

Network Considerations for Distributed Multimedia Object Composition and Communication*

Thomas DC Little[†] and Arif Ghafoor[‡]

Syracuse University, Syracuse NY, 13244-1240

November 1, 1990

Abstract—A multimedia application must manage distributed data consisting of text, video, audio, and graphics, maintained in remote, heterogeneous databases. Problems of synchronization, real-time communication, and format conversion arise in this form of system to compose distributed data into complex multimedia objects for final interaction and presentation to a user. Inter-object relationships are specified in terms of a set of timing and spatial integration requirements which dictate the performance characteristics of a supporting distributed system. Described in this paper are the state-of-the-art techniques proposed for multimedia object communication and integration necessary to maintain their relationships and compose multimedia objects. Accordingly, an architecture for distributed multimedia object management and composition is presented which provides a framework for supporting future distributed multimedia information systems (DMIS).

Keywords: multimedia, object composition, distributed system.

*In *Network, IEEE*, Vol. 4, No. 6, November 1990, pp. 32-40.

[†]Thomas Little is now with Boston University, *tdcl@bu.edu*

[‡]Arif Ghafoor is now with Purdue University, *ghafoor@ecn.purdue.edu*

1 Introduction

Recent developments in high-speed communications technology have resulted in interesting new applications for Distributed Computing Systems (DCSs). A DCS generally comprises groups of workstations and shared Input/Output (I/O) devices interconnected by Local Area Networks (LANs). The advent of fiber optics and Broadband Integrated Services Digital Network (BISDN) extends the potential for these systems beyond the LAN environment by providing high-speed, low-bit-error-rate communications channels to remote sites and allowing switched voice and video transmission. One primary beneficiary of this technology is the Distributed Multimedia Information System (DMIS) [1]. Figure 1 describes the components of a DMIS, with the distribution of functions carried out by various system components.

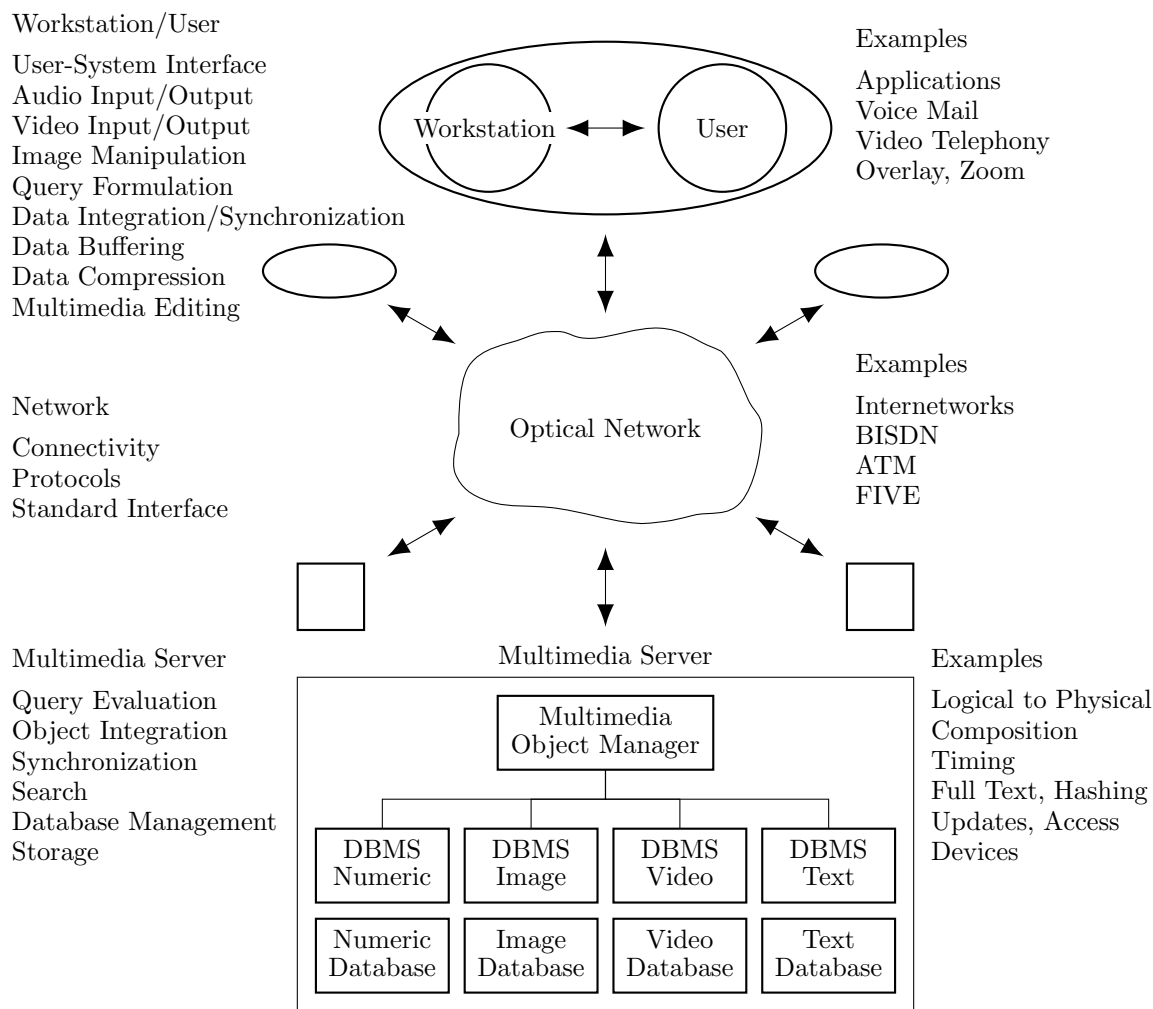


Figure 1: Distributed multimedia information system.

There are many potential multimedia applications. A typical one is the electronic catalog in which a consumer can browse through or query a set of listed catalog items by listening to audio descriptions, reading prices, and viewing video demonstrations of a merchant's products. Such service requires data of three types: audio, video, and numeric. Other

Table 1: Multimedia Applications and Characteristics.

Application	Media	Selected Functions
Office Automation	Images, Text, Spreadsheets, Mail	Composition, Filing, Communication
Medical Information Systems	Video (Telephony), Images, Text	Data Acquisition, Communication, Filing
Geography	Images, Graphics	Data Acquisition, Storage, Image Manipulation
Education/Training	Audio, Video, Images, Text	Browsing, Interactivity
Command and Control	Audio (Telephony), Images	Data Acquisition, Communication
Weather	Images, Numeric Data, Text	Data Acquisition, Simulation, Data Integration
Banking	Numeric Data, Text, Images	Image Archiving
Travel Agents	Audio, Video, Images, Text	Video Browsing, Communication
Advertising	Video, Images	Image Composition, Enhancement
Electronic Mail	Audio, Images, Text	Communication
Engineering, CAD/CAM	Numeric Data, Text	Cooperative Work
Consumer Electronic Catalogs	Audio, Video, Text	Video Browsing
Home Video Distribution	Audio, Video	Video Browsing
Real Estate	Audio, Video, Images, Text	Video Browsing, Communication
Library	Image, Text	Database Browsing, Query
Legal Information Systems	Image, Text	Database Query
Tourist Information	Audio, Video, Text	Video Browsing
Newsprint Publication	Image, Text	Image, Text Composition
Dictionaries	Image, Text	Database Browsing, Query
Electronic Collaboration	Audio, Video, Text	Videoconferencing, Concurrency Control, Communication
Air Traffic Control	Audio, Text, Graphics	Concurrency Control, Communication

multimedia applications exist in the areas of medicine, geography, and education. [Table 1](#) summarizes these application areas with their characteristics. Few of these applications use a single medium by itself without some interaction with other media; data of different media types are combined for use in presentation to the user. The process of combining data in this manner is commonly known as integration, composition, or fusion, and is shown pictorially in [Figure 2](#).

Composition generally takes spatial and temporal forms [2]. Temporal composition refers to synchronization of multiple streams of information consisting of objects of varying granularity [3]. These objects can be continuous types (video or audio), discrete types (images or text), or combinations of both. Spatial composition is concerned with the combination of objects in space, such as image overlay or text with image.

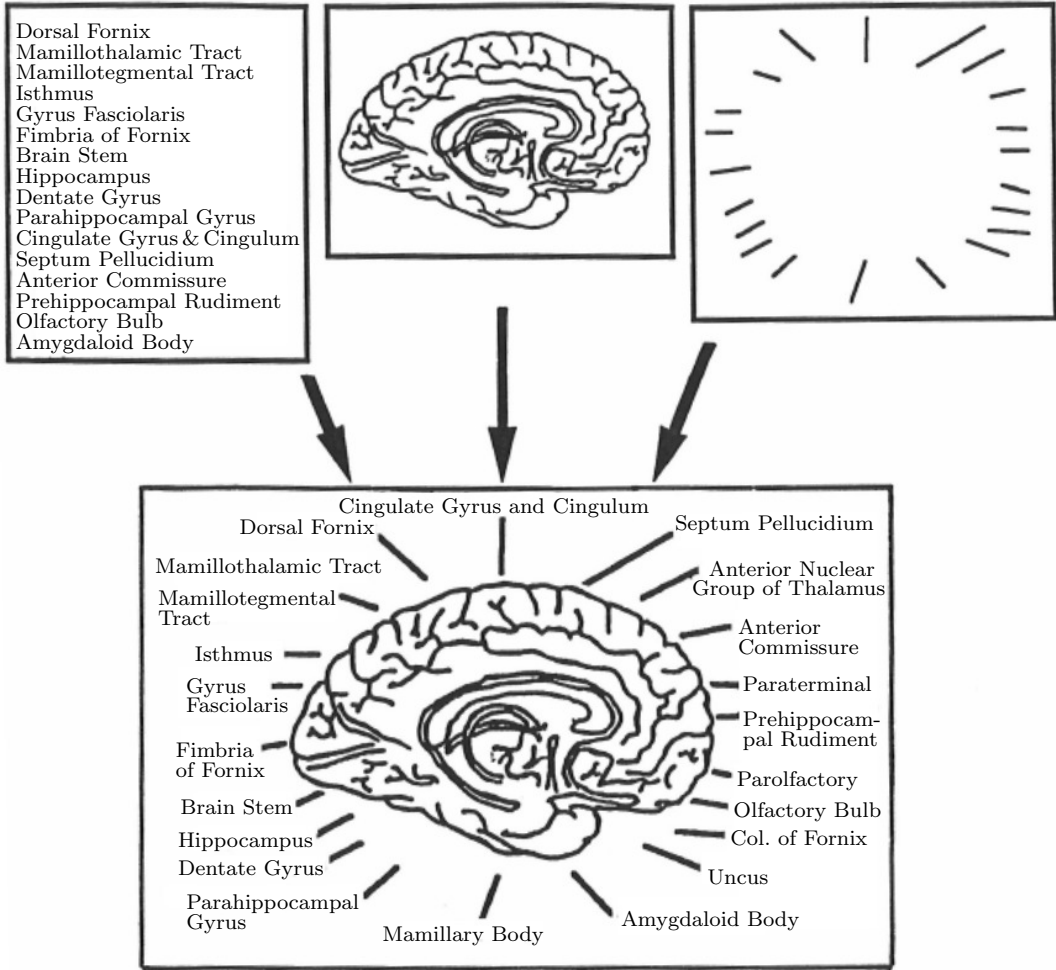
For most applications, such as medical information systems and interactive electronic collaboration (see [Table 1](#)), data are obtained from many dispersed sources. Data are created or fetched from individual locations or sources, and communicated to users at interactive terminals or sinks. In the future, an increasing amount of information will be provided by private and public database organizations that are geographically dispersed [4]. The composition process requires assembling these data based on both spatial and temporal constraints by some component of the system for ultimate presentation to the user(s), a process suitable for cooperating distributed multimedia database servers. The partitioning of this composition process on the set of distributed system resources is one of the issues addressed in this article. Other important topics discussed include requirements for distributed multimedia object composition, possible approaches to the composition problem, and assessment of future multimedia communication technology.

