CHAPTER 6

COMPUTER-AIDED LEARNING AND USE OF THE INTERNET

Mark Huckvale
University College London, United Kingdom

Chris Bowerman
University of Sunderland, United Kingdom
Anders Eriksson
Umeå University, Sweden
Bernd Pompino-Marschall
Zentrum fuer Sprachwissenschaft Berlin, Germany
Mike Rosner
University of Malta, Malta
Mark Tatham
University of Essex, United Kingdom
Briony Williams
University of Edinburgh, United Kingdom
Maria Wolters
University of Bonn, Germany

1 Introduction

1.1 Aims of the Working Group

The working group in Computer-Aided Learning and Use of the Internet aims:

- to bring together information about resources and tools for computer-aided learning (CAL) in the field
- to promote the use of the Internet to disseminate information and teaching materials
- to make suggestions of areas of resource development to potential authors
- to encourage the generation of a comprehensive set of high quality resources
- to create an infrastructure for the management of resources.

1.2 First year activities

Our main activities in the first year were concerned with a review of current practice. We first requested an expression of interest in the area through a message sent on a number of electronic mailing lists. We had over 70 responses from individuals interested in contributing to our survey of computer-based methods, or interested in hearing about the outcome of our survey [URL1].

We set up a Web-based database of teaching resources, to which individuals could contribute through a forms interface accessible by Web browser. The contents of the database are regularly converted to HTML so that it may be viewed from anywhere on the Internet [URL2]. We made the decision to focus on resources explicitly designed for educational purposes, and on materials that were accessible directly over the Internet.

In our reports on our first year activities (Huckvale *et al*, 1997a, 1997b, [URL3]) we reviewed the current use of the Internet for the provision of teaching materials in Speech Communication Sciences. We concluded that: (i) there was a need for computer-based teaching materials in the field, (ii) that the Internet was an exciting medium for the development and delivery of such materials, and (iii) that teaching materials of defined focus and limited range - which we called 'tutorials' - provided the best compromise in terms of flexibility and effectiveness.

1.3 Second Year Activities

In the second year we have further developed the concept of tutorials and considered in more detail the framework in which such tutorials might be used.

We have set up a database of tutorial components which can be accessed and updated via the Web [URL4]. In doing so we had to develop a set of tutorial ‘characteristics’ by which tutorials could be indexed and searched. The database system also allows feedback from users to authors, and is described in Appendix A. In this Chapter we present guidelines for the development of quality materials and give practical information about authoring. In Section 2 we look at general issues about the design of teaching and learning materials in a telematic context. In Section 3 we try to relate conventional classroom approaches to teaching and learning with their telematic equivalent. In Section 4 we try to give practical advice on the design, development and evaluation of tutorial components. In Section 5 we start a discussion as to the future of telematic teaching within the conventional University system of education in Speech Communication.

2 Telematic Teaching and Learning Materials

In this section we look at Web courseware development and design from both a pedagogic-content perspective and from the practical view of implementation.

2.1 Web courseware development

The development of materials and delivery systems for teaching and learning over the Web is complex because it involves two imperfect sciences: educational theory and software development. In our experience it is easy to lose sight of the educational goals of some courseware by concentrating on the software development, or conversely it is easy to be lazy in the conversion of existing materials and not to make the best use of the available technology.

From an educational point of view, teaching and learning materials must be designed to be effective in the context to which they are applied. There are many parallels between the telematic context and what has become known as ‘Open Learning’, where students often work on their own, at their own pace with materials specially designed to stimulate self-study and self-learning (Race, 1994).

From a software development point of view, the use of the Web and its multi-media technologies suggests the adoption of proven software engineering principles such as prototyping and modular organisation.

In the following sections we look at general educational principles which can be applied to the overall design of courseware, look at the kinds of software development techniques that might be applied, and consider how the medium of the Internet might be best exploited.

2.2 Courseware Design

The following is only a brief outline of the educational issues involved on courseware design. The interested reader is encouraged to follow some of the references.

A standard model for the development of a course follows these basic stages:

1. determine the overall *objective* of the course (e.g., to provide a working knowledge of articulatory phonetics).
2. develop a *curriculum* which will deliver the overall objective. In order to develop a curriculum we need to consider the likely background of students entering the course (i.e., determine their expected prior knowledge and hence the prerequisites for the course).
3. determine, in more detail, the *aims* of the course, i.e., the *skills* and *knowledge* which a student will acquire on exiting from the course. These items of skills and knowledge can be diagrammed as a network in order to determine the inter-relationships between them and hence determine a basic teaching order.
4. determine how best to *assess* these new skills and items of knowledge. This could involve assessments that test individual aspects of the course as well as exiting skills.
5. develop *teaching materials* which provide the skills and knowledge for a student to be able to successfully complete the course assessments.

Courses are often organised into smaller components or modules. A standard analysis of a module is to consider it as comprising a number of *events* (Gagne & Briggs, 1992) which need to be sequenced in order to achieve the desired learning outcomes. The key events are:

- *gain attention* (announce the start of a learning session),
- *inform students of objectives* (announce the expected outcomes for the learning session),
- *stimulus* (demonstrate a skill or reveal an item of knowledge),
- *elicit performance* (give the students some practice),
- *feedback* (give feedback on the student's performance),
- *assessment* (determine the student's performance level),
- *enhance retention and transfer* (aid the retention of the skills and knowledge just learned),
- *summarise* (summarise the session and flag the skills and knowledge learned) and
- *conclude* (terminate the session).

These activities follow a fairly standard lesson plan for introduction, presentation, practice and summary/conclusion. The sequence of modules should be related to the curriculum network of skills and knowledge.

From this high-level design, the telematic components of the course can be developed using a multimedia software development methodology and authoring tools as described below. However this is not the end of the design process since we still need feedback from its users.

Having prepared the teaching materials the course is then run with students in *participation*, and then *evaluated* both in a formative sense, to improve the course and in a summative sense to determine how well the course is achieving its objectives, year by year. Evaluation is against a number of criteria: the need and market for the course and whether or not it is more cost-effective than other alternatives, the usability of the telematic system, content and accuracy, or pedagogic effectiveness (skills, mastery, problem solving skills, information retention, attitudinal changes). The evaluation can look at pre- and post-course results as well as the process of learning. These are often judged by means of questionnaires and interviews of sample groups and can be undertaken periodically. Having evaluated the course we make modifications and repeat the course design process again.

2.3 Courseware prototyping

In order to ensure the efficient development of the Web course we need to adopt a methodological system of software development. A common method is based on prototyping. The developer builds a series of 3 prototypes: the first to determine user requirements and system objectives, the second to determine the processing requirements (data structures, media, algorithms) and the third as an actual implementation to deliver to users. The system is then tested and evaluated and adjustments are made before going into production.

Before producing the first prototype a basic specification needs to be produced. First of all, we need to determine the subject area and scope for the tutorial. Certain subject types suit different teaching and learning styles and these then sit best with different multimedia topologies. Linear and hierarchical topologies are good for reimplementing traditional courseware; network topologies are good for discovery environments; and constructional or simulation systems are good for skills learning. The intended role of the finished system (be it a tutor, or a tool, or a microworld, or even a tutee) also needs to be considered before an initial prototype can be built. This is closely tied to the teaching and learning approach: more open systems are likely to be more of a microworld tool with coaching support, while less open systems are more likely to be tightly controlled tutorials. It is useful to produce a one-page overview of the system to indicate how the screens are linked (as a hierarchy or network, for example). The details of any graphics or clip art (e.g., for buttons), sound and video files can also be specified in advance along with the actions performed when buttons are selected and the due-date for delivery of these items.

In later (i.e., the second and third) prototypes the previous system is run, debugged and edited so that it performs as per a modified specification, produced in consultation with end users. Thus prototyping should be seen as an active process of refinement rather than as a series of innovations.

2.4 Effective use of the medium

Viewed simply as a medium for delivery of instructional material, the Internet does not provide much benefit over a CD-ROM. In fact, since the Internet currently has a relatively low bandwidth, it can not provide as much data in real time as can a CD-ROM. Furthermore, the new DVD-ROM (Digital Versatile Disc - Read Only Memory) can hold up to seven times as much data as a conventional CD-ROM, and also provides video-quality playback standard for moving pictures, as well as greatly enhanced sound quality. This development means that except over specialised high-speed networks (such as the ATM technology used in the Scottish Universities' Metropolitan Area Network) the Internet cannot normally hope to compete with the non-interactive static delivery medium.

The advent of DVD-ROM means that it is especially important to work with the strengths of the Internet as an instructional medium, emphasising what the Internet can offer that is not available on ROM of any kind. The major strengths of the Internet are that:

1. The Internet is a distributed carrier: it is just as easy to access information across workstations locally as remotely.
2. The Internet is capable of creating a “virtual community”: it supports a limited amount of human interaction which brings distance learners together in a social community which has many of the advantages of a traditional classroom.

Newly-designed courseware should play to these strengths, combining the new multi-media opportunities of Web publishing with computer-mediated communication tools such as videoconferencing, audio-streaming, e-mail, chat and newsgroups. Conversely, when the capabilities of the medium and the requirements conflict, we should not be afraid to use paper, books, CDs and DVD-ROMs, audio-tapes, videos and face-to-face meetings in combination with telematic resources.

3 Adapting Existing Approaches

In this section we try to bridge classroom approaches to teaching and learning and the telematic approach. We consider what would be the telematic equivalents of lectures, laboratory work, tutorials and assessments. In each case we give some real examples of cases where this has been attempted.

3.1 Lectures

We would be the first to admit that the lecture format has many limitations. Yet on a world scale, like COBOL as a programming language, it probably remains the most commonly adopted format. For both systems, one consequence of direct importance is that they both leave a legacy. In the case of COBOL it takes the form of computer programs and systems, including well designed ones whose functionality awaits re-exploitation on more modern software platforms. There is also a legacy of possibly excellent lecture courses that are waiting to be "telematicised" i.e. adapted to the criteria of open learning and to a telematic, rather than face-to-face style of delivery. The question is, how is this done? Are there criteria by which to judge whether such an adaptation will be advisable, easy or impossible?

In order to approach these questions we pause to consider the advantages of certain basic characteristics that lectures have. We then consider what is involved in adapting lectures to Internet delivery.

What is a lecture?

A lecture is a specialised kind of teaching event in which a lecturer (usually one person) presents one or more ideas (collectively the *topic* of the lecture) to an audience. Despite huge variations in lecturing style, topic, and size of audience, lectures are distinguished from other kinds of teaching event, such as seminars, tutorials, discussion groups, etc. Without giving a hard and fast definition, we can state with some confidence that lectures generally have the following characteristics:

- **Limited level of interaction**. Advantage: more coverage of material. Within a given time frame, there is clearly a trade-off between coverage and interaction. However, minimal interaction can be extremely valuable, allowing the lecturer to judge directly whether the material being presented is actually being understood, and allowing the student to obtain immediate clarification on material being presented.

- **Face-to-face contact**. Advantage: supports multimodal interaction. Lecturer and students are in direct, two-way audio-visual contact. Despite the limited interaction, the lecturer can count on the full range of modalities including not just text and graphics but also gesture, intonation, facial expression, rhetorical questioning etc. Another effect of face-to-face contact is the social dimension. Because all participants come into physical proximity on a regular basis, it contributes to the growth of a course-centred community.

- **Restricted duration and topic**. Advantage: continuity and focus. Lectures tend to last between 40 and 80 minutes, reflecting the fact that a lecture can command the undivided attention of all participants. This allows material requiring a certain level of concentration to be presented within the tight constraints of the lecture format by ensuring minimum external distraction during the lecture period.

- **Sequential organisation**: Advantage: knowledge of preceding elements can be assumed and built upon. Sequence enters both into the organisation of individual lectures and into the ordering between lectures.

Are lectures an effective educational medium?

It is currently very fashionable to insist that with the advent of new teaching methods that have evolved hand-in-hand with improving communications technology, lectures are now an outmoded medium of instruction. Some would go further and claim that lecture courses are not in the spirit of open learning because the student is not in control of the taught material.

