GR*i*NS: An Authoring Environment for Web Multimedia

Dick C.A. Bulterman[1], Lloyd Rutledge[2], Lynda Hardman[2], Jack Jansen[1] and K. Sjoerd Mullender[1]

Oratrix Development B.V.[1]
Amsterdam, Netherlands
Dick.Bulterman@oratrix.com

CWI: Centrum voor Wiskunde en Informatica[2]
Amsterdam, The Netherlands
Lloyd.Rutledge@cwi.nl

Abstract: The W3C has recently released a language for Web-based Multimedia presentations called SMIL: the Synchronized Multimedia Integration Language. GR*i*NS *is* an authoring and presentation environment that can be used to create SMIL-compliant documents and to play SMIL documents created with GR*i*NS or by hand. This paper discusses GR*i*NS as a tool for creating multimedia education materials on the Web.

1 Introduction

Since its beginning, hypermedia has been used to make education presentations with advantages over text-only and mono-medium material. The presentation of multiple types of media provides different conceptual channels through which the user can perceive information. The level of interaction provided by hypermedia gives the user more control over how the information is presented. Hypermedia synchronization enables the various media components to be integrated into one cohesive presentation, enriching the user's perception of them. The use of different media, the unification and synchronization of their display and the high level of user interaction make education hypermedia presentation not only information but also engaging. However, education hypermedia presentation are typically local and isolated to individual machines or institutions.

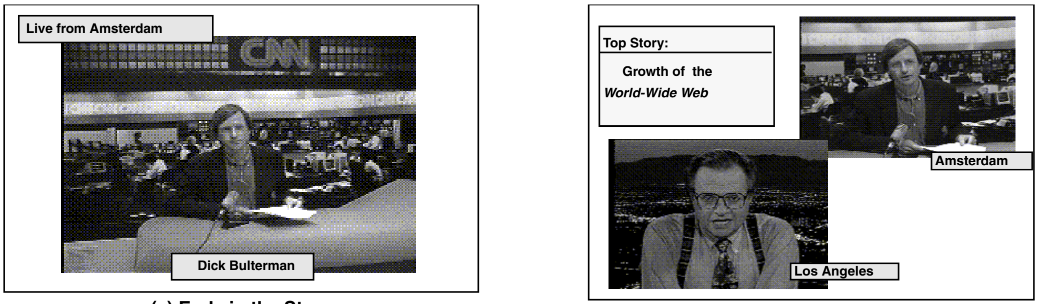
In June 1998 the World Wide Web Consortium released the Synchronized Multimedia Integration Language (Hoschka 1998) (SMIL, pronounced *smile*). In SMIL, the control interactions required for multimedia applications are encoded in a text file as a structured set of object relations. A declarative specification is often easier to edit and maintain than a program-based specification, and it can potentially provide a greater degree of accessibility to the network infrastructure by reducing the amount of programming required for creating any particular presentation. The first system to propose such a format was CMIF (Bulterman et al. 1991, Hardman et al. 1994, Bulterman et al. 1993). Other more recent examples are MADEUS (Roisin et al. 1997) and RTSL (Roisin et al. 1997). In developing SMIL, the W3C working group involved has restricted its attention to the development of the base language, without specifying any particular playback or authoring environment functionality.

This paper presents an authoring and runtime environment called GR*i*NS: a GRaphical *i*Nterface for SMIL. GR*i*NS consists of an authoring interface and a runtime player which can be used together (or separately) to create/play SMIL-compliant hypermedia documents. The authoring part of GR*i*NS can be used to encode SMIL presentations for any SMIL compliant browser or stand-alone player. The GR*i*NS player present any SMIL-compliant. GR*i*NS is based largely on earlier experience with the CMIFed authoring system and the CMIF encoding format, both of which strongly influenced the development of SMIL.

2 A Typical Example SMIL Document

This paper's discussion is illustrated with one type of an educational hypermedia application: that of a informative newscast-style presentation. The educational properties of this example are that it informs the user of a particular body of knowledge. It provides a default passive mode of presentation which the use can simply follow. This passive mode can be overridden by interaction from the user requesting more detailed information about some aspect of the information presented.

The basic premise of this newscast is a story on the explosive growth of the WWW. Several media objects have been defined that together make-up such a newscast. (Note that the selection of objects is arbitrary; we have selected a moderate level of complexity to illustrate the features of GR*i*NS.)



(a) Early in the Story **(b) During the Remote Report**

Figure 1: Two views of the Web newscast.

