

## **Introduction**

---

This case study highlights the experiences and lessons learned during a focused website content management system (CMS)<sup>1</sup> research and implementation project. This project was initiated in November 2003 when the Embassy of Finland agreed to fund Ungana-Afrika in their efforts to explore new information and communication technology (ICT) tools within the African development community.

Ungana-Afrika is a South Africa based non-profit organisation that provides technology support, and helps others initiate technology support programs, within the development community of Southern Africa. In partnership with the Finnish Embassy, Ungana-Afrika's goal for this project was to build its own internal understanding and knowledge of CMS solutions, and in turn, help bring the benefits of this technology to the other non-profit organisations and groups that it serves.

Though Ungana-Afrika's long term objective for this CMS research is very broad (and will hopefully prove relevant to a wide variety of development contexts in Africa), this Case study focuses primarily on Ungana-Afrika's own experience with one open source CMS called Midgard. In early 2004, Midgard was piloted internally by Ungana-Afrika in a hands on effort to duplicate the process that any organisation must go through when evaluating a CMS.

## **Project Environment**

---

Though Ungana-Afrika's mission is technology focused, the non-profit structures and environment in which it operates is common to many small non-profit organisations. This section outlines this structure, and the needs Ungana-Afrika has for a web presence.

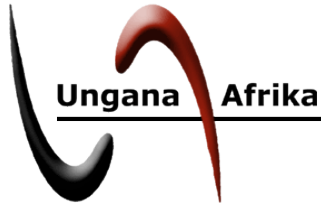
### **Physical Characteristics of The Organisation.**

Ungana-Afrika is a small non-profit organisation registered in South Africa and located in Pretoria. The main office supports 5 full-time employees, and the occasional volunteer. All employees have access to a computer, and most have their own laptop. The organisation has a LAN, linked to a 24-hour ADSL connection, and a Linux machine that provides server functionality only within the local network. As the ADSL connection uses dynamic IP addressing, and is also rather unreliable, it is not possible to operate a public server from the office. As a result, Ungana-Afrika uses an external non-profit hosting service (Kabissa) to manage/host website and e-mail services for the organisation.

All employees are proficient in basic computer operations, and a couple have more advanced knowledge including basic networking, programming and simple Linux administration. No staff members have formal experience with web programming or web administration, but one has self-taught knowledge of common web publishing tools (Dreamweaver, etc.) and simple HTML. This is the same employee who handles in-house infrastructure setup and troubleshooting.

---

1 In a TechSoup.org article by Rob Prodeaux, and CMS is defined as "a set of processes, applications, and databases that help an organization create, store, coordinate, and publish information in a useful format, in a timely fashion, and with a consistent method."



## **Website Content Management Solutions**

### **Case Study Within the Non-Profit Environment**

#### **The Importance of a Web Presence for Ungana-Afrika**

Ungana-Afrika depends on its website to communicate with a large audience both locally and globally. This audience includes potential clients, potential donors, partner groups, and general interested parties. For this audience, Ungana-Afrika wants to clearly communicate its mission and values, along with its service structure, and specific examples of experience/impact. In addition, in an attempt to keep everyone up to date on the most recent activities, general news and shared public resources are provided.

Ungana-Afrika hopes that its website can spark the interest of new clients and supporters while providing simple contact details for these groups to follow up. In addition, the website must serve as a common reference place where Ungana-Afrika employees can link contacts to specific information without re-typing the text every time.

To meet these needs, the Ungana-Afrika website must serve as a very glorified "brochure" of sorts, with some additional features such as news and resource links. However, Ungana-Afrika would like to tap into more interactive website features in the future. This could include providing support services for clients via a web menu/helpdesk, and interactive intranet services for remote employees., etc.

#### **Current Website Solution, and Expectations for a CMS Implementation**

Ungana-Afrika uses a static website structure, meaning that all web pages must be edited individually either by modifying the code directly, or using some kind of "what you see is what you get" (WYSIWYG) editor. One employee has been designated as web master, and he uses Dreamweaver to build and maintain a working copy of the website locally. Then, after any modifications, he uses an file transfer protocol (FTP) program to upload the changes to the public hosting server at Kabissa where these pages are available to the entire Internet.

Dreamweaver offers some level of "content management" as it utilizes templating features to simplify site-wide modifications, and has several file management tools that ensure integrity of the site's structure, links, etc. Despite this, in order to use Dreamweaver (or other similar programs) the web developer must also control the site's underlying "backend" structure, and have some general web programming skills. Though Ungana-Afrika does have an internal employee who can handle this, it means that all web publishing and control falls in the hands of one person. Thus, website updates happen infrequently, or not at all (if this person is not available). As a result, Ungana-Afrika wants to tap into the work flow options of a website CMS which would allow all employees (with or without web skills) to contribute to site updates directly.

