

A MULTIMEDIA DATABASE FOR AN ADVANCED TEleshopping APPLICATION

D. Maino¹, F. Mercalli¹, R. Negrini²

1 - Centro di Cultura Scientifica "Alessandro Volta" - Villa Olmo, Como - Italy

2 - Politecnico di Milano - p.za Leonardo Da Vinci 32, Milano - Italy

ABSTRACT

This paper illustrates some issues arising when designing an advanced teleshopping system based on a multimedia database. In particular, it is focused on a conceptual model (MMAIN) able to support the multimedia database. The studies here reported are part of the RACE Project ESSAI, whose goal is to implement by the end of 1994 a prototype teleshopping system based on advanced multimedia technologies. A large multimedia catalogue of products to be sold is to be implemented as part of the system. The high quality of multimedia data and the high flexibility of the data model required by such a system make the ESSAI multimedia database a rather advanced prototype.

1. INTRODUCTION

Multimedia technology opens the road to completely new ways to conceive computer applications. As a consequence, completely new ways of defining and manipulating data managed by those applications are needed.

The ESSAI Project, sponsored by the Commission of the European Communities under the RACE II Programme, is not an exception: it aims at implementing a teleshopping system prototype - made possible by the development of new hardware technologies - whose data management requirements are not adequately covered by traditional database management techniques.

For this reason, research activities have been started in the frame of the project in order to first define the peculiar requirements, in terms of data management, of this kind of application and secondly to design a feasible solution.

Various databases are present in ESSAI; in this paper we will concentrate on the most innovative one, i.e., the multimedia database containing information about products to be sold through the system.

At present, the requirement analysis has been completed and a conceptual model for the data has been sketched.

The goal of this paper is to outline these main conclusions, on which the implementation of the ESSAI prototype will be based.

In section 2 an overview of the whole ESSAI system is given, in order to better focus the problem. In section 3 the basic requirements for multimedia data management are listed. In section 4 a conceptual model capable to support the ESSAI multimedia database, called MMAIN, is discussed. In section 5 a logical architecture for a system supporting MMAIN is presented. In section 6 some conclusion remarks are reported. In Appendix A a summary of MMAIN concepts is presented.

2. THE ESSAI SYSTEM

The aim of the RACE II Project ESSAI is to implement by the end of 1994 a prototype teleshopping application based on an Integrated Broadband Communication network (IBC network).

The fundamental idea underlying the ESSAI Project is to show that, as the communication networks will offer more and more bandwidth at lower prices, the implementation of a teleshopping system based on many low cost *vending terminals* connected to a unique *control center*, in which almost all the processing power needed by the system is located, becomes achievable.

This will allow the installation of many terminals in a large number of sites where the setting up of a traditional shop is not practicable, greatly extending the market of the companies that will exploit the system.

The vending terminal should offer to the end-user a nice multimedia interface, in order to persuade her/him to buy the offered products.

If all the computing power is to be located in the control center, a large amount of information of disparate kinds should flow between the control center and the vending terminals: this justifies the need for an IBC network.

The overall architecture of the system, stemming from these preliminary considerations, is illustrated in figure 1.

Three main actor types are involved in the architecture of figure 1: the *service provider*, which is the company that

will offer and manage the teleshopping system, the *good suppliers*, which are the companies that will exploit the teleshopping service in order to sell goods to their customers, and the *end-users*, which are the customers that buy goods through the vending terminals.

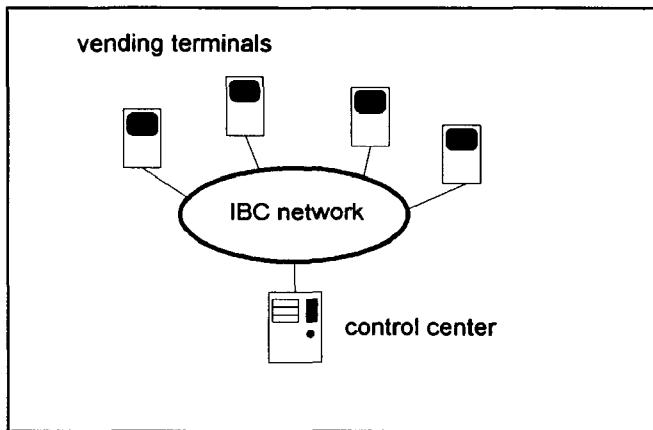


Figure 1. ESSAI general architecture.

