REIHE INFORMATIK

1/93

MCAM: An Application Layer Protocol for

R. Keller und W. Effelsberg
Universität Mannheim
Seminargebäude A5
68131 Mannheim

MCAM: An Application Layer Protocol for Movie Control, Access, and Management

Ralf Keller* and Wolfgang Effelsberg†
Praktische Informatik IV
University of Mannheim, Germany

Abstract

Most of the recent work on distributed multimedia systems has concentrated on the transmission, synchronization and operating system support for continuous media data streams. We consider the integrated control of remote multimedia devices, such as cameras, speakers and microphones, to be an important part of a distributed multimedia system. In this paper we describe MCAM, an application layer architecture, service and protocol for Movie Control, Access, and Management in a computer network. The OSI Reference Model is our framework. We present the protocol data units and the Finite State Machine for our application protocol and outline the automatic generation of the implementation code for layer 7 from our formal specification. MCAM allows complete and integrated control of movie data streams and devices in a heterogeneous multimedia network.
Keywords: multimedia systems, digital movies, ISO/OSI, application layer

*keller@pi4.informatik.uni-mannheim.de
†effels@pi4.informatik.uni-mannheim.de

1 Introduction

The aim of Open System Interconnection (OSI) is to allow the interconnection of heterogeneous and distributed computer systems. The protocols of layers 1-6 are now standardized, and several application layer protocols have also reached International Standard level, including X.400 for electronic mail and FTAM for file transfer, access, and management [13]. What is still missing is a standard for the control, access, and management of continuous media (CM).

CM-streams (such as movies) have different requirements than traditional discrete media (e.g. text and graphics) in terms of bandwidth, delay, delay jitter, error control and flow control. Every component in a distributed multimedia system has to guarantee these requirements. Considerable work has to be done to meet the challenge of multimedia data streams. Examples of current research fields are multimedia file systems [23], operating systems [2], synchronization [22], transmission protocols [4], network support [5], and throughput improvement in upper layers [9].

In this paper we address the application layer (layer 7) issues of continuous media data streams within the OSI framework. From our ongoing work on the XMovie project [19] we have gained experience in implementing CM-streams in a network. We have learned from XMovie and other projects that low-level stream services can be implemented on today's computers successfully. Modern MAC protocols for high-speed networks, i.e. DQDB and ATM, allow isochronous transmission at high data rates [8]. Modern transport protocols employ rate-based flow control and forward error correction in order to maintain the timing relationship between sender and receiver ([1], [26]). But application level protocols for CM streams are still missing.

CM streams have to be enriched in layer 7 by support services. We have identified two such services: *movie directory* and *CM equipment control*. Both services are absolutely necessary for the realization of a multimedia service in a distributed heterogeneous environment.

The movie directory is used as a repository for movie information, such as digital image format and storage location. The equipment control service enables the user to control CM equipment attached to remote computer systems, e.g. speakers, cameras, and microphones.

In our model a user can *access* (create, delete and select), *manage* (query and modify attributes), and *control* (playback or record) *movies*. Thus we call our protocol **MCAM** for Movie Control, Access and Management.

Two other approaches to multimedia control in distributed environments are the Lancaster Architecture and Bellcore's Touring Machine. An overview of the two architectures can be found in [25]. Both architectures make use of a repository/name service, and in the Lancaster Architecture it is also possible to control CM equipment. But there is one major distinction between these two architectures and MCAM: We separate the control protocol from the CM stream protocol and include the two new services in the control protocol. Our assumption is that name services and device control have requirements radically different from those of continuous media streams and should thus be carried over a different protocol stack.

In IBM's "Heidelberg Multimedia Communication System" equipment control and stream control are also available but separated from each other [3].

None of these projects provides a formal description of the control protocol, which we consider to be an important step towards standardization.

This paper is organized as follows. In Section 2, we present the model of the movie service. Section 3 describes the service definition. The protocol specification and use of underlying services is outlined in Section 4. Section 5 concludes the paper.

2 Model of the Movie Service

2.1 Streams and Movies

A multimedia application has to handle streams of continuous media data, e.g. video and/or stereo audio streams. In this paper we concentrate on digital movies. Three types of movies can be distinguished: *FILE*, *SHM*, and *LIVE*. At the beginning of its lifetime a movie will be produced by a source, e.g. a camera (LIVE movie). After this it can be stored on disk for later use (FILE movie), transmitted as a data stream through a network or piped through shared memory (SHM movie). The last step of its career will be a movie consumer, e.g. a graphic display – a window from the user point of view (LIVE movie). We will see later that the three types have different functions and requirements; for example live sources cannot be played backwards.

The reason why we introduce the SHM type is for multicast distribution or filters. We can connect a movie source to a shared memory buffer (SHM) which in turn can be connected to multiple sinks, implementing an application-controlled multicast. The SHM can also be used for filtering, compression, etc. An example is shown in Figure 1.

We call an entity reading from an input device and producing a stream (on a network) a *source* of a movie and an entity consuming a stream by writing on an output device a *sink* of a movie. Sources and sinks can be of either of the three movie types FILE, SHM or LIVE.

