A Framework for Adaptive Networked Multimedia

Dick C.A. Bulterman
The Multimedia Kernel Systems Project
CWI: Centrum voor Wiskunde en Informatica
The Netherlands
Email: dcab@cwi.nl

Accessing multimedia information in a networked environment introduces problems that don't exist when the same information is accessed locally. These problems include: competing for network resources within and across applications, synchronizing data arrivals from various sources within an application, and supporting multiple data representations across heterogeneous hosts. Often, special-purpose algorithms can be defined to deal with these problems, but these solutions usually are restricted to the context of a single application. A more general approach is to define an adaptable infrastructure that can be used to manage resources flexibly for all currently active applications. This paper describes aspects of a research program into adaptive, networked multimedia that started at CWI in 1991.

1. Problem Overview

Networked multimedia is a generic term that describes a model of information distribution in which data sources are located separately from data sinks. Networked multimedia offers a number of advantages to applications: the network provides a convenient means of distributing information to other sites, it provides access to compute servers where special-purpose processing of multimedia data can take place, and it provides access to central servers that can be used to store the often vast amounts of data required to represent multimedia information fragments. At the same time, however, networked multimedia presents an application with a number of disadvantages when compared to accessing and manipulating multimedia data locally: the data delivery characteristics of the network are difficult to predict and control, the contention for critical system and data resources across the network makes balanced data access difficult to achieve, and differences among network hosts may make data objects difficult to share.

In order to make networked multimedia more useful to application designers and users, considerable effort needs to be devoted to studying the way that data servers, operating systems and network infrastructures provide access to time-sensitive data. Most current approaches define extensions to "conventional" means of accessing remote data to provide predictable network service and performance. For example, predictability is provided in data object servers (either file servers or database systems) by supporting efficient object storage and retrieval/delivery [DBL92,RV91] and in operating systems by supporting *quality of service* guarantees for delivery of (pos-

sibly) complex data types [ABL92,D91,GA91,HKN91,LMM92,TNP90]. At the network level, support for predictable multimedia is provided by, among others, admission control techniques that regulate use of resources and by technologies that provide deterministic network/data access [CSZ92, HM91,JST92,LG91,VF90,T90]. The basic premise of this work is that an application will request a data object (or a collection of objects) requiring a specific amount of resources during a specified time. If these resources are available, the application can execute; if not, the application is either delayed or it is denied access to the resources.

An implicit assumption in current approaches is that the application program bears a significant control burden in requesting and coordinating multimedia information. Consider, for example, the application environment shown in Fig. 1. Here, an application running on node *a* requests multimedia data from four sources located on three separate servers. The application must know the resource requirements of each *stream* of data it uses (where we use the term "stream" to mean either a single object or a collection of similarly-typed objects from a single server), it must coordinate the arrival and manipulation of multiple independent streams, and it must take any actions necessary if a given stream cannot be provided by the infrastructure. In general, the application software must control all content-based actions in or among the streams in the context of the application, while the infrastructure will control representation-based actions within a single stream in the context of service guarantees or network/server activity.

The content/representation division of control works well in environments where sufficient resources exist to handle an application's request fully or where insufficient resources exist to handle the requests at all. It is less effective when an application can receive only partial support of its requests. In Fig. 1, suppose that one of the requested data streams could not be made available at the required level of service. An application may decide to skip this data object (or the collection of objects associated with that stream), or it may decide to substitute another data object or object server. In effect, the application program is engaged in a process of resource allocation: it is attempting to match its data needs to the resources available at various locations in the support infrastructure. Unfortunately, to do efficient resource allocation—even if this means only selecting from a set of available data streams—the applica-

