A multimedia database supporting a generic computer based quality management system

Alberto Silva[*a] , Gabriel Andrade, José Delgado

INESC - Instituto de Engenharia de Sistemas e Computadores[**]

Rua Alves Redol, 9

Apartado 13069 - 1000 Lisboa

Abstract

Nowadays, quality management systems (QMS) are a main aspect of modern organizations. Basically these systems manage their business information processes (such as generic documents, quality records, templates, and procedures description) based on several media (text, images, audio, etc.) giving different groups of users different capabilities and visions of the system.

The World Wide Web (WWW or Web) is a real success and an important factor in the current computer and telecommunication fields. It is being used mainly to published and retrieve information in a world wide scale. However, some projects in developing local, or internal "web application" ("Intranets") have already started. Basically, this kind of application will support the new need for collaborative and cooperative work inside an organization, or even inside a restricted set of organizations.

This paper presents an experience in designing and developing a generic and configurable quality management system using the WWW technology. Database characteristics and problems involved with this novel kind of application development will be stressed from the multimedia database and the WWW point of view. Namely points concerning the integration of different information media, meta-databases, transaction mechanisms, and user interface limitations will be focused.

Key Words

WWW, Multimedia Database, Quality Management System, HTML, CGI, User Interfaces, OMT.

[*] Phone: 310 03 07 - Email: Alberto.Silva@inesc.pt (INESC - Grupo de Serviços e Sistemas Telemáticos)

[a] This work was partly sponsored by the Consortium agreement between INESC-CFS and Partex S.A.

[**] Telefone: 310 00 00 (General) - Fax: 314 58 43

1 Introduction

Nowadays, the majority of the manipulated and transmitted information is digital. This information is represented by different media (e.g. text, image, graphic, audio, ...) and their respective formats (e.g. ASCII or ANSI for text data; BMP, GIF, or JPEG for image data; etc.). Actually the Internet is the main transmission vehicle of the majority of this (digital) information, due mainly to the success of the World Wide Web [3] paradigm.

Using the technology involved in the Internet, some services and specific applications concerning the organizations have started to be developed [4,5,6,34,35,36]. These services should support internal characteristics of the organizations, in a way to provide computer support collaborative and cooperative work (CSCW) [7,25], and also to extend their legacy applications. This new class of services integrate a wide area of technologies, that spans from the data model design [8,26,27,28,29,30] and construction of databases, to the use of specific communication protocols (such as electronic mail, file transfer, etc.), through the definition (authoring-in-the-small) of multimedia objects using different descriptive languages and formats [9,22,23,24], or through design and authoring (authoring-in-the-large) hypertext applications [31,32]. To this class of services and systems, which use the technology involved in the Internet, but whose goals are to handle and give support to the internal specificities of one or a co-related group of organizations, we will call "Intranet" in opposition to "Internet".

Modern organizations, due internal and external forces, have been improving their management and productive processes. Consequently, they have been looking for the introduction of some quality norms [1,2], either in the development process of their principal products (or services), or in all their productive and management processes. Assurance quality processes are complex, involve a great quantity of information, involve the majority of the human resources of the organization, and should be dynamic, flexible and configurable to the different realities of organizations.

This paper presents the goals, the design, and some implementation issues of the Web-Q system (from ***Web**-based **Q**uality management system*), is a generic computer based quality management system (QMS) developed in the context of the "Intranet" services research area. Databases characteristics and problems involved with this novel kind of application development will be stressed from the multimedia database and the WWW point of view. Namely points concerning the integration of different information media, meta-databases, and user interface limitations will be focused. Figure 1 depicts a snapshot of one screen (generated HTML page) of the system.

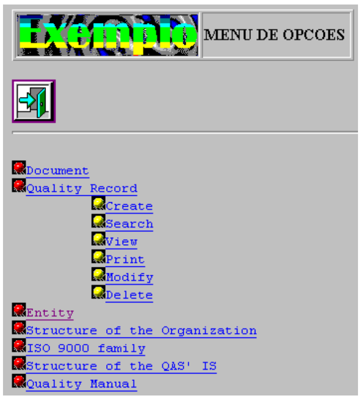


Figure 1: A sample screen (page) of the Web-Q system (text in Portuguese).

The rest of this paper is structured as follows. Section 2 presents the goals and basic principles underlying the Web-Q system development. Section 3 describes the conceptual design of the main object diagrams. Section 4 raises some questions about possible ways in describing and storing multimedia information and presents our decisions in the context of the WWW technology. Section 5 presents and discusses some implementation issues. Finally, conclusions, open questions and future work are presented in section 6.

