

# The WebComfort Project

João Leonardo Carmo, Alberto Rodrigues da Silva

SIQuant – Engenharia do Território e Sistemas de Informação  
Av. Casal Ribeiro, nº18 – 9º Dto, 1000-098 Lisboa, Portugal  
INESC-ID, Instituto Superior Técnico,  
Rua Alves Redol, nº 9 – 1000-029 Lisboa, Portugal  
{joao.carmo, alberto.silva}@siquant.pt

**Abstract.** A Content Management System (CMS) is a critical element for the success of organizational web sites and intranets. This paper focuses mainly on web-oriented CMSs, and demonstrates the effort spent with the objective to produce a web-oriented CMS framework, fully developed with Microsoft technologies, namely ASP.NET 2.0 (C#) and SQL Server 2005. The major technical details of this project are described in this paper, as well as the related and future work.

**Keywords:** WebComfort, Content Management Systems, ASP.NET, Framework, Dynamic Web Pages, Dynamic Content

## 1 Introduction

Nowadays almost every organization has a web site, for the Internet or just in an Intranet context. These systems will inevitably evolve and expand in a considerable way aligned with their respective organization's lifecycle. Consequently, their involved content should also evolve over the years, but it will most certainly be outdated or incorrect, with no navigation or search capabilities. The maintenance of a web site, once a very simple activity, can easily become a nightmare. Recent web sites tend to scale from tens to thousands of pages. Still, aesthetics and visual attractiveness are critical factors for the web site's success, having to be modified and renewed as many times as necessary.

The worst part of this approach is the effort manage and administrate these web sites, normally carried out by just one person or by a limited group of people, commonly named "webmasters". The consequence of this fact is a simple one: it is easy to lose control of the web site's pages. What about its organizational and logical structure? Well, someone has the answer: the webmaster. It is clear that this web site maintenance workflow is funneling to a single break point in the near future, the webmaster, which can eventually compromise the entire web site and collapse it completely, in a very pessimistic, however, possible scenario.

Given this context, the need to structure and manage these informational units is of the utmost importance. An "informational unit" can be looked upon as every entity that is responsible for the production, structuring, storage, search, edition and/or

content use. These tasks, among others, are usually found along with the ones delegated to a CMS (Content Management System) in the heart of an organizational environment.

Generically, a CMS (Content Management System) is a software platform that supports the creation, management, distribution, publication and search of organizational contents (e.g., web pages, images, sounds, other types of documents) [4, 5, 6]. The web version of a CMS will only handle contents targeted to be visualized or used mostly in web sites.

A CMS is responsible for the web site's contents life cycle management, providing tools for new content creation and edition, oriented by a content creation/edition workflow. Further on, a CMS also enables complete site structure and visual appearance management, while delivering automatically generated and completely integrated web site navigation [4, 5, 6].

Although the designation of "CMS" does comprehend more than its "web side" as stated earlier, it is also true that the majority of its current use is precisely for web site management. In some circles of the CMS community, these systems are also known as Web Management Systems, or WMS. From this point forward, "CMS" will designate a web-oriented CMS.

One important point to notice is that a CMS doesn't solve the organization's informational chaos problem by itself. Its correct implementation and operation, by means of adequate administration policies, however, can lead to a much expected solution for this problem, or at least help to minimize it.

This paper is structured in four main sections. Section 1 introduces the context of CMS systems, and presents the structure of the paper. Section 2 overviews the main concepts, features and benefits around the CMS area. Section 3 is dedicated to the WebComfort project [1], providing an extensive insight into the project's foundation and current development. Section 4 presents the conclusions for the project so far and refers some future work to be developed.

## **2 Content Management Systems Overview**

In order to clarify what CMSs are, it is important to analyze the typical CMS's content-based workflow, which can be divided in three main phases, such as suggested in Figure 1: (1) Content Creation/Edition; (2) Content Management; and (3) Content Publication and Presentation.

