Time-Based Media Representation and Delivery[1]

T.D.C. Little

Multimedia Communications Laboratory
Department of Electrical, Computer and Systems Engineering
Boston University, Boston, Massachusetts 02215, USA
(617) 353-9877, (617) 353-6440 fax
tdcl@bu.edu

MCL Technical Report 08-15-1993

Abstract–Temporal relationships are a characteristic feature of multimedia objects. Continuous media objects such as video and audio have strict synchronous timing requirements. Composite objects can have arbitrary timing relationships. These relationships might be specified to achieve some particular visual effect of sequence. Several different schemes have been proposed for modeling time-based media. These are reviewed and evaluated in terms of representational completeness and delivery techniques.

Keywords: Temporal specification, multimedia synchronization.

---

[1] In *Multimedia Systems*, J.F. Koegel, Ed., ACM Press, March 1994, pp. 175-200.

1 Introduction

Multimedia refers to the integration of text, images, audio, and video in a variety of application environments. These data can be heavily time-dependent, such as audio and video in a motion picture, and can require time-ordered presentation during use. The task of coordinating such sequences is called multimedia *synchronization* or *orchestration.* Synchronization can be applied to the playout of concurrent or sequential streams of data, and also to the external events generated by a human user. Consider the following illustration of a newscast (adapted from [36]), shown in Fig. 1. In this illustration multiple media are are shown versus a time axis in a *timeline* representation of the time-ordering of the application.

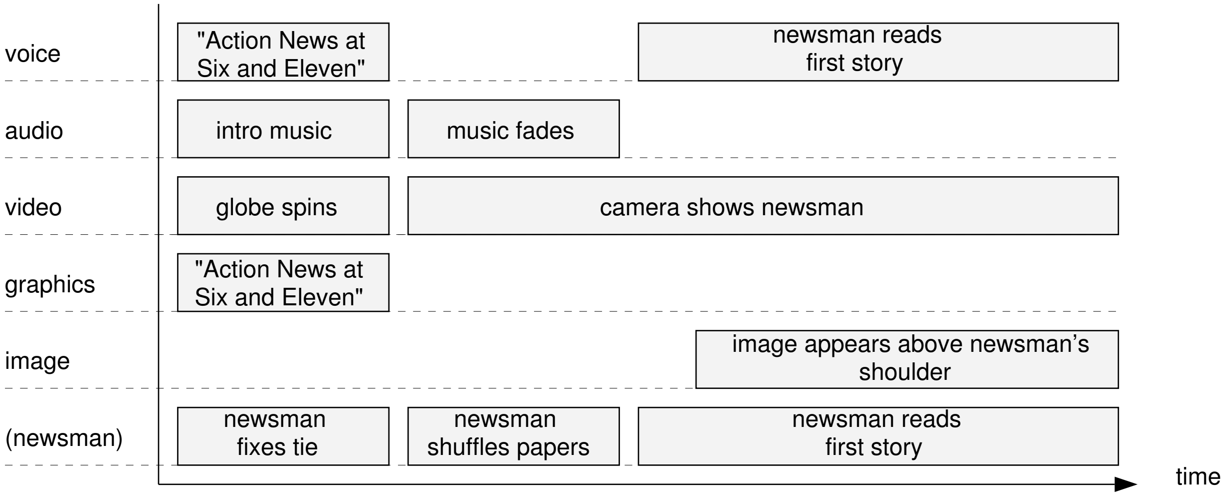

Figure 1: “Action News” Timeline Representation

Temporal relationships between the media may be implied, as in the simultaneous acquisition of voice and video, or may be explicitly formulated, as in the case of a multimedia document which possesses voice annotated text. In either situation, the characteristics of each medium, and the relationships among them must be established in order to provide coordination in the presence vastly different presentation requirements.

In addition to simple linear playout of time-dependent data sequences, other modes of data presentation are also viable, and should be supported by a multimedia information system (MMIS). These include *reverse, fast-forward, fast-backward,* and *random access.* Although these operations are quite ordinary in existing technologies (e.g., VCRs), when non-sequential storage, data compression, data distribution, and random communication delays are introduced, the provision of these capabilities can be very difficult.

In this chapter, we describe time-based media representations with respect to specification, user interaction, and system timing enforcement. The primary issues deal with the *physical level*, the *service level*, and the *human interface level* [73]. At the physical level, data from different media are multiplexed over single physical connections or are arranged in physical storage. The service level is concerned with the interactions between the multimedia application and the various media, and among the elements of the application. This level deals primarily with intermedia synchronization necessary for *presentation* or *playout*. The human interface level describes the random user interaction to a multimedia information system such as viewing a succession of database items, also called *browsing*. We also overview important temporal models necessary to describe time-dependent media, and survey various approaches for their specification. Furthermore, we describe the implications of time-dependent data retrieval in a real system.

The remainder of this chapter is organized as follows. In Section 2 we describe basic models of time. Section 3 describes representations of time with respect to multimedia timing specification and delivery. Section 4 describes the data presentation or delivery problem. Section 5 concludes the chapter.

2 Models of Time

A significant requirement for the support of time-dependent data playout in a multimedia system is the identification and specification of temporal relations among multimedia data objects. In this section we introduce models of time that can be applied to multimedia timing representations, and introduce appropriate terminology.

As a multimedia author, we seek to specify the relationships between the components of this application. The service provider (the multimedia system) must interpret these specifications and provide an accurate rendition. The author's view is abstract, consisting of complex objects and events that occur at certain times. The system view must deal with each data item, providing abstract timing satisfaction as well as fine-grained synchronization (lip sync) as expected by the user. Furthermore, the system must support various temporal access control (TAC) operations such as reverse or fast playout.

Time-dependent data are unique in that both their values and times of delivery are important. The time dependency of multimedia data is difficult to characterize since data can be both static and time-dependent as required by the application. For example, a set of medical cross-sectional images can represent a three-dimensional mapping of a body part, yet the spatial coordinates can be mapped to a time axis to provide an animation allowing the images to be described with or without time dependencies. Therefore, a characterization of multimedia data is required based on the time dependency both at data capture, and at the time of presentation (see Table 1) [45].

Table 1: Definitions of Time Dependencies

static	no time dependency
discrete	single element
transient	ephemeral
natural or implied	real-world time dependencies
synthetic	artificially created time dependencies
continuous	playout is contiguous in time
persistent	maintained in a database
live	data originate in real-time
stored-data	data originate from prerecorded storage

Time dependencies present at the time of data capture are called *natural* or *implied* [45] (e.g., audio and video recorded simultaneously). These data streams often are described as continuous because recorded data elements form a continuum during playout, i.e., elements are played-out contiguously in time. Data can also be captured as a sequence of units which possesses a natural ordering but not neccessarily one based on time (e.g., the aforementioned medical example). On the other hand, data can be captured with no specific ordering (e.g., a set of photographs). Without a time dependency, these data are called *static*. Static data, which lack time dependencies, can have *synthetic* temporal relationships (e.g., Fig. 1). The combination of natural and synthetic time dependencies can describe the overall temporal requirements of any pre-orchestrated multimedia presentation. At the time of playout, data can retain their natural temporal dependencies, or can be coerced into *synthetic* temporal relationships. A synthetic relation possesses a time-dependency fabricated as necessary for the application. For example, a motion picture consists of a sequence of recorded scenes, recorded naturally, but arranged synthetically. Similarly, an animation is a synthetic ordering of static data items. A *live* data source is one that occurs dynamically and in real-time, as contrasted with a *stored-data* source. Since no reordering, or look-ahead to future values is possible for live sources, synthetic relations are only valid for stored-data.

Data objects can also be classified in terms of their presentation and application lifetimes. A *persistent* object is one that can exist for the duration of the application. A *non-persistent* object is created dynamically and discarded when obsolete. For presentation, a *transient* object is defined as an object that is presented for a short duration without manipulation. The display of a series of audio or video frames represents transient presentation of objects, whether captured live or retrieved from a database. Henceforth, we use the terms *static* and *transient* to describe presentation lifetimes of objects while *persistence* expresses their storage life in a database.

In the literature, media are often described as belonging to one of two classes; *continuous* or *discrete* [29, 68, 52]. This distinction is somewhat confusing since time ordering can be assigned to discrete media, and continuous media are time-ordered sequences of discrete ones after digitization. We use a definition attributable to Herrtwich [28]; continuous media are sequences of discrete data elements that are played out contiguously in time. However, the term *continuous* is most often used to describe the fine-grain synchronization required for audio or video.

2.1 Conceptual Models of Time

In information processing applications, temporal information is seldom applied towards synchronization of time-dependent media, rather, it is used for maintenance of historical information or query languages [67, 74]. However, conceptual models of time developed for these applications also apply to the multimedia synchronization problem. Two representations are indicated. These are based on *instants* and *intervals* [5], described as follows.

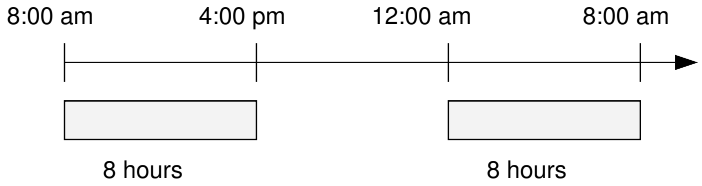

Figure 2: Instants vs. Intervals

A time instant is a zero-length moment in time, such as “4:00 PM.” By contrast, a time interval is defined by two time instants and therefore, their duration (e.g., “100 ms” or “9 to 5”). Intervals are formally defined as follows: let $[S, \leq]$ be a partially ordered set, and let a, b be any two elements of S such that $a \leq b$. The set $\{x | a \leq x \leq b\}$ is called an *interval* of S denoted by $[a, b]$. Time intervals are described by their endpoints (e.g., a and b as defined above). The length of such an interval is identified by $b - a$. The relative timing between two intervals can be determined from these endpoints. By specifying intervals with respect to each other rather than by using endpoints, we decouple the intervals from an absolute or instantaneous time reference, leading us to temporal *relations*.