Figure 1 shows two views of the newscast example, taken at different times in the presentation. Figure 1, part (a) shows a portion of the introduction of a story on the growth of the World-Wide Web. In this portion, the local host is describing how sales of authoring software are expected to rise sharply in the next six months. Figure 1, part (b) shows a point later in the presentation, when the host is chatting with a remote correspondent in Los Angeles.

The ability to link to various homepages makes the semantic content of the document dynamic. As shown in Figure 2, the information content can be augmented during the story depending on the viewer's interests. Links are not restricted to homepages, of course: any addressable object or document deemed relevant by the author should be available. Such behavior comes at a cost, however: we must be able to specify what happens to the base presentation when the link to the homepage is followed: should it pause, should it continue, or should it be replaced by the linked pages. (We discuss this in more detail below.)

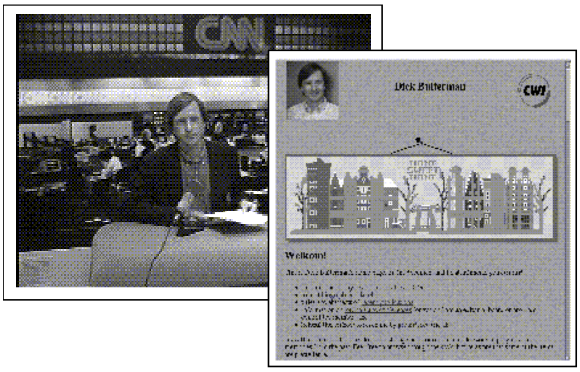


Figure 2: Augmenting information during the news.

SMIL is a declarative language for describing Web-based multimedia documents that can be played on a wide range of SMIL browsers. The format is designed to be easy to author and maintain. Its HTML-like syntax can be written using text editors, and certain aspects of its syntax make the use of text editors easier. This helps take the authoring of hypermedia out of the domain of hypermedia authoring experts and make it more accessible to the general community, including educators.

For larger presentations, authoring systems like GR*i*NS are required to handle the increase complexity. With such systems, aspects of SMIL syntax, such as the temporal hierarchy, make the maintaining of larger and longer-term presentations easier than with non-declarative formats. GR*i*NS and similar SMIL tools further enhance the accessibility of hypermedia authoring to educators and the general computer user community.

Another aspect of SMIL expected to assisted presentation authors is its growing use and availability of examples. As a W3C developed format intended as the HTML for hypermedia, it is receiving a lot of attention. Several different companies have released players and SMIL-related tools (Oratrix 1999, RealNetworks 1999a), and repositories of SMIL presentations are sprouting up across the Web (WebPhD.com 1999). This provides SMIL authors with plenty of example code from which to work. Furthermore, general templates of presentation for particular user communities, such as education hypermedia, could be developed for wide-spread use in authoring individual documents. Example HTML code is frequently used as templates for text HTML editing, and the same practice could be adopted for SMIL. Furthermore, SMIL author environments such as GR*i*NS could use cut and paste features to more easily integrate portions of templates into new presentations.

Architecturally, SMIL is an integrating format. That is, it does not describe the contents of any part of a hypermedia presentation, but rather describes how various components are to be combined temporally and spatially to create a presentation. (It also defines how presentation resources can be managed and it allows anchors and links to be defined within the presentation.) Note that SMIL is not a replacement for individual formats (such as HTML for text, AIFF for audio or MPEG for video); it takes information objects encoded using these formats and combines them into a presentation.

SMIL describes four fundamental aspects of a multimedia presentation: *temporal specifications* to encode the temporal structure of the application; *spatial specifications* to support simple document layout; *alternative behavior specification* to express the various optional encodings within a document based on systems or user requirements; and *hypermedia support*: mechanisms for linking parts of a presentation.

2.1 SMIL temporal specifications

SMIL provides coarse-grain and fine-grain declarative temporal structuring of an application. Coarse grain temporal information is given in terms of two structuring elements:

- `<seq> ... </seq>`: A set of objects that occur in sequence.
- `<par> ... </par>`: A collection of objects that occur in parallel.

Elements defined within a <seq> group have the semantics that a successor is guaranteed to start after the completion of a predecessor element. Elements within a <par> group have the semantics that, by default, they all start at the same time. Once started, all elements are active for the time determined by their encoding or for an explicitly defined duration. Elements within a <par> group can also be defined to end at the same time, either based on the length of the longest or shortest component or on the end time of an explicit master element. Note that if objects within a <par> group are of unequal length, they will either start or end at different times, depending on the attributes of the group.

