Towards a Taxonomy for Guiding Multimedia Application Development

Anne Hannington
Department of Computer Science and Computer Engineering
La Trobe University
anne@cs.latrobe.edu.au

Karl Reed
Bond University
(on leave from La Trobe University)
kreed@cs.latrobe.edu.au

Abstract

For research into multimedia development practices to progress, the actual processes and activities performed need to be recorded and compared. For accurate comparison, some agreed means of describing the material is needed. In other fields such as botany and zoology, taxonomies are constructed to facilitate such recording and comparison.

This paper draws on the published literature on multimedia to obtain appropriate terms, and then constructs a taxonomy using traditional techniques, in particular, faceting. We compare our results with existing taxonomies and argue that the latter's focus on aesthetics is not suitable for research into process and estimating issues. Finally, we discuss the use of the taxonomy as a basis for comparing multimedia projects, and as a basis for identifying the critical factors in multimedia project development. This will be used to support the development of improved process and estimating models.

1. Introduction/Motivation

This paper proposes a taxonomy for use in the study of multimedia development practices as part of our investigations into the relationship between multimedia processes and traditional software development. Appropriate taxonomies are essential for the proper study and analysis of any field, and software engineering and multimedia are no exceptions (the authors located more than 100 publications on information technology related taxonomies in a non-exhaustive search). Within software engineering, taxonomies have been proposed in a wide range of areas. Perry and Kaiser used the IFCS paradigm to produce a taxonomy for development environments [1], while Merrit [2] produced a taxonomy of sorting algorithms, extending Knuth's [3]. Mehta et al [4] proposed a taxonomy of software connectors. For Software Engineering as a whole, the SWEBOK project [5] produced its Knowledge Area Descriptions based upon Bloom's taxonomy. While not specifically software engineering related, Kim and Ellis' [6] taxonomy for workflow architectures, based on collaborative work using distributed systems, may prove useful in software process recording.

In the multimedia related areas there is a significant volume of taxonomy-related work. Tetzlaf and Flynn [7] use a simple taxonomy to describe video servers, while Chakrabarti et al [8] show the use of a pruned version of the Open Directory (http://dmoz.org) RDF to automatically classify web pages, improving significantly upon earlier results reported in [9].

The evidence is that taxonomies can and do play major roles in ensuring that human information exchanges are effective, by permitting the use of shared concepts which facilitate classification, and hence comparison. In general, the existence of an appropriate taxonomy is considered a mark of discipline maturity.

The taxonomy presented here is intended to assist in the comparative study of two information-technology related development activities. The importance of this can be seen from the fact that both areas share common features, for example, the use of specific implementation mechanisms in both, and, the story-construction concepts in software development's Extreme Programming [10, 11], and requirements analysis.

Our interest, however, extends beyond comparative studies. Current research reveals that the multimedia design methods proposed by academia are not widely used by those engaged in multimedia application development [12]. In addition it is suggested that the methods/models currently being used are not adequate to capture much of the unique behaviour of multimedia applications [12, 13]. The absence of a singular defined method is not only due to the different media being integrated, but also due to the diverse origins of multimedia developers and as a consequence the diverse skills and techniques they bring with them.

The authors argue that a formal basis for describing processes, as distinct from describing the design of multimedia projects is required, and propose that the

Proceedings of the Ninth Asia-Pacific Software Engineering Conference (APSEC'02)
0-7695-1850-8/02 $17.00 © 2002 IEEE

development of a taxonomy is a first step in this direction. In order to collect appropriate terms for classification we first discuss multimedia development processes and techniques. The absence of a precise definition for multimedia is then noted, and a faceting technique is used to develop a new taxonomy. Some examples of its use are presented, then it is compared with existing multimedia taxonomies. We conclude by summarising, and proposing future work.

2. Multimedia process models

Multimedia application development, like traditional software development, takes place at various levels. And, like most software process models, the detail specific to the domain is hidden behind the use of generic terms, in order to describe a general model. The following discussion looks at suggested process models for multimedia development, focusing on design. In doing so we look at the steps proposed by the general models and compare these to models derived specifically with educational multimedia applications in mind. We then discuss current multimedia design techniques cited in the literature and their relevance to capturing the many facets of multimedia design.

2.1. Generic multimedia models

The 1994 Apple Computer publication *Multimedia Demystified* was written to address the needs of multimedia developers as identified by a 1992 survey of over 500 participants [14]. The life-cycle model presented is drawn from the disciplines recognised as converging to create multimedia: publishing, film, and software production. The model suggested consists of the following main phases:

- Concept and Planning
- Design and Prototype
- Production
- Testing
- Distribution
- Follow-up

Though these phases may not necessarily be as clear-cut as defined above, particularly between 'Design and Prototype' and 'Production', it is observed that "[e]very creative process follows a general pattern that moves from early conceptualisation to final realization. The details within each step and the extent of each stage will differ with the media or use but overall the process should be the same" [14].

These basic steps can also be identified in the client-centred process model put forward by England and Finney [15], shown in figure 1. Of particular interest is the 'Detailed Specification' phase, where the need to establish details of the content, platform, media and techniques, interface and interactivity is emphasised. These factors correlate with those considered important for 'Concept and Planning' as discussed in [14], and are also identified in [16, 17]. From these, terms that describe a multimedia project have been identified as outlined below.

