An Enterprise Content Management Solution Based on Open Source

Rogerio Atem de Carvalho

Federal Center for Technological Education of Campos (CEFET Campos), R. Dr. Siqueira, 273, Campos/RJ, CEP 28030-130, Brazil ratem@cefetcampos.br

Abstract. Spread out on the Internet, mail servers, and hard-drives everywhere, unstructured information in the form of Web pages, email, RSS feeds, office documents, images, video, and sound, accounts for approximately ten times the structured information stored in databases. Seeking to organize their large volume of non-structured content, companies implement Enterprise Content Management (ECM) systems to make this information more accessible for users and make these systems communicate with relational database backed solutions through a single interface. Small and medium enterprises (SMEs) and local governments also have the same need for unstructured information organization. However, most of times, they cannot afford the high acquisition and customization costs, or don't want to become dependent of proprietary ECM solutions. This paper aims to present NSI^2, an ECM solution totally built on top of open source software that offers the functionalities demanded by this kind of system.

Keywords: *Enterprise content management, EIS for public sector, Enterprise information integration, Open source, Small and medium enterprises*

1. INTRODUCTION

Scattered on the Internet, mail servers, and hard-drives everywhere, unstructured information in the form of Web pages, email, RSS feeds, office documents, images, video, and sound, account for approximately ten times the structured information stored in databases. The ever growing convergence of Information and Communication Technologies confirms this tendency for the years to come. Seeking to organize their large volumes of unstructured content and integrate them to the traditional, database backed enterprise systems, companies implement Enterprise Content Management (ECM) systems to make this information more accessible for users through a single point of access. Small and medium enterprises (SME), education and research institutions, and government organisms also have the same need for unstructured information organization and integration to foster their business processes and become compliant to regulations and modern accountability practices. However, most of times, they cannot afford the high acquisition and customization costs - or don't want to become dependent - of proprietary ECM solutions. This paper aims to present NSI^2, an ECM solution totally built on top of open source software, that offers all the basic - and some not so basic - functionalities demanded by this

kind of system. NSI^2 is a play with the acronym of Information Systems Research Group (NSI in Portuguese, where it is developed), and Non-Structured Information (NSI).

This paper is structured as follows: after this introduction, the main issues on ECM and how NSI^2 address them are presented. Its content management structure comes next, followed by sections describing its infrastructure and current applications. Finally, concluding remarks are given on the last section.

2. NSI2 AND ENTERPRISE CONTENT MANAGEMENT

Enterprise Content Management (ECM) integrates the management of structured, semi-structured, and unstructured information, and related software and metadata in solutions for content production, publication, utilization, and storage in organizations [1], emphasizing the coexistence of technical and social aspects within the content management [2]. It is an emerging topic that have been attracting new commercial and academic players, including the professional forums such as AIIM International, which defines itself as "the ECM association" (http://www.aiim.org).

ECM is related to, and some times confused with, a series of other topics, which include Knowledge Management, Content Management Software, and Web Portals. However, it represents a new area since it focuses on the management of textual and multimedia content across and between enterprises, seeking to establish an integrated perspective on information management, through the integration of content with organizational databases and applications [1]. Moreover, like every enterprise-wide initiative, an ECM system deployment raises many issues. How NSI2 addresses issues identified by [1], [3], and [4], are now briefly discussed.

2.1 Return on Investment (ROI)

ROI is calculated as the present value of the benefits divided by the present value of costs [3], in other words, a combination of cost savings, improved profit margins, and sales enhancements. However, ECM ROI normally is associated to cost savings only. According to [1], these savings can be obtained through a composition of different cost drivers, like reduced time on information searching, lower printing costs, and multi-lingual technical publishing. On the other hand, other indirect benefits must be considered, like the need to answer to external and internal regulations and the development of future capabilities on enterprise systems technologies. Obviously, investment costs also must be taken into account [3]: software licensing fees, hardware and physical infrastructure, bandwidth, support, and management.

