

Information technology — Computer graphics and image processing — Presentation Environment for Multimedia Objects (PREMO) —

Part 3: Multimedia Systems Services

ICS 33.160.60; 35.140

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

BSI
Documents

AVA

Document Sharing Hub

National foreword

This British Standard reproduces verbatim ISO/IEC 14478-3:1998 and implements it as the UK national standard.

The UK participation in its preparation was entrusted to Technical Committee IST/31, Computer graphics and image processing, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the BSI Standards Catalogue under the section entitled “International Standards Correspondence Index”, or by using the “Find” facility of the BSI Standards Electronic Catalogue.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, pages i and ii, the ISO/IEC title page, pages ii to iv, pages 1 to 63 and a back cover.

This standard has been updated (see copyright date) and may have had amendments incorporated. This will be indicated in the amendment table on the inside front cover.

Amendments issued since publication

Amd. No.	Date	Comments

This British Standard, having been prepared under the direction of the DISC Board, was published under the authority of the Standards Committee and comes into effect on 15 March 1999

© BSI 05-1999

ISBN 0 580 30077 3

Contents

	Page
National foreword	Inside front cover
Foreword	iv
Text of ISO/IEC 14478-3	1

INTERNATIONAL STANDARD

**ISO/IEC
14478-3**

First edition
1998-12-15

Information technology — Computer graphics and image processing — Presentation Environment for Multimedia Objects (PREMO) —

Part 3: Multimedia Systems Services

*Technologies de l'information — Infographie et traitement d'images —
Environnement de présentation d'objets multimédia (PREMO) —*

Partie 3: Services pour les systèmes multimédia



Contents

	Page
Foreword	iv
Introduction	1
1 Scope	2
2 Normative references	3
3 Definitions	3
3.1 PREMO Part 1 definitions	3
3.2 PREMO Part 2 definitions	3
3.3 Additional definitions	4
4 Symbols and abbreviations	4
5 Conformance	5
6 Overview of the Multimedia Systems Services	5
6.1 Introduction	5
6.2 Object framework	5
6.3 Subtyping diagram	7
6.4 MSS object life cycle	8
7 Configuration objects	9
7.1 Introduction	9
7.2 Format objects	9
7.3 Transport and Media Stream Protocol objects	10
7.4 Quality of Service Descriptor objects	10
8 Stream Controls	11
8.1 StreamControl objects	12
8.2 SyncStreamControl objects	14
9 Devices, Resources	14
9.1 Virtual Resources	14
9.1.1 Configuration objects on virtual resources	15
9.1.2 Stream control	15
9.1.3 Resource management	15
9.1.4 Quality of Service Management	16
9.2 Virtual Devices	17
9.2.1 Processing element	18
9.2.2 Ports	18
9.2.3 Streams	18
9.2.4 Port configurations	19
9.3 Virtual Connections	20
9.3.1 Examples for connection agreement	20
9.3.2 Connection establishment	21
9.3.2.1 Unicast and multicast	21
9.4 Groups	22
9.4.1 Resource acquisition and end-to-end QoS	23
9.4.2 Stream control	23
9.5 Logical Devices	24
10 Functional specification	25
10.1 Introduction	25
10.2 Non-object data types	25
10.3 Exceptions	25
10.4 Structures	26
10.4.1 Port information structure	26
10.5 Configuration object	26
10.5.1 Format objects	26
10.5.1.1 Format object	26

		Page
10.5.2	Transport and Multimedia Stream Protocol objects	26
10.5.2.1	MultimediaStreamProtocol objects	26
10.5.2.2	IntraNodeTransport objects	27
10.5.2.3	InterNodeTransport objects	27
10.5.3	Quality of Service objects	27
10.6	Stream Controls	28
10.6.1	StreamControl object	28
10.6.2	SyncStreamControl object	29
10.7	Devices, resources	30
10.7.1	VirtualResource object	30
10.7.2	VirtualDevice object	32
10.7.3	Virtual connections	34
10.7.3.1	VirtualConnection object	34
10.7.3.2	VirtualConnectionMulticast object	35
10.7.4	Group object	36
10.7.5	LogicalDevice object	37
11	Component specification	37
Annex A (normative)	Overview of PREMO MSS objects	39
Annex B (informative)	A typical example scenario for MSS usage	41
Annex C (informative)	Basic Devices	42
C.1	Format objects	44
C.1.1	Video formats	44
C.1.2	Audio formats	45
C.1.3	CATV format	45
C.1.4	MIDI format	46
C.2	Digital stream controls	46
C.3	Video and audio processing	46
C.3.1	Video processing	46
C.3.2	Audio processing	47
C.4	Specific devices	47
C.4.1	Defining a device	47
C.4.2	Video	47
C.4.3	Audio	47
C.4.4	Files	48
C.4.5	CD player	49
C.4.6	CATV tuner	49
C.4.7	MIDI device	49
C.4.8	External resources	49
C.5	Functional Specification	49
C.5.1	Area of interest for video objects	50
C.5.2	Format objects	50
C.5.3	Digital Stream Control	54
C.5.4	Video and audio processing	55
C.5.5	Specific devices	56
Annex D (informative)	Examples of virtual connection settings	61
D.1	Hardware connection example	61
D.2	Direct connection example	61
D.3	Local connection example	61
D.4	Network connection example	63

Descriptors: Data processing, information interchange, graphic data processing, image processing, video data, audio data, coding (data conversion), coded representation.

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, government and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee ISO/IEC JTC1. Draft International Standards adopted by the joint technical committees are circulated to the national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

ISO/IEC 14478-3 was prepared by Joint Technical Committee ISO/IEC JTC1, *Information technology*, Subcommittee SC24, *Computer graphics and image processing*.

This International Standard currently consists of the following four parts under the general title *Information technology — Computer graphics and image processing — Presentation environments for multimedia objects (PREMO)*:

- *Part 1: Fundamentals of PREMO*;
- *Part 2: Foundation Component*;
- *Part 3: Multimedia Systems Services*;
- *Part 4: Modelling, Rendering, and Interaction Component*.

Annex A forms an integral part of this part of ISO/IEC 14478. Annex B to Annex D are for information only.

Introduction

The Multimedia Systems Services (MSS) component of PREMO provides a standard set of services that can be used by multimedia application developers in a variety of computing environments. Enabling multimedia applications in a heterogeneous, distributed computing environment is the design motivation for the MSS. This is an increasingly prevalent computing model, and a solution that meets the needs of this environment can more easily be scaled to stand-alone systems than vice versa.

The principal reasons for defining the MSS are:

- a) provide abstractions and mechanisms that make it possible for applications to deal with the problems of distributed multimedia computing successfully;
- b) facilitate the implementation of complex applications, such as video conferencing;
- c) provide abstractions that make it possible for applications to deal with media devices without regard to specific characteristics of the platform, attached devices, or the network(s) connecting the platforms and devices;
- d) to provide a standard methodology, especially for handling “live” data;
- e) insure scalability to large organizations;
- f) insure adequate performance in adverse conditions;
- g) facilitate Quality of Service commitments; and
- h) consider the time critical nature of the data.