In addition, as mentioned above, Ungana-Afrika wants to tap into more advanced and interactive website features that are above and beyond the scope of a simple "brochure" site structure. As a result, Ungana-Afrika was hoping that a CMS could simplify the process and eliminate the need for an employee to learn web scripting or website database management.

For more information about CMS concepts, considerations and advantages, please see the resources outlined in the Appendix.

## **Implementation of Midgard**

---

### **Initial Considerations**

Before implementing a CMS solution, it was first necessary to ensure that Ungana-Afrika has the technology readiness, content readiness and organisational readiness to take one on. In other words, is there a plan for how the organisation works and manages new information that can ultimately be published? Because Ungana-Afrika performed a website planning exercise when first developing the initial website content and structure, everyone was confident that existing documentation and publishing methods were clear and efficient. As a result, it seemed the best course of action was to first adapt the existing site structure to function within a CMS, and then draw upon new functionality only after this was achieved.

Next, Ungana-Afrika needed to consider which CMS to implement as there is a daunting selection of systems and system types available. The project agreement with the Finnish Embassy specified that Ungana-Afrika would pilot an Open Source solution preferably developed by a Finnish entity<sup>2</sup>. One popular choice that fit this description was Midgard, which is a complete open source LAMP (Linux, Apache, MySQL, PHP) CMS framework<sup>3</sup>. Upon additional research, it was clear that Midgard is quite an advanced CMS that offers a great level of customization and functionality, but also requires detailed computer knowledge to implement and take complete advantage of. However, as Ungana-Afrika has some in-house ICT expertise, it seemed that Midgard would be manageable, and that the additional functionality would be quite beneficial for future needs. Therefore, Midgard was selected.

### **Midgard Training**

Most CMS solutions require a great deal of upfront setup before their benefits can be felt, and Midgard is no exception. Despite the fact that Ungana-Afrika has internal ICT expertise (beyond what many small Southern African organisations would have), they were unable to begin the implementation process without external assistance. As a result, in early February 2004, Ungana-Afrika hosted a week-long Midgard CMS training program headed by Finnish Midgard creator and developer Henri Bergius of Nemein software. During this training, Mr. Bergius covered several topics in depth including:

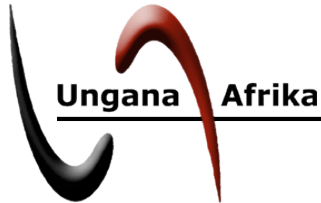
- Conceptual overview of a CMS system and various CMS options
- Overview of the Midgard CMS framework
- Installing and configuring Midgard CMS and related database packages
- Working with Midgard CMS; setting up templates, adding content, etc.
- The application of Midgard and CMS systems for the nonprofit community

After this session the attendees were all able to create a basic website using Midgard and handle the basic server-side administration duties that accompany a CMS package like Midgard.

---

2 In some cases, the CMS selection process can be more intensive than the actual implementation, and most organisations would spend much more time with this step.

3 [www.midgard-project.org](http://www.midgard-project.org)



### **Transfer of Existing Content to Run on Local Midgard Setup**

Throughout early 2004, Ungana-Afrika worked with the Midgard CMS system, and managed to convert its own website to run locally within Midgard. In order to accomplish this, Ungana-Afrika setup its local Linux server to run the Midgard framework and all related server packages, thus duplicating a real web hosting environment<sup>4</sup>. The raw content of the previous Ungana-Afrika.org website was then re-created from scratch using the Midgard CMS system on this server. This affords Ungana-Afrika the chance to test the dynamic functionality of the Midgard on a fully-functional Linux web-server, and to keep a local version of the site for development and backup purposes. Unfortunately, this process proved to be quite challenging, as the Linux/LAMP learning curve was very steep due to a great deal of server-side customizations that are necessary for Midgard.

Most of this initial setup was performed by the same person who administered the previous static website structure. However, after the process was complete, any staff member was able to locally contribute simple real-time content much the same way they would use a word processor. Furthermore, the complete graphical structure and layout of the site was preserved while creating a formatting standard for all content additions. As a result, all users had the freedom to focus on content, while the CMS automatically took care of layout and structural considerations (taking this burden off the web master). So within a local context, Ungana-Afrika was able to show immediate advantages.

