Evaluation and Comparison of Content Management Systems

Bachelor Thesis

University of Rostock, Department of Computer Science
and
Department of Business, Economic and Social Science

submitted by Wersin, Daniela
born Dec 26th, 1979

Academic Supervisors: Prof. Dr. Andreas Heuer
Dr. Ing. Holger Meyer
Mathias Bietz

Submission date: August 15th, 2001

Abstract

Appropriate information in time has developed to a main business resource and requires the same professional management as conventional means of production. However, the huge supply as well as the customary decentralized storage are aggravating the effective handling of information.

Content management approaches to unify the administration of relevant information. There are developing a vast number of software products which aim to apply this concept. However, a clear definition of content management does not exist and thus, the market offers a variety of solutions.

This bachelor thesis analyzes four content management systems (IBM Content Manager, Hyperwave Information Server, Zope, VIP' Content Manager) which represent different aspects of content management, such as Web content management, collaboration in Intranets or management of multimedia. As a basis, general comparison criterions are developed, which orientate to the content life cycle. Examples of application areas for each system conclude the comparison.

Zusammenfassung

Geeignete Informationen zum richtigen Zeitpunkt zu haben - das hat sich zu einer wichtigen Unternehmensressource entwickelt und erfordert das gleiche professionelle Management wie herkömmliche Produktionsfaktoren. Das riesige Angebot und die üblicherweise dezentrale Speicherung erschweren jedoch den effektiven Umgang mit Informationen.

Content Management versucht, die Verwaltung von relevanten Informationen zu vereinheitlichen. Es entstehen zur Zeit eine Vielzahl von Software Produkten mit dem Ziel, dieses Konzept anzuwenden. Es existiert jedoch keine klare Definition von Content Management, und der Markt bietet folglich eine Vielzahl von Lösungen an.

Diese Bachelor Arbeit untersucht vier Content Management Systeme (IBM Content Manager, Hyperwave Information Server, Zope, VIP' Content Manager), die jeweils verschiedene Aspekte des Content Managements umsetzen, wie z.B. Web Content Management, Zusammenarbeit in Intranets oder Multimedia-Verwaltung. Als Grundlage werden zunächst generelle Vergleichskriterien entwickelt, die sich am Lebenszyklus von Content orientieren. Einsatzbeispiele eines jeden Systems schließen den Vergleich ab.

Acknowledgements

I herewith thank everyone that provided me with assistance in writing this bachelor thesis.

Special thanks is for Mathias Mohr and Birger Bösel from PLANET ic GmbH Schwerin, who introduced the VIP' Content Manager to me.

Furthermore, I want to thank Prof. Dr. Andreas Heuer, Dr. Ing. Holger Meyer and Mathias Bietz for supervising the thesis. I show my special gratitude to Mathias Bietz who patiently helped me with all my questions.

Lastly I wish to thank my family and my boyfriend for their loving support.

Table of Contents

LIST OF ABBREVIATIONS

CMS	Content Management System
DMS	Document Management System
DTML	Document Template Markup Language
ERP	Enterprise Resource Planning
GUI	Graphical User Interface
HIS, HWIS	Hyperwave Information Server
ICE	Information and Content Exchange
IE	Microsoft Internet Explorer 4.x or later
LDAP	Lightweight Directory Access Protocol
NIS	Network Information System
ODMA	Open Document Management API
OLE	Object Linking and Embedding
VIP	Versatile Internet Platform
W3C	World Wide Web Consortium
WDDX	Web distributed Data Exchange
WebDAV	Web-based Distributed Authoring and Versioning
XML	Extensible Markup Language
Zope	Z Object Publishing Environment

Introduction

Popular terms like “information age” or “third industrial revolution” show the exceptional role of information nowadays. Appropriate information in time has developed to a main business resource and requires the same professional management as conventional means of production.
The supply of data, which is the raw material information is extracted from, has equally increased. The following facts emphasize this development:

- The number of books in libraries is doubling every 14 years. [26]
- The number of web Pages on larger sites is doubling every year. [8]
- The amount of knowledge which is available world wide is doubling every five years. [26]

An effective handling requires a precise provision and management of the total assets as a basis for a quick retrieval. A good example is the successful development of web portals that sort and search information on the Internet.

What is the status quo of handling these huge amounts of data?

Data that is available as non-digitalized form is managed successfully in various kinds of libraries and archives. However, this method means space-consuming storage and inconvenient retrieval.
Consequently, the share of digitalized data is increasing. Amongst the advantages are reduced storage space, independency of location and automatic cataloging and retrieval. On the other hand, new problems arise. Digital data exits in numerous formats and is subject to a fast development of technology. Additionally, there is another major difficulty: information which was just made handy through digitalization is often managed decentralized and inconsistently.

Consider the example of information storage in an arbitrary company: employees save relevant data either in their local file system or in project folders on LAN file servers. Structured data can be stored in databases, long-term storage uses

external media like magnetic tapes or optical media. In addition, there are still e-mail systems and floppy or Zip disks and CD-ROMs. If now an employee quickly wants to find relevant information for a waiting customer or a presentation about the companies projects should be published on the web site, the effort is huge. To say nothing of the workflow processes of data in projects or of controlled access to confidential information.

Consequently, there is an urgent need for a centralized management system. Content Management is aimed right at this non-trivial task.

Chapter 1

The Essence of Content Management

Before defining content management, it is worth taking a look at its development. Content management is a very young approach[1] - but covers already a large area of information technology. Marketing departments of software companies of any size soon elected it to become their hobbyhorse. The market of content management systems is developing fast and companies face a high pressure of competition. A universal definition of the term content management does not exits yet, companies rather understand it the way their system works. However, "content" and "content management" have become buzz-words and symbolize a breakthrough in information systems [27].

Indeed, the potential of content management is huge. After defining and delimiting all necessary terms in section 1.1, the benefits of content management will be explained in section 1.2. Finally, section 1.3 introduces the content life cycle as a preparation for the proceeding of chapter 2.

1.1 Terms and Definitions

1.1.1 Content

Content is the sum of all relevant single information [12].

According to this definition, content comprises both digital and non digital data. However, digital content is understandably the basis for professional content management software. By the way, in the German linguistic usage, the word "content" is simply borrowed, as the translation "Inhalt" would limit the broad meaning. Examples for content are text documents like bills or statements, web pages and page elements such as text, graphics, multimedia, but also applications and programming logic.

[1] according to [1], the first CMS was "Storyserver" of Vignette, 1997

1.1.2 Content Management

Content management is the process of systematic and structured provision, creation, preparation, administration, presentation, processing, publication and re-use of content [19].

This general definition emphasizes content management as a chain of actions that actually comprises more responsibility than the term “management” in content management suggest.

Another definition that is even more indistinct describes content management as a complex set of processes involving many different people from diverse functional areas, multiple process steps, and overwhelming amounts of information [8]. This statement spells the extent of the overall process out – in contrast to the following definition:

Content management is the controlled publication of Web sites [18].

Obviously, content management moves on a wide area of interpretations, although the last definition probably meant Web content management. This special form of content management will be explained in the next section. Later descriptions of key features and system components will finally clarify the comprehension of content management.

1.1.3 Content Management Systems

A content management system is a software to accomplish content management.

As the definition of content management is very broad, a universal system that supports the whole process is unlikely. Rather, most content management systems have specialized on certain parts of the process or offer customer specific solutions.

The biggest group amongst them are Web content management systems. They restrict the understanding of content to everything that should be published on the Internet. 2 of 4 systems analyzed later in the thesis fall into this group.

Managing large amounts of information is not a new problem - systems with similar intentions exist since years. Document management systems and groupware systems as two major representatives are shortly explained and delimited from content management systems below.

1.1.3.1 Content Management and Document Management

A document management system is a software (sometimes also additionally hardware) that captures, administrates and stores (sometimes even archives) documents. If the documents only exist in hardcopy, they are transformed to softcopy by scanning. A popular document management system is DOCSOpen from Hummingbirds/PCDocs.

Obviously, “content” and “document” make the difference. There are a lot of associations with the term “document”. Text files are often saved as .doc. In everyday life, documents are generally considered to be an important piece of paper. In fact, this understanding was also the initial one of document management. Even if in the meantime document management systems tried to enlarge the term document to files of any type, it will always represent something persistent. In contrast, content can be highly dynamic, e.g. a search result or a personalized Web page. Additionally, content management systems focus on publication of content, whereas document management systems emphasize the storage and access of documents.

For the remaining part of the thesis, the terms “document”, “content” and “object” will be used synonymically as far as the definitions permit it.

1.1.3.2 Content Management and Groupware Systems

A groupware system is a software that supports collaboration of project groups with unified clients for e.g. email, calendaring, group scheduling and to-do lists. A popular example is IBM Lotus Notes.

Groupware systems allow a sophisticated collaboration over Intranets. Similar to document management systems, they do not focus on publication. Additionally, they involve only parts of the available content. Thus, if required, they can be a useful supplemental application to content management systems.

1.2 Benefits of Content Management Systems

The diverse benefits that the use of content management systems will bestow upon commercial as well as institutional organizations are categorized into three groups: Because content management …

1. … provides an organization wide view of all content in one repository system,
    - this means common and consistent storage and access for all employees, and even for authorized users/customers who demand online access to their statements
    - the quality of content is increased. The content is up to date and error reduced, as the system scans e.g. for broken hyperlinks
    - the “time to delivery” is shortened
    - there is more security through a unified and fine grained access control
    - the access is be faster and storage space is saved
    - the unified technical fundament is open and scalable
    - the separate storage of metadata offers better search capabilities

2. … separates pure content chunks from their layout information,
    - the system increases re-usability of content for different target publications
    - easy updates can be achieved by exchanging layout or content

3. … integrates workflow features,
    - activities of team members across the organization can be coordinated
    - distributed and controlled authoring is possible
    - processes can be rationalized and costs be reduced

The successful application of content management depends on both, an appropriate content management system as well as professional and serious strategic planning. Before purchasing a special system, the organization needs to set up its specific catalogue of requirements in order to thoroughly check the available systems. The

acquirement is a major investment and can not easily be revoked. Consequently, the decision for a content management system effects the long-term efficiency of the resource information.

1.3 Content Life Cycle

The definition in subsection 1.1.2 described content management as a chain of actions. This is based on the idea of different stages of content - the content life cycle. Below, the stages of the process are further refined, as there are also intermediate steps which will be of interest.

The content life cycle comprises the following stages:

creation, acquisition, conversion, editing, review, approval/rejection, test, and publication.

However, content management system not only support the process of content publication but also the integrated storage. There are still other, parallel stages that can be determined:

cataloging, storage, access, maintenance, preservation, and disposal.

The connection of these intertwining process chains is depicted in figure 1:

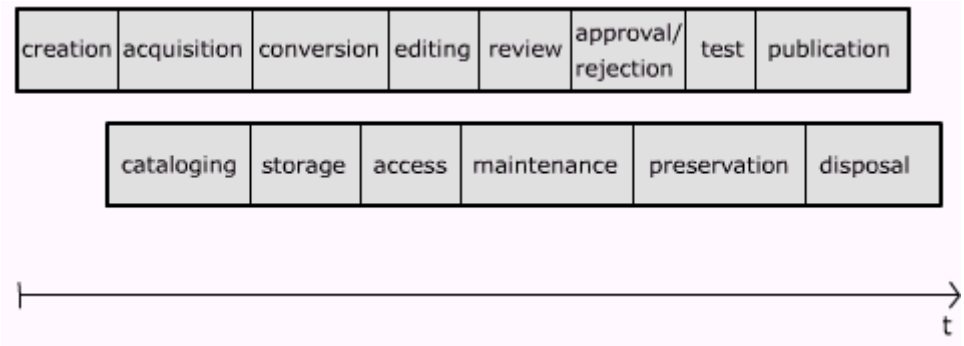

Fig. 1: Content life cycle stages

The content life cycle is a sequential process like any other life cycle. Nevertheless, it allows certain stages to be repetitive. The editing-review-approval/rejection combination, for example, can iterate until the auditing person is satisfied with the edited content. Likewise, accessing and maintaining the stored content are highly

repetitive stages. Even the disposal does not have to be singular, because undo possibilities offer a restoration of the recently deleted content.

