Europe in the Round CD-ROM, Guildford, Vocational Technologies, 1994.

Europe in the Roundis a hypertextual/ hypergraphical database of facts about Europe, biased towards education and training, but taking in its stride many peripheral areas relevant to anyone considering living and/or working in, or even visiting, a member state of the EU – population, economy, institutions, geography, climate and so forth – all making up thousands of interconnected screens. These screens contain maps with a selection of zooms and hotspots, all kinds of facts and figures about available educational and vocational courses (presented both textually and, where appropriate, as graphs, bar charts and such like), details of life in each EU member state, details of cities and what they have to offer, and a lot more. There are also extra facilities, such as being able to set currency exchange rates or to create customized notebooks.

The PC version received for review (on CD-ROM – it is also available on floppy disks) has been written in Toolbook. The Macintosh version (the original) is based on HyperCard. Both were first published in 1991, and the design and structure of the software have changed little since then, but this latest issue contains more information, and is as up to date as software of its kind can be. Improvements and additions include a new section on regional statistics, new sub-sections on work and finance for students, a new section on EFTA, information on the European job vacancy system, enhanced road maps, distances between cities, a powerful index, generally improved links between sections, and more use of colour.

Three years (1991 to 1994) is a long spell in the world of educational technology, yet the user-interface of *Europe in the Round* looks as fresh as it ever did. And perhaps not surprisingly, since many of the innovations it brought with it have become well-tried features of multimedia CAL and reference software. The interface incorporates universal icon meaning and consistent positioning on the screen, never showing inactive icons, navigational clues provided by different transitions between screens, and the like, all of which shows how far-seeing the members of the design team were. The package lacks sound, though one cannot but agree with what Jacquetta Megarry (its producer) wrote not long after its initial publication (Megarry, 1991), that adding sound would in turn have added little or nothing in terms of usefulness, while eating up computer memory and human and financial resources. One could also say that adding sound would be as risky today as it was in 1991: every extra multimedia element brings with it an increased chance of instability.

I mention this because my practice these days when reviewing any multimedia package is to give it a tough test: as well as running it on a standard 'clean' machine (where there is rarely any problem, and indeed none with *Europe in the Round*), I run it too on a PC which conforms to MPC standards but which is also packed with all manner of cards and device drivers. I do this because not everyone has a perfectly clean machine. If a piece of software passes this test, I can be fairly certain of its robustness. Few programs do, nor did *Europe in the Round.* I had severe installation problems, including some non-sensical messages (such as inadequate disk space when I had over 100 MB free), and when I eventually managed to install the software by ignoring the instructions in the manual, it crashed or hung on me with irritating frequency. The other side of this coin is, of course, that the test was particularly acid. Furthermore, at the end of the manual is a troubleshooting section. This does not tell you how to get out of difficulties, but rather how to contact the publishers (far better than the usual advice about checking whether or not you have plugged in the computer). In other words, the publishers are happy to help with problems of installation or running. I should emphasize this praiseworthy attitude rather than the problems I encountered which were, after all, the result of difficult conditions. So I do.

In any case, this is my sole major complaint. Other complaints (such as the occasional use of the word *card* for screen, a long-standing HyperCard hangover, or the fact that the program assumes a VGA display and therefore does not use the full screen in SVGA mode on a large monitor) detract little from the overall impressiveness of the package. For it really is a veritable goldmine of information presented in a digestible form. Indeed, why it has not yet been taken up by large numbers of public information services throughout Europe is a mystery to me. It is certainly the best-presented information on opportunities for work and study in the EU, and its built-in facility for creating slide-show demonstrations seems tailor-made for public access: it should be available in the entrance halls of every educational establishment with European links. It is, very clearly, a tool which can be profitably used by students, whether in HE or not, whatever they are studying, and whether or not they intend to study or work in Europe. It also represents a fascinating case study in good practical CAL design.

Gabriel Jacobs, University of Wales, Swansea

References

Megarry, J. (1991), 'Europe in the Round: theory into practice', *Interactive Multimedia,* 2, 3.

Multilingual Multimedia – Bridging the Language Barrier with Intelligent Systems edited by Masoud Yazdani, Oxford, Intellect Books, 1993. ISBN: 1-871516-30-7.

This book is edited by Masoud Yazdani, who in addition to having written the preface, has a chapter to himself, and is joint author of no less than four of the remaining eight chapters. Yazdani is also the founder of Intellect, the book's publisher. Is this work therefore some kind of ego trip? Well, no more than a single-author book is, and on the whole the work is certainly worth reading.

The main theme is trans-language communication with the help of software tools such as grammar checkers and icon-based (non-linguistic) systems. Along the way, however, other themes pop up, some of which will be relevant to all readers of ALT-J.

The book opens with a descriptive piece about a prototype system for practising foreign languages in a restaurant, based around cartoons, sampled sound and hyperlinks. I have seen this program in action on a Macintosh. It is impressive when demonstrated but, as the authors of the chapter admit, its most serious limitation is that it does not allow for any alteration of the material.

