

A User Friendly Platform for the Automatic Production of HTML Multimedia Title Series

John Garofalakis^{1,2}, Penelope Markellou^{1,2}, John Moutzidis^{1,2}

¹ Research Unit 5: Internet & Multimedia Technologies
Research Academic Computer Technology Institute
61 Riga Feraiou str., 26221 Patras, Greece
Email: {garofala,markel}@cti.gr

² Graphics, Multimedia and GIS Laboratory
Department of Computer Engineering & Informatics, University of Patras
26500 Rio Patras, Greece
Email: moutzid@ceid.upatras.gr

Abstract: This paper presents a user-friendly platform for the automatic production of sets of interactive multimedia title series with similar content and structure. Our approach consists of three main parts: the first one is the MTS (Multimedia Title Specifications) module which gets as input the title specifications from the publisher and provides as output their automatic storing into the system database. The second one is the TMDBE (Title and Multimedia Database Editor) that helps the end-user to manage the multimedia data (video, image, text, audio, etc.). The last module CG (Code Generator) gets as input from the database the title editing information and produces as output a source code for the target multimedia platform, in our case web browsers (Netscape and Internet Explorer). The main advantages of our modular architecture arise from the fact that we can use any relational database, the TMDBE can be implemented in any programming language and the modules are completely independent to each other.

1 Introduction

Nowadays, the importance of multimedia technology, services and applications has been widely recognised. Many initiatives, both national and international, are actively promoting the research activities and experiments in this field, which is progressing very fast. Moreover, the ability to automate the multimedia title production is one of the main challenges.

Traditional methodologies follow strict stages from designing, storyboarding, script development, production and authoring. Furthermore, the current approaches [2] require that the publisher should have a wide range of knowledge and specific programming skills. The design and development of such applications by a friendly, robust, correct and efficient way are not trivial or straightforward.

Lang and Barry [7, 8] studied the way that multimedia systems are developed in practice. Their findings revealed that practitioners are not using the multimedia de-

velopment methodologies prescribed by academic literature and that most are using their own in-house methods. Lang in studies [5, 6] mentioned that traditional tools and techniques appear to be inadequate for hypermedia system modelling.

In [1] a method for developing interactive multimedia through the modelling of semantic content is presented. The benefits of the method for the developer are: development of interactive content, re-usable media and model, dynamic media use, full media reasoning, front-end tools that support consistent, precise and quick modelling. From the user's point of view the benefits include: interactive media, clearer identification with concepts, media presented according to user requirements. Other research works are presented in [4, 9].

In this context, this paper presents a user-friendly platform for the automatic production of HTML multimedia title series. This environment is based on a new authoring model that allows the production of title series in a fast and easy way. The whole work has been done in the framework of a research project entitled "AMESA"¹. Its main purpose was to create an environment to facilitate the production of multimedia title series on web browsers (HTML code). In the context of this project two case studies of our system were implemented.

A title series is a set of multimedia titles with common concept and comparable content, structure and organisation. A series can consist of titles as similar as quarterly updates of a product catalogue or as diverse as presentations of different companies. Other examples of title series can be the following: educational lessons on various scientific areas, presentations of touristic places, portraits of prominent people, historical events, sports and leisure, product presentations, etc.

However, it is important to note that under our perspective, all titles in each series come under the generic description of the title series, which we call Model Title Specification (MTS). The MTS embodies the idea, vision and creativity of its designer, who is not necessarily a computer professional. In many cases this person is the publisher himself or the multimedia title series director or whoever wants and can define the requirements for the series. Then the software engineers develop the specific system. The final system offers a user-friendly environment, which can be used by less qualified personnel. These people, who are considered as the end-users of the platform, choose the templates and the forms from the available ones, fill in the content and the multimedia objects, test their titles, improve and correct the titles and finally produce the various titles of the series.