2.2 Functional Model

The functional model of our system is shown in Figure 2. It consists of four parts: Directory System, Equipment Control System (ECS), Stream Provider System (SPS), and MCAM.

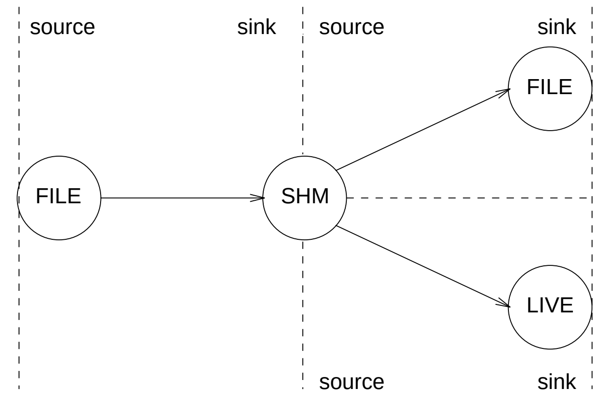


Figure 1: An example of the three movie types

Directory System. We use the ISO/CCITT directory service and protocol to store and access distributed movie information [11]. The X.500 directory standard distinguishes two types of entities: directory user agents (DUAs) and directory system agents (DSAs). Arbitrary information can be stored in the entries of an X.500 directory. Therefore its use for digital movie services is straightforward.

A user interacts with the directory system via a DUA. The directory information is stored by the DSAs. They cooperate to guarantee the consistency of distributed data, and to answer remote and distributed queries.

The directory entry for a movie contains at least the following attributes: a unique name, location (a host), type (FILE, SHM, or LIVE), format (audio or video format), and compression method.

A user can query the directory system using the MCAM service. He can request information about a specific movie or search for movies of a special type or format. If he has the authorization he can create, modify or delete a movie entry.

Equipment Control System. Most of the CM equipment attached to a computer will support more operations than play, pause, resume and stop (basic stream operations). It may be possible to turn, zoom and focus a camera, or to increase or to decrease the volume of a speaker. Therefore we consider remote equipment control to be an important part of a distributed multimedia system.

Our model provides an interface to device operations. MCAM transmits the user requests to an equipment user agent (EUA). The requested operation is performed by an equipment control agent (ECA), which is directly connected to the CM equipment[1]. All ECAs form the equipment control system (ECS), which provides a location-transparent access to CM-equipment.

Stream Provider System. The stream provider system (SPS) provides to MCAM a plain stream service. Such stream services have been proposed in the literature (e.g. [4],

[1]MCAM can use the directory service to determine the location of each ECA.

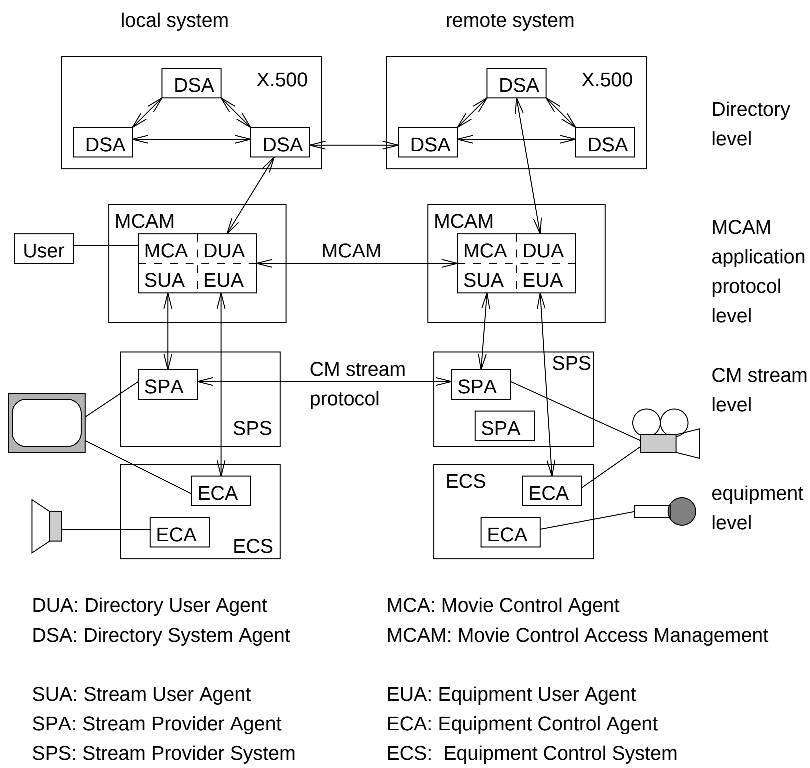


Figure 2: MCAM functional model

[7], [19]), and several research projects report successful implementations. The SPS provides

- a connection service between stream provider agents (SPAs) and
- a stream data interface with the operations play, stop, pause, and resume.

One of the connected SPAs acts as the source and the other as the sink of the stream. The role of each of the SPAs is determined at connection establishment time and cannot be changed until connection release, i.e. the direction of the stream is fixed.