2.1.1. Content and media. The focus on content and its treatment (the media used and the information it conveys) is one of the key differentiators between non-multimedia and multimedia applications, and indeed between multimedia applications.

The choice of media to represent content has a considerable impact on the effectiveness of the application and on development considerations. The creation of media (text, graphics, animation, audio and video) is a creative process, usually involving much iteration, and can begin very early in the development cycle. Early versions of artwork, for instance, can provide the client with a picture of the visuals, and allow the developer to estimate development time and storage requirements for the finished artefact. The use of prototyping thus plays an important role, permitting cost and storage estimates for the full implementation [16].

Most multimedia process models advocate use of strict evaluation and revision within the iterative cycles of development [14-16], see figure 2. This requires tracking the development of the individual media used within a

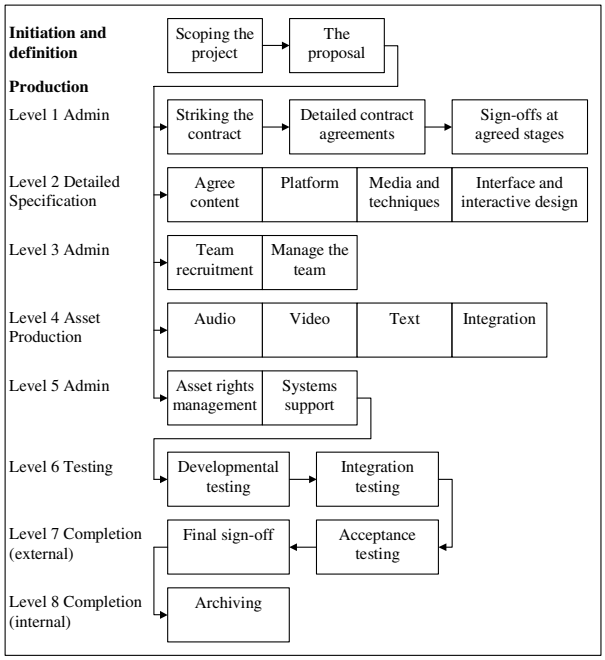

Figure 1. Client-centred process model [15]

project, and any scheme wanting to capture development effort will have to take this into consideration by allowing the various phases of creation to be recorded.

In addition, the temporal nature of animation, audio, and video need to be modelled in design, and suitable techniques are required to capture this behaviour.

2.1.2. Interactivity and interface design. Interactivity is defined in [14] as being either: *passive*, a linear presentation in which the users only interact to start or stop the presentation; *interactive*, where users are able to "chart a personal course through the content"; or *adaptive*, where users are able to enter their own content and control how it is used. "[T]he greater the level of interactivity, that is, the greater the audience involvement and control, the greater the effort required to design and produce a multimedia product. The level of interactivity also gauges complexity when estimating project duration and cost" [14]. Therefore, being able to describe in terms of interactivity is important.

Interactivity considerations subsume the notion of navigation, and the importance of interface design, as indicated by their grouping together in figure 1. Navigation is often described in terms of linear navigation, hierarchical navigation, or networked navigation, and applications can display combinations of these. The more complex the navigational structure the greater the design effort will be in order to ensure usability. This is further complicated when the media mix is considered.

2.1.3. Platform. The choice of media, interface design, and level of interactivity cannot be determined without consideration of the delivery platform. This must occur early in development, since it determines the size of media, and bandwidth availability [14-16]. Security issues are also determined by the platform and need to be represented.

2.2. Domain specific multimedia models

The existence of general models that provide a basis for determining critical factors for multimedia design, however, does not imply a blanket solution for construction. The process is not only going to vary widely depending on the media used, but also on the type of the multimedia application itself. By identifying the different content areas and treatment of projects within a particular domain, a more accurate classification of these projects can be obtained.

This is evident in Beasley's discussion on the analysis and design phases of instructional multimedia software development [18]. While his description of the analysis phase, can be mapped onto that considered good practice in software development (stating the objectives of the application, determining the intended audience, concept analysis, determining restrictions on design, development and implementation, cost/benefits analysis, feasibility report, and client sign-off), the need for a content expert, familiar with instructional design is evident.

This becomes more significant in his discussion on design. Each activity and sub-task requires specific domain knowledge. For instance, the considerations required when mapping instructional objectives to lesson modules, the appropriate length of a given module, or choice of appropriate instructional design models. Knowledge of the instructional design models can aid in further classification of a given project's type. (See Appendix – Instructional Design Model Facet)

Steps that cater for instructional design are also included in the multimedia educational design model presented in [19].

Domain specific steps, such as these, need to be considered when comparing multimedia projects both across, and within domains, to be able to fully compare and contrast development activities.

3. Multimedia Design Techniques

The models reviewed illustrate the focus on techniques from film and publishing (see also Luther [20]). Each model cited has a production phase, and there is an emphasis on prototyping. Navigation-maps, storyboards, scripts and flowcharts, among others, are all stated as design artefacts. However, recent surveys [12, 21] suggest that techniques used in traditional software development are also being employed.

