

# Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks\*

Thomas DC Little<sup>†</sup> and Arif Ghafoor<sup>‡</sup>

October, 1991

**Abstract** – Future multimedia information systems will require the integration of various objects retrieved from databases distributed across the network. Here we discuss a methodology for spatial and temporal composition of multimedia objects

**Keywords:** Distributed multimedia information system, value-added network, spatial & temporal integration, synchronization, communications protocols.

---

\*In *Computer*, Vol. 24, No. 10, October 1991, pp. 42–50.

<sup>†</sup>Thomas Little is with Boston University, *tdcl@bu.edu*

<sup>‡</sup>Arif Ghafoor is with Purdue University, *ghafoor@ecn.purdue.edu*

# Introduction

Multimedia is an emerging application-oriented technology embracing many computer disciplines, including interface design, networks, and databases. The field presents many interesting challenges for research. Advances in workstation and network technologies—and the desire to improve communications—have created great interest in multimedia applications. These applications require the processing, storage, and transmission of such data types as audio, video, text, graphics, and images.

One of the most important requirements of a distributed multimedia information system (DMIS), is the integration, or *composition*, of multimedia objects retrieved from databases distributed across a network. We define *multimedia objects* to be aggregate units composed of data elements taken from the aforementioned data classes. Their composition depends on the spatial and temporal characteristics of the multimedia elements.

Spatial composition of multimedia data is unique to each medium. It describes the assembly of objects in space on a workstation display at certain points in time. For pictorial representations like still images and graphics, composition operations include overlay and mosaic, and require processing such as scaling, cropping, color conversion, and position registration. For audio data, composition consists of mixing signals. Other “spatial” audio operations include gain and tone adjustments, which are useful in videoconferencing applications to prioritize one speaker’s voice among many. Gain or tone differences can signify “distance” in participants via signal-distortion techniques [1]. A typical spatial composition for pictorial data is shown in [Figure 1](#), which was constructed using scaling and mosaic-forming operations.

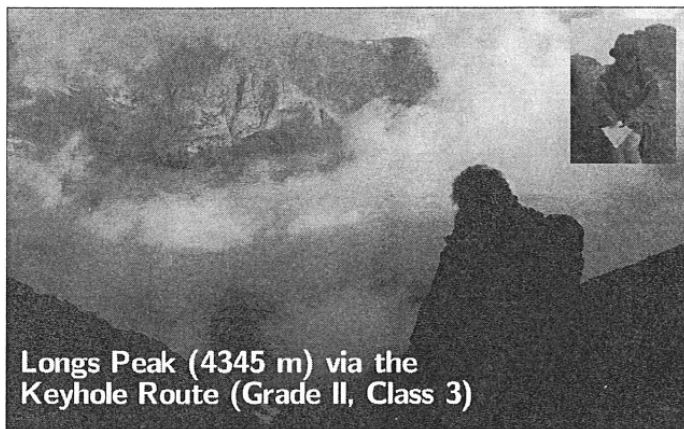


Figure 1: Cropping and overlaying images, and text formatting.

Temporal composition requires evaluating the relationships among component elements and scheduling their retrieval and presentation accordingly. The relationships can occur naturally, as for live audio and video, and they can also be synthetic, consisting of arbitrary temporal constraints on any multimedia data type [2]. [Figure 2](#) shows various data elements retrieved from storage and presented serially at the times indicated.

The primary issue we address is the overall process necessary to perform spatial and temporal data composition for a DMIS. With respect to delays introduced through the

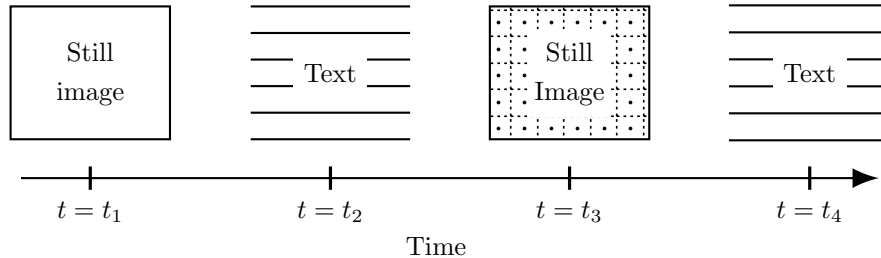


Figure 2: Temporal composition.

network, we find that temporal composition can be most suitably achieved at the workstation. Spatial composition is most effectively performed in a hierarchical fashion as dictated by the availability of system resources. The subsequent composition methodology combines spatial and temporal composition as a network service.

We also investigate database organizations and data distributions, and discuss spatial and temporal composition functions and their composition into the network architecture. Finally, we address the issue of mapping the composition process onto the network resources as a value-added service.

## Databases and distributed objects

A number of common database management systems (DBMSs) can be identified for multimedia applications. These systems can be organized as centralized, master-slave, and federated, as shown in Figure 3. In the simplest case, a multimedia information system resides on a single server system incorporating DBMS functionality for each medium. Data composition occurs entirely at the server. However, access to remote databases expands the potential applications for multimedia services, requiring a multiple database organization as in Figures 3b and 3c. Here, we define a data *server* as an intelligent database machine that can perform complex multimedia database operations such as browsing, linking, and pattern-matching. The server can also perform standard DBMS operations such as selection, projection, and join.

The most suitable database organizations for large-scale multimedia applications are the federated and master-slave types as shown in Figures 3b and 3c. In either case, users must be able to query across the universe of available data. This requires global naming of data and resolution of heterogeneity between DBMSs, server hardware, data formats, etc. After data elements are identified and located, the spatial and temporal composition of accessed data can be viewed as an added service for the creation of complex multimedia objects. The remaining tasks include specifying the composition function and distributing it over the set of computing elements in the DMIS in a manner appropriate for the DBMSs.

