

First SICMA Trial: The Multimedia Server and the Virtual Museum Application

Y. Mavraganis¹, Y. Maragoudakis², N. Pappas³, G. Kyriakaki⁴, K. Meyer⁵, W. Lamotte⁶, C. Brandt⁷, P. Berenbrink⁸

Abstract. This paper presents the SICMA Multimedia Information Server that was demonstrated during the first public trial of the ACTS SICMA project, which took place in Natural History Museum of London and in Gallo Romeins Museum of Tongeren (Belgium), from June to September of 1997. The demonstrator application, namely the “Virtual Museum”, is also presented. The SICMA server complies to the DAVIC standard and is built on top of the KYDONIA multimedia information server. Core DAVIC Server Service elements (Service Gateway, Stream, Content, File) have been developed and the DAVIC User Plane information flows (S1 and S2) are supported. The KYDONIA multimedia information server implements special techniques- object-oriented multimedia data model, scheduling mechanisms, parallel placement- for efficient management of multimedia data on the internal level of SICMA server. In the context of the first trial, the role of the SICMA server was to store, manage and distribute on demand, over ATM networks, multimedia content to Set Top clients of the Virtual Museum application, which accepted the requests of museum visitors (end users). We describe the overall SICMA server architecture and explain the main factors that lead to server design. We also identify the critical points which affected the overall system performance and determined the degree of compliancy with the accepted standards, such as DAVIC.

1. Introduction

In recent years advances in storage mediums and communication technologies make it possible to store an enormous (and still growing) amount of information like video, audio or text in digitised form. These stored information can be made accessible and available (using digital processing) to many users simultaneously, in applications like Video on Demand, Telemedicine, Distance Learning, or Home Banking. Real-time requirements, very large data transfer rates and storage space requirements make the design of a Multimedia Information Server so difficult, and make it necessary to study the problem from a different point of view than the research about file systems or usual databases does.

The SICMA project (*Scaleable Interactive Continuous Media Server - Design and Application*) is part of the Advanced Communication Technologies and Services (ACTS) Programme of the European Union, since the first call of 1995. The general aim of SICMA is to design a scaleable server for the delivery of images, data and continuous multimedia

¹ Multimedia Systems Institute of Crete/Technical University of Crete (MUSIC/TUC)

² MUSIC/TUC

³ MUSIC/TUC

⁴ MUSIC/TUC

⁵ University of Paderborn

⁶ Limburg University

⁷ University of Paderborn

⁸ University of Paderborn

information, over high speed networks, to clients, and to demonstrate its efficiency with the help of a relevant application, namely the "Virtual Museum".

There is a lot of ongoing research in the area of multimedia information servers. In [GVKR95] a tutorial and survey for multimedia storage servers is given. [V94] presents a multimedia system architecture consisting of an information management, a storage and a network subsystem. In [CT95] research and development issues- such as multi-level scheduling, content-based retrieval of multimedia data- for large-scale multimedia information servers are discussed. [TJHHJLCR94] describes the design, implementation and analyses of a distributed parallel storage system, able to supply image streams fast enough to permit multi-user real-time video-like applications. In the research project MARS (Massively-parallel And Real-time Storage, ([BPC94]) a large scale multimedia server is designed and implemented. It uses some of the well-known techniques in parallel I/O, such as Data Striping and Redundant Arrays of Inexpensive Disks (RAID). It also uses an innovative, ATM based, interconnect to achieve a scaleable architecture that transparently connects storage devices to an ATM-based broadband network. The experiences and implementation details of an Ethernet-based server called 'Stony Brook Video Server' are presented in [VVC96]. The server integrates a software-based disk array with a Real-Time Ethernet Protocol, called RETHER, which guarantees the smooth delivery of multimedia data. The paper focuses on three main design and implementation issues: buffer management, retrieval algorithms and client-side support for rewind of MPEG videos. Design and implementation issues for video on demand servers, compliant to state-of-the-art standards, are also presented in [CEA96].

The SICMA server complies to the DAVIC standard. DAVIC (Digital Audio Video Council) is an international organisation, founded in 1994, that includes major telecommunication companies, CATV operators and manufacturers of video servers and Set Top Boxes. Its purpose is to promote the introduction of residential interactive audio-visual services in an open way. This can be achieved by enabling end-to-end interoperability across systems, applications and services, through the establishment of internationally agreed open interfaces and protocols. DAVIC has released a set of specifications that describe the overall infrastructure (core systems architecture, APIS, protocol stacks) for the provision of interactive multimedia services from large information servers, through high speed networks, to Set Top Units at home. MUSIC/TUC and MSC have developed the appropriate software for the DAVIC compliancy of the server. Thus, the SICMA server can provide multimedia services to DAVIC compliant execution environments, and support DAVIC-compliant applications with minimum implementation effort.

The DAVIC server is built on top of the KYDONIA system [CPKMMT97], a multimedia object database management system, that has been developed by MUSIC/TUC and is expanded, within SICMA, for providing full multimedia support. The main features of the system are: storage management for multimedia objects, real-time data pumps towards multiple clients over ATM, Multimedia DBMS functionality, multimedia object modelling, text and video access methods and browsing techniques. KYDONIA acts as a multimedia data manager inside the SICMA server.

The final aim of SICMA is to build a parallel multimedia server because parallel systems are able to deal with a very large disk array. Compared to sequential systems, they are able to store a larger number of media streams and to serve the requests of a larger number of users at the same time. While parallel systems are hard to use for certain problems, they seem to be well suited for the delivery of continuous media. This is because users usually access streams stored on the server independently from each other. Thus, a natural parallelism is given and can be exploited most efficiently by a parallel system.

The demonstrator application presents Captain Cooks exploration of Australia. In 1768, Captain James Cook sailed to Australia in order to explore the unknown south pacific. The goal was to find a mysterious continent and to explore the plants and animals living there. The visitors of the exhibition in Natural History Museum of London, are enabled to go on board of a virtual model of Captain Cooks ship 'Endeavour'. The exploration of the ship is accompanied by videos, pictures, stories and 3D objects which are presented in the virtual

rooms of the ship. Incorporating Virtual Reality techniques, video animation and still picture presentation makes this Virtual Museum application interesting to be viewed within the NHM, and also as an in-house application.

The first SICMA trial took place in Natural History Museum of London and in Gallo Romeins Museum of Tongeren (Belgium), from June to September of 1997. The client subsystems were interconnected via ATM networks with the SICMA server (Interactive Continuous Media Server or ICMS), and retrieved multimedia content (videos, audios, images) from the server, each time an end-user asked for a certain object (on demand). The two SICMA servers, for some periods, were interconnected via the ATM JAMES network for the demonstration of a Remote Access scenario, where the clients retrieved multimedia content, not from the local, but from the remote server. The following figure describes the overall SICMA infrastructure for the first trial.