As a result of this approach, a significant reduction of time and cost may be achieved. Moreover, highly qualified and expensive personnel are not needed except for the phase of system development. Even in this case, the experience that has been acquired at the first development of such a system is really fundamental since parts of the initial code can be reused in new systems.

From the abovementioned, it is clear that the main contribution of our architecture is the modularity and consequently the independence of components, front-end, back-end and Code Generator (CG) respectively. The front-end (MTS and Title and Multimedia Database Editor or TMDBE respectively) component interacts with the human users in order to store in the back-end component (database) the appropriate information about the specifications, the multimedia data and the editing of the desirable title, which satisfies the specifications above. The CG component is the "heart"

of our environment since taking directions from the database (specifications, multimedia data and editing) produces HTML code for an interactive multimedia title on the Web.

Previous work in this field was presented in the project entitled “GMI - Greek Multimedia Initiative: Development of a Methodology and an Innovative Authoring Environment for Series of Interactive Multimedia Tourist Guides on CD-I” [3]. In this attempt, a set of titles with content and structure similarities were specified through a model, which was a blueprint about content, presentation and interactivity and whose design was modelled in OMT (Object-oriented Modelling Technique).

Moreover, in [10] the concept of GMI was exploited through the project “VALMMETH – Validation of a Multimedia Title Series Production Methodology”. This project aimed to produce customized authoring tools for multimedia title series through a large-scale validation. In their development the Java-poet class was used, which was very restrictive to future upgrades especially for the database that was developed in POET.

In our architecture we automate the previous work since the specifications can be given electronically via appropriate forms and can be stored at the specific part of back-end component. We can also use any relational model for database, since the TMDBE component can be implemented in any programming language and all modules are completely independent to each other.

So, the advantages of our approach relate to:

- The production of series of multimedia titles in a fast and easy way.
- The development effort for each new title does not require special computer knowledge.
- The production cost of a new system reduces referring to the conventional one.
- The reusability of the code that has as a result the significant reduction of development time of each new system.
- The effective and friendly user interface of the system, which speeds up the title production process.
- The education cost and time for such a system is minimum.
- The system is very easy to be upgraded with new functionalities.

This paper is organized as follows: section 2 illustrates the functionalities of the developed environment. Section 3 presents the proposed system architecture and describes software modules and system integration. Two case studies are described in section 4. Conclusions and points about future directions of our work are presented in section 5.

2 The Proposed Approach

The proposed approach is suitable and applicable for the production of interactive multimedia titles series. For each series a customised system is designed and developed. This system provides the end-users with a usable authoring environment for the production of the multimedia titles. Our approach is especially appropriate for the production of a variety of similarly structured titles. The system functionality and

methodology relies on the definition of three users categories: (a) the publisher, (b) the developer and (c) the end-users (editors). In small firms, two (a and c) or even all three users categories could be realized by the same person.

The steps that are followed for the production of a multimedia title series, according to our definition are:

1. The publisher defines the requirements that the specific system (Title Series) should satisfy via the MTS tool.
2. The developers take the output of the MTS tool (which consists of forms, reports and pseudo-code), design and implement-customize the specific system.
3. The editors produce the various titles.

As it has already been mentioned, the first step is the definition of user requirements and needs. The intended audience of this kind of systems is the people that will use it for the production of a series of multimedia titles. Besides the details of the computer configuration section, no particular technical expertise is required by the user's part. However, it is assumed that the user will have a firm understanding of the corresponding Model Title Specification (MTS) tool.

The MTS is the generic description of a series of similar multimedia titles. The MTS grew from the need to specify with ample flexibility the multimedia presentation requirements of different potential customers. It also serves the purpose of explicitly streamlining the work to be done in the preparation of such a multimedia title. Specifically, MTS represents the thorough domain analysis where all concepts are clearly expressed. It also includes a description of all types of multimedia assets that will be used. A typical example is the specification of the format of image, video and audio files (such as TIFF, QuickTime, AVI, WAV, etc.).

