The evolution of multimedia applications at the University of the Balearic Islands

Antoni Bibiloni, Ricardo Galli and Llorenç Valverde

This article presents different projects and results from a research and engineering process at the IT and Multimedia Technologies Laboratory (ITMTL) at the University of the Balearic Islands. This process was developed over the past decade, providing a certain perspective on the evolution multimedia applications underwent during that time. With this objective, we will list some of the most recent projects and briefly explain their aims, the development process followed and some of the results obtained.

Introduction

In December 1992, the multimedia CD-ROM entitled "UIB 92: A Goal for the Future" was released, the materialisation of the vision and support of the then Chancellor of the University of the Balearic Islands, Nadal Batle, who some 18 months earlier had listed the challenges and power of innovation associated with new multimedia technologies to his colleagues at the Department of Mathematics and Information Technology. It should be pointed out that at that time CD-ROM readers were very rare and it would not be until early 1993 that the first computers with an integrated CD-ROM reader would appear on the market. A multi-disciplinary team was charged with the tasks that resulted in the aforementioned CD-ROM. It is also important to mention, because of its curious nature and to demonstrate the extent to which Dr. Batle's explanations were on the mark, that the presentation of the official CD-ROM was accompanied by the presentation of another, alternative CR-ROM, also made by a multi-disciplinary group led by members of the same department. These early experiments with multimedia technologies at the UIB would be followed by others, including a Masters in Interactive Multimedia with European funding. Since then, experimentation with multimedia technologies has experienced highs and lows but has not stopped being a factor at the department and led to the creation of the IT and Multimedia Technologies Laboratory (ITMTL), the basic function of which is to coordinate training, research and development activities relating to multimedia technologies which, as is well known, no longer rely on the CD-ROM as their only or main support but are now shared with the Internet, DVDs and, more recently, such unlikely supports as electronic organisers and mobile telephones.

This article describes some of the latest projects developed at the ITMTL, such as an educational game *Hudà*, first developed for a CD support and later adapted to the Internet. There is also a brief description of the Minority Newspaper to New Media (MnM) European project aimed at promoting the adaptation to multimedia technologies of European newspapers written in minority languages. Finally, we present the most recent project, *Escripnauta*, an Internet-based multimedia story competition geared towards schoolchildren in the Balearic Islands, based on a web application developed at the ITMTL. But to begin with, we would like to take a brief look at the general features of multimedia applications, such as the evolution of specifications and classification criteria.

Antoni Bibiloni
Ricardo Galli
Llorenç Valverde

Professors of the Department of Mathematics and Computer Sciences at the University of the Balearic Islands

Evolution

a) Markup languages

With regard to markup languages, we believe we are currently in one of the final phases of the developmental process of specifications, fruit of an evolutionary process that began with the first HTML specification and of which XML is one of the most popular exponents today. We are therefore witnessing how the needs that arose over many years have become a tangible reality.

One example is XML (Extensible Markup Language), which is simply a standardised text format that serves to represent any type of information structured in any type of Internet support. Its beginnings can be traced to the 1960s, when IBM tried to structure documents in an organised fashion to facilitate data exchange and handling. GML (Generalised Markup Language) was created for this purpose, but the first standardised and structured information technology of any importance was SGML (Standard Generalised Markup Language), also from IBM. In 1968 it was considered an ISO standard. Although it is still very powerful, SGML is very complex and needs a great deal of software for processing, something which involves a high level of complexity and elevated costs. SGML was therefore not a clear alternative to HTML hypertext at the beginning of the 1990s.

HTML was defined by CERN researchers to share formatted documents online and was basically a simplified SGML application. Over time, webpages became increasingly sophisticated and HTML had to add new technologies to meet demands from Internet users and companies. Dynamic content technologies like database connections using HTML as an interface were among the applications that led to a great change in the idea of the Internet. In particular, interaction with corporate systems made it possible to develop fields such as electronic commerce, bringing meaning to the fields of economic transactions. Of course, HTML was not designed for this purpose and the W3C Consortium began to study a broader and more flexible alternative (without a fixed set of labels). In 1996, a study began into the possible adaptation of SGML to a Web context. Extensibility, structure and validation were the three premises for maintaining SGML in the new language, which was named XML. Its first specification date was February 1998.

