MODELS TO SUPPORT MULTIMEDIA COURSEWARE PRODUCTION

Philip Barker and Susan Giller

ABSTRACT
Many new technologies are now available for delivering and/or providing access to computer-based learning (CBL) materials. These technologies vary in sophistication in many important ways - depending upon the bandwidth that they provide, the interactivity that they offer and the types of end-user connectivity that they support. Invariably, appropriate combinations of the available technologies are needed in order to produce the most effective and efficient learning environment for any given application. Bearing this in mind, it is important to consider how multimedia resources, interactivity and global connectivity can best be used in order to produce a software product that best fulfils the requirements identified in any given courseware requirements specification. In this paper we discuss the types of model that are needed to create effective interactive, multimedia courseware. We also indicate the nature of the interactions that exist between these models and the ways in which these can be used to optimise the trade-offs that are inherent in the creation of multimedia CBL materials.

KEYWORDS
learning products, courseware, interactive multimedia software, design and development, methodologies, models, evaluation, project life-cycle

INTRODUCTION

During the last decade or so, multimedia computing technology has opened up many interesting opportunities for the creation of new types of teaching, learning and training product. These products have covered a wide range of subject areas and have been aimed at many different audiences. Typical examples of the types of product that are now available include: electronic books [10, 13]; educational games; various sorts of interactive product based on the use of compact disk (CD) technology [11]; intranet and World Wide Web pages [4]; and tools to facilitate 'electronic' knowledge sharing [3, 5, 6]. Together, products of this sort can be used in a variety of ways to enrich, quite significantly, the scope and quality of computer-based learning (CBL) experiences.

Within the remainder of this paper we shall use the term '*interactive multimedia courseware*' to refer to computer-based learning products in which some skill or knowledge is intentionally transferred to a user of that product as a consequence of its use. Such software can be defined in terms of 'learning products' in which optimal combinations of text, sound and images are used to achieve particular learning outcomes. This software also uses interactivity in various ways to achieve these objectives. Typically, interactivity is used: to facilitate navigation through the corpus of materials that make up the courseware product; to provide assessment

and feedback mechanisms; and to facilitate communication between the group of users that constitute the 'learning community' at which the product is focused.

Naturally, when designing courseware, it is important to consider how multimedia resources, interactivity and global connectivity can best be used in order to produce a software product that successfully fulfils the conditions identified in any given courseware requirements specification. Because of the complex nature of both software development and the cognitive processes that take place during learning, this is no easy task. The design and development processes therefore need to be guided by appropriate models that encapsulate 'best practice' with respect to both system design and implementation issues. In addition, relevant pedagogic models also need to be employed in order to ensure that the subsequent learning processes are relevant to the knowledge and skills that are to be acquired. Therefore, in this paper we shall discuss the types of model that are needed in order to create effective interactive, multimedia courseware. We also indicate the nature of the interactions that exist between these models and the ways in which these can be used to optimise the trade-offs that are inherent in the creation of multimedia CBL materials.

UNDERLYING MODELS AND METHODOLOGIES

There are two basic approaches to developing courseware products; we shall refer to these as the '*empirical approach*' and the '*theoretical approach*'. The first of these uses a strategy that is based essentially on a 'trial and error' procedure; that is, a learning product is produced and its effectiveness is determined; if necessary, an iterative approach is then used in order to improve it. In the second approach, appropriate theories and models are used, in so far as they exist, in order to create a learning product that falls directly within a given 'region of acceptability'. This latter approach is attractive because it reduces the amount of uncertainty involved in product creation, that is, the amount of wasteful trial-and-error activity that is often involved. Bearing this in mind, it is our opinion that design and development models (and methodologies derived from them) should play a fundamental underlying role in guiding the production of interactive multimedia courseware. The underlying importance of these basic tools is reflected schematically in Figure 1.

