Lloyd Rutledge, Jacco van Ossenbruggen, Lynda Hardman, and Dick C.A. Bulterman, "Cooperative Use of MHEG-5 and HyTime", Proceedings of Hypertexts and Hypermedia: Products, Tools, Methods (H^2PTM'97), *September 1997*

Cooperative Use of MHEG-5 and HyTime

Lloyd Rutledge, Jacco van Ossenbruggen*, Lynda Hardman and Dick C.A. Bulterman

CWI
P.O. Box 94079
1090 GB Amsterdam, The Netherlands
{lloyd,lynda,dcab}@cwi.nl

**Vrije Universiteit*
Dept. of Math. and Computer Science
De Boelelaan 1081
1081 Amsterdam, The Netherlands
jrvosse@cs.vu.nl

ABSTRACT. The standards MHEG-5 and HyTime are interchange formats for hypermedia information. While they may seem to compete, they actually play separate and complementary roles in a complete and open hypermedia environment. MHEG-5 is used for portable final-form hypermedia presentations. HyTime is used for the long-term, presentation-independent storage of hypermedia documents. Given these tasks, MHEG-5 can be used to encode presentations of HyTime documents. This paper explores these two standards, the cooperative roles they play and their application to the CMIF hypermedia environment architecture. The issues discussed include the semantic overlap between the hypermedia models each standard represents and how the use of each standard affects the cooperative use of the other.

KEYWORDS. MHEG-5, HyTime, DSSSL, CMIF, Hypermedia Authoring

1. Introduction

MHEG-5 and HyTime are both standards for hypermedia documents, but each functions on a different layer of document processing methodology. MHEG-5 is a standard for representing portable hypermedia presentations. HyTime is a standard for representing the hypermedia structure of documents independently of any presentation. It provides an encoding format, a general hypermedia model and facilities for transformations to other formats and to variations in document presentation. MHEG-5 is one potential output format for such transformations of HyTime documents.

Our research group has investigated the issues involved with hypermedia authoring and presentation with the development of the *CWI Multimedia Interchange Format (CMIF)* and the *CMIFed* environment for the creation and playback of CMIF hypermedia documents [HAR 94]. CMIF encodes crucial hypermedia functions such as synchronization, hyperlinking, screen display layout and presentation style. Our main goal for CMIF is that it be applicable to a wide variety of presentation situations. To facilitate this adaptability, we have developed an encoding for CMIF documents in HyTime.

In this paper we discuss the application of both HyTime and MHEG-5 to hypermedia processing. We first provide background material on HyTime, MHEG-5 and other formats and models for hypermedia. Second, we present a general model for hypermedia systems with roles that are potentially played by HyTime, MHEG-5 and other formats. Then we discuss correlations between constructs in MHEG-5 and HyTime and the effect the different roles played by MHEG-5 and HyTime have on these correlations. These relationships are illustrated with the CMIF format, its use of HyTime constructs and its potential presentation with MHEG-5. Then the potential role DSSSL has is specifying the mapping between HyTime presentation-independent structure and their presentation with MHEG-5 is discussed. Finally, the application of MHEG-5 and HyTime to the CMIF hypermedia environment is described.

2. Standards for Hypermedia

The *Multimedia and Hypermedia Information Encoding Experts Group (MHEG)* is the ISO effort for standardizing the coding of multimedia and hypermedia information. MHEG's *Part 5: Support for base-level interactive applications (MHEG-5)* [ISO 97] encodes portable final-form hypermedia presentations transferred over a distributed network. It assumes minimal capabilities for the user interface system, ensuring that MHEG-5 presentations can be processed for a wide variety of front ends. Presentations are transferred over the network on an as-needed basis. Each step of the presentation, as defined by the user interaction, is fully encapsulated and communicated separately, minimizing network traffic and the transfer of unnecessary information. MHEG-5 is defined as a collection of related object-oriented data structures.

Hypermedia/Time-based Structuring Language (HyTime) [ISO 92] builds upon and extends into hypermedia the presentation-independent text structuring of *Standard Generalized Markup Language (SGML)* [ISO 85]. SGML is a widely used ISO standard for representing electronic text documents. It supports and facilitates document exchange, reuse and longevity by representing a document in its permanently stored form without defining its processing into any particular presentation. SGML also provides for the definition of separate document sets, called *Document Type Definitions (DTDs)*, which represent different types of documents used by separate communities. A DTD defines a structure within which a class of documents fits. An individual SGML document is an instance of the DTD associated

with it. SGML documents consist of *elements*, which can contain direct text or other elements. Each element has a collection of *attributes*. Two types of attributes, the *unique identifier (ID)* and *unique identifier reference (IDREF)* enable one element to refer to another. A DTD defines a set of element types, which say what type names elements of a document can have and what attributes each element type can have.

HyTime's additional structure includes complex hyperlinking, locating of document objects and the scheduling of objects within measured coordinate systems such as space and time. HyTime inherits SGML's ability to define distinct document sets. HyTime, like SGML, requires external mechanisms for specifying the presentation of its document. HyTime's extension beyond SGML is defined as an *SGML architecture*. An architecture defines a collection of *architectural forms*, each of which is a collection of attributes that an element can have. This specification is more flexible than the DTD specification because elements of each form can have additional attributes and can have any element type name. Multiple DTDs can be specified for documents that conform to a given architecture.

