Content Management-System

Projektarbeit: Content Management-System
Fachbereich Kommunikationstechnik
Stefanie Dimpfl, Ronny Ellermann, Sabrina Erdle
Höhere Berufsfachschule Medien in Neustadt
November 2002 bis Februar 2003

Inhaltsverzeichnis

1 Geschichte 03

1.1 Geschichtlicher Hintergrund
1.2 Digitalisierung der IT-Abteilung in den letzten 30 Jahren
1.3 Weiterentwicklung im CMS-Bereich

2 Begriffe 09

3 Managements 12

3.1 Content
3.2 Content Management
3.3 Content Management-System
3.4 Asset Management
3.5 Knowledge Management

4 Architektur eines CMS 25

4.1 Anatomie der Dokumente
4.2 Content Lifecycle
4.3 Kernkomponenten
4.4 Serverkonzepte

5 Entwicklung von HTML zu XML 38

5.1 HTML
5.2 DHTML
5.3 XML ergänzt HTML
5.4 Aktueller Trend XHTML

6 Akzeptanz 42

6.1 Einführung neuer Konzepte und Modelle
6.2 Webpublishing
6.3 Mitarbeiter im Bezug auf CMS
6.4 Personal
6.5 Kunden

7. Softwarebeispiele 47

7.1 Dante
7.2 Zope

8. Anhang 57

8.1 Quellen
8.2 Protokolle
8.3 W-Fragen
8.4 Ziele unserer Website
8.5 Zeitplan
8.6 Arbeitsplan
8.7 Expertengespräche
8.8 Erfahrungsberichte

1. Geschichte

1.1 Geschichtlicher Hintergrund

Der Begriff "Content Management" lässt meist den Gedanken entstehen, dass es sich hier um eine neumodische Erscheinung aus dem Bereich Management handelt. Doch dem ist nicht so! Wie bei vielem anderen auch, gab es schon zu früheren Zeiten eine Grundidee, aus deren Wurzeln sich dann das "Content Management" weiterentwickelte.
Bereits in der Antike (2500 v.Chr. bis 476 n.Chr.), zu Zeiten Alexander des Großen, gab es den ersten Versuch "... das Wissen der Welt nicht nur zu sammeln, sondern auch zu systematisieren."[1] Ptolemäus I. , ein General Alexander des Großen, ließ in Ägypten, dass er bei der Aufteilung des Alexanderreiches zugesprochen bekam, eine Bibliothek errichten - die berühmte Bibliothek von Alexandria. Seine Idee war alles Wissen der Welt an einem bestimmten Ort zu sammeln. Er erteilte den Auftrag, "alle Bücher dieser Welt zu beschaffen."[2] Es wurden Such-Agenten im ganzen Mittelmeerraum umhergeschickt, nach Büchern zu suchen. Die Bedeutung Alexandrias, als größtes Handelszentrum, erwies sich von großem Vorteil. Handelsschiffe die zahlreich im Hafen Alexandrias einliefen, wurden gezwungen, falls Schriftrollen an Bord waren, "... diese den Kopisten der Bibliothek zum Abschreiben zur Verfügung zu stellen, woraufhin sie dann diese Abschriften wieder in Empfang nehmen durften. Die Originale verblieben in der Bibliothek."[3] So entstand das größte Informationszentrum ("antiker Wissenspool, mit 400.000 bis 700.000 Buchrollen") jener Zeit. Die Bibliothekare schufen eine "antike Buchverzeichnung", in dem sie ihre Schätze inventarisierten und systematisch ordneten.

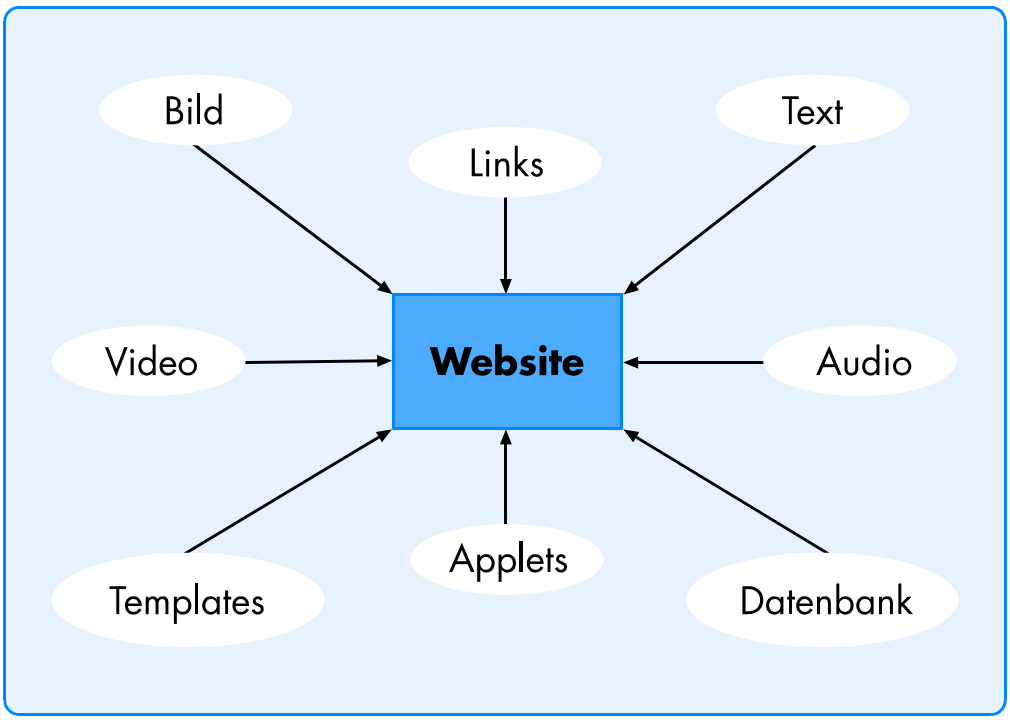
Einer der größten hellenistischen Dichter und Grammatiker, Kallimachos von Kyrene, erfand "... sogenannte Pinakes, annotierte Bestandsverzeichnisse (...), die wir heute als Metainformationen von Dokumenten zu bezeichnen pflegen: Autor samt Bibliographie, Entstehungszeitraum, genauer Zeilenumfang und die einleitenden Sätze des jeweiligen Werkes. Auch die Rollen, die das eigentliche Werk enthielten, wurden mit einem Etikettierungssystem versehen, das es ermöglichte, sie zu identifizieren und bezüglich Inhalt zu klassifizieren, ohne sie dazu auch nur aus den Regalen ziehen zu müssen."[4]

Wenig später jedoch stellte sich heraus, wie sinnvoll diese Art von Archivierung war.

Als Caesar bei seiner Ankunft in Ägypten in den Geschwisterstreit zwischen Kleopatra und ihrem Bruder Ptolemäus XIII. verwickelt wurde, musste er im Hafen von Alexandria um sein Leben kämpfen. Er ließ darauf eine Flotte von Ptolemäus anzünden, die unglükklicherweise auch eine Außenstelle der Bibliothek in brannt setzte. 40.000 Schriftrollen wurden dadurch vernichtet. Die Bibliothekare hatten jedoch zum Glück durch die Strukturierungsmaßnahmen die wesentlichen Informationen in der Hand, was genau verlorengegangen war. "... diese Informationen - sowie eine zugegebenermaßen durchaus rigide Wiederbeschaffungspolitik - versetzte sie in die Lage, den Bestand der wertvollsten der verlorengegangenen Dokumente nach nur rund 30 Jahren wiederaufzufüllen."[5]
Demzufolge könnte man glauben, dass in der heutigen Zeit "Dokumentstrukturierung und Klassifizierung" keine Probleme mehr mit sich bringen. Aber wie aktuelle Untersuchungen zeigen, ist dies keine Selbstverständlichkeit. Es bestehen alle Ansichten, dass die 90er Jahre des 20 Jahrhunderts, "... als das undokumentierteste Jahrzehnt in die Geschichte der Inhaltsaufbereitung einzugehen (...)"scheint.[6] Dies ist auch kein Wunder, denn die Möglichkeiten der Speicherung von digitalen Medienobjekten nimmt immer rascher zu im Gegenteil zu den Strukturierungsmöglichkeiten, die mangelhaft sind, was der

1-6 (http://www.community-of-knowledge.de/cp_artikel_d.htm?artikel_id=60)

Wiederauffindbarkeit und Wiederverwendbarkeit von Objekten äußerst störend im Wege steht. Laut Schätzungen geht man davon aus, "... dass 90% aller Unternehmensdokumente Inhalte in unstrukturierter Form enthalten, dabei verbraucht eine moderne Volkswirtschaft ungefähr ein Viertel ihrer Ressourcen für Informations - i. e. Inhaltsgenerierung und -verwaltung."[1] Die Verluste die Unternehmen und Dienstleister durch schlecht aufbereitete Informationsquellen und deren mangelhaften Pflege erreichen, liegen bis 2003 bei rund 31,5 Mrd.$. Zusätzlich wird das Ganze verstärkt durch Webauftritte mit tagesaktuellen Informationen, die in unserer heutigen Zeit zu den üblichen Standards gehören nicht nur für Unternehmen in der Medienbranche, wie es sich an der Entwicklung elektronischer Bestellvorgänge und Abwicklung von Kundenaufträgen zeigt. Des weiteren benötigt auch das firmeninterne Intranet und Extranet mit Geschäftspartnern und Lieferanten einen reibungslosen Austausch mehrwertschöpfender Informationen. Die Informationen bzw. das gespeicherte Datenmaterial sind jedoch das eigentliche Problem. Da die unterschiedlichen Dateiformate zum Beispiel bei einem Webauftritt zur Ansicht, einen eigenen Verarbeitungs- und Aufbereitungsprozess benötigen. Noch komplexer sieht es bei Intranet und Extranet aus.
Das folgende Schaubild spiegelt die gängigsten, möglichen Dateiformate für Webauftritte wieder:



Es stellt sich die Frage, wie schafft man es, benötigte Informationen überhaupt aufzufinden, wie trennt man veraltetes von aktuellem, wie verwaltet und katalogisiert man völlig unterschiedliche Dateiformate, macht sie zugänglich und stellt sie auf einer möglichst einheitlichen Benutzeroberfläche dar? Die Antwort auf diese Frage finden sie im Kapitel 3.3 „Das Content Management System".

1 (http://www.community-of-knowledge.de/cp_artikel_d.htm?artikel_id=60)

1.2 Digitalisierung der IT-Abteilung in den letzten 30 Jahren

In den letzten 30 Jahren wurde in der IT-Abteilung alles elektronisiert.
Somit entstanden viele "Informationsinseln", d.h. es wurden viele zusammengehörende Informationen an vielen unterschiedlichen Stellen gespeichert.

Seit geraumer Zeit wird versucht sich mithilfe von Customer Relationship (CRM) einen Überblick über zusammengehörende Informationen zu machen.
Dies erfolgt, indem man mühsam und kostspielig durch eine Systemintegration, zumindest eine Sammlung und Zusammenfassung vorhandener Informationen aus den unterschiedlichen IT-Systemen angestrebt. D.h. die Inseln werden miteinander Verbunden bzw. miteinander in Bezug gesetzt!

Man Versucht also, historisch gewachsene Informationen, die irgendwo im Unternehmen verstreut sind, nutzbar und wieder findbar zu machen! Aus Daten und Informationen werden so Nutzbare Contents.

Aber dennoch gibt es viele Unternehmen, bei denen sämtliche alte Organisationsmittel geblieben sind (Hängeregistratur, Ordnerablage,...). Dies führt zu einer konsequent gelebten Degradierung von Contents (findbar, transferierbar, nutzbar) zu Informationen, („irgendwo da unten im Stapel müsste es sein!"[1]). Somit gibt es überall in Unternehmen Inseln mit Informationen, die schlecht oder gar nicht findbar sind und damit nicht weitergegeben werden können.

1.3 Die Weiterentwicklung im CMS-Bereich

Wie Sie bereits aus dem Kapitel 1.1 „Geschichtlicher Hintergrund" erfahren haben, ist Content Management (CM) ein alter Hut. Denn der Umgang mit Inhalten war schon immer ein wichtiger Faktor gewesen und ist in unserer, immer mehr informationsüberfluteten, Gesellschaft zu einem elementaren Bestandteil geworden. Er „... spiegelt sich (lediglich) in unterschiedlichsten Szenarien mit immer wieder neuen Namen wider."[2]
1999 und 2000 war CM sogar als das Unwort des Jahres gewählt worden, auf das keine Unternehmensberater oder Online-Magazine „... verzichten konnten und als absolute Notwendigkeit betrachteten."[3] Obwohl CM eigentlich nur den Wunsch beinhaltet, in größerem Sinne, Ordnung zu schaffen, speziell bezogen auf das anarchistisch entwickelte Internet. Trotzdem ist festzustellen, dass es immer noch kein System gibt, welches die gewünschte Flexibilität besitzt, allen Unternehmen gleichermaßen gerecht zu werden, obwohl sich „Web Content Management Systeme" (WCMS) immer mehr zu „All-in-one-Werkzeugen" entwickeln.

Hier ein kurzer Einblick in die zukünftig möglichen Weiterentwicklungen im CMS-Bereich:

- XML in WCMS
- Portale
- Content Syndikation
- personalisierte Inhalte

1 (Koop, H.-J.;Jäckel, K.K.; Van Offern, K.K.: Erfolgsfaktor Content Management - Vom Web Content bis zum Knowledge Management, 1. Auflage, Wiesbaden, Vieweg, 2001, S.30, Z. 32);
2-3 (Zschau, O.; Traub, D.; Zahradka, R.: Web Contnent Management - Websites professionel planen und bearbeiten, 2. Auflage, Bonn, Galileo Press GmbH, 2002, S.233/ff)

XML in WCMS

- XML-Definition vgl. Kapitel 5 Entwicklung von HTML zu XML
- XML ist bereits weit verbreitet in WCMS und meist ein integraler Bestandteil von Import- und Exportschnittstellen

Vorteile durch XML:
- die Strukturierung und die Medienunabhänigigkeit von Inhalten
- der vereinfachte Datenaustausch

Nachteile:
- die Vorteile von XML werden noch nicht voll ausgenutzt
- es gibt noch keine Browserumstellung auf XML und XSL
- für Webpräsentationen ist noch die Umwandlung von XML in HTML nötig

Fazit:
- die XML-basierte Datenhaltung wird sich mit der Zeit etablieren
- sie ist die Voraussetzung für „Content Syndikation"

Portale

Definition:
- Portale sind zentrale Plattformen für viele Benutzer gleicher Interessensgebiete
- sie bedienen den Massenmarkt
- stellen eine Ansammlung von wichtigen Informationen, in konzentrierter Form zur Verfügung
- verweisen meist auf interne und externe Websites
- zum Beispiel haben sich viele Portale aus ehemaligen Suchmaschinen entwickelt, wie Yahoo, AltaVista, AOL, ...

Es gibt drei verschiedene Arten:
- vertikale Portale
- horizontale Portale
- Unternehmensportale („Enterprise Portals")

Vertikale Portale:
- setzen sich mit bestimmten Themen auseinander
- werden meist von einem Anbieter vertrieben
- bieten:
  - fokussiertes Wissen und weiterführende Themen
  - direkte Unternehmensanbindungen, dieser Branche

Horizontale Portale:
- sind personalisierte Portale, mit der Absicht den individuellen Wünschen des Benutzers gerecht zu werden
  Vorteil:
  - der Benutzer kann die Inhalte seiner Startseite nach belieben selbst zusammenstellen

Unternehmensportale (Enterprise Portals):
- dienen als Startpunkte für Mitarbeiter eines Unternehmen, bei Informationssuche
- bieten:
  - Intranet- und Internetzugang

- fasst firmenrelevante Informationen für den Mitarbeiter zusammen und macht sie ihm zugänglich

- Zugriffe auf E-maildienste und Datenbankinhalte
- Kommunikationswerkzeuge
- Kalenderfunktionen

Fazit:
- Portale sind momentan voll im Trend
- sie erweitern bisherige Geschäftsmodelle

Content Syndikation

Def.: - bezeichnet den Austausch und den Handel von Inhalten für das Publizieren im Web über „Content-Provider"

Content-Provider:
- sind Händler/Lieferanten von Inhalten, auch bezeichnet als Syndicator
- erstellen ICE-Pakete, die beim Empfänger entpackt und automatisch in die Website integriert werden können
- führen Inhalte unterschiedlichster Quellen zusammen
- stellen Inhalte in verschiedene Sprachen oder Ausgabeformaten zur Verfügung
- klären die rechtliche Lage für ihre Auftraggeber
- der Austausch läuft meist über eigene Protokolle und Übertragungsmöglichkeiten (z.B. über ICE)
- seine Hauptaufgabe ist es für den Kunden den gewünschten Inhalt zielgerecht bereitzustellen

ICE:
- „Information and Content Exchange"
- ein Standard, der all die verschiedenen Protokolle vereinheitlichen soll
- entwickelt von der Authoring Group
- basiert auf XML
- bidirektionales Protokoll, steuert die Übertragung zweier Systeme in beide Richtungen
- kontrolliert, automatisiert und reguliert den Austausch von Webinhalten

Fazit:
- das Content-Angebot kann über ICE versendet, verrechnet und sogar mit Copyrights und weiteren Metainformationen versehen werden
- erhebliche Kostenersparnis durch die Vereinfachung

Personalisierte Inhalte

- die Personalisierung erzielt, dass Webseiten genau die gewünschten, benutzerspezifischen Informationen enthalten
- man spricht hier auch von dem „gläsernen Benutzer", dem Inhalte präsentiert werden, ohne dass er danach suchen muss

Die Personalisierung wird in drei Vorgänge unterteilt:
- Profiling und Tracking
- Matchmaking
- Channeling

Profiling und Tracking:

- Benutzerdaten (Geburtsdatum, Hobbies, Besuche spezieller Websites, ...) werden gesammelt und ausgewertet

Matching:
- vorhandene Inhalte/Informationen werden mit dem Benutzerprofil abgeglichen

Channelling:
- die ausgewählten Informationen erhält der Benutzer automatisch oder vorausgewählt über seine favorisierten Kommunikationskanäle (E-Mail, SMS, FAX,...)

Fazit:

- die Methoden klingen theoretisch einfach, sind aber von der Realität der WCMS noch weit entfernt
- nur speziell dafür entwickelte Systeme (wie z.B. Cycosmos), ermöglichen diese Art von Personalisierung

Problem:

- man muss mit der Speicherung von gesammelten Benutzerdaten äußerst sensibel umgehen, wegen dem Datenschutz
- das Tracking, von dem der Benutzer nichts weiß, stößt immer noch auf Kritik

Résumée

Der Wunsch aller wäre doch ein System, dass alle Funktionalitäten in sich vereint und so immer die richtige Lösung darstellt. Man spricht dann auch von dem Wunsch nach einer „eierlegenden Wollmilchsau". Sie ist ein Fabeltier, die alle wichtigsten Bedürfnisse der Menschen abdecken würde. Leider handelt es sich hierbei um reine Utopie, die es in unserer Realität nie geben wird, obwohl es schon einige Lösungen gibt, die eine Annäherung an dieses Wesen erkennen lassen. Jedoch ist meistens, wie bei allen Dingen im Leben ein kleiner Hacken bei der Sache. Viele dieser Lösungen sind zum Beispiel an ein bestimmtes Projekt, eine Website, eine vorgegebene Technologie, ... gebunden. Aus diesen Gründen kann es wohl nie ein System geben, dass alles kann und darüber hinaus keine großen Kenntnisse, zur Anwendung erfordert.
Standardisierte Technologien wie Java oder XML nähern sich dieser allgemeinen Lösung schon an, sind aber jedoch noch sehr jung auf dem Markt. Die Entwicklung bleibt weiterhin spannend und wer weiß, vielleicht gibt es doch irgendwann eine „eierlegende Wollmilchsau".

2. Begriffe

Daten:

- unterste Ebene von Informationsverarbeitung;
- einzelne Bits und Bytes, die einem bestimmte Buchstaben, Wort, ein Pixel eines Bildes o.a. repräsentieren

Information:

- nächsthöhere Ebene von Daten;
- setzen sich aus Daten zusammen, Voraussetzung: dass die zugrunde liegenden Daten eine Struktur aufweisen

Content:

Informationen sind gleich bedeutend im Austausch mit Anderen. Als Content kann man Informationen als Objekte oder Austauschgegenstände verstehen
Liegt eine Information also in einer Form vor, in der man sie an Andere weitergeben kann, spricht man von Content (z.B. Texte, Bilder...).
Man kann ganze Informationspakete mittels eines Mediums weitergeben, egal ob dies für den Anderen einen Nutzwert darstellt.

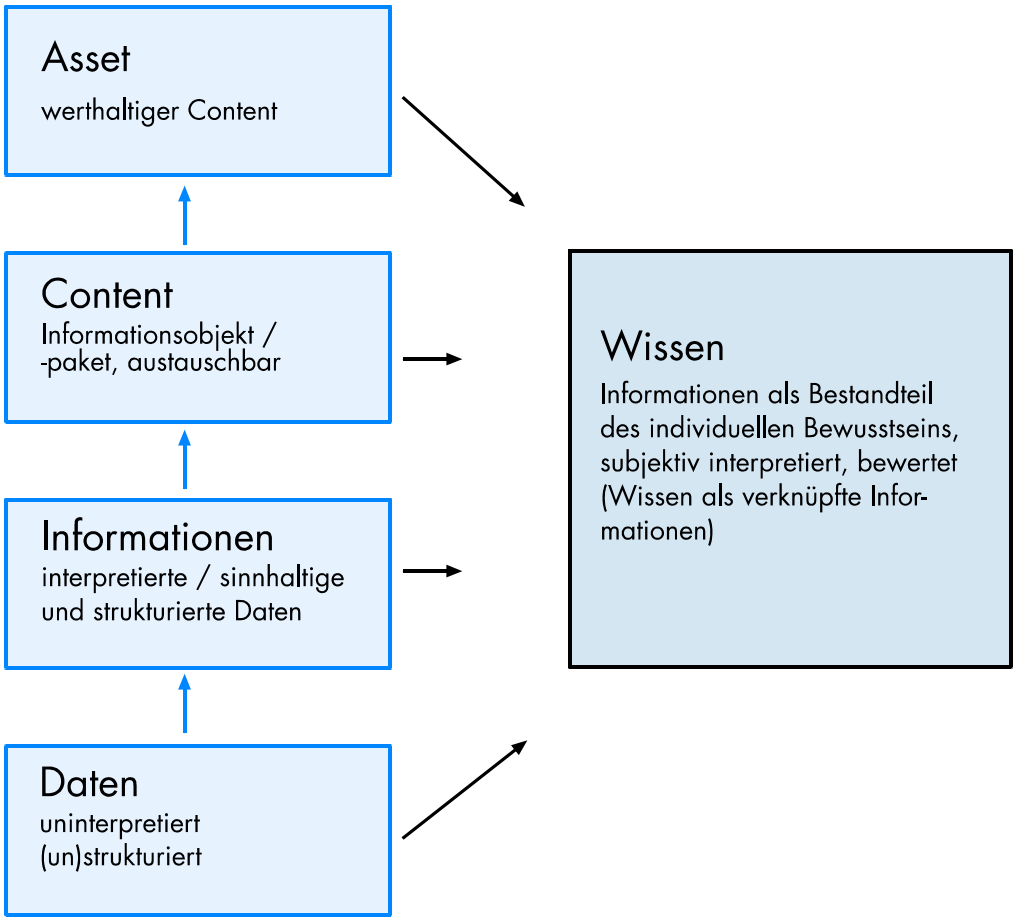
Asset:

- Charakteristikum des Nutzwertes von Contents
Asset sind Contents die für den Nutzer einen bestimmten Wert repräsentieren. Aufgrund des Wertes ist ein potenzieller Nutzer bereit, Mühe für die Beschaffung des Asset aufzuwenden, diesen gegebenen falls zu bezahlen oder ihn gegen andere Assets zu tauschen.

