AUTOMATYKA • 2007 • Tom 11 • Zeszyt 1–2

Grzegorz J. Nalepa*, Antoni Ligęza*, Igor Wojnicki*

From Content to Knowledge: a Perspective on CMS**

1. Introduction

The Internet has become a single most important resource for instant information sharing. From the point of view of ordinary users it can be considered to constitute a very flexible and powerful version of the concept of the so-called *blackboard architecture*. The rapid growth of the Internet prompted a rapid development of specific software. This software addresses different aspects of information organization and interchange related issues, such as storing, retrieval, searching, indexing, aggregating, sharing, updating, etc. Taking into account the size and flexibility of the system as a whole, these problems constitute a real challenge.

Content Management Systems (CMS for short) emerged as a technology providing a unified interface for large databases or data warehouses. In a broad sense a CMS consists of: a RDBMS (Relational Data Base System) which stores the data, and a complex web-based application or interface providing the access to the data. The web interface is built around using common web technologies. This software requires a proper and efficient runtime environment, including a web server. An important aspect of a CMS system is the possibility of user personalization, tracking, and interactive use.

Today web technologies constitute an advanced and universal programming framework. This framework has a very heterogeneous structure including: data encoding and structuring languages (such as HTML and XML), meta-data languages (such as RDF), data transformation languages (such as XSL/T/FO), data presentation languages (such as CSS), server-side programming languages (such as PHP, JSP), and client-side programming languages (such as Javascript).

Rapid development of CMSs has been stimulated by the so-called opensource LAMP platform, which provides a vary portable and accessible deployment environment for a CMS. LAMP originally stood for “GNU/Linux Apache SQL PHP”. PHP is an advanced

* Institute of Automatics, AGH University of Science and Technology

** The paper is supported from the AGH UST grant No 11.11.120.44.

and very common server-side programming platform for developing web applications. First versions of PHP were best suited for Apache, the most important (and most popular) web server on the Internet. SQL has been the first open source RDBMS which provided a relatively high efficiency along with good integration with PHP, today it may be often replaced by some technologically superior solutions such as PostgreSQL. All of these software tools were originally tailored for the GNU/Linux operating system environment, which provided an accessible, efficient, and low-cost deployment platform. Today there is a very large group of FLOSS (*Free/Libre, Open Source Software*) CMS systems.

Multiple CMS categories can be identified erg., web portals, group ware suites, forum sites, and e-learning toolkits. In each of these cases the content they manage differs. From knowledge management perspective this content may be seen as a particular kind of knowledge. In order to design and implement a CMS, knowledge representation methods should be taken into account. While all of the CMS systems provide some kind on knowledge acquisition facilities, few of them focus on using proper knowledge representation method, that would simplify the acquisition and design process.

This paper presents results an critical overview of architecture and main classes of CMS. The overview has been conducted in order to identify main issues related to knowledge representation in CMS. In the paper some practical guidelines for improving existing CMS, and turning them into *knowledge-oriented systems* have been given.

2. CMS Architecture

Content Management Systems (CMS for short) emerged as a technology providing a unified interface for large databases or data warehouses.

In a broad sense a CMS consists of:

– the server side: a RDBMS (Relational Data Base System) which stores the content, and a complex web-based application or interface providing the access to the content;
– the client side: a web browser that renders the content in the visual manner.

The clients are connected to the server via the computer network. It is the Internet in the general case, but it can also be an Internet network, an Internet.

On the server side several several software elements can be identified:

– a RDBMS server which stores the content, and provides a query facility, in case of FLOSS;
– the main CMS application, implemented in some high level language, such as PHP, JSP, or ASP; this application accesses the content RDBMS system using SQL (or in some cases technologies such as ODBC), and encodes the results of the query using XML or HTML;

– the application runtime, such as PHP interpreter, Java Virtual Machine, or ASP run-time;
– the web server, which is integrated with the application runtime, and serves the XML-encoded content via the HTTP protocol;
– the operating system environment (erg. GNU/Linux, BSD, Unix), which provides a proper and efficient run-time for all of the above components, including an efficient TCP/IP network stack.

