Presenting Multimedia on the Web and in TV broadcast

Warner ten Kate* **Dick Bulterman†**
Patrick Deunhouwer* **Lynda Hardman†**
Lloyd Rutledge†

Philips Research Laboratories*
Research Institute for Mathematics and Computer Science†

Abstract

This paper investigates the main issues related to the translation of SMIL into MHEG-5 documents. This is driven by the more general objective to achieve interoperability between the domains of Web and digital-TV, where MHEG-5 is used in the digital-TV environment and SMIL is the Web format to specify interactive synchronized multimedia presentations.

A summary of both formats is presented, on the basis of which it is shown how SMIL translates into MHEG-5. Although the formats have differences, such translation appears to be feasible. Aspects of authoring for both domains and other interoperability issues are discussed.

keywords: multimedia; languages; conversion; MHEG; SMIL; WWW; TV;

*mailing address:
Prof. Holstlaan 4,
5656 AA Eindhoven, The Netherlands.
{tenkate,deunhwr}@natlab.research.philips.com

†CWI, Centrum voor Wiskunde en Informatica,
Kruislaan 413,
1098 SJ Amsterdam, The Netherlands.
{Dick.Bulterman,Lynda.Hardman,Lloyd.Rutledge}@cwi.nl

1 Introduction

The MHEG group of ISO/IEC (Multimedia and Hypermedia information coding Expert Group) has issued the MHEG-5 standard for representation and interchange of interactive multimedia applications [1]. The MHEG-5 standard has been adopted in the DAVIC specifications (Digital Audio Visual Council) as the application format for use by digital-TV services [2].

The W3C (World Wide Web Consortium) has established the SYMM working group (SYnchronized MultiMedia) to specify a format for synchronized multimedia presentations [3]. The format is called SMIL: Synchronized Multimedia Integration Language [4].

This paper discusses some issues related to the interoperability between both multimedia presentation formats. The discussion will be along the lines of presenting a SMIL document in the MHEG realm, where throughout the paper MHEG will be used to indicate MHEG-5.

Aside from the presentation format, the Internet and DAVIC environments differ in their communication protocols for data access and transport as well as in the content representation formats they support. With respect to streaming protocols (upper layers) the Internet is based on RTSP and RTP [5, 6], where DAVIC has specified DSM-CC and MPEG [7, 8]. Compared to DAVIC, which uses the principle of a single format for each content type, the Internet imposes no restrictions. The paper will not go into the interoperability aspects related to these protocol and representation differences.

The paper is organized as follows. First a summary is given of the MHEG and SMIL formats. Being a new format under development, SMIL is presented in some detail, and the description may deviate from the final format. Subsequently the translation of SMIL into MHEG is discussed, followed by a discussion on interoperability. A summary of the findings concludes the paper.

2 MHEG-5

The MHEG-5 standard defines the syntax and semantics of a set of object classes to create interactive multimedia presentations that can be distributed in a client/server architecture [1, 9]. The client machine typically resides on a minimal resource platform. The server provides the presentations and downloads the application and content portions to the client as needed. In a digital-TV broadcast environment the download is typically arranged through cyclic retransmission. The client can cache the content, using the related hooks MHEG-5 provides.

The basic notion in MHEG is the Scene object, using capitalization to indicate class names. A Scene contains the media objects such as graphics, sound and video. It also describes behavior like user interaction and synchronization between objects. This is done in an event-action paradigm where an event, such as button clicked, stream ended or timer fired, triggers a set of actions defining the behavior.

Scene objects are grouped into the so-called Application object, which is unique within the presentation and serves as the root entry. Only one Scene at a time is active; per Scene the objects are grouped intended for coordinated presentation. Objects that are shared across Scenes are contained in the Application object. A possible Action is to traverse to another Scene object, providing a way to navigate through the presentation.

The top-level MHEG classes are illustrated in Fig.1, using the OMT notation [10]. Root, Group and Ingredient are abstract classes, indicated by non-bold printing. They model the common aspects of the concrete classes such as Application and Scene, printed in bold. The Root class defines the main properties of all MHEG classes, such as activation and deactivation of the object. The Ingredients are the objects which are Grouped together into either a Scene or an Application.

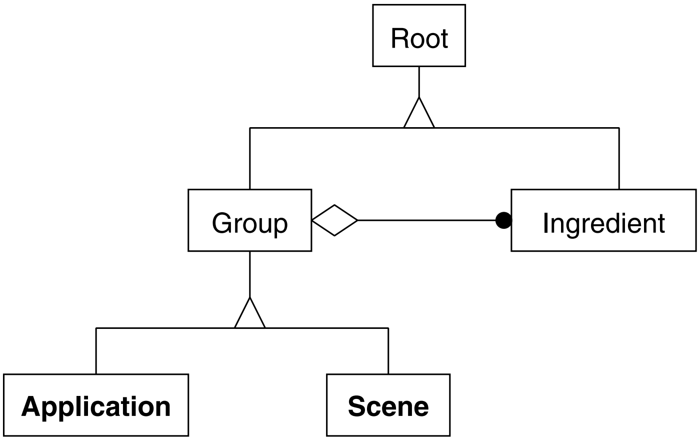

Figure 1: *Top-level MHEG classes.*

The Ingredient class has the subclasses Link, Program, Variable, Presentable, Palette, Font, and CursorShape, as illustrated in Fig.2, where the latter three classes are depicted together. In the context of this paper the Link and Presentable classes are the ones of interest. The Presentable class, which is partly depicted in Fig.3, represents the visible and audible content, including some dedicated components such as hotspots and choice lists. The Link class arranges the coordinated presentation of these, including their interactions.

