



Seminar XML und Datenbanken

Arbeitsgruppe Datenbanken und Informationssysteme
Universität Kaiserslautern

**XML-basiertes Dokumenten-Management /
Content Management**

von

Andreas Kümpel
eMail: andreas.kuempel@gmx.de

12. Februar 2003

INHALT

	Seite
1. Einleitung	3
2. Dokumenten- und Content-Management-Systeme	4
2.4 Nutzenpotentiale	4
2.1 Begriffsbestimmung	5
2.2 Anforderungen	7
2.3 Architektur	9
3. XML-Anfragesprachen	10
3.1 Anforderungen	11
3.2 XML-QL	12
3.3 XQL	14
3.4 XXL	15
3.5 XIRQL	16
4. Produkte	18
4.1 MS Content Management Server	18
4.2 Zope	19
5. Ausblick	22
Referenzen	23

Abstrakt

Content Management. Wieder geistert ein neues Schlagwort durch die Unternehmen. Es verspricht die Lösung für die Bewältigung der immer größer werdenden Informationsfluten in den Firmen. Diese Ausarbeitung gibt einen kurzen Überblick über Dokument Management Systeme und Content Management Systeme. Da XML in vielem Content Management Systemen zum Einsatz kommt, wird hier auf einige XML-Anfragesprachen eingegangen. Die Softwareindustrie hat auf den Bedarf der Unternehmen nach Content Management Systemen erkannt und sofort reagiert. Es existieren mittlerweile mehrere hundert unterschiedliche Systeme, von denen zwei in dieser Ausarbeitung kurz vorgestellt werden.

1. Einleitung

In vielen Unternehmen verbringen Mitarbeiter bis zu 60% ihrer Arbeitszeit [2] mit der Informationssuche, seien es Geschäftsbriefe, Regelwerke oder Konferenzbeschlüsse. Ein sehr großer Teil dieser Informationen ist immer noch lediglich in schriftlicher Form auf Papier vorhanden und oftmals befinden sich sogar mehrere Kopien davon im Unternehmen im Umlauf. Die Verwaltung dieser Informationen erweist sich als äußerst schwierig, besonders dann, wenn mehrere Kopien eines Dokumentes in unterschiedlichen Versionen vorhanden sind. Auch der Schriftverkehr innerhalb eines Unternehmens kann sich auf diesem traditionellen Wege als zeitaufwendig herausstellen.

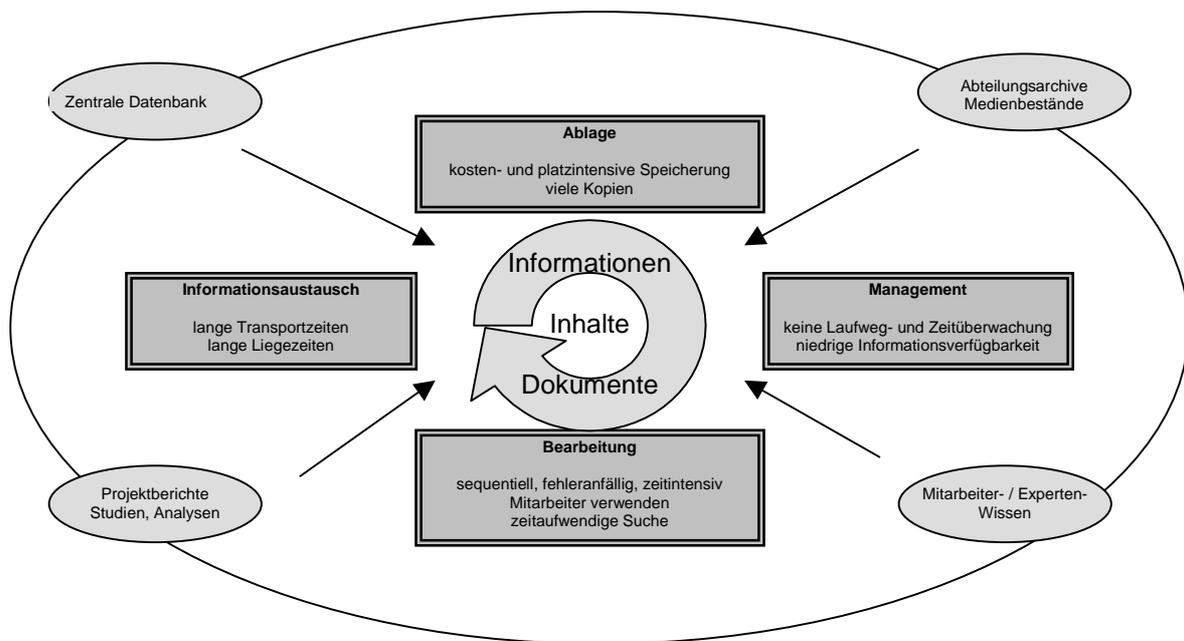


Abbildung 1: Ausgangssituationen in Unternehmen in punkto Informationen / Inhalten [5]

Die Abbildung 1 verdeutlicht eine Ausgangssituation in einem Unternehmen in Bezug auf Informationen und deren Verarbeitung. Informationen, Inhalte und Dokumente werden zentralen Datenbanken, Projektberichten, dem Mitarbeiter-Wissen oder Abteilungsarchiven entnommen und kursieren innerhalb eines Unternehmens. Vier Problembereiche treten dabei auf. Zum einen ist allein die Ablage der schriftlichen Inhalte kosten- und platzintensiv, oftmals existieren sogar in unterschiedlichen Abteilungen der Unternehmen mehrere Kopien. Zudem ist das Management von Inhalten in diesem Fall schwierig, Laufweg- und Zeitüberwachung sind kaum möglich. Sollen bereits vorhandene Inhalte bearbeitet oder als Quelle für neue Inhalte verwendet werden, so ist die Suche nach Inhalten, wenn diese nur in schriftlicher Form vorhanden sind, gerade in großen Archiven enorm zeitaufwendig. Sollen Informationen beispielsweise zwischen einzelnen Abteilungen ausgetauscht werden, sind lange Liegezeiten und Transportzeiten unvermeidbar.

Die Lösung bei der Behebung solcher Probleme sind sogenannte Dokumenten Management oder auch Content Management Systeme, Software, mit deren Hilfe Dokumente in elektronischer Form zentral gespeichert, verwaltet und publiziert werden können. Die Zunahme an elektronischen Brief- und Datenverkehr war in den letzten Jahren erheblich und wird auch in Zukunft immer mehr an Bedeutung gewinnen. Aus diesem Grund sind DMS oder CMS in immer mehr Bereichen einsetzbar, überall dort, wo ein hoher Informationsverkehr aufkommt.

Die Vorteile von DMS / CMS sind enorm, insbesondere für mittelständische und große Unternehmen. Neben der erheblichen Zeitersparnis bei der Suche nach Informationen und Dokumenten, bieten die Management-Systeme Vorteile, wie die sichere Speicherung, den Datenschutz, Zugriff über das

gesamte Intranet (oder Internet) für beispielsweise Außendienstmitarbeiter, oder die Unterstützung zur einfachen Darstellung von Dokumenten via Internet.

Diese Ausarbeitung soll einen kleinen Einblick in Dokument-Management-Systeme und Content Management Systeme bringen. Im zweiten Kapitel erfolgt zu aller erst eine Begriffsbestimmung von Dokument- und Content Management Systemen, eine Abgrenzung der beiden Systeme, sowie eine Beschreibung des grundlegenden Aufbaus solcher DMS / CMS und deren Anforderungen. Viele der heute verfügbaren CMS-Systeme benutzen zur Speicherung der Informationen und Dokumente das XML-Format (Extended Markup Language). Übliche Datenbank-Anfragesprachen reichen hier zur Suche in den Datenbeständen nicht mehr aus. Aus diesem Grund wird im dritten Kapitel auf einige XML-Anfragesprachen eingegangen. Die Software-Industrie hat den Bedarf von DMS und CMS erkannt. So existieren mittlerweile eine nicht unerhebliche Anzahl an solchen Systemen. Im vierten Kapitel werden zwei unterschiedliche Systeme kurz vorgestellt. Schließlich erfolgt im fünften Kapitel ein kurzer Ausblick auf die zukünftige Bedeutung von CMS.

2. Dokumenten Management Systeme / Content Management Systeme

2.1 Nutzenpotentiale

Die Nutzenpotentiale von CMS sind sehr umfangreich. Grob kann man sie in drei Klassen unterteilen: erstes Zeit, zweitens Kosten und drittens Qualität [5].

Zeit

Alleine durch die digitale Speicherung und den damit schnelleren Zugriff auf Inhalte verkürzt sich die Durchlaufzeit, d.h. die Transportzeit und die Zugriffszeit bei der Suche, erheblich. Zudem ist ein simultaner Zugriff auf die Inhalte möglich. Aufgrund der layout-neutralen Speicherung ist die Wiederverwendung und die Integration von bereits vorhandenen Inhalten in neue Datenbestände auch erheblich weniger zeitaufwendig. Aber auch der Zugriff auf die Daten selber führt zur Zeitersparnis, seien es einfache Zugriffe von Mitarbeitern innerhalb des Unternehmens oder Außendienstmitarbeiter, die über das Internet beispielsweise auf Konstruktionspläne oder Hilfen zugreifen können.

Kosten

Sicherlich sind die Anschaffungskosten für ein CMS relativ teuer. Es sind zwar sogar kostenlose CMS (z.B. „OpenCMS“, „Zope“, „Midgard“) verfügbar, allerdings dürften diese kaum den Bedarf an Funktionalitäten von mittelständischen bis großen Unternehmen decken. Enterprise Content Management Systeme sind für bis über 100.000 € verfügbar. Allerdings werden durch diese einmalige Anschaffung (wenn man von der Wartung und der Schulung des Personals absieht) erheblich Kosten gespart. So entfallen Kosten für Personal (Arbeitsstunden), die Archivfläche und die Sachkosten. Ist in dem CMS ein W-CMS (Web-based Content Management System, siehe Kapitel 2.2) integriert, so werden an dieser Stelle zusätzlich Kosten für die Erstellung der Internetpräsenz eines Unternehmens verringert.