The second chapter is concerned with a multilingual approach to databases intended to be accessed by speakers of different languages. It mostly says little other than comparatively obvious things: such databases ought to have multilingual interfaces, language-specific browsing facilities, non-linguistic systems of communication (such as icons, or video sequences without the use of language) where appropriate, and so on. However, the chapter does pick up when it treats cultural differences.

Chapter 3 is the first of two dealing with communication by icons and pictures. The idea of using graphics alone to communicate ideas is studied in a fair amount of depth, and while the design of such graphics is fraught with all manner of difficulties (neatly outlined here), it is clear that there is considerable scope for using pictures in a multilingual environment provided one is careful about which pictures to use.

By the time we reach Chapter 5, on generalizing language tutoring systems, we have left the soft stuff and have entered the realm of clause analysis, syntactic representation and lexical processing. We are also introduced to LINGER, an expert-system grammar-checking program for would-be programmers with little programming experience.

LINGER also forms the basis of Chapter 6, while Chapter 7 treats its successor known as eL. The problems involved in checking grammar are formidable, and the two chapters (especially Chapter 7) are at least useful in showing just how difficult it is to get a computer to 'understand' natural language.

Chapter 8 continues the theme of checking natural-language syntax by looking at some problems and possible solutions in this area when error-checking software is used with learners of English as a foreign language. There is a product review, together with a long comparison between the seven programs considered. The unsurprising conclusion is that all the programs fail, in different ways, in terms of the (rightly) exacting criteria used for evaluation.

The final short chapter is a critical assessment of Artificial Intelligence as used in language teaching, again comparing some available products.

For me, the downside of this book is that it appears to have been put together somewhat artificially. The unifying motif in the mind of the editor has clearly been that of addressing the problem of multilingual electronic communication, but the reader has to work hard to see some of the direct connections between chapters. Furthermore, there is an appreciable disparity between material of interest to researchers but probably beyond beginners, and more basic material for relative novices.

Nevertheless, the book will be of interest both to those concerned with computer-assisted language learning and those concerned with making courseware in general available to speakers of different languages. The chapters on iconic communication will also interest courseware developers in all disciplines.

Gabriel Jacobs, University College of Swansea

Using Records of Achievement in Higher Education edited by A. Assiter and E. Shaw, London, Kogan Page, 1993. ISBN: 0-7494-1111-2.

This is the latest in the *Using*... series of books, and it will be of particular interest to all in higher education who have responsibility for curriculum design and delivery. The material presented is drawn from the Using Records of Achievement (ROA) conference organized by the HEC in 1992, and like that event the main thrust of contributions is very much directed towards educational practice, the enhancement of student learning and informed professional action.

The origins of ROA and the use of associated portfolios are rooted in two perceived deficiencies of the higher-education system. First, an acknowledgement of the limitations of conventional examinations in doing justice to the full range of learning achieved by students; and secondly a growing realization of the educational value to students of involving them more fully in the reviewing, monitoring and recording of their own learning. This latter stance has, of course, been at the centre of Enterprise and Capability programmes, although its importance is also to be found in a much older tradition which values autonomy in learning as one of the distinguishing features of university education.

The publication is timely, and presents a nicely balanced blend of case studies, contextual issues and practical guidelines presented in an easy-to-read format. Both new and experienced staff will find much to interest them here, and the use of bullet-point listings works well for readers who want to engage with the text quickly. Some 22 short chapters are presented under a number of organizing terms:

- the use of ROAs and profiles for access, admissions and accreditation of prior learning;
- the development of skills, both personal and professional;
- the empowerment of learners through insight into their own motivations, needs and actions;
- the facilitation and assessment of work-based learning.

Any reader pondering these issues will be interested in the cases and descriptions of practice from fellow-travellers who have already grappled with them and who have begun to formulate responses. It is particularly welcome to see the odd contribution from employers interleaved with those of academic staff, and salutary to note that others from very different contexts may have much of value to share with us. To this reader, the material certainly achieves its stated intention of 'providing a glimpse of what others are doing and what they have learned by doing it', and herein lies its paramount strength.

An important sub-theme running through the variety of perspectives is the deployment of learning contracts and their importance as a mechanism for mediating the needs and interests of students, tutors and employers. First-hand accounts of the management of these processes provide valuable insights into the practical difficulties surrounding attempts to introduce innovation into higher-education institutions.

If there is a deficiency in the book – and this is a small one – it is that the students' voice and perspective is not placed alongside those of the other key participants. This is an irony, given the commitment in many of the sections to a student-centred approach and learner empowerment. Despite that reservation, however, this volume makes a significant contribution to what is an evolving area of work in many institutions, and as such is recommended as an excellent starting point for those considering similar initiatives.

Gus Pennington, University of Teesside

Handbook of Educational Technology by Henry Ellington, Fred Percival and Phil Race, 3rd edition, London, Kogan Page, 1993. ISBN: 0-7494-0849-9.

This publication gives a taste of a wide field, and is supported by an extensive glossary of terms used in educational technology, and a bibliography of recent articles and books.

