Multimedia Resources: An Information Model and its Application to an MPEG2 Video Codec

Constant Gbaguidi[1], Simon Znaty[2], and Jean-Pierre Hubaux[3]

Still today, diagnosing a problem with multimedia resources, such as video and sound cards, is insufficiently automated. These resources therefore cannot be accurately managed. One reason for this is the lack of their thorough modeling. In this paper, we fulfill this need, by proposing a generic information model, which we further apply to an MPEG2 video codec. We highlight the main characteristics of this kind of codec, identify parameters that influence these characteristics, and reveal some of the trade-offs that the application developer can consider in order to design efficient software for MPEG2 codecs. In addition to the benefits of this modeling for the user and the application developer, we also show how useful it could be for the providers of distribution services, such as "live" video transmission. These providers can use our model to achieve resource management on an end-to-end basis.

KEY WORDS: Multimedia resources, information model, MPEG 2, video codec.

1. INTRODUCTION

Currently, there is no easy way to obtain management information from devices such as video or sound cards. Users often return equipment to the vendors, for very simple repairs that they should be able to do themselves. In turn, the vendors exchange the misbehaving equipment for a new - as they, too, are unable to identify the cause of the problem. Having a management information model that is implemented with the aforementioned devices, will bring only benefits to both users and equipment vendors. This information model can also be used - as a

1. DI-ICA, Swiss Federal Institute of Technology (EPFL), 1015 Lausanne, Switzerland. Email: constant.gbaguidi@epfl.ch.
2. Multimedia Networks and Services Dept., ENST-Bretagne, 35512 Cesson Sévigné, France. E-mail: simon.znaty@enst-bretagne.fr.
3. DI-ICA, Swiss Federal Institute of Technology (EPFL), 1015 Lausanne, Switzerland. Email: jean-pierre.hubaux@epfl.ch.

resource management tool - by the service providers to manage incompatibility situations wherein the users in a communication session employ devices with totally different features. A user, with a Motion Joint Photographic Experts Group (MJPEG) card, may wish to interact, over videoconferencing, with someone using an H.261 board. The service provider must then furnish a transcoder that can be employed by a large body of users at a time; this transcoder is part of the Quality of Service (QoS) provisioning system, as well as the network. The management of the entirety of this system requires managing the transcoder, which is an example of what we call a multimedia resource because it needs to be aware of the nature of the medium (audio, video, etc.) that it handles. The users' video boards should also be managed in order to achieve complete end-to-end resource management.

Until now, very few works have been devoted to multimedia resources. They are part of what the Telecommunications Information Networking Architecture Consortium (TINA-C) calls, without much characterization, information converters [1,2,]. Pavlou and Griffin [4] consider these resources as service resources and contribute a generic resource configuration management scheme. The information model that we develop in this paper can be used with this resource configuration scheme.

A relevant review of multimedia resources - under the broader heading of media handling resources, which include multipoint control units as well - is given in [5], again, without much study of their features.

We address this study, by proposing a generic information model of multimedia resources (Section 2). We then specialize the proposed model for a specific resource, namely an MPEG2 (Moving Pictures Experts Group) [6] video codec (Section 3). We chose this codec because of its richness in features and, therefore, in information. Concluding remarks follow in Section 4.

2. AN INFORMATION MODEL OF MULTIMEDIA RESOURCES

First, we present the general principles behind information modeling (section 2.1). Then, we describe multimedia resources and point out their main differences from the classical network resources (section 2.2). The modeling itself is presented in section 2.3.

2.1. Principles for Information Modeling

Sclavos *et al.* [7] identified certain principles that must be followed during any information modeling process. The main principles are:

- *Information structuring.* The relevant information in the system should be highlighted and structured in a consistent way. For instance, we may model a computer by saying that it contains some hard-drives, files, Central Processing Unit (CPU), memory, Input/Output (I/O) controllers, and so forth. This listing of components does not tell much about the actual relationships among them. Management is easier when the relationships among the management information are clearly determined and described. A more accurate modeling of a computer could look like this: a computer is made up of a motherboard, one or more I/O controllers, one or more hard-drives, etc. A motherboard hosts the main memory and, possibly, I/O controllers. Hard-drives contain file-systems that hold files. The hard-drives are controlled by, and accessed through, I/O controllers. And so forth. This improved structuring shows the relationships among the components of a computer, including the hierarchy among them.

- *Abstractness*. The system being modeled must be conceptualized, i.e., described possibly using concepts that help grasp the characteristics and activities of the system. This conceptualization highlights the information that needs to be managed.
- *Genericity*. The common features among entities of the same type should be identified. From a practical point of view, genericity helps devise a common management scheme for entities that share some common features. This scheme may then be implemented differently for each individual entity.
- *Independence from the implementation*. The designer should not think of a particular implementation solution when modeling a system. A thorough understanding of the system is the key issue in modeling.
- *Modeling purpose*. A complete information model should exhibit the information necessary to perform the tasks within the five management functional areas identified by the International Organization for Standardization (ISO): fault, configuration, accounting, performance, and security management [8]. Very often, the designer concentrates on a subset of these areas when modeling a system. Then, she must mention which areas are of interest to her.

We chose an object-oriented technique - specifically the Object Modeling Technique (OMT) [9] - to model the management information. There are other ways (not object-oriented) of modeling this information, but these means generally fail to express fairly complex characteristics such as the relationships among information components. An example is the Internet Engineering Task Force's (IETF) structure of management information [10] derived from the Abstract Syntax Notation One (ASN.1) [11].

2.2. Characterization of Multimedia Resources

Multimedia resources can be realized in software or hardware or in both. Rodriguez *et al.* [12] uphold that software-only video codecs, yet less complex than hardware-assisted codecs, currently deliver a lesser quality. Even though many efforts are being deployed to offer complete software solutions, it is too early to bet on their real performance. Personal Computers (PC) now come with multimedia-enabled extensions, promising much suitability to the needs of multimedia applications.