There are thirteen ways in which two intervals can relate in time [26], whether they overlap, abut, precede, etc. These relations are indicated graphically by a timeline representation shown in Fig. 3 [1]. The thirteen relations can be represented by seven cases because six of them are inverses. For example, after is the inverse relation of before, or equivalently, $before^{-1}$ is the inverse relation of $before$ (a *equals* b is the same as b *equals* a). For inverse relations, given any two intervals, it is possible to represent their relation by using the noninverse relations only by exchanging the interval labels. The equality relation has no inverse.

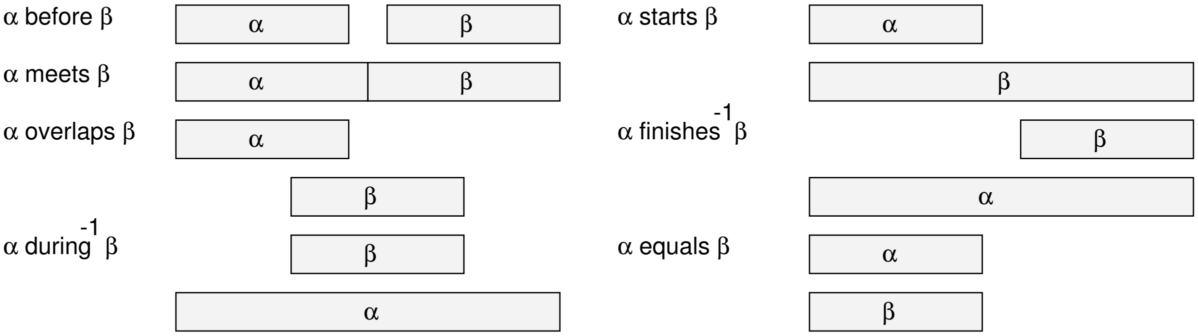

Figure 3: Binary Temporal Relations

Temporal intervals can be used to model multimedia presentation by letting each interval represent the presentation time of some multimedia data element, such as a still image or an audio segment. These intervals represent the time component required for multimedia playout, and their relative positioning represents their time dependencies. Fig. 4 shows audio and images synchronized to each other using the *meets* and *equals* temporal relations. For continuous media such as audio and video, an appropriate temporal representation is a sequence of intervals described by the *meets* relation. In this case, intervals abut in time, and are non-overlapping, by definition of a continuous medium. With a temporal-interval-based (TIB) modeling scheme, complex timeline representations of multimedia object presentation can be delineated.

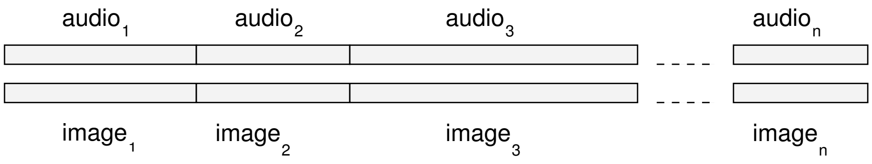

Figure 4: Temporal Interval Representation of Audio and Video

3 Time and Multimedia Requirements

The problem of synchronizing data presentation, user interaction, and physical devices reduces to satisfying temporal precedence relationships under real timing constraints. In this section, we introduce conceptual models that describe temporal information necessary to represent multimedia synchronization. We also describe language and graph-based approaches to specification and survey existing methodologies applying these approaches.

The goal of temporal specification is to provide a means of expressing temporal relationships among data objects requiring synchronization at the time of their creation, in the process of orchestration. This temporal specification ultimately can be used to facilitate database storage, and playback of the orchestrated multimedia objects from storage.

To describe temporal synchronization, an abstract model is necessary for characterizing the processes and events associated with presentation of elements with varying display requirements. The presentation problem requires simultaneous, sequential, and independent display of heterogeneous data. This problem closely resembles that of the execution of sequential and parallel threads in a concurrent computational system, for which numerous approaches exist. Many concurrent languages support this concept, for example, CSP [30] and Ada, however, the problem differs for multimedia data presentation. Computational systems are generally interested in the solution of problems which desire high throughput, for example, the parallel solution to matrix inversion. On the other hand, multimedia presentation is concerned with the coherent presentation of heterogeneous media to a user, therefore, there exists a bound on the speed of delivery, beyond which a user cannot assimilate the information content of the presentation. For computational systems it is always desired to produce a solution in minimum time. An abstract multimedia timing specification concerns presentation rather than computation.

To store control information, a computer language is not ideal, however, formal language features are useful for the specification of various properties for subsequent analysis and validation. Many such languages or varying capabilities exist for real-time systems [81], (e.g., RT-ASLAN, ESTEREL, PEARL, PAISLey, RTRL, Real-Time Lucid, PSDL, Ada, HMS Machines, COSL, RNet etc.). These systems allow specification and analysis of real-time specifications, but not guaranteed execution under limited resource constraints. Providing guaranteed real-time service requires the ability to either formally prove program correctness, demonstrate a feasible schedule, or both. In summary, distinctions between presentation and computation processing are found in the time dependencies of processing versus display, and the nature of the storage of control flow information.

Timing in computer systems is conventionally sequential. Concurrency provides simultaneous event execution though both physical and virtual mechanisms. Most modeling techniques for concurrent activities are specifically interested in ordering of events that can occur in parallel and are independent of the rate of execution (i.e., their generality is independent of CPU performance). However, for time-dependent multimedia data, presentation timing requires meeting both precedence and timing constraints. Furthermore, multimedia data do not have absolute timing requirements (some data can be late).

A representation scheme should capture component precedence, real-time constraints, and provide the capability for indicating laxity in meeting deadlines. The primary requirements for such a specification methodology include the representation of real-time semantics and concurrency, and a hierarchical modeling ability. The nature of multimedia data presentation also implies further requirements including the ability to reverse presentation, to allow random access (at a start point), to incompletely specify timing, to allow sharing of synchronized components among applications, and to provide data storage of control information. Therefore, a specification methodology must also be well suited for unusual temporal semantics as well as be amenable to the development of a database for timing information.

In this section we investigate the requirements of of multimedia data types with respect to time. The important requirements include the ability to specify time in a suitable manner for authoring, the support of temporal access control (TAC) operations, suitability for integration with other models (e.g., spatial organization, document layout models). In Section 4 we describe related requirements with respect to enforcing temporal specifications and delivery.

3.1 Relative Versus Absolute Timing Specification

Timing relationships can be described using relative or absolute timing. In this section we describe the limitations of these two representations.

3.1.1 Temporal Instants

An instant-based temporal reference scheme has been extensively applied in the motion picture industry, as standardized by the Society of Motion Picture and Television Engineers (SMPTE). This scheme associates a virtually unique sequential code to each frame in a motion picture [66]. By assigning these codes to both an audio track and a motion picture track, intermedia synchronization between streams is achieved. This absolute, instant-based scheme presents two difficulties when applied to a computer-based multimedia application. First, since unique, absolute time references are assumed, when segments are edited or produced in duplicate, the relative timing between the edited segments becomes lost in terms of playout. Furthermore, if one medium, while synchronized to another, becomes decoupled from the other, then the timing information of the dependent medium becomes lost. This scenario occurs when audio and image sequences are synchronized to a video sequence with time codes. If the video sequence is removed, the remaining sequences do not have sufficient timing information to provide inter-media synchronization (Fig. 5). Instant-based schemes have also been applied using MIDI (Musical Instrument Digital Interface) time instant specification [49] as well as via coupling each time code to a common time reference [31]. Other work using instant-based representation includes [75] for editing multimedia presentations using timelines.

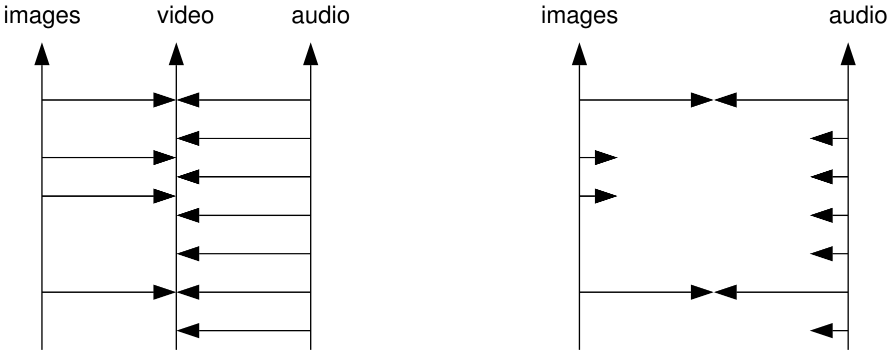

Figure 5: Decomposition Anomalies in an Instant-Based Representation

3.1.2 Temporal Intervals

Temporal intervals can be used to model multimedia presentation by letting each interval represent the presentation of some multimedia data element, such as a still image or an audio segment using TIB modeling. TIB representations are fundamental to the study of time and temporal logic [1]. TIB representations asing *temporal relations* applied to multimedia include HyTime [10], ODA [34] and ODA extensions [29], the work of Little and Ghafoor [41, 46], Dimitrova and Golshani [19] for supporting temporal queries.