The remainder of this article is organized as follows: definition of the terminology and issues related to providing temporal data integration in a distributed environment, review of the current technological approaches to the issues and an assessment of the state of the art, and presentation of a unified model for synchronization at three levels, based on previously reviewed approaches. The article concludes with a course for future efforts.

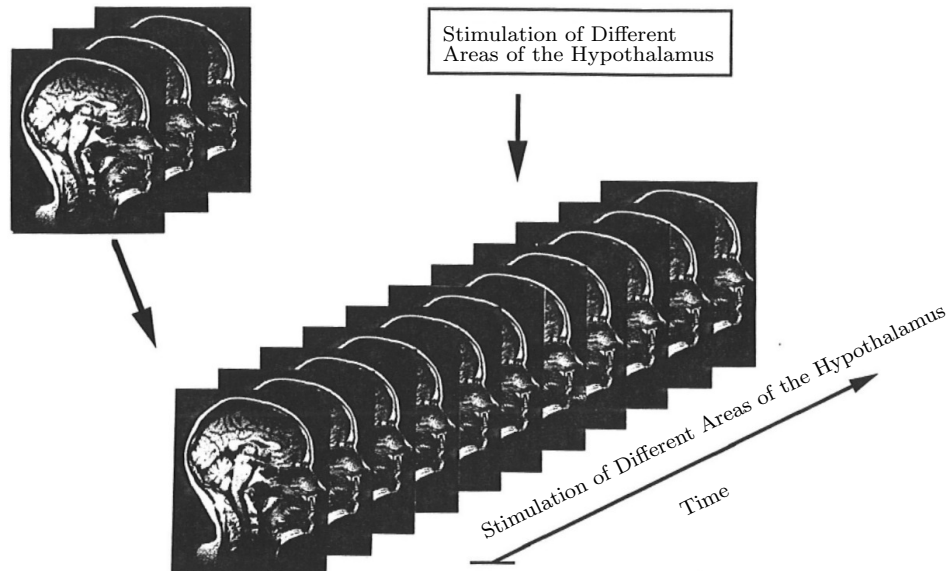
2 Object Composition and Integration

An “object” in this article is any unit of data, whether complex or simple, that can be distributed throughout a DMIS or, alternately, can be presented to a user in some desirable manner. This definition encompasses a spectrum of object type complexities, including composite multimedia documents, text-annotated images, simple numeric data, and encoded audio. [Table 2](#) [5] summarizes the units of information associated with the various media types. It can be noted that for each media type, multiple levels of decomposition result in different units of information; for example, the basic unit for the text type is the character, but increasingly complex objects of word, sentence, paragraph, and document can be composed from the atomic or base types.

The size of objects varies extensively with the application. Textual objects such as words, sentences, paragraphs, or pages comprise some of the smallest objects; color images form the largest. The size of objects can be unbounded. However, generally the largest objects are composed from the sequences of still image objects that make up color video. [Table 3](#) [6–9]




(a) Spatial Composition



(b) Temporal Composition

Figure 2: Spatial and temporal composition.

Table 2: Information Units for Various Media Types.

Medium	Increasing Complexity of Objects 				
	Atomic Unit	Intermediate Unit		Frame	Composite Object
Text	Character	Word		Sentence	Paragraph Document
Image	Pixel			Image	
Motion Video	Pixel	Raster		Image	Segment Film
Graphic	Vector			Polygon	Drawing
Audio	Sample	Phoneme	Word	Sentence	Paragraph Discourse

indicates typical uncompressed object sizes in bits per object for various object types taken from the application areas of Table 1. The sizes indicate a requirement for both massive data storage and high bandwidth communication for utilization of these objects in a DMIS. With suitable approaches, however, data compression can reduce these object sizes significantly. For still images, a compression ratio of 3 : 1 has been reported [8] without loss of information. Greater compression ratios, exceeding 12 : 1, are possible by using lossy approaches, without significant image degradation [8].

2.1 Multimedia Object Types: Persistent Versus Nonpersistent

Objects can be classified in terms of their presentation and application lifetimes. A “persistent” object is one that can exist for the duration of the application in a persistent store such as a database. A “nonpersistent” 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 created dynamically or retrieved from a database. Objects are “static” during presentation if they exist for an extended period for their possible manipulation. A still image is an example of a static object. Henceforth, we use the terms static and transient to describe presentation lifetimes of objects while persistence expresses their storage life in a database.

2.2 Multimedia Object Composition: Spatial Versus Temporal

Spatial composition involves assembling data based on overlaying or linking multiple objects into a single entity, as for example, the composition of textual and image information of Figure 2a. For such objects, the order in which the composition is performed is of no significance since there are no specific temporal relationships between the data elements and no transient data. Spatial composition techniques must consider size, rotation, and placement of participating objects. If an overlaying operation is performed, various merging operations can be used such as Exclusive-ORing of pixel elements. Performance of spatial composition assumes an underlying data model supporting the aggregation of objects. Standardization efforts have resulted in both logical and layout representations for multimedia documents described by the Office Document Architecture (ODA) [10]. These representations, as well

Table 3: Typical Uncompressed Object Sizes.

Application			Total	
Geography [6]	Multispectral Scan Image (corrected)	$3548 \text{ b} \times 2983 \text{ b} \times 6 \text{ b/pixel}$	64 Mb	
	Return-Beam Vidicon (corrected)	$5322 \text{ b} \times 5322 \text{ b} \times 6 \text{ b/pixel}$	170 Mb	
Medicine [7]	Digital Chest X-Ray	$1024 \text{ b} \times 1024 \text{ b} \times 12 \text{ b/pixel}$	13 Mb	
	Emission Computed Tomography	$128 \text{ b} \times 128 \text{ b} \times 16 \text{ b/pixel}$	260 kb	
	Nuclear Medicine	$256 \text{ b} \times 256 \text{ b} \times 16 \text{ b/pixel}$	1 Mb	
	Nuclear Magnetic Resonance	$512 \text{ b} \times 512 \text{ b} \times 16 \text{ b/pixel}$	4.2 Mb	
	Ultrasound	$512 \text{ b} \times 512 \text{ b} \times 8 \text{ b/pixel}$	2.1 Mb	
	X-Ray Computed Tomography	$512 \text{ b} \times 512 \text{ b} \times 14 \text{ b/pixel}$	3.7 Mb	
Videotelephony Stills [8]	Medium Resolution (b/w)	$512 \text{ b} \times 400 \text{ b} \times 8 \text{ b/pixel}$	1.6 Mb	
	Motion Video (30 f/s)	Medium Resolution (color)	$512 \text{ b} \times 400 \text{ b} \times 24 \text{ b/pixel}$	4.9 Mb
		High Resolution (color)	$1024 \text{ b} \times 1024 \text{ b} \times 24 \text{ b/pixel}$	25 Mb
		5 s Medium Resolution (b/w)	$1.6 \text{ Mb/f} \times 30 \text{ f/s} \times 5 \text{ s}$	240 Mb
		5 s Medium Resolution (color)	$4.9 \text{ Mb/f} \times 30 \text{ f/s} \times 5 \text{ s}$	735 Mb
		5 s High Resolution (color)	$25 \text{ Mb/f} \times 30 \text{ f/s} \times 5 \text{ s}$	3.8 Gb
Office Automation	VT100 ASCII Text Screen	$80 \text{ c/l} \times 241 \times 8 \text{ b/c}$	16 kb	
	8.5 in. \times 11 in. Page ASCII Text (Courier)	$66 \text{ c/l} \times 551 \times 8 \text{ b/c}$	29 kb	
	Scanned 8.5 in. \times 11 in. Page (b/w)	$8.5 \text{ in.} \times 11 \text{ in.} \times (300 \text{ pixel/in.})^2 \times 8 \text{ b/pixel}$	67 Mb	
	Scanned 8.5 in. \times 11 in. Page (color)	$8.5 \text{ in.} \times 11 \text{ in.} \times (300 \text{ pixel/in.})^2 \times 24 \text{ b/pixel}$	200 Mb	
	5 s Telephone Quality Audio	$7000 \text{ S/s} \times 5 \text{ s} \times 8 \text{ b/S}$	280 kb	
	5 s Stereo CD Quality Audio	$44 \text{ kS/s} \times 2 \text{ ch} \times 5 \text{ s} \times 16 \text{ b/S}$	7 Mb	
	5 s LPC Coded Voice [9]	$2.4 \text{ kb/s} \times 5 \text{ s}$	12 kb	

as other approaches [11, 12] develop a hierarchy of objects with object-oriented paradigm. Similarly, a methodology is defined for exchange of temporal relationships occurring within documents [13], including parallel, sequential, and independent relationships. This representation describes concurrent, serial, and arbitrary temporal relationships, respectively.

