# New Wine in Old Bottles: Multimedia Design Methodology

Robert (Bob) Godfrey
*Department of Applied Computing and Mathematics*
*University of Tasmania*
*B.Godfrey@appcomp.utas.edu.au*

**Abstract**

Multimedia technology opens up a whole new world for the developer of computer-assisted learning (CAL) material. From the outside, the technical challenges of this new area seem quite different from the conventional CAL development, but on closer inspection the techniques needed to design and develop multimedia applications are the same as those used to develop more conventional applications - problem definition, scope definition, top-down development, prototyping, structured walkthroughs, modularity and data dictionary / repository among others.

This paper outlines a hypothetical multimedia interactive role-playing CAL system (used to explore Just-in-Time inventory management). It then proceeds to describe the steps of a methodology for the design and development of the system, comparing the various aspects of the methodology (genre, character, interface, location, plotting, scripting and testing) to more conventional systems development processes.

The paper concludes that while the technical side of multimedia is new and a knowledge of new devices and interfaces is required, and new skills of animation, film and audio need to be integrated, this new wine is still contained in old bottles, for the tried and tested methods of conventional system development are readily adapted to these new media. The biblical exhortation not to put new wine into old bottles does not apply to multimedia design methodologies.

**Keywords**

*ASCILITE'95, interactive multimedia, multimedia design, system design, multimedia, methodology*

## 1. Introduction

Multimedia is seen as having great potential as a pedagogical tool (Daily, 1994), and some would argue that it is an almost essential component of technology-based training (Sims, 1994) but early reports suggest that it is a difficult and expensive technology to use well (Ellis and Sims, 1994). While some might agree with Clark's (1994) assertion that there is demonstrable proof that media atrributes influence learning, and that most claims are the product of enthusiasm, perhaps a greater number would side with Reiser (1994) and his view that while new media are not essential delivery tools, they can ease the path of the learner. So while we could manage without multimedia, its existence opens up new learning possibilities that we would be wise not to ignore.

There are suggestions that multimedia is fundamentally different from other computer-assisted learning techniques, and that accordingly design and production must be approached differently. Much of this concentration on the unique character of multimedia is centred around two aspects; first the technological richness of the media, and second the broadness and diversity of the audience or

users (Trueblood et. al., 1995) While these two aspects of multimedia design are significant, there is a danger that people will see them as a reason to reject traditional computer system design methodologies and to try to build multimedia methodologies from the ground up.

We might draw a parallel here with end-user computing systems, when end-users are experiencing all the development problems that computing professionals have encountered and solved over the years - structured coding, effective testing, documentation and the like. Computer professionals learnt the hard way what techniques worked and which ones did not, and it is a waste for end-users to retread the path and to learn the hard way. Similarly in multimedia systems there is much to be gained for adopting as a basis the traditional computer systems development techniques, and then to recognise the areas that need modification in the light of experience and to refine the methodology accordingly. Just because the technology is different does not mean that everything else must be different. If traditional computer system development methodologies are readily adapted to multimedia, then in the huge pool of computer professionals there exists substantial resources for accelerating the pace of multimedia development.

## 2. The waterfall model

Figure 1 shows on the left the traditional waterfall model of the systems development life cycle (SDLC), and on the right a similar diagram of the stages in the production of an interactive multimedia system - a multimedia development life cycle (MDLC).