Qualität

Ein weiterer positiver Aspekt der Verwendung eines CMS ist die Steigerung der Qualität. So kann (siehe Kapitel 2.3) eine Qualitätssicherung der zu erstellenden Dokumente sichergestellt werden. Ein Beispiel hierfür ist eine Broschüre einer Firma. Wurde sie bisher von einer beauftragten Agentur erstellt und vor dem Druck vom Lektorat kontrolliert und nachgebessert, so wird dies jetzt innerhalb sehr kurzer Zeit von Mitarbeitern des Unternehmens durchgeführt. Ein in einem CMS vorhandenes

Qualitätssicherungssystem könnte dazu dienen, vor der Veröffentlichung die Broschüre per Mausklick bei einem Qualitätssicherer vorzulegen und dort absegnen zu lassen. Aber auch im Bereich Dienstleistung kann sich ein CMS positiv auswirken. Durch effektiveren Zugriff auf die Daten kann die Auskunftsfähigkeit verbessert und die Reaktionszeit auf Anfragen verkürzt werden. Sind beispielsweise in einer telefonischen Hotline einer Firma sofort Anleitungen oder Hilfen verfügbar, kann einem Kunden unmittelbar Auskunft gegeben werden.

2.2 Begriffsbestimmung

Ein *Dokumenten Management System (DMS)* ist ein Softwaresystem, das in der Lage ist, binäre Daten (wie Dokumente oder Multimedia-Daten), zu verwalten und zu archivieren. Im Endeffekt scheint also ein DMS nichts weiteres zu sein als ein Archivsystem. Der Unterschied eines DMS gegenüber normalen Archivsystemen besteht jedoch darin, dass eine Versionsverwaltung von Dokumenten möglich ist. Einfache Dateisysteme der Betriebssysteme bieten lediglich nur wenige Kriterien zur Suche nach Dokumenten an. Meist sind dies der Dateiname, Dateityp und das Erstellungs- bzw. Änderungsdatum. Bestenfalls kann eine Suche nach Inhalten bei ASCII-Textdateien erfolgen. Wird ein Dokument gesucht, bei dem keines der oben genannten Kriterien genau bekannt ist, kann sich eine Suche danach schnell als zeitraubend erweisen.

Um eine etwas erweiterte Suche zu ermöglichen, wird bei Dokument Management Systemen zusätzlich zu den eigentlichen Dokumenten eine separate Datenbank mit Metadaten für diese Dokumente aufgebaut. Das bedeutet allerdings auch, dass bei jeder Speicherung von Dokumenten in der DMS zusätzlich der Benutzer diese Metadaten erst einmal eingeben muss, d.h. die Daten, die der Benutzer persönlich für relevant hält in Bezug auf das zu speichernde Dokument. Zur Definition von Metadaten stehen Metadaten-Modelle wie beispielsweise *Dublin Core Metadata* zur Verfügung. Danach ist jedoch eine strukturierte Suche nach Dokumenten mit Hilfe vorgegebener Suchbegriffe möglich. Ein oft für Dokument Management Systeme alternativer verwendeter Begriff ist der des *Dokumenten-Servers*. Allerdings liegt bei Dokumenten-Servern der Schwerpunkt auf der Suche in virtuellen Bibliotheken (Dokumenten-Servern) über das Internet.

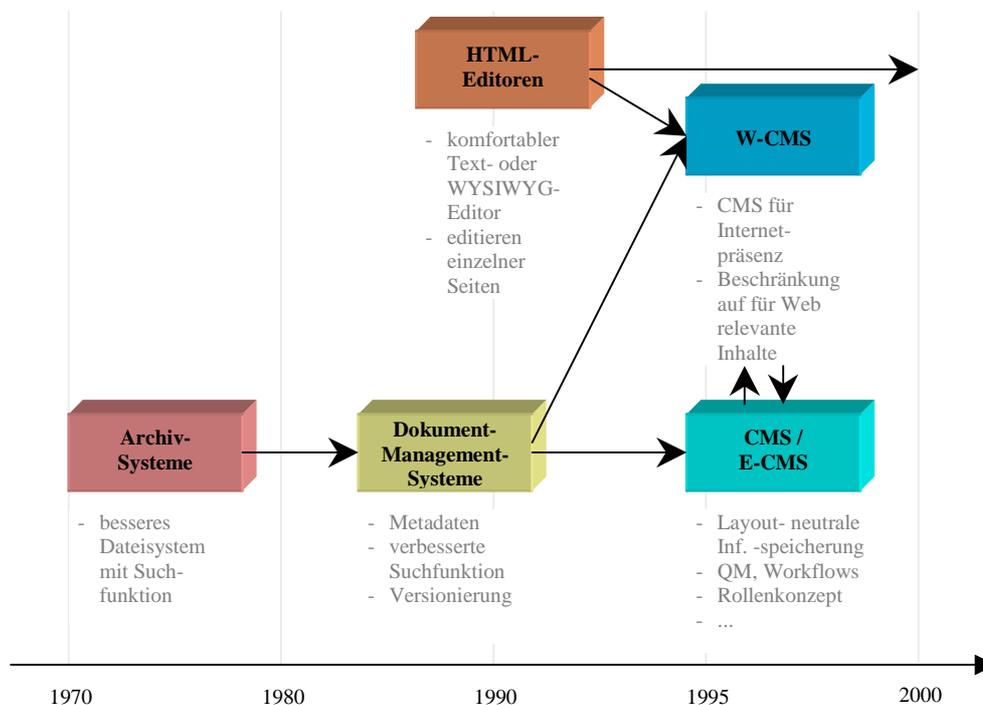


Abbildung 2: Zeitliche Entwicklung von DMS / CMS

Das Fraunhofer Institut für Arbeitswirtschaft und Organisation definiert den Begriff „*Content Management*“ wie folgt: [4]

„*Content Management* ist die Handhabung digitaler Informationen in allen Prozessen bzw. Prozessschritten von der Entstehung bis zur Distribution und Verwendung. Planung, Steuerung und Produktion von digitalen Inhalten und Inhaltselementen erfolgt dabei in der Form, dass eine bedarfs- und benutzergerechte Aufbereitung von Informationen aus unterschiedlichen Quellen für unterschiedliche Medien erfolgen kann.“

Kurz gesagt bieten Content Management Systeme (kurz CMS) eine Möglichkeit zur Verarbeitung für alle im Unternehmen relevanten Inhalten (engl. *Content*) an, wobei die Planung, Steuerung und Produktion von Inhalten seitens des CMS unterstützt wird. Diese Inhalte reichen von einfachen Produktinformationen, über den unternehmensinternen und -externen Briefverkehr bis hin zur Verwaltung von Rechnungen und Kundendaten.

Allerdings findet man auch andere Definitionen des Begriffs *Content Management*. So definiert die Forrester Research Group [28] den *Content Management* folgendermaßen:

Eine Kombination aus wohl-definierten Rollen, formalen Prozessen und unterstützenden System-Architekturen, die Unternehmen dabei unterstützen, Seitenelemente, wie Text, Grafiken, Multimedia und Applets zu erstellen und zu kontrollieren.

Nach dieser Definition dient ein *Content Management System* also in erster Linie dazu, den Web-Auftritt einer Firma im Internet wesentlich zu vereinfachen, so dass Inhalte ohne Kenntnisse des Webdesigns im Internet präsentiert werden können. An dieser Stelle ist der Begriff „*Web-Content Management System*“ (W-CMS) angebracht, um es gegenüber CMS abzugrenzen, die eher die Informationsverarbeitung und -präsentation *innerhalb* eines Unternehmens unterstützen. Leider wird diese Unterscheidung von vielen Softwarefirmen jedoch nicht vorgenommen, was dazu führt, dass viele Softwareprodukte, die als CMS-Produkte verkauft werden, nicht mehr sind als sehr komfortable Webseiten-Editoren. Mittlerweile taucht aber auch der Begriff „*Enterprise Content Management System*“ (E-CMS) auf. Dieser Begriff wird allen Anschein nach zunehmend für die *Content Management Systeme* im klassischen Sinne verwendet. Die Darstellung von Inhalten im Internet stellt für die E-CMS nur einen Teil des Einsatzbereiches dar. An dieser Stelle scheint es zu einer Namensverschiebung zu kommen, d.h. vom W-CMS zum CMS und vom CMS zum E-CMS. Die Bezeichnung „Enterprise“ soll verdeutlichen, dass das System „alle“ unternehmensinternen Inhalte verwalten sollen, nicht nur für den Internetauftritt relevante Inhalte.

Abbildung 2 verdeutlicht in etwa die zeitliche Entwicklung der unterschiedlichen Systeme, deren Einfluss aufeinander, sowie einige wichtige Eigenschaften dieser Systeme. Aus den ersten HTML-Editoren haben sich mittlerweile kleine W-CMS entwickelt. Eine exakte Abgrenzung von W-CMS und CMS / E-CMS erweist sich als schwierig, da von der Softwareindustrie diesbezüglich keinerlei Standards vorgegeben werden und es jeder Softwarefirma freisteht, ihr Produkt als CMS zu bezeichnen. Kennzeichen von W-CMS bleibt jedoch, dass die ganze Inhalts- und Informationsverarbeitung in erster Linie zum Ziel hat, neben der eventuell vorhandenen Möglichkeit zur unternehmens-internen Weiterverarbeitung, komfortabel, schnell und einfach eine Internetpräsenz zu erstellen und zu warten.

Der große Unterschied von Content Management Systemen gegenüber Dokument Management Systemen oder klassischen Archiv-Systemen besteht darin, dass das System weiß, wie der Inhalt eines Dokumentes strukturiert ist. Dieses „Wissen“ über die Struktur eines Dokumentes bietet eine Vielzahl von Vorteilen und ermöglicht es einem CMS eine Reihe von Funktionen zur Verwaltung von Inhalten zur Verfügung zu stellen. Abbildung 3 verdeutlicht einige wichtige Aspekte eines CMS.

