Principles of Video Annotation Markup Language (VAML)

Tina T. Zhou and Jesse S. Jin

School of Information Technologies
University Of Sydney, NSW 2006, Australia

{tzhou, jesse}@it.usyd.edu.au

Abstract

The creation of hypertext and World Wide Web provided a powerful mechanism for organizing and distributing the warehouse of information. However, video-based hypermedia may represent the future of the Internet since it provides a much better viewing experience. Our ongoing project - the Video Annotation Markup Language (VAML) aims to enhance the hypervideo framework. In this paper, we give a detailed description of VAML and its principles.

Keywords: HyperVideo, HyperVideo Authoring, Video Annotation.

1 Introduction

We are seeing that the volume of digital video information is growing drastically with the advances of video technologies. People like watching video because the video is much better than text in the way of enhancing users' viewing experience. However, when people are watching video, interaction between video and people seldom happens except the actions of stoping, pausing, back warding, forwarding or slowing down. In the real world, the author of a video may like to embed more related information in the video and allow viewers to browse this information without shattering the integrity of the video. To achieve this, the concept of hypervideo is introduced. With hypervideo, viewers are now able to choose the information by navigating within the video or even among more videos through indicated links.

Hypervideo acts like hypertext. It offers its viewers a path to follow the narrative moments that determine what lies ahead and explain what came before. Unlike a web page, in which static and simultaneous links are within the same space, hypervideo-linking opportunities come and go as the video sequences play out in time. Like static hypertext, more than one opportunity can occur at a time – several clips can play at once, or parts of the video frame itself can contain more than one active link.

Many multimedia-authoring tools have exploited the hypervideo concept. We will give a brief description about these tools in Section 2. However, current systems either use programming script or text file that contains the specification about when and where the link opportunities occur to implement the linking structure. To the best of our knowledge, none of them changed the raw video stream.

Our ongoing project VAML is designed to have tags embedded in the raw video data stream and work in a similar manner as a HTML document does. Structural elements such as scenes, shots, and objects are identified in VAML video. VAML established a general structure model that describes the relationships between structural elements. With the respect to the video structure model, VAML renders the elements with certain effects, such as colour, brightness, fading, 3D animation, and even more complicated ones, for example, the user may be able to rotate the actor's head in a movie video, or segment a video stream into foreground (moving object) and background. However, the essential feature of VAML would be the concept of computer-supported links, including temporal links, spatio-temporal links and text-based links. It is this linking capability that allows a non-sequential organization and presentation of video. Physically allocating tags inside of video data stream to refer the certain point of video allows jumping around in the video, and the development of digital watermarking technology makes this become possible by inserting addition information such as structure definition, description, and machine-processable features (i.e., links) to a raw video data stream without affecting the integrity of video. Liu [Liu01] addressed various design issues and implementation issues of VAML application. In this paper, we give more specific information on VAML.

The rest of this paper is organised as follows. Section 2 briefly describes the development of hypervideo and technologies exploited in VAML. Section 3 presents the detailed design of VAML. In Section 4, we outline the implementation of VAML video. Section 5 concludes this paper with a discussion of our future research directions.

2 RELATED WORK

In early 1990s, hypervideo applications such as InterVideo [Kahn91] and Elastic Charles [Br∅ndmo91] were based on analogue video and laser disc technology. In these systems, two video segments could not be presented simultaneously on the same computer screen. Along with the development of digital technology, solving this problem became possible as showed in Interactive Kon-Tiki Museum [Liest∅l94]. The Interactive Kon-Tiki Museum achieved continuous integration in linking from video to text and video to video. However, the links represented by buttons did not appear in the video frame.

HyperCafe [Sawhney96] [Sawhney97] developed by Sawhney et al. illustrated the general concept of hypervideo framework and offered to its users and authors

Copyright 2004, Australian Computer Society, Inc. This paper appeared at the Pan-Sydney Area Workshop on Visual Information Processing (VIP2003), Sydney. Conferences in Research and Practice in Information Technology, Vol. 36. M. Piccardi, T. Hintz, X. He, M. L. Huang, D. D. Feng, J. Jin, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

the richness of multiple narratives. It implemented a playback tool called Hypervideo Engine in Macromedia Director's Lingo programming language. Hypervideo Engine provided a high-level scripting interface for authoring hypervideo narratives. The scripts specified the spatial and temporal placement of hypertext and video clips on the screen along with the linking opportunities. Other multimedia-authoring tools such as HyperSoap [Dakss98], Hyper-Film [hyperfilm00] [Tua02], and Flash, also implemented the hypervideo manner in different scripting language. However, the learning process of using these tools is very complicated and the structure of these final products is also hard to tell [Sawhney 97].