For temporal composition there exists a time ordering assigned to the presentation of the elements of the multimedia object. Consider a multimedia slide presentation in which a series of verbal annotations, or audio, must accompany a set of images presented. The presentation of the annotations is sequential, as it is for the images. Points of synchronization correspond to the change of image segment and the end of a verbal annotation, an example of a rather coarse synchronization between objects. A multimedia system must preserve the timing relationships between the elements of the object presentation at these points of synchronization by the process of temporal composition. Figure 2b indicates a pictorial representation of sequences of audio and video elements presented continuously as they are generated. For this kind of presentation, the video component consists of a sequence of frames, each displayed for 1/30 s to maintain the nominal video display rate of 30 frames/s. However, during the display of one video frame, the corresponding sequence of audio samples does not comprise a logical information unit for which we can apply synchronization points. Some other form of coordination is necessary to satisfy a tight synchronization requirement (e.g., lip synchronization) between the two streams during playback. Specific coarse synchronization points between these types of multimedia objects do not exist except at the beginning and end of the data sequences; rather, an ongoing form of synchronization is necessary. This type of temporal composition is called “continuous” [5], “stream”, or “isochronous” [14] synchronization.

2.3 Temporal Composition: Continuous Versus Synthetic

In this article, the problem of temporal object composition is specifically considered. Based on the multimedia application areas presented, two kind of temporal relationships are defined that can occur between objects. We call temporal relationships between objects that occur as streams “time-continuous” relationships, or simply “continuous” relationships. The relationship between audio and video streams generated by recording the image and voice of a speaker is continuous or isochronous [14] since these streams are produced, communicated, and presented as finite sample sizes and in fixed time intervals. Artificially created temporal relationships formed between objects that do not require continuous stream synchronization are called “synthetic relationships” [15]. Synchronized images and text for the composition of a multimedia display fit this category of temporal relationship. Synthetic relationship composition primarily indicates applications that rely on composition from stored, preorchestrated presentations with less stringent synchronization and delivery requirements, while continuous relationships are critical for real-time applications with strict needs for interobject coordination and communication. In the following sections, we elaborate on the distinctions of continuous and synthetic composition.

2.3.1 Time-Continuous Composition

Continuous temporal relationships occur among objects that are acquired simultaneously. The most illustrative example of a multimedia application exhibiting continuous composition

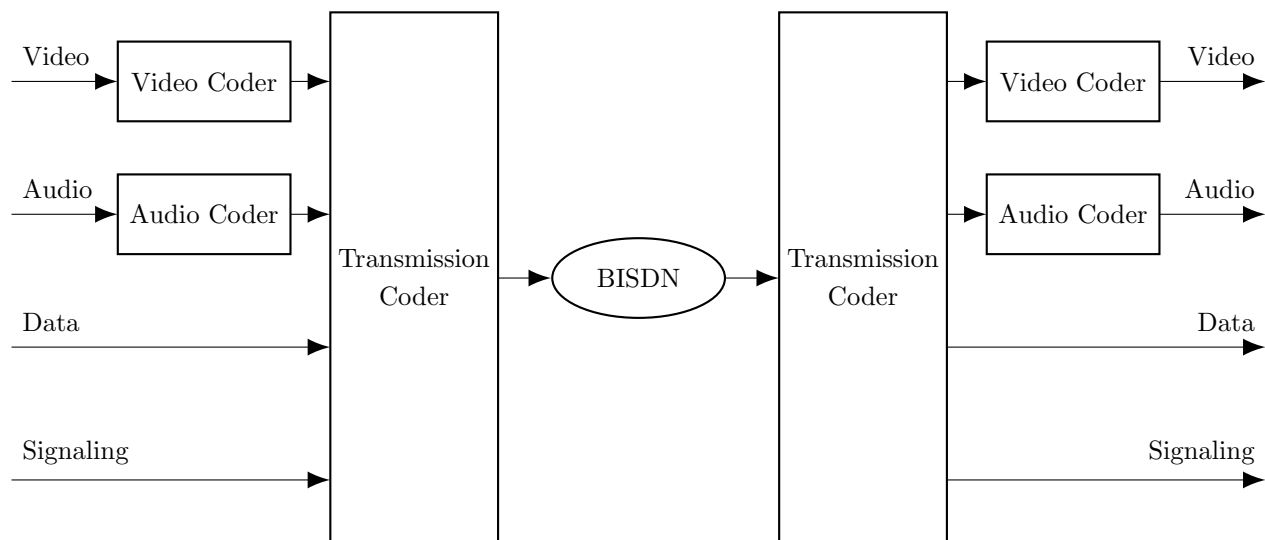


Figure 3: Visual Telephony.

is visual telephony, where audio and video are acquired, encoded, and transmitted between remote sites over a network. Figure 3 [16] indicates the components of a simple visual telephony system. Here, a transmitting station acquires data, signaling, and analog audio and video signals in real time and encodes and sends them via a high-speed network such as BISDN to a receiver, which performs the inverse function. To minimize communication cost, some form of compression and decompression is performed on the transmitted data. This can be variable bit rate or fixed bit rate, and intra- or interframe coded. Network bandwidth must be guaranteed to meet the variable rates associated with the video signal, with a hard bound on delay. Also, variations in packet arrival time (jitter) introduced by independent routing paths, packet loss, and packet buffering must be eliminated, via synchronization, by the network or receiver. A detailed discussion of the communication requirements for this type of composition is presented in the next section.

2.3.2 Synthetic Composition

Examples of composition based on synthetic temporal relationships occur whenever complex temporally related objects are created and stored in a database. Object storage is facilitated by a Database Management System (DBMS) capable of storing multimedia data (objects) and their attributes. In addition to maintaining the data comprising the objects, the DBMS must also maintain information necessary to perform temporal composition for both continuous and synthetic synchronizations. Specification of the temporal relationships can be modeled using the Object Composition Petri Net (OCPN) [2], which allows subsequent storage of coarse synchronization information at the object level (see Table 2). Figure 4 shows a hypothetical Medical Diagnosis and Teleconferencing System (MDTS) that employs synthetic relationships among component objects. In this example, it is assumed that the application requires synchronization of verbal utterances with video segments and diagnostic images, which constitutes a subset of the synthetic relationships and synchronization

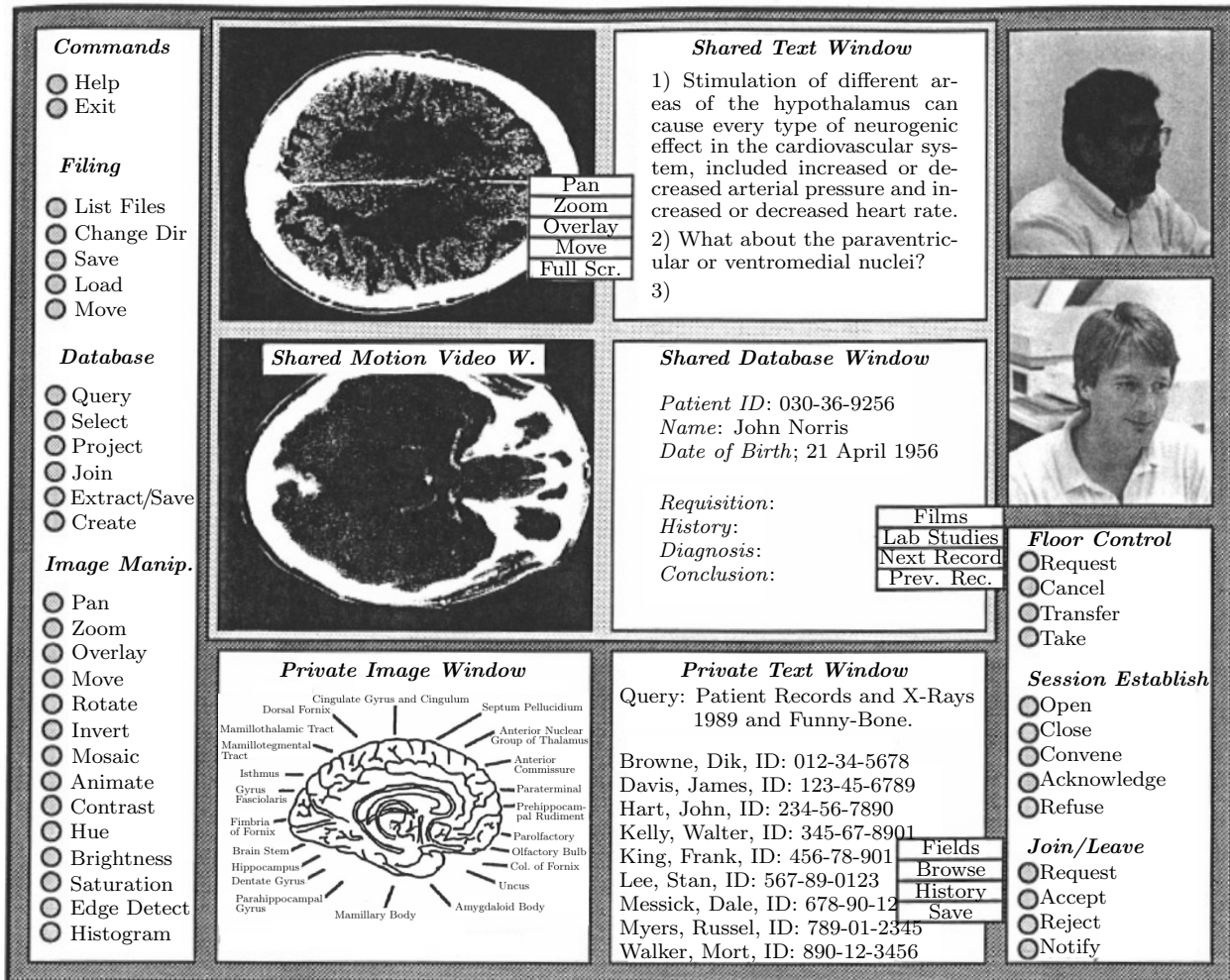


Figure 4: Medical diagnosis and teleconferencing system.

requirements. An OCPN for these objects captures the synthetic relationships and indicates synchronization points. For a given application and its data, by employing the OCPN, a database schema can be constructed for maintaining the object hierarchy and temporal relationships for storage and subsequent retrieval [2]. Figure 5 shows a segment of the OCPN for the MDTS. Parallel streams of audio and video data are synchronized at various points with text and image objects, as specified by the OCPN for this example.