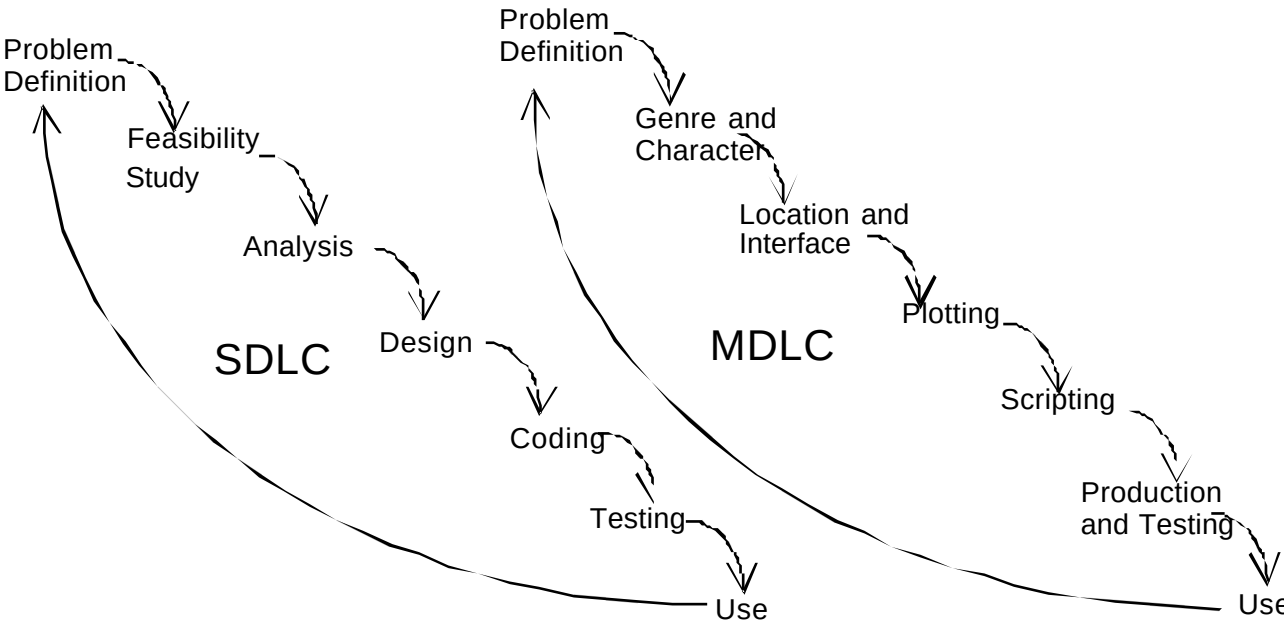


Figure 1. The SDLC and the Interactive Multimedia Development Life Cycle

The SDLC clarifies the problem definition, then establishes the feasibility of the project. This is followed by analysing the requirements in detail to determine the scope and basic character of the system. Design turns this outline into formal specifications, which are then coded and tested to produce the final product.

The interactive multimedia system defines the target audience and their requirements, then chooses an appropriate genre. This is followed by characterisation and then location setting. Plotting via mapping and flowcharting follows, then scripting precedes construction.

The two methodologies use very different terms, but closer inspection exposes great similarity. To illustrate this similarity let us take as an example a proposal to develop an interactive multimedia system that will be used by students learn about the intricacies of demand and supply in a just-in-time inventory system.

## 3. Problem definition

Problem definition has to be the first stage of any design cycle. It shouldn't need saying but regrettably has to be said that successful systems design and implementation is only possible when developers have a clear idea of the problem to be solved. There are four basic issues - identifying the client / sponsors / target audience, eliciting their needs / wants, identifying the scope of the project, and understanding the resource limits. In reactive projects much of this information is given by the client, but there is still a need to understand power relationships and decision making structures so as to avoid satisfying only the less important, but not the most influential, members of the client organisation. In proactive projects where it is the developers that initiate the process, there is greater freedom to determine the nature of the system, but poor choices at this stage will lead to trouble in subsequent marketing.

As with any system, greater success in problem definition is achieved when one understands the target users, the technology, and the problem domain. It is essential then, for multimedia development as for any system, to recognise your strengths and weaknesses, and to seek help from people with greater knowledge in the areas you have least experience in. So it is difficult to switch directly from a lifetime of developing accounting applications to multimedia educational systems, but no different from switching to real-time control of a chemical plant, a security alarm data communication system or the like. In any system there is a need to identify the development skills required, and to build a team with those skills. It is not necessary, nor indeed it is often possible, to find all these skills in one person.

## 4. Genre

Choosing the appropriate genre is basically a feasibility decision. In a traditional feasibility study, the analyst uses a mental model of a proposed system, aided by rough sketches and calculations, and tries to imagine the idea working. They examine feasibility from three perspectives - technical, economic and organisational feasibility. In assessing technical feasibility, they ask the question as to whether they think the system can be built. In economic feasibility they estimate the likely cost of production and in organisational feasibility they ask whether the idea would work given the intended user organisations and people.

