The KYDONIA Multimedia Information Server*

Christodoulakis S., Pappas N., Kyriakaki G., Maragoudakis Y., Mavraganis Y., Tsinaraki C.[1]

Multimedia Systems Institute of Crete, Department of Electronic and Computer Engineering, Technical University of Crete, Chania

Abstract. This paper describes the development of a multimedia information server that supports state-of-the-art distributed applications. This is achieved through techniques for efficient storage management of large objects, the provision of real-time data pumps controlled by scheduling mechanisms, multimedia object modelling and ODBMS functionality for the manipulation of objects. The system delivers MPEG-2 video and audio streams to multiple clients and supports browsing on the components of HTML documents and on MPEG-2 video sequences. The server functionality is exported as a proprietary application development interface in a sophisticated client subsystem. Current research focus is on content-based browsing and retrieval techniques as well as for the completion of the implementation of the DAVIC-compliant server functionality of the system.

1. Introduction

The KYDONIA project started in MUSIC/TUC with the aim to investigate design issues and develop technology for the multimedia information systems of the future. 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. The main objectives of the system have been to serve as a testbed for the research and development of multimedia information systems that support distributed multimedia applications, such as movies-on-demand and digital libraries.

The development of the system is based on research in areas such as : storage servers and storage hierarchies, real-time scheduling of disk access requests, transmission of delay-sensitive data, multimedia representation techniques and content-based retrieval and browsing. Several applications have been developed, in order to generate more detailed system requirements and tests, and then have led to specific decisions regarding the system implementation. Another important issue is the integration of state-of-the-art standards, such as MPEG-2 compression for video and audio and HTML for the representation of documents with the provision of adequate functionality for their management.

Current and future development directions are towards a DAVIC-compliant architecture that involves the use of MPEG-2 transport streams for the delivery of

* The project has been financially supported by the ACTS-SICMA and HERMES-ESPRIT Long Term Research, European Community projects.

[1] Authors' address : MUSIC, P.O. Box 133, 73110 Chania, Crete, Greece. E-mail : {stavros, nikos, georgia, imarag, jm, chrisa}@ced.tuc.gr.

delay-sensitive data over ATM, as well as the integration of standards such as DSM-CC and CORBA, that provide interoperability among distributed systems. It is our belief that a database implementation of a DAVIC-compliant multimedia server has significant advantages over an implementation of the DAVIC server functionality on top of a filing system.

The structure of the paper is as follows : section 2 overviews the system architecture and functionality and section 3 reports on techniques that provide for the requirements of multimedia applications. Section 4 summarises the status of the current generation of system, while sections 5 and 6 discuss future directions in terms of research focus and in terms of implementation and compliance with the DAVIC standard.

2. Architecture

The system follows a client-server architecture, the main components of which are (Figure 1) and are described below.

2.1 The Client Subsystem

The *Client Subsystem* has been developed with the aim to support the development and execution of multimedia applications on top of the KYDONIA server. The architecture comprises four modules : the *Communication/Request Manager* (C/RM.), the *Object Virtual Machine* (OVM), the *Object Manager* (OM) and the *Application Development Interface* (ADI).

The Communication/Request Manager is responsible for the construction of request messages that will reach the KYDONIA server via the Line Processing Module, the interpretation and manipulation of server messages and data replies and the control (allocation-deallocation methods) of the Communication Pool, which is used for transferring data to/from the server. In order to handle overflow and starvation in the Communication Pool, when the transfer rate of the delay-sensitive data is greater than the consumption rate, a pause message is sent by the C/RM module to the server. The client continues to consume the available data and when the Communication Pool is about to underflow, a resume message is sent to the server and the normal stream transmission is restarted.

OVM constitutes the kernel of the object-oriented data model and the Object Manager is responsible of accessing and caching objects in the client. It allocates a main memory area named *Object Pool (OP)*, and space on the local disk, called the *Swap Space (SS)*, which is a logical extension of the Object Pool. When main memory space overflows, the Object Manager is responsible of moving a number of objects from the Object Pool to the Swap Space. The objects to be moved are selected on LRU policy. Again, when the Swap Space tends to overflow, objects are moved from to the server storage space and the allocated area is released. The Object Manager operates tightly with OVM in order to identify the structure of objects and convert them to main memory format or secondary storage format. The Object Manager also co-operates with the Communication/Request Manager when objects have to be sent to, or received from the server.