The primary goal of the MSS is to provide an infrastructure for building multimedia computing platforms that support interactive multimedia applications dealing with synchronized, time-based media in a heterogeneous distributed environment. Operation in a distributed environment is important because of significant trends in the computer industry towards client/server and collaborative computing. Another significant trend is towards multimedia enabled computing. The inevitable result will be an intersection of these trends to produce a distributed multimedia environment with a topology similar to Figure 1.

The MSS is intended to address a broad range of application needs. It extends the multimedia capabilities of stand-alone computers to capabilities that are usable both locally and remotely. The Multimedia Systems Services gives applications the ability to handle:

- i) live data remotely;
- j) stored data remotely;
- k) both live and stored data simultaneously;
- l) multiple kinds of data simultaneously;
- and
- m) new kinds of devices and media types.

To provide support for remote media device control and remote media access that derive from the above application scenarios, the Multimedia System Services uses two distinct mechanisms. To support interaction with remote objects, the Multimedia Systems Services depends upon an underlying object model and infrastructure, as described in ISO/IEC 14478-1 (PREMO). To support the media independent streaming of time critical data, the Multimedia Systems Services defines a Media Stream Control.

The MSS does not address:

- n) encryption and security;
- o) intellectual property rights and accounting;
- p) scripting;
- q) user interfaces; or
- r) sharing of data between applications.

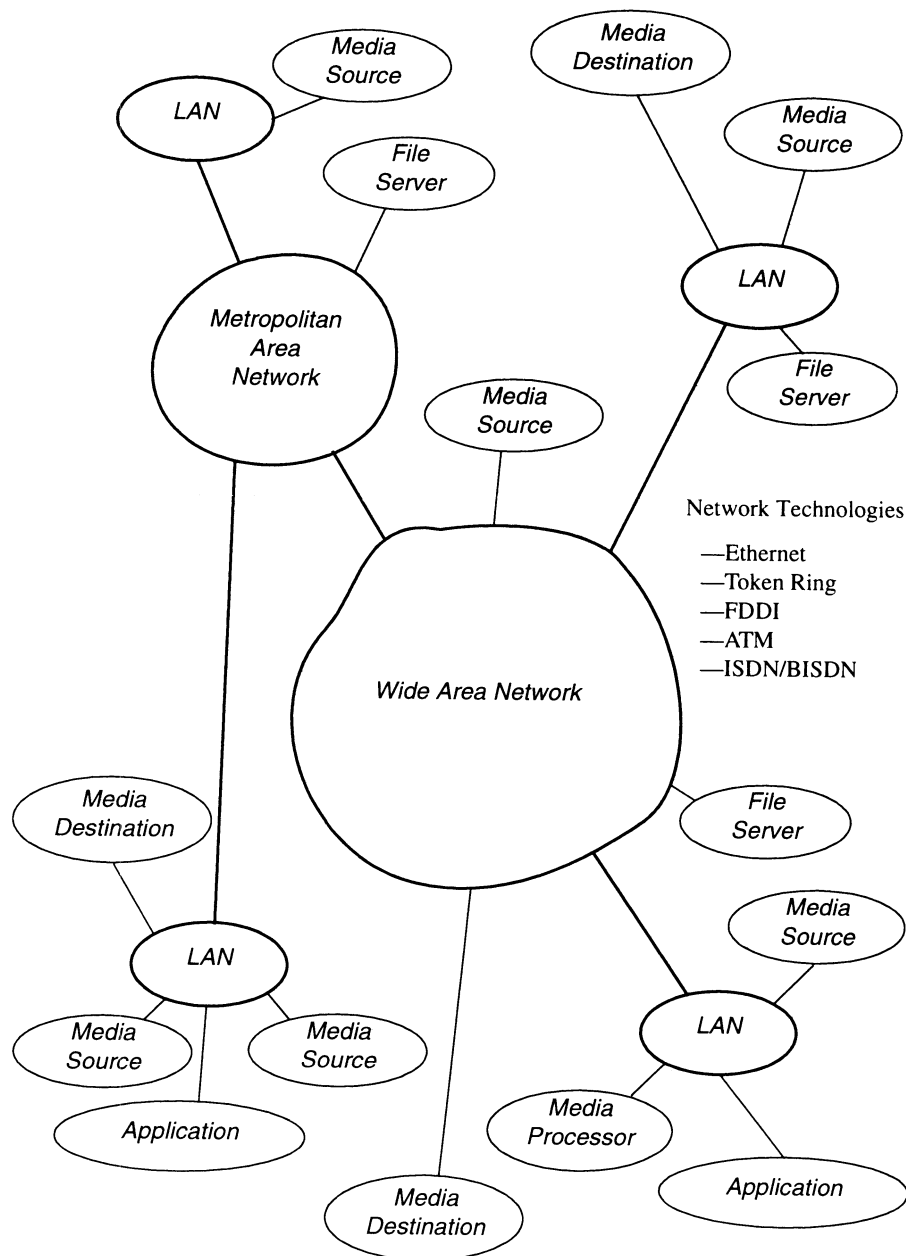


Figure 1 — Distributed multimedia environment

1 Scope

This part of ISO/IEC 14478 defines a standard set of multimedia system services that can be used by multimedia application developers in a variety of computing environments. The focus is on enabling multimedia applications in a heterogeneous, distributed computing environment. Throughout this part of ISO/IEC 14478, this component will also be referred to as "Multimedia Systems Services", and abbreviated as MSS.

The Multimedia Systems Services constitutes a framework of “middleware” — system software components lying in the region between the generic operating system and specific applications. As middleware, the Multimedia Systems Services marshals lower-level system resources to the task of supporting multimedia processing, providing a set of common services which can be used by multimedia application developers.

The Multimedia Systems Services encompasses the following characteristics:

- a) provision of an abstract type for a media processing node, extensible through subtyping to support abstractions of real media processing hardware or software;
- b) provision of an abstract type for the data flow path or the connection between media processing nodes, encapsulating low-level connection and transport semantics;
- c) grouping of multiple processing nodes and connections into a single unit for purposes of resource reservation and stream control;
- d) provision of a media dataflow abstraction, with support for a variety of position, time and/or synchronization capabilities;
- e) separation of the media format abstractions from the dataflow abstraction;
- f) synchronous exceptions and asynchronous events;
- g) application visible characterization of object capabilities;
- h) registration of objects in a distributed environment by location and capabilities;
- i) retrieval of objects in a distributed environment by location and constraints;
- j) definition of a Media Stream Protocol to support media independent transport and synchronization.

The Multimedia Systems Services rely on the object model of ISO/IEC 14478-1 (Fundamentals of PREMO) and the object types and non-object data types defined in ISO/IEC 14478-2 (PREMO Foundation Component).

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 14478. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of this international standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 14478-1:1998, *Information technology — Computer graphics and image processing — Presentation Environment for Multimedia Objects (PREMO) — Part 1: Fundamentals of PREMO*.

ISO/IEC 14478-2:1998, *Information technology — Computer graphics and image processing — Presentation Environment for Multimedia Objects (PREMO) — Part 2: Foundation Component*.

ISO/IEC 14478-4:1998, *Information technology — Computer graphics and image processing — Presentation Environment for Multimedia Objects (PREMO) — Part 4: Modelling, Rendering, and Interaction Component*.

ISO/IEC 10918-1:1994, *Information technology — Digital Compression and Coding of Continuous-Tone Still Images (JPEG)*.

ISO/IEC 11172:1992, *Information technology — Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5Mbit/s (MPEG)*.

