

Universität Leipzig

Fakultät für Informatik

Prof. Dr. Hans-Gert Gräbe

Pro-Seminar „Knowledge Environment Engineering“

Integration von Content-Management-Systemen in Portal-Lösungen

René Plötz, 9125375

Diplomstudiengang Informatik, 7. Semester

E-Mail: Rene.Ploetz@medizin.uni-leipzig.de

Leipzig, den 19.01.2007

vorgelegt bei:

Prof. Dr. Hans-Gert Gräbe

Inhaltsverzeichnis

1	Einleitung.....	4
2	Grundlegende Technologien.....	5
2.1	Content-Management-Systeme.....	5
2.1.1	Begriffsbestimmung.....	5
2.1.2	Kernprozess.....	6
2.1.3	Funktionen von Content Management.....	7
2.2	Portal-Technologie.....	11
2.2.1	Begriffsbestimmung.....	11
2.2.2	Anforderungen.....	11
3	Integrationsebenen	14
4	Ausgewählte JAVA-basierte Umsetzungen.....	17
4.1	Jahia.....	17
4.2	Graffito.....	20
5	Zusammenfassung.....	23
6	Quellenangaben.....	24

Abbildungsverzeichnis

Abb. 1: Content Lifecycle, Seite 6

Quelle:

Rita Warren: What is Content Management? Seattle, 2003, Seite 3

Erhältlich (Stand: 18.01.2007) unter:

<http://www.ziacontent.com/downloads/What%20is%20Content%20Management.pdf>

Abb. 2: Aufbau Jahia

Quelle:

Jahia Ltd.: Jahia 5.0 End User Guide, Carouge (CH), 2006, Seite 21

Erhältlich (Stand: 18.01.2007) unter: [Jahia End User Guide](#)

Abb. 3: Jahia Workflow

Quelle:

Jahia Ltd.: Jahia 5.0 End User Guide, Carouge (CH), 2006, Seite 23

Erhältlich (Stand: 18.01.2007) unter: [Jahia End User Guide](#)

Abb 4: Graffito Architektur

Quelle:

The Apache Software Foundation: Graffito Architecture, o.O. 2004-2006.

Erhältlich (Stand: 18.01.2007) unter:

<http://incubator.apache.org/graffito/architecture.html>

1 Einleitung

In diesem Dokument sollen mögliche Schnittpunkte zwischen der Portal-Technologie und Content-Management-Systemen (CMS) herausgearbeitet und Stufen der Integration von CMS in Portale aufgezeigt werden.

Die Verknüpfung beider Technologien ist unter dem Gesichtspunkt der Notwendigkeit Informationen in organisierter Form zu strukturieren und diese seinen Kunden zur Verfügung zu stellen und aufgrund der gegebenen Gemeinsamkeiten sinnvoll. Insbesondere deswegen ist eine Vereinigung von Lösungen von sog. „content producer applications“ - zu denen Content-Management-Systeme gehören - und „content consumer applications“ - unter anderem Portale - angebracht.

Hieraus ergeben sich Anforderungen an bestehende Portal- oder CMS-Lösungen die im folgenden herausgearbeitet werden sollen.

Keinesfalls sollen hier Anwendungsfälle für CMS oder Portale besprochen werden, ebensowenig wie dies eine Abhandlung über deren Vorteile bzw. Nachteile sein soll. Es stehen also ausschliesslich die technischen Aspekte im Vordergrund.

Der erste Teil umfasst sich mit den Funktionalitäten beider Technologien, wobei der Schwerpunkt auf dem Content-Management-System liegt, da dieses letztlich die Verwaltung der für das Portal relevanten Informationen verantwortlich ist und damit den wichtigeren Faktor für die Qualität des durch die Kombination entstehenden Gesamtsystems darstellt.

Im zweiten Abschnitt werden dann die verschiedenen Stufen der Integration und die hieraus resultierenden grundlegenden Eigenschaften besprochen.

Im letzten Teil werden Java-basierte Lösungen vorgestellt, die auf die eine oder andere Art Content-Management und Portal-Technologie miteinander verknüpfen.

2 Grundlegende Technologien

2.1 Content-Management-Systeme

2.1.1 Begriffsbestimmung

Folgende Charakterisierung von Content-Management (CM) findet sich unter anderem in der Literatur:

„Software-basiertes Content Management befasst sich mit der systematischen Sammlung und Verwaltung von Informationsbausteinen in einem einzigen (logischen) Bestand. Es stellt Anfragemethoden und Mechanismen für die sichere Arbeit ganzer Nutzergruppen mit diesem Inhaltsbestand („Content Base“) bereit.“[1]

Neben dieser allgemeinen Definition, welche auch auf Dokumenten-Management-Systeme (DMS) und Datenbanken zutreffen kann gibt die obige Quelle auch folgende Definition für CM 2. Ordnung an:

„Software-basiertes Content Management befasst sich mit der systematischen Sammlung, Erstellung, Speicherung und Veredelung von strukturierten Inhalten und Mediendaten aller Art in einem einzigen, fein granulierten (logischen) Bestand. Es unterstützt gezielt die sichere Aggregation, Veredelung, Verarbeitung, Auswertung und Wiederverwendung dieser Content Base durch ganze Benutzergruppen.“[1]

Diese Definition setzt die Schwerpunkte im Gegensatz zum DMS auf einen Kreislauf bestehend aus Erstellung, Verwertung bis zur Wiederverwendung. Der entscheidende Unterschied besteht hier in der Veröffentlichung der Dokumente im Gegensatz zur ausschließlichen Speicherung, also die Unterstützung des gesamten Content Lifecycles¹.