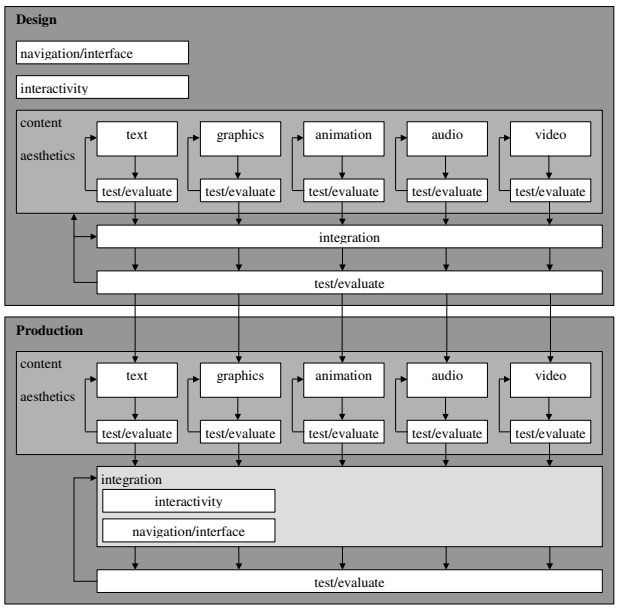

Figure 2. The iteration and revision cycles blur the boundary between design and production

Non-multimedia software design has traditionally been captured through the use of a wide range of techniques, e.g. data-flow diagrams, and structure charts, for data and information flow; E-R diagrams, for modelling database requirements; flow charts, and Petri-nets, for modelling behavioural issues; and OO techniques using UML notations to capture all aspects.

While there is evidence that these techniques are used in multimedia development [12], it is not clear exactly what artefacts and behaviour they are modelling. Traditional software design techniques are not equipped to model many of the new artefacts and behaviour that multimedia applications have introduced. For instance, they do not adequately capture the navigational aspects of multimedia design, nor do they capture the temporal aspects, and resulting synchronisation that may be required. Also little guidance is given to content modelling, and while OO-techniques may be used to model the existence and behaviour of interface widgets, they give no representation of the "look". Hence, if they are to be used and "early visualisation" [14] is required, they need to be combined with a prototyping technique that captures the visual.

As already remarked, techniques having their roots in the film and publishing industries are being applied, as the predominant use of terminology from film, and consequent focus on creativity in design shows. Storyboards, scripts, mock-ups etc. have all been cited [12, 13], however these too lack notations for aspects of multimedia, such as interaction.

Researchers, however, have begun to address this issue by working on tool support for the use of storyboards in early design [22, 23], including the ability for navigation and interaction with basic widgets [13, 24, 25] and more recently the ability to respond to more complex user behaviour [26]. The goal is an effective means for capturing and communicating early design ideas, through techniques that simulate the behaviour of the design.

These promising "informal" techniques for capturing the behaviour required [26] must still be transformed to a more formal notation for implementation.

The problem of capturing human activities in a manner that is sufficiently informal for non-programmers to understand, yet sufficiently precise for developers to use as a specification, is well known in systems analysis [27] and requirements engineering domains. It would appear that this is an even greater difficulty where multimedia projects are concerned, due to the presence of artistic content and the difference in representational techniques between artistic and software personnel.

Formal design techniques, (e.g., HDM, RMDM, OOHDM) [28-30], have their basis in hypermedia and concentrate on the issue of navigation. Notations, terminology, and design approaches for navigation are imported into the software design models they use as their basis, namely ER and OO. While these techniques also allow for the modelling of basic interaction, they currently do not allow for temporal considerations.

The preceding discussion has provided a basis for describing a multimedia application, showing important elements for consideration within the design phase and how these elements have been the focus of existing design techniques. However, for the effectiveness of these design techniques to be determined a means of recognising the contribution and relevance of these techniques to particular steps in design is required.

4. Taxonomy

Various models for development, and methods for design exist for multimedia, but they deal with different domains and stages of design. Further, they have similarities, and differences, which need to be formally recognised before comparisons and discussions can accurately be made. A widely accepted system capable of classifying multimedia applications, using the characteristics identified in the previous discussion, is required. As has been shown in other fields, an appropriate (multimedia) taxonomy, supporting different views, can be used to address this problem. We derive our taxonomy by considering its classification capabilities.

Firstly, we need to be able to classify multimedia projects into their different application domains, allowing comparison between projects of similar motivation and parent disciplines. By parent disciplines we mean those that have embraced the use of multimedia.

Secondly, we need to determine those characteristics of a multimedia application that impact on development time and effort. This involves consideration of both "authoring-in-the-large" and "authoring-in-the-small" as defined in [28].

Thirdly, we need to classify the existing design techniques in order to establish which activities in design they model effectively and to ensure all characteristics identified in the second step are adequately represented during development.

Hence, it will be possible to establish where the techniques will be of most use and identify those cases where techniques are either inappropriately applied, or do not exist. This will also assist in the process of comparing traditional software development processes to their multimedia counterparts.

