

The Intelligent Helpdesk: Supporting Peer-Help in a University Course

Jim Greer, Gordon McCalla, John Cooke, Jason Collins, Vive Kumar,
Andrew Bishop, Julita Vassileva

ARIES Laboratory, Department of Computer Science, University of Saskatchewan,
Saskatoon, S7N 5A9 CANADA
Email: {greer,mccalla,jiv}@cs.usask.ca

Abstract. Universities, experiencing growths in student enrollment and reductions in operating budgets, are faced with the problem of providing adequate help resources for students. Help resources are needed at an institution-wide and also at a course-specific level, due to the limited time of instructors to provide help and answer questions. The Intelligent IntraNet Peer Help Desk provides an integration and application of previously developed ARIES Lab tools for peer help to university teaching. One of its components, CPR, provides a subject-oriented discussion forum and FAQ-list providing students with *electronic* help. Another component, PHelpS, suggests an appropriate peer to provide *human* help. In both cases it is *peer* help, since the help originates from students themselves. The selection of the appropriate help resource (electronic or human) is based on modelling student knowledge and on a conceptual model of the subject material.

1 Introduction

Universities are faced with the difficult problem of providing adequate help resources for their members, i.e. the staff, faculty and students. Help resources are needed at an institution-wide and also at a course-specific level, due to the limited time of instructors to provide help and answer questions. Computer technology offers several approaches to facilitating and providing the necessary personalized help resources that can be made available to a mass audience. By deploying IntraNets in universities different kinds of resources (lecture notes, exercises, quizzes, syllabi, etc.) can be made available on request to any student. There are numerous positive examples of implementing on-line course materials and discussion groups at other universities, for example, the Virtual-U Project [14], WebCT [8], and Quorum [3].

However, merely providing access to appropriate material via a network doesn't solve the problem of providing help. One way to decrease the load of teachers is to provide conditions for students to help each other. Peer help has many pedagogical advantages [12]. First, it promotes the socializing of students in the context of work and increases their motivation by giving social recognition for their knowledge and helpfulness. Second, peer help is deeply situated in a shared context and can therefore

provide a stronger learning experience for the person asking for help. Third, it is a way to make learning happen “just in time”, i.e. when the problem arises. Fourth, it promotes processes of self-explanation [5] and reflection in the helper, and in this way “reciprocal” learning takes place. Fifth, it is cost effective, since it uses the learners themselves as a teaching resource. And finally, it facilitates social interaction in a group of learners and helps to create knowledge-anchored personal relationships among them.

Peer-help happens naturally within small groups of learners. When the group of learners is too large or distributed, however, obstacles arise. In a university environment this is often the case. For example, the Department of Computer Science at the University of Saskatchewan offers an introductory service course in computer science (CMPT 100) for students from various faculties (commerce, arts & science, nursing, agriculture, etc.) which involves about 600 students per academic term. Learners who need help may not know whom to ask, since they may not be able to identify which student is knowledgeable; in fact they may not even know the other learners. In addition, many student questions relate to their assignments or laboratory activities, arising while they are working at their computers at home or in the labs. These factors combine to make peer help more challenging to provide. If the students did know a potential helper, they wouldn't know whether or not the helper was currently available, which means a loss of time and a loss of the immediate context in which the problem has arisen. Computer technology can be applied to help overcome some of these problems facing peer help. There are many Computer Supported Collaborative Work (CSCW) tools that facilitate communication among peers. However, they rarely ever provide personalized help on demand. We have been developing cognitive tools that support peer help in the context of university teaching in an individualized way and just in time. The tools can provide support in several ways:

- by providing an appropriate “discussion facility” where people can ask subject-related questions and give answers, discuss problems of common interest either synchronously (like a chat-facility), or asynchronously (like a newsgroup-facility);
- by finding appropriate peer helpers who are knowledgeable and likely to be able to answer a learner's question;
- by providing a shared workspace for helper and helpee where they can share the context of the problem;
- by helping the helper better understand the problem of the person asking for help and suggesting a pedagogically appropriate way of helping. For example, sometimes it is pedagogically better not to give any help, but to encourage the learner to try harder to solve the problem alone.

