Bridging the collection management system-multimedia exhibition divide: a new architecture for modular museum systems

Gabriele Scali [*] *and Flavio Tariffi* [#]

[*] Technical and R&D Director of Space SpA, Prato (PO), Italy
E-mail: gab@acm.org
[#] CEO of Space SpA, Prato (PO), Italy
E-mail: flavio.tariffi@spacespa.it

ABSTRACT

A major cause of architectural inconsistency in current museum software systems is the non-interoperability of Collection Management (CMS) and Multimedia Exhibition Systems. Until today, building an *integrated* museum system, i.e. a software system encompassing the two types of application, has proven to be a particularly difficult task, often leading to unsatisfactory results.

On one side, Collection Management Systems are expected to host a variety of diverse media formats, and lag behind when it comes to granting access to them in ways they were not designed for, on the other side, multimedia applications deployed inside museums are just static incarnations of ordinary multimedia productions, failing to keep up with the dynamic reality of museum collections behind them.

To effectively provide improved access and management of multimedia contents, Collection Management Systems and multimedia delivery systems inside museums must be rethought and designed in parallel, with each other in mind.

The present paper describes our efforts on this subject, the design of an integrated museum system easily accommodating new and unexpected media formats and description standards. The CMS backstage application is capable to provide active online feed to the multimedia presentation applications built relying on it. The result is a sustainable and *alive* system, heavily based on the concepts of distributed objects, scripting and skinning, easily adaptable when deployed inside a new environment.

KEYWORDS: collection management systems, multimedia exhibition systems, interoperability.

INTRODUCTION

The paradigm behind software systems currently used inside museums dates back to many years ago, when it was first understood that computers where going to be very important to cultural heritage operators.

At that time, the available hardware was generally expensive and not very fast, and programmers concentrated their efforts to satisfy the specific needs of the restricted class of users represented by

museum insiders: namely curators and administrative personnel. The software systems that emerged out of this situation, called Collection Management Systems, have traditionally been of powerful but complicated products, generally regarded as providing unfriendly interaction and poor looking user interfaces.

In the last decade, as technology, following Moore's Law, continued to become progressively more affordable and powerful, multimedia came of age. Museums were soon identified as ideal environments in which to apply the exciting new possibilities, and a completely different class of museum software emerged in the form of multimedia exhibits and kiosks: software made for museum visitors, which, on the contrary of CMS, had to be attractive, informative and intuitive.
The developers who created these first systems, however, took the less than ideal path of borrowing their approach and philosophy from the world of static multimedia presentations they were used to. The resulting systems, as found today, are very prone to aging because they live disconnected from the live museum, only freezing ideas and needs at the moment they are created. An analysis of the differences and historical factors that brought to the current situation can be found in [2] and [5].

Current museum software is still strictly derivative, at least on the conceptual level, of the above two classes of applications. Very few attempts have been made to let the two worlds cooperate, although it is generally recognized that museums would benefit very much from such an operation ([3], [1]).

ANALYSIS OF CURRENT SOLUTIONS

Most CMSs on the market today tackle the multimedia problem by adding features to accommodate rich media content as *attached* information relative to the items contained in the collections they manage. This, in most cases, translates in permitting to browse a series of pictures, or at most watch a video, while focusing on a specific collection item.

Only a few products provide instruments permitting to use the CMS content for presentations directed to the public. Some of them, as System Simulation Limited's Musims, offer an external Content Development System: a publishing pipeline able to use material contained in the CMS to produce presentations for the web, interactive kiosks, workstations, CD-ROMs or print. The pipeline permits to add new or revised content, keeping the presentations fresh, and to hold alternative versions of texts, tagging it for presentation to different audiences: general public, educational, special needs, foreign languages, etc...

On the content creation side, Hypermedia design methodologies and dedicated authoring environments have enormously developed and many of them reached a full maturity in recent years, but even the most advanced CMS systems, when trying to exploit them as a source for multimedia exhibitions, can't take advantage of this progress.