Standardization activities have resulted in several approaches to synchronization for electronic documents, including hypermedia. For electronic document representation and interchange, the Office Document Architecture (ODA) [34] describes parallel, sequential, and independent temporal control [22, 56] but does not support synchronization for continuous types. However, there are proposals to extend the ODA model for this purpose [29].

Other existing approaches to timing specification for multimedia either rely on simple time precedence relationships or are based on temporal intervals. Of the ones based on intervals, most only provide support for the simple parallel and sequential relationships. Synchronization can be accomplished using a purely TIB representation, with explicit capture of each of the thirteen temporal relations [65], or with additional operations to facilitate incomplete timing specification [29].

The notion of temporal intervals can also support reverse and partial playout activities. For example, a recorded stream of audio or video can be presented in reversed order. For this purpose, *reverse* temporal relations can be defined. These relations, derived from the forward relations, define the ordering and scheduling required for reverse playout. Furthermore, *partial interval* playout is defined as the playout of a subset of a TIB sequence [46].

3.1.3 Parallel and Sequential Relations

A common representation for time-dependent media relies on a subset of the thirteen temporal relations by using only the parallel (*equals*) and sequential (*meets*) relations. By restricting temporal composition operations to these relations, most temporal interactions can be specified. This approach has been used by Poggio et al. in the development of the Command and Control Workstation Project (CCWS) [55], by Postel [57] (also including an *independent* relationship, and by Ravindran [61] using AND-OR graphs and an *occurs–after* relation to specify timing precedence.

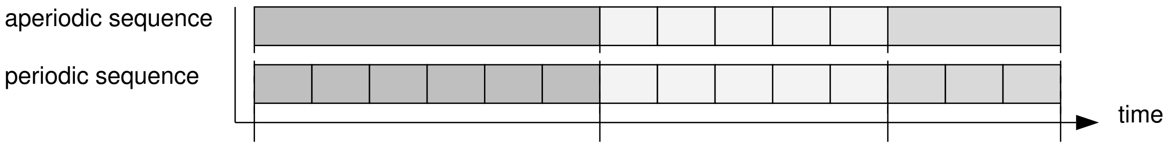

Figure 6: Aperiodic Interval Conversion

This definition also requires uniform representation of data elements to eliminate overlap in time. Conversion from aperiodc representations can be achieved by decomposing larger intervals into a uniform size as shown in Fig. 6.

3.2 Temporal Access Control

A significant requirement for a media representation is the support of TAC operations. These operations provide the base functionality on which time-based multimedia applications can be built, including the system support (delivery) described in Section 4. Clearly there are common characteristics required by the media authoring system, the user TAC functionality, and the system support primitives. Here we introduce and identify the following TAC operations:

- reverse
- fast-forward
- fast-backward
- midpoint suspension
- midpoint resumption
- random access
- looping
- pseudo-sequential access (browsing)

These operations can be implemented in various ways. For example, fast-forward can be provided either by skipping video frames, or by doubling the rate of playout. Therefore, these operations can imply vastly different data structures and system delivery functionality.

3.3 Incomplete Timing

Under some conditions, it may be desirable to introduce incomplete timing specifications as can often arise when a time-dependent data are to be played out in parallel with static ones [29] (Fig. 7). For example, if an audio segment is presented in synchrony with a single still picture, the time duration for image presentation could be unspecified and set to the duration of the audio segment. Incomplete specification can allow the static medium to assume the playout duration of the continuous medium. It is always possible to incompletely specify the timing for the parallel *equals* relation when only one medium is not static. For other types of relations, more information is required to describe the desired temporal result.

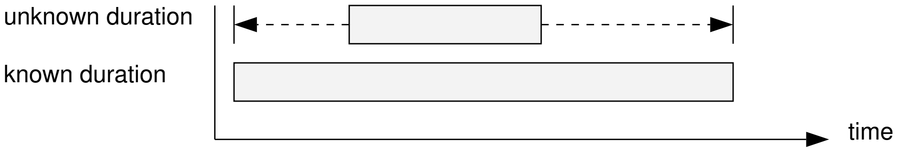

Figure 7: Incomplete Timing Specification

If both media have preassigned but unequal time durations, synchronous playout requires forcing one medium to alter its timing characteristics by time compression/expansion or data dropping/duplication. This kind of timing *coercion* is straightforward in theory [41, 29] but has limited applicability to some media (e.g., music).

3.4 Temporal Transformations

Temporal transformations change one frame of time reference to another as illustrated in Fig. 8. These transformations can meet the conceptual TAC operation requirements although they can also be restricted by limitations of system delivery mechanisms. Temporal transformations include

- scaling
- cueing
- inverting
- translation (shifting)

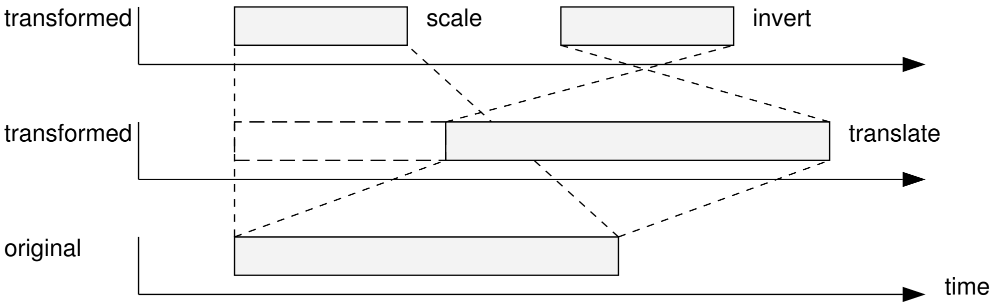

Figure 8: Common Temporal Transforms

Temporal transformations can be applied to many time-based representations. If a time-based representation expresses precedence and ordering, or relative timing, temporal transformations can provide a mapping from the representational domain to a playout time coordinate system. This approach is used by the Athena Muse system [31], HyTime [51], Herrtwich [28], Gibbs et al. [24], Dannenberg, [12], and many others. For example, the relative timing of video frames as described by sequence numbers ($i = 1, 2, 3, ...$) can be mapped to real-time units of 15, 30, or n frames/s as illustrated in Fig. 9. In the Athema Muse system [32], time is described as a dimension that can be manipulated apart from real-time, and is treated as a virtual dimension. The system must provide support for any transformation or manipulation of this dimension (a virtual time coordinate system).

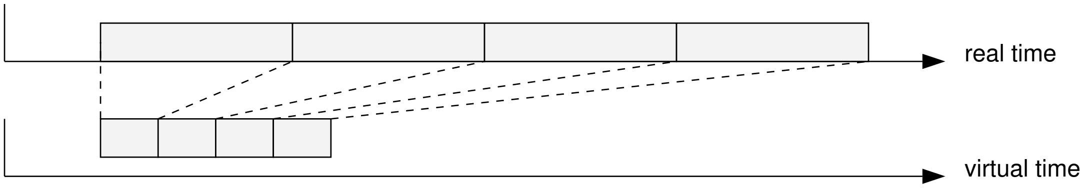

Figure 9: Projection of Virtual Time to Playout (Real) Time

Some of the requirements for time-dependent data are not well described by either the graph or language-based specifications. For example, to reduce (slow motion) or increase (fast-forward) the speed of a multimedia presentation via temporal transformation, these temporal models are deficient. These requirements can be addressed by *temporal abstractions*, which are means to manipulate or control the presentation of a temporal specification via time reference modification.

Various virtual time abstractions have been described in the literature [28, 3, 10]. These describe the maintenance of a time reference that can be scaled to real-time and adjusted to appropriate playout speeds (e.g., Fig. 9). If real-time is defined as nominal clock time as we perceive it, then virtual time is any other time reference system suitable for translation to real-time. For example, a unitless reference can be converted, or projected [10] to real-time system by any scaling or offsetting operations. In this manner, the output rate and direction for a sequence of data elements can be changed by simply modifying this translation, i.e., an entire temporal specification, language or graph-based, can track a specific time reference or translation process.

For continuous data, temporal information can be encapsulated in the description of the data using the object-oriented paradigm [28]. Using such a scheme, temporal information including a time reference, playout time units, temporal relationships, and required time offsets can be maintained for specific multimedia objects.

A timed data stream $s = \{m_i\}$ where $m =$*(data value, time stamp, duration).* This is combined with access modifier called *clock*, which acts upon the encapsulated data:

$$\mathit{clock} = (R, S, V_0, T_0)$$

where R is the clock rate ($1/R$ is its period), S is the speed (nominally of unit value – similar to utilization), V_0 is the initial clock value, and T_0 is the absolute start time of the clock. The clock provides a time reference system to the timed data stream. Using the clock, the ith item is defined as

$$m_i = (V_0 + iS/R, T_0 + i/R, 1/R)$$

i.e., the time capsule can define the behavior of a periodic stream and also provide various TAC operations through transformations.

If the data are periodic, this approach can define the time dependencies for an entire sequence by defining the period or frequency of playout (e.g., 30 frames/s for video). Similarly, for mixed-type time-dependent data, there have been several proposals for their conceptual modeling, most based on TIB schemes [11, 29, 65].

Another scheme, used in the Etherphone environment [78], defines abstractions for continuous media of audio and video. Here *samples* and *frames* describe the basic audio and video data unit sizes, respectively. *Strands* are defined as sequences of either audio samples or video frames. When strands become aggregated or interleaved, they become *ropes*.

