Canonical Processes of Media Production

Lynda Hardman*
CWI
P.O. Box 94079, 1090 GB
Amsterdam, The Netherlands
Firstname.Lastname@cwi.nl

ABSTRACT

Creating compelling multimedia presentations is a complex task. It involves the capture of media assets, then editing and authoring these into one or more final presentations. Tools tend to concentrate on a single aspect to reduce the complexity of the interface. While these tools are tailored to support a specific task, very often there is no consideration for input requirements for the next tool down the line. Each tool has the potential for adding semantic annotations to the media asset, describing relevant aspects of the asset and why it is being used for a particular purpose. These annotations need to be included in the information handed on to the next tool.

We specify inputs and outputs to a number of canonical processes we identify in multimedia production. We do not specify the intricate workings of the processes, but concentrate on the information flow between them. Our claim is that by specifying the inputs and outputs required for processes that occur in widely differing uses of media we can identify a small set of building blocks that can be supported in semantically aware media production tools.

Categories and Subject Descriptors

H.1 [**Models and Principles**]: General; I.7.2 [**Document Preparation**]: Multi/mixed media, Hypertext/hypermedia

General Terms

Design, Standardization, Human Factors

Keywords

Media capture, Media annotation

1. INTRODUCTION

There is substantial support within the multimedia research community for the collection of machine-processable semantics during established media workflow practices [2, 3, 6, 8, 10]. An essential aspect of these approaches is that a media asset gains value by the inclusion of information about how or when it is captured or used. For example, metadata captured from a camera on pan and zoom information can be later used for the support of the editing process.

Though the suggested combination of description structures for data, metadata and work processes is promising, the suggested approaches share an essential flaw, namely that the descriptions are not sharable. The problem is that each approach provides an implicit model for exchanging information that serves the particular functionality and process flow addressed by a particular environment. The situation is similar to data formats with included metadata, such as the `mov`[1] or `XMP`[2] formats, where the combination of features hinders general exchange when the application does not support the features supported by the data format.

Our aim here is to establish clear interfaces for the information flow across processes between distinct production phases so that compatibility across systems from different providers can be achieved. We see this as a first step towards a longer term goal — namely, to provide agreed-upon descriptions for exchanging semantically annotated media assets among applications.

The processes should not be viewed as prepackaged, ready to be implemented by a programmer. Our goal is rather to analyse existing systems to identify functionality they provide. On the basis of the processes supported within the system to determine which outputs should be available from the system. In this sense, we hope that system creators will be open to providing the outputs we identify when the processes are supported within the system. We hope that in this way the multimedia community will be able to strengthen itself by providing not just single process tools, but, in the same way that the pipe[3] was so important in the evolution of UNIX, allow these to belong to a (global) suite of mix and match tool functionality.

To validate the model we need to describe a number of workflows in terms of the processes identified in the model and specify the inputs and outputs. Contributions in the workshop include descriptions of video production [9] and new media artwork [7] in terms of the model.

*The model was formed during discussions with Frank Nack, Kurt Piersol and Brigitte Kerhervé. Other contributors are Stephen Kimani, Nicu Sebe and Freddy Snijder as part of the "Multimedia for Human Communication" working group. Lynda Hardman is also affiliated with the Technical University of Eindhoven.

MHC'05, November 11, 2005, Singapore.

[1] `http://www.apple.com/quicktime/technologies/`
[2] `http://www.adobe.com/products/xmp/in-depth.html`
[3] `http://www.cf.ac.uk/psych/CullingJ/pipes.html`

The structure of the paper is as follows. We derive the requirements for describing a single process, then identify and describe a number of processes we see as being canonical to media production. The paper concludes with a discussion of the proposed model.

2. CANONICAL PROCESSES OF MEDIA PRODUCTION

An important requirement for the description of the processes is to ensure that the description details only the process taking place and is independent of whether the process can, or should, be carried out by a human or a machine. This allows for a gradual shift of the processing burden from human to machine as the technology develops. In addition to a description of the process, we also need clear specifications of the inputs and outputs to the process.

We identify the following list of processes based on en examination of existing multimedia systems. We give an explanation of what they are and state their inputs and outputs. An initial diagram can be found at the Dagstuhl web site[4].

While we use single words to name each of the processes, these are meant to be used in a very broad sense. The textual description of each one should give a flavour of the process we wish to express. We refer to the processes by means of the single name.