¹ Siehe Abbildung 1

2.1.2 Kernprozess

Die Erzeugung von Inhalten stellt sich bei einem CMS in der Regel in einer Kette von Prozessen dar, welche im Folgenden kurz dargestellt und erläutert werden sollen²:

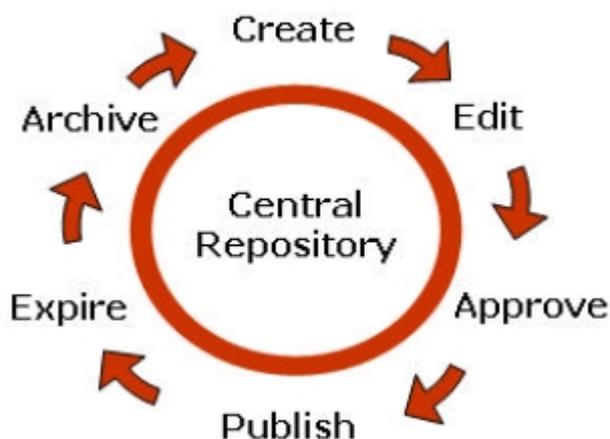


Abbildung 1: Content Lifecycle

2.1.2.1 Produktion

Der Lebenszyklus eines Inhalts startet mit seiner Erstellung bzw. erstmaligen Digitalisierung. Es hat sich für Medienunternehmen als sinnvoll erwiesen auch fremde Assets im Zuge des sog. Prozesses der „Content Syndication“ einzukaufen und dann zur Verfügung zu stellen. Wichtig für die Produktion neuer Inhalte ist allerdings auch die Nutzung eines internen Archivs für ältere Inhalte um eine Komposition dieser mit Neueren zu ermöglichen.

2.1.2.2 Publikation und Anreicherung von Inhalten

In diese Kategorie fallen sämtliche wohldefinierten Arbeitsschritte die nötig sind um einen bestimmten Inhalt zu publizieren (der sog. Workflow). Hier spielen insbesondere auch Fragen zu Veröffentlichungsregeln, Verbindungen zwischen Inhalten und Rechtemanagement eine Rolle.

² vgl. hierzu auch [2], Seite 16ff.

2.1.2.3 Content Lifecycle Management

Dieses Management umfasst im Wesentlichen Funktionalitäten zum Strukturieren des Inhalts, dessen Beschreibung mit Metainformationen, das Integrieren in ein CMS und die Archivierung bzw. Speicherung in selbigem. Diese Funktionalitäten sind insofern wichtig, da mit Ihnen die Effektivität Inhalte zu finden steht und fällt.

2.1.2.4 Distribution

Dieser Teil der Kette umfasst den Verbreitungsprozess von Inhalt, Struktur und Layout über gegebenenfalls verteilte Distributionsumgebungen in aller Welt mittels Technologien des Hostings, Netzwerkzugang oder Satellitenübertragung. Hierdurch wird eine vielfältige Wiederverwendung in den Distributionskanälen möglich.

2.1.2.5 Präsentation

Auf dieser Ebene spielen vor allem Fragen bezüglich der Ausgabe des Contents auf den spezifischen Endgeräten eine Rolle. Insbesondere umfasst dieser Punkt auch nicht-digitale Bereiche, beispielsweise die Verwertung in Printmedien. Netzwerke bilden nur einen Teil der möglichen Präsentationsmedien.

2.1.3 Funktionen von Content Management³

2.1.3.1 Funktionen erster Ordnung

a) Zugangskontrollen

Zugangskontrollen sind in einer arbeitsteiligen Umgebung unumgänglich. Hierfür ist eine Benutzerverwaltung nötig, welche die Berechtigungen verwaltet und an dieser sich Nutzer authentifizieren können.

Die Leistungsfähigkeit solcher Zugangskontrollen wird begrenzt durch die Sicherheit der Kontrolle selbst, die Einteilung der Nutzer in Gruppen, Rollen oder die Nutzung von Profilen, sowie die Feinheit der Berechtigungen.

³ vgl. hierzu [2], Seite 22ff. bzw. [3] Seite 22ff.

b) Protokollfunktionen

Wichtig ist weiterhin das Erfassen von Bestandsänderungen der Content Base, da hier Grundlagen über das Nutzerverhalten oder wichtige Metainformationen zum Content gewonnen werden können.

Es sind in diesem Zusammenhang abgestufte Implementierungen denkbar, beispielsweise wäre im Extremfall auch eine Speicherung der Zeit und des Datums jeglicher Interaktion eines Nutzers mit der Content Base denkbar.

c) Datensicherung

Da Informationen die wichtigste Ressource eines CMS darstellen ist eine Datensicherung unerlässlich. Hierbei ist es von Vorteil, dass ein CMS in der Regel die Daten zentral ablegt.

d) Rollback

Als Rollback bezeichnet man das Zurücknehmen von Änderungen an Inhalten unter der Voraussetzung, dass diese hierdurch nicht an Konsistenz verlieren. Dies ist im Verlauf des redaktionellen Prozesses insbesondere bei der Erstellung dringend erforderlich.

e) Mehrplatzfähigkeit

Im Rahmen der Erstellung der Inhalte sind mehrere Personen in verschiedenen Rollen involviert - zumeist arbeiten mehrere Personen gleichzeitig an Artikeln. Daher nutzen datenbankgestützte Systeme hierfür oftmals die Funktionen zur Sperrung von Tabellenzeilen oder -seiten oder allgemein das Transaktionskonzept des zugrunde liegenden DBMS.

f) Check-In und Check-Out

Die Mehrplatzfähigkeit ist bei nicht datenbankgestützten Systemen mittels eines sog. Check-In/Out-Systemes realisierbar. Ein Redakteur teilt dem CMS mit, dass er bestimmte Inhalte bearbeiten will. Sofern kein anderer diese aktuell über dieses System gesperrt hat, wird das System dies erlauben und sie als „ausgecheckt“ bzw. „checked-out“⁴ vermerken. Solange die Inhalte nicht wieder „ingecheckt“⁵ werden sind sie vor Bearbeitung durch andere Nutzer geschützt.

