Multimedia Art Directory — a multimedia application for an ATM network*

Álvaro Almeida[1], Paulo Proença[1,2] and Manuel Ricardo[1]

1. INESC
Praça da República, 93 R/C
4000 Porto, Portugal
tel: +351-2-2094200 fax: +351-2-2084172
e_mail: {aalmeida, pproenca, mricardo}@inescn.pt
2. ISEP
Rua de S. Tomé
4200 Porto, Portugal

Abstract

This paper presents a Multimedia Art Directory, that has been developed for an ATM network, by describing its main characteristics and the creation environment and the network facilities used.

1 Introduction

In this paper a multimedia service that has been deployed in an ATM network is described.

MAD, the Multimedia Art Directory, appeared as a test case in the project RACE 2017 SCORE – Service Creation in an Object Oriented Reuse Environment, used to evaluate its results that are, basically, a set of methods and tools for the efficient provision of Integrated Broadband Communication services.

In order to demonstrate these results, MAD was initially required as a service that should be realistic, multimedia, use an ATM network, have real end–users and should not require too much man–power for its development.

To satisfy some of these requirements, an agreement was reached with another project, RACE 2068 LACE – Local Applied Customer Environment for Integrated Communications, which was developing an ATM demonstrator that would be placed in the Ashmolean Museum, in Oxford. With this collaboration, the ATM network and the real end–users were gained. To

*This work has been partially supported by the RACE 2017 (SCORE) project.

satisfy the non too much man–power consuming and realistic service requirements, a decision to built a service that should be strongly based on already existing software was also made: the X.500 implementation of the ISO–DE consortium [1] was so adopted.

The MAD most valuable achievement, from a user point of view, is the ability to allow a user to navigate through a directory of museums, that contain works of art, query the directory by some attributes and, when the desired work of art is found, the ability to show the relevant information about it, presented in text, high quality images and video sequences.

In section 2, the service is described by introducing the X.500 directory, the work carried out to form the art directory and the architecture of the multimedia directory over the ATM network.

In section 3, the MAD creation environment, that is a specialisation of the SCORE generic Service Creation Environment, is presented. The creation process model, the methods and the tools used, as well as its main characteristics, are introduced. The transport services offered by the target network, the LACE demonstrator, and its usage are also presented.

Finally, in section 4, some conclusions are drawn about the creation activity as well as the functionality and usability of MAD.

2 Service description

The Multimedia Art Directory is a service that provides the access to art information stored in a database distributed across a network. By combining the Directory service specified by CCITT [2] with multimedia technology, objects of art, in MAD are described by text, pictures and video.

2.1 The X.500 directory

The X.500 directory or, as it is known, the Directory, is generally used as an electronic telephone book, which allows users to query about names and addresses of, for instance, persons and organisations.

The information of the Directory is kept on a tree like structure whose hierarchy is based on the natural fact that persons work in departments of organisations that, in turn, are grouped in countries, as described in Fig. 1.

The Directory, that is composed by servers distributed across the network and called Directory System Agents (DSAs), is accessed by users through a Directory User Agent (DUA) process, as described in Fig. 2.

A users' DUA is connected to a server DSA. When the information required is not stored on that particular server, a contact with the server that holds the information is performed transparently. From the user point of view, there is just one Directory to which he connects himself and which provides him with the information.