We identified a list of desirable features for a software aiming at being valuable to a museum department taking care of both the internal cataloguing and public presentation of its material:

- An expanded handling of controlled dictionaries: CMSs often store only basic information about Authors, as they are not considered part of the core museum information. Author descriptions are limited to the encoded ULAN or ICCD data fields. The software should permit to focus on them as central points of interest, as they in reality are so for visitors
- The capability to store rich media about related entities, not just Items. It should be possible to store pictures of authors, for example
- The possibility to customize displayed information for a specific audience or purpose, for example showing a picture to adults and a different version of it to children
- The general ability to store *connective* or *context* information, which is very important to Multimedia applications, think for example to introductory texts for a particular historical period or transversal information about a series of items. More generally it would be advisable to keep interdisciplinary information, in the form of context and links between different arts or historical, ethnographic, cultural, scientific, political and socio-economic context, and semantically rich enough relationships, for instance, when focusing on a painter, which authors have been similar or who influenced whom work,
- The possibility to integrate geographic mapping of the information not only in the form of location names but really geo-referenced to real maps stored in the system and accessible by the content presentation functions
- The integration of the time perspective. For example information on how a map evolve over time or on which other artists or scientists were contemporary of a given Painter
- The acknowledgement of the concept of aggregations: e.g. "schools" or "styles"

An explicit support for educational material such as tests, evaluations, games etc...

OUR APPROACH

To provide a solid conceptual framework to our integration we needed a hypermedia design methodology and we chose the Hypermedia Design Model (HDM) [9], [17] as it was one of the most mature approaches available and, at the same time, one we already knew.

Starting from the previous musings about the inadequacies of current CMSs we chained a number of consequent considerations, trying to understand what would have to be changed in a typical CMS to function as the basis of multimedia presentations inside the museum. One of our central considerations was to revise the way a multimedia object is regarded inside the system, and raise it (at least from the database point of view) to the same importance of a collection item.

- the CMS database must be broadened, some of the elements that were considered context (e.g.

authors) become central, at the same time much more context information must be freely permitted
- this context must capture and link relationships with items and data which lives also outside the single museum, this concept is very important for both scientific and educational purposes, for the concept of museum networks and for the socio-economic implications of the museum as a link to its own territory [16],
- this information is very difficult to identify a priori. Trying to capture it in the data structure would be short sighted; it must be based on an adaptable mechanism, permitting extension by adding metadata: in this area XML immediately seemed the right solution,
- the relationship between rich media objects and collection items should be a many-to-many. Not only a collection item could be represented in an unlimited number of multimedia objects but also one MM object could well be relevant to different museum items and topics,
- the relationship semantics must be enriched with a some information about the *role* the object plays in the MM, its relevance, meaning etc...

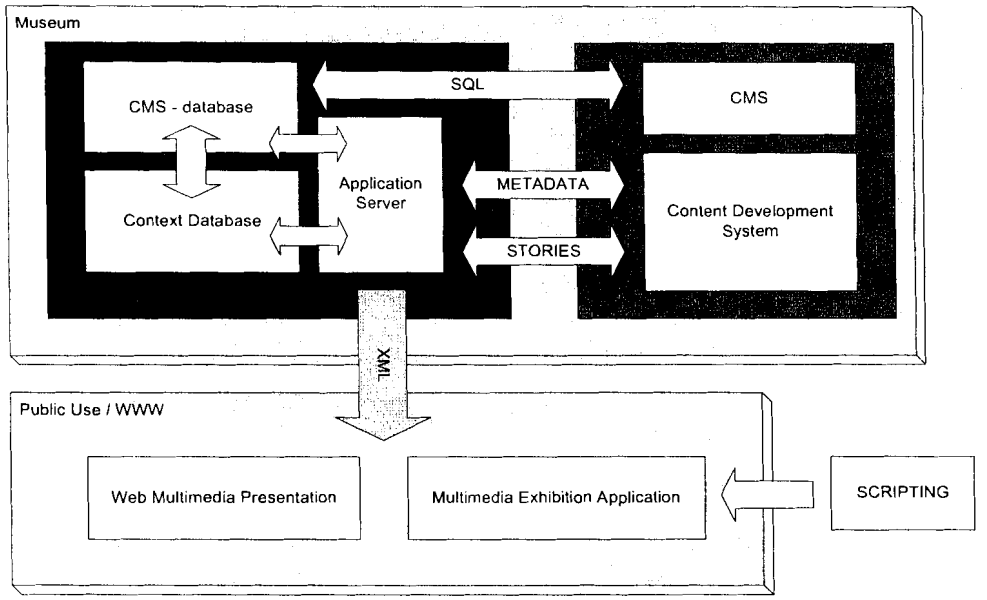
With these topics in mind, we looked for a design that would tackle the twofold intent of permitting metadata extension to the CMS information with context information and to hybridise an HDM-like approach with current CMSs' databases operations.

For a number of reasons, we wanted to enhance the information contained in the CMS without building a monolithic repository: we wanted to the preserve subsystems' modularity, performance, ease of realization and maintenance, ease of integration and data interchange with other systems. As a result, we decided to make as little modifications as possible to the CMS database itself. Instead, we implemented into it software hooks to permit to externally leverage the system without interfering with the operations of regular CMS clients. Then we designed a dedicated Application Server: a piece of middleware capable of enriching the content of the DB with semantic information based on the Dublin Core Initiative and the CIDOC CRM ([12], [7]), providing XML output to feed the creation of HDM-like entities and / or collection thereof.