Document Style Semantics and Specification Language (DSSSL) [ISO 96] is a standard that encodes the translation of SGML, and thus HyTime, documents into other forms of output. DSSSL defines a system independent way to describe the processing of SGML documents of a given DTD into those of another DTD or into various forms of page-based output. DSSSL specifies such mappings with a powerful document query language, a document transformation language and a style language to apply a set of formatting characteristics to portions of a document. Such a DSSSL mapping is encoded in a *style sheet*, which is applied to a document or document set to specify its mapping into one format or one presentation style.

3. A Layered Hypermedia Model

In this section we describe a general model for hypermedia processing based primarily on the Dexter and AHM models but also taking into account the processing model used for SGML, HyTime and some related formats. The Dexter/AHM model and its relation to these SGML-based formats is illustrated in Figure 1. Our Berlage document architecture extension to HyTime, which is included in this diagram, is described in Section 3.1.

The *Dexter Hypertext Reference Model* [HAL 94] is a model representing the common abstractions of different hypertext systems. Dexter specifies significant abstractions shared by typical hypertext systems to provide a common reference for discussing and comparing these systems. It is designed to aid in the general discussion of different hypermedia documents and applications. It defines a layered model onto which individual hypertext systems can be mapped. It also defines commons terms to be uniformly applied to the hypertext systems when they are described by Dexter. We have extended Dexter from hypertext into hypermedia with the development of the *Amsterdam Hypermedia Model (AHM)* [HAR 94]. AHM extends the Dexter model into multimedia by adding to it concepts such as the

synchronization of multimedia object display and more complex specifications of how link activation affects document presentation.

Dexter/AHM divides hypertext processing into three layers: the *run-time*, *storage* and *within-component* layers. The run-time layer is concerned with the final presentation of documents, including how the user interacts with the interface and what activities occur during presentation. The within-component layer contains the media objects used within a document. This layer represents the individual items contained in the document. The storage layer contains the atomic components making up a document plus structures grouping and relating these components. It is related to the within-component layer through *anchoring*, which enables media type-independent linking to portions of atomic media content. The run-time and within-component layers of the Dexter/AHM model are covered minimally in this treatment, with the focus being on the remaining components of the model.

In SGML and HyTime terminology, a document is separated into *content*, *structure and presentation*. The *content* makes up the individual perceivable components of the document. The SGML and HyTime constructs that locate files stored on a system fit in the within-component layer. These constructs include SGML entities and the HyTime location address module. The *presentation* is how the document appears to the user. SGML focuses on document *structure*, which specifies the relations between the content components that affect their potential final presentations without specifying any particular means of presenting them.

The Dexter/AHM storage layer provides the hypermedia structure of a document. This includes hyperlinking and the composition of document objects from atomic media components. The HyTime hyperlinks module establishes non-hierarchical and non-linear relationships between document components. The structures defined by this module correspond very closely with links in the Dexter/AHM storage layer.

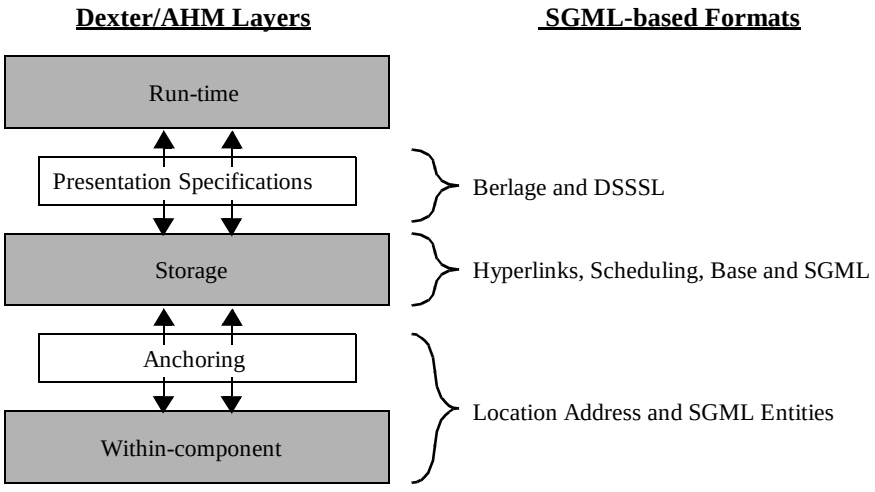

Figure 1. *The Dexter/AHM Layers and Corresponding SGML-based Formats*

SGML and the HyTime base module define several types of composites. We refer to the information of this layer as the *generic hypermedia structure* of the document.

A document's generic hypermedia structure cannot by itself be presented to a user. This structure must accompany a mapping that specifies how that structure is translated into a particular presentation. This mapping references aspects of the document's generic hypermedia structure to determine the document's appearance to the user. The mapping may also reference aspects of the document's presentation environment to determine the best means of displaying the document under the conditions given. This functionality corresponds with the Dexter/AHM *presentation specification* layer, which defines how the contents of the storage layer are to be processed when the user browses through them. DSSSL performs actions of this layer because it specifies how document structure is to be translated into a presentation under particular circumstances. HyTime does not address presentation-specification because no HyTime constructs refer to presentation-time document characteristics. An extension to HyTime for representing presentation specification information appropriate for storage in the document itself is described in Section 3.1.