In this paper we focus on the first two ways of supporting peer help. The last two ways are the topics of other ongoing research projects in our lab and are not discussed here. The discussion facility, called the Cooperative Peer Response (CPR) system, provides a suite of WWW-based tools to facilitate cooperative learning, peer help, and expert help within a University course. The CPR discussion forum encourages peer help; it has been deployed in a number of University courses and has proven to be an effective learning support tool.

The second tool, called the Peer Help System (PHelpS) provides a facility for locating somewhere on the network a peer helper who is ready, willing, and able to provide help to a particular help request. A PHelpS prototype has been deployed in a distributed workplace environment [10] and preliminary experiments suggest that it is an effective training and performance support tool.

CPR and PHelpS, together with various related spin-off projects, have led to an attempt to integrate several cognitive tools into a new style of Intelligent IntraNet Peer Help-Desk facility. Such a Help-Desk draws together a variety of cognitive tools, particularly tools for peer help, into a comprehensive environment to support many styles of learning. The advantage of using an IntraNet instead of the Internet is that access is restricted to only those students attending the course, so that they can communicate with their peers and teachers. In this way students are protected from possibly disturbing comments made by occasional “visitors”, who are not involved in the course. This makes it easier to track down what students are doing and to collect information about the topics that were discussed. This information is used both for improving the peer-help facility and as feedback to teachers to adapt the course accordingly. It also eliminates some concerns about the privacy of user information gathered during students’ work with the system.

This paper outlines the design of the Intelligent IntraNet Help-Desk. It does not focus on the architecture of the proven peer-help tools (PHelpS and CPR) that act as its structural skeleton, since they are described in detail elsewhere [6, 10], but rather focuses on the integration of these tools in the context of university teaching.

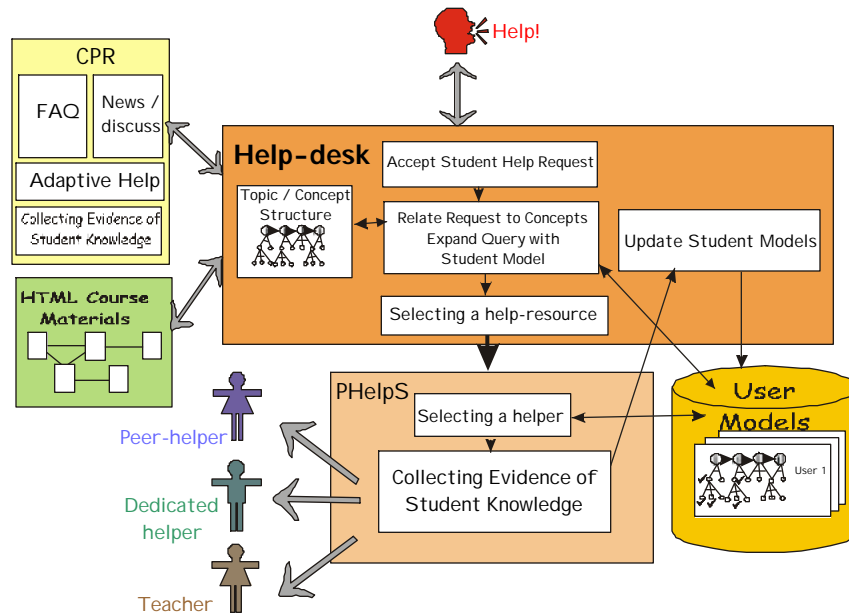


Fig. 1. Architecture of the Integrated Help-Desk

2 The Intelligent IntraNet Help-Desk: an Integration of Cognitive Tools

The help-desk provides individualized on-line multi-modal peer-help, as shown in Figure 1. In this integrated system, CPR acts as a medium for multiple users to communicate with one another in electronic and asynchronous ways and provides a resource for adaptive help. The role of PHelpS is to select an appropriate human helper when necessary and to facilitate the subsequent direct communication between the peers.

The Intelligent IntraNet Help-Desk accepts and interprets help-requests coming from students. Help requests can be made directly, or while browsing through the course materials, or while working with CPR. The Help-Desk locates an appropriate help-resource (e.g. related FAQs, a discussion thread related to the help-request, or web-pages addressing the concepts involved in the help request) or a knowledgeable peer-helper. Next we shall discuss the Help-Desk, which is the central component in Figure 1.