The vending terminal is the most important component in determining the success of the system on the end-user side. It is important to design a man-machine interface able to *attract* end-users. This means not only that a multimedia user interface is needed, but also that the quality of media data should be as high as possible.

A high resolution graphic display and a hi-fi reproduction system will be used for the output of data. For data input, it has been decided to make extensive use of a touch screen device, since it represents the most intuitive form of data input available on the market.

Besides the multimedia user interface above sketched, the terminal should feature a payment system, based on some sort of electronic payment form, such as a credit or debit card, in order to make it possible for the end-user to pay the goods she/he chooses.

This involves two more types of actor, a *payment authorization center* and a *bank*, that will interact with the teleshopping system through the service provider.

For what concerns goods delivery, immaterial goods, such as for instance cinema movies, may be directly delivered by the vending terminal through ticket printing, while material goods, as for instance a photo camera, may be indirectly delivered by mail; in this later case the terminal will hand out to the end-user a voucher, proving the purchase of the good.

In the control center three main components are present: the *payment system*, the *availability database* and the *catalogue database*.

The payment system is in charge to obtain from the terminal the payment data relative to the end-user, for instance data read from the credit card, and to ask the necessary payment authorization.

The availability database is the database containing information concerning the availability of the goods to be sold.

This is substantially derived from the stock databases of the good suppliers.

The catalogue database is the system in charge to manage all information about the products offered by good suppliers; it is, in a sense, the ESSAI electronic form of the usual paper catalogues that a supplier prepares for his customers. The catalogue database should show in the most comprehensive way the supplier offer.

The catalogue database is the most innovative kind of database present in the ESSAI teleshopping system since it must store and control a large amount of multimedia information, which is peculiar also in the requirements it imposes on the system in charge to manage it.

3. BASIC REQUIREMENTS

The catalogue database of ESSAI is the multimedia database management system (MMDB), which is in charge to manage data presenting the products available for sale through the teleshopping application.

These are mainly advertisement data and, as such, they are the product of the creative mind of an artist. This is a very important consideration, since it means that the MMDB cannot be based on a very rigid data model: it would be impossible to ask an artist to work following a predefined rigid scheme because her/his work has to be included in a computer system. Moreover, as opposed to usual data management environments, the external appearance of the data becomes as important as its content.

For this reason, we need a data model allowing extreme flexibility not just in defining how data are structured but also in defining how data are to be presented to the end-user.

In order to cope with this important requirement, information stored in the MMDB should include two different types of data: *content data* and *layout data*.

Content data is data as usually conceived: in the case of ESSAI it will be text, image, sound etc. to be shown or played at the teleshopping terminal.

Layout data - which is not considered in traditional database management systems - carries the information concerning how content data is to be externally organized and represented. This idea is not a new one: it has been included in ODA standard ²⁾ and it has been recognized as fundamental in many works ^{3) 4)} but it becomes vital for the presentation of information to the end-user of a teleshopping application.

From the end-user point of view, it is clear that a teleshopping system will be accepted only if it will be able to simulate in the closest way the real selling world. End-users should be faced only with the buying activity, which they execute everyday and for which no training is needed.

This implies a couple of important consequences on the MMDB component of ESSAI.

A first requirement is to support *multimedia queries*. If the usual way to sell goods is to be simulated as close as possible, besides the traditional hierarchical catalogue search and the traditional queries based on alphanumeric data, multimedia queries should also be considered. In other words, multimedia data such as images and sound should not be simply attached to the traditional textual information, but should play a significant role as *active components*, that is, components that can be involved directly in query formulation.