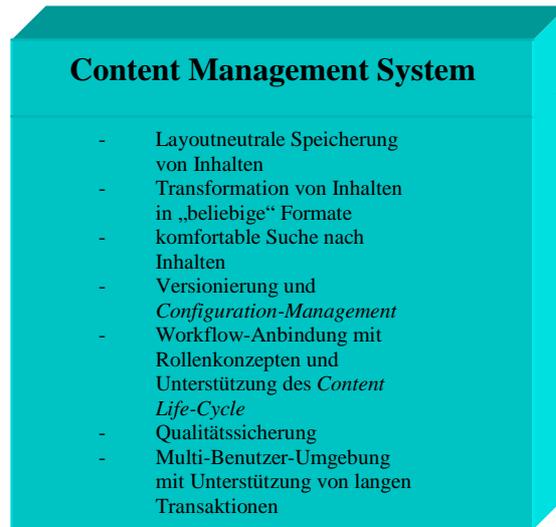


Abbildung 3: Wichtige Aspekte eines CMS

Ein wichtiges Kennzeichen von Content Management Systemen ist die Trennung von dem eigentlichen Inhalt bzw. dessen Struktur und der Darstellung. Um eine einfache Wiederverwendung von Inhalten zu ermöglichen, ist eine layout-neutrale Speicherung erforderlich. Die Sprache XML bietet sich an dieser Stelle an. XML bietet gerade für Content Management Systeme die Möglichkeit, die Inhalte der Dokumente bis auf die kleinste Inhaltskomponente aufzubrechen und ihren logischen Zusammenhang anhand einer Baumstruktur zu hinterlegen und so zu beschreiben. Die Struktur wird bei XML in der DTD („Document Type Definition“) bzw. im XML-Schema niedergelegt. Zur Bearbeitung eines Dokumentes braucht ein CMS lediglich diese DTD / Schema auslesen und entsprechende Felder und Funktionen zur Eingabe bzw. Änderung zur Verfügung stellen.

Auf diese Weise wird auch die Erstellung von Inhalten als HTML-Dateien (oder ganze Internet-Portale) oder PDF-Dateien mittels Templates, vordefinierte Grundgerüste für die Darstellung von Inhalten, vereinfacht. Wird beispielsweise ein Dokument von einem Server aus abgerufen, werden die entsprechenden Bestandteile des Dokumentes aus der Datenbank gelesen und in ein Template geladen, um so dem Nutzer dargestellt zu werden.

2.3 Anforderungen an CMS

Trennung von Inhalt und Layout

Innerhalb von Content Management Systemen werden nicht fertige Dokumente, sondern nur einzelne Bestandteile (XML mit Text, Bildern, Videos, etc.) gespeichert. Sobald der Benutzer beispielsweise ein bestimmtes Dokument abrufen, wird anhand der in XML abgespeicherten Struktur eine HTML-Seite erstellt oder es soll möglich sein, eine PDF-Datei dieses Dokumentes zu erstellen. Diese Möglichkeit setzt voraus, dass der Inhalt und das Layout eines Dokumentes getrennt werden.

Unterstützung des „Content-Lifecycle“ / Workflow-Anbindung

Eine weitere wichtige Anforderung an CMS ist die Unterstützung des Lebenszyklus (engl. *Lifecycle*) von Inhalten. Ein Content Management System soll nicht einfach nur zur Speicherung und Suche in bereits vorhandenen Inhalten dienen, sondern den gesamten Ablauf des Entwicklungsprozesses unterstützen, der mit der Erstellung der Inhalte beginnen soll.

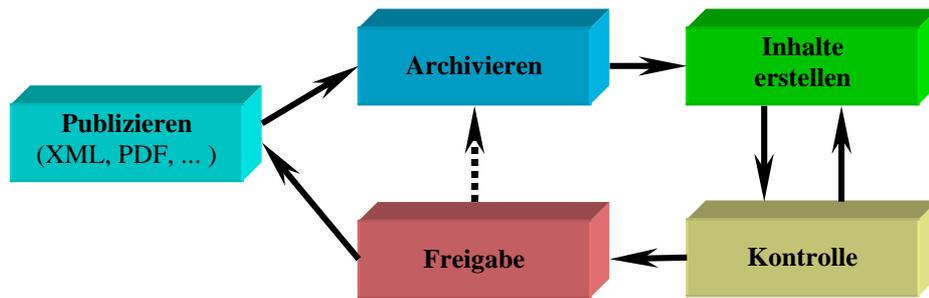


Abbildung 4: Content-Lifecycle [12]

Im ersten Schritt dieses Lebenszyklus (siehe Abbildung 4) wird der Inhalt erstellt. Als Quelle können bereits vorhandene Inhalte aus dem Content Management System sein (Archiv), aber auch gerade im Bereich von Web-Content Management Systemen sogenannte Angebote zur *Content-Syndication* [12]. Unter Content-Syndication versteht man die Verbindung von Inhalten unterschiedliche Websites, also Internetpräsenzen von Firmen oder Instituten. Gerade für die Aufwertung mit business- und branchenrelevanten Informationen ist dieses Angebot interessant, so können beispielsweise Börsenkurse oder aktuelle Nachrichten in die selbst zu erstellenden Inhalte eingebunden werden. Grundlage für das Content-Syndication ist das „Information-and-Content-Exchange-Protokoll“, kurz ICE. Das ICE verwaltet und automatisiert das Einrichten von Geschäftsbeziehungen zum Austausch von Informationen. Das ICE-Protokoll setzt auf XML auf. Es besitzt keine eigenen Datenformate, sondern eine ICE-Document-Type-Definition, die frei eingesehen werden kann. Ein Beispiel für solch einen Content-Anbieter ist die Firma iSyndicate [12].

Bei der Kontrolle wird der vorher erstellte Inhalt auf inhaltliche und ggf. gestalterische Korrektheit überprüft bzw. die Qualität gesichert. Das kann durch die Person geschehen, die den Inhalt selber erstellt hat, oder durch andere Mitarbeiter, einem Redakteur. Man spricht an dieser Stelle von dem 4-Augen-Konzept (siehe auch ISO 9000), d.h. einer Kontrolle durch den Autor und einen unabhängigen Redakteur. Wird der Inhalt von diesen zwei Personen akzeptiert, so kann der Inhalt freigegeben werden. Dieser Vorgang könnte durch ein CMS einfach unterstützt werden, indem gewisse Abläufe automatisiert werden. So könnte die Überprüfung eines Dokumentes auf das Einhalten vorgegebene Strukturen durch das CMS vorgenommen werden.

Nach der Freigabe des Inhalts wird dieser in ein gewünschtes Format konvertiert (z.B. wieder als XML-Datei). Handelt es sich bei dem Inhalt z.B. um eine aktualisierte Internet-Seite für das Internetportal einer Firma, so müssen ggf. neue Referenzen auf die Internet-Seite gesetzt werden, die Internet-Seite als HTML zusätzlich generiert und auf dem Webserver abgespeichert werden, zusammen mit den dazugehörigen Bildern und Grafiken.

Letztendlich werden die Informationen archiviert, wobei die Archivierung nicht unbedingt in demselben Format geschehen muss, wie bei der Publikation, da auf diese Weise eine Wiederverwendung von Inhalten stark vereinfacht werden kann. Beispielsweise kann bei einem W-CMS der eigentliche Inhalt im Archiv im XML-Format gespeichert werden, gleichzeitig aber auf dem Web-Server im mittels Templates generiertem HTML-Format.

Wünschenswert für ein CMS ist zusätzlich die Möglichkeit eigene Workflows für einzelne Inhalte oder ganze Projekte erstellen zu können. Gerade bei großen Redaktionsteams ist eine Aufgabenverteilung und ggf. eine Qualitätskontrolle durch gestaffelte Reviewzyklen erforderlich. Die digitalisierten Inhalte sollen dabei einen „vorgefertigten“ Weg durch das Unternehmen gehen. Ein Workflow ist eine Kette von Arbeitsschritten, die vom CMS gesteuert durchgeführt werden. Die Workflow-Engine realisiert diese Steuerung. Diese Workflow-Komponente eines CMS ist dann dafür verantwortlich, dass der zu erstellende Inhalt entsprechend eines vorher definierten Arbeitsablaufs

(Workflow-Modell) in der richtigen Reihenfolge an alle betreffenden Personen (Rollen) weitergeleitet wird. So können auf diese Weise auch Einhaltungstermine überwacht werden. Mittels einer Monitor-Komponente lässt sich dann der Weg eines Inhalts durch das Unternehmen verfolgen [27].

Rollenkonzepte

Unter Rollenkonzept wird die Benutzerverwaltung verstanden und die Regelung der Zugriffsrechte. Ein CMS soll es auch ermöglichen, dezentral zu arbeiten. Bei einer höheren Anzahl von Mitarbeitern (und Abteilungen) bietet es sich an, einzelnen Mitarbeitern verschiedene Rollen zuzuordnen (Redakteur, Designer, ...), die die Rechte besitzen, auf bestimmte Inhalte zuzugreifen, zu erstellen und freizugeben. Gerade bei der Verwendung von Workflows ist solch ein Rollenkonzept unverzichtbar. Aber auch in puncto Sicherheit kommt einem Rollenkonzept eine sehr wichtige Bedeutung zu. Ein Außendienstmitarbeiter sollte sicherlich nicht Zugriff auf Geschäftsbriefe der Firmenleitung besitzen und in der Lage sein, deren Inhalte auf der Internetseite der Firma zu veröffentlichen. Zudem könnte je nach Art des Benutzers das Benutzerinterface entsprechend gestaltet werden. Weiterhin sollte ein CMS die parallele Bearbeitung von Inhalten durch mehrerer Personen (einer Rolle) ermöglichen. Entsprechende Mechanismen zum Transaktionsschutz (lange Transaktionen) sind an dieser Stelle erforderlich, da oft mehrere Personen über längere Zeit parallel an Inhalten arbeiten.

Anbindung an Fremdsysteme

Ein CMS sollte nicht für sich isoliert arbeiten. So sollte es möglich sein, Fremdsysteme, seien es einfache Textverarbeitungssysteme, bis hin zu anderer im Unternehmen bereits eingesetzter Software (Bibliotheken, etc.), an das CMS anzubinden und zu integrieren. Solch eine Anbindung könnte durch einfache Speicherung von mit Fremdsystemen erstellten Inhalten im CMS bis hin zu einem Zugriff des CMS auf die von einem Fremdsystem verwalteten Inhalte realisiert werden.

2.4 Architektur von CMS

Die folgende Abbildung 5 veranschaulicht die allgemeine Architektur eines (W-) Content Management Systems. Grundlage des Systems sind die eigentlichen Daten oder Inhalte, die im Repository gespeichert werden. Bei dem Repository muss es sich nicht um eine einzige Datenbank handeln. Es können auch Dateisysteme oder andere Datenbestände hinzuzählen oder Inhalte aus dem Internet in das CMS mit eingebunden werden.

Kern des Content Management Systems ist die Verwaltungs-Komponente. Dort werden die wichtigen Aspekte eines CMS (siehe Abbildung 3) realisiert.