2.1 Knowledge Representation

Central to the intelligence of the HelpDesk is a knowledge base representing course topic and concept structure. This knowledge base is needed in order to “understand” the student’s help-request so as to match it to relevant articles and discussion threads in CPR and to find peers with the knowledge to deal with the request. We decided to use a two-layered knowledge representation, similar to the one used in BIP [2]. On the first, coarser “topic level”, the organization of the course and the activities taking place during it are represented (lectures, chapters or sub-chapters, exercises, labs, assignments, tests). The deeper concepts addressed by the topics taught in the course are represented on the second level. Several topics can relate to one concept (for example, several lectures, exercises or assignments may relate to various aspects of recursion). Similarly one topic may address several concepts (for example a lecture on web searching might refer to concepts such as “browsing’ or “search strategies”). The topic-concept structure is shown in Figure 2. We shall discuss it again in more detail later.

2.2 Student Modelling

One of the limitations of cognitive models (which sometimes discourage people from using them for instructional purposes) is the absence of elaborate analytic models of group learning. Therefore, it difficult if not impossible to fully apply existing cognitive theories of group learning in the construction of intelligent learning or help environments. Despite this, we believe that partial solutions should be sought. In the Help-Desk we have applied two simple and well-known representation techniques for student modelling: a numeric overlay and a profile of several general parameters. Two types of evidence are used to update the student models: direct and indirect. The

direct evidence comes from observed students' actions. During students' work with the web-based materials, with CPR and with PHelpS, the Help-Desk collects evidence about student knowledge and updates the individual student models. There are at least 10 sources of direct evidence about the student, which can be used: the history of studied topics in the course, the assignment marks of students, explicit testing on topics, the student's self-assessment (see the personal check-marks in Figure 4), the teacher's assessment (see the lecturer check-marks in Figure 4), votes in the newsgroup in CPR (about which answers are good), posted questions and answers in CPR, observation of CPR browsing (threads visited, participation), observation of browsing in the web-based course materials, feedback about the student, given by the peer helper, feedback about the peer-helper, given by the student.

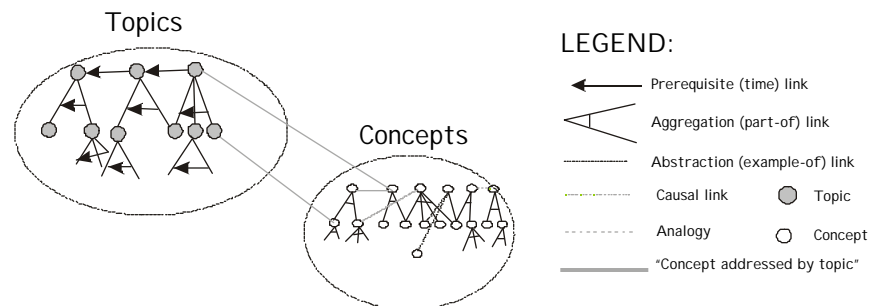


Fig. 2. The concept-topic structure

Every student model has two parts. The first part contains general information about the student, such as name, alias (if he / she wishes to have one for privacy reasons) and several parameters providing a general evaluation of the student. These include general helpfulness, general knowledgeability, overall willingness to help, and history data (e.g. how active he / she has been in general and how many times he / she has given help recently). These general parameters contribute to the calculation of the score for every student when a peer-helper has to be selected by PHelpS.

The kernel of the student model is a numeric overlay over the concept-topic structure. This overlay model provides information about how much the student knows about each concept and topic. The Help-Desk uses it in order to better "understand" the student's help-request, i.e. to place it in the right context (of the current topic) and to expand it eventually with related concepts, which are considered not known by the student (according to the student model).

The indirect evidence for updating the student models is gathered from directly observing evidence about knowledge of certain concepts and topics and propagating it to related concepts and topics. In order to make this clear we explain the concept-topic structure in more detail (see Figure 2). The topic structure includes prerequisite links (what *should be taught* before what) and temporal links (which topic *was actually taught* before which). Each topic can be broken down into sub-topics to decompose the structure further. In this way the topic structure is represented in an

aggregation hierarchy (AND/OR graph) of concepts ordered according to prerequisite links and temporal links.

Each of the topics may be connected to multiple concepts in the concept structure. Concepts represent the teaching goals, i.e. the elements of knowledge that the student has to develop as a result of taking the course. Concepts can be related to other concepts through various semantic links, including abstraction and aggregation (which in turn may use AND/OR clustering semantics), causal links, analogy links, and prerequisite links.