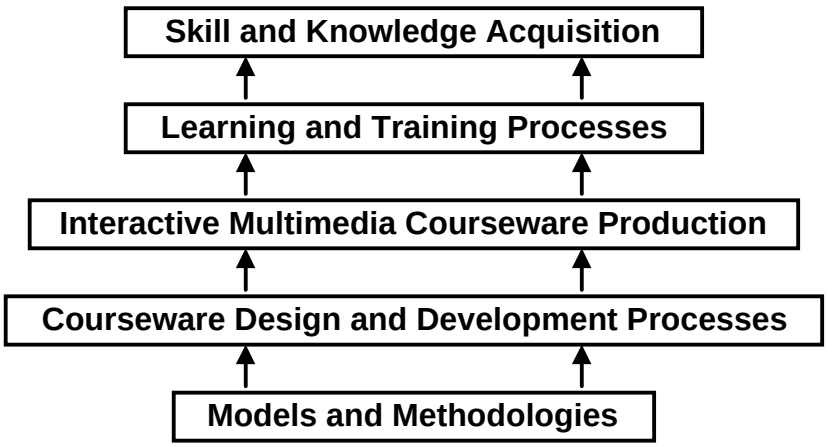

Fig. 1: The Role of Design

Some years ago we introduced a basic model that we felt could act as a basis for designing and developing interactive multimedia courseware. We referred to this model as the '*MAPARI*

model' [2]. MAPARI is an acronym for '**M**imicry, **A**pprenticeship, **P**ractice, **A**ssessment, **R**efinement and **I**mprovement'. The MAPARI model was based upon the concept of 'expert performance' of a physical or intellectual task. Its origins are rooted in the study of 'electronic performance support systems' [1].

Implicit in the use of the MAPARI approach is the application of an interacting set of lower-level design models that cover each of the aspects involved in the overall set of activities involved in creating an interactive multimedia courseware product. For example, there are *pedagogic models* that describe the basic nature of learning processes and the effects that learning materials are likely to have upon their users. There are also *design models* and *architectural models* that recommend how a particular product should be designed in order to meet a particular learning need. In addition, there are various s*ubject-based models* that define the relationship between the important topics and concepts within a particular learning domain. In addition, there are also *development models* that encapsulate 'best practice' with respect to the mechanisms involved in actually creating a learning product from its design 'blueprints'. Sometimes these latter models are referred to as 'methodologies'. It is this latter type of tool that we are primarily concerned with in this paper.

The need for the introduction of methodologies into software development first became apparent as large programming projects became increasingly complex. In 1981, Checkland [8] defined a methodology as *'… a set of principles or a method which, in any particular situation, has to be reduced to a method uniquely suited to that particular situation'*. Following on from this, Bruce and Pederson [7] stated that: *'a software development project is defined as a planned undertaking that is to result in producing one or more computer programs with specified functional capabilities. These programs are to be developed in accordance with a defined schedule and within the planned resource budgets: staff support; computer time; computer capacities in terms of throughput; memory and peripheral storage, and funding constraints.'*

The British Computer Society (BCS) Information Systems Analysis and Design Working Group in 1983 first defined an information system methodology as: *'a recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation, management and training for developers of information systems'* [12].

In practice, there are many models and methodologies now available for software development - particularly commercial ones. Undoubtedly, one of the simplest models for software production is the 'waterfall' model [14]. Of course, many more advanced models and methodologies now exist. Typical examples of these include: CASE tools, manuals; pro-forma documents; model building templates; education and training; and consultancy support. To date, these methodologies are mostly used in the development and implementation of business-oriented information systems. They have not usually been successfully used for the development of multimedia learning products. In order to produce multimedia learning products effectively and efficiently, it is imperative that development projects are carefully controlled and managed [9]. The production of high quality, robust multimedia applications is dependent on development projects that are adequately costed, well-planned, efficiently prepared and effectively managed.

Designing and developing multimedia learning products necessitates close co-operation between people with specific skills and expertise. Furthermore, the existence of diverse modes of storage and delivery necessitate the adoption of appropriate methodologies and suitable

design models in order to create high quality learning products. In the remainder of this paper we shall discuss some of the more important issues relating to the formulation of the various models and methodologies that we feel are needed in order to underpin the successful development of interactive multimedia courseware products.

In the following section of the paper we proceed to identify why development models and methodologies are needed. In subsequent sections we outline the essential features of an interactive multimedia courseware development methodology that has been successfully used for the production of a number of interactive multimedia learning products.