The character of the stages can mostly be guessed from the semantics of their names. Nonetheless, they should shortly be described in the following.

Creation is the act of producing the content piece. If created externally (e.g. purchased news), the content needs the be acquired, thus undergoing the control of the content management system. Conversion is an optional step. It strips off unnecessary information or changes the markup language. Editing is one of the main stages. Here people from different departments assemble, re-arrange, and prepare the content for its publication. Graphics artists, application developers, Web page developers, or accountants are involved. After that, the content is reviewed and approved or rejected, which means evaluation of correctness and adequacy. This is also optional, but common practice and an essential part of workflow. The testing stage is facultative, but medium and large systems should offer this useful feature, where a real application environment is simulated and scripts or database access can be verified. Finally, content is disseminated to different target publications, such as CD-ROM, print media or Internet/Intranet.

Cataloging means indexing the content in order to simplify later retrieval and capture additional descriptive information about the content in metadata. Storage keeps the content on different storage media persistent and enables access to the content by searching it. Maintenance ensures the consistency and up-to-dateness, which was an important benefit of content management. In contrast, preservation guaranties future access and keeps the look and feel of the content through hardware and software migration. Finally, disposal is required if the content contains obsolete or superfluous information.

This chapter defined and explained relevant terms and thereby established the necessary basis for the comparison criterions of the following chapter.

Chapter 2

Comparison Criterions for Content Management Systems

The four systems that the thesis compares are as different as content management systems can be. Their scope and functionality aim at entirely different target groups. This matter of fact demands a general approach of comparison criterions. Furthermore, the criterions should not only be adapted to these four content management systems. Finally, as expressed in chapter one, the market situation is highly dynamic and competition will surely be a motor for future innovations and new trends. That's why, the approach explained below tries to provide general validity.

As a starting point, the diploma thesis of Oliver Zschau [1], which was submitted at the University of Applied Sciences, Mittweida, in 2000, had to be evaluated and, if required, refined and completed.
In his thesis, Zschau gave a broad overview of Web content management systems and analyzed the features these systems can have in detail. He set up a list of functionalities and grouped them into main categories. Generally, this is a useful approach, because the categories are mainly kept wide (e.g. asset management), though they sometimes include unusual relations, such as "search function" in the sub-category "authoring".
On the one hand, it is not stated explicitly that this feature list is to be used for a comparison of content management systems. However, at the end of the thesis, Zschau introduces eight Web content management systems and uses this feature list to describe the systems. Also, the Website www.contentmanager.de, which was established by Oliver Zschau, offers a dynamic comparison of content management systems and thereby uses a schema that clearly follows the mentioned feature list.

Obviously, this method does not meet the requirements of general validity set up above. It sticks to functionality, moreover to Web content management system

functionality. As an example, the repository system, which plays a major role also in Web content management systems, is treated marginally and can be found spread across the main categories of the feature list.

It was not possible to merge these two different approaches. Consequently, this thesis will present a comparison method from a different point of view.

2.1 Central Idea

Firstly, content management systems are broken down into components. Secondly, the processes within these components are analyzed.

The main components of a content management system are:
collection system, publication system, repository system, workflow system, and administration system. Figure 2 shows the relations between the components:

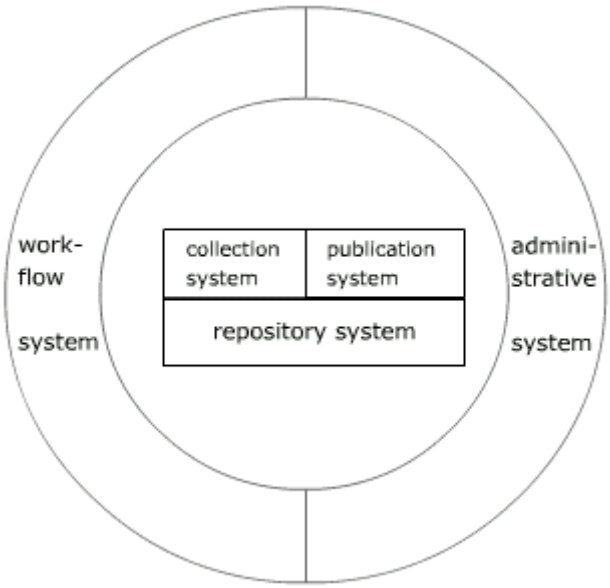

Fig. 2: Main components of a CMS

These five components do not only divide content management systems, they also clearly separate the long chain of processes involved in content management. In the following, the stages of the content life cycle are simply assigned to the main components:

- **collection system**
  creation, acquisition, conversion
- **publication system**
  editing, review, approval/rejection, test, publication
- **repository system**
  cataloging, storage, access, maintenance, preservation, disposal

The collection system is responsible for gathering the content and hands in the content chunks to the repository system. The publication system takes the content from the repository and prepares it for the requested target publication. Both, collection and publication system need access to the repository system.
Obviously, all stages of the content life cycle are already assigned and the remaining workflow and administration system did not get any stages. These two components are not only responsible for certain stages but rather accompany the whole content life cycle.

Content management systems can now be compared by asking:
Which content life cycles are supported and with which functionality?

According to the respective requirements of an organization which plans to purchase a content management system, emphasis can be laid on certain stages or components. As an example, a library will probably wish to get strong support in the repository system, whereas an insurance company would rather focus on the collection and workflow system.
In the run-up to an evaluation of content management systems, on organization can determine if all stages of the content life cycle need to be supported and where they attach special importance to.

Below, there is an overview of the main components, their stages of the content life cycle and the functionality that current can systems offer in these stages. A more detailed analysis of functionality within the life cycle stages can be found in the next chapter, where the four systems have been examined according to the described approach.
As it is the nature of any categorization, there are always “black sheeps” that do not fit into any of the categories. These “black sheeps” are gathered at the end of the overview in the category “general properties”.

2.2 The Universal System

This section offers an overview of the functionality of a universal content management system as it would look today. However, it is unlikely that a real system will offer all functions stated in the overview. The overview rather mirrors the sum of functions of the four analyzed systems. The following subsections each deal with a main component by listing possible functions and explaining interesting aspects.

2.2.1 Collection System

- **Creation**
  - create new object:
    - fix object type:
      - user defined type
      - automatically recognized by system
      - selective list of system defined object types
    - metadata input form

- **Acquisition**
  - import
    - from local workstation, Intranet or Internet
    - large amounts of data supported, e.g. through ftp server or import of multiple files, or import whole website with link structure
    - automated file type recognition
    - automated meta data recognition, derived e.g. from structure in emails, articles
    - Content Syndication, supported protocols: ICE
  - scan directly into CMS
    - scan-GUI
    - support of double and multiple pages
    - OCR/ICR

- **Conversion**
  - unnecessary information stripped from the content
  - markup language changed, e.g. conversion to XML

Content that is not directly created in the content management system can be acquired either by importing it or by scanning the hardcopy document. Beside the usual way of importing data, content can also be acquired from vendors, e.g. news from a press agency. This is called content syndication and is defined as a method of content exchange between authors, agencies and publishers via the Internet.
If content is scanned into the system, techniques of automatic recognition of standard text (OCR) or even of handwriting and forms (ICR) can save time and costs.
Concerning conversion of content, none of the four systems offered any functionality.

2.2.2 Publication System

- **editing**
  - internal editors
    - html editor, allows graphical and source code editing
    - text editor
    - syntax check
  - external editors
    - external application depends on object type
    - interfaces to existing applications
      - protocols OLE, ODMA, WebDAV
    - editing steps in external application are comprehended by CMS
    → steps can be rolled back and are kept in protocol
  - publication templates for each publication
    - chose before editing

- template editor
    - GUI

- style sheets, layouts, themes

- programming/scripting language
    - native or external
    - has access to complete repository
    - able to generate dynamic output

- create a navigational model (=site map)

- validation
    - hyperlinks, syntax

- **review**

  → compare workflow system

- **approval / rejection**

  → compare workflow system

- **test**
    - private working areas
    - preview for dynamic content

- **publication**
    - deploy (Web)
        - publishing server
            - development space in Intranet
            - is “staged” to either
        - staging server: holds static HTML
        - live server: offers dynamic and static pages
        - rollback of live server to earlier state
        - runtime dependent resolution

- check links
- display active links
- hide but remember dead links

- print, fax
  - internal print function
  - annotations of document can be printed separately
  - parts of object can be printed

- export
  - supported export types
  - convert to different format, e.g. PDF
  - annotations, metadata and protocol exported to extra file

- provide
  - collaboration across Intranet

Editing of content will always stay the primary domain of external applications. That's why, content management systems should offer appropriate interfaces to ensure that editors can continue to work within their habitual editors. Protocols like OLE or ODMA support a tight integration of both applications, so that for example the menu bar of the external application is integrated into the user interface of the content management system. WebDAV allows for distributed editing over networks.
Templates are an essential concept of content management, because they enable the editor to easily publish the same content with different layout. Equally important is the support of programming/scripting languages, because they can add dynamic and personalization to static Web pages. If the language is a native product of the manufacturer, a syntax validation feature should be offered.
The review and approval stages are explained in the workflow system, because they are often the main focus of workflow in content management systems.
During the test stage, private working areas enable editors to experiment in a simulated publication environment and only open the changes to others at the end of the test session.
The final publication can have different targets: content can be provided on the Intranet, printed, exported or deployed to a Web server. Actually, even output to

audio, Emails or WAP is imaginable. Web servers are often referred to as “live servers” if they offers dynamic content or “staging server” if only static HTML files are stored there. The respective German terms are “Produktivserver” for the two Web servers and ”Redaktionsserver” for the development space in the Intranet.

2.2.3 Repository System

- **cataloging**
    - indexing
        - different types of indexes offered
            - full-text
            - specific key words
    - metadata
        - high degree of customization:
            - user-defined and/or admin-defined metadata
            - powerful predefined key words, like expiration date
        - multiple-value attributes supported
        - all metadata can be searched for
        - stored separately from object
        - can be inherited if required

- **storage**
    - model for storage system
        - databases / file systems / hybrid = both
        - commercial dbs supported
            - interface JDBC/ODBC
        - native dbs (RDB, ORDB, OODB)
        - distributed storage
    - content
        - supported file types
        - handling of large files
            - divide them into components, e.g. a large text doc
            - no max. content size

- object oriented view on files

rights, metadata, hyperlinks, history protocol, status

- **access**
  - rights management (views)

    → compare administration system

  - search function
    - scope: metadata, full text, multimedia
    - execute “mass operations” on search result
    - store queries
    - invoke stored queries
      - in case of events
      - according to time schedule
    - offer search functionality even for Web users
    - special search features for multimedia content

  - search & replace function within files, across files

  - direct API for programmers

  - hyperlink system
    - overview of all hyperlinks from and to an object
    - check for dead links
    - hide, remember and restore dead links
    - GUI for editing hyperlinks
    - links only visible if user has access rights to target object

- **maintenance**
  - service programs
    - optimization and consistency checks of storage

- **preservation**
  - migration of content
    - time schedule

- event driven

- versioning of objects
  → compare workflow system

- scheduled server backup, restorable

- **disposal**
  - undo possibility

  - automatic disposal if expiration date reached

When the content is handed over to the repository, it is at first catalogued. In order to allow a faster search, the content is indexed for certain key words, such as author or date. This means that after pre-searching an object for specified key words and recording their locations for entry in an index file, the later search can be executed on the index file, which is must fast than searching through the whole object. Ideally, all metadata for an object is indexed and thus searchable. Additionally, full-text searching should be provided. Metadata values are partly predefined, such as size or author, and partly defined by the user, such as description. It must also be possible to define new metadata fields for individual purposes.
The actual storage of content can be in databases or file systems. Databases should be preferred, as they support the repository with sophisticated searching, sorting, linking and reporting capabilities. For bigger application scenarios, storage on multiple servers should be possible (distributed storage). Each piece of content is regarded as an object with certain behavior and properties, e.g. access rights, status, hyperlinks, history protocol and metadata. These properties are stored separately from the object.
Access to the repository system demands fine tuned rights mechanisms, which will be described in the administration system. Search functionality should be extensive, as the amount of content will be large. It is useful to store search queries and start them later in case of triggering events or according to a time schedule. On the search results, operations can be executed, such as “show all

rejected documents of last week and write reviewer's name to file". Additionally, a search & replace function should allow to replace content within one or multiple objects or content objects themselves. The content in the repository system should also be accessible through and Application Programming Interface to allow an alternative access for programmers. Furthermore, the content is linked within the repository. This is straightforward with HTML and XML objects, but also imaginable for every other object type.