2.1 Premeditate

Any media capture occurs because someone has made a decision to embark on the process of capturing — whether it be image capture with a personal photo camera, professional news video, Hollywood film or security video in a public transport system. In all cases there has been premeditation and a decision as to when and for how long capture should take place.

In all these cases what is recorded is not value-free. A decision has been made to take a picture of this subject, conduct an interview with this person, make this take of the chase scene or position the security camera in this corner. Already there are many semantics that are implicitly present. Who is the "owner" of the media to be captured? Why is the media being captured? Why has this location/background been chosen? Whatever this information is, it should be possible to collect it and preserve it and be able to attach it to the media that is to be captured. For this we need to preserve the appropriate information that can, at some later stage, be associated with one or more corresponding media assets.

The input to this process is from outwith the system. The output is a set of annotations (set of `annID`), with no associated media asset.

2.2 Capture

After a process of premeditation, however short or long, at some point there is a moment of capture. Some device is used to collect images or sound for a period of time, be it photo or video camera, scanner, sound recorder, heart-rate monitor etc..

Note that in this process, we do not restrict capture to only purely recorded information. Media assets can also be created. For example, images can be created with image editing programs or generated by computer programs. What matters is that a media asset comes into existence, we are not interested in the method of creation per se. If the method is considered as significant, however, then this information should also be able to be recorded as part of the metadata.

In summary, the input to the capture process is a collection of metadata (set of `annID`), for example information available from the premeditation process, and/or the message construction process. As a result of the capture process we have a media asset (`medID`), associated information about the capture and/or creation process (set of `capID`) plus the associated information given as input (set of `annID`).

2.3 Archive

The process of archiving is that of storing a media asset plus existing associated annotations and assigning this grouping an identity so that it can be retrieved as a unit. The input to the archival process is a `medID` plus the associated set of `annID` and `capID`. The output of the process is a component identity (`compID`) plus the identity of the archive in which is has been stored (`archID`). The `compID` is equivalent to the component identity specified in hypertext literature [4, 5].

While archiving is trivial for an individual media asset (assignment of a `compID` within the `archID`), merely increasing the number of items in an archive does not necessarily increase the value of the archive. A valuable archive is one that contains a high percentage of valuable assets in a small archive, rather than a low percentage of valuable assets in a large archive. To maintain or increase the value of the archive, the process of archiving may actually be to discard existing media assets plus their associated annotations. In other words, the process of archival includes deciding whether to select the media asset for inclusion in the archive, and, if so, to archive the annotations already associated with the asset and if necessary include others.

2.4 Annotate

Once a media asset exists and has been included in an archive we still need to be able to add extra information about it. This may include information that could have been collected during the premeditation, message construction or capture processes, but is added later. Any information added does not change the original media asset.

We do not prescribe the form of annotations, but require that they can be created and associated with one or more media assets. The structure of an annotation (`annID`) is a reference to a vocabulary being used (`ontID`), one of the terms from the vocabulary (`attID`) plus a value describing the media asset (this may or may not have an ID). The `annID` can refer to the complete media asset, but the annotation could be more specific. In this case, an anchor mechanism is needed to refer to the part of the media asset to which the annotation applies [4]. An `anchID` is needed to give a media independent means of referring to the part of the media asset and a media-dependent anchor value is required to specify the part of the media asset. For example for an image this could be an area, for an object in a film a time-dependent description of an area of the image. For further discussion on anchor specifications see [5] p53.

We use the term annotation, but wish to emphasize the breadth of our intended meaning. Annotation is often used

[4] `http://www.dagstuhl.de/files/Proceedings/05/05091/05091.PiersolKurt1.Slides.pdf`

to denote a single human user adding metadata to enable search at some later date. Here we see annotation as the broader process of adding partial (more easily machine-processable) descriptions of the content of the media asset. The annotation process can never be complete, since different aspects of the media asset may be made explicit in different contexts. The description assigned to it, however, can be viewed as providing "potential for organisation", or as a step prior to a cataloguing step.

How the annotations are created is not of essence to the process description: they may be human-created or automatically generated, for example, from feature extraction processes. The meaning of the attribute can be obtained through its association with the ontology (recorded in the attribute). The value of the annotation may be one of those specified for the attribute. For example, for the attribute "modality" a value may be "spoken language" or "sound effect". The value may also be numeric, for example for the attribute "colour", in which case the units need to be specified. (Note that specification methods already exist [14].)

Note also that the annotations may not be explicitly assigned by a user, but may be assigned by an underlying system through interaction by the user with the media asset. The information gained in this way should be treated on an equal footing with the information assigned by a human user — the difference being that different values for the "assigner" (if this is deemed significant) would indicate the difference. The "assigner" would be one of the attributes from the annotation ontology.