Wissen:

Wissenmanager u.a. bauen auf der Annahme auf, Wissen erfassen, speichern und wieder verfügbar machen zu können. Unter Wissen sind Informationen als Bestandteil des individuellen menschlichen Bewusstseins zu verstehen Wissen besteht aus subjektiv interpretierenden und bewerteten Informationen, die erst durch die individuelle Verknüpfung mit anderen Informationen ihren speziellen Wissenscharakter erhalten.

Begriffs-Schaubild:



2.6 Anwendungsformen der Internet- und Webtechnologien

Mit Web sind hier alle drei möglichen Anwendungsformen der Internet- und Webtechnologien gemeint, nämlich Internet, Intranet und Extranet.

Intranet:

Besitzt eine definierbare, meist unternehmensinterne Zielgruppe. Die Mitarbeiter eines Unternehmens tauschen hier Informationen aus und haben die Möglichkeit, Anwendungen zur Reisekostenabrechnung etc. zu nutzen.

Internet:

Angebote im Internet sind für die Öffentlichkeit gedacht und meist uneingeschränkt für alle Nutzer einsehbar.

Extranet:

Es ist eine Internetapplikation für die Kommunikation von Unternehmen mit ihren Außenstellen, Partnern, Zulieferern oder Investoren. Die Zielgruppe ist hier weiter gefasst als beim Intranet, aber dennoch eingeschränkt.

In allen Erscheinungsformen des Webs muss Content gemanagt werden. Die Unterschiede liegen jedoch in den Ansprüchen an das System.

3. Managements

3.1 Content

Das von der Globalisierung geprägte Wirtschaftsleben erfordert Geschwindigkeit in Abläufen und zugreifbare, nutzbare Informationen als Rohstoff. Diese werden erreicht durch Rationalisierung, Automatisierung und Standardisierung.

Zugreifbare und nutzbare Informationen sind abhängig davon, wie umfassend und integrativ im Unternehmen Informationen und Contents „gemanagt" werden.

Als Content bezeichnet man findbare, transferierbare und nutzbare Informationen. Content ist dadurch, der Interessenpunkt jedes Unternehmen, da man durch den effizienten Umgang Produktivitätsvorteile erzielen kann.

Integrationswirkung des Contents

Customer Relations Management ist nur dann sinnvoll, wenn eine Integration von Informationen sowie eine Übersicht über Informationsinseln besteht.

Die Nutzbarkeit von Informationen beruht auf der Transferierbarkeit und somit der Fähigkeit, Informationen zu finden. Die Zusammenfassung und Bündelung von Informationen aus den unterschiedlichsten Quellen generiert Contents.

Content und Beschäftigung ist die Hauptschnittstelle für zentrale und erfolgskritische Informations- und Austauschprozesse im Unternehmen.

Wofür Contents?

Interner Vorteil:
- Nutzbare Informationen such- und wiederfindbar
- Verbesserung der Informationsflüsse
- Wissenszusammenhänge werden transparenter
- Entscheidungen stehen auf einer besseren Basis, können schneller getroffen werden

Externer Nutzen:
WETTBEWERBSFÄHIGKEIT
  - Corporate Identity (CI)
  - Corporate Communication (CC)

- Eine Content Erstellung, Bearbeitung und Produktion geht nicht nur schnell, sondern auch einfach vonstatten, sie erfüllt zudem gleichzeitig die CI-Vorgaben und transferiert den minimierten Arbeitsaufwand für die Mitarbeiter in ein nach außen einheitliches und aktuelles Erscheinungsbild. Dadurch wird eine einheitliche Außenwirkung, unabhängig von der internen Quelle und der Beschaffenheit des Contents, erzielt.

- Durch das neu gewonnene Wissen kann man neue Geschäftsfelder erschließen z.B. Content Management als Dienstleistung.

- Die (Geschäfts-) Abläufe beschleunigen sich, Rückfragen werden seltener, die Geschäftsprozesse mit Ihrem Unternehmen werden für Kunden und Lieferanten noch einfacher und somit angenehmer, d.h. aus der Sicht ihrer Partner steigt die Zuverlässigkeit, Kompetenz und die Geschwindigkeit ihres Unternehmens. Folglich wird das Unternehmen wettbewerbsfähiger.

Merkmale von Content

Inhalt:

Eigentliche Basisinformation z.B. einen reinen unformatierten Text

Struktur:

Überschriften, Absätze, Querverweise gehören zur Struktur, gleichgültig, ob sie sich auf den selben Text beziehen oder auf das ganze Content. Die Gesamtstruktur eines Content besteht aus einer inneren Struktur und gegebenenfalls aus der Einbindung in oder mit anderen Informationen (der Externen Struktur).

Formatierung:

Aufmerksamkeit gewinnt man mit Schriftart, Schrifttyp und Hervorhebungen.

Layout:

- es umfasst eine Vielzahl textlicher u. graphischer Gestaltungsmöglichkeiten, bis hin zu Designelementen
- zweispaltig, Bildunterschriften, Tabelle, Schriftfarben, Form

Medienformat:

Entscheidung zwischen einer internen Formatierung und der äußeren oder medialen Form
- dem Medienformat
- qxd-Datei, HTML-Site,...

Medium:

Entscheidung für ein bestimmtes Trägermedium (Medium) Diskette, Papier...

Diese Merkmale der Klassifizierung gelten für alle Arten von Informationen. Je komplexer das Medium desto vielfältiger werden die unterschiedlichen Ausprägungs- und Gestaltungsmöglichkeiten.

Medienneutralität

Als medienneutral werden nur solche Informationen bezeichnet, die hinsichtlich des Mediums nicht kategorisiert oder gebunden sind. Sie sind somit also in jede Medienform konvertierbar oder sollten dies zumindest sein. In der Praxis sieht dies schnell anders aus, da durch konvertieren oft Informationen und Qualität verloren gehen.

Content-Orientierung

Jeder Content ist hinsichtlich der genannten Merkmale genau spezifiziert. Das Content-orientierte Management von Informationen ist demnach eine Speicherung und Verarbeitung von Informationen, bei denen unabhängig vom späteren Medium, die einzelnen Elemente des Content getrennt voneinander erfasst, gespeichert und abgerufen werden können als:
- Inhalt
- Form
- Struktur
- Layout

Asset

Ein Asset ist ein fertiger Content, der vermarktbar ist, der einen Wert hat und dem man möglicherweise ein „Preisschild" aufkleben könnte.

<u>Folgende Faktoren machen einen Content zu einem Asset:</u>

- Markt:
  Bereitschaft für den Content Geld auszugeben

- Aktualität:
  Ein Asset ist aktuell und wird aktuell gehalten

- Nutzwert:
  Ein Asset bietet einen gewissen Nutzwert für einen Kunden, der über kurzfristige freiverfügbare Informationen hinausgeht

- Relevanz/Zweck:
  Ein Asset muss auf ein Bedürfnis, bzw. eine Fragestellung genau passen

- Herausgeber:
  Ein Herausgeber hat eine bestimmte Anerkennung und einen guten Ruf in Bezug auf die Contents

Durch diese Faktoren entsteht die Forderung nach immer besseren, schnelleren und spezielleren Assets.

Aus der heutigen Sicht wird der optimale Asset digital geliefert und vermittelt fundiertes und aktuelles Know-how, welches den Nutzer ein Stück weiterbringt und gleichzeitig das Interesse erstickt, den Asset weiterzugeben oder zu kopieren.
Fraglos sind solche Assets recht selten! (vergl. Erfolgsfaktor Content Management; S39ff)

3.2 Content Management

Der zielgerichtete-, systematische- und durchgängige Umgang mit Informationen (Contents) bezeichnet man als Content Management. Content Management übernimmt meist folgende Funktionen:

- Erzeugung (Generierung)
- Verwaltung (Organisation und Aufbereitung)
- Zur-Verfügung-Stellung (Distribution)
- Schaffung von Nutzungs- und Verarbeitungsmöglichkeiten (Nutzung)

Content Management schafft lediglich die notwendigen Voraussetzungen dafür, Content einfach, schnell, und reibungslos in einer Vielzahl von Verwendungsmöglichkeiten und -formen nutzen zu können. Ein Content Management ersetzt keine Redaktion oder Layoutabteilung.

In der Praxis finden sich häufig Systeme, bei denen Funktionalitäten zur Verarbeitung und Nutzung (Redaktion, Layout usw.) mit eingebunden sind - speziell bei Web Content Management-Systemen. Diese Funktionalitäten stellen aber nicht den eigentlichen Kern des Systems dar, sondern sind eigentlich als sinnvolle Zusatzfunktion zu betrachten.

Gerade bei sehr leistungsstarken Content Management-Systemen mit umfangreicher Funktionalität hinsichtlich Erzeugung, Verwaltung, und Zur-Verfügung-Stellung von Contents ist eine Funktion zur Nutzung der Contents häufig nicht integriert, sondern wird mit Hilfe separater Redaktions- oder DTP-Systeme durchgeführt. Gerade wenn diese nicht Bestandteil des Content Management-Systems sind, muss der funktionierenden Verknüpfung dieser Systeme und der damit verbundenen Schnittstellen-Problematik besondere Aufmerksamkeit gewidmet werden.
Über die drei Grundfunktionen - Erzeugung, Verwaltung, und Zur-Verfügung-Stellung hinaus, dient ein echtes Content Management aber auch zur Erfüllung weiterer Zielsetzungen:
- Aktualität
- Reliabilität (Zuverlässigkeit)
- Qualitätssicherung
- Gewährleistung inhaltlicher Konsistenz

Darüber hinaus gibt es Einschränkungen hinsichtlich der Leistungsfähigkeit. Selbst ein professionelles Content Management kann nicht Relevanz oder Nützlichkeit sicherstellen.

„... ,dass bei der Auswahl innerhalb eines Content Managements zu erfassenden Contents, potentielle Relevanz und Nützlichkeit in besonderem Maße zu beachten sind."[1]

Entscheidend:

Der Sinn eines Content Managements liegt darin, zu vereinfachen, zu standardisieren, Zeit zu sparen und bessere Ergebnisse zu erzielen.
Das spricht auf Dauer gegen eine manuelle Lösung des Content Managements mit Papier und Bleistift. Dies sollte aber nicht eine Einführung eines Content Management System erzwingen, denn mit diesen Systemen wird nur dann etwas erreicht,

1 (Koop, H.; Jäckel, K.; Offern, A.: Erfolgsfaktor Content Management, 1. Auflage, Braunschweig/Wiesbaden, S.16, Z6);

wenn sie:
- tatsächlich Vereinfachungen in den Abläufen und Verbesserung der Informationsflüssen bewirken
- nicht noch eine Insellösung oder noch ein zusätzliches Organisationssystem bedeuten
- kostenmäßigen Rahmen bleiben
- von den Benutzern akzeptiert werden

u.a. (siehe Erfolgsfaktor Content Management; S.50)

3.3 Content Management-System

CMS ist ein IT-basiertes System zur Organisation, Verwaltung und Durchführung des Content Managements. Es ist also folglich ein elektronisches Tool für Content Management. Ein solches System besitzt zudem Zusatzfunktionen, sogenannte Funktionalitäten, die zur Verarbeitung und Nutzung (Redaktion, Layout) integriert sind.
Achtung, wenn die Redaktions- und DTP-Systeme nicht Bestandteil des Content Management-Systems sind, muss der Verknüpfung und der somit verbundenen Schnittstellenproblematik besondere Aufmerksamkeit gewidmet werden.
Man sollte sich merken, dass ein Content Management-System weder ein Redaktionssystem, noch ein Knowledge Management-System ist. Ein Redaktionssystem besitzt lediglich Bausteine/Elemente zum Content Management-System.

Teile von Content Management-Systeme sind:
- Content Organisation
- Aufbereitung
- Content Generierung
- Distribution

Ein CMS ist also ein Teil des unternehmenseigenen Intranets, ggf. des Extranets, sofern Kunden, Lieferanten oder sonstige direkt in das System eingebunden werden sollen.

Aufgaben eines CMS

Ein Content Management-System soll die Aufgabe übernehmen, bestimmten Benutzern die Befugnis zu erteilen, definierte Bereiche einer Seite zu verändern und zu pflegen, ohne dass diese einen technischen Sachverstand aufweisen müssen.

Häufig trifft man CMS in der Form eines Forums an, denn auch hier haben Benutzer die Möglichkeit, Beiträge zu verfassen die dann veröffentlicht werden. Diese greifen aber genau wie bei „normalen" CMS niemals tiefer in die Programmierung ein.

Meistens besitzen solche Seiten einen gesonderten Redaktionsbereich, auf den man nur mit Benutzererkennung und Passwort zugreifen kann. Ist man hier angemeldet, so kann man beliebige Änderungen auf der Seite vornehmen.

Auswahl eines CMS

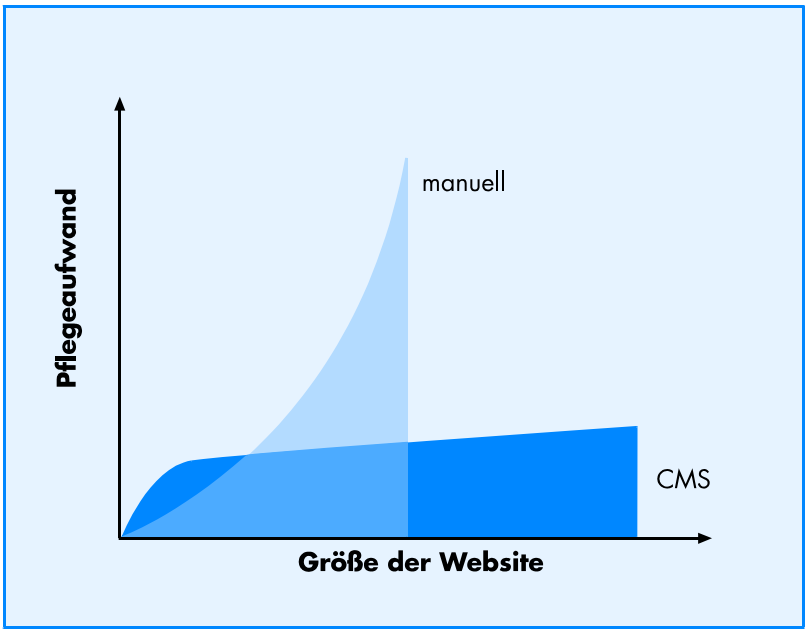
Es gibt zwar sehr einfach gestrickte Content Management-Systeme, doch die meisten verfügbaren CMS sind sehr umfangreich und bieten vielerlei Funktionen.
Einige Content Management-Systeme sind speziell auf die Anforderungen einer bestimmten Branche zugeschnitten oder eignen sich zum Beispiel besonders gut für Kommunalverwaltungen. Manche beschränken sich auf die konkrete Aufgabe der Inhaltsverwaltung, andere sind gleichzeitig ein ganzer Baukasten für Webseiten, mit dessen Hilfe man schnell eine Seite zusammenbasteln kann.

die Auswahl eines CMS ist von strategischer Bedeutung, da ein späterer Wechsel hohe Kosten verursacht. Wenn die Anforderungen nicht zum System passen oder umgekehrt, ist nichts gewonnnen.
Man sollte sich daher einen klaren Kriterienkatalogs um zu jedem Punkt klare K.o.-Kriterien zu definieren, also Punkte die für ihr Content Management-System unabdingbar sind und möglichst vollständig genauso wie gefordert funktionieren oder vorhanden sein müssen.

Wann lohnt sich ein Content Management-System?

Um so häufiger eine Webseite oder sonstige Seiten aktualisiert werden soll und je mehr Personen an der Seite arbeiten, desto lohnenswerter ist ein Content Management-System.
Jedoch ist die Anzahl der an einer Seite arbeitenden Personen keine Bedingung für ein solches System, denn selbst wenn nur eine einzige Person arbeiten würde, könnte ein CMS sehr hilfreich sein. Denn damit könnte man nämlich von jedem Rechner mit einem Internetzugang Änderungen vornehmen. Man bräuchte somit kein FTP-Programm mehr und die meiste Administrationsarbeit wäre stark vereinfacht.

Die Einrichtung eines CMS würde zwar anfänglich Mehraufwand, primär an Zeit, jedoch abhängig vom System, auch durchaus an Geld bedeuten. Der Einsatz ist daher vor allem nur dann sinnvoll, wenn die dank des Systems eingesparte Wartungsarbeit größer ist als die zu dessen Konfiguration benötigte.



Die typischen Problembereiche eines CMS sind:

- Berücksichtigung verschiedener Beschaffungswege
- Unterstützung unterschiedlicher Informationsquellen und - qualitäten
- Unterstützung verschiedener Informationsformen (Foto, Text,...)
- Erstellung von Content
- Strukturierung und Organisation des Contents
- Schnittstellendefinition
- Retrieval und Transfer

In der Praxis werden nur wenige optimale Lösungen gefunden, sofern nicht die Bereitschaft von Seiten des Unternehmens besteht, die bestehende IT-Struktur zumindest teilweise erheblich zu verändern.

Wie funktioniert ein Content Management-System?

Die meisten Content Management-Systeme funktionieren nach dem gleichen Prinzip. Sie laufen als Programm auf dem (Web)Server, verwalten die Daten der Seite und lassen sich über eine Oberfläche steuern.
Diese Programme sind meist in einer Skriptsprache programmiert, entweder PHP oder Perl. Der Server muss über einen Interpreter für die jeweils verwendete Sprache verfügen, damit das Programm funktioniert.

Das CMS erzeugt normale HTML-Dateien, über die man die Inhalte der Seiten verwalten kann. Dazu werden meist Formulare verwendet, in die man neue Daten eingeben kann. Diese werden dann an den Server gesendet und dort ausgewertet. Der Server kann die gesandten Informationen entweder in ganz normalen Dateien ablegen oder in einer Datenbank speichern. Die meisten CMS unterstützen Datenbanken und verlangen unbedingt eine Datenbankunterstützung.

Greift nun ein Besucher auf die Seite zu, so erzeugt das CMS eine HTML-Datei aus den in der Datenbank gespeicherten Informationen. Für den Besucher ist daher nicht ersichtlich, dass die Seite dynamisch generiert worden ist.

Welche Unternehmensbereiche werden bei der CMS-Einführung betroffen?

<u>a) direkt</u>

- IT
- Geschäftsprozesse
- Organisation
- Personal
- Produktion

...

b) indirekt

- Strategie
- Unternehmenspolitik
- Finanzen

Inhalt und Gestaltung

Bei solch einer Lösung ist das Design also völlig getrennt von dem Inhalt einer Seite. In der Datenbank werden nur die reinen Informationen ohne darstellungsrelevante Formatierungsanweisungen gespeichert. Es ist deshalb sehr leicht möglich, das Aussehen der gesamten Seite zu verändern - dafür muss nur der für die Ausgabe verantwortliche Teil geändert werden.

Um ein solches Verfahren zu ermöglichen, muss die Webseite genau strukturiert werden. Alle Inhalte müssen einer Kategorie zugewiesen werden, damit das Content Management-System versteht, wie mit den Informationen umgegangen werden soll und wo sie später angezeigt werden. Dies schränkt vielleicht die Kreativität ein wenig ein, garantiert aber auch einen übersichtlichen und einheitlichen Seitenaufbau.

Somit kann ein CMS als Kernkomponente oder als Gesamtlösung eingesetzt werden!!!

Technik

Die Einführung eines CMS ist durch dessen Integrationspedanterie, eine große technische Herausforderung. Schnittstellen müssen erstellt, getestet und stabilisiert werden. Dazu kommt zusätzlich das Erlernen des neuen Systems und das „Einfahren" des Zusammenspiels vom Server-Betriebsystem bis hin zum Anwender-Client.
Dies bindet Ressourcen und benötigt Know-how.
Wichtige Punkte sind:

Basissysteme:

Vorfestlegung und Beschränkungen für die CMS-Einführung, wenn der Status Quo nicht verändert werden soll:
- Besonders beachten sollte man die vorhandenen Basisinstallationen:
  Hardwarearchitektur, Betriebsystem,… . Denn dies dient der Anpassung des CMS an die aktuelle Struktur.
- vom CMS (aus technischer Sicht) geforderte Standards:
  z.B: XML, HTML, ICE, TCP/IP,… .
- Zusätzliche Hardware- und Software-Anforderungen an die IT-Infrastruktur durch das CMS:

z.B: Clients, Server und die zu Grunde liegende Netzwerkinfrastruktur.

Sicherheit:

Die spezifischen Sicherheitsanforderungen durch ein und in ein CMS:
- Speicherung von empfindlichen Daten
- Betriebssicherheit des Content Management-Systems:

z.B.: Zugriffsschutz und Verschlüsselung der Daten benutzter Subsysteme. Etwa solche, die zwischen dem Content Management-System und seinen Umsystemen benutzt werden.
- Verschlüsselung oder Transparenz der Datenübertragung im LAN und WAN
- Systemstabilität

Schnittstellen und Integration:

Diese sind nur dann von Bedeutung wenn das geplante Content Management-System eine Schnittstelle zum Altsystem haben soll.
- Lösen der integrationsbedingten Beschränkungen, die vor der Einführung eines CMS
- Fehlende Schnittstellen zum CMS aus den Umsystemen feststellen
- Festlegungen für vorhandene Datenbanken

Je Datenbank muss festgelegt werden, ob sie:

- in das Content Management integriert/angebunden wird
- für das CM Daten liefert
- durch das CMS ersetzt werden soll
- nichts mit dem CM zu tun hat, also eine Informationsinsel bleibt

- Schnittstellen und Ihre Einführungsstrategie:
  z.B. ein Parallelbetrieb zwischen Altsystemen und dem CMS aufgrund der vorliegenden Anforderungen des Content Management-Konzepts

IT-Bereich

Es ist besonders wichtig, Anforderungen zu formulieren, die sowohl die Personalbestattung als auch das Know-how der IT-Mitarbeiter betreffen:

Man unterteilt in folgende Phasen:
- Implementierungsphase:
  Hier wird installiert, adaptiert, parametrisiert und gegebenenfalls noch programmiert wird.
- Einführungsphase:
  Erste Benutzer gewöhnen sich an das System
- Betriebsphase:
  Der „Normalbetrieb" hat sich eingependelt und die User können mit dem System brauchbar umgehen.

Die Anforderungen werden sich mit der steigenden Erfahrung der Mitarbeiter wandeln.

Trotzdem sollte es verschiedene Hilfethemen geben:

- Technischer Support für die IT-Abteilung
- Inhaltlicher Support für die Benutzer des CMS
- Schnittstelle zwischen technischen und inhaltlichen Support und Zuständigkeit der IT-Abteilung für inhaltliche Fragen

- Notfallpläne für eventuelle Extremfälle wie Brand, Wassereinbruch, Stromausfall...
- IT-seitige Stützung des Content Managements

Zusätzlich sollte man sich Gedanken darüber machen, wie viele Mitarbeiter mit welchen Know-how für das CMS benötigt werden.

Grundidee

1. Alles wird einmal abgespeichert. Bei der Änderung einer Teilinformation ändern sich alle Objekte, die diese Teilinformation nutzen.
2. Alle relevanten Contents müssen nach individuellen Vorgaben strukturierbar und wiederfindbar gemacht sein, d.h. sinnverwandte Contents können gefunden und in einen neuen Zusammenhang gestellt werden.