The Visible class is an abstract class for visuals to be displayed on a screen.

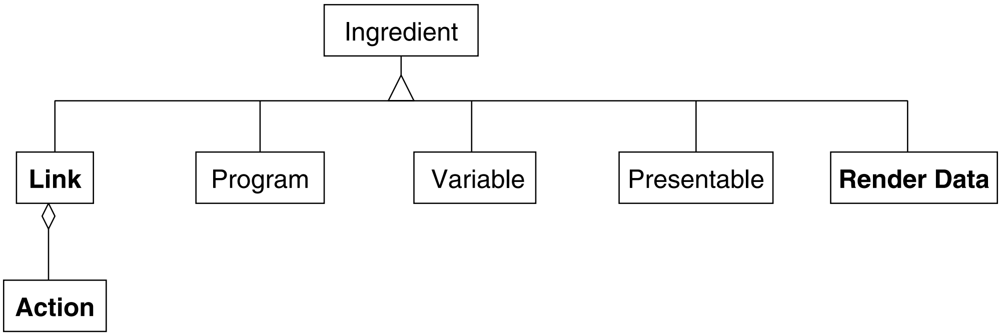

Figure 2: *The MHEG Ingredient class.*

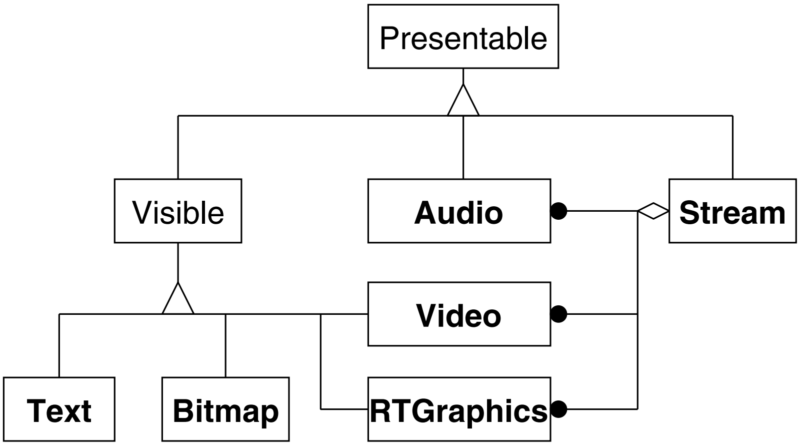

Figure 3: *The MHEG Presentable class (partial).*

MHEG uses a rectangular coordinate system with a fixed size. Concrete classes which inherit from the Visible encompass the Video stream, RTGraphics (animation), Bitmap (graphics) and Text. Objects not shown in Fig.3 include the HyperText and EntryField (form) classes, which inherit from the Text class. Others are Slider and Button, and vector graphics such as Rectangle. The Visibles such as HyperText and Button support direct user interaction, bypassing the otherwise availably event-action mechanism.

The Link object defines a so-called LinkEffect to be executed when a LinkCondition becomes true. The LinkEffect is a list of Actions. In order to fire the LinkEffect, the Link object itself should be in the active state. The LinkCondition is triggered by an event, and consists of an EventSource, EventType and EventData. EventSource identifies the object from which the event may emanate; EventType identifies the type of the event emanated. EventData is optional and distinguishes between events of the same type from the same object, and can thus be used to select among different LinkEffects.

3 SMIL

The SMIL syntax is XML compliant [11]. Elements are represented by start and end tags, in between which their content, usually other elements, is listed. In the case of empty elements, the start and end tags may be combined in a single tag. The hierarchy of elements containing other elements represents the structure of the multimedia presentation, which is given as a tree, where the leaves are the empty elements and the nodes are the container elements. An example is a sequence of images, where empty elements contained by the element 'seq', represent the images. The values of elements, e.g. the URL where to retrieve the image's content data, are assigned through the attributes listed with the start tag.

The constructs SMIL provides can be divided into four categories which will be described below:

1. schedule
2. layout
3. linking
4. alternates

3.1 Schedule

The schedule part of SMIL concerns the synchronization constructs for specifying the temporal order in which the various media types are to be presented. There are three elements: parallel (par), sequence (seq), and media-object. Media-object[1] specifies the media: its type (audio, video, etc.) and location (URL) where the actual content data reside. par and seq are grouping constructs: elements contained by a par are presented together; elements contained by a seq follow in series. par and seq elements can contain other par and seq elements, thus allowing hierarchical structuring of the temporal presentation.

All three elements can contain the attributes 'begin', 'end', and 'dur' (duration) for specifying detailed synchronization within their container context. The assignments are relative to the time of that context or relative to a sibling element. Time starts at zero within each container element.