3.5 Non-Temporal Transformations

Another important requirement of a temporal representation scheme is the ability to indicate non-temporal transformations of data. These transformations include the *fading* of audio and video signals, mixing of channels, color enhancement of images, generation of fonts, etc. To distinguish these from temporal transformations we call these *spatial* operations [43]. A number of time-based data representations have been proposed to capture both spatial and temporal transformations. These include the work of Herrtwich [28], and Gibbs et al. [24].

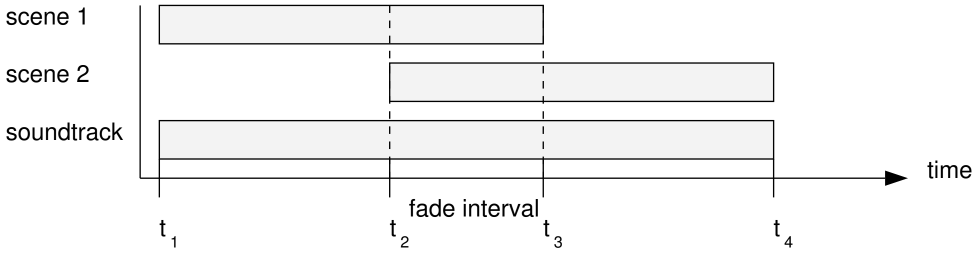

Figure 10: Fading Interval

Typically both a time-based representation and a spatial transformation model is required to describe a complex multimedia orchestration. Fig. 10 illustrates this point. Here, two video scenes are synchronized with a soundtrack (music) but overlap during a *fade interval*. Over this interval the visual effect is one of a fading of one scene to the other. This activity can be described as a spatial transformation or process applied to the video data. Few representational schemes have an appropriate means of represeating this time-dependent transformation in a uniform data model (i.e., video is not preprocessed to provide the fading effect).

3.6 Abstractions for Authoring and Visualization

Multimedia authoring (described elsewhere) of time-dependent multimedia presentations requires some means of abstracting the final product via a representational scheme. Both language-based (including scripting) and flow graph- (or icon) based approaches have been proposed. In each case, the significant requirement is the ability to represent concurrency and to specify real-time presentation timing.

3.6.1 Graph-Based Representations

Although some language-based or script-based representations satisfy these requirements, graphical models have the additional advantage of pictorially illustrating synchronization semantics, and are suitable for visual, icon-based orchestration of multimedia presentations. Graph-based representations satisfying these requirements include timelines, flowgraphs, the Timed Petri net [54, 41, 70, 21], and temporal hierarchies [61, 46].

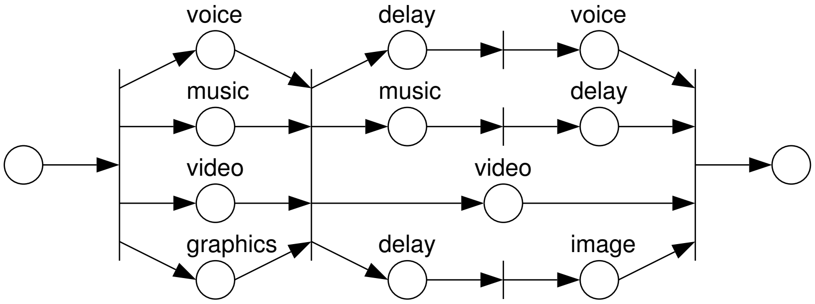

Figure 11: OCPN for the "Action News" Example

For example, Fig. 11 represents an OCPN (Object Composition Petri Net - a TPN derivative [41]) of the "Action News" example of Fig. 1. This TPN explicitly captures any of the temporal relations, and can provide simulation in both the forward and reverse directions. Each *place* in this TPN represents the playout of a multimedia object while transitions represent synchronization points. In contrast, the TPNs of Stotts and Furuta [70] and Fujikawa [21] are designed to capture the nondeterministic actions introduced by a user through browsing.

3.6.2 Script-Based Representations

There are a number of language-based temporal representation schemes generally formulated on parallel and sequential programming language features. *Scripts* represent a subset of this classification, but are derived from the scripting of theatrical works.

For language-based schemes, an extension for the language CSP has been proposed to support multimedia process synchronization, including a resolution of the synchronization blocking problem for continuous media [68]. Various other language-based approaches have also been proposed (e.g., specification using LOTOS (Language Of Temporal Ordering Specification) [79], and process-oriented synchronization in CCWS [55]).

HyTime (Hypermedia/Time-based Structuring Language – ISO/IEC Draft International Standard DIS 10744) [10, 51] and the HyTime application SMDL (Standard Music Description Language) [11] are language-based approaches to synchronization based on SGML (Standard Generalized Markup Language, ISO/IEC 8879-1986). The HyTime specification provides a scripted form of language specification. HyTime uses a SGML format, comprehensive, deals with temporal intervals and relationships, formulated to handle music, beyond the scope of this discussion (the complexities of the representation, and this author's knowledge (me) of music representations). At this time there is no HyTime "engine" available for interpreting HyTime documents.

Scripting is another possibility. Scripts are used in the description of music (e.g., Dannenberg [14, 15, 16] as well as for indicating "paths" though documents [84]. Rennison et al. present a scripting language for multimedia (MuXScript) [63]. Tsichritzis et al. describe a representative scripting language [77] for multimedia time dependencies using sequential, parallel and looping operations. For sequential operations, a_1 occurring before a_2 is specified as

$$a_1 >> a_2$$

Similarly, parallel operations such that a_1 occurs simultaneously with a_2 can be specified as

$$a_1 || a_2$$

Furthermore, iteration of n times is specified with n^*a.

Some of the requirements for multimedia presentation are not well described by either the graph or language-based specifications as described previously. To provide temporal transformations such as playout rate changes, additional abstractions can be used as modifiers to either a language-based or graph-based representation.

Additional requirements for authoring include support for spatial transformations, composition of composite orchestrations, relative timing management, recursion, encapsulation (hierarchy), iteration (looping), incomplete timing specification, and support for user input (discussed in Section 3.7).

With respect to relative timing management, media editing operations such as cutting and pasting result in segments of time-dependent data that must be repositioned in a multimedia orchestration. Timing of individual data elements must be maintained with respect to the selected elements rather than some absolute time.

3.7 Interaction and Synchronization

When a human user interacts with a multimedia system, the application must synchronize the user and the external world. This can take the form of starting or stopping the presentation of an object, posing queries against the database, browsing through objects, or other inherently unpredictable user or sensor-initiated activities. For continuous-media systems, user interaction also implies random access to a sequential form of information. Consider a database of video stills representing scenes from an automobile, shot while looking out at a city's streets [40]. If the scenes are recorded at regular intervals, then a virtual "drive" down the street is possible through animation. When the database contains images from all possible orientations (e.g., all streets of a city), "driving" may include "turns" and corresponding jumps out of the sequential nature of the sequence of images corresponding to a street. Synchronization in this case requires coordination of the multimedia presentation with random external events created by the user. This application has been implemented [64] for interactive movies by using the hypertext paradigm.

The essence of the hypertext paradigm is the nonlinear interconnection of information, unlike the sequential access of conventional textual information. Information is linked via cross-referencing between keywords or subjects to other fragments of information. It is possible to represent this interaction using a graph-based model such as the Petri net [70, 71] and also rely on a detailed time-based representation for individual data objects. Such a Petri-Net-Based-Hypertext (PNBH) expresses information units as net places and links as net arcs. Transitions in PNBH indicate the traversal of links, or the browsing of information fragments. For example, in Fig. 12 we show a PNBH network consisting of segments of an interactive movie (such as Lippman's). These segments can be played-out in a random order, as selected by the user and restricted by the semantics of the net.

Unlike the OCPN, which is a form of *marked graph* [54], net places in PNBH can have multiple outgoing arcs, and therefore can represent nondeterministic and cyclic browsing.

Instead, the OCPN specifies exact presentation-time playout semantics, useful in real-time presentation scheduling. Clearly these two models complement each other for specifying both user interaction and presentation orchestration.

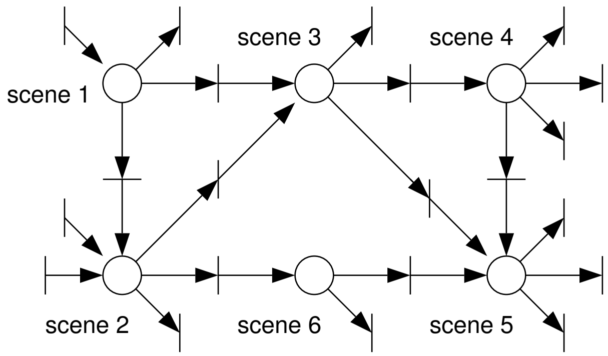

Figure 12: PNBH for Representing Relationships Between Scenes of a Motion Picture

Other important time ordering in multimedia includes the coordination of multiple users in computer-supported collaborative work (CSCW) or group work technology. Important considerations here are concurrency control for shared objects and the management of distributed data sources. Time management in CSCW is beyond the scope of this chapter, however, some recent work on this topic includes Ravindran & Prasad [62], Ramanathan et al. [58], and Yavatkar [82].

4 Support for System Timing Enforcement – Delivery

Temporal intervals and instants provide a means for indicating exact temporal specification. However, the character of multimedia data presentation is unique since catastrophic effects do not occur when data are not available for playout, i.e., deadlines are *soft* in contrast to specification techniques which are designed for real-time systems with *hard* deadlines [20].