⁴ auch: gesperrt

⁵ auch: entsperrt oder „checked-in“

Integration von Content-Management-Systemen in Portal-Lösungen

g) Metainformationen und Verwaltungsfunktionen

Die Anreicherung von Inhalten mit Zusatzinformationen ist eine der zentralen Funktionen eines CMS. Hierbei existieren 2 Formen:

1. vorgefertigte („eingebaute“) Metainformationen

Diese umfassen in der Regel den Typ des Inhalts, Datum und Uhrzeit, Themen, Schlagworte, Redakteure oder andere Statusinformationen.

2. freie definierbare Metainformationen

h) Anfragefunktionen

Hierzu zählen spezielle Anfragen, beispielsweise eine Suche nach Content eines bestimmten Typs.

i) Massenoperationen zur Modifikation größerer Mengen von Inhalten des CMS, beispielsweise das Umgruppieren oder Löschen von veralteten Informationen.

2.1.3.2 Funktionen zweiter Ordnung

a) Bearbeitung und Verifikation

Einfache CMS-Lösungen nutzen externe Applikationen zum Bearbeiten, hierbei ist der Bearbeitungsverlauf dem CMS unbekannt (sog. „black box“-System – vergleichbar mit einem schwarzen Karton, dem man den Inhalt nicht ansieht).

Komplexe Redaktionssysteme hingegen versuchen wahlweise trotz des unterschiedlichen Formates der Quellen ein Bedienkonzept für die Bearbeitung zu etablieren. Dies kann unter anderem durch Fernsteuerung externer Applikationen, Nutzung eines übergeordneten, einheitlichen Dokumenten- bzw. Containerformats oder aber die direkte Implementierung eines Editors im CMS geschehen.

b) Aufzeichnung von Änderungen

Jedwede (atomare) Änderung muss mittels der Protokollfunktionen für ein Rollback aufgezeichnet werden.

c) Beziehungen zwischen den Inhalten

Inhalte müssen sich zu größeren Bausteinen (Gruppen, thematische Blöcke) zusammenfassen lassen, da dies auch notwendig für Mehrfachverwendung oder Versionierung ist.

Integration von Content-Management-Systemen in Portal-Lösungen

Weiterhin müssen Querverbindungen (Links) zu anderen Inhalten möglich sein, um Beziehungen zwischen ihnen hervorzuheben.

d) Versionsverwaltung

Falls das CMS verschiedene Versionen eines Inhalts bzw. eines Teils dieses in verschiedenen Revisionen verwalten kann, so hat dies den Vorteil, dass eine Anzahl an Redakteuren gleichzeitig an verschiedenen Versionen arbeiten kann.

e) Internationalisierung

Die Internationalisierung ist faktisch ein spezieller Fall der Versionsverwaltung. Hier werden verschiedene Sprachversionen der Inhalte gespeichert. Allerdings sind auch gesonderte Strukturierungen oder spezialisierte Übersetzungssysteme denkbar.

f) Workflow

Ein Workflow modelliert die zugrunde liegenden Geschäftsprozesse. Hierdurch wird vor allem der Content Lifecycle von der Erstellung, zur Überarbeitung, Publikation, Archivierung und Wiederverwendung unterstützt.

Letztlich ist diese Kette zwingend – nach der Erstellung muss die Überarbeitung folgen. Der aktuelle Zustand, in dem sich der Inhalt befindet, stößt also unweigerlich Folgeprozesse an.

g) Gestaltung

Leistungsfähige Content-Management-Systeme trennen Inhalte von der Struktur oder der Gestaltung der Anzeigeform. Struktur und Inhalt werden im Dokument einer Metasprache (z.B. XML) gehalten und liegen da entsprechend strukturiert vor. Die Gestaltung der Inhalte wird dann über Templates (Vorlagen) erreicht.

h) Verarbeitungsfunktionen

Viele CMS bieten über API-Dokumentationen die Möglichkeit bestimmte Funktionen nachzureichen. Dies betrifft beispielsweise eine Rechtschreibkorrektur, die Automatisierung von Prozessen im CMS oder Funktionen zur Formatierung von Inhalten.

i) Fremdformate

Kein CMS kann jedes Format unterstützen. Daher ist es zweckmäßig, dass ein Import bzw. Export von Fremdinhalten möglich ist. Dies setzt leistungsfähige Filter voraus.

2.2 Portal-Technologie

2.2.1 Begriffsbestimmung

Der Begriff „Portal“ ist im allgemeinen Sprachgebrauch nicht eindeutig festgelegt. Die Definitionen reichen von einfachen Linksammlungen bis hin zu großen Unternehmensportalen mit vielen Personalisierungsmöglichkeiten für den Nutzer. Folgende Definition lässt beispielsweise viele Interpretationen zu:

„Ein Portal ist ein zentraler und persönlicher Einstieg (Single Point of Access) in die Informationswelt des Internet oconsumptionder Intranet, von dem aus Verbindungen zu den relevanten Informationen und Diensten hergestellt werden können.“[4]

Grundsätzlich ist allen Definitionen eines Portals gleich, dass sie ein Portal als Einstiegsseite zu Inhalten des entsprechenden Netzes (Internet, Intranet) klassifizieren. Verschiedene Inhaltsquellen und Suchfunktion sind weitere Eigenschaften. Im Folgenden sollen neben den genannten weitere grundlegende Funktionalitäten größerer Portale erläutert werden.

2.2.2 Anforderungen⁶

Eine Portal-Software besteht grundsätzlich aus 2 Teilen – den Basisdiensten und den Anwendungen des Portals.

Diese Portal-Anwendungen realisieren im Allgemeinen die gesamte Anwendungsfunktionalität des Portals, indem sie auf die Basisdienste über vorher festgelegte Schnittstellen zugreifen. Eine mögliche ist der Portlet-Standard, es existieren aber auch proprietäre Schnittstellen je nach Realisierung der Basisdienste.