3 Definitions

3.1

PREMO Part 1 definitions

this part of ISO/IEC 14478 makes use of all terms defined in ISO/IEC 14478-1 (Fundamentals of PREMO)

3.2

PREMO Part 2 definitions

this part of ISO/IEC 14478 makes use of all terms defined in ISO/IEC 14478-2 (PREMO Foundation Component)

3.3 Additional definitions

For the purposes of this part of ISO/IEC 14478, the following definitions apply.

3.2.1

configuration objects

collective name for format, quality of service descriptor, and media stream protocol objects

3.2.2

jitter

delay variance

3.2.3

processing element (for virtual devices)

conceptual entity describing the internal behaviour of virtual device objects

3.2.4

virtual connection adapter

conceptual entity describing the configuration process performed by a virtual connection object

3.2.5

unicast connection

one-to-one connection; an output port may be connected to one input port only, and an input port may be connected to one output port only

3.2.6

multicast connection

one-to-many connection; an output port may be connected to several input ports, and an input port may be connected to several output ports

the following alphabetical list gives the subclause of each definition

configuration objects	3.2.1
jitter	3.2.2
multicast connection	3.2.6
processing element (for virtual devices)	3.2.3
unicast connection	3.2.5
virtual connection adapter	3.2.4

4 Symbols and abbreviations

ATM:	Asynchronous Transfer Mode.
CATV:	Cable TV
DMA:	Direct Memory Access.
FSM:	Finite State Machine.
IEC:	International Electrotechnical Commission.
IS:	International Standard.
ISO:	International Organization for Standardization.
JPEG:	Joint Picture Experts Group.
LAN:	Local Area Network
MIDI:	Musical Instrument Digital Interface
MPEG:	Moving Picture Experts Group.
MSS:	Multimedia Systems Services
PREMO:	Presentation Environment for Multimedia Objects.

QoS:	Quality of Service.
TCP:	Transmission Control Protocol.
UDP:	User Data Protocol.
RTP:	Real-Time Protocol

5 Conformance

A conforming implementation of the PREMO Multimedia Systems Services shall comply with the general conformance rules defined in clause 5 of ISO/IEC 14478-1 and the component specification in clause 11 of This part of ISO/IEC 14478.

6 Overview of the Multimedia Systems Services

6.1 Introduction

This clause presents several comprehensive views of the Multimedia Systems Services, which, taken together, represent a broad, architectural summary. These views include:

- a) an object interaction diagram, which characterizes the dynamic relationships among instantiated objects and to illustrate client visible interfaces;
- b) a subtyping diagram, which describes the subtyping hierarchy among MSS objects;
- c) a short description of the life cycle of Multimedia Systems Services objects.

A somewhat larger example of how the various MSS objects may be used is also given in Annex B.

6.2 Object framework

Figure 2 on page 6 summarizes the interactions between Multimedia Systems Services framework objects and a client; Figure 3 on page 7 summarizes the interaction among framework objects. As seen in Figure 2, only a subset of the objects and interfaces are actually visible to a client of MSS. In particular, much of the interaction between the virtual connection and other objects in the framework is not client visible. This part of ISO/IEC 14478 is concerned primarily with client visible interfaces implementations may extend these interfaces for implementation-specific reasons, to comply with the behaviour of the objects, as defined in this standard.

Figure 2 is suggestive, rather than realistic, since the objects shown are abstract types, rather than concrete types which would normally be instantiated. Also, object creation and destruction are not shown in this diagram.

In Figure 2, the client is communicating with a small dataflow graph, comprised of two virtual devices and a virtual connection. A group object, which assists the client, is also shown. The client interacts with the objects indicated by the arrows. Each of these objects may be local or remote.

Each virtual device is a processing node in the dataflow graph. The nature of the processing (capture, encoding, filtering, etc.) varies according to the specific object (and is implemented by subtyping). Each virtual device contains a stream control object and one or more format objects shown by the boxes in the shaded areas. Virtual connections and groups also have an associated stream object and this association is represented similarly. These associations are referred to as inclusion. Although not explicitly shown in the diagram, the client interacts directly with the included stream and format interfaces.

A stream control object provides the client with an interface to observe media stream position in various terms (as a function of media transport, media samples, or logical time; see clause 8). Stream control objects also provide synchronization operations. Stream control objects do *not* perform the effective transfer of media data; they merely act as an entry for the clients to control media flow.

In addition to a stream control, a virtual device also contains one or more ports, describing an input or output mechanism for the virtual device. Ports are framework abstractions that do not have a client-visible interface. Virtual devices provide operations to select a specific port, using an abstract non-object data type as an opaque handle.

Just as the stream control object allows a media stream control abstraction which is separable from media processing, the format object provides an abstraction of the details of media formatting which is separate from both processing and flow-control. For example, the details of a frame-dependent video encoding, like MPEG, would be represented by a subtype of format.