Time-dependent data differ from historical data which do not specifically require timely playout. Typically, time-dependent data are stored using mature technologies possessing mechanisms to ensure synchronous playout (e.g., VCRs or audio tape recorders). With such mechanisms, dedicated hardware provides a constant rate of playout for homogeneous, periodic sequences of data. Concurrency in data streams is provided by independent physical data paths. When this type of data is migrated to more general-purpose computer data storage systems (e.g., disks), many interesting new capabilities are possible, including random access to the temporal data sequence and time-dependent playout of static data. However, the generality of such a system eliminates the dedicated physical data paths and the implied data structures of sequential storage. Therefore, a general MMIS needs to support new access paradigms including a retrieval mechanism for large amounts of multimedia data, and must provide conceptual and physical database schemata to support these paradigms. Furthermore, a MMIS must also accommodate the performance limitations of the computer.

Once time-dependent data are effectively modeled, a MMIS must have the capability for storing and accessing these data. This problem is distinct from historical databases, temporal query languages [67, 74], or time-critical query evaluation [33]. Unlike historical data, time-dependent multimedia objects require special considerations for *presentation* due to their real-time playout characteristics. Data need to be delivered from storage based on a prespecified schedule, and presentation of a single object can occur over an extended duration (e.g., a movie). In this section we describe database aspects of synchronization including conceptual and physical storage schemes, data compression, operating system support, and synchronization anomalies.

4.1 Synchronization

The term *synchronization* defines the occurrence of simultaneous events. By using dedicated devices and complete parallelism (e.g., the sound and image tracks of a motion picture) synchronization can be achieved. In a more general purpose computer system, it is more difficult to attain this goal. Instead, storage devices, the network, the system bus, and dynamic memory must be carefully reserved and scheduled to provide similar but more flexible functionality. At each component and at different levels of system abstraction the synchronization requirements differ in character and control.

For multimedia data, the absolute synchronization requirement can be relaxed to different degrees for each medium without adversely affecting their presentation quality. For example, Table 2 shows the synchronization tolerances for various media (adapted from [27]).

We define timing parameters characterizing intermedia and real-time synchronization for the delivery of periodic (e.g., audio and video) and aperiodic data (e.g., text and still images). Parameters applicable to aperiodic data are *maximum delay, minimum delay,* and *average delay* as measured with respect to real time or with respect to other aperiodic data [48] (Table 2). For periodic data, maximum, minimum, and average delay are also applicable to individual data elements, but in addition, *instantaneous delay variation* or *jitter* is important

Table 2: Synchronization Tolerances for Various Media

medium	maximum delay (s)	maximum jitter (ms)
voice	0.25	10
video (TV quality)	0.25	10
compressed video	0.25	1
text	1	not applicable
data (file transfer)	1	not applicable
Image	1	not applicable

for characterizing streams. These parameters can describe time skew with respect to real-time as well as to other periodic streams in an manner analogous to a phase angle.

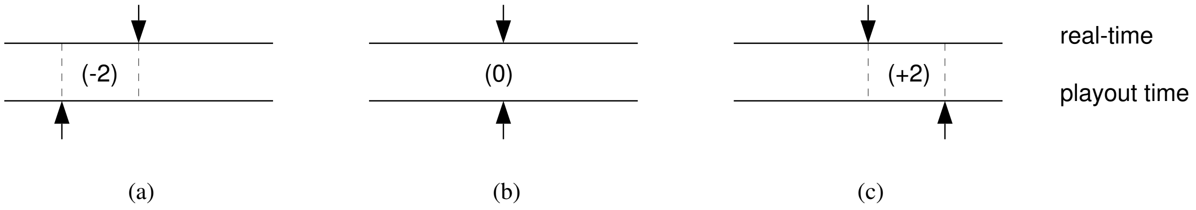

Figure 13: Skew: (a) lagging, (b) none, (c) leading

Synchronization implies the occurrence of multiple events at the same instant in time. If they occur at different instants they are *skewed* in time (Fig. 13). For two sequences of events (e.g., sequences of audio and video data frames) individual differences between corresponding events are called *jitter*, whereas the average difference over some interval of n frames is called *skew*.

For periodic data such as audio and video, data can be lost resulting in *dropouts* or gaps in playout. Such losses cause the stream of frames to advance in time or cause a stream *lead*. Similarly, if a data frame is duplicated, it causes the stream to retard in time or a stream *lag*.

Because many streams are possible, we characterize both intermedia and real-time reference skew for k streams using a matrix representation as,

$$skew = \begin{bmatrix} 0 & sk_{1,2} & sk_{1,3} & sk_{1,k+1} \\ sk_{2,1} & 0 & sk_{2,3} & sk_{2,k+1} \\ sk_{3,1} & & 0 & sk_{3,k+1} \\ & & & \\ sk_{k+1,1} & sk_{k+1,2} & sk_{k+1,3} & 0 \end{bmatrix}$$

where $sk_{p,q}$ describes the skew from stream p to stream q (q to p is negative) and the $k+1$th element corresponds to a real-time reference. We also define a target skew matrix $\Theta_{p,q}$ (similar to Ravindran's divergence vector [61]) which indicates target values which can be interpreted by a skew control function. Related to skew is data *utilization* [42]. Utilization U describes the ratio of the actual presentation rate to the available delivery rate of a sequence of data. Frame drops will decrease utilization whereas duplicates will increase utilization from its nominal unit value.

We define three measures of synchronization. The first defines single event synchronization. The second is a generalization of the event synchronization, and the third is specific to periodic stream data characteristic of audio and video data and is more practical to implement. In each case, synchronization depends on multiple event occurrences including a reference event and a synchronized event.

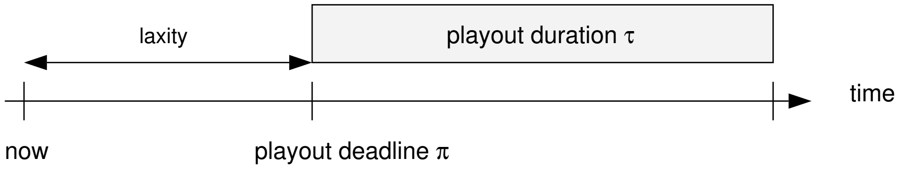

Figure 14: Single Event Synchronization

Definition 1 *A data item (object) with actual playout time ρ is synchronized with playout reference time π iff*

$$|\rho - \pi| \leq \theta$$

where θ is the synchronization tolerance. Note that the units and time reference are arbitrary as long as they are consistent.

Definition 2 *Definition: A composite object with actual playout times $P = \{\rho_i\}$ is synchronized with playout reference times $\Pi_i = \{\pi_i\}$ iff*

$$\forall i, |\rho_i - \pi_i| \leq \theta_i$$

where $\Theta = \{\theta_i\}$ *are the synchronization tolerances between each element and the reference.*

This definition is similar to Ravindran's divergence vector [61] and Gibbs' definition of synchronization [24].

Definition 3 *A continuous, periodic data stream comprised of* n *elements with actual playout times* $P = \{\rho_i\}$ *is synchronized with respect to a playout reference time* π *and period* τ *iff*

$$\forall i, 0 \leq i < n, |\rho_i - \pi - i(\tau + \epsilon)| \leq \theta$$

where θ *is the synchronization tolerance for the stream and* ϵ *is the clock drift tolerance.*

This definition allows for a drift in the actual playout times with respect to the reference. Furthermore, this tolerance can be described as a *maximum*, or a statistical value.

The synchronization tolerance can also be defined with respect to sequential and parallel events as shown in Fig. 15, where θ_p describes the timing tolerance between parallel envents and θ_s describes the tolerance between sequential events.

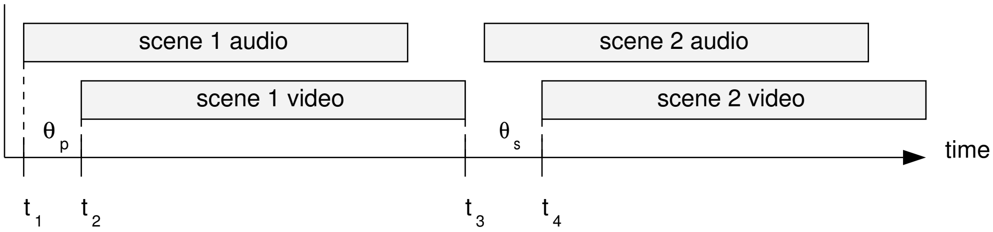

Figure 15: Synchronization Tolerance

In each definition of synchronization, the reference and actual playout times can be in any consistent frame of reference or coordinate system and therefore can be applied in conjunction with the aforementioned (linear) temporal transformations.

Definition 3 is intended for continuous audio and video for which *cumulative* error can be more significant than the relative error. The cumulative error is associated with the ϵ in the aforementioned definition. *Relative* timing refers to the θ in the previous definitions whereas *absolute* timing refers to the θ_i. For these data, sample rates and frame rates define the playout times for individual data elements. For aperiodic data, the simplicity of Definition 3 can be applied by conversion to a periodic stream [28] as illustrated in Fig. 6.

4.2 Data Structures for Temporal Representation

To to support a time-based representation, a MMIS must capture the representation using an appropriate data structure that is suitable for subsequent application and TAC functionality and is also appropriate for object evolution through object editing. Because multimedia data are large with respect to data storage, a data structure supporting a temporal representation must be suitable for efficient data retrieval via database indexing and query.