Non-sibling elements, i.e. elements outside the current container, cannot be referred to. This guarantees that the given temporal tree structure spanned by

[1] Strictly, the entity is an element, not an object; its name is media-object element. For brevity media-object will be used in the text.

the par and seq elements is maintained. In other words, any interpreter can rely on that structure being valid.

Each element can also use the 'repeat' attribute, which specifies the number of times the element should be played. The par element has the additional attributes 'lipsync' and 'endsync'. Lipsync specifies the synchronization accuracy in which the elements in the parallel are reproduced. In so doing it influences the presentation behavior in cases where one of the media items in the parallel experiences a hiccup, e.g. due to bandwidth fluctuations in the network. Endsync provides a way of controlling the end of a parallel. For instance, if an audio and a video stream are played in parallel while both are of unknown duration, endsync can specify that both should end when the first one ends.

3.2 Layout

It is the base philosophy of markup languages to separate document structure and presentation quantification, in order to ease interoperability across various display media. Structure is specified with the document, presentation is handed over to the display device. This philosophy makes a lot of sense in the case of text-oriented documents. However, the more graphical information gets added to the document, the more control the author wants to have on the exact presentation, certainly in case of a multimedia presentation. CSS can be seen as result of this with respect to HTML [12].

Layout is a core functionality of a multimedia presentation. On the other hand, several schemes are available on the Web which support layout, e.g. CSS-positioning [13]. Therefore, layout is provided by SMIL through indirection. The media-object elements can have an attribute, called 'loc', which refers to a layout specification outside the body of the document. The header specifies which layout scheme is used; the layout scheme uses the same names as referenced by the loc attribute, thus connecting the layout constructs with the corresponding media-objects.

Within this framework SMIL has defined a layout scheme to support the basic constructs. So-called 'tuner' elements are provided which have attributes for specifying position (left, top, z) and size (width, height). The ID of the tuner element equals the name referenced by the loc attribute in the media-object element.

By using this basic-layout it is possible to create a self-contained SMIL-document. No other parsers and interpreters are required to construct a player. This is felt to be an important feature to support interoperability with other formats, in particular with MHEG.

3.3 Linking

In addition to schedule and layout, (hyper)linking is the third major component in a SMIL document. Besides extending multimedia to hypermedia, the linking mechanism will also be the foundation of user interaction (linking within the document).

A link specifies a traversal from one or more source anchors to one or more destination anchors. The anchors identify an object in the current or some other document. In SMIL the anchor elements are specified with the media-object. This ensures that anchoring to subparts of the time-area space consumed by the media-object complies with the constructed schedule and layout of the media-object. The link elements, called 'hlink', are specified separately. They contain so-called 'src' and 'dst' elements which list the respective (internal and external) source and destination anchors constituting the hlink. The behavior upon traversing the link is also specified with these src and dst elements, e.g., replace or pause the current presentation.

As well as the hlink, SMIL has an in-line version, which is similar to the a-element in HTML. Being compacter in notation, it facilitates simple linking constructs. It is, however, less powerful in its expressiveness.

3.4 Alternates

Internet delivery uses different grades of Quality-of-Service; not only because of the actual network traffic load, but also because of the spread in bandwidth of access networks, e.g. due to modem speed. It is therefore of interest to support the concept of alternates, which is a set of (container) elements from which one is selected. The 'switch' element specifies such choice situations. The author's preference is assumed to be the first one in the list. For example, the switch could contain media-objects each stemming from the same content source but coded at different compression ratios, the media-objects being ranked in order of decreasing bandwidth. In this case the switch allows the overall performance of the presentation to be traded-off against delivery bandwidth and content quality. It can also be used for other types of choices, such as selecting between alternate languages.

4 Translating SMIL into MHEG-5

SMIL and MHEG both offer functionality for describing a multimedia presentation, including ways to specify

1. content material
2. synchronization
3. layout
4. hyperlinking
5. user interaction
6. dynamic behavior

(User interaction and dynamic behavior are not yet in SMIL, but will certainly be added.) They differ in their characteristics as expressed by Table 1. It is not the purpose to compare the pros and cons of these characteristics, but to discuss how conversion between both formats can be achieved. The discussion will concentrate on translating SMIL into MHEG.

Table 1: Characteristics of SMIL and MHEG.

language	SMIL	MHEG
language type	declarative	declarative
scripting	-	provisions for calling procedural code
language model	structured	Scenes, event-driven
compositional units (entities)	tag-markup (elements)	object-oriented (classes)
representation encoding	XML-compliant/Unicode	ASN.1/DER
representation independence	adaptable	final form
platform	typically Web	minimal resource platforms
resource management	alternates	hooks for resource management
presentation management	interlude	preparing and activating objects

Both SMIL and MHEG are declarative languages, where MHEG provides hooks for calling procedural code. Declarative means that the functionality is described, but not the detailed implementation of how to accomplish that functionality, i.e. the result rather than an algorithm causing that result is coded. Control flow constructs (if-then-else, while/until, for-loop) are typically absent. SMIL describes a multimedia presentation in a structured way. MHEG provides Scenes containing the various objects; their interrelations are described in a event-action paradigm. In MHEG all presentational dependencies (temporal synchronization, user interaction, etc.) are event-driven mechanisms. This provides flexibility in authoring the various presentations, as any event can cause any action(s). The cost of this compactness is the reduction in structure.