Fine grain synchronization control is specified in each of the object references through a number of timing control relationships:

- *explicit durations*: a DUR="*length*" attribute can be used to state the presentation time of the object;
- *absolute offsets*: the start time of an object can be given as an absolute offset from the start time of the enclosing structural element by using a BEGIN="*time*" attribute;
- *relative offsets*: the start time of an object can be given in terms of the start time of another sibling object using a BEGIN="*object_id + time*" attribute.

The temporal composition of the <par> and <seq> constructs in SMIL provides an authoring and maintenance convenience by providing meaningful boundaries for presentation components. Multimedia presentations are typically subdivided in terms of time, and authors typically view the composition of their multimedia documents as such.

2.2 Layout specifications

In order to guarantee interoperability among various players, SMIL-V1.0 supports basic primitives for layout control that must be supported on all SMIL players. This structure uses an indirect format, in which each media object reference contains the name of an associated drawing area that describes where objects are to be presented. A separate layout resolution section in the SMIL <head> section maps these areas to output resources (screen space or audio). Non-visible objects (such as audio) can also be defined. Each player/browser is responsible for mapping the logical output areas to physical devices. A priority attribute is being considered to aid in resolving conflicts during resource allocation. (As of this writing the name channel will probably be used as an abstract grouping mechanism (Bulterman et al. 1991) but the deadline for text submission came before this issue was resolved.)

2.3 Alternate Behavior Specifications

SMIL provides a means for defining alternate behavior within a document via the <switch> construct. The <switch> allows an author to specify a number of semantically equivalent encodings, one of which can be selected at runtime by the player/browser. This selection could take place based on profiles, user preference, or environmental characteristics. In the example described, a specification is given that says: play either the video or still image defined, depending on system or presentation constraints active at runtime.

The adaptation SMIL presentations to user characteristics provides what is perhaps the most important use of the <switch> construct. User language can be adapted to, enabling authors to specify one document that can present its content in one of several languages. Varying user abilities are also accounted for in <switch> construct adaptation, such as the use closed-captions that can be turned on and off passed on browser-specified user preferences.

2.4 Hypermedia and SMIL

HTML supports hypertext linking functionality via a straight-forward process: each document has a single focus (the browser window or frame) and anchors and links can be easily placed within the document text. When the link is followed, the source text is replaced by the destination. In an SMIL presentation, the situation is much more difficult. First, the location of a given anchor may move over time as the entity associated with the anchor moves (such as following a bouncing ball in a video)—and even if it does not move, it still may be visible for only part of the object's duration. Second, since SMIL is an integrating format, conflicts may arise on ownership of anchors and the semantics of following any given link.

In SMIL-V1.0, a pragmatic approach to linking is followed. Anchors and links within media objects are followed within the context of that media object. An anchor can also be defined at the SMIL level, which affects the presentation of the whole SMIL document. This effect depends on the value of a `SHOW` attribute, which may have values:

- `REPLACE:` the presentation of the destination resource replaces the complete, current presentation (this is the default);
- `NEW:` the presentation of the destination resource starts in a new context (perhaps a new window) not affecting the source presentation; or
- `PAUSE`: the link is followed and a new context is created, but the source context is suspended.

General support for hypermedia is a complex task. Readers are invited to study the Amsterdam Hypermedia Model (Hardman et al. 1994), which served as the basis for the initial SMIL hypermedia proposal.

3 GR*i*NS: Authoring and Presentation for SMIL

GR*i*NS is an authoring and presentation system for SMIL documents. The GR*i*NS authoring tool is based on a structured approach to document creation (Bulterman et al. 1991). The GR*i*NS presentation tool provides a flexible presentation interface that supports all of the semantics of the SMIL V1.0 specification. In the following sections, we give examples of how GR*i*NS can be used to create the WebNews example. We then discuss some of the issues involved in providing support for the GR*i*NS presenter.

The GR*i*NS authoring environment supports creation of a SMIL document in terms of three views:

- the *logical structure view*, where coarse-grain definition of the temporal structure takes place;
- the *virtual timeline view*, where fine-grain temporal interactions are defined; and
- the *playout view*, where layout projections are defined.

Hyperlink definition and specification of alternative data objects can occur in either the logical structure view or the virtual timeline view.

3.1 The Logical Structure View

If we were to define the Web Growth story in terms of its over-all structure, we would see that the story starts with a opening sequence (containing a logo and a title), and ends with a closing jingle. In between is the "meat" of the story. It contains an introduction by the local host, followed by a report by the remote correspondent, and then concluded by a wrap-up by the local host. This 'table of contents' view defines the basic structure of the story. It may be reusable.

