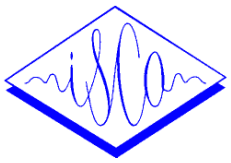

ESCA/SOCRATES Workshop on Method and Tool Innovations for Speech Science Education (MATISSE)
University College, London, UK
April 16–17, 1999

Tutorial Design for Web-based Teaching and Learning

Chris Bowerman[1], *Anders Eriksson*[2], *Mark Huckvale*[3],
Mike Rosner[4], *Mark Tatham*[5], *Maria Wolters*[6]

[1]University of Sunderland, [2] Umeå University, [3] University College London,
[4] University of Malta, [5] University of Essex, [6] University of Bonn
M.Huckvale@ucl.ac.uk

Abstract

The Computer Aided Learning working group of the SOCRATES thematic network in Speech Communication Sciences have studied how the Internet is being used and could be used for the provision of self-study materials. In this paper we build on our findings and make recommendations that should be useful to any current or potential author of tutorial materials designed for use on the Web. Our recommendations to authors of tutorials include: to choose topics of appropriate size, to design tutorials for re-use, to take care in the use of language, to design for open-learning, to use interactive assessment, and to make appropriate use of the medium. We provide a checklist which authors can use to measure the effectiveness of their tutorials and which can help improve the quality and utility of the final result. We give some Internet links to tutorials, which can serve as models of either good design or good use of the medium.

1. Introduction

The Computer-Aided Learning (CAL) working group of the SOCRATES Thematic Network in Speech Communication are studying the current use and the future potential of the Internet to support teaching and learning in the field [3] & [5]. Our findings are that Speech Communication education has special characteristics that lead to particular interest in Internet/CAL: it is strongly interdisciplinary, distributed over many types of university departments, involves multiple languages and usefully exploits interactive media.

However our survey of existing Internet teaching resources in the field [4] highlighted the fact that much educational material on the Web was poorly designed from the students' point of view. We found material that was no more than a text-book presentation of ideas, material that did not encourage the student to discover new concepts, material that did not challenge the understanding of the student, and material that did not make the best use of the medium.

In [3], we proposed that progress could be made through the building of small autonomous teaching components that we called, simply, *tutorials*. The key features of tutorials were that they (i) were oriented to self-study, (ii) had restricted scope and clear pre-requisites, (iii) had restricted conceptual difficulty, (iv) had a defined internal structure, (v) were largely self-contained, and (vi) contained mechanisms for self-assessment. Our aim was to define a re-usable teaching component that was small enough to be written by one person.

In this paper we discuss general principles of courseware development and make recommendations that should be useful to any current or potential author of tutorial materials designed for use on the Web.

2. Life Cycle of CAL Materials

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 simply convert existing materials and not to make the best use of the available technology.

From a software development point of view, the use of the Web and related multimedia technologies suggests the adoption of proven software engineering principles such as prototyping and modular organisation. These principles are often embodied in methodology, which emphasises that software products have a clear *life-cycle* from requirements through design, implementation, evaluation and maintenance.

A standard model for the development of a tutorial includes these basic stages:

1. Determine the overall *objective* of the tutorial (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.

3. Determine, in more detail, the *aims* of each tutorial component, i.e., the *skills* and *knowledge* that a student will acquire on completion. 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 should involve the assessment of individual aspects of the tutorial 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 tutorial assessments.

In a typical tutorial lifecycle, a series of three prototypes are built: 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 (to verify the content as well as the software product) and adjustments are made before going into production.

Different immediate goals are best achieved with different multimedia topologies. Linear and hierarchical topologies are good for reimplementing traditional courseware with a more closed, didactic approach. More open-learning approaches are better served by network topologies - which are good for discovery learning - or constructional and simulation systems - which are good for skills learning.

A tutorial can often be broken into a number of stages (sometimes, screens) each of which contributes towards part of the learning experience. As such, the learning experience can be seen as a set of *events* [2], which need to be structured 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 student 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 events can be used as listed to produce a conventional, linear lesson, e.g. to introduce, present, practise and conclude. Alternatively, a number of them can be selected to give rise to a more exploratory, network-based learning experience, e.g. gaining attention, presenting a stimulus, eliciting performance and providing feedback in response to activities selected by the student.

3. Recommendations

In this section we give some general advice on the design of Web tutorials.

3.1 Design for Open Learning

Web-based tutorials are basically intended for self-study. Therefore, they are a form of open learning and should 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 in [6].