PROBLEMS ARISING IN MULTIMEDIA DEVELOPMENT

In principle, multimedia courseware can be developed by anyone who has the necessary equipment. However, because of the variety of specialist skills that are usually required, it is teams of people that most often develop multimedia learning products. These teams are usually based in 'private' multimedia companies (producing commercial applications), large corporate companies (that develop in-house training materials) and educational establishments (providing opportunities for the development of specific multimedia learning applications). Whilst many of the private multimedia production companies have adopted their own approaches to the analysis, design and development of their products, there are no generally accepted methodologies specifically designed to assist in the creation of multimedia learning products.

Naturally, as the complexity of the products and the size of the development teams increase, it becomes more and more difficult to manage the production process efficiently. Existing structured methodologies help in analysing data, the relationships between entities and the behaviour and events taking place within a system. Most of the methods stress the importance of operation, providing information, handling events, and so on. However, these elements are not so important in multimedia application development, where the process involves the integration of all the available resources into a single piece of software. In this situation, the emphasis is on how to present a topic to users in a way that increases their knowledge and/or skills. For this reason, the existing system development methodologies are not considered suitable for the creation of multimedia learning products.

The development of multimedia learning products is a complex task and, in addition to technical issues, problems can occur in all aspects of the development process. These often result in poor quality products or products that are delivered outside time and budget limits. Projects that result in inferior products are considered to be unsuccessful. There are a great number of reasons for project failure. These can include: poor project management, inadequate resources and budget; lack of skills, tools and equipment; insufficient development time or inappropriate learning strategies. Some of the most common problems arise due to poor communication and conflicting demands during the development process.

A multimedia project is normally composed of four equally important components as shown in Figure 2a. Ideally, if we had adequate models, it would be possible to predict exactly how much time should be spent on each of the four phases. Invariably, the time spent on each individual phase is not equal and is often not predictable. Indeed, as can be seen from Figure 2b, in reality, many multimedia projects spend a much greater proportion of time on resource and program development and relatively little time on planning and design, testing and quality

control, and dissemination and evaluation. Neglecting these major elements of the project life cycle could ultimately lead to a deficient product.

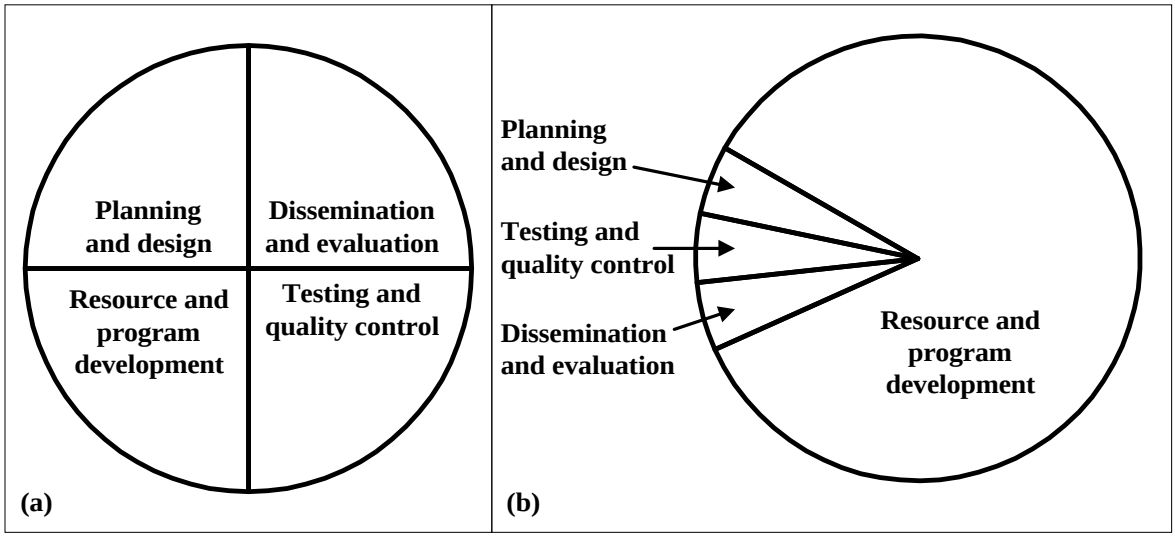

Fig. 2: Multimedia Project Organisation in Terms of (a) Major Phases and (b) Time Spent on These Phases