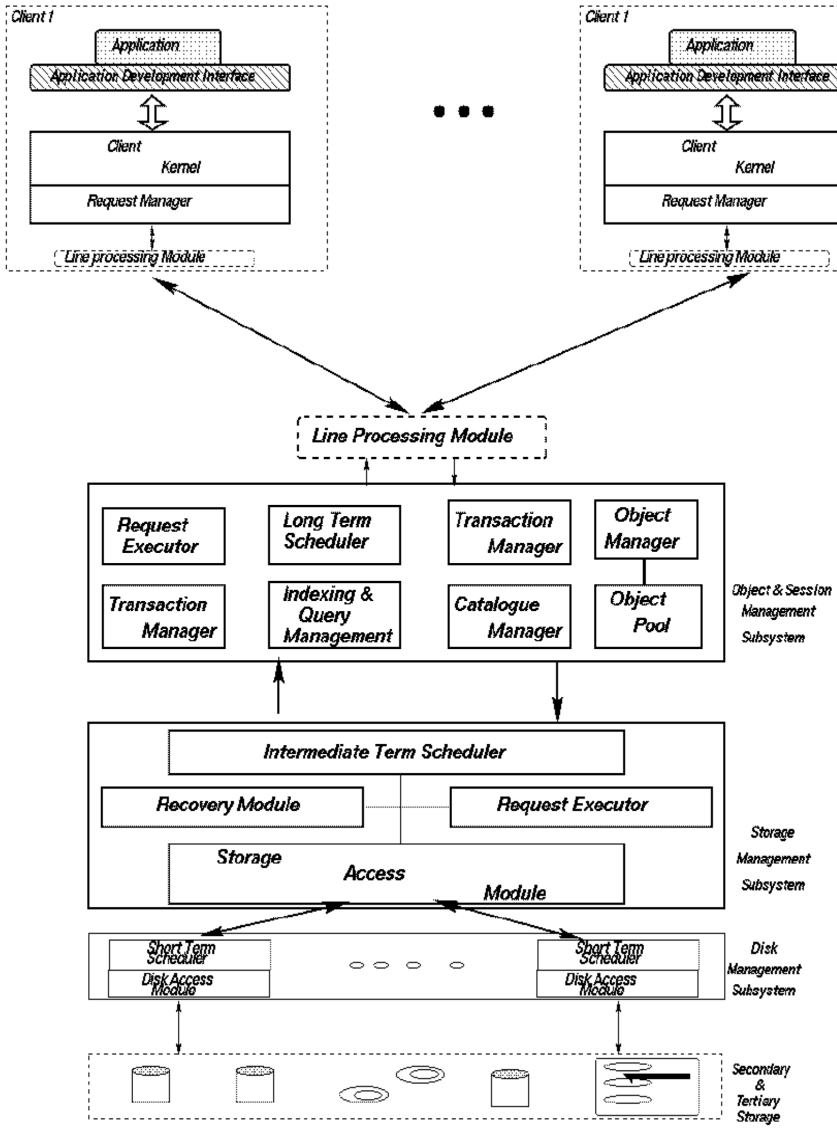

Figure 1. The KYDONIA System Architecture

The Application Development Interface (ADI), is the means for new applications construction and installation. ADI exports to application developers the supported functionality for: object database schema manipulation (e.g. create classes/instances, delete/update instances), session and transaction management, index creation and query submission and browsing on HTML documents and MPEG-2 video streams. The main goal of the ADI library is to provide the application programmer with a transparent view of the system features. Therefore ADI operations embody the capabilities of client caching, message and data exchanging with the server and local object operations (e.g. update).

2.2 The Network and Communication Module

The *Line Processing Module* takes over the distribution of data and control messages over the network. An LPM process runs on each client site and is responsible for the communication between the client and the server. The LPM and client processes have a common shared memory space that is used as for transferring data between them. The client LPM provides the option of establishing either TCP or ATM connections with the server, based on client requirements and capabilities.

A sophisticated version of the Line Processing Module runs on the server site, based on a multi-threaded implementation in order to support multiple clients. A main thread is always listening for new requests for connection and fires a new thread for each client, which is responsible for the communication service of the specific client. TCP connections are used for control messages and ATM connections are used for transferring data.

Current research and development concerns scheduling schemes that should be applied on LPM threads in order to handle the transmission of delay-sensitive streams in a more efficient manner.

2.3 The KYDONIA Server

The KYDONIA server, apart from the server LPM module, consists of two core modules for storage and object and session management, which introduce two separate layers: one for handling all issues of storage management and one for providing a logical view of the data and the operations on them.

2.3.1 Object and Session Management

The Line Processing Module, after receiving client requests from the network, forwards them to the *Object & Session Management Subsystem* (OSM). OSM, in addition to interpreting and scheduling the incoming requests, is also responsible for the provision of functionalities such as session and transaction management, manipulation of main-memory objects, indexing, query, catalogue support. The main components of the subsystem are shown in Figure 1.

The *Long Term Scheduler* performs an initial scheduling of incoming requests, by assigning static priorities which affect the order of execution. It co-operates with other modules (i.e. the Catalogue Module) for collecting information, in order to estimate system load. The module is also being enhanced with more sophisticated admission control algorithms, which will determine the acceptance of delay-sensitive requests.

