Multimedia Synchronization[1]

T.D.C. Little, A. Ghafoor†, C.Y.R. Chen, C.S. Chang, and P.B. Berra

Department of Electrical and Computer Engineering
Syracuse University
Syracuse, New York 13244 USA

† School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907 USA

Abstract – Multimedia synchronization has been recognized by many researchers as a significant requirement for applications using time-dependent media. The orchestration of static data elements such as images and text, and the "lip-sync" of audio and video are examples of such synchronization. In this paper, many aspects of multimedia synchronization are reviewed at the physical, service, and human interface levels of integration. Applicable areas include temporal modeling with intervals and abstractions, conceptual and physical models for databases, and systems support for synchronization including real-time scheduling and communications protocols.

[1] In *IEEE Data Engineering Bulletin*, Vol. 14, No. 3, September 1991, pp. 26-35.

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 movie, and can require time-ordered presentation during use. The task of coordinating such sequences is called multimedia synchronization. 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. Fig. 1 shows an example of time-ordered multimedia data for a sequence of text and images.

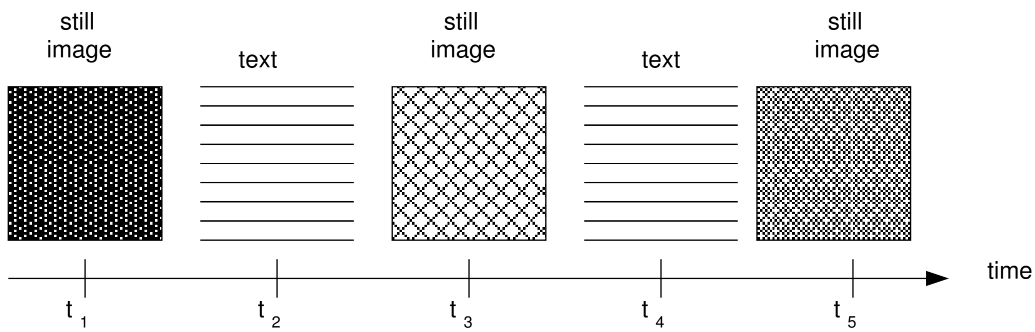

Figure 1: Time-Dependent Presentation

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 synchronization in the presence vastly different presentation requirements. Consider a multimedia slide presentation in which a series of verbal annotations coincides with a series of images. The presentation of the annotations and the slides is sequential. Points of synchronization correspond to the change of an image and the end of a verbal annotation, representing a coarse-grain synchronization between objects. A multimedia system must preserve the timing relationships among the elements of the object presentation at these points of synchronization by the process of temporal integration.

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 database management system (MDBMS). 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 paper, we describe the multimedia synchronization problem for time-dependent MDBMSs with respect to three levels. These are the *physical level*, the *service level*, and the *human interface level* [1]. 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 when data can be distributed.

The remainder of this paper is organized as follows. In Section 2, we define appropriate terminology for synchronization of time-dependent data. Section 3 surveys conceptual models for describing multimedia synchronization including temporal intervals, process models, user interaction, and temporal abstractions. Section 4 provides a examination of the database issues related to multimedia synchronization, including logical and physical models of temporal data, and systems support. Section 5 concludes the paper.

2 Classification of Time-Dependent Data

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. We must therefore develop a characterization of multimedia data based on the time dependency both at data capture, and at the time of presentation.

Time dependencies present at the time of data capture are called *natural* or *implied*. Audio and video recorded simultaneously have natural time dependencies. Data can also be captured as a sequence of units which possesses a natural ordering (e.g., see the medical example above). 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*. 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* [2, 3, 4]. This distinction is somewhat vague 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 [5]; 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.

3 Conceptual Models for Describing Multimedia Synchronization

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.

3.1 Modeling 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 [6, 7]. 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* [8], described as follows.

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 can be decoupled from isolated instants by specifying durations, leading us to temporal *relations*.

There are thirteen ways in which two intervals can relate in time [9], whether they overlap, abut, precede, etc. These relations are indicated graphically by a timeline representation shown in Fig. 2 [10]. Only seven of the thirteen relations are shown since the remainder are inverse relations.