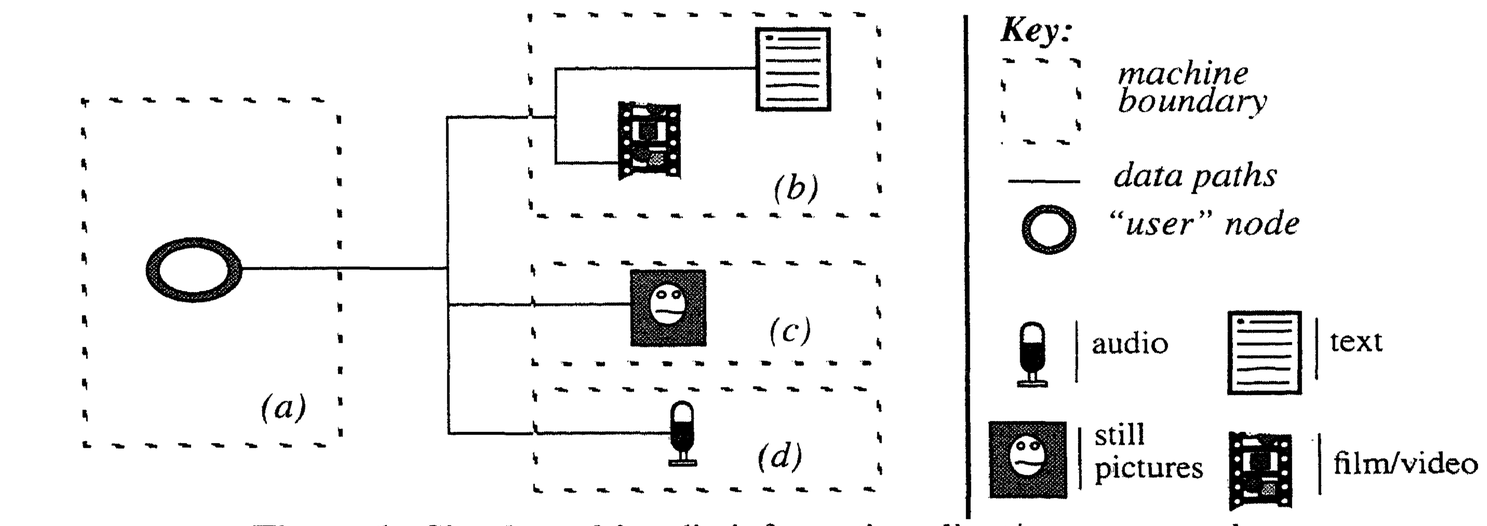

Figure 1. Simple multimedia information client/server example.
The client (a) is fed by three servers (b, c, d), one of which supplies two data types. (The structure of the client and the client's application is not shown.)

tion needs to know how to best make use of the available infrastructure. This involves issues that most applications programs are ill-equipped to resolve. (It also requires applications to be rewritten when they are moved to new environments.) Alternatively, the operating system or the data servers could handle all resource allocation, but the (local) operating system will have only limited knowledge of the state of each of the servers and other applications active within the networked environment, and the data servers will be able to manage only their own streams, not other streams in the infrastructure.

This paper presents an alternative approach to supporting networked multimedia that is being studied within the Multimedia Kernel Systems group at CWI. Our work is aimed at studying coordinated application and infrastructure-based support for *adaptable* applications. Here, adaptable means that an infrastructure can be defined so that an application can adapt to the resources available at the time the application is run. The types of adaptability we consider include responding to (possibly transient) variations in the number and composition of network and remote resources that are available during application execution, as well as application and server support for heterogeneous collections of input/output devices. Our approach is based on two mechanisms. First, we define an application specification that explicitly describes the data objects used by an application, the manner in which the objects interact, and the available ranges of alternatives that are acceptable to the application at run-time. Second, we define an interface to the data objects that allows alternative representations to be selected at run-time by a process of application-transparent negotiation at run-time. This approach is specifically geared to applications that have a *document* or *presentation* structure. An authoring system (such as [RJM93]) can be used to generate a specification that can be accessed/executed at some later time. By allowing the execution to be adaptable, one specification can potentially allow an application to be available within a heterogeneous environment under a range of resource availability conditions. As will be discussed, this can help to reduce the high cost of authoring multimedia applications and it can lead to more efficient use of multimedia infrastructures.

In the sections below, we describe the framework for partitioning control responsibility within the system infrastructure to support adaptable applications. This framework, the *Amsterdam Multimedia Framework*, distinguishes itself from other approaches because of the cooperative and distributed nature of resource allocation and control among a collection of independent multimedia applications.

2. Requirements for Adaptable Networked Multimedia

In order to support adaptable networked multimedia, an underlying framework is necessary that defines how information is structured, composed, accessed, and manipulated, as well as how it is stored and transmitted among sources and sinks. In this section, AMF: the *Amsterdam Multimedia Framework* is presented. To put AMF in context, its description is prefaced with a discussion of the type of multimedia applications it was intended to support and a review of the control issues that the framework must address.

2.1. *Multimedia Application Descriptions: The Document*

Our abstraction for organizing multimedia information is the *document*. A document defines a collection of *data objects* and a description of how these objects interact. Each object may consist of previously-stored information or information that is generated dynamically. Such information can be of either a single data type (such as pure audio or video) or of a composite data type (such as video with embedded audio). An active document is called a *presentation*.