The *Object Manager* is responsible for providing access to the objects that are located in the main memory of the server (*Object Pool*). The *Transaction Manager* guarantees correct execution of the transactions and co-operates with : the *Concurrency* module, which sets and releases locks on object basis and with the *Recovery* module in the storage subsystem.

The *Session Manager* starts and releases client sessions and maintains information describing the usage of system resources during a session context. The *Request Executor* interprets client requests and forwards them to the appropriate modules. Queries on object attributes are expressed in an SQL-like language and the *Indexing and Query Manager* optimises them to use path indexes during their execution, and proceeds in the optimised plans. The Catalogue module maintains the object database schema of the applications and it also keeps information concerning the placement of objects on disks.

The Object & Session Management module, comprises the front-end of the server to the client subsystem and enhances the overall flexibility of the system by enabling the manipulation of application and system objects internally in the server.

2.3.2 The Storage Machine

The storage machine of the KYDONIA server consists of the *Storage Management* subsystem and the *Disk Management* subsystem. The Storage Machine (SM) is responsible for the storage, retrieval and delivery of objects. In the following subsections we describe all the features and techniques of storage management in the server.

2.3.3 Storage Management Subsystem

The original versions of the Storage Management subsystem, supported an object-oriented management environment for storage hierarchies. Progressively, it has been enriched with capabilities and mechanisms for supporting multiple clients and most recent efforts have concentrated on its transformation to a powerful multimedia storage server.

The organisation of the information within the storage management subsystem is hierarchical and consists of three levels : *Logical Files*, *Physical Files* and *storage devices*. Since this subsystem is the final target of requests for storing and retrieving objects, techniques such as scheduling of secondary and tertiary storage accesses, object clustering, striping across many disks, object duplication, buffering policies and data replacement strategies, have a great impact on performance, are supported by the system.

Data clustering in the logical and the physical layer have also been developed. The logical clustering of structured (or traditional) objects, is based on their *class* and their relative positions in the *class hierarchy*. The application can explicitly specify the logical clustering rules. All objects belonging to a logical cluster are finally stored in a number of physical clusters (Physical File level). Since data in a physical cluster are stored contiguously on disk, the physical cluster is actually the unit of storage. Upon creation, each logical cluster corresponds to one physical cluster. As objects are stored/updated in logical clusters, more physical clusters are created according to rules that have been specified by the system administrator (defaults also exist) and are mapped to the same logical cluster.

The application developer has the capability of configuring the storage subsystem, in order to achieve better results. This is done via a strictly-defined script language, through which the application developer can also specify physical-file characteristics (e.g. mapping of storage devices to a certain physical file), the range of physical cluster sizes, as well as the size and placement of disk partitions. The same script can be used to specify the initial data placement for long objects. Another script language is available for the database administrator to configure and inform the storage manager about the hardware storage devices that are available.

Long objects and operations on them have been modelled in the storage subsystem, in order to support multimedia data with large storage space requirements. The storage server divides video and audio objects into logical blocks while storing them on disks. These blocks can be stored contiguously or scattered in the storage device. Both strategies are supported by permitting the contiguous storage of a multimedia long object on a single disk, or by allowing the combination of object striping and interleaving across multiple storage devices. In Figure 2, the general architecture of the storage management subsystem and its component modules, are shown.

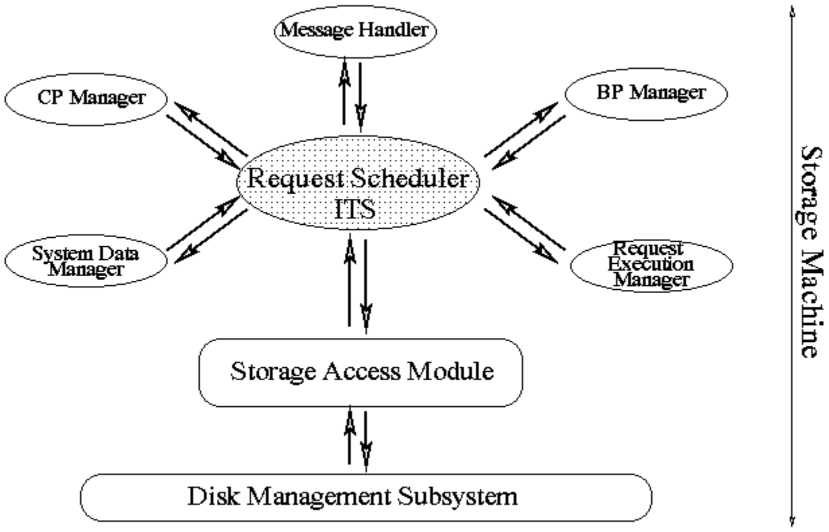

Figure 2. Storage Management Architecture

The *Message Handler* of the storage subsystem is responsible for accepting messages from higher and lower levels in the system hierarchy and for feeding the queues of the request scheduler. The request scheduler in the level of storage management is called *Intermediate Term Scheduler (ITS)* and is part of the three-level scheduling hierarchy that has been developed in the KYDONIA server. Section 3.3 describes the overall scheduling approach of the system in greater detail.