With this architecture, the existing CMS client server system was left almost untouched and could still be deployed independently from its multimedia counterpart.

The application server must be able to keep enough information to satisfy a request from a client willing to know something on Picasso with: a Biography, a list of everyday life pictures, a list of paintings and their thumbnails (for further data one would go refer to the item itself), a list of related geographical locations, a list of used techniques, some historical data such as a timeline, a list of related painters, questions and answers for educational purposes etc... All these information should be dynamic, i.e. if a related Item or more context information is added to the database the client MM application would reflect the change.

The whole system is designed to use the tightly integrated distributed objects architecture depicted below:



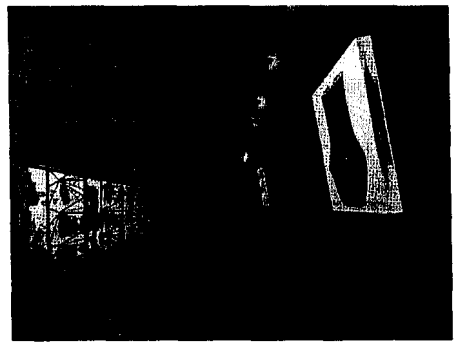
The backstage (dark blue) is represented by a common relational database containing the CMS database paired with another relational or object-oriented database storing context (meta)data. They both connect to an application server providing a series of high-level conceptualization services leveraging both the collection data and the metadata. On the curator side (light blue) we find the CMS, which keeps working as usual in cooperation with its own relational database but could in principle be modified to take advantage of the application capabilities in the future. The content editing system also resides on the curator side, permitting to manage the context information. The content editing system will permit to add stories (see below), create new entity types, define their associated metadata and handle associations between them and the existing information. In general the content editing system permits to do all the background work necessary to *decorate* the CMS contents and make them usable by multimedia presentation applications.

INNER WORKINGS

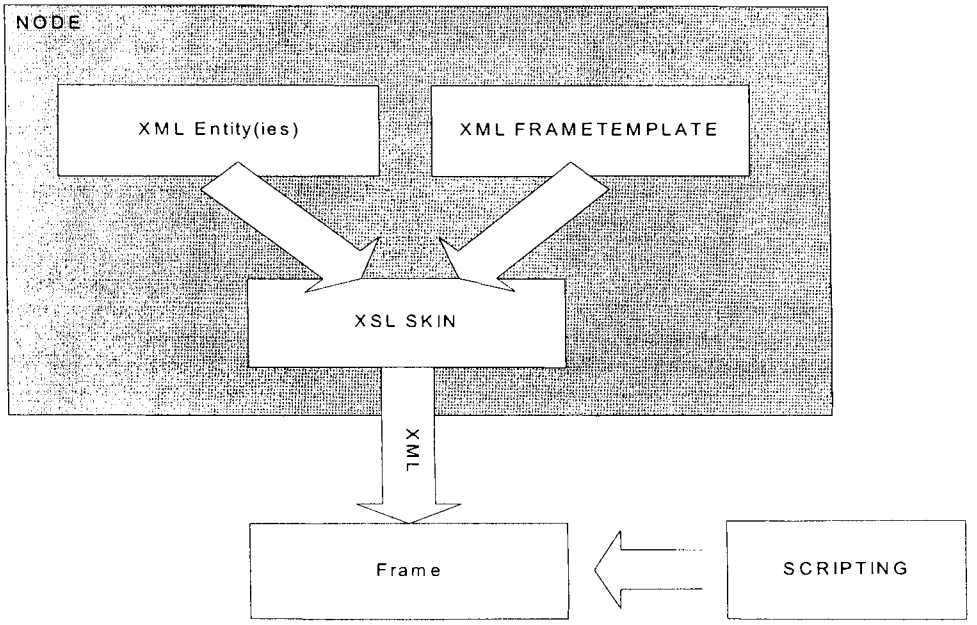
Let's examine in more detail how multimedia content is dynamically generated from the Application server

and gets incorporated in a client presentation, for example a website. Let us say, for example, that a museum containing information about local artists and their work wants to create a presentation in which these topics are presented in the perspective of regional history, folklore and typical products and food. First, a set of new entities types should be identified and modelled in terms of XML. Entities would then be inserted in the context database along with any multimedia information regarding them. These new entities would be connected to the information contained in the CMS (for example associating authors to their preferred food), ready to form the basis for the production of a multimedia presentation.