In a centralized environment where the data are not distributed (Figure 3a), a single server performs all object composition. However, the centralized environment is not rich enough to support most multimedia applications, since only local data are accessible. For a DMIS, multiple database servers can perform spatial composition to reduce the load on the destination workstation and the required network bandwidth for object communication. In

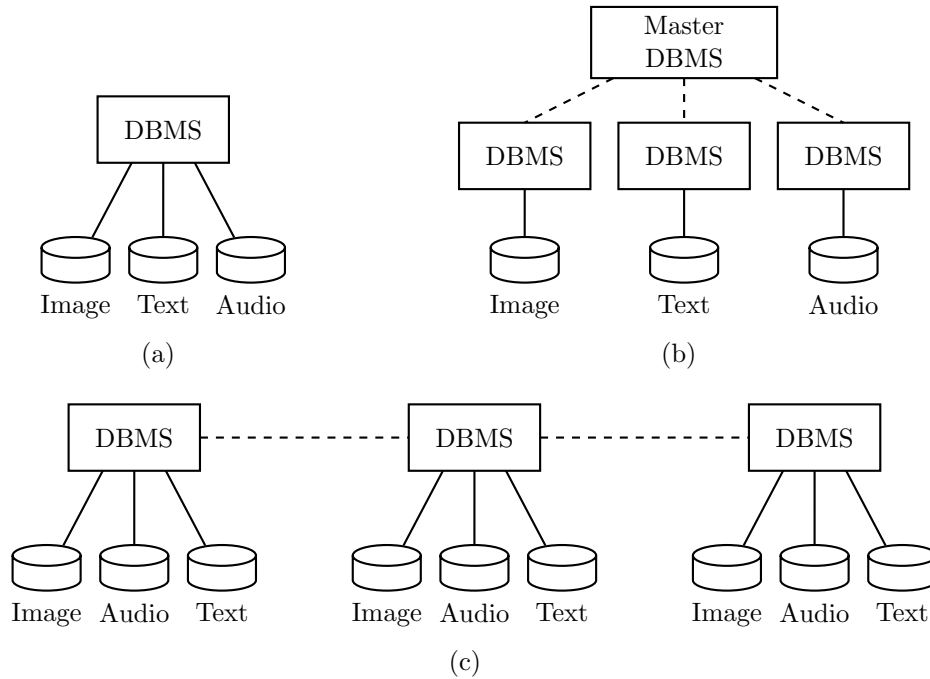


Figure 3: Centralized (a), master-slave (b), and federated (c) database-management systems.

particular, spatial operations such as generating mosaics, cropping, and color conversions, which are common in window-based environments, can significantly reduce superfluous data transfer when performed at server sites. Temporal composition does not exhibit the same benefit at remote servers because no data reduction occurs between source and destination. For strictly temporal composition without spatial composition requirements, a point-to-point virtual connection between source and destination best maintains synchronization for continuous communication and presentation of objects. No additional processing can occur within the network at some intermediate site.

Based on the assumption of a distributed DBMS organization, five types of point-to-point connections for object retrieval and composition in a DMIS can be identified. These composition architectures, shown in [Figure 4](#), include

- single source to single destination,
- single source to multiple destinations,
- multiple sources to single destination,
- multiple sources to single destination using an intermediate site, and
- multiple sources to multiple destinations using an intermediate site.

[Figure 4a](#) illustrates a point-to-point connection for a client-server relationship between a single multimedia server and the workstation. [Figure 4b](#) represents a shared-object architecture in which a single object is displayed simultaneously to various users via multicasting.

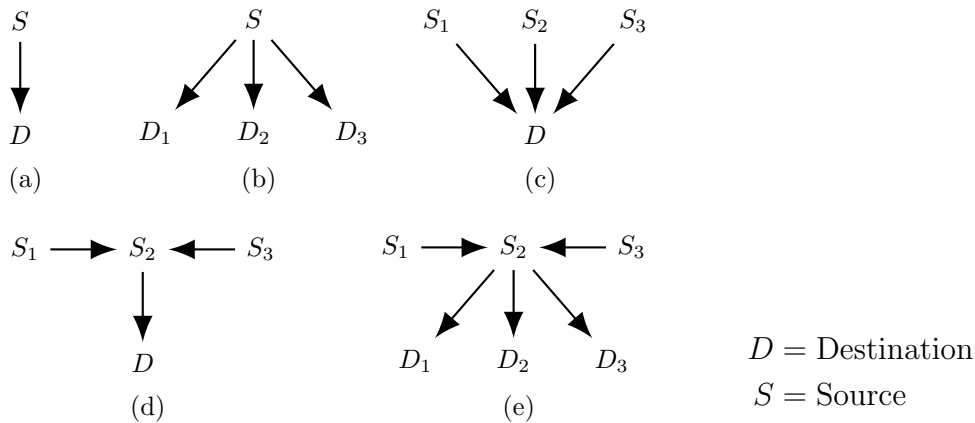


Figure 4: Multimedia connections in a distributed information system: single source to single destination (a), single source to multiple destinations (b), multiple sources to single destination (c), multiple sources to single destination using an intermediate site (d), and multiple sources to multiple destinations using an intermediate site (e).