Every play operation creates a *play context*, and a stop releases it. A play context determines which operations are allowed and how they are performed on a sink. On a LIVE or SHM sink new data can only be appended, whereas on a FILE sink new data can be inserted or appended, and old data can be replaced or erased. The play context also carries information about the current value of the parameters for *reliability*, *speed*, *mode*, *quality*, *section* and *direction*. Some of them can be modified by the user, others are fixed (see Table 1; a missing • indicates that the corresponding parameter is fixed).

With the parameter *reliability* the initiator declares whether he demands error-free service or whether he accepts transmission errors. Most of today's multimedia applications do not require 100% reliable transmission because the errors are often not recognizable to human end users (e.g. pixel errors in a video frame). But if the sink is a disk archive for production-quality movies, or if medical data is involved, the movie transmission might have to be error-free. The default depends on the type of the source. The reliability parameter controls the error correction scheme used on the CM data stream.

Parameter	Type of source		
	SHM	LIVE	FILE
Reliability	•	•	•
Speed	•		•
Mode		•	•
Quality		•	
Section			•
Direction			•

Table 1: Movie parameters

Each movie has a default data rate, called the real-time rate. The initiator can request a *speed* (presentation rate) higher, lower, or equal to the real-time rate. E.g. an NTSC video has a real-time rate of 30 frames/second; if the initiator requests a presentation rate of 15, the movie is played at half speed.

Scientific and commercial applications often use slow or fast motion (*mode*) for special effects. Whether this parameter will have any effect depends on the type of source. A FILE movie must be recorded with 50 frames/second to allow a 50% slow motion. Fast motion is easier because uniform skipping is sufficient to create the desired effect.

The ability to use the *quality* parameter depends on the kind of source. Its main purpose is to control the data compression algorithms. The requested quality influences the resulting amount of data (e.g. the quantization factor of JPEG [27]).

If the source is a FILE then the initiator can request transmission of only a *section* of the stream. The default value for this parameter is the whole movie. It is also possible to choose between two *directions*; the movie can be transmitted in the default direction (forward) or in the reverse direction (backwards).

If the responder cannot guarantee either reliablity or speed at connection establishment the request is rejected. In contrast, if the guarantees are violated during operation the responder can react in one of two ways: he can send a notification to the initiator and continue or he can stop and issue a "provider stop". This option can be selected at connection establishment time.

Let us consider an example. A laboratory assistant wants to observe a dangerous experiment with a remote camera. He uses his MCAM user agent to establish a connection between the SPA of the remotely controllable camera (LIVE source) and one of his windows on his screen (LIVE sink), and starts the movie. If he wants to see the details of the experiment, he can turn and zoom the camera to the desired position, using his mouse and the MCAM control protocol over the network.

Movie Control, Access, and Management. All the services provided by MCAM can be accessed at the service access point (SAP) of the MCAM layer. The movie control agent (MCA) uses the directory, ECS, and SPS to provide the MCAM service to the initiator. MCAM itself uses the directory service to obtain information. For example, let us assume that a client requests to connect movie source A with movie sink B. If MCAM has no cached information about A or B, it sends a query to its DUA and uses the information received to fill in the parameters for the SPS call.

3 Service Definition

3.1 Movie Service Provider and Movie Service User

In order to provide remote connections between sources and sinks in a heterogeneous environment, the MCAM component uses a well-defined service and protocol.

Our MCAM service definition uses the abstract model and terminology defined in the OSI Basic Reference Model [12] and the OSI Upper Layer Architecture [10]. Our model describes the interactions between the two movie service users and the movie service provider. Information is passed between a movie service user and the movie service provider by movie service primitives which may carry parameters.

One of the movie service users is defined as the *initiator* (or the user side) and the other is defined as the *responder*. The initiator and the responder both are Multiple Association Control Functions (MACFs) within layer 7. They control an MCAM service element, a directory service element, a stream control service element, and an equipment control service element.

A movie transmission and management service is in many respects similar to a file transfer and management service. Therefore our MCAM service follows closely the FTAM service already standardized by ISO [13]. Similar to FTAM, the movie service is subdivided into regimes. Three types of movie service regimes are defined:

- the MCAM regime which exists while the application association is used for the MCAM protocol, and
- the movie connection regime during which one sink and one source are associated with the MCAM regime
- the movie control regime.

The MCAM service provides for a sequence of movie connection regimes within a MCAM regime and a sequence of movie control regimes within a movie connection regime. Termination of a regime implies termination of all regimes nested within that regime. The nesting of MCAM regimes is shown in Figure 3. The notion of regimes follows the FTAM standard.

3.2 MCAM Service Primitives

This section provides a short description of the MCAM service primitives. For each service, the user of the service (i.e. the application service element that invokes the service primitive) is stated.

MCAM Regime Control. Three services are associated with MCAM regime control. The MCAM regime establishment service (M-INITIALIZE) is used by the initiator to create and bind a MCAM regime to the application association linking the two movie service users. It also establishes MCAM specific information such as authorization and accounting data. The orderly MCAM regime termination service (M-TERMINATE) is used by the initiator to dissolve the MCAM regime and unbind it from the application association between the movie service user and the movie service provider. The abrupt MCAM regime termination service is used by either one of the service users (M-U-ABORT) or the service provider (M-P-ABORT) to dissolve the MCAM regime and its binding to the application unconditionally.