Fig. 2 provides an example of a document-based multimedia application—in this case, a fragment of a walking tour of Amsterdam. This fragment contains a title bar using text data, a description of typical shopping street using video data, several "buttons" using text data that control navigation through the document, a CWI logo using still-image data, and two sets of captions (one in English, one in Dutch) using text data. The document from which this example is taken also has two sound tracks (one in Dutch, one in English) that provide audio commentary during the tour. The data objects can be stored on various servers located throughout the environment. When the document is accessed, each of the individual object streams is sent to a document *player*, which implements any high-level (non-embedded) synchronization constraints among the streams (such as matching the subtitle text with the audio data). Each document, such as the tour of Amsterdam in our example, is specific to a particular application; the player is a general-purpose program that must be able to play many different documents.

The primary advantage of using a document model is that it provides an explicit behavioral specification. This behavioral description can be used to fetch individual data objects by a player, but it can also be used prior to execution to analyze expected application resource use and feasibility for a given environment [BZ92a]. Assuming

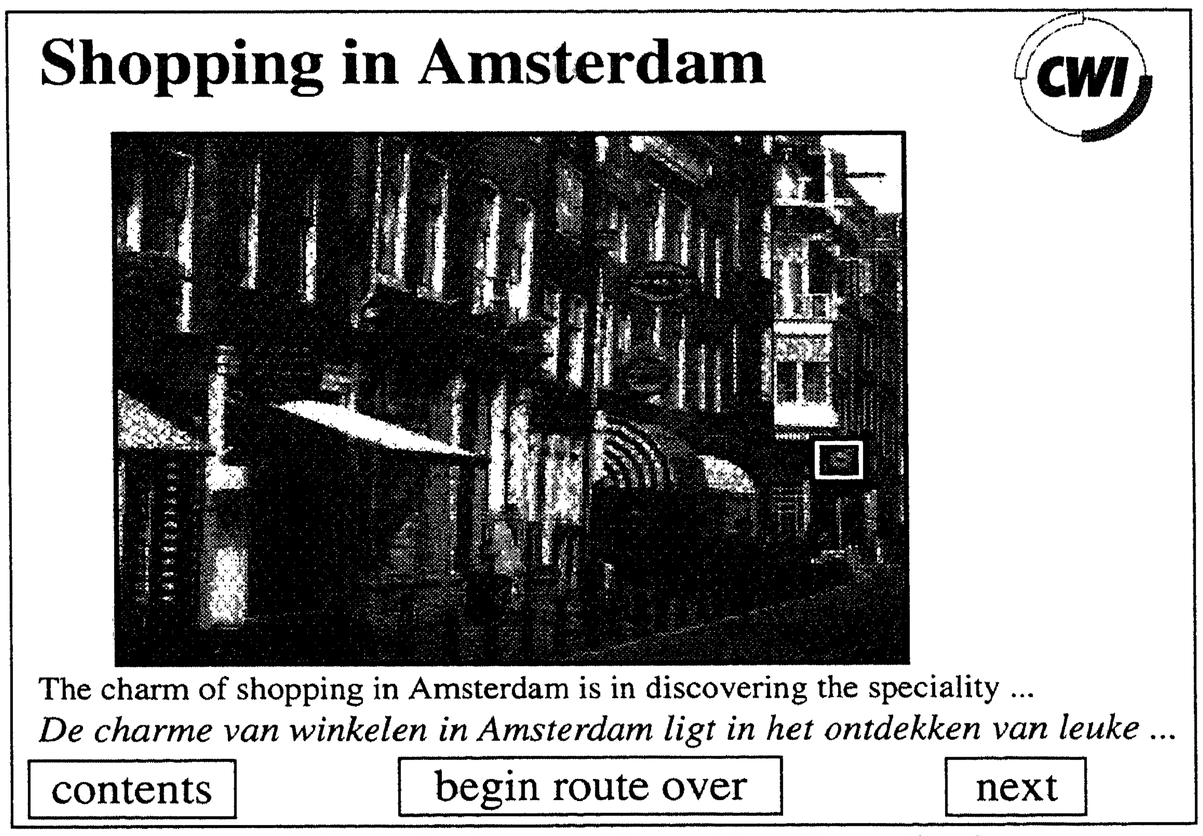

Figure 2. An example multimedia application.
The rectangles along the bottom are navigation controls; the square in the picture is a hyperbutton. The lines of text are captions that accompany multi-lingual audio.

the specification was defined to run in a general-purpose environment (that is, it was not designed for use on one particular platform), the specification can also be used to determine how (and if) the synchronization needs of the application can be supported at run-time [BRL91].