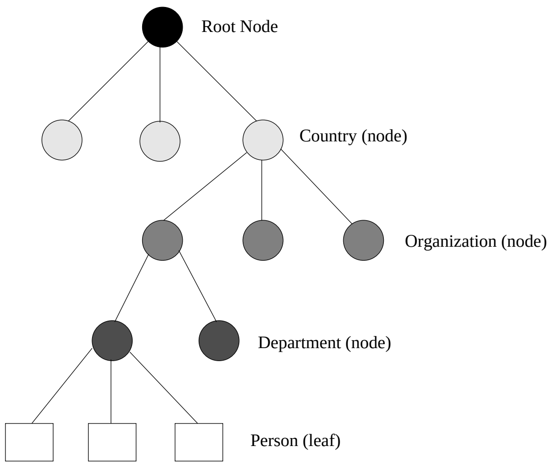

Figure 1: The Directory Information Tree

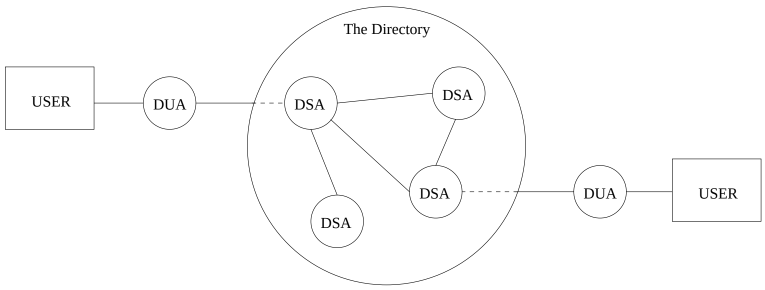

Figure 2: The Directory Functional Model

2.2 The art directory

The X.500 Directory service defines a set of classes for each object type, being the attributes of a class used to characterise the object.

In order to accommodate objects of art into the Directory, a new set of classes was created and the Directory Information Tree was adapted to hold the new objects.

With the help of an art expert from the Ashmolean Museum, the attributes that uniquely identify an object of art were identified and the internal organisation of a museum was learned.

A museum was then modelled, for this service, by *Art Organisation*, *Art Departments*, *Art Sections* and *Art Sub–Sections* classes.

Works of art were characterised by a *Work Of Art* class with the following attributes: *Work of Art Name*, *Artist Name*, *Artist Birth Date*, *Artist Death Date*, *Artist Birth Location*, *Artist Nationality*, *Origin* (where and when the work of art was created), *Technique* and *Dimensions*.

On a request of a work of art, the user is presented, as shown in Fig. 3, with information about the work of art itself, the artist which has created it and a video sequence that, for instance, can represent the work of art from various points of view.

By specifying a value–attribute pair, a list of objects that match the specification under a node of the tree, which is described as a path, is presented to the user. The objects found can be either a work of art or any other object such as an art section.

With this service, museums have a new and expeditious form to reach the public in general and art galleries are offered a new way to advertise their products. Potential users of this service can then be art students, art professors, art researchers, art investors and public in general.

2.3 Service architecture

The MAD architecture is shown in Fig. 4. The ISO–DE implementation [1], represented in Fig. 4 as the *text/image client* and the *text/image server* blocks, of the X.500 protocol is used for the transmission of the text and image types of information. The TCP/IP (Transmission Control Protocol/Internet Protocol) are the communication services used by these blocks.

The *video client* and *video server* blocks are used for the transmission of MPEG video sequences [11] according to a video transmission protocol that is, in a simplified form, presented in Fig. 5. The TPI/AALPI (Transport Provider Interface/AAL Provider Interface) are the LACE network ATM services used by these blocks and described in section 3.3.

