EARLY EXPERIENCES IN MEASURING MULTIMEDIA SYSTEMS DEVELOPMENT EFFORT

Tim Fletcher, Stephen G. MacDonell and William B.L. Wong
Computer and Information Science
University of Otago, New Zealand
Tel: +64 3 479 8142   Fax: +64 3 479 8311
email: tfletcher@sol.otago.ac.nz, {stevemac,wwong}@commerce.otago.ac.nz
Keywords: Multimedia, management, metrics

ABSTRACT

The development of multimedia information systems must be managed and controlled just as it is for other generic system types. This paper proposes an approach for assessing multimedia component and system characteristics with a view to ultimately using these features to estimate the associated development effort. Given the different nature of multimedia systems, existing metrics do not appear to be entirely useful in this domain; however, some general principles can still be applied in analysis. Some basic assertions concerning the influential characteristics of multimedia systems are made and a small preliminary set of data is evaluated.

INTRODUCTION

With its inherent use of original and pre-sourced media components as fundamental building blocks and a focus on screen-oriented authoring, multimedia systems development does not lend itself easily to the 'traditional' software processes commonly used in other domains. Multimedia information systems (MMIS) development has therefore evolved as its own sub-discipline over the last ten years to a point where a range of high productivity specialist tools and more appropriate development methodologies have been formulated. This is a reflection of an increasing maturity in MMIS development. Another outcome of this greater maturity is a realisation that the development process must be managed effectively, so that it can be measured, controlled and improved (Gao and Lo, 1996) - for *all* systems, we are concerned with increasing quality and productivity whilst minimising cost.

Despite tool and methodology advances, MMIS development continues to demand significant effort. England and Finney (1996) believe this is because multimedia development projects have so many influencing variables (more than for other system types) that must be considered, while Marshall et al. (1994) suggest that the development of system content is the major constraint to the widespread use of commercial multimedia, despite faster cheaper hardware and more powerful authoring environments. In terms of the current project, we are concerned with the implications of the 'non-standard' MMIS development methodologies and the particular characteristics of MMIS in terms of software development effort measurement.

The next two sections of the paper consider the specific characteristics of MMIS that mean that commonly used measurement methods in the business software domain are less applicable. This is followed by a discussion of the current empirical study. Preliminary data analysis is then presented and the implications of the findings are discussed. The limitations of the study are described and the paper is concluded with a short summary of ongoing work.

SPECIAL CHARACTERISTICS OF MULTIMEDIA SYSTEMS

Multimedia information systems development is sufficiently different from that of other development paradigms to mean that models, tools and methodologies from these domains are not entirely suitable (Marshall et al., 1994; Gao and Lo, 1996). Four important differences have particular impact: the use of specialised high productivity authoring tools, the preparation of the media content, the cross-functional composition of many development teams, and the methodology phases applicable to MMIS development.

Authoring tools are used in a similar manner to the way in which fourth generation languages (4GLs) are used in business-oriented systems development. In general, authoring tools are utilised to integrate and build the *system* (rather than the content) at a very high level of abstraction. Of concern is what the system should do, rather than how it should do it (hence the analogy with 4GLs). These environments offer very high productivity, and some even allow systems to be built without coding or scripting. Research into development effort associated with the use of authoring tools is not widespread, however. Moreover, this mirrors a similar dearth of knowledge in regard to 4GL-based development in the business domain. Thus the degree of impact of the use of such tools on development effort is unclear.

The authoring process generally occurs after the development of the specific media components. Construction of the media is arguably the most difficult task in the overall MMIS development process - it certainly appears to be the most time consuming. Marshall et al. (1994) and others (Merrill et al., 1991) indicate that elementary computer-based training (CBT) software requires 100 hours of effort per delivery hour, a figure that can rise to 800 hours or more per delivery hour if multimedia elements are added (Beautement, 1991). Unfortunately, these figures are generally anecdotal - it is to be hoped that continued empirical work may provide more objectively derived indications of effort required.