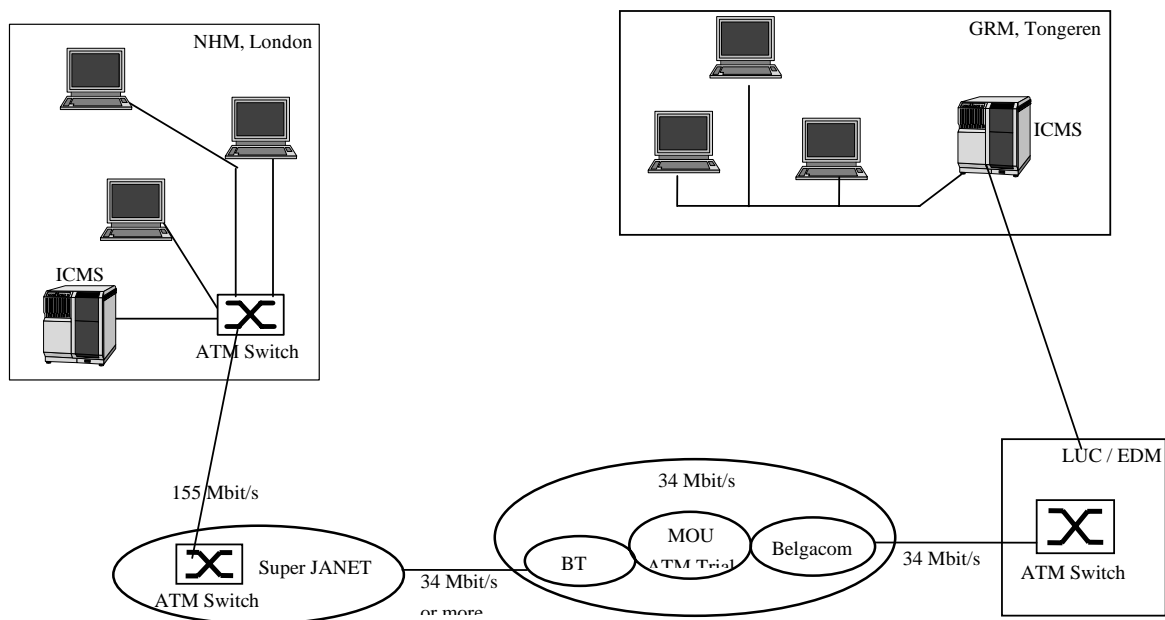


Figure 1-1 First SICMA Trial Infrastructure

The main goal of this paper is the extraction of useful conclusions concerning the development of DAVIC multimedia servers, as well as the overall presentation and evaluation of a trial that used state-of-the-art technologies and which finally proved to be very interesting for the end users (visitors of the museums). The structure of the paper is as follows: in the following section we describe the overall SICMA server architecture. Section 3 presents the DAVIC layer of the SICMA server and section 4 the KYDONIA server. Section 5 describes the D-Server modules of the SICMA server, which are responsible for parallel data block access. In next section, the Remote Access Scenario is described. Section 7 presents the Virtual Museum Application and section 8 concludes and presents future directions.

2. Overall SICMA server architecture

In this part, we describe the overall software architecture of the SICMA server that was installed in Natural History Museum of London, and provided multimedia services to clients of the Virtual Museum application, during the first trial phase of SICMA. The basic layers of the SICMA server software architecture as presented in Figure 2-1 are: the DAVIC Shell, the KYDONIA agent, the KYDONIA server and the D-Servers.

The SICMA server follows the server architecture identified in the DAVIC 1.0 standard (see Figure 3-1 DAVIC Server Reference Model). The DAVIC Server Service Elements have been implemented and the protocol stacks of the basic server information flows are supported. The DAVIC Shell (upper layer of the SICMA Server) is comprised of the Service Elements: ServiceGateWay (SGW), Stream, Content and File, that were used by the Content Provider and the consumer clients for installation and retrieval of the Virtual Museum application multimedia content (video, audio, images), in a DAVIC compliant way.

The SGW Service Element acts as the entry point into the server. It organises the service domain of the DAVIC server and enables external clients to discover and select available services as well as to install new services within the server system. The Stream Service Element is responsible for the control of media streams. The File Service Element is used for the retrieval of images from the server and the Content Service Element provides functionalities for installing content inside the server. A CORBA-compliant Object Request Broker (Iona's Orbix product) was used for the transfer of client requests (DSM-CC commands) to DAVIC service elements of the server.

For each Service Element -except for the Service Gateway, that is unique for all clients- there exists a corresponding *Factory Object* which manages the lifecycle (creation, deletion) of service element instances. The Service objects, that will serve the requests of a specific client, are created on demand, when the clients requests the opening of a certain service, and deleted when the client no longer needs that service. The DAVIC Service Elements use multimedia data base functionalities provided by the KYDONIA server for accessing (storage, retrieval, getting attributes) the application's multimedia content.

The communication of the DAVIC layer service element objects with the KYDONIA sublayer is achieved via the KYDONIA Agent module, that acts as a filter by which functionalities of the KYDONIA server are exported to the DAVIC layer. KYDONIA Agent is also responsible to direct data flows from the KYDONIA Transmission module to the corresponding service element object which will send it, over ATM (S1 flow), to the client.

The KYDONIA server, described in [CPKMMT97], is responsible for organizing the multimedia content (storage, retrieval) in the internal layer of the SICMA server. It is comprised of the Transmission module, the Object Manager and the Storage Manager.

The Transmission module is responsible for sending the requested multimedia data objects from the KYDONIA layer to the service objects of the DAVIC layer, via the KYDONIA Agent. The Object Manager is responsible to manage the KYDONIA objects in main memory of the server. It maintains and provides access to system objects, such as descriptors and indexes. It is also responsible for the transaction management inside the server.

The Storage Manager is responsible of the permanent storage of data objects. It is capable of servicing multiple clients ,that request either simple objects or streams, simultaneously. The storage system applies efficient strategies for data placement, appropriate buffering schemes, and scheduling policies in order to manage satisfactory multimedia data streams.

Finally, the D-servers, which are found in the lower level of the software hierarchy, are responsible of writing and reading data on/from the disk subsystem. D-servers accept requests, on a data block level, from the KYDONIA Storage Manager.

In the following figure the SICMA Server software architecture is presented.