Moreover, the definition of all types of interactivity required in the series titles is supported. Typically this refers to buttons, lists, hot-spots, menus, etc. A clear resolution on the functionality of the application is provided to the user (for example, access to Internet addresses, database support, etc.). The presentation structure of the series (for example, navigation between screens) and the types of events that trigger transitions are also helpful. Normally, screen layout and temporal synchronisation details may be left for the final stage of the MTS definition. If some of these issues are deemed important then they should also appear on the draft MTS.

The development of the system is based on the basic requirement: a minimal title production effort. This implies the need for a simple, straightforward, intuitive and friendly user interface. Therefore, from the editor's point of view, there is no direct distinction between the database editor (to enter and edit all multimedia assets that are necessary to build a title) and the title editor (to compose a title with assets from the database). These two parts are integrated into one application.

The process that the editor should follow in order to produce a multimedia title with whichever system consists of the following stages:

1. Definition of the specific title scenario. This scenario should follow the general structure of the system as defined at the MTS by the publisher.
2. Multimedia data collection or creation.
3. Categorisation and insertion of multimedia data into the system (multimedia database).
4. Improvements, corrections and final montage of the title.

5. Automatic production of the web-based multimedia title.

The editor has also the opportunity to create a multimedia library containing texts, photographs, buttons, backgrounds, etc. This library will be capable of handling the content of every possible multimedia title. It will also reduce the time needed for the production of any new multimedia title with the specific system and will increase overall production.

The themes that such a system can cover are infinite. An interesting future area will be continuous learning. It is widely acknowledged that multimedia technology, whether delivered from a CD-ROM or via the web, offers great potential benefits to learning. The combination of audio, video and text offers particular benefits since this kind of multimedia material carry particular added value to the learning process. The attraction of our approach lies precisely on: the reduction in time associated costs, the reduction in personnel associated costs, the ability to involve educators directly in the production process, or parts of the process, by providing them with an easy to use tool, the simplicity and low-cost of the system.

Another important issue is that the cost of customising the system depends on the complexity of the multimedia title series. When the development team has already developed one system before, then the pure labour cost for customising a new system for another series is less since:

- most of the components of a system are reusable.
- our experience indicated that all systems use the same key design and implementation concepts.
- the systems are transferable and applicable to various domains.

3 System Architecture, Modules and System Integration

The architecture of the system presented in figure 1 has been designed in such a way that would make our platform even more flexible. Each component is completely independent from each other, so that different components can be developed in different languages depending only on their basic functions. This helps maintenance and upgrading of the system.

The DBMS should be a relational one and all connections to the DB are implemented using ODBC to ensure independence from the RDBMS. Choosing a relational DBMS over Object Oriented or Object Relational satisfies the need for a common, world wide spread DB. As the purpose of the tool is not to built Multimedia DB but to easily deploy a multimedia title, and as only information about the multimedia files are stored in the DB, the choice made to use a relational DB doesn't affect the efficiency of the tool.

The Code Generator (CG) is implemented using COM technology, which is supported by many development platforms. Implementing the CG as a DLL file is a de-facto standard procedure used also in many commercial products that allows quick and easy future upgrade of the code generation process (simply replace the DLL file with a new one).

The connections to the DB and the calls to the CG are completely transparent to the end users whose only interaction with the system are the two user tools, that are the MTS Tool and the Title and Multimedia Data Base Editor (TMDBE), two fully graphical user interface tools designed especially for people that are not code experts to help them build applications or multimedia presentations for browsers.