Of course, there are many other reasons why multimedia projects fail. For example, a survey of the members of the multimedia development team at Singapore Polytechnic [15] identified the following important points that contributed to the failure of multimedia development projects:

- lack of proper documentation and set procedures
- unclear roles and responsibilities
- poor analysis and feasibility studies
- insufficient testing and lack of quality control
- inadequate communication
- lack of guidance in regard to file naming strategies
- ineffective design scripts
- deviation from the actual design

The main contributing factors to unsuccessful interactive multimedia projects can be grouped into three major categories: poor design; ineffective project co-ordination; and inefficient communication channels. Undoubtedly, most of these arise due to lack of a structured approach. Therefore, in the following section of the paper we introduce a comprehensive methodology that is designed to improve the efficiency and effectiveness of multimedia development teams. We also illustrate how the methodology can be applied to the creation of multimedia learning products.

A METHODOLOGY FOR CREATING MULTIMEDIA LEARNING PRODUCTS

There are several good arguments in favour of adopting a sound methodology for developing multimedia learning projects. These include the necessity for good communication; consistency; effective quality control strategies; and accurate cost estimation. In the remainder

of this section we outline some of the important issues which an effective interactive multimedia courseware development methodology must address if it is to be successful.

Project Planning The scope of a project, its budget and its intended completion date are usually negotiated prior to the commencement of any development activity. Once a project is formally agreed upon, management strategies are identified, milestones are established and tasks and resources (including individual team members) are identified.

Team Management Most multimedia development teams comprise a number of people who contribute various skills and expertise. For example, project managers orchestrate the project life cycle, subject matter experts (SMEs) provide the necessary content input and courseware designers produce the 'blueprint' for the product. Development staff, consisting of resource developers (graphics, animation, video, and audio) and programmers 'build' the program.

Effective team management helps to establish a good working atmosphere, deals with problems more readily and shares the workload efficiently [9]. Figure 3 shows the structure of a typical multimedia development team. In this diagram, the impetus is provided by the courseware designer and project manager who co-ordinate the planning, design, development and testing throughout the project life cycle. The output in this case provides the necessary documentation to be used throughout the project life cycle.