Note that we make an explicit distinction between the process of associating an annotation with a media asset and archival of a media asset. The former associates information with the media asset. The latter allows the media asset (along with its associated annotations) to be located from a repository of components.

The input to the annotation process is a component (`compID`). The output is the same component plus the additional annotations (set of `annID`). An `annID` contains a reference to an `ontID` and potentially a reference to a `valID`.

2.5 Query

Up until now the processes we describe concentrate on capturing, storing and describing media assets. These are needed for populating the media repository. Once there is an archive (but not before) it can be queried for components whose associated media assets correspond to desired properties. Again, we do not wish to use a narrow definition of the term "query", but intend to include any interface that allows the archive to be searched, using query languages of choice or (generated) browsing interfaces that allow exploration of the content of the archive.

Any query of the system may be in terms of the `medID` of the media assets, or in terms of any of the annotations (`annID`) stored with the media assets or the `compID`. A query would need to specify (indirectly) the annotation(s) being used. The mechanisms themselves are not important for the identification of the process.

The input to the query process is an archive of media components (`archID`) plus a specification of a subset of these. The output is a (possibly empty) set of identified media components (set of `compID`) corresponding to the specification. Note that the output is not a set of `medID`s, but a set of `compID`s that include references to the media assets (`medID`).

2.6 Message Construction

A query implicitly specifies a message, albeit a simple one, that an author may want to convey (since otherwise the author would not have been interested in finding those media assets). The query is, however, not itself the message that the author wishes to convey. Neither is the set of media assets returned as the result of the query. Just as capturing a media asset is input into the system, so is the specification of the message an author wishes to convey. In some sense, there is no input into the process. However, the real input is the collection of knowledge and experience in the author her /himself. The output of the process is a description of the intended message, whether implicit or explicit, to either an author or the system. For example, a multimedia sketch system such as described in [1] allows an author to gradually build up a description of the message. For the message to be machine processable the underlying semantics need to be expressed explicitly. It is expected that a message will be conveyed by the presentation of one or more media assets to an end-user.

While we do not exclude this process as being carried out by a system, we expect that, at least in the near future, it will predominantly be carried out by a human user.

In general, we give no recommendation in this paper for the syntax of the message. We expect that it contains information regarding the domain and how this is to be communicated to the user, but we do not assign anything more than a means of identifying a particular message - the `messID`. The input to the process is thus from outwith the system and the output is a `messID`.

2.7 Organise

While querying allows the selection of a subset of media assets, it imposes no explicit structure on the results of one or more queries. The process of organisation is to create some document structure for grouping and ordering (a subset of) the selected media assets for presentation to a user. How this process occurs is, again, not relevant, but includes the linear relevance orderings provided by most information retrieval systems. It certainly includes the complex human process of producing a linear collection of slides for a talk; creating multimedia documents for the web; ordering shots in a film; or even producing a static 2-dimensional poster.

The document structure is guided by the message, in the sense that if the presentation is to convey the intended underlying message then the document structure should emphasize this, not work against it. The document structure may reflect the underlying domain semantics, for example a medical or cultural heritage application, but is not required to. The structure may be colour-based or rhythm based, if the main purpose of the message is, for example, aesthetic rather than informative.

In the arena of text documents, the document structure resulting from organisation is predominantly a hierarchical structure of headings and subheadings. The document structure of a film is a hierarchical collection of shots. For more interactive applications, the document structure includes links from one "scene" to another. In a SMIL [13] document, for example, `par` and `seq` elements form the hierarchical backbone of the document structure we are referring to here.

The input to the organise process is the message (`messID`) plus one or more media components (set of `compID`s). The output is the document structure (docID) which includes pointers to the media components associated with the substructures.

2.8 Publish

The output of the organise process is a prototypical presentation which can be communicated to an end-user. This serves as input to the publication process which selects appropriate parts of the document structure to present to the end-user. The publication process takes a generic document structure and makes refinements before sending the actual bits to the user. These may include selecting preferred modalities for the user's task or device.

Publication can be seen as taking the document structure from the internal set of processes and converting it (with potential loss of information) for external use. Annotations may be added to describe the published document. For example, the device or bandwidth for which the publication is destined. Annotations and alternative media assets may be removed to protect internal information or just reduce the size of the data destined for the user.