For example, when a customer asks a shop assistant to show her all hats of a given style she is formulating a query on an image - the image of the hat - and the shop assistant answers with a collection of hats - similar to each other - much in the way a usual DBMS answers to a query with a collection of records having some properties in common.

In order to accomplish this task, an appropriate access method for multimedia data must be provided. As actually happens in traditional Information Retrieval Systems for textual information, indexes must be derived from multimedia data too. Indexes in this case take the more general name of *access structures*. A query involving multimedia data will determine a search into these access structures.

The current state-of-the-art in the image and sound processing fields seems not to allow the automatic generation of the access structures. Though this may be further investigated, it seems more reasonable to study methodologies and/or tools to support the manual generation of them.

A query language able to express multimedia queries should define, in addition to the usual operators, those ones needed to specify conditions on multimedia data.

It should be possible, for instance, to apply on image data CONTAINS, SIMILAR and ATTRIBUTE operators. The operator CONTAINS means "search for images containing a particular subimage"; SIMILAR means "search for images similar to a specified image"; ATTRIBUTE means "search for images with a particular attribute (color, brightness, etc.)".

A second requirement is the use of the hypermedia paradigm when organizing information. Consider, for instance, a person entering a shop to buy a camera. He initially looks at a rack full of cameras, which may be regarded as a first hypermedia node. Then he chooses a camera and asks the shop clerk to show it, getting a more detailed view corresponding to a second hypermedia node. Then, for instance, he asks the price, getting a third hypermedia node, which in this case is not an image but a sound item: the voice of the clerk answering the question.

It is worth to stress that customers are not tied to a predefined path to get information. Different customers may choose to "read" different "nodes" and to read them in different orders. This implies that the traditional way to linearly organize the information is impracticable.

Hypermedia is a must. The MMDB component of ESSAI should thus be provided with functionalities to support hypermedia technology.

A last important requirement is the ability to manage multilanguage information. In fact, among the most appropriate sites to install ESSAI vending terminals there are airports and railway stations, where a lot of people speaking many different languages are likely to be attracted by the teleshopping application. A way to easily manage the representation of printed and spoken information in different languages is to be found.

4. DISCUSSION OF THE MMAIN MODEL

Taking into account the basic requirements illustrated in section 3, a conceptual model for the multimedia catalogue database of the ESSAI system called *MMAIN* (*Model for Multimedia Advertisement Interfaces*) is discussed in this section.

From the technical point of view, the main characteristic of advertisement data is its loose structuring. This is in contrast with traditional computer application areas, as for example office automation, where generally data possess, or should possess, a well defined structure.

Moreover, a large number of different media data should be supported at best.

According to these concepts, we devised a paradigm, to support the ESSAI catalogue database, derived from the one used in making movies. It uses fundamentally the notion of *scene*, composed by a set of *multimedia items*.

These are two well-known concepts for movie directors, who create movies by grouping together different scenes, each scene being in turn constructed by means of several items, at least an image sequence and a sound track.

In the ESSAI teleshopping application, in addition, it will be possible to have concurrently on the terminal screen still pictures, graphic animations, moving video, text and, most important, it will be possible for the end-user to interact with all these items through, for instance, screen buttons, joysticks, speech recognition.

In MMAIN, a scene is a set of text, sound, images (still or moving), etc. to be presented in the same interval of time, according to a well-defined layout structure. It can be thought as the unit of presentation on the teleshopping terminal, much in the same way a transparency is the unit of presentation in a conference speech.

A scene can also be viewed as a sort of container for multimedia data. You can imagine it as an interval on a temporal axis with a starting point and an ending point. Between these two points different *multimedia items* rise, live and die.

In figure 2 the scene concept is graphically represented.

The multimedia items composing a scene are of different types - images, audio, text etc. - and for this reason they exist in fundamentally different ways. Nevertheless they all have in common three attributes that clearly locate and

characterize them within a scene: a *start time* - relative to the beginning of the scene - a *duration* and a *content*.