Maintaining the repository's content can be helped by service programs, which look for stale content, optimize the storage or check the consistency, especially if distributed storage is being applied.

Preserving the content is a very important aspect, because it should be accessible tomorrow as well as in a couple of decades. This is actually the field of archives, and content management systems thus offer few functionality here. Content can be migrated, which means consistent transportation to probably another storage media.

The development of objects over time is kept with versioning. Each version contains the state of the object at a certain time. The principle of versioning is explained in the workflow system. Concerning the short-time preservation, back-ups of the content are a used.

Even content will not live forever. Disposal is a naturally function of the universal content management system, as well as the undo possibility.

2.2.4 Workflow System

- **conceptual model**
  - involved objects and paths
  - scope: all stages of content life cycle are supported

- **easy changeability**
  - GUI, API
  - freely configurable

- **tools**
  - visual interface, that shows the process graphically
  - task lists for users

- protected editing
    - check in/out:
        - automatically check out when edited
        - automatically check in after stored in system
    - lock/unlock objects
- annotations
    - great variety: sticky notes, comments, attach files
    - private, public, group
    - stored separately
- history protocol for content
    - date, time, user, detailed description of action
- notifications
    - automated
    - via email or pop-up window
- versioning of files
    - automatic or manual determination of version number
    - sorted list of all versions
    - every version is restorable

The workflow system controls the flow of activities within the organization. It coordinates users, applications and operations in order to achieve a defined aim within a certain time schedule. The necessary objects are provided by the workflow system [16].

In the universal content management system, workflow plays a major role. It supports all stages of the content life cycle and is freely configurable by the administrator. He can define the exact path of the object between the system users. There are a lot of useful tools which support the workflow:

After looking at his individual task list, an editor begins to work on a document. He checks the document out of the repository, so that no one can interfere with his editing steps. The others now only have read access to the state of the document when it was checked out. The editor can attach several annotations to the document, which have informational purpose for the following editors. If the editing stage is finished, the editor checks the documents in, thus handing over the changed object to the repository and sending it to the next station – either another

editor or optionally, a reviewer. The next user can be automatically informed by email or pop-up windows.
Another concept of workflow is the versioning of files. The term “versions” is generally understood as the different states of a document during the editing process over time, but not as different and fix states of a document for different publication purposes. These versions of the document are kept in a version history, each with a version number, date and time of creation and author. Every earlier version of a document can be restored and get the most current version.

2.2.5 Administration System

- **security**
    - user management
        - types of users:
            - anonymous guest on Web site
            - identified user on Web site
            - identified user within system
        - assign users to self- or predefined user groups and roles
        - GUI

    - rights management
        - fine tuning of rights (create, change, delete, add metadata)
        - double storage of access rights:
            - in each object
            - in each user group/role
            → possibility to locally restrict access users
        - objects inherit rights from parents upon creation

    - reliability
        - of authentication
            - SSL to encrypt passwords and all other content
        - of storage of passwords

    - integration of other user management systems
        - single-login through connection to LDAP, Windows NT, NIS

- **personalization**
    - customize interface for system users
        - general appearance
        - for different users or user groups
    - content classification
        - data types defined by system programmer,
        - instantiated by any authorized user

- **tools**
    - client tools (extra tool, browser)
    - administration tools (extra tool, browser)
    - reports:
        - system (performance, up time of server, slow pages)
        - Web users (popular areas, average visiting time, OLAP)
        - system users (logged-on users, current action, idle time)

An important task of the administration system is to manage the variety of users working with the system and determining which content they have access to. These users identify themselves during the login procedure. The universal content management system is able to integrate user records from e.g. LDAP or Novell user databases, so that the user does not need to re-login to the system. Web content management systems additionally have anonymous users (guests) and identified users (e.g. community member) that browse their Web site.
Users can be assigned to user groups or roles and thus receive defined access rights. Roles mirror the responsibility of the user, such as employee, designer, editor or system administrator. The rights should not only be “read-write-delete”, but contain fine grained actions, such as renaming, adding attributes or moving the object. The rights defined in the roles can be restricted (not enlarged!) by defining access rights to each object. As objects inherit access rights from their parents, it is possible to restrict access only in certain areas.

Personalization is often understood as offering an individual Web site for registered Web users. Because this task is mainly executed by so called “portal management systems”, it is not supported by the universal content management system. However, the universal content management system offers personalization of the system interface, so that for example users only see the functions they are allowed to carry out. Furthermore, system programmers can define custom object types. These object types can then be instantiated and used by authorized system users.
As an interface to the system serve tools – client tools as the classical user interface for everybody and administration tools made for the role/group of administrators. Both interfaces can be offered either with an extra tool or within the browser. The latter method is advantageous as it allows users to access the system from any workstation with Internet/Intranet connection and does not require multiple installations. However, the working speed heavily depends from the bandwidth.

2.2.6 General Properties

- **API**
  - open
  - standard language, e.g. C, Java

- **architecture**
  - modular architecture
  - system is open source

- **availability**
  - platform
  - operating system
  - Web server

- **language support**
  - system interface in different languages
    - for each user saved in user preferences
  - content in different languages
  - virtually all languages available, scalable

- bi-directional languages supported

- **extensibility**
  - add-ons from supplier/ to other products
  - eCommerce
  - application servers
  - XML
  - interfaces to programming languages or internet protocols

- **price, licenses**
  - best case: free

- **support**
  - documentation: extensive, understandable, examples
  - on/offline help
  - support of manufacturer
  - community, mailing lists

This last category contains general properties of content management systems. The API should, for example, be open to allow programmers to implement additional functionality or build connections to external programs. The architecture of the universal system is modular and flexible, the source code is public, so that everyone can self-fix problems and there is no dependency on the existence of the manufacturer. It is available for every environment and can be extended by other applications, such as shopping or ERP systems. The interface to XML will be dealt with in chapter 3.5.
Language support applies to the language of the system's interface as well as to content. The users can define their favorite interface language in their preferences and virtually every language is supported. Concerning the content, the system can manage different headlines and text elements for the same object in different languages. As the universal content management system represents the best possible case, it is for free, but nevertheless, an extensive support is offered.

This concludes the chapter about general criterions for a comparison of content management systems. The following chapter will use the approach in order to present four content management systems: IBM Content Manager, Hyperwave Information Server, Zope, and Gauss VIP Content Manager.

Chapter 3

Comparison of four Content Management Systems

This chapter will present the following content management systems:

- IBM Content Manager 7.1
- Hyperwave Information Server 5.1
- Zope 2.3.0
- Gauss VIP' Content Manager 5e

Thereby, the comparison follows the method introduced in the last chapter.
Generally seen, there are two ways to accomplish a comparison of elements.
1. The first way takes an element, analyzes it according to the comparison criterions, takes the second element, analyzes it … etc.
2. The second way sequentially examines the comparison criterions and analyzes for each criterion all elements to be compared.

The following comparison will proceed according to the first way. Each section below will examine one of the systems and present their strengths and weaknesses during the content life cycle. This overview corresponds to the current release and does not claim to be complete. Patching holes by self-implementing functionality or purchasing any manufacturer's add-ons is generally indicated or not considered.
Finally, the last section will give a summarized evaluation of each content management system and point out application examples.

3.1 IBM Content Manager

IBM Content Manager combines the former IBM products EDMSuite Image Plus, VisualInfo and DB2 Digital Library. The latest version is Content Manager 7.1, version 8 is coming soon.

The help function [20] within the client applications was a major source of information for the following overview.

3.1.1 General Properties

Architecture, Availability, API

The IBM Content Manager (IBM CM) is built on a triangular client/server architecture of client, library server and object server, as shown in figure 3:

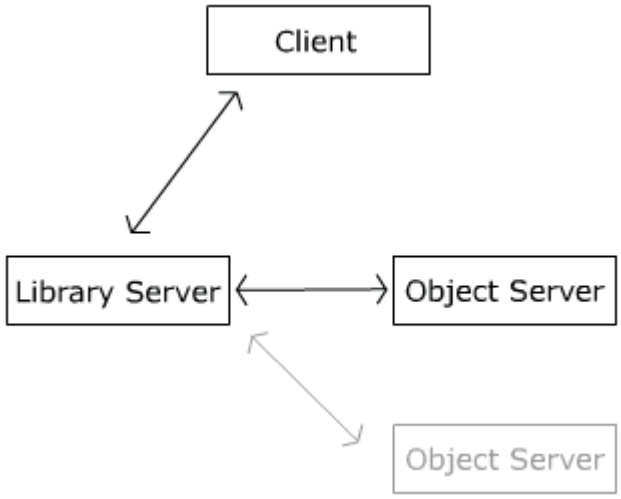

Fig. 3: Triangular architecture of IBM CM

Each of the components must occur one or more times.

The actual content is stored on the object server. However, the client sends its requests to the library server. There, the request is checked for access permission and forwarded to the object server. The content is sent back to the library server and finally to the client. The servers are available for Windows NT, Windows 2000, AIX and MVS/ESA. The client application is available for Windows platforms, or can be implemented with the C API.

Language Support

The IBM CM supports 22 languages, which are identified by a 3-character code. The language of the client application and of the content is simply defined by stating the desired language code. Administrators can define metadata (="index classes") in several languages for each object. The system user will then only see metadata in the language of his client application.

Especially interesting is the support for bi-directional languages, such as Arabic or Hebrew. If a user inputs data into a system that is defined to be bi-directional, the cursor in the input field is on the right side.

Extensibility

IBM offers several add-ons for its CM product.

The "IBM Enterprise Information Portal Client Kit for CM" offers a browser based interface for the client application, therefore opening it for all platforms. Additionally, Java Server Pages are supported.

Furthermore, the optional "Image Search Server" and "Text Search Server" allow the storage and retrieval of text documents and images. They will later be explained with the repository system.

The IBM CM can be enabled as a Windows Active Directory and a Windows NT service application.

The system was certified as "Tivoli Ready" and tightly integrates with Siebel Systems' Call Center Application Siebel 2000. Interfaces to SAP R/3 or Lotus Domino can be purchased.

Price, Licenses

IBM CM aims at large-scale enterprises with serious amounts of information to be managed. The price and license policy is therefore aligned to the financial potential of the customers.

3.1.2 Collection System

Acquisition

IBM CM does only support the acquisition of content. There is no major support for content creation within the client application, aside from annotations or stored search queries.