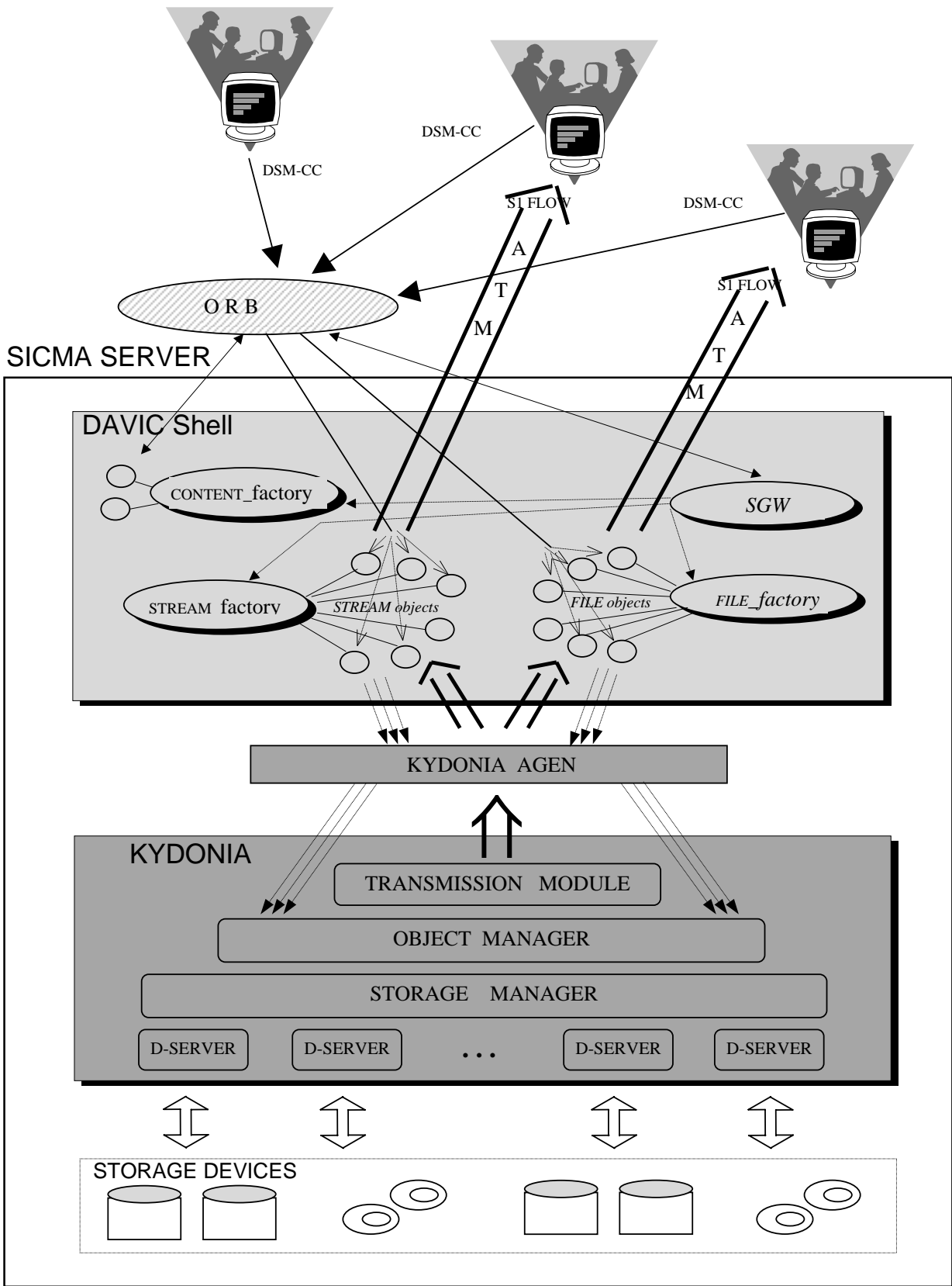


Figure 2-1 Overall SICMA Server Architecture

3. The DAVIC Shell

3.1 Overview

The architecture of a DAVIC Service Provider System (Server) follows a distributed, object-oriented approach. It is described as a collection of services which support multimedia-oriented applications, each with a known function and interface. The following figure presents the DAVIC Server Reference Model (only the core Services are represented).

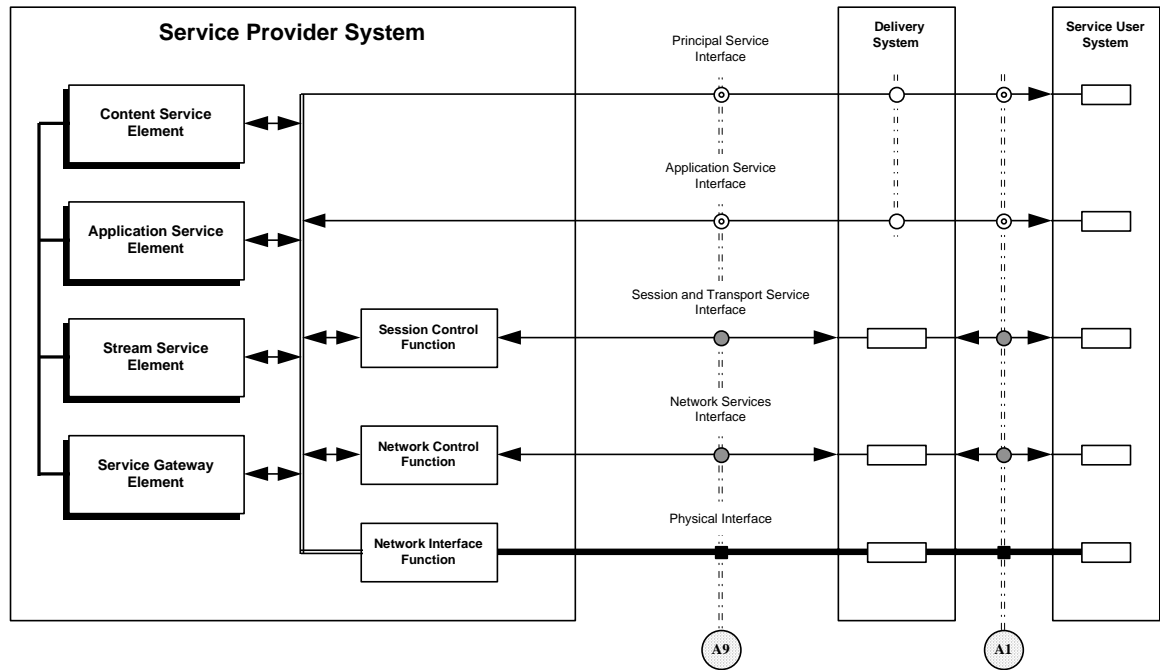


Figure 3-1 DAVIC Server Reference Model

The DAVIC Server Services are the following:

- Service Gateway
- Application Service
- Stream Service
- Content Service
- Session Gateway Service
- File Service
- Download Service
- Client Profile Service

The interfaces and protocol stacks supported by a DAVIC-compliant Server are categorized in five information flows (S1-S5), which are described in the following table.

S1	Used to carry high-bandwidth, bulk data over a transport stream such as MPEG-2.
S2	Used for RPC requests/replies between clients and server.
S3	Used for managing sessions.
S4	Used for User-to-Network and Network-to-User Signalling.
S5	Used for management interfaces and MIB's.

Table 3-1 DAVIC Information Flows

During the first phase of the SICMA project, most of the DAVIC Server Services have been developed and the protocol stacks for S1 and S2 information flows were supported. Specifically, the Services: Service Gateway, Stream, Download, File and Content have been designed and implemented. During the first trial in Natural History Museum of London, all of these server services, except the Download Service, were used by the client application. In the following subsections, we will present design and implementation issues of the DAVIC Service Elements and information flows, and discuss the degree of DAVIC compliancy achieved in the first trial.

3.2 The Service Gateway Service Element

The ServiceGateway is a fundamental Service Element that organises the service domain of the DAVIC server. It enables external clients -producer and consumer clients- to discover and select available services as well as to install new services within the server system. The overall functionality of this Service Element can be categorized as following:

- creating namespaces of services by binding Services to Names and browsing within these namespaces.
- asossiate names with data values
- establish and release session contexts within the distributed environment

They clients of the system "use" the ServiceGateway element, for discovering and selecting (opening) server services. The ServiceGateway provides to clients, by establishing communication paths (sessions) and returning object references, the means for invoking operations on a service's exported interface. The interface of the Service Gateway element, as it is identified in the DSM-CC standard and adopted by DAVIC, is comprised of the DSM-CC User-to-User Interfaces: *Access, Naming Context, Directory and Session* [DSM-CC].