The Intermediate Term Scheduler interacts with, and manages the operation of, the *Request Execution Manager*, the *Buffer Pool Manager*, the *Communication Pool Manager* and the *Storage Access Module*.

The BP manager supports the internal buffering mechanisms while the CP manager supervises allocation and deallocation actions in communication pool (CP holds the data that are going to be transferred from server).

The Storage Access Module *(SAM)*, is the core of the Storage Machine. It is responsible for applying the clustering mechanisms, handling long multimedia objects, supporting recovery and managing devices (bypassing the OS file system by maintaining its own *media access information*). The Storage Access Module also manages all storage media in a transparent way (device level). A generic device model has been developed, that takes into consideration all the characteristics of the different, possibly removable storage devices (optical, magnetic , tertiary storage etc.) and performs a mapping between a general set of operations to storage-medium specific operations.

2.3.4 Disk Management Subsystem

The Disk Management Subsystem is the lowest level in the architecture of the KYDONIA system. Its main goal is to serve the requests that need accesses to secondary or tertiary storage devices. It consists of a number of disk access modules called *DAMs*. Each DAM corresponds to a single system disk drive and is responsible of managing, scheduling and delivering requested data to the storage management subsystem for further processing. The DAMs can be distributed to a number of other computing machines, and thus give greater flexibility to the storage server. An important issue of this approach is that the Storage Management subsystem, together with the Disk Management subsystem can support parallel accesses of storage devices, a feature that is necessary for handling multimedia data.

3. Multimedia Management

3.1 Multimedia Modelling

Raw multimedia values, are usually represented as long, unstructured sequences of bytes. However, multimedia applications need a more sophisticated and flexible representation model, that will provide advanced presentation and manipulation requirements. In the KYDONIA system, multimedia data are modelled as objects and therefore have the characteristics of the object-oriented paradigm. The important features of the KYDONIA multimedia model are that :

- it allows the structuring of raw multimedia data in meaningful entities in a hierarchical or/and hypermedia manner,
- it provides the basis for the representation of multimedia content, i.e. the real-world semantics of the entities represented by multimedia values,
- it supports access methods and browsing primitives,
- it provides extensibility according to application-specific needs.

The task of multimedia data modelling has been divided in three distinct levels : the *logical*, the *interpretation* and the *presentation model* and they correspond to the three degrees of content perusal: structural, semantic and presentation. This approach also reflects the different levels of multimedia objects management that are supported by the system storage of the logical structure, content-based browsing and retrieval and presentation.

The logical structure model is based on the concepts of media *BLOBs (Binary Long Objects)*, *Media Values*, *Elementary objects* and composite multimedia objects or

Documents. Media elements represent entities that are associated with a single-type media value object (such as video only or audio only). Multimedia documents are further classified in *Hierarchical* and *Hypermedia Documents*. Hierarchical documents represent tree-structured composite objects (e.g. movies), while hypermedia documents can have a network-like structure.

The representation of media values that belong to a standard, such as MPEG-2 video, and audio and HTML documents, has been incorporated in the above generic model, in order to exploit their specific characteristics and enhance their manipulation in content-based retrieval techniques.

3.2 Browsing and Content-Based Retrieval

Even though browsing is widely considered an informal presentation of multimedia values, browsing primitives in the KYDONIA system extend this notion and are defined in terms of the multimedia model. They belong to one of the following categories : browsing in the media value, browsing in the logical structure of multimedia objects and browsing in semantic properties.

Media value browsing primitives are defined for every media type, in terms of its physical dimension(s). Video and audio object values are of spatiotemporal and temporal nature respectively, while images have a spatial realisation. Browsing through temporal values is defined as normal and variable-rate (i.e. skipping time content), bi-directional presentation and is supported by access methods for MPEG-2 video and MPEG-2 audio.

Browsing in hierarchical multimedia documents is called *object skeleton presentation* and is defined as a set of navigation paths from the root of a hierarchical composite object, towards the lower levels of the composition . Hierarchical structure browsing is combined with *element expansion,* which is the presentation of the value of the structural elements.

Hypermedia document browsing is defined by means of the following primitives : across documents, across elements of a document, across elements that belong to different documents, across documents and external objects and across document elements and external objects. External objects are neither multimedia documents nor multimedia elements (e.g. conceptual database objects, etc.).

Apart from access methods for MPEG-2 audio and video streams that have been developed in the KYDONIA system and support media value browsing which can be extended to provide more advanced content-based retrieval techniques, searching is supported in text values via the *BO.SI. (Boolean and Similarity)* access method which is a mixture of the traditional signature and inverted file organisations supporting searching in hierarchically structured text objects and ranking of text objects of any kind, anywhere in the hierarchy.