Our proposal for a taxonomy for establishing the characteristics of a multimedia application (focusing on design) follows.

4.1. Defining Multimedia

The first step in taxonomy construction is establishing the subject field [31]. In the preceding discussion we have freely used the term "multimedia". Clarification of this term is required to establish the sort of applications we are dealing with. Examination of various multimedia definitions offered suggests that within a certain framework the definition depends on the user's needs. This leads to inconsistencies regarding a number of aspects. Some sources define multimedia as "using several media" [32], implying more than two, others only require more than one [33]. The presence of interactivity is considered essential in some definitions [15, 34], while others make the distinction between linear multimedia and non-linear, interactive multimedia [16, 35, 36]. A further categorisation of hypermedia is recognised in [15], where a structure of linked elements is provided through which the user can navigate.

The following definition given by Feldman is commonly cited [37] [15]:

> *The seamless integration of data, text, images and sound within a single digital information environment.*

We conclude that applications may be either online, offline or hybrid, and contain linear and/or interactive multimedia, where multimedia is the combination of two or more different types of media controlled by a computer.

4.2. Multimedia Application Domains

As indicated previously, determining a multimedia project's application domain (by considering the parent disciplines involved) is the first step towards selecting a viable development process, and establishing which elements are critical for the project's success. For instance, when designing a multimedia educational package, the consideration of instructional design principals may require the inclusion of the steps, Lesson Module Identification and Sequencing, Instructional Design Models Identification, and Assessment Strategies Determination [18], which would be of no benefit to, an e-Commerce application, for example. While these steps may fit broadly into the category of "content authoring", and occur somewhere in the design phase, hiding them there is of no benefit to development time estimation and metrics establishment, nor when classifying a project.

The grouping of multimedia projects according to their application domains will provide a number of insights:

1. Identify the personnel likely to be working on the project
2. Identify the origins of process models that may already be in place
3. Identify the design techniques that may be prevalent
4. Identify the tools that may be utilised

It is then necessary to determine how well existing techniques have adapted to the unique requirements of multimedia i.e. do they cater for all aspects adequately.

Considering the application domains that have embraced multimedia we recognise the following categories, based on [38]:

- *Multimedia information systems*: databases, information kiosks, hypertexts, electronic books, and multimedia expert systems
- *Multimedia communication systems*: computer-supported collaborative work, videoconferencing, streaming media, and multimedia teleservices
- *Multimedia entertainment systems*: 3D computer games, multiplayer network games, infotainment, and interactive audio-visual productions
- *Multimedia business systems*: immersive electronic commerce, marketing, multimedia presentations, video brochures, virtual shopping…
- *Multimedia educational systems*: electronic books, flexible teaching materials, simulation systems, automatic testing, distance learning…

Multimedia information systems, communication systems, and business systems development processes are founded on software development, while multimedia entertainment systems derive theirs from the film industry. Multimedia educational systems (the goal setting of these is normally achieved using Bloom's taxonomy) have their foundations in both Education and Publishing. We have already seen the need for instructional design considerations.

These backgrounds, however, are not mutually exclusive, with the result that the semantics of some elemental terms are contextual. Further, examination of Gonzalez's [38] categories shows that they are not distinct, nor orthogonal. In fact, the elements of "Multimedia Information Systems" for example, may be used to construct a collaborative work system. A multimedia expert system may include elements of a Multimedia entertainment system, and so on. Despite this, such classifications are useful, since the concepts that they represent are well (or should be) well defined.

In addition, one must ensure that the problem domain and the solution space are classified separately, to allow the appropriate linking between the two. The significance of this is discussed in [39].

4.3. Classification System

A faceting technique [31, 40] was used to classify the elements of multimedia development. This involves grouping the various subject elements within the field of knowledge into their component elemental classes, known as "facets", where each facet is "derived from the parent universe by a single characteristic of division" [31]. The elements that are categorised within each facet are called terms. The combination of facets used to represent a given artefact is known as the schedule. The ordering of facets within the schedule depends on the ranking determined by the classification maker with the following basis: "[b]y convention, this ordering consists of a one-dimensional list where the conceptual closeness between any two terms is indicated by the number of terms listed between them. When classifying in a faceted scheme, the most significant term in the classification description is a term selected from the facet most relevant to the user" [40]. Other arrangements may also be defined, such as simple to complex, spatial, chronological or an alphabetical listing if the terms do not suggest a natural order of precedence [31].[1]

The choice of a faceted system arises from the need for a representation that allows classification from different views, and by a combination of terms [31].

4.3.1. Project classification. The following factors have been identified as descriptors for a multimedia project in terms of characteristics influencing development:

- Domain
  - Type
- Platform
  - Security
- Interactivity
- Interface
- Content
- Programming

From these, the following schedule of facets can be identified to describe a multimedia project[2]:

<domain/type>,<platform>,[<security>]

The facet, *domain*, represents the multimedia application's domain. *Type* is a further classification of the project within the domain. The terms used are listed in the Appendix.

Once the application domain and type have been established, certain characteristics of the project need to be identified. Projects displaying similar characteristics based upon their element classification, can be grouped further, facilitating the comparison of process modelling techniques, design techniques etc.