2 Goals and principles of the Web-Q system

The main goal of the Web-Q system is to give computational (and digital) support to quality assurance processes of the involved organizations. We will name Web-Q as a computer based quality management system (QMS). The system should be driven by the following general principles:

- To be configurable in order to adapt to different organizations;
- To provide a common graphic user interface, easy to use and to understand;
- To be (conceptually) independent from any specific platform, or database management system.
- To support collaborative and cooperative work. This means the system will be used concurrently by different groups of users, with different visions and capabilities.
- To be based on the client-server computation model.
- To support different types of multimedia information, eventually without any structure;

The main functionalities of the current version of Web-Q are:

- Loading and configuration of the QMS information system (QMS IS) “configuration” structure (this functionality enables the system to be easily adapted to different organizations). This means configuration of the following components:
  - menus, options and actions;
  - users and groups of users,
  - organization structure (different kinds of structures should be possible, such as tree or list).
- Management (insert, change and remove), retrieval and maintenance of the QMS IS kernel. Conceptually the QMS is based on the following basic elements: *documents*, *processes*, *activities* and *quality records*. These elements formally keep the procedures related to the set of processes that each organization executes in their business cycles. The system must provide support to:
  - create all the elements referred to above;
  - establish relationships between the different elements;
  - associate elements with the QMS IS;

- associate elements with the organization structure;
- navigate between all the interrelated elements;
- edit and print all the elements.

- Management, retrieval and maintenance of users, or group of users; and of the organization structure.
- Access, in a controlled way, by everybody of the organization, or even by someone outside the organization. This implies that the system must provide a distinct vision of the information as well as a distinct set of available operations, for each user, or group of users.

3 Data model

The kernel of the QMS IS consists basically in a structure set of QMS elements, which in general is hierarchical (e.g. chapter 1, section 1.1, section 1.2, ... chapter 2, ...). Each structural level (e.g. section 3) of the IS could contain several documents and several quality records. A document represents, basically in text media, descriptive information about one or more processes and it is associated with only one structural level. A quality record represents a template, or a generic form, used by the processes of the QMS (for instance, an invoice, a fax-form, or a receipt template could be a quality record) and can be found in more than one structural level. Processes and activities have graphic representation in the Web-Q system. A process is described by a set of activities (represented by circles) that interact with quality records (represented by squares) and has scheduled times and costs associated. The current version of the prototype developed doesn't support process and activity elements yet. Figure 2 depicts, using the OMT [8] notation, the object model of the elements involved in the QMS IS.

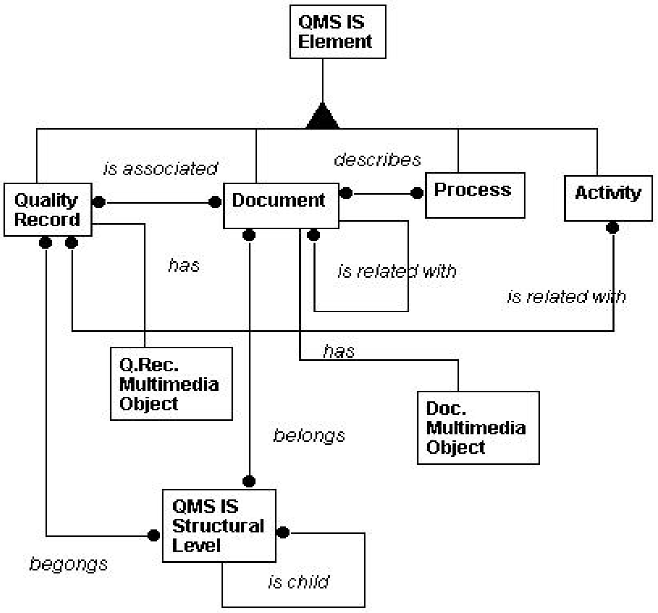


Figure 2: Elements Object diagram.

It is particularly important to note the relationships between the different elements and that both document and quality record have a 1-to-1 connection with an associated multimedia object (implemented as a HTML file). Another point to note is the association '*is child of*' of the QMS structural level with itself. It is this kind of association that enables the QMS to be structured hierarchically.

The Web-Q system supports three different types of entities (users) according to the organization point of view: external entity; employer entity; and structural entity (Figure 3).

Each external entity has a type (for instance: supplier, client, public institute, etc.). The organization must be structured in a set of levels (in general hierarchical levels) to which the employer and structural entities are bound. Structural entities have a 1-to-1 relation with the organization level, which means that structural entities could be created to correspond to a virtual entity associated with a group of entities (e.g. Direction, Operators, etc.). The employer entities are associated with real users, that must belong to any organization level.

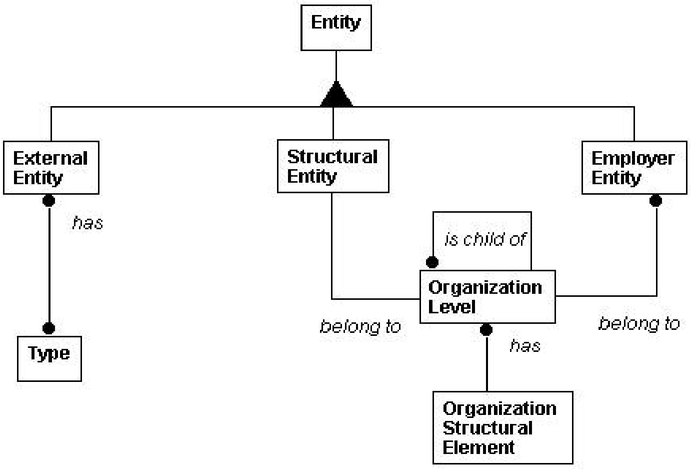


Figure 3: Entities Object diagram.