If the benefits can vary for each case, costs savings can be obtained for almost all cases – at least for small and medium deployments - by the use of Free/Open Source Software (FOSS) and commodity hardware. In fact, the sole use of FOSS, in general, reduce the Total Cost of Ownership (TCO) [5] [6] and, simultaneously, broaden the user base for a given solution [7]. Following this approach, NSI^2 is totally based on

FOSS components that can run on clusters of commodity hardware, since it is aimed at SME, educational institutions, and local governments. For these cases, cost has been a major hindrance to the widespread ECM adoption, in special in developing countries. Since NSI^2 is a young solution, detailed ROI calculations are yet to be obtained, but observed cost reductions on its first user organization is about 65% in hardware, and 100% in software licensing.

2.2 Content Model

The core of any ECM solution is to understand the role of content in the organization context [1]. According to [8], content is normally organized under two different approaches: database-based and document-based. The first evolved from the database community and consists basically of storing documents and their components as database objects, while the other emerged from the publishing communities, and consists basically of defining document templates. Relying on Zope, NSI^2 uses a hybrid approach, based on the concept of *typed content*, where document templates are defined as classes. Therefore, according to the specific necessities, a content element can be treated as a document or as an object. While keeping all the advantages of an object stored in Zope Object Database (ZODB), like versioning, use of semantic relations, semantic checking, and undo; for content producers it is a document that can be edited (using a built-in text editor) and exported as a XML object if needed. Objects are stored in a hierarchical folder structure, and are accessed through their URL, which is constructed from the server name followed by the object path in ZODB. Zope also offers default templates that provide different types of metadata that can be customized for specific uses. Following the types identified by [9], template sheets provide content (keywords, title, language and other Dublin Core fields), data lineage (date created, date modified, change log etc), and technical (content type, encoding etc) metadata. Using object-orientation, new types can be created by both composition and inheritance from basic template sheets.

2.3 Infrastructure

According to [1], for wide-scale ECM initiatives, information technology involves a number of challenges, all addressed by NSI^2:

1) Integration of standardized applications and tools: NSI^2 uses Zope's features on relational databases access and XML processing for integration with other applications. When needed, Python scripts can be implemented to build-up more sophisticated integration routines.
2) Developing user-friendly content management: the Zope based Plone Content Management system is used to manage content in NSI^2. This tool allows users to edit documents through the web, and specific workflows can be created to administer content publishing.
3) Updates in software and hardware (scalability): NSI^2 relays on a series of FOSS with strong developer communities that have been keeping the software up-to-date for years. In terms of hardware, clusters of commodity

hardware are used, which reduces costs and the need for more specialized support personnel.

4) XML and other open format compliance: although Zope has its own object-oriented storage formats and algorithms, it is possible to convert every object into a XML document. In the contrary way, XML objects can be easily converted into Zope objects.
5) Information security issues: content is secured through Zope's fine-grained and flexible access routines, built around the concept of "safe delegation of control", that allows to turn control over parts of collections of documents to certain user roles. More restricted security measures can rely on proxy roles, where a user only has the right to run a certain script that indirectly execute operations on ZODB.

2.4 Customization

Customization is a key issue for ECM, due to its considerable costs, being functional customization the main issue on this matter. ECM customization and can be divided in three main areas:

1) Content Model Management: functionality for structuring of content, metadata model, taxonomy, and templates.
2) Content storage and delivery management: user roles, versioning, classification, transformation, retention and tracking.
3) Process support and automation: workflows.

For the first two areas, besides taking advantage of Zope and Plone features, NSI^2 extends both to manage automatically fragments of documents and store huge collections of indexable objects, as will be seen afterwards in this article. Workflows are used as Zope offers them.

3. NSI2 CONTENT MANAGEMENT STRUCTURE

NSI^2 content management structure is built on top of Plone (http://www.plone.org), a Content Management Systems (CMS) based on the Zope (http://www.zope.org) object publishing and application server. Extensions are also provided to enhance the CMS capabilities of Plone. The next topics will discuss briefly each of these components of NSI^2.