Ein Content Management-System ist nie Selbstzweck, sondern entfaltet seine Stärken einerseits gerade in Bereichen, in denen Informationsblöcke strukturiert werden und mit nur einmal vorhandenen Daten (-sätzen) gefüllt werden können. Und andererseits dort, wo Contents inhaltlich strukturiert werden sollen.

Organisation (CMS)

Veränderungen im Unternehmen führen nicht nur zwangsläufig zu Änderungen in der Unternehmenskultur, sondern fast immer auch zu Veränderungen der bestehenden Organisationsstruktur.
„Man sollte also einen prüfenden Blick auf die Organisationsstruktur und -kultur im Unternehmen werfen, denn ein noch so gutes Content Management - gestützt auf ein optimales Content Management-System - ist noch lange keine Garantie dafür, dass die erhofften und erwarteten Effekte auch eintreten, wenn die Organisation nicht stimmt.

Content Management ist zu wichtig für das Unternehmen und zu kostenintensiv, als dass man das mögliche gute Ergebnis mangels durchdachter organisatorischer Vorarbeit gefährden sollte."[1]
Überhaupt ist es schwierig die wichtigsten relevanten organisatorischen Aspekte zu finden, die durch die Einführung eines CMS entstehen können. Hierbei sollte zuerst eine Analyse stehen, in wie weit Struktur, Kultur und Zustand der Organisation die weitreichenden Veränderungen überhaupt erlauben. Nach der Erfüllung dieser Voraussetzungen geht es daran, die unternehmensindividuellen organisatorischen Fragen zu beantworten. Dabei sollte man sich Klarheit darüber schaffen, wo die Rahmenbedingungen für die Organisation liegen und diese dementsprechend positionieren. Abhängig ist das ganze vom Aufbau des Unternehmens, der Unternehmens-, Kommunikations- und Führungskultur.

Finanzierung und Controlling

Zwei Besonderheiten bei der finanzierungsorientierten Betrachtung eines Content Management-Vorhaben sollt man beachten:

a) die differenzierte Erfassung aller gegebenenfalls auftretenden Kostenarten, speziell der oft vorkommenden nicht offensichtlichen indirekten Kosten
b) die fundierte Analyse möglicher Einspareffekte und Erlöspotenziale, vor allem unter Berücksichtigung der indirekten Effekte

1 (Koop, H.; Jäckel, K.; Offern, A.: Erfolgsfaktor Content Management, 1. Auflage, Braunschweig/Wiesbaden, S.116-117)

Aufwendig ist die finanzielle Analyse von CM-Projekten weder wegen den anzuwendenden Methoden, noch wegen den Verfahren. Sondern vor allem wegen der Vielzahl möglicher Kosten- und Erlösfaktoren, die in direktem oder indirektem Zusammenhang mit Content Management stehen. Dies ist ein Charakteristikum, dass für alle Projekte im Bereich Informationsbearbeitung festzustellen ist.
Besondere Aufmerksamkeit gilt somit der vollständigen Erfassung aller Faktoren im Rahmen der Budget- und Finanzierungsplanung. Da dies ein komplexes Vorhaben ist, sollte man möglichen Risiken, die mit der Einführung eines eingebundenen Content Managements/Content Management-Systems verbunden sind, erhöhtes Interesse widmen.

Content Management - Nomenklatur

Asset Management ist das Management von Assets

- Asset Management ist der Überbegriff für alle Aktionen und Funktionalitäten eines Content Management (-Systems) für den Umgang, das Verwalten sowie das Management echter Assets, also werthaltiger, vermarktbarer Contents.

Staging
Wenn ein Content freigegeben ist, soll er (be-) nutzbar sein, also Menschen zur Verfügung stehen.
Staging ist die Veröffentlichung

W3C
Standards sind wichtig, damit sich Programme und Systeme verstehen und miteinander kommunizieren können. Diese Standards werden genau definiert. Für CMS gibt viele relevante Standards und Empfehlungen des W3C (World Wide Web Konsortium), einer Non-Profit-Organisation

Funktionalität

Funktionalität klärt, welche Operationen auf und mit dem Content ausgeübt werden sollen und welche sonstigen Funktionalitäten die Mitarbeiter und späteren Benutzer benötigen werden, um effizient und effektiv mit dem System arbeiten zu können.
z.B.:
Suchmöglichkeiten und Retrievalfunktionalität (=Verfahren zum Wiederfinden elektronisch gespeicherter Daten)

Zugriff auf die Contents

- Wie viele Nutzer dürfen gleichzeitig mit dem System arbeiten und welche Antwortzeiten werden erwartet?

Sperren von Content zur Bearbeitung

(z.B. „Check-in / Check-out"-Mechanismus)

Metadaten-Funktionalität

Metadaten beschreiben Datenstrukturen. D.h. Attribute und Zusammenhänge eines je-

den Contents werden verwaltet
bsp: Metadaten eines Bildes:
- Untertitel, Stichwörter, Auflösung, Größe...

Metadaten sind i.d.R. in Content Management-Systemen fest „verdrahtet" und können vom Unternehmen gar nicht oder nur mit Tricks geändert oder erweitert werden. Dann sollte man aber Anforderungen die an die Metadaten gestellt werden genau spezifizieren.

Template-Funktionalitäten/Vorlagen/Schablonen

Templates sind Vorlagen. In Bezug auf Content Management sind dies Form- und/oder Strukturvorlagen zur Füllung mit Inhalten. Sie sind also strukturierte Platzhalter für Inhalte.

Archivierung:

Es hat nur Sinn, wenn es gebraucht wird. Ein Content Management-System, das Content archiviert und diese Contents gleichzeitig aus der Suchwelt herausnimmt, verfehlt seinen Zweck.
Durch die Archivierung darf auf keinen Fall der Betrieb aufgehalten werden.

Konfigurations-Management

Hierzu zählen frei definierbare Bildschirmumgebungen, damit jeder Benutzer seine individuelle und optimale Arbeitsfläche konfigurieren kann.

Customer Relationship Management-Funktionalitäten

Durch das CMS werden Kundendaten zugreifbar bzw. integriert und können somit als Basis für CMR genutzt werden.

3.4 Asset Management

Über die Zielsetzung des Content Managements hinaus soll das Asset Management gleichzeitig die Werthaltigkeit der verwaltenden Contents erfassen und gewährleisten.

Es verfügt darüber hinaus noch über die Funktionalität und Bewertung von Contents (= preisliche Kennzeichnung) und der Möglichkeit zur Erfassung und Abrechnung von Content-Transfers.

Um Asset Management zu gewährleisten, muss ein Content Management-System um entsprechende Tools und Funktionalitäten zur Content-Bewertung, zur Erfassung von Content-Transfers und zur Generierung entsprechender verifizierbarer Transaktionswerte verfügen und diese in Form geeigneter Schnittstellen zur Abrechnung zur Verfügung stellen.

3.5 Knowledgemanagement/Wissensmanagement

Knowledge Management ist die größte Herausforderung in einem Unternehmen. Man versucht, aus einmal erfassten Informationen Wissen zu generieren. Knowledge Management besteht nur 20% aus Technologie und der Rest findet in den Köpfen von Menschen statt. Die Vielgestaltigkeit der Knowledge-Management-Lösungen zeigt, dass es beim Versuch, Wissen aus strukturierten Informationen zu gewinnen und zu aggregieren, die verschiedensten Ansätze gibt. Dabei geht es weniger um den Einsatz einer Technik, sondern darum das Bewusstsein Einzelner oder von Gruppen für diesen Prozess zu schärfen.

Um „Wissen managen" zu können, bedarf es einiger nicht ganz unwichtiger Voraussetzungen, die prinzipiell mit IT-Systemen nichts zu tun haben und durch diese heute gar nicht oder nur schlecht übernommen werden können:

- Input: Grundvoraussetzung ist, dass die Mitarbeiter bereit sein müssen, eigenes Wissen in Content zu repräsentieren. Das setzt mehr voraus als das Vorhandensein eines Content Management-Systems, nämlich mindestens eine entsprechende Kooperationskultur im Unternehmen

- Strukturierung: Diese „Wissen-Contents" müssen von „Wissenden" so strukturiert werden, dass man sie wieder finden kann. Dies ist zwar in einem Content Management-System darstellbar, aber die Strukturierung an sich wird immer von den wissenden Menschen vorgenommen

- Output: „Die Menschen, die „Wissen-Contents" nutzen wollen, müssen nicht nur „richtig" suchen können. Viel schlimmer: Sie müssen verstehen können, was sie finden!"[1]

„Content Management-System ist also nur ein kleines Hilfsmittel, ein Werkzeug oder Rädchen im Rahmen eines umfassenden Knowledge Managements."[2]

1-2 (Koop, H.-J.;Jäckel, K.K.; Van Offern, K.K.: Erfolgsfaktor Content Management - Vom Web Content bis zum Knowledge Management, 1. Auflage, Wiesbaden, Vieweg, 2001, S.52);

4. Architektur eines CMS

4.1 Anatomie der Dokumente

Content Management Systeme wurden aus bereits bestehenden verschiedenen Anwendungen entwickelt. So lassen sich noch heute ihre Urheber aus Branchen- oder Technologiefokus erkennen, diese sind z.B. Redaktions- oder Datenbankmanagementsysteme. Des weiteren ist der Begriff „Content" bei einem Content Management System von großer Bedeutung. Die Summe von wesentlichen Einzelinformationen, d.h. Struktur, Darstellung und Inhalt, wird als „Content" eines Content Management System bezeichnet. Daraus lässt sich schließen, dass die Informationen nicht nur für das Aussehen bestimmt sind, sondern sie werden auch für die automatisierte Weiterverarbeitung in Datenbanken und für die Weiterverwendung auf anderen Websites eingesetzt. Dies führt zu einer Zerlegung, mit separater Datenhaltung in drei wesentliche Bestandteile: Struktur, Darstellung und Inhalt. Diese Dreiteilung ist in jedem Dokument enthalten. Hierbei wird auch von der „Anatomie der Dokumente" gesprochen.

Struktur:

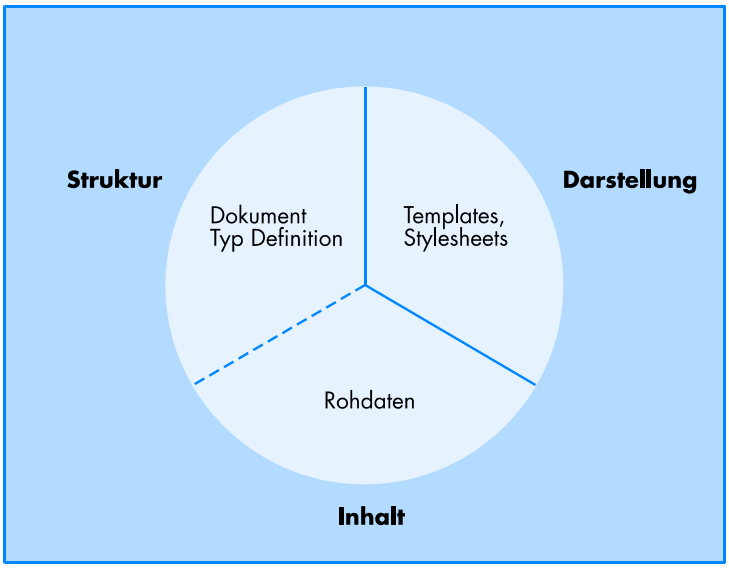
Die Struktur ist die Definition des Inhalts von jeder Einzelinformation samt ihrer Abfolge bzw. Verschachtelung, z.B. Presseinformation (Titel, Detailinformation).

Darstellung:

Die Darstellung dient als formale Beschreibung zur Präsentation auf einem Ausgabemedium. Dazu werden Stylesheets verwendet, in ihnen regelt man die Formatierung und Positionierung des Inhalts. Dies wird für verschiedene Ausgabemedien angewandt, der Inhalt bleibt immer gleich, lediglich durch Stylesheets der Ausgabe angepasst.

Inhalt:

Beim Inhalt wird die Definition der Struktur in Datenelemente abgebildet. Damit die Metainformationen erhalten bleiben, ist die Datenhaltung des Inhalts von sehr großer Wichtigkeit.

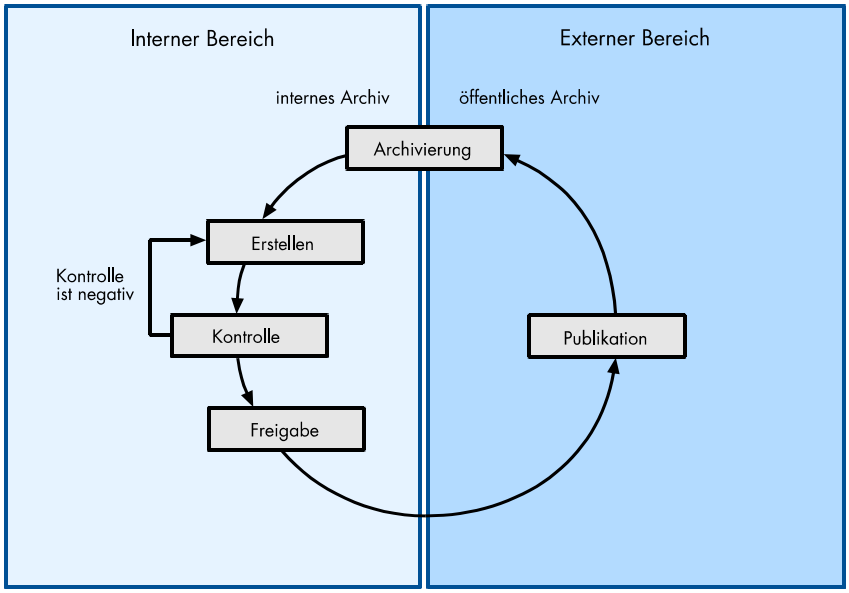


Durch die Aufteilung in drei Teile entstehen sämtliche Vorteile:

- Trennung zwischen inhaltlichen und formalen Daten
- Darstellung der Informationen auf verschiedenen Ausgabemedien
- Benutzer- und bedarfsgerechte Aufbereitung der Informationen
- Geringere Durchlaufzeiten gewährleisten eine höhere Aktualität

4.2 Content Lifecycle

Der Content Lifecycle ist einerseits ein theoretisches Modell zum besseren Verständnis von Webpublishing. Andererseits ist er ein Hilfsmittel, um einen effizienten und effektiven Produktionsprozess zu gewährleisten, sowie die Ermittlung der Anforderungen an ein Content Management System. Jeder Inhalt einer Website durchläuft während seines Lebenszyklus mehrere verschiedene Abschnitte, die im Content Lifecycle enthalten sind. Bevor er auf einer Website zu sehen ist, durchläuft er all die Stationen, die im Konzept des Content Lifecycles enthalten sind. So werden Inhalte systematisch produziert und publiziert.
Im ersten Lebensabschnitt wird ein neuer Inhalt erstellt, dies kann z.B. Text, Bild, Grafik oder HTML sein. Nach Erstellung des Inhaltes wird er einer höheren Instanz zur Kontrolle weitergegeben. Bei der Kontrolle wird der Inhalt auf seine Qualität geprüft. Besteht der Inhalt die Qualitätskontrolle nicht, d.h. bei negativer Kontrolle, so muss er wieder zur Erstellung zurück und nochmals verändert werden. Ist die Qualitätskontrolle positiv, dann wird der Inhalt über die Freigabe zur Publikation veröffentlicht. Wird der Inhalt nicht mehr benötigt, kommt er zur Archivierung. Dabei wird zwischen internem und öffentlichem Archiv unterschieden. Soll der Inhalt für interne Zwecke archiviert werden, z.B. für eine Dokumentation oder zur Wiederverwendung, dann kommt er zum internen Archiv. Ins öffentliche Archiv kommt er, wenn er z.B. zum recherchieren für Nutzer der Website dienen soll. Des weiteren haben viele Betreiber von Websites das Problem ihren Inhalt in mehreren Sprachen veröffentlichen zu müssen. Eine wesentliche Aufgabe eines CMS ist es, den Übersetzungsprozess zu verwalten und die Verbindung zu den Ausgangsdokumenten stets aktuell zu halten. Das folgende Schaubild zeigt, das der Content Lifecycle der Lebenszyklus der Inhaltselemente im definierten Workflow ist .



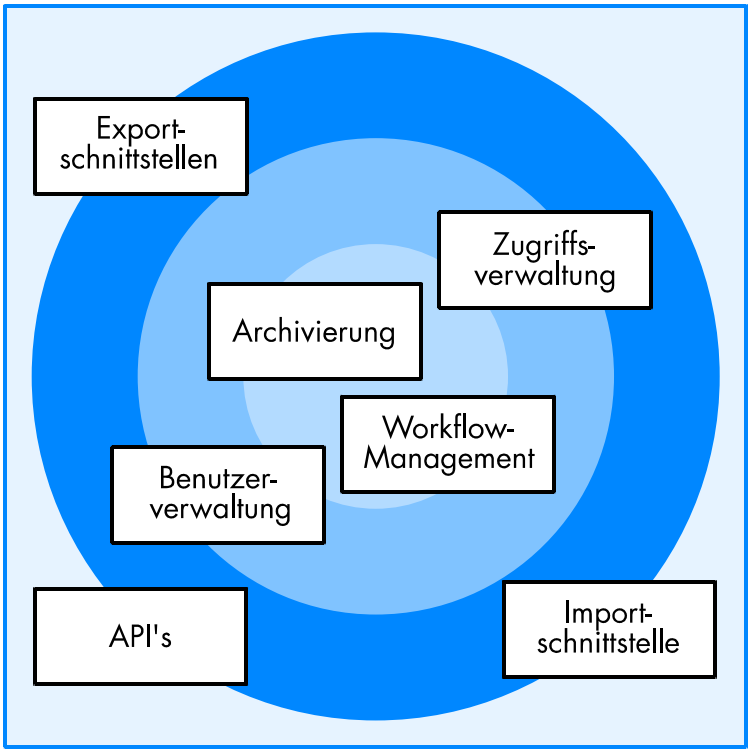
Geschütztes Editieren

Grundsätzlich kann nicht ausgeschlossen werden, dass ein gleichzeitiger Zugriff auf ein und denselben Content passiert. Dadurch könnte es im schlimmsten Fall zu Datenverlusten kommen. Daher ist das sogenannte geschützte Editieren eine wichtige Grundlage eines CMS. Dies bedeutet, so lange ein Content von einer Person bearbeitet wird, ist der Inhalt für den editierenden Zugriff anderer gesperrt. Dies wird in einem Content Management System als Check-in und Check-out bezeichnet. Öffnet eine Person zur Bearbeitung einen Content, so erfolgt der Check-out und ist zum Editieren für andere nicht mehr verfügbar. Wird der Content wieder geschlossen, dann erfolgt der Check-in und steht für andere wieder zur Verfügung.

4.3 Kernkomponenten

Jedes Content Management System unterscheidet sich von anderen, teilweise im Aufbau oder in der Art der Anwendung. Trotzdem lassen sich aber alle CMS durch fünf Kernkomponenten beschreiben, die aufeinander aufbauen:

Zentrale Bestandteile sind:



1. Archivierung - „Content Repository“

„Content-Repository“ spiegelt die Speicherung und Ablage des Content in einem CMS wieder. Hier werden alle Assets zentral gespeichert, die zum Beispiel auf einer Website benutzt werden. Natürlich gibt es noch andere Formen zur Speicherung von Inhalten wie z.B. die Ablage in Dateien und Verzeichnissen, XML - Dateien, objektorientierte Datenbanken oder Mischformen. Der Vorteil von Content Repository liegt aber darin, dass durch die zentrale Speicherung der Inhalte ein konsistenter Datenstamm entsteht, auf den allen Mitarbeiter, je nach Zugriffsrecht, zugreifen können. Durch diese Art von Speicherung wird die Datensicherung und Administration des Systems erheblich erleichtert. Des weiteren ermöglicht sie, dass der dezentral organisierte Workflow trotzdem den Zugriff auf den zentralen Datenbestand von überall aus hat. Zuvor müssen jedoch auf der Datenbank verschiedene Strukturen, aufgrund der Trennung von Inhalt und Layout, definiert werden. Das bedeutet, dass man für jede Content-Art eine leere Hülle in der Datenbank erstellen muss. Diese Vorarbeit ist meist etwas mühsam und zeitaufwendig, was sich aber mit der Zeit, d.h. mit der Nutzung des Systems erheblich bezahlt macht, da das Datenbankdesign dann für längere Zeit fest steht. Dies ermöglicht, dass Daten nicht als in sich geschlossene Dateien gespeichert, sonder in ihre Einzelkomponenten nur referentiell zueinander gespeichert und verwaltet werden. Das bedeutet, dass man auch einzeln Zugriff zu den Komponenten hat und sie an unterschiedlichen Stellen, zum Beispiel in einer Website einbauen kann. „Der Zwang zur Strukturierung der Inhalte wirkt sich an dieser Stelle positiv aus. Warum sollten Informationen wie Adressdaten in einer unstrukturierten Datei gespeichert werden?“[1] Durch die Verwendung einer Datenbank kann man Informationen mehrfach nutzen und einfacher nach gewünschten Daten suchen. Der Zugriff auf die Datenbank kann nur über bestimmte Tools erfolgen, die das CMS mitliefert (z.B. eigene Applikationen oder Formulare im Webbrowser).
Zur Archivierung gehört auch die Versionsverwaltung und - kontrolle, sowie die Einbindung von externen oder internen Editoren, um sowohl Inhalte als auch Stylesheets zu erstellen.

Versionierung

Sie ist nicht zu verwechseln mit der Protokollierung aller Aktivitäten im System (vgl. Workflowmanagement). Sie dokumentiert die Änderungen an Inhalten und speichert diese als einzelne Version ab. Die Anzahl der Versionen sind im CM-System einstellbar. Dadurch lässt sich immer wieder eine Rekonstruktion durchführen man spricht in diesen Fällen auch von „Roll-Back„.

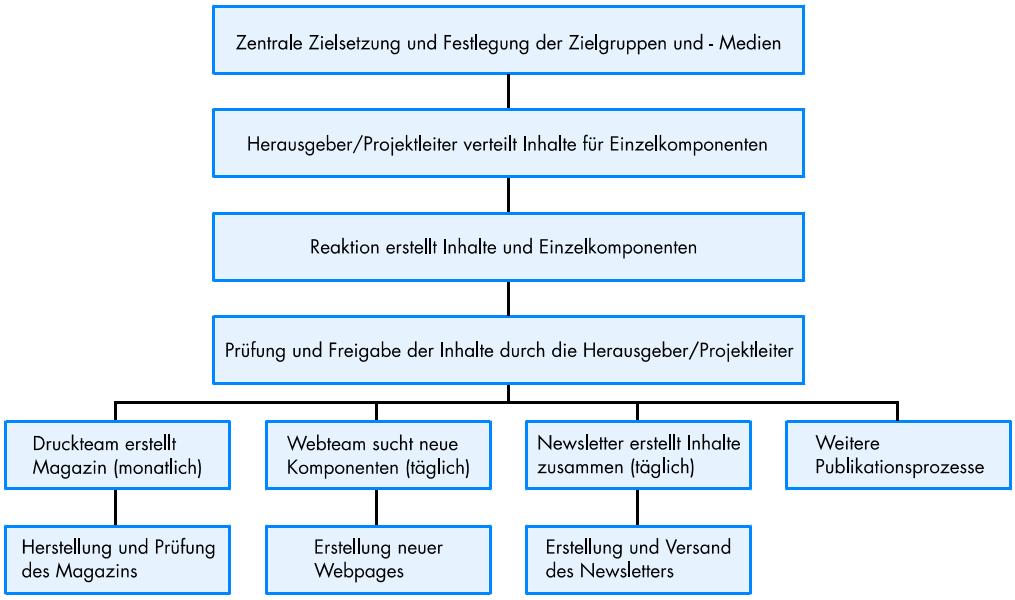
4. Workflowmanagement

Die Trennung von Inhalt und Layout bringt ebenfalls große Vorteile für die Gestaltung des Workflows mit sich. Auf die im Content Repository gespeicherten separierten Assets kann dadurch einzeln über den Workflow zugegriffen werden. Die Workflowkomponente definiert, automatisiert und überwacht die Arbeitschritte im Publishingprozess. Sie gibt Rückmeldung an die Mitarbeiter und ermöglicht dadurch erst die dezentrale und aufgabenbezogene Arbeit an z.B. einer Website. Über den Freigabezyklus (vgl. Content Lifecycle) wird die inhaltliche Qualität gesichert.