Figure 4 shows the associations between entities and QMS IS elements. The association "access" maintains, in terms of consult and edit (write) operations, the access that every entity makes to every element. The association "involve" keeps the set of capabilities/operations that every entity may execute over every element. Finally, the association "situation" keeps a historic record of every event/operation executed on every element, in such a way that we could determine in any time, for every element, what its current state is and what next events could occur (a simple control for workflow).

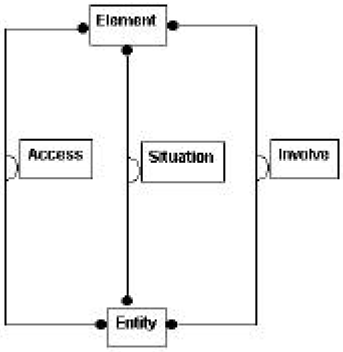


Figure 4: Elements and Entities Relationship Object diagram.

Other data visions of the system won't be presented here due to space limitations, particularly those concerning configuration meta-information, such as menu/options and respective associations with entities.

4 Multimedia issues

After the presentation of the general principles of the Web-Q system and its respective data model, some questions must be raised in order to take some implementation decisions. Basically, these questions could be summarised in the following points:

- How to represent the QMS IS elements that, as we have seen, aren't structured and could involve different data media?
- How to save and efficiently manage this information?
- How to integrate this information characteristics with the flexibility requirements demanded by the Web-Q system? That is, how to represent this information together with all the meta-information involved (entities, menus/options, organization levels, etc.)?

One possible representation of the QMS elements could be to use one format based mainly on text media, such as the simple ASCII format or the proprietary Microsoft document format DOC or the RTF (rich text format), or a more complex format based on the multimedia object description, such as Hytime [22], ODA [23], or MHEG [24]. Text based formats weren't adopted because of their lack of support for other media (mainly images and graphics) and their portabiblity deficiencies (except the ASCII format). Multimedia based formats weren't adopted due to their complexity and the fact that they lack edition tools. Our choice focused on the HTML language [12] that is a SGML [14] Document Type Definition. Our decision is based on the fact that HTML language can support simple multimedia object description in an extensible way [9], there are many viewers and even authoring-in-the-small tools, it can support hyperlinks between a generic set of documents, its documents are portable, and mainly it is becoming a "de facto" document standard.

The second question raised, could have the following two solutions: either maintain the multimedia objects (in our case, HTML pages together with images and eventually audio records and animations) directly in the file system of the server; or maintain these objects in either a relational, using BLOBs (binary large object) fields, or in an object-oriented database management system [20,21]. Due to simplicity and easy integration with the WWW model, the first solution was adopted. We think that this solution implies a better performance in comparison to the second one, due to the fact that an access to the file system is faster than the equivalent to the DBMS. (Nevertheless, this assumption must be conveniently checked to reach a final conclusion). However, the option adopted presents some drawbacks, such as a lower integration level with the rest of the information, poor and deficient protection and maintenance mechanisms.

The third question concerns the integration between the QMS multimedia objects and the rest of the information of the system. Figure 5 depicts its conceptual databases.

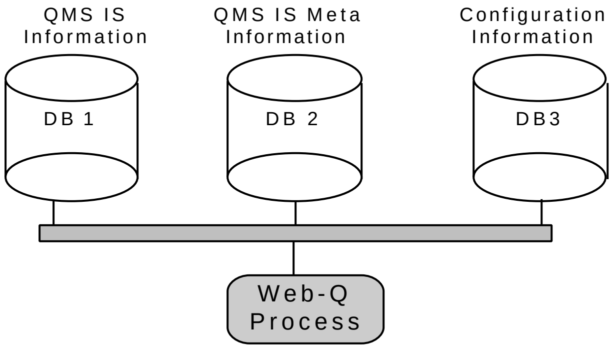


Figure 5: Conceptual databases organization.

There are conceptually three integrated databases. DB1 keeps all multimedia objects associated with QMS elements. DB2 maintains the associations between the different QMS elements and the QMS structure, and other related information. As figure 2 presents, each document or quality record maintains, in DB2, the meta-information as well as a direct link with a “real” multimedia object. DB3 maintains the rest of the information, such as: the entities (or users), its accesses and associations with QMS elements; the organization structure and its relationships with entities; the menus/options and related configuration information. According to the schema of figure 5, all the integration and cohesion of the information is the responsibility of the Web-Q process.

5 Implementation issues

The implementation of the system presented in the previous sections is based on the WWW paradigm. This section presents some interesting points involved with the development of the Web-Q system. In section 6, the major problems and limitations found using this technology will be focused.

5.1 Computational model

W3 follows the client-server model that uses a simple stateless protocol (HTTP) over the current TCP/IP stack. This protocol is very adequate for the retrieval of hypermedia information but presents some drawbacks when used to simulate interactive connections, or when the clients need to handle some event notification that occurs at the server side.