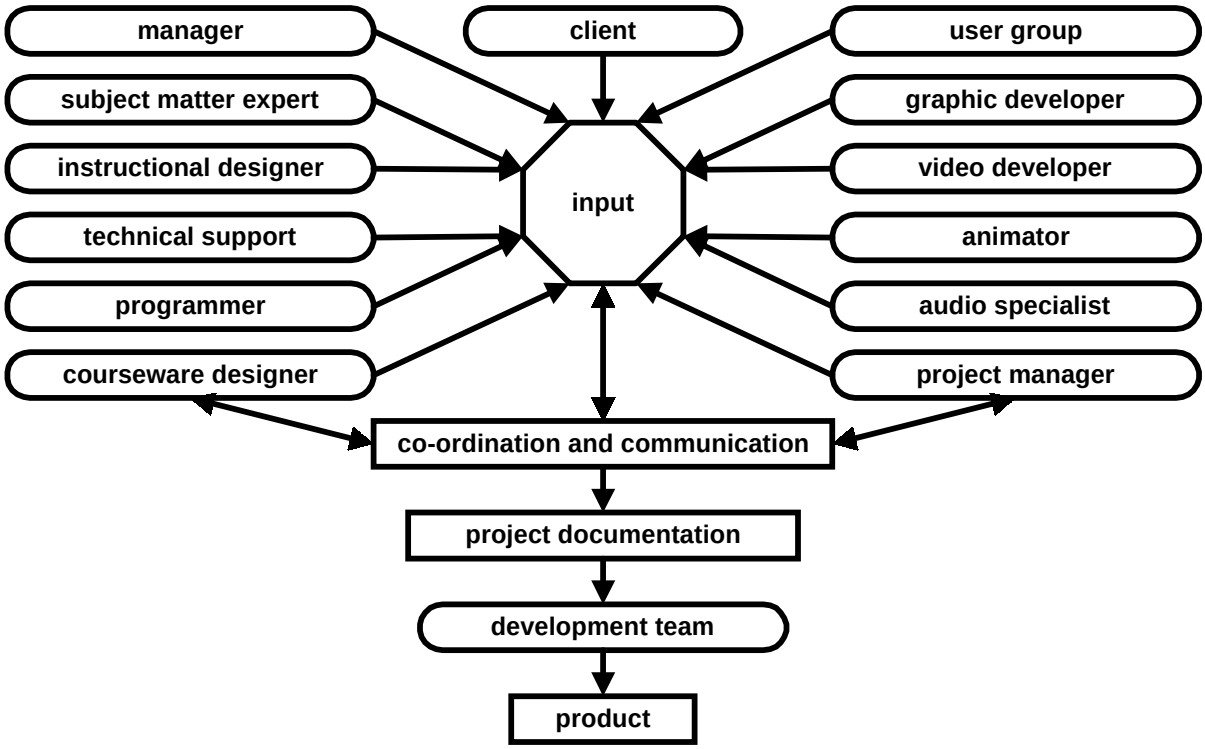

Fig. 3: Structure of a Typical Multimedia Development Team

A clear definition of the roles and responsibilities of each team member helps to ensure that all project tasks are allocated. It also allows contingency plans to be put in place in the event of any member of the team becoming unavailable.

Project Life Cycle Figure 4 shows the four main phases involved in a typical multimedia project. The first three of these (preparation, design and development) are iterative processes - the amount of iteration depending critically upon the values of the time and cost variables for a project. The fourth phase (dissemination) can be repeated but it is usually not of an iterative nature - unless the product is revised or modified in some way between releases.

Analysis of Requirements Multimedia learning products are a relatively new approach to education and training. Many clients are unaware of the advantages and disadvantages of these methods of instruction and can often request an end-product that is unsuitable or impractical. Other projects may be impossible to fulfil within the time or budgetary constraints. These needs are difficult to assess without a comprehensive project analysis. This should determine the specific requirements of the end-product and should include an in-depth study of the users, the intended delivery systems and the scope of the project. It is important that these are fully documented so that developers and clients can refer to them at various stages during the life cycle to ascertain whether the product is meeting the required standards. The results enable designers and developers to identify a suitable interface and appropriate content treatment.

Design Specifications A design specification is a referral document that provides a detailed description of the program requirements. It incorporates essential information such as the font sizes to be used, the colour palettes, file formats and flow charts outlining the proposed content. This document should be available to all members of the development team.

Human-Computer Interface (HCI) The HCI encompasses a number of factors that are fundamental to the development of the product. These include the proposed screen layouts, navigation controls, user interaction, 'look and feel' and the nature of the various metaphors that are embedded within the courseware.

Delivery Medium Multimedia learning products can be delivered to their users in a number of ways for example, CD-ROM publication, the Internet, an intranet or local area network, and various types of turnkey solution. Each type of delivery method has its own constraints and limitations; it is therefore important to address these at the planning stage of the project. Of course, combinations of these different delivery methods may also be used within any given project.

Resource Production Many multimedia learning products are made up of a series of 'pages' containing text, graphics, audio, video and animation. The establishment of appropriate parameters for each of these at the outset of the project avoids inconsistencies within the final product.

Program Integration When all the individual multimedia elements are complete they have to be integrated into the final product by members of the programming team. The ease with which incorporation and integration can be achieved depends critically upon the availability of a comprehensive script and detailed flowcharts to assist the programmers and resource producers. In addition, it is essential to have a good file naming strategy in order to avoid chaos when integrating resource files into the final program.