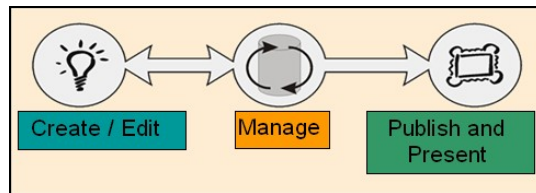


Figure 1. Generic CMS's Content-based Workflow

To discuss these phases it is essential to identify specific roles in the CMS-supported organizational context. Although diverse roles can exist, the most important ones are: (1) Content Authors/Editors; (2) Content Managers; (3) Publication and Presentation Managers; and (4) Administrators. These roles can later subdivide themselves hierarchically into more specific ones, such as text author, designer or layout publisher. There is also another role, commonly known as "anonymous". It's through this role that public access to the system (by external entities, i.e., that do not possess an identity) is achieved.

CMS technology gathers a set of concepts and terminologies, as suggested in Figure 2, which is important to explain, to assure a better comprehension of this paper, namely:

- **CMS (Content Management System)** – a web portal manager computer system focused on the content, providing an abstraction layer of the technological details for the end user, allowing him to focus on the most important web portal asset: it's content.
- **Web Site** – a logical and contextualized set of web pages managed by a CMS. Typically the pages will include a navigation banner and an associated visual style to all of them.
- **Dynamic Web Page** – a logical set of components (designated by Web Components, description follows) in the context of a Web Site. Typically the components will also include an associated visual style. (e.g., page, section, tab, container, placeholder).
- **Web Component** – an independent component with associated business logic (e.g., module, webpart, portlet). It is responsible for storing and presenting content information with a determined visual style, and performs actions on the content (e.g., edit, remove, search).
- **Visual Style** – a set of layout settings (i.e., content spatial disposition information) and of visual themes, i.e., content appearance information (colors, fonts, sizes, etc.).
- **Page Workflow** – logical sequence of publishing steps associated to a Web Page.
- **Component Workflow** – logical sequence of publishing steps associated to a Web Component.

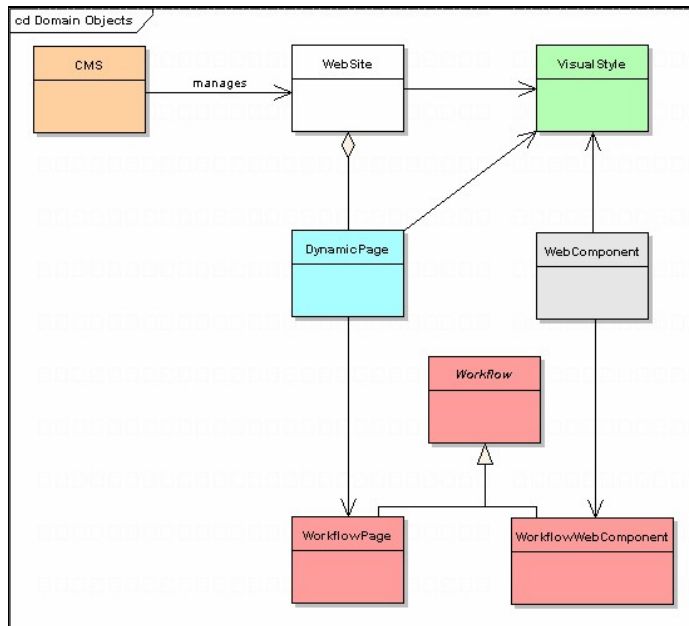


Figure 2. A generic domain model for CMS's concepts

### 3 The WebComfort Project

#### 3.1 Overview

WebComfort is a Web Portal and Content Management System Framework promoted by SIQuant [3]. WebComfort is fully developed and supported by Microsoft technology, including ASP.NET 2.0 (C#) and SQL Server 2005. It enables web portal management and operation in an integrated manner, supporting the required portal and content management tools and mechanisms through simple web browser access, without the need to use an additional standalone application.

As suggested in Figure 3, a WebComfort portal consists in a variable number of tabs (or dynamic pages) based on ASP.NET masterpage technology, hierarchically organized, in which each tab structures different contents through different predefined module types, based on ASP.NET user controls. In addition, module positioning is supported by a flexible structure of dynamic containers to satisfy specific web site design and presentation needs.