Why do we need a two-layered knowledge representation? The topic–structure provides a natural way to represent the position of a student in a course. However, it is not fine-grained enough to represent the differences in knowledge/ understanding among peers who are taking the same course. All it can state is the historical fact that the students have attended a certain lecture or have done a certain assignment. A finer distinction is needed in order to find capable peer-helpers, which reflects the knowledge of students and their ability or understanding. This distinction can only be found at the concept level, since every topic, sub-topic, assignment and test are related to a (set of) concept(s). Another advantage of maintaining a concept level is the possibility to take into account the various semantic links among concepts and to propagate knowledge values in user profiles through the concept network. In this way the system knows not only what the student has been taught, but by observing knowledge about one concept (e.g. good performance on a test), the system is also able to deduce that the student is likely to have knowledge on a related concept. In this way the knowledge value on one concept can be propagated to related concepts. As a result, a help -request addressing a given concept may be directed to a helper who hasn't exactly demonstrated knowledge on this concept, but is expected to have it because of having successfully mastered a closely related concept.

Knowledge propagation can happen among concepts or among topics and also between the two levels (from topics to concepts). Following the prerequisite links at the topic level, the system can conclude that if a student is currently working on topic B which has topic A as a prerequisite, the student has some knowledge of A (see Figure 3). Following temporal links, the system can conclude that if a student is currently working on topic C that was preceded by B, the student should have some knowledge of B.

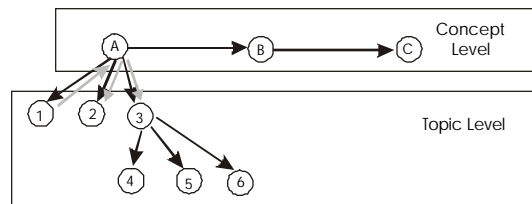


Fig. 3. Propagation of knowledge values among topics and concepts

Following the topic-concept links, the system can conclude, that if a student has learned topic A with related concepts 1, 2 and 3, the student has some knowledge of these concepts. Propagation in the opposite direction is also possible. For example, if

there was strong evidence that the student knows concept 1, which is related to topic A, which is related to two other concepts, 2 and 3, the system can conclude that the student has learned topic A, and is also knowledgeable about concepts 2 and 3 (the gray arrows in Figure 3).

Following the semantic links among concepts, one can also make some conclusions about general knowledge levels. For example, if the student knows concept 3, which is an abstraction of 5, 6 and 7, the system can conclude that the student knows at least one of the examples 5, 6 or 7.

For our early experiments, the propagation techniques have been tailored specifically for the concept structure of the CMPT 100 course. We intend to generalize these techniques by overlaying Bayesian belief networks on these concept structures in the future. The updating and propagation of knowledge values through the topic and concept level of all student models takes place all the time, also while the student is not on-line. In this way the models are always kept up-to-date, and when a help-request is posed or a peer-helper is required, the system can react quickly. One important decision we took is to make the student models inspectable by both students and teachers. In this way the unavoidable imprecision and errors that can happen in diagnosis (i.e. in the interpretation of direct evidence and propagation) can be corrected by a human.

2.3 Help-Desk Operation

The student requests help about a certain topic by clicking on a question mark associated with a particular topic, subtopic, or activity (see Figure 4). The help-request is expanded with related concepts and topics which he or she is believed not to know (according to the student model) and is passed further to CPR in order to recommend a thread or posting in the FAQ, which corresponds to the concepts related to the query. If no such resource is available in CPR, or if the student has chosen explicitly to request a human helper, the expanded request is passed on to PHelpS to find an appropriate peer-helper.

PHelpS finds a peer (student) who is currently on-line, and who is knowledgeable about the concepts related to the help request. The identification a peer is carried out by an applet, which is downloaded automatically for every registered user when he /she starts his / her browser. This applet periodically sends messages to the server indicating that the person is still on-line and active. When the peer-matching algorithm has identified some active user as a good potential helper, PHelpS contacts this student by starting another applet. It presents to the helper the topic about which help was requested and the list of the related topics and concepts, which the HelpDesk believes the student asking for help does not know, and opens a chat window. In this window, the helper can answer the question or explore some follow-up questions with the student requesting help. If such a dialogue is established and carried through to conclusion, then the Help-Desk presents an evaluation form to both the student who asked for help and the helper so that they can both provide feedback on the quality of help and the level of knowledge of each other. This feedback is used to update the models of the helper and the student requesting help.