CM Equipment control. The M-EQUIPMENT-CONTROL service primitive allows the integrated handling of remote devices, e.g. to perform a camera zoom, or to adjust the volume of speakers. This service is visible in every regime. It interfaces to the user on the initiator side and to the ECS on the responder side.

Movie Management. The movie management primitives create and delete movie sources and sinks. With the create sink service (M-CREATE-SINK) the initiator can create a new sink (e.g. a new file for the storage of a movie or a movie window on his screen) or select an existing one (e.g. a speaker at a workstation) and register this new

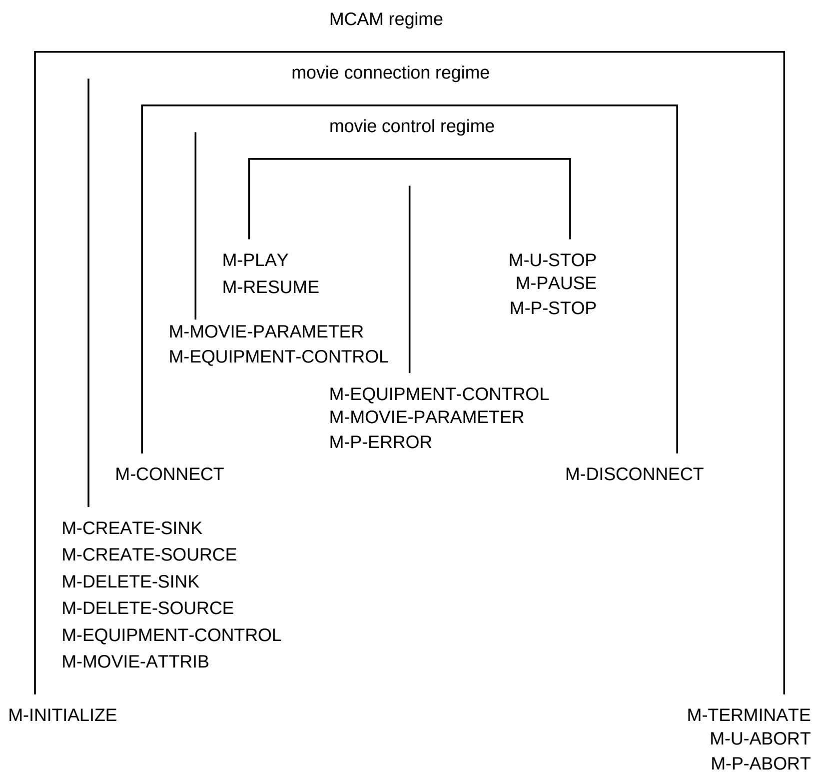


Figure 3: Nesting of movie service regimes

sink with the MCAM server and the directory service. The create source service (M-CREATE-SOURCE) creates a new source or selects an existing one and registers it with the MCAM server and the directory service. Depending on the lifetime parameter of the create source or create sink service, the movie object will exist only for the duration of the association or until an explicit delete is performed. The delete source service (M-DELETE-SOURCE) enables the initiator to release the binding between the MCAM service and the specified sink in either of two ways: the sink ceases to exist or only the registration of the sink at the MCAM service is removed. In its functionality the delete sink service (M-DELETE-SINK) corresponds to the delete source service. Both services remove the respective directory entry. If the initiator wants to interrogate or to modify the attributes of the specified movie source/sink he can use the movie attribute service (M-MOVIE-ATTRIB).

Connection Regime. With the movie connect service (M-CONNECT) the initiator can connect one movie source to one movie sink. This defines the direction of the data flow and establishes access rights for movie control. The disconnect service (M-DISCONNECT) enables the initiator to release the specified connection.

Movie Parameters. The adjustment of movie parameters is performed by a single service: M-MOVIE-PARAMETER. The initiator can interrogate and modify the value of each of the parameters described in Section 2.2. This service is visible in both movie connection and movie control regime.

Movie Control Regime. In order to control movies, MCAM defines five services. The play service (M-PLAY) is used by the initiator to start a stream; the source starts sending and the sink begins to receive in a way determined by the kind of source and sink. Also the play context is established. The pause service (M-PAUSE) enables the initiator to stop the movie; the play context is preserved. The resume service (M-RESUME) restarts a movie which was stopped by the pause service. With the user stop service (M-U-STOP) the initiator can stop the movie and release the play context. The provider stop service (M-P-STOP) is used by the provider to signal the initiator that the end of the movie has been reached or that an unrecoverable error has occurred and the movie has been stopped.

Provider Error. The provider error service (M-P-ERROR) is used by the responder or by the provider to signal the initiator that an error occurred, without stopping the movie. This can happen if the reliability or speed parameters cannot be met at runtime, and the initiator had stated at connection establishment time that he wants the movie to continue in such cases.