This mode enables computer-supported cooperative work (CSCW) [3], or teleconferencing, and can require concurrency and consistency control mechanisms for shared objects. Figure 4c represents a distributed-object environment, for which complete composition is performed at the sink in a multidrop fashion. This case can be handled by independent network connections between the sink and each source. However, it poses a challenge to workstation control of intermedia synchronization.

Figure 4d shows a scenario in which objects are composed at an intermediate site after arriving from distributed sources and are sent via a single connection to the final destination. By using a single connection, data sequencing ensures strict ordering and thereby provides intermedia synchronization [4].

Figure 4e illustrates the general multicast and multidrop case of distributed object composition in a shared-object environment.

As mentioned, the process of data composition can be distributed within the network to minimize various system costs. In particular, the criterion for the selection of composition locus is a function of various system costs including communication, storage, processing, and desired system-performance characteristics such as the reliability and quality of communications. The problem is analogous to optimization of queries in a distributed DBMS [5]. However, some differences exist, which we discuss later.

## Spatial composition

Current data-composition schemes incorporating video data are mostly analog. For example, image mosaics are generated by capturing and placing video stills from the analog output of a videodisc player. Analog schemes differ from the composition of video, images, audio, and text in a completely digital domain. In the digital realm, we can use storage, processing and communication technology, and open system applicability to achieve device independence in

an integrated environment.

The task of composing data spatially as shown in Figure 1 can be specified using an object-oriented paradigm and implemented using DMIS components. Many data representations for composite multimedia data have been proposed based on the object-oriented paradigm [6]. We concentrate on the task of composition rather than discussing a complete object-oriented spatial data representation. We assume a simple layout structure based on ODA [7] (Office Document Architecture), as shown in Figure 5. Figure 5a shows a typical spatial translation. Figure 5b illustrates a spatial data representation, or a spatial *hierarchy* for these translations.

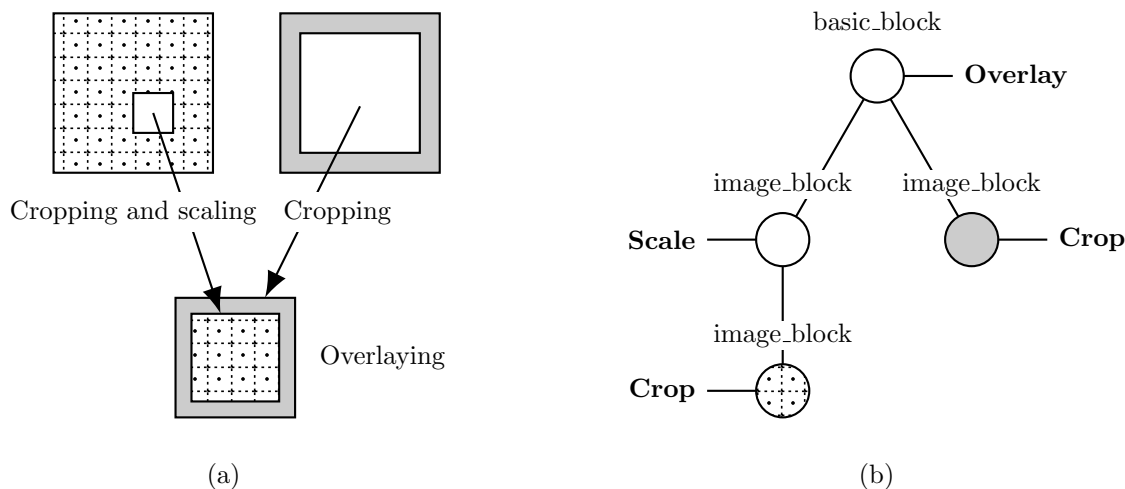


Figure 5: Multiphase spatial composition: transformations (a) and spatial hierarchy (b).

If data are composed at the workstation, user interaction can be most readily incorporated into the composition process. For example, users can choose window placement, resolve occlusion, evaluate view-space coordinates, and perform scaling without dialogue with a remote composition service. Similarly, object modification is most readily achieved on the workstation. However, some objects can exceed the local storage capacity of the workstation, and modification must occur on fragments of the original object stored remotely.

Composition at the workstation requires transmission of complete multimedia objects from the database, such as full-size color video or full-resolution images. At the destination, much of this resolution is not used after scaling, cropping, color mapping, or other spatial operations. A remote service, by contrast, can perform these spatial operations prior to data transmission, reducing the transmission overhead and freeing the workstation for other tasks. Specialized hardware like array processors and video special-effects devices optimized for high-speed spatial manipulation can also reside at the composition sites.

To provide a value-added service facilitating these kinds of spatial operations within a network, we describe a set of spatial manipulation primitives that can be implemented as standard functions at various sites within the DMIS. These primitives fall into two classes: unary and binary operations. Unary operations are applied to single data objects like the low-pass filtering of an audio segment or image cropping. Binary operations combine pairs

of objects, creating their composition. Table 1 summarizes examples of these operations and lists approximate processing costs.