3.3 Scheduling Mechanisms in the KYDONIA Server

The KYDONIA system incorporates several schedulers. The design assumes the existence of a three-level scheduling hierarchy which permits independent operation of each scheduler.

The highest level scheduler is the Long Term Scheduler (LTS) and is embedded in the Object and Session Management subsystem. LTS is responsible for determining the order in which the incoming requests will be submitted to the next level of scheduling. Parameters such as system load, the number of active clients, the network delays, and resource availability are calculated and processed in order to apply admission control algorithms. Currently, only simple and deterministic techniques are used for admission control.

In the middle of the scheduling hierarchy is the Intermediate Term Scheduler. ITS is part of the Storage Management subsystem and its main goal is to manage the secondary and tertiary storage accesses in an efficient way. Thus, in order to improve the performance of retrieval, ITS exploits the parallel access of storage media mechanisms and supports real-time request servicing.

At the bottom level the Short Term Scheduler (STS) takes over the scheduling of accesses of a single storage device (disk). There exist many STSs in the Disk Management subsystem, distributed across the Disk Access Modules. The main responsibilities of each STS are: to specify an efficient order for submitting requests to disks, by applying algorithms that ensure the minimisation of disk-head movements, to schedule disk operations (insert ,eject etc.), and finally, to guarantee that the data retrieval deadlines are met for real-time requests.

ITS may be consider as the heart of the Storage Management subsystem. The reasons that impose the existence of a central common scheduler in the storage subsystem are:

- Accesses to critical common resources, such as buffers and transfer links, should be scheduled by one central scheduler
- A single module should handle the retrieval of data that are striped across multiple disks (e.g. video)
- The synchronization of disks and the optimal use of buffer space when servicing delay-sensitive requests can be guaranteed by a common scheduler for all disks.
- Finally, only a central scheduler can determine whether disks belonging to tertiary storage devices (jukebox) should be on-line or not.

3.3.1 Intermediate Term Scheduling

The main responsibility of ITS is the scheduling of: tertiary storage accesses, batch requests, non delay-sensitive requests and requests with real-time constraints. Figure 3 shows the data model of ITS. Three priority queues, the *waiting queue*, the *active queue* and the *reply queue*, together with two FIFO queues for disk requests, the *disk request & reply queue*, one EDF queue and a dynamic hashing table of blocking requests, are the main structures that support the Intermediate Term Scheduler.

The priority of requests in the waiting queue (that holds incoming requests) and in the active queue (which holds requests that are being served concurrently), is determined by several factors such as the *request type*, the *data type*, *aging*, and *processing state* and is calculated via a weighted heuristic formula. The request with the highest priority is chosen in each round from the active queue, to start or continue execution. When request service cannot further proceed due to lack of resources or if there is a need for disk access, the request is exited from the execution manager process, with a status flag that indicates the current request processing phase. If this flag indicates that access of secondary or tertiary storage will be performed, ITS removes the request from the active queue and places it to the Blocked Request Table, until the necessary data are available in the system's buffer pool. Otherwise, the priority of the request is re-evaluated, depending on the exiting status, and its position in the active queue is updated.

All system queues are periodically processed by the System Data Manager. Activities that take place by this module are: 1. Insertion of waiting requests in the active queue 2. Submission of disk access requests to idle disks 3. Notification of the Blocked Request Handler about recent disk replies and 4. Sending request replies to upper system levels.

3.4 Stream Support: the Video Data Pump

ITS has been designed and developed in order to be capable of supporting requests with time constraints and efficiently multiplexing them with non delay-sensitive services. Thus, ITS takes into account, factors such as *consumption rate*, *disk load*, *buffer space* management and *disk characteristics* (transfer rates, seek delays etc.), in order to schedule requests for the retrieval of multimedia objects.

When a request for video playback is inserted in the active queue, its execution continues until it exits with a status indicating that video is ready for playback. This mean that space for a double-buffering scheme is reserved in the communication pool, the first block of data is already available from disk, and all the necessary indexing information is also available (in order to retrieve the rest of the video stream). During that phase, the video request is removed from the active queue, and inserted in the Deadline Queue, while the deadline for the next service is defined .

The servicing of the video request is then left to the Delay-Sensitive Data Controller. The DSD Controller is activated in discrete time periods, at time interrupts which indicate that the earliest deadline for a request service has expired. When the DSD Controller is active, it serves the first request from the Deadline Queue, sends the appropriate disk access requests and repeats the same process for all requests that have almost the same deadline. If the next deadline is far enough, the DSD Controller sleeps and remains idle until the next interrupt occurs. During these idle time periods ITS serves all other requests that have no real-time constraints. All DSD Controller actions are shown Figure 4.