Temporal composition of two objects can occur based on either sequential or parallel time relationships. There are thirteen ways in which two objects can relate in time [17]. However, it has been shown that seven relationships are sufficient to describe composition based on any temporal relationship between pairs of objects [2]. These seven relationships are shown in Figure 6a. The OCPN can capture the semantics of any of these temporal relations for the purpose of specifying timing and display requirements of various objects, as demonstrated in Figure 6b.

These relationships and OCPN models have been shown to be sufficient for specifying the temporal relationships of complex multimedia interactions constructed by pairing related

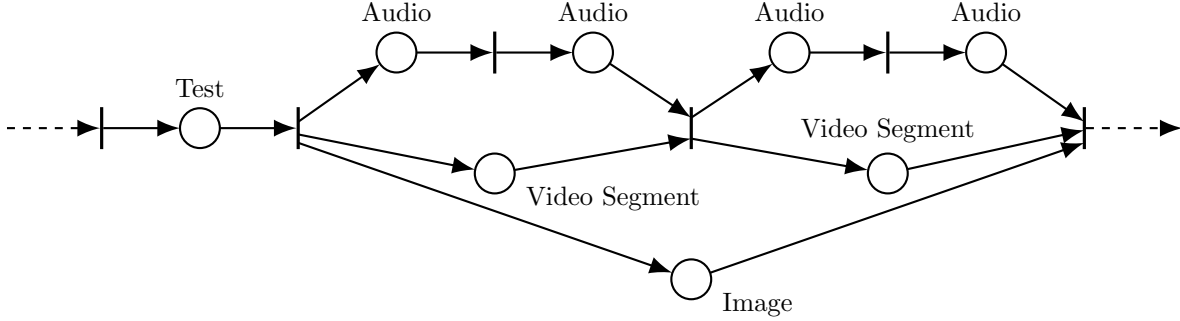


Figure 5: OCPN for subset of MDTs.

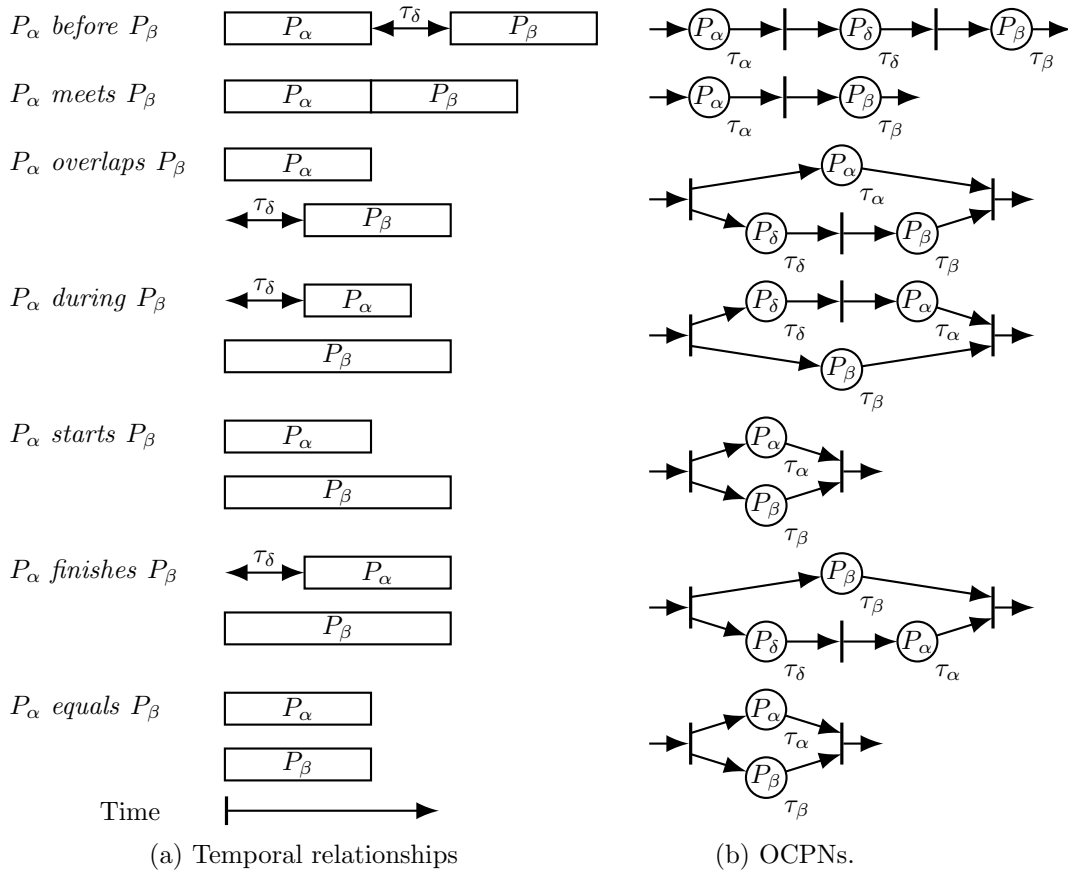


Figure 6: Temporal relationships and corresponding OCPNs.

objects [2].

2.4 Multiple Levels of Integration

There are three levels of integration defined for the provision of multimedia services in a DMIS [3]. These are defined as the “human-interface” level, the “service” level, and the “physical”

level. At the human interface level, integration consists of presentation and interaction with the user through multiple I/O devices. Service-level integration of multiple media describes the set of services offered by the DMIS to provide object composition functionality to multimedia applications. Interactions between objects and applications are achieved by an interface at the service level. Physical-level integration describes the consolidation of the data comprising multimedia objects onto communication and storage media. This integration consists of multiplexing for physical channels or clustering for physical storage.

The OCPN modeling technique describes the composition of multiple media for objects at the human interface and service levels of integration. To characterize the remaining levels required for a DMIS, we introduce “Quality Of Service” (QOS) to describe composition service and physical channel features.

2.5 Quality of Service for Object Composition

An important parameter for supporting distributed applications over a network is QOS, which characterizes communication services. For multimedia object communication, a QOS can be defined in terms of a tuple indicating target levels of performance. Specification of performance through this tuple influences the development of a DMIS. To develop this tuple some definitions are introduced.

We assume that those media which occur as continuous streams are transformed into sequences of discrete values to provide a uniform mechanism for intermedia synchronization. This assumption is consistent with the process of digitizing analog inputs such as audio and video. Single instances of discrete values or aggregations of these values can be considered objects. Whether continuous or synthetic relationships exist among synchronized entities, an end-to-end delay exists between the source and destination. This delay depends on the characteristics of the underlying network. A sampling delay is introduced in the digitization of analog data streams such as real-time video and voice. For stored objects, various overheads such as query evaluation, seek, and access delays in the storage devices are associated with object retrieval. Remote data access introduces communication delays due to network transmission, packetization, buffering, and depacketization. Finally, delay associated with presentation is incurred at the sink. Additional delay can be introduced due to compression and decompression of data at the source and sink, respectively. The relationship between these delays is summarized in [Figure 7](#).

The presentation rate of sequence of objects, such as video frames, is nominally equal to the rate at which they are recorded; however, this rate can be increased or decreased for certain circumstances. We define the presentation “speed ratio” for a sequence of objects to be the ratio of the actual to the nominal object presentation rate. Similarly, object “utilization” describes the ratio of the actual presentation rate to the available delivery rate of a sequence of objects. When utilization is equal to unity, all objects are presented. When it decreases, some objects can be discarded in order to maintain synchronization between two object streams. Ideally, both object speed ratio and utilization are equal to unity. However, there is a tradeoff between these parameters that depends on the policy established by the composition service. Degradation of one of these parameters can reduce system load, as discussed later. Object “skew” refers to the average difference in presentation times between two synchronized objects over n synchronization points at their synchronization instant. The

Table 4: Parameter Values.

Parameter	$t_0 - t_1$	t_1	$t_1 - t_2$	$t_2 - t_3$	$t_3 - t_4$	$t_4 - t_5$	$t_5 - t_6$
Speed Ratio	3/3	—	1/1	2/2	4/6	4/2	6/6
Utilization	1	—	1	1	1	1	1
Skew	0/3	—	1/1	-1/2	13/6	3/2	0/3
Jitter	—	1	—	—	—	—	—

instantaneous difference is called object “jitter”.

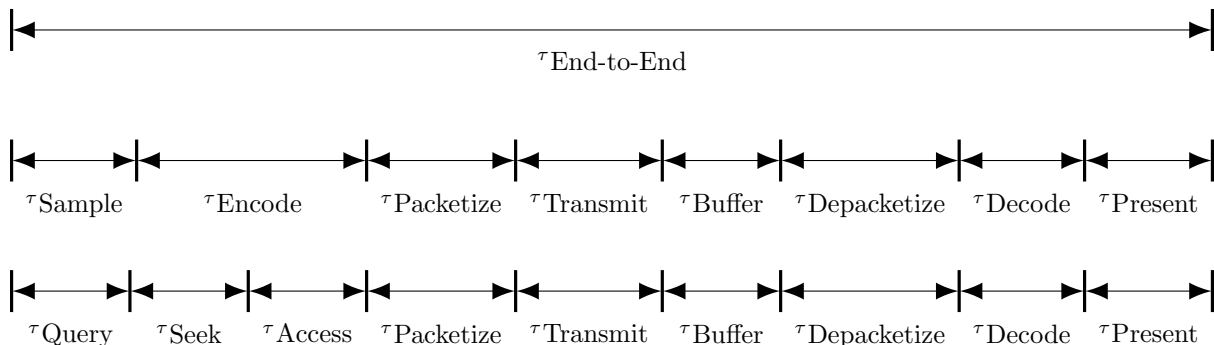


Figure 7: End-to-end delays.

In [Figure 8](#), the characteristics of object skew, jitter, utilization, and speed ratio are shown. If we assume that stream A represents the nominal presentation of a sequence of objects, then the characteristics of the presented stream, A' , can be determined. These characteristics are shown in [Table 4](#).