SMIL follows the Web's convention of markup notation and is XML compliant; SMIL components are elements. MHEG adheres to an object-oriented approach; its components are class instances (objects). Their interchange is encoded in a final-form format using ASN.1/DER [1, 14, 15]. A textual notation is also supported. A SMIL document is essentially a text file, usually ASCII but extendable to full Unicode, and is adaptable to some extent.

SMIL's primary platform is the Web environment; MHEG is designed for minimal resource platforms. Because of this, SMIL provides the alternates construct to optimize performance, while MHEG provides a variety of means to assist resource management. Next to using the division in Scenes, these encompass actions to set cache priorities, to prefetch and prepare content, and to store data persistently for later retrieval. MHEG explicitly distinguishes between preparing/preloading and activating/running an object (also deactivating/stopping and destructing/unloading, respectively). SMIL has the notion of 'interludes' to mask waiting times during fetching.

Broadly, there are two approaches to realizing a translation:

1. mapping
2. conversion

Mapping refers to the direct replacement of SMIL-code by MHEG-code, i.e. to apply a type of table look-up algorithm. Conversion involves analysis and interpretation of the SMIL-code and subsequent replacement by MHEG-code, which expresses (nearly) the same application appearance. The new code should be more efficient than that resulting through mapping. The conversion process itself consumes more processing power. Where mapping requires local action, conversion enforces global action.

After outlining how SMIL constructs map onto MHEG, some options for optimizing the conversion will be discussed in the following subsections. In a third subsection an implemented conversion environment will be presented.

4.1 Mapping SMIL onto MHEG-5

Table 2 summarizes how the main SMIL constructs map onto MHEG. There are two issues here: that of mapping SMIL elements onto MHEG objects, and that of mapping a SMIL structure onto the MHEG Scenes/event-action paradigm. It will be seen that most SMIL constructs map straightforwardly onto MHEG. A comparison will show that currently MHEG has a richer variety of constructs. However, one should recall that SMIL is still in its development phase.

Table 2: Mapping SMIL on MHEG.

	SMIL	MHEG
media-object		
	audio	Stream/Audio
	video	Stream/Video
	img	Bitmap
	text	Text
	ref	-
schedule		
	top-most par	Application
	top-most seq	Scenes
	other par and seq	flatten out into begin/end assignments
	begin-attribute	Run-action
	end-attribute	Stop-action
	dur-attribute	Stop-action
	begin-, end-, dur-value	Stream-, Timer-event
	lipsync	Stream
	endsync	Stream-event/Stop-action
	repeat	Looping
basic-layout		
	loc-attribute	resolve from tuner-element
	position-, size-attributes	identical
linking		
	anchor	AnchorFired-event, Hotspot
	hlink	Link
	src	LinkCondition
	dst	LinkEffect
	traversal behavior	LinkEffect
alternates		
	switch	resource management, ListGroup

media-object: The elementary media-object elements clearly have their counterpart in MHEG. Composite objects, i.e. multiplexed media, are not supported by MHEG. Here, the translation process should split these elements into their constituent parts. This process is considered to belong to the format conversion of content representation (transcoding), which is not the subject of this paper.

schedule: The par and seq elements give temporal structure in a SMIL document; they do not have a direct counterpart in MHEG. The children of the top-most seq element can be mapped onto MHEG Scenes. The siblings of this seq, if present, transfer to the Application object. If one of those siblings is also a seq, or if there is no seq at all, it may be better to map all elements onto one Scene. All descendants of the accordingly mapped elements should be contained in the same Scene or Application object as that element. This is because MHEG allows only one Scene to be active. The implicit begin and end assignments from their parent construct (par or seq) should be made explicit. If subsequent Scenes turn out to consist of single objects, it may be better to group these objects into fewer or even one Scene; for instance, by mapping to the same Scene those media-objects which are listed contiguously.

Synchronization between elements can be specified using event-action pairs. For example, a delay implicit from a shifted begin maps onto a timer. Synchronization relative to a sibling maps onto the events StreamEvent, StreamPlaying, StreamStopped, or CounterTrigger. Lipsync is ensured by grouping the objects within one Stream object. Endsync maps to a combination of (Stream)events and actions, depending on the actual endsync assignment.

The begin and end operations map to the Run and Stop actions. Repeat maps to Looping the Stream object.

layout: SMIL's own basic-layout scheme is observed only. The layout indirection resolves simply by substituting the loc-assignment in the media-objects with the layout specification of the corresponding tuner elements. The position and size attributes have a one-to-one mapping with their MHEG counterparts defined with the Visible class.

linking: MHEG considers an Application as a closed self-contained presentation. The concept of a web of presentations is moderately supported, e.g. by launching other Applications, or by requesting and reloading new content data. As with all interactions in MHEG, hyperlinking is provided through the event-action paradigm. The hlink clearly maps to the Link object, where the src and dst elements correspond to the LinkCondition and LinkEffect, respectively. The behavior upon link traversal is added to the actions of the LinkEffect. The Link object is contained in the Scene which contains the EventSource.