From the above, we can summarise that, ITS can handle and support multiple video service requests. It succeeds to establish video data pumps between the storage subsystem and the clients through the communication pool, that follows a double-buffering scheme, and by transferring data blocks periodically, controlled by a deadline-based scheduling approach.

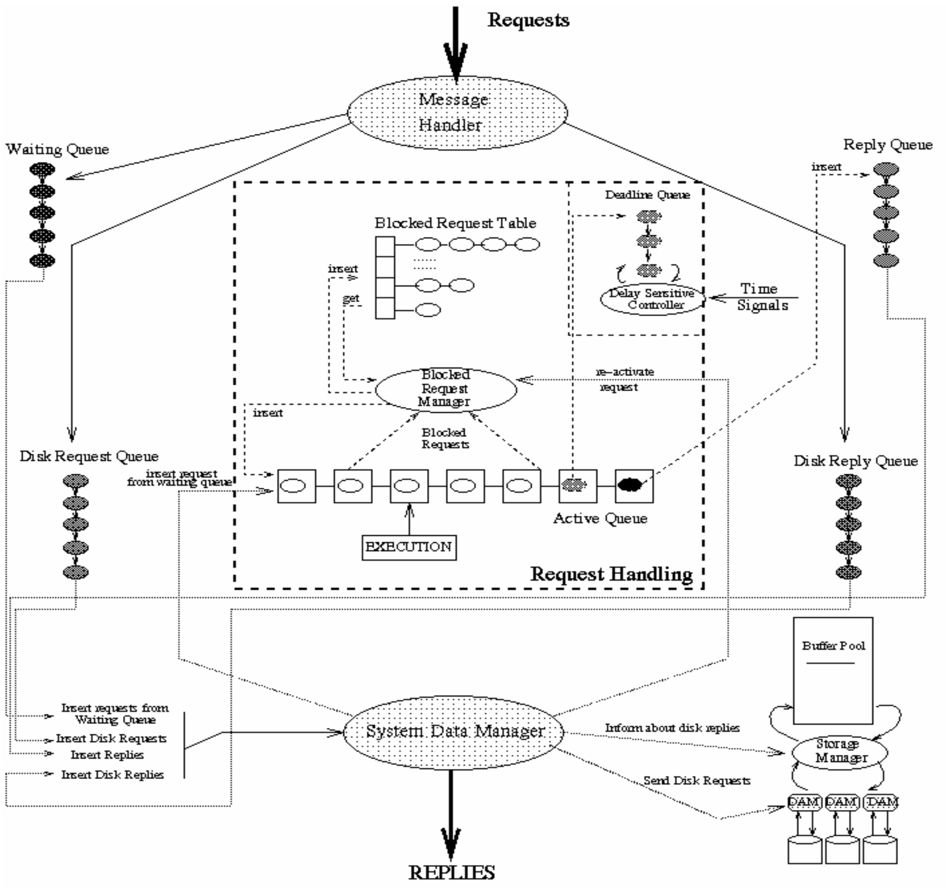

Figure 3: The Intermediate Term Scheduler Data Model

ITS is also capable of supporting other functions, such as video broadcasting. In that case, ITS delivers video data to multiple clients over the network, by using a shared buffer space in the communication pool and by retrieving the popular video stream, only once from the disk subsystem.

Once a video retrieval request is entered in the Deadline Queue, it is being served by the DSD Controller until its execution completes. However, there are cases that the service process should be interrupted from its normal operation. For example, an application user may wish to pause or stop the video presentation, or due to low consumption rates at client sites, the stream retrieval should be paused for a while. Thus, the Storage subsystem should support stop, pause and resume operations. These actions are carried out by the DSD Controller which is responsible to seek and find delay-sensitive requests that are to be paused or stopped and release or reserve the appropriate resources of corresponding requests.