The multimedia presentation would later get the entities from the Application Server, and change them, by means of a series of XSL transformations, into a presentation for the web or into a customized VRML environment or again into an XML file driving a presentation hosted by the proprietary multimedia player our company is developing as part of the same integrated system. This would all happen in real time, so if a better picture of a certain vase would be introduced into the system, all the multimedia presentations using it would soon reflect the change.



To give a basic overview of the inner workings of our design, the following is a short description of the main concepts used inside our system:



EntityType

Defines the semantics of an entity (e.g.: Painter, Painting etc..). It is expressed as a DTD which can be used to validate entities of that EntityType. All EntityTypes are inserted in an ontology framework like the CRM (CIDOC Object Oriented Reference Model) [7], this semantic unity permits to exploit the richness of content in the database and opens the possibility of heuristic exploration of the represented conceptual domains.

Entity

Is an instance of an EntityType as provided by the Application Server in the form of an XML document. It defines an object with related information, and eventually provides (part of) the data for presentation. Collection Items are obviously entities of a particular type depending on what they represent, but they are not the only entities provided by the CMS: authority files also provide data that constitutes entities, as does much of the contextual information kept by the application server and not inherent to collection objects. Together with a FrameTemplate and a Skin, an Entity or a collection of entities completely define the Frame by an XSL transformation.

Structure

Actualises the navigational structure of a multimedia presentation, contains a series of Nodes connected by structural or semantic links much like HDM. Structure also provides default navigation mechanisms such as history stacks and map.

Node

Provides the unit of navigation in a multimedia application. It is associated with an Entity or collection of entities, a FrameTemplate and a Skin. It uses them as inputs and performs an XSL transformation to actualize all the necessary data to create a Frame.

Frame

It is a screenshot or series of screenshots, a situation, which actualises the navigation to a Node for the user. The Frame will contain a number of user interface components, which will play the role of *actors* of the presentation. Our design includes an interface specification for components that want to be usable in our system. As new and more sophisticated components are developed as standard COM or .NET components in conformance to our interface, they will add to the expressiveness of the possible presentations. Frames are also capable of taking full advantage of multimedia web technologies and can display DHTML, Flash, QTVR and anything possible on a web page. This makes possible (but not mandatory, when necessary to take full advantage of the power of off-line technologies) to express, with a single authoring action, a system deployable both inside the museum and on the World Wide Web.

FrameTemplate

A FrameTemplate is an XML document defining the static appearance and interactive objects contained in a Frame. It also defines the Frame mechanics, either statically or parametrically respect to presentation information. FrameTemplate is the connection point to the ability of scripting the appearance and mechanics of the presentation from outside the system. The scripting interface can be used with a standard scripting language, like Javascript.

Skin

An XSL document containing actualisation information, which combines an Entity or collection of Entities with a FrameTemplate and defines the final look of the Frame as it will be presented on screen. It is possible to change the characteristic of a presentation by simply adding new Skins. Ultimately, a Skin is a package containing parameters for the visualization of user interface components, graphics and directions to apply them to a particular user interface; it permits to completely change the look and feel of an application independently of the functionality. The same presentation will have a completely different look and feel depending on the skin: this will be very useful, for example, to customize the same material for different audiences or presentation devices. Children will prefer a nice colorful interface featuring their favorite cartoons, a presentation to be used with a touch screen or seen by sight-impaired people will need larger buttons, and so on.

Story

In traditional multimedia applications the role of 'what' the application has to show is given to the storyboard, a rather rigid piece of information cast on paper or

some other electronic form. In our idea of a dynamically fed multimedia application, the storyboard leaves way to the more abstract concept of Story. A Story is a particular piece of metadata representing a series of items and connective tissue made to link them ad-hoc (in the forms of texts, other rich media, relations), it defines an ordered collection of topics and information items, which can be specified either directly or by means of their attributes. If we do not link a presentation *directly* to the description of an item we gain the possibility of changing a parameter, say language, and have the same Story told another way. If more pictures of an object are introduced in the collection management system, and we linked a guided tour presentation to "the whole set of pictures available for that given item", then the presentation will be automatically updated with the new data.

CONCLUSIONS

The shift of importance and role of cultural institutions in the net economy must be fostered by the growth of a new generation of software systems that would take in consideration the strong *integrative* and *interactive* nature of the new approach. These systems will present a richer, dynamic and more enjoyable experience to the public (well fitting the web spirit) while providing all the necessary power and expressiveness to curators. The software architecture sketched here is an attempt to turn the tables in this direction and we hope it will start a profitable new thread of discussion. Our future work will be in the direction of refining and enriching our current design, especially in the direction of user modeling, context sensitiveness and intelligent content customization.