In diesem Zusammenhang sind folgende Basisdienste und Problemstellungen relevant:

a) Struktur-Management

Das Struktur-Management legt den strukturellen Aufbau und die Navigationsweise fest, insbesondere wo Portal-Anwendungen innerhalb der Struktur angezeigt werden. Im Fal-

⁶ vgl. hierzu [5]

Integration von Content-Management-Systemen in Portal-Lösungen

le eines personalisierbaren Portals werden hier Anwendungen an feste Positionen gesetzt (die Anwendung für den Login wäre ein Beispiel hierfür) oder für den Nutzer als optional deklariert, damit dieser das Portal für sich anpassen kann.

Auf diesen Basisdienst bauen im Allgemeinen diejenigen Personalisierungsdienste auf, welche die Struktur anhand vom Verhalten des Nutzers, Tageszeit oder Ereignissen ändern.

b) Layout-Management

Die Aufgabe des Layout-Managements ist das Zusammenstellen der aufgerufenen Portal-Seite aus den einzelnen Portal-Anwendungen und die Erzeugung der Seite für die Übertragung zum Nutzer. Besondere Berücksichtigung finden hierbei die Vorgaben des Struktur-Managements, insbesondere die persönlichen Einstellungen des Nutzers hinsichtlich aktivierter Anwendungen und deren Platzierung.

c) Content-Management

Der Basisdienst „Content-Management“ ist zumeist nur sehr abgestuft im Vergleich zu echten Content-Management -Systemen implementiert. Hierbei kann man die Herkunft eines Portals erkennen – wenn es ursprünglich ein reines CMS war, welches zu einem Portal erweitert wurde, so ist die Funktionalität dieses Dienstes deutlich ausgeprägter als bei den Portalen, welche das CMS nachträglich eingebaut haben um Redakteuren die Möglichkeit zu geben Inhalte einzugeben anstatt nur zu verlinken. Oftmals wird auf externe Content-Management-Systeme zurückgegriffen, vor allem wenn die redaktionellen Prozesse komplex sind.

d) Single-Sign-On

Der sog. „Single-Sign-On“ ist Möglichkeit, sich einmalig auf einem Portal einzuloggen und dann in allen Dienste und Anwendungen, die von diesem Portal integriert werden, ebenfalls als Portal-Nutzer erkannt zu werden. Hierdurch entsteht dem Nutzer letztlich ein Gewinn durch das Merken eines einzigen Passwortes bzw. Account-Namens und der Zeitgewinn durch den nur einmaligen Login, allerdings auf Kosten der Sicherheit aller Portal-Anwendungen bzw. Dienste im Falle, dass der Account durch Dritte übernommen wird.

e) Rechte- und Benutzerverwaltung

Die Rechte der Benutzer werden mit der Strukturierung in Gruppen oder Rollen von diesem Basisdienst gesteuert. Besondere Bedeutung erhält diese Verwaltung bei Portalen, die es ermöglichen Inhalte selbst zu erstellen oder das Aussehen des Portals zu per-

Integration von Content-Management-Systemen in Portal-Lösungen

sonalisieren. In diesen Fällen ist es je nach Umfang der Nutzer auch sinnvoll Verzeichnisdienste (beispielsweise LDAP) zu nutzen. Er sie ermöglichen den oben erwähnten Single-Sign-On.

f) Suche

Die Suche über oftmals heterogene Datenbestände oder verschiedene DBMS ist im Regelfall schwierig, da hierbei zumeist proprietäre Schnittstellen genutzt werden müssen oder aber die Verschiedenartigkeit der Daten ein Problem ist.

Je nach Anwendungsfall stehen dementsprechend viele Connectoren zu den Datenbeständen oder Diensten zur Verfügung.

g) Integrationsdienste

Die Schnittstelle zum Backend (also die Datenquellen, externe Datenbestände usw.) wird durch die Integrationsdienste gebildet. Diese Dienste führen die Datenaggregation über Datenbankschnittstellen (JDBC, ODBC) oder andere Schnittstellen bis hin zu Enterprise Application Integration (EAI). Die Integrationsdienste sind oftmals fest in den zugrunde liegenden Application Server des Portals integriert.

h) Transaktionsdienste

Die Transaktionsdienste abstrahieren ein Transaktionskonzept über die Integrationsdienste (soweit nötig), und sorgen gleichzeitig für eine Abstraktion des Datenzugriffs für die Basisdienste und gegebenenfalls für die Portal-Anwendungen.

Die Portal-Anwendungen realisieren die spezifischen Funktionen eines Portals. Sie werden im Regelfall aus einer Portal-Anwendungsklasse abgeleitet, die je nach verwendetem Produkt Portlet, I-View, I-Let oder Gadget genannt wird. Sie werden dann über die standardisierte API der Anwendungsklasse vom Portal angesteuert und nutzen die API des Portals für ihre Funktionalität.

3 Integrationsebenen

Wie bereits in den obigen Ausführungen ersichtlich bestehen eine Reihe von Gemeinsamkeiten zwischen Content-Management-Systemen und Portalen, besonders hinsichtlich dezentraler Veröffentlichung von Inhalten, dem freien Zugriff auf diese von externen Benutzern als öffentliche Inhalte bzw. von Innen heraus als Inhalte im Archiv. Ferner ist es möglich, web-gestützt über wohldefinierte Schnittstellen auf Datenbestände im Backend zuzugreifen, was insbesondere für CRM-Portale relevant ist. Dies führt zu schnelleren Geschäftsprozessen, zu einer schnelleren Entscheidungsfindung durch verbessertes Informationsmanagement und eine offensichtliche Intensivierung der Teamarbeit.

Ein CMS speichert seine Datenbestände an eine festgelegte Stelle – im folgenden Content Repository genannt. Dieses Repository kann über Zugriffsschnittstellen angesprochen werden. Ein Portal wird dies nutzen um auf Datenbestände des CMS zurückgreifen zu können, wobei es nicht unbedingt das einzige zugreifende System sein muss. Es wäre auch denkbar, dass das CMS unabhängig vom Portal arbeitet insofern, dass es Inhalte gibt, deren Lebenszyklus außerhalb des Portals liegt.