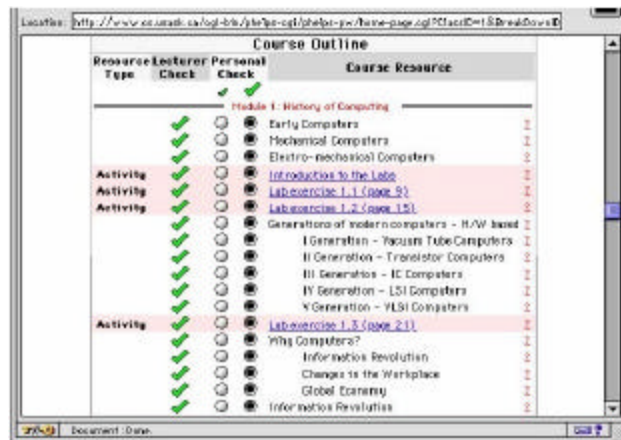


Fig. 4. The Help-Desk Interface

3 Comparison with Other Work

There have been numerous approaches in the field of AI and Education aimed at providing peer help for the learner. Most of them, however, try to create an artificial peer, i.e. an intelligent component or agent, who collaborates with the learner, an approach that was originally proposed by Self in 1986 [13]. Examples of such artificial peers are Dillenbourg & Self's [7] artificial co-learners, Chan & Baskin's learning companions [4], and Aimeur & Frasson's "troublemaker"[1]. All of these systems are focussed on collaborative problem solving (and consequently have a very restricted domain of application). They generate help and utterances themselves (using their knowledge bases) and decide when to interfere (using their pedagogical strategies). In this sense they are classical "Intelligent Tutoring Systems".

Our approach to providing peer help differs significantly from these classical approaches. First, the subject domain of the Help-Desk can be as broad as needed; the only requirement is the existence of some kind of domain structuring (into topics, concepts, tasks or skills) to which help-requests can be indexed. Second, there is minimal *fully automatic* generation of computer-based help, so the system can perform with a less extensive knowledge base and less sophisticated reasoning mechanisms. All the help entries are generated by the students themselves, by means of posting questions and answers to the discussion groups in CPR and by providing direct help via PHelpS. Third, the system doesn't interfere with the help dialogue and doesn't make pedagogical decisions. It is activated only by an explicit request from the student. In this way the Intelligent IntraNet Peer Help-Desk naturally involves human intellect at precisely those points that are currently considered as the

“Achilles’ heels” of AI-based learning environments: the diagnosis of a student’s knowledge, pedagogical decision-making, and generating the instructional content.

The Intelligent IntraNet Peer Help -Desk can be compared with other student model-based approaches for selecting an appropriate human helper. Hoppe’s [9] COSOFT is the first ambitious project to address several issues related to the use of student modelling in order to parameterize human-human collaboration. Later Mühlenbrock et al. [11] pursued this research further. The questions raised by Hoppe in 1995 include the composition of a learning group from a known set of students, and especially the selection of a peer-helper, the identification of problems to be dealt with in a collaborative session or the selection of tasks that are adequate for a given learning group. Hoppe’s approach has been primarily targeted at exploring possible improvements to group student modelling to support human collaboration. It focuses on a limited domain since it employs classical ITS diagnosis, representation and matching. In addition it is intended to support only human-human collaboration, but not to be integrated with an automatic advice or help utility. Our Help-Desk uses the same student models both for selection of human partners and for providing electronic help (based on the FAQ facility). Unlike COSOFT, the student modelling approach employed in our Help-Desk doesn’t rely so much on computer diagnosis, but on human feedback. This makes it easily transferable to new domains. The modification of PHelpS (which is a task-based help system designed to work in procedural workplace domains) to a concept / topic-based help system designed to work in a post secondary course environment took about four weeks of work for one programmer. We expect that changing to a different topic/concept structure for a different domain will take even less time.