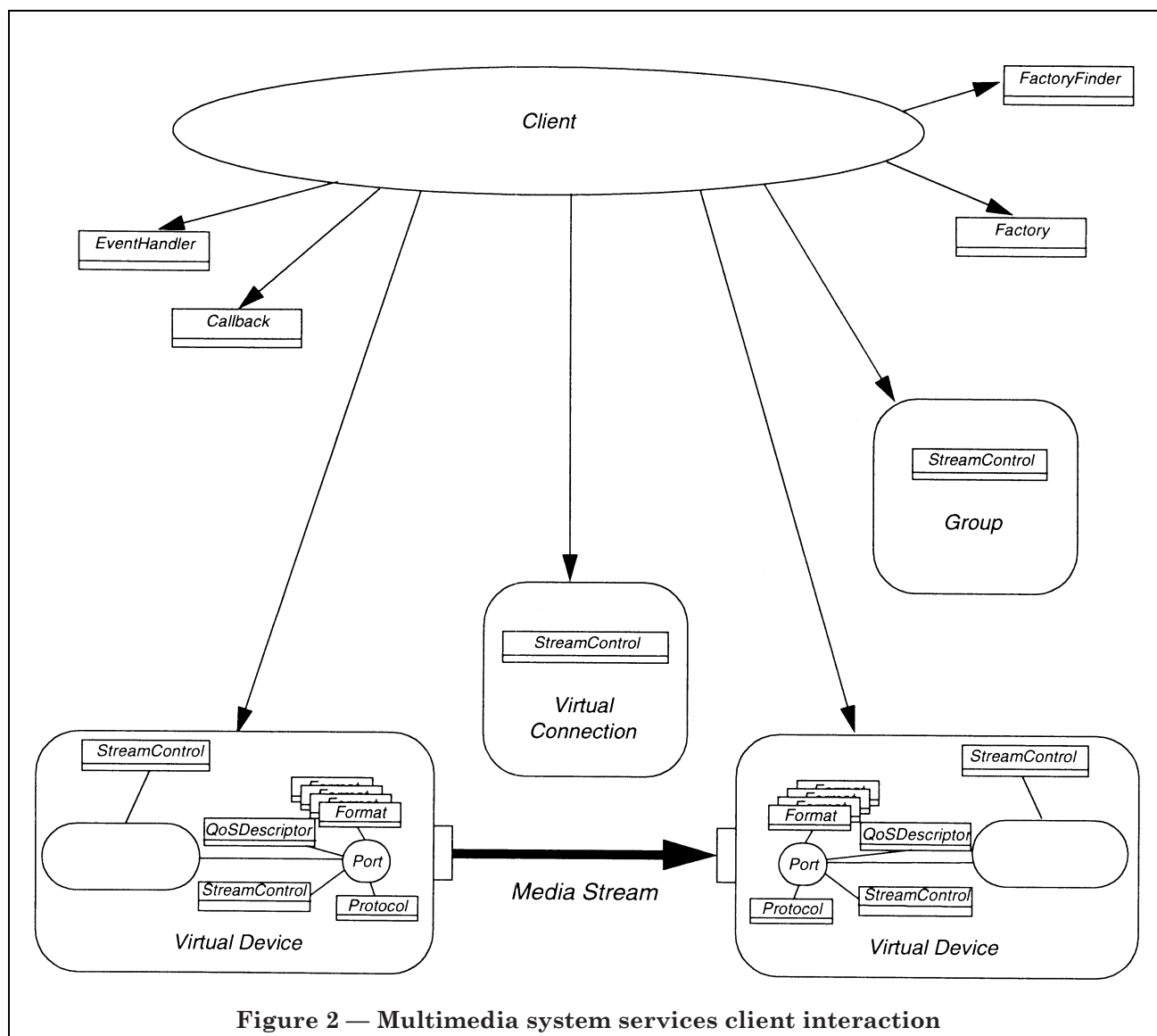


Figure 2 — Multimedia system services client interaction

The virtual connection provides operations to create a connection between an output port of one virtual device and an input port of another, fully encapsulating low-level transport semantics. Virtual connections also provide support for multicast connections. An included stream object provides operations for controlling the dataflow on the virtual connection.

The group object, shown in Figure 2, provides assistance to the client to manage the dataflow graph of the two virtual devices and the virtual connection. A group object provides a convenient mechanism for atomic resource allocation and specification of end-to-end Quality of Service (QoS) values for the whole graph. The group object gives access to a stream object, through which the client can control dataflow for the encapsulated graph.

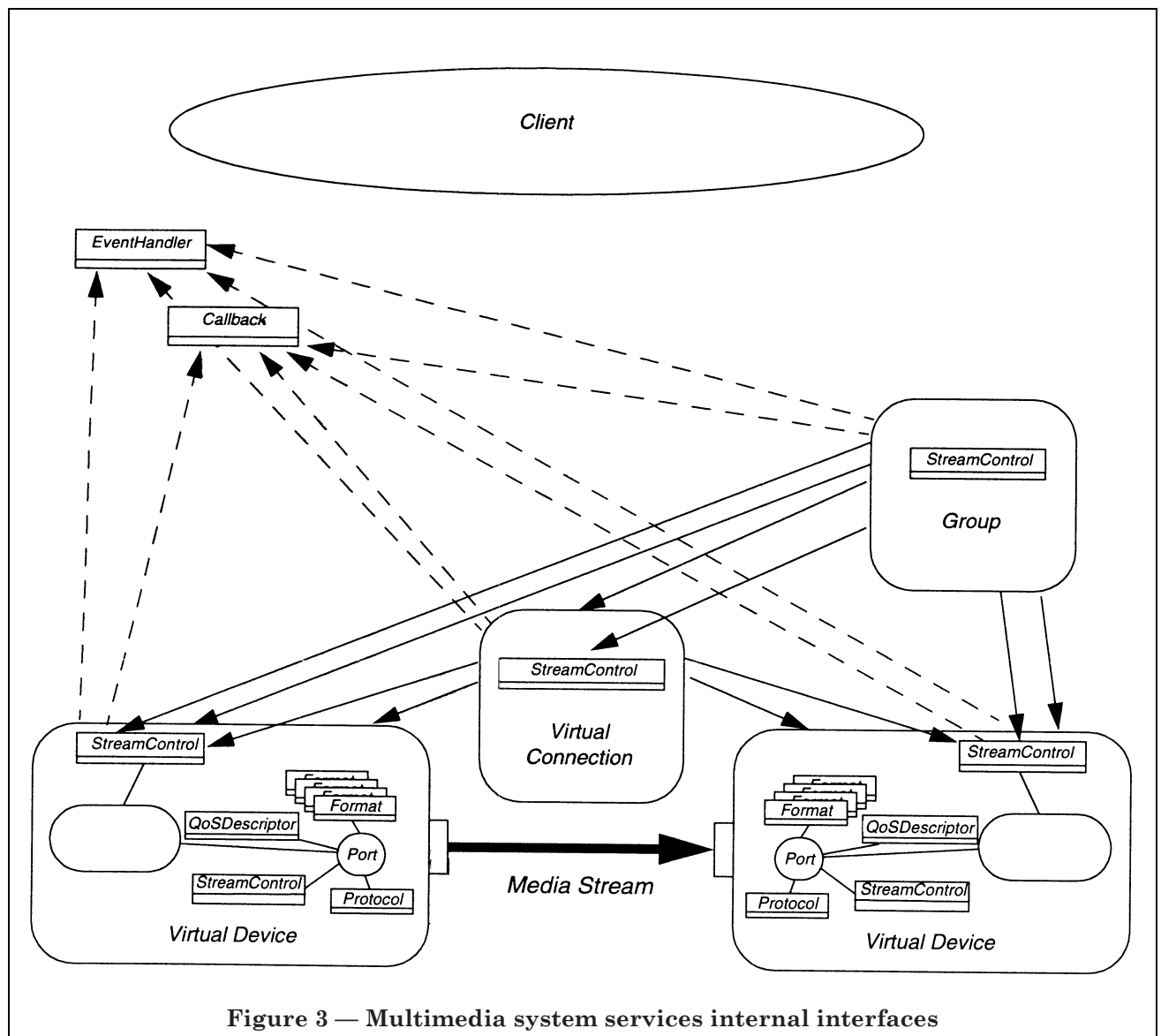
Multimedia Systems Services objects are instantiated by factories. A factory provides the client facilities to select among the various objects that the factory is capable of creating. A client can also use the factory finder service to find a reference to a factory capable of instantiating an object whose properties satisfy a list of constraints.

A client can register interest in receiving specific events produced by the various objects. This is done using the callback mechanism and the event model as described in ISO/IEC 14478-2 (part 2 of PREMO).

Figure 3 shows the internal connections between MSS objects. For the most part, the client is unaware that these connections exist, and this part of ISO/IEC 14478 will not focus on the low-level details of these connections and their relevant interfaces. They are shown here to help explain the Multimedia Systems Services architecture. The primary purpose of most of the internal connections is to off-load work from the client. Note, for example, that the virtual connection interacts with the formats of both the source and target virtual devices. This allows the virtual connection to match those formats without client intervention. The group and the stream associated with the group provide similar assistance to the client; the group can assist in resource allocation, while the associated stream can assist in stream control. The dashed arrows show that the objects send events to the client via the event handler.