Es ist möglich ein Dokument in einem client-seitigen Textverarbeitungsprogramm erstellen, es dann in das CMS „einchecken“ bzw. „auschecken“ und dann nur mit dem Portal anzeigen lassen. Andererseits ist es denkbar, dass es spezifischen Inhalt gibt, welcher über das Portal erstellt, verwaltet und ausgeliefert wird. In dieser Hinsicht sind Portale und CMS komplementär.

Man kann abhängig vom Grad der Verzahnung zwischen beiden Technologien 4 Stufen oder Ebenen der Integration unterscheiden. Diese sind Anwendungen eines CMS – die Stufe der Verzahnung hängt wie bereits oben erwähnt von der Bedeutung der CMS für den Workflow bei der Erstellung von Inhalten für das Portal ab. Je nach Schwierigkeit der redaktionellen Prozesse wird man ein CMS mit anderen Funktionalitäten in das bestehende Portal integrieren. Die einzelnen Ebenen bauen dabei aufeinander auf.

1. Bereitstellen von Inhalten

Auf dieser Ebene der Integration stellt das CMS Funktionalitäten zur Verfügung um Inhalte zu publizieren, also sie zu erstellen, zu überprüfen und dann zu veröffentlichen. Der Zugriff finde dann über das personalisierte Portal mit entsprechenden Zugriffsrechten und Rollenverteilungen statt. Man kann daher sagen, dass das CMS das Portal mit

Integration von Content-Management-Systemen in Portal-Lösungen

Inhalten füttert – das Portal selbst integriert sich allerdings nicht in das CMS. Die Kommunikationsrichtung ist also einseitig.

Sämtliche Portal-Anwendungen können dementsprechend keine Änderungen am Inhalt vornehmen, sind aber daher auch weniger komplex.

Eine weitere Unzulänglichkeit besteht darin, dass unterschiedliche Zugriffsverwaltungen existieren. Daher trennt sich auch die Nutzerschicht in CMS-Nutzer und Portal-Nutzer, was vor allem bei größeren CMS-Portal-Lösungen ineffizient ist.

2. Interaktives Ausschöpfen von Inhalten

Im Gegensatz zur ersten Stufe wird hier ein Web-Service zur bidirektionalen Kommunikation zwischen beiden Komponenten des Gesamtsystems eingesetzt. Dies führt dazu, dass das Portal Abfragen zu Inhalten direkt an das CMS stellen kann anstatt gefüttert zu werden. Die grundsätzliche Richtung bleibt aber erhalten – es besteht also nur eine zusätzliche Möglichkeit der Abfrage.

Der Sinn dieses Web-Services liegt in der Personalisierung der Inhalte. Das Portal kann je nach Aufbau den nutzerspezifischen Teil dynamisch generieren, wofür direkte Anfragen an das CMS nötig sind. Weiterhin sind hierdurch Aktualisierungen der Inhalte denkbar, ebenso wie das automatische Entfernen von alten Inhalten nach einer gewissen Zeit, ohne allerdings andere Inhalte wieder zu veröffentlichen. Ein weiterer Vorteil liegt in der Verbesserung der Suchfunktion, welche nun auch nach Metainformationen suchen kann, welche nicht angezeigt werden.

3. Inhaltliche Beiträge

Auf dieser Ebene bietet das CMS seine Funktionen direkt im Portal an. Dadurch können die Nutzer direkt Inhalte erstellen, freischalten und veröffentlichen. Die Rollenverteilungen des Portals sind hierbei maßgeblich für die Berechtigungen eines Nutzers am integrierten CMS.

Wichtige Funktionalitäten für dieses Integrationslevel sind:

- Portal-Anwendungen für das Check-In und Check-Out
- gegebenenfalls eine Anwendung für Statusinformationen bzw. Aktionen für den Inhalt
- eine Anwendung zur Freischaltung des Inhalts am CMS und / oder eine Abstraktion des dem Portal zur Verfügung gestellten Inhalts durch sog. „Syndikationskanäle“ bevor der Inhalt im Portal selbst verfügbar wird.
- einheitliches Aussehen und Steuerung von CMS- und Portal-Funktionen

4. Inhaltsverwaltung

Die höchste Ebene zeichnet sich dadurch aus, dass Nutzer hier die Administration des CMS direkt im Portal vornehmen können. Die Administration des CMS wird also durch das Portal hindurch zum Endnutzer geleitet. Diese Funktion ist vor allem für Portale interessant, bei denen stark verteilte, laufend wechselnde Benutzergruppen bzw. Benutzertypen an Inhalten für das Portal arbeiten.

Mögliche Einsatzzwecke oder Funktionalitäten hierfür sind:

- Bereitstellung von Workflow-Vorlagen über Portal-Anwendungen um Prozesse wie das Anlegen oder Editieren von Inhalten zu optimieren und zu vereinheitlichen
- Abbildung von administrativen Funktionen (Hinzufügen von Nutzern, Gruppen oder deren Berechtigungen) über Portal-Anwendungen
- Portal-Anwendungen für (administrative) statistische Zwecke, beispielsweise Zugriffstatistiken oder Trends für Inhalte des CMS
- Speicherung der Konfiguration des Portals direkt im CMS als eine Form des Inhalts. Hierbei ergeben sich weitere Funktionalitäten wie das Rollback einer Konfiguration oder von externen Inhalten.

4 Ausgewählte JAVA-basierte Umsetzungen

4.1 Jahia

Der CMS und Portal-Server Jahia⁷ ist ein Open-Source-Projekt der Jahia Ltd., welche neben einer freien Community-Version mehrere im Funktionsumfang erweiterte Kaufversionen des Produktes anbietet.