The World Wide Web Consortium (W3C) developed the Synchronized Multimedia Integration Language (SMIL) [W3C98] [Hoschka98] to allow the use of text editor to write multimedia presentation. SMIL is a HTML-like language. The syntax of SMIL conforms to the XML standard. SMIL can be used for choreographing multimedia presentations where audio, video, text and graphics are combined in real-time [W3C03]. With SMIL, the presentation designer can indicate the spatial layout and temporal relationship of media, and where and when the objects are shown. SMIL also defines a set of navigation constructs to support the functionality of HTML-style hyperlinks. However, hyperlinks between videos are in a synchronized manner where the linked video and the linking video are playing in the same time in the SMIL player. Since the action with the video is along with the time line, author cannot define a certain point in a video to let viewers directly go to that point from the current viewing point. The video has to be either continuing playing or simply stop.

While SMIL is organizing its multimedia presentation along with the time line, other applications try to seek solutions in video's spatio-temporal domain. Bertolino's hypervideo [Bertolino98] is one of typical examples. In Bertolino's hypervideo system, the video is decomposed into shots, and then objects are extracted and tracked within each shot. Thus, linking occurrences can be associated with objects among the shots. HyperSoap we mentioned before also exploits the object tracking technique.

As we discussed in Section 1, we are trying to develop a new method to simplify the learning process. We exploit SGML and digital watermarking technologies.

2.1 SGML

The Standard Generalized Markup Language (SGML), is an international text processing standard (ISO 8897) proposed by the International Standards Organization (ISO) in the early 1980s [Connolly95]. It was designed as a means for managing information and increasing the portability documents among computers and text processing systems.

One of the tenets of SGML is the separation of document content from its rendering. The division is achieved with markup, a series of instructions, embedded in the text of the document to provide the system with necessary processing rules for interpretation during presentation. Procedural markup is used to give the necessary rules for rendering document text – how the text should appear on the page. Descriptive markup defines the purpose of the text in the document. A Document is broken into elements that represent object semantics within the system. Tags are used to define the logical elements of the document. Associated attributes of tags also identify the arbitrary types of elements. Elements can be nested within other elements to define the organization of the document. Document Type Definition (DTD), establishes the document structure. It provides a framework for the types of elements that constitute a document. It also defines the hierarchical relationships between elements and sets the context rules ensuring a consistent and logical structure of the document. Tags have to strictly respect the set of context rules defined in the DTD.

In summary, SGML enables the information sharing amongst users, applications and information maintenance in storage. We apply SGML as a format to represent the complex video structure, identify objects, and express hyperlinks or other relationships among the identified object. This is actually a video annotation process.

2.2 Digital Watermarking

Digital watermarking is a technique by which imperceptible digital code is embedded in media content such as image, video, and audio. It delivers a solution for protecting digital image, audio and video assets.

Digital watermarking technology enables users to embed invisible and inaudible messages, called watermarks into still image, video data stream and audio data. The watermark does not increase the size of the data file. It should not be affected by format conversions, compression, or signal processing as well. The detected messages are used to prevent unauthorized use, detecting illegal copies or modified locations.

We consider the digital watermark as a device of unbreakable connection of video content with SGML markups. Whenever the video content is transmitted or converted into a different format, the watermark continues to exist and remains unchanged. So if the watermark data include an identifier of the SGML markup, the SGML markup created for the original video content will never be orphan.

3 VAML

Based on SGML, we define a VAML video as consisting of its elements, markup, attributes and entities. Elements refer to the content of video such as shots, frames, objects, hyperlinks etc. The combination of elements forms the normal video stream. Markup tags are extra information inserted into the video stream to define the meaning and context of video, thus, the applications that process the VAML video would identify scenes, shots, and objects of the video precisely and quickly. There are four types of markup in VAML. We give the detail explanation of these markups in Section 3.1. Attributes are the properties associated with markup. The detail of attributes will be given in section 4.2. Entity refers to the unit of virtual information storage that contains part of document. An entity can be referenced from one or more places in a VAML video, thereby causing its information to be included in the video at the points of reference. Entity could be an image, a text, or any other media types.