The presentation of a document in this model is the result of processing its generic structure with its presentation specifications. A document's presentation falls within the run-time layer of Dexter/AHM. MHEG, being a presentation specific format, falls within this layer.

3.1. *The Berlage SGML Architecture*

HyTime contains no constructs for specifying structure that fall in the presentation specification layer of document modeling. In writing a HyTime DTD for CMIF, we encountered some presentation specification concepts that were broad enough to apply to many hypermedia document sets. We defined and SGML architecture to represent these presentation specification concepts, the same mechanism with which HyTime was defined to represent generic hypermedia structuring concepts. We call this SGML architecture and the forms it defines the *Berlage architecture*, named after the Dutch architect of several important buildings in Amsterdam. As architectural forms, these presentation specification constructs can be used in DTDs for other HyTime document sets. These new forms are defined with HyTime form attributes and thus elements conforming to these presentation specification forms also conform to their HyTime forms and will be recognized as such by HyTime engines, even if the engine does not recognize Berlage forms. The Berlage architecture extends HyTime, rather than overriding it. Berlage documents are also HyTime-conforming documents. Further, semantics deemed appropriate for HyTime representation are encoded with HyTime constructs in the Berlage architecture. What Berlage provides are some presentation-specification layer constructs that style sheets can refer to in addition to HyTime generic structure constructs in determining a presentation.

4. Potential Relationships Between MHEG-5 and HyTime

The relationship between HyTime and MHEG has been explored in previous work. Gopal and Price [GOP 95] have described this difference, stating the following:

- HyTime makes no assumption about its user, while MHEG assumes a particular model of interaction.
- HyTime link defining is general and flexible, while MHEG links are basic and in their final form.
- A HyTime engine processes an entire document, while MHEG processes only the portion of the document presented.

Early in the development of both standards a comparison between them is provided by Markey [MAR 91]. This work first suggests the two standards are complementary rather than competing and proposes MHEG as a format for presented documents whose long-term storage format is HyTime.

In this section we discuss potential relationships between MHEG-5 and HyTime constructs. CMIF and its use of HyTime and MHEG-5 are used as examples to illustrate these relationships. The corresponding constructs from the three formats are shown in Table 1. The DTD for CMIF documents and its use of HyTime forms is illustrated in Figure 2. We divide hypermedia into separate areas and discuss the constructs from each area in turn. These areas are *content*, *hyperlinking*, *composition*, *measured positioning* and *appearance*. For each area, we describe the MHEG-5, HyTime and CMIF constructs involved. We also discuss how these constructs address either the presentation-independent or presentation-specific issues of their use within their format.

4.1. *Content*

In a single hypermedia document, much of the media content shown in any presentation of the document is referenced as external files. However, some of content can be stored directly in the document file itself. This is particularly true for text. In an MHEG-5 document, a *Text* class holds text that is to be displayed on the screen. An integral part of the SGML format is this direct containment of text data in the document itself. Every element in an SGML document can contain other elements or straight text. HyTime inherits this characteristic. The most direct way to incorporate text into an SGML or HyTime document is in this fashion. In the CMIF DTD, the *immediate* element type is used to include text directly in the document file. Its HyTime encoding includes the text as direct content. The generation of an MHEG-5 application from a CMIF document using an immediate element to hold text would likely result in such text being placed in a Text class instance.

Another MHEG-5 class that suggests content semantics is *OpenConnection*. It retrieves the contents of an external file for presentation as a media object. The source document information from which the calling of OpenConnection would result is the specification of a file whose contents are considered a media object

component of the document. It would be called at the point at which the user's interaction or the timing of the application's presentation makes it appropriate for the media object contained by that file to be displayed. For example, an image may be displayed 5 seconds into a presentation. When 5 seconds have passed, the OpenConnection action retrieving that image would be called.

The primary SGML construct for specifying files is the *SGML entity*. An entity is a string which can be processed by a file server to return a file's contents. This string corresponds with the *address* field of OpenConnection. The *external* element type of the CMIF HyTime encoding uses an SGML entity to specify a file on the local system as a media object.

	MHEG-5	**SGML/HyTime**	**CMIF DTD**
Content	Text	Element Content	Immediate
	OpenConnection	Entity	External
	address	Entity, Notation Location	External, URL notation
	protocol	Notation	URL notation
Hyperlinking	Link	Hyperlink	Hyperlink
	LinkCondition	Starting Anchor	Starting Anchor
	LinkEffect	Ending Anchor	Ending Anchor
	HyperText	Anchor	TextAnchor
	SwitchButton, Pushbutton	Starting Anchor	none
Composition	Application	HyTime Document	CMIF
	Scene	Event Group	Composite
	TokenGroup	SGML hierarchy, Event Group	Channel, Sequential, Parallel
Measured Placement	implicit coordinates of display	Finite Coordinate Space	explicit coordinates of display
	Visible	Event	Event
	OriginalBoxSize, OriginalPosition	Extent List	ScreenExtent
	Timer	Marker	TimeBound