Our own view is that it is that the quality and openness of the educational experience is determined not by the medium *per se*, but by the way the medium is used. A good lecturer will know how to exploit the limited characteristics of the medium to maximum advantage, and this will be reflected in many organisational aspects of the course including the syllabus itself, the organisation of topics, choice and quality of materials, and distribution of topics over lectures. In addition to all this, the lecturer may plan various activities outside the lectures intended to strengthen interaction with the course community.

For this reason, authors of existing courses need not throw up their hands in despair at the thought of adapting an existing course to Internet delivery. The main problem, when a high-quality lecture course already exists, is how best to exploit the features of the new medium.

Although many aspects of lectures can be successfully adapted to Internet delivery, some cannot. Sometimes, for example, the success of the course relies upon the personality of the lecturer or upon the entire experience that is linked to standard lecturing conditions. In such cases, it might be best to consider other forms of telematic delivery such as video or audio cassette.

What would an Internet-based lecture course be like?

Since there are few examples of Internet lecture courses that exist, we can suggest a few pertinent characteristics that might serve as guidelines:

- **Asynchronous delivery**. The course can be followed where and when the student prefers, rather than according to a pre-existing timetable that specifies the location and timing of individual lectures. Of course, there have to be some constraints of this kind (e.g. the course must have finite duration, and assessment will presumably be carried out according to some schedule defined by the educational institution).

- **Browsable course structure**. All the major aspects of the course (e.g. prerequisites, syllabus, descriptions of individual lectures, relations between courses) should be browsable by the interested student to any desired depth. Of course, the reason for browsing the course structure is to enable students to make an informed choice about the particular trajectory they intend to follow through the course materials.

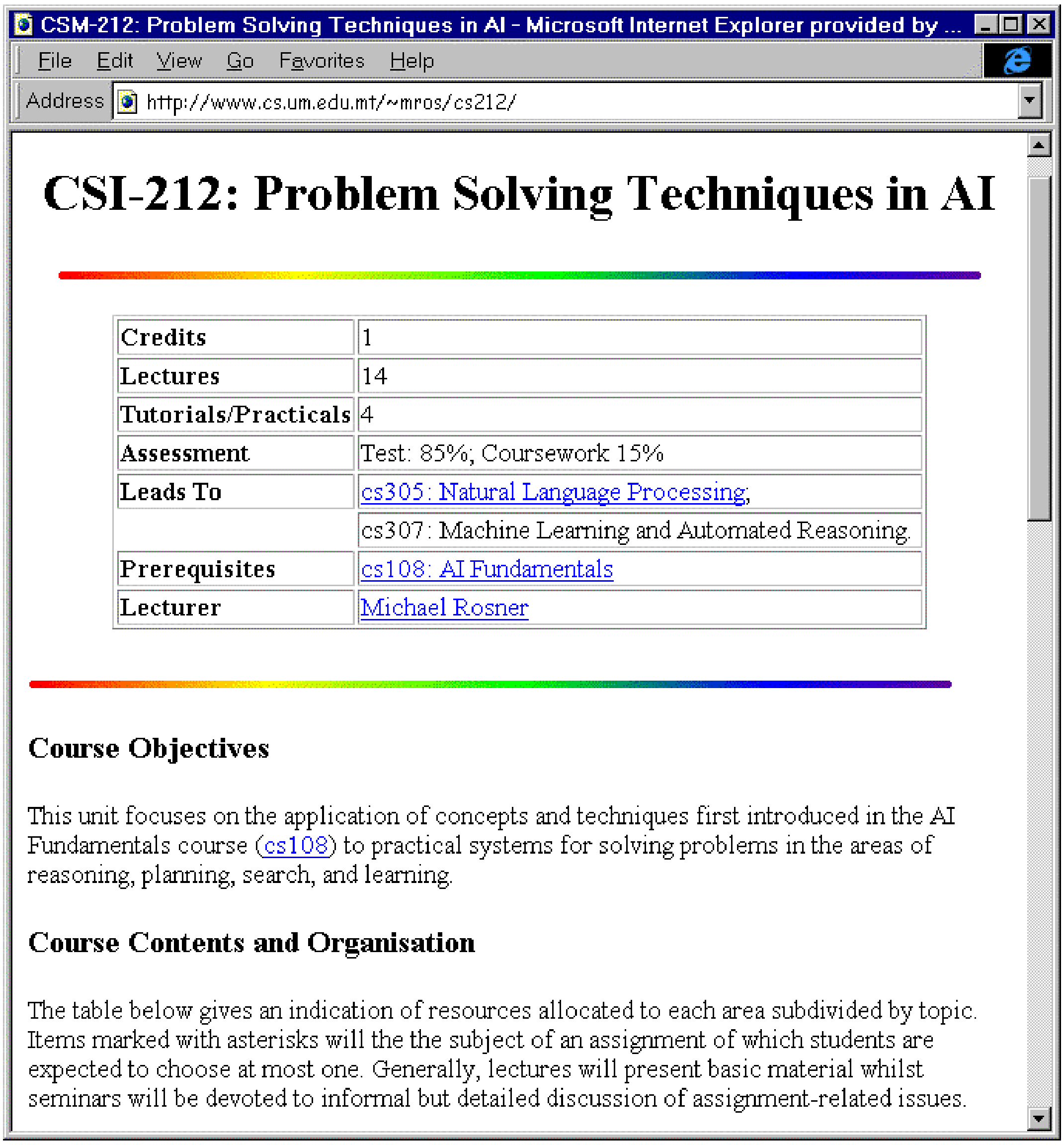
- **Downloadable course materials**. Where students are at a distance from learning resource centres, or when the best resources are difficult to obtain, there is a benefit to using the Web to disseminate course materials. These might include not just lecture notes, but also other kinds of background reading, exercises, software etc.

- **Communication with lecturer and other students**. This is, of course, an integral part of any course for which e-mail or other methods of communication might be used.

Adapting existing lectures to Internet delivery

So much for the final goal. Next we turn to consider two main routes towards it.

One route is to use the existing structure of lectures to as the main organising principle. This can then be embodied in the structure of a course home page as a first step towards sensitising students to Internet delivery. Subsequently the depth and breadth of the home page can be elaborated. Below is an example of the home page for a course for which lectures are still delivered face-to-face, but where course materials are made available over the Internet.



CSM-212: Problem Solving Techniques in AI - Microsoft Internet Explorer provided by ...
File Edit View Go Favorites Help
Address http://www.cs.um.edu.mt/~mros/cs212/

CSI-212: Problem Solving Techniques in AI

Credits	1
Lectures	14
Tutorials/Practicals	4
Assessment	Test: 85%; Coursework 15%
Leads To	cs305: Natural Language Processing;
	cs307: Machine Learning and Automated Reasoning.
Prerequisites	cs108: AI Fundamentals
Lecturer	Michael Rosner

Course Objectives

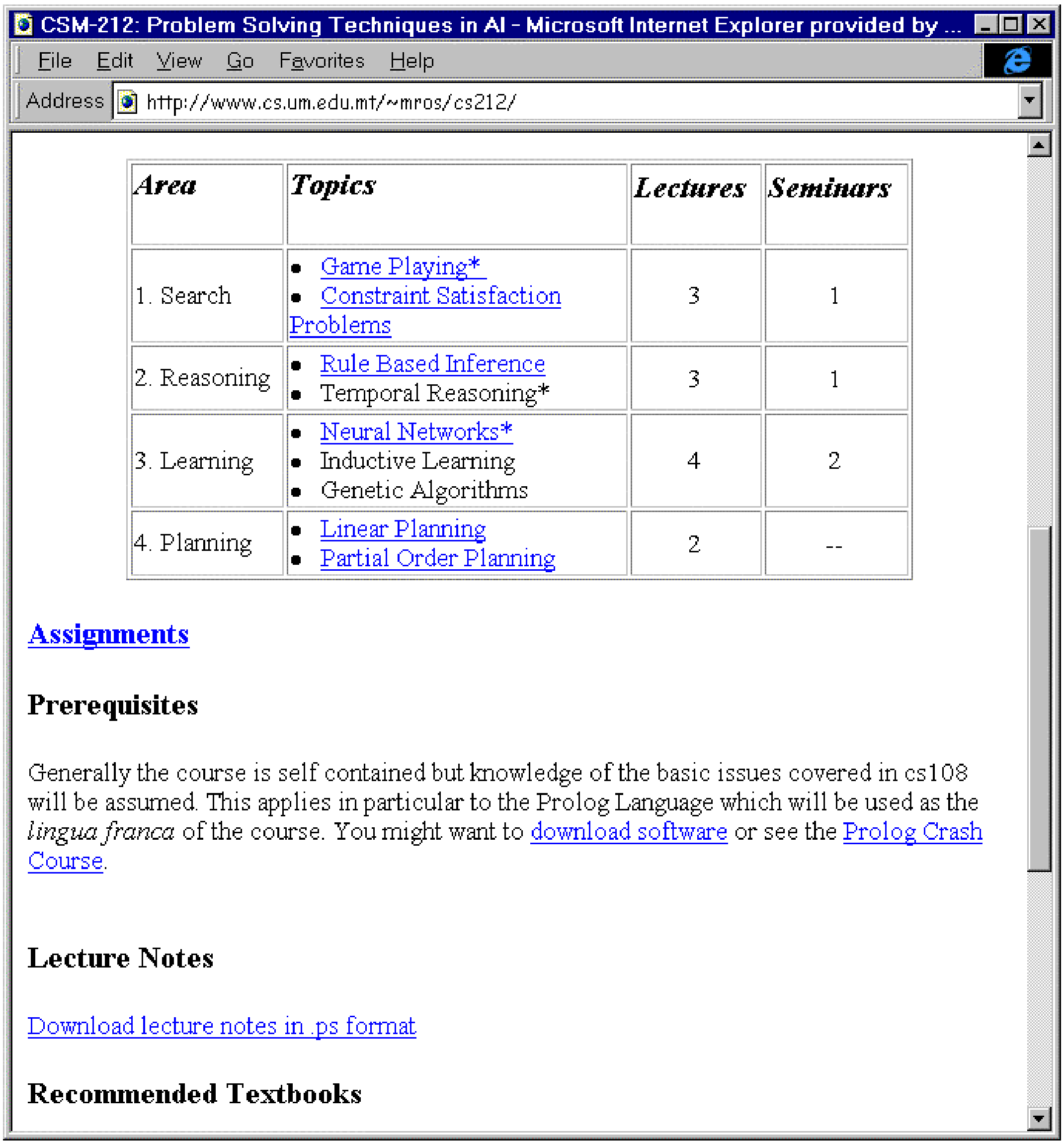
This unit focuses on the application of concepts and techniques first introduced in the AI Fundamentals course (cs108) to practical systems for solving problems in the areas of reasoning, planning, search, and learning.

Course Contents and Organisation

The table below gives an indication of resources allocated to each area subdivided by topic. Items marked with asterisks will the the subject of an assignment of which students are expected to choose at most one. Generally, lectures will present basic material whilst seminars will be devoted to informal but detailed discussion of assignment-related issues.

Figure 6.1 A Course Home Page (such as this one by Mike Rosner [URL5]) can serve as bridge between existing organisation and telematic organisation.

The table in Figure 6.1 gives some information about the course duration, prerequisites, mode of assessment, whilst a second table (shown in Figure 6.2) provides a guide to the contents, indicating the course resources devoted to each subtopic, and a link to the relevant lecture notes and slides. These can also be accessed “sequentially”. Other links point to assignment descriptions, reading materials, and course-related software.



CSM-212: Problem Solving Techniques in AI - Microsoft Internet Explorer provided by ...
File Edit View Go Favorites Help
Address http://www.cs.um.edu.mt/~mros/cs212/

Area	Topics	Lectures	Seminars
1. Search	• Game Playing* • Constraint Satisfaction Problems	3	1
2. Reasoning	• Rule Based Inference • Temporal Reasoning*	3	1
3. Learning	• Neural Networks* • Inductive Learning • Genetic Algorithms	4	2
4. Planning	• Linear Planning • Partial Order Planning	2	--

Assignments

Prerequisites

Generally the course is self contained but knowledge of the basic issues covered in cs108 will be assumed. This applies in particular to the Prolog Language which will be used as the *lingua franca* of the course. You might want to download software or see the Prolog Crash Course.