The DSM-CC Access, Naming Context and Directory interfaces were supported by the server for the first Trial. The Session operations (*attach, detach*) were not available at that time- the development of the Session Service Element and the DSM-CC User-to-Network messages was in progress- and the application used the CORBA compliant *_bind* operation for initial interconnection with the Service Gateway Element of the server. The Naming Context operations (*bind, resolve* etc..)- supported by the Orbix Naming Service- were mainly used by the client application.. The *resolve* operation was used by the client for obtaining an object reference to a Server Service (e.g to a Stream Service) and the *destroy* method for releasing access to the Service, when it was no longer needed.

The Service Gateway, each time the opening of a new service is requested by a client (in the context of the *resolve* operation), it calls the corresponding Factory Element for creating a new service object for this client. The Factory Objects are responsible for managing the lifecycle (creation, deletion) of service element instances. In this way, the service instances are created on demand and not during the service installation phase. The Factory concept, although not specified in the DAVIC 1.0 specifications, it is introduced in the DAVIC 1.1 document for the Distributed Server Reference Model. For each kind of service- except for

the Service Gateway Service Element- there exists a corresponding Factory Element (Stream Factory, Download Factory etc..) which is responsible to create and destroy service instances of this kind.

3.3 The Stream Service Element

The Stream Service Element is responsible for the control of media streams, by exporting an appropriate interface through which the client controls the advance of the media stream. Stream provides some primitives which are used to emulate VCR-like controls for manipulating MPEG continuous media streams. The Stream interface is comprised of the methods: *pause*, *resume*, *status*, *reset*, *jump*, *play*, which are described in the DSM-CC standard [DSM-CC]. Each Stream Service Element employs a stream state machine to keep information about the operation that is currently executed, the progress of the stream transmission and the allowed future choices. The Stream objects communicate, through shared memories (double buffering techniques), with the KYDONIA server, for receiving the data streams coming from the storage level. The implementation is based on threads, for achieving better performance.

The operations used by the application were the *resume* and *pause* operations. All the streams were requested by the client on the normal play rate, because the end-user was not provided by the ability to change the rate (and the direction) of the stream. Thus, operations such as fast forward and fast backward did not take place. The Stream Service Element was further expanded for receiving the stream data, not only from the local KYDONIA Storage Manager, but also from the ATM network in order to support the Remote Access Scenario, that is described in a following section.

3.4 The File Service Element

The File Service Element is used by the clients for reading and writing files which are stored inside the server. The File service exports an appropriate DSM-CC interface (*open*, *close*, *read*, *write*) for accessing files. In the context of the first trial, the application used the File Service Element to get/read the images that were stored as separate files in the DAVIC server. So, the application firstly obtained, through the *resolve* operation of the Service Gateway Element, an object reference to a File Service Instance and then called the *read* operation for retrieving the whole image from the server. The File Service Objects, like the Stream objects, retrieve the image objects from the KYDONIA server by calling the appropriate methods of the KYDONIA Agent and using shared buffers with KYDONIA, for receiving the image data. The File *write* operation was not used by the application, because for writing the image files to server the *load* method (described in the following section) of the Content Service Element was used at content installation phase.

3.5 The Content Service Element

The DAVIC Content Service Element defines the standard content types of which a Content Provider can specify instances, as well as the interface to manipulate them, inside the overall architecture of the Service Provider System. The Content Service Element has a standard IDL interface, that inherits from DSM-CC Directory and defines 4 additional operations : *copy*, *modify*, *move* and *load*. All the operations of the Content interface function on content packages or content elements within a content package and apart from copying, modifying, moving and loading content, also handle content binding in the naming context of the specific Content Service element. Thus access to content is based on the naming mechanism provided by the CORBA standard.

In the context of the public trial of the SICMA project, that was held in the Natural History Museum, the Content Provider that was the museum itself, owned a set of content objects consisting of :

- MPEG-2 video and audio streams that were multiplexed and coded as MPEG-2 Transport Streams for real-time delivery and synchronization
- RGB (SGI format) images
- a set of other data that were residing in the application platform and were never uploaded to the server

The content loading process consists of the DAVIC Content Transfer API (A10 interface) and loads individual content items from local storage. The SICMA server internally manages content through ODBMS functionality supported by the KYDONIA system.

3.6 DAVIC Information Flows

As it was previously described, the interfaces of a DAVIC-compliant server are categorized in five information flows (S1-S5). The protocol stacks of these information flows follow the OSI Reference Model. DAVIC has selected appropriate protocols for the different layers, for each information flow. The main efforts during the first trial phase of SICMA have been concentrated on supporting the S1 (data channel) and S2 (command channel) User Plane information flows [DAVb95].

The S1 information flow is for content-information flow from a source object to a destination object. It is defined to be a uni-directional flow from the server to the STU carrying encoded video/audio content over a transport stream [DAVb95]. The protocol stack adopted by DAVIC, for carrying S1 information flows is shown in the following figure (ATM-based transmission).

Other Data	MPEG Video Elementary Stream	MPEG Audio Elementary Stream	
MPEG-2 Private Section	MPEG-2 Packetised Elementary Stream (PES)		MPEG-2 Program Specific Information
MPEG-2 Transport Stream (Single Program TS)			
AAL 5			
ATM			
PHY			

Figure 3-2 S1 Information Flow Protocol Stack

For supporting the S1 information flow, MUSIC/TUC and MSC have developed a module that constructs MPEG-2 Transport Streams from audio and video files that must be synchronized. This module has been used in the first public trial of SICMA, in order to support a DAVIC-compliant environment, a key issue of which is the real-time delivery of data as MPEG-2 Transport Streams. This module has been tested both by the MUSIC/TUC and MSC teams, as well as by the UPB team. The Transport Streams that it produces are of very high quality and fully compliant with the MPEG-2 Systems standard, as well as the VideoPlex video board that was used in the hardware environment of the first trial application.

After verifying the quality results of the module, UPB used the software provided by MUSIC/TUC and MSC and created the MPEG-2 Transport Streams from all the audio and video material that the NHM provided.

The construction of MPEG-2 Transport Streams was an important success factor for the trial of SICMA in the following ways :

- the DAVIC standard specifies that the MPEG-2 TS protocol must be used for real-time data delivery.
- this transport protocol is the best suitable for transmission over ATM networks and actually resulted in the correct and efficient transmission over the European JAMES network.
- the MPEG-2 Transport Stream standard achieved the perfect synchronization of the audio and video content files.
- the option of storing MPEG-2 Transport Streams and having them ready for real-time delivery at any time, saved the SICMA server from the overhead of constructing them real-time.

Apart from off-line creation of the stored Transport Streams, the S1 interface of the SICMA server as has been developed by MUSIC/TUC and MSC, has the ability of constructing Transport Streams in real-time and transmitting them over ATM networks using the AAL5 (mapping of MPEG-2 Transport Stream into AAL5 according to [DAVb95]). For the trial, the protocol stack for carrying the MPEG-2 transport streams was IP over ATM, because the ATM cards of the PC clients (simulated Set Top Boxes) did not support the direct use of ATM AAL5.

The S2 information flow is for control information flow from an application service layer source object to a peer destination object. The protocols to support user-to-user interaction are defined in the following figure:

DSM-CC User-to-User
OMG- CDR
OMG- UNO
TCP
IP
AAL5
ATM
Lower layer Protocols

Figure 3-3 S2 Information Flow Protocol Stack

The compatibility with the DAVIC protocol stack for the S2 Information flow is achieved mainly from the following facts:

- the high-level interfaces of the DAVIC Service Elements are based on the DSM-CC User-to-User Interfaces.
- the Orbix product, that implements a CORBA compliant Object Request Broker, is used for the transfer of the control messages between the client entities and the Service objects in the Server.