Table 1: Correspondence MHEG-5, HyTime and CMIF Constructs

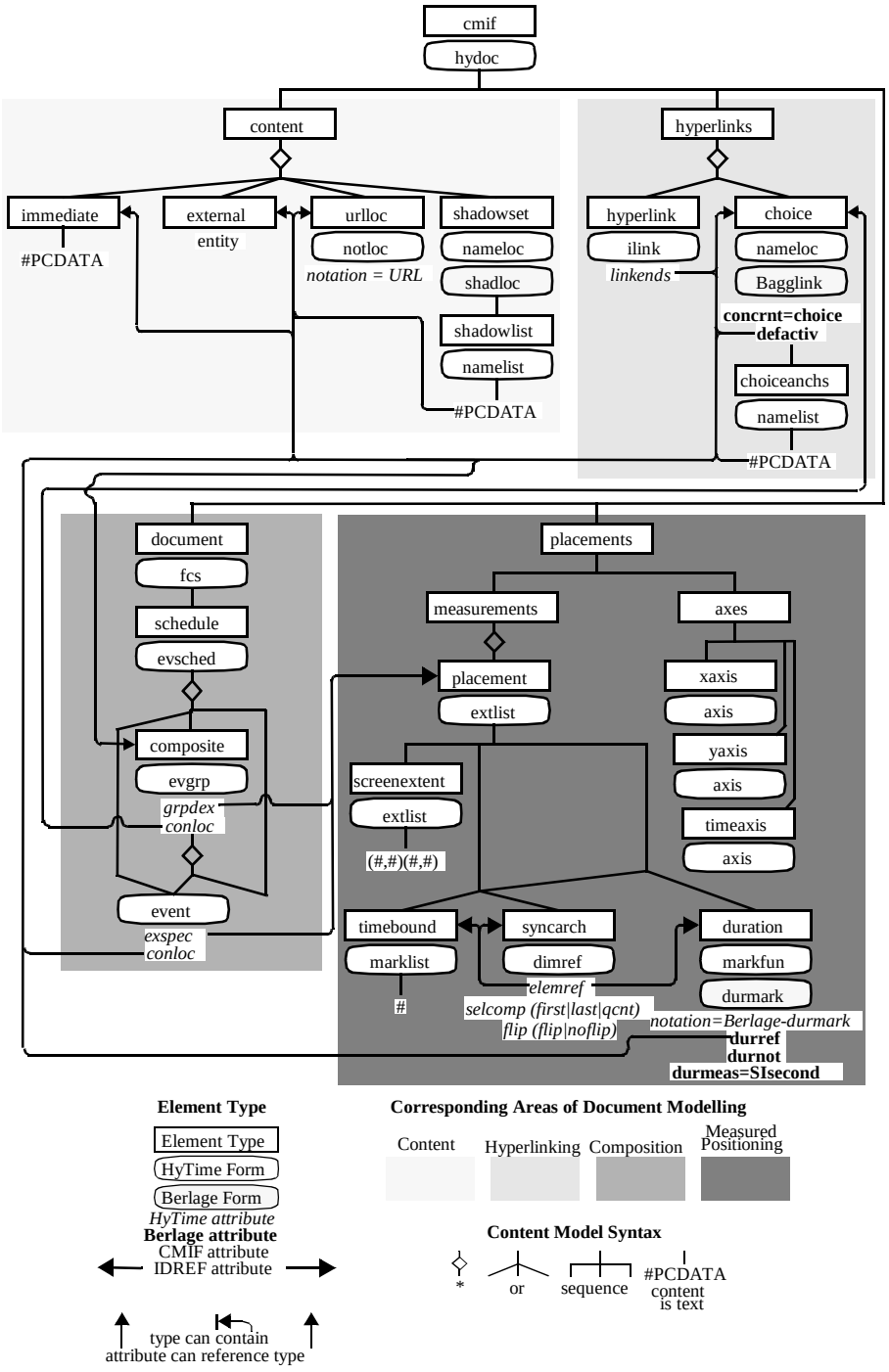

Figure 2. *The HyTime DTD for CMIF [RUT 97]*

The content specifying capabilities of the SGML entity are extended by the HyTime *Notation Location (notloc)*. The contents of a notloc are processed to return some data and correspond again with the address field. A notloc also specifies a *notation* for its address text string to determine how they are to be processed to return the data. The CMIF HyTime encoding uses notloc to specify files located using a URL in its *urlloc* element type. The contents of a urlloc are a string in URL notation specifying the located file. This notation corresponds with the *protocol* field of OpenConnection.

Content is also defined in CMIF with the *shadowset* element type. A shadowset specifies a collection of alternative media objects stored on the distributed system that each represent the same conceptual media object in the document's presentation. This allows, for example, a single video clip to be represented with a variety of video formats, file sizes and resolutions. The actual video clip file used in a presentation of the document is determined by the available resources and bandwidth in the presentation's environment. HyTime constructs such as the *Named Location Address (nameloc)* are used to specify that a collection of located objects are to be considered as a single unit. There is no MHEG-5 equivalent for the shadowset element type. When generating an MHEG-5 presentation for a CMIF document, the most appropriate member of each shadowset must be determined for inclusion.