Table 1: Characteristics of typical spatial manipulation primitives.

| Operation          | Operands | New Size           | Cost Estimate        | Data Types |
|--------------------|----------|--------------------|----------------------|------------|
| Overlay (mix)      | $A, B$   | $\max( A ,  B )$   | $\min( A ,  B )zP_m$ | Image,     |
| Overlap            | $A, B$   | $> \max( A ,  B )$ |                      | video,     |
|                    |          | $<  A  +  B $      | $( A  +  B )zP_m$    | graphics,  |
| Abut (concatenate) | $A, B$   | $ A  +  B $        | $( A  +  B )zP_m$    | and        |
| Crop (cut)         | $A$      | $c A $             | $c A zP_m$           | audio      |
| Scale (filter)     | $A$      | $s A $             | $ A P_s$             |            |
| Text format        | $A$      | $f A $             | $f A zP_m$           | Text       |

$|A|$  = Size of  $A$  in image pixels, audio samples, or text characters

$c$  = Fraction of unaffected pixels in a crop operation

$f$  = Number of pixels per formatted character

$P_m$  = Unit cost per memory move

$P_s$  = Unit cost per scaled pixel

$s$  = Scale factor

$z$  = Number of words per image pixel, audio sample, or text character

We envision spatial composition to be a multiphase process consisting of unary operations that adjust the data elements, followed by binary operations that compose the adjusted elements into final form. Further, we define a logical entity, called the *spatial translator* (ST), which can be distributed throughout the network to perform these operations. Note that these operations constitute the overall translation from storage to workstation display, a simplification of the many intermediate transformations between the various coordinate systems and canonical representations usually indicated in graphics programming standards like the Graphical Kernel System (GKS) and Programmer's Hierarchical Interactive Graphics System (PHIGS).

Distributing operations with a uniform interface throughout the DMIS lets users tailor spatial data composition to individual object requirements. As data are retrieved, users can invoke spatial operations at remote servers through the uniform interface to reduce data traffic.

As mentioned, spatial operations are basically transformations on multimedia objects to achieve such functionality as cropping, filtering, and overlaying. However, some operations may result in data reduction, while in other cases the new data elements can be larger than the old (for example, coloration and scale-up). If the criteria to select between a server or workstation for composition are based solely on reduction of network traffic, clearly we want to perform unary spatial transformations at the server site only when a data reduction occurs. However, if the criteria include minimizing processing at the workstation, we may want to perform the data-increasing transformations at the server as well.

Considering binary spatial manipulation, suppose the criterion to select a site for composition is based on reducing network traffic and minimizing the workstation processing load. Let  $R_1$  and  $R_2$  represent two objects to be merged spatially in some arbitrary way, with characteristic data sizes of  $|R_1|$  and  $|R_2|$ . The binary spatial merge for some transformation

$$b_\gamma$$

will result in a final display object  $R_d$  with size  $|R_d|$  and is denoted as

$$R_d = b_\gamma(R_1, R_2).$$

Clearly  $\max(|R_1|, |R_2|) \leq |R_d| \leq |R_1| + |R_2|$ , since the merge represents in the worst case the union of the two objects (for instance, overlay, abut, and mix). This relationship is valid for any type of data object including text, image, video, and audio, assuming the absence of data compression.

If  $|R_d| = \max(|R_1|, |R_2|)$ , performing composition at the server potentially reduces data traffic to the workstation by  $\min(|R_1|, |R_2|)$ . On the other hand, if  $|R_d| = |R_1| + |R_2|$ , no savings result. Within this range, the choice of a site for composition depends on the penalty associated with the composition processing versus the necessary data communication bandwidth and the composition requirements of other related objects. The overall savings in data transmission includes all operations required in forming the final composite object.

For example, consider again the multimedia object in [Figure 1](#) (shown here in gray scale). The data constituting the object are assumed to exist in several distributed servers. Each image is stored in 24-bit, full-color format (8 bits per color, three colors per pixel). The subimage in the figure is created by cropping its corresponding (1200 × 925-pixel) source image to a size of 80 × 120 pixels. This is superimposed onto the background image along with the ASCII text (61 characters × 8 bits per character) with a selected font. The final composite object is entirely in bitmapped form and is the size of the cropped background image (888 × 685 pixels).

If composition is performed at the workstation, the raw, unprocessed images and text would have to be transmitted to that location (1200 × 925 pixels × 24 bits/pixel × 2 images + 61 characters × 8 bits/character = 53,280,488 bits). If composition occurs before transmission, the data-transmission requirement is 888 × 6 pixels × 24 bits/pixel = 14,598,720 bits. The savings in data transmission for remote composition is 38,681,768 bits, or a traffic reduction of 73 percent.

Note that in these calculations we have assumed no data compression or the overheads associated with storage and communication. However, the percentage savings remain unchanged with compression. This example illustrates the reduction of data possible during the process of data composition at some intermediate site in the network as data are retrieved from a database, merged, and sent to the final presentation site.

The spatial processing requirements depend on the size and type of the object considered, the type of composition function, and its implementation, whether parallel or serial. [Table 1](#) summarizes processing approximations for some of the various spatial operations on images, audio, and text. Either time or number of operations can be applied as costs. For example, the move cost relates to memory-access time; that is,  $P_m$  equals two memory-access cycles (one read, one write).



For a given composite multimedia object, the spatial formatting requirements can be quantified by using the cost estimates for the various spatial transformations. Chaining the cost estimates as described by the spatial hierarchy provides an estimate of overall processing cost. The evaluation of such processing requirements is important in determining the real-time computational performance of the workstation and the distribution of composition operations within the network.

## Temporal composition

Multimedia data objects are time-dependent and must be synchronized accordingly. This synchronization requirement encompasses static objects, such as still images and text, and continuous streams of audio and video. A DMIS must satisfy this requirement in the presence of random network delays due to the inherently asynchronous nature of a packet network [1,8,9] and the latencies associated with storage devices. This problem is particularly acute, since several streams of different origin can require synchronization with each other. For continuous streams of data, the problem is to ensure the proper time of play-out (presentation time) of each data element in spite of random network delays, as illustrated in Figure 6.