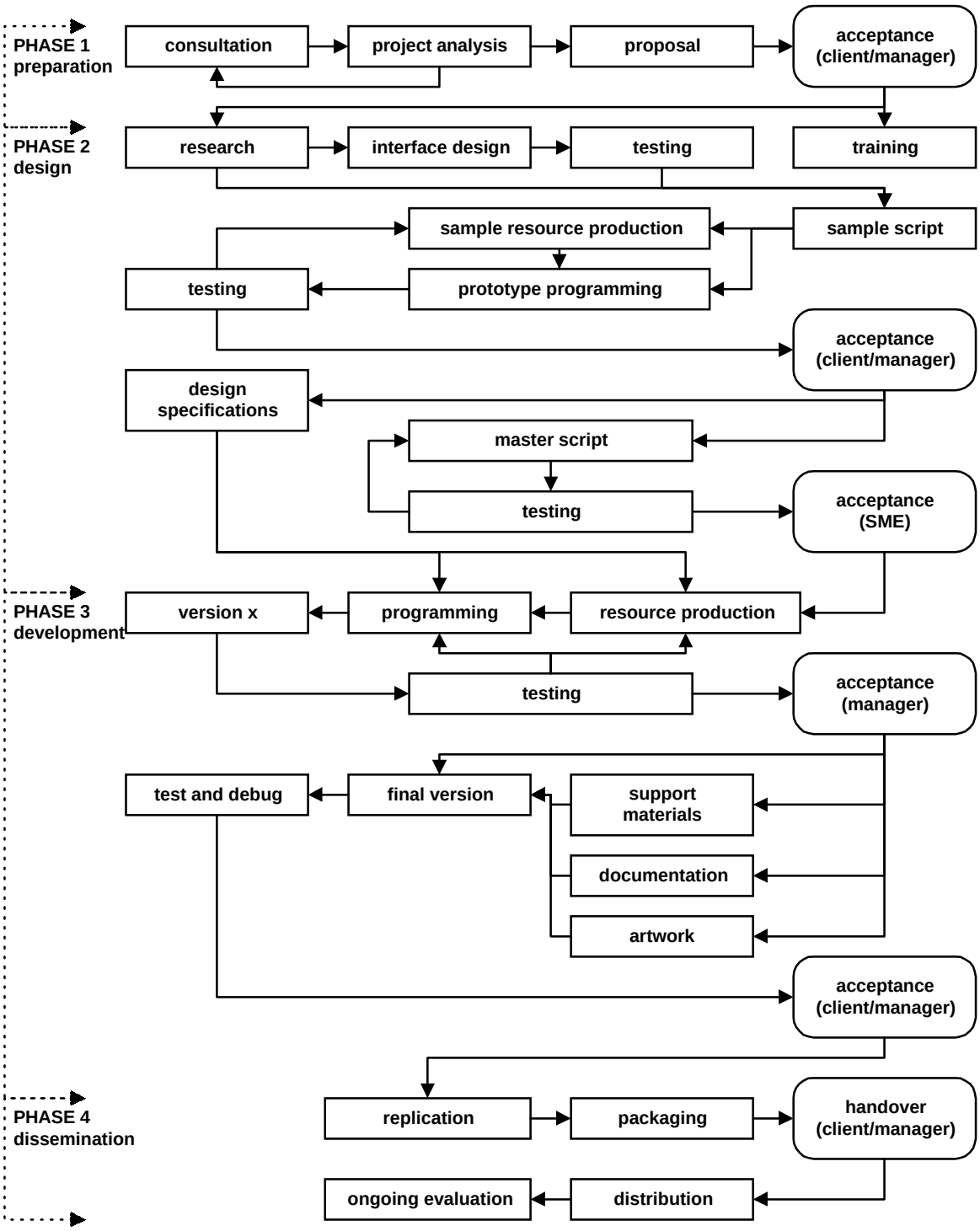

Fig. 4: Development Methodology for Interactive Multimedia Courseware

Quality Control and Evaluation Testing and quality control should ideally occur at all stages of the development process. A thorough testing strategy should be established that allows all errors to be noted and amended in an effective way. Both formative and summative evaluation strategies must be identified during the planning stage. It is important to stress that evaluation and product testing should be regarded as continuous processes - both during the development of a product and after it is complete. Although it is not explicitly indicated in Figure 4, we believe that ongoing testing and evaluation should be employed throughout Phases 1, 2 and 3.