A genre 'comprises a class of communicative events, the members of which share some common set of communicative purposes. These purposes are recognised by expert members of the parent discourse community and thereby constitute the rational for the genre.... Exemplars of a genre exhibit various patterns of similarity in terms of structure, style, content and intended audience. If all high probability expectations are realised, the exemplar will be recognised by the parent discourse community' (Swales,1990, p. 58). Thus when an appropriate genre is chosen for a particular communication, the user is encouraged to and is able to make implicit assumptions about the communication, and these need not be detailed explicitly. If a detective / mystery genre is chosen, the user will expect to ask questions, interpret clues, and eventually "find the murderer" or its equivalent. In a romance setting, there is an expectation of an initial rapport, followed by conflict or misunderstanding, and concluding with a rapprochement. Similarly, they are clear expectations of other genres such as adventure, science-fiction, western and comedy. The romance genre might suit

learning about relationships, the western or the detective, notions of right / wrong conduct, and the adventure or science-fiction, the exploration for new knowledge as say in a campus orientation tour.

For example, if we want to produce an interactive multimedia system for fact finding techniques during computer systems analysis, a detective genre is a fairly obvious choice. The classic detective in the belted raincoat or their counterpoint in Miss Marple could interview a variety of people, take notes, seek clarification of ambiguities, and gradually build up a picture of the facts. We might try out other genres, but might finally conclude that the detective genre makes our task easier - we have decided that the approach is technically feasible.

The choice of an unusual genre may increase the level of interest by combining fantasy and realistic elements. Suppose for example we have decided to create an interactive multimedia game to teach the concepts of Just-in-Time Inventory Management. We might use the traditional Christmas song “on the first day of Christmas my true love gave to me a partridge in a pear tree.....” and choose a romantic setting with the hero requiring to get supplies of partridges, turtle doves, French hens, geese and swans in ever increasing number to satisfy the object of their affections, with a second wholesaler being a rival for those affections. As in the song, the complexity of the task increases at each of the twelve stages, and one could work in industrial disputes, differential pricing structures, lead times, minimum order quantities, and inventory processing into this alien setting.

## 5. Characterisation

Characterisation is an extension of genre choice where we choose the primary character, secondary (support, foil, catalyst) characters and work with archetypes and stereotypes within our genre. By defining characters with physical attributes, specific knowledge and role, language style and vocabulary, attitude and the like, we are attempting to cover the bases of our learning objectives while still remaining true to our genre. So, returning to our example, taken from our original problem definition, we might produce a check list of character traits (helpful, competitive, obstructive, friendly, aloof, rigid, deceitful, vague) we would like our user to experience and we would attempt to attach these traits to our characters. We have in effect started to build a data dictionary / repository as used in Database Management or CASE tools.

## 6. Location

When working with locations we use a top-down approach. Our macro view or big picture links micro views (the little pictures) into a network and is concerned with the usable content of each location and the possible cause and effect dynamics that are available. The creation of locations and sub-locations is a manifestation of top-down design, and as in program design, for reasons of economy we will often reuse locations or sub-locations.