In general, the W3 clients (also called “browsers”) have GUI capabilities, are multi-platform (X-Windows, MS-Windows, Macintosh, etc.), are multi-protocol (HTTP, FTP, GOPHER, NEWS, file, etc.), and parse and conveniently show HTML information. On the other extreme, there are the W3 servers (also called "HTTPd" or "HTTP daemons"). HTTP servers are in general multi-threaded, have configuration, logging and protection/authorization files, and support the CGI (Common Gateway Interface) protocol [13]. Servers answer client requests by returning HTML pages retrieved from the file system or by forwarding the data received from the CGI.

The CGI provides a mechanism to extend the basic functionality of an HTTP server by connecting it to other specific processes, called "gateways", invoked at run time. It is important to note that those gateways are completely independent processes that are launched by the HTTP server process. The way the data is communicated between the server and the gateway is conditioned by the CGI specification.

Figure 6 illustrates the general computation model described above.

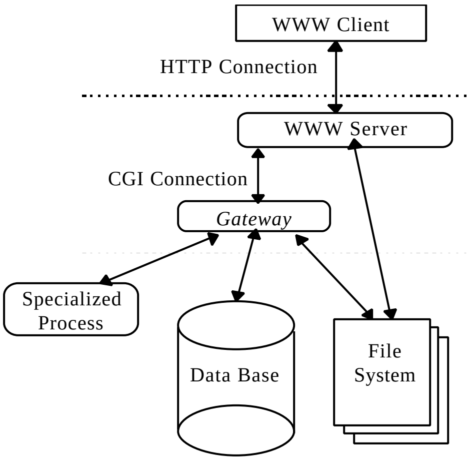


Figure 6: The WWW generic computational model.

Gateways are specific applications that can interact with several data repositories (e.g. relational databases systems, file systems, X.500 distributed directories [16], etc.) or even with other specialised processes. However, all the CGI gateways are "short life" processes; this means that they are not interactive processes.

The Web-Q system was developed in the MS-Windows (3.1, 3.11 and 95) environment. This means that the Web-Q system (the database and the CGI gateway itself) runs in this environment, while any client could be run in any platform as long as a generic WWW client will exist. From the server point of view it is just required an HTTP server supportting of CGI for Windows (Win-CGI). The generic computation of this kind of execution is depicted in figure 7.

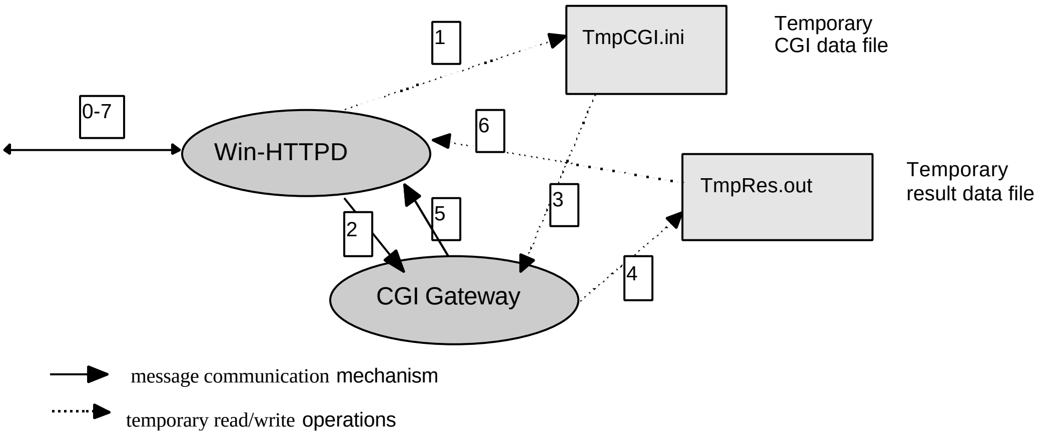


Figure 7: The CGI protocol adapted for MS-Windows environment.

In the response to each WWW client request (0), the server creates a temporary file "tmpcgi.ini" and eventually other additional temporaries files, which contain all the necessary information (1) involved. After that, the server starts the execution of the gateway (2) giving it the path of the file with the CGI information, and the path of the file where it will expect to receive the result information. The gateway reads and parses the content of the file "tmpcgi.ini" (3), executes a set of actions and writes its results in the file "tmpres.out" (4). Finally, the gateway notifies the server that it has finished(5). The server reads the content of the temporary file "tmpres.out" (6), and sends a HTTP response to the client (7).

The Web-Q process is a Win-CGI gateway developed in MS-VC++ [10]. We have used the GENESIS library (***Gene****ric Library for WWW* ***S****erv****i****ces Authoring*) [4,6] as the WWW support (basically dynamic generation of HTML information and CGI interface), and MFC (Microsoft Foundation Classes) to all other particularities of the system, mainly to use the ODBC (open database connectivity) interface.

5.2 Database implementation

The conceptual database model presented in section 3 is mapped in the implementation phase, in the schema depicted in figure 8.