3.1 Zope and Plone

Zope stands for the Z Object Publishing Environment, it is and open source web application server implemented in Python language, and can run on a variety of operational systems that include Unixes, Linux, Windows, and Macintosh. Zope offers an integrated solution for web applications that relies on an object oriented and transaction enabled database (ZODB), an administrative interface (ZMI), a web

server, a search engine (Zcatalog), and a workflow engine. Also, it can connect to all major relational database management systems, work together with Apache, and is XML-RPC, DOM, WebDAV and other open standards enabled. It also offers more than three hundred plug-ins (called products) developed by the community, which address e-Commerce, content, integration with other tools and environments, helpers, internationalization, navigational, visualization, and user management. It also has around it a strong and active community of developers that interact through mail lists and wikis that account for thousands of entries a month. In other words, Zope supplies the entire infrastructure to develop web-based applications, in special, content-driven solutions, its original focus.

Plone is a CMS that works on top of Zope. It is highly extensible, with more than six hundred plug-ins for a variety of uses such as layout and presentation, versioning, communication, object storage, media management, project management, calendars, statistics, and many others. It offers all basic content types like folders, links, files, documents, events, images, news items, and dozens of more sophisticated types in the form of plug-ins. All content types have a rich metadata set associated to it, based on the Dublin Core Metadata Initiative (http://www.dublincore.org), providing content, lineage and technical metadata. Content objects can have keywords, historical data, type, and even discussion threads associated to them. Using specific plug-ins it is possible to have enhanced search options through the implementation of ontologies.

Plone is highly extensible and even users with very basic programming skills can create new content types, define views and security roles, and implement workflows to manage the new type. Using a code generation tool denominated ArchGenXML, it is possible to model in UML new sets of related content types, managed by specific workflows, and generate all the code necessary to implement a new application. Also, new layouts and default content views can be configured through a series of specific forms, liberating content managers of deep HTML knowledge. It also supplies basic accessibility features.

Analyzing other ECM and CMS solutions, like the one presented in [11], it is clear that the Zope/Plone infrastructure addresses all the basic features demanded by this kind of applications. More than that, it can be used to implement highly sophisticated Knowledge Management solutions [12].

3.2 NSI^2 Extensions

A Web Services based architecture [13] is envisioned for future developments of NSI^2, however, current efforts focuses on improving user experience and solution scalability, as described by the following topics.

Granular Document

An ECM system must be a content application tailored to a specific audience, instead of a limited search engine. According to [14], modern content applications should deal with the representation of semi-structured content – stored and indexed securely in a content repository. Following this reasoning, NSI^2 extends Plone with a new type, named Granular Document. Granular documents can be divided into grains

or fragments according to their representation inside the document. For instance, a research article stored as granular document can have its paragraphs, tables, formulas, and figures indexed and accessed separately from the composite (the own document). This concept is related with the learning objects concept from distance learning [15] and fragment concept from caching of web pages [16]. The first concept refers to those pieces of information inside a learning content that can be identified themselves as smaller learning objects. Fragments are pieces of information that can be reused in different contents and, therefore, can be served and cached as such, reducing network traffic and I/O tasks, and also improving user experience.

Granular Document is implemented in a very straightforward way: the XML representation of a document is parsed for identifiable grains of information, like paragraphs, images and tables. These grains inherit the metadata from the owner document and also have their own metadata and text automatically associated in proper indexes. For instance, an image with a specific caption can be found through its owner document's full text and metadata, and also by its own metadata (for instance, related to its format) and the caption text. When a user opts for a granular search, he or she will get a results page with a hierarchy of documents and their respective grains. Granular search can be restricted by all kinds of metadata, like format, type, date of upload, author, and such, to reduce the result set size. Moreover, if ontology is in use, and the user also opts for an enhanced granular search, the grains inherit the semantic network associated to the owner document. In that way, it is possible to give users the option of a more focused search, like "return all images about fuzzy aggregation operators" – where image is the grain type and "fuzzy aggregation operators" the full text term to search for. If the search is based on ontologies, documents containing related terms and their grains would be returned too.

When created, a Granular Document object first converts HTML or MS-Office to XML and then parses it in the search for grains. Documents in ODT format, like the ones from Open Office, are unzipped and then parsed. The user can view and save documents and grains in plain HTML, PDF or in its original format. These options can be configured by the system administrator, according to the enterprise content reuse policies. It is important to note that, since grains are also objects, they are cached separately from their owner documents.