Obviously, most of the services described above require *access control*, e.g. a remote microphone can only be turned

Service primitive	MPDU	Carried by
M-INITIALIZEreq	INIRQ	A-ASSOCIATE
M-INITIALIZEresp	INIRP	A-ASSOCIATE
M-TERMINATEreq	TERRQ	A-RELEASE
M-TERMINATEresp	TERRP	A-RELEASE
M-U-ABORTreq	UABRQ	A-ABORT
M-P-ABORTind	UABRQ	A-ABORT

Table 2: MCAM protocol data units – regime establishment

on by authorized users. The details of the access control and authentification mechanisms are beyond the scope of this paper.

4 Protocol Specification

The MCAM service is distributed in nature. In order to operate in a heterogeneous environment, we have to define the MCAM protocol precisely.

We are convinced that the new generation of high-speed networks, and in particular ATM, provides the technical foundation for the transmission of digital movies in worldwide networks. But equipment is heterogeneous, and will probably always be so. In order to enable movie transmission in a heterogeneous network the *standardization* of a movie transmission and control protocol is of major importance. We therefore provide a formal definition of our MCAM protocol, using the widely accepted OSI conventions.

In this section we describe each of the MCAM protocol data units (PDUs), their mappings to underlying services, and give an example for a state/transition diagram for correct sequencing.

4.1 MCAM Protocol Data Units

ISO has defined a data description language for application layer PDUs called ASN.1 (Abstract Syntax Notation One [16]). We are currently in the process of defining all MCAM PDUs in ASN.1. This allows us to generate C/C++ data structures for our implementation automatically. The ASN.1 specification can also be used to automatically create an encoder/decoder for MCAM PDUs for our runtime implementation. ASN.1 tools are now provided with many ISO/OSI communication software packages; a well-known example is ISODE [24]. The PDUs of MCAM Regime Control are mapped to the ACSE service primitives, all others are mapped directly to the Presentation Layer service.

ACSE Mappings. The application association and related application context for the movie protocol are established by the Association Control Service Element (ACSE [14]). The service primitives used for initializing and terminating the instance of the MCAM regime carry the parameters of the association context and authorization and accounting information. The mapping of the MCAM application PDUs (MPDUs) to the ACSE service is shown in Table 2.

Service primitive	MPDU	Carried by
M-CREATE-SINKreq	CSIRQ	P-DATA
M-CREATE-SINKresp	CSIRP	P-DATA
M-CREATE-SOURCEreq	CSORQ	P-DATA
M-CREATE-SOURCEresp	CSORP	P-DATA
M-DELETE-SINKreq	DSIRQ	P-DATA
M-DELETE-SINKresp	DSIRQ	P-DATA
M-DELETE-SOURCEreq	DSORQ	P-DATA
M-DELETE-SOURCEresp	DSORQ	P-DATA
M-EQUIPMENT-CONTROLreq	ECORQ	P-DATA
M-EQUIPMENT-CONTROLresp	ECORP	P-DATA
M-MOVIE-ATTRIBreq	MATRQ	P-DATA
M-MOVIE-ATTRIBresp	MATRP	P-DATA
M-CONNECTreq	CONRQ	P-DATA
M-CONNECTresp	CONRP	P-DATA
M-DISCONNECTreq	DISRQ	P-DATA
M-DISCONNECTresp	DISRP	P-DATA
M-MOVIE-PARAMETERreq	PARRQ	P-DATA
M-MOVIE-PARAMETERresp	PARRP	P-DATA
M-PLAYreq	PLYRQ	P-DATA
M-PLAYresp	PLYRP	P-DATA
M-PAUSEreq	PAURQ	P-DATA
M-PAUSEresp	PAURP	P-DATA
M-U-STOPreq	USTRQ	P-DATA
M-U-STOPresp	USTRP	P-DATA
M-P-STOPreq	PSTRQ	P-DATA
M-RESUMEreq	RESRQ	P-DATA
M-RESUMEresp	RESRP	P-DATA
M-P-ERRORreq	PERRQ	P-EXPEDI-TED-DATA