Unverzichtbar gerade bei großen CMS, die von vielen Autoren und Redakteuren parallel genutzt werden, ist eine Versionierung. Da bei CMS die Wiederverwendung eine große Rolle spielt und oftmals Referenzen auf bereits bestehende Inhalte und Dokumente gesetzt werden, muss ein CMS sicherstellen, dass ein Dokument auf einen Inhalt einer ganz bestimmten Version referenziert. Wird dieser Inhalt im Nachhinein von einem Autor weiter verändert, so muss eine neue Version des Inhalts angelegt werden. Andernfalls könnte ein Dokument auf einen Inhalt verweisen, der aufgrund seiner Änderung nicht mehr in den Kontext des Dokumentes passt.

Eine weitere wichtige Komponente ist die Workflow-Komponente, besonders im Zusammenhang mit der Qualitätssicherung. Wie bereits in Kapitel 2.3 erwähnt ist bei großen Redaktionsteams eine Aufgabenverteilung unbedingt notwendig. Über die Workflow-Komponente können eigene Workflows erstellt und die beteiligten Rollen bzw. Personen festgelegt werden. Die Komponente ist dann während dem Ablauf des Projektes dafür verantwortlich, den geplanten Workflow sicher- und den aktuellen Status darzustellen.

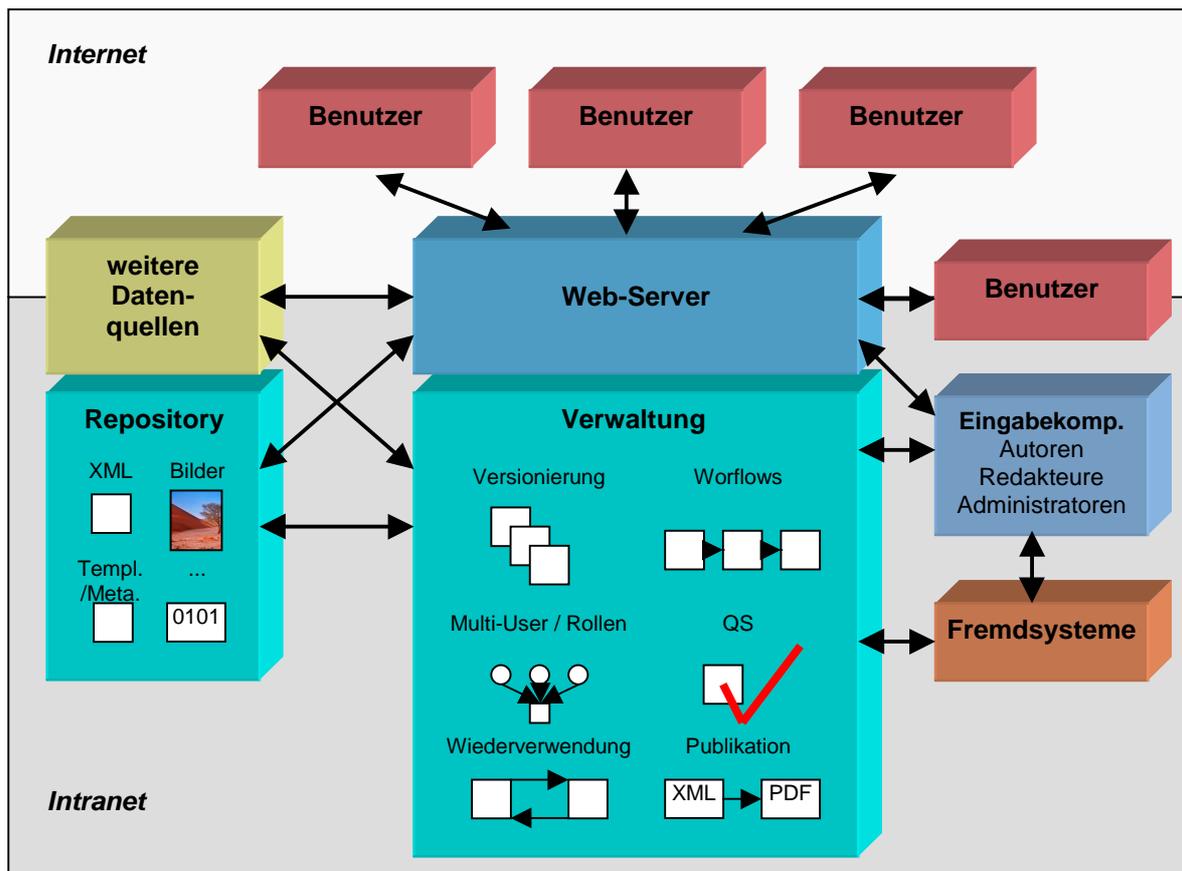


Abbildung 5 - Architektur eines (W-) CMS

Eng zusammen mit der Workflow-Komponente arbeitet die Mehrbenutzer- bzw. Rollenkomponente. Hier werden Rollen festgelegt und bestimmten Personen diese Rollen zugeteilt. Je nach Rolle bekommt ein Benutzer oder Autor ein unterschiedliches Benutzerinterface und Rechte auf den Zugriff von Inhalten und die Funktionalität des CMS zugewiesen. Zudem wird hier, zusammen mit der Versionierung, der Transaktionsschutz auf Dokumente für den simultanen Zugriff sichergestellt.

Die dritte große Komponente in einem CMS ist der Web-Server. Der Web-Server dient in erster Linie für den Zugriff auf die erstellten Inhalte seitens des (End-) Benutzers. Diese Komponente ist allerdings nur bei W-CMS zwingend erforderlich. Je nach System erfolgt der Zugriff der Autoren, Redakteure oder Administratoren ebenfalls über den Web-Server, was auch im Kapitel 4.2 vorgestellten Produkt Zope der Fall ist. Einige CMS unterstützen zusätzlich die Anbindung von Fremdsystemen. Ein CMS ist in der Regel Bestandteil einer ganzen IT-Infrastruktur in einem Unternehmen. Eine Anbindung an andere Systeme in dem Unternehmen ist also erforderlich.

3. XML-Anfragesprachen

Durch die Speicherung der Daten in den Datenbanken von DMS und CMS im XML-Format taucht ein Problem auf: Die Suche in solchen DMS und CMS. Übliche Anfragesprachen zum Suchen in Datenbanken, wie SQL oder OQL, erweisen sich bei der Suche hier als ungenügend oder gar unbrauchbar, da sie den Anforderungen (siehe 3.1) nicht entsprechen. So soll auch die Suche in im Internet verteilten Dokumenten möglich sein. Neue Anfragesprachen, die direkt auf die Datenstrukturen von XML-Dokumenten angepasst sind, sind erforderlich. Im Folgenden Kapitel werden einige solcher XML-Anfrage-Sprachen genauer erläutert.

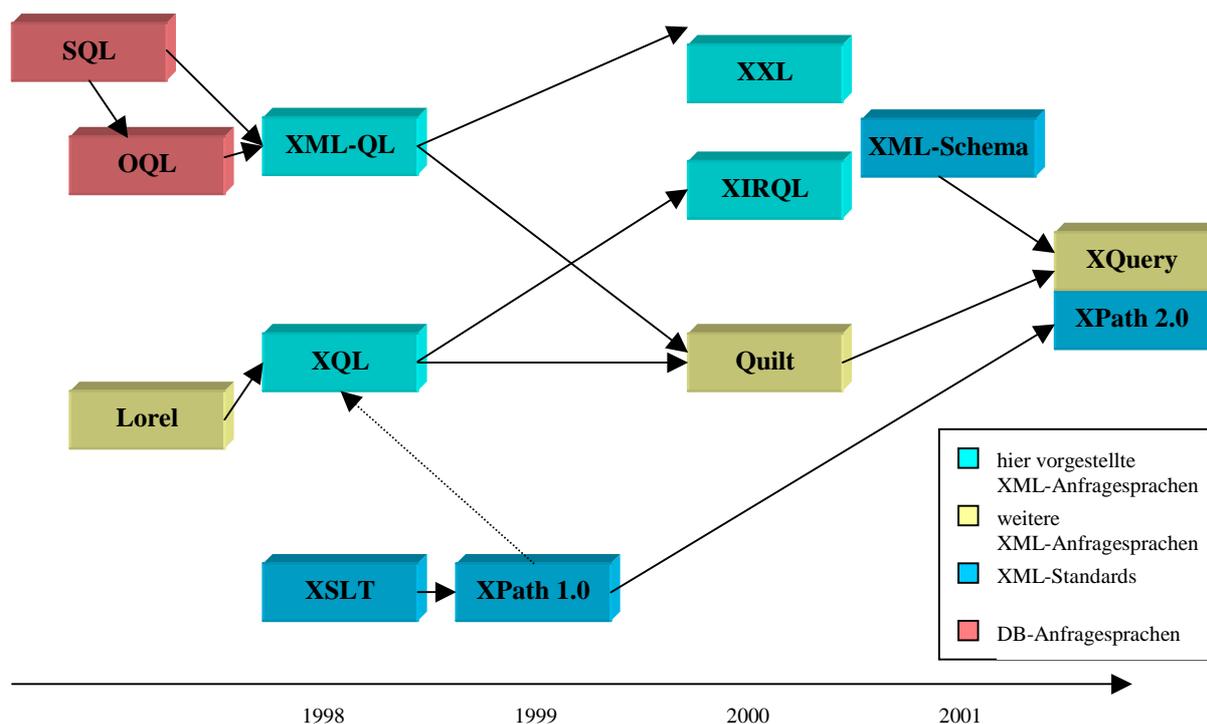


Abbildung 6 - Historische Entwicklung von XML-Anfragesprachen [u.a. 16]

Abbildung 6 zeigt die zeitliche Entwicklung und die Einflüsse der unterschiedlichen XML-Anfragesprachen, Datenbank-Anfragesprachen und XML-Standards. XQL stellt hier u.a. eine Erweiterung von XPath dar, wobei XPath aber erst 1999 vom W3C standardisiert wurde.