A multimedia resource exhibits *modes* and deploys *processing layers*. We define modes as the external view of the resource's activities, while processing layers refer to the internal activities (processing steps) leading to the provision of the overall resource's functionality.

Depending on the granularity intended for the model, multimedia resources can aggregate others (Fig. 1); indeed, each processing layer may in turn be considered as a multimedia resource. Thus, the *recursiveness* property, well known in the network management community, can be applied here as well; it denotes that a multimedia resource may contain many others. For instance, the buffer used in video codecs may be considered as a multimedia resource, if its management needs to be emphasized. In addition, multimedia resources aggregate ports and paths (Fig. 1).

An important difference between network elements and multimedia resources is that the latter resources usually need to know the real nature of the stream that they process. For instance, MPEG codecs do not process audio and video information in the same way. On the other hand, the network elements do not have to know whether the information relates to audio or video (or whatever). Moreover, multimedia resources can highly affect the content of the incoming

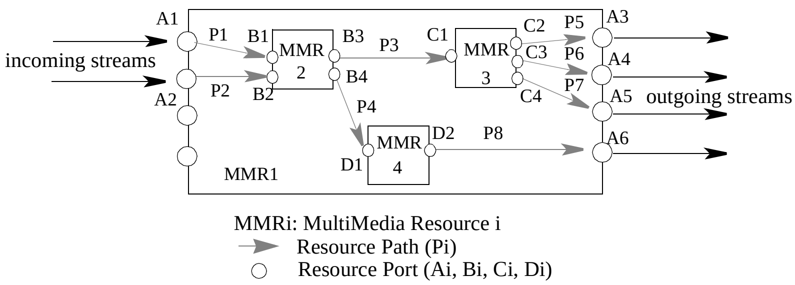


Fig. 1. Architecture of a Multimedia Resource.

information, and the transformation that they perform on this information is generally irreversible; it is impossible to recover the original information from an MPEG-encoded stream. Network elements may generate such non-linearities *only* when losses occur. To take into account these two observations (i.e., the importance of the characteristics of the incoming flows, and the non-linearities between the input and the output flows), any modeling of multimedia resources should clearly differentiate the characteristics of the incoming stream from those of the flow after treatment. We did this in our model, which is presented next.

2.3. A Generic Information Model

First, we explain why we chose to not re-use the existing information models of network elements. Then, we present our model, by describing each of its elements in turn.

There are many abstract information models of the network, the most famous being M.3100 [13] and G.805 [14]. M.3010 - a recommendation from the International Telecommunications Union (ITU) - provides a generic model of *network elements*. G.805 provides a generic functional architecture for the connectivity services offered by *transport networks*.

These models apply to the key elements, in the network, whose design is more complex than that of multimedia resources. Therefore, we think it reasonable to present, as a first step, a much simpler model than those mentioned above. We follow the simplicity philosophy exemplified by the network modeling works in [7,15,16]. These works, however, did not consider the specific features of multimedia resources as described in section 2.2.

The model proposed (Fig. 2) results from the discussion in the previous section. It focuses on the information related to configuration and performance management. Modes appear in the shape of streams (*InputStream* and *OutputStream*), while processing layers are essentially represented by lower level multimedia resources, called *(N-1)-MultimediaResource*, assuming that the level of the containing resource is *(N)*. Processing layers also appear in the shape of the *Function* objects aggregated by the *(N)-Service* object. Each entity in Fig. 2 is described below.

2.3.1. The Architecture_Element Class

As stated in section 2.2, multimedia resources may be designed in software and/or hardware. Therefore, their architecture may consist of a software part and/ or a hardware part. The latter is an equipment characterized by its description, its physical location, the contact information in case of trouble, and the name under which it is managed.

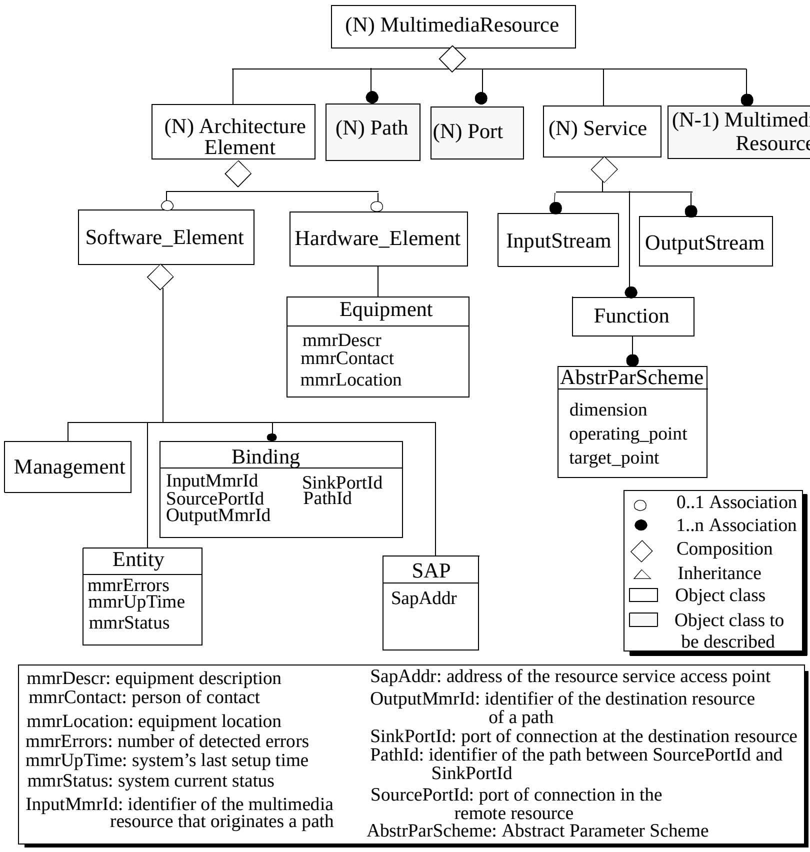


Fig. 2. A Generic Information Model of Multimedia Resources.