Creating documents using authoring systems or program-based toolkits is typically an arduous task. One motivation for investigating adaptable networked multimedia was to provide reduce the overall effort of producing multimedia presentations by a means of reusing document structures in multiple environments once they were authored [BRL91,HBR93b].

2.2. *Supporting Adaptable Documents: Data-Representation and Document-Content Issues*

During analysis of a document, it is typically assumed that the specification provides a precise description of the needs and characteristics of the application. Our work investigates the use of a specification as a guide to *possible* resource and data use, depending on the resources available at execution time of the document. While pre-execution analysis can provide a useful first step in determining specification feasibility, it cannot resolve all of the issues that may influence the run-time needs or run-time behavior of an application. In defining a basis for adaptable documents, two classes of issues can be identified that influence document analysis and support: issues associated with the physical representations of multimedia data and issues associated with the content-based interactions of users with multimedia data.

2.2.1. *Representation-based issues.*

One major difference between multimedia data and "conventional" electronic data is that multimedia information can require specific service guarantees to preserve synchronization properties of the data. These properties are the consequence of how multimedia data is represented, not the meaning of the data itself. While the representations of each data type vary, there are several common issues that are relevant for all time-sensitive multimedia data:

- *intra-object synchronization:* each component can have synchronization constraints that are related to the type of data being retrieved. For example, the video, audio and caption-text data in Fig. 2 each have their own synchronization constraints. These constraints must be supported by the source environment, the network infrastructure being traversed and the destination environment. These constraints can usually be managed on an end-to-end basis [D90,D91];
- *inter-object synchronization:* in general documents, data will be encoded in separate streams of objects, each of which may be located at different hosts. While inter-object synchronization is often controlled in the context an of application, the composite transfer of data may need to be coordinated to improve system efficiency. For example, synchronization of audio data and caption-text can be done by the application, but it can be done more efficiently using markers placed in the data objects and evaluated by the support software;
- *heterogeneity:* in general environments, all of the presentation workstations will not be identical. Information may need to be adapted at either the source or the

sink to meet the needs of a presentation environment, where the adaptation process may itself have an influence over which parts of a document are available to a user—a process that may also impact scheduling, resource allocation and synchronization with the network.

Bandwidth management can also be included among the representation-related issues. In spite of the trend toward faster networks and more highly-encoded information, the transfer capacity of the various interconnects will remain a critical resource that must be managed—either because application demands will grow or because multiple types of networks will coexist at a site, requiring a degree of coordination and management when allocating local and global resources efficiently.

2.2.2. Content-based issues.

The reason for isolating representation-based issues is to consider ways of providing other than worst-case resource allocation in an adaptable environment. In a similar manner, the actions that occur based on the content of a document will also affect the way that documents are fetched, composed and delivered. These include:

- *user selectivity*: not all of the information available in a document may be used each time the document is accessed; for example, although the document in Fig. 2 supports multilingual audio and/or captions, users usually don't want to hear or read all of the available languages simultaneously. (Note that the selection of desired information is made at run-time—not author-time—and that the selection may be influenced by the facilities available on a given playback platform.)
- *presentation non-linearity:* the order in which objects are accessed and presented depends on the document structure *and* the result of user interaction at run-time. For example, users may want to jump around in a document by scrolling forward or backward or by following *hyperlinks* that have been defined statically or dynamically in the document; in Fig. 2, a small rectangle is visible over a traffic sign in the mid-right portion of the street—selecting this button will transfer the user to a section discussing the merits of getting around by bicycle, car and tram in the city.
- *user flexibility:* in general, documents are activated because a user wishes to obtain information. Given a choice, it is our experience that users will tolerate a lower quality presentation instead of being denied access to a presentation totally. Such lower quality may manifest itself as (slight) delays in the presentation of parts of a document or in the substitution of a lower-resolution form of information for a higher-resolution one. (The term "resolution" is used broadly: it could mean substituting a piece of text for a picture or an audio fragment for a piece of video.)