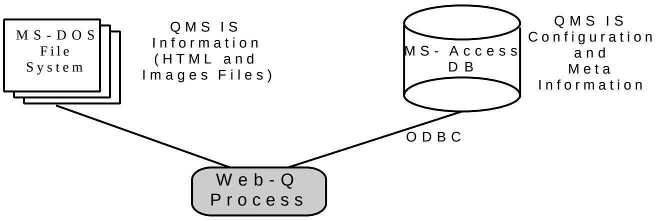

MS-DOS File System
QMS IS Information (HTML and Images Files)
MS-Access DB
QMS IS Configuration and Meta Information
ODBC
Web-Q Process

Figure 8: Web-Q database schema.

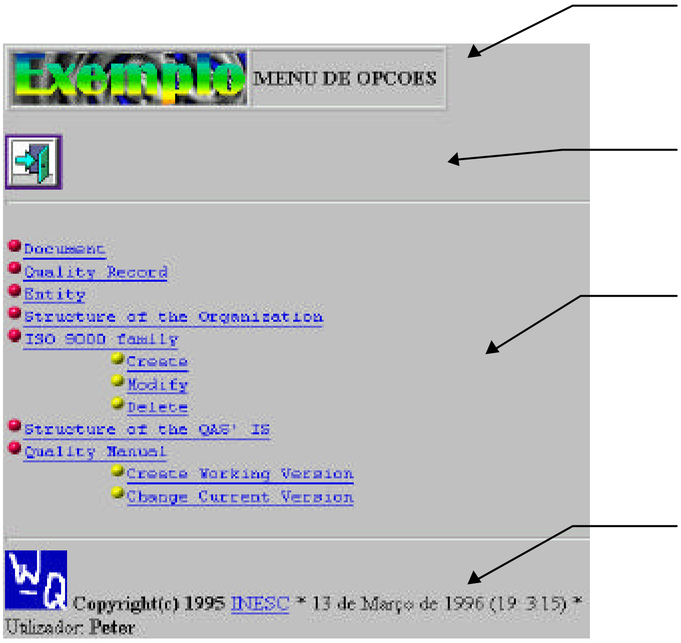
The Web-Q database consists in a hierarchy of directories, where the QMS elements (documents and quality records) are maintained as HTML and related image files (corresponding to the DB1 of figure 5); and the structured information is maintained in a MS-Access relational database (corresponding to the DB2 and DB3 of previous figure 5) and it is accessed from the CGI gateway from the ODBC (open database connectivity) interface [19]. Because the use of the "open" ODBC interface, the movement to a powerful DBMS should be possible and (apparently) transparent.

5.3 Some particularities concerning the user interface

5.3.1 Consistency in the generation of all user interfaces

All HTML pages generated "on fly" by the gateway correspond to the user interface of the system. In general, each page is structured sequentially by the following blocks (figure 9):

- A configurable **header**, that records the logotype of the organization and an information (a title) that identifies each page.
- A **set of icons** (images) that correspond, each one, to different actions. This set of icons depends on the capabilities that a given entity could have, and of course of the semantics of the page itself.
- The **body** of the page. This block corresponds to the semantic of the generated page and consequently is the most important block of each generated page.
- The **footer** of the page, which keeps the identification of the current entity, the date/time of the Web-Q system at the moment of generation, and other descriptive information.



The configurable header area

The set of icons/short-cut links area

The body area

(in this case, with a menu tree)

The configurable footer area

Figure 9: The menu page of the Web-Q system (text in Portuguese).

This structure keeps the cohesion and consistency of all generated pages of the system. These properties are important to avoid the problem "lost in the hyperspace" of hypermedia applications.

5.3.2 A flexible and generic navigational tree

The figure 9 presents the first page (after the authentication page) of the Web-Q system. One interesting component developed was a generic navigational tree. This component has been adopted in several situations around the system implementation such as: the menu tree, the organization structure tree or the QMS IS structure tree. A generic tree could have several branches and/or leaves. Each branch could be expanded or retracted, just by clicking over the corresponding link, so that new branches and/or leaves would be shown. Our component supports the opening of several branches concurrently. Figure 9 shows the tree of menus/options, where there are two branches expanded. When a leaf is selected by the user, a new context is generated. For instance, you could image the QMS IS structure tree with several hierarchical branches (corresponding to the hierarchical structure of the QMS IS), each one with potentially different types of leaves, for instance documents or quality records.

5.3.3 Controlled edition of QMS elements

In the actual version of the prototype, all the normal user interfaces consist of HTML page sequences, that are dynamically generated from all the information kept in the database and in the file system. Although there is information generated from the Web-Q gateway, there is also information that should be edited "by hand". This is the case of the edition of the QMS elements. The system enables the edition of QMS elements in a controlled way (only for those entities that have associated write access) but only in the local machine. Figure 10 depicts the mechanism involved.