Lecture Notes

Download lecture notes in .ps format

Recommended Textbooks

Figure 6.2 A course homepage can also include links to related resources: articles, books, data and software.

A second route to follow (which is of course not incompatible with the first) is to exploit e-mail connectivity, since e-mail is the most fundamental vehicle of communication which has the great advantage of being easy to use and set up. E-mail connectivity serves both lecturer/student as well as student/student interaction. The former serves a variety of purposes, e.g. as a means of distributing "bulletin-board" information, such as exercises or lecture materials, as shown above. The latter is of more use for Internet-based discussion, collaborative exercises, FAQs (Frequently Asked Questions and answers) and the like.

3.2 Laboratory classes

In phonetics and speech science, laboratory work is an important component which is well suited for telematic teaching. Normally students work alone or in small groups, analysing their own speech pre-recorded speech material on-line using computer-based analysis tools. Laboratory courses could therefore quite easily be transformed into telematic courses as far as content and methods are concerned. Present bandwidth limitations may make the transfer of speech data too slow to be practical, but this problem can be circumvented by storing the sound files on the local computer and running the analysis programs locally for example as Java applications. At present there are, however, no analysis tools available which run inside Web browsers, locally or over the Internet. Also sound handling in Java, which may be of sufficient quality for demonstrations, is not good enough for serious acoustic analyses. But these limitations and shortcomings will almost certainly be overcome in the near future given the rapid technical development the field.

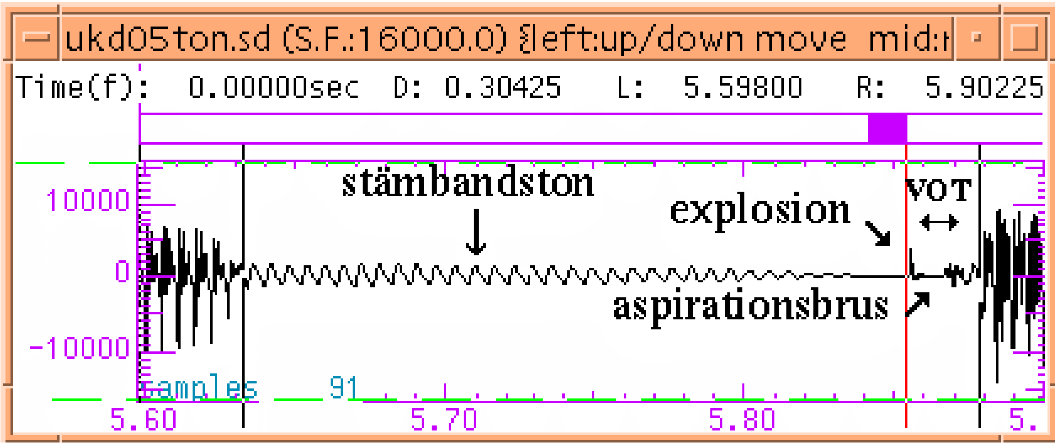
Given the above mentioned limitations, setting up a laboratory course today requires a combination of Internet based components and components run outside the Internet environment. Instructions, examples and demonstrations, course administration and student tracking, typically run in an Internet environment while audio files and acoustic analysis tools run outside the Internet environment, typically locally on the users' own machines

A laboratory course for undergraduate students of phonetics, recognising the present limitations, has been developed and tested at the department of phonetics at Umeå University. The objective of the course was to familiarise the students with basic concepts of acoustic phonetics and analysis tools used in the acoustic analysis of speech. A section of the course was also devoted to concepts and experimental methods in speech perception.

The course consisted of 17 laboratory exercises, each centred on a given topic. All exercises included suggested background readings as well as demonstrations and experiments. The total number of contact hours was thirty, but students had access to the laboratory outside these hours and were free to work through the course at their own pace but observing certain deadlines.

Instructions for the exercises were in the form of Web pages and the course recommended textbook was used for the background readings. Assignments, questionnaires, and student tracking were administered using Internet-based tools developed with the help of the WebCT course development tool.

Acoustic analyses were made outside the Internet environment using the ESPS/*Waves+*™ analysis package. All instructions for the exercises were made with this particular analysis package and computer environment in mind which made it possible to design all accompanying illustrations so that they corresponded exactly to what the students encountered on their screens as they performed the experiments. One such illustration is shown in Figure 6.3.



ukd05ton.sd (S.F.:16000.0) left:up/down move mid:

Time(f): 0.00000sec D: 0.30425 L: 5.59800 R: 5.90225

Figure 6.3. The figure shows an example of an illustration from the instructions accompanying a laboratory exercise on duration measurements of stop consonants.

Perception experiments were accessed from a Web page. The programs were written in Java which means that they could run as Java applications inside the browser. Due to the rather poor sound quality in Java, this option was only used for one of the experiments. The other experiments ran in windows outside the browser and sounds were handled using UNIX audio tools.

Data collection and analyses were also done in the UNIX environment. Students were given immediate feedback upon completion of a given perception test in the form diagrams presenting their results. In addition the combined result of all students which, at a given moment had completed the same experiment, was shown in another diagram. The diagram showing the results for the whole group could be updated at any time to include the most recent results until all students had completed the task.

The laboratory course took part towards the end of the first semester of a full-time course on phonetics. Most students had no, or very little, experience with computers, but this fact did not seem to produce any problems.

3.3 Tutorials, group work, and student discussions

Tutorials in contact classes often involve student presentations and discussions, as well as work in small groups. In addition, students may participate in informal discussion of the course outside contact hours. All these aspects of learning may be labelled “interpersonal”, since they involve interaction with other human beings (whether synchronously or asynchronously). Specifically, they involve interaction with other students as well as the tutor. The same technological considerations are relevant to all these areas, and will be presented in terms of the categories introduced in the table below. To begin with, the different types of interaction will be discussed, followed by a more detailed consideration of the technologies involved.

	Audio-visual	Text-only
Synchronous	Video-conferencing	On-line chat
Asynchronous	Web pages (often updated)	Web conferencing; E-mail discussion list

3.3.1 Interaction: types and benefits

Synchronous interaction

If students can make themselves available at a set time, then synchronous interaction is possible. Using video-conferencing (or possibly a text-only "chat" system), students can give a presentation and subsequent live discussions can ensue. Synchronous interaction may be found more enjoyable, but students' timetables and even variation across time zones may make it impossible.

Asynchronous interaction

The alternative is asynchronous interaction, where discussions may take place over a matter of days, using Web-based conferencing systems, newsreader software, or even a simple e-mail discussion list. This way of holding a discussion has been used informally for many years on the Internet, in the form of the public "newsgroups" and also subscription-only text-based discussion lists. The Web has increased the options available, so that a student's presentation can now be made by means of a Web page, which is particularly valuable in the case of students who are nervous when making live presentations. In addition, Web-based conferencing software makes it easier than before to track the course of a discussion and add one's own comments. It also makes it easier to limit access to any given discussion to only those people authorised to view it.

Benefits of interaction

Student interaction has the advantage of introducing an element of cognitive conflict (i.e. differing viewpoints) into the class situation which can enhance adult learning considerably. It also fosters co-operation, decreases the isolation between distance learners and allows for greater reflection time than normal (see e.g., O'Mally, 1995).

Student-tutor interaction

The same tools can also be used to enhance interaction between students and the tutor on an individual basis. An example is "surgery" tutorials, where frequently asked questions from previous years are stored in a "FAQ" file, which students consult before e-mailing a question to their tutor. Should the question not be answered by the FAQ the student e-mails the tutor, who places the answer for all to see in the FAQ. This approach not only reduces the number of redundant queries to the tutor, but also enables subsequent students to benefit from the experience of previous students.

3.3.2 Technologies for interaction

Video-conferencing

The sound and picture quality of Internet video-conferencing can vary widely, according to the speed of the connection available. The highest quality is available over an ATM network where the quality is of television standard. More usual is an ISDN connection, where the lower speeds (such as ISDN-2) can be so slow as to make the medium almost unusable for serious pedagogical purposes. The most popular speed, ISDN-6, is adequate but can impose limitations on the spontaneity of interactions.

Video-conferencing needs to be moderated and planned in order to be successful, as it is less easy to generate a useful spontaneous discussion compared to the live classroom. The greater accumulation of equipment and controls tends to slow down the speed of interactions, and some training is necessary for all participants, not merely the facilitator. A possible technique for planning and evaluating a video-conferencing (VC) session is given below (Chen, 1994). This looks at the cycle of activities which need to take place before each online meeting. In the pre-meeting (before going online) phase the general management of the online session needs to be worked out in detail (video conferencing is only really effective when adequate preparation has been made). During the meeting phase the most effective strategy is to brainstorm and synthesise ideas. Then, after the meeting the moderator or manager of the project needs to review progress and make plans for the next meeting, i.e., return to the pre-meeting preparations again. This cycle continues until project completion.

Stage	Actions
Pre-meeting	develop statement of problem and goals
	decide on idea generation and consensus-building approach (brainstorm, delphi)
	develop background & support materials
	prepare presentation
Meeting	in plenary state goals, give material, explain approach
	elicit ideas and capture them
	organise ideas and assess them
	present idea (best not plenary)
Post-meeting	Review and cycle again

Text-based conferencing

“Internet Relay Chat” was the informal precursor of this mode of interaction. This text-only synchronous medium allows a surprisingly large amount of interaction to take place over a limited-bandwidth connection. Large conferences tend to work best (Webb, 1995) and a group of eighteen students works much better than a group of nine. Students need to develop their computer “chat” skills by intensive use over a number of courses. It is a good idea to have a welcome message for each participant on a conference, and to encourage students to engage in an initial ice-breaking activity in order to overcome initial reticence. Without such measures, there is a danger of students simply “lurking” (reading messages without participation). Even with these measures, the number of “lurkers” can be up to one-third of participants.

File-based conferencing

File-based conferences are a key, successful telematic tutoring technique. These noticeboard systems (e.g., newsgroups, FirstClass) allow a group of users to post messages which can be read by all. The messages can be organised into “threads” of different topics as required - and it is this ability to organise the information that gives file-based conferencing its power over plain video conferencing and e-mail. A number of approaches have successfully been tried with file-based conferencing and these are listed below. They comprise: moderated conferences (which are managed by a moderator, like video conferences, and require planning and guiding); Jigsaw and consensus building techniques (in which groups of students merge to form ever larger groups - synthesising views as they go to form a group view); and the OU technique (which rewards students for discussing ideas, group work and a short report).

- **Moderated conferences**. The moderator should tidy up the conference, set the agenda, keep things moving (e.g., by Socratic techniques, asking leading questions, contributing their own research and focusing away from distracting topics) and create the desired environment (Hardy, 1992). This environment should emphasise that people are primarily responsible for meeting their own needs and that all are responsible for helping others identify and meet needs by offering themselves as a flexible resource to the community.