3.1 Anforderungen an XML-Anfragesprachen

Die Aufgaben von XML-Anfragesprachen bestehen nicht nur darin, das komplette XML-Dokument zurückzuliefern, welches die Bedingungen der Anfrage erfüllt. Es gibt auch eine Reihe von weiteren Anforderungen:

- Einbettung von XML: Anfragen lassen sich in XML einbetten und XML-Fragmente können in Anfragen enthalten sein [16].
- Suche in verteilten XML-Dokumenten: XML-Dokumente befinden sich nicht notwendiger Weise in einer Datenbank, sondern können sich im Intra- oder Internet befinden. Auch in diesen Dokumenten sollte eine Suche möglich sein.
- Suche auf XML-Dokumenten ohne und mit Schemata: XML-Daten sind nicht immer mit der dazugehörigen DTD (Document Type Definition) vorhanden. So sollte eine Anfragesprache auch diese XML-Daten durchsuchen können. Natürlich sollte eine Anfragesprache dem gegenüber auch ein vorhandenes Schema für XML-Daten nutzen können, um so die Daten schneller und effizienter durchsuchen zu können.
- Einfluss auf die Struktur der Ausgabe: Es ist selten sinnvoll auf eine Suchanfrage hin einfach nur das entsprechende XML-Dokument selber zurückzuliefern oder einfach nur einen bestimmten Teilbaum eines Dokumentes. Mit einer XML-Anfragesprache sollte es möglich sein, Einfluss auf die Struktur der zurückgegebenen Daten haben zu können. Wird

beispielsweise nach dem Begriff „Julius Cäsar“ gesucht, so sollte das Ergebnis unabhängig von der Struktur der Quelldokumente folgende Form haben:

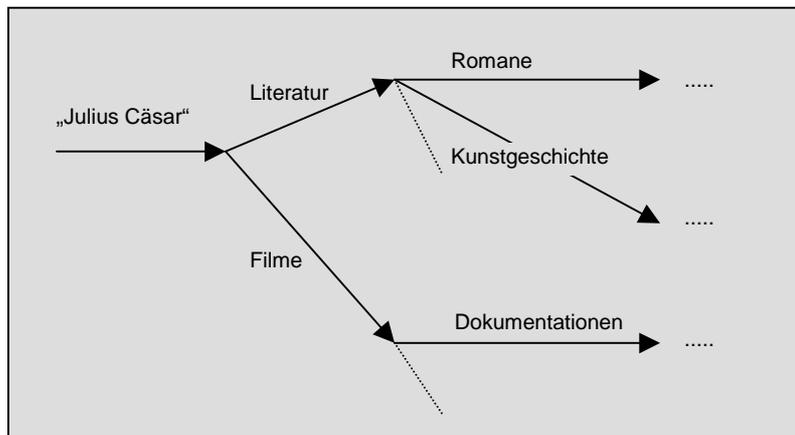


Abbildung 7 – Struktur der Ausgabe eines Anfrage

In diesem Beispiel (Abbildung 7) soll in einem Datenarchiv nach „Julius Cäsar“ gesucht werden. zurückgeliefert werden soll allerdings nur der Titel, Autor und Jahreszahl eines Dokumentes geordnet nach unterschiedlichen Quellentypen (Schriftliche Medien, Visuelle Medien).

3.2 XML-QL

Die Anfragesprache XML-QL (*XML-Query Language*) wurde im Rahmen eines AT&T Labs-Research-Projektes [18] entwickelt. Die Ziele beim Entwurf dieser Sprache waren u.a. folgende: [15]

- die Daten sollten aus großen XML-Dokumenten extrahiert werden können
- eine Konvertierung zwischen Daten im XML-Format und Daten aus relationalen oder objektorientierten Datenbanken sollte möglich sein
- die Sprache soll die Möglichkeit bieten, XML-Daten in andere XML-Daten (mit unterschiedlichen Schemata) zu transformieren
- Gewährleistung der Datenintegration aus vielen verschiedenen Quellen

Die folgende Abbildung 8 stellt eine Beispiel-DTD dar, auf die folgenden Anfragen und Darstellungen aufbauen:

```

<!ELEMENT book (author+, title, publisher)>
<!ATTLIST book year CDATA>
<!ELEMENT article (author+, title, year?, (shortversion|longversion))>
<!ATTLIST article type CDATA>
<!ELEMENT publisher (name, address)>
<!ELEMENT author (firstname?, lastname)>
  
```

Abbildung 8 - Beispiel DTD [18]

In dem Beispiel beinhaltet das zu durchsuchende XML-Dokument folgende Daten (Abbildung 9):

```

<bib>
  <book year="1995">
    <title> An Introduction to Database Systems </title>
    <author> <lastname> Weber </lastname> </author>
    <publisher> <name> Addison-Wesley </name > </publisher>
  </book>

  <book year="1998">
    <title> Foundation for Object/Relational Databases: The Third Manifesto </title>
    <author> <lastname> Weber </lastname> </author>
    <author> <lastname> Darwen </lastname> </author>
    <publisher> <name> Addison-Wesley </name > </publisher>
  </book>
</bib>

```

Abbildung 9 - Beispiel XML-Datei

XML-QL bietet für XML zwei verschiedene Datenmodelle an. Zum einen das *ungeordnete Modell*, zum anderen das *geordnete Modell*. [15]

Bei *ungeordneten Modell* wird die Reihenfolge der XML-Elemente nicht beachtet. Das XML-Dokument wird als Graph dargestellt. Von der Wurzel ab werden die Kanten mit den Namen der XML-Elemente gekennzeichnet. Die XML-Attribute werden den Knoten zugeordnet, wobei jeder Knoten einen sogenannten *object identifier* (OID) besitzt. Die Blätter des Graphen werden mit den Werten (Strings) belegt. Zu beachten ist dabei, dass es keine ausgehenden Kanten mit der gleichen Bezeichnung und dem gleichen Wert geben kann.

Im Gegensatz zum *ungeordneten Modell* ist das *geordnete Modell* ordnungserhaltend. Dazu werden bei dem Graphen Indexzahlen hinzugefügt. Die oben genannte Einschränkung des ungeordneten Modells ist damit aufgehoben. Bei der Suche mit XML-QL kann zwischen beiden Modellen gewählt werden.

Anfragen mit XML-QL

Die Anfrage in XML-QL ist eine **WHERE...CONSTRUCT...**-Klausel. In dem WHERE-Block wird die Auswahlbedingung und das XML-Dokument angegeben, in dem die Suche vorgenommen werden soll. Eine Suche in mehreren Dokumenten ist möglich. Im CONSTRUCT-Block wird die Struktur des zurück zu liefernden Ergebnisses der Suchanfrage festgelegt. Eine Beispielanfrage könnte wie folgt aussehen:

```

WHERE <book>
  <publisher><name>Addison-Wesley</></>
  <title> $t</>
  <author> $a</>
</> IN "www.a.b.c/bib.xml"
CONSTRUCT <result>
  <author> $a</>
  <title> $t</>
</>

```

Bei dieser Anfrage werden in allen vorhandenen Büchern, die vom Addison-Wesley-Verlag erschienen sind mit Autor und Titel aufgelistet. \$t steht hier als Variable für den Titel des Buches, analog steht \$a für den Namen des Autors. Diese Suchanfrage würde folgendes Ergebnis liefern:

```

<result>
  <author> <lastname> Weber </lastname> </author>
  <title> An Introduction to Database Systems </title>
</result>

<result>
  <author> <lastname> Weber </lastname> </author>
  <title> Foundation for Object/Relational Databases: The Third Manifesto </title>
</result>
<result>
  <author> <lastname> Darwen </lastname> </author>
  <title> Foundation for Object/Relational Databases: The Third Manifesto </title>
</result>

```

Bei dieser Anfrage wird deutlich, dass für jeden Autor eines Buches, das Buch selber mehrmals separat aufgelistet wird. Eine alternative Suchanfrage bietet ein übersichtlicheres Ergebnis:

Suchanfrage:

```
WHERE <book>
  <title> $t </>
  <publisher><name>Addison-Wesley </> </>
</> CONTENT_AS $p IN "www.a.b.c/bib.xml"
CONSTRUCT <result><title> $t </>
  WHERE <author> $a</> IN $p
  CONSTRUCT <author> $a</>
</>
```

Ergebnis:

```
<result>
  <title> An Introduction to Database Systems </title>
  <author> <lastname> Weber </lastname> </author>
</result>
<result>
  <title> Foundation for Object/Relational Databases: The Third
  Manifesto </title>
  <author> <lastname> Weber </lastname> </author>
  <author> <lastname> Darwen </lastname> </author>
</result>
```

Der Inhalt des passenden Elementes wird in der Variable \$p gespeichert, die in dem CONSTRUCT-Block erneut ausgewertet werden kann.

3.3 XQL

Die Anfragesprache XQL (*XML Query Language*) wurde 1998 von J. Robie, J. Lapp und D. Schach entwickelt [19] mit dem Ziel eine Sprache mit einer einfachen Syntax zur Suche, Adressierung und Filtern von XML-Daten zur Verfügung zu stellen. U.a. wurden folgende Entwurfs-Ziele vorgegeben:

- XQL-Strings sollten kompakt sein, leicht zu lesen und schreiben
- XQL sollte einfach zu parsen sein
- XQL sollte als String ausgedrückt werden, um sie beispielsweise in URLs zu integrieren

XQL besitzt kein eigenes Datenmodell. Es benutzt XML und dessen implizite Baumdarstellung [15].

Anfragen mit XQL

Anfragen werden in XQL als String formuliert. Im Gegensatz zu XML-QL besitzt XQL keinen CONSTRUCT-Block, so dass viele Anfragen nicht möglich sind (bspw. die Rückgabe der Ergebnisse als neue XML-Dokumente). Zudem kann XQL die Struktur des Eingabedokumentes nicht ändern, es unterstützt keine Variablen und die Ausgabedaten können nicht sortiert werden. Ebenfalls können mit XQL keine Verschachtelungen von Anfragen vorgenommen werden [17]. Die Selektion aus einem XML-Dokument erfolgt durch die Navigation der Elemente mittels eines Pfades.

Grundsätzlich hat eine Anfrage mit XQL folgendes Format :

pfad-ausdruck[filter]

Beim **pfad-ausdruck** wird der Pfad in ein Dokument spezifiziert, während **filter** die Bedingungen beschreibt, die an den Knoten gestellt werden. Die folgenden Beispiele sollen dies verdeutlichen. Es sei folgendes XML-Dokument (Abbildung 10) vorgegeben:

```

<filme>
  <film datum="01.05.1977">
    <titel>Georges Science Fiction Saga, Teil 4</titel>
    <regie>George L.</regie>
    <produzent>...</produzent>
    <genre>Science Fiction</genre>
    <beschreibung>...</beschreibung>
  </film>
  <film datum="01.05.2001">
    <titel>Planet der Affen</titel>
    <regie>Tim Burton</regie>
    <produzent>...</produzent>
    <genre>Science Fiction</genre>
    <beschreibung>...</beschreibung>
  </film>
</filme>

```

Abbildung 10 - Beispiel XML-Datei

<i>XQL-Anfrage</i>	<i>Bedeutung</i>
<code>/filme[/filme/film/regie='George L. ']</code>	Liefert alle Filme ab dem Knoten „filme“, bei denen George Lucas Regie geführt hat.
<code>/filme/film/titel[//regie='George L. ']</code>	Liefert die Titel aller Filme, bei denen George Lucas Regie geführt hat.
<code>//film[genre=//film[titel='Planet der Affen']/genre]</code>	Liefert alle Filme, deren Genre dasselbe ist, wie das Genre des Films „Planet der Affen“.
<code>//film/@datum[//film/titel='Planet der Affen ']</code>	Liefert das Attribut Datum des Filmes „Planet der Affen“.