XML established a way of transmitting structured data. It was a meta-language that made it possible to define new markup languages, such as XHTML, WML, MathML, SMIL, etc. One of the main goals behind XML was to separate content and presentation. This was possible because it could support personalised markup vocabularies, meaning that data and the relationships among them could be described.

b) Distribution

Multimedia applications form part of society's communication network today, as another way of distributing information and knowledge, using applications that can range from different physical supports, like CD-ROMs, DVDs and communication networks, through to the Internet, mobile phones, and digital television platforms, thanks to the fact that the interactivity requirement can now be implemented on all digital content-distribution networks and/or platforms.

Content suppliers need to offer content with multimedia features, i.e., where the same source of content can be distributed on many distribution channels. We believe this multi-distribution possibility acts as an element to invigorate technological evolution in the development of multimedia applications. One example are mobile phone networks. The initial premises of the technology include already existing open standards, such as http protocol or XML language, the independence of the mobile communications technology on which it is implemented, e.g., GSM or GPRS, and the independence of the mobile terminal, from a telephone through to a PDA.

WAP is the wireless applications protocol and WML is a "markup" language that allows text formatting in a WAP environment. It uses a range of markup labels that determine the appearance of content visualisation. It is defined through the use of XML regulations and is therefore an XML application. It is therefore the developmental standard in this environment.

With regard to the distribution of audiovisual content, the standard formats on distribution methods are practically already defined. The MPEG family includes MPEG-2 for DVD-Digital Video TV, MPEG-4 for the Internet and mobile networks.

c) Classification criteria

It wasn't long ago that multimedia applications were classified according to parameters such as the nature of the application with respect to the use availability by individual or multiple users. They were also classified by the physical location of both the application and the data, distinguishing between stand-alone and distributed. They were even classified by their distribution support, i.e., CD-ROM, CD-I, DVD, Internet, etc.

The evolution of developmental tools, available resources and current distribution platforms (both on physical supports and communications networks) led to new parameters for classifying and assessing multimedia applications in general.

The properties we should bear in mind include the use of parameters such as extensible, compatible, standardised formats, communication-network distribution, data and document sharing, self-updating data, new-content generation, meta-content and meta-data, multi modal, multi user, usability, interface optimisation, etc. These will determine whether our multimedia application is independent of environments and platforms, whether we can guarantee its evolution or life cycle and whether it will be a reference point for research and development.

Extensible is understood to mean an application developed using open environments and languages, which is not dependent on a specific environment or particular multimedia author system. It includes applications that can be freely applied or used where we like or where they are needed. Permits do not have to be requested or payment made for extending or broadcasting these multimedia applications.

We understand **compatible** to include the possibility of applications able to extend their sphere of application, with spheres understood as a set of platforms available for executing a multimedia application. For example, from our point of view, we should not today talk about an application that only runs on a particular hardware platform or a particular operating system.

Using compatible and standardised formats, such as MPEG, we can design applications independently of multimedia data formats or the way data are presented to users. We understand that storing audiovisual data is a process that has to be based on standard formats, where we are able to share communication networks on standardised protocols. For example, it is hard to imagine today that anyone could create a streaming video server without using the MPEG-4 format or RTSP protocol.

The possibility of sharing data between applications initially developed with different objectives can also be considered. We only have to use data models and meta-content on standardised regulations for this feature to offer the ability to share repository networks and completely extend our application's possibilities. A clear example is the use of XML for documents. This property enables us to define applications capable of updating their data with regard to other data in a simpler format.

The wonder of content and knowledge self-generation represents a new step forward in the design of multimedia applications. Now not only are we able to present information to users but users can generate and broadcast their own multimedia content. Obviously, this property is not necessary for all applications, but it does represent the real interactivity as understood today. We could say that multimedia applications are no longer multimedia data + browsing interactivity, but rather involve active participation and the generation of new interactive presentations with personalised content by users, who have now become authors.