The GR*i*NS logical structure view allows an author to construct a SMIL document in terms of a similar nested hierarchy of media objects. The interface provides a scalable view of the document being created. Note that only structure is being edited here: much of this creation takes place before (or while) actual media objects have been created.

This logical structure is determined by the temporal hierarchy defined by the SMIL `<par>` and `<seq>` constructs. Cutting and pasting of subtrees of this logical structure is provided by this view. This facilitates the cutting and pasting of portions from helpful presentation templates deposited on the Web.

Figure 3, shows a part of the WebNews hierarchy in terms of the logical structure view. (The green box in the middle shows two alternates in the presentation: a video or a still image, one of which will be selected at runtime. The logical structure view has facilities cutting and pasting parts of the presentation, and it allows individual objects or sub-structured to be previewed without having to play the entire application.

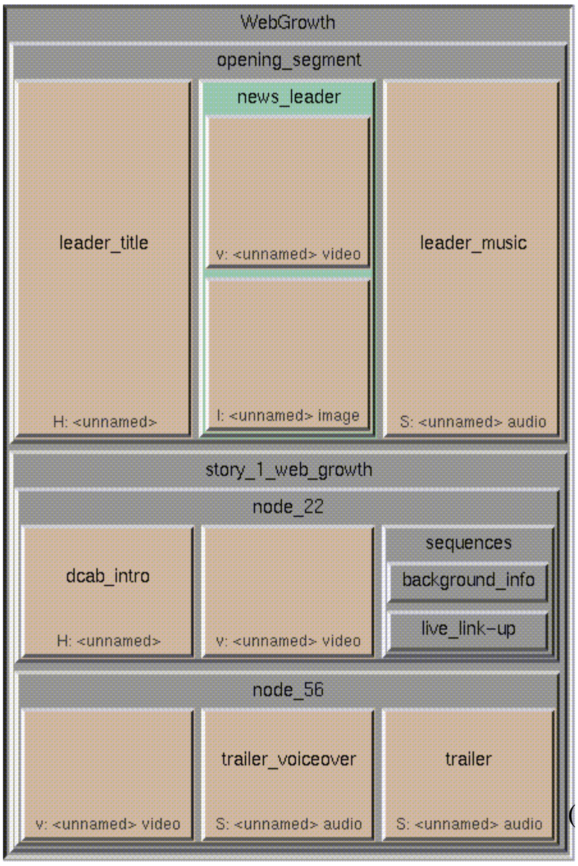


Figure 3: The hierarchy view and attributes

During design and specification, the values of object attributes—including location, default or express duration or synchronization on composite objects—can be entered by the author. In practice, duration of an object or a group will be based on the enclosing structure, will which be calculated automatically. Note that while the basic paradigm of the logical structure view is a hierarchical structure, the author can also specify loop constructs which give (sub-)parts of the presentation a cyclic character.

The logical structure view of GR*i*NS also provides the creation and editing of `<switch>` constructs, which provide presentations their adaptive character.

3.2The Virtual Timeline View

The logical structure view is intended to represent the coarse timing relationships reflected in the `<par>` and `<seq>` constructs. While the attributes associated with an element (either an object or a composite structure definition) can be used to define more fine-grained relationships—such as DURATION or the desire to REPEAT an object during its activation period—these relationships are often difficult to visualize with only a logical structure view. For this reason, GR*i*NS also supports a timeline projection of an application. Unlike other timeline systems, which use a timeline as the initial view of the application, the GR*i*NS timeline is virtual: it displays the logical timing relationships as calculated from the logical structure view. This means that the user is not tied to a particular clock or frame rate, or to a particular architecture. (The actual timing relationships will only be known at execution time.) Note that the virtual timeline view is a generated projection, rather than a direct authoring interface. It is used as a visualization aid, since it reflects the view of the GR*i*NS scheduler on the behavior of the document.

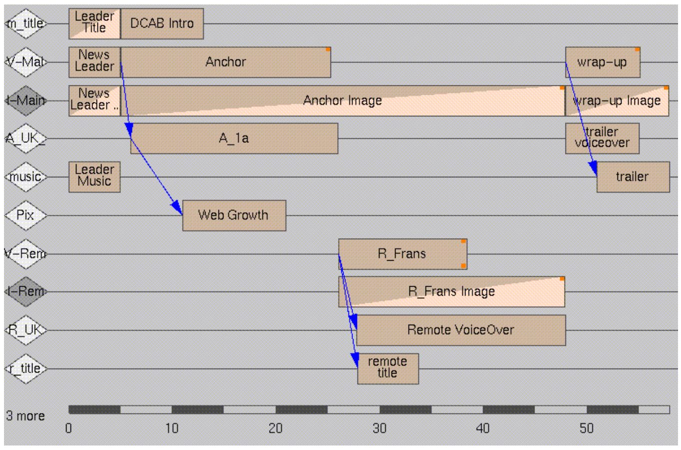


Figure 4: The virtual timeline view of the WebNews. (Time flows from left to right.)