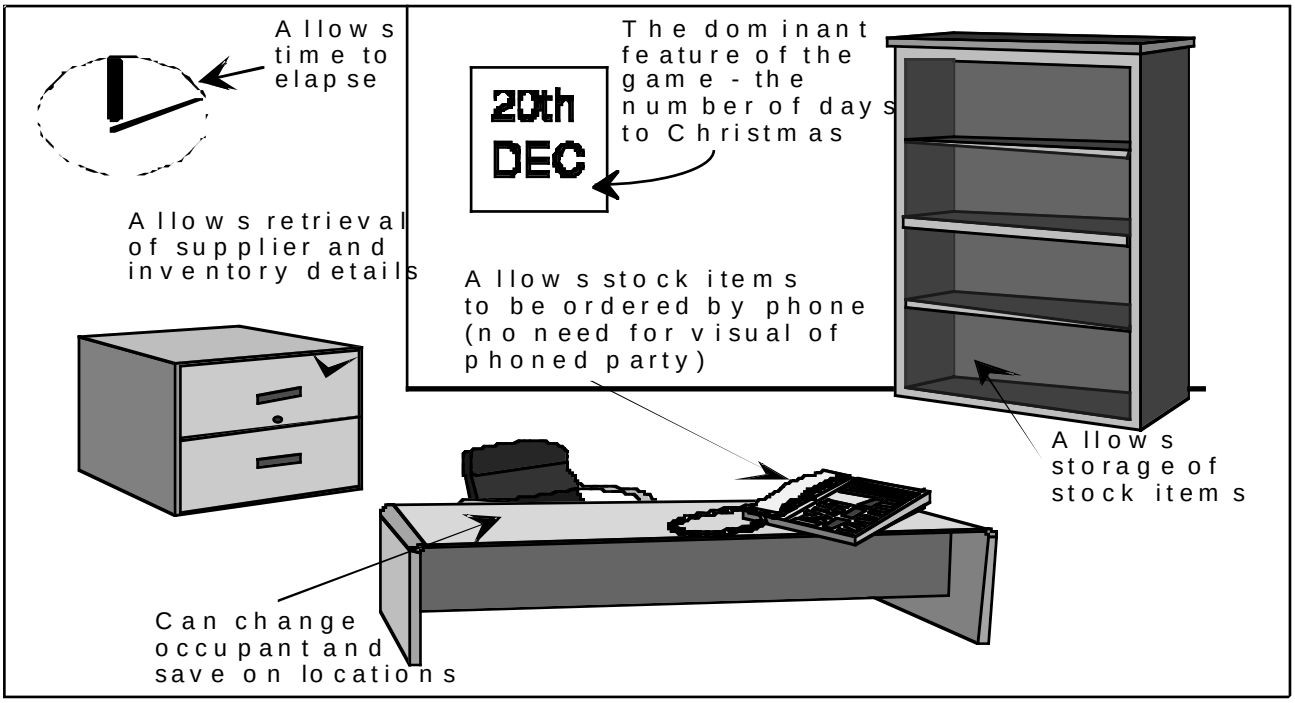


Figure 2. An office location.

In this phase of the design we are concerned with sketching each location and placing movable objects and fixed property within the location, and adding these things into our dictionary or repository (figure 2). The multimedia location sketches serve the same purpose as the rich pictures of Soft Systems Methodology (Lewis, 1992), with their emphasis, not on accurate depiction, but on describing the essence of some activity, and where humour and characterisations can be employed to focus attention and to stimulate discussion.

## 7. Interface

Part and parcel of location design is the specification of the interaction interface and its commands - talk, examine, use, take, give and so on. Over the years, a variety of user interface software tools have been produced (Myers, 1995) for traditional computer systems. Tools like pop-up windows and icons, and widgets like menus, buttons, scroll bars, text type-in fields, alpha sliders, range sliders and the like can be readily adapted as part of the interactive multimedia user interface. These techniques need to be integrated into a learning framework so that the user can express their thoughts, feelings and desires, and is not just a passive object to be manipulated (Teck, 1994).

## 8. Plotting

Having decided upon the locations, it is now time to connect them up with actions and events to indicate the various story paths, rewards and obstacles of the interactivity. Traditional dataflow diagrams and flowcharts are readily adapted for this stage of multimedia design. In linking locations and sub-locations, the lessons of programming modularity need to be borne in mind, particularly the emphasis of data coupling.

Another systems method is of use here - the walkthrough (Yourdan 1985). We can use this technique to assign location sketches to role-playing individuals, and then to walkthrough the plotting to check that there are no unwanted loops or dead-ends, and no inconsistencies in the plot.

## 9. Scripting

Scripting is the process of defining all the dialogue, actions and reactions, location by location, scene by scene, for the whole interaction. Scripting is like coding a program, only in a new language and perhaps with different layout rules. In order for dialogue to flow conversation trees, similar to decision trees, are a useful visualisation (figure 3).