All the interfaces of the Server Services are written in the OMG Interface Definition Language (IDL) and compiled with the Orbix IDL Compiler. The use of IDL to describe the Service interfaces is adopted by DSM-CC and by the DAVIC Standard. DAVIC has also adopted the following:

- Universal Network Objects (UNO) as the Remote Procedure Call (RPC) mechanism

- Common Data Representation (CDR) as the data encoding for User-to-User messages
- Interoperable Object Reference as the encoding of interoperable addressing an identification of objects
- the IIOP (Internet Inter-Orb Protocol) Protocol Profile as a required Object Reference Encoding

All of the above requirements are supported by the Orbix product (version 2.0.1). The Object References used by the application were Interoperable Object References that complied with the IIOP Protocol Profile. The Object References returned by the server and used by the application were those created by Orbix and did not include association tags of resources (DSM ConnBinders). For example, when a Stream Service was opened, the information about the address from which the client should receive the incoming data stream was not included in the Stream Object Reference. Provided also that the Session establishment phase was not supported in the first trial- where the Network Manager could inform the client for such kind of information- the client discovered the ports for receiving information from configuration files.

Concerning the use of Orbix, although the learning curve to become familiar in developing Orbix applications was quite long, it solved many problems in the integration phase. The performance of transferring the commands was also very good.

4. The KYDONIA Server

4.1 Overview

The DAVIC standard, aiming mainly on the concept of interoperability between subsystems, does not define the internal implementation of a DAVIC server. It only defines functionalities, interfaces and protocol stacks on external reference points of the server. It is outside the scope of the standard to define the way in which data is managed (stored, retrieved) internally in the server. The SICMA server uses the KYDONIA system [CPKMMT97], a multimedia object database management system, for efficient data management in the internal level. The KYDONIA system provides a set of functionalities to the upper level (DAVIC Service Elements) for organising and accessing the multimedia content of applications supported by the server. KYDONIA supports an object-oriented data model, with appropriate extensions for supporting multimedia data types, that is used for the creation and manipulation of structured content (e.g MPEG-2 video objects, stream objects). Moreover, on the storage level, it manages the physical representation of multimedia through efficient placement techniques (striping) and supports multiple concurrent data pumps via scheduling mechanisms. In general, KYDONIA acts as a Multimedia Object & Storage Manager inside the SICMA Server that provides transaction support, retrieval of continuous data streams, creation of multimedia objects (images, videos etc..) and accessing parts of multimedia objects.

4.2 The KYDONIA Agent

Kydonia agent is a module responsible for the communication with the Kydonia server. The kydonia server provides an object-based interface that the DAVIC shell uses to exploit the server's facilities. Once an operation to Kydonia server is wanted a kydonia object must be created. Through that object's methods stream operations or simple data storage and retrieval functions can be triggered in Kydonia server.

The communication between the Kydonia Agent and the Kydonia server follows the datagram architecture. Kydonia object acts as an encoder of the requests that are to be sent in Kydonia

server, which have to be in a special format (marshalling). Additionally when stream playback is requested Kydonia Agent takes over the periodic data block transfer. The kydonia agent reads data from a shared communication pool, follows a double-buffering scheme for each stream and is responsible for the deallocation of appropriate space whenever a data block is transferred to network. This module is also capable of using either TCP-UDP/IP protocols or pure ATM primitives in order to transfer MPEG2-Transport streams over ATM (AAL5).

4.3 The Object Manager

The Object Manager of the KYDONIA Server accepts, through the KYDONIA Agent module, all the incoming requests from the DAVIC Service Elements. It is responsible for the management of objects in main memory level. It maintains an Object Pool, where application and system objects reside. In this area of main memory, there exist descriptor objects, indexes, but not large multimedia objects such as images and video streams. The Object Manager is also responsible for the transaction management inside the server. It implements a concurrency control mechanism and with the support of the Recovery Manager, which is part of the Storage Manager Subsystem, preserves the basic properties of a transaction (ACID properties). The transaction support is mainly needed by the Content Service Element that installs and updates multimedia content inside the server. The interface supported by the Object Manager also includes operations that have to do with the naming of the objects inside the server system. Every KYDONIA object can have a name and it is uniquely identified inside the system by its Object Identifier (Object Id). Object Manager holds in Object Pool indexes (Symbol Tree) that maintain the mapping between the name of an object and its Object Identifier. Applications and end-users access objects and services by browsing through namespaces and calling their names (see Service Gateway operations). There is a need to map the external names to names or identifiers which are meaningful for the data management system and constitute unique identifiers for the multimedia objects and services. So, when a Service Element needs to access an object, whose only knows the name, it invokes the appropriate Object Manager's function to find the Identifier from the name and then, by using the Object Identifier, it can retrieve or update the attributes of the Object.

4.4 The Storage Manager

The Kydonia's storage server is responsible of the permanent storage of data objects and able of handling multiple data streams. The main characteristics of multimedia streams are the high data transfer rate requirements, their continuous nature and the high storage space demands. Thus, the KYDONIA system applies efficient strategies for data placement, appropriate buffering schemes, and scheduling policies in order to manage satisfactory multimedia data streams. The most important module of Storage Manager, in terms of data stream support, is considered the Intermediate Term Scheduler (ITS). ITS is responsible to manage the secondary and tertiary storage accesses in an efficient way in order to improve the information retrieval performance. ITS exploits the parallel access of storage media mechanisms and supports real time request servicing. It has been designed and developed in order to be able to support requests with time constraints and efficiently multiplexing them with non-delay sensitive services. More information about the Storage Manager architecture, as well as a detailed description of the scheduling algorithms can be found in [CPKMMT97].

At this point, we will present all the actions that take place between the Kydonia server and the Stream Service Element of the DAVIC server, when a contiguous data stream is requested.

Let us assume for example that a video playback request arrives in the DAVIC server and that the appropriate Stream Service object is created. Stream object acquires video data from the Kydonia data management system, processes and delivers streams over the network. Figure 4-1: Kydonia's video data pump presents the interaction between the stream object and the

Kydonia system, and also the way that a video data pump is established. Initially the stream object activates an *open_stream* operation that is handled by the Object Manager. During the *open_stream* operation all the necessary preprocessing, for the, requested by name, stream, is performed by the Object Manager. The next action that is carried out by the stream object is the *get_stream*. This action is forwarded again to the Object Manager which submits the corresponding request to the storage management subsystem. The procedure is handled by ITS that is responsible to service the video request ensuring the time constraints and synchronizing the disks where the stream is stored in. ITS controls the preserved double-buffer space in communication pool and decides in which buffer the video data of the currently retrieving block are going to be written on each request time service round. The stream object reads the data from the buffers alternately following the stream's transfer rate. Defense mechanisms are also supported in case problems in synchronization of stream objects and the storage subsystem appear. Finally, the video data is transferred to its destination, through the ATM network. This procedure between the ITS and the stream object, as it is shown in Figure 4-1: Kydonia's video data pump, continues periodically until all the video data is sent and represents the kydonia's video data pump.