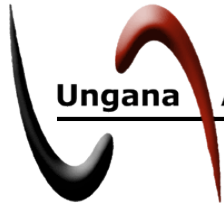
### **Attempts to Host a Public Version of the Midgard-Driven Site**

Because the local web server is not public (it is not open to the rest of the Internet), Ungana-Afrika began working with its long-time web host, Kabissa, to host a public Midgard-driven version of the site content that was produced locally. Unfortunately, this proved to be a request that is not realistic for Kabissa, due to the same installation complexities that Ungana-Afrika experienced in its own local implementation. Kabissa experimented with the process, but concluded that they are not comfortable implementing a complete setup on their public server which already hosts hundreds of websites (and thus cannot experience downtime). This left Ungana-Afrika with two primary options for a Midgard solution:

- Choose a Midgard-Specific Host. Other web hosts offer Midgard-specific hosting, and can guarantee immediate hosting of any Midgard-created site. However, the Midgard hosts that were recommended to Ungana-Afrika were more expensive and less customizable than Kabissa, and therefore not realistic options. Furthermore, as Kabissa is a non-profit host that caters specifically to the African development community, it (or local hosts like it) may be the best options for other African non-profit groups. As a result, Ungana-Afrika wants to stay true to solutions like Kabissa, that its own client base will be most comfortable with.
- Install Midgard Lite. There is a limited install of Midgard called "Midgard Lite," which is much less intrusive to install on a server (it does not require special server-side alterations the way the full version does). However, this version is also very limited in features and functionality. In fact, the lead developer and our contact person in Nemein, Henri Bergius, advised against this option for an application like ours.

---

<sup>4</sup> setting up a local web-server is a task most non-profits will leave in the hands of an external hosting provider, but it is necessary that Ungana-Afrika understand the hosting requirements of a CMS, and how non-profits must interface with external hosting providers



So, as noted, neither of these alternative options were realistic solutions and the process of using a permanent public Midgard solution needed to be abandoned.

### **Implementation Conclusions and Next Steps**

Though Midgard has proven to be a very powerful CMS, its complexity and installation limitations have made it a truly unrealistic solution for Ungana-Afrika at this point. Furthermore, as Ungana-Afrika's web needs are relatively simple, it is believed that other, less complex/intrusive options can meet current requirements instead (while still providing the fundamental CMS advantages Ungana-Afrika is seeking). Midgard has proven to be a powerful and reliable solution for enterprise applications, and organizations that are looking to operate interactive and detailed web portals or intranet sites (and have the means to maintain such resources). However, on a small-scale, small-budget application, with limited internal LAMP skills and without an internally managed server, it does not seem to be an applicable solution.

However, it should be noted that many efforts are taking place from the Midgard developers to address these complications and make Midgard a more installation and user friendly without sacrificing functionality. As a result, Midgard should not be ruled out for future consideration<sup>5</sup>.

Currently, Ungana-Afrika is continuing to use a static Dreamweaver solution, as it still meets immediate needs. In order to further evaluate this decision, and the competitive differences between this static HTML solution and a CMS solution, Ungana-Afrika created a critical comparison that appears in the Appendix. This comparison is based on research that happened throughout the project, and highlights the advantages and disadvantages of each solution in different contexts. From this evaluation, it was clear that a static solution is acceptable for current "brochure-website" needs (assuming an in-house web master is available), but more advanced needs will most likely be better matched with a CMS. Though this breakdown is by no means exhaustive, it has helped Ungana-Afrika justify the continued use of a static HTML solution, until internal needs dictate otherwise or more research can be performed.

However, in order to prepare for future needs and changing requirements, Ungana-Afrika has created a short-list of CMS options that should be researched as a priority for internal use, or as a recommendation to other clients. This list includes packages such as Drupal, EzPublish, Mambo, PageTool, etc.

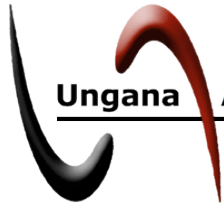
### **Applicable Lessons and Resources for other Implementations**

---

Ungana-Afrika works with a wide variety of organizations, all of which have special needs when it comes to the scale and complexity of their web presence. As a result, there is no one single CMS solution that will fit everyone's environment. Some organizations seek to upgrade to a more complex and interactive website while others simply need a tool that gives them the power to create a basic website from scratch without special web or computer skills. As there is a wealth of open source website management tools available, different solutions can be better adapted for different environments. Because of this, Ungana-Afrika has been researching, and will continue to research, several

---

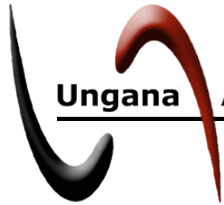
<sup>5</sup> More details about these advancements can be found in Henri Bergius blog space at: [http://www.bergie.iki.fi/blog/making\\_open\\_source\\_content\\_management\\_suck\\_less.html](http://www.bergie.iki.fi/blog/making_open_source_content_management_suck_less.html)



supplementary tools and concepts in addition to the Midgard CMS solution that has been piloted locally. In addition, the following notes are offered as a result of this implementation experience for other small organisations considering a CMS.