1 (Zschau, O.; Traub, D.; Zahradka, R.: Web Contnent Management - Websites professionel planen und bearbeiten, 2. Auflage, Bonn, Galileo Press GmbH, 2002, S.207)

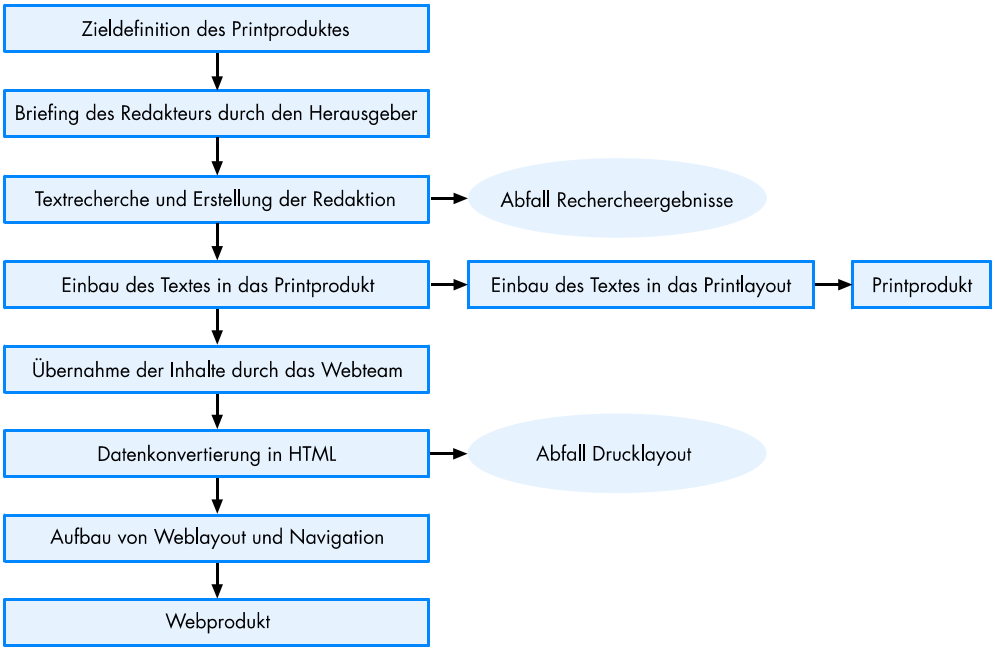
"Alter Workflow" - lineare Verwertung von Inhalten

In den Frühzeiten des Internets erstellte eine Redaktion lediglich Inhalte für das bestellte Medium, die dann später für das Internet angepasst wurden oder schlimmsten Falls 1:1 ins Web umgesetzt wurden.



"Paralleler Workflow" - eine Wertschöpfungskette

Die Redakteure gehen gemeinsam mit Programmierern und Herausgebern vor und erstellen individuelle Texte für zielgruppennahe Medien, die im CMS miteinander verknüpft und erfolgreich eingesetzt werden können. Man spricht hier auch von "Cross Media Publishing".



Zu den wichtigsten Workflow-Tools zählen :

1.) Benachrichtigungen
2.) Statusanzeigen und Filter
3.) Aufgabenverteilung und -listen
4.) Protokollierung
5.) Preview
6.) Vertreterregeln

Benachrichtigungen

- über E - Mail oder Systeminterne Anzeigen

Statusanzeigen und Filter

Statusanzeigen:
- sie überprüfen Zustände (z.B. wann die letzte Aktualisierung eines Objektes erfolgte)
- können auch benutzt werden wenn keine Änderungen/Aktionen statt gefunden haben

Filter:
- sie erlauben zusätzlich komplexe Abfragen über das Content Repository und seine Meta - Informationen (Nutzer, Zeitraum, Erstellungdatum, Änderungsdatum)

Aufgabenverteilung und -listen
- sie sollten im CMS erfasst sein und direkt mit den Objekten verknüpft werden
  Vorteil :- ein Redakteur kann somit feststellen was er als nächstes zu tun hat und erledigte Aufgaben abhaken

Protokollierung

- es gibt zwei Arten der Protokollierung:
  a) die objektbezogene, sie erfasst alle Aktivitäten bezogen auf ein bestimmtes Objekt (z.B. Bearbeitung, Löschen, ...) ebenso den Zeitpunkt der Aktivität und den Nutzer
  b) die administrative, sie speichert die Systemzugriffe

Preview

- bietet dem Freigebenden und dem Autor eine Vorschau, um neue oder veränderte Inhalte zusammen mit den bestehenden darzustellen
- Vereinfacht die Qualitätssicherung

Vertreterregeln

- ermöglichen das lückenlose Funktionieren des Publishingprozesses, da jedem Mitarbeiter bei einem Ausfall (z.B. Krankheit, Urlaub) ein Vertreter zugeordnet ist

3. Benutzer- und Zugriffsverwaltung

Unter Benutzerverwaltung versteht man die Festlegung von Gruppen, Rollen und deren Zugriffsrechte. Somit erhalten einzelne Benutzer, Gruppen und sogar ganze Abteilungen Zugangs- und Bearbeitungsrechte in einem CMS. Der Zuständigkeitsbereich wird über die Gruppe zugeordnet, die Aufgaben und Rechte der Gruppe über die Rollen. Zum Beispiel wird der Benutzer Müller der Gruppe Grafik zugeordnet und der Rolle Abteilungsleiter. Diese festgelegten Benutzerdaten werden dann in einer Benutzerdatenverwaltung gespeichert.

Wieso aber ist die Benutzer- und Zugriffsverwaltung so wichtig?
Ein CMS ist die Schnittstelle eines Unternehmens für Informationen die für die Allgemeinheit (z.B. dem Internet) oder einzelnen Nutzergruppen (Intranet und Extranet) bestimmt sind. Die Datenbank auch „Content Repository" genannt verwaltet eine Vielzahl von Informationen und Daten, auch solche über die nicht jeder verfügen sollte (z.B. Informationen über Mitarbeiter). Ein weiterer Faktor ist, dass CM-Systeme Multiuserfähig sind, d.h. mehrere Benutzer können gleichzeitig und unabhängig voneinander darauf zugreifen. Die Folge ohne Benutzer- und Zugriffsverwaltung wäre, dass es zu Manipulation, unautorisiertem Lesen oder dem Bearbeiten von Informationen bishin zum Systemzusammenbruch kommen würde. Die Benutzer- und Zugriffsverwaltung bilden daher ein Sicherheitskonzept welches das CMS nach außen hin absichert und intern nur die nötigen Zugriffe erlaubt. Zum anderen wird durch die Benutzerverwaltung der Workflow klar definiert und die anfallenden Aufgaben direkt an die verantwortlichen Gruppen bzw. Mitarbeiter verteilt. Denn Prozesse lassen sich „... nur effektiv gestalten, wenn der Beteiligte einen abgesteckten Aufgabenbereich und entsprechende Kompetenzen hat."[1] Somit kommt es auch zu einer erheblichen Entlastung des Administrators, da er seine Rechte, natürlich nicht alle (z.B. Rechte zum Bearbeiten des Systems), an die verschiedenen Gruppen und Rollen abgibt bzw. verteilt.

Die Zugriffsverwaltung steuert, wie der Name es schon sagt die Zugriffe auf das CMS. Die Clients kommunizieren mit dem Server über das HTTP-Protokoll, die sicherste Variante ist aber das HTTPs-Protokoll. Das Login geschieht über eine bestimmte URL auf dem Webserver. Durch die Eingabe von Nutzerkennung und Passwort wird der Benutzer authentifiziert. „Aus der Authentifizierung lassen sich die Berechtigungen des Nutzers im System ableiten"[2], d.h. das CMS schaut in seine Benutzerdatenbank und überprüft welche Rechte der Nutzer hat. Bei WCMS gibt es zwei Ansätze, die sich auf die Sicht des Benutzer auf die Funktionen und Inhalte im System beziehen:

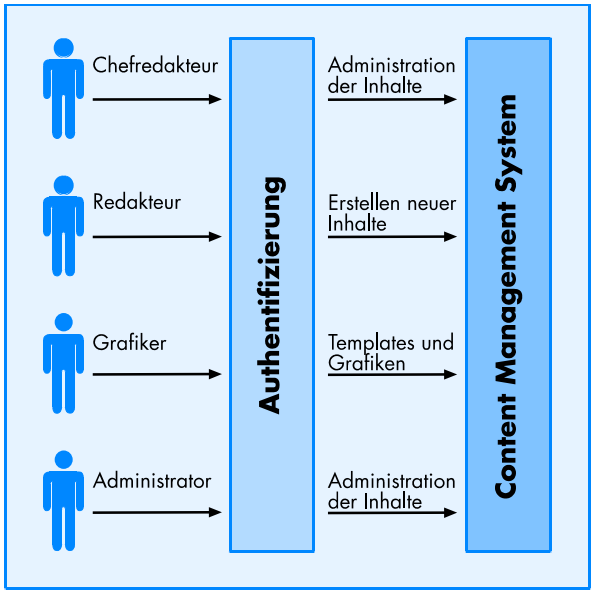
a) die Zugriffsrechte werden bei Aktionen geprüft (z.B. beim Klick auf Ordner, ...)

b) durch die speziellen Zugriffsrechte jedes einzelnen Benutzers werden ihm nur die Funktionen und Inhalte zur Verfügung gestellt, auf die er auch ein Recht hat (= sicherste Variante)

1-2 (http://www.community-of-knowledge.de/cp-artikel_d.htm)

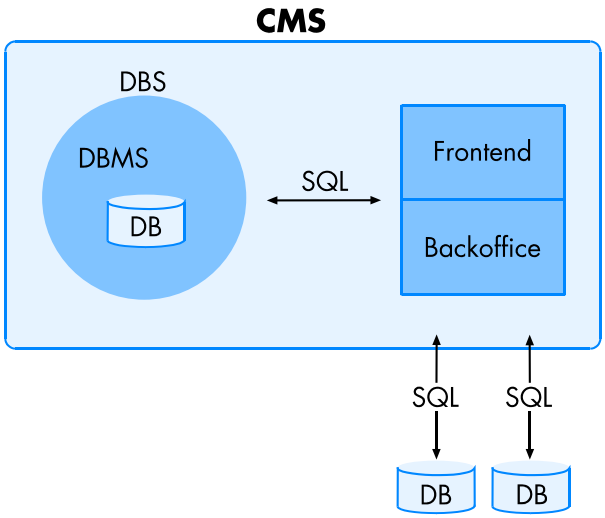
Vorteil:

Der Benutzer erhält einen übersichtlicheren speziell für ihn eingestellten Client, dadurch wird der Umgang mit dem System extrem vereinfacht und der Nutzer nicht unnötig mit anderen Funktionen verwirrt.



4. Im - und Exportschnittstellen

„Diese Funktion sorgt dafür, dass Inhalte (z.B. Datenbanken, Websites, ...) integriert und die Daten nach Ablauf Ihrer Lebensdauer archiviert werden, indem sie etwa als XML- oder Word-Dokumente formatiert werden".[1] Die Zugriffe auf andere Datenbanken erfolgen mittels spezieller Datenbankschnittstellen (z.B. Sybase, Oracle, MySQL, ...), die sich je nach Anwendungsgebiet teilweise unterscheiden.
Es gibt aber auch noch die ODBC- und JDBC-Schnittstellen, die von vielen Datenbankherstellern angeboten werden, über die man herstellerunabhängig die Datenbank per SQL abfragen kann. Somit kann man mit fast jeder Datenbank kommunizieren, die diese Standards benutzt.



1 (Zschau, O.; Traub, D.; Zahradka, R.: Web Contnent Management - Websites professionel planen und bearbeiten, 2. Auflage, Bonn, Galileo Press GmbH, 2002, S237/ff)

5. APIs

APIs („Application Programming Interface") sind offengelegte Programmschnittstellen zur funktionalen Erweiterung. Darüber hinaus stellen sie Standardfunktionen für die Verwaltung und Darstellung des Contents zu Verfügung.

Die Strukturverwaltung

Die Strukturverwaltung ist eine weitere Komponente von CM-Systemen, die aber nicht so stark gewichtet wird wie die anderen. Sie stellt bei aus unterschiedlichen Quellen zusammengesetzten Dokumenten die Gültigkeit und Vollständigkeit der herangezogenen Webseiten und benutzter Links sicher. Zudem „Content Syndikation" ohne eine leistungsfähige Strukturverwaltung nicht möglich wäre.

4.4 Serverkonzepte

In den vorangegangenen Kapiteln wurden vorrangig die verschiedenen Kernkomponenten eines CMS und deren Funktionsweise vorgestellt. Diese Funktionen, werden dem Anwender über den Client, ein Programm, das die Protokolle eines Servers versteht und mit diesem auch kommunizieren kann [z.B. Browser - Clients wie Netscape oder anwenderspezifische Clients (= eine plattformabhängige Applikation)], zur Verfügung gestellt. Der Server enthält also den funktionalen Kern des CMS, damit Mitarbeiter überhaupt am Client des Systems arbeiten können. Er beinhaltet das Content Repository, verwaltet den Workflow, das Rechtekonzept und bietet Im- und Exportschnittstellen nach außen. Das bedeutet, dass alles was vom Client aus gestartet wird funktional über den Server abläuft.
Technologisch basieren also alle CM-Systeme auf dem „Client-Server-Modell". Die Zentrale Aufgabe des Servers ist demzufolge die Bereitstellung von Information im Web. Dabei entstehen theoretisch zweierlei Lasten auf dem Server:
- interne Zugriffe von Seiten der Mitarbeiter zur Pflege der Inhalte
- externe Zugriffe von Webbenutzern

Aus Sicherheits- und Performancegründen haben sich verschiedene Technologien entwickelt, die diese beiden Zugriffsarten mehr oder weniger trennen:
- dynamisches Publishing
- Publishing-/Staging-Server
- Publishing-/QA/Staging-Server

Bei der Auswahl des Serverkonzeptes ist projektbezogen zu überlegen, ob es den Anforderungen gewachsen ist. Vor allem aber sollte man bedenken, dass der Workflow stark von der verwendeten Technologie abhängig ist. Ein weiterer Aspekt den man bedenken sollte ist die Skalierbarkeit. Ein CMS ist umso besser skaliert, d.h. an wachsende Ansprüche anpassbar, je mehr Komponenten über mehrere Server verteilt sind. Der Vorteil dieser Aufteilung liegt darin, dass beim Ausfall einer Komponente wenigstens die restlichen noch betriebsbereit sind.

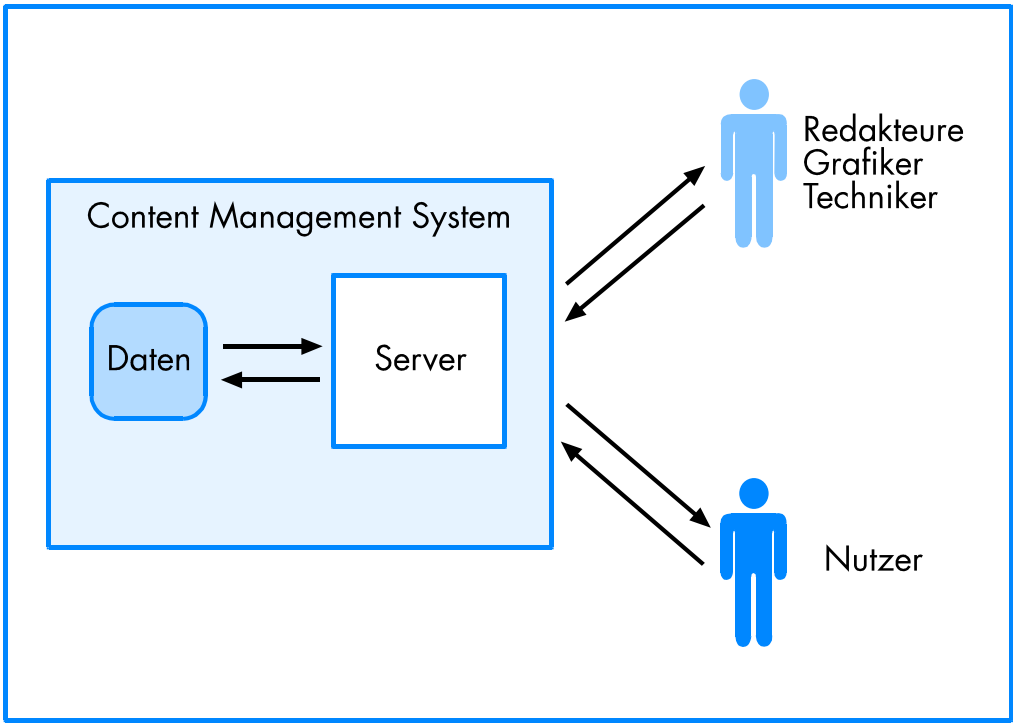
Dynamisches Publishing

Bei diesem Serverkonzept erfolgt die Redaktionsarbeit und Zugriffe von Webnutzern zentral über ein und denselben Server. Dynamisches Publishing bedeutet, dass wenn z.B. ein User einen Link aktiviert, automatisch ein Prozess gestartet wird der das gewünschte Objekt hochlädt und es mittels einer Template-Vorlage formatiert und online stellt. Erst durch die Verwendung von Caching werden lange Ladezeiten verhindert. Dieses Konzept ist meist bei datenbankbasierenden CM-Systemen vorzufinden.

Caching:
- speichert die am häufigsten geladenen Seiten in einem Zwischenspeicher ab, damit sie nicht immer wieder neu generiert werden müssen

Vorteile:
- besonders für die dynamische Integration von Informationen aus mehreren Systemen in eine Website geeignet

Nachteile :
- bei einem Serverausfall kann keiner der beiden Seiten mehr darauf zugreifen



Publishing-/Staging-Server

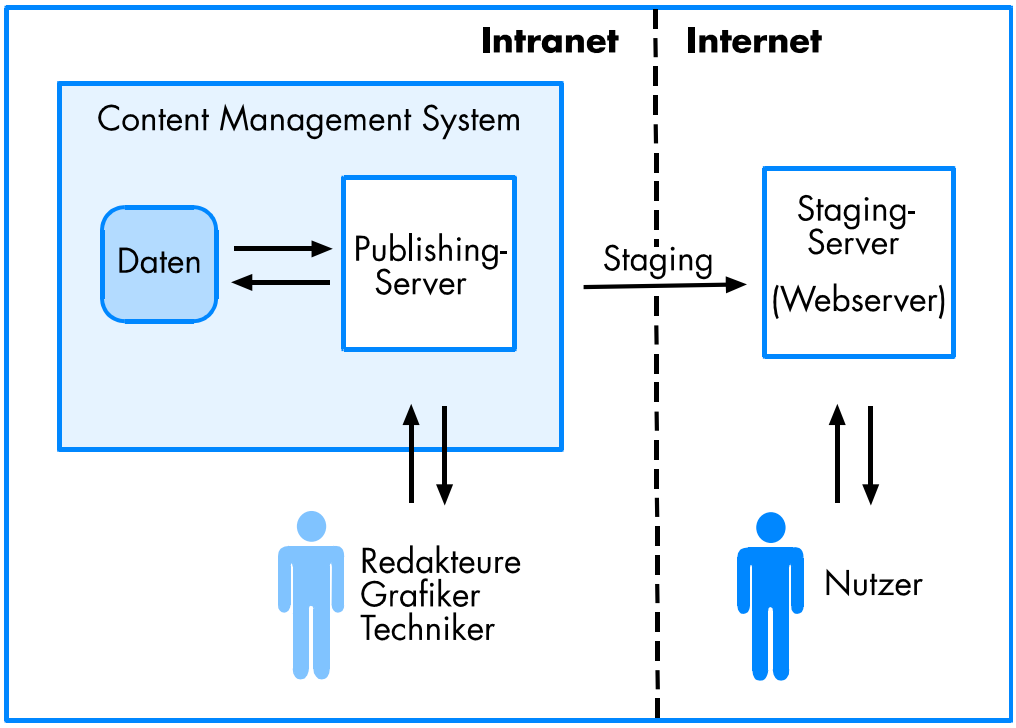
Der Publishing-/Staging-Server ist ein erweitertes Serverkonzept, das zwei separate Umgebungen schafft. Einmal durch den internen Publishing-Server (= Redaktionsserver) für die Redaktionsarbeit und andererseits durch den Staging-Server (= Webserver) für die User.

Publishing-Server:
- beinhaltet alle Funktionalitäten eines CMS und die Content Repository
- hier wird der Content erstellt, verändert und verwaltet

Staging-Server :
- ist ein ganz normaler Webserver der die statischen Seiten verwaltet, die entweder manuell oder zeitgesteuert vom Publishing-Server auf ihn übertragen werden

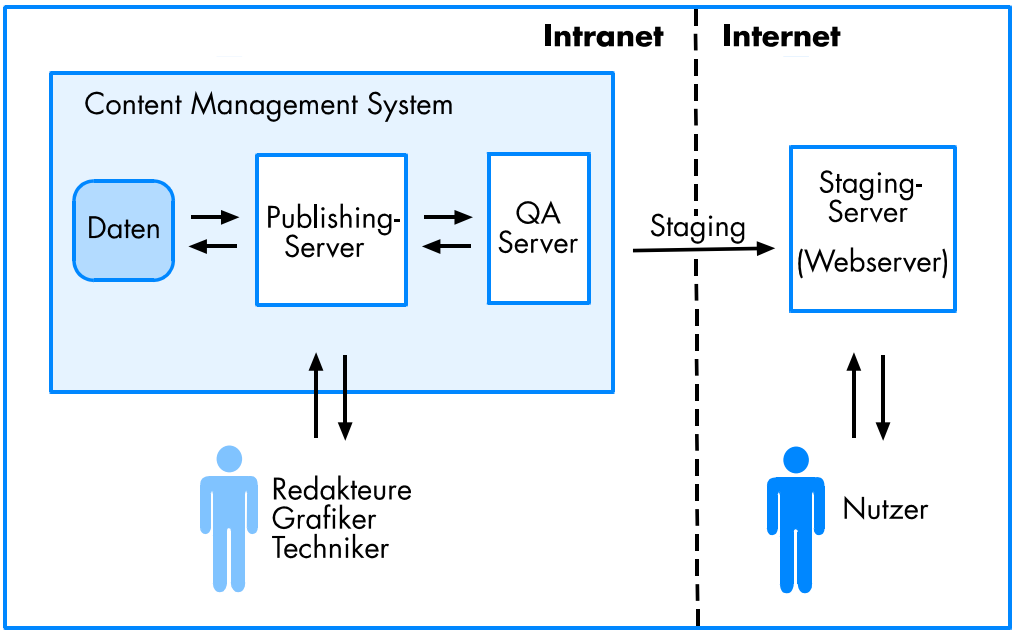
Vorteile :
- die getrennte Einrichtung beider Server ermöglicht so eine bessere Sicherung des Systems und der Betriebssicherheit, z.B. kann jeweils ein Server vor und einer hinter einer Firewall positioniert werden
- bei Ausfall eines Servers kann weiterhin gearbeitet werden
- gute Performance
- es ermöglicht die Verwendung eines beliebig filebasierten Webservers (z.B. Apache)
- die Webseiten werden hier nicht wie beim Dynamischen Publishing ständig neu zusammengebaut, sondern aktuell aber statisch auf dem Staging-Server abgelegt.