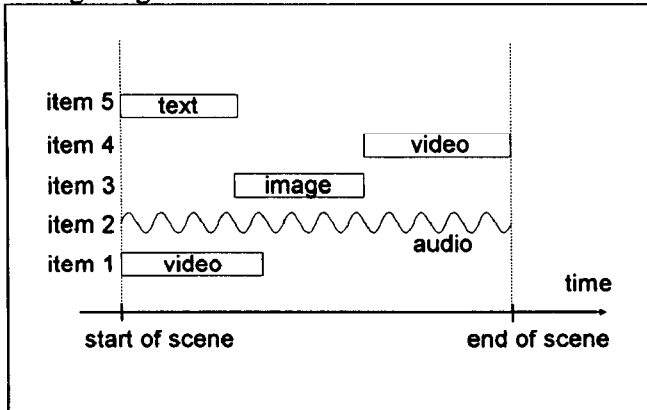


Figure 2. The scene concept

This is modeled introducing the type *MMItem*, representing the general case of multimedia item, featuring the attributes *StartTime*, *Duration* and *Content*.

The *StartTime* and the *Duration* attributes are the first components of the layout data mentioned in section 3. The underlying assumption here is that a generic multimedia item has its own, limited life.

Within the scene context an item may start at a generic point in time, not necessarily corresponding to the scene beginning, specified by the value of *StartTime*.

The *Duration* attribute represents the amount of time the item stands on the scene. The duration is limited to a maximum corresponding to the total scene duration.

The *Content* attribute represents the content data mentioned in section 3. In the most general case the *Content* attribute consists of a *BLOB* (*Binary Large Object*), since for a generic *MMItem* the structure of the contained information is unspecified.

Further attributes forming layout data of a multimedia item depend on the particular type of the item itself.

For example, to model items that can appear on the screen - such as images, video sequences or text - a more specific type *VisualItem* is derived from *MMItem*. Besides inheriting the three *MMItem* attributes, *VisualItem* objects have additional attributes referring to their spatial representation: *XPosition*, *YPosition*, *ScaleFactor* and *RotationAngle*. Moreover, two attributes related to the temporal behavior of a generic *VisualItem* have been introduced: *RiseTime* and *FallTime*. They can be used to specify cross-fading effects.

An *AudioItem*, on the other hand, will have the additional attributes *Volume* and *Tone*.

These observations lead to the definition of a type hierarchy for multimedia items, which is depicted in figure 3. In this figure three further items are derived from *VisualItem*: *ImageItem* representing still pictures, *VideoItem* for full motion video and *TextItem* for the usual text.

As above mentioned, one can imagine the activity of creating a scene like as the making of a scene by a movie director. Continuing with this parallel, when a set of good scenes has been collected, the director proceeds in the editing of the movie, following her/his own favorite, well-defined, order. This means that, in *MMAIN*, a way should be also provided for the definition of a structure that plays

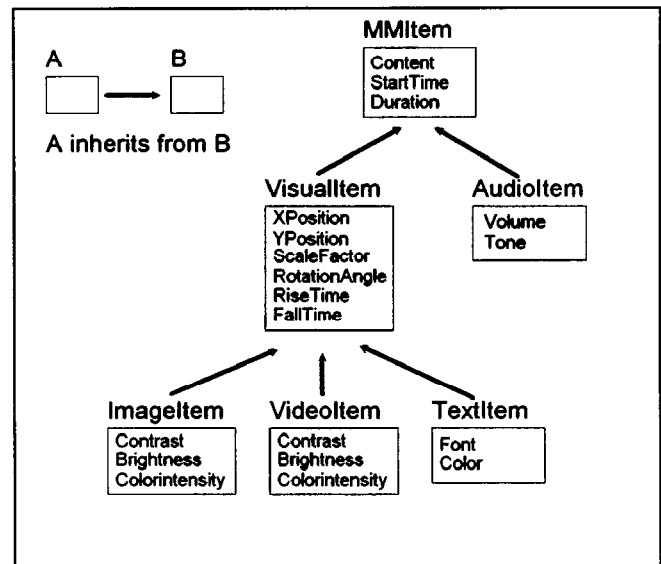


Figure 3. The multimedia item hierarchy

the role of the movie itself, that is a sort of container for scenes. This structure is called *presentation*.