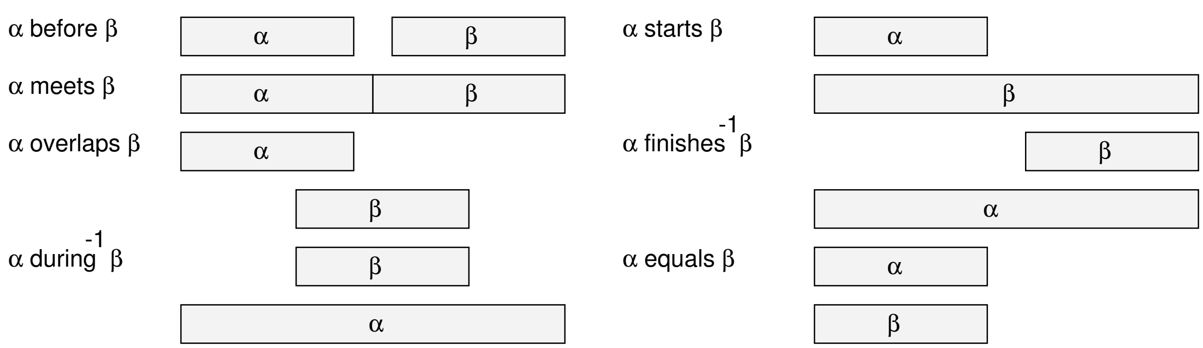

Figure 2: Temporal Relations

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. 3 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.

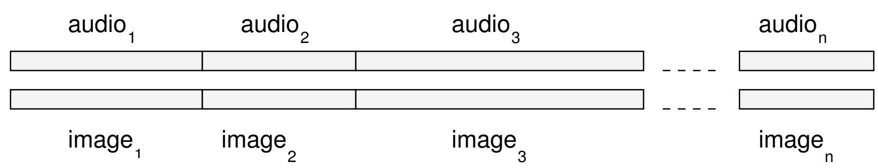

Figure 3: Synchronization of Audio and Images

With a temporal-interval-based (TIB) modeling scheme, complex timeline representations of multimedia object presentation can be delineated. 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.

Under some conditions, it may be desirable to introduce incomplete timing specifications using the TIB model, as can often arise when a time-dependent stream is to be played out in parallel with a static one. For example, if an audio segment is presented in synchrony with a single still picture, the time duration for image presentation would nominally be greater than or equal to that 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.

3.2 Process Synchronization

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 realtime systems with *hard* deadlines [11]. When such specification approaches are used, the presentation of a nondecomposable multimedia element is assigned to an executable process and the processes are indicated as synchronized via inherent language constructs. To be applicable to multimedia synchronization, these methods must allow synchronization on component precedence and on 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.

Although some language-based models satisfy these requirements, graphical models have the additional advantage of pictorially illustrating synchronization semantics, and are suitable for visual orchestration of multimedia presentations. A graph-based model satisfying many of the requirements listed above is the Petri net [12] which is both a graphical and mathematical modeling tool capable of representing concurrency.

3.3 Interaction and Synchronization

When a human user interacts with a multimedia system there is a requirement to synchronize the application with the user or 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 sensorinitiated 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 [13]. 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 [14] for interactive movies by using the hypertext paradigm [15]. The hypertext paradigm is a mode of information access and manipulation suitable for facilitating user-level interaction and database browsing.

The essence of hypertext is a nonlinear interconnection of information, unlike the sequential access of conventional text. Information is linked via cross-referencing between

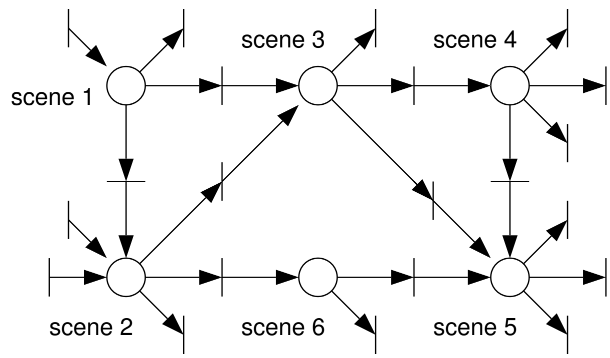

Figure 4: PNBH Petri Net

keywords or subjects to other fragments of information. One hypertext representation also uses Petri nets [16]. 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. 4 we show a PNBH network consisting of segments of the aforementioned interactive movie. These segments can be played-out in a random order, as selected by the user and restricted by the semantics of the net.

3.4 Temporal Abstractions

Some of the requirements for multimedia presentation 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, the 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 [5, 17, 18]. These describe the maintenance of a time reference that can be scaled to real-time and adjusted to appropriate playout speeds. 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 (see Fig. 5). For example, a unitless reference can be converted, or projected [18] 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.

We now describe existing methodologies for representation intermedia synchronization of time-dependent data.