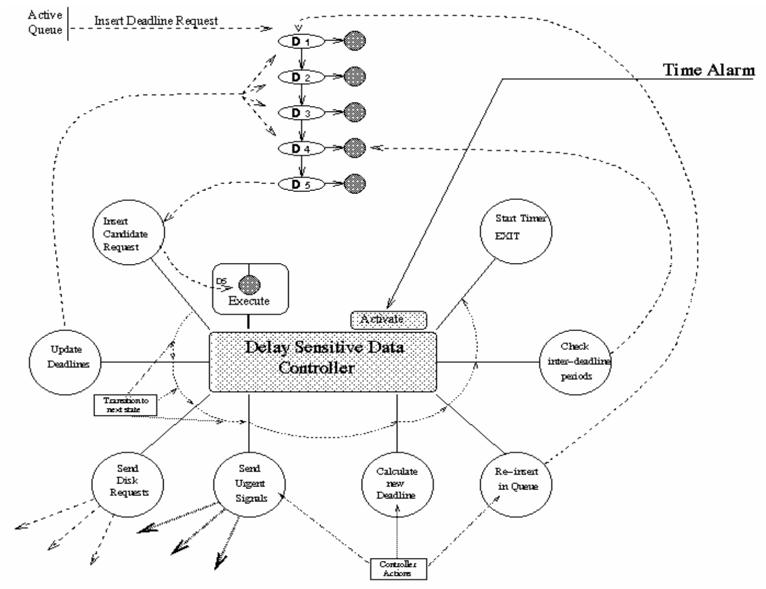

Figure 4. Delay Sensitive Data Controller State Machine

4. Current Status

The system is implemented in C++ language and the total size of the code is well above 200,000 lines. At present, the KYDONIA system runs on a Sun Sparc 20 workstation and SunOS 5.4. Clients and application programs run on several Sun Sparc 4 and on one Sparc 20. The communication network is either Ethernet or ATM.

Several prototypical applications have been developed on top of the multimedia information server. The applications are built with the Application Development Interface (ADI) library that exports the functionality of the system. The majority of these applications intend to test and experiment in the KYDONIA environment.

Hypermedia document presentation and *media presentation*, are two of the most important applications that have already been developed. The hypermedia application manages HTML-document presentation.

Figure 5 shows the graphical interface of a media application for the storage, retrieval and presentation of MPEG-2 video and MPEG-1 audio streams and JPEG images. A video library is available to the user, the contents of which are represented by the miniature of the first frame and a set of descriptive characteristics (title, size, frame rate, etc.) about each video object that has been stored in the server. The user has the capability to store new video streams. The figure shows two clients requesting the retrieval of two video streams concurrently. Video value browsing is available in the form of VCR functionality that is, play, pause, resume, stop, step, jump and play from a frame forward or backward and variable-step forward or backward fast playback, which can be activated by a control window that appears when a specific video clip is selected. An application of virtual reality developed by the University of Paderborn and the Natural History Museum will be soon integrated on top of the server.

Figure 5. User Interface of a KYDONIA Application

5. Research and Experiments

The KYDONIA information server has been the core platform for system oriented research projects of MUSIC/TUC. MUSIC participates in the ESPRIT long term basic research project HERMES which aims to develop and validate the fundamental system-oriented theories for the design of the very large multimedia information systems of the future. The KYDONIA system serves as a testbed for the needs of the project and a series of benchmarking tests are currently taking place on the as one of the activities of the HERMES project. The results of these benchmarking experiments are important in order to improve overall system performance.

The system is also used as a development platform for the ACTS-SICMA project that concerns the development of scaleable interactive continuous media applications and servers, according to a DAVIC-compliant architecture, some general issues of which are described in the next section.

6. Activities towards a DAVIC Server

The DAVIC standard [DAV95], specifies the infrastructure for providing multimedia services from large servers through delivery systems to Set Top Units at home (Figure 6) . Its main goal is to ease interoperability between the distinct components in order “to advance the success of emerging digital audio-visual applications and services”. The overall service architecture is defined in terms of service element entities, functionalities, interfaces, information flows and protocol stacks. Of great importance is the transfer of MPEG-2 transport streams over ATM networks.

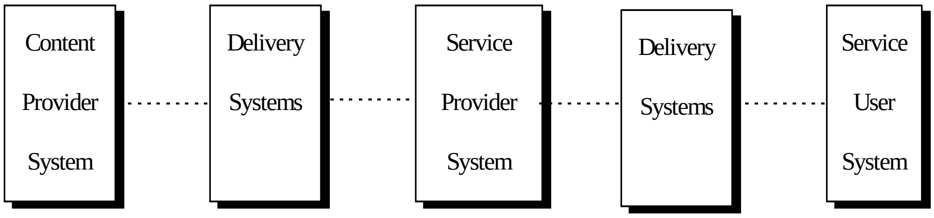

Figure 6. DAVIC System Components

According to the standard, the server follows a distributed object-oriented approach and is comprised of service elements (objects) which provide, through a well-defined interface, a variety of services to support multimedia applications.

We believe that the implementation of the DAVIC server functionality on top of a multimedia database management system like KYDONIA provides significant advantages in the management of the multimedia objects in the server, in the organisation and extension of the functionality provided by the system, in the support for content addressibility, browsing, and associations among multimedia objects, in the implementation of additional supporting mechanisms like security, accounting, queries, etc.

The KYDONIA multimedia server is enhanced with features, such as new interfaces and protocols, in order to become capable of providing services to DAVIC-compliant execution environments. It also aims to incorporate new mechanisms for the efficient manipulation of multimedia data internally in the server and thus manage to fulfil the quality of service (QoS) requirements of the DAVIC standard.

