Museums and the Web 1998
A&MI
OVERVIEW
ATTEND
WORKSHOPS
SESSIONS
SPEAKERS
ONLINE
EXHIBIT
SPONSOR
BEST OF THE WEB

# Papers

## Adapting Museum Structures for the Web: No changes needed!

**Jorge Gustavo Rocha, Jose Carlos Ramalho, Pedro Rangel Henriques, Jose Joao Almeida, Jose Luis Faria, Mario Ricardo Henriques, Universidade do Minho, Portugal**

## Contents

# Introduction

This work is being developed under the GEIRA project, an EC supported project, under the INTERREG II program. The goal of the GEIRA project is the development of a service to support the publication of multimedia information, with respect to science and technology, cultural resources and environment protection in the of North of Portugal and Galiza (North-west Spain). In this paper we will focus on part of the work being done to provide the museums of this region with tools for data management and methodologies that enable the publication of their data, preferably on the Web.

Most of the existing museums in this region are small with little or no computer-based information support. Only a few have the opportunity to be assisted by a software engineer. Our goal is to get an information system in each museum, running with the minimal external support. It should be as easy as possible to publish information on the Web from that information system. Each museum will have a presence on the Web, as a stand-alone entity.

The aim of project GEIRA is to provide an additional value to the users searching for information. Instead of iterating through a set of institutions, the user can search and browse in a knowledge level above each particular institution. GEIRA will provide that knowledge level, rather than just aggregate a collection of links in well organized site.

### About the title

The title of this paper reflects our search for knowledge representation models to be applied to museums. We want models rich enough to be applied for both the daily data management and for multimedia publishing, like CD-ROMs and the Web. The knowledge must also be suitable for building relations between heterogeneous sources, like different kinds of museums.

In this paper, we will present two different approaches for representing museological information: one based on relational databases and the other based on annotated documents. For each, we will try to identify the advantages and the difficulties, resulting in a trade off that should be analyzed before deciding on one of these models. We have found that annotated documents are a suitable data representation for unstructured information, like long descriptions of objects, of archaeological sites, biographies, etc.

### Structure of the Paper

This paper starts by identifying our goal: the development a system that will minimize the museum's effort in order to maintain a presence on the Web. In order to do that, we will analyze two different models of data representation: the relational model, and the use of annotated documents. The relational database model is presented more briefly because it is traditionally used and is already well understood. The second approach will be presented in more detail, with examples of what and how to explore the semantics embedded in the annotations.

## Relational Database based Knowledge

In the context of GEIRA, there was no specific application to support the daily data management activities of the museums so, the first step was the development of one. The design and the development of this application was done for the Windows operating system. The application was written in Delphi, and the data is stored in a relational database. This application runs, typically, on a single or small network of PCs.

For the definition of the data structures, we followed, as much as possible, the SPECTRUM [Ass97] recommendations. During the development, we had the support of people from the museums. At this time, the application is ready to be installed, and only minor adjustments are expected. In the next weeks, we will evaluate the application, in daily activity. Meanwhile, we are preparing a technical report about the application built, considering all design and implementation issues.

For the purpose of this paper, we refer this application as a practical solution to fulfill the needs of museum data management. This application was not written overnight, and some problems had to be solved. But it was not far from the usual development of an interface over a database (the definition of the data structures, the tables of the database and their relations, became easier using SPECTRUM, as we said). The development of this kind of application, built on top of a relational databases is easier due to the set of existing sophisticated and affordable tools, like the Delphi and SQL Server used. This means that, when deciding for this model, we can expect the existence of a rich set of tools.

### The first Web site

To make possible the presence of this variety of museums on the Web, in a systematic way, some alternatives were explored.

Another team of this project, working on Vila Real, developed a common abstract structure for the museums [LC97]. Each site is then constructed to instantiate that structure.

The root of each museum starts in a atrium, and then is divided in 5 subsections: collections, activities, free theme 1, free theme 2 and contacts. The pages generated for each section are structured with several frames, each with a specific functionality. This mapping between the abstract structure and the concrete HTML pages was done by a multimedia designer.

To make this model systematic, almost all the information is stored in a relational database, rather then on HTML pages. The pages are dynamically created using Microsoft Active Server Pages. The construction of a new site consists of filling in database fields (with text, images, etc.), rather than writing and composing HTML pages. In this way, the task of building a new site is faster and more accurate. Maintaining the sites is also easier, and can be made by people aware of HTML details.