Concerning the software element, it gathers four classes, namely *Management*, *Entity*, *SAP* (Service Access Point) and *Binding* (Fig. 2). The *Management* class holds the information necessary for managing the resource. The *Entity* class is concerned with the behavior of the resource; it gathers the information necessary for computing statistics about the resource. The *SAP* class represents the point where the service, provided by the resource, can be requested. The values of the attributes of the *SAP* object class depend on the protocol used. Lastly, the *Binding* class illustrates the associations among the entities aggregated by the modeled resource.

Table 1. Examples of Binding objects.

Object identifier	Input MMR	Source Port	Output MMR	Sink Port	Path Identifier
Bd1	MMR1	A1	MMR2	B1	P1
Bd2	MMR1	A2	MMR2	B2	P2
Bd3	MMR2	B3	MMR3	C1	P3
Bd4	MMR2	B4	MMR4	D1	P4
Bd5	MMR3	C2	MMR1	A3	P5
Bd6	MMR3	C3	MMR1	A4	P6
Bd7	MMR3	C4	MMR1	A5	P7
Bd8	MMR4	D2	MMR1	A6	P8

Examples of *Binding* objects are given in Table 1; they correspond to the multimedia resource depicted in Fig. 1. Each row in the table represents a *Binding* object.

2.3.2. The Service Class

In order to provide the service required by external entities, multimedia resources must deploy certain functions in order to process the incoming streams (Fig. 2). The management of these functions relates to the five basic management functional areas defined by the ISO (i.e., configuration, performance, security, fault, and accounting management). Most of these functional areas together serve to manage the QoS required to the resource. According to the ITU [17], QoS is “the collective effect of service performances which determine the degree of satisfaction of a user of the service”. This definition focuses on the technical aspects of QoS. Accounting aspects are not emphasized. Our view of QoS management versus the management areas is depicted in Fig. 3. Performance, security and fault management contribute to the management of the QoS perceived

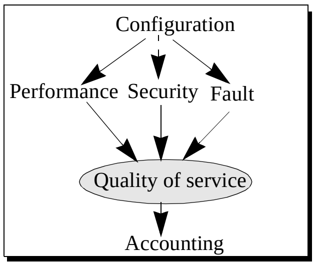


Fig. 3. QoS Management versus the Five Management Areas.

by the customer. QoS management provides account management with the information to be used for pricing and billing.

Unlike accounting, configuration management implicitly contributes to QoS management, via performance and fault management. The resource configuration influences the way fault, security, and performance are managed. Detecting faults within a well-configured device, that can collect and export its configuration information, is much easier than detecting faults in an entity whose configuration is secret and therefore cannot be appropriately managed. Security and performance are likewise much simpler to manage in the case of a device with a clear configuration.

QoS is the main quantifier of the service required from a multimedia resource. The functions related to its management (basically configuration, performance, fault and security management) are characterized by certain parameters. Distortion and resolution are examples of abstract parameters for the performance function. It is worth noting that these parameters are represented by their schemes in our model (Fig. 2). Indeed, especially in hardware-assisted devices, abstract parameters depend on other factors. For example, distortion depends on the level of the incoming signal. The accurate way to highlight this is to use curves.

Next, we describe another object class (i.e., *Port*) aggregated by the *(N)-MultimediaResource* object class (Fig. 2).

2.3.3. The Port Class

Our model of a resource port is depicted in Fig. 4. It looks very much like the global model for multimedia resources (Fig. 2), except that the *Binding* class does not appear in Fig. 4, and the traffic at the port is represented by a single *Stream* object class. Here, it is not necessary to have a *Binding* object because, in a port as depicted in Fig. 4, there are no components to bind with one another. Moreover, a port is assumed to not carry streams in the two directions; having input and output

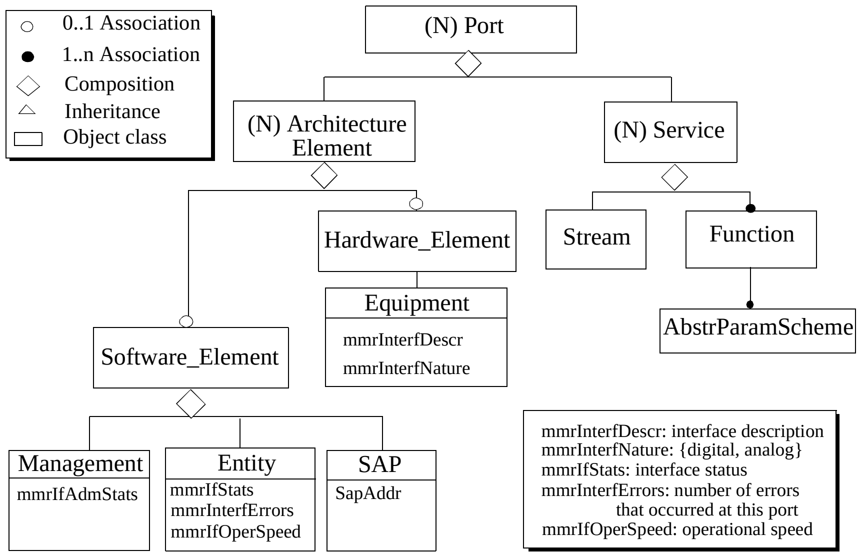


Fig. 4. A Model of the Port Object Class.

streams does not make sense in this case. Ports are linked to one another by a path, as described in the next section.