The authors go to some lengths to explain that by 'educational technology', they are emphasizing more the sense of 'the technology of education' than 'technology in education'. In this respect, they deal extensively with the way in which effective education may be crafted or engineered rather than considering the technological devices which might be employed. To this end, the book has chapters on basic educational strategies, educational objectives, instruction and learning techniques, assessment and evaluation, and includes a chapter on resource centres. Each chapter is well structured with a basic but detailed outline of the topics under consideration, with their advantages and disadvantages.

In the first chapter, the authors introduce a simple model of learning comprising the four stages of 'wanting, doing, feedback and digesting'. This is an interesting addition to the informative but rather dry chapter. The authors continue to work with this model throughout the book, often referring to it to support a particular instructional technique or educational approach. This helps the book to be more cohesive and rounded than it would otherwise have been, and will be very helpful to a reader with little or no exposure to theories of learning. The authors also posit a simple idea that there is a continuing trend from mass education, through individualized education, to group learning, something which is again used throughout to link otherwise diverse sections.

However, the reader whose interest is in technological devices which can facilitate educational techniques, and moreover is looking for basic detail on the specific educational applicability of hypertext or hypermedia, groupware or cognitive strategies, or is keen to delve into the wider social or political aspects of the introduction of educational technology, will be disappointed. And the disappointment may be particularly acute in the last chapter where four out of seven of expected future trends in education have recently been under considerable attack by Government ministers (two others are supported, but possibly for reasons of cost-saving or social engineering, rather than in the interests of effective education).

Chapter 10 deserves a detailed look in the light of my overall criticism of a lack of being up to date: it offers a quite limited review of computers in education. I have not read the earlier editions of this handbook, but the chapter has a rather dated look-feel, with illustrations of a monstrously huge mainframe and of 'typical' textual computer interfaces capable of displaying only very basic graphics. Although the end of the chapter seems to have been updated to include some sections on email, hypermedia, CD-I and Photo-CD, these do not seem to hang together very well, and the chapter as a whole still seems out of date. Unfortunately, in such a fast-moving field, the successive updating of a handbook first published in 1984 leaves an overall effect rather like that of an archaeological dig, where each era overlays another. A handbook on educational technology that has more on filmstrip projectors than hypermedia must be a candidate for a complete re-write.

This is a suggested handbook for trainee teachers and other students of education or computers in education, and as a basic introduction I suppose it works reasonably well. Supplemented as it is with a good bibliography, it will provide a fair starting point for any student, or indeed lecturer, with no previous exposure to educational theory. However, any novice in the field of educational technology would have to be encouraged to move on to more comprehensive texts as quickly as possible to avoid the over-simplified Weltanschauung promoted by this publication.

Terry King, University of Portsmouth

The Principles of Screen Design for Computer Based Learning Materials by Alan Clarke, 2nd edition, Moorfoot, Sheffield, Learning Methods Branch, Employment Department, 1992.

This book aims to inform authors of computer-based learning materials of findings in the research literature relevant to screen design. In reality, it draws on work that relates to information presentation in general, whatever the medium. There are chapters devoted to electronic text, colour, and graphics, with a collection of guidelines at the end of each chapter (the guidelines are also summarized at the end of the book). Unlike many other sets of guidelines, this book does actually go into some detail of the research it draws on, rather than simply providing references. However, although this format allows the reader to make an assessment of how generalizable the guidelines may be, it does make it harder to find relevant advice, a problem compounded by the lack of an index.

Apart from minor niggles with the number of typographical errors, ungrammatical sentences, and the use of sexist language, I have other deeper concerns with this book. My first is that it contains no discussion of the design process itself. I found no mention, for example, of involving users in developing material, no mention of evaluation, and no mention of iterative design. Although none of these things can guarantee producing good displays, they will certainly be at least as effective as guidelines in helping to avoid unusable ones.

A second concern is that the book contains an uncomfortable amount of what might be termed naive psychology. For example, the assertion that 'long term memory seems to operate in a similar way [to short term memory]' (it doesn't), and the suggestion that Miller's classic 'magic number seven' paper on the limits of short-term memory informs the decision of how many colours to use in any one display (it doesn't). This simplistic style of presentation does little to point to the importance of considering psychological factors when designing screen displays and learning materials (see, for example, Helander 1988).

A third concern is that the book is not comprehensive. Seen simply as a collection of guidelines, there are surprisingly few of them (less than 80 in all). There are also surprising omissions from the list of references. For example, there is no mention of the classic work edited by Smith and Mosier (1986) who list over 900 guidelines (along with supporting references) for designing user-interface software, of which almost 300 are concerned with data display alone. Finally, there is little discussion of the way particular authoring tools impose their own style on the finished product, or of the recommendations made in the interface style guides produced by Apple and Microsoft (for example, Microsoft 1992).

Interestingly, Clarke is careful to distinguish between guidelines ('hypotheses which [are] supported by evidence and dependable within limits') and principles ('guidelines which [are] dependable in a wide range of situations, locations and with different groups of learners'). He concedes that much of the material presented in the book should be considered as hypotheses and guidelines, rather than principles, which makes one wonder about the title of the book. However, if you are looking for a few simple guidelines for a computer-based learning project, it may be a useful starting point, even if, in my opinion, Smith & Mosier or Helander would almost certainly be better.

