Content Management for Mobile Multimedia Art Guides

Extended Abstract⋆

Augusto Celentano, Marek Maurizio, Renzo Orsini

Dipartimento di Informatica, Università Ca' Foscari, Venezia, Italia
`(auce,marek,orsini)@dsi.unive.it`

Abstract. This paper discusses issues about content management in mobile multimedia art guides elaborating on experiences made in three exhibitions held at Ca' Foscari University since Spring 2009. A family of mobile art guides on Apple iPod touch devices have been designed, featuring rich multimedia content and simple interaction styles. The guides are based on the Webkit run-time system wrapped by an iPhoneOS application interfacing an SQLite database for dynamic HTML generation and a gesture control system. A desktop application allows a content designer to configure, populate and maintain a database of multimedia material and to organize and select data components for the mobile guide application.

1 Introduction

In this paper we present a system for content management for mobile art guides. The system has been developed in the context of a project about the design and evaluation of novel interactive multimedia systems for art exhibitions, involving the Departments of Computer Science and of Art History of Università Ca' Foscari Venezia. The project has delivered so far multimedia installations and mobile guides on Apple iPod touch devices[1] for several exhibitions held in the building of Ca' Foscari University. The first exhibition, *Nigra Sum Sed Formosa. Sacred and Beauty of the Christian Ethiopia*, was held on Spring 2009, featuring artwork and testimonial objects from private collections. In the second exhibition, *Bruce Nauman: Topological Gardens* Ca' Foscari hosted a section of the USA pavilion at the Venice Biennial Art Exhibition in Summer 2009; the project is now continuing with the exhibition *Russias! Memory, Deception, Imagination* about Russian art of the XX Century, opening in April, hosting interactive multimedia installations and a new version of the mobile guide, still stemming from the original design; a spin-off of this project continuing during 2010 in cooperation with Palazzo Grassi will build multimedia catalogs and guides for the François Pinault collection of contemporary art at Punta della Dogana in Venice.

⋆ Parts of this paper have been published or accepted for publication in CHItaly 2009 and AVI 2010 conferences cited in the References Section

[1] iPhone devices share the same OS but we shall refer to iPods only for brevity.
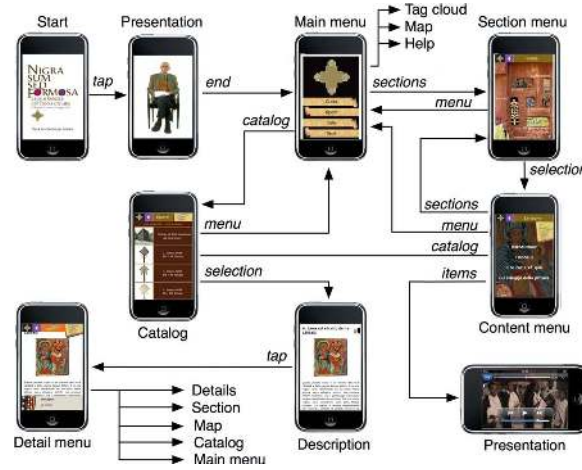
Fig. 1. Navigational structure of the *Nigra Sum Sed Formosa* guide(adapted from [3])

From the art visitors perspective the project, which is described at length elsewhere [1–4], aims at experimenting rich multimedia content on on-site installations and on mobile devices, to provide a comprehensive context about an exhibition including historical, cultural, social and environmental information, leading to a participative emotional mood with a multi-facet experience, as suggested by recent studies in the field [5–7]. From the designer perspective the project aims at building a development environment based on a multimedia repository on which multiple perspectives, multiple navigation paths and multiple interaction styles can be adapted, fulfilling the requirements of different categories of users. Indeed, very few experiences exist in real world exhibitions even in primary cultural institutions; while portable guides are commonly available, rich multimedia content is still lacking in the majority of cases, but for some experimental pioneering experiences [1].

In this paper we shall focus our discussion on the organization and management of multimedia content by presenting the underlying architecture of the mobile guides and of the content management system, discussing with reference to our experience the flexibility to derive different types of run-time models for content delivery.