In case of a HyperText object, the SMIL-anchor maps to an AnchorFired event to emanate from that HyperText object, the EventData of that event being the tag of the actual anchor fired. In the other cases, an anchor has to be constructed by means of the Hotspot class. This class inherits from the Button class and allows to define an area on the display for user interaction (i.e. to fire off events).

alternates: MHEG doesn't have an equivalent to the switch construct in SMIL. The switch is thus to be resolved upon translation, and, because in most cases bandwidth is not a restriction, it is very likely that the first element in the switch is the one to select. Switches on language choices can be translated into a user dialogue, e.g. using the ListGroup class, to be run at the start of the presentation. This assumes that no user preferences are known to the translator, which might equally be the case for a SMIL-player.

4.2 Converting SMIL into MHEG-5

As some constructs in SMIL do not have a one-to-one relation to MHEG-5, there is room for the translator to optimize the mapping in a way dependent on the actual document. This concerns mainly the conceptual difference between structured and event-driven approach, noteworthy w.r.t. scheduling. Others are, for instance, the switch and the interlude constructs from which the translator could derive preload and cache instructions, while selecting the best quality in the switch. SMIL's basic-layout, in contrast, has a one-to-one relation with layout in MHEG, and further optimization is less opportune. This section discusses some optimization aspects related to the structured approach.

The container elements par and seq are mapped onto MHEG by substituting implicit begin and end assignments through explicit ones. A straightforward way to implement this flattening procedure is recursively: the begin and end attributes of the container's children are adapted according to the semantics expressed by the container element; the children which are container elements themselves are subsequently resolved by calling the same mapping function. This is a top-down approach. The process could also be bottom-up, making the recursive nature less explicit: the begin and end assignments in the leaf elements get repeatedly be substituted. It requires the translator to first analyze the complete document in order to determine the tree structure.

Instead of simply flattening the document tree, the convertor could use the structure information. One of the nice features of SMIL is that the begin and end assignments are restricted in scope to the siblings of the element they are attribute of. This means that a container element and all its descendants can freely be taken out or be replaced by another element or (sub-)tree without affecting the other synchronization assignments in the document. Vice versa, a sub-tree can be taken apart for presentation on its own. This characteristic allows a SMIL presentation to be considered as a tree of sub-presentations. This extends to the media-object, which can be a composite media, and the SMIL presentation, which can be taken as a sub-presentation into another document.

The sub-presentation characteristic offers one use of the structure information. If, for example, the player has a limited resource capacity in video decoding, the convertor has to decide how to handle a plurality of video elements. Instead of analyzing the flattened presentation, the knowledge which elements are in a seq would already reduce the problem. In addition, it may be assumed that elements near the top of the tree are the more relevant ones, such as to steer the conversion. For instance, a complete container element could be replaced bluntly by a less complex element, e.g., the container's main element. Although altering the original presentation, this compensates for the limited player capabilities. It is believed that replacement by sub-tree will lead to a more coherent rest-presentation than an approach where such selection is to be performed after flattening the tree.

The sketched scenario exploits the concept of replacing sub-presentations, respectively, reproducing a sub-presentation only. The exact rule base upon which a convertor performs such replacement, remains under study, including the circumstances when to apply the replacement. Note, that this principle could also be used in a SMIL-to-SMIL conversion process, where the user's (SMIL) player also runs on a minimized platform. Possible criteria in such a rule base include

- the amount of additional resources needed to reproduce the children of the container element
- depth in the tree
- the amount of resources needed to reproduce the parts below and above the container element
- the location of the element's content (URL)
- the content representation format

Replacing sub-presentations or reproducing a sub-presentation only can be seen as a form of layered encoding: a base presentation gets extended to a richer presentation by adding additional layers of presentation code. The replacement functionality could be authored by using the switch element; the switch would contain a single, easily displayable element together with a container element holding the richer presentation.

Another way to use the structure information given by the temporal tree is in adding preloading actions, so that the MHEG-player fetches the corresponding objects right before their activation. Likewise, the scheduling of the (cyclic) transmission of the related content data can benefit from the given temporal structure.

A convertor may adapt the mapping of the top-most seq element into Scenes in a more context dependent manner, for instance based on the resulting number and type of objects per Scene, or on the effect of Scene transitions on presentation and resources. Similarly, a convertor might deal more intelligently with situations where a seq element contains an embedded seq, or a par contains a par.

4.3 Translating SMIL into MHEG-5 via a DSSSL-DTD

In order to support the translation of SMIL to MHEG-5, a structured conversion environment has been realized that is based on the CWI CMIF document modeling and presentation system [16]. The CMIF environment uses a special-purpose document encoding that can be translated to a number of different output formats, or which can be played on a native CMIF player.

(CMIF as a format is similar to that of SMIL. CMIF uses a hierarchical structure as the basis for a hypermedia document model. CMIF introduced the concepts of synchronization arcs and the par/seq temporal encoding that have been incorporated by SMIL. CMIF supports a broader hypermedia model and a richer set of presentation attributes than SMIL, but for operational purposes, SMIL can be considered to be a 'CMIF-lite.')