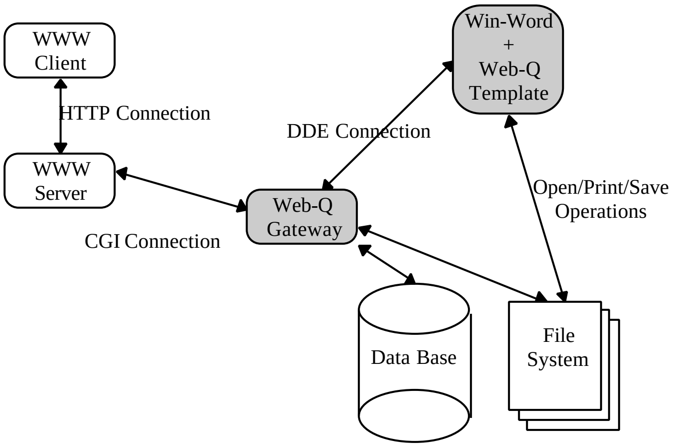
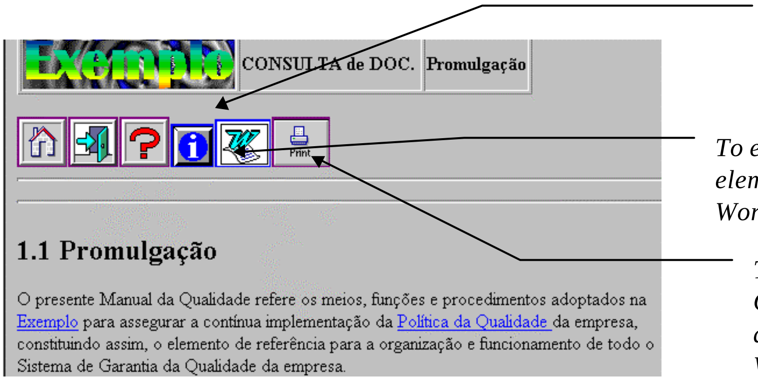


Figure 10: QMS element edition mechanism.

To edit a document (or a quality record) the user must access the respective HTML page wrapped in the Web-Q system. Figure 11 presents a snapshot of a generated HTML page for an entity that has the capability to edit a given document. In this case the user must select the icon "Word" in order to edit the corresponding HTML file. The client sends that command to the Web-Q system (via HTTP and CGI protocols), which in turn raises the execution of the Win-Word and gives by DDE the command "open file" with a given attach Win-Word template. This template is specific of the Web-Q system, however, it is based on the HTML-Assistant (which is an HTML template to Win-Word). To print the document according to the characteristics of this template, the user may select the "Print" icon. Figure 12 shows part of the meta-information involved with the document element. It is showing its current version; the place where the document should be maintained; an external code or identification; a list of the entities involved with it and its respective capabilities; a list of other situations/states that the document has gone through (the current state of the document is “validated on 12/12/1995 by the entity pmsff”); a list of the related ISO 9000 related codes; a list of other related documents; a list of related quality processes; and a list of the entities that may access the document. For these last four lists, it is possible to select a specific item and consult its detailed information, using the hypermedia capabilities inherited from WWW technology. Basically, each of there lists consists of a FORM element with two sub-elements: a list-box with single selection (SELECT tag) and a consult button (INPUT tag with attribute TYPE = Submit).



1.1 Promulgação

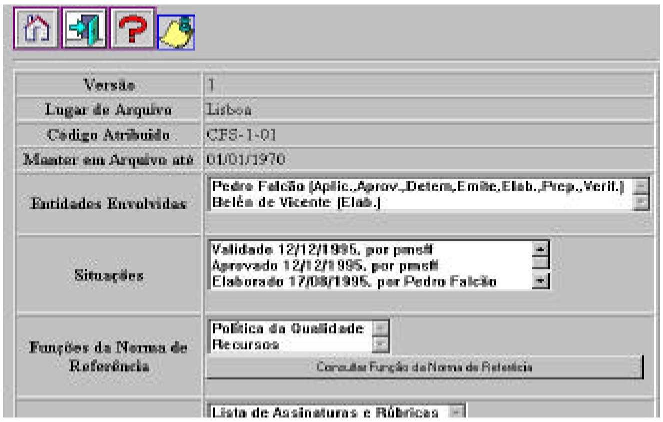
O presente Manual da Qualidade refere os meios, funções e procedimentos adoptados na Exemplo para assegurar a contínua implementação da Política da Qualidade da empresa, constituindo assim, o elemento de referência para a organização e funcionamento de todo o Sistema de Garantia da Qualidade da empresa.

To access to the current QMS meta information

To edit the current QMS element using the Win-Word as a HTML editor

To print the current QMS element using a customized Win-Word template.

Figure 11: Snapshot of the HTML page associated with the QMS document.



Versão	1
Lugar de Arquivo	Lisboa
Código Atribuido	CFS-1-01
Manter em Arquivo até	01/01/1970
Entidades Envolvidas	Pedro Falcão (Aplic.,Aprov.,Detem,Emite,Elab.,Prep.,Verif.) Belén de Vicente (Elab.)
Situações	Validado 12/12/1995, por pmsff Aprovado 12/12/1995, por pmsff Elaborado 17/08/1995, por Pedro Falcão
Funções da Norma de Referência	Política da Qualidade Recursos Consultar Função da Norma de Referência
	Lista de Assinaturas e Rúbricas

Figure 12: Snapshot of the HTML page associated with the QMS document meta-information.