The main element on the client side is the web browser. In a simplest case its job is to simply render the XML-encoded content. It often has some basic programming capabilities, mainly Java Script. In some cases it is integrated with the JVM (Java Virtual Machine) which allows for full-scale application programming

Considering the architecture, the core software, called the CMS runtime here, can be identified. The runtime includes: an RDBMS, a server side programming language, a web server, content encoding and description languages, web browser, and client-side programming language. For all of this components some well established technologies are provided.

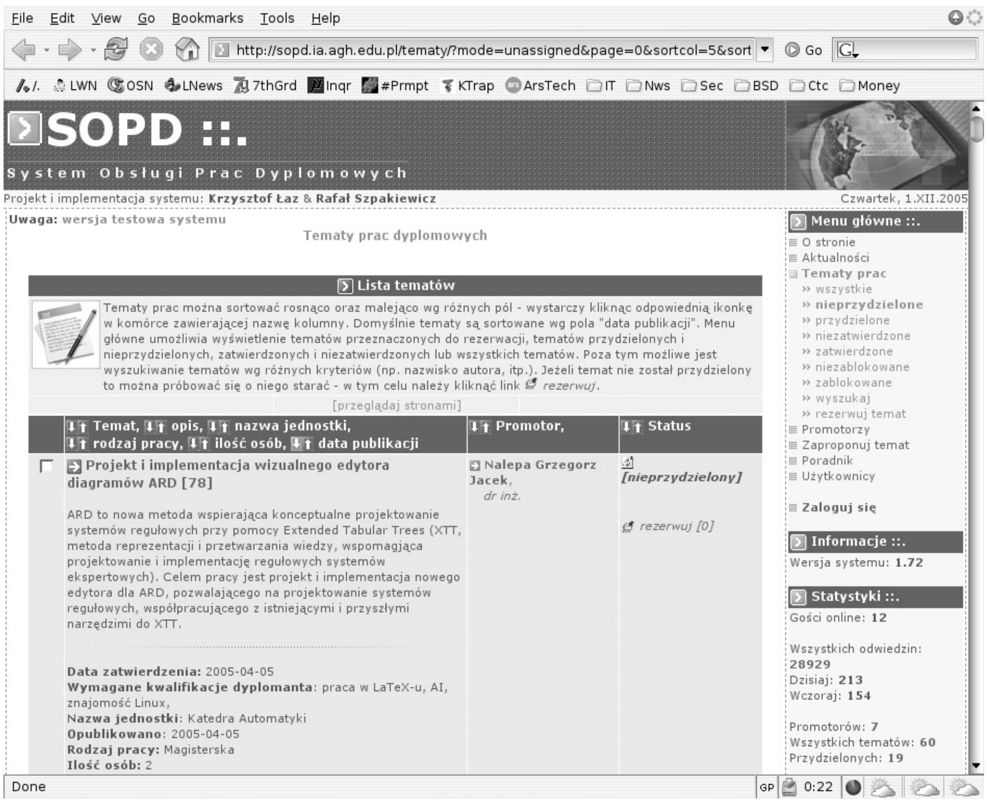
3. CMS Classes

CMS are not a heterogeneous software collection. Several classes or groups can be identified. The main differentiating factor is the type of web system they are oriented for. Basing on research, supported by [1, 2] in the following subsections main CMS classes are identified and described.

3.1. Portals

This is the main class of CMS, grouping the largest number of software solutions. Portal CMS are designed to build a so-called *web portal,* a web site presenting multiple categories of content, with advanced user profiling capabilities. Portals are extensively used by news or domain-specific content providers. They are often combined with advertisement and e-commerce facilities. They provide user accounts in order to profile the content for the specific user, and track user activity. Portals are usually heavily linked with other web sites concerning the content. There is a large number of this class of CMS implementations. Some most mature and well established are: Dru pal (www.drupal.org), Mambo (www.mamboserver.com), Plone (www.plone.org), PHP Nuke (phpnuke.org), PostNuke (*postnuke.com*).

An example application of such a system may be found in [6] and is shown in Figure 1. It shows the SOPD systems, which supports students thesis presentation, and assignment in Institute of Automatics AGH.