WebComfort claims the separation between the content and its presentation on the web site. Content is presented through information modules, whose presentation can

be configured without changing the data model and the underlining contents. In particular, regarding content presentation, a layout management mechanism is provided, at portal, tab and module levels.

WebComfort's authorization and security policy is defined according to a flexible role-based mechanism. It is possible to create and manage numerous roles according to different functional and business requirements. There are two main different types of access, tab-wise and module-wise: viewing access and management access.

The "framework" designation comes from WebComfort's easiness of extension, by allowing adding new module types to manage and display existent information, or even new types of information, supporting the development of new module logic and design through a well-defined module API.

Current available modules support typical web site functionalities, such as *Announcements, Events, Contacts, Links, Image, Documents, HTML Document, Rich HTML Editor, XML Document, Discussion, Forum, Chat, Navigation Menu* and *Navigation Tree*. Additionally, specific modules for electronic commerce, portal usage statistics, project management and GIS (Geographic Information Systems) are also available.

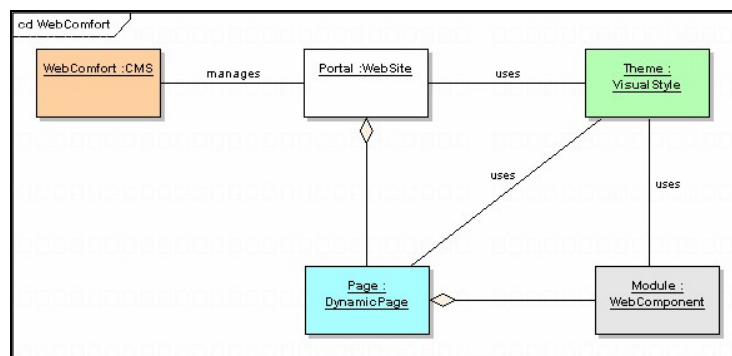


Figure 3. WebComfort's basic elements

### 3.2 WebComfort Portal Structure

Each WebComfort portal is composed by a series of pages, called tabs, which can be accessed through a portal navigation banner.

In its standard form, each tab is composed of three vertical containers. However, this structure can be customized to different configurations. These containers hold the page's modules: the elements responsible for managing and displaying content within a tab (e.g., links, text, images, HTML code).

Each module has a title, a content editing area and a possible help page. The module's content editing area is only revealed to users with editing permissions to the referred module.



Figure 4. A WebComfort portal example, using a visual style with two vertical containers

### 3.3 Users, Roles and Tab Management Delegation

Each user as an identity within the portal, represented by a user account identified by the typical username and password pair. Other than his personal identity, a user needs permissions to gain access to private tabs, or to be able to perform management tasks within the portal.

In WebComfort permissions are not granted directly to users. Instead, permissions are granted to roles. This way, it is possible to associate a user with one or more roles, granting them the respective permissions. Permissions can be issued on modules or tabs (see Figure 5).

There are two types of roles: system roles and local roles. A system role has a scope that spans through the entire portal's tabs, whereas a local role's scope is confined to a specific tab. Module management can be delegated to a local role, but tab management is only available through a system role.

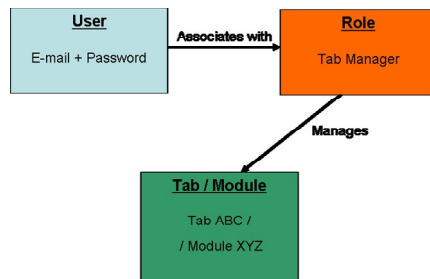


Figure 5. The relationships between users, roles and tabs / modules

WebComfort provides a flexible and easy role management feature called role hierarchy, which allows the definition of role hierarchies within the portal.

To better explain, suppose that there are two roles in the portal with associated permissions, Father and Mother. Using WebComfort's role hierarchy feature, it is possible to create a third role, Son, which inherits the permissions of both Father and Mother roles. This way, any user that becomes associated with the Son role will have the same permissions that a user associated to both Father and Mother roles. As a final note on role hierarchy, it just needs to be said that a local role can be inherited by a system role, but the opposite is not allowed, for obvious security reasons, as a local role could "escape" its tab-contained scope.