- Availability of connectivity is an important factor for low-resource applications. Most CMS systems are designed such that updates are performed through a live web interface. As a result, an Internet connection is required for all the time spent making changes (editing and publishing time). Some small organisations that do not have this option may benefit from an off-line solution that only requires a connection to publish changes after they have been made. This can be achieved easily with a static HTML solution, or by implementing an off-line server from which to replicate changes. In this case study, Ungana-Afrika had both a permanent Internet connection, and the means to run an off-line server. Please note that practical considerations may be very different in applications where these options are not available.
- In-house skills can vary dramatically in low-resource and small-scale applications. In this case study, Ungana-Afrika had basic internal expertise related to web concepts and web publishing, and the ultimate conclusions drawn reflect this. In environments where this in-house experience does not exist, external support considerations will play a much larger role. In these cases, the CMS selection process may also take on a very different form. This is because some CMS solutions are designed for simplicity while sacrificing customisability and features. These systems may provide better alternatives to static HTML solutions or full-featured CMS options.
- A CMS is never a replacement for good website planning and organisational content development strategy. This case study did not touch upon the importance of these considerations. Good website planning and content development must be in place before any CMS or website management tool can be successfully considered.

Again, every situation is unique, and requires sufficient research, planning, and dedication. Furthermore, Ungana-Afrika hopes to build off of this internal case study example to provide application-specific support to a variety of organisations looking to build or expand their web presence.



## Appendix – Additional Resources

### Web Resources

Check out the following web links for additional details about CMS solutions for the non-profit community, and general CMS concepts:

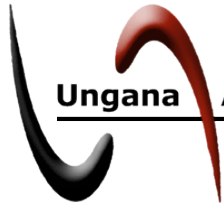
- A good summary of CMS concepts from TYPO3:  
[http://typo3.com/What\\_is\\_a\\_CMS.1351.0.html](http://typo3.com/What_is_a_CMS.1351.0.html)
- A nice selection of CMS resources from TechSoup:  
[http://www.techsoup.org/search/moreresults.cfm?query=cms&type=howto&pro\\_number=1&howto\\_number=9&comm\\_number=68&other\\_number=8](http://www.techsoup.org/search/moreresults.cfm?query=cms&type=howto&pro_number=1&howto_number=9&comm_number=68&other_number=8)

### Comparison Between Static HTML and CMS Solutions

The following tables were generated by Ungana-Afrika in an effort to outline several advantages and disadvantages between static HTML website and CMS solutions within different applications. See the key below for an explanation of each application example.

<b>Term Key</b> (clarification of row labels)	
All Applications	Applies to all applications
"Simple" Applications	Applies epically to "simple" sites that do not require special interactive functionality or complex structure. Usually small in scale.
"Complex" Applications	Applies to more "complex" sites that may incorporate a detailed structure of many layers along with interactive features. Usually large in scale.

(cont. on next page)



**Static HTML Driven Sites:**

	<b><i>Advantages</i></b>	<b><i>Disadvantages</i></b>
All Applications	<ul style="list-style-type: none"><li>• Off line Editing options are built-in</li><li>• Not dependent on server scripting</li><li>• Not many "non-standard" concepts</li><li>• Many editing options are available (dreamweaver, frontpage, etc...)</li></ul>	<ul style="list-style-type: none"><li>• Specific skill required to add content (HTML, Dreamweaver, etc.)</li><li>• Hard and unrealistic to update frequently</li><li>• Software purchase usually required</li><li>• Error prone as some structural concepts are human-dependent</li></ul>
"Simple" Applications	<ul style="list-style-type: none"><li>• Simple administration<ul style="list-style-type: none"><li>• Easy to manage and organize centrally</li><li>• simple setup</li></ul></li><li>• No separation of content and presentation (complete WYSIWYG). This is useful for simple sites</li></ul>	
"Complex" Applications		<ul style="list-style-type: none"><li>• Very static, so complex features and structure and problematic</li><li>• Dependent on one person to publish content</li></ul>

**CMS Driven Sites:**

	<i>Advantages</i>	<i>Disadvantages</i>
All Applications	<ul style="list-style-type: none"> <li>• Workflow Options</li> <li>• Content contributors don't need special skills</li> <li>• Many people can drive content easily</li> <li>• Fast update processes encourage dynamic content</li> <li>• No editing software required</li> </ul>	<ul style="list-style-type: none"> <li>• Often reliant on Internet connection (offline functionality is limited or complicated – local server etc..)</li> <li>• Editing functionality is limited by CMS</li> </ul>
"Simple" Applications		<ul style="list-style-type: none"> <li>• Requires special admin to setup and maintain (change structure, etc.). This may mean an <i>ongoing</i> external consultant, and may be excessive for simple sites</li> <li>• Dependent in server-side scripting (hosting features) which may be excessive for simple sites</li> </ul>
"Complex" Applications	<ul style="list-style-type: none"> <li>• Modular structure makes complex functionality easier</li> <li>• More complex web scripting and database management are handled by CMS, so advanced interactive features are easier to implement/manage</li> </ul>	