Projekt i implementacja systemu: **Krzysztof Łaz** & **Rafał Szpakiewicz**

Czwartek, 1.XII.2005

Uwaga: wersja testowa systemu

Tematy prac dyplomowych

Lista tematów

Tematy prac można sortować rosnąco oraz malejąco wg różnych pól - wystarczy kliknąc odpowiednią ikonkę w komórce zawierającej nazwę kolumny. Domyślnie tematy są sortowane wg pola "data publikacji". Menu główne umożliwia wyświetlenie tematów przeznaczonych do rezerwacji, tematów przydzielonych i nieprzydzielonych, zatwierdzonych i niezatwierdzonych lub wszystkich tematów. Poza tym możliwe jest wyszukiwanie tematów wg różnych kryteriów (np. nazwisko autora, itp.). Jeżeli temat nie został przydzielony to można próbować się o niego starać - w tym celu należy kliknąć link *rezerwuj*.

[przeglądaj stronami]

	Temat, opis, nazwa jednostki, rodzaj pracy, ilość osób, data publikacji	Promotor,	Status
☐	**Projekt i implementacja wizualnego edytora diagramów ARD [78]** ARD to nowa metoda wspierająca konceptualne projektowanie systemów regułowych przy pomocy Extended Tabular Trees (XTT, metoda reprezentacji i przetwarzania wiedzy, wspomagjąca projektowanie i implementację regułowych systemów ekspertowych). Celem pracy jest projekt i implementacja nowego edytora dla ARD, pozwalającego na projektowanie systemów regułowych, współpracującego z istniejącymi i przyszłymi narzędzimi do XTT. **Data zatwierdzenia:** 2005-04-05 **Wymagane kwalifikacje dyplomanta**: praca w LaTeX-u, AI, znajomość Linux, **Nazwa jednostki**: Katedra Automatyki **Opublikowano**: 2005-04-05 **Rodzaj pracy:** Magisterska **Ilość osób:** 2	**Nalepa Grzegorz Jacek**, *dr inż.*	***[nieprzydzielony]*** *rezerwuj [0]*

Menu główne ::.
- O stronie
- Aktualności
- Tematy prac
  - wszystkie
  - **nieprzydzielone**
  - przydzielone
  - niezatwierdzone
  - zatwierdzone
  - niezablokowane
  - zablokowane
  - wyszukaj
  - rezerwuj temat
- Promotorzy
- Zaproponuj temat
- Poradnik
- Użytkownicy
- **Zaloguj się**

Informacje ::.

Wersja systemu: **1.72**

Statystyki ::.

Gości online: **12**

Wszystkich odwiedzin: **28929**
Dzisiaj: **213**
Wczoraj: **154**

Promotorów: **7**
Wszystkich tematów: **60**
Przydzielonych: **19**

Fig. 1. SOPD system page example
Source: own research and [6]

3.2. Group ware

Group ware CMS provide an on-line environment supporting the problem of group work coordination. This includes tasks like: contact management, email and instant messaging support, distributed editor environment, possibly with version control, calendars and schedule controllers, as wheel as message boards, etc. There are numerous applications of such systems, including the support for a large local company, as well as distributed worldwide enterprise; these task can allow for communication with customers, by including a CRM part (Customer Relationship Management). Some popular implementations of such systems include: PhpGroupWare (*www.phpgroupware.org*), WebCollab (*webcollab.sf.net*), NetOffice (*netoffice.sf.net*).

3.3. Forums

Forums provide a complete solution for asynchronous discussion of large number of people. They allow for exchanging ideas between experts. They have also become one of the preferred solution for on-line technical customer support for number of software and hardware companies. Some best examples of forum CMS are: Forum (*www.phorum.org*), PhpBB (*www.phpbb.com*).

3.4. E-Commerce

A special group of CMS systems is oriented towards e-commerce solutions. These are usually dedicated portals, with some additional features built-in. These features include a secure user data management, advanced user profiles and tracking, on-line shopping carts, and complex presentation modules for product catalogs, etc. Several system exist in this class: osCommerce (*www.oscommerce.com*), PhpShop (*www.phpshop.org*), ZenCart (*www.zen-cart.com*).

3.5. E-Learning

CMS are one of key components in on-line e-learning solutions. They support the process of entering the content for e-learning suites as well as the learning process itself. They gain an increasing popularity by adopting some established standards such as SCORM. One of the best examples of such systems is Noodle (*www.moodle.org*), others include: Atutor (*www.atutor.ca*), Dukeos (*www.dukeos.com*).