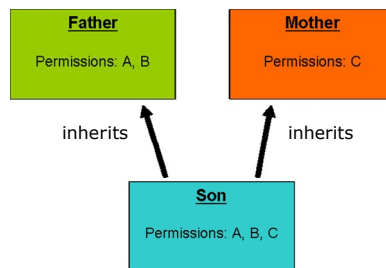


Figure 6. WebComfort's Role Hierarchy

Another useful and critical WebComfort feature is management delegation. Using management delegation, the web portal administrator can delegate a system role the task of managing an entire tab, without giving it the rest of the administration capabilities, like user registration or tab creation. This is called a Responsible Role. When empowered by a Responsible Role, the user can perform almost every action the portal administrator can, with the exceptions of not being able to remove the tab from the portal and not being able to define a new Responsible Role to it.

WebComfort modules are organized by module categories, and these categories have instantiation permissions. A Responsible Role can only instantiate modules in the tab that belong to authorized instantiation module categories.

With WebComfort's management delegation feature, it is possible to free the portal administrator from all tabs' management details, allowing for more flexible, easy and distributed site management. The administrator will still have full control over the entire portal. It is also important to refer that only a system role can become a Responsible Role, and that only a portal administrator can empower a system role with Responsible Role capabilities.

### 3.4 WebComfort Content Operation Features

WebComfort supports several important and useful content operation features, which are explained in this section.

#### 3.4.1 Content Referencing

Content referencing between modules allows that a module's content to be referenced by another module, acting as a direct pointer to the content of the targeted module.

This way, every time the content of the referenced module is updated, these changes will automatically be reflected to all the modules referencing it.

Content referencing promotes content reuse and helps minimize duplicate information within the portal. When a module is displaying referenced content, it loses the ability to edit its content, since the original content belongs to the module being referenced.

Content referencing doesn't delete the original module contents. When the reference is removed, the original module content is reinstated.

One important aspect about WebComfort's content referencing is the fact that it is only allowed between two "compatible" modules (i.e., two instances of the same module type). This way, WebComfort prevents a Tree module to reference a Text one, for instance.

Content referencing is available to all WebComfort modules natively, even for upcoming modules. The programmer doesn't have to interact with WebComfort's API in order to enable content referencing.

#### **3.4.2 Content Copy**

Content copy between modules is very similar to content referencing. The main difference lies in the fact that in content copy, the source module content is copied to the destination module, creating an independent instance of the same content. Then it can be altered without worrying about changing the original content. Once again, this operation can only be applied between two compatible modules.

In order to enable a module to support content copy between its instances, a public method named "moduleCopy" with two integer arguments representing the source module and the destination module must be developed by the programmer. This provides the appropriate content copy programmatic support to interact with WebComfort.

#### **3.4.3 Content Import and Export**

Content import and export is a very useful feature that allows a module's content to be exported or imported via a WebComfort XML Content File.

In order to enable a module to support content export and import features, two public methods named "moduleExport" and "moduleImport" with two arguments each (source module and file name) must be developed by the programmer. This provides the appropriate content export and import programmatic support to interact with WebComfort.

A WebComfort XML Content File has a very simple structure. It consists on a "module" tag that provides the support for the name and the title of the module, and on several "element" tags, that basically represent the module's elements via a tuple <element name, element text>.

It is also possible to export and import a whole tab's contents to and from a single WebComfort XML Content File. WebComfort basically makes recursive calls to the tab module's export and import features to accomplish this.

#### **3.4.4 WebComfort Static Content Language Engine (SCLE)**

WebComfort supports static content multi-language through its Static Content Language Engine or SCLE. This way it is possible to incorporate several languages



into the module's static content (i.e., labels, buttons) without having to develop the same module in another language. The words and phrases from the different languages are store in XML files called WebComfort Language Packs.

When a new module is developed, the programmer can now provide it with multi-language support by creating a new WebComfort Language Pack for it. To activate the multi-language feature, the programmer just has to make a call to the WebComfort API to activate the SCLE for his module. The module will automatically be rendered in sync with the actual portal language.