If this is done, then the final product that is generated should fall well within that product's pre-specified 'region of acceptability' - both in terms of its usability and in terms of the skill/knowledge development that takes place as a result of using it. As we discuss later, '*ongoing evaluation*' is an important aspect of the dissemination phase of the methodology.

Documentation Good documentation is essential to ensure that any project life cycle evolves efficiently and within the allotted time. The basic documentation required for most multimedia projects comprises: a project proposal; design specifications; scripts and storyboards; quality control and testing strategies; and progress reports.

Scripts and Storyboards Scripts are paper-based documents that provide a detailed description of the courseware content and how this is to be implemented within the final product. Scripts and storyboards are usually prepared by courseware designers working in conjunction with SMEs. Each page contains details of all on-screen text and audio transcripts. It includes written descriptions of graphics, video and animation along with their file names. These form the basis of audio scripts, graphics listings and video and animation storyboards. It also provides special instructions for the programmer and detailed flow charts.

Dissemination The dissemination phase of product development involves all the processes necessary to package the product and make it available to customers. Of course, even after a learning product has been released for distribution (see Phase 4 in Figure 4) it is important to undertake a programme of '*ongoing evaluation*'. The results of this type of evaluative study can subsequently be used to fine-tune future releases of the learning product and also inform the design and development processes involved in the creation of new courseware products.

DISCUSSION

There are a number of fundamental reasons why models and methodologies are important within the context of the courseware engineering activity involved in the production of an interactive multimedia learning product. Undoubtedly, one of the most convincing arguments for the use of models stems from the fact that they enable us to make accurate predictions about the properties and behaviour of the systems that they describe. Naturally, an important aspect of any interactive multimedia project is the overall cost of producing the final product. It is therefore important to consider what models are available to enable us to predict this - and other important parameters such as development time and how this is partitioned over the four quadrants depicted in Figure 2a. In conventional software projects there are a number of models that can be used to estimate algorithmically the likely cost that a development project will accrue and how long it is likely to take to create the final product [14]. However, as far as we are aware, there are no analogous models available to enable us to predict the overall cost or the duration of an interactive multimedia development project. Development is either undertaken to a 'fixed budget' - which may involve trimming down and cutting back in order to keep the product within cost - or it may go 'over budget' - with the consequences that this will have on the final price of the product for customers. Naturally, the development of appropriate models and/or methodologies to support the various activities involved in creating interactive multimedia learning products will go some way towards alleviating these issues.

In this paper we have described a methodology to facilitate the development of interactive multimedia learning products. This methodology has been extensively used within Singapore Polytechnic - where it has been adopted as a standard design and development tool. The

approach embedded within the methodology has been successfully used to produce a number of interactive multimedia titles for learning applications. Some examples of the products that have been developed include '*A Multimedia Guide to Hydroponics*' (1995), '*Nutrition*' (1997), '*Ornamental Fish - A Multimedia Guide to Aquaculture in Singapore*' (1997) and '*Wafer Fabrication Safety*' (1999).

The methodology that was described in the previous section has been derived primarily from our experiences with conventional multimedia learning products involving the creation of interactive CDs - such as those listed above. It is therefore necessary to discuss whether or not the methodology (as presented in Figure 4) could be applied to the development of other types of interactive learning product - particularly those involving the use of an intranet or the World Wide Web. Our experiences to date suggest that the basic methodology (with some minor amendments) will easily handle this latter type of product. Essentially, Phases 1 through 3 could be used as they stand. So far, we have found that the only area where minor amendments may be needed is in Phase 4. In this phase, the 'replication' and 'packaging' steps are usually less important for an intranet/Internet project than they would be for a CD project - unless, of course, some form of paper-based ancillary of CD-based materials are used to support the under-lying web-based learning resources. We are therefore optimistic that the proposed methodology (with or without minor modifications) can be used to cater for a wide range of interactive multimedia learning product development projects

CONCLUSION

Developing interactive multimedia courseware products is a complex and costly process that requires a wide range of powerful models and a workable, practical design and development methodology. In this context, a structured design approach has a number of advantages. For example, templates and structural documents provide essential information in easy-to-read formats that can be tailored to meet the requirements of individual team members. In addition, the production process can be made simpler and more efficient - as fewer program and resource changes are required at the development stage. Furthermore, quality control is made more effective by providing testers and evaluators with a program 'blueprint'.