Figure 1 shows the outline of the navigational structure of the iPod touch guide for the exhibition *Nigra Sum Sed Formosa. Sacred and beauty of Christian Ethiopia*, that first implemented the content management system. The guide included audio comments, slide shows, video interviews, detailed images and descriptive texts, organized according to four navigational paths: a set of sections each presenting a different theme of the exhibition, with audiovisual presentations, interviews and artwork descriptions; a general catalog presenting the exhibition artworks according to a sequential visit, each with a short description, audio comments and detailed images; a map of the exhibition rooms providing

the visitor with a spatial reference for organizing the visit; a set of keywords from a folksonomy traversing the exhibition themes and artworks with different association patterns. Hence, it was necessary to decouple the data items from their association in the guide interface and from the access paths to the collections.

2 The Run-Time Architecture

The guide design is based on a desktop web application for the organization of the multimedia repository, on the Apple X-Code development environment and an SQLite database for the content management application in the device, and on the Webkit run-time system for content rendering and user interaction management. Such an architecture was chosen primarily for rapid development constraints, since the time to design and implement the guide was very short; the whole exhibition has been set up in only two months, with iconographic and multimedia material arriving at irregular pace during the month preceding the opening date. The choice of rendering the content through an already available run-time system based on HTML and CSS standards was thus compulsory. The Webkit component of the iPod touch was the perfect choice, matching time and quality of results. It is wrapped by a native iPod application providing a finer gesture control, data management, complete support for multimedia playback in a local environment and, finally, the ability of logging the user actions, i.e., the interaction events, for on-field evaluation of the guide use.

The application has been developed using the capabilities of the native iPod touch operating system for the playback of images, audio and video, and on the embedded Webkit rendering engine for displaying HTML pages using iPod specific CSS constructs for visual transition effects. Figure 2 shows the run-time architecture of the application.

The contents are stored in the iPod as three different sets of data:

1. static pages, which are HTML files with linked external style sheets and JavaScript code;
2. files with images, video and audio content in a format accepted by the device and with a default folder structure;
3. templates and variable data stored in an SQLite database containing information on the artworks; information includes the artworks' location and thematic classification according to the exhibition navigational paths.

The application behavior is simple but not trivial: the initial page is loaded into the Webkit rendering engine, configured to use the whole screen area, hiding the navigation bars to improve the screen appearance and gain full control over the user actions. User gestures are interpreted as usual, but some of them are blocked, such as screen free orientation, to avoid misalignment of the carefully designed graphical interface.

The whole guide content is stored in HTML pages organized as a local web site. When the user requests a new page the request, identified by a link, is trapped by the `WebView Delegate` module. This module, based on the name of
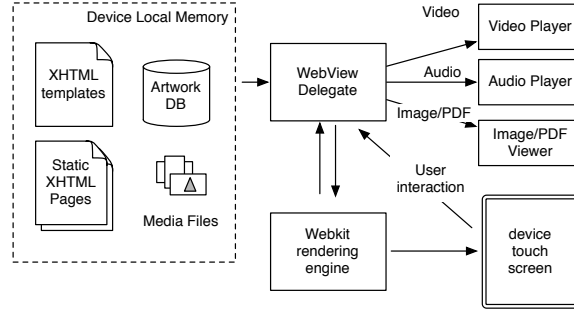
Fig. 2. The application run-time architecture

the referenced link and on the information stored in the database, recognizes static pages references and instructs the Webkit rendering engine to load and display a normal HTML page. The `WebView Delegate` module itself takes care of the following cases:

1. if the link is a reference to play a video, the `Video Player` module is invoked to display the video as a standalone presentation. At the end the application takes control and loads the appropriate next page;
2. if the link is a reference to an image not embedded in an XHTML page or to a PDF document, the image or the document are displayed. Such documents can be zoomed with usual iPod style pinch gestures. An overlaid *Close* button allows the user to return to the previous screen;
3. if the link is the name of an audio file, the file is played asynchronously; the execution returns to the HTML page containing the audio link; this behavior is typical for the audio comments in artwork pages;
4. if the link is to a page containing a list of artworks the list is recorded to allow the user to browse the relevant items in the correct sequence.
5. absolute and relative references to artworks (i.e., references by artwork numbers or by *next* and *previous* commands) are handled by merging information from the database with the artwork template, dynamically generating HTML pages that are sent to the Webkit rendering engine.