Overall, though, I have to confess to a certain scepticism that mechanically applying guidelines, whatever their source, will help anyone to design effective displays or learning materials. Involving users at an early stage, conducting proper evaluations, and using a strategy of iterative design are far more likely to produce successful results.

Tony Gillie, University of Reading

References

Helander, M. (ed) (1988), *Handbook of Human-Computer Interaction,* Amsterdam, North Holland.

Microsoft (1992), *The Windows Interface: An Application Design Guide,* Microsoft Press.

Smith, S.L. and Mosier, J.N. (1986), *Guidelines for Designing User Interface Software,* The Mitre Corporation, Report ESD-TR-86-278. Copies of this monograph are available free of charge from Cambertown Ltd, Unit 8, Goldthorpe Industrial Estate, Goldthorpe, Rotheram, South Yorkshire S63 9BL. Quote reference 50-OL121 when ordering.

Multimedia Mania by D. Paulissen and H. Frater, Michigan, USA, Abacus Software and Dusseldorf, Germany, Data Becker, 1993. ISBN 1-55755-166-9.

It is well known that information is more effectively communicated, and messages more persuasively delivered, through the use of an appropriate mixture of text, sound and images. What is new about current approaches to multimedia is the technological opportunity that now exists for delivering such information via computers, but of course handling the various types of data involved is not easy. Indeed, in order to complete a multimedia project successfully, specialized hardware and software are usually needed, as well as relevant experience. Unfortunately, few (if any) of the currently available books on multimedia cover all the necessary techniques in a way that is suitable for newcomers to the field. *Multimedia Mania* is a two-part polymedia publication that attempts to fill the gap. It consists of thirteen chapters, three appendices and a subject index (all printed on paper) and a CD-ROM containing a large number of demonstration and shareware programs.

The first chapter briefly describes various multimedia applications, providing a useful introduction for newcomers. The second chapter describes the multimedia PC (MPC) standard and the various technologies (CD-ROM, audio, video, image acquisition, etc.) normally associated with multimedia.

Chapters 3 and 4 will be of interest to those involved in acquiring and installing a new MPC system. Configuring a multimedia PC is often a tricky job – particularly, for sound recording and playback. And in this context, the tips provided here will be found very helpful, especially with respect to the installation of sound cards. Following on from this, the fifth chapter (on sound recording) describes the processes involved in making digital recordings using a PC. The accompanying CD-ROM is used to expose readers to a sample music database developed using Microsoft Excel as a retrieval engine. It also takes readers through a hands-on session with the EZSound FX sound-editing utility.

Capturing and processing images are covered in chapter 6. Here, the Hijaak screen capture program (provided on the accompanying CD-ROM) is described in some detail – anyone who intends to prepare a multimedia presentation will find this software useful. Scanners and image processing are the other two important topics covered in the chapter, which also provides up-to-date information on Kodak's Photo CD.

Chapter 7 provides very good coverage of animation and its associated techniques. The examples presented include animation with Toolbook and Autodesk Animator. Examples of 3-D animation are also there, and again, the demonstrations provided on the CD-ROM will be of help in learning about this topic. For those interested in MIDI, chapter 8 will be of substantial practical value: it contains a wealth of tips and guidance on musical applications of the MPC.

Sounds and pictures as essential components of multimedia presentations are the subject of chapter 9, the style and content of which make it of significant value for those starting off in multimedia. The wide range of demonstrations (based on MMPLAY, TEMPRA SHOW, Authorware Star, Excel Slide Show, and PictureBook Professional) provide excellent hands-on experience. Authorware Star (which looks very similar to Authorware Professional) provides a useful way of sampling some of the features of a more sophisticated authoring system. The use of Authorware Star is further discussed in chapter 10 on authoring and hypertext, where methods for creating hypertext systems using Microsoft's Viewer are also described, as is the use of the Multimedia Development Kit (MDK).

Digital video is covered in chapter 11 which is in some ways disappointing: although a number of different video cards are now available commercially, the authors of this book choose to discuss only the MicroEye video output card, which obviously this limits the scope of the treatment of video hardware. The software-based approach for handling video is illustrated, however, using Microsoft's Video for Windows – a far more widespread product – which is discussed in some detail. And it is good to see that the authors have not omitted the important topic of virtual reality, treated in Chapter 12.

The last chapter, on Visual Basic, will be of particular interest to programmers who want to write code for multimedia applications. The examples on the CD-ROM will help them explore and experiment with the MCI (Media Control Interface) commands within Windows 3.1.

The three appendices provide a glossary, an overview of various types of multimedia products, and a summary of the material held on CD-ROM. The second of these gives a wide range of information sources that multimedia developers will be glad to have: it includes details of animation software, authoring systems, image editors, MPC upgrade kits, PC sound cards, sound editors, video editing systems, virtual-reality equipment, and suppliers of CD-ROM magazines.

This book will undoubtedly be a useful information resource for anybody wishing to enter the field of multimedia PC computing, and the tips it offers, as well as the CD-ROM bundled with it, make it a good buy.

Ashok Banerji and Check Meng Tan, University of Teesside

Designing Usable Electronic Text – Ergonomic Aspects of Human Information Usage by Andrew Dillon, London, Taylor and Francis, 1993. ISBN: 0-7484-0113-X.