We designed our applications with the above-mentioned properties in mind, although obviously not all the applications cover all the properties for a number of reasons, such as application date, requirements, the availability of resources, etc. Below we will explain how these properties were designed and particular applications carried out.

The projects

Hudà

Hudà is the name of a multimedia application developed for the Department of Education and Culture of the Government of the Balearic Islands. It is an educational game about local literature, where users learn while playing and enjoying the game. To reach the target, two clearly differentiated parts were created:

- Virtual Library is where the game is played. There are

eight interactive VR (Virtual Reality) rooms with 40 interactive objectives that make up the game environment. Within each room, users are asked questions on the topic at hand as they browse and select interactive objects (hot spots). The aim is to go from one room to another, winning points by opening doors until you reach the final room where the major prize is located.

- Authors Library, where multimedia content is organised by author, works, life, literary movements and glossaries. It is the informational basis that provides the logic for the virtual game.

Hudà was initially developed to run on a CD-ROM platform and was adapted to a web environment two years later. *Hudà CD* and *Hudà Web* are currently sharing the same or very similar graphic design and functionalities. The differences are in the restrictions: *Hudà CD* is a single-user game while *Hudà Web* can be played by a number of people at the same time, i.e., *Hudà Web* is a multi-platform, multi-user application that can provide access to the Authors Library and game statistics, organise competitions among users and schools, store and recover matches and enable servers to manage sessions.

It is not easy to find applications initially designed for local environments like CD-ROMs that are later translated into network-distributed environments like the Web. This project features the *extensible* concept, in that there is a significant amount of sharing of engineering resources, as *Hudà Web* was defined according to the functionalities developed for *Hudà CD*.

Obviously, there are still applications, data, virtual environments and game logics on the CD environment. For the application to be extended to a distributed environment, the first step is to be able to separate the semantics, presentation and multimedia units (data) to use. Luckily, when the application was being designed, the work was split into different areas at the same time: while virtual environments were being defined, the designers created 3D environments, teachers defined the game logic and semantics of the future units and philologists generated the literature files (library) and the questions for the interactive game. This environment promoted working to the philosophy of separating data programming and their semantic data.

This initial approach meant that the semantics were kept intact, the multimedia units only required a recompression process for the new distribution media and a new storage site, while the repository would be a distributed database susceptible to being updated. The presentation software varied, although many functionalities such as those developed in Java language were used by both distribution environments. Functionalities like counters for correct and incorrect answers, the current situation, being within the virtual world, etc. were reused without any great changes. *Hudà Web* was developed for Linux, Apache, MySQL, PHP and Java.

This example shows a clear methodology when it comes to developing a type of application, i.e., the need to separate semantics, content and data presentation. It is possible to use compatible data that can easily be converted to other formats and to try to develop your own particular functionalities for an application.

MnM

Minority Newspaper to New Media (MnM) is a project of the European eContent 3316 Programme, aimed at newspapers published in Europe and written in minority or regional languages. The aim is to promote the use of these languages in new technologies, show them how to adapt their traditional products to digital technologies and make more profit from digital formats. The newspapers that collaborate on the project are *El Segre* and *El Diari* from the Balearic Islands, written in Catalan, *Primorski Dnevnik*, the only newspaper of the Slovene-speaking community in North Eastern Italy and *Hufvudstadsbladet* and *Vasabladet*, Swedish-language newspapers in Finland.

The project was originally established with two main objectives:

- Data sharing. The central and common point of all the members of the consortium are data, i.e., news stories or data that initially belonged to each producer, which are recorded, stored and represented in a particular way. The project aims to describe these data in a way that provides a common significance and entertainment for all members to be able to share and prepare different news stories.

- Multiple data presentations. With the idea of extending distribution channels, the aim is to focus on the functionality of transformation, which makes it possible to control how an output presentation document based on the initial data

(news) is created, according to the output devices, including webpages, mobile phones, paper, PDA electronic organisers and multimedia presentations.