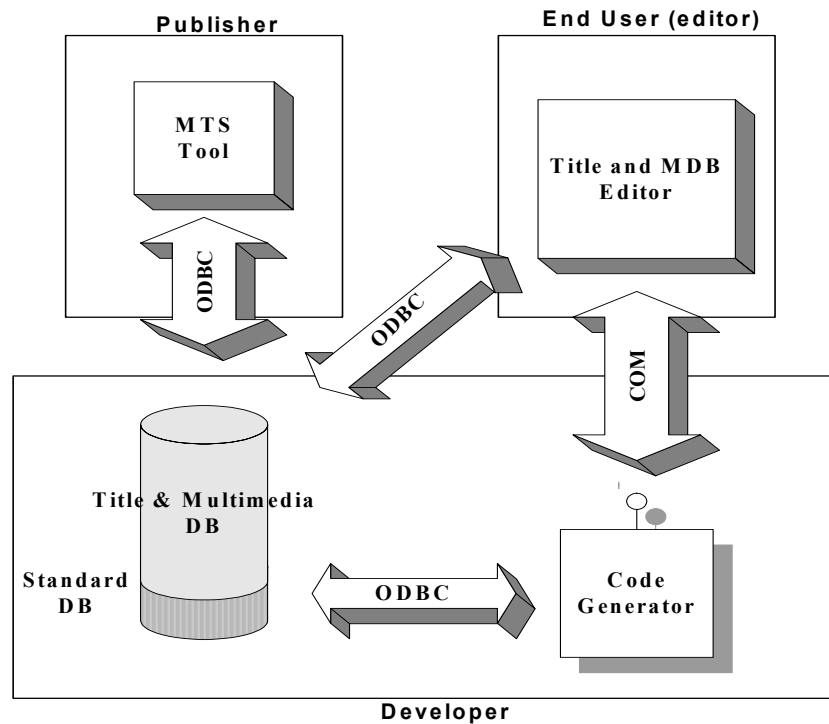


Figure 1: System architecture and user interfaces.

3.1 Title and Multimedia Database

Title and Multimedia Database (TMDB) is the repository of all applications developed using the application-platform. TMDB stores information about all multimedia objects that might be used in the process of developing the complete series. The multimedia objects could be stored as BLOB object but, considering the fact that the system makes no use of the multimedia object itself but information concerning both the style of the object (position, dimensions etc) and its interoperability with other objects, the BD stores only a reference to a physical location in a disk.

The schema of the TMDB depends on the focus-purpose of the series and in the general case is variable although parts of it can be used in another application that handles a different series.

3.2 Standard Database

The standard Database (DB) contains all the data required by the Code Generator about the HTML objects. The schema of the Standard DB is well defined, cannot be altered and is the same for all applications. The HTML objects that are described are the following: Page, Paragraph, Anchor, Image, Multimedia object (video, sound), Table, List, Text Area.

The information stored concerns the different objects used in the application, as well as their style and their interoperability (actions, events that trigger them and arguments). Each HTML object stored in the Standard DB, has a unique id which is used by the Code Generator and the TMDBE to retrieve and modify the attributes of the object.

3.3 Model Title Specification Tool

This tool is used by the publisher to define the maximal application of the series. Each application of the series can be a subset or the complete set of the application that the user has defined using the MTS tool. MTS stands for Model Title Specifications. MTS tool uses a user-friendly fully graphical interface to help the user define the model of the application. The user creates category pages, which then fills in with HTML objects. For each HTML object inserted in a category page the user defines not only the way the object looks (style) but also the way it interacts with the user. The MTS tool stores in the Standard DB all the information about the style of each category page and the objects that contains. Finally, the MTS tool gives out a printout all the MTS that is used by the development team to implement the Title and MDB Editor. The MTS tool is standard for all application series.

3.4 Code Generator

The Code Generator (CG) is a COM object in the form of a Dynamic Link Library, implemented in C++, which has several interfaces used to call several methods that get the HTML and JavaScript code. For each HTML object a different interface has been implemented. The corresponding interface gets the id of the object and returns the HTML and JavaScript code as text.