### 3.5 WebComfort Development Team: Work in Progress

WebComfort is presently in its second version. Version 2.5 is currently under strong development (release candidate: July 2006). WebComfort v2.5 new features list includes:

- **Online Visual Style editor.** Presently, visual styles are supported in WebComfort, but the development of new visual styles is achieved manually. With the new online Visual Style editor it will be possible to create and edit visual styles easily and without the hassle of manual coding. The editor also promotes a much more error-free visual style development environment.
- **WebParts technology integration.** ASP.NET 2.0 WebParts will boost the actual WebComfort module semantics, providing them with inter-module communication and drag-and-drop functionalities within the tab.
- **Multi-Language Dynamic Content.** WebComfort will add support for multi-language dynamic content, allowing a module's dynamic content language to match the portal language, just like static content already does. This way it will be possible to write a piece of text in several languages, and the correct version will be shown according to the portal's language definition.
- **Integrated Content Search Engine.** A new integrated content search engine will allow fast searches through the entire web portal and through any type of content.

## 5 Conclusions and Future Work

Nowadays there are many CMS systems with respective flavours, varying from proprietary to open-source offers, supported by Java, ASP.NET, PHP and other technologies [9].

CMSs tend to be modular, extensible and versatile systems. Most of the corporate web sites are (or will be, in a near future) supported by this kind of software frameworks, with the benefits and features discussed broadly in this paper.

WebComfort is a CMS developed preliminary at INESC-ID's Information Systems Group (GSI), mostly for the sake of its academic and research interests. Some of its features, presented in this paper, were designed and developed in that context. Due to the recognition of WebComfort's stability and versatility, it became a commercial product promoted by SIQuant.

The current state of WebComfort provides the following features: (1) content and presentation separation; (2) extensibility through modules developed as required; (3)

authorization and security policies based on a flexible role-based approach; (4) content operation features such as (i) content referencing; (ii) content copy; (iii) XML content import and export; (4) online visual style editor; (5) webparts technology integration; (6) multi-language support; and (7) content search support.

With all these features involved, WebComfort supports very well the design, development and operation of web sites. However, we still think there are some areas to improve, in particular at the following levels: (1) the definition and development of specific toolkits (e.g., for documental and organizational management); (2) content definition and management workflows; (3) research new approaches to produce web sites and respective modules yet in a faster and productive manner.

Regarding this last topic, in the near future WebComfort will be used in a much wider sense to support a new form of CMS deployment technology. Following an MDD approach (Model Driven Development) [7], a new Web Portal Modelling Language (WPML) will be defined. The WPML will allow the complete specification of a CMS system, from visual style definition and elements composition down to component workflow specification. The WPML will be an extension to UML XIS 2 profile [8]. The resulting web portal models will then be instantiated automatically, deploying a fully functional and configured CMS system (WebComfort, in this case).

## References

- [1] WebComfort (<http://berlin.inesc-id.pt/alb/WebCoMFORT@78@-1.aspx>),
- [2] J. L. Carmo, L. Pereira, "WebCoMFORT – Web Content Management Framework", Final Course Report, 2003, Instituto Superior Técnico
- [3] SIQuant (<http://www.siquant.pt>),
- [4] Guruge, A., Corporate Portals Empowered with XML and Web Services, Digital Press, 2003,
- [5] Sum, P. et. al., Content Management Systems, Glasshaus, 2003,
- [6] Robertson, J, "So, what is a CMS?", KM Column, 2003 ([http://www.steptwo.com.au/papers/kmc\\_what/index.html](http://www.steptwo.com.au/papers/kmc_what/index.html)),
- [7] Model Driven Architecture (<http://www.omg.org/mda/>).
- [8] A. R. Silva, "The XIS Approach and Principles", Proceedings of the 29th EUROMICRO Conference, Turquia, Antalya, Setembro de 2003, IEEE Computer Society (<http://berlin.inesc-id.pt/alb/static/papers/2003/EuroMicro2004-Alberto-Silva.pdf> )
- [9] CMS Watch Top 13 Enterprise Portals (<http://www.cmswatch.com/Portal/Report/Vendors/>)