Self Service

Self-Service is a product that works on top of Granular Document and is similar to solutions like SafariU (http://www.safariu.com), which let users build new textbooks from parts of existent ones, providing automatic table of contents and index. Self-Service allows a user to search a NSI^2 catalog and tick the grains (or whole documents) of interest. Then Self Service will build a new document, in HTML or Open Office format, with the selected grains, including the source for each object and a complete reference list at the end. With this document the user will have a basis for creating new content, like reports, articles, news, and learning material.

Enhanced User

Enhanced User is a way of stimulating the continuous use of the solution, giving registered users (not simple guests) features like search storage, push and RSS services according to selected document categories, file and grain cabinet, polls and a series of customization options. Also included is a Personal Information Management (PIM) structure that can hold email with full text search (including in attachments, even when zipped), personal contacts, *personal* calendar, text notes, and instant messaging.

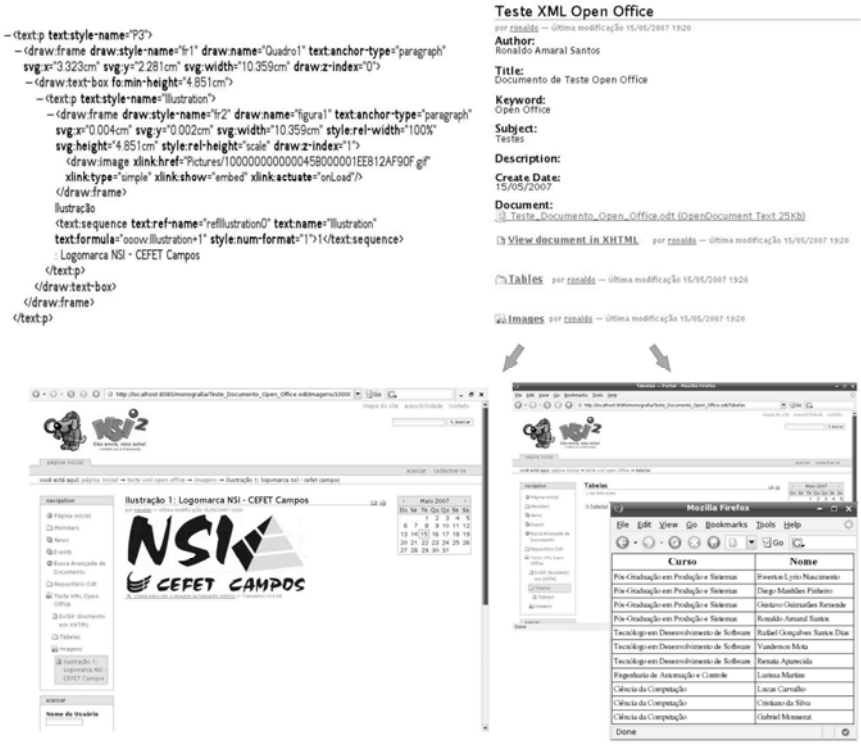

Figure 6. Granular Document Grain Extraction From XML That Represents the Original Document. Grains Can be Opened and Saved in Original, HTML or PDF Format.

Enhanced File System Storage

Zope stores objects and indexes inside its object database, ZODB, which, however, looses efficiency when it comes to store more than tens of gigabytes of objects. Moreover, no software is better than the operational system to store and serve large collections of files, making it a better choice, providing that a good retrieval method is offered. Although Plone already offers a content type (FSFile) that keeps metadata on ZODB and stores the file itself on a specific directory in the file system, this solution also have some limitations when dealing with a large number of files in the

same place. To overcome these deficiencies, NSI^2 extends Plone by offering a content type called Enhanced File System File (EFSFile).

This new type uses the same philosophy of FSFile, however, it is optimized to transparently work with the PVFS2 file system and the Lucene search engine, which are used to enhance NSI^2 scalability, as will be presented in the Deployment Architecture section. When referenced on a search result page, EFSFile objects are viewed as ordinary Zope objects, but their content and index entries are stored outside ZODB. In that way, file serving and search processing can be delegated to specialized software running on dedicated clusters, while developers still take advantage of all object-oriented resources of Zope, and users still access these objects as "first-class content", since for both groups, it is like the objects were inside ZODB.