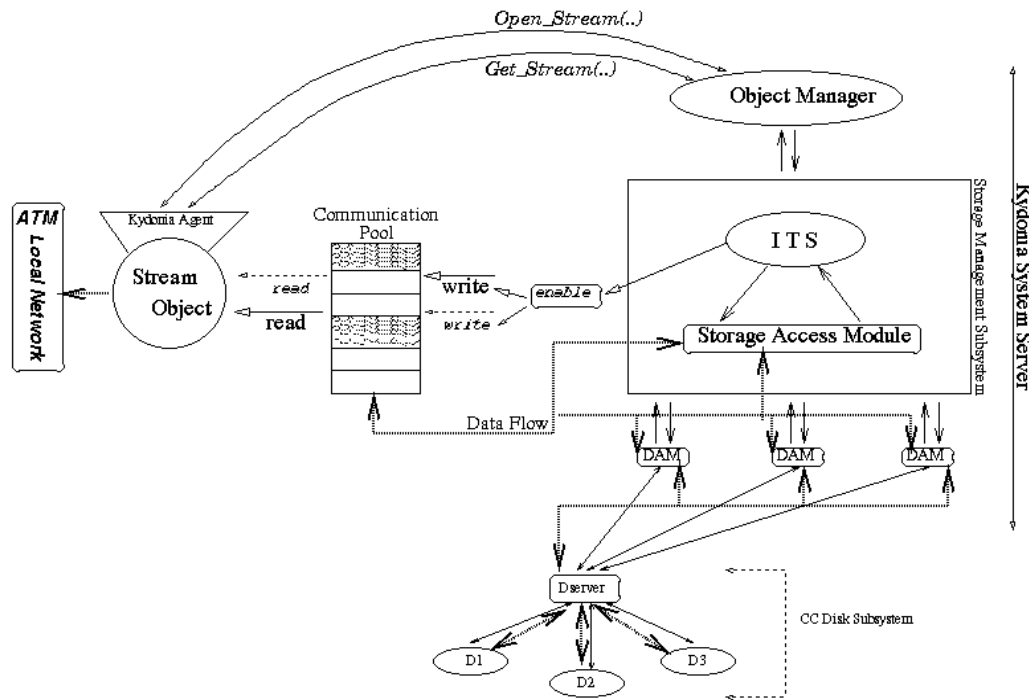


Figure 4-1: Kydonia's video data pump

From the above we can summarize that ITS is able to support and handle multiple video service requests. It succeeds to establish and maintain video data pumps among the storage subsystem and the stream service element. The data flow through the communication pool (that follows a double-buffering scheme) satisfies consumption rate requirements since ITS insures periodic transfer of data blocks, that is controlled by a deadline based scheduling approach. ITS is also capable to support and other operations such as video broadcasting. In that case ITS is responsible to deliver video data to multiple clients through the network using a shared buffer space in communication pool and retrieving only once the popular video stream from the disk subsystem. Operations such as stopping, pausing, or restoring a stream service are also supported by ITS.

5. The D-Server

5.1 Overall concept and functionality

The basic functionality of the D-Server-Process is to store and retrieve data block elements on/from the disk subsystem or another fast storage device. It is responsible for maintaining the disk subsystem that can be configured in a number of partitions as well as a local cache that is implemented on top of the main memory of the processor performing the D-Server. The D-Server is able to receive read- and write - requests with the additional option to read (or write) the requested data into(from)a local cache. It has no own intelligence but is triggered from the higher layers of the server system software, namely the request scheduler (RS). Thus, the process just reacts to incoming requests.

The D-Server is able to handle two types of requests: deadline requests and ABC-Requests. Deadline requests are marked with the point of time they have to be sent to the user. The D-Server Process performs requests in the order of increasing time markers. The data requested by ABC-requests is stored several times on randomly chosen disks. An ABC-Request is issued from the RS-Process to a special ABC-Process. This ABC-Process stores the port number of this request and forwards the sub-requests to the corresponding D-Servers holding the copies of the requested data elements. After the first D-Server replies to these sub-requests by sending an appropriate message to the ABC-Process, a message is submitted from the ABC-Process to one D-Server which fulfills the final request. According to the ABC-Protocol all other sub-requests are canceled. This proceeding minimizes the time the requests have to wait until they are performed [BLR97].

5.2 Processing of requests

The whole SICMA-Server works in rounds and each request from the RS-Process is marked with the round it belongs to. In a given round, the RS-Process issues a number of requests to the D-Servers that have to be processed to serve the different streams within the next time period. The RS- Process sends all this deadline requests and ABC-Requests to the corresponding ABC-Process or directly to the D-Server in the case of a deadline-request.

For each time marker available at the moment in the server, a D-Server stores one list with ABC-Requests and one with Deadline-Requests. During each round, he fulfills the requests in the following order:

1. Urgent Deadline-Requests (e.g. requests of previous rounds that are still not processed by the D-server when requests of the next round already arrived)
2. ABC-Requests of the corresponding round
3. Deadline-Requests of the corresponding round

After the three phases, the D-Server starts with the next round, i.e. it expects to receive requests from the RS-Process or the ABC-Process for the next round. To switch from one round to the next, the D-server does not communicate with other D-servers, the switching is performed asynchronously.

5.3 Delivery of request results

The result of a request is submitted as a Reply-Message to the calling instance. For a request issued directly by the RS-Process to a D-Server, the request carries the port number (RETURN_PORT) of the RS-Process. After completion of the request, the D-Server

acknowledges this by sending a Reply-Message to this port. If a request is issued from the RS to an ABC-Process, the ABC-Process stores the port number of this request and forwards the sub-requests to the corresponding D-Servers holding the copies of the requested data elements. These sub-requests carry the port number of the ABC-Process issuing these sub-requests. After the first D-Server replies to these sub-requests by sending an REQ_READY message to the ABC-Process, a message is submitted from the ABC-Process to one D-Server which fulfills the final request. According to the ABC-Protocol all other sub-requests are canceled. This final message carries the port number of the calling RS-Process so that the D-Server can submit an acknowledge to the overall request. The transport of mass data (video information) is done via shared memory. To do this, the calling RS process requests a shared-memory block on it's processor. For a read requests, this shared-memory block is written, for a write-request, this shared memory block is read by the D-Server.

5.4 Performance Results

The performance of the D-Server is tested in the following environment. One scheduler process generates read requests that are transmitted to the D-Server. The scheduler sends as much requests as possible as long as no backlogs occur. All requests are issued for 10 large files mapped onto the D-Servers. A number p1 D-Server processes are set up. A number p2 of receiving processes are set up to which the data retrieved from the D-Server is send to. All submitted requests are deadline requests.

We measured the data transfer rate from the D-server to the receiving processes in Mbyte/sec. The following values were observed:

Block size	p1=1, p2=1, files are opened/closed after every read operation	p1=1, p2=2, files are opened only once for reading	p1=3, p2=2, files are opened once for reading
20 KByte	1.93 Mbyte/sec	1.98 Mbyte/sec	6.10 Mbyte/sec
50 KByte	2.20	2.30	6.20
100 KByte	3.06	3.08	8.60
200 KByte	3.25	3.27	9.15
500 KByte	3.59	3.64	10.10
1000 KByte	3.68	3.69	10.67

Table 5-1 D- Server Performance Results

This shows that the overall retrieval performance is scaleable when the number of D-Servers and receiving elements is increased.