Each time an interface of the COM object is called the TMDBE passed the id of the object for which the code is requested. The Code Generator establishes an ODBC connection to the Standard DB, and by using the object id, retrieves all information about the style and interoperability of the object stored in it. As information is retrieved, it is formatted into HTML and JavaScript code and the Code Generator returns that in the form of text to be further processed by the TMDBE. All layout information about the object is formed and returned as HTML code where as all interoperability information is formed and returned as JavaScript code. The CG places each object in a unique-different layer so that absolute positioning and unique reference of the object can be achieved. The code returned by the Code Generator follows the Document Object Model Level 2 specification of the W3Consortium.

The code generated by the CG can be used for standard HTML pages as well as for dynamic HTML pages such as ASP. At this point, it must be stressed that the CG does not generate the JavaScript Functions code possible used in some actions of an HTML object, although it natively supports simple ones such as hide/show layers, change source of an image (rollover effect) and alert messages. The code of those, more complex, JavaScript Functions, is supplied – developed by the team responsible for developing the TMDBE.

3.5 Title and Multimedia Database Editor

The Title and Multimedia Database Editor (TMDBE) is an application developed especially for each series according to its' needs. The user facilitates this application to manipulate multimedia objects stored into the TMDB that is to insert, update and delete objects. After filling the MDB with objects the user can use the Title Editor to build a title-application of the series. Using a page template that has been described with the assistance of MTS tool, the user can create pages just by selecting the desired objects from the TMDB. After completing the editing of the title-application the user can choose to build the application code.

At the beginning of the development process the publisher uses the MTS tool to define the maximal application of the series, which is the MTS. During this process, the styles of the various objects of the application (information such as width, height, absolute position) are stored directly into the Standard DB. The rest of the information is given to the development team as input so that they can design and implement the TMDB as well as the TMDBE. As soon as the development process has been completed, the end user can start using the Title and MDB Editor to insert into the MDB multimedia objects that may be used in an application of the series and to start building an application. After completing the procedure of selecting objects from the MDB and inserting them into each page of the application the user may choose to produce the code for the application. At this point the Title and MDB Editor uses the information stored during the previous procedure concerning the users' selections, to make calls to the Code Generator and format the returned values into complete HTML pages. The user may also store the application developed by the TMDBE (pages created, objects used, actions defined) into the corresponding DB, as well as to load stored applications and modify them.

4 Case Studies

Sample applications have been designed and implemented in order to demonstrate the proposed approach. The first application is a bookstore online catalogue, where the user navigates the pages and reads information about the books. Specifically, this application produces web-based multimedia catalogues according to the requirements of the specific publisher. The final system enables the user to incorporate information on every aspect of the bookstore's online web catalogue, categories and books. This online web catalogue has no need for dynamic pages as all information about the

books is static and the web catalogue doesn't include an e-Commerce part where users can buy books.

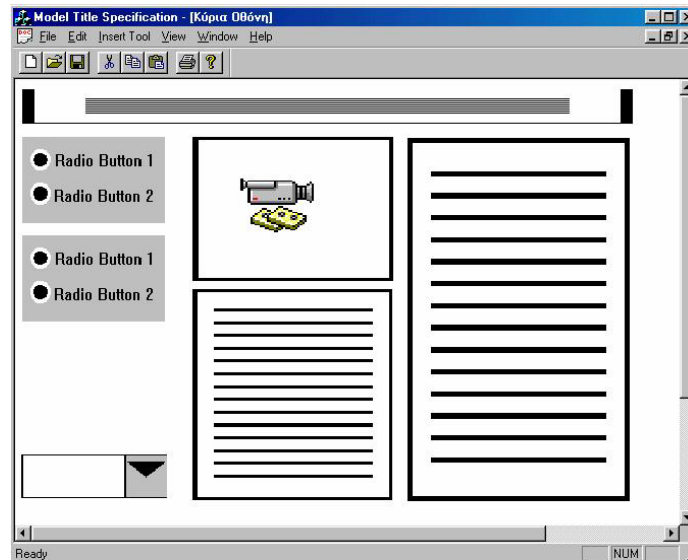


Figure 2: The MTS Tool.