3.6. Miscellaneous

Besides all of the above systems there are number of other solutions. They support some common on-line activities, such as: logging, wiki creation, personal image galleries, and so one. Some popular systems are: Bologna (*www.bblog.com*), PhpWiki (*www.phpwiki.org*), Gallery (*gallery.sf.net*).

4. Wiki Systems

The wiki *concept* emerged in 90's. The main idea behind was to create a tool for communication and knowledge sharing to be simultaneously simple and powerful. Thus, the main features follow this idea. A wiki system is mainly a collaboration tool. It allows multiple users access, read, edit, upload and download documents. Documents are text based enriched with so-called wiki markup. It is a simplistic, tag based, text only language which allows to annotate text with information regarding the way how it should be presented. Such an enriched text is called Cronkite. The tags allow to make sections, subsection, tables, items and other typographic operational Cronkite remains human readable, tags are intuitive and easy to learn.

Each document is identified by a keyword uniquely, which makes a wiki concept similar to the encyclopedia concept. Furthermore a document, in addition to typographic tags mentioned earlier, can contain hyper links to other documents. A hyper link is a document name enclosed by a link tag. Furthermore a wiki allows to upload images, and other files as well. Wikis are mostly web based. A web interface allows to access a wiki, render pages, and edit them. Depending on particular solutions there might be access control and authorization mechanisms guarding access implemented as well.

One of the most important features of a Wiki is an *integrated version control*. Each page modification is recorded. At any time a user can access a page and its previous versions. A wiki system is usually based on some server side processing technologies providing the web application, and optionally a database backed. As a fronted a web browser is used.

It is worth pointing out that wiki Systems currently blend with CMS. Some CMS provide wiki functionality while some Wilkins evolve into CMS. Similarly Wilkins are more and more often merged with e-Learning systems to support collaborative knowledge gathering and sharing.

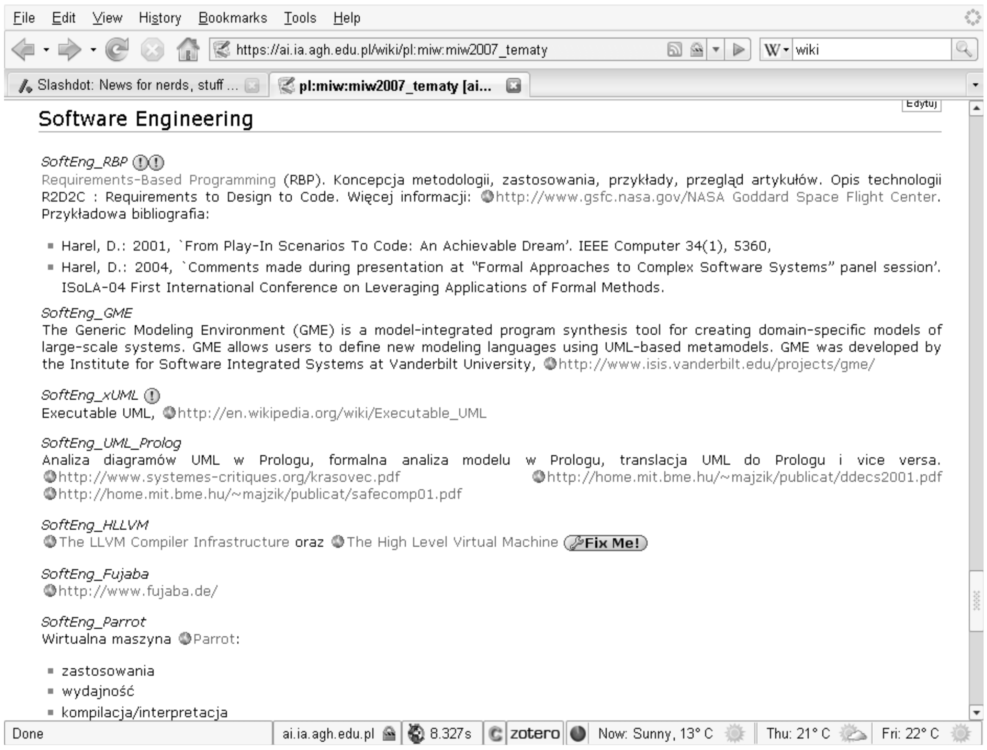
One of the most interesting wiki systems for developers is *DokuWiki* (*http://wiki.splitbrain.org/wiki:dokuwiki*). It is designed to be both easy to use and easy to set up. DokuWiki is based on PHP and does not require a database backend. Pages are stored as versioned text files which enables easy backup-restore operations. It allows image embedding and file upload/download. Pages can be arranged into so-called name spaces which act as a tree-like hierarchy similar to directory structure. It also provide syntax highlighting for in-page embedded code for programming languages such as: C/C++, Java, Lisp, ADA, PHP, SQL and others, using GeSHi (http://qbnz.com/highlighter/). Furthermore it supports extensive user authentication and authorization mechanisms including Access Control Lists (ACL). Its modularized architecture allows to extend DokuWiki with plugins which provide additional syntax and functionality. The most popular plug ins provide: user and ACL management, log, gallery of pictures, discussion board, calendar, and LaTeX math and symbols.