However, the VAML video we defined above, would not work on certain system without giving the notification to that system about its markup tags used. Therefore, a VAML video will has the following three parts.

1. **VAML Declaration**. This is a header file that contains the system specific information needed to run the VAML video on the target system. It specifies which character set will be used, which code will be used as VAML delimiter, and how long the names of the generic identifiers can be. Normally the VAML declaration will be held in the form of compiled tables by the VAML processor and thus be invisible to the user.
2. **Document Type Definition (DTD).** This is a set of rules defining the structure of a VAML document and listing all permissible elements.
3. **Instance**. This is the actual video and its accompanying tags conform to the specifications and restrictions set forth in the DTD.

3.1 Markup and Tag

Markup is not a part of the intellectual content of a video, but instead provides information about how the video is structured and how it should be interpreted or presented. In VAML, we designed four types of markup:

- Media Descriptive Markup
- Structure Presentational Markup
- Render Procedural Markup
- Referential Markup

Media Descriptive Markup provides general information about the entire video, including the title of the video, the brief introduction of the video, the author of the video, the video copyright, the recommendatory frame rate, the video file type, etc. These general descriptions are nested within the HEAD part of the VAML. TITLE, AUTHOR, INTRO, COPYRIGHT, FRAMERATE, FILETYPE, DATE, and CATEGORY are used to embody these descriptions.

Structure Presentational Markup has two sub-types. One is Document Structure Presentational Markup. The other one is Video Structure Presentational Markup. Document Structure Presentational Markup intends to clarify the organization of the VAML document. It includes HEAD and BODY two tags. The entire Media Descriptive Markup set is nested in HEAD and HEAD contains no other markups. BODY can contain other types of markup but no Media Descriptive Markup. Video Structure Presentational Markup specifies video breaks, video transitions and even the designation of the structure of the video. DOC, SCENE, SHOT, and OBJECT are Video Structure Presentational Markups. They construct the hierarchical structure of video and can be refer to the structure element of the video. The hierarchical relationship between these tags is showed in Figure 1.

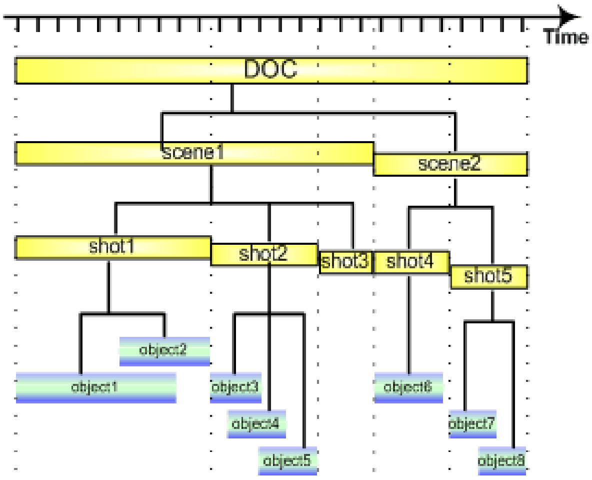
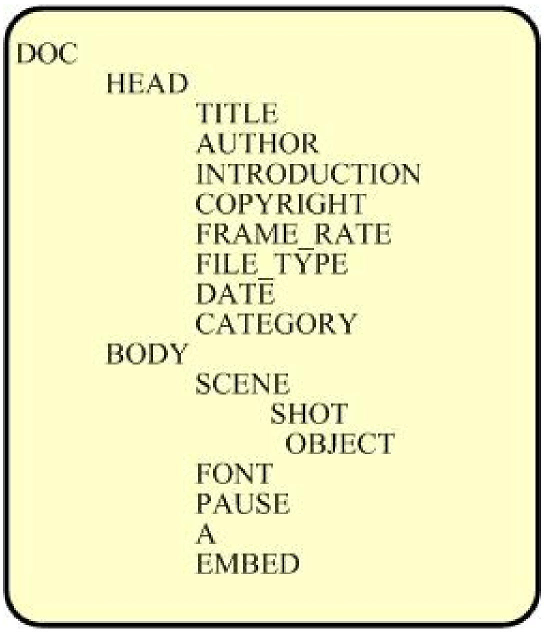


Figure 1: Hierarchical Structure of VAML Document

Render Procedural Markup consists of commands indicating how video should be formatted. It generally deals with the features of a particular video element (scene, shot, or object). This includes editing the colour of shot or object, segmenting the front moving object from the background, and so on. The FONT tag is one of important procedural markups in VAML. It performs the tasks we mentioned above. Procedural markup in VAML also makes schedule for video playing. For example, the PAUSE tag can suspend the play of a video at a certain point for certain amount of time.