Like a movie, a presentation is a set of related scenes, but with a major difference: the movie is an ordered sequence of scenes, and the order is defined by the director once and for all. This means, as everyone knows, that no interaction is possible between the audience and the movie.

On the contrary, in the *MMAIN* presentation an user should be allowed to interact with the system to change the order of scenes displaying.

Consider for instance a presentation for a car: there will be scenes regarding engine features, internal look, general performances and so on. An user should be allowed to jump with very few constraints from a scene to another, in order to examine those details in which she/he is particularly interested.

In figure 4 the presentation and movie parallel is shown.

It is clear that a non-trivial way to bind scenes in a presentation must be provided. The better solution is to exploit the hypermedia paradigm. In this paradigm information is organized in a non linear way by subdividing it into chunks, called *nodes*, and providing connections among them, called *links* ¹⁾.

In the *ESSAI* case, nodes correspond to scenes, and links may be used to connect related scenes, as above mentioned.

Links between scenes are activated through *anchors*, which are special items attached to the scene whose selection

allows to walk along the corresponding link. The main type of anchor in ESSAI will be a button on the touch screen of the selling terminal, but other solutions as for instance "audio buttons" activated through speech recognition, or physical input devices activated by some sort of mechanical action, are possible.

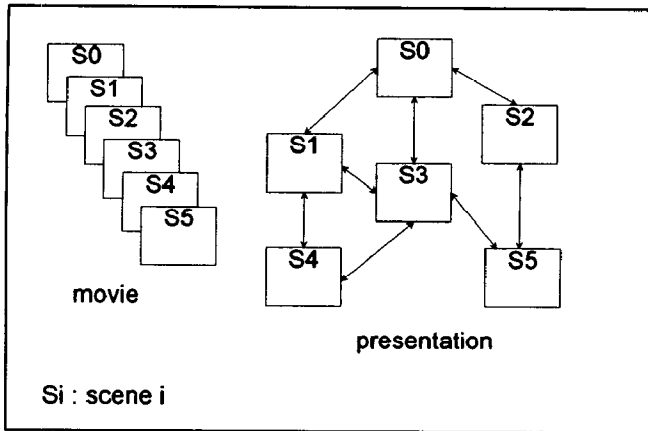


Figure 4. Movie and presentation parallel

Two more concepts complete the definition of MMMAIN: the concept of *perspective* and the concept of *synchronization* of MMItem objects inside a scene.

The basic idea underlying the perspective concept is that information is an abstract entity that, in order to be communicated to a human being, must take one of many possible concrete representations. We say that a given information may be presented from different *perspectives*¹⁾.

The notion of perspective may be useful in many different cases. For example, it can be used in solving the multilanguage problem mentioned in section 3. We can imagine, for instance, an advertisement regarding a car: generally a good mixture of images, sound and text will be useful for explaining all the features of the car. But text and speech items should be displayed or played in the language of the end-user. So information that is to be displayed as text or played as speech may have different perspectives, each perspective corresponding to its translation in a given language.

The perspective concept may be useful in many other cases. For example, since sound items result completely unuseful for deaf people, information represented by default as voice explanations can have other perspectives associated, representing it as a text.

The concept of *synchronization* among multimedia items is needed to better control item behavior within a scene in time domain.

In fact, items in a scene are frequently related to each other in the temporal axis and cannot begin and display freely.

In normal situations the *StartTime* and the *Duration* attribute above defined may suffice to define these

temporal relationships but in abnormal cases more specific information is needed.