Another important communication parameter is the “reliability” of communication services, which can be viewed at multiple levels: per bit, frame, packet, channel, or connection. The reliability, expressed in terms of Bit Error Rate (BER) and Packet Error Rate (PER), represents the number of errors per time unit for bits and packets, respectively. The detected error rate of a communication channel is dependent on factors including the transmission medium, check-summing algorithm, and expected rate of packet loss from buffer overflows. The effect of packet and bit errors can have very different consequences, depending on the data transmitted. For example, an error in digitized voice can result in an audible click in a phone connection. For interframe coded video, lost bits or packets can interrupt image display for several seconds [\[18\]](#).

The level of error provision also impacts time performance. To provide an error-free service at any level of synchronization (bit, packet, etc.), error control protocols are required. These protocols must provide error detection and retransmission and/or correction, each reducing the transmission performance and ability to meet real-time performance specifications.

Based on these definitions, the QOS for multimedia communication can be defined as a tuple that includes speed ratio, utilization, average delay, maximum jitter, maximum BER, and maximum PER. This tuple can specify the requirements necessary to provide a multimedia service. For example, video telephony requires that voice and video be properly synchro-

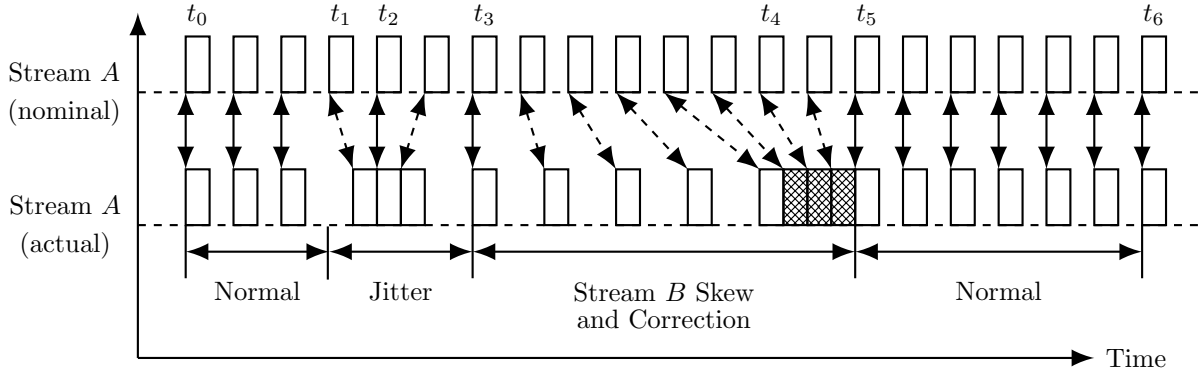


Figure 8: Skew, jitter, utilization and speed.

Table 5: Layers of Functionality.

Layer
User Interaction
Schedule Enforcement
Object Management
Network

nized with moderate reliability. For the video stream, this can be reflected through a QOS tuple with the following characteristics; speed ratio = 1.0, utilization = 1.0, average delay = 0.25 s, maximum jitter = 10 ms, maximum BER = 10^{-2} , and maximum PER = 10^{-3} [19].

3 Temporal Object Composition in a Distributed System

A high QOS for object composition in a DMIS places strong requirements on both hardware and software associated with the server, network, and workstation components. These requirements can be partitioned by several more or less independent layers of functionality, having different synchronization characteristics. The major layers of functionality of a DMIS system have been defined [2]. They include the user interface, scheduler for the object composition process, object manager, and network services. The relationship between these functional layers is shown in Table 5. Each layer provides services to the preceding layer as the hierarchy is traversed from top to bottom. The functional requirements of each layer with respect to temporal composition are discussed next.

3.1 User Interaction

Ultimate functionality and utility to the user depend on the transparency of data and communication access to the distributed object. The user should have, for example, the ability

to setup and use coordinated (synchronized) multimedia channels (for example, visual telephony) by simple means, and browse through sets of various objects within a database without regard to their location or media type. User interaction can be in the form of queries, sequential stopping and starting, spatial panning and zooming, animation, and various editing operations.

The user interaction layer is characterized as the most coarse level of synchronization in the DMIS. Synchronization points occur at invocations and terminations of application programs, object selection or query, pause operations, or any other user-object interaction. These synchronization events occur randomly, based on the user interaction, and cannot be predicted by the system. The synchronization required in this fashion has implications for the process of scheduling object presentation for concurrent real-time activities at the workstation.

3.2 Object Presentation Scheduler

The activities of multimedia object presentation to the end user are inherently concurrent, while the contemporary output devices (workstations) are not. Virtual, real-time concurrency is provided by multitasking for these devices and facilitated by a scheduler. With respect to synchronization, the scheduler must manage concurrent processes with real-time requirements. These requirements encompass both user interaction and the continuous and synthetic temporal relationships that can exist between objects.

The scheduler for the presentation of objects must have access to a schema specifying the temporal relationships of the objects currently undergoing presentation, the types of media, the desired QOS, and the location within the decentralized database of the selected objects and their components with respect to the display site. The various media types have different communication requirements (see “Network Layer for Distributed Multimedia Objects” below), which can vary depending on the desired QOS. From this information, decisions can be made to provide suitable capability for scheduling.

For objects with synthetic relationships, temporal relationships are known *a priori* by virtue of their database storage. With this information, the scheduling task is significantly simplified [15]. A similar argument is valid for scheduling synchronization for continuous objects. This type has predictable scheduling requirements. However, scheduling the presentation of objects influenced by intermittent activities introduced by the user is more difficult.

Scheduling for synchronization of objects is based on temporal constraints, including “deadline”, “minimum delay”, and “maximum delay.” Minimum delay defines the least amount of time before presentation of an object is to occur or between the presentation times of two objects. Similarly, maximum delay indicates the greatest amount of time before presentation is to be initiated or between the presentation instants of two objects. The time instant identified by the end of these intervals is the deadline. These temporal constraints can be derived from the QOS tuple supplied to the scheduler. An error condition results in the event that a constraint is violated (a schedule cannot be met due to the heavy load). A policy to solve this problem is to degrade the QOS for an object to satisfy the schedule. The degradation can be in the form of dropping frames (i.e., a reduction in object utilization) or decreasing the object speed ratio. The approach taken in [5] is to use “restricted blocking,”

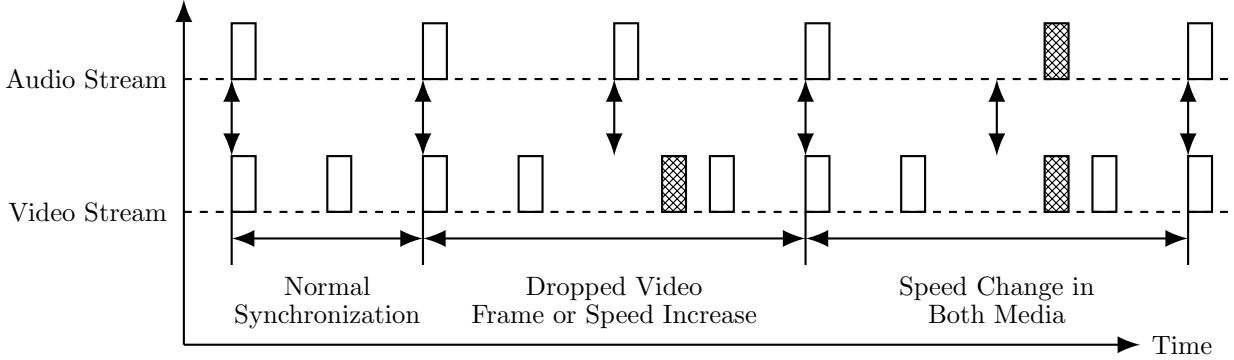


Figure 9: Synchronization anomalies.

whereby if during the presentation of two media (such as audio and video in Figure 9) a synchronization point is reached but one stream is delayed, the policy is to perform some appropriate alternative action while the other stream catches up. For the example, in Figure 9, the alternative is to hold the last frame of the video stream while the audio stream is presented normally. This practice ultimately results in an object skew, dropping of frames, or increased object speed for one of the data streams as the scheduler handles the burst of delayed frames when they arrive.

3.3 Object Management

A DMIS consists of data storage, processing, and communication components. Using the layered view of Table 5, the underlying communication system provides services to the upper layers of the DMIS. Such services must support unicast and multicast connections with remote sites, global naming of objects, concurrency control, resolution of heterogeneity, etc. Operations and interface primitives between layers must be defined to facilitate management and control of synchronization over established connections [14]. This control consists of establishing a QOS for a multimedia connection, and maintaining or modifying it for a connection. Maintenance of QOS can include adjusting the skew between synchronized data streams and informing the application of the state of synchronization.

Synthetic composition is intended to be performed at the level of complex stream-type or static objects. This level of the hierarchy represents the intermediate level of synchronization between stream synchronization and user or external event synchronization. Spatial registration, synchronization, and QOS information must be maintained for each object to perform composition. It is the role of the object manager to maintain this information, including the objects themselves.

To specify temporal relationships between two objects, four parameters must be specified and maintained as indicated by the synchronization tuple $(\tau_\alpha, \tau_\beta, \tau_\delta, T_R)$ [2]. τ_α and τ_β define the time durations of the individual objects. The delay between them is expressed in terms of τ_δ . Their temporal relationship is given by T_R (see Figures 6a and b). By combining objects in a pairwise fashion, a hierarchy of temporally related objects can be created and managed in a distributed multimedia database. The granularity of synchronization for synthetic composition is determined by the size of the objects in the hierarchy; however, this

technique is not intended for providing continuous relationship composition. It is designed for a coarse level of synchronization at the complex object level rather than at the frame level (see [Table 2](#)). An example of object synchronization is provided later.

3.4 Network Layer for Distributed Multimedia Objects

In a distributed system, the communication subsystem plays a central role in its organization and operation. The network is considered to provide basic services to the system as a whole. With respect to temporal object composition, this subsystem must be able to provide synchronization down to a fine level in what is called stream synchronization.

For multimedia data types, there is a unique set of requirements imposed on the communication component of a DMIS due to the size and characteristics of multimedia objects (see [Table 3](#)). For some types, delayed data are of little or no use for a multimedia application. Voice and video require real-time delivery, whereas text and images merely require timely delivery. With respect to transmission reliability, multimedia types also have differing requirements for PER and BER. Voice and video data can suffer errors in transmission without major degradation in service, depending on their coding algorithms. For data transfer (text and numeric), errors cannot be tolerated at the destination. Therefore, error detection, correction, and recovery schemes are used to provide reliable communication for these types. Error recovery protocols are sacrificed to provide real-time delivery performance for perishable objects while real-time delivery is traded off to provide reliability for numeric data.