2.3.4. The Path Class

A *Path* object class represents the link between two *Port* object classes. This link may be physical, i.e., hardware-based, or logical. An example of a physical path is a wire connecting two ports. A logical path, on the other hand, refers to a function receiving data from one port, processing it and passing it on to an exit port. For instance, you can require a microprocessor to process an information and then send the result through a specific port. In such a case, the link between the input and the output port is strictly virtual. Fig. 5 illustrates both the software and the hardware parts of a path. The software element provides a logical view of the resource; this view is used to manage the underlying resource.

Paths can be bound to one another through a *PathBinding* object. As an example, paths that belong to the same communication session can be bound through a *PathBinding* object. Properties such as synchronization can then be accurately expressed. This object class may be used to set priorities on the tasks to be held by the resource. Assume that paths P5 and P6 (Fig. 1) belong to the same communication session. The processing tasks held by MMR3 may give a higher

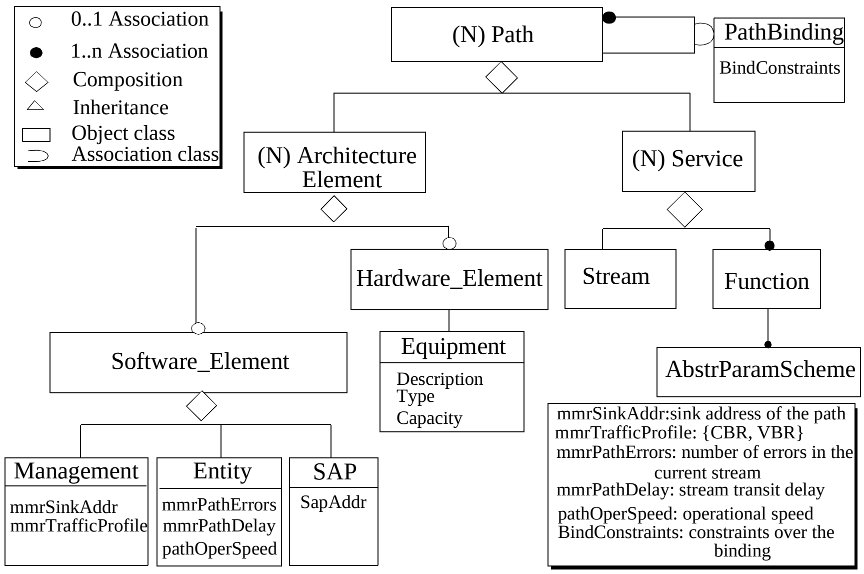


Fig. 5. A Model of the Path Object Class.

priority in treating these two paths, thereby affecting the processing of P7. Several instantiations of the *PathBinding* classes may be needed, depending on the purposes to be achieved; in the previous example, the purpose was synchronization.

In this Section, we proposed a generic information model for multimedia resources. This model was targeted for configuration and performance management. Although configuration information can be easily found in our model (paths, ports, and embedded resources), performance aspects are less evident to seize; this is because the model is still abstract. We emphasize performance aspects in the following section, where we specialize our model for a real multimedia resource: an MPEG2 video codec.

3. CASE STUDY: A MODEL OF MPEG2 VIDEO CODECS

For the sake of readability of the information model, we first describe MPEG2 video codecs; the latter's parameters that can influence the quality (performance) of a video service are highlighted. In order to keep the case study as simple as possible, we do not consider the systems layer standardized for MPEG2; we rather study the codec alone. The intent behind the design of this layer was to synchronize and transport MPEG2 streams over the network.

In section 3.1, we present the main features of MPEG2 video codecs. Then, we instantiate the generic model (Fig. 2) for these codecs (section 3.2). Lastly, we give examples that could advantageously use the model instantiated (section 3.3).

3.1. A Characterization of MPEG2 Video Codecs

An MPEG2 codec is organized into four layers which describe its internal activities: the block, macroblock, slice and picture layer [18]. A block is an image piece with a size of 8x8 pixels. A macroblock is a 16x16 pixel image piece. A slice is a horizontal strip within a frame; its coding is independent of that of its neighbors. These layers hierarchically lead to the construction of the encoded stream. The MPEG2 encoding scheme uses both spatial and temporal compressions in order to high compression ratios, while still providing an acceptable perceptual video quality.

There are 3 kinds of frames that come out of an MPEG2 encoder: intraframes (I), interpolative frames (B) and predictive frames (P). Together, they constitute the codec's modes - according to statements in [18], and to our terminology in section 2.2 -, i.e., the codec's activities as perceived from the outside. I frames are constructed using only spatial compression, whereas B and P frames result from both spatial and temporal compressions. "Spatial compression" refers to the reduction of the spatial redundancy. As a result, in sequences made up of only spatially-compressed frames, the loss of any frame does not undermine the decoding of the rest. On the contrary, temporal compression produces frames that can be tightly correlated. In order to reduce this correlation over time, the codec periodically encodes I frames which are "self-contained" (i.e., they are not correlated with the preceding frames). Moreover, I frames serve to reset errors due to the predictive scheme employed to construct P, as well as B, frames.

