

Joomla! und Plone - ein Vergleich zweier Content-Management-Systeme

Bernd Steinacker
HfT Stuttgart

15. Januar 2009

Inhaltsverzeichnis

1	Einleitung	1
2	Joomla!	2
2.1	Installation	2
2.2	Administration	2
2.3	Sicherheit	3
2.4	Barrierefreiheit	4
3	Plone	5
3.1	Installation	5
3.2	Administration	6
3.3	Sicherheit	7
3.4	Barrierefreiheit	7
4	Bewertung	8
4.1	Vergleich der beiden CMS	8
5	Fazit	9
5.1	Zusammenfassung	9
5.2	Ausblick	9

1 Einleitung

Informationsmanagement wird in der heutigen Zeit immer wichtiger. Die zunehmende Marktdurchdringung mobiler Endgeräte, die über einen Internetzugang verfügen, trägt maßgeblich dazu bei.

Ein wichtiger Schritt zu einem optimierten Informationsmanagement ist, wichtige Informationen ortsunabhängig zugänglich zu machen. Hierzu wird an Softwarelösungen gearbeitet, die anfallende Tätigkeiten technisch unterstützen. Beispiele hierzu sind browserbasierte E-Mail-Services, Gruppenkalender, Dokumentenmanagementsysteme oder auch Online-Banking.

Das Erstellen einer Webseite, die solche Dienste zur Verfügung stellt, erfordert spezielle Kenntnisse und einige Übung in der Entwicklung und der Gestaltung von Internetauftritten. In der Vergangenheit führte dies immer wieder dazu, dass Projekte durch teure Spezialisten umgesetzt werden mußten. In einigen Fällen wurden geplante Projekte sogar ganz gestrichen.

An diesem Punkt setzen Web-Content-Management-Systeme wie Joomla! oder Plone an. Ihre Funktionalitäten werden über einen Browser verwaltet und können dadurch örtlich ungebunden genutzt werden. Ein Web-Content-Management-System ist ein komplexes Redaktionssystem. Ziel ist u.a. eine vollständige Trennung der Inhaltserstellung und -pflege von der technischen Generierung des HTML-Codes der Webseiten zu erreichen. Für Redakteure ist es damit sehr viel einfacher, Webseiten zu erstellen, die allen Ansprüchen an die Einhaltung internationaler HTML-Standards und des Corporate Designs gerecht werden. Grund hierfür ist, dass sie sich nur noch um die eigentlichen Inhalte kümmern müssen.

In dieser Ausarbeitung werden Joomla! in der stabilen Version 1.5 und Plone in der Version 3.1.6 verglichen.

Eine Gegenüberstellung der Kernfunktionen zeigt eine klare Überlegenheit von Plone. Dieser Vergleich wird jedoch auch Bereiche wie Handhabung und Einsatzgebiete einbeziehen.

2 Joomla!



Abbildung 1: Joomla! Logo [4]

Joomla! entstand als Open-Source-Variante zu Mambo, einem im Jahr 2001 von der Firma Miro entwickelten Content-Management-System [7]. Seit Anfang 2008 liegt nun die Veröffentlichung der stabilen Version 1.5 vor, die auch den momentan aktuellen Stand abbildet (1.5.6). Joomla! ist GNU/GPL 2.0 lizenziert. Abbildung 1 zeigt das Logo, unter dem Joomla! aktuell vertrieben wird.

2.1 Installation

Voraussetzung für die Installation von Joomla! ist die Installation eines Webserver, der PHP unterstützt. Außerdem wird ein von Joomla! unterstütztes Datenbanksystem benötigt [7].

Für Joomla! ist hierzu das vorkonfigurierte Paket namens XAMPP sehr hilfreich. Es enthält den Webserver Apache in der Version 2 sowie einen MySQL-Server. Bei der Installation des Apache wird bei XAMPP berücksichtigt, dass dieser PHP-Unterstützung anbietet. Natürlich kann man diese Komponenten auch einzeln installieren bzw. Alternativen benutzen, muss diese jedoch für die Nutzung von Joomla! anpassen.

Ist diese Umgebung eingerichtet, ist die weitere Installation von Joomla! auch für ungeübte Benutzer einfach durchzuführen. Das Joomla!-Paket wird in den gewünschten Ordner entpackt. Per Browser kann dann auf den Webinstaller, eine Art Hilfsprogramm, zugegriffen werden. Das Programm führt den Benutzer leicht verständlich durch die Konfiguration.