A set of traffic classes can be identified by bandwidth requirement, delay distribution, and end-to-end packet loss. We will ignore the distinction between high- and low-bandwidth traffic since ample bandwidth is necessary for either case (e.g., interactive versus bulk traffic). Critical to the network component of a DMIS is the communication delay distribution and reliability. Three types of network performance characteristics [20] can be identified for types of multimedia objects:

- Delay-sensitive, nonblocking service—This service is used for communication requiring high data rates, short delay, small delay variation, and zero packet loss due to contention. Video telephony requires this kind of service since video data is transient in nature and therefore perishable. Although this service provides no loss due to blocking, it does not guarantee freedom from packet errors.
- Delay-sensitive, blocking service—This service is used for communication requiring high data rates and can tolerate moderate delay, jitter, and blocking. This service has a relaxed delay distribution and a nonzero probability of lost packets, but with a bounded value for the ones lost consecutively. Voice and video traffic can use this kind of service. Bulk information transport that can tolerate errors is provided by this service without a requirement for minimum delay.
- Delay-insensitive, error-free service—This service is used to provide error-free transfer of data provided by a retransmission policy for blocked, discarded, or erroneous packets. Traffic of this class is guaranteed zero end-to-end errors and specified by a minimum

average throughput and maximum average delay. This service is intended for any traffic requiring error-free service, such as file transfers.

These three classes characterize most traffic possible in a DMIS. However, it is intended that QOS be dynamic according to the requirements of the applications. Communication protocols need to be provided with a spectrum of reliability levels and real-time characteristics to support a wide range of QOS levels indicated by each media type and dictated by individual multimedia applications. In the following sections, we review techniques to provide synchronization or maintenance of delay variation as dictated by QOS.

4 Technological Assessment: Multimedia Services

In this section, we discuss solutions to the problems of object synchronization at three levels and present a model for distributed object composition. This model consists of a series of layers representing object composition protocols for imposing various types of synchronizations as proposed in [2, 14, 21]. These levels of synchronization deal with user interaction, object composition and management, and stream synchronization at the network layer (see Figure 9). They are described below.

4.1 Low-Level Synchronization Protocols

At a very low level, synchronization of multimedia data requires synchronization of data streams with very tight skew, jitter, and delay requirements (we defer discussion of reliability). Audio and video have these requirements. The low level must deal with methods of maintaining synchronization of data streams from disks, communication channels, and real-time inputs.

A proposed approach to the provision of a flexible communication mechanism for variable QOS uses variable-bandwidth channels provided by packet switching schemes [22, 23]. Rather than using fixed, dedicated, low-bandwidth (circuit-switched) channels with independent signaling channels, a high-bandwidth Virtual Circuit (VC) can be used to support different data transfer rates. BISDN is proposed to support data transfer operations necessary for distributed multimedia applications such as transmission of audio, video, and data. The proposed transport mechanism for BISDN, called Asynchronous Transfer Mode (ATM), seeks to provide [24]:

- A single network interface to communication channels for each media type: audio, video, image, and text
- Adaptability of an application's bandwidth requirements
- Flexibility for handling different data types
- Common signaling structure

The ATM transport technique uses a multiplexing scheme in which data is organized into units called "cells." Each fixed-length cell contains a header, which includes media access,

connection, and priority control information as well as a data field. Cell length has not been standardized but is proposed to be less than 100 octets. Calls are associated with cells by the header contents; therefore, no channel bandwidth is occupied in a VC except during the actual transmission of an ATM cell. The implication of this multiplexing strategy is that dynamic bandwidth allocation is possible as applications require varying communication performance dictated by differing data types or QOS. Each media type can utilize the same transport mechanism irrespective of the data type's bandwidth requirements.

By using a transport mechanism like ATM, anomalies occur that make synchronization at the receiver difficult. These include packet loss, delay due to buffering and independent communication paths, and clock variation at the sender and receiver. Several approaches have been proposed to resolve these problems of synchronization at the stream level as well as the object level.

In [21], the approach to stream synchronization is to establish a single "Multimedia Virtual Circuit" (MVC), whereby synchronized media are multiplexed onto a single VC with an inherent variable bandwidth. A similar approach is proposed by [3]. The rationale for this approach is twofold. First, since a single VC is used, the network guarantees that packets are received in the same order in which they are sent without incurring delay variations (jitter) due to multiple VC connections. Transmission of the multiplexed media is serialized. Second, by using a single VC, connection establishment can be performed prior to data transfer without the requirement for additional signaling channels as in fixed-bandwidth circuits.

An MVC can be viewed as a broadband digital pipe that consists of multiple channels, one per medium or data stream, as shown in Figure 11. The exact number and characteristic (QOS) of the channels an MVC comprises depend on the application's requirement and request at connection setup. To support connection establishment and control, interfaces between the network and transport layers and the transport and session layers provide functions of "create," "listen," "modify," and "clear," and "create," "add," "modify," and "delete," respectively. These allow connection management, including specification of service class, priority, flow control, or other QOS parameters.

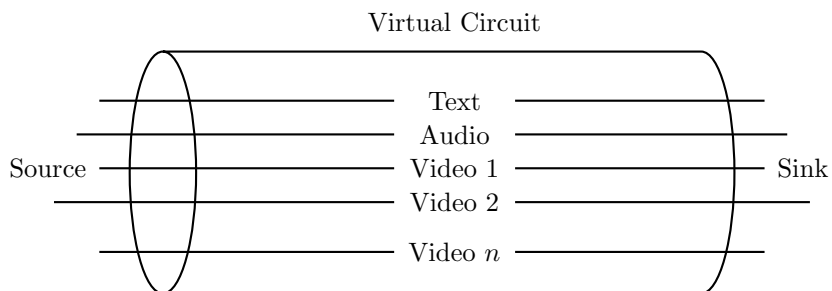


Figure 10: Multimedia virtual circuit.

Priority and service class describe the tolerance to delay and to peak and the average bandwidth requirement. As channels are needed in a connection, the "add" functionality allows additional media to be multiplexed over the same MVC with desired QOS.

For this scheme, in order to provide synchronization at the receiver, it is assumed that

packets arrive in sequence. Two levels of packet multiplexing and demultiplexing are defined in [21]. At the sender, streams of data (packets) from multiple channels of a multimedia application are multiplexed onto a single MVC. Streams of packets from multiple MVCs are then multiplexed onto the network for transmission. The inverse operation is performed at the receiver. Different media of an application possess different delivery requirements, which can be represented by a priority assignment. To maintain temporal synchronization within an MVC, all channel priorities are raised to the level of the media with the highest priority. MVCs with the same priority are called a “Virtual Circuit Group” (VCG). The approach to ensure synchronization requires that first, transmission priority is given to packets of the MVC in the VCG with the highest priority, and second, a round-robin scheme is used for packets within a VCG. This scheme has some limitations [14]. First, by promoting the priorities of media channels within an MVC, inefficiencies result in the form of lost bandwidth. For example, the QOS for a text data channel can require error recovery, while this is not necessary for an audio channel. If both require the same priority and error protocol, unnecessary handshaking is performed. Second, the MVC model assumes point-to-point connections, with no provision to support applications requiring multiple sources or destinations, such as shared windows, teleconferencing, and distributed objects.

We can identify four types of connections for object retrieval and composition in a DMIS, each shown in Figure 12: single source to single destination, single source to multiple destinations, multiple sources to single destination, and multiple sources to multiple destinations.

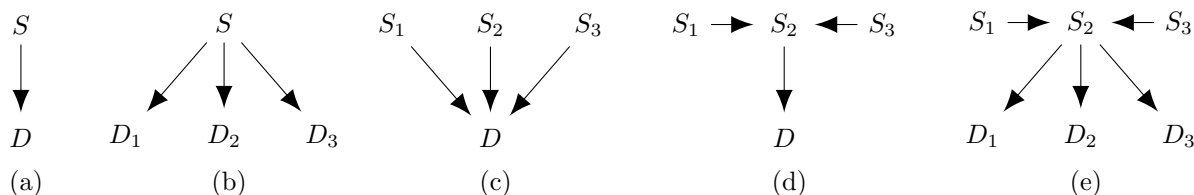


Figure 11: Multimedia connections in a DMIS (S: Source, D: Destination).

The first case is a point-to-point connection for which a master-slave relationship exists between a single multimedia server and a sink. A single MVC can serve the purpose of this connection. The second case represents a shared object environment where a single object is displayed simultaneously to various users. This mode is necessary for the collaborative or teleconferencing environment and requires concurrency and consistency control mechanisms. The MVC approach can handle this case via multicasting or establishing channels for each destination, under a single controller as the source. The third case represents a distributed object environment, for which complete composition is performed at the sink. This case is also handled by independent MVC channels, assuming that there are no dependencies between data from the different sources. If there are data dependencies between the channels, the MVC model is not applicable. However, by using an intermediate site, data dependencies can be resolved and an independent connection established with the single destination. Figure 12d, the fourth case, shows a scenario in which dependent objects are assembled from distributed sites and sent via a single MVC to the final destination. The fifth case defines the general case of distributed object composition in a shared object environment.

In [14], it is proposed that continuous data streams be exchanged on independent channels for each media, rather than on a multiplexed MVC. Synchronization of related media is performed at coarse intervals, with the requirement that short-term synchronization drifts (skew) be kept small. This is based on the assumption that bounded skew can be provided for a short period of time between temporally related data streams using appropriate buffering. At coarse synchronization points, the rate skew change and the instantaneous jitter are evaluated to provide corrective resynchronization via feedback to the buffering scheme. Frames in this context are called “Physical Synchronization Frames” (PSFs) [14] and refer to a physical unit of communication, such as the packet. A more coarse grain of synchronization is described by “Logical Synchronization Frames” (LSF). The relationship between PSFs and LSFs is shown in Figure 13. These frames indicate discernible units of information (see Table 2), which comprise sequences of PSFs. The PSFs are for synchronization within the communications mechanism, and the LSFs are for the controlling application. Furthermore, by specifying the relationship between LSFs and PSFs, a degree of the QOS can be indicated for synchronization. A one-to-one relationship implies tight synchronization at the control level, while an n -to-one relationship indicates control-level resynchronization every n PSFs.