6 Conclusions and future work

This paper presented the work done in the context of a very specific area of modern organizations, which are the quality assurance systems, where the information involved could use different media, and isn’t structured.

Our prototype was developed in the Windows environment, using a relatively large number of tools and technologies in a novel way. We proved that the HTML language can be used, with advantage, as a simple and flexible format to represent and define multimedia objects [9], and that it is possible to integrate HTML browsers, with HTTP servers, specialized servers (Web-Q), and others tools (such as Win-Word with a specific Web-Q template), in a relatively transparent way.

However, several problems and limitations appeared with our prototype, mainly due to the lower level of interactivity with the user and a lower performance level inherent from the CGI stateless connection. Another capability not yet implemented, was the specific treatment of the processes and activities. These elements must basically have a graphic representation and consequently, some customized graphic editor should be provided in order to enable the users to specify (to design) them.

The HTML 2.0 was adopted in our prototype version, which defined only a primitive mechanism for inserting media into HTML documents - the IMG tag - that was worthwhile to the Web-Q goals. However, major developers involved have been presenting new ideas to handle the new media: Sun's APP and APPLET tags [15] for executable code; Netscape's EMBED tag [17] for compound document embedding, and Microsoft's DYNSRC attribute [16] for video and audio. To unify these different perspectives, a W3C Working Draft specification [9] was written to extend HTML in order to support the insertion of multimedia objects such as Java applets, Microsoft Component Object Model (such as OLE components), and a wide range of different media plug-ins. (The new INSERT tag is being proposed).

We think that more research and work must be done to improve the generic functionality of the Web-Q. Namely, we identify the following points:

- Use the Netscape's FRAME tag to give a better level of interactivity and user interface cohesion and integration. For instance, we should use a fixed frame for the heading and footer, and variable frames for the list of icons/actions, and for the body part (see section 5.3.1).

- Develop some specific Java applets to give users a better level of interactivity and to improve the general performance of the system. For instance, an applet associated with the tree mechanism, with the expansion and retraction capabilities would be desired. Another important applet should be a customized graphic editor, to enable the remote editing of graphic (processes and activities) Web-Q elements.

- Develop a mechanism to "keep-alive" the CGI gateways in order to significantly improve the performance of the system. This technique could be be based on a third process, called *dispatcher*, that would be a generic CGI gateway that could know how to communicate and cooperate between the other processes (the previous CGI gateways). This kind of solution is being used by some developer providers and researchers in different ways [33].

- Integrate Web-Q with other workflow and document management applications. This should be another interesting worthwhile task in order to prove the well adopted characteristics of WWW as an integrator of technologies and applications.

After the design and development of the project presented, some questions and related multimedia aspects were kept open and should be discussed further:

- Is the HTML language, with its new INSERT tag [9], sufficient enough as a generic multimedia representation format? What are its major advantages and drawbacks relatively to languages such as HyTime or MHEG? Would HTML become an "umbrella" to cover several different formats?

- If defining an HTML file, with its related media files, as a multimedia object, what will be the best way to save and to handle these objects? At the operating system (i.e. file system) level (as it is currently done and as adopted by the Web-Q system)? Or at an object-oriented DBMS? Are there any significant differences between the related capabilities found in either operating systems or database systems? Or, should we just talk about persistent objects?

Acknowledgments

The basic ideas involved the generic definition of a computer supported QMS was due to Pedro Falcão and Vicente de Belén. The preliminary discussions involving the implementation phase was done with José Borbinha. A significant part of the implementation work was done by Pedro Ribeiro.

References

[1] James Vincent, Albert Waters, and John Sinclair. *Quality Assurance* Vol. 1 - Practice and Implementation. Vol. 2 - A programming Guide. Prentice Hall, 1988.

[2] A. Ramos Pires. *Qualidade*. Colecção Sílabo Gestão. Edt. Sílabo, 1993.

[3] T.Berners-Lee, R. Cailliau, A. Luotonen, H.F. Nielsen, A. Secret. The World-Wide Web. *Communications of the ACM*., pp. 76-82, August 1994.

[4] Alberto Silva, José Borbinha, José Delgado. Organizational management system in an heterogeneous environment - A WWW case study. In *Proceedings of the IFIP working conference on information systems development for decentralized organizations*, pages 84-99, Trondheim, Norway, August 1995. http://bruxelas.inesc.pt/~alb/papers/ofw3.ps

[5] José Borbinha, Alberto Silva, Pedro Ribeiro, José Delgado, Paulo Caetano. Desenvolvimento de Sistemas e Serviços para a Internet com interface WWW. *2º Encontro Nacional do Colégio de Engenharia Electrotécnica da Ordem dos Engenheiros.* Dezembro 1995.