The translation process is illustrated in Fig.4 [17]. First, a SMIL document is created, e.g. with a CMIF authoring tool. Next, an existing CMIF-to-HyTime conversion tool was used to produce a version of the SMIL document in a HyTime-compliant CMIF DTD; this produced a HyTime-encoded CMIF/SMIL document. Next, a DSSSL/XSL style sheet was applied to the converted document to provide an MHEG-5 encoding of the original SMIL presentation.

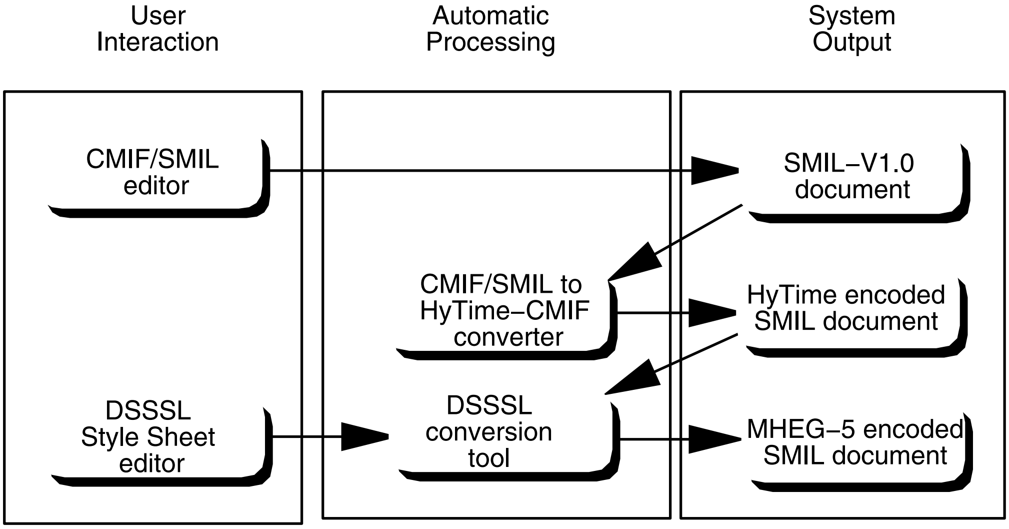

Figure 4: *Structured conversion based on CWI CMIF.*

Note that DSSSL, the Document Style Semantics and Specification Language [18], can be used in a number of ways. Most of the reported uses apply DSSSL as a common language for encoding style sheets for use with SGML/XML and HyTime when converting documents from one DTD to another. Our work applies DSSSL differently. Instead of converting between DTDs or applying specific styles to a presentation, we have successfully used DSSSL to map the HyTime-encoded CMIF/SMIL document directly to a native MHEG-5 application. This reduces the amount of post-processing required to obtain a presentation that can be used by an MHEG-5 player.

At present, the approach described has been applied to document containing HTML text and various standard image formats. The conversion supports mapping of basic datatypes and hypermedia anchors and links. The limited interaction possible in SMIL documents makes the conversion process relatively straightforward for documents with par/seq temporal composition. We are currently experimenting with more complex document encodings.

5 Interoperability

In order to include interoperability with the World Wide Web, translating SMIL into MHEG (and vice versa) is a factor. Fig.5 depicts schematically a possible architecture to achieve such interoperability. As with the discussion on translation, only the situation is considered of presenting SMIL documents on the TV.

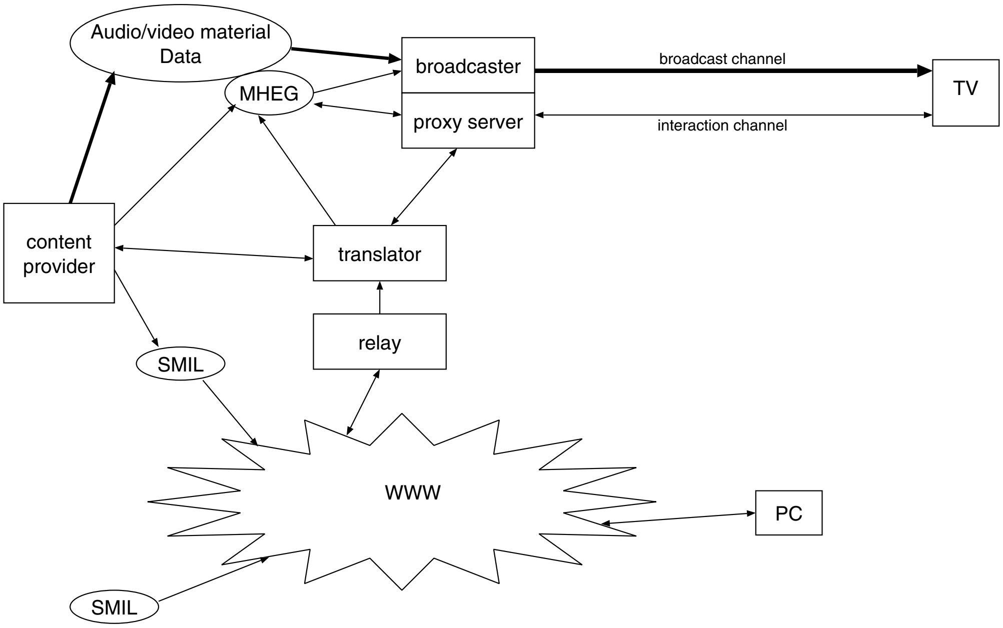

Figure 5: *The architecture for interoperability between WWW and digital-TV domains.*