Da XQL direkt auf XML aufsetzt und eine einfache und kompakte Syntax besitzt und die Möglichkeit, eine Suchanfrage als einfachen String darzustellen, ist diese Anfragesprache besonders für Internet-Anwendungen, beispielsweise auch für im Internet verfügbare Datenbanken, interessant, da eine Suchanfrage mit in der URL angegeben werden und bei der dynamischen Generierung eines Dokumentes ausgewertet werden kann. Genaueres zu XQL befindet sich unter [19].

3.4 XXL

Ebenso wie bei XIRQL unterstützt die Anfragesprache XXL („*Flexible XML Search Language*“) [20] *Ranked Retrieval* (Bewertung von Dokumenten nach ihrer Relevanz bezüglich der Suchanfrage) und nicht das *Boolesche Retrieval* (einfaches Durchsuchen von Dokumenten nach Zeichenketten). An dieser Stelle soll nur kurz anhand von Beispielen auf die Sprache eingegangen werden. Eine Ausführliche Beschreibung dieser Sprache ist in [21, 20] zu finden. Die Suchanfragen beziehen sich auf die Beispiel-XML-Datei aus Abbildung 10 (Kapitel 3.3).

Eine Suche nach Filmen, die als Genre das Stichwort „Fantasy“ besitzen, sieht wie folgt aus:

```

Select P
From http://mymoviecollection.xyz/movies.xml
Where Filme.Film As P
And P.# LIKE „%Fantasy%“

```

Die fett gedruckten Wörter sind reservierte Schlüsselwörter der Sprache, Großbuchstaben stehen für Variablen (in diesem Falle „P“), die an Knoten gebunden werden. Das Zeichen „#“ ist ein Wildcard-Platzhalter für beliebige Pfade, Punkte stehen für die Pfadkonkatenation.

Die oben genannte Beispielanfrage würde in diesem Fall zu einem leeren Ergebnis führen, da nirgendwo Fantasy explizit in der XML-Datei erwähnt wird, obwohl man Science Fiction durchaus als

Fantasy zählen lassen kann. Für eine semantische Ähnlichkeitsanalyse (z.B. „Fantasy oder ähnliches“), die eine nach Relevanz absteigend sortierte Rangliste von Treffern als Ergebnis liefert, muss die XML-Suchmaschine mit Eigenschaften einer Text-Retrieval-Maschine ausgestattet werden [20]. Zu diesem Zwecke wurde XXL mit einem sogenannten „Ähnlichkeitsoperator“ (Zeichen „~“) ausgestattet. Diese Ähnlichkeitsvergleiche können sowohl für Element- und Attributsinhalte, als auch für Element- und Attributnamen angewandt werden (z.B. „... **Where** Region.~Ort ...“). Die oben gestellte Anfrage würde dann wie folgt abgeändert:

```

Select P
From http://mymoviecollection.xyz/movies.xml
Where Filme.Film As P
And P.# ~ „Fantasy“

```

Zur Ähnlichkeitssuche steht eine Begriffshierarchie als Ontologiegerüst zur Verfügung (Abbildung 11). Solch ein Gerüst ist ein Baum mit einer Knotenmenge, den Begriffen (z.B. Elementnamen) und einer Kantenmenge. Die von einem Knoten ausgehenden Kanten zu den Kindern sind geordnet. Die Ordnung wird durch die Kantengewichte ausgedrückt und gibt die semantische Ähnlichkeit zwischen Geschwistern an. In dem folgenden Beispielbaum ist *Schwimmen* mit *Schnorcheln* enger verwandt (Abstand 1) als mit *Surfen* (Abstand 6).

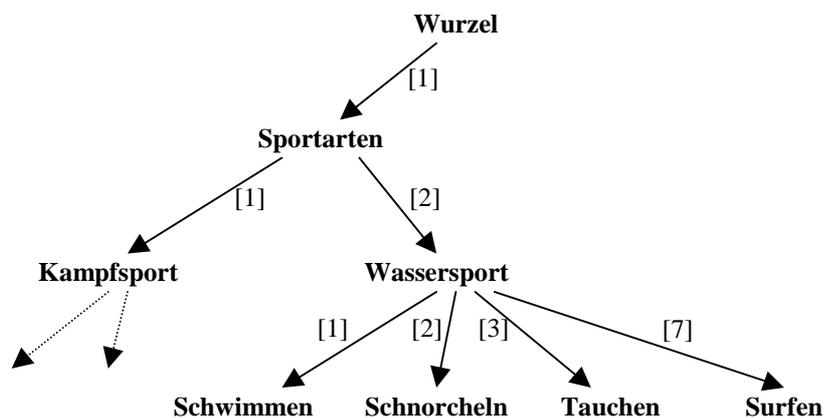


Abbildung 11 - Ausschnitt einer Ontologie [20]

Eine genaue Definition eines Ontologiegerüsts ist [20] zu entnehmen.

3.5 XIRQL

Bei der Anfragesprache XIRQL (XML Information Retrieval Query Language) [22] wurde die in Kapitel 3.3 vorgestellte Sprache XQL um einige Aspekte erweitert, mit dem Ziel *Information Retrieval* auf XML-Dokumenten zu ermöglichen.

Datentypen und vage Prädikate

Oftmals sind bei der Suche nach spezifischen Daten die genauen Parameter nicht exakt bekannt. Ein Beispiel dafür wäre die Suche in einer Enzyklopädie: „Gib mir Informationen über einen Politiker, die so ähnlich wie Manela heißt und der um 1950 Mitglied des African National Congress war.“ Diese Anfrage sollte natürlich alle Artikel über Nelson Mandela in der Enzyklopädie auflisten. Dass bedeutet, dass für die unterschiedlichen Datentypen (z.B. Personennamen, Orte oder Zeitangaben)

vage Prädikate benötigt werden. Neben der Ähnlichkeit (vage Gleichheit) sollten auch andere datentypen-spezifische Vergleichsoperatoren („klingt wie“, „</>“, „in der Nähe von“, ...) unterstützt werden. Je nach Ergebnis des Vergleichs zwischen der Anfrage und dem im Dokument vorgefundenen Daten, sollen die Prädikate ein bestimmtes Gewicht zurückliefern. Um dies zu ermöglichen werden neue Datentypen eingeführt. Ein Datentyp besteht aus einer Domäne und einer Reihe von vagen Prädikaten [22]. Dazu werden die Datentypen in einem XML-Dokument mittels XML-Schemata neu definiert.

Gewichtung

Jeder, der schon das Internet nach Dokumenten zu bestimmten Themen gesucht hat, wird das Problem kennen: Auf der Suche nach Informationen beispielsweise über die XML-Anfragesprache „XQL“ wird man zwar von der Suchmaschine von Ergebnissen geradezu überschwemmt, allerdings wird dabei jedes Dokument aufgelistet, in dem irgendwo die Bezeichnung „XQL“ vorkommt, sei es im Titel des Dokumentes oder in der Referenzliste am Ende des Dokumentes. In XQL könnte solch eine Anfrage auf eine XML-Datei (mit unterschiedlichen Artikeln als Inhalt) wie folgt aussehen:

```
/document[.//heading cw "XQL" and .//section/* cw "XQL"]
```

Diese Anfrage („cw“ steht für *contains word*) durchsucht den Titel und den Inhalt eines Dokumentes nach dem Begriff „XQL“. An dieser Stelle sind für uns aber Dokumente, bei denen „XQL“ im Titel vorkommt, von höherer Relevanz. Dies wird realisiert, indem die Knoten eines XML-Dokumentes unterschiedliche Gewichtungen bekommen. Die gewünschte Gewichtung könnte dann in eine Anfrage eingebaut werden:

```
/document[0.6 * .//heading cw "XQL" + 0.4 * .//section/* cw "XQL"]
```

Relevanz-orientierte Suche

Wenn bisher bei der Suche nach Inhalten der Type des zurückzuliefernden Elementes angegeben werden muss, so soll es nun möglich sein, lediglich den gewünschten Inhalt anzugeben. Das IR System soll selber in der Lage sein, die für uns relevanten Teile eines XML-Dokumentes zurückzuliefern, d.h. der Benutzer (bzw. Anwendung) muss nicht unbedingt genaue Kenntnis über die Struktur des zu durchsuchenden Dokumentes haben.

Structural relativism

XIRQL unterscheidet bei der Suche nicht zwischen Attributen und Elementen. D.h. XIRQL ignoriert leichte Variationen bei XML-Dokumenten. So kann das Datum eines Artikels im Titel als Attribut eingetragen sein, oder als eigenständiges Element in der XML-Datei.

Hyperlinks

Eine große Erweiterung gegenüber XQL ist die Unterstützung von Hyperlinks. So kann mit XIRQL nach Dokumenten gesucht werden, die Hyperlinks auf andere bestimmte Dokumente besitzen.

```
/artikel/zitate/zitat => /artikel[autor="Prof. Jones"]
```

Diese Beispielfilter für eine Suchanfrage sucht nach Artikeln, in denen auf ein Artikel von „Prof. Jones“ referenziert wird.

Die genaue recht umfangreiche Syntax kann [22] entnommen werden. XIRQL wurde mittlerweile in dem Open Source Prototypen „HyREX“ (Hypermedia Retrieval Engine for XML) implementiert.

4. CMS - Produkte

Nach meinen letzten Recherchen sind momentan mehr als dreihundert verschiedene CMS-Produkte auf dem Markt verfügbar. Die Unterschiede zwischen den einzelnen Produkten schwanken enorm, nicht nur von ihren Funktionsumfang, sondern auch von ihren Preisen. Angefangen von kostenlosen Produkten (wie Zope, OpenCMS oder Midgard) bis hin zu E-CMS wie Vignette, deren Kosten im sechsstelligen Euro-Bereich liegt. Im folgenden werden zwei ausgewählte CMS-Produkte kurz näher erläutert.