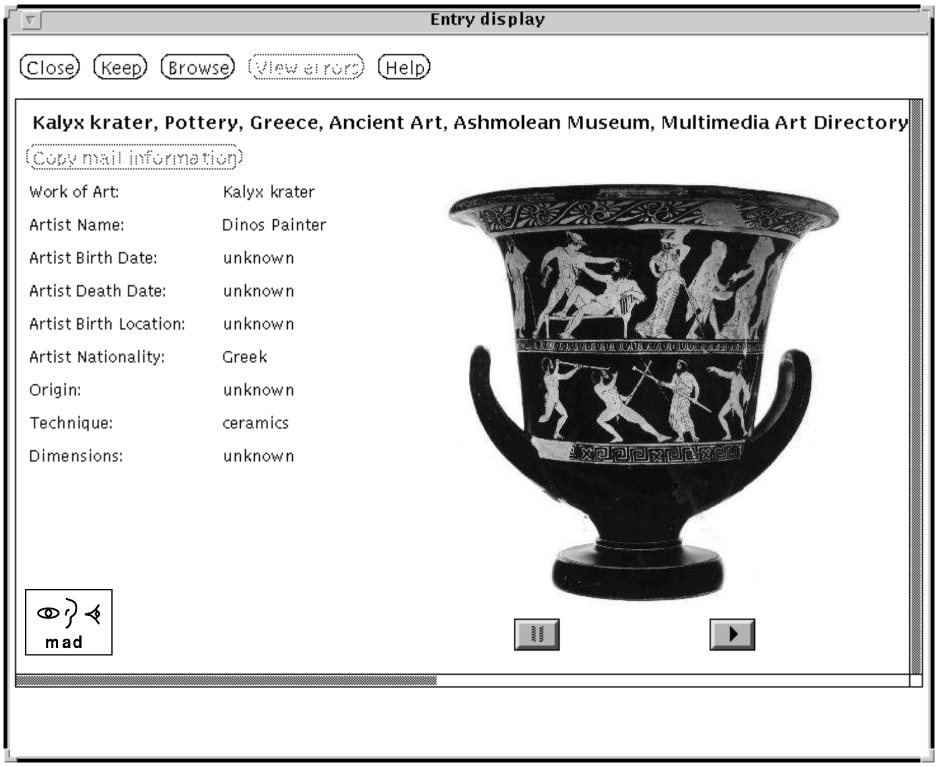

Figure 3: An Object of Art

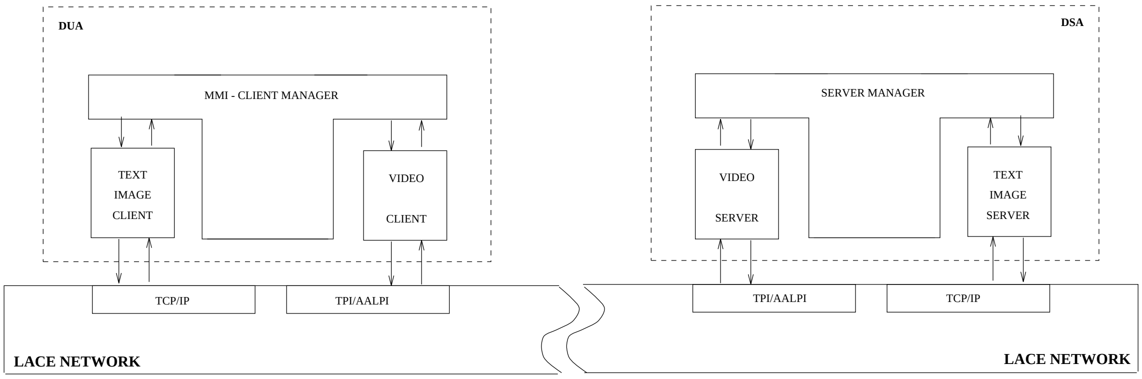

Figure 4: MAD architectural scheme

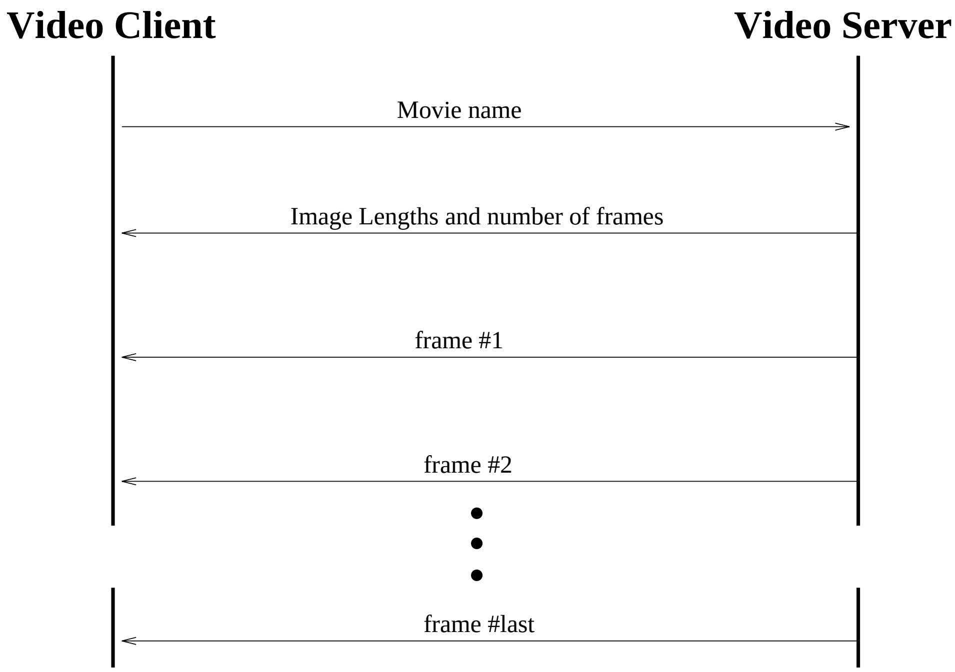

Figure 5: Simplified video transmission protocol

3 Service creation environment

MAD was developed under the project RACE 2017 SCORE – Service Creation in an Object-oriented Reuse Environment, in cooperation with the project RACE 2068 LACE – Local Applied Customer Environment for Integrated Communications.

The main objective was to evaluate the applicability of the methods and tools recommended by the SCORE project in the context of a real multimedia service for an ATM network.

3.1 Model of the creation process

MAD has followed the SCORE process model [3] which defines the following activities: Service Analysis, Service Specification, Service Design, Building Block Specification, Building Block Design, Building Block Coding, Service Implementation, Test and Deployment.

During Service Analysis, the differents actors involved in MAD and the roles they play were identified: public in general as the Service Subscriber/End-User, INESC organisation as the Service Provider, Ashmolean Museum as the Data Provider, SARI (INESC group) SCORE team as the Service Developer and the LACE project as the Network Provider. Service needs, based on the actors requirements and expectations, were also identified. As the

SCORE project emphasises reuse, some components, with potential for that, were also identified. Finally, a first description of the service was produced using an Object–Oriented technique.

During Service Specification the service was specified in an implementation independent way using a formal description technique.

Based on the information gathered during the Service Analysis and the Service Specification, the service was designed in the next activity. The architecture was produced, building blocks were identified and the communication protocols between objects were designed.

After that, new components were specified, designed and coded. Once created and tested, the new components were integrated with the reused ones in order to compose the service. After several tests, the service was deployed in the LACE network.