Referential Markup refers to entities that are external to the document and is replaced by these entities during processing. For example, EMBED is used for this kind of markup to embed images, text, video or other media with various media types. The A markup refers to the links between documents (out-links) and within documents (in-links).

Figure 2 shows all the markups in VAML. The indentation in the figure indicates the nest relationship between markups. We will discuss each markup in detail in Section 3.2.



```
DOC
      HEAD
            TITLE
            AUTHOR
            INTRODUCTION
            COPYRIGHT
            FRAME_RATE
            FILE_TYPE
            DATE
            CATEGORY
      BODY
            SCENE
                 SHOT
                  OBJECT
            FONT
            PAUSE
            A
            EMBED
```

Figure 2: VAML Markups

3.2 ATTRIBUTES

Attributes are the properties associated with markups. Its value associated with markup describes the elements but are not part of elements. For example, the ID attribute of a shot provides a label for shot so that the shot can be referenced explicitly from any place in the video. Figure3 shows the detail of all the attributes associated with the respective markup in VAML.

Tag	Attribute	Val Value
A	HREF	*URL
	NAME	*text
	TARGET	_blank, _self
AUTHOR		*text
BODY	WIDTH	*amount
	HEIGHT	*amount
CATEGORY		*text
COPYRIGHT		*text
DATE		*date
DOC		
EMBED	SRC	*url
	TYPE	_text, _image, _video…
	WIDTH	*amount
	HEIGHT	*amount
	SHAPE	_triangle, _circle, …
	BORDER	*amount
	TRANSPARENT	_yes, _no
	NAME	*text
	ALIGN	_top, _mid, _left, _right
	ALT	*text
FONT	COLOR	*color_code
	ACTION_TYPE	_fadeOut, _loop,
	TIME	*amount
	BACKGROUND	*color_code
FILETYPE		_mpg, _avi, …
FRAMERATE		*number
HEAD		
INTRO		*text
OBJECT	ID	*text
	NAME	*text
	DESCRIPTION	*text
	TYPE	_motion, _static
	START_FRAME_NO	*amount
	END_FRAME_NO	*amount
	MOTION_DSCR	*text
	POSITION	_rectangle, _triangle,…
PAUSE	TIME	*amount
SHOT	ID	*text
	NAME	*text
	KEYFRAME_NO	*amount
	START_FRAME_NO	*amount
	END_FRAME_NO	*amount

Figure 3: Markups and its associated attributes of VAML

4 VAML VIDEO

In the design of VAML we could integrate an automatic segmentation tool so that shot, object detection is automatically performed. Once the video elements have been recognized, the structure presentational markups could be automatically inserted into the video stream by using watermarking technology. Users may add other markups according to their needing. Figure 4 shows the whole process of VAML video creation.

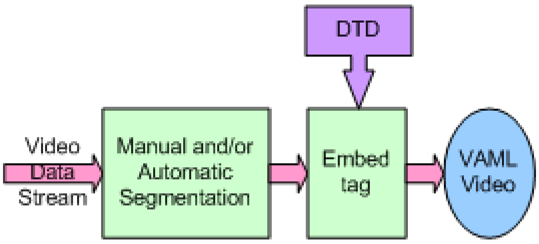


Figure 4: Data Flow in VAML Video Creation Process

To play the VAML video, a VAML video player is designed. The player first parses the VAML video and then renders video element in the video play area. Figure 5 shows this process.

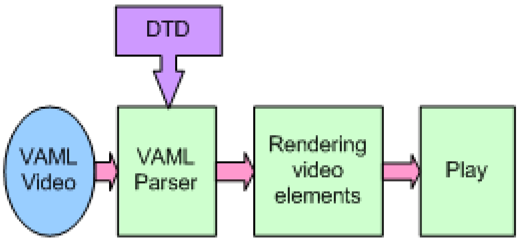


Figure 5: Data Flow in VAML Video Play Process

5 CONCLUSION

This paper describes the principles of VAML system, an easy and efficient way for constructing hypervideo. Comparing to traditional hypervideo authoring tools, VAML simplifies the learning process since it uses SGML as a basis for the structure metadata model. In operation VAML interprets the input DTDs to generate a view on video structure and then embed the tags into the video data to create a VAML video stream. The VAML player firstly extracts the VAML video elements according to the input DTDs, and then renders the elements in play screen.