Das System selbst integriert neben einem Content Management Server einen Portal-Server, ein Document-Management-System, Proxy-Server und einen BPM-Server, sowie eine Auswahl an fertigen Portal-Anwendungen. Die Verfügbarkeit unterscheidet sich je nach Version (beispielsweise besitzt die Community-Version kein Single-Sign-On und keinen Proxy-Server). Nach eigener Aussage ist Jahia der führende Anbieter von kommerziellen Open-Source Enterprise Content Management Systemen⁸.

Jahia ist ein Content Management Server mit integriertem Portal-Framework basierend auf dem JSR168-Standard[6] sowie einem Document-Management-System.

Der Aufbau besteht aus folgenden 3 Modulen (siehe auch [7]):

1. Der Jahia Foundation Server stellt die Funktionalität des Backends zur Verfügung
2. einem Portal-Server
3. dem Content-Management-Server

⁷ <http://www.jahia.net> (Stand: 18.01 2007)

⁸ <http://www.jahia.net/jahia/Jahia/cache/bypass/pid/554?ctnid=3257> (Stand: 18.01 2007)

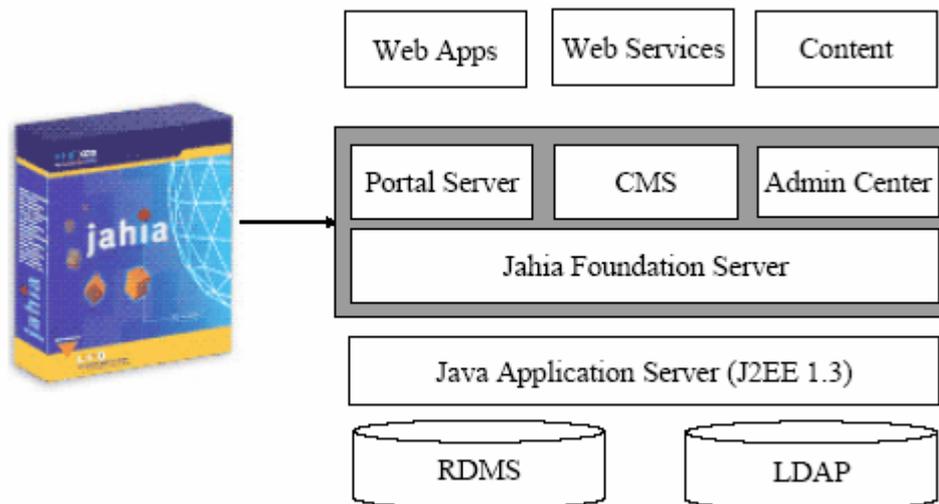


Abbildung 2: Aufbau Jahia

Wichtige Merkmale der (zu kaufenden) Vollversion:

1. Benutzung von Templates zur Anzeige der Inhalte
2. Internationalisierung
3. Versionierung der Inhalte
4. Portlet-Unterstützung nach JSR168
5. Document-Management-System
6. Rollenbasiertes Zugriffssystem
7. Virtuelle Unterseiten durch Integration eines JSR128⁹-Proxy-Servers
8. Statistische Funktionen
9. Zeitbasiertes Veröffentlichen von Inhalten
10. JSR170¹⁰-kompatibler Import/Export von Inhalten
11. Active Directory-, LDAP- und Single-Sign-On-Unterstützung

⁹ siehe hierzu <http://www.jcp.org/en/jsr/detail?id=128> (Stand: 18.01 2007)

¹⁰ siehe hierzu [8]

Integration von Content-Management-Systemen in Portal-Lösungen

Abstriche hinsichtlich der Funktionalität muss man bei der Community-Version machen. Hier fehlen vor allem die für große Seiten interessanten Single-Sign-On-Fähigkeiten, vor allem aber auch die Unterstützung für die Mehrsprachigkeit sowie die Versionierung. Dies hat zur Folge, dass Jahia in der Community-Version für größere Projekte nur bedingt geeignet ist.

Jahia zeichnet sich durch eine tiefe Integration des CMS in das Portal aus. Dies wird am Workflow zum Erstellen der Inhalte für das Portal deutlich:

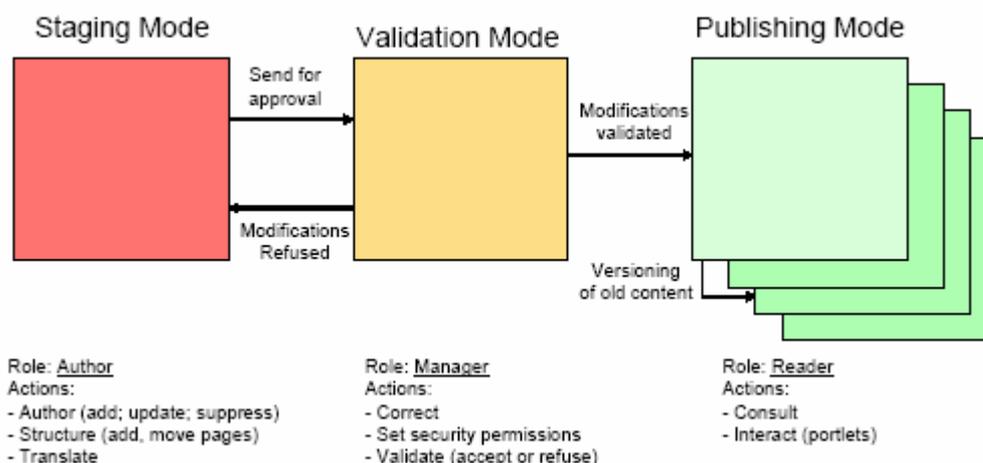


Abbildung 3: Jahia Workflow

Wie ersichtlich ist es unnötig bei der Erstellung von Inhalten zwischen Portal und CMS zu wechseln, ebenso existieren keine 2 Benutzerverwaltungen. Daher kann man die Integration von CMS und Portal bei Jahia als Ebene-4-Integration klassifizieren.