Each of these factors affects the support mechanisms required to provide adaptability in a document. The notion of *user selectivity* means that static analysis of a document before it is executed may not provide an insight into how a document will actually be used. Similarly, *presentation non-linearity* could result in "jumping" to various parts of a document, each with its own quality of service requirements. As a result, efficient use of an infrastructure will require dynamic rather than static assignment of resources across the network. *User flexibility* means that some degree of run-time

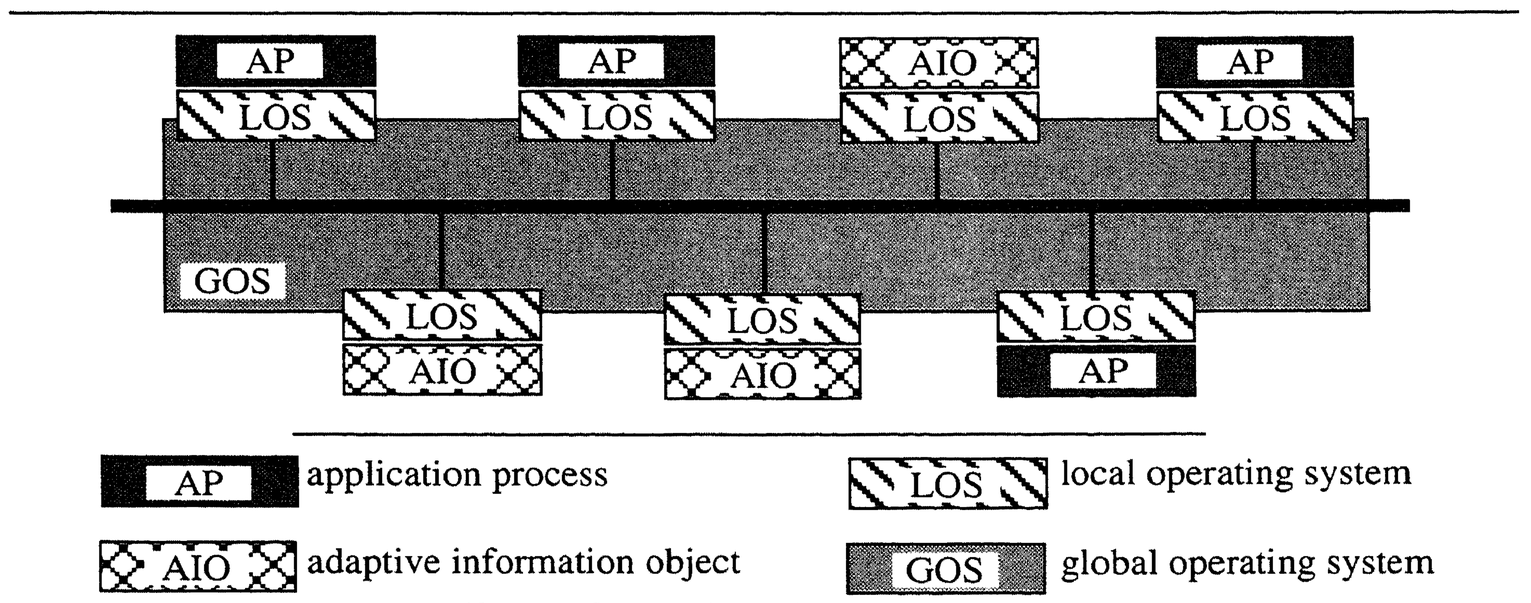

Figure 3. AMF "active" components.

negotiation may need to be supported so that the information presented to the user can be matched to the resources available at the time individual data access requests are made.

3. AMF: The Amsterdam Multimedia Framework

Although many of the techniques required to support representation-based control and, to a lesser extent, content-based control can be taken from existing research results, it is important that these results be applied within a framework that provides an explicit partitioning of control concerns across components in a network infrastructure. This provides a definition of the scope of each technique and can result in better interaction among components. The AMF provides this partitioning for our work.

Fig. 3 illustrates AMF. Here, many applications (AP) communicate with *adaptive information objects* (AIOs) via an infrastructure that is managed by a set of local operating systems and a global operating system. The LOSs and GOS coordinate resource allocation, while the APs and AIOs request and deliver information, respectively. Note that the AMF does not solve the multimedia data transfer problem, it only characterizes the components in an environment and it indicates their interactions. Individual models still need to be developed that implement the general functionality of the framework.

The general structure of AMF is similar to client/server models of networked computing. The difference is that within AMF, the control of multimedia is a cooperative process that requires content-based coordination among all components. For example, assume that one of the APs requests two object streams, each from separate AIOs on two separate hosts. Assume further that one of the AIO is able to meet the service quality request of the application directly, while the other one is not. In this case, both could inform the application of their available degree of service (leaving the application to select an appropriate recovery action) or the two AIOs could communicate with each other to determine if there was a common level of service that both could provide that was acceptable for that application. This could be possible if:

- each of the AIOs was aware of the other's presence,
- each AIO was aware of other's service constraints, either directly from copies of the application specification or by intervention of the GOS and/or each LOS, and

- both AIOs were aware of the range of options acceptable to the application and supportable by the LOS/GOS.