6.3 Subtyping diagram

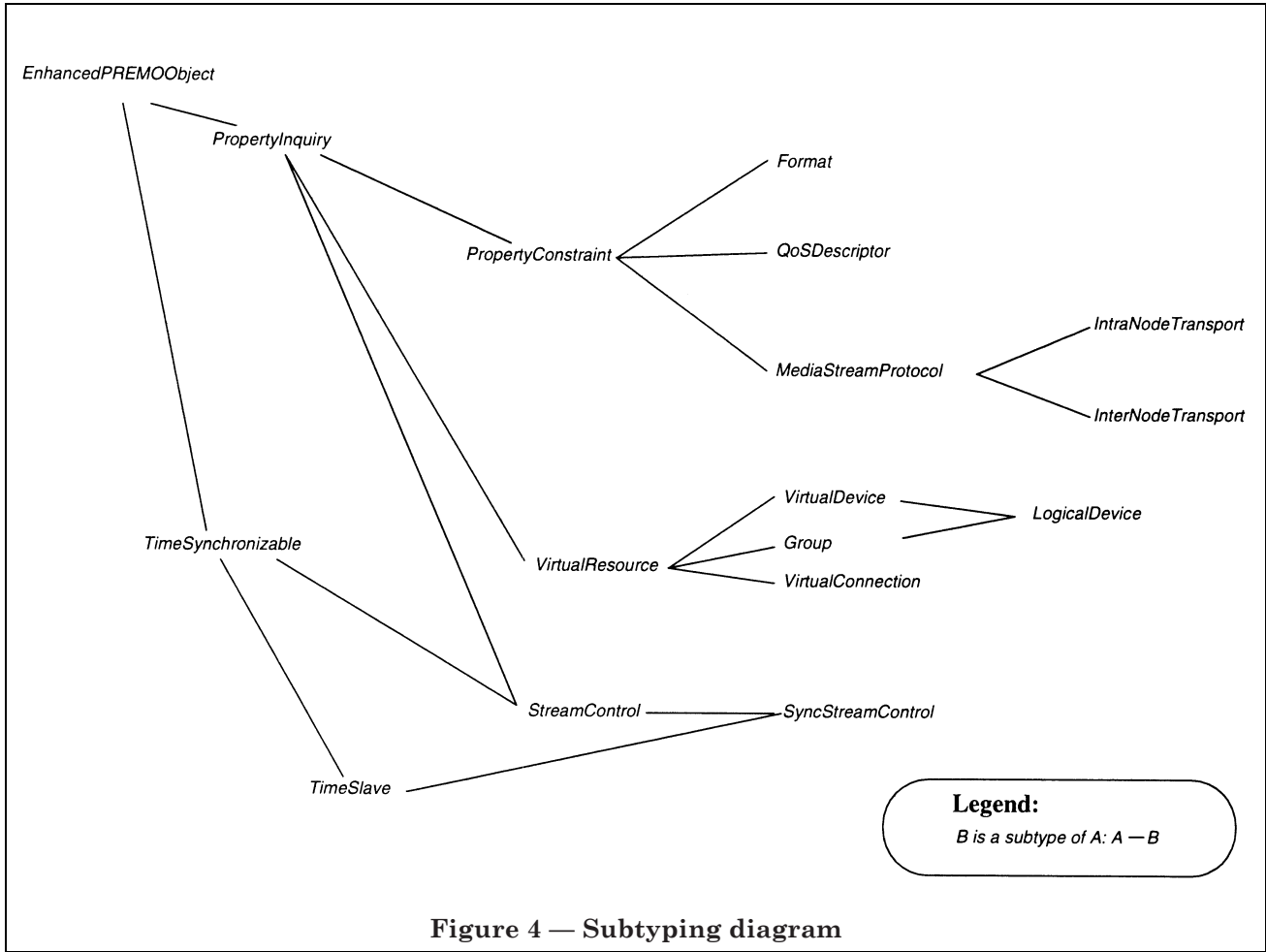
Another view of the MSS Architecture is given in Figure 4 on page 8. This is a simplified subtyping diagram for the MSS objects which does not contain all object types defined in this part of ISO/IEC 14478 neither their full type graph; the reader should refer to Annex A for a more complete diagram.



The types in the diagram will be discussed in detail elsewhere in this part of ISO/IEC 14478. For the time being the only important characteristic to note is that all types defined in MSS are subtypes of *EnhancedPREMOObject* (see ISO/IEC 14478-2), thereby inheriting all features described in the Foundation Component of PREMO. Also, all objects are subtypes of *PropertyInquiry*, which is further described in 8.1.2 of ISO/IEC 14478-2 (part 2 of PREMO).

6.4 MSS object life cycle

Figure 5 on page 9 gives a schematic view of the life cycle of an MSS object and its reference. Beyond the basic life cycle of PREMO objects, as described in ISO/IEC 14478-1, all MSS objects rely on the concepts of factories and factory finders defined in the extended profile of ISO/IEC 14478-2 (see 8.2.1 of ISO/IEC 14478-2 for a detailed description of these facilities), which encapsulate normal object and object reference creation. The client has the ability to create objects through these factories by specifying constraints on the capabilities of the object to be created (e.g., media formats it can process, quality of service it can provide, etc.). The client receives from the factory an object reference to an object obeying these requirements. This object reference goes then through the life cycle as described in 8.11 of ISO/IEC 14478-1.



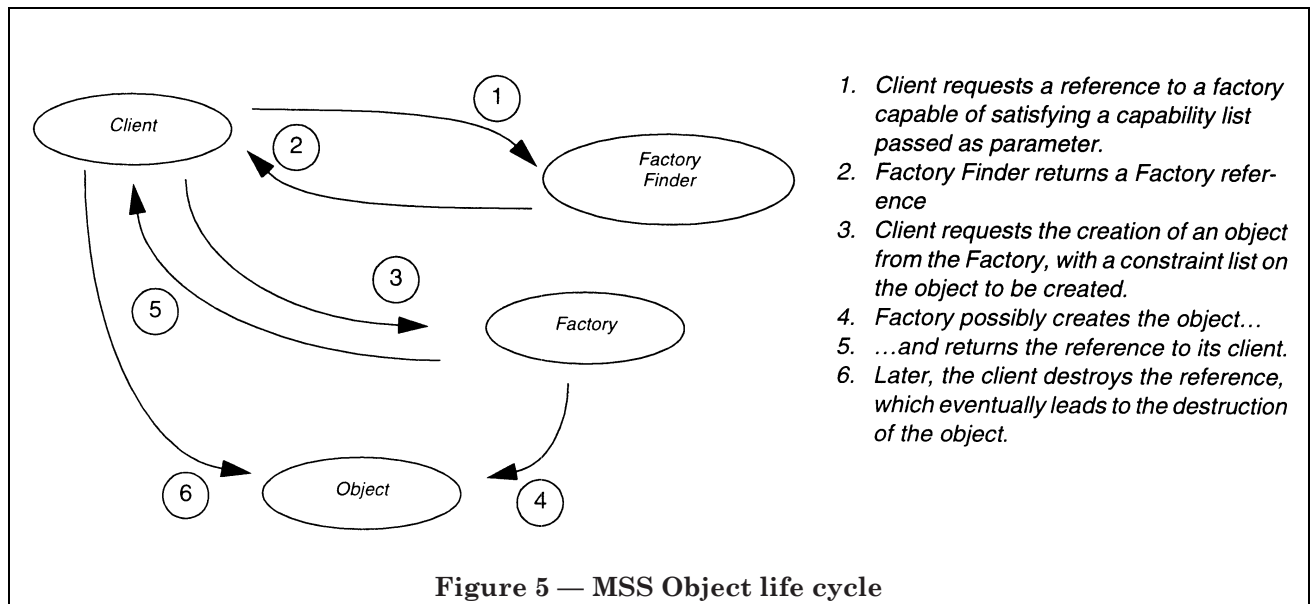


Figure 5 — MSS Object life cycle

7 Configuration objects

7.1 Introduction

The various MSS object types, categorized as *configuration* objects, are used as information depositories for other objects. Configuration objects do not provide complicated behaviour to their clients; instead, their role is to act as placeholders for the necessary parameters which allow other objects to function properly. Configuration objects are not used in isolation; they are always associated with an object of type *VirtualResource* object (see 9.1) whose behaviour they affect.