6. The Remote Access Scenario

One of the main goals of the first trial phase was to demonstrate a Remote Access Scenario, between the systems installed in London and Belgium, over the Pan-European ATM JAMES Network. The idea was to support the interconnection and data communication between the two SICMA servers in order to achieve transfer of multimedia content to a local client from a remote server. This is the first step to identify neseccary expansions in server software for supporting a hierarchy of distributed servers. In such a distributed system, based on factors such as locality/availability of multimedia content and the load of the system, clients could be

served transparently by other servers and not by the one with which they were initially interconnected. Another important issue, was to investigate mechanisms for communication and content transfer between servers. This option could provide the ability for a Service Provider to install multimedia data to servers (cache servers) which are closer to the clients that ask more frequently for this kind of data.

It is necessary that the location of the source for data streams is completely transparent to the client application. So, the code of the application remains the same no matter where the data is coming from. In the Remote Access case, when the client asks for a video stream, the stream is coming from the remote server, through the local server, to the client. Each time, one of the servers acts as a broker and the other as a source for the stream. The main expansions have been made in the implementation of the server's Stream Service Element, where it behaves, in the case of the remote access, as a client for the remote server. The KYDONIA Agent was also modified for retrieving the data stream not only from the local data pump (through shared buffers with the KYDONIA system) but also from the network, when the server is acting as a broker.

The Remote Access scenario was demonstrated to the press on the opening of the trial and took place on certain dates during the trial, whenever access to the JAMES network was available. Through this effort, it became clearer the mode of operation of a distributed DAVIC server, where the services are not all in the same machine, but interconnected via a network. Future work on this task could involve the real time storage of the data streams in the broker server. More specifically, to store the data coming from the remote server and not only forward them to the client (a basic requirement for the Hierarchy of Cache Servers). Moreover, a direct communication between the local client and the remote server could take place, without the interference of the local server. In this case the DSM-CC *Service Transfer* functionality should be used.

In order to assess the bandwidth usage in the remote access scenario, a server-to-server connection between the two museums' servers was made. During the whole session, measurements were performed at the Fore switch port to which the LUC Sun is connected. A sample interval of 30 seconds was used. Of course, this significantly narrows the pattern of the graph shown next. Each of the peaks corresponds to the retrieval of a video sequence on the remote server.

Traffic from LUC Sun during press demo (18/06/97)
(sample interval = 30 seconds)

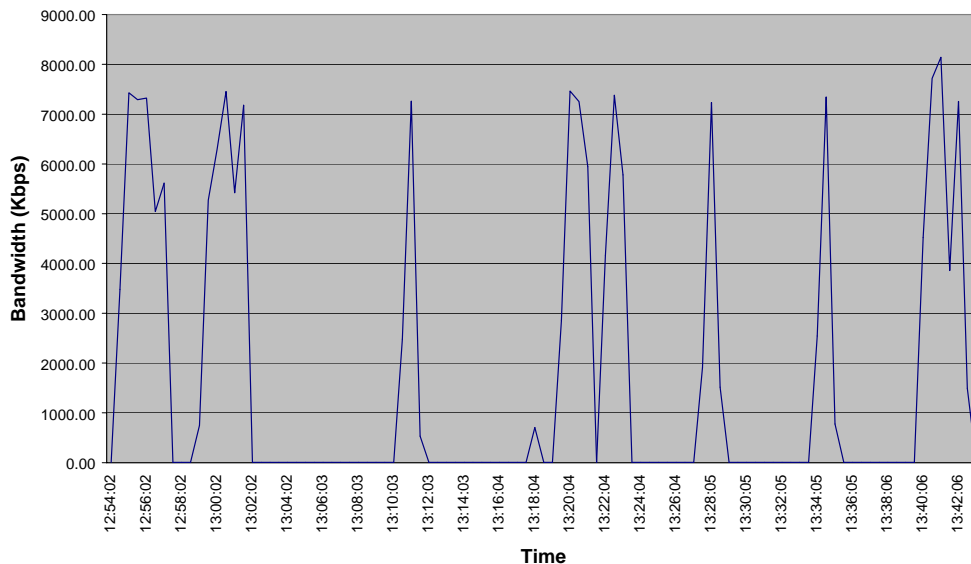


Figure 6-1 Traffic Measurements

In this particular session, the average bandwidth observed during video retrieval was 7.334 Mbps, with peaks up to 8.14 Mbps. All of the retrieved videos were shown without interruptions or drops in the sound or video, which means that the average bandwidth was sufficient to transmit the MPEG-2 transport streams adequately from the server to the client.

During all measurement sessions, an average bandwidth usage around 7.4 Mbps was observed. This is what was expected, given the following parameters:

- the MPEG-2 TS in itself requires 6.384 Mbps (384 Kbps audio + 6 Mbps video)
- segmentation into ATM cells adds 5 bytes for each 48 bytes of payload (for the ATM cell header), which yields 7.049 Mbps
- the IP overhead yields the finally observed bandwidth usage

Of course, due to the buffering techniques used in the transportation software, dips in the network can be compensated. This is also visible in the peak rates observed.

7. The Virtual Museum Application

Among emerging media technologies Virtual Reality is one of the most promising technologies which could lead to new ways on how museums interpretate and present their material to the public. The conceptualisation of a 'virtual gallery application' is therefore based on a 3D environment populated with representations of artefacts with supporting interpretative materials. In context of the SICMA project this means that such materials are stored on the SICMA server. In order to prove the functionality of the server system a special application was developed, the so called virtual museum, which was demonstrated in two museums from June to end of August in the Natural History Museum London and Gallo Romains Museum, Tongeren (Belgium).

7.1 The virtual gallery application

The Natural History Museum holds a significant collection of artefacts that are derived from the first voyage of Captain Cook. This historic voyage culminated in the detailed charting of New Zealand and the East Coast of Australia in 1771. The voyage also had a particular significance, as it was the first to carry on board a group of professional scientists who made detailed astronomical and biological observations supported by detailed collecting and recording. After considering alternative options the concept of creating a digital replica of Cook's ship Endeavour and populating this with digital objects derived from the museums and other collections was adopted as the 'virtual gallery' application.

The exhibition therefore took place in a realtime rendered 3D model of the Endeavour. The digital replica of the ship is based on the best available detailed historical data. The application enables users to explore a continuous, seamless, virtual exhibition defined on the first level by this virtual environment (the ship) inhabited by 2D and 3D rendered objects. The visitors can treat these objects as interfaces to reveal further levels of multimedia data. These include interactive programmes, video clips, animated renderings, digitised 2D and 3D scans of original artefacts ranging from navigation equipment, charts, and specimens collected on the voyage, to the mundane stores of a sailing ship of the period. The objects are supported with interpretative information.

7.2 Description of the virtual exhibition and interaction modes

The exhibition begins with a video sequence, which is downloaded from the SICMA server as all other videos. After this introductory video sequence, the visitor finds himself standing in the Great Cabin of Captain Cook.

Aboard the Endeavour, there are artefacts that the user can interact with. For the purpose of this project, these artefacts have been grouped into three categories; *zoom*, *video* and *3D object mode*. Each artefact has a 'narration hotspot'. This is a zone around the object. When the visitor enters this zone for the first time, an audio clip of a narrator plays, in which the object is described and the user told how they can interact with it. In addition to the real-time virtual reality interface of the Endeavour, the visitors mode of interaction is changing according to the accessed object. Depending on the type of accessed object one of the following interfaces will be presented to the visitor. The content is downloaded from the SICMA server.