The facet, *platform*, represents the delivery platform for the project, and influences security, storage and efficiency issues.

The facet, *security*, represents the issues of security that may need to be considered, depending on the delivery platform and project type.

Applying the first three facets in the schedule yields the following description of an education package for online (intranet) testing:

Multimedia education system, automatic testing, online, authorization, authentication

Interactivity, interface, content, and programming are treated differently, as these may not be uniform for the whole project. The following grouping of facets is proposed:

1. Those specific to artefact integration and interaction
2. Those specific to artefact creation

This requires the division of a multimedia project into sections, permitting its description as being composed of a number of sections, where each section has associated screens and a means of navigating between them. Each screen then has an interface, an associated level of interactivity and programming, and displays content. Media artefacts represent content.

This yields the following sample schedule:

```
mmProject   ::= <domain/type>,<platform>,
                [<security>],<section>+
<section>   ::= <screen>+, <navigation>
<screen>    ::= [<interactivity>], <interface>+,
                <content>, <programming>+,
                [<navigation>]
<content>   ::= <media artefact>+
```

Note this is only one example of a possible schedule, other arrangements, for instance, including programming and content at the section level as well, may be desirable.

The inclusion of marginal subjects [31] must also be considered in this process. This allows categorisation using terms and facets that may be central to other related

[1] Faceting as a classification technique has been used in requirements re-use [27]. Further we note that taxonomy production could be automatic, as could classification. However, the process would no longer be transparent to humans.

[2] We use an extended BNF, where
<> represents a non-terminal
[ ] implies optional
+ implies 1 or more
* implies 0 or more
Terminals are the terms within the facets.

classifications. We have seen the usefulness of this with instructional design (see Appendix).

4.3.2. Artefact Creation. The facets for media artefacts have been chosen to highlight the media-specific characteristics that need to be considered for the design of the particular artefact. These are:

- behaviour/media,
- origin,
- state,
- size,
- format, and
- duration

Since the behaviour of media artefacts can be classed at the highest level as either being static or temporal, this level of classification has been included in view of its importance to the design techniques chosen (i.e. current design techniques for hypermedia fail to account for temporal aspects of media – as mentioned previously in Section 3).

The *media* type of the artefact, has implications in terms of developer skill required, creation time, and format and space requirements for the final product. The *origin* facet, describes the origin of the media artefact, and is particularly important for development time and cost estimation, as well as determining the required skills. These facets are outlined in figure 3.

An artefact, media or otherwise (see Design Technique/Artefact facet in Appendix), may pass through many revisions from design to implementation and require different tools for different steps. Knowledge of the current *state* of development of an artefact becomes important for effective project tracking and management.

The storage size and format of an artefact are also of importance, as is duration for temporal media.

Therefore, a multimedia artefact could be classified using the following schedule (for details of facets see Appendix):

Media facet	*Origin facet*
Static	Acquired
Text	Repurposed
Graphics	Created
Photographs	
Temporal	
Animation	
Audio	
Music	
Voice	
Sound Effect	
Video	

Figure 3. Media and Origin Facets.

<media>, <origin>, [<state>], [<size>], [<format>], [<duration>]

As an example, the use of a sound effect already in our archives or supplied by the client could be described using the first three facets as follows:

Temporal/audio, sound effect, acquired

We remarked earlier that the taxonomy can be used to identify methodologies required at some point in the project, or the steps necessary to "acquire" an artefact. In this case, our description shows that the design technique must be able to capture the temporal aspects of this artefact, that there is no requirement for audio equipment to produce it as it already has been acquired, and that the development time is negligible. If it is contained within our sound effect library then we may be able to assume we own the rights to it, if the client is the supplier, we need to check ownership. If the delivery platform of the project has been classified we may also know our constraints as far as duration, size and format.

We are now able to define schedules depending on the user's need for applying the taxonomy. For instance, should the user want to keep a record of the current development state of an artefact, and the phase at which it reached that state, they can include the *state* and *operations* facets in its description, and define a schedule as follows:

<media>,<origin>,<state>, <operations>

Consider an example, where the production of an animation was begun during the concept and planning phase to permit the "look" of artwork to be viewed, but the artwork was not yet fully rendered. This would then yield:

Temporal, animation, partially-rendered, concept and planning

While this gives information regarding the artefact, note that a scheme for its identification will also be required. This would usually be achieved by a file-naming and version scheme, imposed by the developer. This is distinct from the artefacts identification within the classification scheme. In this preliminary phase of the taxonomy a formal notation scheme [31] is seen as a future enhancement, as the taxonomy stabilises.

5. Existing Taxonomies

Our taxonomy differs from existing multimedia taxonomies as a result of the impact of its intended use upon its construction. The motivation for Heller and Martin's initial media taxonomy [41], extended and

renamed as a multimedia taxonomy in [42], was to provide a mechanism for choosing appropriate media, and media representation, for a given application. Their taxonomy classifies along three dimensions, media type, media expression, and context. A media artefact is classified by: its type, (text, sound, graphics, motion, multimedia) i.e. the type of expression used to represent the media; general, a category provided to allow the evaluation of characteristics that are medium specific, "but may be present in all expression formats" [42]; elaboration, where the information is presented without editing; representation, where the information is presented in "a more abbreviated stylised mode"; and abstraction, where information is represented in an "iconic" form; and placed in a context. The taxonomy is then used to aid in the selection of media expression, and combination of media and expression, to best represent the content at hand [41].