Very high compression ratios can be reached by using interpolative, or B, frames, which interpolate between a given (I or P) frame and the latter's predicted version. Hence, B frames introduce non-causal properties; being inserted between two frames on which they depend, B frames cannot be decoded until the two related pictures are actually decompressed. Therefore, the compression ratio, yet high, increases the decoding delay. Such a trade-off among characteristics of a codec should appear in the model; we meet this goal in our model in section 3.2. Generic codec characteristics, as well as some of their influencing parameters, are identified by Rodriguez *et al.* in [12]. These characteristics are:

- the *image fidelity*, i.e., how similar the encoded picture is to the original one from a perceptual point of view

- the *bandwidth*, i.e., how high the compression ratio is
- the *playback performance*, i.e., how easy the decoding process is
- the *coding-to-decoding asymmetry*, i.e., how similar the encoding features, such as the frame size, are to the display aspects
- the *memory consumption*, i.e., the amount, of memory and video buffer verifier, that is necessary for performing the encoding/decoding activities
- the *processing delay*
- the *scalability*, i.e., the property conferred to a codec whose algorithm compresses the video into multiple frame resolutions, multiple representations of the pixel color depth, and so forth; the encoded information is organized hierarchically in two kinds of layers: lower or base layer, and enhancement layer. The base layer is necessary to decode the enhancement layers; it ensures the minimum QoS.

We specialized and extended the work of Rodriguez *et al.* for an MPEG2 video codec (Table 2), using inputs from [6,12,18,19]. Our model is described next.

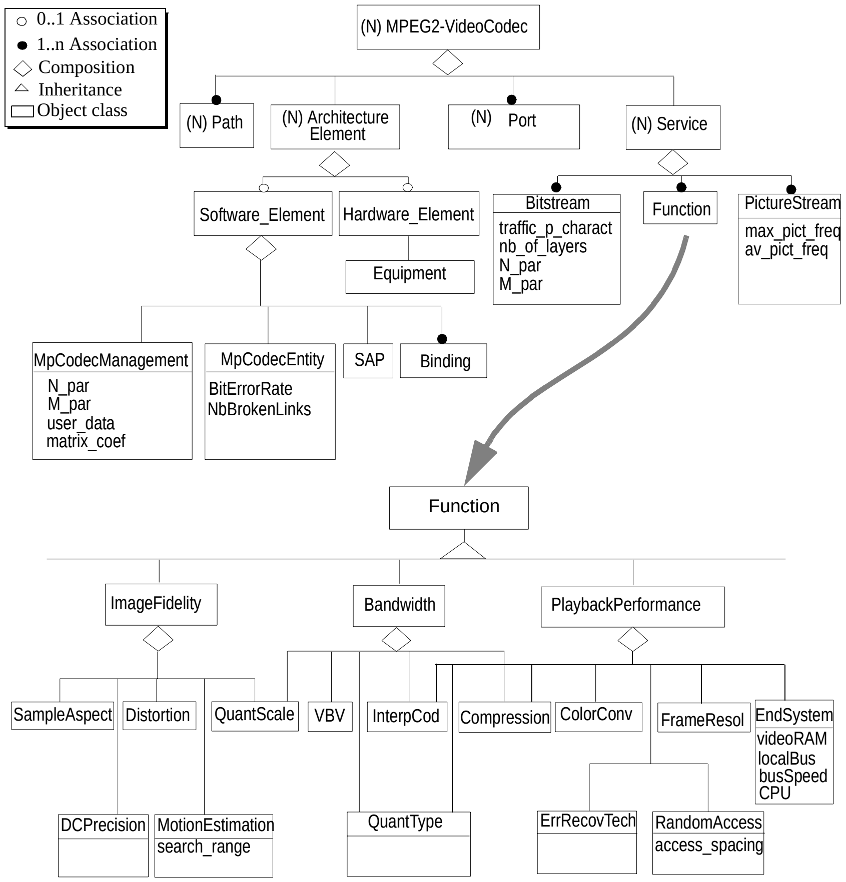
3.2. An Information Model of MPEG2 Codecs

The information model proposed (Fig. 6) comprises the functions to be deployed by the codec in order to meet a required QoS. These functions relate to either task (encoding or decoding) held by the codec. Functions that relate mainly to the encoding task are image fidelity, bandwidth and scalability. The function that relates mainly to the decoding task is the playback performance. All the other functions relate to both tasks. For the sake of brevity, we model only three functions: image fidelity, bandwidth, and playback performance (Fig. 6). Each function is described along with its influencing parameters. We can see (Fig. 6) that some parameters influence simultaneously many functions (in ways described in Table 2). For instance, the use of interpolative coding (*InterpCod* object class) has an impact on both the bandwidth and playback performance functions. The management information that we gather in our model can therefore be used when designing a software for MPEG2 codecs. The application designer may have a look at the model and get an idea of the parameters to take into account. Futhermore, the resource manager can obtain information about the behavior of these parameters.

The management of the functions and their influencing parameters is the responsibility of the *MpCodecManagement* entity whose attributes should be selected among the MPEG2 bit-stream fields [6],as we did in [20].

Table 2. MPEG2 Video Codecs: The Main Characteristics and Some Parameters.

Characteristics	Parameters	Influence
Image fidelity	- quality saturation (distortion threshold) - quantizer scale - motion estimation search range - interpolative coding - quantization matrices	- variable in CBR coding in order to maintain the bit rate; constant in VBR in order to diminish the variations in the image quality - its non-causal nature renders the stream more sensitive to errors
Bandwidth	- quantization scheme - traffic profile - interpolative coding	- adaptive or predictive schemes save bandwidth - it achieves high compression ratios
Playback performance	- frame rate - frame resolution - color-conversion operations - quantization scheme - interpolative coding	- its non-causal nature is unsuitable for real-time applications
Coding-to-decoding asymmetry	- pulldown - end-system features	- takes into account the difference between the source aspect ratio and the sink display aspect ratio
Memory consumption	- interpolative coding - picture structure	- it induces out-of-sequence transmission of frames - an interlaced frame may be coded either as a frame or as two separately coded field pictures; using these pictures saves memory and reduces the decoding delay.
Scalability	- frame resolutions - temporal resolutions - pixel color depths	
Processing delay	- lookup tables - quantization scheme - picture structure	- simplify (color) conversion operations - influences the number of color conversions - cf. row “Memory consumption”