7.3 Technical realization

The application is designed to serve as a 3D information browser connected to the multimedia server via a variety of networks. The basic service of the server is to deliver media streams, containing video, audio, images or general data over a network infrastructure. The client systems retrieve all application multimedia data from the server across an ATM network.

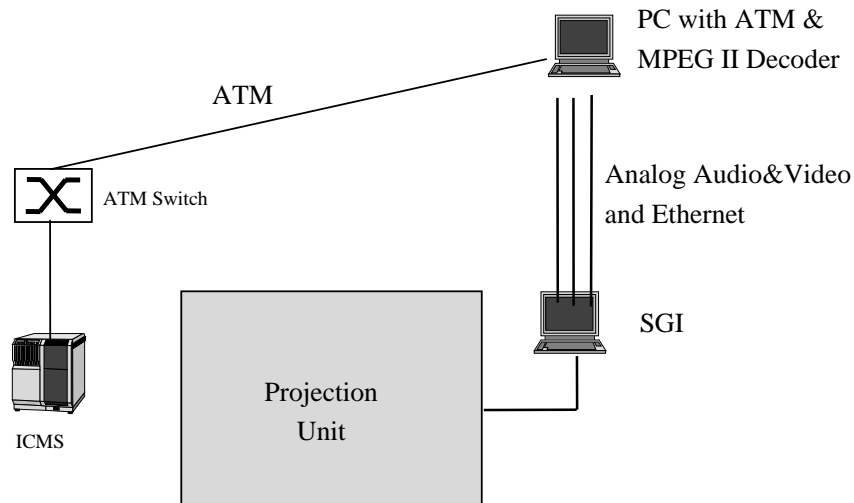


Figure 7-1 Local Setup at each Museum

Video information is stored in MPEG-2 Transport Stream format, audio information as MPEG audio layer 2 streams on the ICMS. All other information is stored in common used formats, for example pictures has been stored as RGB files.

The Natural History Museum and the Gallo Romeins Museum provided high-end SGI rendering systems and PC systems. The PC systems contain an ATM interface and MPEG II decoder board. If a visitor will enter a sensitive zone in the virtual museum, the SGI triggered

a signal to the PC via an ethernet connection. The PC sent a DSM-CC request to the DAVIC-compliant ICMS for the retrieval of the demanded information. If the information is a video or audio stream, the PC forwards this stream to the MPEG II decoder board for decoding. The analogue outputs of the decoder card were plugged into a genlock interface of the SGI machine. If the information are raw data such as 3D objects or pictures, this items are send from the PC to the SGI via ethernet.

During the trial conducted at The Natural History Museum in London the application was displayed on SGI Octane MXI systems in a public gallery. Two duplicate presentation theatres gave visitors access to the 'Virtual Endeavour' application where it was projected onto large, 1.7m x 3m screens. A navigation console was installed in front of each projection. At any given time, navigation through the virtual environment was controlled by one user in each theatre. Up to ten other people were able to view the projected graphics. This provided the opportunity for a semi-immersive environment where group participation was possible. More information about the Virtual Museum application can be found in [BJM97] and [B97].

8. Conclusions and Future Work

The SICMA Server supported the Virtual Museum application, for the three-month trial period, without any performance or stability problems. A high degree of compliancy with the DAVIC standard has been achieved, through the development and use of the DAVIC Service Elements. The integration of technologies and standards adopted by DAVIC (DSM-CC, CORBA, ATM, MPEG-2 Transport Stream) was a complicated and difficult task. The throughput requirements (two clients retrieving videos streams of 6 Mbits/sec) have been succesfully covered via scheduling mechanisms on the storage level. The expansion of the SICMA server is continued, towards the 2nd SICMA trial, where it has to support a video on demand (Distance learning) application, that will be accessed by a large number of students (users). The main efforts are concetrated to support the scalability property in a multiprocessor system as well as the strict throughput demands. The development of new modules, e.g Session Service Element, for full DAVIC compliancy is continued.

The virtual musem application was also very successful. The results of the evaluation were enormous positive. For instance the average time spent by a visitor for one exhibit is between 30 seconds and one minute. In case of the SICMA virtual museum application it was around 20 minutes, that was never expected by the museums experts. It is a strong belief that such kind of exhibitions will have a great impact on the information society, in the near future.

9. References

[BJM97] Brandt, C.; Johnson J.; Meyer K.: "SICMA - A Networked VR Application", The Fourth International Conference on Hypermedia and Interactivity in Museums, Paris 1-5th September 1997.

[B97]Bloomfield, R.: "Remote Access to Museums Exhibits", The 6th International Forum on Leisure and Entertainment Attractions, Straßbourg 24th-26th June 1997.

[BLR97] Berenbrink, P.; Lüling R.; Rottmann V.: "A Simple Distributed Scheduling Policy for Parallel Interactive Continuous Media Servers", Workshop on Parallel Computing and Multimedia, 11th International Parallel Processing Symposium, Geneve1997.

[BPC94] M.M. Buddhikot, G.M. Parulkar, J.R. Cox: "Design of a Large Scale Multimedia Storage Server". In Computer Network and ISDN Systems 27 (1994), pp 503-517.

- [CEA96] Chang S., Eleftheriadis A., Anastasiou D., Development of Columbia's Video on Demand Testbed, Image Communication Journal - Special Issue on Video on Demand and Interactive TV, 1996.
- [CT95] Stavros Christodoulakis and Peter Triantafillou: "Research and Development Issues for Large-Scale Multimedia Information Systems". ACM Computing Surveys - Special Issue on Multimedia Information Systems, December 1995
- [CPKMMT97] Christodoulakis S., Pappas N., G. Kyriakaki, Maragoudakis Y., Mavraganis Y., Tsinaraki C: "The KYDONIA Multimedia Information Server", in Proceedings of Second European Conference on Multimedia Services Applications and Techniques, ECMAST '97, Milan
- [CORBA] The Common Object Request Broker: Architecture and Specification, OMG group
- [DAVa95] Digital Audio-Visual Council, "Service Provider System Architecture and Interfaces", DAVIC 1.0 Specification Part 03, 1995.
- [DAVb95] Digital Audio-Visual Council, "High and Mid Layer Protocols", DAVIC 1.0 Specification Part 07, 1995.
- [DSM-CC] Digital Storage Media Command and Control International Standard. ISO/IEC JTC/SC29/WG11 N1300 p1 MPEG 96/. (Tampere).
- [GVKR95] L. Gemmell, H.M. Vin, D.D. Kandur, P.V. Rangan: Multimedia Storage Servers: A Tutorial and Survey. IEEE Computer, 1995.
- [TJHHJCR94] B. Tierney, W.E. Johnston, H. Herzog, G. Hoo, G. Jin, J. Lee, L.T. Chen, D. Rotem: "Distributed Parallel Data Storage Systems: A Scalable Approach to High Speed Image Servers". In Proc. of the 2nd ACM Multimedia, 1994.
- [V94] H.V. Vin: "Multimedia System Architecture". In Proceedings of the International Symposium on Photonics for Industrial Applications, 1994.
- [VVC96] Michael Vernick, Chitra Venkatramani, Tzi-cker Chiueh: "Adventures in Building the Stony Brook Video Server". In Proceedings of ACM Multimedia 1996, pp 287-295.