Perhaps one of the most important objectives in the design of a multimedia application is the ability to share data on various content presentation platforms. To this end, the project first defined a data model and application that understood all the data to be processed, as understanding them at a semantic level made it necessary to provide some type of diagram.

A diagram is a conceptual box that describes the structure underlying a collection of participating elements. We used the concept of a diagram firstly to define the "vocabularies" to represent multimedia units, i.e., the types of elements and attributes of a particular type of unit, so they could be shared with all the applications and utilities developed.

One example of this concept was the application whereby journalists defined an "ownership" vocabulary to allow them to trade news stories on the Web, among all the participants in the consortium. Another was the application that generated news stories, where a set of definitions and regulations was created that told the application how to use and interpret the units.

The representation of the news stories was based on XML technology. With regard to the structure of the documents, they had to be able to deal with elements such as headlines, leads, journalist and copy. This was possible thanks to the NITF (News Industry Text Format) standard. NITF is an XML vocabulary designed to represent the information used in journalism. There was also XML News, which uses a more simple representation diagram. MnM was developed on XMLNews, in particular XML-News-Story for content meta-data.

Transformations were defined in XSLT (Extensible Stylesheet Language Transformation). The generation of outputs was done using XMLNews-Story documents and XSLT visualisation. Once the user request was gathered, the information was requested from the database, from where it returned an XMLNews-Story document and the output generated for different devices, including the web, mobile phones, a PDF output document for PDAs, printed document or paper and multimedia output.

The Web module used an XML processor that implemented the XSL language and made it possible to visualise XML documents through XSL stylesheets, checking SML documents against DTD. The specification of this processor was a W3C recommendation and was presented as a Document Object Model (DOM), while the complete specification could be found on the W3C Web.

The WAP module consisted of generating outputs for a mobile phone and focused on generating content for a WAP. Its objectives were to create WML pages, adapt content to the device and work with news stories not initially adapted to mobile phones. A WAP emulator served as a mobile browser (micro-browser) for visualising WML pages as if it were dealing with a mobile phone, making use of a simulator. We used the Nokia Wap Toolkit and Motorola Application Development Kit for the project.

The SMS module generated system output in the form of short messages. It used SMSMessage to represent the short messages to send. SMSMessage is an XML diagram that represents protocol sections for sending short messages to mobile phones. It is used to facilitate communication between an SMSC short message gateway and a service provider (SP) and allows the exchange of sections between the two to be carried out using XML documents. The SMS module was charged with generating short messages automatically. The information source was an XMLNews-Story that contained the news story.

The Doc module was aimed at generating the printed system output in PDF or RTF format. The objectives were to generate news stories in a suitable format for printing and which could be worked on with a text processor. Content generation in a printed format required transforming an XML document into a text document. The module used a JADE processor, an interpreter that implements the DSSSL presentation language (the language in which SGML representation styles are defined). An SGML document was transformed into a text document through the use of DSSSL stylesheets. This basically added an SGML document and checked and translated it to rtf, pdf, text, sgml, xml, dvi, ps or html. It was also an OpenSource tool and was available for different platforms.

This project made us aware of the real importance of using standardised forms for defining meta-data and meta-content. As well as language selection, without the use of XML it was quite hard to combine different data sources and generate so many different outputs.

Escripnauta

Escripnauta was a Web application developed for Banca March, with the purpose of organising a multimedia story competition over the Internet.

The *Escripnauta* application was used to create and assemble multimedia presentations based on a time line in a web creation and presentation environment. *Escripnauta* defines types of elements able to combine audio, animation, images and text to create a complete multimedia presentation. These elements can be created using the multimedia creation tool chosen by the user. When the user has created the multimedia story components, he or she can use Escripnauta to orchestrate the multimedia story on the time line.