4.1 MS Content Management Server

Ein sehr bekanntes Content Management System kommt vom Microsoft [26]. Beim MS Content Management Server handelt es sich um ein komfortables W-CMS-System, mit Workflow-Unterstützung und Rollenkonzepten.

Die Benutzer des Systems werden grob in vier Gruppen eingeteilt: Seiten-Entwickler, System-Administratoren, Seiten-Manager und der Seiten-Autor. Den unterschiedlichen Rollen werden dabei unterschiedliche Benutzer-Interfaces zur Verfügung gestellt. Die folgende Abbildung 12 zeigt die Struktur des MS CMS.

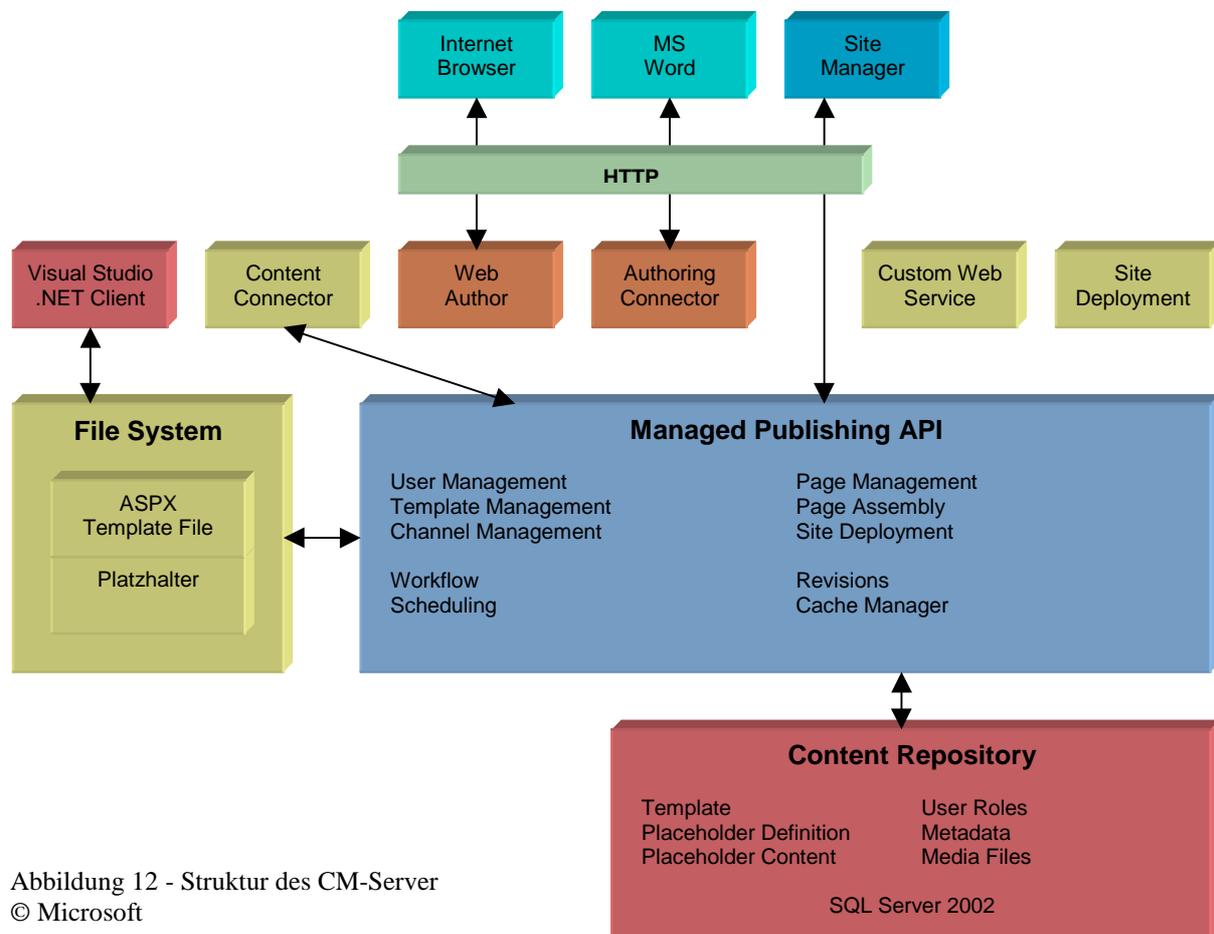


Abbildung 12 - Struktur des CM-Server
© Microsoft

Im folgenden werden einige Komponenten des Systems näher erläutert.

Web Author: Die Komponente Web Author wird zusammen mit dem Authoring Connector vom Autor einer Seite verwendet. Hier können Autoren neue Seiten erstellen, vorhandene editieren und Webseiten veröffentlichen. Über die browser-basierte *Web Author Console* können auch mehrere Autoren parallel an unterschiedlichen Bereichen der Seite arbeiten. Diese Konsole kann mittels Änderung an Templates auf die Bedürfnisse eines Autors speziell angepasst werden.

Authoring Connector: Mit Hilfe eines *Authoring Connector Wizards* kann ein Autor Inhalte mit MS Word erstellen und in die entsprechende Seite einbinden. Dabei wird der Inhalt nicht selber in der Webseite gespeichert, sondern ein Platzhalter.

Site Manager: Der Site Manager ist ein Werkzeug für Administrationszwecke verwendet. An dieser Stelle wird die sichtbare Struktur der Webseite festgelegt, sowie die Rechte an die unterschiedlichen Benutzer verwaltet. Mit dem Site Manager können auch sogenannte *Container* erstellt werden, mit deren Hilfe eine Hierarchie in eine Webseite integriert werden kann. Jedem Container können dann unterschiedliche Benutzer zugeteilt werden.

Die folgende Liste stellt nur einen kleinen Teil des Funktionsumfangs des MS CMS dar [26,11].

- Versionierung und Archivierung
- flexible und anpassbare Workflows
- Einbindung von Inhalten in mehrere Layouts durch „Platzhalter“
- Template- und Ressourcengallerien
- dynamisch erzeugbare *Sitemap* (grafische Struktur-Übersicht der Webseiten)
- .NET-Integration
- Erstellung und Einbindung von Web-Services
- XML-Unterstützung (XML, XSLT, XSD)

Eine Test-Version des MS Content Management Servers kann unter [26] kostenlos heruntergeladen werden. Voraussetzung für die Installation ist MS Windows 2000 Server.

4.2 Zope

Zope ist ein Web-Content Management System, ein Framework, um Web-Applikationen zu entwickeln. Der Web Application Server wurde von der Firma Digital Creations zuerst als kommerzielles Produkt entwickelt, wurde aber dann 1998 in ein OpenSource-Projekt umgewandelt, dass bedeutet, es kann kostenlos auf der offiziellen Seite [23] heruntergeladen werden (binäre Installationsdateien oder Source-Code) und ist Lizenz- und gebührenfrei, ebenso wie die Dokumentation. Seit der Umwandlung zu OpenSource hat dieses Projekt eine große Entwickler-Gemeinde gewonnen, wodurch viele Zusatzprodukte (*Zope Products*) entwickelt wurden. Zope wurde größtenteils in der objektorientierten Script-Sprache Phyton geschrieben, einige zeitkritische Teile in C [9, 23].

Um die von Zope verwendeten Inhalte zu speichern, wird keine externe Datenbank benötigt. Zope verfügt über eine eigene Objekt-Datenbank, kurz *ZODB*. Jeder Objekt-Typ (Text, Grafik, etc.) ist durch eine Klasse implementiert. Dadurch können bestehende Objekt-Typen einfach erweitert werden. Mittels Vererbung können neue Objekt-Typen von bestehenden abgeleitet werden. Alternativ zu der Zope eigenen Objekt-Datenbank bietet das CMS jedoch auch die Möglichkeit, auf bereits existierenden Datenbanken aufzusetzen, wie Oracle, MySQL und vielen anderen [23]. Modifikationen im CMS können in Transaktionen gepackt werden und so im Fehlerfall wieder zurückgesetzt werden.

Die komplette Steuerung des CMS erfolgt über einen Web-Browser. Für das Benutzer-Interface wird ein kleiner Web-Server, der ZServer, mitgeliefert. Alternativ dazu kann Zope per CGI auch an bereits vorhandene Web-Server angebunden werden. Abbildung 13 stellt die grundlegende Architektur von Zope dar.

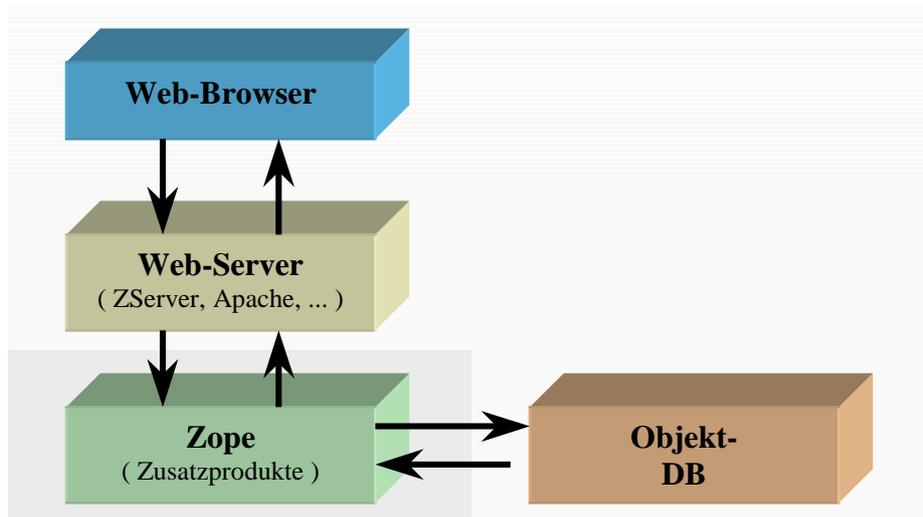


Abbildung 13 - Zope Architektur

Mittels Browser gelangt man auf die graphische Benutzeroberfläche von Zope, das *Zope Management Interface* (siehe Abbildung 14). In der linken Seite des Fensters ist eine Baumstruktur erkennbar, mittels derer auf in Zope vorhandene Objekte oder Einstellungen (Benutzerverwaltung, Datenbank-Management, Verzeichnisse, DTML-Seiten, Grafiken, etc.) zugegriffen werden kann.