Once a document structure is published it is no longer part of the process set. All that can happen is that the publication itself can be distributed to the end-user. If republication needs to take place, then this needs to start from the document structure used as input to the process.

The input to the publication process is a docID (including the references to the compIDs and medIDs) and the output is a presID.

2.9 Distribute

The final process is the, synchronous or asynchronous, transmission of the presentation to the end-user. This can be through the internet, streamed or file-based; via a non-networked medium such as a CD-ROM or DVD; or projected for example during a film, performance or talk.

The input to the process is a docID plus appropriate software/hardware for displaying/playing the media assets. The output is the real-time display or projection of the media assets to the end-user.

While describing the processes we have tried to keep the descriptions as simple as possible. In a number of cases, the results of a process can feedback into a different process in a different role. We do not wish to exclude these "loops" in anyway, nor do we seek to present a complete list but rather give two illustrative examples. The document structure (docID) resulting from the organise process can be treated as a media asset (medID) and fed back into any of the processes that accept a medID as input. Similarly, a film script is the result of a long premeditation process in film whose semantics can only be captured with difficulty. The complete script, however, can be treated as a media asset. Similarly, an annotation can be treated as a "media asset", that is an object of an undefined data format that does not change.

3. DISCUSSION

Even though based on a small set of applications our investigation on the applicability of our canonical processes provided relevant insights for our hypothesis that real parity between tools for the various tools in multimedia production environments can only be achieved if we get a better understanding of the connection between processes.

The main outcome of our investigation is that, despite the large amount of instantiated annotations as well as media document structures during a production, our canonical processes can provide the means for easing the access to potentially relevant information. As the defined ID types are related to processes that in turn are associated with particular stages within a production workflow it is the IDs that facilitate, if known, the entry points to purpose driven exploration or manipulation of the material.

Another interesting finding is that the IDs not only work on this global level of workflows but also help to access more detail areas of content description, as being exemplified by the ontID, attID and annIDs. These provide a particular view on the material, without being able to clearly determining the full semantic meaning of the media asset. For that the associated annotation structure needs to be investigated. Their advantage is, however, that they act like a messenger in cell communication and thus trigger decisions about the path through the knowledge space that is generated in a production. All that works because there is a relation between production tasks and the complexity level of the annotation system, which the defined IDs exploit. Making those generally available will improve the compatibility across systems from different providers which supports the tendency of users to make use of tools that are crafted to give high quality support for the tasks they need to perform without losing the capability to access media assets from various sources.

The current stage of our investigation is just a first step towards this aimed for process fluidity. There are still a number of obstacles that need to be investigated.

In the processes described in the previous section we do not address the issue of manipulating a media asset in the form of changing the intrinsic content of it - as tools like Photoshop allow to do on images. The intention is rather that the descriptions be independent of data format and editing/authoring system. If an editing action on a media asset, medID, does take place then a new media asset is created with a new identity. The author and/or authoring system/editing suite may choose to select large numbers of the previously associated annotations and include these in a new compID. The problem here is, that some aspects of the established annotations up to that stage might still be valid and thus could be kept even for the new media asset. How this detection of static and changeable media semantics can be detected within annotations and hence be exploited remains to be investigated.

When we select media assets for archival, are we able to archive them in multiple archives and preserve the connection between the compIDs in different archives? A more preferred solution might be of having a single conceptual UID that is stored in a conceptual single archive, but is potentially available from different physical places. We do not go further into naming schemes here, and refer to the work on URIs, URLs and URNs [12] but are aware that this point needs further investigation..

In the processes as presented, we have note addressed the complications of being able to annotate annotations. This is of course what one would like to do, but it is out of scope of this paper. An annotation can, just as a (docID), be treated as a "media asset" - i.e. something of undefined data format that does not change and can have semantic annotations associated with it.

Within the processes specified we have assumed that the compID is an atomic entity. In the hypertext literature, composite components also exist. These however are more associated with the document structure rather than the underlying semantic annotations associated with a media asset. Hyperlinks similarly remain in the realm of the document structure and not in the underlying semantic relations. (See [11] where this distinction is explored.)

Outside the scope of the current discussion is the notion of interaction. For a film or a book there is no (system-processed) interaction. However, for an interactive media artwork or an informative hypermedia presentation (e.g. blood flow in the body) interaction is part of the expression of the message and needs to be incorporated in the presentation distributed to the end user. For example, in [7], the interaction is specified as Event-Condition-Action rules. Further work is needed to integrate the processes relating to interaction in the process model.