Few language or graph-based representation techniques specify appropriate data structures that support subsequent TAC operations on a database schema. One such approach provides a mapping from a specification methodology to a database schema using the TPN and the relational database model [46]. In this case, temporal intervals and relationships are described by a timeline representation in an unstructured format, or by a TPN in a structured format. Using a TPN, temporal hierarchy can be imparted to the conceptual schema as sets of intervals bound to a single temporal relation are identified and grouped. For example, this process is applied to the TPN of Fig. 11, resulting in the conceptual schema of Fig. 16.

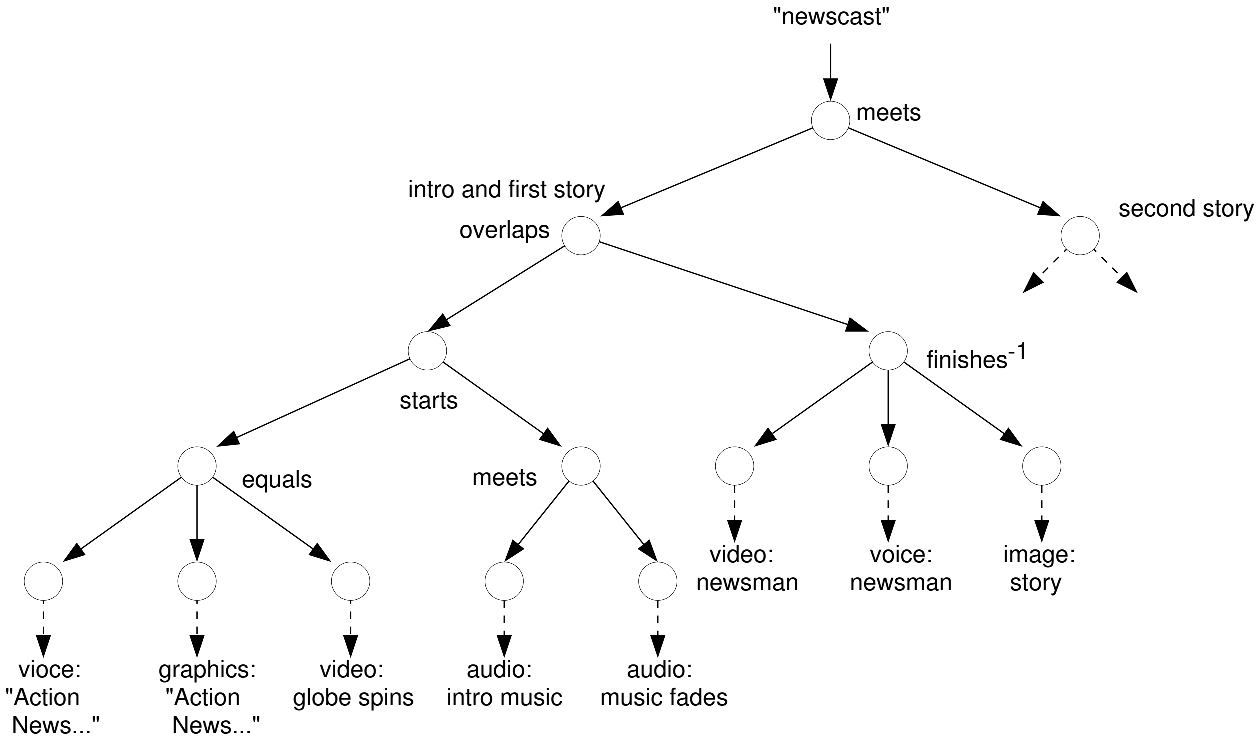

Figure 16: Temporal Hierarchy for the "Action News"

With this approach, the time-base representation can be translated to a conceptual schema in the form of a temporal hierarchy representing the semantics of the specification approach. Subsets or subtrees of this hierarchy represent subsets of the specification, illustrating the capability of composing complex multimedia presentations. Leaf elements in this model indicate base multimedia objects (audio, image, text, etc.), and additional attributes can be assigned to nodes in the hierarchy for conventional DBMS access. Timing information is also captured with node attributes, allowing the assembly of component elements during playout.

4.3 Physical Storage Models

Given a timing specification for a multimedia application, the physical system must meet these requirements. Problems arise due to the strict timing requirements for playout of time-dependent data.

The multimedia types of audio and video require very large amounts of storage space and will exist, when not live, in secondary storage. In order to meet the presentation requirements for these data, there are some obvious storage organizations to facilitate data transfer from secondary storage to display. For example, data can be stored in contiguous blocks on disk in the same order as playout. If disk transfer rates are not attainable for a certain data type, then disk interleaving can be used to produce the necessary data rates, as has been successfully implemented for a monochrome video-in-windows display [35]. When multiple streams originate from the same storage device, interleaving of data is necessary both to maintain data rates suitable for the quality of the stream, as well as to prevent conflict between the interacting streams. Approaches for the placement of audio and video data streams on a rotating-disc storage device has been investigated by Yu et al. [80, 83], Gemmell and Christodoulakis [23], and Rangan and Vin [59].

Some data placement schemes rely on maintaining a fixed transfer rate between disk and display. In the event that the system becomes busy with some other task, it is possible to corrupt the playout sequence, causing a perceptible shortage of data resulting in a blank screen or silence in video and audio output. An alternate approach proposed resolves this contention problem by providing a variable quality of data transfer if there is contention in the system [53, 25]. The key to this scheme is the storage (or generation) of data frames in such a manner as to provide high or low-resolution data retrieval of the same sequence.

4.4 Data Compression

Since multimedia data types have enormous storage and communications requirements, data compression is desirable, if not essential to enable multimedia applications. In this section, some of the implications of data compression on continuous-mode data are discussed.

Many data encoding formats exist for stream-type data. For most uncompressed formats we can delineate fixed data units, for example, video frames at 30 frames/s or audio samples at 3000 samples/s. Since these data are produced at regular intervals, they are called constant bit rate (CBR). Compression schemes for stream-type data can result in CBR or variable bit rates (VBR). The advantage of compressed data is clearly the savings in storage space and communication bandwidth. However, it becomes more difficult to identify points of synchronization between streams requiring synchronization when VBR compression is applied. The reason is as follows. Compression schemes use both intra and inter-frame coding. Intra-frame coding applies compression schemes within a single time-dependent frame. Therefore, a timing specification can apply to the self-contained frame before, during, and after compression. For inter-frame coding, compression schemes apply across a sequence of frames. For the proposed MPEG (Moving Picture Experts Group) coding scheme [38], differential values are generated for sequences of frames between inter-frame coded ones. This approach presents several problems for synchronization. First, it is desirable to have the ability to start at an arbitrary point in the continuous stream. With inter-frame coding this is not possible without first regenerating intermediate frames. Second, in order to provide reverse presentation, differential values must be available in both directions. The ability to begin presentation at an arbitrary point in a stream or to choose direction is part of the larger problem of providing random access or random insertion points into a stream-type object. These problems are approached in the MPEG scheme. To provide random insertion points, intra-frame coding is used at intervals as often as required for access as specified by the application. Reverse playout is accommodated by bidirectional differential frames.

4.5 System Support for Synchronization

For supporting time-dependent media, a MDBMS must deal with storage device latencies including ones due to data distribution across a network. MMIS support for time-dependent data requires an operating system that is tailored to the specific requirements of real-time multimedia data. Unlike hard real-time systems, the inability to meet a deadline for multimedia data is unlikely to cause a catastrophic result. However, the design of such a system must account for latencies in each system component in the delivery of data from storage to the user. Specific scheduling is required for storage devices, the CPU, and communications resources. Recent work in the design of systems support for multimedia data includes [17, 18, 44, 4, 2].

Similarly, providing a transport mechanism for time-dependent data requires managing the resources of a computer network. For delay-sensitive media, these resources are communication bandwidth and end-to-end delay. The problem of synchronization across a network is most acute when providing intermedia synchronization for multiple independent stored-data sources. In this case, to achieve intermedia synchronization, random network delays on each connection must be overcome, in spite of variations in clock rates at each remote data source. Typically, the delay variations on each channel are estimated during connection set-up, and an end-to-end delay, called a *control time* is introduced, representing an interval over which buffering is applied. The result of this buffering is a reshaping of the channel delay distribution to reduce variance.

4.6 Synchronization Anomalies

When data are delayed and are not available for playout, a synchronization anomaly occurs. At the output device, this can result in a gap in the sequence of presented elements, or a shortage of data to present per unit time. Policies for handling late-arriving data include discarding them or changing the playout rate to maintain a constant number of buffered elements. When data are lost or discarded, reconstruction can also be used. Steinmetz [68] proposes performing some alternate activity when a data element is not available, such as extending the playout time of the previous element. Generally, when gaps in a data sequence are ignored with respect to the playout rate, the loss of data elements when subsequent data are available advances the sequence in time, and can be corrected by slowing the playout rate until the schedule is correct. Approaches to synchronization of received packets include varying the playout rate and the utilization of received data [50]. The *expansion* method lets each packet be played out even if late. The result is the delay of all successive packets and an accumulation of skew with time. Another approach is to *ignore* some data since much redundant information is contained in the data streams, thereby preserving the duration of the overall sequence. This is analogous to a reduction in packet utilization [42]. One further gap-compensating technique for continuous media reconstructs the missing data elements. This approach is to substitute alternate data for the missing data in the stream. The data are chosen as null or non-null values (zero amplitude and waveform stuffing), or are interpolated from previous values [72].