4 Conclusion and Future Work

The Intelligent Intranet Helpdesk is being currently employed to support peer-help among the 480 students in the CMPT 100 introductory computer science course for non-majors at the University of Saskatchewan. Data about the discussion groups facility (CPR) usage and about interactive peer-help sessions mediated via PHelpS are being collected. Helpers and helpees have to fill out a short questionnaire after each peer-help session mediated by the system. We will analyse

- the correlation between the system’s choice for best peer helper (based on the student models of the helper and helpee) and the quality of the help provided (based on the evaluation of the helpee);
- the quality of the system’s representation of a student’s (helpee’s) knowledge based on the feedback of the peer helper;
- the overall picture of students’ success on tests and assignments in comparison with results from previous semesters when no HelpDesk was available.

The results obtained will be used to tune the user modelling techniques and to improve the peer-matching scheme of the HelpDesk. In future, we will be pursuing a development of the HelpDesk “in depth” (using more artificial intelligence techniques to amplify the abilities of the HelpDesk in diagnosis, pedagogy and collaboration support). We believe this provides us with a broad and interesting research

perspective, that should result in the construction of flexible, usable, robust and sophisticated tools to support human learning that are characterized by their ability to react to individual differences among learners.

Acknowledgement. This research was carried out under the auspices of the Canadian Telelearning Network of Centres of excellence, project 6.2.4.

References

1. Aimeur E. & Frasson, C. Analysing a New Learning Strategy according to Different Knowledge Levels". *Computer and Education, An International Journal*, (1996) **27**, 2, 115-127.
2. Barr A, Beard, M., Atkinson, R.C. The Computer as a Tutorial Laboratory: the Stanford BIP Project. *International Journal of Man-Machine Studies*, (1976) **8**, 567-596.
3. Canas, A., Ford, K., Hayes P., Brennan, J., Reichherzer, T. Knowledge Construction and Sharing in Quorum, in Greer J. (Ed.) *Artificial Intelligence and Education, Proceedings AIED'95*, (1995) AACE, 218 – 225.
4. Chan T.W. & Baskin A. Learning Companion Systems. In C. Frasson & G. Gauthier (Eds.) *Intelligent Tutoring Systems: On the Crossroads of AI and Education*. (1990) Ablex, NJ, 6-33.
5. Chi, M.T.H., de Leeuw, N., Chiu, M.H., La Vancher, C. *Eliciting self-explanations improves understanding. Cognitive Science* (1994) **18**, 439-477.
6. Collins, J. Greer, J., Kumar, V., McCalla, G., Meagher P., Tkach, R. Inspectable User Models for Just in Time Workplace Training. In A. Jameson, C. Paris, C. Tasso (Eds.) *User Modelling, Proceedings of the UM97 Conference*, (1997) Springer Wien New York, 327-337.
7. Dillenbourg, P. & Self, J. A Computational Approach to Socially Distributed Cognition. *European Journal of Psychology of Education*, (1992) Vol.**VII** (4), 353-372
8. Goldberg M. WebCT – Word Wide Web Course Tools (1997) (Web-Page) <http://homebrew1.cs.ubc.ca/webct/>
9. Hoppe, H.-U. The Use of Multiple Student Modelling to Parameterise Group Learning, in J. Greer (Ed.) *Artificial Intelligence and Education, Proceedings of AIED'95*, (1995) AACE, 234-241.
10. McCalla, G., Greer, J., Kumar, V., Meagher, P., Collins, J., Tkatch, R., Parkinson, B. A Peer Help System for Workplace Training. In B. duBoulay and R. Mizoguchi (Eds.) *Artificial Intelligence and Education, Proceedings of AIED'97*, (1997) IOS Press: Amsterdam, 183-191.
11. Mühlenbrock M., Tewissen F., Hoppe H.U. A Framework System for Intelligent Support in Open Distributed Learning Environments, In B. duBoulay and R. Mizoguchi (Eds.) *Artificial Intelligence and Education, Proceedings of AIED'97*, (1997) IOS Press: Amsterdam, 191-198.
12. Pressley, M., Wood E., et al. Encouraging mindful use of prior knowledge: Attempting to construct explanatory answers facilitate learning, *Educational Psychologist*, (1992) **27**, 91-109.
13. Self, J. The application of machine learning to student modelling. *Instructional Science*, (1986) **14**, 327-388.

14. Virtual-U. Simon Fraser University, Virtual-U Research Project (Web-Page) (1997)
<http://virtual-u.cs.sfu.ca/vuweb/>