Figure 4 shows a virtual timeline of the Web Growth story. Rather than illustrating structure, this view shows each of the components and their relative start and end times. The timeline view provides an insight into the actual temporal relationships among the elements in the presentation. Not only does it show the declared length and start times of objects whose duration or start offset is pre-defined, it also shows the *derived length* of objects who duration depends on the structure of the document. The timing view shows pre-defined durations as solid blocks and derived durations as blocks with a diagonal line. Events on the timeline are partitioned by the various output channels defined by the user. In the figure, channels which are darkened are turned off during this (part of) the presentation. This allows the author to see the effect of various <switch> settings during the presentation.

When working with the virtual timeline view, the user can define exact start and end offsets within

<par> groups using declarative mechanisms (the *sync_arc*), shown as an arrow in the figure. Sync arc are meant to provide declarative specifications of timing relationships which can be evaluated at run-time or by a scheduling pre-processor.

If changes are made to the application or to any of the attributes of the media objects, these are immediately reflected by in the virtual timeline view. As with the logical structure view, the user can select any group of objects and preview that part of the document. When the presentation view is active (see next section), the virtual timeline view also displays how the player schedules, arms and pre-fetches data during the presentation.

Note that both the logical structure and the virtual timeline views are authoring views; they are not available when a document is viewed via the playout engine only. Both views isolate the user from the syntax of the SMIL language.

4 Current Status and Availability

The GR*i*NS authoring environment and playout engine have been implemented on all major presentation platforms (Windows-95/NT, Macintosh and UNIX). They are intended to provide a reference architecture for playing out SMIL documents and to integrate structured authoring concepts in SMIL document creation. While a full function description of all of the features of the environment are beyond the scope of this paper, we do have documentation available for interested parties. General information is available at GR*i*NS Web page (Oratrix 1999). Free GR*i*NS players are also available from this site.

5 References

Bulterman, D.C.A., van Liere, R. and. van Rossum, G (1991). *A* Structure for Transportable, Dynamic Multimedia Documents (with R. van Liere and G. van Rossum), *1991 Usenix Spring Conference on Multimedia Systems*, Nashville, TN, 137-155. See also `http://www.cwi.nl/~dcab/`.

Bulterman, D.C.A. (1993). Synchronization of Multi-Sourced Multimedia Data for Heterogeneous Target Systems, *Network and Operating Systems Support for Digital Audio and Video* (P. Venkat Rangan, Ed.), LNCS-712, Springer-Verlag, 119-129. See also `http://www.cwi.nl/~dcab/`.

Hardman, L., Bulterman, D.C.A. and van Rossum, G. (1994). The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model. *Communications of the ACM*, 37 (2), 50-62. See also `http://www.cwi.nl/~dcab`.

Hoschka, P. (ed.) (1998). *Synchronized Multimedia Integration Language*, World Wide Web Consortium Recommendation. `http://www.w3.org/TR/REC-smil/`.

Hoschka, P. (1997). Towards Synchronized Multimedia on the Web. *World Wide Web Journal*, Spring 1997. See: `http://www.w3journal.com/6/s2.hoschka.html`

Macromedia, Inc. (1999). *Director 6.0.* `http://www.macromedia.com/software/director/`

Oratrix Development, B.V. (1999). *GRiNS*. `http://www.oratrix.com/GRiNS/`.

RealNetworks. (1999). *RealSystem G2 Products*. `http://www.real.com/g2/products/index.html`.

Roisin, C., Jourdan, M., Layaïda, N. and Sabry-Ismail, L. (1997). Authoring Environment for Interactive Multimedia Documents*, Multimedia Modeling 97*, Singapore. 247-262. See also: `http://opera.inrialpes.fr/OPERA/multimedia-eng.html`

RealNetworks (1999) RTSL: *Real-Time Streaming Language*. See `http://www.real.com/server/intranet/index.html`

WebPhD.com. (1999). *The Just SMIL Gallery.* `http://www.greatsmil.com/smilnow/html/gallery.html`.

W3C (1999). *HTML 4.0: W3C's Next Version of HTML*. `http://www.w3.org/MarkUp/Cougar/`