Another motivation for this taxonomy was as a basis for evaluating multimedia presentation. The context dimension added in the 1999 version aids this by providing the following sub-categories: audience, discipline, interactivity, quality, usefulness, and aesthetics. This dimension helps further specialise questions to aid in evaluation of media usage, by placing them in context. For example, questions regarding representation text can be asked in terms of quality, or usefulness in conveying a given message.

The taxonomy is also being used as a basis for suggesting the appropriate media representation of different information types and their organisation [43].

Aleem [44], extending Heller and Martin's work, further defined the interactivity category. This terminology can be included in our taxonomy (see Appendix). This is of particular interest if a level of associated development effort is assigned to each type of interactivity.

While Heller and Martin et al are seeking to define a taxonomy that aids design choices and evaluates the effectiveness of these choices, they have focused specifically on possible treatments of media. The fact that these treatments have a representation in design is of importance to our taxonomy. A possible connection between the two taxonomies could arise with regard to the classification of media within the expression dimension. It has been suggested that as one moves from elaboration to abstraction the effort in creating the media increases [41]. By grouping like media in terms of expression and recording development time, a suitable metric for estimation may be discovered.

6. Conclusion/Future Work

This paper has presented a new taxonomy for classifying multimedia projects, based upon the concept of a "schedule" which uses facets as means of describing projects. We have shown its relationship with other major taxonomies of this type, identifying similarities and differences, and common material. Observations on the importance of taxonomies in general have been made. As stated earlier, our objective is to use our result as a basis for comparing multimedia projects, and to describe the relationships between them and software development projects. The taxonomy, when used in conjunction with an appropriate taxonomy for software projects, will allow us to reliably extract a range of information not limited to:

- determining which facets are significant in estimating projects
- which activities are common/*not* common to both classes of development
- what the dominant technologies in the multi-media projects are, and the factors determining their choice
- whether there exist dominant artefact types whose frequency of use impacts cost and development time
- establishing the relationship between artefact usage and tool and design technique usage
- whether there exist approaches in one class which might be imported into the other
- whether the patterns of organisation and deployment of skills, artefact re-use etc. from one class are evident in the other, and
- the impact of content richness and artistry

We will also be able to compare proposed multimedia process models with those used in the field, using the taxonomy as a basis for detailed process recording. The tests on the use of the taxonomy included herein show that it will perform satisfactorily, however, there are facets that are not complete, and which need further development. Other facets that are to be added address testing, metrics and estimating.

Another major use for the taxonomy will be as a basis for the design of surveys and process and project recording case–studies.

Acknowledgements

Authors wish to thank Dr. Rahayu and Prof. Dillon of the Computer Science and Computer Engineering Department at La Trobe University for their critical comments and encouragement with this work. The first author acknowledges the financial support from the Victorian Government (ICT scholarship), and of La Trobe University. The second author gratefully acknowledges the support in the form of a visiting position in the School of Information Technology, Bond University, and the encouragement of Prof. Gupta.