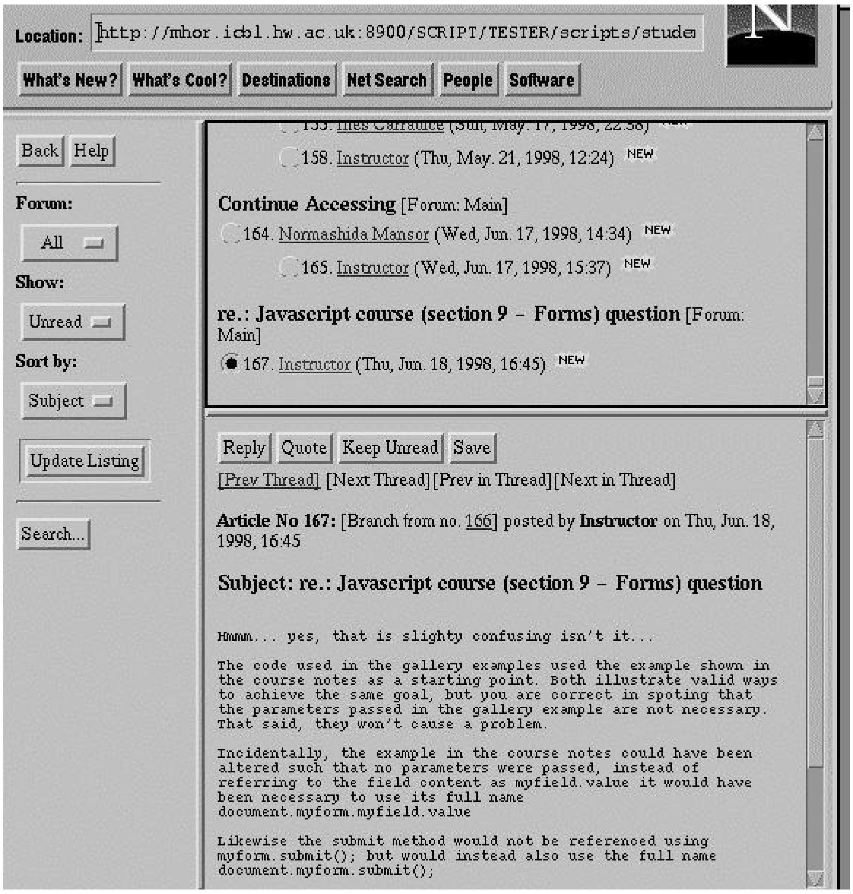
- **Jigsaw conferences**. The Jigsaw method is a useful technique for encouraging co-operation (McMannus *et al.*, 1995). In this technique students are arranged in expert groups, responsible for developing an approach to solving part of the problem. Students are then rearranged in home groups with one person from each of the expert groups and are expected to find an overall solution. This is then brought together by the tutor.

- **Consensus conferences**. A consensus building approach (Lyness and Raimin, 1993) worked well in conjunction with a set of directed reading materials given a few weeks before the conference and after students had been trained to use the telematic tools. There were four sessions, one per week. In the first students were expected to look for key themes and post their position on the conference. In the second students read others messages, looked for an ideal framework and posted a message supporting more than one position on the conference. In the third week students also had to post a message supporting more than one position on the conference. The final session involved a debriefing, discussion and final evaluation.

- **OU conferences**. Learning students are set a question which is marked by their submitting three to seven messages to the conference. Marks accrue according to the extent that they raise issues in the course material, that the messages build on and criticise ideas of others, lead the discussion on and are succinctly expressed. The students must also submit a brief summary of the conference. The marks are balanced 30% own contribution, 40% written submission and 30% for group work and team analysis (Mason, 1993).

Individual student contributions to a file based conference can take the form of Web pages, which have the advantage of a wide variety of authoring tools, mechanisms to include audio and graphical elements, and a standard mechanism for upload to a Web server.

An alternative to the newsgroup technology for text-based conferencing is the Web-based conference. This has the a similar look to newsgroup interaction, but can be driven from a normal browser, with no need for separate newsreader software. An example is seen in Figure 6.4 below, from the TALiSMAN Online Study Centre. This software provides the student with an overall presentation of the discussion threads, a second window showing an individual posting, and hotlinks to use when replying to a specific posting. It is also possible to sort the postings by subject, date, etc., and to specify which forum to view.



○ 155. Ines Carrance (Sun, May. 17, 1998, 22:58) NEW

○ 158. Instructor (Thu, May. 21, 1998, 12:24) NEW

Continue Accessing [Forum: Main]

○ 164. Normashida Mansor (Wed, Jun. 17, 1998, 14:34) NEW

○ 165. Instructor (Wed, Jun. 17, 1998, 15:37) NEW

re.: Javascript course (section 9 – Forms) question [Forum: Main]

● 167. Instructor (Thu, Jun. 18, 1998, 16:45) NEW

Article No 167: [Branch from no. 166] posted by **Instructor** on Thu, Jun. 18, 1998, 16:45

Subject: re.: Javascript course (section 9 – Forms) question

```
Hmmm... yes, that is slighty confusing isn't it...

The code used in the gallery examples used the example shown in
the course notes as a starting point. Both illustrate valid ways
to achieve the same goal, but you are correct in spoting that
the parameters passed in the gallery example are not necessary.
That said, they won't cause a problem.

Incidentally, the example in the course notes could have been
altered such that no parameters were passed, instead of
referring to the field content as myfield.value it would have
been necessary to use its full name
document.myform.myfield.value

Likewise the submit method would not be referenced using
myform.submit(); but would instead also use the full name
document.myform.submit();
```

Figure 6.4: Web-based conference, showing threads, single posting, and viewer option buttons.

Discussion list

Discussion lists have been a feature of the Internet for several years, and are surprisingly valuable, even in the absence of sound, video, or graphics. Most lists are set up in many-to-many mode, such that any postings are immediately distributed to all on the list. However, it is also possible to run a list in one-to-many mode, where all messages are sent from one point only (either originating from the list owner, or originating from list subscribers and then approved for transmission by the list moderator).

This kind of tool is perhaps most useful where Internet connectivity is less than ideal, since text-only messages require minimum bandwidth. Even this restricted medium, however, can contribute towards building up a sense of community among students who may be widely separated.

E-mail-only activities

E-mail activities work rather like managed video-conferences or moderated file-based conferences. An initiating group proposes a project (e.g., weather) and invites participation (via mailing lists etc.). Interested respondents are assigned tasks (e.g., to collect rainfall or sunshine data) and mail information back to the initiating partner. The initiating partner acts as the manager/moderator of the activity and collates the ongoing data and circulates it to all other participating groups as well as preparing a final report (after all iterations of tasks are complete) and publishing it to all. The table gives an outline structure:

Stage	**Activity**
Proposal	Give snappy title, project description (based on example activity), timeline
Refinement	Clarify points as they are e-mailed
Organisation	Acknowledge all messages received. One person Sends, another reads.
Pursuit	Actual activity
Wrap-up	Send answers to co-ordinator of each participating group; discuss at lead partner and assess
Publication	Publish results and send to all

3.4 Assessment

The assessment of students can be broadly categorised under the headings: formative and summative. Formative assessment is designed to aid the student in their learning of the subject; it provides them with an opportunity to respond to a challenge of their understanding in a situation where there is constructive criticism. Summative assessment, on the other hand, is designed to aid the instructor in determining the abilities of the student; it is less often associated with constructive feedback, and more to do with formal measures of competence, leading for example to a qualification.

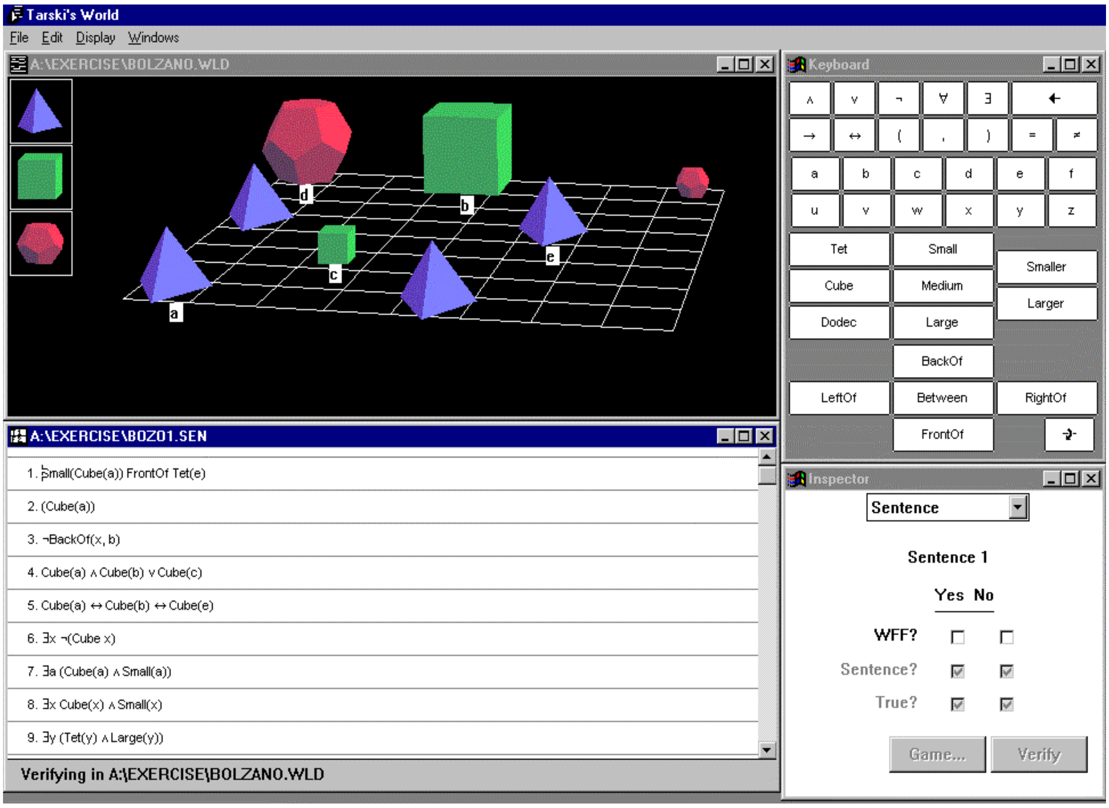
Formative assessment, conventionally, takes the form of homework, self-study material, tutorial questions and problem classes. Summative assessment more often takes the form of written examinations, oral examinations, essays and project work.

In the telematic context, we can classify assessment in terms of the mechanisms by which students make their responses to assessment:

- **via Web forms**: using forms-based Web pages, questions can be posed and answers selected and returned to the Web server. Such responses can be passed on to a human tutor, or may be marked automatically. This type of question and response is most well suited to simple structured questions: multiple choice and single word responses. Here the identification of the correct response can be easily determined automatically, and in the case of formative assessment, returned as feedback [URL6 is a system of this type]. Forms based interfaces can also be used for summative assessment and for responses delivered by e-mail of course.
- **via e-mail**: student responses in the form of any kind of document can be delivered to a tutor as e-mail or e-mail attachments.

- **via Web pages**: an interesting alternative to e-mail is the capability of Web-browsers to upload pages to a central server. In this way students can be asked to create Web pages as an assignment and display the result to all the class. Assessment can then be made by both the tutor and other students.
- **via specialist software**: a final possibility is to record a student's interaction with a software tool. For example Tarski's World (CSLI Publications [URL7]) is a software package designed by J. Barwise and J. Etchemendy (1993) for teaching fundamental concepts of logic, and in particular, the relationship between the language of first order logic and the worlds it models. Tarski's World allows the students to build three-dimensional worlds and to describe them in first-order logic, see Figure 6.5. They evaluate the sentences (bottom left window below) in the constructed worlds (top-left window below) and if their evaluation is incorrect, the program provides them with a game (bottom right window below) that leads them to understand where they went wrong.

Common concerns about telematic assessment are issues of student identification and plagiarism. Although it has always been possible for students to pass off another's work as their own, the telematic format seems to worry teachers and administrators more than does conventional assessment. To enable positive identification it is possible to use digital signatures on telematic work, and though not perfect would seem to be reasonably secure from unwarranted impersonation. On the other hand it will always be easy to pass off the work of others when the student is not being invigilated. Thus the solution to plagiarism in the telematic context is much the same as the normal context: to provide controls over the time and place where responses are delivered.



A:\EXERCISE\BOZ01.SEN

1. Small(Cube(a)) FrontOf Tet(e)
2. (Cube(a))
3. ¬BackOf(x, b)
4. Cube(a) ∧ Cube(b) ∨ Cube(c)
5. Cube(a) ↔ Cube(b) ↔ Cube(e)
6. ∃x ¬(Cube x)
7. ∃a (Cube(a) ∧ Small(a))
8. ∃x Cube(x) ∧ Small(x)
9. ∃y (Tet(y) ∧ Large(y))

Verifying in A:\EXERCISE\BOLZANO.WLD

Sentence

Sentence 1

	Yes	No
WFF?	□	□
Sentence?	☑	☑
True?	☑	☑

Figure 6.5: Tarski's World is a microworld within which students can explore and record their solutions to problems of predicate logic. [URL7]

An interesting use of the Web for both formative and summative assessment is being undertaken within the Computer Science Department in University College London [URL 8]. Students on a course in Systems Analysis and Design can view and practise on a large body of multiple-choice questions accessed via the Web. At the end of the course the student registers his or her intention to undertake the formal assessment. He or she is then presented with a random subset of the questions and a given time limit in which to answer them. Once started the student is obliged to finish the questions within the time available and return their responses.