traffic_p_charact : traffic characteristics
nb_of_layers: number of enhancement layer within the bit stream
N_par: ratio of nterframe to intraframe
M_par: ratio of intraframe to predictive frame
max_pict_freq: maximum picture frequency
av_pict_freq: average picture frequency
NbBrokenLinks: number of B frames that could not be decoded because I or P were corrupted
ErrRecovTech: error recovery technique
FrameResol: frame resolution
user data: information supplied by the user
matrix_coef: coefficients to be used for operations on the luminance and the chrominance
SampleAspect: distance between samples
DCPrecision: precision of the DC coefficients used
search_range: search range for estimating the motion vector
Quant*: Quantization *
VBV: Video Buffer Verifier
InterpCod: interpolative coding
ColorConv: color conversion
RandomAccess: random access to fields of the bitstream
access_spacing: minimum spacing between fields to access

Fig. 6. A Model of MPEG2 Video Codecs.

In addition to the *MpCodecManagement* class, the *Software_Element* class aggregates an *MpCodecEntity* class. The latter records information about the codec performance; examples are the codec bit error rate (BER), the number of broken links (number of B-pictures that could not be decoded because of corruption of the reference frame), the number of unrecoverable frame errors, and the number of recovered frame errors. Error recovery mechanisms (e.g., error concealment and forward error corection) can be used to increase the playback performance.

Finally, we can notice from Fig. 6 that two kinds of stream are treated by an MPEG2 video codec: *PictureStream* and *BitStream*. *PictureStream* relates to the pictures "captured" by the video codec and that are encoded further on, while *BitStream* describes the bit-stream resulting from the encoding process.

3.3. Examples of Use of the Model

The model can be used to equip the customer with tools that help her detect quickly the causes of any trouble with an MPEG2 board. Instead of returning to the vendor each time a simple problem occurs, the customer can develop her own management solutions.

Having such a model benefits also the service creator. When developing an application for an MPEG2 card, the service creator can simply have a look at our model, and find out the balanced influence of MPEG2 parameters on the functions deployed by the card. This would help her find the best trade-offs among these parameters, in order to fulfill the specific requirements of the application in design. We do not presume that a model will solve the problems all together, but it can be the first tool that calls the attention of the developer to specific trade-offs that should be taken into account for an efficient software design.

Our final example is "live" transmission of video, which is one of the primary targets of MPEG2. The example is borrowed from a work by Brenner and Meandzija [21] on broadband video/audio/data distribution networks (Fig. 7). Brenner and Meandzija proposed a distribution network architecture as a solution to the growing attraction of users for interactive services, whose contents change dynamically. Faced with this more stringent requirement, Local Exchange Carriers (LEC), Competitive Access Providers (CAP) and Multiple Systems Operators (MSO) are currently seeking network architectures and distribution technologies that help them minimize their investment and optimize the QoS delivered to the users.

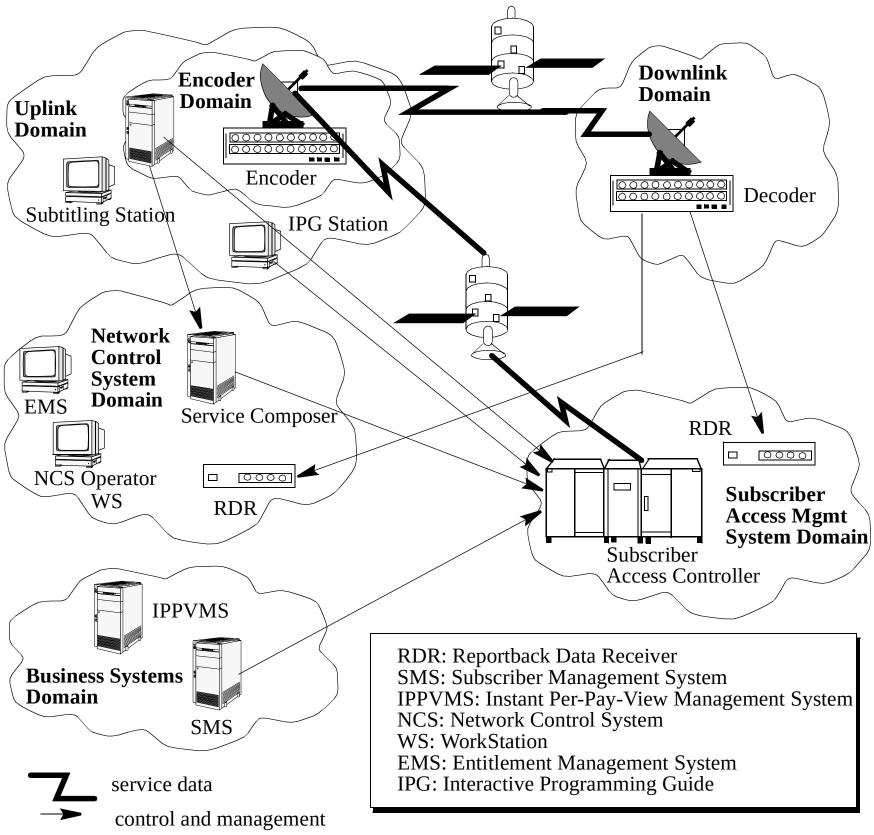


Fig. 7. Use of the MPEG2 Model in the Encoder Domain of Distribution Networks [21].

Briefly, the distribution architecture proposed by Brenner and Meandzija is made up of many domains (Fig. 7) described below.