References

1. Perry, D.E. and G.E. Kaiser, "Models of software development environments", in *10th ICSE*, IEEE Comput. Soc. Press, Washington, DC, USA, 1988, pp.60-8.
2. Merritt, S.M., "An inverted taxonomy of sorting algorithms", *Communications of the ACM*, 1985. **28**(1): pp. 96-9.
3. Knuth, D.E., *The art of computer programming*, 2nd ed. Addison-Wesley, Reading, Mass, 1973.
4. Mehta, N.R., N. Medvidovic, and S. Phadke, "Towards a taxonomy of software connectors", in *ICSE 2000*, ACM New York, NY, USA, pp. 178-87.
5. Bourque, B., et al., *Approved Baseline for a List of Knowledge Areas for the Stone Man Version of the Guide to the Software Engineering Body of Knowledge*. http://www.swebok.org, 1999.
6. Kim, K-H. and C.A. Ellis, "Workflow Performance and Scalability Analysis Using the Layers Queuing Modeling Methodology", in *GROUP'01, Sept. 30 - Oct. 3, 2001*, ACM, Boulder, Colorado, USA, 2001, pp. 135-43.
7. Tetzlaff, W. and R. Flynn, "Elements of scalable video servers", in *COMPCON'95. Technologies for the Information Superhighway*, IEEE Comput. Soc. Press, Los Alamitos, CA, USA, 1995, pp. 239-50.
8. Chakrabarti, S., et al. "The Structure of Broad Topics on the Web" in *WWW2002, May 7 - 11*, ACM, Honolulu, Hawaii, USA, 2002, pp. 251-62.
9. Chakrabarti, S., B. Dom, and P. Indyk. "Enhanced hypertext categorization using hyperlinks", in *SIGMOD '98*, ACM, Seattle, WA, USA, 1998, pp 307-18.
10. Wells, D., *The Rules and Practices of Extreme Programming - User Stories*, http://www.extremeprogramming.org/rules/userstories.html, 2002.
11. Paulk, N.C., "Extreme programming from a CMM perspective", *IEEE Software, 2001*. **18**(6): pp. 19-26.
12. Barry, C. and M. Lang, "A Survey of Multimedia and Web Development Techniques and Methodology Usage", *IEEE Multimedia, 2001(April - June 2001)*, pp. 52 - 60.
13. Bailey, B.P., J.A. Konstan, and J.V. Carlis. "DEMAIS: Designing Multimedia Applications with Interactive Storyboards", in *MM'01*, ACM, Ottawa, Canada, 2001, pp. 241-50.
14. *Multimedia demystified: a guide to the world of multimedia from Apple Computer, Inc*. Apple-new media series, Random House, New York, 1994.
15. England, E. and A. Finney, *Managing Multimedia - Project Management for Interactive Media*, Addison-Wesley, 1999.
16. Vaughan, T., *Multimedia: making it work*, Osborne/McGraw-Hill, Berkeley, Calif., 1996.
17. Rout, T.P. and C. Sherwood. "Software Engineering Standards and the Development of Multimedia-based Systems", in *ISESS'99*, IEEE Comput. Soc., Los Alamitis, Ca, USA, 1999, pp.192-8.
18. Beasely, R.E., "Instructional Multimedia Software Development: Implications for the Analysis & Design Phases of the SDLC", *Journal of Computer Information Systems*, 1999(Winter 1999 - 2000), pp. 2-6.
19. McDaniel, K. and M. Liu, "A Study of Project Management Techniques for Developing Interactive Multimedia Programs: A Practitioner's Perspective", *Journal of Research on Computing in Education*, 1996. **29**(1): pp. 29-48.
20. Luther, A.C., *Authoring interactive multimedia*, AP Professional, Boston, 1994.
21. Britton, C., et al., "A survey of current practice in the development of multimedia system", *Information & Software Technology*, 1997. **39**(10): pp. 695-705.
22. Baecker, R., et al., "A multimedia system for authoring motion pictures", in *ACM Multimedia 96*, ACM, New York, NY, USA, 1996, pp.31-42.
23. Harada, K., et al., "Implementation of the Anecdote multimedia storyboarding system", in *ACM Multimedia 96*, ACM, New York, NY, USA, 1996, pp.403-4.
24. Lin, J., et al., "DENIM: finding a tighter fit between tools and practice for Web site design", in *Proceedings of CHI 2000,* ACM, New York, NY, USA, 2000, pp.510-17.
25. Newman, M.W. and J.A. Landay. "Sitemaps, storyboards, and specifications: a sketch of Web site design practice", in *Proceedings of DIS 2000,* ACM, New York, NY, USA, 2000, pp.263-74.
26. Lin James, Thomsen Michael, and J.A. Landay, "A Visual Language for Sketching Large and Complex Interactive Designs", in *CHI 2002*, ACM, Minneapolis, Minnesota, USA, 2002, pp. 307-14.
27. Cybulski, J.L., *Application of Software Reuse Methods to Requirements Elicitation from Informal Requirements Texts*, Phd Thesis, La Trobe University, 2001.
28. Garzotto, F., P. Paolini, and D. Schwabe, "HDM-a model-based approach to hypertext application design", *ACM Transactions on Information Systems*, 1993. **11**(1): pp. 1-26.
29. Schwabe, D., G. Rossi, and S.D.J. Barbosa, "Systematic hypermedia application design with OOHDM", in *Proceedings of Hypertext '96.* ACM, New York, NY, USA 1996, pp.116-28.
30. Isakowitz, T., E.A. Stohr, and P. Balasubramanian, "RMM: a methodology for structured hypermedia design", *Communications of the ACM*, 1995. **38**(8): pp. 34-44.
31. Vickery, B.C. and Classification Research Group, *Faceted classification : a guide to construction and use of special schemes*. Aslib, London, 1968.
32. *American Oxford Dictionary*. Oxford University Press, 1980.
33. *The Oxford English Dictionary*. Clarendon Press - Oxford 1989.
34. Tannenbaum, R.S., *Theoretical foundations of multimedia*, Computer Science Press, New York, 1998.
35. Furht, B., *Multimedia tools and applications*, Kluwer Academic Publishers, Boston, 1996.
36. Szuprowicz, B.O., *Multimedia technology: combining sound, text, computing, graphics, and video*, Computer Technology Research Corp, Charleston, S.C., U.S.A, 1992.
37. Wise, R. and J. Steemers, *Multimedia: a critical introduction*, Routledge, London, 2000.
38. Gonzalez, R., "Disciplining Multimedia", *IEEE Multimedia*, 2000. **7**(3): pp. 72-8.
39. Glass, R.L. and I. Vessey, "Contemporary Application-Domain Taxonomies", *IEEE Software*, 1995(July 1995): pp. 63-76.
40. Prieto-Diaz, R. and P. Freeman, "Classifying software for reusability", *IEEE Software*, 1987. **4**(1): pp. 6-16.
41. Heller, R.S. and C.D. Martin, "A Media Taxonomy", *IEEE Multimedia*, 1995(Winter 1995): pp. 36-45.