All configuration objects are subtypes of *PropertyConstraint* objects (see 8.1.3 of ISO/IEC 14478-2), and the various parameters they provide to other objects are stored and manipulated through the property constraining mechanism which characterize the *PropertyConstraint* objects. Typically, configurable MSS objects (e.g., resource objects, see clause 9) contain several instances of configuration objects, whose references can be accessed by the external clients; through the *constrain*, *select*, etc., operations these clients can then set the required values for the configuration objects.

Usage of configuration objects is the basic mechanism for configurability in PREMO. Clients may inquire the key-value pairs associated with these objects and, through the property constrain and selection mechanism (see 8.1.3 of ISO/IEC 14478-2) may restrict these values. The algorithms for constraining the possible values (within the limits defined by the capabilities and the native property values of the configuration objects) depend on the client and is not standardized by PREMO. These algorithms may take into consideration the full set of configuration objects associated to the same *VirtualResource* instance. Note that further subtypes of configuration objects may be defined with an inherent constraint algorithm, which is made available through the *select* operation of the configuration object (see again 8.1.3 of ISO/IEC 14478-2).

The MSS has three categories of configuration objects, described below:

- a) Format objects;
- b) Transport and Media Stream Protocol objects;
- c) Quality of Service objects.

7.2 Format objects

The role of the *Format* object type is to represent the details of the media format (that is, the organization of the bitstream) at a particular port of a device (see clause 9). The characteristics of a specific media format are described in form of object properties defined on the corresponding *Format* object.

NOTE For example, the organization of data in an MPEG data stream is different from the organization of data in an a-law audio stream. The *Format* object allows the client to specify as little or as much about a particular encoding as it wants. When a connection is made to a port, the virtual connection interacts with the *Format* object to negotiate the details of the bitstream to be passed between virtual device ports.

The Multimedia Systems Services do not define specific format objects; instead, it defines a general architecture which makes such a specification possible. However, the (informative) Annex C of this part does describe a hierarchy of format and device objects which provide a good example of how this general architecture should be used.

7.3 Transport and Media Stream Protocol objects

When a virtual connection determines that two virtual devices cannot be directly connected (often because they are on different machines), it creates a virtual connection adapter to transport media data between them (see 9.3). The virtual devices may reside in different implementations of the Multimedia Systems Services, so the virtual connection adapter must share a common protocol in order to interoperate.

The purpose of the Transport and Media Stream Protocol (MSP) objects is to provide information on how media data is conveyed among processing nodes. It is not the role of MSS to give a detailed specification of the various possible communication protocols, only references to existing protocol specifications are made here. In MSS, MSP objects have two subtypes:

- a) *IntraNodeTransport*, which refers to communication among nodes taking place through shared memory (e.g., two nodes processing in the same address space of a workstation using, e.g., DMA)
- b) *InterNodeTransport*, which refers to communication among nodes taking place over a network, or through some interprocess communication means. The various protocol names which can characterize these protocols include IPC, UDP, RTP, ATM, or NETBIOS.

The virtual connection determines the format of the media data, including its bit-level representation, by negotiation with the relevant *Format* objects (see 7.2) and perhaps with client intervention. Similarly, the client will conduct negotiations on the detail of the transfer protocols, using the MSP objects. From the point of view of MSP objects the media data is an opaque entity. Neither the virtual connection adapter nor the underlying network transports know how to extract information from the media data.

To allow for future revisions, the MSP properties include a protocol version number.

7.4 Quality of Service Descriptor objects

In order for a virtual resource to be useful, it must obtain the physical resources required for it to do its job. Resources include both system resources that are typically not multimedia specific, such as those provided by the CPU, memory, and network sub-systems, as well as specialized multimedia resources such as audio and video devices.

Because the quality of service that can be provided by many resources varies considerably, the client must also specify the desired quality of service when requesting a resource; this is done by setting the properties of a special information depository object in MSS, called the *QoSDescriptor* object.

Though quality of service can take on many meanings, many of them media and device specific, the Multimedia Systems Services defines a core set of *QoSDescriptor* properties that can be used by a client to specify the quality of service of interest. The core *QoSDescriptor* characteristics defined by the Multimedia Systems Services are:

- a) guaranteed level (key “*GuaranteedLevelK*”): provides options for “*Guaranteed*” service, “*BestEffort*” service or “*NoGuarantee*” service;
- b) reliable (key “*ReliableK*”): the delivery of data is reliable or not;
- c) delay bounds (key “*DelayBoundsK*”): the minimum and maximum delay;
- d) jitter bounds (key “*JitterBoundsK*”): the minimum and maximum jitter (delay variance);
- e) bandwidth bounds (key “*BandwidthBoundsK*”): the minimum and maximum bandwidth.

The client specifies the desired *QoSDescriptor* properties by accessing the quality of service information from a *VirtualResource* object, and manipulating the property values with the property management operations.

The guaranteed level property is an indication of the performance required of the virtual connection (see 9.3 for a more detailed presentation of virtual connections). A “*Guaranteed*” connection will reserve resources appropriate to handle worst-case needs for the media transfer in order to make sure that the data always arrives and is on time. A “*BestEffort*” connection will provide the best possible performance while using optimistic amounts of resources. This may produce situations where the data occasionally arrives late. “*NoGuarantee*” uses the minimum amount of resources for the connection and do as well as possible. The reliability property indicates whether the application is willing to suffer data loss.

Delay is the amount of time between the transmission of the data and the receipt of the data. Different applications will have different requirements. For instance, an audio conference would be unwilling to live with a 2 second delay, whereas a non-inter-active video playback application might find it acceptable.

Bandwidth is the amount of data per unit time that the connection will be required to support or, in the case of an input port, to expect. For example, a video conference might require 384 Kbits/second while an MPEG stream might require 1.5 Mbits/second. By defining the range of the bandwidth required, the connection will understand the maximum burst that it must handle.

Jitter is the amount of delay variance. For example, an ISDN channel that presents a “lot” of data every 125 microseconds has a jitter of 0, since there is no variance in the arrival time of the data. If an application requests a jitter close to 0, then the connection will try to find an isochronous network connection between the two virtual devices.

Subtypes of *QoSDescriptor* objects may extend the *QoSDescriptor* management with additional characteristics.

8 Stream Controls

The *Stream Control* type, and its subtypes, provide a single point of focus for all inquiry and control of media stream progress in a media type independent way. *StreamControl* objects are never created in isolation; they are included objects of *VirtualResource* objects whose role is to monitor and control stream progress for the overall resource. Various virtual resource objects need sub-types of the stream objects defined in MSS to cope with the various media-dependent characteristics, e.g., details of media presentation. The MSS gives, a formal specification of two objects only, namely the *StreamControl* and the *SyncStreamControl* object types; semantic details of the various subtypes are beyond the scope of MSS and are not considered in This part of ISO/IEC 14478.