3   The Content Management System

Figure 3 shows the functional architecture of the content development environment. The external CMS interface allows the user to insert both structured data about each artwork (such as a title, a machine-readable id, a formatted description, dates, etc.) as well as multimedia information about it: a primary image (automatically processed to derive icons and thumbnails), an audio comment, and a variable-sized set of additional multimedia contents (other images, audio files, movies, textual documents, etc.). Such material is packaged in an SQLite
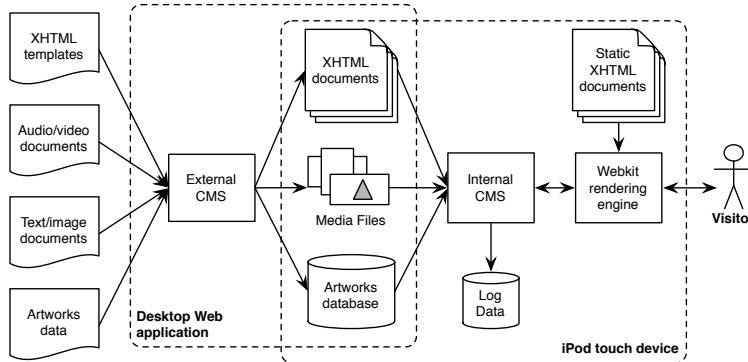
4

Fig. 3. The system architecture

relational database and in a hierarchical file system structure, which constitute the data repository transferred to the iPod device together with the SQLite database.

The external CMS takes also care of the catalogues. A catalogue represents both a physical feature of the exhibition (such as a room) or an abstract one (such as a theme or an historical period) in a transparent way. Artworks in a catalogues are arranged in sequences suggesting the proper visiting order of each collection.

The schema of the data managed by the CMS is not fixed but can be configured by the user to accommodate the needs of different exhibitions and different types of artworks. Predefined data types include text, image, audio and video in formats accepted by the iPod touch run-time.

The need for such a dynamic architecture arises from the guide organization model which must adapt to very different cases. For example, the three exhibitions so far supported are very different in their organization, reflected in different types of artwork collections supported. In the guide for the first exhibition the catalogs for themes, rooms and keywords are different, due to a mixture of themes to which the objects displayed in each room were bound, and to the intrinsic orthogonality of the folksonomy keywords with respect to the exhibition themes. The second exhibition had only one general catalog, due to the small number of large artistic installations each in a different room. In the third exhibition, for which the guide implementation is still in progress at time of writing, the themes are many and correspond very closely to the physical spaces (indeed, each room is denoted by the name of the theme). For each theme (and room) the number of the introductory presentations is higher than in the other two guides, justifying a more interrelated organization of the catalogs.

The basic artwork pages and the catalog pages are created from HTML templates designed by the interface designer. Such templates are included in the packaged application and used together with static pages which implement gen-

eral purpose content, and with the catalogs providing the different navigational structures of the visit.

Multimedia content is categorized in visual, video, audio, and textual content. All files are uploaded through the content management system so that validations on the file's content type can be accomplished, as well as any needed form of preprocessing. Images, for instance, can be automatically watermarked, textual documents can be sanitized, re-encoded, and so on.

The content management system has been designed to be a general tool to allow the representation of a large set of different artwork types and exhibitions. For this reason the underlying data model used to store artwork information is not fixed but can be updated dynamically by the CMS administrators through the administration section of the system. New fields can be added to describe additional information or a new catalogue can be created.

4   Guide Content Generation

The output of the external CMS is a set of three entities: a collection of web pages, an SQLite database, and a collection of multimedia files with a hierarchical structure. It is possible to tailor the result to obtain three different kinds of outputs to be used as a foundation for applications built around different management models.