Increasingly, text is being viewed not in printed form but on a computer screen. This book is therefore a timely one, as it concentrates on the ergonomic and human-factors issues underlying the design of electronic text.

It has ten chapters, a bibliography, a single appendix and a subject index. Following a short introductory scene-setting chapter, the author moves straight into a discussion of usability and the potential utility of ergonomics and human factors within this area. This is followed by a review of the experimental literature on reading from paper and from screens. Two basic strands are considered: outcome measures (speed, accuracy, fatigue, comprehension and preference) and process measures (eye movements, manipulation and navigational issues). The review is organized into three basic parts, covering reported differences between paper and screen, analysis of differences in terms of physical, perceptual and cognitive processes, and issues relating to text and task variables.

In chapter 4, the author examines the value of existing human-factors literature to electronic document designers. The problems of applying this knowledge to the design and evaluation of an electronic-text system are then illustrated by means of a case study (an interactive document retrieval system based on a CD-ROM called ADONIS).

Chapter 5 delves into the problems of classifying texts by means of repertory grid analysis using the FOCUS program, and the implications of the results for electronic text design are then discussed. Chapter 6 explores in more depth readers' interactions with two particular types of text – academic journals and software manuals. The methodology used in the investigation is based on the WWH (Why, What and How) approach to document usage). The chapter concludes with a discussion of the design implications of the research findings.

The material presented in chapter 7 is concerned with the literature dealing with readers' impressions of structure and shape in information space (based on schemata theory), and how this might relate to navigational problems within electronic information. Two simple experiments designed to investigate document structure are then briefly described.

In chapter 8, the author proposes a framework which represents the ergonomic factors involved in using a text, and suggests the variables to consider when designing an electronic document. The framework consists of four interactive elements which reflect the issues dominating a reader's attention at various stages in a reading process. The components of the framework include the task model, the information model, a set of manipulation skills and facilities, and a serial reading processor.

The validity and utility of the proposed framework are explored in chapter 9. This describes two experiments involving readers' use of an ordinary text (on paper) and electronic texts (both linear text and hypertexts) for information retrieval. Various tools were employed to produce the electronic texts (HyperCard, TIES, a word processor and GUIDE). The experiments were conducted in a usability laboratory, and verbal protocol techniques were used to analyse readers' behaviour.

In the final chapter of the book the author summarizes and discusses what has previously been said, and then turns his attention to potential applications of the framework – particularly for the development of systems based on the use of electronic text.

I found this book quite useful. It draws together much of the documented scientific knowledge on reading processes, and reviews its worth in the context of comparing paper-based and screen-based visual interaction with text. Although there are many more avenues yet to explore, the ideas contained here provide some useful starting points on which to base future research.

Philip Barker, University of Teesside

CAL Into the Mainstream: Computer-Assisted Learning (selected contributions from the CAL 93 Symposium, 5-8 April 1993, University of York), edited by M.R. Kibby and J. Hartley, Oxford, Pergamon Press, 1994. ISBN: 0-08-041945-3.

It is somewhat difficult to choose the most useful way to summarize a book of selected contributions from a conference. The reader does not know if the contributions selected (in this case, 21) are meant to represent as faithfully as possible the range of interest areas and contributions presented at the conference, or if they are chosen on a basis of attempted balance among institutions or regions represented among the participants, or some other criterion such as a mix of 'old' and 'new' contributors. Perhaps the standard is purely scientific: the editors' opinions of the most valuable entries. But because of this uncertainty, the reader cannot fairly approach the book as an inventory of British activities with computers in education, nor as conference proceedings. Thus, how to proceed? By treating the book as a book, and thus expecting that the title is an organizer for the content? This approach does not help very much either, in that the reader entering the book through interest in its title gets very little sense of how much computer-assisted learning has gone 'into the mainstream' of UK education after reading the entries. This said, perhaps the only way to address the book is as a hard-covered journal, with a collection of 21 articles offered without any overt editorial synthesis.

That rather long-winded introduction is by way of explaining my choice of procedure for this review. I will give a general comment about the topics and settings addressed by the articles, then note some that I have found to be particularly useful.

Thus the overview of contents: 17 of the 21 articles are from UK author teams, one a team of mixed UK and non-UK authors, and three from other countries (Canada, Germany and Finland). As to content and setting, my analysis is that:

- three deal with software development methodology;
- ten could be described as 'specific-case' studies situated at the university level,
- seven of which I would most generally categorize as case studies of particular prototype packages, including descriptions of the theory of their designs and development processes and evaluations based on the author/designers' experiences with the in-house use of the package,
- one of which is a more cognitively-oriented study of an experiment involving an existing package (a concept mapping tool),
- and two of which relate, in different ways, to distributed learning (OU students' experiences with home computer use in a particular OU course, and experiences at Lancaster with CMC);
- four relate to experiences with children as they use a particular software environment;
- one, from Finland, studies a group of lower-secondary students and a LEGOLogo experience;
- three, including one from Canada, look at groups of primary students as they use either modelling software, or a prototype developed by the authors, or a CD-ROM 'talking storybook';
- two describe case-study type observations about groups of students in initial teacher education at two institutions (articles 17 and 19 – these articles were so similar in style and substance that I looked back twice to see if I was not reading the same article a second time); and
- two describe, from large-scale and longitudinal perspectives, implementation experiences about computers in education.