Table 3: MCAM protocol data units – control, access, and management

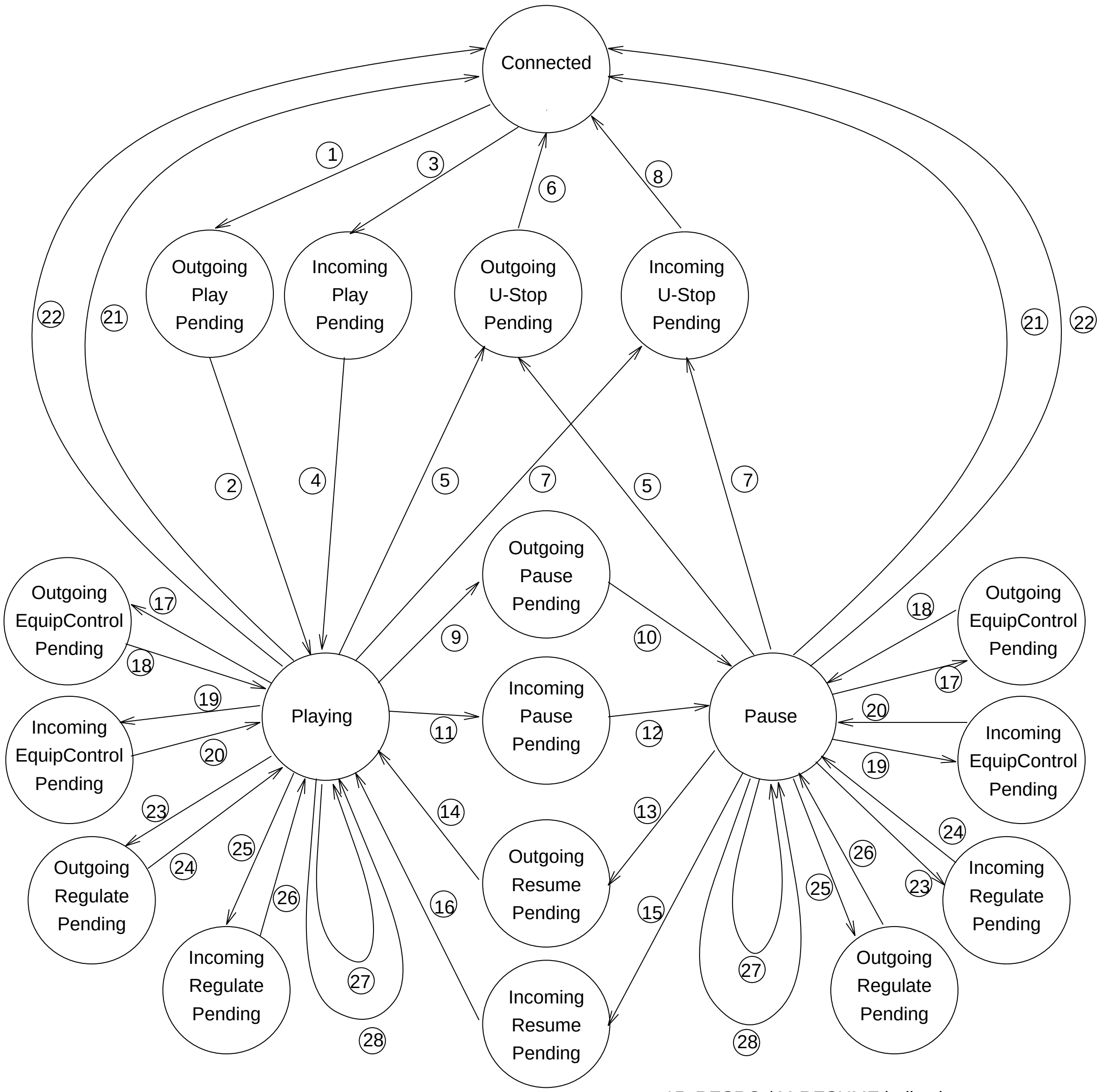
Presentation Service Mappings. All remaining MCAM protocol data units are mapped to the OSI Presentation Service [15]. Only one presentation context is active when the MCAM regime is established. Remember that MCAM handles only the interchange of control information between two movie service users. No CM data are transferred on an MCAM association. The transfer of MCAM PDUs has no real-time or high-bandwidth requirements, and thus no special presentation context is needed. The mapping of MCAM application PDUs to the Presentation Services is shown in Table 3.

4.2 State Transition Diagram

In addition to defining the exact PDU formats in ASN.1 and their mapping to lower layer services, a protocol specification must state the allowable sequences of events. We use a conventional state/transition diagram to specify the dynamic behavior of the MCAM protocol. Since the diagrams of the MCAM regime and the movie connection regime are relatively simple, only the more interesting diagram of the movie control regime is presented in Figure 4.

4.3 Example

A simple example of the usage of the movie service and the corresponding protocol flow is shown in Figure 5 in the form of a time sequence diagram. An initiator establishes an asso-



1 M-PLAY request / PLYRQ
2 PLYRP / M-PLAY confirm
3 PLYRQ / M-PLAY indication
4 M-PLAY response / PLYRP
5 M-U-STOP request / USTRQ
6 USTRP / M-U-STOP confirm
7 USTRQ / M-U-STOP indication
8 M-U-STOP response / USTRP
9 M-PAUSE request / PAURQ
10 PAURP / M-PAUSE confirm
11 PAURQ / M-PAUSE indication
12 M-PAUSE response / PAURP
13 M-RESUME request / RESRQ
14 RESRP / M-RESUME confirm
15 RESRQ / M-RESUME indication
16 M-RESUME response / RESRP
17 M-EQUIPMENT-CONTROL request / ECORQ
18 ECORP / M-EQUIPMENT-CONTROL confirm
19 ECORQ / M-EQUIPMENT-CONTROL indication
20 M-EQUIPMENT-CONTROL response / ECORP
21 M-P-STOP request / PSTRQ
22 PSTRQ / M-P-STOP request
23 M-MOVIE-PARAMETER request / PARRQ
24 PARRP / M-MOVIE-PARAMETER confirm
25 PARRQ / M-MOVIE-PARAMETER indication
26 M-MOVIE-PARAMETER response / PARRP
27 M-P-ERROR request / PERRQ
28 PERRQ / M-P-ERROR indication