5 Conclusion

We presented a review of temporal representations for time-dependent multimedia data, and their suitability for supporting data various delivery operations including presentation. Significant issues remain for providing time-dependent delivery of multimedia data in a general multimedia information system. The primary issues are specification and storage of temporal information describing the time dependencies of multimedia data, provision of an enforcement mechanism for temporal specifications, and accommodation of the latency in the retrieval of time-dependent data when they are not available for playout.

References

[1] Allen, J.F., “Maintaining Knowledge about Temporal Intervals,” *Comm. of the ACM*, November 1983, Vol. 26, No. 11, pp. 832-843.

[2] Anderson, D.P., Homsy, G., “A Continuous Media I/O Server and Its Synchronization Mechanism,” *Computer*, Vol 24, No. 10, October 1991, pp. 51-57.

[3] Anderson, D.P., Tzou, S.Y., Wahbe, R., Govindan, R., Andrews, M., “Support for Continuous Media in the Dash System,” *Proc. 10th Intl. Conf. on Distributed Computing Syst.*, Paris, France, May 1990, pp. 54-61.

[4] Anderson, D.P., Herrtwich, R.G., Schaefer, C., “SRP: A Resource Reservation Protocol for Guaranteed Performance Communication in the Internet,” *International Computer Science Institute Tech. Rept.* TR-90-006, February 1990.

[5] Anderson, T.L., “Modeling Time at the Conceptual Level,” *Improving Database Usability and Responsiveness*, P. Scheuermann, Ed., Academic Press, New York, 1982, pp. 273-297.

[6] Blakowski, G., J. Hubel, and U. Langrehr, “Tools for Specifying and Execution Synchronized Multimedia Presentations,” *Proc. 2nd Intl. Workshop on Network and Operating Support for Digital Audio and Video* (Lecture Notes in Computer Science 614, R.G. Herrtwich, Ed., Springer-Verlag), Heidelberg, Germany, November 1991, pp. 271-282.

[7] Bulterman, D.C.A. and R. van Liere, "Multimedia Synchronization and UNIX," *Proc. 2nd Intl. Workshop on Network and Operating Support for Digital Audio and Video* (Lecture Notes in Computer Science 614, R.G. Herrtwich, Ed., Springer-Verlag), Heidelberg, Germany, November 1991, pp. 108-119.

[8] Cheng, S., J. A. Stankovic, and K. Ramamritham, "Scheduling Algorithms for Hard-Real Time Systems - A Brief Survey," *Hard Real-Time Systems*, J. A. Stankovic and K. Ramamritham, Ed., IEEE Computer Society Press, Washington D.C., 1988, pp. 150-173.

[9] Christodoulakis, S., Faloutsos, C., "Design and Performance Considerations for an Optical Disk-based, Multimedia Object Server," *Computer*, December 1986, pp. 45-56.

[10] Committee Draft International Standard, "Information Technology Hypermedia/Time-based Structuring Language (HyTime)," ISO/IEC CD 10743, April 1, 1991.

[11] Committee Draft International Standard, "Information Technology – Standard Music Description Language (SMDL)," ISO/IEC CD 10743, April 1, 1991.

[12] Dannenberg, R.B., "Remote Access to Interactive Media," *Proc. SPIE Symposium OE/FIBERS'92, (Enabling Technologies for Multi-Media, Multi-Service Networks)*, Boston, MA, September 1992.

[13] Dannenberg, R.B., "Real-Time Scheduling and Computer Accompaniment," in *Current Directions in Computer Music Research*, M.V. Mathews & J.R. Pierce, Eds., MIT Press, Cambridge, MA, 1991.

[14] Dannenberg, R.B., McAvinney, and Rubine, "Arctic: A Functional Approach to Real-Time Control," *Computer Music Journal*, 10(4) (Winter 1986), pp. 67-78.

[15] Dannenberg, R.B. "Arctic: A Functional Language for Real-Time Control," in *Conference Record of the 1984 ACM Symposium on LISP and Functional Programming*, August 1984, pp. 96-103.

[16] Dannenberg, R.B. "Expressing Temporal Behavior Declaratively" CMU Tech Report (to appear).

[17] *Proc. 1st Intl. Workshop on Network and Operating Support for Digital Audio and Video*, Berkeley, CA, November 1990.

[18] *Proc. 2nd Intl. Workshop on Network and Operating Support for Digital Audio and Video*, Heidelberg, Germany, November 1991.

[19] Dimitrova, H., Golshani, F., “EVA: A Query Language for Multimedia Information Systems,” *Proc. 1992 Workshop on Multimedia Information Systems*, Tempe, Arizona, February, 1992, pp. 1-20.

[20] Faulk, S.R., Parnas, D.L., “On Synchronization in Hard-Real-Time Systems,” *Comm. of the ACM*, Vol. 31, No. 3, March 1988, pp. 274-287.

[21] Fujikawa, K., Shimojo, S., Matsuura, T., Nishio, S., Miyahara, H., ”Multimedia Presentation System “Harmony” with Temporal and Active Media,” *Proc. Summer 1991 Usenix Conf.*, Nashville, Tennessee, June 1991, pp. 75-93.

[22] Garcia-Luna-Aceves, J.J., Poggio, A., “Multimedia Message Content Protocols for Computer Mail,” *Proc. IFIP WG 6.5 Working Conf. on Computer-Based Message Services*, Nottingham England, May 1984, Hugh Smith, Ed., North-Holland, NY, pp. 87-98.

[23] Gemmell J., and S. Christodoulakis, “Principles of Delay-Sensitive Multimedia Data Storage and Retrieval,” *ACM Trans on Information Systems*, Vol. 10, No. 1, January 1992, pp. 51-90.

[24] Gibbs, S., L. Dami, D. Tsichritzis, “An Object-Oriented Framework for Multimedia Composition and Synchronisation,” in *Object Composition*, Tech. Rept., University of Geneva, June 1991, pp. 133-143.

[25] Gilge, M., Gussella, R., “Motion Video Coding for Packet-Switching Networks–An Integrated Approach,” *Proc. SPIE Conf. on Visual Communications and Image Processing*, Boston, MA, November 1991.

[26] Hamblin, C.L., “Instants and Intervals,” *Proc. of the 1st Conf. of the Intl. Society for the Study of Time*, J.T. Fraser, et al., Ed, Springer-Verlag, New York, 1972, pp. 324-331.

[27] Hehmann, B.B., Salmony, M.G., Stuttgen, H.J., “Transport Services for Multimedia Applications on Broadband Networks,” *Computer Communications*, Vol. 13, No. 4, May 1990, pp. 197-203.

[28] Herrtwich, R.G., “Time Capsules: An Abstraction for Access to Continuous-Media Data,” *Proc. 11th Real-Time Systems Symp.*, Lake Buena Vista, FL, December 1990, pp. 11-20.

[29] Herrtwich, R.G., Delgrossi, L., “ODA-Based Data Modeling in Multimedia Systems,” *International Computer Science Institute Tech. Rept.* TR-90-043, August 1990.

[30] Hoare, C.A.R., "Communicating Sequential Processes," *Communications of the ACM*, Vol. 21, No. 8, August 1978, pp. 666-677.

[31] Hodges, M.E., Sasnett, R.M., Ackerman, M.S., "A Construction Set for Multimedia Applications," *IEEE Software*, January 1989, pp. 37-43.

[32] Hodges, M.E., Sasnett, R.M., "Plastic Editors for Multimedia Documents," *Proc. Summer 1991 Usenix Conf.*, Nashville, Tennessee, June 1991, pp. 563.473.

[33] Hou, W.C., Ozsoyoglu, G., Taneja, B.K., "Processing Aggregate Relational Queries with Hard Time Constraints," *Proc. ACM SIGMOD Intl. Conf. on the Management of Data*, Portland, OR, June 1989.

[34] International Organization for Standardization, ISO Document No. 8613, ISO, Geneva, March 1988.

[35] Katseff, H.P., Gaglianello, R.D., London, T.B., Robinson, B.S., Swicker, D.B., "An Overview of the Liaison Network Multimedia Workstation," *Proc. 1st Intl. Workshop on Network and Operating Support for Digital Audio and Video*, Berkeley, CA, Nov. 1990.

[36] Kipp, N., V. Newcomb, S. Newcomb, "Hytime Review," TechnoTeacher Inc., 1990.

[37] Naylor, W.E., Kleinrock, L., "Stream Traffic Communication in Packet Switched Networks: Destination Buffering Considerations," *IEEE Trans. on Comm.*, Vol. COM-30, No. 12, December 1982, pp. 2527-2524.

[38] Le Gall, D., "MPEG: A Video Compression Standard for Multimedia Applications," *Comm. of the ACM*, Vol. 34, No. 4, April 1991, pp. 46-58.

[39] Li, L., Karmouch, A., Georganas, N.D., "Real-Time Synchronization Control in Multimedia Distributed Systems," *Proc. 4th IEEE ComSoc Intl. Workshop on Multimedia Communications (Multimedia '92)*, Monterey, CA, April 1992, pp. 294-305.

[40] Lippman, A., "Movie-Maps: An Application of the Optical Videodisc to Computer Graphics," *Computer Graphics*, July 1980, pp. 32-42.

[41] Little, T.D.C., Ghafoor, A., "Synchronization and Storage Models for Multimedia Objects," *IEEE J. on Selected Areas in Comm.*, April 1990, pp. 413-427.

[42] Little, T.D.C., Ghafoor, A., "Network Considerations for Distributed Multimedia Object Composition and Communication," *IEEE Network*, November 1990, pp. 32-49.