It is essential that each tutorial and perhaps each part of the tutorial should have clear *learning objectives*. On the basis of the objectives, students know what they can expect from the tutorial and decide whether they want or need to work through it. Students can also use the objectives to determine when they have sufficient understanding of the material at the end of the tutorial. Clear objectives also help teachers in deciding whether to use the tutorial for their course. Lastly, objectives are important guidelines for the designer when she selects and arranges the material to go into the course.

Try to make the tutorial *self-contained* as far as possible: this means providing sufficient introductory material and reference material that it accessible for a wide range of abilities. Match *pre-requisite knowledge* to the abilities of the expected audience, and give suggestions where students can get that knowledge.

Give the material a *well-defined structure*. Split the tutorial into chunks small enough to work through in 15-minute sessions to let the student work at his own pace. Provide suitable means for *navigation*, which allow the student to move around the material and which indicate his current position.

For each major section, provide *summaries*, which highlight central material. Use figures, graphics, tables and checklists to sum up important concepts and relationships between concepts.

Provide a means for *human help*. This could be e-mail to a teacher or on-line *via* a discussion group. Foster internet-based communication between students.

Provide opportunities for *feedback* from users and teachers to the author. Ideally get a cohort of potential users to *trial* the software in a way where you can study how the tutorial was used.

3.2 Design for Student-Centred Learning

Good tutorial materials *motivate* the student - they aim to explain why the subject is interesting and worthwhile to study rather than assume the student is already motivated.

Good materials let the student work out answers themselves and develop their own *self-esteem*. Learners are encouraged and rewarded to think for themselves.

The tutorial should be written in *a clear and concise language style* that is appealing to read; be friendly but not patronising. Keep sentences short, define jargon terms and avoid unnecessarily complex vocabulary.

Tutorials can be made interesting for the student through the effective use of *multimedia*. But while it is tempting to pack in as many graphics, animations and video clips as you can, these can also distract from the aims of the tutorial. For each graphical component, consider what role it is playing in learning.

Finally, consider the *computer literacy* of the likely users of the tutorial. Be prepared to give explicit instructions for any software referenced by the tutorial.

3.3 Design for Interactivity

A tutorial is not a textbook. It should not consist of a passive presentation of facts. Good tutorial materials are *centred on interaction*; they make learning an active process. Since the materials are designed for use without a human tutor, they should be designed to question and challenge the student's knowledge and understanding. CAL courseware packages have a number of means for posing questions and giving the student a choice of responses. Similar effects can be obtained with Web authoring languages. Use these facilities to let the student check their understanding of the material. You might even want to write the questions first and then create tutorial material to match.

Through interactivity, the student should be able to *learn from his/her mistakes.* When a student makes an incorrect response to a question, this is an opportunity for the tutorial to improve the student's understanding. Make sure your tutorial gives feedback for incorrect answers as well as correct ones.

Use the facilities for asking questions to *monitor the performance* of the student. Reward good progress and suggest other types of help for those who are struggling.

Select a *suitable topology* for the material that reflects the underlying conceptual structure: linear for logical development, hierarchical for material that can be studied at different depths, or a network for explorations. Too many constraints on the path the student takes through the material can be frustrating, while complete freedom may mean the student cannot plan an effective learning path

3.4 Design for Re-use

An advantage of dividing material into *manageable chunks* is that it makes it easy for others to re-use the material. Other teachers will find it easier to make use of your tutorial if they can select which modules are useful for their students.

Use widespread technologies, where possible, to make the tutorial readily *portable* to the maximum number of computing platforms. Remember that a large proportion of Internet users do not have access to fast network links, high-resolution displays or powerful workstations. If you use Java, JavaScript or Dynamic HTML, make it very clear at the outset which browsers are required to run the tutorial. If you use plug-ins such as Shockwave or Real Audio, provide users with a link from which they can get the software. The same holds for special fonts.

If you want your tutorial to be used by students from foreign countries, you need to design for *multilinguality* right from the start. Points to consider: • avoid culturally specific language: especially slang and euphemisms; • try to avoid text in your graphics, so that they do not need to be edited for translation; • use simple graphic icons for navigation.

4. Checklist

In this section we provide a simple checklist that authors of Web-tutorials may find useful. Readers may also wish to refer to [1] for an interesting article on the evaluation of tutorial material.

Goals

❑ Are learning objectives clearly defined?
❑ Does the tutorial contain means for self-assessment?
❑ Are these clearly related to the stated goals?

Relevance

❑ Is there a real student need for this tutorial?

❑ Is the content relevant for teaching and learning in the subject area?

Correctness

❑ Is the subject material accurate and up-to-date?

Pre-Requisites

❑ Are any necessary pre-requisites clearly stated?

❑ Are they realistic with respect to the intended audience?

Interactivity

❑ Does the tutorial allow interaction?