3.2 Methods adopted

Object–oriented methods, characterised by encapsulation, inheritance and polymorphism of the information, were chosen for the creation of MAD. The two reasons behind this choice were that OO methods *i)* were selected in SCORE for service creation; *ii)* are known as providing more correct, robust, modular and reusable systems than the non object–oriented methods.

The methods and languages selected were: the Object–oriented Modelling Technique (OMT) [6], the Specification and Description Language (SDL) [7] and C++ [8].

3.2.1 General description

OMT is a methodology, with a graphical notation associated for the object–oriented analysis, specification and design of software. In order to represent the system, OMT uses three types of models: the *object model*, that describes the objects and their associations, the *dynamic model*, that describes the behaviour of the objects, and the *functional model*, that describes the data flow. Each model is supposed to give a different but complementary view of the system under construction.

SDL is a language recommended by the International Telecommunication Union (ITU–T) for the specification and description of telecommunications systems such as data protocols, services and control systems. It has concepts to describe behaviour, model data and structure systems. Behaviour is modelled by means of extended finite state machines that communicate by mean of messages. Data are described by means of algebraic data types. Structure is achieved by means of hierarchical decomposition and types hierarchies. In this way, a system can be seen as a set of blocks connected by channels. Each block has processes (state machines) that communicate by signals (messages) and are concurrent.

C++ is an evolution of the C programming language [9] that with the additional features of *i)* support for creating and using data abstractions; *ii)* support for object–oriented design and programming; *iii)* refinement of existing C constructs.