## Annotated Documents based Knowledge

From the beginning, in parallel, we started the study of SGML [Her94], to learn how useful it could be for representing and reasoning on museum data.

The study we are reporting in this section was carried out to deal with archaeological sites, where museum objects were found. It is being used in a particular museum with thousands of archaeological artifacts. That is why the data (documents) we will use to exemplify some SGML concepts are mostly related to archaeology.

This data is entered by archaeologists, rather than by the museum staff, but will be available through the museum. The archaeological data will be cross-referenced against information associated with the objects available in the museum collections.

Cross-referencing the particular information about a museum object, with the information about the archaeological site where it was found, can be useful to create a framework where the full context of that piece can be explained. The precise location of the piece, and even all the archaeological site, can be seen through a GIS (Geographical Information System) plug-in.

## Writing SGML Documents

For this approach, the data should be written as SGML documents. This can be created either by transforming the original documents collected by the archaeologists (in some other format) into SGML, or asking them to use SGML. After showing them how to work with a DTD driven editor, and the benefits of automatic syntax validation along with structural manipulation of text within the editor, it was easy to have archeologists adopt the latter approach.

Because they used Winword, we have adopted SGML Author from Microsoft, but soon it was clear that it was not a very good option. We changed to Word Perfect, also affordable, which supports SGML more conveniently. Word Perfect has lots of interesting features: automatically highlights violations to the DTD structure, computes a list of the valid choices dependent of the cursor position, has support for tagging non-structured documents, can ask automatically for the values of required attributes, etc.

The SGML documents are then processed in order to execute, at least, two tasks: to check additional constraints, and to generate HTML. The additional constraints are necessary to ensure consistency among the data. This validation task is only possible when we check the contents of a document compared to others. This validation task and the related discussion about assurance of quality is reported in detail in [RRAH97].

As said above, the second result of processing the original SGML documents, is the generation of the HTML pages. This is not an assisted step; this is completely automatic, which was one of our strongest requirements. Being automatic does not mean that it is a blind process. We can introduce as much intelligence in this step, as there is knowledge to do so. That knowledge depends on the design of the DTD, and on how the text is being tagged (with more or less detail).

## Generating HTML pages

To do the generation of HTML we had to choose an SGML processing tool. An SGML processing tool can have two "operating modes": transforming and formatting. Although, it seemed that for this task we only need a formatting processor, to meet our goals we also need a transforming processor (to produce different structured views of the data).

We compared the tools available in order to choose the best for our intended use:

**Perl - sgmlspl.pl and SGMLS.pm [Meg95b], [Meg95a]**

> has the advantage of being freely available; has major drawbacks if you move deep into transforming; programming gets highly complex (ex. processing sub-DTDs).

**Omnimark and Balise**

> two commercial tools more or less equivalent; the major difference are the conditions of acquisition; Omnimark made a light version freely available that can be used in small to medium projects.

We chose Omnimark [Omn96] and we are generating HTML with OMNIMARK scripts. OMNIMARK is a complex processor, focusing on SGML processing. In our case we are just using a small subset of its functionality, mainly 'Down-Translation'.

Example of a simple script to generate an HTML list of all the entries in the archaeological SGML file:

```
DOCUMENT-START

  OUTPUT "<UL>%n"



ELEMENT IDENTI

  OUTPUT "<LI>%c%n"
```

```
        ELEMENT #IMPLIED

          SUPPRESS



        DOCUMENT-END

          OUTPUT "</UL>%n"
```

In this script, before the processing starts ("DOCUMENT-START") we open an HTML list; during processing if we find an element "IDENTI" which identifies an entry we generate a list item ("`<LI>`") with its contents ("`%c`"); for all other elements we may find, we will ignore them ("SUPPRESS"); when we reach the bottom of the file we close the HTML list.

At this point, any person new to SGML document processing can notice a major advantage of keeping documents in SGML. Since we can define DTDs and maintain information according to those, we have a richer format. It becomes very easy to generate a set of different HTML pages for the same SGML document. Those HTML pages can reflect the structure of the source document or have completely new structures. For example, in the above script, we could collect the "IDENTI" elements into an associative array, sort this array, and generate a sorted list of entries (although we are not modifying the structure we are changing content order).