2.2 Administration

Die gesamte Administration von Joomla! erfolgt über das sogenannte Backend, also einem Bereich der Webseite, der nur den Administratoren zugänglich ist.

Joomla! versucht die Administration durch die Verwendung von JavaScript sowie der Integration von Ajax-Elementen grafischen Oberflächen wie z.B. Windows, KDE oder Gnome ähnlich zu gestalten [7].

Über die Startseite im Administratorenbereich, dem Kontrollzentrum (vergleiche Abbildung 2), erfolgt der Zugriff auf die einzelnen Bereiche der Webseite, wie z.B. den Menüs, den Benutzern, den Spracheinstellungen usw.. Die Konfiguration dieser Bereiche wird hier ebenfalls durchgeführt.

Desweiteren können hier Erweiterungen installiert und konfiguriert werden. Hierzu zählen etwa Templates, also Vorlagen für das Aussehen der Seite, oder auch Zusatzprogramme (Plugins genannt) wie z.B. dem Editor oder einem Bewertungssystem für Beiträge oder ähnliches.

Ein wichtiges Merkmal, das Joomla! bietet ist die so genannte Objektstandhierarchie. Das bedeutet es können beliebig viele Bereiche, Abteilungen und Seiten eingerichtet werden.



Abbildung 2: Kontrollzentrum Joomla! [1]

2.3 Sicherheit

Im Zusammenhang mit der bei Joomla! verwendeten Scriptsprache PHP entstehen einige Probleme.

Wegen der Verwendung von globalen Variablen oder durch unsichere Serverkonfigurationen können Sicherheitslücken entstehen.

Es existiert zum Beispiel die Möglichkeit, dass über die URL einer Seite Variablen verändert werden, wenn diese vorher nicht korrekt initialisiert wurden. So erhält etwa durch den Aufruf von

`http://beispiel.de/beispiel.php?page=http://boese.de/schadcode.txt`

die Variable page den Wert „http://boese.de/schadcode.txt“.

Verwendet nun ein Skript `include($_GET['page'] . '.php')` wird der schädliche Code eingeschleust [3].

Durch eine gewissenhaft vorgenommene Administration können Sicherheitslücken von PHP natürlich vermieden werden. Die Gesamtzahl an bekannten Sicherheitslücken liegt jedoch deutlich über der Zahl der bekannten Schwächen der im Rahmen von Plone verwendeten Programmiersprache Python.

Abbildung 3 bietet einen Überblick über die sicherheitsrelevanten Funktionen der beiden CMS.

Security	Joomla!	Plone
Audit Trail	No	Yes
Captcha	Free Add On	Free Add On
Content Approval	Yes	Yes
Email Verification	Yes	Yes
Granular Privileges	No	Yes
Kerberos Authentication	No	Free Add On
LDAP Authentication	Yes	Yes
Login History	Yes	Free Add On
NIS Authentication	No	Free Add On
NTLM Authentication	No	Free Add On
Pluggable Authentication	Yes	Yes
Problem Notification	No	Free Add On
Sandbox	No	Yes
Session Management	Yes	Free Add On
SMB Authentication	No	Free Add On
SSL Compatible	Yes	Yes
SSL Logins	Yes	Free Add On
SSL Pages	Yes	No
Versioning	Free Add On	Yes

Abbildung 3: Vergleich der sicherheitsrelevanten Eigenschaften [2]

2.4 Barrierefreiheit

Bisherige Versionen von Joomla! haben den Inhalt standardmäßig in Layouttabellen ausgegeben.

Die Verwendung von Cascading Style Sheets (CSS) und der seit Version 1.5 neu strukturierte HTML-Code erlaubt es nun die Seiten in kurzer Zeit barrierefrei zu gestalten. Durch das System des sogenannten Template-Overwrites wurde es ermöglicht, die HTML-Ausgabe in ein Template zu verlagern. Das bedeutet, wenn Joomla! im Template-Verzeichnis einen HTML-Ordner mit entsprechendem Inhalt findet, greift es darauf zu. Andernfalls weicht es auf den Standard-Code aus, der aus Gründen der Abwärtskompatibilität noch mit Tabellen arbeitet. Joomla! wird seit der Version 1.5 standardmäßig mit dem barrierefreien Template Beez ausgeliefert, welches genau diese Funktionen nutzt um dann barrierefreien Inhalt zu erzeugen [7].