Figure 4: State transition diagram of the movie control regime

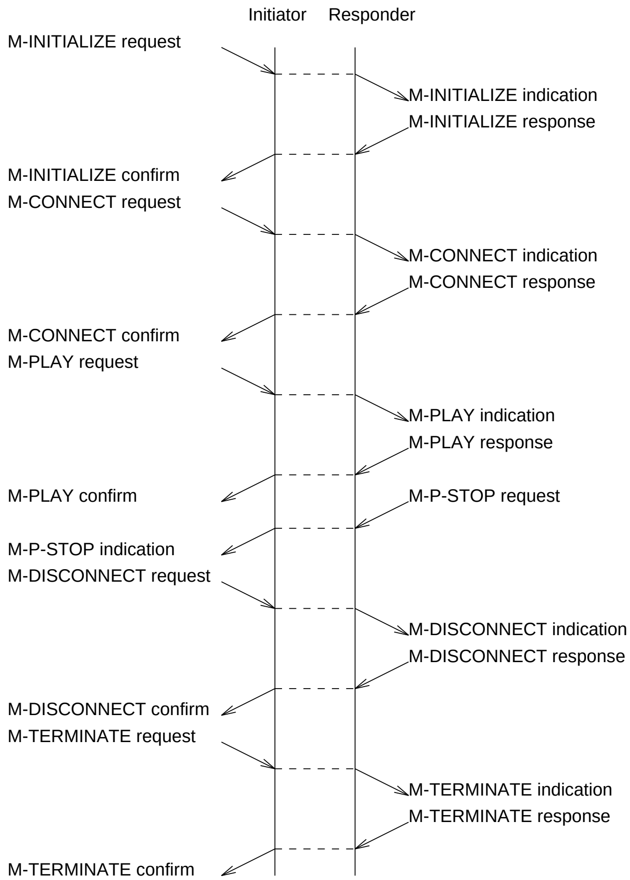


Figure 5: Time sequence diagram of a movie control example

ciation and connects a FILE source and a LIVE sink. Then he issues a play command. When the end of the FILE source is reached, the movie automatically stops, and the responder sends a provider stop signal to the initiator. The initiator then disconnects the sink and the source and terminates the association.

5 Summary, Status and Outlook

We have introduced MCAM, a new OSI Application Layer Services Element for the access, control, and management of digital movies in open networks. The movie control service is enriched by support services for the storage and retrieval of movie information and the control of CM equipment attached to multimedia computer systems. The functional model of MCAM was presented, and the main components and their interworking were described.

The formal MCAM service definition and protocol specification were outlined. These formal definitions can be used to derive an implementation automatically.

We are currently writing a complete formal description of the MCAM protocol in Estelle, and an abstract syntax for the movie service using ASN.1. Estelle [17] is one of three internationally standardized specification techniques (Formal Description Techniques: FDTs). It is based on the Extended Finite State Machine (EFSM) model. Compared to hand-written code, an EFSM specification is more natural and easier to understand for the developer. Using an FDT specification and a code generator has many advantages:

- the code generated from an FDT has fewer bugs than hand-written code
- the code is well structured and easy to understand
- protocol machine aspects are nicely separated from other aspects of a communication system node, such as buffer management, timer management, file I/O, etc. and
- the code is more portable than hand-written code.

It is often claimed that generated code is less efficient than hand-written code. We do not see a good reason for *inherent* inefficiency in generated code provided that the code generator is well implemented. In another project our research group is investigating possible performance enhancements by parallel execution. The overall speedup gained by parallel execution is much higher than a possible speedup by manual code optimization [9]. The parallel code runs on OSF/1, and is also generated automatically from the same Estelle specification [6].

Work is also in progress to refine and improve our own CM stream protocol MTP [20] to provide the following services:

- reliable and unreliable stream service using an adjustable forward error correction mechanism
- support for several uncompressed and compressed formats for movies[2] and
- rate-based flow control for isochronous transmission.

The improved Movie Transmission Protocol (MTP) and other parts of the XMovie project form the basis of our SPS implementation at the University of Mannheim. The XMovie prototype currently allows the transmission of digital movies between RISC workstations from DEC, Sun and IBM at up to 25 frames/s. The movies are displayed in an X window, using our own extension to the X Window System [18].

We have conducted initial experiments with QUIPU, the X.500 implementation of ISODE, as a basis for our movie directory service. Our movie directory prototype is not yet completed.

In a joint project with IBM's European Networking Center in Heidelberg we have also developed a remote camera control system. The camera operator uses his mouse, scroll bars and push buttons on his window surface to control a remote video camera [3].

As a final goal, we envision several SPAs, ECAs, and MCAM entities in an ATM network to provide a digital movie service in a heterogeneous environment using standardized transmission and control protocols.

[2] With the exception of MPEG [21], no international or de facto standard for the format of stored digital CM streams exists. Therefore we have developed our own format to store sequences of audio samples or video frames in a great variety of both compressed and uncompressed formats.

Acknowledgement

We would like to thank Bernd Lamparter for his invaluable and ongoing contributions to the XMovie project. The remote camera control system would not have been possible without the help of our colleagues at IBM ENC Heidelberg, in particular Ralf Steinmetz.

References

[1] Ernst W. Biersack. Performance Evaluation of Forward Error Correction in ATM Networks. *Computer Communication Review*, 22(4):248–257, Oktober 1992.