As a more sophisticated example, we could want a new document having all the entries grouped by geographical areas ("`<CONCEL>`" - in our SGML files). This would imply reordering and restructuring of the source document.

```
  DOWN-TRANSLATE
```

```
global stream area-stream variable initial-size 0

global stream temp

global stream concelho



ELEMENT #IMPLIED

  SUPPRESS



ELEMENT arqueo

  OUTPUT "%c"



ELEMENT identi

  OPEN temp as buffer

  PUT temp "<li>%c%n"

  CLOSE temp



ELEMENT concel
```

```
  OPEN concelho as buffer

  PUT concelho "%sc"

  CLOSE concelho

  DO WHEN !(area-stream has key concelho)

    NEW area-stream key concelho

  DONE

  REOPEN area-stream key concelho as buffer

  PUT area-stream key concelho temp

  CLOSE area-stream key concelho



DOCUMENT-END

  REPEAT OVER area-stream

    OUTPUT "<h2>Entrys of  '" ||

           key of area-stream ||

           "':</h2>%n<ul>%n%g(area-stream)</ul>%n"

  AGAIN
```

Moreover, if we distinguish specific visitors, we can generate pages on the fly, according to user attributes, like level of expertise, etc.

As another example, if we want a LaTeX version of our documents we just have to write a script to do the job.

The thing that should be reinforced here is that we write all these scripts once. The documents to which they apply can vary in their contents but the scripts will remain functional. If we keep the structure (we do not change the DTD) all the processing remains stable. at the other end, if one of the formats that are being used to present our information (HTML, LaTeX, ...) is upgraded, we only need to change the scripts to reflect this. We do not need to go through all our documentation upgrading texts. SGML is standard and platform independent and that is the major advantage we expect to take from it.

Of course, there is the effort of making the scripts. But importantly this effort is to be taken by us now, to enable each museum, in the future, to work with our minimal support.

## Tools: Search, thesaurus and encyclopedia.

In fact, our interest in SGML has to do with document reasoning. And the reasoning carried out is to provide the information requested by sophisticated users, who are not interested in quantity. Answers "in quantity" can easily be obtained through several blind automatic keyword indexing engines (some more blind than others). Even these search engines are trying to be more adequate for users not so impressed with "more is better" and their sophisticated technology, but really searching for specific things.

The main goal of this section is to show how to profit from the structure of the SGML annotated information in order to see it as a knowledge data base capable of inference. We will also show how to exploit meta information in the process of building new documents (ex. html pages), and building new tools (ex. browsers, search engines).

The definition of the DTD and the tagging process, associates a type to each element tagged. The element tag (and sometimes the attributes) indicate the type of the information.

In order to build a knowledge database with the different sorts of information, a classification structure is necessary. In our case a thesaurus will be created.

### Building a Thesaurus

In order to establish relations between the different kind of objects. The thesaurus will:

- establish relations of equality and normalization (alternate or UseFor terms)
- define relations of being a particular case of (isa relation)
- define some properties related to a term
- establish other relations between terms (writers write books, etc.)

The thesaurus is a important tool to define relations over heterogeneous sources of information participating in the GEIRA project (museums, etc. ) and it is a way to reconcile different classification strategies.

A browser and search engine over this heterogeneous information, with some conceptual structure, will work like an "encyclopedia".

### Building an Encyclopedia

In this context an "encyclopedia" should be defined as a view over the information, and also as a navigation tool.

The encyclopedia contains terms and associated information, which can be:

- types and their relations (from the thesaurus)
- instances and associations to the provider, information sources, pointers to the document source, the context in which the term appeared. (Typically, many instances for each specie)

## Conclusions and Future Work

The adoption of SGML by the people entering data was a easy step, as good editors that give the authors promptly feedback, in a WYSIWYG environment are available. These editors made entering SGML documents as easy as entering any other unstructured document, but with the benefit of constructing a structured document. In our study, the archaeologists do prefer to enter SGML documents, instead of unstructured descriptions, because they prefer some assistance to ensure structural and some content validation.

The SGML approach, for some daily data management, is not as adequate as the relation database. This is because there is already a rich set of sophisticated tools available to implement the applications relying on the underlying data model. These tools have been incrementally developed in the last decades.