SMIL documents are available on the WWW. There are content providers who deliver their content in either SMIL or MHEG format, or both, depending on the actual application they provide. Other content providers provide the "traditional" AV program material, possibly enhanced with interactivity. Broadcasters or, more generally, service providers transmit the MHEG applications to the end users, along with the traditional material. Parts of this content may be transmitted at the user's request, via a proxy server at the broadcaster's site. If that request is a SMIL document located on the Web, the proxy server will consult a translator to translate the document into a MHEG application. The SMIL document itself is retrieved via a relay between the Internet and Broadcast networks. In one scenario, the relay is a gateway converting the access and transmission protocols from both networks. Further, the translator possibly takes care of the transcoding of content representation formats.

It should be stated that other architectures are conceivable. For example, the relay to the Internet can merely be a bridge, in which case a so-called IP-tunnel is established through the broadcast channel leaving the protocol incompatibility with the agent at the user's terminal. Further, the sophistication of the translator (mapping or conversion) is an issue, which might be related to design decisions as where to locate the translation (at proxy or with user) and whether to execute the translation in a preprocessing step or on the fly in real-time. These various schemes will not be discussed here.

The scenarios where a SMIL presentation arrives at a TV receiver can be split into two main categories. One is where the TV operates as a Web-access terminal; the broadcaster, or actually the proxy server, acts as an ISP. The other is where the broadcast AV program material is related to the SMIL presentation. From the TV program making perspective this latter is the more interesting use of the Web.

In an earlier paper it was studied how TV program making could benefit from the Internet other than through the browser functionality [19]. It lead to the so-called *trigg&link* concept: The program provider inserts triggers in the broadcast stream; they are related to the program's content and provide entrance to additional information (links). The use of *trigg&link* should 'invite and guide' the viewer to more entertaining value. It was found that in most cases where the Internet could serve as a data base of content, the actual data would be broadcast together with the AV stream and not by way of a URL reference. If the viewer requests for the data, they will be available immediately from the broadcast stream, and there is no need to set up a session with the proxy server to retrieve the data. The implication of this is that the Web-content can be prepared, including its translation, during the generation of the other AV material. This has an impact on the requirements on the translator. Not the proxy server, but the content provider is the main user of the translator, see Fig.5. Also, the conversion should be optimized towards efficient MHEG code rather than to short response times.

Similar as *trigg&link* could guide the viewer to Web-pages (HTML), it could introduce SMIL-presentations. Instead of providing a static picture a small, interactive, movie could enhance the main program. For instance, during the weather forecast the viewer is invited to inspect recent satellite pictures of the earth. A scenario where the viewer moves from Web to broadcast is an EPG available at the broadcaster's Web site, which is enhanced with preview trailers. The EPG connects to the programs being broadcast.

With respect to the content provider in Fig.5 work has been undertaken at CWI to support open authoring and presentation of adaptive, transportable structured hypermedia presentations. The work is focused on the development of an underlying model for hypermedia documents, the Amsterdam Hypermedia Model [20], which provides temporal extensions to the Dexter Hypertext reference model [21]. The development of applications using this model are supported by the CMIFed authoring environment [22], which is being used as the basis for the ESPRIT-IV project CHAMELEON to support multi-platform adaptive documents [23].

The basic premise of CWI's CMIF work has been to define a single document encoding that can be adapted to the needs of the presentation environment. We call this the 'author-once' approach to portable document creation. Once a document has been encoded, it can be adapted at run-time to meet the constraints of the presentation environment, or the encoding can be used for more complex post-processing of a document (such as described in this paper). A diagram of the CMIF architecture is given in Fig.6.

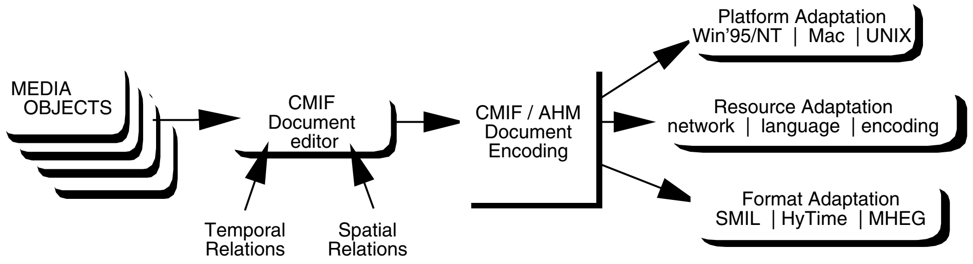

Figure 6: *The CWI's authoring architecture.*

At present, two approaches have been used to provide support adaptive documents. For presentations that preserve the original document encoding, we provide authoring support to describe how parts of a presentation can be transformed to meet the requirements of a particular run-time projection. This can include selection of context format (video/still images) based on bandwidth availability, or it can allow users to select the language that they wish to receive when multiple encodings are available. For documents that need basic format conversions, we use a CMIF-based HyTime projection (that is, a HyTime DTD that understands CMIF presentation semantics) as a basis for conversion, as explained in sec.4.3.