The media preparation component has further implications for effort from a personnel perspective. Whereas 'traditional' development of information systems is generally dependent on a team of software specialists (albeit with particular strengths), the development of MMIS is often undertaken by cross-functional development teams, with one group responsible for the software and the other responsible for the content and design. This can bring added complications to the development process, given that communication even among software specialists is notoriously poor, particularly as the size of the team increases. The impact of a 'non-software' design team on development effort may be significant.

Finally, the overall development methodologies and the activities that occur within them are by necessity quite unique to the development of multimedia systems. As a consequence, many of the components or models on which measurement has been based in the past are simply not available in the multimedia domain. For example, data-oriented specification methods and models used in commercial transaction-based systems development are not easily mapped to multimedia projects. Similarly, the algorithm-centred models used in scientific systems development are also inappropriate for multimedia systems. Multimedia information systems development processes are probably most similar to prototyping methodologies in that significant emphasis (and hence effort) is required in the iterative development of a suitable and appropriate interface with adequate system functionality, although Marshall et al. (1994) have used an adapted waterfall model to represent multimedia courseware development (Figure 1). It should again be acknowledged, however, that the inherently necessary media preparation stage is a key distinguishing factor. The IBM Multimedia Consulting Methodology (Gruskin, 1994) shown in Figure 2 provides a useful

illustration of this consideration, with media preparation depicted as a separate and important activity that is carried out concurrently with authoring and integration.

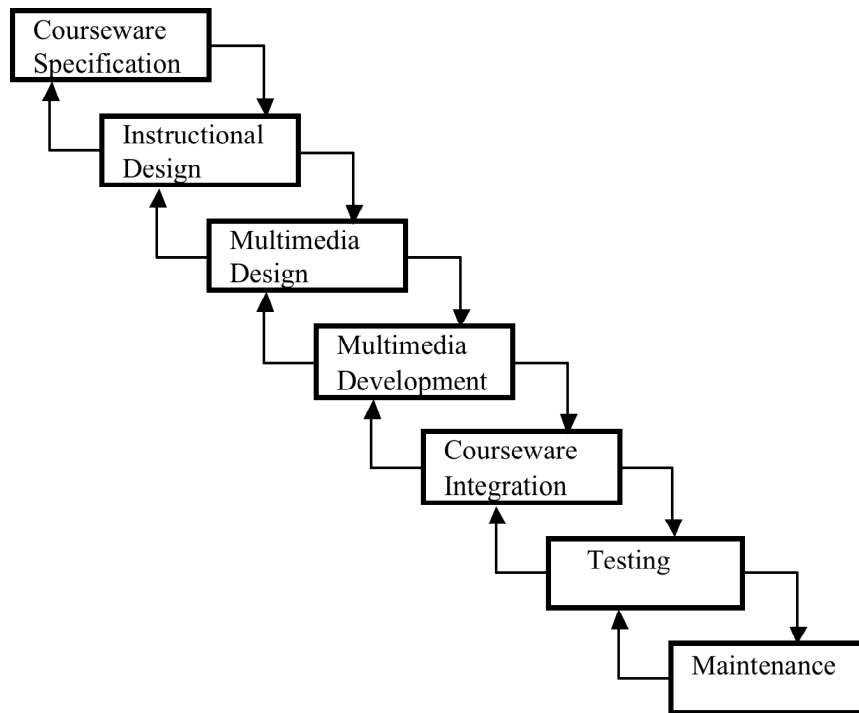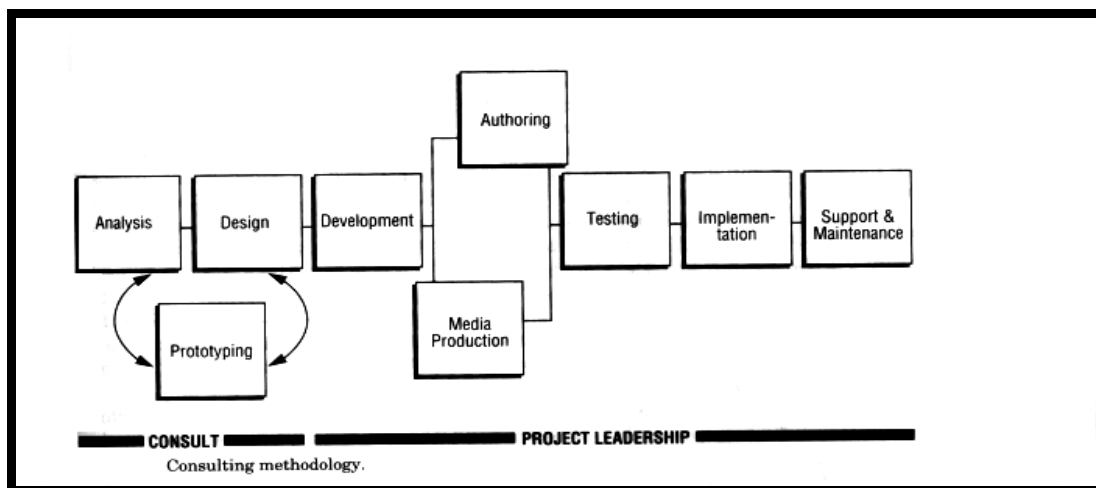FIGURE 1: THE MULTIMEDIA WATERFALL MODEL (MARSHALL ET AL., 1994)