Content can be imported from the local workstation or from any LAN resource. The user must chose the file type from a list, which can include self defined file types. It is possible to import multiple objects at once. Furthermore, content can be scanned directly into the system. Duplex-scanning, i.e. double sided documents, is supported if hardware-supported. Very long documents can be split into several content chunks, though they remain contiguous and the document as a whole can be browsed through. OCR can be integrated with third-party products.

3.1.3 Publication System

Editing

There are no internal editors. Instead, protocols like ODMA and OLE are supported. With ODMA, the editor can save a document from a desktop application directly into the CM.

OLE interacts with the applications to which the file type is linked within the operating system. With the “OLE-Linking” option, the editor activates the object by double-clicking it. The external application is opened and the content is transferred. With “OLE-Embedding”, the editor can double-click an object in a compound document (like an Excel-table in a Word document) and edit the object within the IBM CM client. However, the menus, toolbars, palettes and other controls from the external application are merged into the client application of IBM. This method called “in-place activation” is especially advantageous, if the compound document consists of objects each created with a different external application.

Yet, the editing steps in the external application can not be comprehended by the CM. The result is treated as a black box and is protocoled as a single operation.

There is an internal viewer called for Windows called “VisualInfo Client”. The user can open a document, rotate it in case it was wrongly scanned, zoom into it and show the content in negative colors.

Content in the IBM CM is not prepared for any publication. That’s why, the editing functions of the universal content management systems do not exist. Again, annotations are the only way to edit content. This, however, is not intended to be for publication, but as a communication means within the workflow system.

Review, Approval/Rejection

compare workflow system

Publication

Content can be printed, exported and provided on the Intranet.

Concerning printing, the user can chose if the various annotations or the directory structure should be printed, as well.

When exporting content, the user can chose the target location, but not the export file type. The result is always a file *.~VI, where * represents a cryptic name. Thus, the user should keep the export location, as otherwise it will be hard for him to find it. Annotations can be exported, as well. However, only text notes are readable by humans in the exported file, the various graphic annotations result in a file with cryptic symbols.

3.1.4 Repository System

Cataloging

In IBM CM, content is linked with index classes. These classes can be imagined like predefined classes of metadata. After acquisition, the user must chose an index class. They consist of key fields with values of a common data type, such as long integer. The name of index classes, their key fields and data types are defined by the administrator. Index classes influence content storage, invocation, display and processing. There are no meaningful system defined index classes.

Storage

As described above, the CM is built on a triangular architecture and thus offers powerful, distributed storage. There can be multiple object servers and library servers.

An object server stores the actual digital objects. It contains at least one database in order to keep track of the exact position of each chunk of content. The databases must be either DB2 Universal Database or Oracle. For improvement of security and availability, content can be replicated from a primary object server to a secondary one. In case of a crash of the primary object server, the content can be viewed on the secondary server and restored if the primary object server is available again.

The object server supports the attachment of storage volumes such as DASD, Tivoli Storage Manager and others.
The library server manages a central database with information about the objects in the object server. This database must be DB2 UDB or Oracle. The server is responsible for cataloging, searching and secure access to content in the object server.
The separation of content and its catalogue information has several advantages. It is scalable, as new servers can be added easily. The library server protects the actual repository from direct client access. Thus, the client does not need to care for the content's location on the servers.

The repository supports all file types that are system-supported.
If large text documents have to be stored, they can be separated into smaller pieces, but still remain contiguous. This improves the performance, because the user can already look at the first pages of a document while the remaining parts are still transported. Versions of the CM older or equal 7.1, the variable "MAXPIECE" has been removed. Thus, objects can have any size. A utility program can divide huge objects to smaller ones.

Content is treated from an object-oriented point of view: It is indexable, searchable, can have a protocol of notes and other annotations, can move across workbaskets, is erasable, can be checked in /out and has a priority between 1 and 31,999.
The additional "VideoCharger" can store terabytes of video and audio stream data.

Access

The search interface can search in index classes and perform searching within full-text and multimedia files. The latter two engines, however, must be purchased separately.

IBM CM offers a simple search and an extended search function.
With the simple search, the user can choose index classes and define, which values should occur in their keyfields.
With the extended search, multiple index classes are allowed. Search queries support any combination of logical operators (NOT, AND, OR) and of the

placeholders “*” for {0..n} occurrences or “?” for (0,1) occurrences of a letter or number. Furthermore, queries can be saved for later use and parameters can be defined. Parameters make the query generic, because the required values have to be inserted by the user whenever the query is executed. Each query can be checked for syntax errors before execution. The extended search’s scope also comprises documents that are currently checked out or in a workflow.

The optional Text Search Server supports searching in unstructured text as well as in structured (e.g. XML). It can deal with various file types.

The optional Image Search Server is a search engine for digital images. It does not store the images, but contains different indexes of the images and executes the searching. The images are indexed according to four features, such as QBColorFeatureClass for the average color of all pixels.

There is no search & replace function and the documents can not be linked with each other.

Maintenance

IBM CM offers the following service programs: a “validation utility” checks inconsistency between the library servers and the object servers. Databases on the object and library server can be optimized, as well.

Preservation

With the migration function, content can be transferred from one storage class to another one. The user can determine which cycle (time between two migrations), heap size and storage classes should be used. For example, a university’s library could shift diploma theses to an optical storage medium after a certain number of years.
Versioning of objects is supported and will be explained in the next subsection.
A scheduled backup and restoration is possible through the replication of the object server to a secondary object server.

Disposal

There is no undo-function, so if content is deleted, it can not be restored.

3.1.5 Workflow System

Conceptual Model

Documents are stored in a hierarchical folder structure. Both, folders and documents have a priority ranging from 1 to 31,999. They are kept in workbaskets, a temporary place where they are being processed or waiting for it. The concept corresponds to a company's filing system for employees, departments and processes between them.

Workflow is understood as the routes of documents and folders between the workbaskets. This again corresponds to a company's internal flow of reports and forms. The routes are defined by an administrator. The operations within a workflow are: start, change, go on, finish, dispose. The user can always get information about the state of an object in the process.

There are "system chosen" and "user chosen" workbaskets. With "system chosen" workbaskets, the user can only see the current document to be worked on. If this is finished, the next document becomes visible. The order of visibility is determined according to the priority of the documents.

The scope of this workflow model is relatively small but powerful. It covers the editing, review and approval/rejection stage.

Easy Changeability

The route of workflow can be configured with the API and with an GUI offered in the administrator's client. The route can be arbitrary.

Tools

IBM CM supports the workflow with protected editing, annotations, versioning of files and history protocols.

If an employee wants to work on a document, he simply opens it – the document is automatically checked out. Checking out can also be triggered by an event or according to a time schedule. The employee can use a variety of annotation tools, such as pen, emphaziser, box, circle, line, arrow, stamp or post-it notes. Annotations can be erased and are only placed above the document onto a virtual glass plate. They don't change the document itself. External notes can be attached to the document, as well, the date and author are added to the employee's note

automatically. These notes are kept in a sequential protocol of notes, which can be printed and exported separately.

If during the workflow a document is edited, the different states can be kept is versions. If the editing is finished, the editor must decide if the last current version should be overwritten with the new document or if it should be kept as an old version. Old versions are only allowed to be read, not changed. Consequently, they only have informational purpose and can not be restored. This inconvenient property will surely be changed in the next release of IBM CM.

During a defined workflow route, all operation on a document are saved in a history protocol. It contains the operation, such as “check in”, date, time and user. The protocol is stored in a table of events in the library server. Again, operations can not be rolled back.

3.1.6 Administration System

Security

The management of users and rights is done by an administrator with the help of a Java based system management client or via C API.

Users can be assigned to freely configurable user groups, thus getting certain access rights and a standard library server. There are no anonymous users, like guests on a web site.

As usual, access rights are stored two times: in the user group and with each object in its access control list. The permissions in the user group should be maximal, as the object can only restrict the permission of users. The real permission is determined by the minimum of the user’s group privileges and the object’s access control list. The rights are very fine grained, such as creating folders or adding objects to workbaskets.

There are no indications about reliability.

On Windows systems, CM can cooperate with MS Active Directory.

Personalization

The user interface can be customized via the Application Programming Interface. Content can be customized by defining index classes.

Tools

There is client application available for Windows which was written with CM APIs. Administration tasks can be performed with the Java GUI or via a Web browser.
CM offers several reports about the system state and problem determination tools. For example, there is a validation utility that reports about inconsistence between the object and the library server.

3.1.7 Evaluation of IBM Content Manager

The IBM Content Manager focuses on storing and managing content in a robust and scalable way. Its main component is definitely the repository system with distributed storage and various migration, backup and restoration utilities. It handles all data formats and is able to keep content of any size. It offers a lot of interfaces for external storage management systems, such as Tivoli SM. The separation of library server and object server ensures integrity of data and special security.
Another strength of IBM CM is the workflow system. The automatic and arbitrary transfer of documents with the multiplicity of annotations ensures a fast and effective communication within the company.
However, the publication system is hardly supported. IBM CM does not know different target publications. Printing and exporting are only added for individual needs of users. Content is merely provided on the Intranet for collaboration purposes.

To sum up, the IBM CM offers a common, solid infrastructure for content. However, this should be better called “document management with integrated workflow”. The important property of content management systems, publication of content for different purposes, is not supported at all. Consequently, the term “Content Management System” is questionable.

3.2 Hyperwave Information Server

The following overview corresponds to Hyperwave Information Server 5.1. In the meantime, version 5.5 is available. The information is mainly taken from the User's Guide [13] and Administrator's Guide [23].

3.2.1 General Properties

Architecture, Availability, API

Necessary components are merely the Hyperwave Information Server (HIS) and a client workstation with a browser interface. They communicate via HTTP.

The server itself comprises three layers:

A The protocol conversion layer "WaveMaster" is able to transform HTTP into the native HG-CSP protocol.

B. The session layer creates instances for every client connection. The client application in the Web browser uses cookies for the session-oriented behavior. However, if cookies are disabled in the browser, session functionality can not be offered. (e.g. user identification or preferences)

C The database layer is responsible for storage of the content and management of links and attributes.

Hyperwave Information Server is available for all major platforms.

A separate Web server is not required, as an HTTP server is already included.

Application Programming Interfaces are provided for Java and all languages that support the Microsoft COM object, thus C++, VB, etc.

Language Support

HIS supports 11 European languages.

Each user can define his preferred interface language for the browser interface. If documents are available in different languages, the document for his preferred language will be displayed. Documents have an attribute "language", which determines the language of its content.

Extensibility

Hyperwave offers two major products to enhance to HIS:
"Hyperwave Information Portal" offers a single point of exchange via a Web browser for information that is not managed in a common repository, but spread in a decentralized network within the Intranet/Extranet.
"Hyperwave eLearing Suite" supports the continuous education of employees.

HIS offers interfaces to a lot of external products, e.g. SAP R/3, MS PowerPoint, Adobe FrameMaker, and to programming languages like Java, JavaScript or C++. As scripting languages, JavaScript, Perl, CSS, VBScript and PHP3 are supported.
The ability of HIS to cooperate with XML will be explained later.

Price, Licenses

Hyperwave does not publish its price policy.

Support

The guides delivered with the product are detailed and easy to follow, yet they describe the sequence of steps instead of explaining why and giving examples. The offline help is rudimentary, but all guides can be found online again. The company offer support for their customers, of course, but there is no community.

3.2.2 Collection System

Creation

When creating a new object in HIS, the user must chose from a detailed list: collection, discussion forum, text, HTML, remote object, CGI object or note.
A remote object is a reference to an object on another server, e.g. WWW or FTP server. CGI object can be executed on the server, that's why they can only be created by the administrator.
Collection is just another name for folder. There are amazingly many types of folders in HIS, each with a special behavior and attributes:

- Collections hold documents and other collections.
- Sequences additionally link their elements with a double-chained list and generate the navigation elements "next" and "previous" for every element.