NOTE Further PREMO components may define subtypes of the *StreamControl* and *SyncStreamControl* objects.

The *StreamControl* type provides operations to observe and control media position. A complication is that different objects require different concepts of position:

- a) *Transport aware object*: the object might understand transport packets, but not understand the structure of the media stream. The object can only report the stream address, that is the byte count since the stream began to flow.
- b) *Media Stream aware object*: the object might understand media samples, but not how media samples translate into stream time. The object can report the stream sample count.
- c) *Time aware object*: the object understands how to extract stream time from the media stream. The object can report the stream time.

To answer to these concerns, the *StreamControl* object is defined in MSS to be a subtype *TimeSynchronizable*, described in detail in 7.9 of ISO/IEC 14478-2. Inheriting the behaviour of these objects, clients of *StreamControl* objects have the ability to access and control the stream position in terms of:

- d) *Media stream data*: The internal progression space of the object (i.e., index of a sample).
- e) *Relative time*: The time elapsed from an origin, settable by the client. One can achieve effects like “pause at 100ms from origin”.

Synchronization elements can be put at various points of the stream, using either media stream data or relative time. This means that a *StreamControl* object will send events when these points are crossed. Such synchronization elements can be set periodically, i.e., one can achieve effects like “send an event at each 100 ms”.

Stream control objects do *not* perform the transfer of media data; they merely act as an entry for the clients to control media flow. Conceptually, media data transfer is done by the processing element of the *VirtualDevice* object and, hence, is opaque to the client.

The *StreamControl* object does not specify further the internal progression space of *TimeSynchronizable* (which is a generic type), i.e., whether the internal progression is done in the integer, real, or time domain. Specific *VirtualDevice* objects have to actualize the *StreamControl* objects they use by making this decision.

8.1 StreamControl objects

The *StreamControl* object adds a finer media control to its supertype *TimeSynchronizable*. This finer control is achieved through the refinement of the finite state machine governing the behaviour of *TimeSynchronizable*.

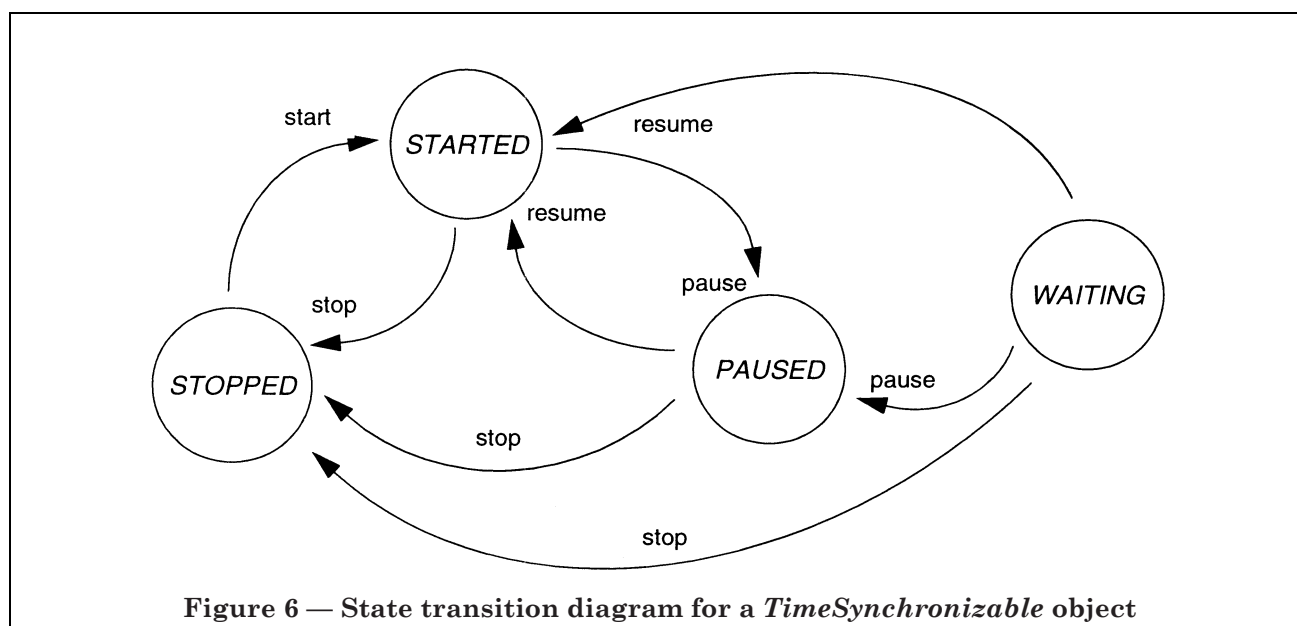


Figure 6 depicts the finite state machine of *TimeSynchronizable*, as defined in ISO/IEC 14478-2 (not all state transition operations are present on the figure, only the most important ones); for a detailed discussion on the behaviour of *TimeSynchronizable*, see 7.9 (page 12) of ISO/IEC 14478-2. Figure 6 gives a rough overview of the finite state machine of a *StreamControl* object, again omitting some details (the detailed state transition table is in 10.6.1).

The *StreamControl* object adds three new states to the state machine of *TimeSynchronizable*, namely *MUTED*, *PRIMING*, and *DRAINING*. Three new operations are also defined, which control the state transitions to and from these new states: *mute*, *prime*, and *drain*.

MUTED and *PRIMING* are refinements of the *STARTED* state of *TimeSynchronizable*. The additional semantics in these states is related to the notion of data presentation. In describing the *STARTED* state, ISO/IEC 14478-2 refers to data presentation in very general terms only, as one abstract processing step of the object at that level of abstraction. This abstract step is performed through the invocation of a protected operation, called *processData*. The specification leaves the semantic details of what presentation means to the various subtypes of *Synchronizable*. Although the *StreamControl* object does not specify what presentation means either (and leaves the details to the subtypes of *StreamControl*), the specification of *MUTED* and *PRIMING* gives a somewhat finer control on the behaviour of the *StreamControl* object with respect to presentation. This refinement is as follows:

- *MUTED*: no presentation occurs while the object is in this state, and the media data are disregarded. In other words, progression on the stream occurs (i.e., all synchronization actions are performed) but the *processData* operation, which represents the abstract notion of processing media data, is not invoked. The operation *mute* is analogous to the operation *start*, inherited from *TimeSynchronizable*, the only difference being the target state of the transition (*MUTED* instead of *STARTED*).

— *PRIMING*: no presentation occurs while the object is in this state, and the media data are buffered in an internal buffer. In other words, progression on the stream occurs (i.e., all synchronization actions are performed) and the media data are stored internally instead of being presented, i.e., instead of calling the *processData* operation. The operation *prime* is analogous to the operation *start*, inherited from *TimeSynchronizable*, the only difference being the target state of the transition (*PRIMING* instead of *STARTED*). If the internal buffer of the object is full, i.e., no stream data can be stored any more, the object makes an internal state transition to *PAUSED*.