FIGURE 2: THE IBM MULTIMEDIA CONSULTING METHODOLOGY (GRUSKIN, 1994)

Consulting methodology.

MEASURING DEVELOPMENT EFFORT

Almost all current effort assessment and early prediction methods assume the existence of data-centred products (e.g. data flow diagrams, data models, screen and report layouts) and/or a 'traditional' development process (incorporating analysis, design, coding and so on). As previously discussed, multimedia systems development incorporates little of these aspects. A more appropriate assessment and estimation method is therefore suggested here.

Existing Measurement Methods

Development effort estimation is primarily undertaken using either a size-based measure or a function-based measure. The most popular size-based methods use estimated product lines of code as input to a derivative of Boehm's Constructive Cost Model (COCOMO) (Boehm, 1981) with adjustments for various 'cost drivers'. Marshall et al. (1994) have adopted an adjusted version of this approach in their analysis and prediction of multimedia courseware development effort, incorporating consideration of course delivery time and a large number of drivers (influential factors) under four classes of course difficulty, interactivity, development environment and subject expertise. Although their general approach seems sound, the work considered here has no courseware emphasis, so the use of this factor as a part of the model is inappropriate for our purposes. Furthermore, the inclusion of more than twenty drivers, some of which must be assessed subjectively, is an aspect that should be avoided if possible.

Determining effort using function-based measures may be useful for multimedia systems. Albrecht's function point analysis (FPA) method (Albrecht, 1979) considers the contribution of system inputs, outputs, enquiries and files to system scope and complexity, with final adjustment based on the characteristics of the development and operating environments. As defined, however, the approach is not entirely appropriate for MMIS development, for several reasons (Gao and Lo, 1996): MMIS make use of large databases but these are not actually maintained by the system; MMIS development tends to adopt extensive component reuse; and the output forms for MMIS are far more complex than for standard business systems. Gao and Lo (1996) have therefore produced an adjusted FPA assessment method that has attempted to take account of these differences, with some success. In their model, however, output forms (text, sound, animation) are weighted equally whereas it may be that some forms of output are significantly more influential than others in regard to their contribution to effort. Moreover, the subjectivity of FPA in general has been widely questioned (Jeffery and Low, 1990), so an alternative approach may be desirable.