4.2. *Hyperlinking*

For our discussion of hyperlinking we use the HyTime perspective of it, which is that hyperlinking is the establishment of relationships or associations between document objects. Hyperlinking fits in the Dexter/AHM storage layer. HyTime hyperlinking is handled by the *hyperlinks* module. HyTime's consideration of its hyperlinks is at the generic hypermedia structuring level. This perspective is more general than that of more presentation-specific approaches of MHEG-5 and typical hypermedia systems, in which hyperlinks are considered isomorphic with navigation. Generically structured hyperlinks upon processing can be mapped to either navigational or non-navigational components of the presentation.

A HyTime *hyperlink* joins together a collection of *anchors*. It specifies that each anchor plays some role in an association or relationship with the other anchors. HyTime hyperlinking constructs define allowable *directions of traversal* between anchors in a hyperlink. Traversal in HyTime is considered to be more general than navigational, referring to any processing of a hyperlink that refers to one anchor from another. For each anchor in a link, it can be specified whether or not traversal to or from the anchor through that link it allowed. With this specification, it can be said for each pair of anchors in a hyperlink whether or not traversal can start at the first anchor and go to the second through that hyperlink. In such a case, the first anchor would be the *starting anchor* and the second would be the *ending anchor* for the pair.

HyTime defined *hyperlinks* correspond with MHEG-5 *links*. *Starting anchors* and *ending anchors* as defined by HyTime *traversal directions* correspond, respectively, with the MHEG-5 *LinkCondition* and *LinkEffect* classes. For each potential starting

and ending anchor pair recognized by HyTime hyperlink, the starting anchor is encoded as an MHEG-5 LinkCondition and the ending anchor is encoded as an MHEG-5 LinkEffect. The MHEG-5 *HyperText* class enables a portion of directly-contained text to be the starting anchor of a hyperlink. It specifies that the user's selection of that portion of text can potentially activate a LinkCondition of some MHEG-5 Link.

CMIF presentations have navigational links. With these, a mouse click on a word or image changes the information displayed in the manner common in hypermedia interfaces. CMIF navigational links are represented with HyTime hyperlinking constructs that specify a single source and single destination for traversal. The CMIF navigation structure has a more direct correlation with MHEG-5 navigational structure. For each CMIF link starting anchor, the user's selection of that anchor triggers a LinkCondition which results in a LinkEffect being executed. CMIF starting anchors can be text, which translate to MHEG-5 HyperTexts.

MHEG-5 also defines *SwitchButton* and *PushButton* classes for the navigational interface. These establish specific constructs of the GUI as triggering certain conditions when selected by the user. These selectable constructs are strictly part of the user interface and do not encapsulate portions of the document's content as do HyperText class instances or CMIF starting anchors.

4.3. *Composition*

In this paper composition is the grouping together of document components for a single purpose. The Dexter/AHM model provides a composite construct for grouping document components to be presented together or to inherit common characteristics. The defining of elements in SGML and HyTime makes hierarchical composition an integral part of document structure.

The most basic compositional construct in each document format is the representation of a whole document or presentation as a unit. This provides the highest level of document composition: the root node. The *CMIF* element type and the *HyTime Document (HyDoc)* form contain all the components of one document. The corresponding class of MHEG-5 is the *Application*. A CMIF document can be encoded as a HyTime document, which in turn can be converted to an MHEG-5 Application.

In SGML and HyTime, composition is an inherent aspect of how documents are constructed, since SGML provides a hierarchical model where an element, the most basic SGML construct, can contain other elements. When a reference is made to an element it can also refer to, depending on the circumstances, all of the element's descendants. MHEG-5 and CMIF provide composition for more focussed purposes, as shown below with the MHEG-5 Scene and TokenGroup classes and their corresponding CMIF element types.

An MHEG-5 Application contains one or more *Scenes*. Each Scene specifies a group of media objects to be displayed at one stage of the application's presentation. It is indicative of MHEG-5's presentation-oriented nature that the structural division

in terms of presentation components is made at the top level. In HyTime documents this division is more general in nature. The division of the document into portions that are to be presented at separate stages typically occurs further down in the hierarchy, or is not delimited in terms of the hierarchy but in terms of more intricate aspects of the document's structure. The CMIF hierarchical structure is also more presentation-independent than MHEG-5. The separation of presentation stages from other aspects of document structure, such as hyperlinking and anchoring, contributes to presentation-independence by facilitating the reuse and separate editing of these separable aspects.

The CMIF element type that determines presentation stages is the *composite*. The composite groups together document components to coordinate the timing of their presentation. The contents of a composite will typically be played starting at the same time or one after another. By defining the ordering in time of document component presentation, the composite element type defines presentation stages.

The composite element type uses the HyTime *Event Group (evgrp)* form, which positions media objects within and relative to an area of a measured coordinate system. Unlike Scenes, evgrps do not necessarily correspond to single stages of a document's presentation. A HyTime document's event groups may be structured in such a presentation-independent manner that there is not a one-to-one correspondence in the mapping between evgrps and stages of the document's presentation in a given context. The composite element type of the CMIF DTD does, however, affect the stages of a document's presentation.