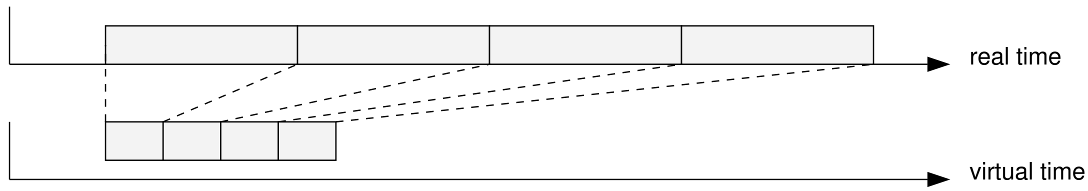

Figure 5: Projection of Virtual Time to Real Time

3.5 Existing Temporal Specification Methodologies

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 [19]. 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. Instant-based schemes have also been applied using MIDI (Musical Instrument Digital Interface) time instant specification [20] as well as via coupling each time code to a common time reference [21].

The 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 repre- sentation, with explicit capture of each of the thirteen temporal relations [22], or with additional operations to facilitate incomplete timing specification [2].

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 [3]. Various other language-based approaches have also been proposed (e.g., specification using LOTOS (Language Of Temporal Ordering Specification) [23], and process-oriented synchronization in CCWS [24]).

The OCPN (Object Composition Petri Net) [25] represents the only graph-based specification scheme that we are aware of. The particularly interesting features of this model are the ability to explicitly capture all of the temporal relations, and to provide simulation in both the forward and reverse directions. Each place in this Petri net derivative represents the playout of a multimedia object while transitions represent synchronization points. For example, the audio and image sequence of Fig. 3 can be represented by an OCPN and is shown in Fig. 6.

Unlike the OCPN, which is a form of *marked graph* [12], 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.

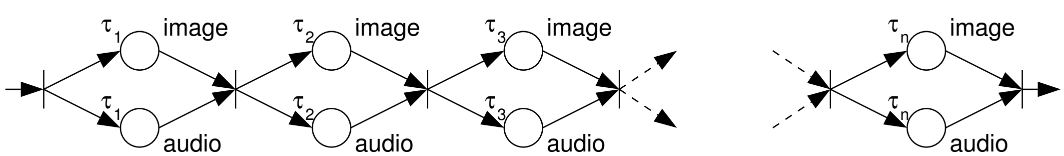

Figure 6: Image and Audio Synchronized with the OCPN

Standardization activities have resulted in several approaches to synchronization for electronic documents, including hypermedia [15]. For electronic document representation and interchange, the Office Document Architecture (ODA) has been standardized [26]. This standard describes parallel, sequential, and independent temporal control [27, 28] but does not support synchronization for continuous types. However, work is underway to extend the ODA model for this purpose [2]. HyTime (Hypermedia/Time-based Structuring Language) [18] and the Hytime application SMDL (Standard Music Description Language) [29] are language-based approaches to synchronization based on SGML (Standard Generalized Markup Language, ISO 8879). The HyTime specification provides a scripted form of language specification.

4 Database Aspects of Multimedia Synchronization

Once time-dependent data are effectively modeled, a MDBMS must have the capability for storing and accessing these data. This problem is distinct from historical databases, temporal query languages [6, 7], or time-critical query evaluation [30]. 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 Conceptual Storage Models

A conceptual data model for time-dependent data must support forward, reverse, and random access in addition to conventional DBMS queries. Temporal intervals can be described by a timeline representation in an unstructured format, or in a structured format such as the OCPN. Using the OCPN, temporal hierarchy can be imparted to the conceptual schema as sets of intervals bound to a single temporal relation can be identified and grouped. For example, this process is applied to the OCPN of Fig. 6, resulting in the conceptual schema of Fig. 7.

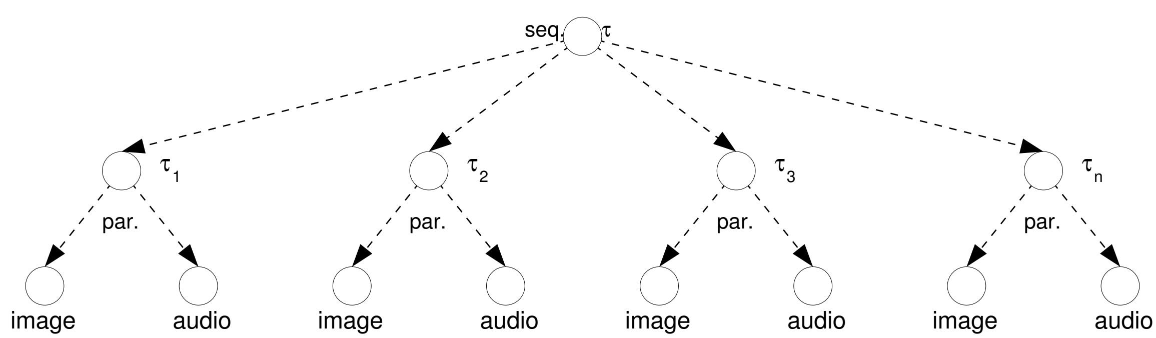

Figure 7: Conceptual Schema Based on an OCPN

With this approach, the conceptual schema forms a temporal hierarchy representing the semantics of the OCPN. Subsets or subtrees of this hierarchy represent subnets of the OCPN, illustrating the capability of composing complex multimedia presentations. Terminal 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. Temporal information can also be encapsulated in the description of the multimedia data using the objectoriented paradigm [5].