The relational database model, however, does not adapt well when we try to use it for less structured data, with textual characteristics, and with the notion of sequence. This kind of data is frequent in object's descriptions, in discussions of their importance, in the context where the objects were made or discovered, in biographies and so on. If the data are less structured, how can it be incorporated in a fixed structure? If there are many variants, the relational model also tends to grow in the number of fields, many of which will be null in a particular instance. In textual description, we read a clear sequence of words containing meaning. Putting discrete elements in a database, we loose the sequence of the elements. Due to this limitation of the relational model (mainly when managing less structured data), we usually create memo fields big enough to store the textual descriptions. All the information can be stored in memo fields, but the only thing to do with it is to store and retrieve all the field as a block.

The SGML standard is suitable for documents with less formal structure, enabling further processing and reasoning. It is possible to manipulate parts of the data, to build relations between parts of each, etc. The drawback of this model is that this processing and reasoning does not come without a price, in the sense that this processing must be programmed. There are sophisticated tools to process SGML documents, but these are very expensive, and require some practice to get useful results. SGML has also advantages over the relational model in the preliminary stage of data manipulation, when a structure is not yet clearly defined. We can only start to work on the relational model, after that model is built. The SGML approach accepts an incremental refinement.

### Future Work

In this stage, we need to evaluate and test as many tools and systems as we can for SGML processing. We are also considering the possibility of using just XML, as it seems to be powerful enough for our purposes, without some of the difficulties of SGML which is a more general standard. It seems that XML is being well accepted in the community.

From a more scientific point of view, our work in the short term is the investigation of the combination between the relational database model with SGML. In this kind of architecture, we would have the usual fields of the relational tables, but some of them containing annotated data. To take advantage of the annotations, some improvements must be made in the relational engine, enabling processing of the fields containing SGML, according to the respective DTD, or even without a DTD.

## Acknowledgments

The authors would like to thank Prof. Alberto Proença, responsible for the GEIRA project at University of Minho. We also would like to thank the contribution of the Vila Real team, working on the same project. We also acknowledge all the help received from the archaeological unit of University of Minho.

The GEIRA project is supported by the EC INTEREG II program.

Finally, we thank the organization of MW'98, for their work and the opportunity to share our work with other institutions.

## References

**Ass97**
Museum Documentation Association. Spectrum: The UK documentation standard, 2nd edition. Technical report, 1997.

**Gro95**
CIDOC Archaeological Sites Working Group. CIDOC core date standard for archaeological sites and monuments. Technical report, 1995.

**Her94**
E. Herwjnen. *Practical SGML*. Kluwer Academic Publishers, 1994.

**LC97**
Leonel and JosÈ Bulas Cruz. Motor para criaÁão de sÌtios web para museus. Technical report, 1997. In Portuguese.

**Meg95a**
D. Megginson. sgmlspl: a simple post-processor for sgmls and nsgmls. Technical report, Dep. English - Univ. Ottawa, October 1995.

**Meg95b**
D. Megginson. Sgmls.pm: a perl5 class library for handling output from the sgmls and nsgmls parsers. Technical report, Dep. English - Univ. Ottawa, Canada, October 1995.

**omn**
Defining microdocument architecture. Technical report. http://www.omnimark.com/white/microdoc/microdoc.html.

**Omn96**
Omnimark. *Omnimark Programmer's Guide*. Omnimark Technologies, 1996.

**Roy**
Bruce Royan. Quality control in electronic networks. Technical report.

**RRAH97**
José Ramalho, Jorge Rocha, José Almeida, and Pedro Henriques. Sgml documents: Where does quality go? In *SML/XML'97 Conference Proceedings*, pages 171-177, Washington, 1997.

**Sev95**
Eric Severson. The art of sgml conversion: Eating your vegetables and enjoying dessert. Technical report, 1995. WHITE PAPER 4001-II.

**Smi97**
Alastair Smith. Criteria for evaluation of internet information resources. Technical report, 1997. http://www.vuw.ac.nz/ agsmith/evaln/index.html.

*Last modified: 09/28/2004 16:39:52 This file can be found below http://www.archimuse.com/mw98/*
Send questions and comments to info@archimuse.com