The DokuWiki system has been successfully used in AGH Institute of Automatics as the following applications: Research Project Team Collaboration, Student Projects, Student Collaboration (*https://ai.ia.agh.edu.pl/wiki*). Thanks to the DokuWiki extensive user authentication and ACL mechanisms user collaboration becomes seamless and information access can be controlled with a great precision.

Regarding the Project Team Collaboration the Wiki pages were used to collect and publish domain specific information and data. Project Work Breakdown Structure and progress were stored and updated in the Wiki together with domain specific knowledge. Certain namespaces were chosen to publish information publicly and others to make it visible to the team members only. Since all information stored in the Wiki is subject to the version control feature gradual improvements to the gathered domain knowledge were possible.

DokuWiki was also used as an information platform to hold student project subjects and descriptions. The subjects and descriptions were collaboratively developed by faculty members. Students were not allowed to modify the pages, since the projects were supervised by faculty members. Only faculty were allowed to add or modify information about project assignments and its progress. The third application regarded collaborative knowledge gathering regarding availability of certain technologies. Students were allowed to create, edit, delete and update pages belonging to a given namespace. Since the Wiki syntax is simple and easy to learn the was completed quickly.

DokuWiki proved to be a great tool aiding teaching and research processes. It can act as an easy to use and set up collaborative work environment. An example of the AGH IA wiki is shown in Figure 2.


File Edit View History Bookmarks Tools Help
https://ai.ia.agh.edu.pl/wiki/pl:miw:miw2007_tematy
wiki
Slashdot: News for nerds, stuff ...
pl:miw:miw2007_tematy [ai...
Edytuj
Done
ai.ia.agh.edu.pl
8.327s
zotero
Now: Sunny, 13° C
Thu: 21° C
Fri: 22° C

Software Engineering

SoftEng_RBP
Requirements-Based Programming (RBP). Koncepcja metodologii, zastosowania, przykłady, przegląd artykułów. Opis technologii R2D2C : Requirements to Design to Code. Więcej informacji: http://www.gsfc.nasa.gov/NASA Goddard Space Flight Center. Przykładowa bibliografia:

- Harel, D.: 2001, `From Play-In Scenarios To Code: An Achievable Dream'. IEEE Computer 34(1), 5360,
- Harel, D.: 2004, `Comments made during presentation at "Formal Approaches to Complex Software Systems" panel session'. ISoLA-04 First International Conference on Leveraging Applications of Formal Methods.

SoftEng_GME
The Generic Modeling Environment (GME) is a model-integrated program synthesis tool for creating domain-specific models of large-scale systems. GME allows users to define new modeling languages using UML-based metamodels. GME was developed by the Institute for Software Integrated Systems at Vanderbilt University, http://www.isis.vanderbilt.edu/projects/gme/

SoftEng_xUML
Executable UML, http://en.wikipedia.org/wiki/Executable_UML

SoftEng_UML_Prolog
Analiza diagramów UML w Prologu, formalna analiza modelu w Prologu, translacja UML do Prologu i vice versa. http://www.systemes-critiques.org/krasovec.pdf http://home.mit.bme.hu/~majzik/publicat/ddecs2001.pdf http://home.mit.bme.hu/~majzik/publicat/safecomp01.pdf

SoftEng_HLLVM
The LLVM Compiler Infrastructure oraz The High Level Virtual Machine Fix Me!