Think, for instance, to the most simple case of synchronized items: a video sequence and its sound track. If no problems occur, it may be enough to specify the same start time and duration for both items, in order to get things work fine. But if the video sequence is delayed for some reason, the system should also postpone the start time of the sound track in order to still have audio and video synchronized. Problems like these are likely to occur in the ESSAI environment, since, as explained in section 2, multimedia data items travel through a network and thus may suffer random delays.

To deal with this problem, in MMMAIN two main abstractions have been introduced: *sequential rendering* and *simultaneous rendering*.

Two items are sequentially rendered if the start time of the first plus its duration must be equal to the start time of the second.

Two items are simultaneously rendered if their start time must be equal.

In Appendix A a more formal definition of MMMAIN is given. This is not a complete result since research activities are still ongoing, but it can be nevertheless useful to better fix the ideas discussed in this section.

5. LOGICAL ARCHITECTURE

In figure 5, the logical architecture of a system supporting MMMAIN, is illustrated.

At the top level it is composed of three major blocks.

The first block contains the *MultiMedia Storage Subsystem*, which provides storage containers for multimedia data and maintains information for retrieving these data (the *access structures*). The MultiMedia Storage Subsystem has to take care of data retrieval and, if necessary, of short term recovery. It is the real database engine for MMMAIN.

A second block is composed by the *Data Analyzer* and *Data Formatter* modules; the first module should be able to analyze multimedia data (image, sound, text etc.), in order to construct for each of them the appropriate access structures to be stored in a portion of the database. The second module should be able to format (compress) data in order to store them in a convenient way. The operations performed by this block may be carried out off-line.

A third block is responsible for the interaction with the teleshopping application and is composed by the following parts. The *Query Composer* is the module that interacts with the application in order to get queries formulated by the user.

Queries may be specified in an implicit and non-precise way, for instance touching an object on the screen, and asking to get information on all objects similar to it. The Query Composer should be able to compose a precise query expressed in a formal query language, possibly

starting from an unprecise one. The *Query Processor* should analyze the query elaborated by the Query Composer and, after having performed some checks on it, should decompose, optimize and send it to the MultiMedia Storage Subsystem. The *Hypermedia Browser* is in charge to manage the navigation among nodes, through anchors and links. Finally, The *Scene Builder* should provide the application with the retrieved data and manage all the directives useful for their presentation to the end-user.

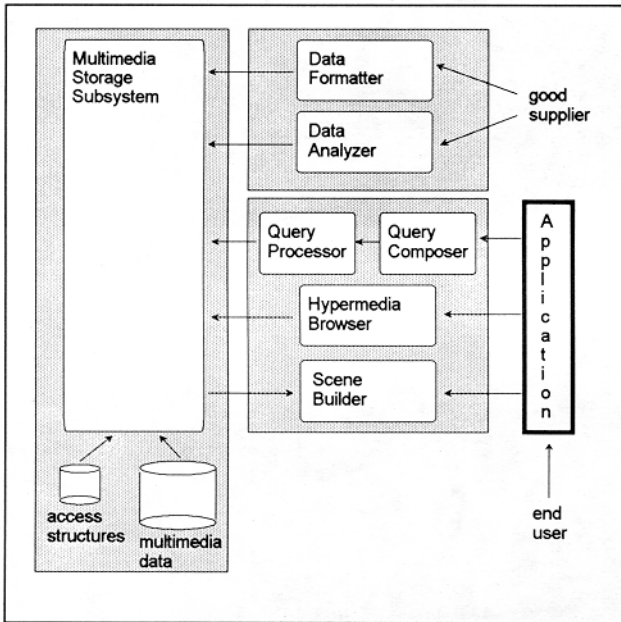


Figure 5. Logical architecture.

6. CONCLUSIONS

In this paper, the requirements in terms of data management needed by an advanced teleshopping system, being developed in the frame of the RACE II ESSAI Project, have been analyzed. The most important of them are: extreme flexibility, non-linear browsing of information and multiple language management.