The Proposed Measurement Approach

When compared to existing assessment/estimation methods, the proposed method considers software products more relevant to multimedia systems (e.g. animation sequences as opposed to data entities) and examines their contribution to systems development effort. Measurement of development effort is itself made more relevant to multimedia systems, in that it is suggested that effort data be recorded alongside tasks such as audio/visual editing, digitising, video recording and sound capture. (It should be noted that the empirical analysis described below does not fully evaluate the proposed method, for reasons discussed in the Lessons Learned subsection. In the interests of research, however, the approach is more fully described here.) The proposed approach is based on the assertion that MMIS development effort is a function of (i) building the system content and (ii) authoring the system. Each of these tasks is evaluated in terms of the components manipulated and the activities carried out:

> (i) building the system content - for each media component created, the following data items were to be recorded: filename, media type (graphic, audio, video, animation, photograph, scan), original or pre-existing, creation effort (for original media), digitising effort (for scans, video and audio), editing effort, and component duration (for temporal media i.e. animation, sound and video). The assumption underlying this collection is that each media form may have a different impact on development effort.

(ii) authoring the system - for screen authoring the screen name and authoring effort were to be collected for each screen. An inspection of each project was also to be conducted to ascertain the 'complexity' of each piece of media and each screen. The data to be collected for each screen were: the number of objects on the screen (including sounds), the number of links between that screen and other screens, the number of events on a screen and the average number of actions per event. Procedures that respond to a mouse being clicked, or any other scripted actions, are considered as events. The associated task is normally a generic activity to be performed; typically most link buttons contain only two actions for the click event: play 'click' sound and go to another screen. The media complexity was to be based around graphics data: the number of objects on the component, whether it had been reused elsewhere in the project, and the form it took (button, toolbar, screen, background, component i.e. part of the foreground or a source in an animation sequence). This approach is based on the assumption that a screen that incorporates a greater number of objects and events will take proportionally greater effort to develop.

SMALL-SCALE EMPIRICAL INVESTIGATION

In order to test the validity of the approach and to determine which factors were the main contributors to development effort, data were collected from several senior student projects.

Data Collection

Preliminary data was gathered from five 4th-year group projects in multimedia information systems from a joint course taught by the Information Science and Design Studies departments at the University of Otago. The group sizes were four (three groups) or five (two groups) and the ratio of Information Science to Design Studies students were 3:1 (two groups), 2:2 (one group) and 3:2 (two groups). As described previously, this mixture of personnel is common in commercial projects where groups are frequently made up of people from very different backgrounds, such as graphic design, programming, video/audio production, project management and interface design (England and Finney, 1996). This situation should therefore be considered as reflective of the commercial environment.

For most of the class this was their first exposure to MMIS development. Projects were to be delivered in two phases: the prototype then the final system. During the prototype stage students were 'learning by doing', gaining experience with the various tools and technologies. While most students had little experience of multimedia *per se* when entering the course, they did have sufficient training is the various areas of multimedia listed above. Typically the Information Science students were skilled in the areas of programming, requirements gathering and systems management while the Design students had skills in the areas of graphic design, interface design and the preparation of audio-visual material.

The projects were undertaken with the goal of producing a cross-platform system. Applications included an interactive shopping mall, a guide to mountain bike tracks and a music catalogue. Media components were built with commercial development packages and two authoring environments were used. The choice of environment was important since it affected authoring time and the manner in which the media components were prepared. The two environments were Macromedia Director and Apple Media Tool (AMT). These tools employ quite different metaphors. Director uses a time line with media included on different