SoftEng_Fujaba
http://www.fujaba.de/

SoftEng_Parrot
Wirtualna maszyna Parrot:

- zastosowania
- wydajność
- kompilacja/interpretacja

Fig. 2. AGH IA system page example
Source: own research and [6]

5. Critical Perspective

A review of the selected CMS revealed some common limitations. Even though the number of classes and implementations is quite large, few of the solutions are innovative. Most of them share some architectural patterns, and compete merely on technical level (e.g. Customizability, performance, etc.). Several types of limitations can be identified, namely:

– technical limitations,
– content management and portability problems, and most importantly,
– knowledge representation simplicity.

A common technical limitation is the dependence on selected run-time technology, namely RDBS. Most of PHP-based solutions use an simplified PHP API for selected

RDBMS. For performance reasons MySQL has been a popular solution for FLOSS CMS. However, until recently, it lacked some important RDBMS functions, such as transactions, which have always been present in PostgreSQL or Oracle. An obvious solution is to use some kind of database abstraction, such as ADOdb (adodb.sf.net) which supports number of RDBMS: MySQL, PostgreSQL, Interbase, Firebird, Informix, Oracle, MS SQL, Foxpro, Access, ADO, Sybase, FrontBase, DB2, SAP DB, SQLite, Netezza, LDAP, and generic ODBC, ODBTP, or some advanced modules available in PHP.

Another technical limitation is poor support for W3C standards and technologies. Number of solutions are still HTML-dependent, whereas they should be based on XML. XML along with CSS not only allows for advanced content encoding, structuralization and presentation, it is also foundation for other important standards. XML-based meta-data languages such as RDF provide a unified framework for meta-level content description, including semantic information.

When it comes the the actual management of the content the main problem is, that CMS systems provide limited content portability. As long as content is simply data, such as HTML, PDF documents, pictures and so-on CMS acts as a repository. When it comes to sharing information *about* the content, meta-data, or meta-knowledge, CMS systems do not provide appropriate facilities. This is mainly related to the lack of some common knowledge representation standards, while focusing on low-level encoding, and visual presentation of content.

From knowledge management point-of-view, the crucial problem with CMS systems is, that they escape the knowledge representation and management pitfalls by assuming some predefined CMS structure, e.g. a portal. In this sense these CMS are only a kind o modifiable portal *templates*. It is difficult to choose knowledge representation for current CMS solutions, because the knowledge they store can be considered implicit, or even hidden.

This criticism implicitly assumes that what CMS where designed to manage, and make available is somehow knowledge. That would put the process of deploying a CMS in the context *knowledge engineering*. In this field concepts such as knowledge *representation, management, or validation* are used. They refer to some common tasks in this field. However, considering how most of CMS are built, they cannot be considered Knowledge Management Systems. Turning current CMS into knowledge-oriented CMS should involve number of features.

6. Guidelines for Knowledge-based CMS

There seems to be seven generic functionalities/areas that should be supported by any “knowledge server”; these are [5]:

– knowledge representation and organization support; the system should provide appropriate structures and languages;
– knowledge processing and inference; the system should be capable of multi-paradigm reasoning and automated operations on knowledge;
– inference and operation control, user support, and operational decision making;

- knowledge acquisition support and extraction of knowledge from different sources;
- user interface – both for operational use and administration/design;
- truth-maintenance, verification and validation support;
- learning and optimization.

These top-level general features are often translated into numerous detailed features.

A critical area is an advanced explicit knowledge representation. In the field of knowledge engineering a number of representations, such as frames, trees, tables, concept networks, mind maps are available. Some new solutions, such as XTT, include even knowledge processing and logical analysis features [7].

Due to more and more complex nature of the encoded knowledge, as well as to the rapidly increasing size of knowledge bases, efficient techniques for supporting visual design and knowledge acquisition are becoming more popular. It is believed that more than 80% of information accepted by human comes through our eyes. Simultaneously, graphical knowledge representation is more intuitive and can be performed at different levels of abstraction. The XTT approach [5, 7] allows for hierarchical visualization of both knowledge (encoded in the form of tabular trees) and inference control (encoded in links of the tree/graph).