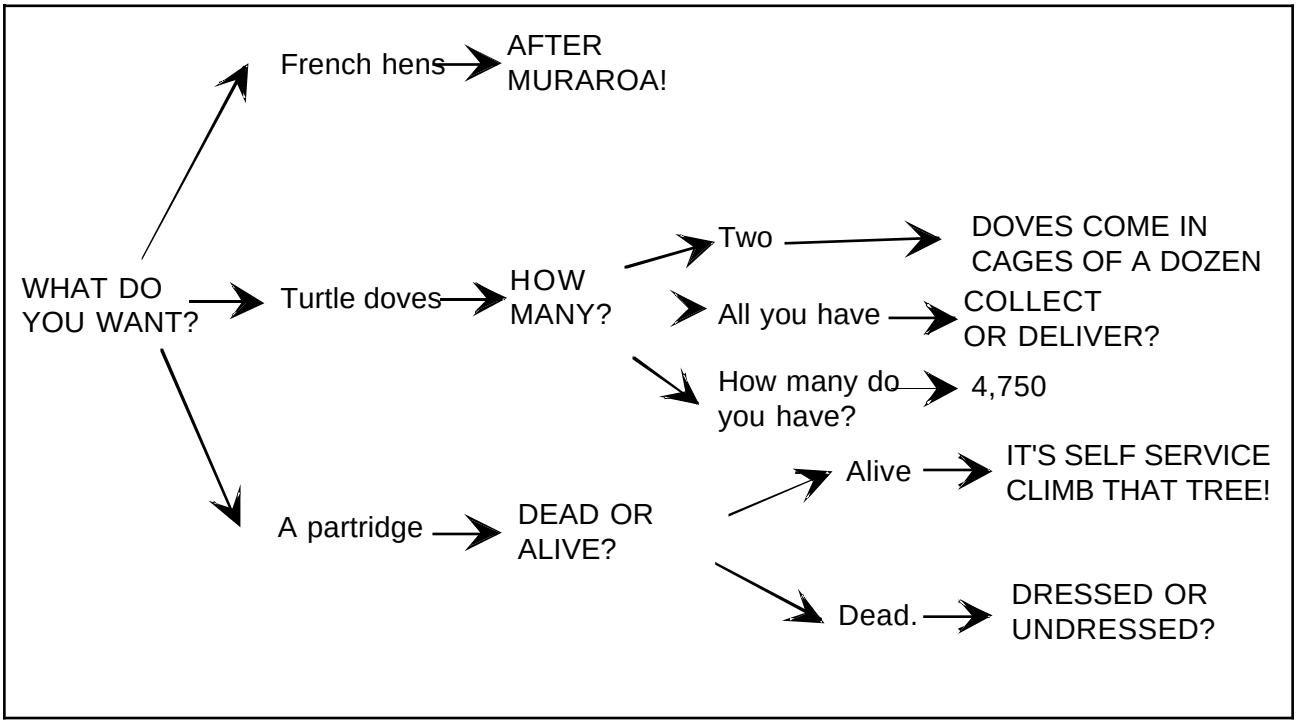


Figure 3. A conversation tree.

Scriptwriting is like program coding in one other way. It is a process that mixes art and science. No amount of perspiration can make up for a lack of inspiration. If you cannot write lively, humorous dialogue then you will have to use professional writers to engage your target audience. Trying to do it yourself if you don't have the gift is to produce an amateurish product (Wharton, 1994). Technical writers have frequently been used by computer professionals to turn draft programmer documentation into user (friendly) manuals. Computing people would be wise to continue this tradition and use professional writers to produce polished multimedia dialogue.

## 10. Production

If the design has been carried out well, production is simply a process of letting each skill group - film makers, graphic artists, animators and programmers, go about their tasks in their own way. Of course, as in any computer project, poor design will be magnified during production, and major problems will necessitate returning to previous design stages to overcome them.

## 11. Testing

Computing people are experienced in the process of prototyping, unit testing and system integration, and these skills come to the fore in multimedia development. Producing a CD-ROM is an irreversible process that requires a high level of testing before the product is frozen. It is useful to prototype small

sections of the interactivity using PCs, and to integrate sections progressively before the whole process is passed over to media production.

## 12. Overall project management

A critical feature in multimedia development is the sizing of the finished product to fit say a CD-ROM - so much video, so much audio, so much program, just as PC software must control and balance ROM, RAM and disk storage and meet desired response times.

Interactive multimedia projects can rarely be carried out by a single individual. Teams must be built and staffed with the large range of skills required - educators, animators, film makers, scriptwriters, programmers and so on (Pearce, Riddle and Nott, 1994). The range is perhaps greater than in many computer systems with their analysts and programmers, but in other computer systems we need database administrators, data communications experts, process control experts, knowledge engineers and the like. As with many computer system developments, we need more than the tool and some perspiration, we need inspiration and specialisation. So CAL software requires educators and instructional designers, and desktop publishing requires editors.