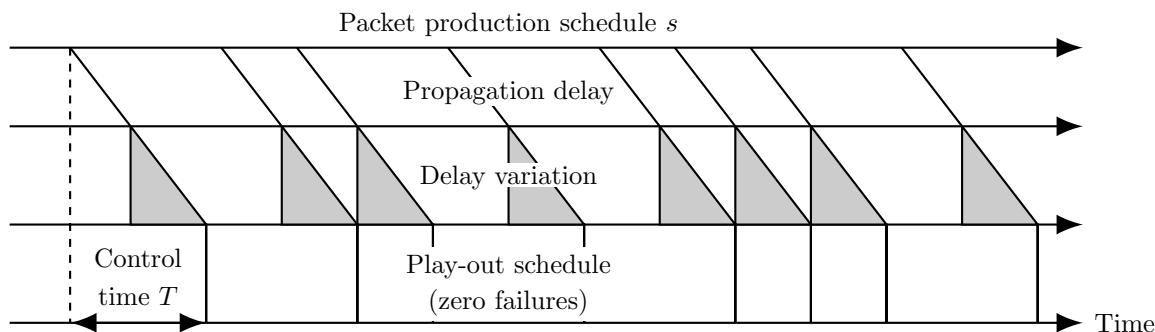


Figure 6: Play-out in the presence of random network delays.

Synchronization of this type has been typically applied to single streams of packetized audio and video data [8] but can be generalized to multiple streams and nonstream data such as still images and text. Approaches to synchronization determine the delay and buffering required to establish a level of packet loss for various network delay distributions. This delay, called a control time  $T$ , can be found for audio and video streams [8,9] given a target packet-loss probability. The same principle can be applied to nonstream data. However, in the former analyses, it is assumed that the packet-generation rate is equal to the consumption rate and that the capacity of the communication channel is not exceeded. When arbitrary sequences of multimedia objects are presented, the data can easily exceed the channel capacity for some intervals. However, due to flexibility in object retrieval for stored-data applications, data-retrieval times can be reorganized so that the channel capacity does not adversely impact the presentation. Of course, this is not possible for live data sources. In essence, database sources provide more freedom in controlling when data are acquired by the application, as shown in Figure 7.

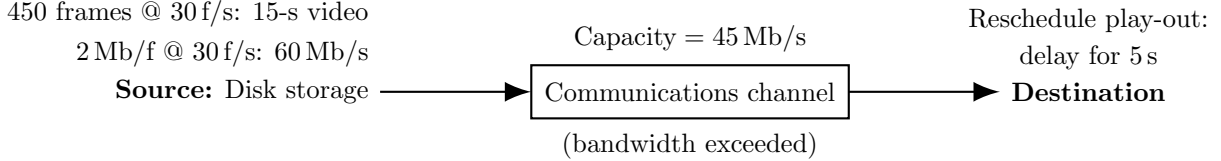


Figure 7: Flexibility in database retrieval.

Beyond the problem of compensating for random network delays, multiple-stream synchronization can be controlled by the destination workstation or by any other intermediate server within the network prior to data delivery. An object’s temporal specification needs to be known to the synchronization controller irrespective of the controller location within the network. A Petri-net-based approach to specify the temporal requirements for multimedia objects can be used for this purpose [1, 10]. If synchronization is performed at the destination (as in Figure 4c), the workstation must evaluate the temporal specification of objects and carry out a synchronization dialogue with the remote servers prior to data transfer. If synchronization occurs at any other intermediate server (as in Figure 4d), the workstation does not need to evaluate the temporal specification. However, due to such factors as network latencies, retransmission (if applicable), and workstation-performance limitations, significant skew can be introduced among these synchronized streams, and an intermediate server has little control over the final result at the workstation. Therefore, an intermediate server cannot reliably provide the fine synchronization required for audio/video streams, which generally require a skew of less than 150 milliseconds.

A typical application might use the synchronization service for preorchestrated presentations, teleconferencing, or CSCW. Typically, an object is identified from a database through browsing or querying operations. Once identified, the object can be retrieved, composed, and presented. Since both spatial and temporal composition specifications must be met, an important operation in this scheme is the decomposition of multimedia data into classes for independent transfer. This separation can increase network performance by identifying unique data-traffic classes and using different transfer protocols tailored to each class [11]. In essence, such transport protocols can provide different levels of guaranteed quality of service for each data type based on its tolerance to packet delay and loss. For example, an image object requires error-free service, while audio objects can tolerate errors.

We present two communication protocols to perform synchronization as a value-added network service between sources and a destination [12]. The Application Synchronization Protocol (ASP) and Network Synchronization Protocol (NSP) allow the communication of complex multimedia presentations from distributed sources for play-out at a single site. The protocols utilize a Petri-net-based temporal specification of a multimedia object [12]. This model basically specifies the precedence relations among all subobjects in the form of a partial and strict ordering. Partially ordered objects have simultaneous play-out deadlines and require concurrent presentation, while strictly ordered objects require time-sequential play-out (as in Figure 2).

The ASP sets up and initiates data transfer as specified by temporal requirements on an end-to-end basis. Spatial requirements of an object affect the ASP, since they must be passed during connection setup to the remote sites. The ASP takes as input a selected object

representing the aggregation of a complex multimedia presentation requiring synchronization. It then returns independent streams of synchronized data traffic that can then be routed to specific output devices for play-out at the workstation. The ASP also provides control over the quality and the cost of transmission service by negotiating a target packet-loss probability and delay with the underlying guaranteed-service data-transport mechanism [11].