Since modern systems can accept knowledge coming from heterogeneous sources and specified in different formats, it is more and more important to assure appropriate capabilities for knowledge evaluation, verification and validation. Some formal techniques are developed for knowledge encoded in logical and algebraic (tabular, tree-like) form [5, 7].

The knowledge encoding should not be limited to structure encoding (XML), but also meta-data, and semantics, for example using RDF. There are number of valuable developments in the field of languages for formal ontologies (OWL) which could improve CMS. In case of knowledge specified with rules, the RuleML [11] language offers some limited capabilities for knowledge specification, mostly with respect to structural specification useful for exchange and analysis. Unfortunately, semantic operations are not supported. Moreover, complex inference mechanism are neither designed not implemented.

Knowledge Management [9] is a relatively new domain located at the intersection of knowledge engineering [5, 8], management in general, informations systems including expert systems [8], and intelligent web applications. An example of a system integrating various knowledge management techniques for enterprise applications is the PYTON system [12]. There are numerous efforts to meet the emerging requirements, however, in case of lack of consistent and uniform theoretical foundations efforts oriented towards building a knowledge server replacing, subsuming and covering database services, web applications and decision processes are far from being satisfactory.

An example of an ongoing effort to build a so-called “knowledge server” (it could be considered knowledge-based CMS) is the CyC Project (cyc.com). The Cyc knowledge base is a formalized representation of a vast quantity of fundamental human knowledge: facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life. The medium of representation is the formal language CycL. The KB consists of terms – which constitute the vocabulary of CycL – and assertions which relate those terms. These assertions include both simple ground assertions and rules. The Cyc KB is divided into many „microtheories", each of which is essentially a bundle of assertions that share a common set of assumptions; some microtheories are focused on a particular domain of knowledge, a particular level of detail, a particular interval in time, etc. The microtheory mechanism allows Cyc to independently maintain assertions which are *prima facie* contradictory, and enhances the performance of the Cyc system by focusing the inferencing process [4].

7. Concluding Remarks

The paper discusses certain issues concerning the so-called Content Management Systems (CMS) which can be regarded as a partial solution with respect to knowledge storing, retrieval and presentation. A critical perspective and state-of-the art of such systems is outlined. Contemporary tools and techniques applied in CMS are presented in brief and future problems to be solve are identified. Limitations of such systems are an inspiration to build knowledge management CMS in the form of a *Knowledge Server.*

Some most important functionalities of potential Knowledge Sever are identified. A short discussion of possible application of existing and experimental tools to form components of the system is provided. It is pointed out that new technologies, such as XML related ones are not sufficient to satisfy the requirements.

References

[1] The Opensource CMS Website, www.opensourcecms.com, Miro International Pty Ltd, 2005

[2] The CMS Matrix Website, www.cmsmatrix.org, Plain Black Corporation, 2005

[3] Ernst S., Pacewicz D., Klimek R.: *Nowoczesne systemy zarządzania treścią w bazach danych i witrynach internetowych*, 2005

[4] What is Cyc?, www.cyc.com/cyc/technology/whatiscyc_dir/whatsincyc, Cycorp, Inc, 2005

[5] Ligęza A.: *Logical Foundations for Rule-Based Systems*. Kraków, UST-AGH Press 2005

[6] Łaz K., Szpakiewicz R.: *Projekt i implementacja systemu obsługi prac dyplomowych – baza danych z dostępem internetowym*. Kraków, AGH 2004 (praca dyplomowa, promotor: A. Ligęza)

[7] Nalepa G.J., Ligęza A.: *A graphical tabular model for rule-based logic programming and verification.* Systems Science, vol. 31 no. 2, 2005

[8] Aitech Katowice. Sphinx 4.0. http://www.aitech.gliwice.pl/. Vide: System documentation

[9] Liebowitz J.: *The Handbook of Applied Expert Systems*. CRC Press, Boca Raton, 1998

[10] Liebowitz J.: *Knowledge Management. Learning from Knowledge Engineering*. CRC Press, 2001

[11] Boley H., Tabet S., Wagner G.: *Design Rationale of RuleML: A Markup Language for Semantic Web Rules.* In SWWS'01, Stanford, 2001

[12] PYTON. http://www.pyton.pl/