Based on the DAVIC specifications for the Server Architecture, the following core Service elements are being developed in KYDONIA: the *Content Service* that enables the installation of new applications within the server, the *Service Gateway*, which organises the service domain and provides a means to external clients to discover and open services and the *Stream Service*, which is a repository and source for streams and allows a client to control the flow of a media stream.

The interfaces (APIs) of the Service Elements are mainly based on the DSM-CC User to User specifications. The data streams are transferred through the MPEG-2 Transport Stream Protocol and for the transfer of operation calls (RPCs) between the server and the clients the Orbix 2.0.1 product is used.

As we have already mentioned, the aim of DAVIC is to increase interoperability and thus the internal implementation of the server is not defined. It is the Server's responsibility to organise and manage multimedia content in order to meet the requirements stemming from the upper layers of the system (Quality of Service, persistence support etc.). The KYDONIA acts as a multimedia data manager inside the DAVIC server and is used by the DAVIC Service elements for creating, organising, storing and retrieving multimedia content.

7. Conclusions

Multimedia will find important applications in many aspects of everyday life. Digital libraries and video- or movies-on-demand, are such commercially important applications for multimedia since they are beginning to become feasible, due to technological improvements in storage media, networks, standards etc. Their requirements of the above are very high in terms of presentation and interaction, as far as the end-user is concerned, and are translated into very strict quality-of-service requirements for multimedia service providers. The development of powerful servers with capabilities of efficient storage and retrieval, advanced representation techniques and network resources management, is needed for the support of large-scale multimedia application repositories.

The KYDONIA system, as a multimedia information server, attempts to investigate and develop technologies for the implementation of such systems, and is currently being used for the extraction of valuable results, regarding: the overall development process, and the particular algorithms and implementation techniques that are being designed, in a large scale testbed environment.

8. References

[ANA95] Analyti A., Christodoulakis S., "Multimedia Object Modelling and Content-Based Querying", Proceedings of Advanced Course Multimedia databases in Perspective, Netherlands 1995.

[ATM95] The ATM Forum, "Audio-Visual Multimedia Services : Video on Demand", Specification 1.0, December 1995.

[CHR91] Christodoulakis S., Ailamaki A., Fragonikolakis M., Kapetanakis G., Koveos L., "The Design and Performance Studies for a Multimedia Information Server", IEEE Database Engineering Bulletin, September 1991.

[CHR92] Christodoulakis S., Ailamaki A., Anestopoulos D., Argyropoulos S., Fragonikolakis M., Kapetanakis G., Kazasis F., Koveos L., "The OVM WorkPackage - Design Documentation and User Manual", MUSIC Technical Report for the KIWIS P2424 project, Chania, Greece, September 1992.

[CHR93] Christodoulakis S., Anestopoulos D., Argyropoulos S., "Data Organizations and Storage Hierarchies in a Multimedia Server", Proceedings of IEEE COMPCON, 1993.

[CHR96] Christodoulakis S., Tsinaraki C., "Digital Library support in the KYDONIA OODBMS", (submitted for publication), 1996.

[DAV95] Digital Audio-Visual Council, "Description of DAVIC Functionalities", DAVIC 1.0 Specification Part 1, 1995.

[GEM90] Gemmel J., "Principles of storage and retrieval of delay-sensitive multi-media data", M.Sc. Thesis, University of Waterloo Canada, 1990.

[ISO93] ISO/IEC, "Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbps Part 1: Systems", ISO/IEC 11172-1, 1993.

[ISO94] ISO/IEC, "Generic Coding of Moving Pictures and Associated Audio: Systems", (MPEG-2 Systems Specification), ISO/IEC 13818-1, November 1994.

[KIM89] Kim W., "A Model of Queries for Object-Oriented Databases", Proceedings of the 15th International Conference on VLDB, Amsterdam 1989.

[KIM90] Kim W., “Introduction to Object-Oriented Databases”, The MIT Press, 1990

[KYR95] Kyriakaki G., “Query processing in an Object-Oriented Database”, Diploma Thesis, Dept of Electronics and Computer Engineering, Technical University of Crete, 1995.

[MAV95] Mavraganis Y., “Design and Implementation of the Object and Session Management Subsystem in the Object Oriented Database Management System KYDONIA server”, Dipl. Thesis, Dept of Electronics and Computer Engineering, TUC, 1995.

[MAR96] Maragoudakis Y., “Client subsystem and client-server communication in the OODBMS KYDONIA”, Dipl. Thesis, Dept of Electronics and Computer Engineering, TUC, 1996.

[PAP95] Pappas N., “Design and Implementation of the Intermediate and Short Term Scheduler of the KYDONIA server”, Dipl. Thesis, Dept of Electronics and Computer Engineering, TUC, 1995

[TSI95] Tsinaraki C., “Design and Implementation of a uniform and transparent system for document and link representation in the KYDONIA Object Oriented Database Management System and the World Wide Web”, Dipl. Thesis, Dept of Electronics and Computer Engineering, TUC, 1995