4.2 Graffito

Graffito¹¹ ist Teil des Apache Incubator Projektes, welches das Ziel hat Projekte einem Revisionsprozess hinsichtlich Qualität und Legalität der Codebasis zu unterziehen damit diese als Projekte der Apache Software Foundation akzeptiert werden können.

Das Graffito-Projekt selbst hat vergleichbar mit Jahia das Ziel einen Portal-Server mit einem CMS zu verknüpfen um ebenso die höchste Integrationsstufe zwischen CMS und Portal zu erreichen, allerdings befindet sich das Projekt in einem sehr frühen Stand, da insbesondere die Funktionen des CMS unfertig sind. Als Portal dient dem Projekt Jetspeed 2¹², letztlich ist also die gesamte Integrationsarbeit hier noch durchzuführen.

Die geplante Architektur von Graffito stellt sich folgendermaßen dar:

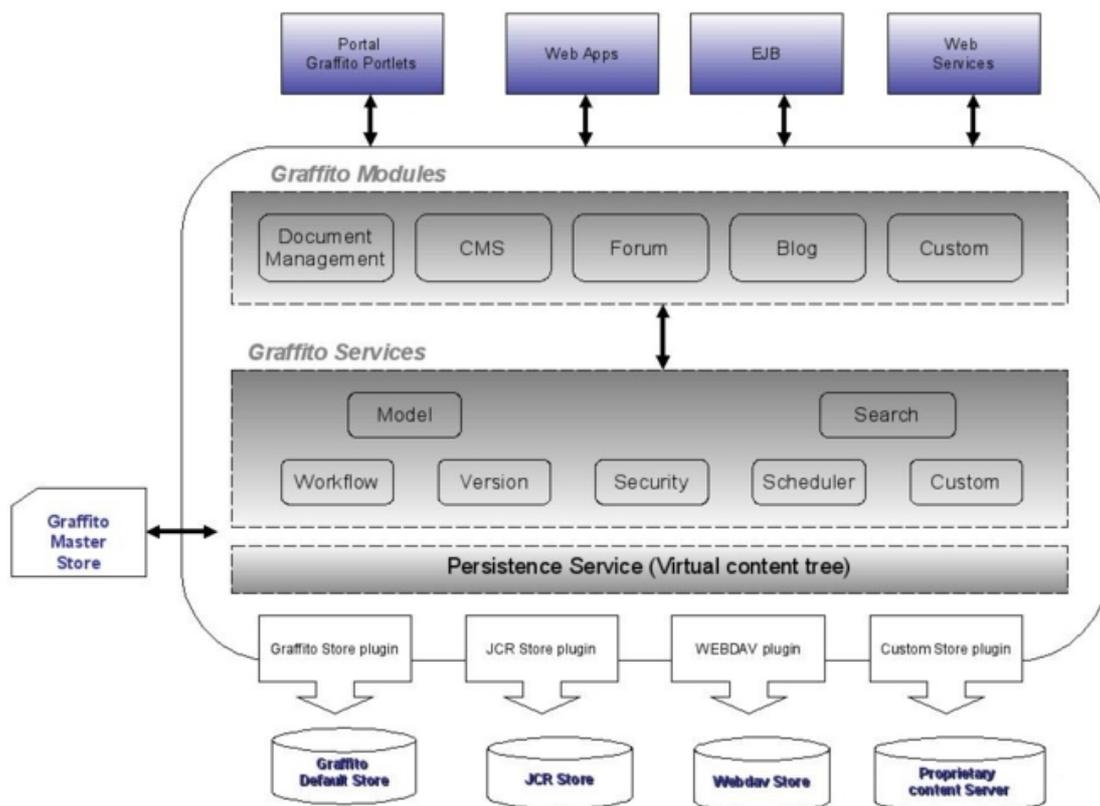


Abbildung 4: Graffito Architektur

¹¹ <http://incubator.apache.org/graffito/> (Stand: 18.01 2007)

¹² <http://portals.apache.org/jetspeed-2/> (Stand: 18.01 2007)

Integration von Content-Management-Systemen in Portal-Lösungen

Wie ersichtlich existieren eine Reihe von sog. Graffito-Services, welche vergleichbar mit den Basisdiensten des Portals die Funktionalität für die Graffito-Module zur Verfügung stellen.

Es gibt im Bereich des Graffito-Projektes eine Reihe von zu implementierenden und implementierten sog. „Repository Services“ (siehe auch [9]):

1. Model Service

Management aller Objekte des CMS, inkl. Verwaltungsfunktionen (erstellen, löschen, verlinken, aktualisieren).

2. Lock Service

Organisation des gleichzeitigen Zugriffs auf CMS-Objekte (beispielsweise gleichzeitiges Editieren)

3. Version Service

Graffito soll mit Funktionen des Backends eine Versionierung der Inhalte ermöglichen, insbesondere auch von Statistiken und Metainformationen. Dies soll auf Basis von Transaktionen stattfinden.

4. Security Service

Ferner sollen die CMS-Benutzergruppen in die zugrunde liegenden Jetspeed-Gruppen integriert werden, weswegen der Security Service in den entsprechenden Service von Jetspeed 2 eingebaut werden muss.

5. Search & Indexing Service

6. Workflow Service

Wie erwähnt folgen die Schritte zur Erstellung von Inhalten bei großen Content-Management-Systemen einem festgelegten Ablauf. Genau das soll hier auch implementiert werden, allerdings soll dieser Ablauf anpassbar sein, damit der redaktionelle Prozess je nach Anwendungsfall verbessert wird.

7. Event Service

Der Event Service soll dafür sorgen, dass das CMS auf bestimmte Ereignisse, wie beispielsweise das Erstellen von Inhalte dynamisch reagieren kann.