The elements the *Escripnauta* application offers makes it possible to control:

- The size of the visualisation window for the multimedia story, background audio and duration of the multimedia story.
- The place where components are placed, start and end times, visual effects and transitions between the different components
- The components on the time line in the multimedia story if they had to be reproduced sequentially or at the same time, as well as the process of putting the different scenes within the complete presentation into sequence.

To achieve the independence of the process of creating a multimedia story with the available local resources, the following general guidelines were proposed:

- A user could communicate with the multimedia story server-creator using any Internet browser. The user's computer simply had to be connected to the *Escripnauta* server.
- Upon requesting the visualisation of a story, the *Escripnauta* server would be responsible for sending the multimedia story created in Flash format from the server to the user who had requested its visualisation.
- The components available for creating a multimedia story could be offered from servers. Users could use the elements on common databases, create their own elements and even offer them to other users if they liked.

In analysing *Escripnauta* we defined a syntax for specifying matters such as order, permitted elements, optional elements, elements that could be repeated within the structure, etc.

Within the time lines, storywriters could define the duration of the slide, the participating elements and their time sequences, the spatial composition and the visual input and output effects of the elements within the slide. Input forms were controlled using JavaScript, always checking they were compatible with all Internet browsers.

The fact that a document could describe itself in the face of external applications meant that the applications could process the document more intelligently. The meta-content concept was applied to a design application, as we understand that a multimedia presentation can describe itself, i.e., it has the key information for its presentation (resident on a specific server) and its resident content on any server of the resources network. This meant we could generate different processors capable of reading the presentation and generating distinct outputs on different formats and supports.

When it came to developing the application, we used various possible output formats, in particular Flash, SMIL and dynamic HTML. Users could choose the format on which they wanted to visualise the selected multimedia story. This meant that the server included three modules able to generate presentations, read multimedia stories in XML and generate the desired format.

To present the multimedia story to the user, you would have to have a plug-in, or viewer, installed. If the chosen format were Flash, this could be visualised on any Internet browser, the only requirement being that the Flash plug-in were installed. A plug-in visor is required for SMIL formats and there are no visualisation requirements for HTML formats.

The process of generating the audiovisual presentation is done in real time. Each time a presentation is requested the visualisation index is generated, so that the stories are dynamic and can be modified through to when they are published and closed. When a multimedia story is processed, the generator syntactically analyses the multimedia story with its participating elements, attributes, comments and processing instructions (effects, transitions, etc.) and then generates the output file. The file is then sent to the customer who requested it and who can then visualise it.

The success of the application is based on the formats, both those admitted for creation and the ones generated by

the audiovisual presentation. It is also fairly important to be able to share all the element resources, as well as the independence of particular browsers. We believe it is not appropriate to develop applications that only run on specific Internet browsers, as we have shown there are solutions for making multi-browser applications.

Conclusions

Multimedia applications engineering is an area still under development but at the same time already mature, although problems and questions are still arising to which there are no general solutions. Possibly the area with the greatest lack of specifications, consensus and standards has been digital video, where there is significant incompatibility between the three most commonly used architectures (QuickTime, Windows Media and Real). Today, working with MPEG offers a number of guarantees. As long as it is possible, it is appropriate to adapt applications to their corresponding MPEG standard, i.e., MPEG-1 for CD-ROM applications, MPEG-2 for DVD-Video, MPEG-4 for Internet applications and MPEG-7 for working at a more semantic level.

We have been able to develop applications under the concept of re-utilisation, generate presentations for different output devices and offer the possibility of users generating new content. The *Escripnauta* experience led us to define an extension and to offer digital video units as a component of creation, which means we can talk about an application for generating audiovisual content by content from other audiovisual content.

We used XML interfaces in most cases, making it possible to make the modules that generate output independent from the rest of the system. This shows the possibility of integrating components by using the XML standard as an interface. We also made translations between different XML vocabularies. This illustrates how systems based on different vocabularies can be integrated, e.g., digital publications based on XMLNews-Story and bibliographic management based on DocBook. It is only necessary to determine the correspondence between the two vocabularies to allow information to be exchanged between them.