4 Tutorial Design & Implementation Advice

4.1 Introduction

In Huckvale *et al.* (1997a), Section 6, we gave a formal definition of a courseware component well matched to the requirements of telematic teaching and learning in speech communication science. We simply called these 'tutorials'. Such components, we argued, were small enough for them to be written by one person, yet large enough to be worthwhile reusing by many people.

The key features of such tutorial components were that they i. were oriented to self study, ii. were delivered asynchronously, iii. had a restricted topic and pre-requisites, iv. had restricted conceptual difficulty, v. had a defined internal structure, vi. were largely self-contained, and vii. contained mechanisms for self assessment.

In this section we aim to give practical advice to the designers of such tutorials. In 4.2 we look at issues of Open Learning materials in general, in 4.3 we look at issues of multimedia development, including the choice of topology, in 4.4 we list a number of special issues worth considering about the design and development of Web tutorials, and in Section 4.5 we list some of the authoring tools that might be used for their construction. In addition Appendix B provides a short introduction to writing a Web-based tutorial using a popular authoring system called Authorware, and Appendix C provides a bibliography for further reading.

4.2 Relationship to Open Learning

Web-based materials are basically intended for self-study. Therefore, they are a form of open learning and have to meet the criteria for good open leaning materials. Tutorials are well suited for moving open learning materials to the Web, since they combine passive elements providing information with active elements of assessment, but require only a very limited amount of actual interaction between teacher and student.

A set of tried and tested principles for the development of open learning materials is given by Race (1994). Here we summarise the issues most relevant to Web tutorial developers and comment on any particular telematic aspects.

Open learning materials:

- *are centred on interaction*. It is what the learner does that matters. Learning has to be made an active process. In the absence of a human tutor, the materials must be designed to question and challenge the student’s knowledge and understanding. In contrast text books or lectures are passive presenters of content.

- *aim to motivate the student*. They aim to explain why the subject is interesting and useful, whereas text books assume the reader is already interested.

- *develop the self-esteem of the learner*. Learners are encouraged to think things out for themselves and gain reward by doing so.

- *have clear learning objectives*. They are purpose built for a target group of learners and structured for their needs.

- *allow the user to learn from his/her mistakes.* Responding to the incorrect answers of students is as important as responding to the correct answers. We learn a lot from our mistakes, since they indicate where we need to improve our understanding.

- *are broken up into manageable chunks.* The material should acknowledge that attention is hard to maintain over long periods. Headings are very efficient indicators of structure, while numbering makes it easy to cross-reference. The use of indentation and different font sizes to distinguish between core material and additional information helps. Symbols in the margins can be used to indicate self-assessment questions, definitions, examples and so on. There should be mechanisms for students to make the material their own by adding their own comments. This can be done with the Web with ingenuity.

- *contain structural and summative information.* Figures and graphs should be used to structure and summarise the information and form the basis of activities. Other important components are summaries, active checklists, glossaries of important terms and an index. Finally, the text should contain tips for learners which enable them to use the material more effectively.

- *use a clear and concise language style and friendly tone.* Keep the sentences short and avoid complicated vocabulary. You may want to consult language specific books on good style such as Fowler (1961) for English. The IEEE has published a collection of papers on good technical writing (Beer ed., 1992; Woelfle ed., 1992).

- *should be piloted and edited as a result of feedback.* Provide telematic mechanisms for students to comment on the material.

- *should have some human back-up.* Always provide a human to answer questions beyond the material itself.

4.3 Tutorial Design & Development

When developing CAL materials you must always first look at the design of the course content. A number of courseware development methodologies have been produced (e.g., ASSURE, Heinich, 1993) These methods take care of high-level design issues (e.g., aims and objectives) and require the use of other lower-level design approaches in order to determine the individual learning experiences (see e.g., Gagne & Briggs, 1992). What a methodology such as ASSURE can do is to outline the key steps required in courseware production: setting overall goals with due consideration of the target students, writing material considering the most appropriate media (book, video, Web), trialling the course and adapting it in the light of feedback and then updating it as a matter of course in use.

Once you have decided on a methodology to help develop your material, you should think about the structure to adopt. Small tutorials consisting of a couple of practicals might need no superordinate structure, but more complex tutorials contain such a wealth of material that it should be organised and made accessible to the student in a coherent way. The "pages" of CAL courseware can be organised in a number of ways. Some organisations favour more structured, traditional teaching and learning approaches and others favour more exploratory, progressive approaches. The organisation, or topology, of the "pages" is thus related to the teaching and learning approach taken by the course design team.

There are four key topologies: linear, hierarchy, web/network and simulation:

- The *linear* topology is a simple serial sequence of pages - rather like a book. It is good for rapidly converting existing paper-based courseware to Web and multimedia formats.
- The *hierarchical* topology is commonly used to convert paper courses to the electronic medium. This topology is a tree-like structure which consists of a top-level (home) page which then leads to other sub-pages at lower levels. It is often effectively used when the subject area naturally subdivides into categories and subcategories (e.g., the animal kingdom).
- The *web* or *network* topology consists of a set of pages through which several routes are possible. There is generally a start page and then a number of interlinked pages which the user explores according to their own desire. It is rather like exploring an unknown town: the user starts at a given place (e.g., the railway station) and then wanders around the streets exploring those places that look the most interesting. Such multimedia titles are best considered as a set of "experiences". Each screen is an experience and clicking on buttons leads to other experiences. It is the collection of these experiences that constitute learning. An example might be "Cool Britannia". The title screen, would lead off to experiences on music, art, theatre, history, industry etc. The order in which the experiences are encountered is not important - what's important is that each experience is interesting and that they are appropriately linked, e.g., the section on industry might lead to a historical section on the industrial revolution. Such an exploratory approach is, clearly, in tune with progressive teaching approaches and, at its best, tries to make learning active and fun.
- The *simulation* topology presents a user interface which models a real-world artefact, e.g., the space shuttle. In attempting to fly the shuttle successfully the learners gain much valuable experience of their task. This approach is useful for skills-based learning and is, once more, best suited to more progressive teaching and learning approaches.

Once we have organised the material in one of the topologies described in the previous paragraphs, we need to help the students in exploiting the way the material is organised. Students come to the tutorials with different backgrounds and aims. Some just want to complete the module, others want to acquire specific skills they need for a project, and still others like to explore the topics covered.

The linear topology is the most straightforward and least flexible for both student and teacher. Since the units build on each other, their sequence is fixed. A table of contents and "back" and "next" is all the navigation help students need. If the material is structured more finely, linear paths can help students with very specific needs to acquire the skills they need more efficiently than if they had to navigate the whole structure.

If content is structured in a hierarchical, tree-like fashion, students can decide which topics they want to cover in depth. To facilitate orientation, the teacher can mark the core elements in the tree and group branches under headings which indicate clearly which aspect of the tutorial topic is explored there. The navigation control should make sure that at every point in the module, the student knows exactly where he is in the hierarchy and which options are open to him.

The most difficult structure to navigate is the network of material. It is easy to lose sight of the original goal and be caught in a maze of hyperlinks. Therefore, several paths through the material should be already available. The most adequate tool is surely a simple search engine. However, we can still chart some more explicit paths through the material, either by indexing the components by theme, author, level of difficulty etc. or by setting up a sequence or a tree of components suitable for a given purpose.

Navigating simulations is fundamentally different from the types of interaction discussed above. The student navigates by interacting with and controlling the system, there are no passive components any longer. In order to allow self-paced learning, the students should be able to exit the simulation at various stages and save their current position. Ideally, it would also be possible to control the degree of difficulty (easy, medium, hard).

4.4 Special Considerations

When designing open learning materials for the Web, several additional issues should be considered:

Student Computer Literacy: Students should be familiar with the technology they have to use. This means that they should know at least how to use e-mail and the Web. Experience with a simple word processor is also useful. If the material is structured as a web itself, students should be able to use search engines.

Infrastructure: Remember that bandwidth is limited, especially for students studying from home. Using large audio and graphics files or short movies is therefore feasible only if you can be sure that you have the appropriate infrastructure, for example when the students are using the tutorial in a university lab. You also need to ensure that the students' browsers have adequate plug-ins for handling the data. If students will be using your tutorial at home, create a page which specifies the necessary plug-ins and provides pointers to pages from where they can be obtained.

Fonts: Some texts require special fonts, such as the IPA font for phonetic symbols. Until the advent of Unicode, which will incorporate all of the IPA symbols, there are three options: define your own symbols in the text, use one of the ASCII versions such as SAMPA [URL9] or Arpabet, or require that a phonetic font be installed for the browser. If you do use an ASCII version of the IPA, indicate which one and, if possible, give an online reference.

Availability: Do you want to make your material available only to the students of your course? Then you either need to issue them with special passwords or make sure that the pages can be accessed from specific machines only. Most departments have local pages that can be read from within the department network, but not by outsiders. If your tutorial is to be accessible from outside your department, then what kind of support are you going to give to people who work through it? You can invite their feedback via e-mail, offer to answer questions related to the material, or, once a common European credit assignment system is in place, convert the tutorial to a full blown course for credit.

Multilinguality: If a tutorial is to be used by students from other European universities, it should be easy to translate. This has three consequences. First, try to write in a clear and concise style, avoiding complicated sentences and words and examples that are specific to your cultural background. Second, use more graphics whenever you can, because the language of graphics is universally understood. Third, try not to use text in the graphics. This makes them hard to adapt to a new language, since the translator needs the original file and a program to edit it. Rather, put the necessary information into the caption. Even if you do not translate your material yourself, you can collaborate with native speakers from other universities who want to adapt it. For example, you can provide them with the text in ASCII format, they translate it, send it back to you, and you insert the passages in the appropriate places in your tutorial.

Reusability: There are two levels of reusability, reusing your material for your own courses and making it available on the Internet for the use of others. First of all, you should modularise your material as much as possible. Each module should be described concisely by the prerequisites a student needs to work with it and a set of teaching objectives or a brief description of its content. When your tutorials share a common layout, material from different tutorials can be integrated more easily. If you want to publish it on the Internet, make sure that users know what students can learn from using the tutorial and what type of student it is best suited for. Take care to specify which equipment is needed, such as browser plugins or sound cards, whether the tutorial can be used via a slow Internet connection or whether there is a text-only version. You might also want to add some notes on its design, if teachers are not familiar with the pedagogical technique you use.

Use of technology: The computer makes it possible to supplement traditional teaching media such as print or lectures in a variety of ways. We decided to focus on the Web because it provides excellent means for integrating all sorts of information from plain text to sound and video. It would be beyond the scope of this chapter to provide a systematic overview of all these technologies and an assessment of their use for teaching. Appendix C provides a short bibliography of readily accessible survey articles for those who want to explore these issues further. For a comparison between the Internet and CDROM, see Huckvale *et al*. 1997a, Sec. 3.2. The following checklist, adapted from Race (1994), is intended to help you decide whether a particular technology T is suitable for your students:

- Why do you need T? For example, when teaching sign language, you may well want to use videos, while for demonstrating the effect of vocal tract configurations, a combination of audio and graphics files could be sufficient.
- What objectives can be covered using T? Of what use will it be to the students, will it clarify concepts, demonstrate phenomena, provide data, be used for self-study?
- Can the learners work with T? If the students cannot use your material because they lack the equipment or are not computer literate enough, your efforts will be basically in vain.
- What will the students do with T? Incorporating the use of T into a well-defined task is more effective than having the students just consume the material.
- Should the material be produced from scratch or adapted? Adapting the material is often the better choice since designing new material for an unfamiliar technology can be difficult and time consuming.

4.5 Tools

There are a number of authoring tools available, ranging from simple (e.g. Office packages) to complex authoring suites (e.g. Authorware). These tools can often be used to develop not only Web courseware but also stand-alone (local) CAL material. Table 6.1 presents an overview of some popular tools: more information is listed in Appendix C.