4. NSI2 DEPLOYMENT ARCHITECTURE

An ECM solution to be effective must be scalable and be available in a 24x7 fashion. In NSI^2, scalability and availability are understood as a composition of three elements: access, storage, and indexing. The structure used to comply to these requirements (Figure 2) is based on a set of open source software that guarantees both load balancing and failover:

Access: provided by LVS (Linux Virtual Server) servers that distribute HTTP and FTP requests over a farm of Zope servers. Additionally, ZEO (Zope Enterprise Objects) cluster makes Zope servers share a common content objects collection.

Storage: the PVFS2 distributed file system stores files in a cluster of storage servers, as they were a single server.

Indexing: the Lucene search engine is a more scalable and manageable replacement to the built-in Zope search and indexing functionalities, allowing the setup of a grid of machines exclusively dedicated to run parallel searches and store the indexes.

Using this structure, if one server fails, another one is automatically elected to replace it, in a transparent way. Moreover, it is possible to add new servers "on the fly" to any of the clusters, thus keeping the quality of services when the number of users and information stored and indexed grow.

5. APPLICATIONS

Besides various smaller applications for local governments, NSI^2 is the basis of a series of initiatives sponsored by the Secretary for Professional and Technological Education of the Brazilian Ministry of Education (Setec/MEC) in collaboration with Unesco. Those initiatives offer a series of services for the Professional and Technological Education (PTE) network, which accounts for approximately three thousand public and private institutions in Brazil. Starting on 2009, NSI^2 will form the basis for extending these services to Mercosul (common market for South American countries) and Portuguese-speaking countries in Africa, like Angola and

Mozambique. This deployment of NSI^2 based solutions in development countries shows the two-pronged importance of the use of FOSS to implement it: reduce TCO and foster information technology development – through the availability of all source code and documentation. Currently, four different NSI^2 applications are in development, all in the realm of the PTE:

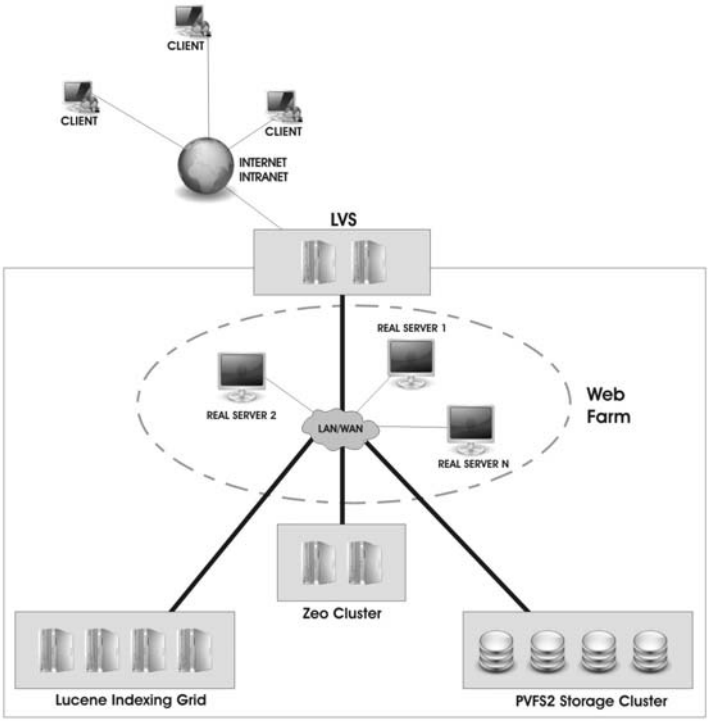

Figure 2. NSI^2 Deployment Architecture: Users View a Single Point of Access for the ECM Solution

1) Digital Library: a library of articles and thesis will concentrate the academic work on PTE in Brazil. Workflows for revision, translation, metadata association, and publishing will support the correct upload of these documents.
2) Digital Document Center: using a series of Open Office templates and proper workflows, the administrative documents from Setec/MEC will be generated, managed, and stored using NSI^2. In the medium term, taking advantage of the structure offered by the Granular Document objects, a XML Datawarehouse [17] will be created, allowing analytic operations on the document collection stored in the solution. This document warehouse will be used for statistics, governmental accountability, and historical data analysis purposes.
3) Observatory of the PTE: similar to the Digital Library, it stores analytical documents on various statistics about labor positions and alike related to the PTE. Using the Self-Service option for Granular Documents, new reports and case studies will be better supported.