The work to update links if the elements are shifted is taken over by the sequence.

- Language Clusters contain the same document in different languages. When opening a language cluster, the users only see the document that corresponds to their preferred language.
- Multi Clusters combine all elements they contain into one compound document. Users only see those elements in the document to which they have permissions.
- Alternative Clusters only return one element of all elements they contain. This element is determined according to the preferences of the visiting user. This is useful if, for example, the elements hold the same information, but in different qualities. A user with a low bandwidth would set “quality=low” in his preferences and receive the respective element, e.g. a lower resolution image.

The final dot on the i is the fact that elements may be a member of different collections. This is shown in Figure 4:

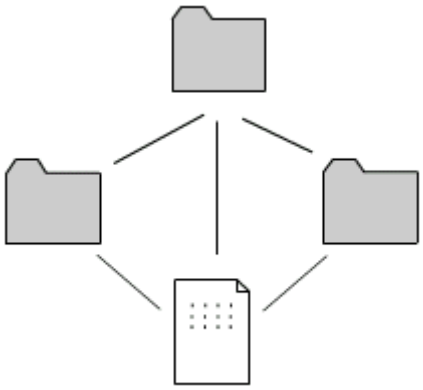

Fig. 4: Collection hierarchy in Hyperwave

This increases the re-usability of content. Collections and documents may occur several times in the hierarchy, though they are not physical copies, but exist only once. Of course, it is also possible to duplicate objects.

Metadata for an object is inserted into a form before it is created. This includes title, language, description and URL, others can be defined, as well.

Acquisition

Content can be imported into HIS, but scanning is not supported.
With the “Hyperwave Publishing Wizard”, it is possible to upload documents from applications or upload whole directory structures. File types are recognized automatically, and additional types can be defined. Certain metadata is determined by the system, like GOid, MIME-Type, Owner or TimeCreated. Objects inherit rights from the collections they were imported to.

3.2.3 Publication System

Editing

If the user wants to edit any content, he must at first enter the “authoring mode”. Then he can author, modify and administrate the objects.

There are a few internal editors: one for plain text and one for graphical HTML editing. These editors should probably be an aid for small, additional changes, because the main authoring still happens in external applications.
Beside these two editors, there is still a discussion forum builder included. This builder is a complete GUI for building forums and newsgroups within the Intranet. It can be a useful means of communication in projects group.

The interfaces that HIS offers for external authoring mainly support Windows platforms and all require local installation. The editor can use ODMA compliant applications to externally edit content. Rights and version control are integrated. Furthermore, he can use the standard MS Office tools and the Windows Explorer to access content on the HIS. The menu entries of the external application are then added by HIS functions, such as “save to HIS”. In the Windows Explorer, the HIS appears as an additional network file system, this method is called “Virtual Folders”. The objects to be edited are checked out on the HIS and a copy is sent to the local file system of the editor. A spooler program periodically checks the object and the Virtual Folders for changes and in case there are some, the corresponding objects on the HIS are updated.

There are publication templates, but they can only be used in the internal HTML editor. The templates can be implemented with the native PLACE language. Other supported languages are Java, JavaScript, and CGI.
Hyperlinks do not need to be checked for validity, because HIS manages links separately from the content. This will be explained in the next subsection.

Review, Approval/Rejection
compare workflow system

Test
There are no test functions offered.

Publish
Content on HIS that has come so far in its life cycle is considered to be already published. The hierarchy of documents is not supposed to be deployed to a Web server in order to become a Web site. Neither should it now be printed to books or used for a CD production. Of course, users can do so, but HIS does not support it. Content can be exported only with the “Virtual Folders”. Consequently, publication on HIS is equivalent to providing content on the Intranet.

3.2.4 Repository System

Cataloging
In HIS, metadata are stored in extra “information objects”. It can always be changed, added or deleted. They are referred to as “attributes” and can be sorted into four groups:
The administrator can define attributes, e.g. departmentID, and even standard system users can define attributes, e.g. MyDescription. This customized metadata is generally not indexed and thus not searchable, if not explicitly stated differently by the administrator.
The other two groups are system defined metadata. Here one can distinguish between read-only attributes and read-write attributes. As the name suggests, the value of read-only attributes can only be looked at and not changed by neither the system user nor by the administrator. Examples are “Size” and “TimeCreated”. On

the other hand, the values of read-write attributes can be changed by the user, e.g. “Rights” or “Title”.
Multiple-value attributes are not supported. Attributes or even parts of them can be inherited upon creation of an object in a collection type.

Storage

HIS contains an internal relational database called “wavestore”. Additionally, it is possible to connect to the Oracle8 Database Server (“waveoracle”).

The database layer (C) which was described in the “General Properties” subsection consists of three sub layers:

C1 The object server (“wavestore” or “waveoracle”): it creates, modifies and deletes objects and their relations, i.e. links and collections. Furthermore, it indexes them for searching and manages the user records.

C2 The full-text server, which can again be an native engine or the Verity Search Engine, stores indexes of all text documents

C3 The document cache server is the place where the actual content is stored. It also servers as a cache for documents that are stored on remote servers.

Distributed storage is supported in two ways. When using the Oracle database, the content can be distributed over several hard disks. Additionally, several Hyperwave information servers can shape a server pool, share the load and yet behave as one single server.

HIS initially knows 142 file types, this however can be adapted by the administrator. Content is treated as objects, as it has hyperlinks, information objects with attributes, annotations, access rights and can be searched for.

Access

There are two search engines, a native one and Verity Search Engine. Both engines can search for indexed metadata, and Verity offers sophisticated full-text searching with thesaurus, stemming, phonetic and proximity search. Multimedia searching is not supported.

Within the extended search function, the user can state the name of the XML tag that the search term should appear in and restrict the area to search in to a certain subset of the collection hierarchy. Also, he can filter the results by MIME-Type.
Search queries can be stored for later use and be executed according to a time schedule. Search results can even be sent via Email.

A search & replace function is not offered. One can replace objects with a menu function, but the searching for the objects to be replaced has to be done by the user himself.

HIS possesses a brilliant HTML hyperlink system. It is very flexible and entirely managed by the system. Hyperlinks are stored separately from the content they are linking. This means for example, that before storing an HTML document to the repository, the links are extracted and dynamically merged back into the document whenever it is later served to a user. The link's anchor can point to either a destination within the same document, to another document/collection or to a group of other objects (="multi-ended link"). The destination object can even be on another server in the server pool.
If the destination object is ever deleted, the link gets invalid and invisible, but is still kept! The link becomes active if the destination object is inserted again.
As hyperlinks are merged into the document only when it is served, links can became invisible for a user if he has no access permission for the destination object. There are "link maps" for each object which show all incoming and outgoing hyperlinks.

Maintenance

Content on a HIS can be updated with the content on another HIS with the mirror function. The wavestore database can be optimized through re-organization.

Preservation

HIS supports backup for wavestore and waveoracle. Backups can be scheduled. As they capture the state of the database at the start of the backup, all changes in the meantime are lost in case of a restoration. For details about versioning of objects compare the workflow system.

Disposal

On wavestore, content records are not removed from the database when content is deleted by the user. This only happens during the next database re-organization.

3.2.5 Workflow System

Conceptual Model

In order to support the editing, review and release of content, HIS offers the following functionality: protected editing, annotations, history protocols, notification and a version control for single objects and for whole directory hierarchies. The release flow defines the route of documents between editors and reviewers.

Easy Changeability

Release flows can easily be edited by an administrator through an API.

Tools

To ensure protected editing, documents can be checked out or locked. Checking out differs from locking concerning the additional features of version control. Each version controlled object has is a version history which shows former versions and the most recent one. Figure 5 shows a possible process:

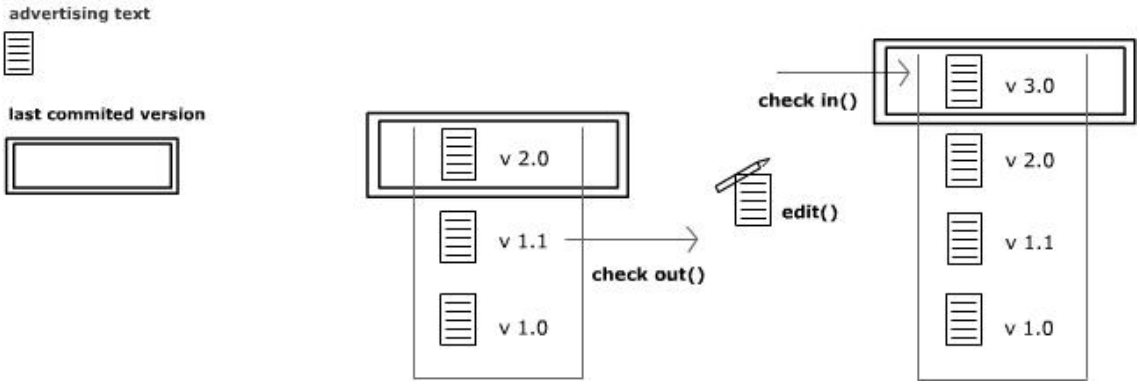

Fig. 5: Version Control in Hyperwave

Before editing, the last committed and thus the actual version of the advertising text is version 2.0. However, the editor wants to modify another version and checks out version 1.1. When finished, he assigns version number 3.0 to the changed

document and marks it as the last committed version. This is what users with read-only access will be able to see.
It is also possible to apply the version control to whole directory hierarchies. This is called “configuration” and saves not only the content, but also its structure and metadata.
Notifications can be configured as automatic Emails or pop-up windows. They are employed within newsgroups, stored queries or in a release flow, for example if a reviewer should be informed about a recently checked in document which he is supposed to review.

3.2.6 Administration System

Security

The types of users in a HIS are unidentified guests, identified users in viewing mode or authoring mode. For identified users, preferences can be set in cookies, such as interface language, MIME-Type or quality. Users and user groups are defined by an administrator.
The rights are only laid down within each object. In a rights wizard, permissions can easily be managed. However, the permissions only include read, write, and delete. Objects inherit the rights from their parents when they are created.

Encryption is needed as the browser is the standard client application and thus, confidential information as well as all the content is sent via HTTP. HIS supports SSL 3.0 for encryption.

HIS is able to integrate other user management systems, namely LDAP, Windows NT user directory and NIS.

Personalization

Based on HTML, JavaScript, CSS and the native language PLACE, the programmers can customize the user interface. Furthermore, the functions available in the user interface can vary depending on the user’s IP address. This is useful in Intranet/Extranet applications, where employees and partners should both access the content.

With “document classes”, data types with special characteristics can be implemented by a system programmer, e.g. an address book. The Object is later instantiated by a system user, e.g. MyAddressBook.

Tools

Both, client and administration tool must be accessed via a browser. Configuring the server is also possible in an extra configuration file.

HIS can generate reports about the server status, such as server name, up-time, or total amount of data retrieved in MB. Concerning the users, one can get a list of online users, their host name and their idle time.

3.2.7 Evaluation of Hyperwave Information Server

Hyperwave describes its flagship product as an “Intranet information server”. HIS offers a collaborative environment for companies with project organization. Teamwork is facilitated through the browser based interface, the ease of use, a tight integration of Windows applications and the newsgroup builder.
The system mainly supports the stages of the lifecycle that play a role in collaborative processes: creation and acquisition, editing, review and approval, cataloging and access of enterprise wide content. Its strengths are the flexible structure of documents which is mainly achieved by a sophisticated link management and various predefined collection types. The extended possibilities of metadata definition and search queries allow a fast retrieval of desired information.