The latter two in the categorization I offer above are, in my opinion, the two articles which most substantially relate to the title of this book. They are the two I solidly recommend to anyone interested in what happens when large-scale initiatives try very hard to stimulate computer use in schools. They are also two of the three articles from the collection I have already cited in my own writings, and will continue to use as valuable references, not only because they are based on broadscale samples of experience, but also because of the value of the insights they offer.

The first of these two, whose title does not do much to suggest its contents, is by Alan Brown ('Processes to support the use of information technology to enhance learning') and is based on an evaluation of the UK Information Technology Teacher Training Development Programme (1988-1992). It presents many rich and useful insights from seven projects focusing on the management of IT use by teachers, and ways teachers use IT to enhance learning in the classroom. The second, by John Gardner *et al.* ('Learning with portable computers'), is a fascinating study of a broadscale attempt, involving nine schools and 235 pupils, to give students unlimited access to personal computers (at home and at school) over an entire school year, then to investigate the impact of this access on their performance in core school subjects. The article, whose findings deserve careful study by anyone interested in 'the mainstream', again points out the critical influence of the teacher in what happens with computers and learning.

The third article I found particularly useful is by Mike Aston and Bob Dolden ('Logiciel sans frontières'), again a title which will not help the reader to identify its contents if seeing only a list of titles. This article is of less general interest than the ones by Brown and Gardner *et al.*, but of particular value to those involved in software development from a broad perspective. The article summarizes some of the experiences from a number of European-level projects involving cross-national educational software portability.

As for the rest of the articles, well, I certainly can say they are well-written, well-edited, and a number of them I have noted for myself because of their relation to my various research interests. But the reader will most benefit from them by seeing them in wider contexts. Studies of the generic type: 'What a certain group of our students did with our prototype and what we have learned from the experience', for example, are most valuable when one can place such observations into the context of other studies involving similar cases in order to see where trends and insights transcend the particular case under description. For the reader interested in the various types of software or IT-use situations described in the case-study-like articles in this book, there will certainly be good references to add to his or her collection. For example, because I am currently very interested in CMC, I quickly went to the article by Steeples *et al.* on the topic, and am pleased to add it to my CMC Insights and Experiences file.

One accepts a journal as a collection of articles, and the articles included here are quite interesting. Nonetheless, one would like a book, destined to live for years on library shelves, to go one step beyond the collection level to bring some sort of coherence in the articles for the reader who in the future seeks out the book because of its title. One hopes that those readers will find the articles by Brown and by Gardner and his colleagues, if they want a reflective summary of a wide set of experiences from the field, from mainstream practice.

Betty Collis, University of Twente, Netherlands

NOTE

The content of this book also forms Volume 22 (Issues 1 and 2) of the journal *Computers and Education.*

Hyperprogramming: Building Interactive Programs with Hypercard by G. Coulouris and H. Thimbleby, Wokingham, Addison-Wesley, 1992. ISBN: 0-201-56886-1.

This book is aimed at those wishing to develop HyperCard applications and who already have some programming experience (for example, in a conventional third-generation language such as Pascal or BASIC). The authors suggest that their work is suitable for a number of target reader groups: novice programmers, expert programmers, students, advanced students, and HyperCard developers, and individual chapters are identified in the Preface as being of particular relevance to each target reader group.

A floppy disk containing HyperCard stacks accompanies the book, and complements it by providing the HyperTalk code discussed within it. In this way, readers with a Macintosh and HyperCard are able to run the programs and, more importantly, cut and paste both objects and code for use in their own programs (known as stacks in HyperCard).

There are a number of books about HyperCard, but this one is quite novel in its approach. It goes straight into HyperTalk (without lengthy description of the HyperCard interface) and combines this with the pedagogic strategy of teaching by example. Throughout the book, example code is used to introduce new HyperCard and HyperTalk topics to readers. A key benefit of this topic-based approach is that it presents material in problem-oriented ways. Readers are thus able to discover hints and tips while learning about programming in HyperTalk. They are also presented with solutions to some of the practical problems that might be encountered when they come to develop their own HyperCard stacks.

There are fifteen chapters, all oriented towards practical use. The first provides an immediate introduction to programming in HyperTalk. This is followed by two chapters on the basics of working with HyperCard and HyperCard objects. The fourth chapter then describes the basic commands and control structures used in HyperTalk programming.

Chapters 5 to 11 adopt the topic-based

approach mentioned earlier, with each chapter presenting a new theme. They take the reader through a wide variety of practical programming issues while presenting new syntax, structures, and tools. Programming topics covered include graphics, timing, animation, menus and searching.

Chapters 12 to 14 deal with the concept of hypertext. A number of issues are addressed: design, authoring, reading (browsing), and so on.

The final chapter presents ideas and strategies for undertaking a variety of projects in HyperCard; the aim is to provide some indication of the flexibility and scope of environments like it. A wide range of problems are discussed, including the development of books, games and a simple spreadsheet. In addition, techniques for protecting software from tampering and virus attack are outlined.

