. S/he now has the option of resubmitting the initial query along with those items marked as relevant to its resolution. The items marked as relevant are used internally within the system to definea new set ofquery terms which form the basis of extracting further information from the database.

Exploring the database of questions and answers in this manner has several advantages over conventional traditional query based systems. First, that the student is not asked to redefine their query in the language of the system, as is the case in most query based systems (Campbell 1996). Rather all the user has to make is a judgment of relevance. Secondly, and more importantly from an educational perspective, the method used optimises the possibility of incidental learning during the process of query refinement. Query refinement becomes similar to browsing. Query refinement proceeds by the student simply selecting or indicating documents of relevance among the set that is presented ie browsing through the information space of relevant documents. While gaining the advantages of browsing, namely that of promoting incidental learning, our system avoids one of the major limitations of browsing systems - namely that the information space being explored is statically defined. In Organik, the relevance feedback is used to dynamically reconfigure the user's information space. This makes Organik like a query based systems. In the latter, when an initial query is refined and resubmitted, the view of the information space returned to the user changes accordingly. Thus, the strategy adopted by Organik for information retrieval, in principle, combines the advantages of both browser and query based systems.

This process of query refinement is not only useful to the student in discovering information that is already on the database. A common problem tutors encounter in answering questions on-line, is that the context in which the question is asked is lost. However, if the student is able to notify the tutor of items within the database that they have already found relevant to their query (though not the total answer) then the probability of the tutor being able to provide a pertinent answer is increased.
Similarly, when a tutor receives a question, the system itself can also provide the tutor with a list of items in the database that are deemed to be related, thus facilitating the tutor's ability to respond to the question in a coherent and consistent manner.

Making effective use of distributed expertise through Organik

- Where a group of tutors are collaborating in the use of the system, it is critical that no-one tutor becomes overloaded with questions or receives questions that are not within their field of expertise. The mechanism that Organik implements for achieving this goal, is through dynamic tutor pro-

files which act as filters of student queries. These profiles are initially seeded by the expert with information which usefully indicates his/her area of expertise/ responsibility. These determine which queries a tutor receives. If no tutor profile matches a submitted question, then the question is sent to all the registered tutors. A gap list is also maintained of all the unanswered questions in the system. Any tutor can chose to answer these questions. Once a tutor has answered a question, an analysis is performed on the question and the profile of the expert's interests updated accordingly. Profiles are thus dynamic. They are automatically modified over time as the tutor answers new questions. The tutor can also fine tune the operation of a profile through hand-crafting its specification.

Work is currently taking place on the design of a work-flow component in Organik which will regulate the number of queries referred to an tutor in a given period of time. Student demands on tutor's time fluctuates considerable, often reaching a peak just before an assignment deadline. Some regulation of the volume of requests sent to a tutor is obviously required in order that a tutor does not become overburdened or a student deterred from using the system because of a slow response time.

Evaluation

- Whether the design of Organik meets the needs of tutors and students is being evaluated. As indicated, various design decisions have been taken which, while derived from theory, require empirical testing. Organik aims to facilitate effective and accurate information retrieval through combining traditional retrieval algorithms with relevance feedback from the user. The design of Organik has purposely avoided any attempt to model formally the structure of the question-answer dialogues that are retrieved from the database. Such formal modeling would have allowed the development of automatic analysis and indexing tools but the overheads involved in so doing were felt not to warrant the effort. Other projects, such as the Distributed Learning Dialogues project at the HCRC, Edinburgh are exploring the latter option.

The success, in practice, of a system such as Organik will depend also on many non-technical factors. The quality of answers to student queries, and thus the quality of the evolving database, depends on the participating tutors. Tutors, knowing that their answers to student queries will be stored and available world-wide, may be tempted to give answers that are more directed at impressing the world-wide audience than meeting the information need of the student. Students can be reluctant to pose queries as this displays their lack of understanding of a key concept to other students. Also, while the system is designed to grow naturally through use, the database needs to be initially "seeded" with questions and answers in order than student use is encouraged and the growth cycle initiated. While existing FAQ files (frequently asked questions) such as those found in Newsgroups can be imported into Organik, effort will still be required from the tutor if the initial seeding of the system is to be successful.

The significance of many of these non-technical issues to the performance of the system can be anticipated to decline in the future as networked learning becomes more common. Then the value of being able to access a resource, day or night, which deals with the common questions asked about the topic of interest will be recognised. The sheer volume of information flow will create a demand for knowledge management systems such as Organik which can filter and respond to queries before they reach the tutor's e-mail box.

References

Ackerman, M 1994a Augmenting the Organizational Memory: A Field Study of Answer Garden. In ACM Conference on Computer Supported Cooperative Work (CSCW'94):243-252.

Ackerman, MS 1994b Answer Garden: A Tool for Growing Organisational Memory. Cambridge MA: Massachusetts Institute of Technology.

Ackerman, MS and TW Malone 1990 Answer Garden: a tool for growing organisational memory. In Proceedings of the ACM Conference on Office Information Systems:31-39. New York: ACM Press.

Ackerman, MS and DW McDonald 1996 Answer Garden 2: Merging Organisational Memory with Collaborative Help. In ACM Conference on Computer Supported Co-operative Work (CSCW'96):97-105.

Boder, A 1992 The process of knowledge re-ification in human-human interaction. Journal of Computer Assisted Learning 8: 177-185.

Brailsford, T. and PMC. Davies 1997 Knowledge Tree: putting discourse into computer based learning. Alt-J 5 (1): 19-26.

Buckingham Shum, S 1996 Negotiating the Construction of Organisational memory Using Hypermedia Argument Space. In Workshop on Knowledge Media for Improving Organisational Expertise, 1st International Conference on Practical Aspects of Knowledge Management. Basel, Switzerland.

Campbell, I 1996 Applying Ostensive Functionalism in the place of Descriptive Proceduralism: "The query is dead". In British Computer Society Workshop on Information Retrieval and Human-Computer Interaction, ed. C.W. Johnston and M.D. Dunlop. University of Glasgow.

Eisenstadt, M, Buckingham Shum, S and A Freeman. 1996 KMi Stadium: Web based Audio/Visual Interaction as Reusable Organisational Expertise. In Workshop on Knowledge Media for Improving Organisational Expertise, 1st International Conference on Practical Aspects of Knowledge Management. Basel, Switzerland.

Goodyear, P and C Steeples. 1992. IT-based open learning: tasks and tools. International Journal of Computer Assisted Learning 8: 163-176:.

Gundavaram, S 1996 CGI Programming on the Internet: O'Reilly & Associates.

Laurillard, D 1993Rethinking University Teaching: Routledge.

Mayes, JT and IE Neilson 1995 Learning fron other people's dialogues: questions about computer-based answers. IFIP Transactions A -Computer Science and Technology 61: 31-47.

Mayes, T 1993 Commentary: the impact of cognitive theoryon the practice of courseware authoring. Journal of Computer Assisted Learning(9): 222-228.

McKendree, J and TJ Mayes 1997 The Vicarious Learner: investigating the benefits of observing peer dialogue. In Proceedings of Computer-Assisted Learning Conference (CAL'97). Exeter, UK.

Neilson, I, Thomas, R, Smeaton, C, Slater, A and G. Chand 1996 Education 2000: Implications of W3 Technology. Computers and Education 26 (1): 113-122.

Smeaton, C and I Neilson. 1995. The Answer Web. The Fourth International World Wide Web Conference (Poster Presentation), Boston, USA.

Thomas, R and I Neilson. 1996. Harnessing simulations in the service of education: The Interact Simulation Environment. Computers and Education .