The weak point of HIS 5.1 is the coarse access control. However, this has already been improved in HIS 5.5. Similarly to IBM Content Manager, external publication of content is not the major aim of the system. Again, this is (partly) improved in HIS 5.5, where a CD publisher program is added.
In contrast to IBM CM, the storage of content is not the center of interest. Responsibility is shifted to the Oracle database. The new product of Hyperwave, the Hyperwave Information Portal, could also be a reason for this approach, because it does not require a common repository any more. This convention of content management systems is understandably hard to realize in large companies with many business partners.

3.3 Zope

Now the functionality that Zope 2.3.0 offers during the content life cycle is described. The current stable version of Zope is 2.4.0. Basic references for the following overview are the “Zope Book” and own project experiences.
A great part of Zope’s features is gained through “add-ons”, which were developed by community members and can be downloaded separately. They are included in the overview, but indicated as such. The overview does not claim to have included all available add-ons. Furthermore, as new versions and add-ons are developed very fast, some missing functionality described here may already be patched tomorrow by a new add-on.

3.3.1 General Properties

Architecture, Availability, API

Zope is an Open Source Web application server of the Zope Corporation, formerly known as Digital Creations. Zope, which was mainly written in Python, comprises the following components:

- **ZServer** acts as a Web server and supports many Internet protocols, such as HTTP, FTP, FastCGI or XML-RPC.
- **ZCore** includes a Web Object Request Broker, a search engine and the security layer.
- **ZObjectDataBase** keeps all Zope objects and supports transactions, undo possibilities and private versions. Optionally, Zope objects can be stored in a relational database.
- **Management Interface**, which is the standard way to access Zope for users as well as administrators by simply using a Web browser.

Not only is Zope implemented in an object oriented way, its whole concept is based on object orientation.
Zope is available for Windows, Linux and Solaris. For other platforms, the source code must be downloaded and compiled. As the source code is open, everyone can extend it and build custom applications.

Language Support

Zope is available only in English, but there are add-ons for other languages. To achieve multi-language content, users can assign self-defined language properties to the content or, again, use add-ons.

Extensibility

There is a great variety of tools to extend Zope. Users can implement additional functionality on their own with so-called “Zope Products”. Thereby, they either use the management interface, write Python programs or combine both ways. Additionally, they can download add-ons (407 different products in August 2001), e.g. for eCommerce, credit card processing or web discussions.

Price, Licenses

Zope is free of any costs.

Support

Zope documentations are downloadable for free and offer an extensive pool of information for every requirement. In the management interface, a help function and parts of the API are included. There is a huge community all over the world which support users with “How-to”s, “add-on”s and mailing-lists. The manufacturer offers training, technical support and consulting for sale.

3.3.2 Collection System

Creation

To create content, the users chooses a product from a list, e.g. DTML method, external method, folder, image, user folder, ZSQL Method, etc.

If there are any self-defined Zope classes, they will also appear in the list.

Objects will be inserted into the folder they were created in. The structure of Zope objects is like a hierarchical tree of objects of any kind.

Concerning the metadata, few system defined attributes are automatically added, such as “Size” or “TimeModified”. For those with editable values, such as “Title”, a little form is generated. Every other metadata needs to be inserted later on by the user. →Compare cataloging

Acquisition

Zope has no scanning interface, but supports the import of content.

Users can choose between “file” or “image” to import from their local workstation or any source within the LAN. The MIME-Type is recognized automatically – users can even call an image “myPic.txt” – Zope does not care for type endings in names. It is possible to replace certain object at every time through importing another object. Large amounts of data can be uploaded through the integrated FTP server. Also, it is possible to exchange data on two different Zope installations with import/expert utilities. When using external scripts in Python or Perl, only the directory path of the code is required for execution.

3.3.3 Publication System

Editing

There is an internal editor for plain text. This is used for text or source code authoring and integrates a syntax check for the native scripting language DTML. However, it is inconvenient for larger documents.

There are few ways to edit content on the Zope management interface. Neither OLE nor ODMA are supported, Instead, Internet protocols like XML-RPC, WebDAV or HTTP are used. With HTTP, a copy of the document is sent to the user’s workstation. If the users is finished with editing, the content is sent back after a re-login. Calling and executing scripts is possible via XML-RPC. Remote editing via WebDAV requires a WebDAV client, which must be outfoxed with a extra HTTP-server for working correctly on the content of a document and not on its result.

Publication Templates

Templates must be implemented by the user in DTML or Python. Alternatively, there are several add-ons.

Programming Language

Zope offers its native, server side scripting language called DTML, additionally, Python or Perl scripts can be inserted. With DTML, the user inserts special tags into HTML documents in order to add dynamics, such as database query results. It is

advised that simple programming should be achieved through DTML, whereas more complicated logic should be programmed in Python or Perl.

Review, Approval/Rejection

→ compare workflow system

Test

Zope offers three ways to test the behavior of content.

Firstly, each object can be "view"ed at. With HTML, DMTL or scripts, the view function displays the result as if it would look when published and executed.

Secondly, ZSQL methods can always be tested on the current database connection without the need to call it in a document.

Thirdly, Zope offers "versions". Although the name reminds on versioning of objects during a workflow, this is not meant. Versions are private working areas of users. They open a session and work within a part of the object hierarchy privately. The others do not see his changes. If finished, the user closes the sessions and all content is automatically incorporated. This is advantageous, because it does not require any separation or publishing and live server, and there is no need to copy data or shut down any server.

Publication

Content on a Zope server is intended to be published on a on the Internet/Intranet. It does not need to be deployed, because it is already published if it exists in a folder. The URL to the object is just its path in the directory structure. Their visibility to others depends on the access permissions stated in the object.

There is no support for other publication media.

Objects can be exported to the local workstation or to the Zope machine. However, the fix file extension ".zexp" does not improve the usability. Objects can also be exported to XML, meaning that the content is saved within a huge <CDATA> tag and the attributes into separate tags. Until now, no other application that Zope can understand this structure.

3.3.4 Repository System

Cataloging

Users can set attributes with the help of “properties”. Properties have a name and a name which must belong to one of the predefined data types, such as boolean, float, data or several string types. These user defined attributes are not indexed and thus not searchable.
Zope sets certain read-only attributes, such as “Size” and offers read-write attributes, such as “ID” or “Title”. They are partly indexed.
Multiple values and inheritance of attributes are not supported.

Additionally, Zope offers the product “ZCatalog” – a built in search engine for categorizing and searching Zope Objects and even external data. Content inside a ZCatalog can be indexed for full-text, keywords or fields. This enables searching for attributes’ values or range of values. As ZCatalogues contain anything, the search result may include any kinds of objects, e.g. Email messages.

Storage

Zope supports a hybrid storage model, so the central Zope Object Database can use both, a file system or a relational database as its backing store. There is a built in relational database called “Gadfly”, for other relational databases an adapter has to be downloaded. Each Web request is treated as a single transaction by the ZODB, so that in case of errors, any changes made during the request can be rolled back. The ZServer logs in to the database only once after the start and uses this fast connection for all queries.
With ZopeSQL, SQL queries can easily be integrated, e.g. into HTML pages. There is no need to worry for database specifics, as this is done by the respective adapter. The results of a query are objects and can be treated as any other Zope Objects.

With the add-on “ZEO”, which means Zope Enterprise Objects, the Zope system can be modified to become distributed. This allows load balancing and distributed storage on multiple servers, machines and networks.

There is no special support for large objects.

One of the main characteristics of Zope is object orientation. Content has properties and offers operations that others can use. The focus is not on HTML, but on the components it is made of. An important concept is “acquisition”, which is depicted in figure 6:

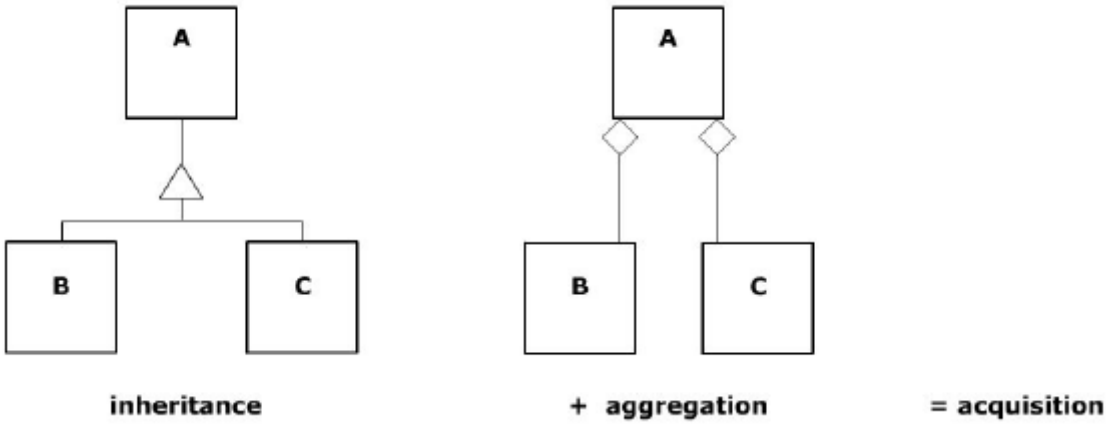

Fig. 6: Acquisition in Zope

Acquisition merges the conceptual concepts of inheritance and aggregation. Objects are placed inside of other objects (folders), and are thus able to reuse their parents’ attributes, content and functionality. To give an example, foot elements on Web pages are often the same. This means in Zope, that they are stored only once on the top-level. Each object, that is deeper in the folder hierarchy can acquire the element for insertion. The “client” object does not need to know the exact location of the wanted element. It only inserts the DTML tag <dtml-var myFooter> into its source code. If requested, Zope will search for it at first in the client object itself, then in the HTTP request and finally in all parent folders up to the root.
Of course, it is possible to override the default elements in special areas of the hierarchy. This is a very useful feature, because it centralizes common resources and avoids redundant data. In case of changes, e.g. the footer should be changed, the object only needs to be changed in one location.
Acquisition is also applied with rights.

Access

There are three ways to search for content on a Zope server.
Firstly, if the data is within the directory structure of Zope, the internal search function can search for specific IDs, object types and contained text.

Secondly, if the data is stored in the database, ZSQL queries can easily be implemented, stored and re-used.
Thirdly, data in a ZCatalog or outside in relational databases, file systems or on remote servers can be indexed and thus search through.
Zope even offers automatic forms for the query input and the results for the last two methods.
All methods can search in metadata as well as in full-text, however, features of multimedia objects are not supported.

Search & replace functionality requires an add-on.
Zope supports the standard HTML hyperlink system. All other special features of the hyperlink system of the universal content management system are not offered.

Maintenance
There are service programs for the ZODB, such as database packing and cache management.

Preservation
Zope offers a modified concept for versioning of files: Each editable object has a history, where every editing step is remembered and can be restored. The long list contains the date, author. It is possible to let Zope generate a comparison table of one older version and the current one.
With an add-on, the Zope server can make scheduled backups.

Disposal
Similarly to the history list of each object, the undo functionality remembers all single steps, which can be made undone. This, of course, also applies to disposing content.

3.3.5 Workflow System

Apart from the history protocol for each object, there is no workflow functionality implemented in Zope. However, there is an additional Zope product for this purpose: the open source workflow toolkit “wftk”. This is a library implemented in

C, which is suited for simple workflow circumstances. The deeper study should be neglected, as add-ons are generally only mentioned.

3.3.6 Administration System

Security

Each folder in the Zope directory structure contains a folder “acl_users”, where the access permissions for this folder are laid down.

Users are assigned to roles, like designer, author, or admin. They must at least have one role. Web site visitors are generally gathered in the role “Anonymous”. Roles can be designed freely by an administrator. In each user folder, permissions are assigned to the roles known there, e.g. “manager may delete objects” or “designer may undo changes”. Web site visitors usually have “View” permissions. These fine grained permissions are valid for the current folder and for all folders under it (=acquisition), as long as it is not stated differently (=overriding). Consequently, access permissions are only set with the objects.