Any heterogenous team will present problems of differences in perspective, failure to understand / communicate, desire to be the central focus and differences in motivation. Multimedia projects present a particular problem with the coming together of two highly creative groups - computing people and media people, both of whom like to retain artistic / creative control of their activities and do it their way. Teams will not succeed unless the creative energies of both groups are harnessed together rather than competing for the centre of the stage. For this to happen, there must be a sharing of ideas and a mutual respect for each group's special skills and experience. So we are not suggesting that IT people do it all, for then presentation might dominate over content (Smoliar, 1994), but to recognise that in developing their separate methods and traditions, computing people and creative arts people have been treading down parallel paths, and there is now the need to merge the two methodologies before moving ahead.

## 13. Summary

The modern systems analyst must be capable of dealing with an ever growing set of methodologies. From early beginnings where the flowchart was perhaps the only tool used, today there is a great variety of tools and techniques - SSADM, Prototyping, Soft Systems, CASE and the like - from which an appropriate choice must be made at each step in the development process. Multimedia simply adds a few more options and variations on tried and tested methods. To ignore this rich collection of existing methods when developing multimedia applications is to repeat in great part the lessons of history. Recognition and exploitation of the similarity will bring two benefits. First, it will encourage traditional computer systems designers to engage with multimedia, and second, it will provide a resource pool that educators, artists and communicators can tap into for assistance with multimedia development.

## 14. References

Clark, R.E. (1994). Media will never influence learning. *Educational Technology Research and Development*, Vol. 42, No. 2, pp. 21-29.

Daily, B. (1994). Multimedia and its impact on training engineers. *Internatianal Journal of Human-Computer Interaction*, Vol. 6, No. 2, pp. 191-204.

Ellis, A. and Sims, R. (1994). Multimedia authoring: Developing effective training programs. *APITITE'94 Proceedings*, pp. 735-740.

Lewis, P.J. (1992). Rich picture building in the soft systems methodology. *European Journal of Information Systems,* Vol. 1, No. 5, pp 351-360.

Myers, B.A. (1995). User interface software tools*. ACM Transactions on Computer-Human Interaction,* Vol. 2, No. 1, pp. 64-103.

Pearce, J.M., Riddle, M.D. and Nott, M.W. (1994). From concept to product: A team approach to multimedia development. *APITITE'94 Proceedings*, pp. 883-888.

Reiser, R.A. (1994). Clark's invitation to the dance: An instructional designer's response. *Educational Technology Research and Development,* Vol. 42, No. 2, pp. 21-29.

Sims, R. (1994). Seven levels of interactivity: Implications for the development of multimedia interactivity and training, *APITITE'94 Proceedings*, pp. 589-594.

Smoliar, S.W. (1994). On the promises of multimedia authoring, *Informtaion and Software Technology*, Vol. 36, No. 4, pp. 243 - 245.

Swales, J.M. (1990). *Genre analysis: English in academic and research settings*, Cambridge University Press, Cambridge, pp. 58.

Teck, T.S. (1994). A framework for button theory implementation: A multimedia learning system. *APITITE'94 Proceedings*, pp. 551-556.

Trueblood, R.P., Lai, V.S., Mahapatra, R.K. and Floyd, S.A. (1995). A multimedia development process, paper to be presented to the *American Information Systems Conference*, August 25-27 1995, Pittsburg, PA.

Wharton, S. (1994). Integrated multimedia ... Is it in the script? or CD-ROMS ... Why, it's the done thing!. *APITITE'94 Proceedings*, pp. 391-396.

Yourdan, E. (1985). *Structured walkthroughs*, Yourdan Press, Englewood Cliffs, New Jersey.

## 14. Acknowledgement

A debt of gratitude, for introducing me to the potential of multimedia in their excellent workshop "Scripting for Multimedia", is owed to Kathy Mueller, a consultant with Beam Software, and to Mark Morrison, the creative force behind Beam Software's film noir detective interactive "The Dame is Loaded" (to be published in 1996 by Philips International, London and Los Angeles).