The third additional state, *DRAINING*, is the counterpart of *PRIMING* in buffer control. When set to this state, the object empties the buffer filled up by a previous *PRIMING* state; when the buffer is empty, the object makes an internal state transition to *PAUSED*. While emptying the buffer, presentation of data also occurs. The operation *drain* is defined to set the *StreamControl* object into *DRAINING* state.

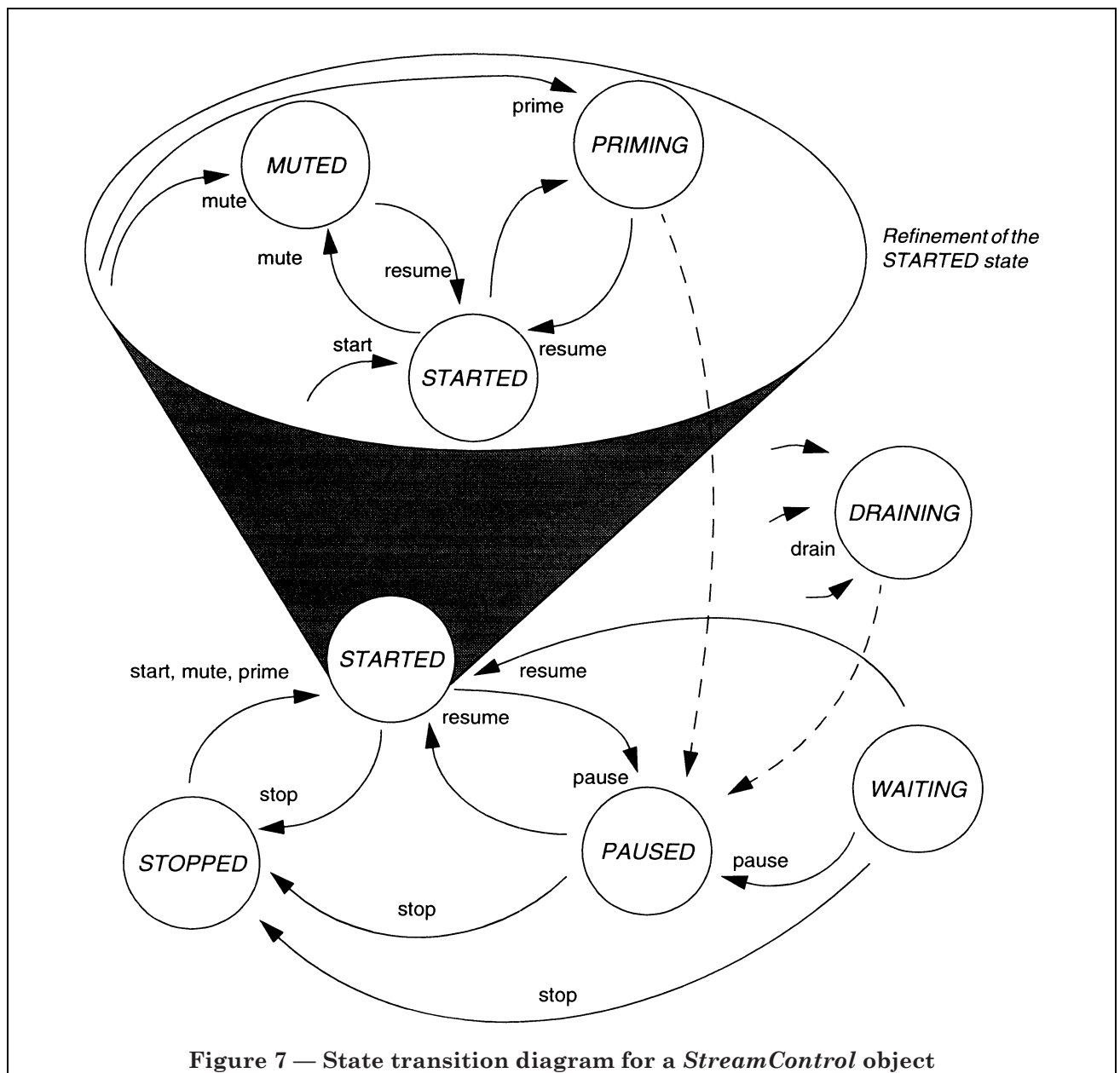


Figure 7 — State transition diagram for a *StreamControl* object

A transition to *STARTED* state from both *PRIMING* and *DRAINING* states is possible; conceptually, the internal buffers are to be instantaneously emptied before the normal media flow is resumed.

Note that, as a subtype of *Synchronizable*, the *StreamControl* object also inherits the possibility to have state transitions monitored by other objects (using the callback mechanism, see also 7.9.1.2 of ISO/IEC 14478-2). In particular, clients can be notified if internal buffers become full or empty while priming, respectively draining (in both cases an internal state transition to the state *PAUSED* takes place, which may be monitored).

Subtypes of *StreamControl* may add additional semantics to buffer control. As a typical case, if the streams are aware of their position within a dataflow network, some of the operations, like *prime* or *drain*, may also generate private control flow among the streams in this network. For example, *prime* on a *StreamControl* may also generate a control information to the *StreamControl* object “up-stream” i.e., the stream providing the data. Whether such additional protocol is defined or not depends on the sub-types of the *StreamControl* object.

StreamControl objects are also subtype of *PropertyInquiry*. This means that subtypes of *StreamControl* objects may also take part in complex negotiations based on properties, as described in 8.1 of ISO/IEC 14478-2 (part 2 of PREMO).

8.2 SyncStreamControl objects

The *SyncStreamControl* type is designed to permit the synchronization of multiple media streams. The client specifies a second *StreamControl* object to provide a master position reference to the *SyncStreamControl*. This functionality is achieved by inheriting the behaviour of the *TimeSlave* objects, defined in 7.9.3 of ISO/IEC 14478-2. *SyncStreamControl* is defined as a (multiple) subtype of both the *StreamControl* and the *TimeSlave* object types, thereby refining the finite state machine of a *TimeSlave* object the same way as *StreamControl* objects refine the behaviour of *TimeSynchronizable* objects.

NOTE The *SyncStreamControl* type provides the client with a variety of options with respect to setting up how synchronization is to be achieved. For example, a video display device can be synchronized with an audio display device by setting up the appropriate relationship between their associated *StreamControl* objects. Assuming the video device supports a *SyncStreamControl* type, the audio device's *StreamControl* object can be made the master of the video device's *SyncStreamControl* object.

In another situation, the client may want to synchronize two displays to a common time reference. In this case, both displays would have to support the *SyncStreamControl* type. The client would make the *StreamControl* object associated with the time reference the master of the *StreamControl* objects associated with both the display objects.

9 Devices, Resources

9.1 Virtual Resources

A *virtual resource* is an abstraction of a physical resource that provides the client developer a consistent programming model, independent of the details of specific implementations. The notion of a virtual resource makes applications more portable across a variety of systems, while at the same time making the transparent sharing of physical resources possible.

The Multimedia Systems Services defines four basic subtypes of virtual resources:

- a) *virtual devices*, which abstract media processors;
- b) *virtual connections*, which abstract connections between virtual devices;
- c) *groups*, which provide a convenient way to interact with a collection of virtual devices and connections;
- d) *logical devices*, which provide a possibility to build a hierarchy of virtual devices.

The MSS defines a *VirtualResource* object type, which serves as a common, abstract supertype for these three categories. This clause describes this object type in more details.