REFERENCES

1. Howard Besser, *Integrating Collections Management Information into Online Exhibits: The Web as Facilitator for Linking Two Separate Processes* in David Bearman and Jennifer Trant (eds.), Museums and the Web (selected papers from the '97 conference), Pittsburgh: Archives & Museums Informatics, 1997, pages 201-205

2. Howard Besser *The Transformation of the Museum and the way It is Perceived* in Katherine Jones-Garmil (ed.), The Wired Museum, Washington: American Association of Museums, 1997, pages 153-169

3. Jim Blackaby, Beth Sandore *Building Integrated Museum Information Systems: Practical Approaches to Data Organization and Access* Museums and the Web Conference, March 16-19, 1997-Los Angeles

4. Leslie A. Carr, Wendy Hall, and David C. De Roure *The Evolution of Hypermedia Link Services* in ACM Computing Surveys, Symposium on Hypertext and Hypermedia, 2000

5. Vanessa Castellano *Museums and Multimedia* Network Museum and Magazine Project International, Tokyo, 1998

6. S. P. Christodoulou, G. D. Styliaras

and T. S. Papatheodorou *Evaluation of Hypermedia Application Development and Management Systems* ACM Hypertext '98, Pittsburgh, PA, USA, June 20-24, 1998

7. CRM: CIDOC Object-Oriented Reference Model *http://www.geneva-city.ch/musinfo/cidoc/oomodel/*

8. Mark Dobie, Robert Tansley, Dan Joyce, Mark Weal, Paul Lewis, Wendy Hall *A Flexible Architecture for Content and Concept Based Multimedia Information Exploration* David J. Harper and John P. Eakins (eds), The Challenge of Image Retrieval, Newcastle - GB, February 1999

9. Franca Garzotto, Paolo Paolini *HDM - a Model for the Design of Hypertext Applications* in Proceedings of the Third ACM Conference on Hypertext, 313-328, 1991

10. Richard Light *Presenting XML* Sams Publishing1997

11. Maria Milosavljevic, Robert Dale, Stephen J Green, Cécile Paris and Sandra Williams *Virtual Museums on the Information Superhighway: Prospects and Potholes* In Proceedings of CIDOC'98, 10-14 October 1998. Melbourne, Australia

12. Gabriele Scali *Beyond the multimedia database concept: a Cultural Heritage-specific Application Server* in Proceedings of the EVA2001 Conference, March 26-30 2001, Florence, Italy

13. Werner Schweibenz The 'Virtual Museum': New Perspectives For Museums to Present Objects and Information Using the Internet as a Knowledge Base and Communication System University of Saarland, Germany, 1998

14. Georgios D. Styliaras and T. S. Papatheodorou *Defining a flexible methodology for supporting automatic generation of structured hypermedia content in the Web* AACE WebNet 2000, San Antonio, USA, October 31-November 4, 2000

15. Georgios D. Styliaras, T. S. Papatheodorou *Using a data-centric approach for developing thematic-based collections in the Web* AusWeb2K-The Sixth Australian World Wide Web Conference, Rihga Colonial Club Resort, Cairns, 12-17 June 2000

16. Flavio Tariffi, Gabriele Scali *Putting museum networks to work: innovative solutions for thematic and territorial clusters of smaller museums* Alcalà 2001 Conference July 9-14 2001, Alcalà de Henares, Spain

17. Lewis P.H., Hall W., Carr L. A., De Roure D. C., *The Significance of Linking,* in ACM Computing Surveys 31(4), December 1999

ACKNOWLEDGEMENTS
The present paper has partially been produced in the frame of the EC financed research project IST-2000-25136 *OpenHeritage*.

ABOUT THE AUTHORS

Gabriele Scali is Director of R&D at SPACE S.p.A., where he currently leads several international research projects. He is one of the software engineers who pioneered the introduction of the object-oriented paradigm to the Italian software scene in the early 90s. His interests ranging from cognitive science to the theory of complex adaptive systems, Scali has formerly been involved in fields as different as biomedical and building science software. He can be contacted at *gab@acm.org*

Flavio Tariffi is the CEO of SPACE S.p.A. and Managing Director of Atlantis S.p.A. After humanistic and computer science studies, he deals since several years with IT for the Cultural Heritage and is now in charge of promoting and marketing high-tech solutions and in charge of several major national and European projects. He can be contacted at *flavio.tariffi@spacespa.it*