The current VAML prototype points to several directions for future work. Firstly, in the existing video data structure we defined in Section 4, the same object appearing different shots need to be distinguished as different objects, for example, object 2, 4 and 6 may be the same object with a slight different motion effect. With current VAML definition, this objects need to be defined three times. This is not efficient. A more efficient data structure model needs to be conducted to cope with this issue in the future. Secondly, the OBJECT markup defined in current VAML only embodies the general description of object. An object annotation system may be needed to describe more precisely the spatial and temporal arrangements of object. Finally, VAML focuses on the presentation of video structure and the rendering of video structure elements. To achieve real data interoperability on the web, the precise semantic meaning of the video data need to be extracted. The VAML parser may be extended to have a function that is able to translate the video structure elements to search engine understandable form according to video's domain knowledge. In this manner, a single VAML parser tool is appropriate for metadata of news, educational materials, entertainment, or other videos in variant domain.

6 REFERENCE

[Bertolino98] Bertolino, P., Mohr, R., Schmid, C., Bouthemy, P., Gelgon, M., Spindler, F., Benayoun, S., Bernard, H. and Recherche, A. (1998): Building and Using Hypervideos. In *Proceedings of the Fourth IEEE workshop on Applications of Computer Vision*, **WZCV'98**: 276-277.

[Br∅ndmo91] Br∅ndmo, P. H. and Davenport G. (1991): Creating and Viewing the Elastic Charles – A Hypermedia Journal. In *Hypertext: State of the Art*, Aleese, R. and Green, C., eds., Intellect, Oxford, U.K., pp. 43-51.

[Connolly95] D. Connolly (1995): Overview of SGML Resources. *World Wide Web Consortium*, http://www.w3.org/MarkUp/SGML/.

[Dakss98] Dakss, J., Agamanolis, S., Chalom E. and Bove, V. M. (1998): Hyperlinked Video. *Proceedings SPIE Multimedia Systems and Applications*, **2528**.

[Hoschka98] Hoschka, P. (1998): An Introduction to the Synchronized Multimedia Integration Language. In *IEEE Multimedia*, **5**(4):84-88.

[hyperfilm00] http://www.hyperfilm.it.

[Kahn91] Kahn, P. and Haan, B. J. (1991): Video in Hypermedia: The Design of InterVideo. In *Visual Resources*, **VII**:353-360.

[Liest∅l94] Liest∅l, G. (1994): Aesthetic and Rhetorical Aspects of Linking Video in Hypermedia. In *Proc. Hypertext 94, ACM Press*, New York, **Hypertext94**:217–223.

[Liu01] Liu, C. and Jin, J. S. (2001): Modelling and Design of VAML. *Conferences in Research and Practice in Information Tchnology - Selected Papers from the Pan-Sydney Area Workshop on Visual Information Processing*, **11**:151 -152.

[Sawhney97] Sawhney, N., Balcom, D. and Smith, I. (1997): Authoring and navigating Video in Space and Time. In *IEEE MultiMedia*, **4**(4):30-39.

[Sawhney96] Sawhney, N., Balcom, D. and Smith, I. (1996): Hypercafe: Narrative and Aesthetic Properties of Hypervideo. In *Proc, Hypertext 96, ACM*, **Hypertext96**:1-10. Also see the HyperCafe Web site: http://www.Icc.gatech.edu/gallery/hypercafe.

[Tua02] Tua, R. (2002): From Hyper-Film To Hyper-Web: The Challenging Continuation Of A European Project. In *EVA 2002 Proceeding*, Pitagora Ed., Bologna.

[W3C98] Hoschka, P. ed. (1998): Synchronized Multimedia Integration Language (SMIL) 1.0 Specificatio. W3C Recommendation, http://www.w3.org/TR/REC-smil.

[W3C03] Synchronized Multimedia Activity Statement. World Wide Web Consortium, 2003, http://www.w3.org/AudioVideo/Activity.html.

[Yankelovich88] Yankelovich, N., Haan, B. J., Meyrowitz, N. and S. M. Drucker, S. M. (1988): Intermedia: The Concept and Construction of a Seamless Information Enviornment. In *IEEE Computer*, **21**(1): 81 – 96.