channels, while AMT uses an iconic metaphor with individual screens (see Figures 3 and 4). The time line approach is more ponderous for interactivity since each screen is simply a point in time. This means that an interactive session will consist of 'jumping' around a timeline; this is not as straightforward as using links between screens. However, Director uses Lingo, a powerful scripting language, whereas AMT has no scripting language as such. This means that although AMT is easy to use for simple systems, for more complicated projects Director's scripting capability may prove to be more useful. It would be interesting to have some appreciation of the impact of each environment on development effort.
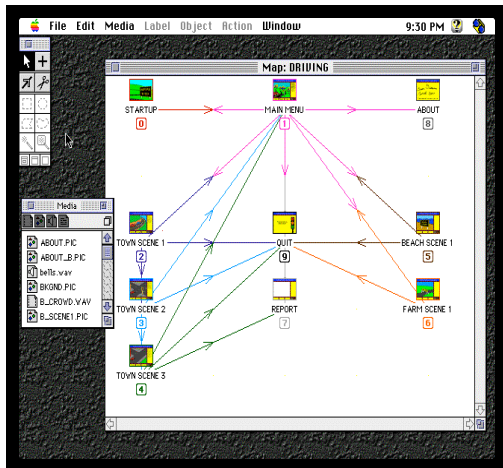
FIGURE 3: AMT SCREEN MAP   FIGURE 4: A DIRECTOR 'SCORE'

Analysis

Two sets of data were available for analysis, the first being the data related to the creation of media components. This data set included 45 observations all associated with graphic media components (thus no comparison of the influence of different media types on effort was possible, for reasons discussed below). For each component, the creation effort, development (authoring) environment, number of objects and component form were recorded. Correlation analysis was performed across the data set to check whether any of the characteristic variables were related to development effort. Although no significantly useful relationships were identified, some observations were made. It was evident that the development environment had some impact on effort (so analysis was performed on two data subsets but with no further success), as did the component form, in that the few screens took significantly greater effort to develop than buttons and animation sequence components.

The screen authoring data set comprised eighteen observations, for which authoring effort, authoring environment, number of links, number of objects and number of events were recorded. It was evident from examining the data that again the two environments showed differing characteristics - the set was therefore split into one of eleven observations and one of seven. Relationship analysis using scatter plots and correlation assessment for the eleven observation data set showed that none of the screen characteristic measures were related to the associated development effort. Analysis of the smaller data set, however, revealed a very strong and significant linear relationship between both the number of objects and the number of events and development effort (the correlation coefficients are shown in Table 1).

6

TABLE 1: EFFORT-SCREEN CHARACTERISTIC CORRELATIONS

	Effort and No. objects	Effort and No. events
Pearson's correlation	0.98	0.98
Spearman's correlation	0.88	0.94

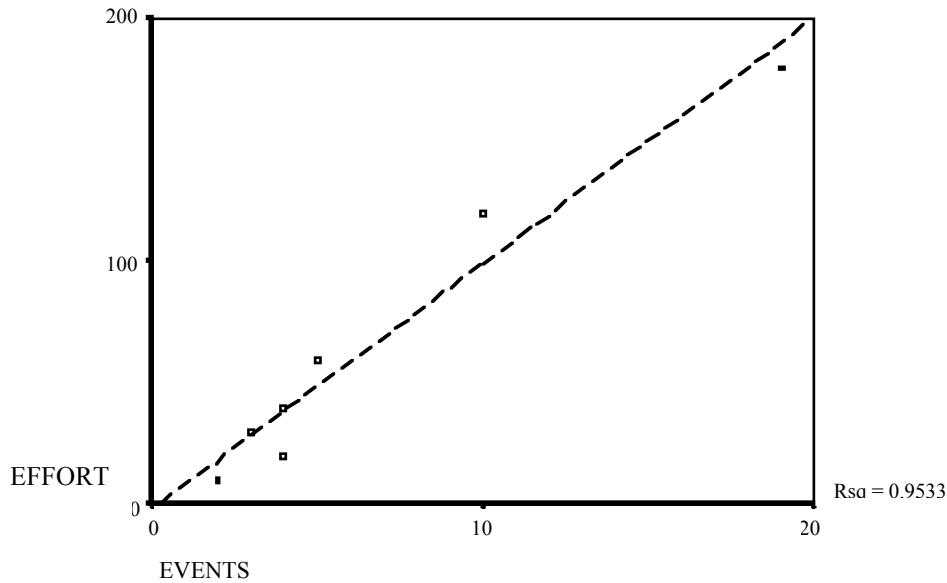FIGURE 5: REGRESSION LINE FOR FIT OF EFFORT BASED ON EVENTS

Figure 5 illustrates the goodness of fit for the regression of screen authoring effort based on the number of screen events for one of the authoring environments. Although the data set is very small, the strength of the relationship gives us encouragement for further investigation.