3.2.2 Usage in MAD

OMT was used in MAD for the Analysis. The object model, clearly the most important of the three, has enabled the identification of some important objects and their attributes. Associations between them as well as aggregation and specialisation relations have also been identified. With the functional model, the kinds of data involved in the service and their transformations were identified. The dynamic model was not used.

SDL was used for the Specification and Design of MAD. It means that, after the OMT analysis, the most important objects were transformed into SDL processes and SDL structured data types. Although SCORE suggests clearly the kind of SDL objects that OMT objects can be transformed in, in MAD these rules were not followed. Instead, a new SDL service was specified, from scratch, but with a constraint — the person in charge of the SDL specification had to be the same that would perform the OMT analysis.

The main reason behind this decision was the lack, in that time, of a automatic OMT–SDL translation tool. Being the specification carried out by hand, the best approach was to specify by refining the OMT description at the same time. It meant that no clear mappings exist between the last OMT description and the first SDL specification, since some "mental" and invisible iterations occurred in between.

SDL was used mainly to describe the protocols between the active objects, modelled as concurrent and distributed processes. Geode [10] allowed the graphical edition, syntactic and semantic verification, as well as the simulation of MAD. Automatic code generation for C language for several platforms is also provided by the tool.

C++ was the language used for the implementation. Again, due to the lack of an automatic SDL — C++ translation tool, to the interest in follow the object–oriented approach and to the curiosity of understanding the SDL — C++ transformations, an hand C++ implementation was performed for the actual target network. Internal operations, such as data decoding and human interfaces, were also developed in C++.

3.3 Target network

The terminal equipment of the LACE network consists of Unix SVR4 SUN and IBM workstations, providing the AAL5 ATM service as a Transport Layer Interface (TLI) like interface.

The Transport Layer Interface is, as the name suggests, the service interface provided by the OSI Transport layer. The four services primitives

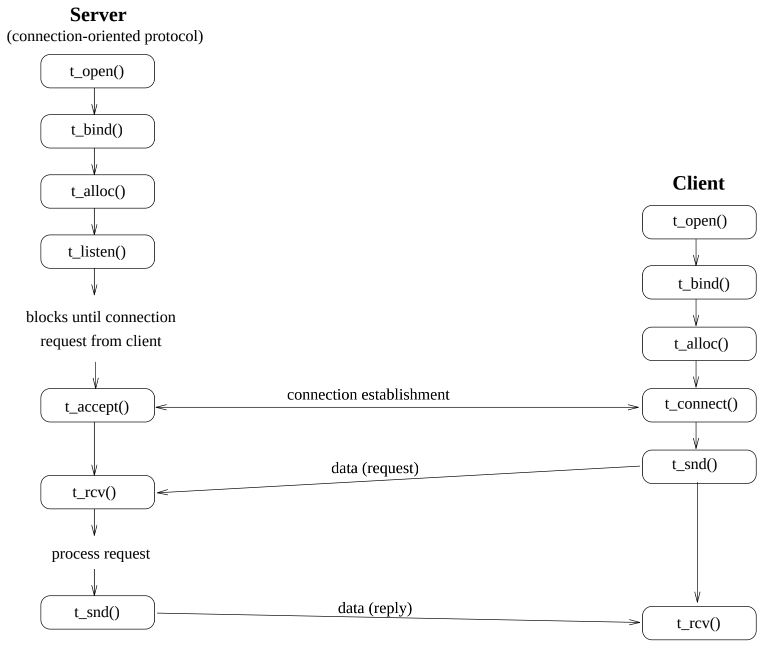

Figure 6: TLI function calls - connection oriented protocol

defined by TLI are: *request*, *confirmation*, *response* and *indication*.

At the initiator (client) side *request* is sent by the user to the provider to order a service. The result of the service is returned by the provider to the user through a *confirmation* primitive. At the responder (server) side the *indication* and *response* primitives are used respectively for the indication of a request, from the provider to the user, and to provide the result of the service, from the user to the provider.

TLI provides two types of service: *connection* and *connectionless*.