Type of tool	Local	Web
Programming Language	C/C++ Java Delphi VisualBasic	Java JavaScript VBScript
Scripting Language	Apple Hypercard Macromedia Lingo Director	HTML Macromedia Lingo Director First Class/Lotus Notes E-mail
Authoring Tool	Authorware ToolBook	MS-FrontPage Jamba

Table 6.1: Authoring tools categorised by type and means of delivery.

The types of tools listed in the table, above, are likely to appeal to different types of Web courseware developer. Programming tools (C, Delphi) and Web scripting languages (JavaScript, VBScript) are likely to appeal to competent software developers, who are already experienced programmers. These tools offer the flexibility and power required to produce totally bespoke courseware but have a steep learning curve and are not for the IT-novice or faint-hearted.

Intermediate and beginner courseware developers are best off considering one of the many existing authoring packages (Authorware, Toolbook, Jamba). These are easy-to-use; have a modest learning curve and can soon be mastered by anyone with basic IT literacy (mouse and windows skills).

Authoring tools offer good flexibility and productivity but do not allow existing (paper-based) materials to be made available on the Web by means of a simple file conversion. Microsoft Office 97 offers such facilities for all the major office applications (e.g., Word, Excel, PowerPoint, Access) - a simple ‘save as HTML’ will turn an existing office document into an HTML/Web document with no need to understand HTML. Whilst such a solution is not pedagogically ideal (as it may not make the best use of the Web) it does enable existing paper-based open learning material to be made more widely available immediately and at little cost.

The software system WTUTOR [URL6] developed by one of the authors combines normal HTML authoring with scripts on the Web server to provide an interactive tutorial environment.

5. Telematic Education in Speech Communication

In order for educational institutions to compete in the emerging, global educational market place, in which students may select a course from India at 1/40 cost of a European course, there is a need for educational institutions to re-engineer themselves to remain lean and competitive. This begs the question of what a re-engineered institution might look like. Brown and Duguid (1996) see an educational institution as having four components: awards, academics, campus, students. There is no need, they argue, for all these components to reside in one location. At its most simple a college might consist of its campus and would support local students. The other services, awarding body and academic tutors would be selected by the student according to their requirements. This type of model implies that virtual colleges might not only offer courses but also accredit post-hoc (e.g., international courses) and offer the services of their staff telematically and offer courses from other awarding institutions. It also implies that students would be attracted on the basis of a number of factors: e.g., mode of study and quality, cost, availability, quality of support, degree of fit to individual needs (individualisation).

The management of virtual educational institution is therefore a more complex matrix then traditional institutions. This becomes even more apparent when catering for just-in-time training for Single Office/Home Office and small companies whose needs are predicated on a semester based study pattern: they may wish to commence studying at any time (not simply at semester boundaries), or at different paces (full time for 1 week or part time evenings for 10 weeks). Also, as users may be distributed both geographically and in time and pay in different currencies for different facilities (e.g., training materials alone or materials and tuition), it becomes critical to make use of some automated student management and course administration/tracking facilities. For further references, see Huckvale *et al.* 1997a, Section 2.2.

The typical workflow for such a system would be: students browse a module catalogue (trying demonstration material), register to become a student, pay on-line, are assigned a tutor, are given access to or are mailed course materials, work through the materials (being assessed and marked) before being certified and accredited. A number of administrative tools are available and can be purchased commercially. Some run on small PC systems and others on larger Unix workstations (dependent on the scalability of the system).

Security is very important in this context, both within the institution's intranet and for transactions over the Internet. Databases with private information about students or marks should not be accessible by unauthorised persons. It is not enough just to protect the data itself with a password, the whole intranet has to be relatively safe from hackers. Firewalls protect the institution's IT system from Internet attacks by allowing only a subset of all possible services, such as e-mail or telnet. When transmitting sensitive data over the Internet, a technology such as Netscape's Secure Socket Layer should be used. Students and tutors can also encrypt messages to each other with software such as PGP (Pretty Good Privacy).

We can conclude that you need a stable, well-maintained IT infrastructure for setting up and managing fully telematic courses. Even if the system administrators are very co-operative, it is both difficult and time-consuming to create material which is suited to different types of students, easy to use, and visually appealing. Although telematic teaching might be a reasonable goal in the long term, creating such courses might not be feasible for those who have to divide their time between research, administration and teaching.

6 Conclusions

6.1 Summary

In this chapter we have taken further the working group's aim to promote the use of the Internet and the Web in the teaching of Speech Communication Science in Europe. We encourage you to consider your own teaching in the light of our discussion.

We have often been asked: How to make a start? How can ideas of telematic teaching be integrated gradually into everyday practice? A good start are course homepages, which can be readily created and maintained. They provide all necessary information about a course such as instructor, schedule, prerequisites and content. In a next step, the lecturer might want to make his or her notes available on that page, either by converting them to Postscript or PDF (Acrobat) format or directly to HTML. A page with pointers to related information might then be added. To foster communication, the teacher can create a mailing list or a newsgroup which may serves as a forum for announcements, discussion, and student papers. All this can be achieved without the complexity of explicit Web-based tutorial as discussed in the last section. To introduce these, a first step would be to isolate some parts of the syllabus that are particularly suited for open, exploratory learning and convert them into a mini tutorial. Lastly, the students can be provided with some training and self assessment materials, e.g. multiple choice tests.

6.2 Further Work

The working group will continue to explore the area of Computer-Aided Learning using the Internet in the coming year. Our directory of Web tutorials (see Appendix A) will grow and we will start to get feedback from students and teachers.

We hope to provide more pointers to resources and tools, as well as more practical advice on authoring. We would like to investigate issues of copyright, of evaluation, of the effective use of computer-mediated communication in teaching, and in the training of students in the effective use of telematic education.

Finally, the SOCRATES Network itself provides a powerful centre within Europe for the collaborative development of curricula and educational resources in Speech Communication, and Web tutorials will have a role in the sharing and spreading of good practice.

Acknowledgements

We are grateful for the help of other members of the working group and other members of the Thematic Network in the preparation of this chapter. We thank all those that have contributed information to our varied questionnaires and data collection exercises.

Uniform Resource Locators

URL1	mailto:calint@essex.ac.uk
URL2	http://www.phon.ucl.ac.uk/resource/cal-top.htm
URL3	http://www.ling.umu.se:80/~anderse/papers/a0518.pdf
URL4	http://www.phon.ucl.ac.uk/home/wbt/
URL5	http://www.cs.um.edu.mt/~mros/cs212/
URL6	http://www.phon.ucl.ac.uk/home/wbt/wtutor.htm
URL7	http://www.csli.stanford.edu
URL8	http://taco.cs.ucl.ac.uk:8080/taco/cwk.html
URL9	http://www.phon.ucl.ac.uk/home/sampa/home.htm
URL10	http://www.byte.com
URL11	http://www.webdevelopersjournal.com/

APPENDIX A Web Tutorial Directory

A.1 Aims

To promote the use and development of tutorials in Speech Communication Sciences the CAL working group has created a Web system for the registration, indexing and criticism of Web tutorials. This appendix describes the aims, operation, and current state of the system. The system itself can be accessed through the home page for the network at:

```
http://tn-speech.essex.ac.uk/tn-speech/
```

The Web tutorial directory system has the following aims:

- to help teachers and learners find tutorials relevant to their needs
- to give authors of tutorials feedback from learners
- to give new authors of tutorials exposure to good tutorial design
- to provide the CAL group with information about the range and utility of tutorials

The Web tutorial directory system has three main components, described in the following sections: (i) a directory of registered tutorials, (ii) a feedback mechanism, and (iii) an index of tutorials for learners.

A.2 Tutorial Registration

Registration of a tutorial is performed via a Web forms interface accessible through the group home page. The following information is requested:

• Title • Description • Keywords • Language • URL • Contact Name • Contact E-mail • Institution • Accessible to • Level • Study Time • Support • Skills/Concepts • Pre-requisite background • Why follow this tutorial?

When this form is submitted, it is stored on a database on a Web server. Periodically this information is indexed and formatted as pages for learners, as describe d in Section A.4.

A.3 Feedback System

When users launch a tutorial they are also encouraged to fill in a feedback form with their comments once they have tried it. This information is also stored in a database on a Web server and co-ordinated with the directory information below. We have designed the feedback form to be short and simple to fill in. The fields and motivation are as follows:

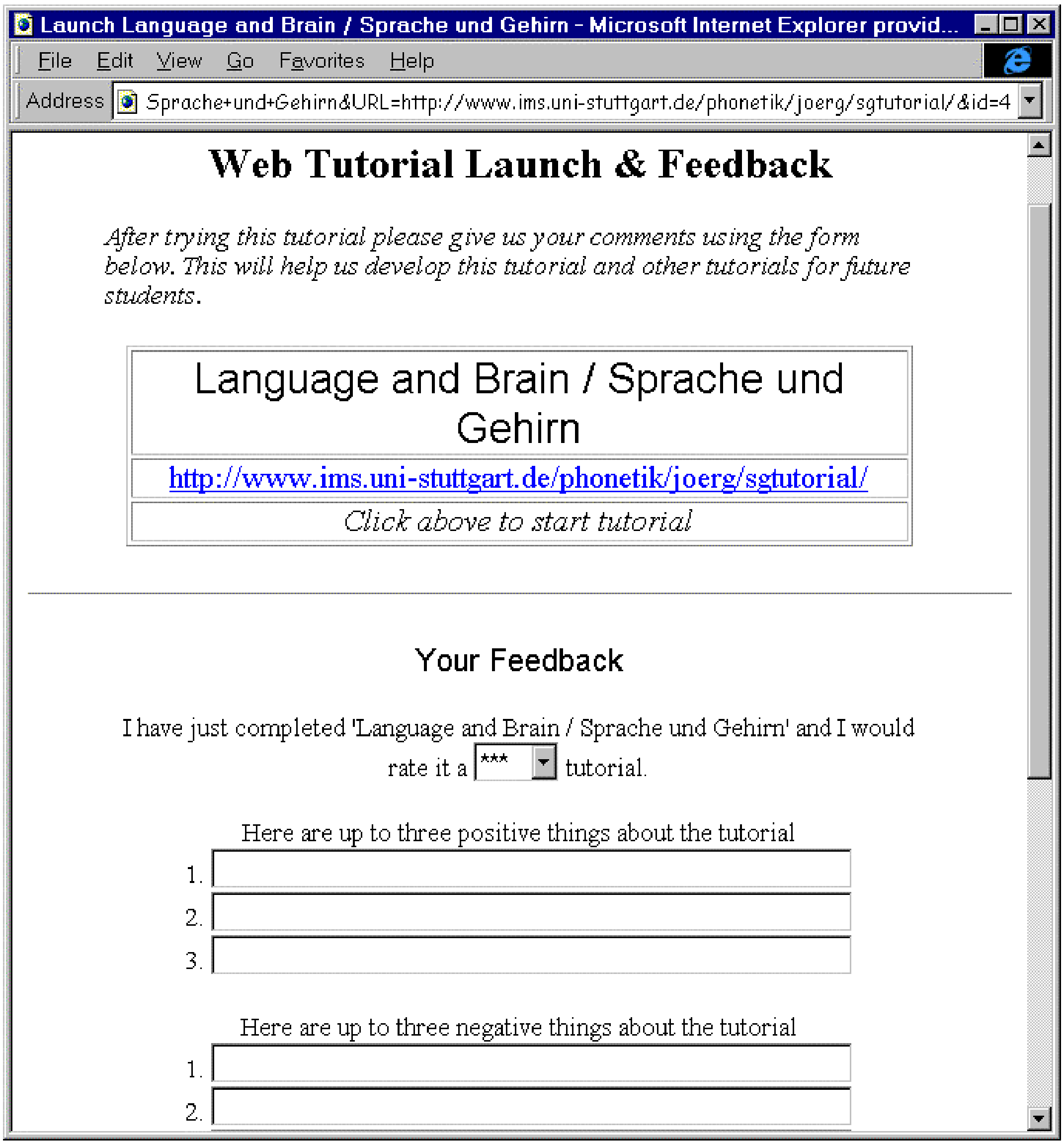
- Overall rating: from 1 to 5 stars.
- Up to 3 positive comments.
- Up to 3 negative comments.
- Other comments.
- Role of user: teacher | specialist student | non-specialist student | interested lay-person.