The NSP provides a data-transfer facility with a predicted end-to-end delay characteristic based on the specified probability of late packets. Neither the ASP nor the NSP specifically provides a mechanism for reconstructing late or lost packets. They rely on selecting the appropriate quality of service parameters for each medium's tolerance to delay and loss.

These protocols allow synchronization of independent network connections, unlike approaches to synchronization [4] that require sequencing of synchronized data onto a single virtual circuit (as in Figure 4c). For the ASP, it is assumed that all synchronization is performed at the destination. Intermediate nodes are considered only if some intermediate spatial composition function is needed. Otherwise a point-to-point connection is implied with corresponding end-to-end properties.

In summary, the ASP and NSP provide synchronization, which involves the following steps:

1. Retrieving the spatio-temporal relationships that describe the components constituting the complex multimedia object.
2. Evaluating the precedence relationships in the Petri net, thereby creating a play-out schedule.
3. Decomposing the schedule into subschedules based upon the different traffic classes represented and the locations of stored data.
4. Determining the overall control time required to maintain synchronization among the traffic classes through interaction with the NSP and cooperating sites.
5. Providing synchronous data transfer.

## Spatio-temporal composition

Here we investigate a combined approach to performing temporal and spatial composition of a DMIS as a service provided by the network. We envision the heterogeneity of the network in terms of speed, computational resources, and topology to motivate the use of a hierarchical composition process. Multiple data servers (DSs) and intermediate sites, or composition servers (CSs), collaborate to compose the requested objects both spatially and temporally.

Generally, an object model consists of information describing the various operations necessary for spatial-temporal composition. The model is stored in the network at a central site. At the time a session is established by a user, the information is identified from the central site, and the object hierarchy is decomposed and mapped onto the set of servers (DSs, CSs, and the workstation).

Although the problem of assignment of composition locus for a given multimedia object is analogous to query optimization for distributed databases [5], it differs in several ways. Because a sequence of spatial operations often cannot be permuted, little optimization can be achieved through reorganizing the sequence of spatial operations. Some spatial transformations (like scale-up) increase data volume and are optimal when performed closer to the data's destination rather than near its source. Second, the optimization technique assumes homogeneity in processing and communication costs and therefore does not account for specialized hardware for each medium, nor does it consider the necessity for load balancing when long-lived database transactions are present (as in the retrieval and play-out of movies).

As mentioned, temporal composition always requires control by the workstation, since independent media cannot be combined in any reasonable manner at intermediate sites in the network en route to the play-out destination, except to provide sequencing [4]. On the other hand, spatial composition can be performed at either the composition servers or at the destination workstation. The choice depends on the object characteristics, the workstation storage and computational capability, and the bandwidth of the network. If remote composition is dictated, the composition is delegated to at least one CS, which we call the primary CS. The secondary servers consist of data sources. The choice of primary server should allow for:

- the locality of objects,
- the amount of spatial processing required on the selected objects,
- the spatial composition capability of each CS (for example, an array processor), and
- the current CS loading.

The first factor means that the composition server with the closest proximity to the largest percentage of data is optimal [5]. The spatial processing consideration should be weighed in determining the primary server. By analyzing the spatial organization of a complex object, one can map a composition function to a specific CS or set of CSs, based upon CS requirements and the capabilities of the servers. Also, the load on a CS can affect the primary server selection.

With respect to the workstation and network components, the assignment of spatial operations must allow for the optimal utilization of bandwidth. This can dictate reducing objects at the CS (through scaling, cropping, and filtering), enlarging objects at the workstation (scaling, coloring, and formatting text), and building windows at the workstation (dealing with occlusion). To minimize workstation computational requirements, all spatial operations must be performed at the CS, including the window management.

Clearly there is a trade-off between traffic on the network, computational load at the workstation, and user control over presentation. When objects are composed at the CSs, the user cannot control data assembly, since this operation is performed prior to the reception of the data. A compromise can be achieved by specifying the object hierarchy so that some objects must be assembled at the workstation and others can be distributed to various CSs. Based on these considerations, we present a scenario for combined spatial and temporal object composition using the ASP and ST entities. We chose the electronic magazine as an example.

This multimedia application is analogous to a printed magazine. A “reader” may browse through “pages” to read articles and view pictures and audio/video presentations, as shown in Figure 8. We model the pages of this application as objects that are browsed sequentially. In addition, the user may perform queries or searches to locate specific articles or advertisements. Spatial and temporal composition is required for elements of text, image, audio, and video, within the context of a page as indicated by the spatial and temporal specifications of Figures 9 and 10. These data are assumed to be distributed across a high-speed network and require remote data access and composition.

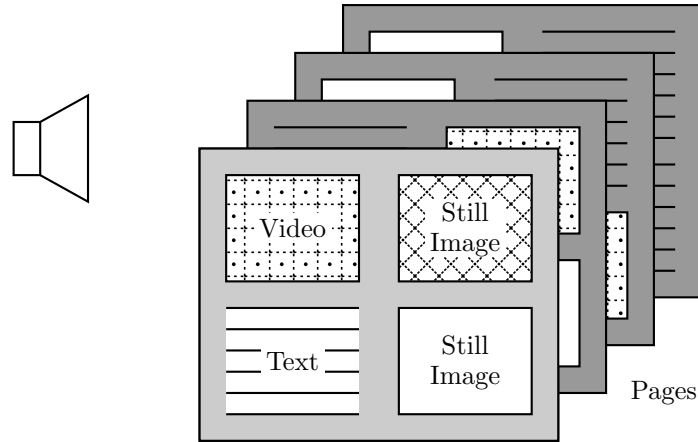


Figure 8: Pages of the electronic magazine.