4) Learning Objects Catalog: a catalog that will maintain references to learning material stored on various PTE institutions. Also using the Self-Service functionality, new learning material can be built from the material obtained through the catalog.

These applications are all in prototype state, and are scheduled to be fully functional by the end of this year.

6. CONCLUSIONS

The solution here presented currently has limitations related to advanced search capabilities, like natural language usage. Also, image and sound files searching are limited to their metadata. However, it is envisioned that these capabilities can be incorporated in the future, thanks to its flexible, object-oriented architecture.

Although NSI^2 is aimed at smaller budgets, it is believed that this solution is compliant to most ECM needs. In fact, even bigger private and public organizations can be satisfied by the use of a proven content management system, a platform that easily integrates to other platforms using open standards, and the scalable infrastructure and extensions provided.

REFERENCES

1. T. Paivarinta and Munkvold, B.E. Enterprise Content Management: An Integrated Perspective on Information Management, in *Proc. of the 38th Annual Hawaii International Conference on System Sciences* (2005).
2. P. Tyrvainen, A. Salminen, and T. Paivarinta, Introduction to the enterprise content management minitrack, in *Proc. of the 36th Annual Hawaii International Conference on System Sciences* (2003).
3. H.E. McNay, Enterprise content management: an overview, in *Proc. of the IEEE International Professional Communication Conference* (2002), pp.396-402.
4. S. Nordheim and T. Paivarinta, Customization of enterprise content management systems: an exploratory case study, in *Proc. of the 37th Annual Hawaii International Conference on System Sciences* (2004).
5. B. Fitzgerald and T. Kenny, Developing an information systems infrastructure with open source software, *IEEE Software*. Volume 21, Number 1, pp.50-55, (2004).
6. S. Krishnamurthy. An Analysis of Open Source Business Models, *Perspectives on Free and Open Source Software*, eds. M. Cusumano,C. Shirky, J.Feller, B. Fitzgerald, S.A. Hissam, K.R. Lakhani (MIT Press, 2005), pp.279-296.
7. D. Riehle, The Economic Motivation of Open Source Software: Stakeholder Perspectives, *IEEE Computer*. Volume 40, Number 4, pp.25-32, (2007).
8. M. Grossniklaus and M.C. Norrie, Information concepts for content management, in *Proc. of the Third IEEE International Conference on Web Information Systems Engineering* (2002), pp.150- 159.
9. W. Kim, On metadata management technology: status and issues, *Journal of Object Technology*. Volume 4, Number 2, pp.41-47, (2005).

10. S. Nordheim and T. Paivarinta, Customization of enterprise content management systems: an exploratory case study, in *Proc. of the 37th Annual Hawaii International Conference on System Sciences* (2004).
11. D. Krechel, M. Hartbauer, and K. Maximini, LENUS - The Hospital Content Management System, in *Proc. of the 19th IEEE International Symposium on Computer-Based Medical Systems* (2006).
12. J. Hartmann and Y. Sure, An infrastructure for scalable, reliable semantic portals, *IEEE Intelligent Systems*. Volume 19, Number 3, pp.58-65, 2004.
13. K.H.S. Kwok and D.K.W. Chiu, A Web services implementation framework for financial enterprise content management, in *Proc. of the 37th Annual Hawaii International Conference on System Sciences* (2004).
14. S. Buxton, Beyond search: content applications, *IEEE IT Professional*. Volume 9, Number 1, pp.29-35, (2007).
15. A.A. Khaing and N.L. Thein, Efficiently Creating Dynamic Web Content: A Fragment Based Approach, in *Proc. of the 6th Asia-Pacific Symposium on Information and Telecommunication Technologies* (2005).
16. P.A. Moumoutzis, N. Christodoulakis, and S. ASIDE, An Architecture for Supporting Interoperability between Digital Libraries and eLearning Applications, in *Proc. of the Sixth IEEE International Conference on Advanced Learning Technologies* (2006), pp.257-261.
17. V. Nassis, T.S. Dillon, R. Rajugan, and W. Rahayu, An XML Document Warehouse Model, in *Proc. of the 11th Int. Conf. on Database Systems for Advanced Applications* (2006).