8. Integration Service

Der Integration Service stellt die Abstraktionsstufe des darunter liegenden Speichersystems dar. Es soll insbesondere dafür sorgen, dass externe Inhalte ebenso wie physische

Integration von Content-Management-Systemen in Portal-Lösungen

behandelt werden können, also die darüber liegende Schicht auch auf externe Quellen zugreifen kann, als wären sie direkt in einem Content Repository verfügbar.

Weitere geplante bzw. verwirklichte Funktionalitäten:

1. Graffito Content Store

Es sollen eine Reihe von Backend-Verbindungen unterstützt werden, namentlich OJB, WebDAV (vergleichbar mit Apache Slide¹³) bzw. Content Repository nach JSR170.

2. CMS-Portlets

a) administrative Portlets

b) Content Browser, Editor und Viewer

c) Content Workflow und Content notification

d) Spezialisierte Portlets für Nachrichten, Foren oder andere, welche in ein Content Repository Daten ablegen müssen

Insgesamt gesehen ist Graffito noch in einem Anfangsstadium. Die obigen Services sind nur teilweise implementiert. Zurzeit baut Graffito stark auf Jetspeed 2 auf und erweitert dieses Portal um Funktionalitäten eines CMS. Allerdings sind (Stand Januar 2007) unter anderem nur der Model Service, der Security Service, eine OJB-Implementierung und die Jetspeed-Integration fertig. Zwar stehen auch eine einfache Suchfunktion (basierend auf Lucene¹⁴), sowie Demo-Portlets und eine einfache Versionierung zur Verfügung, allerdings ist Graffito derzeit für den Produktiveinsatz aufgrund des frühen Entwicklungsstands nicht empfehlenswert.

¹³ <http://jakarta.apache.org/slide/> (Stand: 18.01 2007)

¹⁴ <http://lucene.apache.org/> (Stand: 18.01 2007)

5 Zusammenfassung

Die hier besprochenen Technologien geben einen Einblick in die Evolution, die Content-Management-Systeme und die Portal-Technologie in den letzten Jahren vollzogen haben - von den anfänglichen Doppellösungen des „Fütterns“ des Portals durch das CMS bis hin zu hochintegrierten Lösungen der Enterprise Portale.

Dieses Dokument hat zunächst beide Technologien einzeln besprochen und ihre wichtigsten funktionalen Merkmale aufgezeigt. Darauf aufbauend wurden im folgenden Teil die 4 Ebenen der Integration zwischen diesen aufgezeigt und erläutert.

Im letzten Teil wurden 2 JAVA-basierte Open-Source-Projekte – Jahia und Graffito – vorgestellt und kurz erklärt, inwieweit diese die Integration zwischen CMS- und Portal-Technologie vollzogen haben.

Es ist sinnvoll sich noch einmal die Bedeutung des gesamten Prozesses der Integration von CMS in Portale vor Augen zu führen:

Mittels der Kombination von CMS und Portal in ein Software-Produkt lässt sich die gesamte Kette von der Erzeugung des Inhalts bis zu dessen Distribution über die Vertriebskanäle in einer Applikation steuern und damit vereinheitlichen.

Während früher zwei völlig verschiedenartige Oberflächen und Bedienkonzepte dominierten, sind es heute die Enterprise-Portal-Lösungen. Hierbei spielen nicht nur die für Großunternehmen geschaffenen Lösungen von IBM, SAP, Microsoft oder Oracle eine Rolle, sondern auch Lösungen, welche aus Open-Source-Komponenten zusammengesetzt sind, namentlich Jahia und (in Zukunft) Graffito.

6 Quellenangaben

- [1] Rothfuss, Gunther und Christian Ried (Hrsg.): Content Management mit XML. Springer, 2001.
- [2] Doris Wagner: Vereinheitlichung und Standardisierung von Repositories im Content Management Bereich. Wien, 2006. Erhältlich (Stand: 18.01.2007) unter: http://cocoon.ifs.tuwien.ac.at/lehre/diplomarbeiten/2006_Wagner.pdf
- [3] Eli, Stefan: Konzeption und Realisierung eines Warehouse-Konzepts für das Content Management. Kaiserslautern, 2003. Erhältlich (Stand: 18.01.2007) unter: <http://www.sts.tu-harburg.de/pw-and-m-theses/2003/eli03.pdf>
- [4] Großmann, Martina und Koschek Holger: Unternehmensportale. Heidelberg, 2004, S. 28
- [5] Thorsten Gurzki und Henning Hinderer: Eine Referenzarchitektur für Software zur Realisierung von Unternehmensportalen.
Erschienen in: WM 2003: Professionelles Wissensmanagement - Erfahrungen und Visionen; Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.), GI-Edition - Lecture Notes in Informatics (LNI), P-28 Bonner Köllen Verlag, 2003
- [6] Sun Microsystems: Introduction to JSR 168 – The Java Portlet Specification. o.O. 2003.
Erhältlich (Stand: 18.01.2007) unter: http://developers.sun.com/prodtech/portalserver/reference/techart/jsr168/pb_whitepaper.pdf
- [7] Jahia Ltd.: Jahia 5.0 End User Guide, Carouge (CH), 2006
Erhältlich (Stand: 18.01.2007) unter: [Jahia End User Guide](#)

Integration von Content-Management-Systemen in Portal-Lösungen

- [8] Sun Microsystems: SR 170 for Developers: An Introduction to the Content Repository for Java™ Technology AP. o.O., 2006. Erhältlich (Stand: 18.01.2007) unter:
<http://developers.sun.com/learning/javaoneonline/2006/coreenterprise/TS-4474.pdf>
- [9] The Apache Software Foundation: Graffiti Site v.1.0-a1-dev Project Documentation. o.O, 2005. Erhältlich (Stand: 18.01.2007) unter: <http://incubator.apache.org/graffito/site.pdf>
- [10] CAP Ventures Inc.: Integrating Content Management with Portals: Meeting Enterprise Information Needs. Norwell, MA, 2001.
Erhältlich (Stand: 18.01.2007) unter: [Integrating Content Management](#)