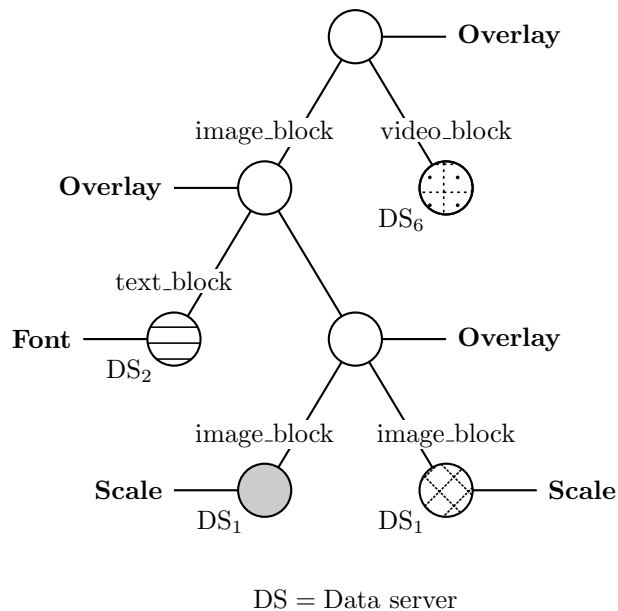


Figure 9: Spatial hierarchy for a page.

After a page is selected for presentation at the workstation, a CS is identified by consulting some centralized *name server* that maintains a global table of objects, names, locations, and

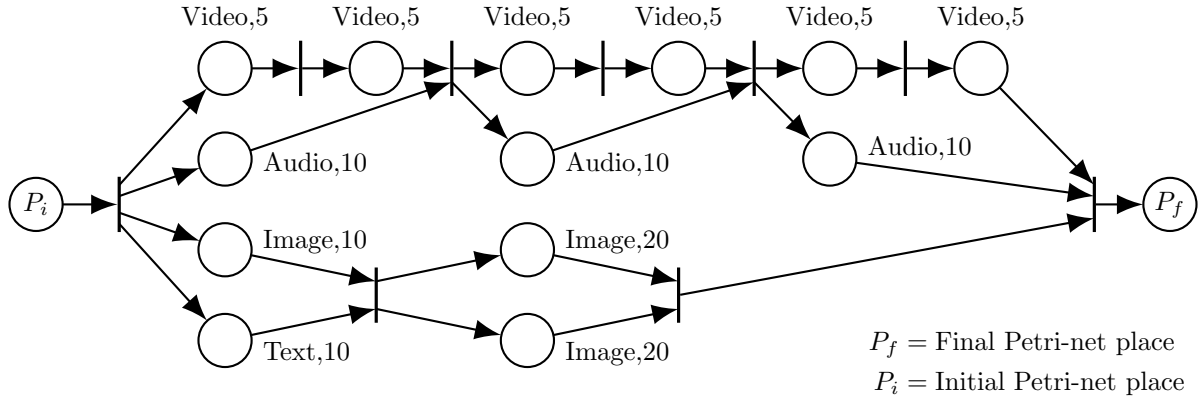


Figure 10: Temporal specification for a page.

other characteristics. The server also maintains DMIS global state information regarding the availability of resources and the load at each CS. This information is applied in the selection of a primary CS for each class of data used to construct an object. Subsequently, the workstation uses the object's temporal specification (Figure 10) to establish an ASP session. The ASP then initiates individual concurrent connections to the primary CSs using the NSP on a point-to-point basis. The primary CSs establish NSP connections with DSs as required for the indicated spatial composition. Load is effectively balanced by placing new sessions onto CSs with the least load, if appropriate, in light of other primary server selection criteria. Based on the DMIS architecture shown in Figure 11, a mapping of spatial operations onto the set of CSs appears in Figure 12. In general, a mapping requires optimization of some cost function such as communication bandwidth.

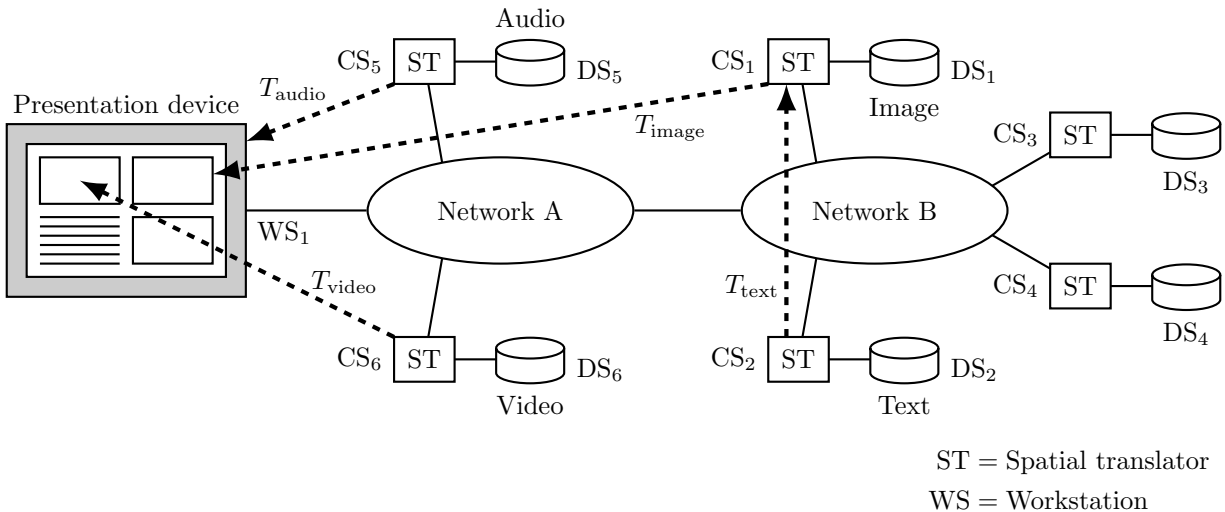


Figure 11: Network example.