[2] Dick C.A. Bulterman and Robert van Liere. Multimedia Synchronisation and UNIX. In *Network and Operating System Support for Digital Audio and Video, 2nd International Workshop, Heidelberg, Germany, November 1991*, pages 108–119. Lecture Notes in Computer Science 614, Springer-Verlag, Heidelberg, 1992.

[3] Andreas Cramer et al. The Heidelberg Multimedia Communication System: Multicast, Rate Enforcement and Performance on Single User Workstations. Technical Report 43.9212, IBM European Networking Center, Heidelberg, 1992.

[4] Bert J. Dempsey, W. Timothy Strayer, and Alfred C. Weaver. Adaptive Error Control for Multimedia Data Transfers. In *International Workshop on Advanced Communication and Applications for High Speed Networks*, pages 279–288, Munich, Germany, March 1992.

[5] Domenico Ferrari, Anindo Banerjea, and Hui Zhang. Network Support for Multimedia. Technical Report TR-92-072, International Computer Science Institute, Berkeley, CA, November 1992.

[6] Stefan Fischer. Generierung paralleler Systeme aus Estelle-Spezifikationen. Master's thesis, Lehrstuhl für Praktische Informatik IV, Universität Mannheim (in German), 1992.

[7] Simon Gibbs. Composite Multimedia and Active Objects. *ACM SIGPLAN Notices*, 26(11):97–112, November 1991.

[8] Rainer Händler and Manfred N. Huber. *Intergrated Broadband Networks: An Introduction to ATM-Based Networks.* Addison-Wesley Publishing Company, Wokingham, England, 1991.

[9] Bernd Hofmann, Wolfgang Effelsberg, Thomas Held, and Hartmut König. On the Parallel Implementation of OSI Protocols. In *Proceedings IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems (HPSC'92)*, Tuscon, Arizona, USA, February 1992.

[10] Information processing systems – Open Systems Interconnection – Application Layer Structure. International Standard ISO 9545.

[11] Information processing systems – Open Systems Interconnection – The Directory – Overview of Concepts, Models, and Service. International Standard ISO 9594-1, December 1988.

[12] Information processing systems – Open Systems Interconnection – Basic Reference Model. International Standard ISO 7498, 1984.

[13] Information processing systems – Open Systems Interconnection – File Transfer, Acces and Management. International Standard ISO 8571, December 1988.

[14] Information processing systems – Open Systems Interconnection – Service Definition and Protocol Specification for the Association Control Service Element. International Standard ISO 8649/50, 1988.

[15] Information processing systems – Open Systems Interconnection – Connection Oriented Presentation Service Definition and Protocol Specification. International Standard ISO 8822/23, 1987.

[16] Information processing systems – Open Systems Interconnection – ASN.1 and its Encoding Rules. International Standard ISO 8824/25, 1988.

[17] Information processing systems – Open Systems Interconnection – Estelle: A formal description technique based on an extended state transition model. International Standard ISO 9074, 1989.

[18] Ralf Keller, Bernd Lamparter, and Wolfgang Effelsberg. Erweiterung von X für digitale Filme. *PIK – Praxis der Informationsverarbeitung und Kommunikation (in German)*, 15(4):196–204, October 1992.

[19] Bernd Lamparter and Wolfgang Effelsberg. X-MOVIE: Transmission and Presentation of Digital Movies under X. In *Network and Operating System Support for Digital Audio and Video, 2nd International Workshop, Heidelberg, Germany, November 1991*, Lecture Notes in Computer Science 614, pages 328–339. Springer-Verlag, Heidelberg, 1992.

[20] Bernd Lamparter, Wolfgang Effelsberg, and Norman Michl. MTP: A Movie Transmission Protocol for Multimedia Applications. *Computer Communication Review*, 22(3):71–72, July 1992.

[21] Didier Le Gall. MPEG: A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4):46–58, April 1991.

[22] Tom D.C. Little, Arif Ghafoor, C.Y.R. Chen, C.S. Chang, and P.B. Berra. Multimedia Synchronisation. *IEEE Data Engineering*, 14(3):26–35, September 1991.

[23] P. Venkat Rangan and Harrick M. Vin. Designing File Systems for Digital Video and Audio. *ACM Operating System Review*, 25(5):81–94, October 1991.

[24] Marshall T. Rose. *The Open Book: A Practical Perspective on OSI.* Prentice Hall, Englewood Cliffs, 1990.

[25] Lillian Ruston, Gordon Blair, Geoff Coulson, and Nigel Davies. Integrating Computing and Telecommunications: A Tale of two Architectures. In *Network and Operating System Support for Digital Audio and Video, 2nd International Workshop, Heidelberg, Germany, November 1991*, pages 57–68. Lecture Notes in Computer Science 614, Springer-Verlag, Heidelberg, 1992.

[26] W. Timothy Strayer, Bert J. Dempsey, and Alfred C. Weaver. *XTP: The Xpress Transfer Protocol.* Addison-Wesley Publishing Company, 1992.

[27] Gregory K. Wallace. The JPEG Stillpicture Compression Standard. *Communications of the ACM*, 34(4):30–44, April 1991.