42. Heller, R.S. and C.D. Martin, "Using a Taxonomy to Rationalize Multimedia Development", in *ICMCS*, 1999, pp. 661-5.
43. Heller, R.S., et al., "Using a Theoretical Multimedia Taxonomy Framework", *ACM Journal of Educational Resources in Computing*, 2001. **1**(1).
44. Aleem, T.A., *Taxonomy of Multimedia Interactivity*, http://www.erols.com/aleem/interact.html, 1997.

Appendix – Partial Listing of Multimedia (MM) Taxonomy

Domain Facet
- MM Business Systems
  - Electronic - commerce
  - Marketing
  - Video Brochures
  - Virtual Shopping
- MM Communication Systems
  - Computer-supported - collaborative work
  - MM teleservices
- MM Educational Systems
  - Automatic testing
  - Distance learning
  - Flexible teaching - materials
  - Simulation systems
- MM Entertainment Systems
  - Infotainment
  - Multiplayer - network games
  - 3D computer - games
- MM Information Systems

Solution Space facet
- Databases
- Electronic books
- Electronic magazines
- Hypertexts
- Information kiosks
- Interactive art and performance
- Interactive music
- Multimedia expert system
- Multimedia presentation
- Streaming media
- Videoconferencing
- …

Delivery Platform Facet
- Online
  - Intranet
  - Internet
- Offline
  - CD-ROM
  - Hard-disk - installation
- Hybrid
  - Online/Offline

Security Facet
- Access levels
- Authorization
- Authentication
  - Digital signatures
  - Time stamping
- File privileges
- Firewall type
- Privacy
  - Algorithm
  - Encryption
  - Key system
- Password storage
- System managed locally
- System managed globally - (remotely)
- E-commerce
  - Transaction - security
  - Secure payment - processing
- …

Navigation Facet
- Linear
- Non-linear
- Hierarchical
- Composite
  - Non-linear/linear
  - Non-linear / hierarchical
  - Hierarchical/linear

Interactivity Facet (based on [44])
- Passive
- Reactive
- Proactive
- Directive

Interface Facet
- Widget
  - Menu
    - Level
  - Button
  - Check box
  - Text box
  - List box
  - Dialog box
  - Slider
  - Form
  - …

Programming Requirements Facet
- Static Web page
- Database
  - Retrieval/storage
  - Retrieval only - (data warehouse)
- Information processing
  - Forms (Web)
- Scripting
  - Client side
    - javascript
    - …
  - Server side
    - php
    - …
- Expert system
- Interface for pre-existing software
- Legacy system
- …

Media Facet
- Static
  - Text
  - Graphics
  - Photographs
- Temporal
  - Animation
  - Audio
    - Music
    - Voice
    - Sound effect
  - Video

Origin Facet
- Acquired
- Repurposed
- Created

State Facet
- Completed
- Demo voice
- Partially rendered
- Sample track
- Space filler
- …

Duration Facet – unit of measure and classification of particular durations into categories would need to be defined by the classifier
- Long
- Medium
- Short

Size Facet - unit of measure and classification of particular sizes into categories would need to be defined by the classifier. I.e. what might be regarded as small when working on a project with CD-ROM as the delivery medium, would be different from when developing for a hand-held.
- Large
- Medium
- Small

Format Facet
- Gif
- Jpeg
- Mpeg
- Pdf
- Plain text
- Post script
- Word
- …

Operations Facet – operations performed on media artefact – reflective of development phases
- Concept and planning
- Design
- Production
- Testing
- …

Design Technique/Artefact Facet
- Mind map
- Information hierarchy
- Content map
- Navigation chart
- Flowchart
- Prototype
- Storyboard
- Interactive storyboard
- Storybook
- Script
- HDM
- OOHDM
- RMDM
- …

Authoring Tools Facet
- Commercial
  - Adobe Photoshop
  - Authorware
  - Corel Draw
  - Dreamweaver
  - Flash
  - Macromedia - Director
  - Netscape - Composer
  - Pro Tools
  - Sound Designer
  - Toolbook
  - …
- Research
  - DEMAIS
  - DENIM
  - …

Skills Facet
- Actor
- Animator
- Content expert
- Editor
- Graphic artist
- Musicians
- Photographer
- Project manager
- Programmer
- Researcher
- Sound/audio engineer
- Sound designer
- Tester
- Testing supervisor
- Videographer
- Video editor
- Writer
- …

Marginal Subjects
Instructional Design Facets (based on [18])

Instructional Design Model Facet
- Tutorials
- Drills
- Practice programs
- Simulations
- Instructional games
- Didactic presentations
- Explorations
- Structured

Observations
- Simulated Personal - Interactions

Instruction Phase Facet
- Present
- Guide
- Practice
- Assess

Instructional Assessment Facet
- Demonstration /performance tests
- Problem solving tests
- Recall tests
  - Fill-in-the-blank
  - Short-answer
  - Essay
- Recognition tests
  - True-false
  - Multiple-choice