Reliability through SSL encryption can by achieved with an add-on, just as the “ZLDAP-adapter” or the “NISUserFolder”.

Personalization

It is possible to customize the management interface, and content classification is supported, as already described in the subsection about creation.

Tools

All Zope users, including the system administrator, use the management interface in their browsers to access Zope.

The system offers reports about the system, e.g. with “profiling”, the speed or requests can be increased. With “snap shooting”, it is possible to determine which operations cause memory leaks.

3.3.7 Evaluation of Zope

Zope is an application server that offers simple content management. Its main advantage is the open source license. This means no costs, no dependence of a manufacturer, a large community that steadily improves the system and the ability

to highly customize it. Zope manages content that is supposed to be published on the Internet, it does not demand to integrate all available content in an organization in contrast to the universal content management system. Its focus is on the collection and publication system.
Dynamic Web pages can easily be built with the convenient integration of scripting languages. There are no different servers for development and publication required, although this was advantageous for larger applications, as not everyone can start his own version-area.
The management of metadata is flexible but decentralized. Self-defined properties of content are not searchable, only through the loop way of creating a ZCatalog. Consequently, the internal search function is rather simple, customized queries need to be self-defined within a ZCatalog.
Zope offers a basis for project development, however, even with the additional workflow kit, it is not suited for greater process chains.
A plus of Zope is the ability to communicate via open standards interfaces, to connect to many databases and file systems and to seamlessly integrate SQL. Furthermore, flexibility of the object structure is achieved through acquisition and object orientation, which will develop to the full during usage. Functionality needs to be implemented only once but can be used multiple times in different ways.
The user management is also flexible and powerful, but it can become confusing for large hierarchies with an inopportune permissions' structure.

Zope highly depends on what you make out of it. It gives major responsibility to the users by offering a system as the basis for further, own developments. Thus, especially professional applications will require programming experience and a deeper study of the system.

3.4 VIP' Content Manager

The VIP' Content Manager is part of the Gauss Versatile Internet Platform (VIP), currently in version 5e. The VIP includes components for content, platform, workflow and document management. In the following, mainly the content manager will be described, with notes on the functionality of the other components.

3.4.1 General Properties

Architecture, Availability, API

The VIP Content Manager (VIP CM) integrates the following components, as shown in figure 7:

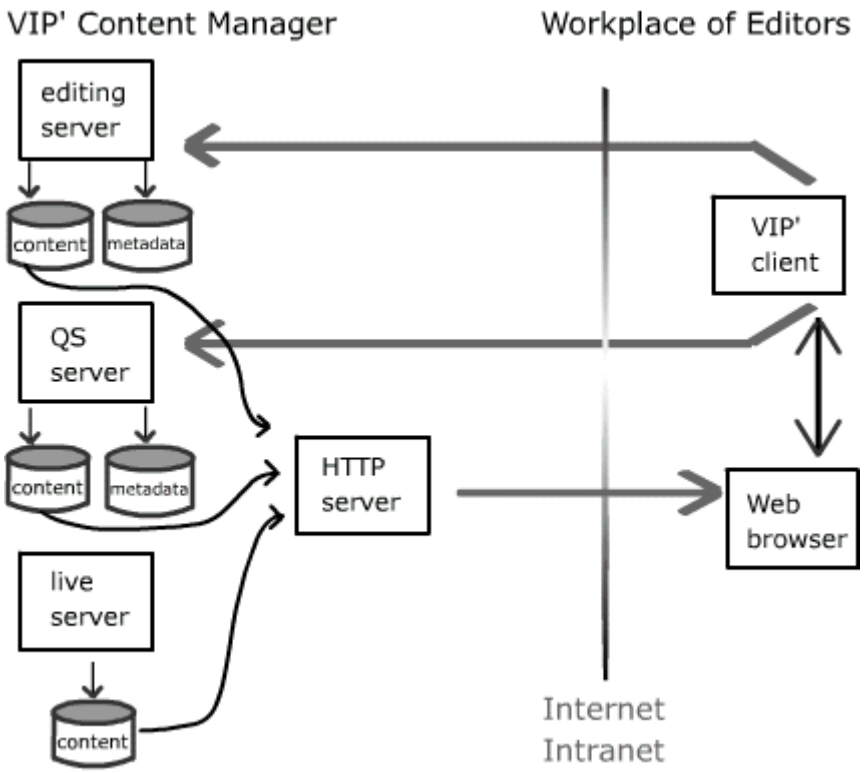

Fig. 7: The architecture of VIP CM

There are 3 main servers, the editing server, the quality control server (QS server) and the live server. Users of VIP CM edit content on the editing server. Thereby, they use the VIP client, which cooperates with a Web browser. If editing is finished, the content is released to the QS server. Here, authorized users like the chief editor

check the content and either reject it and send it back to the editing server or approve and release it to the live server.
The client/server architecture is oriented to the 3-step workflow model of VIP.
VIP CM is implemented in JSP and Java Beans. It offers three APIs. One Java API is used for modules ("agents"), which are activated if case of special states or event, e.g. a user exit. Furthermore, there are an XML-Gateway and a Java Beans API.

VIP CM is available for all major platforms and operating systems.

Language Support

The language of the interface can be set in the user's preferences, there are several European languages available. Content objects have a language attribute and can be displayed according to the user's preferences.

Extensibility

VIP offers a bunch of other products to cooperate with VIP CM:

- VIP Workflow Manager "VIP StaffView": GUI for complex business processes
- VIP Document Management "SpyVision Suite": document capturing, indexing, and archiving.
- VIP Portal Manager: personalization of Web portal

As a direct supplement for the Content Manager, VIP offers a "VIP Content Miner", which is a multilingual and semantic search engine, or the "VIP Intelligent Templates & Forms", which allows page design without any source code editing only with forms.
VIP CM can communicate with Java, Corba and XML. It has interfaces to important application servers and eCommerce applications from e.g. Intershop, BEA or IBM. The XML Gateway will be explained in the next section.

Price, licenses

The price is approximately 25,000€ and depends on the number of concurrent users and CPUs. [2]

3.4.2 Collection System

Creation

When creating an object, the user can choose from a large list of predefined file types and can input metadata in a special form.

Acquisition

VIP CM can import any file types. Additionally, I offers a “Multi Import” of whole Web sites including their link structure.
Scanning and OCR/ICR is only possible in the extra document management system “SpyImage”.
Content Syndication can be achieved through the XML Gateway.

3.4.3 Publication System

Editing

There is a simple internal HTML editor for graphical authoring.
To edit content in external application, they can set their favorite application for every file type. If they want to edit content, they check it out, a copy of it is sent to their local file system where any changes are saved. If finished, the object is checked in, the object copy is sent to the VIP editing server, where the old object is overridden and the object is automatically duplicated to the QS server.
The steps in the external application can not be understood and be rolled back separately by VIP CM.

VIP CM uses an advanced system of templates, which is called “cascading templates”. This is a hierarchy of single templates that can be combined in an arbitrary way. Figure 8 shows a possible combination:

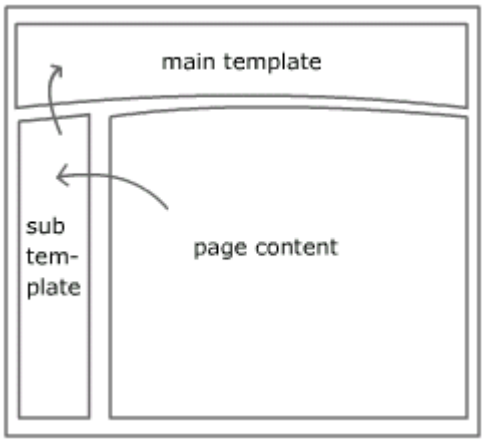

Figure 8: Cascading templates

The main template will probably stay the same throughout a certain area of the site. One level deeper, the sub template is assigned according to the chosen category. This sub template points to the main template. Finally, the page content is inserted and points to the sub template. In a Web page, only references to the respective template are needed.
Because of the easy handling of templates, an external HTML application is sufficient for editing them.
Content can be enhanced with dynamics through the usage of scripting languages, such as ASP, JSP, PHP, etc.

Internal hyperlinks are managed separately. Before a destination object of a links can be deleted, the user is informed with a pop-up window.

Review, Approval/Rejection

→ compare workflow system

Publication

Content that is approved will automatically be released to the live server. However, the live server only contains a static structure. Only with the additional VIP Portal Manager, dynamic and personalized Web pages can be applied.
Target publication of VIP content is the Internet/Intranet.

3.4.4 Repository System

Cataloging

Each object has multiple attributes, which can be inserted during the creation and modified at any time. There are a number of useful system defined attributes, such as title, headline, expiration date, remarks, and even the email addresses of the people involved in editing, reviewing and approving. These attributes can be searched for. The user can also define attributes on their own.
For HTML objects, certain attributes are automatically inserted into the meta tags of the HTML head.

Storage

On VIP CM, the content is stored in a special file system. Actually, the whole structure exists three times on each of the three servers. In the next version of CM, these physical copies should be replaced by instances of the data.
Metadata can optionally be stored in extra files in the file system, in an internal database or via an adapter in a lot of external databases, such as Oracle, DB2, Sybase and others.

Distributed storage is supported with “multi deployment”. This means that several sites can be deployed to multiple live servers.

Large files are not treated in a special way. Content is interpreted as objects with rights, metadata, hyperlinks, protocol of actions and a status.

Access

VIP CM integrates a powerful search called “Object Filter”. It allows fast full-text and metadata search . Mass operations can be executed on the search result or parts of it, e.g. “show all documents of editor X and release them at once”. These filter queries can be stored and re-used later, e.g. “check for rejected documents every day”. Images are stored in separate folders and have extra attributes, which can be searched for, as well.

Even more sophisticated searching and a search function for Web users are offered in the “VIP Content Miner”.

VIP CM applies a special hyperlink system, which administrates links separately. For each object, on overview of all references to and from the objects is possible, where they can also be edited. Dead links are not reminded, but the missing destination object automatically replaces by a standard error page. Additionally, editors are warned with a pop-up window before they can delete a link. A link consistency check for a single object is offered, as well.

Preservation

Short time preservation like back ups are supported, in contrast to migration or other archival methods. There are interfaces for archive systems. Changes of objects over time is kept in versions.

3.4.5 Workflow System

Putting documents into a workflow means in VIP CM to send it to a quality control after editing, where it is rejected or released to the live server. It does not mean the sequential editing between several editors. This relates to the intention to edit content in order to publish it to the Web.

The default route includes three steps, this can be enlarged with the workflow product “VIP StaffView”.

VIP CM supports various notifications, like Email and SMS, protected editing with check out/in functionality, a history protocol for all objects and versioning of files. Annotations, an important means of communication in a general workflow procedure, is only supported with a component of the additional DMS “SpyVision”.

3.4.6 Administration System

Security

User management and access rights are managed as usual. Users are assigned to user groups and roles, thus getting access rights.

Additionally, access rights can be set for individual objects. Rights are very fine grained.

The integration of other user management system is only possible for LDAP. A firewall can be installed in front of the live server, which is called “HTTP-Tunneling”, because the data packets from the client are wrapped as HTTP packets.

Personalization

The client interface can be customized, content, however, only with API programming.

Tools

VIP client and administration tools are browser based. Users can choose between either the browser with some additional JavaScript windows or a JavaGUI with connection to the browser.

There are reports about the users currently logged in and their activities as well as several system reports.