3 Plone



Abbildung 4: Plone Logo [5]

Plone Version 1 erschien 2002. Es ist in der Programmiersprache Python geschrieben und unterliegt ebenfalls der GNU/GPL-Lizenz. Es findet Anwendung sowohl für Intranet- bzw. Extranet-Anwendungen, als auch als Document-Publishing-System. Erweiterungen ermöglichen darüber hinaus den Einsatz für Applikationsserver-Aufgaben wie z. B. als Webshop. Auch ein Einsatz als Groupware-Tool ist möglich.

Plone baut auf dem Open Source Application Server Zope mit seinem Content-Management-Framework auf [9]. Durch Verwendung des Archetypes-Frameworks ist es möglich eigene Web-Applikationen entwickeln. Archetypes ist ein Framework, um die Entwicklung von Plone-Produkten zu automatisieren. Liegt eine formale Beschreibung für einen Inhaltstyp vor, erledigt Archetypes quasi alles. Die Erstellung von Views und Editformularen für den Entwickler eingeschlossen. Dies erlaubt, neue Inhaltstypen schnell zu entwickeln.

Beispielsweise kann unter Verwendung von ArchGenXML aus einem UML-Modell (Unified Modeling Language), ein lauffähiges Plone-Produkt generieren [9].

Desweiteren ist Plone vollkommen Objektorientiert. Inhalte werden in Plone daher in Form von Artikeltypen gespeichert. Jeder Typ speichert Informationen nach einem eigenen Schema. In der Standardversion von Plone gibt es die Artikeltypen Ordner, Seite, Bild, Datei, Link, Nachricht, Termin und Kollektion [8].

Die Benutzerschnittstelle von Plone wurde in mehr als 50 Sprachen übersetzt und eignet sich somit hervorragend für eine Website, die von einem internationalen Team gepflegt und verwaltet werden soll. Mehrsprachige Inhalte werden mit der Erweiterung LinguaPlone verwaltet, die von der Website plone.org heruntergeladen werden kann [8].

3.1 Installation

Auch zur Installation von Plone existiert ein Programm, das den Benutzer verständlich durch die Installation leitet. Die Einrichtung und Konfiguration des Webservers und des Datenbankservers müssen aber in jedem Fall selbst vorgenommen werden. Die Konfiguration setzt im Gegensatz zur Installation des XAMPP-Pakets einiges an Kenntnissen voraus. Plone setzt auf Zope auf. Nachdem Plone aus seinen Quellen installiert wurde muss deshalb eine Plone-Instanz erstellt werden. Hierzu muss sich der Benutzer bei der Schnittstelle, über die

Zope verwaltet wird, anmelden und dort die Plone-Site hinzufügen [9]. Diese Schnittstelle nennt man Zope Management Interface (siehe Abbildung 4).



Abbildung 5: Zope Management Interface [1]

3.2 Administration

Bei Plone sind Front- und Backend nicht deutlich voneinander abzugrenzen. Das Backend wird teilweise quasi in das Frontend „eingebündelt“.

Zum Konzept von Plone gehören Benutzer, Rollen und Gruppen [9]. Je nach Rolle des Benutzers bekommt er Zugriffsrechte zugewiesen. Diese Zugriffsrechte beeinflussen dann die Möglichkeiten, die der Benutzer im Bezug auf die Inhalte der Seite aber auch der Konfiguration der Seite hat.

Für jedes Mitglied, das sich bei einer mit Plone erstellten Seite registriert, besteht die Option einen eigenen Ordner zu erstellen. Für diesen Ordner ist eine spezielle Sicherheitsstufe eingestellt. Diese erlaubt nur dem entsprechenden Mitglied (und Administratoren) Inhalte darin hinzuzufügen und zu bearbeiten [9]. Plone besitzt ein eigenes Authentifizierungssystem. Bei Bedarf arbeitet es somit mit externen Systemen wie OpenID, LDAP oder Active Directory zusammen. Dadurch wird ermöglicht Plone an bestehende Single-Sign-On-Lösungen anzuschließen [8].

Eine besonders interessante Einrichtung von Plone ist die zeitgesteuerte Veröffentlichung, bei der Inhalte zu einem vorgegebenen Zeitpunkt veröffentlicht bzw. versteckt werden.