Standard client/server architectures do not provide a basis for this type interaction. As we will show, AMF was specifically designed to provide it.

The underlying assumption of the AMF is that none of the individual components in a transfer has sufficient information to efficiently control resource allocation and inter-object synchronization. A pair of components, such as an AP and a single AIO, is also insufficient, since both end-points could *think* they could provide a degree of service without realizing that the network interconnect was overloaded or that other applications were about to request service. Instead, by using the information in a document specification to be able to look ahead into an application's future behavior, new techniques for resource allocation in its broadest form can be studied for each component. Unlike typical client/server models, these techniques are not based on a notion of lower-level protocol data independence, but rather, on distributing control so that support decisions can be made in light of the needs of applications throughout the network. The scope of AMF control activity is discussed in the following paragraphs.

3.1. The application process (AP).

The role of the AP is to supply the other components within the AMF with a specification of the object streams used by an application, as well as a definition of any inter-object-stream synchronization requirements and a set of options that can be used in providing adaptable control (see section 3.1 for an example). The AP itself functions like the *player* described in section 2.1: it provides a control interface to the user to provide high-level interaction with the network. ("High-level" means operations like *start, stop, pause, fast-forward, seek,* etc.)

In terms of the issues defined in section 2.2, the player provides a user interface to the execution environment, allowing the user to select the parts of a document that need to be played, to navigate through the document and to define the degree to which a document can be adapted. (For example, if a user plays a document on a disconnected portable machine, more tolerance for missing data object may be specified). The player has only a limited role in *implementing* any representation or content-based control operations other than possibly supporting heterogeneous data—this is because the player is a general-purpose interface, while the specification provides the other AMF components with the information necessary to adapt to the needs of the multimedia application.

3.2. The local operating system (LOS).

The LOS serves as a scheduling authority that controls access to I/O devices attached to the local workstation. The LOS would typically allocate resources based on its architecture-specific knowledge of the local operating environment and the document specification provided by the application. While the LOS has the responsibility for controlling the flow of information in and out of the local environment—including presenting information to and receiving information from the network controller(s)—it cannot control activity outside of its environment because it has only a limited view

of what is happening across the network: individual sources may need to sub-sample or pre-synchronize streams within a document or there may be other active documents generating competing requests for resources that are totally outside the scope of a local operating system.

The LOS can participate in managing various data streams for an application by implementing a negotiation process among data providers within the network. The LOS (together with the LOS of an information provider) can also be used to implement the end-to-end protocols associated with intra-object synchronization. Both of these types of service can be provided directly or in conjunction with a GOS. In general, local resource control should be as light-weight as possible; this provides the user with a responsive environment and the rest of the network with a non-intrusive element.

3.3. The global operating system (GOS).

The role of the GOS is to allocate resources on a network-wide basis. It has a view of network activity that is more comprehensive than the APs, the AIOs or the LOS, since it can coordinate activity among independent applications that use the central network but which originate from different workstations. The GOS can provide support that is independent of any particular workstation architecture, acting as moderator or mediator if conflicts arise. (Such a role may be more appropriate in wide-area implementation than in local area networks.) Note that it would be possible for a given implementation model to combine the functions of the LOS and the GOS, although from the point of view of the framework, it is important to recognize that the functions served by both abstractions are different. The primary practical motivation for keeping the LOS and GOS separate is that workstations in a heterogeneous environment cannot be assumed to have similar local operating systems. (They will also most likely have local systems that cannot be altered or adapted to provide extended multimedia support.) The architecture of the GOS allows global concerns to be factored out of the local environment, even to the point that it is possible to design attached-processor implementations supporting GOS functions [BL91].

3.4. Adaptive information objects (AIO).

The AIO provides applications with an interface to stored, synthesized or interactive information. In supporting access requests, the AIO separates the notions of *multimedia information* and *multimedia information representation*. In this way, AIO presents an abstract interface that is used to control access to one of several representations of a block of 'information.' For example, it can be used to substitute an audio description of a video if the user, the user's workstation, the network or the server's host cannot support video delivery. By providing alternative representations of information, the AIO provides *quality of information* support rather than *quality of service* support. (The latter term is more appropriate for representation-dependent manipulations, while the former is more appropriate for content-based selection.) Note that the AIO does not give you something for nothing: it simply provides a general framework that needs to be filled in by data-dependent code and, if appropriate, alternative representations.