[43] Little, T.D.C., Ghafoor, A., "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks," *Computer* (Special Issue: Multimedia Information Systems), Vol. 24, No. 10, October 1991, pp. 42-50.

[44] Little, T.D.C., Ghafoor, A., "Multimedia Synchronization Protocols for Broadband Integrated Services," *IEEE Journal on Selected Areas in Communications* (Special Issue: Architectures and Protocols for Integrated Broadband Switching), Vol. 9, No. 9, December 1991, pp. 1368-1382.

[45] Little, T.D.C., Ghafoor, A., Chen, C.Y.R., Chang, C.S., Berra, P.B., "Multimedia Synchronization," *IEEE Data Engineering Bulletin*, Vol. 14, No. 3, September 1991, pp. 26-35.

[46] Little, T.D.C., Ghafoor, A., Chen, C.Y.R., "Conceptual Data Models for Time-Dependent Multimedia Data," *Proc. 1992 Workshop on Multimedia Information Systems (MMIS '92)*, Tempe, AZ, February 1992, pp. 86-110.

[47] Little, T.D.C., Ghafoor, A., "Scheduling of Bandwidth-Constrained Multimedia Traffic," *Computer Communications* (Special Issue: Multimedia Communications), Vol. 15, No. 5, July/August 1992, pp. 381-387.

[48] Little, T.D.C., Gibbon, J.F., "Management of Time-Dependent Multimedia Data," *Proc. SPIE Symposium OE/FIBERS'92, (Enabling Technologies for Multi-Media, Multi-Service Networks)*, Boston, MA, September 1992.

[49] Moore, D.J., "Multimedia Presentation Development Using the Audio Visual Connection," *IBM Systems J.*, Vol. 29, No. 4, 1990, pp. 494-508.

[50] Naylor, W.E., Kleinrock, L., "Stream Traffic Communication in Packet Switched Networks: Destination Buffering Considerations," *IEEE Trans. on Communications*, Vol. COM-30, No. 12, December 1982, pp. 2527- 2524.

[51] Newcomb, S.R., N. A. Kipp, V.T. Newcomb, "The "HyTime" Hypermedia/Time-based Document Structuring Language," *Comm. of the ACM*, Vol 34, No. 11, November 1991, pp. 67-83.

[52] Nicolaou, C. "An Architecture for Real-Time Multimedia Communication Systems," *IEEE J. Selected Areas in Comm.*, Vol. 8, No. 3, April 1990, pp. 391-400.

[53] Park, A., English, P., "A Variable Rate Strategy for Retrieving Audio Data from Secondary Storage," *Proc. 1st Intl. Conf. on Multimedia Information Systems '91*, Singapore, January 1991, pp. 135-146.

[54] Peterson, J.L., "Petri Nets," *Computing Surveys*, Vol. 9, No. 3, September 1977, pp. 225-252.

[55] Poggio, A., Garcia-Luna-Aceves, J.J., Craighill, E.J., Moran, D., Aguilar, L., Worthington, D., Hight, J., "CCWS: A Computer-Based Multimedia Information System," *Computer*, October 1985, pp. 92-103.

[56] Postel, J., "A Structured Format for Multimedia Documents," Report RFC767, DDN Network Information Center, SRI International, Menlo Park CA, 94025, USA, August 1980.

[57] Postel, J., Finn, G., Katz, A., and Reynolds, J., "An Experimental Multimedia Mail System," *ACM Trans. on Office Information Systems*, Vol. 6, No. 1, January 1988, pp. 63-81.

[58] Ramanathan, S., Rangan, P.V., Vin, H.M., Kaeppner, T., "Optimal Communications Architectures for Multimedia Conferencing in Distributed Systems," *Proc. 12th Intl. Conf. on Distributed Computing Systems (ICDCS)*, June, 1992.

[59] Rangan, P.V., Vin, H.M., "Designing File Systems for Digital Video and Audio," *Proc. 13th Symp. on Operating Systems Principles (SOSP'91), Operating Systems Review*, Vol 25, No. 5, October 1991, pp. 81-94.

[60] Rangan, P.V., Ramanathan, S., Vin, H.M., Kaeppner, T., "Media Synchronization in Distributed Multimedia File Systems," *Proc. 4th IEEE ComSoc Intl. Workshop on Multimedia Communications (Multimedia '92)*, Monterey, CA, April 1992, pp. 315-328.

[61] Ravindran, K., "Real-Time Synchronization of Multimedia Data Streams in High Speed Networks," *Proc. 1992 Workshop on Multimedia Information Systems*, Tempe, Arizona, February, 1992, pp. 164-188.

[62] Ravindran, K. and B. Prasad, "Communication Structures and Paradigms for Multimedia Conferencing Applications," *Tech. Rept.*, Dept. Computing and Information Sciences, Kansas State University, January 1991.

[63] Rennison, E., Baker, R., Kim, D.D., Lim, Y.H., "MuX: An X Co-Existent, Time-Based Multimedia I/O Server," *The X Resource: A Practical Journal of the X Window System*

(Proc. 6th Annual X technical Conf., Boston, MA, January 1992), Issue No. 1, Winter 1992, pp. 213-233.

[64] Sasnett, R.M., "Reconfigurable Video," *M.S. Thesis*, MIT, 1986.

[65] Schramm, C., Goldberg, M., "Multimedia Radiological Reports: Creation and Playback," *Proc. SPIE Medical Imaging III: Image Capture and Display*, SPIE Vol. 1091, Newport Beach CA, January 1989, pp. 191-201.

[66] SMPTE Recommended Practice: RP 136-1986, "Time and Control Codes for 24, 25, or 30 Frame-Per-Second Motion Picture Systems," April 1986, *SMPTE J.*, August 1986, pp. 862-864.

[67] Snodgrass, R., "The Temporal Query Language TQuel," *ACM Trans. on Database Systems*, Vol. 12, No. 2, June 1987, pp. 247-298.

[68] Steinmetz, R., "Synchronization Properties in Multimedia Systems," *IEEE J. Selected Areas in Comm.*, Vol. 8, No. 3, April 1990, pp. 401-412.

[69] Steinmetz, R., Meyer, T., "Multimedia Synchronization Techniques: Experiences Based on Different System Structures," *Proc. 4th IEEE ComSoc Intl. Workshop on Multimedia Communications (Multimedia '92)*, Monterey, CA, April 1992, pp. 306-314.

[70] Stotts, P.D., Furuta, R., "Petri-Net-Based Hypertext: Document Structure with Browsing Semantics," *ACM Trans. on Office Automation Systems*, Vol. 7, No. 1, Jan. 1989, pp. 3-29.

[71] Stotts, P.D., Furuta, R., "Temporal Hyperprogramming," *J. of Visual Languages and Computing*, Vol. 1, 1990, pp. 237-253.

[72] Suzuki, J., Taka, M., "Missing Packet Recovery Techniques for Low-Bit-Rate Coded Speech," *IEEE J. on Selected Areas in Comm.*, Vol. 7, No. 5, June 1989, pp. 707-717.

[73] Sventek, J.S., "An Architecture for Supporting Multi-Media Integration," *Proc. IEEE Comp. Soc. Office Automation Symposium*, April, 1987 pp. 46-56.

[74] Tansel, A.U., Arkun, M.E., Ozsoyoglu, G., "Time-by-Example Query Language for Historical Databases," *IEEE Trans. on Software Engineering*, Vol. 15, No. 4, April 1989, pp. 464-478.

[75] Toshiharu, T., Shinju, H., Takashi, O., "Multi-Media Visual Information Editing Techniques," *Globecom '87 (IEEE/IEICE Global Telecommunications Conference Record 1987)*, Tokyo, Japan, Nov. 1987, pp. 1192-1196.

[76] Tsichritzis, D., Ed., "Object Composition," *Tech. Rept.*, University of Geneva, June 1991.

[77] Tsichritzis, D., S. Gibbs, & L. Dami, "Active Media," in *Object Composition*, Tech. Rept., University of Geneva, June 1991, pp. 115-132.

[78] Vin, H.M., Zellweger, P.T., Swinehart, D.C., Rangan, P.V., "Multimedia Conferencing in the Etherphone Environment," *Computer*, Vol. No. October 1991, pp. 69-79.

[79] Weiss, K.H., "Formal Specification and Continuous Media," *Proc. 1st Intl. Workshop on Network and Operating Support for Digital Audio and Video*, Berkeley, CA, November 1990.

[80] Wells, J., Yang, Q., Yu, C., "Placement of Audio Data on Optical Disks," *Proc. 1st Intl. Conf. on Multimedia Information Systems '91*, Singapore, January 1991, pp. 123-134.

[81] Wirth, N., "Toward a Discipline of Real-Time Programming," *Comm. of the ACM*, Vol. 20, No. 8, August 1977, pp. 577-583.

[82] Yavatkar, R., "Issues of Coordination and Temporal Synchronization in Multimedia Communication," *Proc. 4th IEEE ComSoc Intl. Workshop on Multimedia Communications (Multimedia '92)*, Monterey, CA, April 1992, pp. 286-293.

[83] Yu, C., Sun, W., Bitton, D., Yang, Q., Bruno, R., Tullis, J., "Efficient Placement of Audio Data on Optical Disks for Real-Time Applications," *Comm. of the ACM*, Vol. 32, No. 7, July 1989, pp. 862-871.

[84] Zellweger, P.T., "Toward a Model for Active Multimedia Documents," in *Multimedia Interface Design*, M.M. Blattner and R.B. Dannenberg, Eds., ACM Press, New York, NY, 1992, pp. 39-52.