The MHEG-5 *TokenGroup* class provides composition to group multiple media objects together in a presentation, either in the same scene or in the same portion of a scene. TokenGroups can contain other TokenGroups, making this composition hierarchical. TokenGroup corresponds with the HyTime evgrp form, since it unites media objects and other evgrps to be collectively positioned in one area of an event schedule. The positioning of its contents are in terms of the bounds established for an evgrp.

4.4. *Measured Positioning*

Measured positioning is the defining of document structure and relationships between document components using measured coordinate systems. Since it establishes relationships between document components, it fits in the storage layer of the AHM/Dexter model. AHM provides constructs in its extension over Dexter for specifying scheduling within the storage layer. While in presentation-oriented hypermedia formats such as MHEG-5 and CMIF coordinate systems are typically measured in terms of time and screen display space, HyTime can use any set of dimension types and any number of axes for its coordinate systems. Upon presentation, these generic schedules can be mapped to screen display coordinates and timelines.

The topmost MHEG-5 class in the hierarchy to start specifying scheduling characteristics is the Scene, which is introduced in Section 4.3. The media objects

within a Scene are positioned along two spacial axes and along a timeline. An evgrp specifies the positioning of media objects within an associated coordinate system, defined with the HyTime *Finite Coordinate Space (fcs)* form. While in MHEG-5 media objects are positioned along two dimensions of space and one of time, an fcs can have any number of dimensions and each dimension can correspond with any type of measurement, including "virtual" ones that have no direct association with the measurements of a document's presentation. MHEG-5 does not have an explicit definition of the coordinate axes of its presentations. The position of objects in two-dimensional space involves the explicit specification of coordinates. The scheduling of objects along a timeline in MHEG-5 involves a separate and more complex combination of constructs, some of which are described below.

The MHEG-5 *Visible* class places a visual media object on the screen display. By specifying that a portion of measured space contains a media object, Visible corresponds with the HyTime *event* form. The Visible class has an *OriginalPosition* field, which states where in the screen display a visual media object appears, and an *OriginalBoxSize* field, which says how much screen space the object takes up. These correspond with the HyTime *Extent List (extlist)* form, which states what measured portion of an event schedule is taken up by an event. The CMIF *ScreenExtent* element type uses an extlist to specify what portion of the screen display is taken up by a media object.

In HyTime the measurement of time is handled with the same constructs and in the same manner as with space and other types of measurement. In both MHEG-5 and CMIF, however, time is handled with different constructs than space. The difference between time and space is more important in presentation-oriented formats than in presentation-independent ones because the processing of a presentation happens in real-time and so has to handle timing during the run of the program, whereas the placing of objects on a screen is a non-continuous process.

The primary MHEG-5 class for handling time is the *Timer*. At a point in a document's presentation, a timer can be established and a time value for it set. Once the amount of time specified by that value has passed, the timer triggers an event, which may in turn trigger an action, such as the presentation of a media object. The establishing of a single numeric value for an aspect of handling a media object's presentation corresponds with the HyTime *marker* form. A marker assigns a single numeric value as part of an extent list. A marker appears as part of a *marker list (marklist)*, which is used by the CMIF DTD as shown in Figure 2. The *TimeBound* CMIF element type, which uses a marker, establishes either the starting time or the duration of a media object's presentation.

4.5. *Appearance*

In this section we discuss aspects of document structure that relate explicitly the document's appearance at presentation. HyTime provides no constructs for this type of structuring, since HyTime is strictly presentation-independent.

Appearance information encoded by MHEG-5 is performed by the *Palette* and *Font* classes. Palette establishes the color of MHEG-5-defined media objects such as text and drawn images. Font establishes the non-color aspects of text appearance. CMIF has corresponding element types defining background color, font size and font color. An important aspect of appearance specification in CMIF is the inheritance of composite elements. Composites group document objects together to define them as inheriting the same properties, such as color and font, from the composite itself. Characteristics assigned to a composite are shared by all objects within it unless the assignments are overridden. This facilitates document editing and reuse because appearance changes to a group of objects can be made in one place and objects can be edited independently of their appearance and have separate appearance specifications applied to them. Since MHEG-5 is a final form presentation, these issues of editing and reusing do not apply.

MHEG-5 also has a *CursorShape* class, which specifies the appearance of the cursor used in the presentation. This is very presentation-specific in nature and is not defined at all by HyTime. CMIF also leaves it up to the individual windowing system to determine the cursor shape used in the interface with the presentation.

5. The Potential Role of DSSSL

Since SGML and HyTime documents describe only generic, presentation-independent structure, they generally need further processing to be indexed, formatted and printed or displayed. DSSSL specifies in a platform-independent way how an SGML document or document set should be processed. To achieve this, DSSSL defines two independent processes. One is the *transformation* process, which is used to convert a document conforming to one SGML DTD to a document conforming to another. Such transformations are used, for example, to convert company-specific documents to an externally defined standard, or for translation of a complex and rich document to a more easily processed one. The second process is the *formatting* process. The document describing the formatting process is traditionally called a style sheet. To be able to develop platform-independent style sheets, DSSSL introduces an abstract target representation to convert to. This abstraction, the *flow object tree*, is a flow-based model of a sequence of pages. Typical objects in the tree are column, header, footer, paragraph and anchor objects. All objects have a specific set of attributes which contain additional information for the formatting application. The use of DSSSL style sheets for hypermedia documents has been discussed in [OSS 97].