- The *uplink domain* (including the encoder domain) is concerned with providing, controlling and managing the encoding devices. The encoders are controlled and managed by Uplink Control Systems (UCS). The uplink domain may embed Interactive Programming Guide (IPG) stations as well as subtitling stations. An IPG station is a cache that is used for giving to the user, the illusion of rapid access to a large interactive database. Subtitling servers may be needed for translating the information into a language that is understandable by the service user. ***The model designed in this paper can be used by the UCS as the information base for controlling and managing the encoders.***

- The *Network Control System* (NCS) domain consists of programs authoring stations, and Entitlement Management Systems (EMS). Program authoring stations run the Service Composer which creates programming services, schedules programs, and manages the UCS accordingly. The EMSs contain Reportback Data Receivers (RDR) and Network Management Stations.
- The *Business Systems domain* is made up of a number of systems dealing with billing, as well as the management of the subscribers' profile. Among these systems are the Subscriber Management System (SMS) and the Instant Pay-Per-View Management Systems (IPPVMS).

- The *downlink domain* can be of two kinds: Direct Broadcast System (DBS) - essentially end-user Integrated Receiver Decoders (IRDs) -, and commercial IRDs feeding a CATV network. The downlink domain may hold head-end transcoders, encoders/decoders, cable head-end, management systems for the head-end, and distribution plants.
- The *Subscriber Access Management System domain* consists of systems performing subscriber authorizations, key management, etc.

Control and management interactions take place between the domains (Fig. 7). Elements in the uplink domain interact with elements in the encoder domain, mainly in order to create, store, and provide the encoders with contents. Elements in the NCS domain issue control and management commands to elements in the uplink domain, including the encoder. Components of the Subscriber Authorization domain interact with the other domains in order to supply information about the customers who are allowed to use the service.

The need for integrated management of the distribution networks, as urged in [21], will definitely require putting together efforts from the signal processing and the network management areas, in order to build up a consistent end-to-end management solution. The work achieved in this paper is intended to be a first step toward this objective.

We have not talked about the protocol to be used for accessing the information gathered in either Fig. 2 or Fig. 6. It is not our intention to tie our information model to any specific management tool, among those which are now at play. Examples are the many management solutions based on the Web, such as the Web-Based Enterprise Management (WBEM) [22], the Java Management Application Programming Interface (JMAPI) [23], Webbin' [24], and [25]. While the previous version [20] of this work was implicitly targeted for management based on the Simple Network Management Protocol (SNMP) [26], the version described in this paper is more “generic”, i.e., independent of management paradigms.

4. CONCLUSION

The main achievements in this paper are the design of a generic information model of multimedia resources, and the specialization of this model for an MPEG2 video codec. Throughout this case study, we were particularly interested in identifying the parameters that influence the functions deployed by MPEG2 codecs. We showed that some parameters may influence many functions in different ways. This observation might be useful for the application developer who wishes to take advantage of the trade-offs among these parameters in order to provide the user with an efficient service. We also illustrated a possible application of our model in distribution networks, particularly for video transmission. This use contributes to the management of the entire QoS provisioning system, i.e., network resources as well as multimedia resources. QoS can thus be managed on an end-to-end basis; the benefit is evident to both the user and the service provider.

This paper paves the road for the overall management of multimedia resources. It is part of a larger project called OAMS (Open management Architecture for Multimedia Services) [27], which aims at designing and developing a management architecture for advanced telecommunication services such as Virtual Private Networks (VPN) and multimedia services.

In the context of OAMS, we also investigated the use of the information model to map QoS across the control and management layers from the end-user’s view down to the connectivity level [28]. We explored QoS mapping with a novel approach which consists in bridging the results of the human vision research area with Telecommunications, in order to take advantage of the user perceptual features. As a matter of fact, not all information can be noticed by the end-user. Consequently, an efficient QoS mapping framework should integrate these psychovisual effects. The QoS framework proposed in [28] specifically exhibits an information collecting and management plane which contains information such as the model of MPEG2 video codecs proposed in this paper, as well as human vision knowledge.

ACKNOWLEDGMENTS

The authors wish to express their gratitude to the anonymous reviewers whose feedback helped to improve the quality of the paper. They would also like to acknowledge Olivier Verscheure, Carlos Lopez and Slim Gara for the fruitful discussions on MPEG2 and perceptual video QoS. Very special thanks go to Holly Cogliati for her numerous comments on the paper.