Eine weitere Möglichkeit die Plone bietet ist die Ereignissteuerung. Hierunter versteht man die Verknüpfung von Aktionen im CMS mit anderen Aktionen, so dass man bestimmte Abläufe automatisieren kann. Möchte z.B. ein Administrator einer Site darüber informiert werden, wenn ein Benutzer einen neuen Artikel ins System eingestellt hat, so kann er für diesen Fall eine E-Mail-Benachrichtung einrichten. Diese automatisierten Funktionen kann man in Plone durch Regeln steuern, die man über ein Menü mit wenig Aufwand formulieren und einzelnen Artikeln oder Ordnern flexibel zuweisen kann [8].

Inhalte in Plone können über RSS (Really Simple Syndication) syndiziert werden. Plone kann für jeden beliebigen Ordner, für Kollektionen und sogar für Suchergebnisse RSS-Feeds erstellen.

3.3 Sicherheit

Zope als Basis von Plone verfügt über ein Sicherheits-Framework, das auf so genannten Rollen aufbaut.

Für jedes einzelne Objekt kann bestimmt werden, für welche Art des Zugriffs welche Rolle benötigt wird. Diese Rollen können den einzelnen Usern dann auch lokal zugewiesen werden, um ihnen z. B. Zugriff nur in bestimmten Teilbäumen zu gewähren.

Im Vergleich zu PHP arbeitet der Web-Benutzer nicht im Dateisystem des Servers, sondern in einer virtuellen Umgebung. Diese Umgebung kann der Benutzer nur verlassen, wenn es vom Entwickler gewünscht ist. Dieses System verhindert Zugriffe auf Daten, die auf dem Server abgelegt sind, und unterbindet so viele der üblichen Angriffsmuster.

Das Sicherheitsmodell von Plone ist mächtig und vielschichtig [9]. Es zu verwalten oder Fehler darin zu finden ist für wenig erfahrene Benutzer sehr schwierig.

3.4 Barrierefreiheit

Mit Plone erstellte Seiten erfüllen die Normen „Section 508“ und W3C-WAI (Web Accessibility Initiative) für barrierefreien Zugang. Durch den Einsatz von Cascading Style Sheets (CSS) und durch die klare Strukturierung des erzeugten HTML-Code können die Plone-Seiten außerdem auf beinahe allen Browsern dargestellt werden [9].

Das Aussehen einer Plone-Site lässt sich allein über andere CSS-Anweisungen stark verändern. Für die Ausgabe auf Druckern ist außerdem ein eigenes CSS vorhanden. Inhalte können in Plone über ein S5-basiertes Stylesheet als Browser-Präsentation angezeigt werden. Die Seiten in Plone werden mit Zope Page Templates (ZPT) hochgradig modular aufgebaut. ZPT ermöglicht hierbei die komplette Trennung von Content, Präsentation und Logik [8].

Visuelle Listen, also z. B. die Personalbar, die Document Actions und die Portal-Tabs wurden für blinde Benutzer optimiert. So erhielten visuell gruppierte Bedienungselemente auch eine semantische Struktur [6].

Das tabellenlose Layout von Plone basiert auf Big Johns Ordered Floats und auf Modifikationen von Tonico Strasser [6].

4 Bewertung

4.1 Vergleich der beiden CMS

Joomla! ist sehr intuitiv zu bedienen und bietet, speziell für kleine Projekte, umfangreiche Möglichkeiten. Die Arbeit mit Plone verlangt im Vergleich ein höheres Maß an Einarbeitung schafft aber auch eine deutlich höhere Flexibilität.

Die Tatsache, dass Plone auf Zope basiert bietet geübten Benutzern große Vorteile, da Zope prinzipiell darauf ausgerichtet ist den Inhalt von der Gestaltung der Webseite zu trennen.

Management	Joomla!	Plone
Advertising Management	Yes	Free Add On
Asset Management	Yes	Yes
Clipboard	No	Yes
Content Scheduling	Yes	Yes
Content Staging	No	Free Add On
Inline Administration	Yes	Yes
Online Administration	Yes	Yes
Package Deployment	No	Yes
Sub-sites / Roots	Yes	Yes
Themes / Skins	Yes	Yes
Trash	Yes	Free Add On
Web Statistics	Yes	Free Add On
Web-based Style/Template Management	Yes	Yes
Web-based Translation Management	Free Add On	Yes
Workflow Engine	No	Yes

Abbildung 6: Vergleich der Funktionen [2]

Die Anpassungsfähigkeit von Zope aufgrund vieler frei erhältlicher Erweiterungen ist enorm. Auch dass Zope sowohl die Funktion eines Application-Servers als auch die Funktion eines Datenbankservers anbietet ist ein Punkt, der diesbezüglich zu nennen wäre. Je nach Bedarf des Projekts können benötigte Funktionen hinzugefügt bzw. weggelassen werden.