Based on the contents of an application specification, the AIO can enter a process of negotiation to provide an application with an appropriate representation of information

that meets the constraints of conditions in the AP, LOS and GOS. The goal of the AMF is that individual implementation models do this negotiation transparently; the motivation for this is that by the time a user goes through the operations necessary to interactively select an alternative representation, the resource constraints that prompted the original negotiation request could have changed. We also assume that most authors would prefer to select the alternative representations which should be used, based on the author's insight into the application domain. (Note that individual AP implementation models may provide both types of control.)

4. Current Status and Summary

The AMF is based on the assumption that resource control in a multimedia network should be adaptable, and that the adaptive process should be distributed over the application, the local operating system, the global (distributed) operation system and the AIOs involved in a transfer. Each of these layers has a specific insight that is important in controlling multimedia transfers. Although each of these insights are necessary, AMF also attempts to limit the scope of any one layer by giving each layer a specific set of concerns to process.

Support for AMF is an on-going research activity. At present, an authoring system has been developed to capture document models in a form that are suited to implementation within the AMF. We have also defined a hyper-information architecture that can be used to describe application-level interactions at runtime within an AMF context. Of the implementation projects, the CMIF authoring environment and its run-time player is the most advanced, while support for general AIO manipulations is at an early stage. Work on the AIO is tied to the development of an LOS/GOS infrastructure and the development of semantic facilities that can be provided to support a wide range of resource, synchronization, and representation control operations. We have performed initial presentation mapping experiments [BW93], but it is too early to draw any conclusions on the utility of this approach.

All of our activity in the Multimedia Kernel Systems project is aimed at understanding the basic relationships that exist in supporting multiple multimedia applications in a heterogeneous network environment. In the current version of our work, this global function is replaced by a separate client and server pair that transparently negotiate the format of the information to be used to satisfy a particular object reference based on the characteristics of the target system, the load on the network, the types of alternative representations that the client will accept, etc. This transparent interaction is important because it offers an opportunity for the system to respond quickly to transient conditions in the environment, but it is difficult to achieve in the light of closed operating systems and multimedia devices. It is our long-term intention to investigate the support of distributed operating systems technology that will allow CMIF specifications (or its successor) to get passed among all of the components of the AMF, each of which will pick out the information it needs to support the synchronization and resource requirements of the application [B92].

Acknowledgments

The general frameworks and various implementation projects described in this article have developed during the past three years as part of the Multimedia Kernel Systems project at CWI. Chief contributors to this project have been Guido van Rossum, Lynda Hardman, Jack Jansen, K. Sjoerd Mullender, Robert van Liere, and Dik Winter. The researchers wish to state their appreciation to Prof.dr. P.C. Baayen, during whose tenure this research program started—and without whose granting of critical resources during the startup phase of the project, this work would never have seen the light of an optical fiber.

References

1 [ABL92]Anderson, T.E., B.N. Bershad, E.D. Lazowska, and H.M. Levy, "Scheduler Activations: Effective Kernel Support for the User-Level Management of Parallelism," *ACM Transactions on Computer Systems*, 10(1), February 1992.

2 [B92]Bulterman, D.C.A., "Synchronization of Multi-Sourced Multimedia Data for Heterogeneous Target Systems," *Proc. 3rd Int. Workshop on Network and Operating Systems Support for Digital Audio and Video,* San Diego, Nov. 1992.

3 [BL91]Bulterman, D.C.A. and R. van Liere: "Multimedia Synchronization and Unix," *Proc. 2nd Int. Workshop on Network and Operating Systems Support for Digital Audio and Video,* Heidelberg, Nov. 1991.

4 [BRL91]Bulterman, D.C.A., G. van Rossum and R. van Liere: "A Structure for Transportable, Dynamic Multimedia Documents," *Proc. Summer 1991 Usenix Conference*, Nashville TN, June 1991.

5 [BW93]Bulterman, D.C.A. and D. T. Winter, "A Distributed Approach to Retrieving JPEG Pictures in Portable Hypermedia Documents," *Proc. IEEE Symp. on Multimedia Technologies and Future Applications*, Southampton, UK, April 1993.

6 *[BZ92a]Buchanan, M.C. and P. T. Zellweger*, "Scheduling Multimedia Documents Using Temporal Constraints", *Proc. 3rd Int. Workshop on Network and Operating Systems Support for Digital Audio and Video,* San Diego, CA, Nov. 1992.