1. The CMS templates can be used to generate a static website. The templates are designed to provide a meaningful structure that can be easily styled with classical CSS and, being human readable also in its final form, easily supports experiments on variants. This solution has the additional benefit to create fast pages that can be presented optimally on slow devices and on virtually any browser. Different stylesheets can be used to make the results appealing in a variety of different screen size, different interaction modes, and so on. This model has been used in the guide for the Nauman's exhibition due to the small number of pages and to the lack of complex navigational structures.
2. The CMS templates can be used to generate a dynamic website. We used this model in the first guide. In that case the artwork pages were dynamically generated by mixing the templates HTML code with the database contents at the iPod application level, while the catalog pages where statically generated.
3. The underlying database and the file system structure can be used as the data repository for any application, such as a native iPhoneOS application. Standard relational queries written in SQL are used to access the artwork and catalogue's data. This information is subsequently used to generate views for the application's users and to retrieve multimedia contents from the file system. This model, which requires more complex native programming but allows designers to get full control of the application, is under consideration together with the second model for the guide for the Russian art exhibition.

Fig. 4. The CMS interface: artwork data.

In all the models the resulting software is an Objective-C native iPhoneOS application deployed on iPod touch devices, which relies heavily upon the primitive Webkit HTML rendering engine of the operating system for the styled presentation of information.

The current prototype of the content management system has been developed as a web application using Rails (http://rubyonrails.org/), an open source web application framework for the Ruby programming language (http://www.ruby-lang.org/). The automatic processing of images is performed through the Ruby interface to the ImageMagick software suite (http://www.imagemagick.org/).

Figures 4 and 5 show two screenshots from the CMS prototype (some fields are omitted, other offuscated for copyright reasons); objects are described by detailed information which is used to fill the artwork pages. Catalogs are built from lists where objects are distributed according to their sequential number (defining the main exhibition path), room or section.

5 Conclusion

Other development environments are available to wrap the Webkit engine of the iPhoneOS for building native-looking web applications, the most notable being at the moment the *PhoneGap* open source development tool (http://phonegap.com). It allows also the application designer to interface, via Javascript, the device hardware to access some features of the iPod hardware. Our application environment is more limited as to the hardware control, but was specifically designed to supply, beside native media playback (which PhoneGap does not support), the complete logging of the user actions, in order to evaluate the guide usability on the field. The reader is referred to [2, 3] for a discussion of evaluation related issues.

Fig. 5. The CMS interface: catalog editing.

Acknowledgment

References

1. G. Barbieri, A. Celentano. Multimedia Technology: a Companion to Art Visitors. In: G. Styliaras, D. Koukopoulos, F.Lazarinis. *Handbook of Research on Technologies and Cultural Heritage: Applications and Environments*. IGI Global, in press.
2. G. Barbieri, A. Celentano, R. Orsini, F. Pittarello. Understanding Art Exhibitions: From Audioguides To Multimedia Companions. *DMS 2009, Int. Conf. on Distributed Multimedia Systems*, Redwood City, USA, 2009.
3. A. Celentano, R. Orsini, F. Pittarello. Towards an Environment for Designing and Evaluating Multimedia Art Guides. In: G. Santucci (ed). *AVI '10, Advanced Visual Interfaces*, ACM Press, in press.
4. A. Celentano, R. Orsini, F. Pittarello, G. Barbieri. Design and Evaluation of a Mobile Art Guide on iPod Touch. *ID&A*, n. 5&6, 2009.
5. A. Damala, H. Kockelkorn. A Taxonomy for the Evaluation of Mobile Museum Guides. In *MobileHCI '06: Proc. of the 8th Conf. on Human-computer Interaction with Mobile Devices and Services*, Helsinki, Finland, 2006.
6. A.M. Ronchi. *eCulture : Cultural Content in the Digital Age*, Springer, Berlin, 2009.
7. O. Stock, M. Zancanaro (eds.). *PEACH: Intelligent Interfaces for Museum Visits*, Cognitive Technologies Series, Springer-Verlag, Berlin, 2007.