6 Conclusion

A summary is given of the MHEG-5 and SMIL formats. Both formats provide the specification of interactive synchronized multimedia presentations. MHEG-5 is the format in use in digital-TV environments; SMIL has been developed for use on the World Wide Web.

Ways how SMIL translates into MHEG are reviewed. Such a translation is needed to achieve interoperability between both domains of Web and TV. Two approaches are distinguished, mapping and conversion. The use of DSSSL-DTD is an interesting way to implement such translation.

A possible architecture realizing interoperability is shown. A way to author for both domains is presented. It has not been discussed how access and transport protocols convert to complete the gateway, as wasn't the transcoding of content representation.

Although the formats have clear differences, their translation in each other appears to be fairly feasible. Therefore, the current SMIL format seems to support the way towards a closer interoperability between Web and TV.

References

[1] ISO/IEC International Standard IS 13522-5 (MHEG), *Information Technology - Coding of Multimedia and Hypermedia Information*, Part 5: Support for Base-level Interactive Applications, November 1996.

[2] Digital Audio Visual Council, *DAVIC 1.x specifications*, <URL:http://www.davic.org/WORKPLAN.htm>, 1995, 1996, 1997.

[3] W3C Working Group on Synchronized Multimedia (SYMM), *Audio, Video, and Synchronized Multimedia*, <URL:http://www.w3.org/AudioVideo/>.

[4] W3C Working Draft 09-November-97, *Synchronized Multimedia Integration Language (SMIL)*, <URL:http://www.w3.org/TR/WD-smil-971109>, November 1997.

[5] IETF Internet Draft, *Real Time Streaming Protocol (RTSP)*, <URL:ftp://ds.internic.net/internet-drafts/draft-ietf-mmusic-rtsp-06.txt>, November 1997.

[6] RFC 1889, *Real-Time Transport Protocol (RTP)*, <URL:ftp://ds.internic.net/rfc/rfc1889.txt>, January 1996.

[7] ISO/IEC International Standard IS 13818-6 (DSM-CC), *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information*, Part 6: Extension for Digital Storage Media Command and Control, July 1996.

[8] ISO/IEC International Standard IS 13818-1 (MPEG-2), *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information*, Part 1: Systems, 1996.

[9] R. Joseph and J. Rosengren, *MHEG-5: An Overview*, <URL:http://www.fokus.gmd.de/ovma/mug/archives/documents/mheg-reader/rd1206.html>, December 1995.

[10] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, *Object-Oriented Modeling and Design*, Prentice Hall, NJ, 1991.

[11] W3C Proposed Recommendation, *Extensible Markup Language (XML)*, <URL:http://www.w3.org/TR/PR-xml>, 1997.

[12] W3C Recommendation, *Cascading Style Sheets, level 1*, <URL:http://www.w3.org/TR/REC-CSS1.html>, December 1996.

[13] W3C Working Draft, *Positioning HTML Elements with Cascading Style Sheets*, <URL:http://www.w3.org/TR/WD-positioning>, 1997.

[14] ISO/IEC International Standard IS 8824-1 (ASN.1), *Information technology - Abstract Syntax Notation One (ASN.1)*, Specification of basic notation, 1995.

[15] ISO/IEC International Standard IS 8825-1 (DER), *Information technology - ASN.1 encoding rules*, Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), 1995.

[16] D.C.A. Bulterman, G. van Rossum and R. van Liere, *A Structure for Transportable, Dynamic Multimedia Documents*, Proc. Summer 1991 USENIX Conf., Nashville, TN, 137-155.

[17] L. Rutledge, J. van Ossenbruggen, L. Hardman and D.C.A. Bulterman, *Cooperative Use of MHEG-5 and HyTime*, Proc. Hypertexts and Hypermedia: Products, Tools, Methods (HHPTM'97), September 1997.

[18] ISO/IEC International Standard IS 10179 (DSSSL), *Document Style Semantics and Specification Language*, Specification available at: <URL:ftp://ftp.ornl.gov/pub/sgml/WG8/DSSSL/>.

[19] W.R.Th. ten Kate, A.F. Schoenmakers, N. de Jong, H. Charman, P. Matthews, *trigg&link - A New Dimension in Television Program Making*, ECMAST, Milan, May 1997, Lect. Notes Comp. Sci. 1242, Springer (1997) 51-65.

[20] L. Hardman, D.C.A. Bulterman and G. van Rossum, *The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model*, Comm. ACM 37 (1994) 50-62.

[21] F. Halasz and M. Schwartz, *The Dexter Hypertext Reference Model*, Comm. ACM 37 (1994) 30-37.

[22] L. Hardman, G. van Rossum and D.C.A. Bulterman, *Structured Multimedia Authoring*, ACM Multimedia '93, Anaheim, August 1993, 283-289.

[23] D.C.A. Bulterman, *CHAMELEON: An Authoring Environment for Adaptive Multimedia Documents*, For information, see: <URL:http://www.cwi.nl/Chameleon/>.