The two appendices are useful. The first provides lists of information about HyperTalk (properties, commands, messages, constants, operators and functions). The second gives a set of comparisons between HyperTalk and Pascal, designed to allow readers with existing Pascal experience to transfer rapidly between the two environments. This second appendix is very well done: the structures used in Pascal are presented with the HyperTalk nearest equivalent presented alongside.

This is an excellent book for anyone wishing to program HyperCard applications, particularly if they already have some programming experience. It also has some potential for people who wish to program in Toolbook on the PC, an environment similar to HyperCard.

Stephen Richards, University of Newcastle-upon-Tyne

STELLA II version 3 (Simulation and Modelling for System Dynamics), Apple Macintosh version.

When STELLA was introduced several years ago, it provided the first opportunity for relatively unskilled workers to solve complex problems which normally would have required skills in algebra and calculus. STELLA was designed to model the behaviour of dynamic systems: that is, those in which the values of various components of a system (the system variables) vary in a complex manner with time. For example, the variations in populations in an ecosystem, or drug levels after taking a medicine, vary in a time-dependent manner which can be calculated from a physical model of the system. Normally, the study of dynamic systems requires the use of calculus to set up and solve differential equations; these describe the interrelations between the quantities of interest. Setting up a dynamic model involves two steps; firstly, the identification of the interrelationships between the variables, and secondly, the writing of differential equations which embody these relationships in an algebraic manner. Finally, the solution to the differential equations reveals the time-dependent behaviour of the system variables.

Prior to the introduction of STELLA, there were a number of rather unfriendly programs available which assisted users in setting up the differential equations, and solved them numerically. Solving the equations has always been considered the 'hard' part of the problem. Unfortunately, if you do not have the skill or training to set up the model or equations, then *that* is the hard part, and finding the solution is irrelevant. STELLA allowed the user to define the problem simply by drawing boxes, which represented the time-dependent quantities in the system, and joining them by flows, which represented the inter-relationships between the quantities. The basic form of the system was defined by the structure of the diagram, and only very minor mathematics was required to define the flows. STELLA then transformed the model into a set of differential equations, and solved them, thus showing how the model system changed with time.

The first version of STELLA was simple and in some ways did not conform to the usual Macintosh interface; however, its idiosyncrasies were easily learnt, and it was widely used. In this form, it was capable of studying most of the problems which its users posed; as long as a simple box and flow model could be used to represent the problem, STELLA could provide an indication of the likely behaviour. Despite this, there were a number of deficiencies: some fairly simple problems could not be broken down into a box-and-flow form, and there were limits to the presentation of the output, which was either a copyable string of numbers or a PICTable graph. STELLA II began to address these problems, and STELLA II 3.0 adds yet more functionality.

The new modelling tools added to STELLA II allowed processes to be used which could only poorly be represented by boxes. These include the conveyor and the oven. A conveyor is a structure into which you can feed material at one end and it will appear at the other end after a delay. This is a much more realistic model of many processes. Take, for example, a model we recently used to study the flow of material through the gastro-intestinal tract. Originally, we used a sequence of boxes to represent the stomach, small intestine, and large intestine. The problem with this model was that as soon as the stomach had emptied some material into the next box (the small intestine), it could be further emptied into the large intestine. Of course, this is not a physically realistic model since the small intestine is a long tube and, as we know, it takes time for material to get to the far end. A conveyor is a much more realistic structure to model this behaviour. STELLA II also allowed one-way or two-way flows, and most usefully introduced a non-negativity constraint. In the earlier versions, accidental negativity of a variable had to be checked for by the model builder, which was most inconvenient. These additions, in our opinion, covered most of the requirements of the average user.

STELLA II 3.0 adds to both the computational and presentation areas. Most of the changes have to do with presentation of the results. There are now two versions, STELLA II 3.0 ordinary, and STELLA II Authoring. This is in line with many programs which now provide a full-featured version for developers, and a simplified version for end-users of the models. We were only able to test the authoring version, so cannot comment on the lesser facilities of the ordinary version. However, the claimed features of the authoring version over the other version include the ability to create stand-alone models ('microworlds' – ugh!) which can be locked so that students can explore their operation but not mess about with their structure. The authoring version also has a mapping mode to aid the construction of complex models, and an overview mode to display the broad structure of the model. This level allows the importation of graphics and Quicktime animation (which we did not test). In this aspect, STELLA 3 (let us call it that) becomes more of a presentation package than a laboratory program, which will be useful for people who need to use simulations as a teaching or negotiation tool. The ability to create stand-alone models is most useful for teaching purposes (students may not need the full ability to create models).

There are rather fewer computational additions to STELLA 3. The built-in function list is slightly longer with 11 new items. Publish and subscribe are supported, which may allow some interesting connections to other calculation packages such as Excel or Mathematica. A valuable feature is the ability to plot several runs of a model on the same graph to allow comparison of changing a particular parameter. In previous versions of STELLA, different graphs had to be drawn and each plot locked before a parameter could be changed and the model re-run to produce the new graph. This made comparison very hard and necessitated printing each individual graph.