A conceptual model satisfying those requirements, called MMAIN, has been proposed.

The MMAIN model solves the problem of flexibility by defining data through an inheritance hierarchy that can be specialized as needed and by introducing the concept of *layout data*.

The problem of non-linear browsing has been solved including the hypermedia paradigm, while multiple languages are managed through the *perspective* concept.

A logical architecture for a system to support MMAIN has also been presented. Future work will be directed to the implementation of such a system.

The step we are at present undertaking in ESSAI is to evaluate if, among existing multimedia authoring toolkits, there are systems that possess enough functionalities to be extended so as to include all MMAIN concepts. If this will

reveal impossible, an ad-hoc system will be developed inside ESSAI.

7. ACKNOWLEDGEMENTS

The authors would like to acknowledge the other companies involved in the ESSAI Project Consortium that made possible the work reported in this paper.

These companies are: COSI (I), ASCOM AUTELCA (CH), SEVA (I), THOMSON-CSF L.E.R. (F), SYSECA (F), OST (F), STZ (D), ELEKTRONIK CENTRALEN (DK), ASCOM TECH (CH), ETH Zurich (CH), SNV (D).

APPENDIX A

Based on the discussion presented in section 4, in this appendix we try to give a more precise definition of the MMAIN model, in terms of data types and operations, using an object-oriented-like notation.

```
class DataBase {
    DBPresentations : setof (Presentation);
operations:
    GetPresentation(Name : String):Presentation;
    SendQuery(Query:String):setof(Presentation);
}
```

```
class Presentation {
    Name : String;
    Scenes : setof (SID);
    HyperMediaLinks : setof (Link);
operations:
    StartScene() : SID;
        // get first scene to be played
    FollowLink(Source : SID, Anchor : AID) : SID;
        // follow link activated by Anchor
}
```

```
class Scene {
    Identifier : SID;
    MultimediaItems : setof(MMID);
    Anchors : setof(AID);
operations:
    GetSimultaneous(Item : MMID) : setof(MMID);
        // Get MMItems to be simultaneously
        // rendered with Item
    GetSequential(Item : MMID) : setof(MMID);
        // Get MMItems to be sequentially
        // rendered with Item
}
```

```
class Link {
    SourceScene : SID;
    SourceAnchor : AID;
    DestinationScene : SID;
}
```

```

class MMItem {
    Identifier : MMID;
    Content : BLOB;
    StartTime : Time;
    Duration : Time;
}

class VisualItem is_a MMItem {
    XPosition : Integer;
    YPosition : Integer;
    ScaleFactor : Real;
    RotationAngle : Real;
    RiseTime : Time;
    FallTime : Time;
}

class AudioItem is_a MMItem {
    Volume : Integer;
    Tone : Integer;
}

class ImageItem is_a VisualItem {
    Contrast : Integer;
    Brightness : Integer;
    ColorIntensity : Integer;
}

class VideoItem is_a VisualItem {
    Contrast : Integer;
    Brightness : Integer;
    ColorIntensity : Integer;
}

class TextItem is_a VisualItem {
    Font : FontType;
    Color : RGB;
}

class MMAnchor {
    Identifier : AID;
    // ... to be further investigated;
}

```

SID, MMID and AID are object identifier internally generated and managed by the system.

REFERENCES

- 1) F. Garzotto, P. Paolini, D. Schwabe, "Authoring-in-the-large: Software Engineering Techniques for Hypertext Application Design", in Proceedings of the Sixth International Workshop on Software Specification and Design, Como, 1991, pp. 193-201.
- 2) ISO, Information Processing: Text and Office Systems; Office Document Architecture (ODA) and Interchange Formats, (ISO 8613, 1989), Parts 1-8.
- 3) Magda M. Mourad, "Object Representation for Multimedia Document Interchange", presented at the Conference ICC 1991;
- 4) C. Thanos, Multimedia Office Filing, (North-Holland, 1990, Amsterdam), Ch. 3 pp. 17-52.