A learning product development methodology, such as the one that has been described in this paper, should enable multimedia courseware products to be developed in a more predictable way - both in terms of learning outcomes and from the perspective of resource utilisation (time and money). Implicit in the various project phases that we have described in our methodology are a number of local and global interactions - both between the underlying models and the system variables that are involved in managing the preparation, design, development and dissemination processes. It is through a greater understanding of these interactions that we will be able to gain further, much needed, insight into the various factors that are responsible for causing multimedia learning projects to fail.

REFERENCES

[1] Banerji, A, (1995). *Designing Electronic Performance Support Systems*, PhD Thesis, University of Teesside, Middlesbrough, Cleveland, UK.

[2] Barker, P.G., (1994). Designing Interactive Learning, 1-30 in *Design and Production of Multimedia and Simulation-Based Learning Material*, edited by T. de Jong and L. Sarti, Kluwer Academic Publishers, Dordrecht, The Netherlands.

[3] Barker, P, (1996). Electronic Libraries of the Future, *International Encyclopaedia of Library and Information Science*, Volume 59, Supplement 22, 119-153, edited by A. Kent and C.M. Hall, Marcel Dekker, New York, NY, USA.

[4] Barker, P.G., (1999). Using Intranets to Support Teaching and Learning, *Innovations in Education and Training International*, 36(1), 3-10.

[5] Barker, P.G., (2000). Developing Teaching Webs: Advantages, Problems and Pitfalls, paper presented at EDMEDIA 2000, in *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications*, June 2000, Montreal, Canada. This paper is available in electronic form at *http://www.philip-barker.demon.co.uk/edmd2000.htm*.

[6] Barker, P.G., (2001). Creating and Supporting Online Learning Communities, paper to be presented at EDMEDIA 2001, *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, University of Tampere, Finland. . This paper is available in electronic form at *http://www.philip-barker.demon.co.uk/edmd2001.doc*.

[7] Bruce, P. and Pederson, S. M., (1982). *The Software Development Project, Planning and Management*, John Wiley & Sons, New York, NY, USA.

[8] Checkland, P., (1981), *Systems Thinking, Systems Practice*, Wiley, Chichester, UK.

[9] England, E. and Finney, A., (1999). *Managing Multimedia - Project Management for Interactive Media*, 2nd Edition, Addison-Wesley, Harlow, Essex, UK.

[10] Giller, S, (1993). *Design Guidelines for Electronic Book Production*, MPhil Thesis, University of Teesside, Middlesbrough, Cleveland, UK.

[11] Kaur, N., (1999). *Patient Education in a Networked Multimedia Environment*, MSc Dissertation, School of Computing and Mathematics, University of Teesside, Middlesbrough, Cleveland, UK.

[12] Maddison, R.N., (1983). *Information Systems Methodologies*, Wiley Hayden, Chichester, UK.

[13] Misanchuk, E.R., Schwier, R.A. and Boling, E., (2000). Discourse on Designing an Illustrated Hypermedia Book, *Journal of Educational Multimedia and Hypermedia*, 9(3), 223-251.

[14] Sommerville, I., (1989). *Software Engineering*, Third Edition, Addison-Wesley, Wokingham, England, UK.

[15] Tan Meng Yoke, (1998). *Methodologies for Multimedia*, MSc Thesis, National University of Singapore, Singapore

Professor Philip Barker
Human-Computer Interaction Laboratory
School of Computing and Mathematics
University of Teesside
Middlesbrough, TS1 3BA
United Kingdom

Tel: +44-(0)1642-342660
Fax: +44-(0)1642-342067
Email: Philip.Barker@tees.ac.uk
Personal Web: http://www.philip-barker.co.uk
Research Web: http://www.isrg.co.uk

Susan Giller
PO Box 16570
Hornby
Christchurch
New Zealand

Tel: +64 (03) 334 1095
Fax: +64 (03) 334 1095
Email: sue.giller@pobox.com
Web: http://www.sue.giller.com