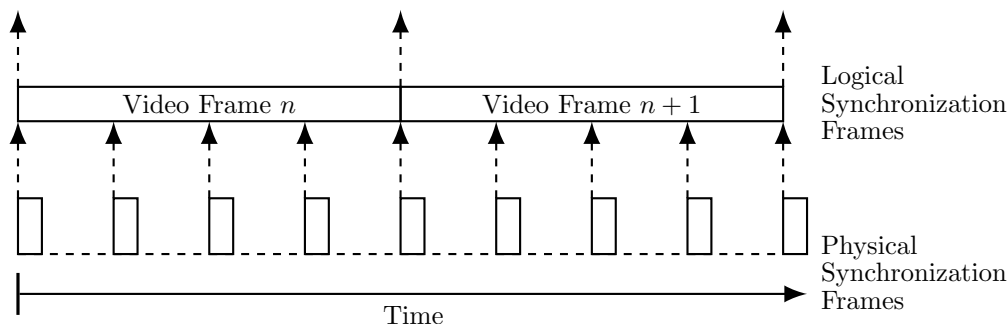


Figure 12: Physical and logical synchronization frames.

4.2 High-Level Synchronization Protocols

At the level of object composition and management, synchronization points are specified by LSFs [14]. These units represent the finest units that an application can manipulate; that is, stoppage of the presentation of a data stream in the middle of an LSF can result in the presentation of the remaining PSFs to the end of the LSF. For representation of the synchronization beyond the low level, object synchronization using an OCPN can be applied.

At the level of complex multimedia objects representing orchestrations of independent media, temporal relationships must be established and maintained. Object modeling is required to identify classes of related presentation elements for the development of a logical database schema and their subsequent storage in a database. For instance, the multimedia objects associated with a conference call or a multimedia document must be classified for these applications. In Figure 14, a hierarchical ordering is assigned to subobjects making up a full-motion video (film) object. Here, an object hierarchy is defined to which object

synchronization can be applied. Stream synchronization can be assigned to a low level with units such as frames, rasters, or pixels.

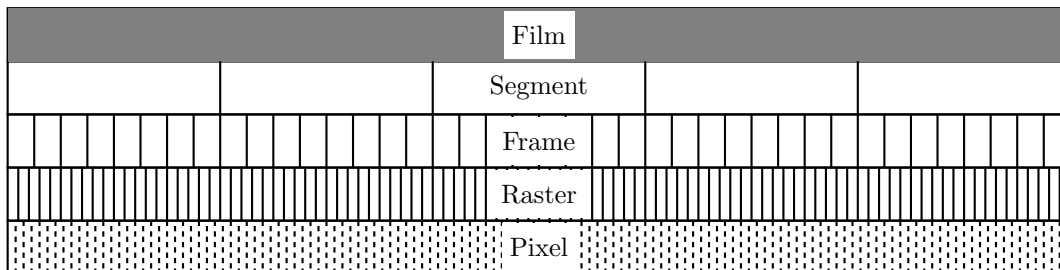


Figure 13: Hierarchical object levels.

Provided with object classes, individual relationships between independent media can be established using the OCPN modeling tool. From the OCPN, a synchronization schema is created, representing the required synchronization at the object level. The relationship between objects and LSFs is dependent on the degree of synchronization the application desires. Synchronization at this high level is facilitated by interacting processes dedicated to presenting individual objects, scheduled by a suitable policy based on QOS. An algorithm proposed in [2] can be used to retrieve objects from database storage and present them based on their temporal relationships. The main features of this algorithm are:

- It identifies temporal relationships from a database.
- It creates process threads for component objects (subobjects) with concurrent temporal relationships.
- It supports arbitrarily nested complex objects with recursion.
- It defines synchronization points at the start and termination of component object presentation as defined by the OCPN for the object hierarchy.

In the following section, an example of a multilevel synchronization scheme is presented.

4.3 Integrated Object Synchronization Model

Based on Table 5 and the description of synchronization protocols given in the previous section, a multilayered model for synchronization is proposed. This model unifies the currently known synchronization approaches at different levels of object composition. The model can be mapped onto the Open Systems Interconnection (OSI) Reference Model, as shown in Figure 14.

The model consists of four layers corresponding to specific layers of the OSI Reference Model. These layers are the “application,” “window manager,” “object composition,” and “transport support” layers. The application and window management layers, corresponding to the OSI application layer, provide the total functionality of the Table 5 user interaction level. The multimedia application is expected to exist within a graphical user interface with

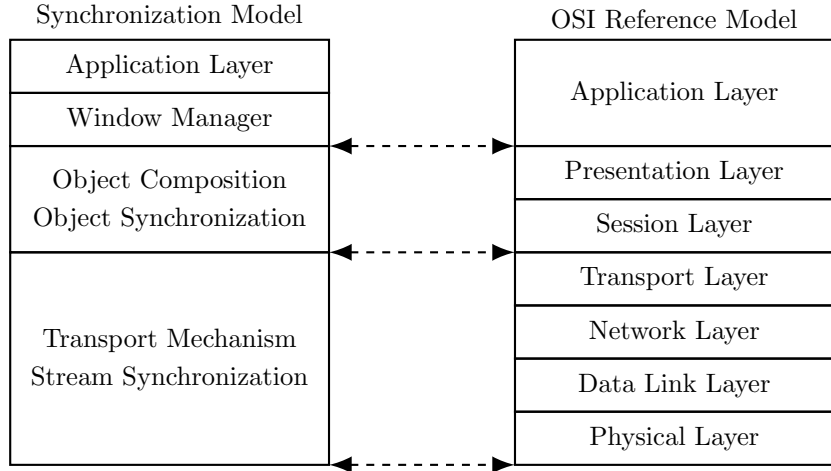


Figure 14: Comparison of OSI and integrated synchronization models.

this functionality. The object composition layer is responsible for providing synchronization at the level of objects with an algorithm suitable to support real-time presentation of concurrent media to the application. This layer corresponds to the [Table 5](#) object presentation and scheduling layers. The transport support layer provides services to support stream synchronization and other network services.

As mentioned earlier, the interfaces between the synchronization layers must provide the ability to monitor and maintain synchronization at higher levels. At the application layer, this means provision for primitives to support user functionality, such as “start” and “stop” of presentation. At the transport layer, it requires functionality to support control of skew, jitter, utilization, reliability, or other QOS parameters. In [Figure 15](#) abstract interface primitives between the synchronization layers are indicated.

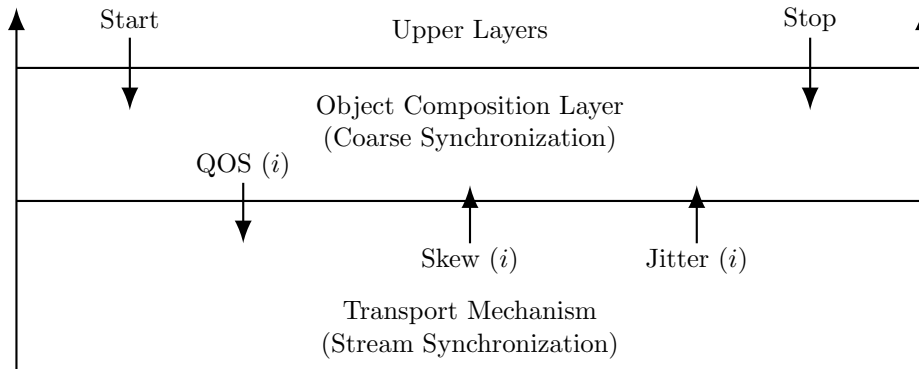


Figure 15: Interfaces between synchronization levels.

In response to synchronization feedback from the transport mechanism, the object composition layer can take corrective action to resynchronize two streams of data, as described previously. This action can indicate a change in QOS to decrease object utilization or increase speed to perform skew correction. In [Figure 16](#), stream *B* undergoes skew correction

by dropping a frame (reduction in utilization). This is representative of possible resynchronization techniques within the transport mechanism layer.

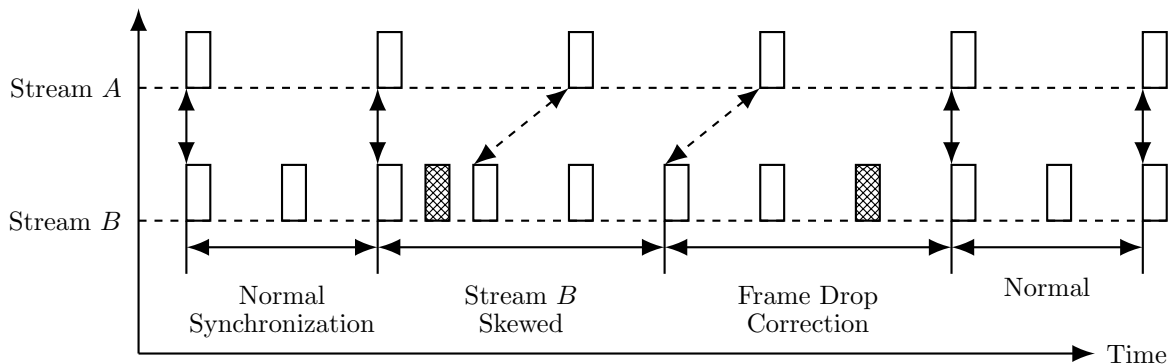


Figure 16: Skew correction.

In [Figure 17](#), a complete example is provided for the integrated synchronization model demonstrating the layered synchronization approach for temporal object composition. The example represents a medical information system incorporating audio, video, text, and graphics. Multimedia objects are synchronized at three levels, as follows:

- At the level of user interaction at starting and stopping points, indicated in [Figure 17a](#) as a set of queries through an information network, shown in [Figure 17b](#). [Figure 17c](#) indicates the hierarchy of database elements forming the multimedia presentation associated with a topic in the browsing model.
- At the object level between audio verbalizations and video segments, as shown in [Figure 17d](#). The various relationships between objects of the presentation are illustrated with their coarse synchronization points and corresponding OCPN.
- At the stream synchronization level between audio packets and video frames, shown in [Figure 17e](#). At this level, sets of PSFs are mapped to LSFs that make up the presentation objects.