Lessons Learned - Observations, Limitations And Difficulties

Clearly much of what we set out to achieve in this study remains unresolved at this stage. The most significant problem was data capture - the students were simply most reluctant to complete the data collection sheets, directly contributing to (i) the very small data sets and (ii) the infeasibility of undertaking much of the planned analysis. For example, all temporal media (video, audio and animation) should have had an associated duration entered on the datasheet. However this was rarely provided. In all cases video and audio pre-existed and only needed to be digitised and edited - however, no records of even these tasks were taken. Similarly, digitising and editing small pieces of audio were seen as 'unimportant' and therefore the times were not recorded. In the final analysis, only three video-associated records were received, too few to be of any use, and no sound or animation data was received. Our objective of assessing the effort needed in the development of various media types was therefore not possible.

A computer-based collection system could solve many of these problems. For example, valid fields only would be highlighted for a particular media type and integrity rules could be enforced at data entry to ensure that all required fields had been completed with valid values. This would also have the effect of streamlining what is at present a tedious, time consuming two stage process (collecting the datasheets and entering the data). Moreover, developers should where possible be given some form of incentive for participation - in a class situation

marks may be a possible reward. If developers are educated concerning the benefits of and rationale for a metrics program they feel part of the process and are more likely to cooperate. Another limitation should also be acknowledged at this point. Student data is often criticised as being unrealistic in terms of 'real' development. Admittedly, the effort required may not be comparable to that needed by commercial developers, particularly as the learning curve formed part of the effort. On the other hand, all of the students began the projects with the same level of experience, so the figures should be appropriate in relative terms for the sample.

CONCLUSION AND ONGOING WORK

It remains our belief that media type, development environment and media component characteristics all have an impact on associated development effort. Although this preliminary study has been unable to empirically illustrate some of these assertions, there is adequate justification to continue with the work. At present, further data collection is being performed under more controlled supervision (something that was lacking in the original exercise) in the hope that larger data sets will enable us to more effectively determine whether useful relationships exist among the data items of interest.

REFERENCES

Albrecht, A.J. (1979), "Measuring application development productivity", in *Proceedings IBM Application Development Symposium*, Monterey CA, 83-92.

Beautement, P. (1991), "Review of interactive video systems and their possible application to training in the 90's", *Interactive Learning International,* 7, 45-54.

Boehm, B. (1981), Software Engineering Economics, Englewood Cliffs NJ, Prentice Hall.

England, E. and Finney, A. (1996), Managing Multimedia, Harlow, Addison-Wesley.

Gao, X. and Lo, B. (1996), "A modified function point method for CAL systems with respect to software cost estimation", in *Proceedings of the Software Engineering Conference SE:E&P'96*, Dunedin, New Zealand, 125-132.

Jeffery, D.R. and Low, G. (1990), "Calibrating estimation tools for software development", *Software Engineering Journal,* 5(4), 215-221.

Gruskin, L. (1994), "Multimedia consulting", in Keyes, J. (ed), The McGraw-Hill Multimedia Handbook, New York, McGraw-Hill, 24.1-24.11.

Marshall, I.M., Samson, W.B. and Dugard, P.I. (1994), "A proposed framework for predicting the development effort of multimedia courseware", in Kappe, F. and Herzner, W. (eds) Multimedia/Hypermedia in Open Distributed Environments, Springer-Verlag, 161-180.

Merrill, M.D., Li, Z. and Jones, M.K. (1991), "Limitations of first generation instructional design", *Educational Technology*, 30(1), 7-11.