Nachteile :
- ungeeignet für dynamische Informationen aus anderen Backend-Systemen



Publishing-/QA-/Staging-Server

Der Publishing-/QA-/Staging-Server ist eine Weiterentwicklung des Publishing-/Staging-Servers. Der Unterschied liegt lediglich darin, dass der Publishing-Server mit einem „Quality-Assurance"-Server erweitert wurde, der nur für die Qualitätssicherung zuständig ist, eine zusätzliche Workflow-Verteilung also. Er beinhaltet die Kontrolle und Freigabe von Inhalten. Somit werden neu erstellte Inhalte vom Publishing Server auf den QA-Server, zur Kontrolle und Freigabe übertragen. Erst mit der Freigabeerlaubnis erfolgt die Übertragung auf den Staging-Server.



Applikationsserver

Netzwerkbasierte Applikationen basieren auf Applikationsservern. Sie weisen speziell bei der Integration in ein System entscheidende Vorteile auf :
- ein Applikationsserver kann Businesslogik abbilden
- sie sind besonders gut geeignet für in bestehende IT-Landschaften integriert zu werden

Wichtige Eigenschaften für Webanwendungen sind:

- Enterprise Application Integration
- Sessionsmanagement
- Offene Integrierung von Businesslogik
- Clustering
- Load-Balancing
- Transaktions-Management

Offene Schnittstellen wie, z.B. ...CORBA („Common Objekt Request Broker Architektur"), COM („Component Objekt Modell"), EJB („Enterprise Java Beans"), ... erlauben den Informationsaustausch zwischen verschiedenen Applikationen (Entwicklungsspra-

chen sind Java, C++). Gerade die Integrationsmöglichkeiten von Applikationsservern sind als Basis für ein WCMS interessant. Es nutzt vorhandene Interfaces und bietet zusätzlich selbst einheitliche Interfaces zu anderen Applikationen. Das Transaktions-Management kann über diese Schnittstellen realisiert werden, was aber kein zwingender Bestandteil eines WCMS ist. Speziell für die Aufrechterhaltung der Performance stellen Applikationsserver Load-Balancing und Clustering zur Verfügung, damit selbst bei einer Vielzahl von Zugriffen auf eine Website, kurze Antwortzeiten möglich sind. Jedoch setzen viele WCMS bestimmte Applikationsserver als zwingend voraus. Der Nachteil besteht in der zu starken Bindung an einen oder zwei Hersteller. Ebenso ist dann die Applikation vom Applikationsserver abhängig und umgekehrt. Das WCMS sollte daher lieber gegenüber Applikationsservern plattformunabhängig sein und auf den meisten Systemen laufen (wie z.B. Zope, vgl. Kapitel 7.2).

FAZIT :

Als Basis für CM sind Applikationsserver eine sehr gute Wahl, ebenso wenn eine Integration in Legacy-Systeme notwendig ist. Die Trennung von Geschäfts- und Programmierlogik ist eine weitere Voraussetzung für den Einsatz eines ganzheitlichen E-Commerce Konzepts im Unternehmen.

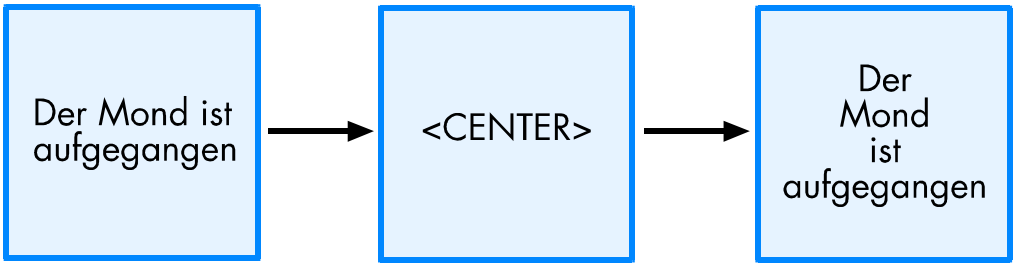
E-Commerce: mit E-Commerce meint man den elektronische Geschäftsverkehr im Web

5. Entwicklung von HTML zu XML

5.1 HTML

Im Rahmen des WWW-Forschungsprojektes am europäischen Hochenergieforschungszentrum CERN in Genf wurde 1990 HTML (Hypertext Markup Language) von dem Forscher Tim Berners Lee entwickelt. HTML ist eine Instanz von SGML (Standard Generalized Markup Language), d.h. HTML bekam von SGML die Grundstruktur vererbt und entwickelte dann weitere Strukturen. SGML beruhte allein auf der inhaltsbasierenden Formatierung von Texten, während HTML außer Texten auch Formeln, Zeichnungen und Grafiken in Texten übermittelt.
In der HTML-Sprache werden die einzelnen Bestandteile eines Dokuments beschrieben, z.B. ist es möglich Überschriften zu definieren, Tabellen festzulegen oder Absätzen eine bestimmte Form oder Farbe zu geben. Mit HTML besteht die Möglichkeit Websites aufzubauen. Der Vorteil von HTML besteht darin, dass es einen einfachen Aufbau sowie eine unkomplizierte Nutzung bietet. Für HTML wird lediglich ein Texteditor benötigt, der ASCII-Texte versteht, welcher auf jedem Betriebssystem standardmäßig vorzufinden ist. Der HTML-Code ist nicht nur auf das Internet beschränkt, daher ist es möglich HTML-Seiten auf dem lokalen PC zu speichern und aufzurufen. Viele Softwarehersteller bieten daher ihre Hilfedateien im HTML-Format an.
HTML ist keine Programmiersprache, sondern eine Seitenbeschreibungssprache für Dokumente. Dadurch lassen sich Informationen der Website eines Computers mit den Informationen anderer Computer verknüpfen, d.h. Computer werden miteinander über einfache Links vernetzt. Durch HTTP-Server werden die HTML-Dokumente den Benutzern weltweit zur Verfügung gestellt, mit Hilfe eines Browsers werden die Dokumente gelesen. HTTP steht für Hypertext Transfer Protocol. Es stellt die Verbindung zwischen HTML-Dokumenten und dem World Wide Web her. Der Aufbau in HTML besteht aus einem Head (Dateikopf), einem Title (Titel des Dokuments) und dem Body (Textkörper).

ASCII-Datei + Formatierungssprache = HTML-Dokument



Nachteile von HTML:

In HTML fehlen ausgeklügelte Formatierungsbefehle, d.h. es besteht aus vorgeschriebenen Tags und Attributen, die keine eigenen Befehle definieren können. Ein Tag beschreibt den Inhalt eines Dokuments. Dem Text können zwar Attribute wie z.B. Fettschrift zugewiesen werden, jedoch ein genaues Platzieren von Elementen auf dem Bildschirm ist nicht möglich. Soll eine Grafik genau über einen Text platziert werden, ist es aussichtslos dies in HTML hinzubekommen, außer es wird die HTML-Erweiterung CSS (Cascading Style Sheets) eingesetzt. Diese können den Text genau beeinflussen, d.h. Abstände regeln, Farben zuweisen, Rahmen ziehen sowie Linien einbauen. Durch CSS lassen sich

z.B. bestimmte Formatierungen angeben, damit der Text auf dem Bildschirm anders dargestellt wird als auf dem Monitor. Weiterhin ist es möglich, Texte mittels Sprachausgabe vorzulesen, aber nicht nur die Stimmlage, sondern auch die Position im Stereo- oder Suroundfeld, sowie die Stimmung des Vorlesers.
Aufgrund unterschiedlicher Browser führt dies bei HTML zu Problemen mit der Darstellung der Inhalte auf dem Bildschirm. Das W3C-Konsortium ist eine internationale Organisation die sich für eine erfolgreiche und standardisierte Entwicklung des Internets einsetzt. HTML unterliegt diesem W3C-Standard, d.h. hier wurde festgelegt, „... wie die HTML-Befehle aufgebaut sind, welche Parameter sie erwarten und welche Aktionen sie auslösen ...".[1] Trotz diesem Standard wird das Ergebnis einer HTML-Site in den verschiedenen Webbrowsern, wie z.B. Netscape oder Internet Explorer, in der Größe und Form unterschiedlich dargestellt. Das liegt daran, dass HTML keine detaillierte Layoutsprache ist, so variiert eben die Ausführung der Befehle.
Hinzu kommt, dass Informationen ihrer Formatierung fest zugeordnet sind. Sollten z.B. die Inhalte einer Website in einer anderen Form aufgebaut werden, so ist dies für den Programmierer sehr schwierig umzusetzen. Das heißt, es entsteht ein hoher Kosten- und Programmieraufwand. Durch diese Problematik wurde eine neue Sprache eingeführt, diese ist XML (Extensible Markup Language). XML wurde im Februar 1998 vom W3C-Konsortium ins Leben gerufen.

5.2 DHTML

Unter DHTML (Dynamic Hypertext Markup Language) wird die einfache Websitebeschreibung verstanden, die sich aus verschiedenen Programmiertechniken zusammensetzt. Es ist keine eigene Sprache, sondern besteht aus dem bisher bestehende HTML-System mit zusätzlichen neuen Techniken. Dadurch besteht DHTML aus dem bisherigen HTML, den Cascading Style Sheets und Javascript. Trotz all dem ist der Erfolg einer Website von ihrer Struktur und ihrem Design abhängig.

5.3 XML ergänzt HTML

Ein Buch wird meist in Titel, Kapitel, Unterkapitel aufgeteilt. Es enthält ein Inhalts- und Stichwortverzeichnis. In HTML ist eine solche Einteilung, oder auch Formatierung genannt, nicht möglich. Mit XML hingegen kann eine solche Formatierung umgesetzt werden. Ein wesentlicher Unterschied von XML zu HTML ist, dass XML dann eingesetzt wird, „... wenn Informationen unabhängig von der späteren Formatierung gespeichert und weiterverarbeitet werden sollen ...".[2]
XML wird für die Definition, Archivierung und den Austausch strukturierter Daten eingesetzt und ist eine festgelegte Methodik. Es ist Plattform- und Anwendungsunabhängig. Des weiteren werden Inhalt, Layout, und Struktur getrennt. Dies bedeutet, dass der reine Inhalt geschrieben und die Formatierung in einer Dokumentenstruktur (DTD) festgelegt wird. In einer DTD-Struktur werden die Informationen strukturiert, aber nicht an einen bestimmten Medientyp gebunden, so ist XML geradezu für ein Content Management System prädestiniert.

Text und Inhalt werden getrennt

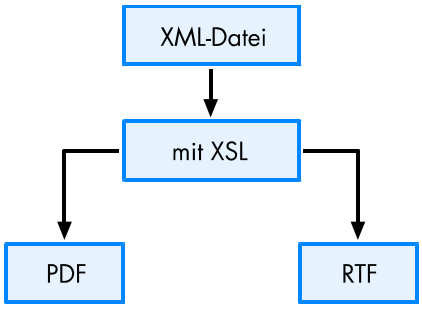
In der DTD (Document Type Definition) sind alle Anweisungen enthalten, die Vorgeben, wie welche Text- und Bestandteile eines Dokuments später angezeigt und vorhanden sein müssen. XML-Dateien, die die gleiche DTD verwenden, besitzen dadurch den gleichen

1 (Harms, F.; Koch, D.; Kürten, O.: Das große Buch, (X)HTML & XML, 1. Auflage, Düsseldorf, DATA BECKER GmbH & Co.KFG, 2000, S.19);
2 (Harms, F.; Koch, D.; Kürten, O.: Das große Buch, (X)HTML & XML, 1. Auflage, Düssel dorf, DATA BECKER GmbH & Co.KFG, 2000, S.20);

Aufbau. Der Vorteil ist das einheitliche Format, welches geradezu für den Datenaustausch geschaffen ist. Des weiteren werden die Attribute der einzelnen Befehle definiert und die inhaltliche Verschachtelung aufgezeigt.
Man unterscheidet zwischen einer internern und externern DTD. Während eine interne DTD am Beginn eines Dokumentes steht, befindet sich eine externe DTD in einer gesonderten Datei. DTD gibt es auch bei HTML, nur wird sich dort nicht sonderlich darum gekümmert.

XSL ist die Stilsprache zu XML

Da HTML zu unflexibel ist, wurde XML entwickelt, mit dem eigene Tags und Elemente für jede spezielle Anwendung kreiert werden können. Dadurch ist es flexibler bei der Webseitengestaltung als HTML. XML ist gegenüber HTML lediglich für das Strukturieren der Daten verantwortlich. Die Ausgabe erfolgt mittels Stylesheet. Dabei können CSS (Cascading Style Sheet) und XSL (Extensible Style Language) hinzukommen. Durch deren Verwendung besteht die Möglichkeit ein Dokument in verschiedenen Medien unterschiedlich darzustellen.
XSL ist vielseitiger als CSS, aber komplexer. XSL ist eine neu entwickelte Formatierungs- und Transformatierungssprache die aus XML resultiert und deren Vorteile benutzt.
Während HTML-Sites schnell aufzubauen sind, ist der Aufbau einer XML-Site mit einem höheren Zeitaufwand verbunden. Bei der HTML-Site stehen alle Informationen in einer Datei, bei XML aber müssen die Inhalte zuerst logisch strukturiert werden. Anschließend geschieht deren Ausgabe mit Unterstützung von XSL. XML achtet auf die genaue Schreibweise des Syntax, in HTML wird dies nicht so streng gehandhabt. Voraussetzung für XML ist, dass ein Betriebssystem sowie ein Browser vorhanden ist, und beide XML fähig sind.



XSLT:

XSLT wird zur Umwandlung von XML-Dateien mit einer bereits vorhandenen DTD, in eine neue DTD eingesetzt. Es ist geeignet für Business Applikationen, das die Datenkommunikation zwischen verschiedenen Systemen ermöglicht.
Mit XML können Daten in Datenbanken abgelegt und durch unterschiedliche Verfahren im Browser oder in anderen Anwendungen ausgegeben werden. Die Bezeichnung der Daten (Tags) wären in der Datenbank dieselben wie in den verwendeten Anwendungen. Die Datenstruktur bleibt dadurch eindeutig und übersichtlich, lediglich die Ausgabeprogramme wandeln diese je nach Bedarf um.

5.4 Aktueller Trend

Die sinnvollste Lösung wäre eine Kombination von HTML und XML. So entsteht aus den beiden Sprachen XHTML. Hier sind die Vorteile beider Sprachen miteinander verbunden und bieten dem Websiteprogrammierer größtmögliche Flexibilität, d.h. eine Website lässt sich zum einen so darstellen wie bisher und zum andern können DTDs genutzt werden. XHTML ist eine Übergangslösung mit Vorteilen gegenüber HTML. Dies wird solange andauern, bis sich XML als neuer Standard durchsetzt.

Der steigende Handel über das Internet (E-Commerce) fordert neue Formen des Informationsangebots sowie der Datenverarbeitung. Daraufhin formulierte das W3C-Konsortium den neuen Standard XHTML als Empfehlung.
Es soll hauptsächlich den Datenaustausch zwischen unterschiedlichen Anbietern ermöglichen. Erreicht wird dies durch die Trennung der Formatierung von den Inhalten. Mittelpunkt einer Website ist nun nicht mehr die Datendarstellung, sondern vielmehr ihre flexible Handhabung. Ein Unternehmen kann mithilfe von XHTML bestimmte Regeln entwerfen und festlegen. Diese geben vor, welche Daten in einem Dokument vorhanden sein müssen. Ganze Bestellsysteme werden so aufgebaut.
Das Internet dient somit als Frontend, dass dem Kunden die Daten anzeigt, gleichzeitig können die Datenstrukturen, die mittels DTDs definiert sind, an andere Anbieter weitergereicht und weiterverarbeitet werden.

6. Aktzeptanz

6.1 Einführung neuer Konzepte und Modelle

Führungskräften und Entscheidungsträgern empfinden neue Konzepte als überaus positiv, im Gegensatz zu den meisten Mitarbeitern. Denn einerseits werfen jene, neue Chancen und Möglichkeiten auf, allerdings birgt es auch Risiken und Restriktionen. Gerade im Hinblick auf ein komplexes Vorhaben wie die Realisierung eines durchgängigen Content Management macht es Sinn sich vorzeitig über alle Vor- und Nachteile klar zu werden. Man beachtet besonders folgende Faktoren:

a) Qualitative Faktoren

- Unternehmen (Geschichte, Visionen, ...)
- Technische Aspekte (Strategie, Organisation, ...)
- Externe Faktoren (Kunden, Prognosen, ...)

b)Quantitative Faktoren (Schwerpunkt)

=> Hieraus entwickelt man dann die erste, grundlegende Zielkonzeption!

Zusätzliche wichtige Grundfragen:

- mögliche Risiken durch die Realisierung
- mögliche Risiken für die Realisierung
- Bewertung von Aufwand und Erfolg
- Klaren Projektablauf definieren und kommunizieren

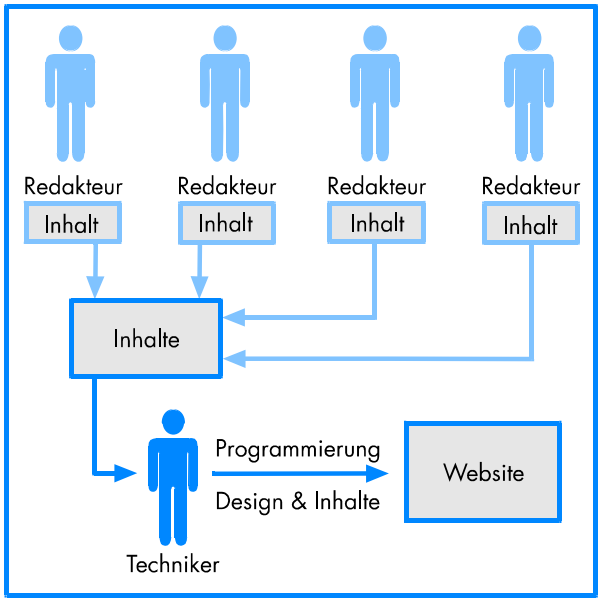
6.2 Webpublishing

Herkömmliches Webpublishing

Beim herkömmlichen Webpublishing verwaltet ein Webmaster die Website eines Unternehmens. Er bereitet diese technisch auf und programmiert die Inhalte mittels HTML. Die Inhalte bekommt er von verschiedenen Fachabteilungen und muss sie Internetgerecht aufbereiten. Wird das Layout der Website geändert, führt dies zu einer aufwendigen Überarbeitung aller vorhandenen HTML-Seiten, d. h. jede Seite muss separat manuell abgeändert werden. Durch den steigenden Umfang der Inhalte erhöht sich gleichzeitig der Arbeitsaufwand. Somit ist der Webmaster überlastet und es entsteht ein hoher Zeitaufwand, bis die Inhalte im Web stehen. Des weiteren sind viel zu wenige Mitarbeiter eingebunden, was den Nachteil hat, dass Inhalte erst auf Anfrage des Webmasters zusammen getragen werden. Hiermit können Aktualität, Qualität und Sicherheit mit dem bisherigen Webpublishing nicht erfüllt werden.

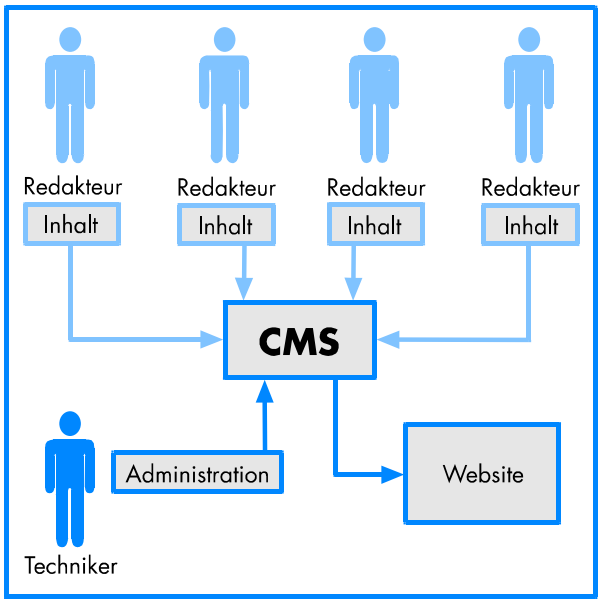
1 (Koop, H.-J.;Jäckel, K.K.; Van Offern, K.K.: Erfolgsfaktor Content Management - Vom Web Content bis zum Knowledge Management, 1. Auflage, Wiesbaden, Vieweg, 2001, S.83, Z.14;)

Schaubild zum herkömmlichen Webpublishing:



Webpublishing mit CMS

Dadurch das Website Programmierung mit einem Content Management System überflüssig wird, kann nun eine Vielzahl von Mitarbeitern die Inhalte erstellen. Durch die Bearbeitung im Browser ist man zusätzlich auch Standortunabhängig. So werden die Inhalte dort erstellt und erfasst, wo man sie produziert. Die Qualität einer Website ist durch den definierten Workflow und Freigabezyklen gewährleistet. Dadurch das jetzt der Webmaster nicht mehr auf sich alleine gestellt ist, wird ein kürzerer zeitlicher Aufwand erzielt. Die Veränderungen auf der Website erfolgt in Echtzeit.



1 (Koop, H.-J.;Jäckel, K.K.; Van Offern, K.K.: Erfolgsfaktor Content Management - Vom Web Content bis zum Knowledge Management, 1. Auflage, Wiesbaden, Vieweg, 2001, S.159, Z.7);

Vergleich:

Webpublishing ohne CMS	Webpublishing mit CMS
Inhalt und Layout sind zusammen	Inhalt und Layout sind getrennt
Webmaster veröffentlicht	Webmaster ist überflüssig
Einheitliches Layout der Website ist nicht gewährleistet	Einheitliches Layout der Website ist gewährleistet
Desto größer die Website, umso höher sind Kosten und Aufwand	Desto größer die Website, umso geringer sind Kosten und Aufwand
Aufwendiges Redesign	Einfaches Redesign

6.3 Mitarbeiter im Bezug auf CMS

Mitarbeiter die täglich operativ mit dem CMS arbeiten sind die wichtigste Zielgruppe des Systems. Wenn diese das System nicht annehmen, kann ein Content Management-System auch nicht helfen.
Man sollte also klare Vorgaben und Zielrichtungen bezüglich den gewünschten und erwarteten Auswirkungen des Content Management-Systems für und auf die Benutzer machen. z.B.:

Benutzerinterface:

Es existieren zwei Möglichkeiten entweder eine Weboberfläche oder eine Client-Software.

Regeln zur Nutzung des Conten Managements:

- Absicherungsregelungen
- Sichtbarkeitsregeln
- Rechte- und Benutzerrollenkomplex

- Übernahmeregeln (wer sonst noch mitarbeiten darf)
- Automatisierungsregeln
- Eskalationsregeln (Hilfe für den Benutzer)

Schulungen

Service und Support

Man sollte einerseits den Benutzerwünschen so weit wie möglich entgegenkommen und andererseits die Benutzung des Content Management-Systems so weit wie nötig absichern.