❑ Does it encourage active/exploratory learning?

❑ Does it create and maintain learner motivation and interest?

Size & Coverage

❑ Does the tutorial have a size appropriate to its value?

❑ Is the subject material covered in sufficient depth?

❑ Is the tutorial broken up into manageable pieces?

Re-Use

❑ Has the tutorial a design suitable for re-use by other teachers?

❑ Is the tutorial portable to other computing platforms?

❑ Are the computational requirements made clear at the outset?

Use of Language

❑ Is the language appropriate with respect to the intended audience?

Navigation

❑ Can users always see clearly where they are in the tutorial and what options are available?

❑ Does the path through the tutorial follow a logical progression?
❑ Is the material organised in such a way that this is obvious to the user?

Feedback

❑ Does the tutorial provide a means for obtaining human help?

❑ Does the tutorial provide a means for users to send feedback to the author?

5. Example tutorials

In this section we review some representative examples of Web tutorials in the light of the preceding discussion.

5.1 Web Tutorial on Knot Theory

`http://cst.uvic.ca/~mmania/knots`

Regarding *content*, the tutorial comes as part of a "chapter" on the mathematical theory of knots (other chapters concern graph theory, finite state machines). The entire chapter has a tutorial flavour, including an introduction, a series of knot related activities and puzzles, a section on practical applications of knot theory, a knotty story and *interactive* exercises.

The material, being of first-rate quality, checks against the criteria of *correctness*, *relevance* and *coverage*. Multimedia are used to great effect (e.g. animations of untying knots). Each tutorial level is of modest size (c. 10 mins) taking exercises into account. No formal *prerequisites* appear to be necessary and difficult language has been carefully avoided. In a general sense, the material has clearly been designed for *open learning*. For example, a lot of attention has been spent on organising the material appropriately so that the reader is encouraged to explore.

5.2. Tutorial on Bayes' Theorem by Stefan Waner and Steven R. Costenoble

`www.hofstra.edu/~matscw/tutorialsf3/frames6_6.html`

This tutorial is also part of a suite covering a variety of topics in finite mathematics that have been published to complement three published textbooks. Despite this, the tutorial itself is extremely informative, brief, and to the point. The contents are highly *relevant* to the needs of someone lacking knowledge in this area. The user's knowledge is continuously monitored by means of single, pertinent questions that test understanding of a well-defined concept, which are interactively assessed. Once the system establishes that certain prerequisites are lacking the user can be accurately directed to another tutorial, which supplies relevant knowledge.

5.3 Corpus Linguistics, by Tony McEnery and Andrew Wilson

`http://www.ling.lancs.ac.uk/monkey/ihe/linguistics/contents.htm`

We include this collection by way of contrast: it is clearly of great educational value, and the material is accurate, pertinent, and well planned. It also includes a useful glossary. Yet it is less obviously a tutorial in the sense addressed by this article. The general impression is more of a focussed collection of book chapters on a series of related subtopics. At the end of each chapter there is an interactive test with some exercises. Wrong answers are flagged by returning a link to the relevant part of the text.

Unlike the Bayes' tutorial, however, full understanding of this material depends on access to a related printed book and possibly other external references. Also the chapters are substantial in size which changes the feel of the educational experience.

Acknowledgements

This work has been sponsored by the SOCRATES Thematic Network in Speech Communication Sciences.

References

[1] Baumgartner, P. & S. Payr. (1997). Methods and practice of software evaluation: The case of the European Academic Software Award. In *Proceedings of ED-Media & ED-Telecom 97*, 138–145. Calgary, Canada: World Conference on Educational Multimedia and Hypermedia.

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

[3] Huckvale, M., Benoît, C., Bowerman, C., Eriksson, A., Rosner, M., Tatham, M., Williams, B. (1997a). Computer Aided Learning and Use of the Internet. In *The Landscape of Future Education in Speech Communication Sciences: 1 Analysis*, G. Bloothooft *et al.* (eds.), Utrecht: OTS Publications.

[4] Huckvale, M., Benoît, C., Bowerman, C., Eriksson, A., Rosner, M., Tatham, M., Williams, B. (1997b). Opportunities for Computer-Aided Instruction in Phonetics and Speech Communication Provided by the Internet. *Proceedings of the 5th EuroSpeech Conference*, Rhodes.

[5] Huckvale, M., Bowerman, C., Eriksson, A., Rosner, M., Tatham, M., Williams, B., Wolters, M. (1998). Computer Aided Learning and Use of the Internet. In *The Landscape of Future Education in Speech Communication Sciences: 2 Proposals*, G. Bloothooft *et al.* (eds.), Utrecht: OTS Publications.

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