The structure hierarchy of this system depicts the page-structure of the web site presentation. It is arranged in three levels, the Introduction, the Categories and the Books.

The Introduction consists of the following types of pages:

- Introduction: optional introductory page containing the logo of the company, images, text and/or animation (animated GIFs, etc.) and/or video (in “heavier” versions).
- Language selection: optional page where the user selects the desired language.
- About: optional page with information about the contributors.

The Categories page embodies:

- Book categories: numerous of optional pages mainly designed to present a category of books.
- Book sub categories: numerous of optional pages representing sub categories of books.

The Books contains the following information:

- Description: information about the book.
- Author: information about the author of the book.

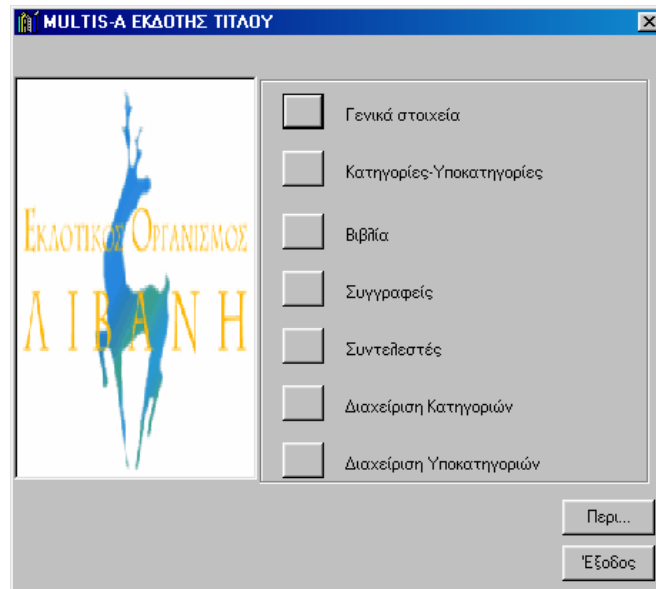


Figure 3: Bookstore online catalogue.

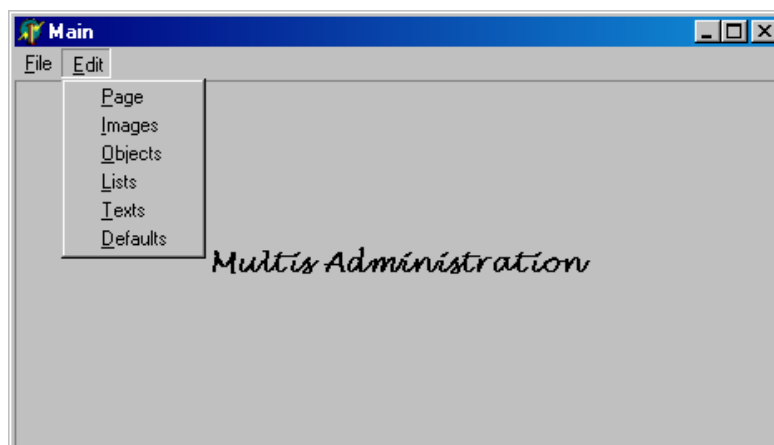


Figure 4: Title and MDB Tool for the Bookstore online catalogue.

The second application deals with the educational presentation of sports. In this case the user has the opportunity to navigate different screens via the use of buttons, hot-spots, hyperlinks, etc. In both applications the multimedia data available are still images, graphs, video and audio clips, as well as plain text. Specifically, this application produces web-based multimedia presentations for various sports e.g. football, basketball, volleyball, tennis, etc. The user can navigate pages related to every aspect of the sport description.