Temporal information including a time reference, playout time units, temporal relationships, and required time offsets are maintained for specific multimedia objects. For stream types, 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) analogous to a set of intervals bound to a single temporal relation.

4.2 Physical Storage Models

Given the conceptual synchronization requirements 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.

In this section, approaches to physical storage of time-dependent data are discussed. 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 (VW) display [31]. 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. An approach to the placement of two audio data streams on a disk is described by Yu et al. [32, 33].

These 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 [34]. The key to this scheme is the storage of data frames in such a manner as to provide high or low-resolution data retrieval of the same sequence.

4.3 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.

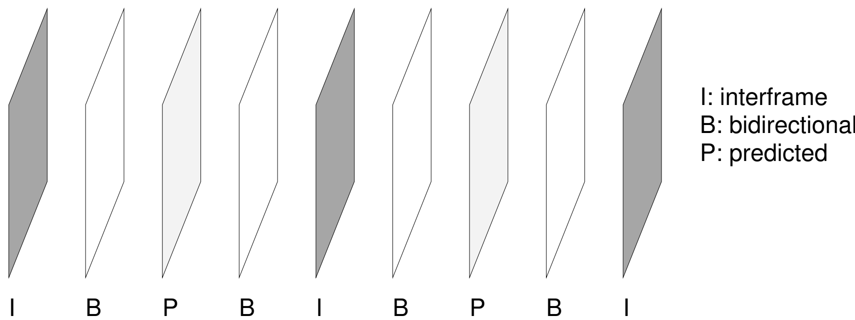

Figure 8: MPEG Coding Scheme

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 [35], 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 (see Fig. 8).

4.4 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. MDBMS 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 [36, 37, 38], the details of which are beyond the scope of this paper.

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-toend 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. This process is shown schematically in Fig. 9, where p(t) is the delay density function and w(t) is the reconstructed, playout density.

Recent proposals to provide continuous media transport in a network rely on a connection establishment phase that reserves sufficient resources to provide a multimedia connection [36, 37, 38]. These reservation protocols evaluate system resources including available bandwidth and channel delay properties. In the event that sufficient resources are not available, the connection is refused. Once a connection is established, buffering is configured to provide the requested level of service in terms of delay and late-arriving data elements. For storeddata

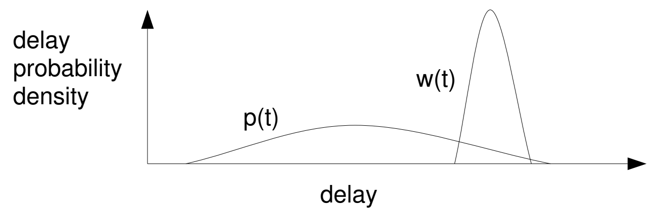

Figure 9: Result of Buffering

applications, the system has flexibility in scheduling the retrieval of data since the playout schedule is known *a priori*. For live multimedia sources, existing stream communication protocols can be applied. Since these protocols assume that some data can arrive late, the system must be able to accommodate shortages of data rather than introduce anomalous playout behavior.

4.5 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 [3] 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 [39]. 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 [40]. 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 [41].

5 Conclusion

In this paper, an overview of the many aspects of the multimedia synchronization problem are reviewed at the physical, service, and human interface levels of integration. Synchronization of this type has been recognized as an important requirement for enabling multimedia applications. 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 laxity in the retrieval of time-dependent data when they are not available for playout.

Acknowledgement This research was supported in part by the New York State Center for Advanced Technology in Computer Applications and Software Engineering (CASE) at Syracuse University, and the NYNEX Corporation.

References

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

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

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

[4] 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.

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

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

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

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

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

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

[15] Conklin, J., "Hypertext: An Introduction and Survey," *Computer*, September 1987, pp. 17-41.

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

[17] 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.

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

[19] 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.

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

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

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

[27] 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, New York, pp. 87-98.

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

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

[30] 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.

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

[32] 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.

[33] 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.

[34] 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.

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

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

[37] Little, T.D.C., Ghafoor, A., "Multimedia Synchronization Protocols for Broadband Integrated Services," to be published in *IEEE J. on Selected Areas in Comm.*, 1991.

[38] Anderson, D.P., Herttwich, R.G., Schaefer, C., "SRP: A Resource Reservation Protocol for GuaranteedPerformance Communication in the Internet," *International Computer Science Institute Tech. Rept.* TR-90-006, February 1990.

[39] 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.

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

[41] 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.