A synopsis of the evaluation of the temporal requirements by the ASP and NSP is as follows: The temporal specification is interpreted to generate a set of deadlines for each class of

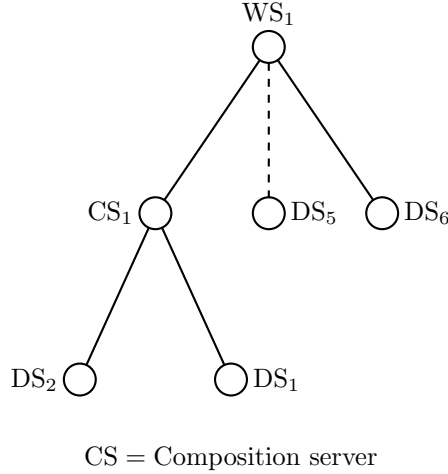


Figure 12: Mapping of spatial operations to composition servers.

multimedia data [12]. Assuming video is synchronized per frame (30 f/s), audio per 10-second interval, and text and image per object, the temporal specification (see Figure 10) generates the subschedules:  $S_{\text{video}} = (0, 0.033, 0.066, 0.099, \dots, 29.966)$  s,  $S_{\text{audio}} = (0, 10, 20)$  s,  $S_{\text{text}} = (0, 10)$  s, and  $S_{\text{image}} = (0, 10)$  s. For each subschedule, an NSP connection establishment procedure is invoked, resulting in a set of control times:  $T_{\text{audio}}$ ,  $T_{\text{image}}$ ,  $T_{\text{text}}$ , and  $T_{\text{video}}$ , which represent the aggregate end-to-end delays over virtual channels between data source and destination for each medium (see dashed lines in Figure 11). The maximum control time  $T_o$  is found and returned to the ASP for generation of the overall start time. Data transfer is provided by the underlying communication mechanism that interprets the derived schedule, which is assured to be feasible by the evaluation of the connection establishment procedure of the ASP and NSP.

## Conclusion

The composition of data objects in a DMIS is an important technical problem considering the complexity of supporting time-dependent media and data heterogeneity in an open systems environment. Object composition requires consideration of both the temporal and spatial characteristics of multimedia elements.

There are trade-offs to partitioning the object-composition process and its mapping onto the resources of the network based on the communication and computational requirements of an object. The mapping of this process, which combines spatial and temporal composition, results in a value-added service provided by the network.

## Acknowledgements

We thank the reviewers for their constructive comments. We also thank P. Rivera for providing the photographs that make up Figure 1. Our work was partially supported by the

New York State Center for Advanced Technology in Computer Applications and Software Engineering at Syracuse University.

## References

- [1] L. F. Ludwig, N. Pincever, and M. Cohen, "Extending the Notion of a Window System to Audio," *Computer*, Vol 23, no. 8, Aug. 1990, pp. 66–72.
- [2] T. D. C. Little and A. Ghafoor, "Network Considerations for Distributed Multimedia Object Composition and Communication," *IEEE Network*, vol. 4, no. 6, Nov. 1990, pp. 32–49.
- [3] I. Greif, ed., *Computer Supported Cooperative Work: A Book of Readings*, Morgan Kaufmann, San Mateo, CA, 1988.
- [4] W. H. Leung, T. J. Baumgartner, Y. H. Hwang, M. J. Morgan, and S. C. Tu, "A Software Architecture for Workstations Supporting Multimedia Conferencing in Packet Switching Networks," *IEEE J. on Selected Areas in Comm.*, vol. 8, no. 3, Apr. 1990, pp. 380–390.
- [5] W. W. Chu and P. Hurley, "Optimal Query Processing for Distributed Database Systems," *IEEE Trans. on Computers*, vol. C-31, no. 9, September 1982, pp. 835–850.
- [6] D. Woelk, W. Kim, and W. Luther, "An Object-Oriented Approach to Multimedia Databases," *Proc. ACM*, ACM, New York, May 1986, pp. 311–325.
- [7] *ISO Document No. 8613*, International Organization for Standardization, Geneva, Mar. 1988.
- [8] M. De Prycker, M. Ryckebusch, and P. Barri, "Terminal Synchronization in Asynchronous Networks," *Proc. IEEE International Conf. Comm.*, IEEE Press, Piscataway, NJ, 1987, pp. 800–807.
- [9] G. Barberis and D. Pazzaglia, "Analysis and Optimal Design of a Packet-Voice Receiver," *IEEE Trans. on Comm.*, vol. COM-28, no. 2, Feb. 1980, pp. 217–227.
- [10] P. D. Stotts and R. Furuta, "Petri-Net-Based Hypertext: Document Structure with Browsing Semantics," *ACM Trans. on Office Automation Systems*, vol. 7, no. 1, Jan. 1989, pp. 3–29.
- [11] A. A. Lazar, A. Temple, and R. Gidron, "MAGNET II: A Metropolitan Area Network Based on Asynchronous Time Sharing," *IEEE J. on Selected Areas in Comm.*, vol. 8, no. 8, Oct. 1990, pp. 1582–1594.
- [12] T. D. C. Little and A. Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Services," to be published in *IEEE J. on Selected Areas in Comm.*, Dec. 1991.