7 [BZ92b]Buchanan, M.C. and P. T Zellweger, "Specifying Temporal Behavior in Hypermedia Documents", *Proceedings of the ACM ECHT'92 Conference on Hypertext*, Milano, Italy Nov 30 - Dec 4 1992.

8 [CSZ92]Clark, D.D., S. Shenker, and L. Zhang. "Supporting Real-Time Applications in an Integrated Services Packet Network", in *Proceedings of ACM SIGCOMM'92*, 1992.

9 [D90]Ferrari, D. "Client Requirements for Real-Time Communication Services", *IEEE Communications Magazine*, November 1990. See also RFC 1193, November, 1990.

10 [D91]Ferrari, D, "Design and Implementation of a Delay Jitter Control Scheme for Packet-Switching Internetworks," *Proc. 2nd Int. Workshop on Network and OS Support for Digital Audio and Video,* Heidelberg, Nov. 1991.

11 [DBL92]Danthin, A., Y. Baguette, G. Leduc, and L. Leonard, "The OSI 95 Connection-mode Transport Service: The Enhanced QoS," in *Proceedings of 4th IFIP Conference on High Speed networking*, Dec. 1992, Liege, Belgium.

12 [FSM91]Fujikawa, K., S. Shimojo, T. Matsuura, S. Nishio and H. Miyahara, "Multimedia Presentation System 'Harmony' with Temporal and Active Media", *Proc. USENIX Multimedia Conference*, June 1991 Nashville TN.

13 [GA91]Govindan, R. and D.P. Anderson. "Scheduling and IPC Mechanisms for Continuous Media," in *Proceedings of the Thirteenth ACM Symposium on Operating Systems Principles*, October 1991.

14 [HBR93a]Hardman, L., D.C.A. Bulterman, and G. van Rossum, "The Amsterdam Hypermedia Model: Extending Hypertext to Real Multimedia," *Hypermedia Journal*, Vol 5(1), May 1993

15 *[HBR93b]Hardman, L., D.C.A Bulterman and G. van Rossum*, "Structured Multimedia Authoring," *Proc. ACM Multimedia '93*, Anaheim CA, August 1993.

16 [HKN91]Hanko, Kuerner, Northcutt, Wall, "Workstation Support for Time-Critical Applications," *Proc. 2nd Int. Workshop on Network and Operating Systems Support for Digital Audio and Video,* Heidelberg, Nov. 1991.

17 [HM91]Hayter, M. and D. McAuley, "The Desk Area Network," *Technical Report No. 228*, Cambridge University Computing Laboratory, 1991.

18 [JST92]Jeffay, K. D.L. Stone T. Talley, and F.D. Smith, "Adaptive, Best-Effort Delivery of Digital Audio and Video Across Packet-Switched Networks," *Proc. 3rd Int. Workshop on Network and Operating Systems Support for Digital Audio and Video,* San Diego, CA, Nov. 1992.

19 [LG91]Little, T.D.C and A. Ghafoor, "Scheduling of Bandwidth-Constrained Multimedia Traffic," *Proc. 2nd Int. Workshop on Network and Operating Systems Support for Digital Audio and Video,* Heidelberg, Nov. 1991.

20 [LMM92]Lesley, I.M., D. McAuley, and S.J. Mullender, "PEGASUS - Operating Systems Support for Distributed Multimedia Systems," *ACM Operating Systems Review* 1(27), January, 1992

21 [RV91]Rangan, P.V. and H. Vin, "Designing File Systems for Digital Video and Audio," *ACM Operating Systems Review*, 25 (5), 1991.

22 *[RJM93]van Rossum, G., J. Jansen, K.S. Mullender and D.C.A. Bulterman, "CMIFed: A Presentation Environment for Portable Hypermedia Documents," Proc. ACM Multimedia '93, Anaheim CA, August 1993.*

23 [T90]Topolcic, C., "Experimental Internet Stream Protocol, Version 2 (ST-II), *Internet RFC 1190*, Oct. 1990.

24 [TNP90]Tokuda, H., T. Nakajima, and P. Rao, "Real-time Mach: Towards a Predictable Real-Time System," in *Proceedings of USENIX Mach Workshop*, October 1990.

25 [VF90]Verma, D. and D. Ferrari. "A Scheme for Real-Time Channel Establishment in Wide-Area Networks", *IEEE JSAC*, Vol. 8, No. 3, April 1990.