A.4 Tutorial Index

The directory of Web tutorials is presented according to 3 possible indexes: (i) by level (elementary, intermediate, advanced), or (ii) by language, or (iii) by keyword list.

From each of these indexes a description of matching tutorials can be viewed. Each tutorial is described using the information listed in Section

A.2. A link is available from each description to any user comments; this information is extracted from the feedback system described in Section A.3. When a tutorial is launched, it is through the feedback page. Clicking on the tutorial URL in the Web directory brings up the feedback questionnaire (see Figure A.1). When the URL is selected from the feedback form, the tutorial proper is launched in a second browser window so that the feedback page is not lost.


Launch Language and Brain / Sprache und Gehirn - Microsoft Internet Explorer provid...
File Edit View Go Favorites Help
Address

Sprache+und+Gehirn&URL=http://www.ims.uni-stuttgart.de/phonetik/joerg/sgtutorial/&id=4

Web Tutorial Launch & Feedback

After trying this tutorial please give us your comments using the form below. This will help us develop this tutorial and other tutorials for future students.

Language and Brain / Sprache und Gehirn
http://www.ims.uni-stuttgart.de/phonetik/joerg/sgtutorial/
Click above to start tutorial

Your Feedback

I have just completed 'Language and Brain / Sprache und Gehirn' and I would rate it a *** tutorial.

Here are up to three positive things about the tutorial

1.
2.
3.

Here are up to three negative things about the tutorial

1.
2.

Figure A.1 From the directory of available tutorials, the tutorial is launched via a feedback page which is used to capture user comments.

APPENDIX B Example Use of Authoring Language: AUTHORWARE

B.1 Introduction

Authorware is a tool which enables Computer-Aided Learning (CAL) tutorials to be produced. Tutorials may contain sound, graphics, video, movies and text and it is possible to assess and mark the user and store these results for later.

This guide provides a basic introduction to Authorware. It is aimed at users who have no previous experience of Icon Authorware (referred to as Authorware, below) and will cover the key features of the product. It does not detail all Authorware's features but should make it easy for the interested reader to pick up the manual and learn about additional aspects of relevance to their intended application but not covered in this guide A. basic familiarity with the computer's user interface is also presumed.

B.2 Starting and Stopping Authorware

Authorware is started by clicking on the Authorware icon In order to leave Authorware just click on Exit from within the File menu.

Authorware can create stand-alone deliverable courseware which can even be published via the Web.

B.3 The Authorware Environment

The key two things to observe about the Authorware environment are the windows and controls. We look at these below.

There are two main windows: the design window and the presentation window. The design window has the ‘flowline’ in it and the various activities which the user will go through are attached to this flowline as small pictures (icons). The other window (when shown) is used to display the course as it runs: you may drag objects around in this window to ensure the best layout.

There are two main sets of controls to keep in mind in Authorware: the menus (along the very top of the screen) and the icon palette (down the left of the screen). The menus provide access to all the Authorware commands and the icon palette provides very convenient access to all the commonly used commands. We shall now look at both of these sets of controls.

The menus can be found along the top of the Authorware screen. They are titled: File, Edit, Data, Libraries, Attributes, Text, TryIt, and Help.

The File menu contains options including:

New File which starts up a new session for a fresh tutorial
Open File which opens up an existing tutorial so that it can be completed
Save which saves a session for next time.
Import which allows text etc. which have been created on other packages to be brought into Authorware.

The Edit menu contains options which include:

Cut which allows objects to be removed
Paste which allows objects to be placed back somewhere else on the flowline

The Data menu contains options which include:

Show variables which is used to select what marks etc. are printed on screen

The Text menu contains options including:

Define styles which allows the formatting of text to be changed
Font which enables the type of font to be selected
Size which allows the size of text to be determined

The TryIt menu contains options including:

Run which starts a tutorial
Trace which allows the tutorial to be debugged

Having looked at the most important facilities available from the Menu let's look at what's available from the Icon Palette. This contains the facilities which are required to create computer-aided learning material, for example displays of text or graphics, and multiple-choice questions. If we wish to show the user (of the tutorial) some text then we just click on the text tool. It then appears on the flowline. We can then enter text to display in this icon. The use of the icons will be explained as needed.

B.4 Starting and Saving a Tutorial

To start a new tutorial select New from the File menu. To save your work select Save from the File menu. The first time you do this you will be asked to enter the name of the file that you wish the tutorial to be saved as. After the first save the file will just be saved.

B.5 Displaying Information: Putting Content into your Tutorial

At the moment we just have a blank tutorial with no contents. In order to make the tutorial present material and get responses from the user, we need to add these interactions into the tutorial. Not only do we need to add the interactions but we also need to ensure that the interactions occur in the right order: there's no point in asking a student for a response before presenting the ideas!

The way in which interactions are added to the tutorial is by dragging the appropriate interaction from the icon palette onto the flowline in the design window. The order in which the interaction icons appear on the flowline determines the order in which the interactions occur. The actual details of the interactions is determined by double clicking on the icons, which then open, adding in the details and then closing the interaction icon, again. We shall look at the details of how to do this in the subsequent sections.

B.6 Text Displays

To add text to the tutorial, drag the display icon onto the flowline. Then double-click on the icon and it will open. Click on the A symbol in the window that appears and then click in the display icon. Click on the A symbol which appears in the tool bar for that window. You may then type in some text.

N.B. It is possible to bring text into Authorware which has been created outside it, e.g., in Microsoft Word. This can be done with the Import option from the File menu and by supplying the name of the file to import. It is best to import text that has been saved in RTF (rich text format) format.

You can now try running the tutorial with Run from the TryIt menu. You should see your text displayed in the presentation window.

B.7 Graphics

Graphics can be incorporated into tutorials in either display (see above) or the interaction (see below) icons. This can be brought into the opened display icon with Import from the File menu and the Paste button. The running tutorial should now display both the text and the graphic.

B.8 Digital Movies and Sound

It is possible to incorporate digital movies and sound into tutorials. Digital movies are video clips which are recorded in digital form on disk. On a Mac, Quicktime, PICS and Macromedia Director movies can be used and on a PC, MPEG, AVI and Macromedia Director movies can be used. In order to incorporate a movie or sound it must have been created beforehand and saved as a file in one of the formats mentioned above. You need to drag a digital movie icon into the flowline and then to open it to specify the name of the file.

N.B. It is best to use the Digital Movie for sounds too because it enables them to be synchronised with other activities. If you are just including sound without any video images then set the Video Options dialogue box to movie with just sound.

B.9 Branching and Interactions 1

So far our tutorial has just displayed things: we shall now look at how to interact with the user.

Let's imagine that we wanted to ask a simple question: Is there a King of France?

Two possible answers are possible here. The correct answer, 'no' and the incorrect answer, 'yes'. The tutorial might wish to do different things according to the answer which the user has typed in. That is it might want to display 'Right' for the correct answer and 'No.' for the incorrect answer. In order to do this the system is going to need two paths in the flowline: one for what happens for the correct answer and one for what happens for the incorrect answer. These paths are known as branches and are handled in Authorware by means of frameworks and maps. When designing interactions with the user it is a good idea to think about the nature of the interaction (e.g., question and possible answers) and plan the possible routes. This is best done on paper before using Authorware.

B.10 Branching and Interactions 2

Let us add the simple interaction, described above, to our tutorial. You need to drag the framework icon to the flowline. Then you need to drag a map icon to the right of the framework icon. The map icon is where all the structure of the quiz is stored. The framework icon just indicates that we are setting up a 'private' network of branches - just for the quiz. The basic intuition is that we set up some form of interaction (by means of an interaction icon) and a network of 'displays' to the right of this. The type of display and the routing through the network is controlled by clicking on the small symbol above the display and selecting various options.

Let's set the framework up first. You need to click on the framework icon and click on each of the triangular navigation icons that are there and press the delete key to remove them one by one. You should drag a triangular navigation icon onto the framework icon's network (to the right of the interaction arrow) and click on OK. Click on the navigation icon (triangle) then select nearby and exit framework from the options. This makes it possible to leave the quiz and return to the main flowline.

Now, you can click on the map icon and set up the quiz. Firstly, drag a display icon onto the flowline and enter the question into it: "Is there a king of France?". Beneath this drag an interaction icon and then, to the right of the interaction icon drag two map icons (clicking button and OK). Name the first map icon YES and the second NO. Click on the small rectangle (called a response icon) above the yes map icon and set the erase feedback to exit interaction. This will make the quiz stop when the user selects the correct answer. Now we need to put some feedback into each map icon. Open the YES map icon and drag a display icon into it. Then open this display icon and enter yes. In the other map icon drag another display icon and enter NO into it. Close the map icons and drag a display icon under the interaction icon. Open this display icon and enter the text Exit?

This completes the definition of the quiz and it can now be run.

N.B. You may need to run the program and click in the presentation window to arrange items for the best layout.

B.11 Erasing the Screen

It is often necessary to erase the screen, e.g., after one interaction and before another. To erase the screen after an interaction select the interaction icon after which you wish the screen to be erased and then select the Erase Interaction option and the desired erase effect.

Erase effects are set in the response icons (small rectangles above display and map icons in frameworks) by clicking on interaction icons and by the Attributes | Transition options.

B.12 Graphic Buttons as a Response to Interactions

It is possible to use graphics as buttons to accept user responses. To set up an interaction and a display in response, you need to drag an interaction icon onto the flowline. Click in the interaction icon and then on OK Edit Display. Enter a query, then drag the icon for the desired response (a display icon in our case) near the interaction icon. This produces a response type dialogue box. Select buttons then select the result icon and type the title of the button (button1). Fill the display icon with the material intended to appear in response. If you add further display icons then there will be more buttons and possible outcomes - add a further display icon (button2) and different response material. You can now try running the tutorial.

B.13 Judging and Recording the User's Performance

In order to be able to offer feedback to the student on their progress it is necessary to judge how they are doing and record this information. Much of this is automated in Authorware. We shall look at how to judge a user's performance and then how to provide feedback.

Student performance can be judged by knowing things such as the length of time they take to complete a task, how many attempts they had and the correct number of right and wrong answers. This information is automatically recorded by Authorware so all we have to do is get it to record these 'judgements'. This is done in the response icon to the above of the display (or map) icon. You need to double click the response icon above button1 and select an option (correct response) from the response tracking list according to the type of response in the interaction ('correct response' increases a count of correct responses, 'wrong response' increases a count of incorrect responses and 'exit interaction' returns flow to the main flowline after the last response). Then you can click the response icon above button2 and set response tracking to wrong response and exit interaction.

We now need to print out the scores recorded. This is done by creating a display icon into which we write the names of the counters. Place the display icon under the interaction icon arrow. Open the display icon and then you may select from all the available counters (including Percent Correct and Percent Wrong) by selecting their name from Show Variables in the Data menu in the Interaction group and clicking on Paste then Done. Select 'Percentcorrect' and 'Percentwrong'.

If you run the tutorial you should see your score.

B.14 Debugging a Tutorial

By now our tutorial is getting quite complex and there may be errors in it which we need to remove. This is called debugging. The debugging tools can be found in the TryIt menu under the Trace option.

You may select to 'step into' a particular icon, i.e., view it as you run the program or 'step over' an icon, i.e., ignore it (because you know that it works).

You can also drag the start and stop flags to section of the tutorial that you wish to debug when running. When you now run the tutorial just the part between the flags will be traced. Remember to remove the flags before delivering your system!

B.15 Hot Text

Hot text is the same as hypertext (the idea behind the world wide Web). The basic idea is that certain words (hot words) in a text are linked to other texts. This means that when you click on a hot word the text linked to that text is then displayed. This text can also have hot words which link to other texts and so on. An example might be in a text about Neural Networks:

> "Neural networks are made up of a large number of neurons which are interconnected by axons."

There are 'jargon' words in this text which might need further explanation, e.g., neurons, axons. A good way of providing further explanation is to link a further, explanatory text to the jargon word by making it into a hot word. Below, we see how to do this.