Betroffene und Beteiligte:

User:

Gruppe aller Mitarbeiter die das Content Management (-System) innerhalb des Unternehmens stützen

Fachspezialisten:

Mitarbeiter die direkt innerhalb eines Teams, einer Gruppe oder Abteilung "Content Management" tätig sind und mit inhaltlichen oder organisatorischen Fragen zu tun haben

IT-Spezialisten:

Mitarbeiter im Bereich IT ode EDV, die für die technischen Aspekte des Content Managements zuständig sind.

6.4 Personal

Die Neueinführung eines Systems, egal ob es IT-gestützt oder nicht IT-gestützt ist, ist immer stark mit dem Bereich Human Resources verknüpft. Denn schließlich sind immer Menschen die Träger der Veränderungsprozesse. In diesem Zusammenhang gilt es User, Content Management-Spezialisten und IT-Experten gleichermaßen bei allen Fragestellungen differenziert zu berücksichtigen. Ebenso ist die bedürfnisgerechte Gestaltung eines durchgängigen Content Managements Voraussetzung für dessen nachhaltige und erfolgreiche Integration.

Zentrale Fragestellungen im Zuammenhang von Content Management und Human Resources sind:
- Ansprüche an ein Content Management (-System)
- Durch die Einführungen eines CM (-System) bedingte Auswirkungen
- Projekt-Realisierung und Personalresourcen
- Methoden und Maßnahmen zur Umsetzung

Folgende Aspekte spielen aus Personal-Sicht, sowohl bei der Bestimmung der Anforderungen als auch bei der Ermittlung der Auswirkungen hinsichtlich eines CMS, eine große Rolle:
- Systemdesign, Personalstruktur, Prozessen und Zielsetzungen
- Mitarbeiterkompetenzen, Mitarbeiterentwicklung und Mitarbeiter-Führung

6.5 Kunden:

„Ein Unternehmen ist dazu da, die Bedürfnisse der Kunden zu befriedigen. Denn wenn ein Unternehmen aufhört, die Bedürfnisse seiner Kunden zu bedienen, dann hat es seinen Platz auf dem Markt - meist recht kurzfristig - verloren."[1]

Kundenzufriedenheit ist das Ergebnis optimaler Dienstleistungen, eines erstklassigen Services und damit einer konsequenten Ausrichtung des gesamten Unternehmens auf den Kunden. Grundlage ist eine echte Beziehung zum Kunden.

Eine konsequente Ausrichtung eines Unternehmens auf die Kunden kann in hohem Maße durch die Verfügbarkeit kundenrelevanter Contents für alle Mitarbeiter des Unternehmens verstärkt werden. Wichtig ist ebenso, dass alle Prozesse für die Mitarbeiter so vereinfacht werden, dass diese sich mit Freude allen Kundenbelangen widmen können, ohne von internen Prozessen behindert zu werden.

Im Brennpunkt des Kundenservice stehen die Kunden, er hat damit direkten Einfluss auf die Kundenzufriedenheit und somit auch auf die Kundenbindung. Durch die zielstrebige Ausrichtung auf den Kunden mit Hilfe kundenrelevanter Contents und die umfassende Berükksichtigung von Kundenwünschen und Bedürfnissen beim Design und der Konzeption des Content Managements können entscheidende Wettbewerbs-Vorteile errungen werden.

1 (Koop, H.-J.;Jäckel, K.K.; Van Offern, K.K.: Erfolgsfaktor Content Management - Vom Web Content bis zum Knowledge Management, 1. Auflage, Wiesbaden, Vieweg, 2001, S. 159)

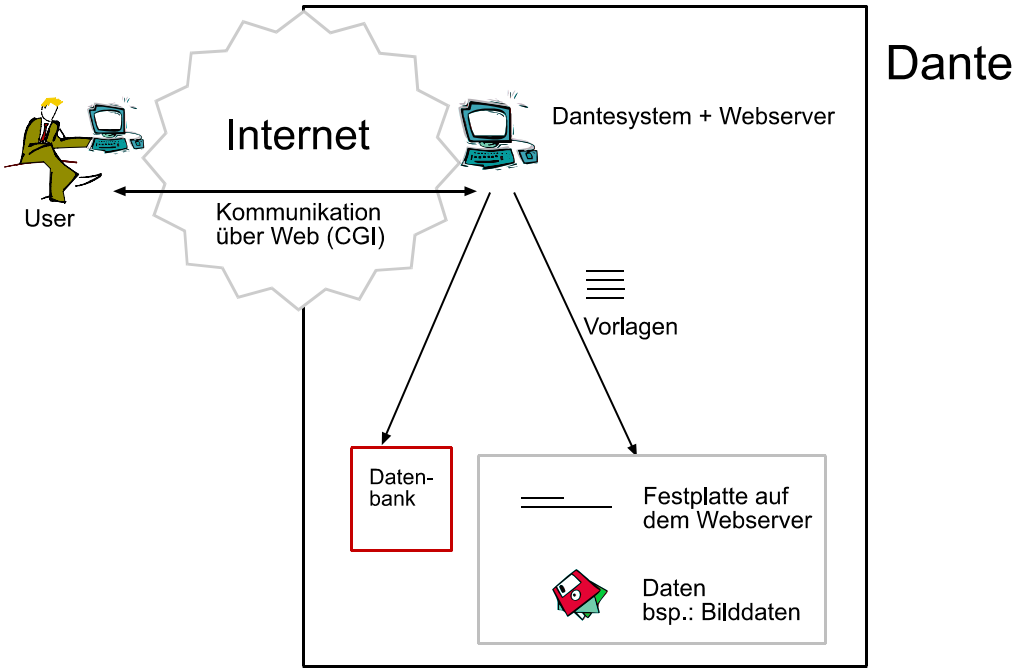
7. Softwarebeispiele

7.1 dante

Das Internet dient als Informations- und Unterhaltungsmedium. Es ist mittlerweile eines, vielleicht sogar das wichtigste Kommunikationsmedium. Die Firma a3design GmbH wurde im Januar 1999 in Zweibrücken gegründet und ist ein national tätiges Internet Systemhaus für Informationslogistik. Diese Firma entwickelte das Content Management System dante. Mit dieser Software unterstützt a3design das Verwalten von Websites Ihrer Kunden.

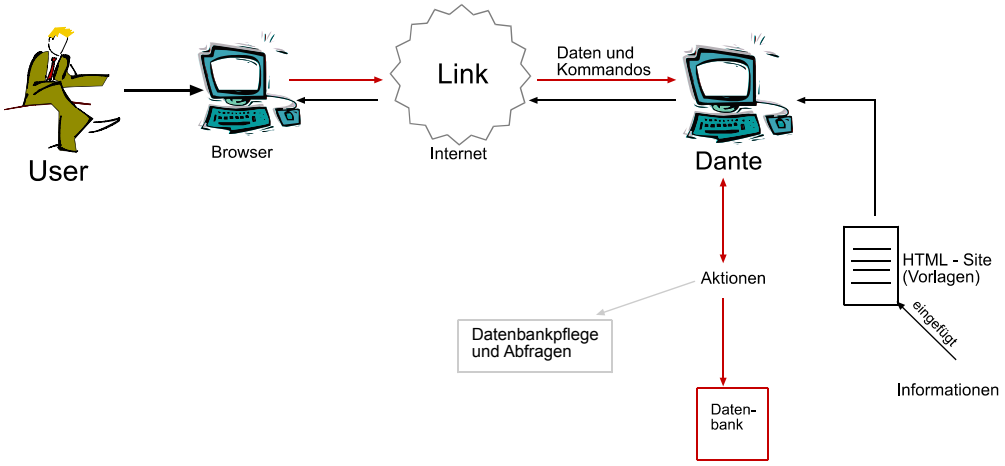
Theoretische Grundlagen

Aufbau eines Informationskanals:
Wesentliche Komponenten um mit dante arbeiten zu können sind:
- Dateien auf der Festplatte (Vorlagen)
- Datenbank (Daten)
- dante System
- Webserver



Kommunikation zwischen User und dante

Das dante System wartet auf Kommandos des Users. Dieser hat die Möglichkeit per Browser auf das System zuzugreifen und ihm mitzuteilen, welche Informationen er haben möchte. Über das CGI (Common Gateway Interface) Protokoll findet der Kommunikationsaustausch statt. Hier wird festgelegt auf welche Art und Weise die Befehle und Daten zwischen Internet Browser und Webserver ausgetauscht werden. Das Besondere hierbei ist, dass dante niemals ohne spezielle Aufforderung agieren kann. Damit Daten und Befehle via Internet transportiert werden können, müssen sie codiert werden. Die Codierung ist im CGI Protokoll festgelegt. Anweisungen sind Daten und Kommandos vom User.



Daten liegen in der Datenbank vor

Datenbanken sind komplexe Gebilde, sie verwalten Daten. Aufgabe von Datenbanken ist es, Massendaten strukturiert aufzubewahren und über eine eigens dafür entwickelte Abfragesprache namens SQL (Structured Query Language) schnellstmöglich auslesen, verändern und schreiben zu können.
Des weiteren besteht eine Datenbank aus einer Tabellenansammlung. Der Tabellenaufbau sieht wie folgt aus: Er besteht aus Spalten und Zeilen. Eine Zeile wird auch als Datensatz bezeichnet. Jede Spalte besitzt einen eindeutigen Namen.

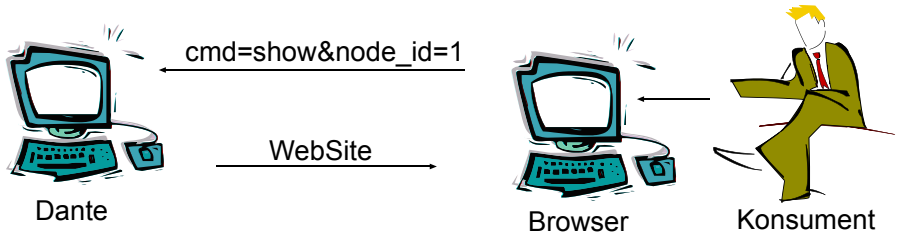
Vorlage

Eine Vorlage kann man am besten anhand eines HTML-Beispiels erklären:

Die Ersetzungsvariable <#titel#> wird mit einem benannten String (=Text) aus einer Datenbank ersetzt. Das Konstrukt <#titel#> dient somit lediglich als Platzhalter. Eine Aufgabe von dante ist es, den benannten String aus einer Datenbank so mit der Vorlage zu verknüpfen, dass die zuvor benannten Ersetzungsvariablen in der Vorlage ersetzt werden. Doch woher weis dante, welche Vorlage mit welchem Datensatz verknüpft oder welcher String in welche Vorlage eingesetzt wird?

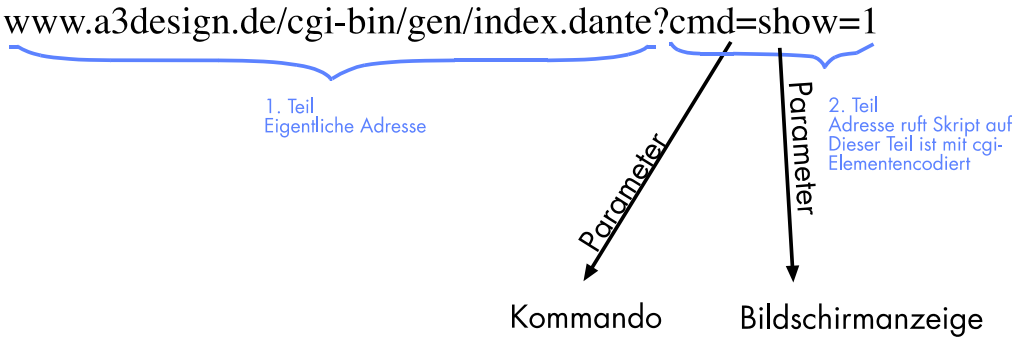
Der User steuert das System mit seinen Kommandos und Daten im CGI-Format. Er sagt also dante, welche Vorlage angezeigt werden soll. Das dante-System verarbeitet den Aufruf und sendet dem User die entsprechenden Informationen zurück. Diese Information wird in Form einer HTML-WebSite im Browser angezeigt.

Wie baut man einen Informationskanal auf?
Jeden Informationskanal kann man mit einem Schema beschreiben:



Das Ziel hierbei ist, aus Datenbankdaten Informationen zu gewinnen und diese zum Konsumenten zu bringen.
Wie das genau passieren soll, entscheidet nur die Art der Vorlagen.

Um diesen Vorgang zu verdeutlichen würde der Link wie folgt aussehen
Beispiel eines CGI-Aufrufs:



Grundlage in dante

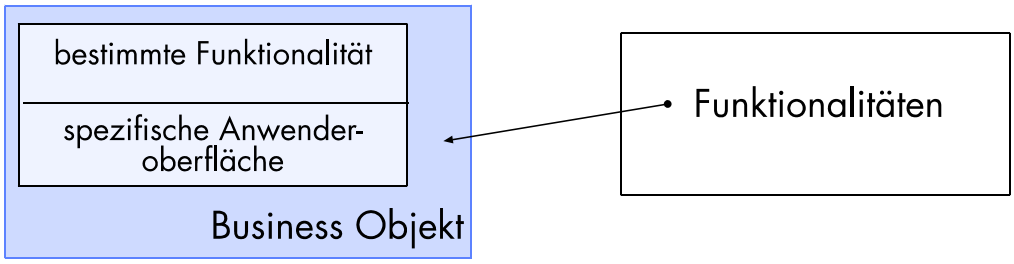
Allgemein:

dante ist ein Client-Server, Webbasiertes Content Management System. Es besitzt folgende Eigenschaften:
- Es ist eine datenbankbasierte, multilinguale Mehrbenutzeranwendung.
- Es ist mit entsprechenden Modulen erweiter- sowie Skalierbar.
- Es besteht aus einem Kern, an den man je nach Bedarf einzelne Zusatzmodule hinzufügen kann.
- Die Module können durch ihre Parameter an die Bedürfnisse der Kunden angepasst werden.

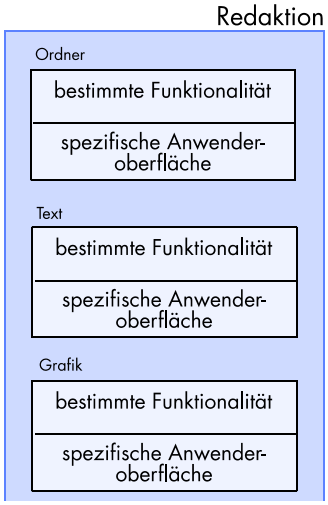
Module

Module stellen Funktionalitäten zur Verfügung, indem sie programmierte Softwarebausteine referenzieren. Diese Bausteine werden Business Objekt genannt.

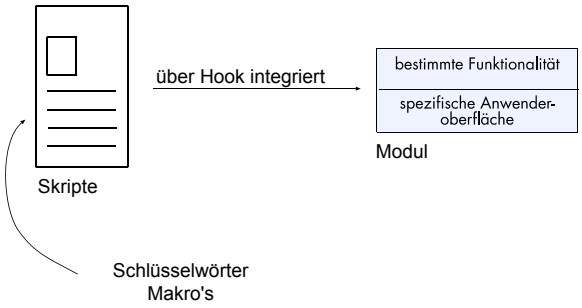
Die eigentliche Funktionalität der Module stellt dieses Business Objekt zur Verfügung. Es besteht aus einer bestimmten Funktionalität und einer spezifischen Anwenderoberfläche.



Zusammengehörende Module, z.B. „Ordner, Rubrik, Text, ...“ werden in einer dante-Klasse zusammengefasst.



Die zu einer Anwendung zusammengefassten Module können beliebig viele Instanzen von Modulen bilden, die dasselbe „Business Objekt“ referenzieren.
Modulen werden über Parametereinstellungen individuell angepasst. Dieses Einstellen von Parametern, wird auch als„Customizing“ bezeichnet.
Ist kein Customizing möglich, kann man den Modulen weiter Funktionalitäten durch sogenannte „Business Rules“ hinzufügen. Business Rules sind Skripte, welche über einen Hook integriert werden können. Ein Hook ist ein Eingriff in den Arbeitsprozess.
Skript können aus definierten Schlüsselwörtern oder komplexen Skripten bestehen, sie werden auch als Makro bezeichnet.



dante ist mit einem Grundstock von Modulen ausgestattet, mit denen man alle Standard Anforderungen im elektronischen Geschäftsverkehr abwickeln kann. Es gibt zwei Methoden:
a) auf ein bestehendes Modul zugreifen
b) ein neues Modul programmieren und einfließen lassen.

Architektur

Bestehende Anwendungen können im laufenden Betrieb verändert und angepasst werden (durch Module und Zusammenstellen), ohne dass die ganze Anwendung erneut konzipiert oder erstellt werden muss. Daher entfällt das Fehlersuchen und ein langwieriges Compilieren. Die Folge davon ist, dass dadurch die Erstellung, Pflege und Wartung der Anwendung sehr schnell vor sich geht und des weiteren auch noch günstig ist.

Anwendungsspezifische Daten werden in der Datenbank gespeichert und mit dante in Form eines Objektes, namens „Item" repräsentiert. Jedes Item besitzt außer dem für ihn spezifischen Datensatz auch eine Referenz auf ein Modul. Des weiteren gehören sie zu einem bestimmten Modul-Typ. Dies bedeutet, dass die Items sämtliche Rechte, Anpassungen sowie „Business Rules" des referenzierten Moduls besitzen.

Dante basiert auf einer webbasierten „Client/Server-Architektur". Die Kommunikation geht zwischen Client und Server über HTTP, CGI und FTP von statten. Der Client ist ein typischer Webbrowser und der Server ein typischer Webserver.
Die Kommunikation mit der Datenbank findet über Sockets oder ODBC statt. Ein Socket ist ein eindeutiger Kommunikationsendpunkt, die Prozesse verständigen sich in TCP/IP Netzen durch das Schreiben und Lesen in Sockets. ODBC steht für Open Database Connectivity und stellt eine Art inoffiziellen Standard für Datenbanken dar.

dante besitzt eine abstrakte Schicht zur Middleware, also zum Webserver und der Datenbank. Dadurch sind die Komponenten austauschbar. Die Folge hiervon ist die Möglichkeit der Skalierung von Kleinsystemen bis Großrechnern ohne zu programmieren zu können.

Kommunikation mit Drittsystemen

Über spezialisierte Module läuft die Kommunikation mit Drittsystemen ab. Die Module werden „Interfaces" bzw. „Business Interfaces" genannt. Fremddaten die als Items repräsentiert werden, werden über die Verwendung der Interfaces auf die gleiche Form gebracht, wie interne Daten. Dadurch ist die Anwendung von Funktionalitäten auf andere interne Module möglich.

Frontend und Backoffice

Die Benutzung und die Entwicklungsumgebung von dante wird im Backoffice, einem einheitlichen dynamisch generierten, passwortgeschützten Bereich, angewendet. Der Clou bei der ganzen Sache ist, dass die Anwendung und die Entwicklungsumgebung, über das gleiche System bedient werden.

Der Unterschied zwischen Entwicklung und Anwendung ist, dass der Anwender einer Rechteeinschränkung unterliegt, d.h. er gehört einer Benutzergruppe an, die keine Rechte fürs Anpassen (Customizen) von Modulen besitzt.

Frontend ist eine individuell anpassbare Oberfläche, über die jede Anwendung neben dem Backoffice bedient werden kann. Die Frontends müssen angepasst und anwendungsspezifisch erstellt werden.
Des Weiteren besitzen sie eine definierte API (Application Programm Interface) zum Backoffice. API ist eine Programmierschnittstelle und Sammelbezeichnung für alle Funktionen, die ein Betriebssystem oder eine Betriebsystemerweiterung zur Verfügung stellt. Mit API gelten für den Anwender die gleichen Zugriffs- und Benutzerrechte wie im Bakkoffice. Die „Business Rules" die im Backoffice definiert wurden, werden nicht mit ins Frontend übernommen.

Frontend-Erstellung

Für die Fronted-Erstellung stehen dem Frontend-Entwickler abstrakte Klassen zur Verfügung. Diese stellen die Software Grundbausteine dar. Der Entwickler kann darauf zugreifen und sie über Methoden-Überlagerung und anwendungsspezifische Eigenschaften erweitern.
Des Frontends abstrakte Klassen unterliegen dem Open Source Gedanken, wodurch die Entwickler alle Freiheiten in Bezug auf Anpassung und Eigenentwicklungen haben.
Die Erstellung der Frontend-Generierung erfolgt mit Hilfe von „semantischen Vorlagen". Semantische Vorlagen werden auch Templates genannt. Hierbei handelt es sich um HTML-Vorlagen, die spezifisch erweitert werden können. Diese Befehle werden als dML (dante Markup Language) bezeichnet. dML sind die Befehle zur Erweiterung. Sie sind ähnlich wie HTML oder XML.
Semantisch bedeutet, dass die Vorlage über das Frontend-API weiß, zu welchem Item und Modul sie gehört, d.h. das Layout kann die Daten interpretieren und darstellen.

Backoffice Customizing

Customizing bedeutet, die vielseitigen Module die dante anbietet, dem Anwender zur Verfügung zu stellen. Diese können Administratoren oder Webnutzer sein. Ein Modul hat eine bestimmte Funktionalität und kann eine Diskussionsform, ein Terminkalender, Benutzermanager oder ein Shop sein.

Das Ziel von Customizing ist, die Schaffung einer Umgebung zur Pflege von Daten für Online-Redakteure. Dazu benötigt dieser keine professionellen dante-Kenntnisse. Der Online-Redakteur loggt sich mittels Passwort in den Backoffice Bereich ein. Er sieht jetzt aber nur den Bereich zum Bearbeiten und Pflegen seines Projektes.

Das schrittweise Vorgehen zum Erstellen einer Website mit dante

Um in das Programm dante zu können, muss man folgende Einstellungen tätigen:

1. Im Internetexplorer über Extras, Verbindungen, LAN-Einstellungen den Proxy-Server deaktivieren.
2. Die Adresse: **http://172.16.107.10/admin.dante** eingeben.
3. Einloggen

Hierbei sollte man beachten, dass man zuvor alle Rechte erhalten hat. Also „Gott" ist, da sonst die Erstellung einer Site unmöglich ist!

Wenn man sich mit der Struktur von Systemeinstellungen und Kunden vertraut gemacht hat (Module und Klassen), sollte man sich in Adam ein neues Projekt anlegen. Dies geschieht, indem man über Einfügen einen Neuen Projektordner erstellt. Man beachte, mit jeder Erstellung wird ein neues Modul geschaffen und verknüpft.
In diesem Projektordner hat man verschiedenen Möglichkeiten seine Website mit weiteren Unterordner zu strukturieren.

Nun sollte man in der Benutzerverwaltung Benutzer anlegen.

Beispiel:
Sie besitzen eine Zeitungsfirma. In dieser Firma arbeiten logisch gesehen mehrere Leute an der Zeitung. Es gibt Redakteure, Fotographen, Texter, Drucker, Korekteure.... Jeder dieser Redakteure besitzt einen Namen. Mein Redakteur besitzt den Namen Klaus. Wenn ich Klaus als neuen Benutzer anlegen möchte, gehe ich in die Benutzer-verwaltung und füge einen neuen Benutzer ein. Diesen nenne ich Klaus und gebe ihm ein Passwort. Wichtig ist jetzt, dass ich Klaus den richtigen Knotenpunkt eintrage. Denn dieser Knotenpunkt regelt, in welchem Bereich Klaus tätig sein darf. Man erkennt einen Knotenpunkt an der Zahl eines Ordners. Die Redaktion, in der Klaus arbeiten soll, besitzt bei mir die Nummer1414. Somit darf Klaus ab dem Ordner Redaktion uneingeschränkt arbeiten. Doch uneingeschränkt heisst, dass er Dinge erstellen, bearbeiten und löschen kann. Wenn man dies vermeiden möchte und Klaus nur Dinge bearbeiten soll, muss man Klaus einer bestimmten Gruppe oder auch Rolle genannt, zuordnen.
Man legt also z.B. die Gruppe, im Rollenmanager, Redakteure an. Somit haben viele gleiche Personen die ebenfalls in dieser Gruppe liegen die gleichen Rechte.