A typical DSSSL style sheet is a declarative specification of a mapping from an SGML document to objects in the flow object tree. It is up to a back-end application to convert the flow object tree to the final presentation form, such as RTF, PostScript or PDF. This back-end needs to implement justification, line breaking and page breaking algorithms. Note that the output of the formatting process is a physical description of the document and all information about the logical structure of the document is lost. As a result, the formatting process can in general not be used for

conversion to structured encodings such as SGML, LaTeX or, in our case, the HyTime encoding of CMIF.

DSSSL is designed for the processing of text documents and does not support any scheduling. Using DSSSL for specifying the presentation of hypermedia documents would require extensions to DSSSL's output model to include scheduling and hyperlinking

5.1. *Hypermedia Output Model*

An advantage of DSSSL is the declarative nature of the style sheets describing a formatting process. DSSSL stylesheets describe mapping in terms of an abstract page model as described by the flow object tree. A drawback of this approach is that since the flow object tree models a two dimensional, physical page representation of the document, DSSSL is only suited for page-based destination formats. As a result, multimedia documents cannot be represented by DSSSL's flow object tree because it lacks a temporal dimension.

There are several ways to overcome the limitations of DSSSL's two dimensional page-based output model.

First, one could use only the transformation process, which does not use the flow object tree. Because this process involves an SGML to SGML transformation it requires that both input and output be encoded in SGML. For MHEG-5 output, this approach requires an extra transformation since MHEG-5 engines use an object-oriented API or text encoding of class instances. This transformation would need to convert from an MHEG-5 equivalent SGML encoding to MHEG-5 itself.

Another approach makes use of DSSSL's *grove* concept. A grove can be thought of as a well-defined collection of trees representing the original SGML or HyTime document. The nodes in the grove are the concepts all DSSSL style sheets act upon. In a DSSSL transformation process, the DSSSL engine builds a grove out of the source document. This grove is transformed by the style sheet into a result grove, which is converted back to SGML by the DSSSL engine. The application programmers interface (API) of a number of SGML tools is anticipated to be based on the same grove concept. This enables the last phase in the DSSSL transformation process, the generating the SGML document from the result grove, to be replaced by a process which could generate the desired text or object-based representation directly from the grove.

A third approach would be to use the DSSSL formatting process, but extend its two-dimensional flow object model. New objects need to be introduced to model hypermedia specific features and MHEG-5's objects can play that role. From a modeling point of view, however, the question remains how well the new multimedia specific (MHEG-5) objects can be integrated into the flow-based objects provided by the DSSSL standard. There are some practical limitations as well, since this approach would require a non trivial extension of the DSSSL back-ends used. These back-ends need to be able to generate an MHEG-5 encoding of the objects in the extended tree.

As a result, the hypermedia style sheet can no longer be processed on other (standard) DSSSL systems.

5.2. *Access to Presentation-time Information*

DSSSL provides hooks to allow the involvement of external processes in the transformation and formatting process. Such processes may provide information that is available only at run-time. Typical examples include information about the available network resources, the results of quality-of-service negotiation, run-time measurement information and user characteristics. The introduction of external processing would make the DSSSL stylesheet more application dependent. However, not all run-time processing needs to be in the style sheet. The style sheet merely needs to provide sufficient information to the MHEG-5 playing environment so it will be able to generate an optimal presentation of the hypermedia document based on that information. Actual processing of that information can be done by an MHEG-5 engine.

So while DSSSL is designed for static, page-based output and not for dynamic and interactive multimedia or hypermedia documents, there are several advantages in using DSSSL for hypermedia as well. First, hypermedia documents still need formatting of static text and graphics. DSSSL already provides facilities to do this. These facilities can be readily reused in a hypermedia environment. Furthermore, DSSSL offers an expressive query language which allows querying of hypermedia documents based on both content and structure. A style sheet could use the query language to retrieve specific parts of the document based on the run-time information described above. With DSSSL's HyTime support, queries can use the powerful locating constructs defined by HyTime to find specific parts of the document. However, before DSSSL can be used for the generation of MHEG-5 presentations, several problems need to be solved. Some of these problems are related to system design and architectural issues, such as the need for ASN.1 based output from either the transformation or the formatting process, while others are modeling problems, such as the integration of the multimedia objects provided by MHEG-5 and the flow objects provided by DSSSL.

Even if these problems are solved, developing style sheets that map HyTime documents to MHEG-5 presentations will not be a task done by a typical hypermedia author. Document engineers will need to provide a template style sheet for each HyTime or SGML document type they design. This template should define the basic and default mappings from the domain specific concepts of their document type onto the concepts defined by MHEG-5. Authors may then override these defaults to specify font, color and other style characteristics that are commonly used in text-based style sheets. But they will be able to adjust hypermedia specific styles as well, such as navigation or interaction styles.