Connection mode is circuit oriented. Data are transmitted in sequence over an established connection. An identification procedure is also provided to avoid address resolution during the data transfer phase. This kind of service is used by applications that require datastream–oriented interactions, like MAD.

Connectionless mode is message oriented. Data are transferred in self–contained units with no relationship between them. This service requires only an established association between the peer users that determines the characteristics of the data. All the information required to deliver a message, such as the destination address, is offerred to the transport provider together with the data to be transmitted, in a service request.

Fig. 6 shows a typical call sequence of the functions provided by LACE in its ATM network library.

In the case of MAD, the *video client* waits for a stream of video frames by binding to the ATM end-point and then blocking on the *t_listen()* call. Once the *video server* is ready to send the data, it opens an ATM end-point and attemps to connect to the client.

4 Conclusions

In this paper MAD has been presented. As a catalogue, it can be used by museums to present their works of art and by art galleries to advertise their products. Being built based on the CCITT X.500, the catalogue can be openly placed and accessed all over the world. Being multimedia, it enables users not to just have the basic information about the works of art and museums, but also to see the pictures and the videos associated with them.

The methods recommended by SCORE were found useful for the creation of multimedia services for broadband networks: *i)* the OMT object model was valuable to think and structure the service in terms of objects; *ii)* SDL was appropriated to describe the interactions between the objects; *iii)* C++ was comfortable to describe algorithms, access network services and build other libraries.

The major drawback found was the inability of evaluating an instance of the service generated automatically, due to lack of OMT — SDL and SDL — C++ automatic translation tools. The difficulty in guaranteeing the quality of the services, mainly for video, was also found.

In the continuation of MAD, future work will include: *i)* refinement of the creation methods to include Quality of Service requirements; *ii)* usage of distributed platforms; *iii)* integration of sound and MHEG; *iv)* construction of a gateway to the World Wide Web (WWW) internet service.

Acknowledgements

The authors would like to express their gratitude, first, to Prof. Eurico Carrapatoso who is their supervisor and has provided them with all the conditions to carry out this work, to Mrs. M. M. Marques for her trustness and strong support within SCORE, to Dr. Kurtz for her art support and, finally, to Mr. J. Soares for his help in the understanding of the LACE network.

References

[1] C. J. Robbins and S. E. Kille, *The ISO Development Environment: Users's Manual*, Vol V, QUIPU.

[2] CCITT, *The Directory - Overview of concepts, models and services*, Recommendation X.500, 1992.

[3] SCORE WP1, *The SCORE Service Creation Process Model*, Deliverable D104, Volume I – R2017/SCO/WP 1/DS/P/016/b2, RACE project 2017 (SCORE), December 1993.

[4] SCORE WP1, *The SCORE Component Model (Version 2)*, Deliverable D104 Volume II – R2017/SCO/WP1/ DS/P/016/b1, RACE project 2017 (SCORE), October 1993.

[5] SCORE – METHODS AND TOOLS, *Report on Methods and Tools for Service Creation (First Version)*, Part V:Targeting, Deliverable D2035 – R2017/SCO/WP2/DS/P/027/b2, RACE project 2017 (SCORE), January 1994.

[6] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorensen, *Object-oriented Modeling and Design*, Prentice-Hall, 1991.

[7] CCITTT, *SDL Specification and Description Language*, Recommendation Z.100, 1993.

[8] Stanley B. Lippman, *C++ Primer*, Addison Wesley, 2 edition, 1991.

[9] Brian W. Kernighan and Dennis M. Rithcie, *The C programming language*, Prentice Hall, 2 edition, 1988.

[10] Vincent Encontre, *GEODE: An industrial environment for Designing Real Time Distributed Systems in SDL*, pp. 105-115 in Proceedings of SDL'89, The Language at Work, eds. Ove Faergemand and Maria Manuela Marques North–Holland, 1989.

[11] ISO/IEC JTC 1/SC 29/WG 11, *ISO/IEC CD 13818–2: Information technology—Generic coding of moving pictures and associated audio information–Part 2: Video"*, December 1993.