[6] José Borbinha, Alberto Silva, Luis Lageira, Pedro Ribeiro, José Delgado. GENESIS - Biblioteca Genérica para o Desenvolvimento de Sistemas de Informação com Interface para WWW. *Comunicações da 1ª Conferência Nacional WWW*, Universidade do Minho, Julho,1995. http://bruxelas.inesc.pt/~alb/papers/genesis.ps

[7] K.L. Kraemer, J.L. Kling. Computer Based Systems for Cooperative Work and Group Decision Making. *ACM Computing Surveys*, vol.20 nº 2, June 1988.

[8] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, and William Lorensen. *Object-Oriented Modelling and Design*. Prentice-Hall, Englewood Cliffs, NJ. 1991.

[9] Dave Raggett, Charlie Kindel, Lou Montulli, Eric Sink, Wayne Gramlich, Jonathan Hirschman, Tim Berners-Lee, Dan Connolly. *Inserting multimedia objects into HTML3. W3C Working Draft*. 21-Dec-95. http://www.w3.org/pub/WWW/TR/WD-insert.html

[10] Bjarne Stroustrup. *The C++ Programming Language*. Second Edition. Addison-Wesley, Reading, MA.1991.

[11] http://www.w3.org/pub/WWW/Protocols/

[12] http://www.w3.org/pub/WWW/MarkUp/

[13] http://www.w3.org/pub/WWW/CGI/

[14] http://www.w3.org/pub/WWW/MarkUp/SGML/

[15] http://www.sun.com/

[16] http://www.microsoft.com/

[17] http://www.netscape.com/

[18] C. Weiser, J. Reynolds, S. Heker. *RFC 1309 - Technical Overview of X.500*.

[19] Mike Satterfield. Introduction to ODBC Driver Development. *Windows/DOS Developers's Journal*. December 1993.

[20] John Joseph, Satish Thatte, Craig Thompson, David Wells. Object-Oriented Databases: Design and Implementation. *Proceedings of the IEEE, vol. 79, nº 1*. January 1991.

[21] Malcolm Atkinson, Fançois Bancilhon, David DeWitt, Klaus Dittrich, David Maier, Stanley Zdonik. The Object-Oriented Database System Manifesto. *Proceedings of Distributed Object Oriented Databases*, December 1989.

[22] S. Newcomb, N. Kipp, and V. Newcomb. The Hytime: Hypermedia/Time-based Document Structuring Language. *Communications of the ACM*., vol. 34, nº 11, pp. 67-83, November 1991.

[23] H. Brown. Introduction to th e Office Document Architecture. In J. Rosenberg, M. Sherman, A. Marks, and J. Akkerhuis, editors, *Multi-media Document Translation: ODA and the EXPRESS Project*, chapter 3, pages 33-76. Springer-Verlag. 1991.

[24] T. Meyer-Boudnik and W. Effelsberg. MHEG Explained. *IEEE Multimedia*, Spring 1995.

[25] C. A. Ellis, S. J. Gibbs, and G. L. Rein. Groupware, some Issues and Experiences. *Communications of the ACM*., vol. 34, nº 1, pp. 38-57, January 1991.

[26] E. Yourdon, and L. Constantine. *Structured Design*. Prentice Hall, Englewood Cliffs, NJ. 1979.

[27] G. Booch. *Object-Oriented Analysis and Design with Applications*. Second Edition. Benjamin Cummings, Redwood City, CA. 1994.

[28] D. Coleman, P. Arnold, S. Bodoff, C. Dollin, H. Gilchrist, F. Hayes, and P. Jeremes. *Object-Oriented Development: The Fusion Method*. Prentice-Hall, Englewood Cliffs, NJ. 1994.

[29] D. Coleman, P. Arnold, S. Bodoff, C. Dollin, H. Gilchrist, F. Hayes, and P. Jeremes. *Object-Oriented Development: The Fusion Method*. Prentice-Hall, Englewood Cliffs, NJ. 1994.

[30] T. DeMarco. *Structured Analysis and System Specification*. Yourdon Inc., New Yourk, 1978.

[31] Franca Garzotto, Paolo Paolini, D. Schwabe. HDM - A Model Based Approach to Hypermedia Application Design. *ACM Transactions Information Systems*, vol.11 nº 1, pages 1-26, January 1993.

[32] Tomás Isakiwitz, Edward Stohr and P. Baiasubramanian. RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM*, vol.38 nº 8, pages 24-34, August 1995.

[33] Ann Lynnworth. *WebHub Developer's Manual*. HREF Tools Corp. January 1996. http://www.href.com

[34] Alexander Clausnitzer, Pavel Vogel, and Stephan Wiesener. A WWW interface to the OMNIS/Myriad literature retrieval engine. *Computer Network and ISDN Systems*, vol.27 nº6, pages 1017-26, April 1995.

[35] Leon Shklar, Kshitij Shah, and Chumki Basu. Putting Legacy data on the Web: a repositary definition language. *Computer Network and ISDN Systems*, vol.27 nº6, pages 939-951, April 1995.

[36] Matthijs van Doorn, and Anton Eliens. Integrating applications and the World-Wide Web. *Computer Network and ISDN Systems*, vol.27 nº6, pages 1105-1110, April 1995.