Based on this example, the integrated model has wide applicability to temporal composition problems of multimedia applications.

5 Conclusion

Presented in this article is a review of current techniques and approaches to the problem of temporal object composition for a DMIS. An integrated model for the combination of these approaches towards the problems of low-level and high-level synchronization and object management is also proposed. Further research on the general problem of object composition needs to be addressed. In addition to the numerous communication areas of research associated with BISDN, various other issues need to be addressed before distributed multimedia

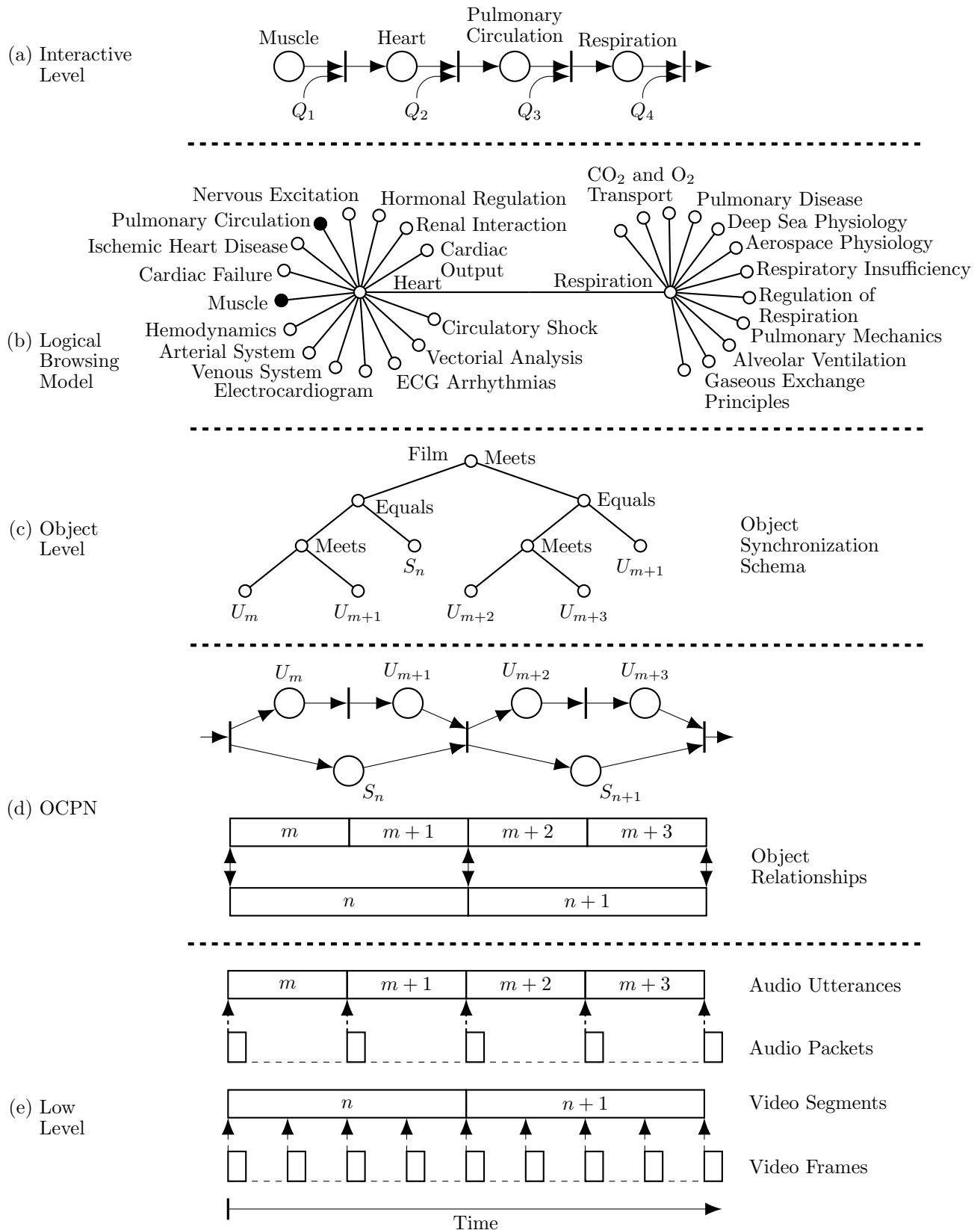


Figure 17: OCPN, LSF, and PSF relationship.

applications have wide acceptance. These include the investigation of a spatial composition mechanism, the use of the object-oriented paradigm for interobject and intermachine communications, the investigation of heterogeneity issues, distributed load sharing and the assignment of composition process based on data distribution [15, 25], and the development of scheduling algorithms for desired QOS.

Presently, research is being conducted at the Multimedia Information Laboratory at Syracuse University towards the development of a DMIS using the integrated synchronization model for a distributed system. The approach towards this work is to develop a working multimedia applications platform on an extensible system of heterogeneous workstations and file servers, implementing a distributed data version initially but with the goal of fully distributed operation [26]. Results of the experience with this work will appear in a later publication.

Acknowledgment

We wish to thank the anonymous reviewers for their helpful comments in the development of this paper, and acknowledge the support by the New York State Center for Advanced Technology in Computer Applications and Software Engineering (CASE) at Syracuse University.

References

- [1] J. J. Garcia-Luna-Aceves, "Towards Computer-Based Multimedia Information Systems," *Computer Message System 85*, R. Uhlig, ed., Elsevier-North-Holland, Amsterdam, 1985, pp. 61–77.
- [2] T. D. C. Little and A. Ghafoor, "Synchronization and Storage Models for Objects," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, April 1990, pp. 413–427.
- [3] J. S. Sventek, "An Architecture for Supporting Multi-Media Integration," *Proc. IEEE Computer Society Office Automation Symposium*, April, 1987 pp. 46–56.
- [4] P. B. Berra, C. Y. R. Chen, A. Ghafoor, C. C. Lin, T. D. C. Little, and C. Shin, "Multiuser Multimedia Application Development System Project," Technical Report, The New York State Center for Advanced Technology in Computer Applications and Software Engineering (CASE), Syracuse University, October 1989.
- [5] R. Steinmetz, "Synchronization Properties in Multimedia Systems," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, April 1990, pp. 401–412.
- [6] A. L. Zobrist and G. Nagy, "Pictorial Information Processing of Landsat Data for Geographic Analysis," *IEEE Computer*, vol. 14, no. 11, November 1981, pp. 34–41.
- [7] J. R. Perry *et al.*, "Performance Features for a PACS Display Console," *IEEE Computer*, vol. 16, no. 8, August 1983, pp. 51–56.

- [8] D. R. Ahlgren, J. Crosbie, and D. Eriqat, "Compression of Digitized Images for Transmission and Storage Applications," *SPIE Proc. Image Processing, Analysis, Measurement, and Quality*, Los Angeles, CA, vol. 901, January 1988, pp. 105–113.
- [9] A. Poggio *et al.*, "CCWS: A Computer-Based Multimedia Information System," *IEEE Computer*, vol. 18, no. 10, October 1985, pp. 92–103.
- [10] International Organization for Standardization, ISO Document no. 8613, ISO, Geneva, March 1988.
- [11] L. Mohan and R. L. Kashyap, "An Object-Oriented Knowledge Representation for Spatial Information," *IEEE Trans. on Software Engineering*, vol. 14, no. 5, May 1988, pp. 675–681.
- [12] P. V. Oosterom and J. V. D. Bos, "An Object-Oriented Approach to the Design of Geographic Information Systems," *Computers and Graphics*, vol. 13, no. 4, 1989, pp. 408–418.
- [13] J. Postel, G. Finn, A. Katz, and J. Reynolds, "An Experimental Multimedia Mail System," *ACM Trans. on Office Information Systems*, vol. 6, no. 1, January 1988, pp. 63–81.
- [14] C. Nicolaou "An Architecture for Real-Time Multimedia Communication Systems," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, April 1990, pp. 391–400.
- [15] T. D. C. Little, "Synchronization of Distributed Multimedia Objects," Ph.D. Dissertation, in Preparation, Syracuse University.
- [16] M. L. Liou, "Visual Telephony as an ISDN Application," *IEEE Communications Magazine*, vol. 28, no. 2, February 1990, pp. 30–38.
- [17] C. L. Hamblin, "Instants and Intervals," *Proc. of the 1st Conf. of the Intl. Society for the Study of Time*, J. T. Fraser *et al.*, eds., Springer-Verlag, New York, 1972, pp. 324–331.
- [18] J. E. Abate *et al.*, "AT&T's New Approach to the Synchronization of Telecommunication Networks," *IEEE Communications Magazine*, vol. 27, no. 4, April 1989, pp. 35–45.
- [19] B. B. Hehmann, M. G. Salmony, and H. J. Stuttgen, "Transport Services for Multimedia Applications on Broadband Networks," *Computer Communications*, vol. 13, no. 4, May 1990, pp. 197–203.
- [20] A. A. Lazar, A. Temple, and R. Gidron, "An Architecture for Integrated Networks that Guarantees Quality of Service," to appear in *Intl. Journal of Digital and Analog Cabled Systems*.
- [21] W. H. Leung *et al.*, "A Software Architecture for Workstations Supporting Multimedia Conferencing in Packet Switching Networks," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, April 1990, pp. 380–390.

- [22] International Telegraph and Telephone Consultative Committee SG VIII Draft Recommendation I.121, 1988.
- [23] National Standards Organization, T1S1 Technical Subcommittee, "Broadband Aspects of ISDN," December 1988.
- [24] M. Wernik *et al.*, "Supporting Multimedia Applications in Asynchronous Transfer Mode Networks," *Proc. 2nd IEEE COMSOC Intl. Multimedia Communication Workshop (Multimedia'89)*, Ottawa, April 1989.
- [25] A. Ghafoor, P. Berra, and R. Chen, "A Distributed Multimedia Database System," *Proc. Workshop on the Future Trends of Distributed Computing Systems in the 1990s*, Hong Kong, September 1988, pp. 461–469.
- [26] P. B. Berra, C. Y. R. Chen, A. Ghafoor, C. C. Lin, T. D. C. Little, and C. Shin, "An Architecture for Distributed Multimedia Database Systems," *Computer Communications*, vol. 13, no. 4, May 1990, pp. 217–231.