REFERENCES

1. N. Natarajan, F. Dupuy, N. Singer and H. Christensen, Computational Modelling Concepts, Vers. 2.0, Document label TB_A2.HC.012_1.2_94, TINA Consortium, February 1995.
2. N. Berndt *et al.*, Definition of Service Architecture, Document label TB_B.MDC.012_2.0_94, TINA Consortium, February 1995.
3. H. Mulder and J, Pavon, (eds.), Special Issue on Telecommunications Information Networking Architecture (TINA), *Journal of Network and Systems Management*, Vol. 5, No. 4, Plenum Press, Dec. 1997.
4. G. Pavlou and D. Griffin, Realizing TMN-like Management Services in TINA, *Journal of Network and Systems Management*, Vol. 5, No. 4, Dec. 1997, Plenum Press, pp. 437-57.
5. Y. Katsube, Towards Flexible Communications Management for Multimedia Multipoint Communications, *IECE Transactions on Communications*, Vol. E77-B No. 11, Nov. 1994.
6. ISO/IEC/ DIS 13818-2, *Information Technology- Generic Coding of Moving and Associated Audio Information- Part2: Video. Draft International Standard*, 1994.
7. J. Sclavos, N. Simoni, S. Znaty, Information Model: From Abstraction to Application. In *Proc. of IEEE NOMS*, Florida, USA, 1994.
8. ITU-T, *Management Framework for Open Systems Interconnections*, Rec. X.700, 1993.
9. J. Rumbaugh *et al.*, *Object-oriented Modeling and Design*, Prentice Hall International Ed., 1991.
10. M. Rose and K. McCloghrie, *Structure and Identification of Management Information for TCP/IP-based Internets*, IETF RFC 1155, STD 17, May 1990.
11. ITU-T, *Information Technology - Abstract Syntax Notation One (ASN.1): Specification of Basic Notation*, Rec. X.680, July 1994.
12. A. A. Rodriguez and K. Morse, Evaluating Video Codecs. *IEEE Multimedia*, Vol. 1, No. 3, pp.25-33, Fall 1994.
13. ITU-T, *Generic Network Information Model*, Rec. M.3100, July 1995.
14. ITU-T, *Generic Functional Architecture for Transport Networks*, Rec. G.805, Nov. 1995.
15. H. Yamaguchi, S. Isobe, T. Yamaki and Y. Yamanaka, Network Information Modeling for Network Management, In *Proc. of IEEE NOMS*, 1992, pp. 57-66.
16. J. P. Gaspoz, J. P. Hubaux and S. Znaty, A Generic Architecture for VPN Configuration Management, *Journal of Network and Systems Management*, Vol. 4, No. 4, Dec 1996, Plenum Press, pp. 375-395.
17. ITU-T, *Quality of Service and Dependability Vocabulary*, Rec. E.800, 1993.
18. P. Pancha and M. El Zarki, MPEG Coding For Variable Bit Rate Video Transmission. *IEEE Communications*, May 1994.
19. F. Gadegast, *MPEG FAQ 4.1*. Available at http://www.mpeg1.de/mpegfaq/.
20. C. Gbaguidi, S. Znaty, J. P. Hubaux and O. Verscheure, Management of Multimedia Resources: From a Generic Information Model to its Application to an MPEG2 Video Codec. In Finn Arve Aagesen, Harald Botnevik, and Dipak Khakhar, (Eds.), *Information Network and Data Communication*, Chapman and Hall, pp.441-457, June 1996, ISBN 0-412-75750-8.
21. A. Brenner and B. Meandzija, Broadband Video/Audio/Data Distribution Networks - The Need

for Network Management. In Aurel Lazar, Roberto Saracco, and Rolf Stadler (Eds.), *Integrated Network Management V: Integrated Management in a Virtual World*, Chapman & Hall, pp.396-417, 1996, ISBN 0 412 80960 5.
22. J. Horrowitz and M. Bolsinger, Accessing TMN through Web-based Enterprise Management, Vers. 1.1, Vertel Inc., Feb. 1997.
23. Sun Microsystems, Inc., Java Management API Architecture, Revision A, September 1996.
24. F. Barillaud and L. Deri, Network Management using Internet Technologies. In Aurel Lazar, Roberto Saracco, and Rolf Stadler (Eds.), *Integrated Network Management V: Integrated Management in a Virtual World*, Chapman & Hall, pp.61-70, 1996, ISBN 0 412 80960 5.
25. M. Maston, Using the World Wide Web and Java for Network Service Management. In Aurel Lazar, Roberto Saracco, and Rolf Stadler (Eds.), *Integrated Network Management V: Integrated Management in a Virtual World*, Chapman & Hall, pp.71-84, 1996, ISBN 0 412 80960 5.
26. J. Case *et al.*, *Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)*, IETF RFC 1905, Draft Standard, January 1996.
27. S. Znaty, Service and Network Management in the OAMS Open Service Architecture, *ACM Computer Communication Review*, Vol. 26, No. 3, pp. 59-74, July 1996.
28. C. Gbaguidi, O. Verscheure and J. P. Hubaux, A New Flexible and Modular QoS Mapping Framework Based on Psychophysics. In R. Boutaba and A. Hafid, (Eds.), *Management of Multimedia Networks and Services*, Chapman & Hall, pp.67-80, 1998, ISBN 0 412 82630 5.

Constant Gbaguidi graduated in electrical engineering at EPFL, Lausanne, Switzerland, in 1995. Since then, he has worked at EPFL as a research and teaching assistant in telecommunication services and management. His diploma thesis was on the implementation of a Virtual Private Network (VPN) over the OSF's Distributed Computing Environment (DCE). In the summer of 1996, he was a Visiting Scholar at Columbia University, New York, NY, where he worked with the COMET group on porting a network prototyping framework to CORBA. Currently, his interests span distributed computing, QoS management, and use of the Internet to facilitate service management.

Simon Znaty obtained a Ph.D. degree in computer networks from ENST, Paris, France, in 1993. From 1993 to 1994, he worked with the Network Architecture Laboratory at NTT, Japan. From 1994 to 1996, he was a senior researcher with the Telecommunications Laboratory at EPFL, Lausanne, Switzerland. Since January 1997, he is professor at ENST-Bretagne, Rennes, France, where he teaches and does research in Telecommunication Services Engineering. His current interests are in the evolution of the Intelligent Network toward mobile networking, management of telecommunication services, active networks and services, and distributed computing.

Jean-Pierre Hubaux has been an associate professor with EPFL since 1990. He is now co-director of the Institute for computer Communications and Applications (ICA). His current areas of interest cover service engineering and multimedia services, with a strong focus on the impact of the Internet on these areas. He supervised projects on service specification and validation, virtual private networks, telecommunications management, and support for multimedia services in ATM networks. Prior to joining EPFL, he spent 10 years in France with Alcatel, where he was involved in R&D projects, mostly in the area of switching systems, spanning architectural and control software issues. He was also in charge of the introduction of artificial intelligence techniques, especially for system troubleshooting. He is currently on sabbatical leave at the IBM T.J. Watson Research Center in Hawthorne through January 1999.