Für Joomla! läßt sich die einfach zu erlernende Handhabung sowie die übersichtliche Installation und Inbetriebnahme als Vorteil anführen. Im Vergleich verläuft die Lernkurve bei Plone sehr steil [8]. Benutzer die sich neu in den Umgang mit Content-Management-Systemen einarbeiten müssen erzielen bei der Verwendung von Joomla! schnell Fortschritte und können durch die im Standard-Paket enthaltenen Themes und Erweiterungen auch mit wenig Übung bereits gute Ergebnisse erzielen.

Eine Übersicht über die im Bereich Management zur Verfügung stehenden Funktionen bietet Abbildung 6.

5 Fazit

5.1 Zusammenfassung

Beide in dieser Ausarbeitung vorgestellten Content-Management-Systeme haben ihre Vor- und Nachteile. Es hat sich gezeigt, dass die Ziele des Projekts und der Kenntnisstand über den die Benutzer und vor allem die Administratoren verfügen für die Wahl des richtigen Content-Management-Systems entscheidend sind.

Benutzer, die große Flexibilität einer gewissen Einfachheit vorziehen ist sicher die Verwendung von Plone zu empfehlen. Die zahlreichen Erweiterungen die für Plone zur Verfügung stehen, sowie die Möglichkeiten die Python als vollwertige Programmiersprache im Vergleich mit der Scriptsprache PHP bietet, machen Plone viel skalierbarer als Joomla!. Plone lässt sich genauer an die Bedürfnisse im Projekt anpassen.

Im Rahmen kleiner Projekte ist die Verwendung von Joomla! aber durchaus empfehlenswert. Der geringe Lernaufwand ist hierbei ausschlaggebend. Auch die Tatsache dass PHP viel weiter verbreitet ist als Python trägt momentan noch hierzu bei.

Im Bereich der Barrierefreiheit ist Plone überlegen, da das System von Grund auf die Grundsätze des barrierefreien Webs berücksichtigt hat.

Die geringe Zahl an bekannten Sicherheitslücken macht Plone im Bereich der Sicherheit überlegen.

5.2 Ausblick

Schon alleine die Tatsache dass Plone Python einsetzt und Joomla! PHP wird dazu führen, dass beide Systeme weiterhin ihre Anwender haben werden. Da beide Projekte als Open-Source zur Verfügung stehen ist wahrscheinlich, dass eine Weiterentwicklung stattfindet.

Literatur

- [1] Autor (Screenshot)
- [2] <http://www.cmsmatrix.org>, zuletzt besucht am 15.01.2009
- [3] [http://www.joomla-tutorials.de/ueber-joomla/google-summer-of-code.html?view=mediawiki&article=Zope_\(Webanwendungsserver\)](http://www.joomla-tutorials.de/ueber-joomla/google-summer-of-code.html?view=mediawiki&article=Zope_(Webanwendungsserver)),
zuletzt besucht am 12.11.2008
- [4] http://www.mz.jena.de/joomla/images/stories/logos/joomla_logo_vert_color.png,
zuletzt besucht am 08.01.2009
- [5] <http://www.zope.de/redaktion/dzug/nachrichten/cms-plone-3-0-fast-fertig/image>,
zuletzt besucht am 12.11.2008
- [6] Interview mit Michael Zeltner,
<http://alt.dzug.org/organisation/presse/PloneInterviews/PloneInterviewZeltner/>,
zuletzt besucht am 15.01.2009
- [7] Joomla! 1.5 stable, Hagen Graf, Addison-Wesley, 2008, ISBN 978-3-8273-2531-0
- [8] Open Source Content Management Plone 3, Jan Ulrich Hasecke,
<http://hasecke.com/texte/plone-3/view>, zuletzt besucht am 15.01.2009
- [9] Plone (Onlinebuch), Andy McKay,
<http://www.zope.de/dokumentation/plonebuch>, zuletzt besucht am 21.12.2008