One of the most serious difficulties with STELLA and STELLA 2 was that they were available only for the Macintosh. 'Is there a PC version?' was probably the most commonly asked question we encountered in our years of showing the program to our colleagues. STELLA 3 promises to provide model transferability between Mac and PC; however a PC version was not available to us so we could not try it.

The review version of STELLA 3 was provided on a single disc and occupied 1.3 megabytes of space. It loaded without difficulty onto a Macintosh 2VI and a 2CI, both running system 7 with the usual heterogenous array of extensions, and claimed 1.5 megabytes of memory, although this could be reduced to 1 megabyte. We did not perform any timing comparisons with earlier versions, but the simple models we created ran smoothly and quickly, giving the same numerical results as those from STELLA 2. Models created in STELLA 2 opened without difficulty in STELLA 3.

Despite the major additions, it must be said that the original version of STELLA has a simplicity which allows new users to grasp its functions rapidly, and will do virtually all that is asked of it. STELLA 3 is easy to learn if you have used earlier versions, but its many features can confuse novices. It is a common situation with all forms of software; for example, all we usually need is old MacWrite and Multiplan, but now we have Word and Excel, with massive functionality that is rarely accessed.

Who will buy STELLA 3? We would expect that people needing to present their models to audiences will find the additions invaluable, and particularly teachers associated with computer-assisted learning in higher education. PC-based departments will benefit from the availability of the package on their machines. Research users of modelling software will find new features in STELLA 3, but most of the time they will not need the glitz.

Clive and Neena Washington, University of Nottingham

Note

STELLA II version 3 is available from Cognitus Systems, 1 Park View, Harrogate, Yorks, HG1 5LY (tel: 0423 562622). It is available only to bona-fide educational users; others must buy the equivalent commercial package. Prices are: Authoring £399, Standard £249; upgrades and LAN packs are also available.

The Open Learning Handbook – Promoting Quality in Designing and Delivering Flexible Learning by Phil Race, 2nd edition, London, Kogan Page, 1993. ISBN: 0-89397-392-0.

The transformation of higher education into a system adapted to servicing the needs of much larger numbers has created a demand for flexibility. Distance and open-learning methods free students from constraints of time and place, and can allow more individualized feedback than traditional teaching programmes. However, there is also an emphasis on auditing for quality – efficiency must be complemented by a concern for standards and excellence. This book is intended both for those already involved in the delivery of open learning, and those moving into the field for the first time. There are two main aims: firstly, to provide guidelines on good practice for those producing open-learning materials; secondly, to advise tutors and mentors on the effective support of open learning.

The book is a readable, attractive reference text, packed with very useful checklists. The author is clearly drawing widely on his own experience, and although not an open-learning package itself, the book provides a feel for the subject by incorporating many of the features of open learning: each chapter begins with an abstract and a list of learning objectives, and some contain self-assessment questions. Chapter 1 discusses the meaning of the term *open* learning, setting it in the context of the whole learning process. Chapters 2 to 7 cover the production of open-learning materials, from the initial design stage through to marking student assignments. Chapters 8 and 9 describe respectively the role of tutors and mentors. Chapter 10 focuses on the integration of open-learning materials into traditional courses. The book concludes with an annotated bibliography and an index. It should be noted that the production, choice and use of non-print media is not covered.

I would single out Chapters 2, 3, 4 and 6, on the design and production of open learning materials, as being of particular merit. In Chapter 2 ('Designing for open learning'), the author takes a realistic approach, opening with a discussion about the relative merits of adopting or adapting existing materials, or starting from scratch. A full description of the difference between open-learning materials and traditional textbooks is given. The importance of piloting materials is highlighted, alongside suggestions for obtaining meaningful feedback from the piloting phase. A Quality Checklist is also provided. Chapter 3 concentrates on the importance of showing learners where they are heading. Syllabus content is now widely expressed (GNVQ/NVQ, for example) in terms of intended learning outcomes: competences, performance criteria and range statements. The section discussing these terms is therefore particularly relevant. Chapter 4 focuses on the purpose and design of self-assessment questions, and the provision of poor, as well as good examples, is most helpful. Chapter 6, on the design of tutor-marked assignments, benefits from covering not only expected topics, such as the development of marking schemes, but goes on to discuss how the delivery of constructive, supportive feedback might be achieved effectively in open-learning situations.

Chapter 5 moves on to the importance of adopting informal user-friendly language when producing open-learning materials, and the readability of this book as a whole demonstrates the efficacy of this approach. However, emphasizing readability above all else can have disadvantages: using an example from this book itself, improving flow by deliberately omitting literature references in the text could hinder those readers wishing to research specific topics in more detail (despite the inclusion of an annotated bibliography). In the same vein, the lack of section numbers gives an informal appearance, but is an irritation when making reference to the book. There is also the danger that 'friendly' language can so easily become patronizing chattiness; thankfully, the author is too experienced to fall into this trap very often.

Despite my minor criticisms, I would have no hesitation in recommending this handbook as a useful resource for those interested in the development of open learning materials, and those involved in tutoring open learners.

Lorraine Warren, Hull