3.4.7 Evaluation of VIP’ Content Manager

The VIP CM is a powerful Web Content Management System. It attaches great importance to the publication system – even the server architecture is adapted to the process of editing, review and publication. The flexible usage of templates in VIP CM corresponds to the intention of content management: to easily present the same content in various ways. However, VIP prepares content only for the publication targets Internet/Intranet. The integration of Java and XML turns it into an open system.

The repository’s file system that stores the same content three times within the system is surely a weak point. This is planned to be improved, soon. The management of attributes and the fast search filter allow a high degree of customization and control over the content.

The workflow system is adapted to the Web publication process, but generally seen in need of enlargement. However, this and other small weak spots can be evened up by “simply” purchasing another component of VIP.

In fact, VIP has split content management into a maximum of single products and offers with the VIP CM pure web content management.

This sales strategy might be advantageous for customers who only need the functionality of one component. However, for an effective and far-reaching application of content management, customers like large organizations will probably not get around to buy at least the Content Manager, Portal Manager and “SpyVivsion” as a document management system.

If the VIP components for document management, workflow management and content management were combined together with the ability to publish to any target media, the result would probably resemble the universal content management system.

3.5 XML in Content Management Systems

In the following, some aspects of XML in content management should be explained and shown, how much support the four system offer in this respect.

XML is a family of standards that was developed by the W3C. XML allows the creation of common information formats.
XML and content management systems share a basic thought: Both distinguish between structure, content and layout. Generally seen, the use of XML in content management systems can bring the following advantages:

- PUBLISH:
  When content is separated from its representation, it can be merged with different presentations for different publications. This is exactly what content management systems support. As an example, imagine different output formats like plaintext, pdf, HTML or WML.
- SEARCH
  The semantics that are contained in the XML tags helps maintaining and retrieving content more conveniently. For example, headlines of a chapter in a text document can be recognized by the tag <chapter_head> and not through a bigger font size.
- EXCHANGE
  The support of XML in application programs is growing, so XML can become the standard format for data exchange, without the need to worry about specifics of the communicating applications. Packets can even be verified for right composition. All this is especially important for content management systems, because they follow a far-reaching approach and consequently need a lot of interfaces.
- FUTURE
  XML is computer and human readable. This is worth a lot in a couple of years, as applications are evolving quickly and are not endlessly downwards compatible. Content management systems should be interested in that, because they keep a lot and important information, which uses wish to access in the future, as well.

Who do the four content management systems meet these high demands?

IBM Content Manager

IBM CM can store XML documents. The optional full-text search engine is able to search for a certain part in that document, it the dtd was introduces to it beforehand.

Hyperwave Information Server

As Hyperwave is no authoring tool, it relies on external applications and browsers to author and visualize XML correctly.

XML and DTDs can be recognized, and search can be executed within special tags only. The Hyperwave Exchange Format “HWX” is used for backups and restoration, archiving, replication and configuration management. It is an open, XML based file format. Thus, HWX files can be created and processed with standard XML tools.

Zope

Zope has founded the Zope XML project in order to discuss and plan the role of XML in Zope and vice versa.

Remote calling of procedures is possible by using the XML-RPC protocol and thus enables all platforms and operating systems to communicate with each other, because the basic format is XML.

XML documents can be dynamically created with DMTL’s batch processing, which can produce a repeating tag structure.

For storing and retrieving XML files, you can use add-ons. Exporting documents to XML is possible, but no one will understand the format but Zope.

In version 2.4.0, the XML core and DOM are supported. This means, for example, that the hierarchy structure of Zope objects can be manipulated with DOM commands.

VIP’ Content Manager

VIP offers the XML gateway as the central interface for communication via XML, so programmers can use every language as long as it is able to evaluate and generate XML.

Importing and exporting content in/to XML is possible if it is conform to a native dtd. With the VIP product “Intelligent Templates and Forms”, HTML editing is achieved by using forms which are XML files, described by a special dtd.

XML content can be converted to HTML or WML via an XSLT interface.

It can be seen that the integration of XML in current CMS plays a growing role, though it is still fragmentary. However, XML will surely develop to a major issue with content management in the medium-term future.

3.6 Summarized Comparison

This chapter gave an overview of four content management systems and listed the functionality that they offer in the different stages of the content life cycle. It is now obvious that they represent very different aspects of content management and aim at different “deployment”. There is no system of which functionality is totally included in another one, each system has strengths in certain fields. A rough but often applicable starting point for a decision between the four content management systems could be the purchase price.

Typical application areas for each system should now be described.

IBM Content Manager

IBM Content Manager can bring value to very large business organizations with a lot of departments which are keeping the same content multiple times and are drowning in masses of paperwork. IBM Content Manager is able to capture all types of content, from small invoices to large multimedia files and store them safely across different object servers. Thus, it gives employees a central access point. They can easily search content with predefined metadata and quickly serve customers’ information requests. Another application is to grant customers online access to their electronic statements. IBM CM behaves close to document management systems with its main focus on content capture, storage, access and archiving. Functionality can be enhanced with other IBM products, such as “CM onDemand” for Enterprise Report Management or “MQSeriesWorkflow” for sophisticated workflow.

Hyperwave Information Server

A typical application example for HIS are large companies with a project organization, probably with branches located in different countries. They can expect to gain versatile collaboration support. HIS is easy to use and integrates seamlessly to Microsoft desktop applications. It offers important functionality “out-of-the-box”.

Zope

Just opposite to the first two solutions, Zope is rather suited for smaller, innovative organizations, probably non commercial ones. The system must be adjusted to specific needs by the “customers” themselves and problems are mainly fixed by

them, as well. However, this guarantees flexibility, self-responsibility and independence of a manufacturer. Zope will evolve quickly because of the huge community, it is always open to standards and, last but not least, Zope is free.

VIP' Content Manager

The VIP content management system is actually the closest to Hyperwave. It offers powerful solutions for large collaboration scenarios. Additionally, the content should also be published to the Internet, thus VIP CM offers support for authoring and publication of content. It integrates well with other applications, for e.g. eCommerce or Customer Relationship Management. However, VIP CM is only one component of the "Enterprise Content Management" suite, which tries to combine sophisticated portal, document, workflow and content management - resulting in a real powerful system.

Summary

This thesis dealt with the comparison of four content management systems:

- IBM Content Manager 7.1,
- Hyperwave Information Server 5.1,
- Zope 2.3.0 and
- VIP' Content Manager 5e.

As a basis, in chapter 1 all relevant terms involved in content management were defined. The definitions were kept very wide in order to analyze all aspects of content management, especially because of its fast development.
CMS were delimited to document management systems and to groupware systems. Their main difference is the understanding of elements to be included and the ability to publish the elements, which is only intended in content management.
General benefits of content management were described. They can all be put down to the facts of a central repository for ALL content, its separation into content chunks and presentation objects, and the integration of workflow features that aim at publication of the same content on different publication media. As a basis for the comparison, the stages of the content life cycle were introduced

Chapter 2 described an approach for comparison criterions. Emphasize was laid on general validity for content management, including future developments. The central idea of the comparison method is to split a content management system into main components, assign the content life cycle stages to them and finally compare, which stages the systems support and how. The main components of a content management system are: collection system, publication system and repository system. Additionally, the workflow and the administration systems accompany the whole content life cycle.
At the end of the chapter, the universal content management system, which was created from the sum of the four systems, was described and its functionality during the life cycle stages shown.

In chapter 3, the comparison criterions were applied for a comparison of the four systems. For each system, the life cycle stages were examined and a final evaluation offered. The integration of XML in the systems was analyzed separately, because XML plays a growing role in content management. Finally, typical application areas of each system and their major strengths were described. Aggregating the system's most compelling properties to one keyword, it could look as follows:

- IBM CM: repository
- Hyperwave: collaboration
- Zope: flexibility
- VIP CM: all-rounder

All system have interfaces to third-party products, as it can be assumed that content management systems need to be integrated into already existing applications of the customer. Additionally, the manufactures offer a lot of attractive supplemental products.

Concluding remark

Though all systems mainly support the content life cycle, they have specialized on certain stages. Moreover, none of the four systems seriously supports the theoretical idea of publishing common content to different target media, not only to the Internet.

Consequently, there is no universal content management system – it is only a theoretical construct to show the position of certain content management systems. Generally, content management should be regarded as a concept or strategy rather than a technology. In the end, successful content management depends on its translation into action within the organization.

List of Figures

Bibliography

[1] Zschau O. : ”Web Content Management – Introduction, Concepts, Software”, diploma thesis, University of Applied Science, department of media and electrical engineering, Mittweida, 2000

[2] Degenhardt U.: “Beispiel einer Evaluation von Web Content Management Systemen”, survey, 2000, http://www.internetmanagement.ch/index.cfm/fuseaction/shownews/newsid/92

[3] Riedl T.: “Zope (Z)open Zource Zoftware”, article, 1999, http://www.linux-magazin.de/ausgabe/1999/06/Zope/zope.html

[4] Tegtmeyer F.: “Objekthierarchie”, article, iX 8/1999, http://www.heise.de/ix/artikel/1999/08/064/

[5] Amos, Grubert N.: “Was ist Acquisition?”, How-to, http://www.zope.de/bibliothek/How-Tos/How-Tos/Acquisition_html

[6] http://www.contentmanager.de

[7] Irlbeck T.: “Computer Lexikon”, Deutscher Taschenbuch Verlag, 1998

[8] Reynolds J., Kaur A.: “Content Management”, white paper, 04/2000 http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/ecommerce/maintain/operate/contmgt.asp

[9] Rohse F., Lennartz S.: “Wuchern mit Pfründen”, article, C’t 21/2000

[10] Winand U., Schellhase J.: “Web-Content-Management”, article, wisu – das wirtschafts-studium, 10/2000, pp. 1334

[11] Eppler T.: “Document Components als Brücke zwischen Dokumenten- und Webcontent-Management”,article,11/2000, http://www.internetmanagement.ch/index.cfm/fuseaction/shownews/newsid/204/

[12] Schuster E., Wilhelm S.: “Content Management”, encyclopedia entry, http://www.gi-ev.de/informatik/lexikon/inf-lex-content-management.shtml

[13] Hyperwave User’s Guide, volume 2, 1999

[14] Heuer A., Saake G.: “Datenbanken: Konzepte und Sprachen”, MITP GmbH Bonn, 2000

[15] Titzler P.: “Entwurf und Implementierung eines Virtuellen Dokumenten-Servers”, diploma thesis, University of Rostock, department of computer science, 1999

[16] Kampffmeyer U., Merkel B.: “Grundlagen des Dokumentenmanagments”, Gabler, 1997

[17] Boiko B.: “Content Manager Bible”, excerpt from unpublished book http://www.metatorial.com/cmbible/cmbible.htm

[18] “contentmanagement”,Mindwrap,Inc. http://www.mindwrap.com/infoblurbs/info_contentmgmt.html

[19] Rothfuss G., Ried C.: “Content Management mit XML”, Springer, 2001

[20] IBM CM for Multiple Platforms 7.1, client application help

[21] Hodge G.: “Best Practices for Digital Archiving”, dlib magazine, 01/2000, volume 6, number 1, http://www.dlib.org/dlib/january00/01hodge.html

[22] “Igniting e-business with e-content”, brochure, http://www-4.ibm.com/software/data/cm/attach/cmfeb01.pdf

[23] Hyperwave Administrator's Guide, Volume 3, 1999

[24] Kappe F.: "XML and Hyperwave Information Server", technical note, 1999
http://www.hyperwave.com

[25] http://www.zope.org

[26] http://www.cybertoch.de/informationszeitalter.htm

[27] http://www.hyperwave.com

Statement of Independence

I herewith assure that I have written the present thesis on my own, using only the indicated references.

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig und nur unter Vorlage der angegebenen Literatur und Hilfsmittel angefertigt habe.

Daniela Wersin
August 15th, 2001