The structure chart of this system includes all the main thematic categories: history, material specifications, rules, technical principles, training issues, food cycle, etc.

5 Comparison, Conclusions and Future Work

Our work specifies and implements a system for rapid development of application-multimedia series and addresses users who have little or no knowledge about development. One could consider Microsoft's FrontPage or Macromedia's Dreamweaver as competitive products to our system. The truth is that both of them produce HTML pages that can incorporate multimedia objects and can define template pages as our system does, and more over are much cheaper to acquire.

The major difference between both of those systems and our own is that our system is a specific purpose tool, with limited capabilities (compared to the first two) that even a person with no knowledge of programming can use it to produce a series of application-multimedia titles just by selecting the multimedia objects that need to be inserted in each page. Further more, for each page the user knows what kind of objects he/she is allowed to insert, and as he/she inserts them there is no need to write even one row of code to ensure interoperability with other objects as this is predefined and the Code Generator takes care of all JavaScript code.

For the future, great step forward for our development platform is being considered, that is the use of XML for representing the MTS and all of its components. XML is the ideal language for describing objects as well as exchanging information between applications or processes. More over, most of the development and application platforms support XML, thus making the produced application-title series platform independent.

Using XSLT and XML Schema on the XML data application-title series production will be simplified and the developing cost of our platform will be reduced as more components of the developed software may be reused. Popular RDBMs like Microsoft SQL Server 2000 already support queries that return XML formatted data, and this will simplify both the use of MTS Tool and Code Generator.

As a conclusion, one could notice that our methodology-tools is not yet another general-purpose tool for non-expert users to build applications for browsers, but a specific purpose tool for users who have no expertise in building applications and want to build a series of applications for browsers, saving time, effort and resources.

References

1. Agius H., Angelides C.: A Method for Developing Interactive Multimedia from their Semantic Content. *Data and Knowledge Engineering* 34(2):165-187 (2000)
2. Dick C., Bulterman A., Hardman L.: Multimedia Authoring Tools: State of the Art and Research Challenges. *Computer Science*, 575-591 (1995)
3. Gardelis D., Hadzilacos T., Kourouniotis P., Koutlis M., Megalou E.: Automating the Generation of Multimedia Title. *Proceedings International Conference on Advanced Science*

- and Technology (ICAST), Entering the 21st Century-Multimedia Information Systems*, Chicago, IL (1994).
4. Jourdan M., Roisin C., Tardif L.: A Scalable Toolkit for Designing Multimedia Authoring Environments. *Multimedia Tools and Applications*, Kluwer, 23:257-279 (2000)
 5. Lang M.: A Study of Practice in Hypermedia Systems Design. Doctoral Consortium, European Conference on Information Systems (ECIS), Bled, Slovenia (2001)
 6. Lang M.: Issues and Challenges in the Development of Hypermedia Information Systems. *Proceedings 11th Annual Business Information Technology Conference (BIT)*. Manchester, England (2001)
 7. Lang M., Barry C.: Contrasting Approaches, a Study of Traditional and Multimedia Information Systems Development in Ireland. *Proceedings Annual Conference of Irish Academy of Management*, Derry, Ireland (2001)
 8. Lang M., Barry C.: Techniques and Methodologies for Multimedia Systems Development: a Survey of Industrial Practice. *Proceedings IFIP Working Group 8.2 Conference*, Boise, Idaho, (2001) 77-86
 9. Marchisio C., Marchisio P.: MediaTouch: a Native Authoring Tool for MHEG-5 Applications. *Multimedia Tools and Applications*, Kluwer, 14:5-22 (2001)
 10. Megalou E., Hadzilacos T.: On Conceptual Modeling for Interactive Multimedia Presentations. *Multimedia Modeling, Towards Information Superhighway*, World Scientific, (1995) 51-66

ⁱ This paper was conducted within the context of "AMESA" project. The Greek General Secretariat for Research and Technology funded the implementation of the project.