Related to the problem of interactivity is the problem of perception. If we assume here that a piece, be it an interactive art performance or a film, is consumed and interpreted over time, then we have to cope with ongoing circles of applying the canonical processes on the material, including the mix of it with other media sources. At the moment we have some vague ideas about the semantic side effects here - not only with respect to the naming of our processes but also on the flexibility that is demanded by combining content descriptions. Our current understanding is that the differences between compIds and messIDs as described during the post-production phase of film production will play an essential role here. We assume that a large amount of additional annotation on an interpretation level will favour dominant relations of messID types where occasionally compID references need to be established to specify particular parts of an audio-visual product. Moreover, at this stage of interpretation, as well as for reuse processes we see the need for including additional information structures of relations, a point that is not at all addressed in this paper.

4. CONCLUSION

We see this paper plus the accompanying process descriptions as an initial step towards the definition of data structures which can be used and accessed by different members of the multimedia community. In this paper we identify processes that we see as being fundamental to different applications of multimedia and wish to discuss them further during the workshop.

Acknowledgments

This paper and workshop were inspired by Dagstuhl meeting 05091 "Multimedia Research - where do we need to go tomorrow" organised by Susanne Boll, Ramesh Jain, Tat-Seng Chua and Navenka Dimitrova.

Parts of this research were funded by the European ITEA Passepartout and Dutch national BSIK MultimediaN e-Culture projects.

5. REFERENCES

[1] B. P. Bailey, J. A. Konstan, and J. V. Carlis. Supporting Multimedia Designers: Towards More Effective Design Tools. In *Proc. Multimedia Modeling: Modeling Mutlimedia Information and Systems (MMM2001)*, pages 267–286. Centrum voor Wiskunde en Informatica (CWI), 2001.

[2] M. Davis. Active Capture: Integrating Human-Computer Interaction and Computer Vision/Audition to Automate Media Capture. In *ICME '03, Proceedings*, pages 185–188, July 2003.

[3] C. Dorai and S. Venkatesh. Bridging the Semantic Gap in Content Management Systems - Computational Media Aesthetics. In C. Dorai and S. Venkatesh, editors, *Media computing - Computational media aesthetics*, pages 1–9. Kluwer Academic Publishers, June 2002.

[4] F. Halasz and M. Schwartz. The Dexter Hypertext Reference Model. *Communications of the ACM*, 37(2):30–39, February 1994. Edited by K. Grønbæck and R. Trigg.

[5] L. Hardman. *Modelling and Authoring Hypermedia Documents*. PhD thesis, University of Amsterdam, 1998. ISBN: 90-74795-93-5, also available at http://www.cwi.nl/∼lynda/thesis/.

[6] R. Jain. Experiential Computing. *Communications of the ACM*, 46(7):48–55, July 2003.

[7] B. Kerhervé, A. Ouali, and P. Landon. Design and Production of New Media Artworks. In *Proceedings of the ACM Workshop on Multimedia for Human Communication - From Capture to Convey (MHC 05)*, November 2005.

[8] H. Kosch, L. Böszörményi, M. Döller, M. Libsie, P. Schojer, and A. Kofler. The Life Cycle of Multimedia Metadata. *IEEE Multimedia*, (January-March):80–86, 2005.

[9] F. Nack. Capture and Transfer of Metadata During Video Production. In *Proceedings of the ACM Workshop on Multimedia for Human Communication - From Capture to Convey (MHC 05)*, November 2005.

[10] F. Nack and W. Putz. Designing Annotation Before It's Needed. In *Proceedings of the 9th ACM International Conference on Multimedia*, pages 251–260, Ottawa, Ontario, Canada, September 30 - October 5, 2001.

[11] L. Rutledge, J. van Ossenbruggen, L. Hardman, and D. C. Bulterman. Structural Distinctions Between Hypermedia Storage and Presentation. In *Proceedings of ACM Multimedia*, pages 145–150. ACM Press, November 1998.

[12] URI Planning Interest Group, W3C/IETF. URIs, URLs, and URNs: Clarifications and Recommendations 1.0. W3C Note for discussion only. Available at http://www.w3.org/TR, 21 September 2001. Report from the joint W3C/IETF URI Planning Interest Group.

[13] W3C. Synchronized Multimedia Integration Language (SMIL 2.0) Specification. W3C Recommendation, August 7, 2001. Edited by Aaron Cohen.

[14] W3C. XML Schema Part 2: Datatypes. W3C Recommendation, May 2, 2001. Edited by Paul V. Biron and Ashok Malhotra.