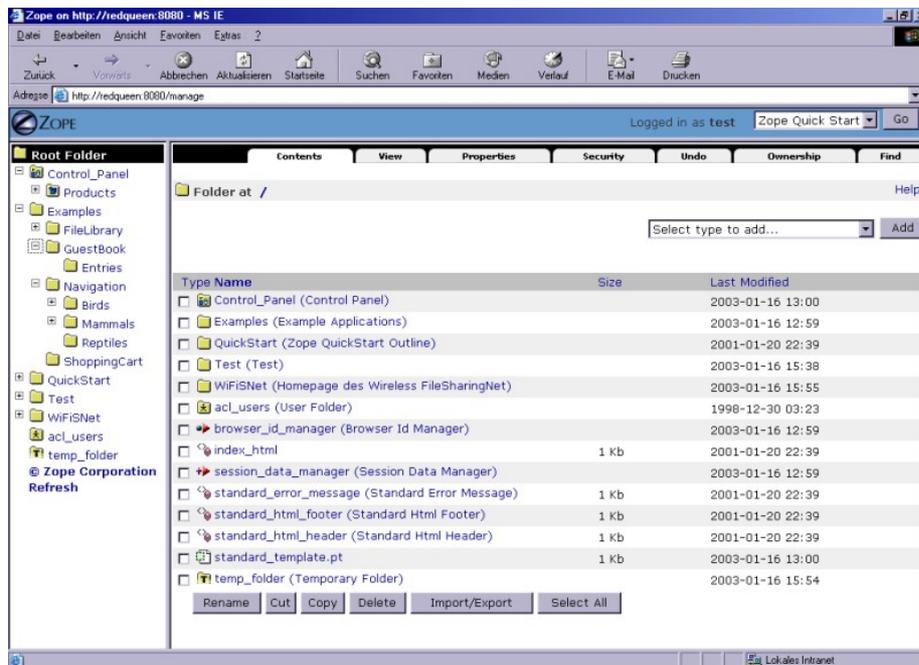


Abbildung 14 - Screenshot Zope

Im zentralen Fenster kann mittels eines Karteikartensystems auf einzelne Details eines Objekts (Einstellungen, Historie, Sicherheit, Undo-Funktionen, Benutzerzugriffe) zugegriffen werden. Auf jedes dieser im Baum verfügbaren Objekte kann (je nach Zugriffsberechtigung) über den Web-Browser zugegriffen werden, wobei jedes Objekt einen Typ besitzt, sowie Methoden und Attribute.

Je nach Objekt-Typ wird seitens Zope auf eine Anfrage entsprechend reagiert. So kann HTML-Code angezeigt, eine Grafik dargestellt oder eine SQL-Anweisung ausgeführt werden. Anstelle von HTML-Dokumenten werden DTML-Dokumente verwendet. In DTML-Dokumenten können zusätzlich zu den Standard-HTML-Tags eigene DTML-Tags, mit deren Hilfe auf andere Objekte zugegriffen werden kann.

DTML-Datei (index)

```
<dtml-var standard_html_header>
<p><dtml-var logo></p>
<p>Willkommen!</p>
<dtml-var standard_html_footer>
```

generierte HTML-Datei

```
<html>
<body bgcolor="#FFFFFF" text="Gray">
<p></p>
<p>Willkommen!</p>
<hr>
<p>Letzte &Auml;nderung 10.03.2003</p>
</body>
</html>
```

Abbildung 15 - Beispiel DTML / HTML

Abbildung 15 zeigt ein Beispiel einer DTML-Datei und der daraus generierten und angezeigten HTML-Datei. Neben der DTML-Datei mit der ID „index“ sind drei weitere Objekte in einem bestimmten Ast des Objektbaumes vorhanden: Zwei weitere DTML-Dateien („standard_html_header“ und „standard_html_footer“), sowie eine JPEG-Datei („logo“). Anhand des Objekttyps ersetzt Zope den entsprechenden Text in der DTML-Datei.

Ein weiteres Feature dieses CMS ist die Möglichkeit der „Acquisition“ [9]. Alle Objekte können auf die Methoden und Attribute von den ihnen im Objekt-Baum übergeordneten Objekte zugreifen. So brauchen beispielsweise immer gleich bleibende Design-Elemente nicht für einzelne Bereiche einer Webseite erneut definiert werden.

Neben einer Versionskontrolle besteht bei Zope die Möglichkeit zur Erstellung von Workflows, also dem vorgegebenen Ablauf der Erstellung von Inhalten. Allerdings ist dies nicht standardmäßig bei Zope verfügbar, sondern muss durch Installation des *Content Management Frameworks* (kurz CMF), ein Zope Zusatzprodukt, nachträglich eingebunden werden. Zusätzlich bietet das CMS die Möglichkeit zur Erstellung ganzer Web-Portale mit Mitgliedskontrolle, Newsticker, Personalisierung, Archivierung und anderem. Die Installation dieses Zusatzproduktes erweist sich als relativ einfach, so müssen die bei CMF mitgelieferten Verzeichnisse einfach in das Zope-Verzeichnis kopiert werden. Allerdings wird durch das CMF die Struktur der zu entwickelnden Homepage stark eingeschränkt.

Auch wenn hier nur ein kurzer Einblick in Zope möglich war, so stellt sich das CMS als komfortables Werkzeug zur Erstellung von Webseiten dar, für dessen Anschaffung keine direkten Kosten anfallen. Nachteil an dem CMS sind jedoch die nicht sonderlich übersichtliche Dokumentation („The Zope Book“) und die relativ lange Einarbeitungszeit. Ein WYSIWYG-Editor ist für die Erstellung von Webseiten nicht verfügbar, so muss der Entwickler neben der bei Zope integrierten Script-Sprache (DTML) und ggf. Python auch über detaillierte Kenntnisse über HTML verfügen. An dieser Stelle bleibt auf entsprechende Zusatzprodukte zu hoffen.

5. Ausblick

Content Management Systeme werden wohl in Zukunft immer mehr Einzug in die Unternehmen unterschiedlichster Branchen halten. Nicht ohne Grund. Kaum noch ein Unternehmen kann es sich heute mehr leisten, nicht im Internet Präsent zu sein. Musste bisher meistens ein externes Unternehmen mit dem Internetauftritt beschäftigt werden, so versprechen W-CMS und CMS hier Abhilfe. Aber auch insbesondere bei der Bewältigung der Informationsfluten leistet ein CMS erhebliche Dienste.

Die Auswahl an Content Management Systeme ist schon jetzt fast unüberschaubar. Da mittlerweile auch die großen Softwarehäuser wie Microsoft eigene Produkte auf den Markt bringen ist zu vermuten, dass sich nach und nach aus der Masse der Produkte auf Dauer nur einige wenige wirklich durchsetzen werden.

Die Bedeutung von XML in Content Management Systemen wird sicherlich weiter zunehmen, gerade im Bereich der Ablage von Informationen und den Informationsaustausch. Förderlich wird dafür das erwähnte ICE-Protokoll im Bereich des Content-Syndication sein. Das ICE wurde beim W3C als Standard vorgeschlagen und hat sicherlich gute Chancen, angenommen zu werden.

Referenzen

- [1] R. Gersdorf; Content Management ist mehr als Web-Site-Verwaltung;
URL: http://www.contentmanager.de/magazin/artikel_13_content_management_ist_mehr_als.html
- [2] S. Schönwald; Dokumenten Management für kleine und mittelständische Unternehmen;
URL: http://www.contentmanager.de/magazin/artikel_185_dokumentenmanagement_kleine_mittelstaendische.html
- [3] Dr. Kampfmeier; Content Management und XML - ein ideales Paar?;
URL: http://www.contentmanager.de/magazin/artikel_86_content_management_und_xml_-_ein_ideales_paar.html
- [4] Redaktion Contentmanager; Content Management in der deutschen Wirtschaft;
URL: http://www.contentmanager.de/magazin/artikel_77_content_management_in_der_deutschen_wirtschaft.html
- [5] M. Michelson; Dokumenten- und Workflowmanagement; Juni 1998
- [6] O. Zschau; Kennzeichen von Content Management Systemen;
URL: http://www.contentmanager.de/magazin/artikel_2_kennzeichen_von_content_management_systemen.html
- [7] V. Gruhn, D. Heymann, M. Kleine; Eine Architektur für Content-Management-Systeme auf Basis von XML
- [8] M. Willenborg; XML-Anfragesprachen; Juli 2001
- [9] B. Schmick; Content Management Systeme; Oktober 2001
- [10] A. Grädler; XML-Anfragesprachen; November 2001
- [11] Kotschy, Marjanovic, Martinovic, Erath, Fink; Content Management Systeme
- [12] W. Wiese; Konzeption und Realisierung eines WCMS; Oktober 2000
- [13] S. Baumann; Der Content Management Kernprozess;
URL: http://www.contentmanager.de/magazin/artikel_61_der_content_management_kernprozess.html
- [14] M. Klumpp; XML Query Languages;
- [15] M. Bietz; Anfragen und XML-Datenbanken; Oktober 2000
- [16] Schek, Türker, Grabs; XML und Datenbanken; 2002
- [17] A. Müller; XML-Anfragesprachen; Mai 2001
- [18] A. Deutsch, M. Fernandez, A. Levy, D. Suci; XML-QL - A Query Language for XML
URL: <http://www.w3.org/TR/1998/NOTE-xml-ql-19980819.html>
- [19] J. Robie, J. Lapp, D. Schach; XML Query Language (XQL)
URL: <http://www.w3.org/TandS/QL/QL98/pp/xql.html>
- [20] S. Sizov, A. Theobald, G. Weikum; Ähnlichkeitssuche auf XML-Daten (XXL); 2000
- [21] A. Theobald, G. Weikum; Adding Relevance to XML, Proceedings of the 3rd international Workshop on the Web and Databases, LNCS, Springer; 2000
- [22] N. Fuhr, K. Großjohann; XIRQL: An XML Query Language Based on Information Retrieval Concepts; Dezember 2001
- [23] Zope Dokumentation und Download; URL: <http://www.zope.org>; Letzter Besuch: Februar 2003
- [24] D. Peczinka, E. Schaub, R. Drüeke; Web Content Management Systeme; 2002
- [25] Vignette: <http://www.vignette.com>; Letzter Besuch: Februar 2003
- [26] Microsoft Content Management Server; URL: <http://www.microsoft.com/cmsserver>; Letzter Besuch: Februar 2003
- [27] G. Versteegen, N. Dietrich, T. Lederer, M. Lindlbauer; Management-Technologien; 2002
- [28] Axel Porst; Content Management und Workgroup Computing