Rechte?

Rechte vergibt man indem man ein Ordner oder ein Dokument aufruft und dieses am Schlüsselsymbol klickt. Hier kann man diesem Dokument verschiedene Gruppen zuordnen und für jede seperat die Rechte vergeben. Rechte wären hier: Access; View; Pool; Grant.

Vergibt man einem Überordner allerdings für z.B . die Gruppe Redakteure das Recht lesen, so darf kein Redakteur in diesem Ordner etwas erstellen oder löschen. Verbietet man allerdings der Gruppe Redakteure ganz den Zutritt, so kommt auch kein Redak-teur an die untergeordneten Dokumente.

Um seine Ergebnisse jetzt im Frontend begutachten zu können, muss man folgende Adresse eingeben: **http://172.16.107.10/index.dante** . Achtung, diese Adresse funktioniert nur mit dem Anonymous (Verwalter)!!! Deshalb diesen niemals löschen!

8. Zope

Entstehungsgeschichte

Zope „Z Objekt Publishing Environment" ist ein Produkt der Firma „Digital Creations" aus Fredericksburg Virginia (USA). Es ist eine Zusammenführung der beiden Digital Creation Produkte Bobo und Principia, die in der Programmiersprache Python mit einigen C-Einbettungen realisiert worden sind. Ursprünglich wurde es von Digital Creation nur für Eigenzwecke eingesetzt. Die Veröffentlichung von Zope 1998, als Open Source Projekt, bewirkte dass seine Weiterentwicklung extrem vorangetrieben wurde. So dass es mittlerweile schon eine große Schar von Anwendern zählt und von vielen Unternehmen eingesetzt wird.

Bobo: - wurde als objektorientierter Ausweg aus der CGI-Misere, zur freien Verfügung entwickelt

Principia: - ist ein kommerzieller Applikationsserver auf Bobo-Basis, der seit der Python-Konferenz 1998 auch frei erhältlich ist

Kurze Beschreibung

Basisfunktionen:
- rollenbasiertes Authentifikationssystem
- objektorientierter Ansatz
- integrierter Webserver (Applikationsserver) und objektorientiertes Datenbanksystem

Serverbetriebssysteme: Linux, Solaris, Windows NT (aus Sicherheitsgründen besser zu verwenden als Windows 95-2000)

Scriptsprachen: Python, DTML
Schnittstellen: XML
Sprache: Englisch

Allgemeines

Zope ist ein Web-Publishing-Werkzeug, das hilft dynamische Websites portabel zu erzeugen und bietet darüber hinaus einen guten Durchsatz bei einer geringen Systembelastung. Es ist als reiner Quelltext oder als binäre Distribution erhältlich. Sein Konzept ermöglicht eine schnelle und einfache Anwendungsentwicklung. Es unterscheidet sich von anderen Applikationsservern dadurch, dass es kein übliches HTML-Dokument zu Verfügung stellt (vgl. Kapitel 4.4 Serverkonzepte). Der Webserver ist übrigens eine erweiterte Version des Python-Servers Medusa. Neben dem Web-Protokoll HTTP versteht der Server auch das FTP und WebDAV-Protokoll, dadurch wird das Laden von Webseiten, Ordner mit enthaltenen Grafiken, … in die objektorientierte Datenbank kinderleicht. Das System kann natürlich auch mit anderen SQL-Servern per CGI kommunizieren und bei Bedarf XML-Dateien einbinden. Zope besitzt ein komfortables und schnelles Web-Front-End, über das fast alle Funktionen der Plattform administrierbar sind. Seine webbasierte Oberfläche ähnelt der des Window-Explorers, da sie ebenfalls eine Baumstruktur mit

Ordnern und darin enthaltenen Objekten wiederspiegelt. Mit Programmierkenntnissen lassen sich in DTML schnell umfangreiche Ergebnisse erzielen. Der Anwender stößt dabei nur ab und zu an seine Grenzen, die Open Source Produkte mit sich bringen. Zudem liegen in Python bereits viele fertige Skripte und Programme vor, die sich verwenden lassen, wie z.B. die pythonbasierte Testdatenbank. Zope ist besonders für wachsende Strukturen von Vorteil, da für zusätzliche Zusatzmodule, Nutzer und CPUs kein Cent extra gezahlt werden muss.

Architektur

Die Architektur ermöglicht es, dass die Entwicklung von Datenbankapplikationen in die Bereiche Datenzugriff und Layout getrennt werden können. Diese Reduktion der Komplexität erleichtert wiederum eine separate Spezifikation und Realisierung. So können Entwickler sich auf den Bereich konzentrieren der ihnen auch liegt.

Installation unter Linux

1. Zope-Version unter http://www.zope.org herunterladen und entpacken
2. das im Ordner enthaltene Install-Skript ausführen

Zope wird dadurch mit dem Standardskript gestartet und ist unter dem Port http://localhost:8080 ansprechbar. Man gelangt zur Webinterface, in dem man an die URL „/manage“ anhängt (http://localhost:8080/manage). Nach dem Login erscheint der Manager-Bildschirm, von dem aus der Entwickler seine Website entwickelt. Hier können neue Manager, User, Unterbereiche (Folder) und Dokumente anlegt, sowie Bilder und HTML-Seiten hochgeladen werden. Dokument-Vorlagen lassen sich über mehrere Mechanismen entwerfen.

Umgang mit Zope

Die Pflege und Bearbeitung von Inhalten geschieht über den Zope-Manager. Er ist in unterschiedliche Bereiche aufgeteilt. Er bietet im linken Frame eine Baumdarstellung, im Rechten sind die Ordnerinhalte zu sehen. Im oberen Bereich, der einer Reiter-Struktur ähnelt kann man auf verschiedene Objektansichten und Managementseiten umschalten (z.B. Edit, Bindings, Test, Proxy, Ownership, Properties, Security und History). Zopes interne Datenbank bietet eine Versionsverwaltung, eine UNDO-Funktion und mehrere Sessions an. Sie ermöglichen die Bearbeitung von Inhalten, ohne die aktuellen Inhalte des Servers zu beeinflussen. Der Vorteil ist, dass man keinen zusätzlichen Testserver benötigt. Aufgrund des objektorientierten Ansatzes kann der Entwickler für jedes Objekt Eigenschaften (Properties) definieren, auf die er mit Hilfe von Template-Sprachen zugreift. Objekte können ihre Eigenschaften auch vererben. Zum Beispiel erbt ein Objekt Eigenschaften eines Ordners in dem es liegt. Ergebnisse von SQL-Abfragen sind wiederum Objekte. Auch sie erben wie die anderen Objekte Eigenschaften und Methoden aus ihrer Umgebung. Zusätzlich können über Python-Klassen weitere Methoden hinzugefügt werden. Es existiert wie bei den anderen Objekten auch ein direktes Mapping von URLs auf Datenbankanfragen. Dieses schließt Standardmethoden und andere Methodenaufrufe inklusive. Parameter ein. Zur Realisierung von SQL-Methoden bietet Zope einen Testmodus und die Erzeugung eines Templates für das passende HTML-Such-Interface an, was bei anderen Systemen oft fehlt. Dieses Template kann individuell angepasst werden. Zusätzliche Parameter gestatten die Festlegung der maximalen Ergebnissreihen und des Cachingverhaltens.

Intelligente Dokumente

Zope stellt wie bereits im Allgemeinen Teil erwähnt wurde keine üblichen HTML-Dokumente zur Verfügung. Dadurch befreit es Autoren, alles fest in HTML-Quelltext zu kodieren, der wie sie in Kapitel 5 „Entwicklung von HTML zu XML" erfahren haben, nicht allzu viel gestalterischen Spielraum bietet. Trotzdem ist es aber möglich, was aber bedeuten würde, dass die Dokumente, die in DTML („Dokument Template Markup Language") erstellt werden, ihre Intelligenz verlieren würden. DTML bettet ähnlich HTML-Tags Befehle in den Quelltext ein. In Zope enthalten DTML-Dokumente lediglich anzuzeigenden Inhalt. Sie werden ebenfalls als Objekt bezeichnet, die ihre Eigenschaften von einer Zope-Klasse erben. DTML-Methoden benutzen jedoch nur andere Objekte, um etwas darzustellen. Sie gelten aber nur innerhalb des Ordners wo sie auch liegen. Zum Beispiel lässt sich ein einheitliches Design für alle Seiten ganz einfach umsetzen, indem man einen Header und Footer als Tags (<dtml-var Name>) darin einbaut.

Vererbung und Benutzerverwaltung

Zur Festlegung der Objekteigenschaften wird nicht nur die klassische Beziehung Objekt A ist ein Objekt B benutzt sondern auch die Akquisition. Sie realisiert die„ ist enthalten in" Beziehung und sorgt für die Vererbung von Eigenschaften der einer URL übergeordneter Ordner. Vererbt werden Dateiinhalte und Fähigkeiten. Akquisition kommt auch bei den Sicherheitsmechanismen zum Einsatz. Wer eine Autorisierung für viele Dokumente und Dienste benötigt legt ein Benutzerobjekt an.

Das Sicherheitsmodell geht von der dezentralen Verwaltung des Servers aus. Der Site-Manager kann Verantwortung an andere Mitarbeiter delegieren bzw. abgeben, die wiederum die Verantwortung von ihrem Teilbereich abtreten können. Diese Delegationspyramide lässt sich in der Baumstruktur anhand der Ordner-Hierarchie ablesen. Zope bietet vier Komponenten zur Benutzerverwaltung an:

1. Benutzer/Benutzerobjekte: - sie dienen zur Authentifizierung im System (Nutzer- und Passwortabfrage)
2. Rollen: - sie sind das Bindeglied zwischen Benutzern und Rechten (bei anderen Systemen ist die Rolle als „Gruppe" anzusehen)
3. Rechte: - sie legen die Autorisierung für Operationen fest
   - Objekte können auch Rechte erzeugen, die in den Standardobjekten nicht vorgesehen sind
4. Akquisition: - sie sorgt dafür, dass sich Rechte, in der URL-Hierarchie, auch nach unten hin durchsetzen

Aus diesen Gründen sollte man die Benutzerverwaltung weit oben im der Baumstruktur ansiedeln, um so die darunter liegenden Ordner/Objekte zu kontrollieren. Mit der Benutzerverwaltung lässt sich somit der Zugriff auf einzelne Objekte detailliert einschränken.

Produkte

Einmal erstellte Arbeiten kann man natürlich wiederverwenden, wenn man eine Vorlage, mit einer Konfigurationsinterface erstellt. Vorlagen heißen in Zope Produkt. Sie lassen sich leicht integrieren und stehen danach als installierbare Objekte über den Zope-Manager zur Verfügung. In Zope werden einige vorgefertigte Produkte, unter anderem auch ein E-Commerce-Produkt über den Produktmanager (im Control-Panel) angeboten. Im Zope-Manager-Guide findet man ein Beispiel, das einem hilft ein Produkt zu erstellen.

Resumee

Zope ist jedoch kein System, bei dem man nach „Knopfdruck" (Login) sofort loslegen kann - ganz im Gegenteil. Man muss erst einmal viel Zeit damit verbringen, sich in dem Zope-Manager zurechtzufinden um alle Funktionen kennenzulernen bzw. auszuprobieren. Was wiederum für Anwender die der englischen Sprache nicht so mächtig sind, ein zusätzliches Problem darstellt. Vor allem durch die mangelhafte Dokumentation ist der Umgang mit Zope nicht gerade leicht.

8. Anhang

8.1 Quellenverzeichnis

Bücher:

Zschau, O.; Traub, D.; Zahradka, R.: Web Content Management - Websites professionel planen und bearbeiten, 2. Auflage, Bonn, Galileo Press GmbH, 2002

Harms, F.; Koch, D.; Kürten, O.: Das große Buch (X)HTML & XML, 1. Auflage, Düsseldorf, DATA BECKER GmbH & Co.KG, 2000

Koop, H.-J.;Jäckel, K.K.; Van Offern, K.K.: Erfolgsfaktor Content Management - Vom Web Content bis zum Knowledge Management, 1. Auflage, Wiesbaden, Vieweg, 2001

Zeitschriften:

Prüfstand - Content Management-Systeme, ct'2002, Heft 20, S.178/ff
The X-Change, Page 2002, Heft 08, S.61/ff

Internetadressen:

http://www.content.manager.de/magazin/artikel_240_print_cross_mediales_publizieren
http://www.community-of-knowledge.de/cp_artikel_d.htm?artikel_id=60
http://www.community-of-knowledge.de/cp_artikel_d.htm?artikel_id=71
http://www.gi-ev.de/informatik/lexikon/inf-lex-content-management.shtml
http://www.unileipzig/artikel_cms
http://www.pcintern.de/artikel_was_ist_eigentlich_CMS?
http://www.zope.org
http://www.dzug.org/zope-book
http://software.zeit.de/software/index
http://www.informationweek.de/channels/ channel09/021228.htm
http://www.pinuts.de

8.2 Protokolle

Protokoll

Datum:	Freitag, den 15.11.02
Uhrzeit:	13.00 Uhr bis 15.00 Uhr
Anwesend:	Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant:	Dimpfl Stefanie

Thema des Treffens: Informationssuche

1. Internetrecherche zu CMS (Ronny, Sabrina):
   Ergebnisse:
   - www. ...

2. Buchrecherche zu CMS in der Landesbibliothek (Stefanie):
   Ergebnisse:
   - keine Bücher über CMS in Speyer vorhanden, daher Buchbestellung

3. Probleme des Treffens:
   - keine

4. Weitere Planungen:
   - Zeit- und Aufgabenplan sowie Regeln aufstellen

Protokoll

Datum: Dienstag, den 19.11.02
Uhrzeit: 13.00 Uhr bis 13.30 Uhr
Anwesend: Ellermann Ronny, Erdle Sabrina
Protokollant: Erdle Sabrina

Thema des Treffens: dante

1. Erläuterungen zu dante:
   - Benutzeranlegen
   - Einloggen
   - Passwörter

2. Erklärung des CM-Systems:
   - IP-Nummer
   - Adresse
   - Adam-Datei

3. Ausgabe des Informationsheft zu Dante:

4. Probleme des Treffens:
   - keine

5. Weitere Planungen:
   - keine

Protokoll

Datum: Mittwoch, den 20.11.02
Uhrzeit: 15.15 Uhr bis 16.10 Uhr
Anwesend: Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant: Dimpfl Stefanie

Thema des Treffens: Themenfestlegung

1. Einführung in dante mit Herrn Scheib:
   - Benutzer anlegen
   - Einloggen
   - Passwörter
   - Übersicht über einzelne Ordner
   - Abrufen verschiedener Server

2. Probleme des Treffens:
   - keine

Protokoll

Datum: Mittwoch, den 27.11.02
Uhrzeit: 15.15 Uhr bis 16.30 Uhr
Anwesend: Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant: Ellermann Ronny

Thema des Treffens: Vergleich, Aufgabenverteilung

1. Vergleich bisheriger Informationen:
   - Wir haben unsere bisher gesammelten Informationen miteinander verglichen
   - Anschließend im Internet nach weiteren Informationen über Content Management gesucht

2. Aufgabenverteilung:
   - Zum Schluss unserer Sitzung haben wir unsere neu erungenen Informationen untereinander aufgeteilt

3. Probleme des Treffens:
   - keine

Protokoll

Datum: Donnerstag, den 19.12.02
Uhrzeit: 12.00 Uhr bis 13.30 Uhr
Anwesend: Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant: Dimpfl Stefanie

Thema des Treffens: Festlegen der Treffpunkte in den Ferien

1. Materialeinteilung:
   - Neue Bücher aus der Landesbibliothek besprechen, erste bisherige Erfahrungen austauschen

2. Zeitrahmen:
   - Bis wann sollten die Bücher gelesen sein?
   - 27.12.2002

3. Neue Treffpunkte festlegen:
   - 02.01.03 Besprechung der Informationen
   - 04.01.03 Ideenfindung/Mindmap

4. Erst nach den Ferien dante lesen!

5. Zusätzliche Termine:
   - Mo. 6.01.03 - dante Praxis
   - Mi. 8.01.03 - dante Praxis

6. Probleme des Treffens:
   - keine

Protokoll

Datum: Freitag, den 27.12.02
Uhrzeit: 14.00 Uhr bis 16.00 Uhr
Anwesend: Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant: Ellermann Ronny

Thema des Treffens: Vergleich und weiteres Vorgehen

1. Vergleich bisheriger Informationen:
   - Wir haben unsere bisher gesammelten, teilweise schon bearbeiteten Informationen miteinander verglichen, und besprochen, welche wichtig und unwichtig sind

2. Weiteres Vorgehen:
   - Zum Schluss der Sitzung haben wir abgeklärt, wie wir weiter vorgehen möchten und was bis zur nächsten Sitzung alles ausgearbeitet werden soll

3. Probleme des Treffens:
   - keine

Protokoll

Datum: Donnerstag, den 02.01.03
Uhrzeit: 16:00 Uhr bis 18:30 Uhr
Anwesend: Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant: Dimpfl Stefanie

Thema des Treffens: Überblick über aktuelle Informationen

1. Klärung der vorhandenen Informationen:
   - Abgleichungen mit anderen Informationen, d.h. welche Informationen sind doppelt?
   - Welche Informationen sind besonders Wichtig?

2. Begriffsklärung: Asset-, Content- und Knowledgemanagement:
   - Begriffsdefinitionen den Gruppenmitglieder verständlich machen,

3. Aktueller Standpunkt, was muss die Gruppe noch erledigen?
   - Was hat die Gruppe schon erledigt?
   - Eingang auf die Ursprungsaufgabe von Herrn Scheib

4. Arbeitsplanerstellung:
   - Einteilung der Informationen in Themengebieten -> 1. Struktur
   - Aufteilung der Themengebiete nach Informationsmaterial an die entsprechenden Gruppenmitglieder

5. Festlegung der nächsten Treffen:
   - 05.01.03 und 07.01.03

6. Probleme des Treffens:
   - keine

Protokoll

Datum:	Dienstag, den07.01. 2003
Uhrzeit:	10.30 Uhr bis 11.15 Uhr
Anwesend:	Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant:	Erdle Sabrina

Thema des Treffens: Besprechung der weitere Vorgehensweise

1. Aufbau/Gestaltung des Projektskriptes:
   - Festlegung der Formatierung der Seiten und die Gestaltung des Inhaltsverzeichnisses

2. Festlegung eines neuen Termins für ein weiteres Treffen:
   - Bildkataloge für das Cover und den Text „Geschichtlicher Hintergrund“ (Antike) sind mitzubringen

3. Probleme des Treffens:
   - keine

Protokoll

Datum:	Freitag, den 10.01.03
Uhrzeit:	13.00 Uhr bis 15.00 Uhr
Anwesend:	Dimpfl Stefanie, Erdle Sabrina
Protokollant:	Dimpfl Stefanie

Thema des Treffens: Ausarbeitung der Musterseite

1. Aussuchen geeigneter Schriftfonts:
   - Probedrucke

2. Anordnen der Seitenbeschriftung:
   - lebender Kolumnentitel, Seitenzahl, Größe der Balken,...

3. Austausch von Informationen:
   - zu Content, Asset, Knowledge-Management
   - Pflege

4. Informationsmaterial via Internet suchen über:
   - dante, Zope, Wirtschaftlichkeit von CMS

5. Probleme des Treffens:
   - keine

Protokoll

Datum: Monatg, den 13.01.03
Uhrzeit: 8.00 Uhr bis 18.00 Uhr
Anwesend: Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant: Erdle Sabrina

Thema des Treffens: Einarbeitung in die zwei Content Management Systeme „dante und Zope“

1. Zope:
   - erstes Kennenlernen und Experimente mit der Navigation und Verwaltung des Systems
   - Einlesen in die Handhabung von Zope mit Hilfe eines „Online Zope-Handbuches“ (www.dzug.org, Zope Book)
   - Begriffsklärungen, der Reiter in Zope und den verschiedenen Objekten (Folder, ...)
   - Ausprobiert: - Objekte anzulegen, zu kopieren, einzufügen, auszuschneiden und zu löschen
   - erste Versuche, eine eigene Website aufzubauen, mit Hilfe eines internen Tutorial (=„vorgefertigtes Praxisbeispiel einer Website“)

2. dante:
   - einlesen im dante-Skript und erste Erfahrungen im Erstellen von Modulen, Ordnern und einzelnen Benutzern
   - es gab Probleme im Umgang mit den Rechten, es war keine Bearbeitung an den Objekten möglich

3. Probleme des Treffens:
   - erst durch die „Neue Rechtevergabe “, durch Herrn Scheib erhielten wir alle nötigen Rechte zur Bearbeitung und den kompletten Einblick in die dante-Struktur (Backoffice)

Protokoll

Datum:	Dienstag, den 14.01.03
Uhrzeit:	8.00 Uhr bis 20.30 Uhr
Anwesend:	Dimpfl Stefanie, Ellermann Ronny anwesend bis 13.30 Uhr, entschuldigt wegen Krankheit, Erdle Sabrina
Protokollant:	Erdle Sabrina

Thema des Treffens: Vertiefung und weiteres Ausprobieren der erworbenen System-Kenntnisse

1. Zope:
   - erster Versuch Texte in Html-Quellcode einzubetten und zu Formatieren
   - Klärung der verschiedenen „Reiter“ in Zope (z.B.: History, Edit, View, ...), durch ausprobieren und Verwendung des Zope-Handbuches

2. dante:
   - weitere Module angelegt und neue Icon-Zuweisung
   - Klärung der Rechtevergabe im Backoffice

3. Allgemein:
   - Vorbereitung für die Expertenbefragung am Mittwoch, den 15.01.03, durchanlegen eines Headers und einer Navigationsleiste in Photoshop 5.5, für den Bau einer Website
   - 13.00 Uhr bis 13.30 Uhr „Besprechung mit Herrn Scheib“
     Thema:
     - Klärung der Begriffe Datenbank und Datenbanksysteme
     - deren Arbeits- bzw. Funktionsweise, anhand einer Skizze zur Ver deutlichung
     - kurze Darlegung des Projektstandes und Klärung der Fragen zur weiteren Vorgehensweise

4. Probleme des Treffens:
   - keine

Protokoll

Datum: Mittwoch, den 15.01.03
Uhrzeit: 8.00 Uhr bis 21.20 Uhr
Anwesend: Dimpfl Stefanie, Erdle Sabrina
Protokollant: Erdle Sabrina

Thema des Treffens: Einarbeitung in die zwei Content-Management-Systeme dante und Zope

1. Zope:
   - Informierung über die Erstellung von Framesets, mit Hilfe eines Html-Handbuches
   - Strukturierung der Website in drei Frames
   - Einbettung eines Headers und einer Navigation in die Site
   - erste Experimente zur Verlinkung, mit selbstgeschriebenen Html-Texten

2. dante:
   - Erstellung von einzelnen Dummies, zur weiteren Erforschung des Submenüs und dessen Funktionen
   - Entschlüsselung der hierarchischen Gliederung des Submenüs

3. Probleme des Treffens:
   - erneute Probleme im Umgang und mit der Handhabung von dante

4. Allgemein:
   - ab 18.00 Uhr die Expertenbefragung mit Marcel Berdyn, über die Handhabung und Funktionsweise von dante