In order to define a set of linked texts we need a way of indicating that all the texts are related. This is done by dragging a framework onto the flowline. Next you need to drag a couple of display icons to the right of the framework icon. They will be grouped together. Now you need to put some text into these display icons. The first is "Neural networks are made up of a large number of neurons which are interconnected by axons." and the second is "Axons connect neurons." We now need to link the texts in these display icons.

Firstly, we must define the style of text which we wish to use for our hot link. This is done by selecting Define Styles from the Text menu. You need to Click on Add but do not hit return. Then enter a name for the style, such as ‘Hyper’. You can select a style of Underline and Bold.

Now we need to define what happens when we click on a hot word. This is achieved by selecting ‘Interactivity’ and ‘single click’, followed by ‘AutoHighlight’. Then by selecting ‘Cursor’ and choosing a cursor which will appear over hot words. Now we must enter the destination of the text we shall go to when the hot text is clicked on. This will generally be 'nearby' (but you can also select first, last or next). You need then to click on modify and the n OK.

We must now define hot words in our text. This is done by opening a display or interaction icon in which you have the text that you wish to make into a hot word. Select the first display icon and enter the text (about neural networks) which will make up the hot link. Now you need to select the text to turn into the link word and to click on the Apply Styles option from the Text menu and the style which we defined for hot text (i.e. Hyper). This will turn the text you selected into hot text. Enter some explanatory text into the second display icon, e.g., ‘axons link neurons’.

B.16 Getting Keypresses

An easy way of giving the user control over the tutorial is to create a menu of choices which can be selected by pressing a single key. We shall create a simple menu which has two choices.

First you need to drag an interaction icon to the flowline and enter the following query into it: "select a or b". Now, to the right of this interaction icon drag two display icons. In the first display icon put the text "You made choice A" and in the second "You made choice B". Name the icons "a" and "b". Also, call the interaction item "Choices". Now you need to click on the response icon above each of the display icons and, with the Change Type button, set it to "keypress". Run the system and you should get a query with two choices (a and b) which you can select.

B.17 Menus

Another easy way of giving the user control over the tutorial is to create a pull-down menu of choices. We shall create a simple menu which has two choices.

First you need to drag an interaction icon to the flowline and, to the right of this two display icons. In the first display icon put the text "You made choice 1" and in the second "You made choice 2". Name the icons "Option1" and "Option2". Also, call the interaction icon "Menu". Now click on the response icon above each of the display icons and, with the Change type button, set it to "Pull-down menu". Enter some different text into each of the display icons (e.g., ‘option one’ into the first and ‘option two’ into the second. Run the system and you should get a menu (called Menu) with two options (Option1 and Option2) which you can select.

B.18 Scripting

Most things which you might want to do can be done by Authorware automatically but there are times when you might need to customise some of Authorware's by adding some simple extensions that you have written yourself. This is known as scripting. Authorware scripting is based on a simple programming language.

Authorware scripts are stored in calculation icons. The script is just typed into these icons as text. We can put calculation icons anywhere in our tutorial just by dragging an icon to the place we want it. One example of needing to use script would be when we wish to check what a user has typed in, do something with that input and then output the result. As an example we shall set up an interaction in which the user enters some text and we then do some processing on the text before printing it out.

You need to drag an interaction icon onto the flowline. To the right of the interaction icon drag a display icon. Then you need to click on the response icon (small box) above the display icon and change the type to text entry. This display icon will now enable us to enter text. After the user has entered something into the text field of the display icon we wish to process this input. To do that we must store the text entry somewhere where it can be accessed. We shall store the text entry in a named location (called a variable) to which we shall give the name 'TextIn1'. All text entries are initially stored in a location called EntryText (actually EntryText@"icon name". What we need to do is get the contents of EntryText (the text field) and put it in a public place which we can process. This is done by means of the := sign (which means assignment):

```
TextIn1 := EntryText
```

Drag a calculation icon to the right of the display icon and enter the above text into it.

What we would now like to do is to be able to perform different actions according to what the user has entered into the system. We shall just make a variable contain 1 (if the user enters hello) or 0 (if the user enters anything other than hello). To do this we need an "if <condition is true> then <do something> statement" in the calculation icon. It looks like this:

```
if TextIn1 = "hello" then Flag := 1
```

Type this straight into the calculation icon.

Another thing that we might want to do is to add one set of words to another, e.g., we might want to add "very" to "very very". This is done, in Authorware by adding a word to some existing words which are stored in a variable. If the words are stored in Words then we can add another "very" with the following line:

```
Words := Words ^ "very"
```

We might wish to repeat something a number of times, for example we might wish to add "very" ten times to a list of words. In order to do this we need to repeat a loop ten times. A loop which will do something ten times looks like this:

```
repeat with Counter := 1 to 10
     <whatever you wish to repeat goes in here>
end repeat
```

Let's try to add very to an existing (empty) list of Words, ten times. We will end up with a list of ten verys. The script to do this looks as, below:

```
Words := ""
repeat with Counter := 1 to 10
     Words := Words ^ "very"
end repeat
```

Add this to the calculation icon.

Our complete script looks as below and is all stored in the calculation icon. The calculation icon is just a place for storing script. Whenever Authorware finds a calculation icon on the flow line it executes the script before continuing. Close the calculation icon and hit OK in response to Authorware's prompts.

```
Message := TextIn1 ^ " there"
Flag := 0
if TextIn1 = "hello" then Flag := 1
Words := ""
repeat with Counter := 1 to 10
     Words := Words ^ "very"
end repeat
```

If we run the script nothing appears to happen: we can not see the results of the script because we need to display them. To display the results you need to drag another display icon to the right of the calculation icon. Insert some variables into this display icon (those our script uses): TextIn1, Flag, Message, Words. To insert variables select: Data | Show Variables. The category must be set to the name of the file you are using and then you will see a list of variables. Select the names of each one we are interested in (in turn) and paste them into the display icon as text. Now try running the tutorial.

B.19 More Information

Authorware is a complex piece of software: this guide gives a basic introduction, only. There are some associated files which you can inspect which illustrate the points made in this guide. They are on the Web at

`http://osiris.sunderland.ac.uk/cbowww/AUTHORWARE/`

APPENDIX C Bibliography

To provide more detailed technical information about WWW and the internet would be beyond the scope of this article. Huckvale *et al.* 1997 already provide a list of good introductory books, so that it will suffice here to point to two series of books which have been translated into all the major European languages: the Dummies series for absolute beginners, published by IDG Books, and the Nutshell series of introductions and references published by O'Reilly & Associates for the more experienced users.

For this chapter, we put together a brief list of introductory articles for those who would like to learn more about specific aspects of the Internet.
These articles provide “executive summaries” and brief tutorials for those who do not have the time or the inclination to work through a complete book. These articles are from the renowned computer journal Byte, which caters for a very wide computer literate audience and whose articles often go beyond the product reviews and market surveys typical for most magazines. All of the articles mentioned can be read at the Byte Web site [URL10].

There are two other very informative magazines whose content can be accessed via the Web.

- *The Web Developers' Journal*. Wide range of information about using the WWW. [URL11]
- *WWW Journal*. Published quarterly, provides up-to-date information about the development of the Web plus a set of in-depth articles which focus on a common topic, such as scription languages, Web page design, or WWW infrastructure. Paper version published by O'Reilly, only back issues are on the Web

We do not cover the more scholarly journals such as relevant IEEE Transactions, Journal of Human Computer Interaction and the like, because they are not as readily available.

Using the Web

- *Applications and the internet*
  Web Applications at Your Service. (Web servers can host applications which can be used by net surfers) Byte 7/97
  Internet-Aware Applications. (potential application programs based on the internet) Byte 3/96
- *Interacting with others*
  HTML + NNTP = Groupware. (using the Web for interactions between members of a group) Byte 9/97
  Groupware Strategies. (Intranet or Internet?) Byte 7/96
  HTML Groupware. (using the page editing facilites of browsers for interaction) Byte 2/98
- *Searching for information*
  The Searchable Kingdom. (basic information about search engines) Byte 6/97
- *Teaching*
  New Ways to Learn. (state-of-the-art survey) Byte 3/95
- *Using Markup Languages*
  Effective HTML Forms. (Web-based forms) Byte 5/98

Dynamic HTML and Scriptlets Add Life. (changing pages dynamically) Byte 1/98
Extensible Markup Language. (new markup standard for Web pages) Byte 1/98
Cascading Style Sheets. (templates for Web design) Byte 11/97
Forms Follow Function. (Web-based forms) Byte 9/97

Infrastructure

- *Miscellaneous*

  The Internet Reinvented. (future developments in Internet infrastructure) Byte 2/98
  Faster Internet Access. (for your students) Byte 8/97
- *Bandwidth*

  Gigabit Ethernet Gears Up. (more bandwidth for the Internet) Byte 1/98
  Smarter and Faster IP Connections. (for transmitting sound and video) Byte 10/97
  Preparing for Gigabit Ethernet. Byte 10/97
  Batter Up for Broadband. (fast connections for home offices) Byte 10/97
- *Internet security*

  Exploring the SSL. (Internet Security Technology implemented in Netscape) Byte 4/98
- *Multimedia*

  Walking, Talking Web. (practical aspects of multimedia and Internet) Byte 1/97
  Data's New Voice (voice communication over the Internet) Byte 9/96
  Virtual Reality Goes to Work. (Virtual Reality in mainstream applications) Byte 3/96
  Multimedia Tools Animate the Web. (using multimedia on the Web) Byte 3/96
  Put the Space in Cyberspace. (3D modeling via Internet) Byte 3/96
  Reliable ATM Networking. (potential tool for transmission of voice and data). Byte 4/97
  Faster, Smarter Nets. (comparison of technologies: ATM vs. routed IP) Byte 4/97
  Multimedia Over the Network. (state of the art report) Byte 3/97
  Merging Voice & Data. (problem solving strategies) Byte 2/96
  ISDN: Give Up And Go?. (ISDN and the transmission of voice+data) Byte 2/96

References

Barwise, J., and Etchemendy, J. (1993). *Tarski's World*. Stanford: CSLI Publications.

Beer, D.F. (1992). *Writing and Speaking in the Technology Professions*. New York: IEEE Press.

Brown, J., Duguid, P. (1996). *Times Higher Educational Supplement*. 5/96.

Chen, H.(1994). Collaborative systems solving the vocabulary problem. *Computer*, May.

Fowler, H.W. (1961). *A Dictionary of Modern English Usage*. Oxford: Clarendon.

Gagne, R., Briggs, L., Wagner, W. (1992). *Principles of instructional design*. Harcourt Brace Jovanovich.

Hardy, V. (1994). Introducing Computer-Mediated Communication into participative management education: the impact on the tutor's role. *Educational training*, 29(4).

Heinich, R., Molinda, M., Russel, J. (1993). *Instructional Media*. Macmillan, USA.

Huckvale, M., Benoit, C., Bowerman, C., Eriksson, A., Rosner, M., Tatham, M., Williams, B. (1997). Computer Aided Learning and Use of the Internet. In *The Landscape of Future Education*, vol 1, G. Bloothooft et. al. (eds.), Utrecht: OTS Publications.

Huckvale, M., et al. (1997). Opportunities for Computer-Aided Instruction in Phonetics and Speech Communication Provided by the Internet. *Proceedings of the 5th Eurospeech Conference*, Rhodes.

Lyness, A., Raimin, J. (1993). *Proceedings NATO advanced Workshop*, NAJADEN papers.

Mason, R. (1993). Designing collaborative work environments for online courses. In Davies, G., Samways, B. (eds), *IFIP Transactions* A29, North Holland.

McMannus, M., Aiken, R. (1995). Using an intelligent tutor to facilitate collaborative learning. In *Innovative adult learning with innovative technologies*, Collis, B., Davies, G. (eds.), IFIP A61, North Holland.

O'Mally, C. (1995). Computer-supported collaborative learning. *NATO ASI Series* F, Vol 128. Berlin: Springer.

Race, P. (1994). *The Open Learning Handbook*. London: Kogan Page.

Webb, B. (1995). *Innovations in education and training international*, 32(2).

Woelfle, R.M. (1992). *A New Guide for Better Technical Presentations*. New York: IEEE Press.