6. Incorporation of MHEG-5, HyTime and DSSSL into CMIF Document Processing

The design of the incorporation of HyTime, MHEG-5 and DSSSL processing into the CMIFed environment is illustrated in Figure 3. This incorporation enables the documents to be presented on and incorporated into other SGML and HyTime processing environments. Documents created with the CMIFed authoring tool are encoded in CMIF. Such documents are then processed by the CMIF to HyTime converter, generating a HyTime-conforming SGML document as output. The DTD and the resulting document can be processed by other SGML and HyTime systems.

The next of this transformation is the generation of MHEG-5 presentations for HyTime-encoded CMIF documents. The user creates a document in CMIF, its HyTime version is generated and then an DSSSL style sheet is processed with the document into its equivalent in MHEG-5, enabling it to be presented on MHEG-5 systems.

7. Conclusion

HyTime and MHEG-5 play different but complementary roles in a complete hypermedia environment. HyTime is appropriate for the authoring of hypermedia information to be presented in a wide variety of circumstances over a long period of time. However, HyTime does not encode how this information is to be presented to the user. MHEG-5, on the other hand, is intended for final form presentation of hypermedia information, but confines that information to one style of presentation and one pattern of user interaction.

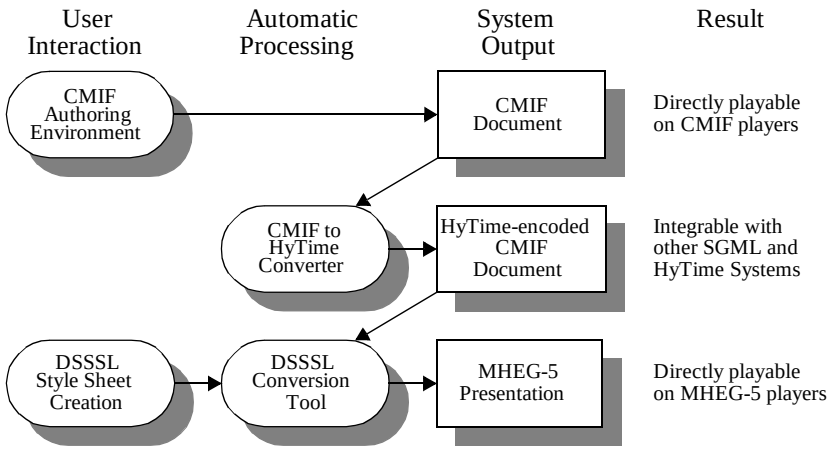

Figure 3. *MHEG-5, HyTime and DSSSL Application to CMIF Processing*

In this paper we presented potential cooperative roles for HyTime and MHEG-5 and explored the issues of their integration. Also discussed were overlaps of hypermedia semantics between HyTime and MHEG-5 which can guide this conversion specification. These issues were illustrated with a description of our CMIF hypermedia environment and the incorporation of HyTime and MHEG-5 into it. These issues include defining the conversion of HyTime documents into MHEG-5 presentations, for which we proposed the use of DSSSL style sheets.

8. References

[DER 94] DeRose, S. and Durand, D. *Making Hypermedia Work: A User's Guide to HyTime*. Kluwer Press, Boston. 1994.

[GOL 91] Goldfarb, C. *The SGML Handbook*. Oxford University Press. 1991.

[GOP 95] Gopal, C. and Price, R. "Multimedia Information Delivery and the MHEG Standard". *Proceedings of the Massachusetts Telecommunication R&D Conference*. 1995.

[HAL 94] Halasz, F. and Schwartz, M. "The Dexter Hypertext Reference Model". *Communications of the ACM*. Vol. 37, No. 2, February 1994.

[HAR 94] Hardman, L., Bulterman, D.C.A. and van Rossum, G. "The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model". *Communications of the ACM*. Vol. 37, No. 2, February 1994.

[ISO 85] International Standards Organization. *Standard Generalized Markup Language (SGML)*. ISO/IEC IS 8879. 1985.

[ISO 92] International Standards Organization. *Hypermedia/Time-based Structuring Language (HyTime)*. ISO/IEC IS 10744. 1992.

[ISO 96] International Standards Organization. *Document Style Semantics and Specification Language (DSSSL)*. ISO/IEC IS 10179:1996. 1996.

[ISO 97] International Standards Organization. *MHEG Part 5*. ISO/IEC IS 13522-5. 1997.

[KIP 91] Kipp, N. *Representation of Hypermedia Objects based on MHEG, HIFF, X11 and HyTime*. ISO/IEC JTC1/SC18/WG8 N2. 1991.

[MAR 91] Markey, B. D. "Emerging Hypermedia Standards: Hypermedia Marketplace Prepares for HyTime and MHEG". *Proceedings of USENIX 91*. 1991.

[OSS 97] Van Ossenbruggen, J., Hardman, L. Rutledge, L. and Eliëns, A. "Style Sheet Support for Hypermedia Documents". *Proceedings of Hypertext 97*. April 1997.

[RUT 97] Rutledge, L., van Ossenbruggen, J., Hardman, L., Bulterman, D. and Eliëns, A. "Generic Hypermedia Structure and Presentation Specifications". *Proceedings of ICCC/IFIP 97*. April 1997.