Protokoll

Datum: Donnerstag, den 16.01.03
Uhrzeit: 8.00 Uhr bis 21.30 Uhr
Anwesend: Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant: Erdle Sabrina

Thema des Treffens: Weitere Einarbeitung in die zwei Content Management-Systeme

1. Zope:
   - Strukturierung und Ausrichtung der Hauptnavigation
   - weitere Verlinkung der Website mit Unternavigationspunkten
   - schreiben des Protokolls
   - erneute Bearbeitung des Headers
   - Entwurf eines Stylesheets zur Formatierung der einzelnen Webseiten

2. dante:
   - Entschlüsselung der dante-Klassen
   - erstes eigenes Projekt in dante angelegt

3. Allgemein:
   - Verfassung von einzelnen Texten für das Projektskript
   - Gestaltung von einzelnen Schaubildern in Freehand

4. Probleme des Treffens:
   - das Frontend in dante funktioniert nicht
   - Fehlersuche mit Herrn Scheib
   - „Anonymous“ nicht auffindbar bzw. wieder herstellbar

Protokoll

Datum: Freitag, den 17.01.03
Uhrzeit: 8.00 Uhr bis 16.00 Uhr
Anwesend: Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant: Ellermann Ronny

Thema des Treffens: Weitere Einarbeitung in die zwei Content Management-Systeme

1. Zope:
   - Neue Formatierung von Stylesheets in Zope
   - Versuch ein Bild in Html-Quellcode einzubinden

2. dante:
   - In dante Redakteure angelegt und verschiedene Rechte verteilt

3. Allgemein:
   - Suche nach Schaubilder im Internet
   - Protokolle verfasst
   - Schaubilder in Freehand nachgebaut
   - Header für die Website für Zope in Photoshop bearbeitet
   - Bildbearbeitung in Photoshop

4. Probleme des Treffens:
   - In Zope gibt es Probleme, den eingepflegten Header im Internet anzuzeigen
   - Probleme mit dem Frontend in dante

Protokoll

Datum:	Samstag, den 18.01.03
Uhrzeit:	8.00 Uhr bis 13.00 Uhr
Anwesend:	Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant:	Ellermann Ronny

Thema des Treffens: Weitere Einarbeitung in die zwei Content Management-Systeme

1. Zope:
   - weiterer Bildeinbau in Html-Quellcode

2. dante:
   - in dante wurde heute nichts gearbeitet

3. Allgemein:
   - Protokolle auf Musterseite eingepflegt
   - Cover für Titelseite eingescannt
   - Verfassen eines Einleitungstextes für die Startseite im Web

4. Probleme des Treffens:
   - der Bildeinbau in Zope hat nicht funktioniert

Protokoll

Datum:	Montag, den 20.01.03
Uhrzeit:	8.00 Uhr bis 18.30 Uhr
Anwesend:	Dimpfl Stefanie anwesend bis 15.30 Uhr, entschuldigt, Ellermann Ronny, Erdle Sabrina
Protokollant:	Ellermann Ronny

Thema des Treffens: Projektabschlussphase

1. Zope:
   - der Bildeinbau in Zope hat funktioniert

2. dante:
   - Frontend funktioniert immer noch nicht

3. Allgemein:
   - Allgemeine Texte verfasst

4. Probleme des Treffens:
   - erneute Probleme mit dem Bildeinbau in Zope

Protokoll

Datum:	Donnerstag, den 23.01.03
Uhrzeit:	16.00 Uhr bis 21.00 Uhr
Anwesend:	Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant:	Ellermann Ronny

Thema des Treffens: Projektabschlussphase

1. Zope:
   - wurde nichts gearbeitet

2. dante:
   - wurde nichts gearbeitet

3. Allgemein:
   - Allgemeine Texte verfasst
   - Schaubilder gestaltet

4. Probleme des Treffens:
   - keine

Protokoll

Datum:	Freitag, den 24.01.03
Uhrzeit:	15.00 Uhr bis 21.30 Uhr
Anwesend:	Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant:	Ellermann Ronny

Thema des Treffens: Projektabschlussphase

1. Zope:
   - wurde nichts gearbeitet

2. dante:
   - wurde nichts gearbeitet

3. Allgemein:
   - Allgemeine Texte verfasst
   - Schaubilder gestaltet

4. Probleme des Treffens:
   - keine

Protokoll

Datum:	Samstag, den 25.01.03
Uhrzeit:	8.00 Uhr bis 17.00 Uhr
Anwesend:	Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant:	Ellermann Ronny

Thema des Treffens: Projektabschlussphase

1. Zope:
   - wurde nichts gearbeitet

2. dante:
   - wurde nichts gearbeitet

3. Allgemein:
   - Allgemeine Texte verfasst
   - Schaubilder gestaltet

4. Probleme des Treffens:
   - die Zeit wird allmählich knapp

Protokoll

Datum: Montag, den 27.01.03
Uhrzeit: 8.00 Uhr bis 21.15 Uhr
Anwesend: Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant: Ellermann Ronny

Thema des Treffens: Projektabschlussphase

1. Zope:
   - wurde nichts gearbeitet

2. dante:
   - das Frontend in dante ist wieder in Ordnung

3. Allgemein:
   - Allgemeine Texte verfasst
   - Schaubilder gestaltet
   - Inhaltsverzeichnis abgeändert

4. Probleme des Treffens:
   - wir stehen immer noch unter Zeitdruck

Protokoll

Datum:	Dienstag, den 28.01.03
Uhrzeit:	17.00 Uhr bis 21.15 Uhr
Anwesend:	Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina, entschuldigt
Protokollant:	Ellermann Ronny

Thema des Treffens: Projektabschlussphase

1. Zope:
   - wurde nichts gearbeitet

2. dante:
   - wurde nichts gearbeitet

3. Allgemein:
   - Allgemeine Texte verfasst
   - Schaubilder gestaltet
   - Cover in Photoshop gestaltet
   - wir haben begonnen unser Skript aufzubauen

4. Probleme des Treffens:
   - wir stehen immer noch unter Zeitdruck

Protokoll

Datum:	Mittwoch, den 29.01.03
Uhrzeit:	8.45 Uhr bis 11.30 Uhr und 15.15 Uhr bis 21.15 Uhr
Anwesend:	Dimpfl Stefanie, Ellermann Ronny, Erdle Sabrina
Protokollant:	Ellermann Ronny

Thema des Treffens: Projektschlussphase

1. Zope:
   - wurde nichts gearbeitet

2. dante:
   - wurde nichts gearbeitet

3. Allgemein:
   - Erfahrungsberichte geschrieben
   - Quellenverzeichnis angelegt
   - Projektarbeit fertiggestellt und ausgedruckt
   - Projektarbeit in 6-facher Ausgabe als Ringbuch gebunden

4. Probleme des Treffens:
   - keine mehr

8.3 W-Fragen

WAS?

- Was ist ein Content Management-System?
  Abgrenzung des Begriffs CMS von anderen:
- Was ist ein Content?
- Was ist ein Content Management?
- Was ist ein Content Management-System?
- Welche Kernkomponenten besitzt ein CMS?
- Gibt es eine Benutzerverwaltung und Gruppenverwaltung?

Genaue Erläuterung anhand der Beispiele dante und Zope

WER?

Projektgruppe:
Dimpfl Stefanie; Ellermann Ronny; Erdle Sabrina;

Projektleiter:
Herr Scheib

WARUM?

Mit dem Projekt versuchen wir der Allgemeinheit verständlich zumachen, wie leicht es ist eine dauerhaft aktuelle Site zu erstellen. Denn heutzutage findet man kaum noch aktuelle Sites im Web. Und die wenigen aktuellen Sites wurden von professionellen Agenturen erstellt.

WOZU?

Dieses Projekt dient dem Erlernen des professionellen Umgang mit Content Management-Systemen, damit man aktuelle, professionelle und wirtschaftlich beständige Websites selbst erstellen kann.

MIT WEM?

Wir erhalten Informationen von Herrn Marcel Berdyn, und von den Autoren der Bücher "Web Content Management" und "Professionelles Content Management".

FÜR WEN?

Dieses Projekt hilft dem "Laien" sich mehr in der Welt der Websites und deren Aufbau zurechtzufinden. Es unterstützt Firmen und sonstige Interessierte Personen.

WIE?

Zuerst sammelten wir verschiedene Informationen rund ums Projektthema. Dies gelang uns mithilfe des World Wide Web und der Landesbibliothek in Speyer.
Danach gab es eine intensive Lesephase, in denen wir uns zum Abgleich der Informationen trafen.
In der Intensivphase beschäftigten wir uns praxisnah mit dante und Zope.
Nach dieser Phase versuchten wir unser erlerntes Wissen schriftlich bzw. elektronisch festzuhalten. Dies nahm die meiste Zeit unseres Projekts in Anspruch.

WOMIT?

Unser einziges Hauptarbeitswerkzeug war der Computer. Wir benötigten lediglich noch einen danteserver sowie eine Zopeserver.
Für alle anderen Ausarbeitungen nahmen wir die Programme Quark und Freehand in Anspruch.

WANN?

Das Projekt begann Ende November, die Hauptintensivphase allerdings verlief über den Zeitraum vom 13.01.03 - 17.01.03.
Daraufhin folgte eine Arbeitsphase inklusive Unterricht vom 20.01.03 - 30.01.03. Wobei allerdings die Woche von 27.01.03 - 30.01.03 ohne Klassenarbeiten vonstatten ging. Der Abgabetermin ist der 30.01.03 um 16.00 Uhr!
Für weitere zeitliche Angaben beachten Sie bitte unseren Zeitplan.

WO?

Hauptsächlich fand unser Projekt in der BBS Neustadt, im Raum 217 statt. Im laufe der letzten Wochen waren wir manchmal bei Stefanie Dimpfl. Genaueres siehe Protokolle!

8.4 Ziele unserer Website

-> Projekt-Präsentation im Internet

-> Bereitstellung von Informationen über CM-Systeme, mit den zwei Fallbeispielen dante und Zope, für Interessierte und evtl. Fachleute

-> Lernen mit den CM-Systemen Zope/dante umzugehen

-> Vor- und Nachteile der beiden Systeme aufzeigen

-> den Sinn und Zweck von CMS-Systemen herauszufinden? Ob sie den Arbeitsaufwand wirklich reduzieren oder eher komplizierter machen?

8.5 Zeitplan

Treffen:

1. Freitag, den 27.12. 2002, um 14.00 Uhr (Ronny, LD)
2. Donnerstag, den 2.01. 2003. um 16.00 Uhr (Stefanie, Edenkoben)
3. Samstag, den 4.01. 2003, um 14.30 Uhr (Sabrina, Ramberg)
4. Montag, den 6.01. 2003, 8 Stunde, Arbeits-Daten auf den Server legen, eventuell dante - Praxis (Einarbeitung)
5. Mittwoch, den 8.01. 2003, Praxis - dante
6. Freitag, den 10.01. 2003, ab der 6. Schulstunde

Check-Points

1. Allgemeines zu CMS lesen, bis zum 27.12. 2002
2. Erste Zusammenfassungen zum Allgemeinen CMS, bis 2.01. 2003
3. Ideen zum Aufbau der Projektarbeit, Grafik und Bilder, bis 4.01.2003
4. dante - Skript lesen, bis spätestens 6.01. 2003 (1. Schultag)!!!

8.6 Arbeitsverteilungsplan

Stefanie Dimpfl:

Geschichtlicher Hintergrund:

- Digitalisierung der IT-Abteilung in den letzten 30 Jahren

Begriffsdefinitionen:

- Daten
- Information
- Content
- Asset
- Wissen
- Anwendungsformen der Internet- und Webtechnologien

Managements:

- Content
- Content Management
- Content Management-Systeme
- Asset Management
- Knowledge Management

Akzeptanz

- Einführung neuer Konzepte und Modelle
- Mitarbeiter im Bezug auf CMS
- Personal im Bezug auf CMS
- Kunden im Bezug auf CMS

Softwarebeispiele

- dante

Ronny Ellermann:

Architektur eines CMS

- Anatomie der Dokumente
- Content Lifecycle

Entwicklung von HTML zu XML

- HTML
- DHTML
- XML ergänzt HTML
- Aktueller Trend XHTML

Aktzeptanz

- Webpublishing

Softwarebeispiele

- dante

Sabrina Erdle

Geschichte:

- Inhaltlicher Hintergrund
- Weiterentwicklung im CMS-Bereich

Architektur eines CMS:

- Kernkomponenten
- Serverkonzepte

Softwarebeispiele:

- Zope

8.7 Expertengespräche

Expertengespräch im Internet mit Oliver Zschau

Oliver Zschau ist Initiator und Mitbegründer des Informationsportals Contentmanager.de, der wichtigsten deutschen Website zum Thema

Thema: Web Content Management-Systeme

1. Welche Faktoren muss ein Web Content Management-System bieten:
   „Die Kernfunktionalität ist das Abbilden des Publishingprozesses einer Website. Dazu gehören: Workflow-Funktionalitäten, Asset-Management, Schnittstellen für In- und Export von Inhalten, leichte Integrierbarkeit in bestehende IT-Strukturen. (...)
   Voraussetzungen:
   Verwendung von offenen Standards in der Programmierung und in der Speicherung des Contents.“[1]

2. Was ist bei der Einführung eines WCMS zu beachten und welche Arbeiten müssen zuvor erledigt werden?
   „Es muss klar definiert werden, welche Content-Typen verwaltet und in wie weit die Mitarbeiter in den Publishingprozess einbezogen werden sollen. (...)
   Technisch stellt sich die Frage der Integrierbarkeit und die Weiterverwendung von bestehenden Inhalten. Vorraussetzung dafür ist die interne Bereitschaft bestehende Prozesse zu ändern. (...)“[2]

3. Benötigt jeder Sitebetreiber ein Web Content Management-System?
   „ Ein WCMS wird dann sinnvoll, wenn abzusehen ist, dass die Zahl der Contents so zunimmt, dass sie mit herkömmlichen Mitteln nicht mehr zu verwalten sind, sowie die Zahl der Mitarbeiter, speziell der technisch nicht versierten (...).“3

4. In welchen Punkten ist der Know-how-Bedarf derzeit am größten?
   „Ganz oben stehen Verständnisprobleme mit der Thematik an sich. Die Frage nach der Auswahl des Systems wird häufig diskutiert. Hier ist eine gewisse Unsicherheit zu erkennen. (...)“[4]

5. Sind heutzutage nicht alle wichtigen Faktoren einer erfolgreichen Website bekannt?
   „ WCMS ist die Grundlage von effizientem Webpublishing und baut zwingend auf den Faktoren einer erfolgreichen Website auf. (...) Es besteht immer noch großer Nachholbedarf in grundliegenden Faktoren. Von der Broschüreware-Phase zur E-Business-Phase ist ein WCMS ein bedeutender Baustein. Sind aber die anderen Faktoren nicht gegeben, wird ein WCMS alleine nicht zum Erfolg führen, bzw. die E-Business-Phase wird nicht erreicht.“[5]

1-5 (http://www.content.manager.de/magazin/artikel_...)

Expertenbefragung mit Herrn Marcel Berydn

Herr Marcel Berydn ist Diplominformatiker mit dem Schwerpunkt Mediendesign. Er war längere Zeit bei a3design beschäftigt. Während dieser Zeit, hat er eine Internetsite erstellt.

Thema: Vorstellung der CMS-Dante-Software

1. Kurze Einführung:
   - in den danteaufbau (Frontoffice/Backoffice)
   - Erläuterung des Frontoffice/Backoffice

2. Erläuterung der Handhabung mit dante:
   - Gruppenverwaltung
   - Benutzerverwaltung
   - Rechtenvergabe
   - danteklassen
   - Module und Kindmodule

3. Praxisübung:
   - ein sneues Projekt anlegen
   - Beispiel anhand eines Unternehmen

8.8 Erfahrungsberichte

Ronny Ellermann

Nachdem Projekt Content Management-System kann ich von mir behaupten, dass es richtig war, an dieser Projektarbeit mitzuwirken. Zuvor hatte ich keine Ahnung, wie ein CMS aufgebaut ist, geschweige denn, wofür es eingesetzt wird und was man sich überhaupt darunter vorstellt.
Nachdem Herr Scheib das Thema CMS in unserer Klasse bekannt gab und erläuterte, um was es dabei geht, war es ein Thema, welches sich für mich sehr interessant anhörte. Zwei Wochen später habe ich mich dafür entschieden an diesem Projekt teilzunehmen. Zudem erfreute es mich, dass mit Stefanie Dimpfl und Sabrina Erdle zwei engangierte Personen mit mir zusammen in einer Gruppe waren.
Trotz das wir vor den Winterferien noch viele Klassenarbeiten in verschiedenen Fächern geschrieben hatten, haben wir schon recht frühzeitig mit der Informationsbeschaffung angefangen, d.h. mit der Internetrecherche, den Büchern aus der Landesbibliothek in Speyer und einige Computerzeitschriften.
Die Arbeitstage während und nach der Intensivphase waren bis zur Abgabe des Projektes war stets kraftraubend, da wir viele Tage von morgens 8.00 Uhr bis abends um 21.00 Uhr sowie an den Wochenenden fast pausenlos gearbeitet haben. Aber in dieser Zeit haben wir uns zum Ausgleich in verschiedenen Restaurants immer gut ernährt.
Des Weiteren fand ich es gut von den beiden anderen, dass sie auch an den zwei Tagen während der Projektintensivphase, an denen ich wegen einer Erkältung nicht mitarbeiten konnte, mir nicht böse waren und weiter zum Ziel hin gearbeitet haben.
Mit der Erarbeitung dieses Projektes war ich mit unserer gemeinsamen Teamarbeit sehr zufrieden, da wir damit auch wirklich etwas vernünftiges vorweisen können und ich dabei viel gelernt habe, vor allem in den Programmen QuarkXPress und Photoshop sowie das ich nun den Canon-Farbkopierer innen- und auswendig kenne. Einen Dank möchte ich noch an unseren Projektleiter senden, da er uns stets zur Verfügung stand, natürlich auch dem Hausmeister, der er abends immer recht geduldig war, bis wir die Schule verließen.

Stefanie Dimpfl

Ich habe mich für das Projekt CMS entschieden, weil mich interessierte wie man Sites Benutzergerecht zuschneidet. Denn in meiner damaligen Agentur, wusste dies Niemand. Schlecht war dies daher, da wir einen Auftrag genau in diese Richtung hatten. Also, mussten wir uns so gut wie möglich in die Software einlesen und den Auftrag umsetzen, was uns auch gelang.
Heute weis ich, dass CMS ein viel weitläufigerer Begriff ist und dass es nicht nur CMS sondern auch viele andere Management-Arten gibt. Mir hat dieses Projekt sehr gut gefallen und im Vergleich zum letzten Jahr, war dieses Projekt etwas weniger stressig. Wir wussten diesmal genau was auf uns zukommt und wir hatten uns gut vorbereitet. Mit Sabrina Erdle und Ronny Ellermann war die Teamzusammenstellung perfekt. Leider erwischte uns in der Intensivphase ein übler Grippevirus, sodass meine Kollegen kurzfristig, Gott sei dank an verschiedenen Tagen, ausfielen. Alles in allem würde ich ein solches Projekt immer wieder machen.
Ein Dank geht auch an unseren Projektleiter Herrn Scheib, der uns so gut wie möglich unterstützte, auch als ich den Anonymous abschoss.

Sabrina Erdle

Vor dieser Projektphase hatte ich diesmal überhaupt keine Bedenken, da ich ja so ungefähr wusste, was auf mich zukommen würde. Im Nachhinein musste ich jedoch feststellen, dass es doch weit mehr Kraft und Zeit erforderte, wie das erste Projekt. Auch ich habe mich aus Interessensgründen für das CMS-Projekt entschieden, obwohl ich noch nie zuvor etwas über CMS gehört hatte. In unserer Gruppe die nur aus drei Mitgliedern bestand fühlte ich mich von Anfang an pudelwohl und es gab immer etwas zu lachen. Wir teilten uns die Arbeit, was das Skript betrifft, ein und jeder konnte soweit es ging seinen Vorlieben und Interessen folgen. In der Intensivphase versuchte jeder so gut es ging sich in die Systeme einzuarbeiten. Leider konnten wir, aus Zeitgründen und den krankheitsbedingten Ausfällen, nicht beide Systeme kennenlernen. Wir entschlossen uns dazu, dass sich jeder in das System einarbeitet das ihn am meisten interessiert. Anfangs hatte ich extrem große Schwierigkeiten mich in das System einzuarbeiten und war fast am verzweifeln bis ab und zu kleine Lichter sich am Horizont auftaten. Als ich soweit "alles" verstand und einzelne Webseiten erstellen konnte, machte es mir riesigen Spaß. Je näher der Abgabetermin jedoch rückte desto mehr stellten wir fest ist noch zu erledigen, und somit waren die letzten Tage noch stressiger. Leider warf ich unsere Gruppe etwas zurück, da ich zwei Tage vor Abgabe an einer schlimmen Mittelohrentzündung erkrankte. Was sich jedoch nicht allzu viel auf das Projekt auswirkte, da Ronny und Stefanie doppelt soviel arbeiteten. Hiermit möchte Ich mich bei ihnen für diese tolle Zeit und die super Zusammenarbeit, vor allem für die Unterstützung als ich krank war bedanken (Ihr seid echt Spitze). Trotzdem will ich diese chaotische und zugleich lustige Zeit nicht missen, so gut wie in dieser Woche habe ich schon lange nicht mehr gegessen. Ich habe sehr viele Erfahrungen gesammelt und neue Kenntnisse speziell im Umgang mit QuarkXPress, Freehand und den CM-Systemen gesammelt.

Ein besonderes Lob möchte ich unserm Projektleiter, Herrn Scheib, ausprechen für sein Engagement und sein stets offenes Ohr für „kleine“ Probleme.

Gruppenerfahrungsbericht

Unsere Projektarbeit über CMS begann Mitte Dezember, zu dieser Zeit hatten wir noch viele Klassenarbeiten zu bewältigen. So konnten wir nicht all zu viel Zeit dem Projekt zukommen lassen. Trotzdem gelang es uns schnell Informationen zu beschaffen. Zum Teil erhielten wir diese aus Büchern, die wir uns aus der Landesbibliothek in Speyer ausgeliehen hatten. Des weiteren aus sämtlichen Sites aus dem Internet sowie Computerzeitschriften. Während dem Treffen vor den Ferien, beschlossen wir unsere Informationen bis zum 06.01.03 zu lesen. Dies war leichter gesagt als getan. Bei unserem Treffen am 02.01.03 stellten wir fest, dass wir unser Ziel bis zum 06.01.03 nicht erreichen werden. Dadurch wussten wir aber, dass wir unsere Anwesenheit in der Schule voll ausreizen werden müssen, d.h. dass wir selbst nach der Intensivphase bis abends um 21.15 Uhr am Projekt arbeiten werden. Im Laufe der Zeit kamen wir mit dem Projekt immer weiter voran. Es war aber jederzeit ein Wettrennen mit der Zeit, aber es gelang uns die Projektmappen rechtzeitig abzugeben. Kurz gesagt war es für uns alle ein erfolgreich abgeschlossenes Projekt, mit dessen Ergebnis wir hochzufrieden sind. Unser Team hat sich jederzeit, auch wenn wir oft lange arbeiten mussten, sehr gut untereinander verstanden. Hinzu kommt, dass wir in dieser Zeit sehr viel gelernt haben.
Ganz herzlich bedanken möchten wir uns bei unserem Projektbetreuer Herr Scheib, der uns immer zur Verfügung stand, wenn Probleme auftauchten.