

Alternative: Open-Source CMS

Content-Management mit Zope und Plone – Ein Erfahrungsbericht

Matthias Bärwolff, Frank Pallas – Technische Universität Berlin – Informatik und Gesellschaft
{mbaer, pallas}@cs.tu-berlin.de

Professionelle Systeme für das Content-Management von Webseiten sind für viele mögliche Anwender zu teuer. Oftmals werden auch viele der gebotenen Funktionen gar nicht benötigt. Doch es gibt Alternativen wie zum Beispiel das Open-Source-Gespinn aus Zope und Plone.

Trotz vollmundig angekündigter Bildungsoffensive ist die Finanzausstattung deutscher Universitäten weiterhin alles andere als rosig. Kleinen und mittelständischen Unternehmen geht es kaum anders. Was also tun, wenn man sich trotz leerer Kassen einen ansehnlichen Internetauftritt jenseits von statischen HTML-Seiten wünscht, der leicht zu pflegen ist und außerdem neben der Außenwirkung einige weitere Aufgaben übernehmen soll?

In unserem Fall handelt es sich um einen vergleichsweise kleinen Lehrstuhl an der Technischen Universität Berlin. Die Online-Präsenz bestand bisher aus einer Fülle statischer Webseiten, die über den fakultätseigenen Webserver ausgeliefert wurden. Über Jahre hinweg ist so eine Struktur gewachsen, die ihren Zweck zwar erfüllte, die aber im Laufe der Zeit immer schwerer zu verwalten war und zudem kaum Lösungen für neue Anforderungen bot. Darüber hinaus war das äußere Erscheinungsbild zumindest renovierungsbedürftig.

Was also tun? Die Anforderungen waren vergleichsweise klar: Das neue System sollte das Einstellen und Ändern von Inhalten deutlich vereinfachen, denn Universitätsmitarbeiter werden nicht für die Pflege statischer HTML-Seiten bezahlt. Um den Kontakt mit ausländischen Forschungseinrichtungen zu pflegen und möglichst auszubauen sollte das Angebot zumindest auch in englischer Sprache verfügbar sein. Die Unterstützung bei der Verwaltung von Lehrveranstaltungen sollte deutlich über das bisherige Maß hinausgehen und zumindest ein minimales Studentenmanagement beinhalten. Außerdem sollte für die in der praktischen Lehre durchgeführten Projekte eine Plattform geschaffen werden, die ein komfortables Austauschen von Informationen und Dokumenten unter den Beteiligten ermöglicht.

Dass bei derartigen Anforderungen schnell die Grenzen statischer – auch generierter – Webseiten erreicht sind, ergibt sich von selbst. Ein Content-Management-System musste her. Da zu diesem Zeitpunkt noch nicht absehbar war, auf welcher Plattform dieses laufen sollte (MacOS X oder Solaris), möglichst ein plattformunabhängiges. Wir entschieden uns also für ein System, das aus den beiden Open-Source-Lösungen Zope und Plone besteht.

Zope und Plone

Das Fundament des gesamten Systems ist Zope – ein Open-Source Application-Server auf Basis der Programmiersprache Python. Zope bietet neben einem integrierten eigenem Webserver (Zserver) und einer integrierten Datenbank (Zope Object DataBase, ZODB) bereits von Hause aus Möglichkeiten zum Zugriff per FTP oder WebDAV. Benutzermanagement, eine Suchfunktion und vor allen Dingen das dynamische Generieren von Webseiten auf Basis von Templates und Skripten existieren ebenfalls. Für Inhalte stehen bereits einige Basistypen zur Verfügung.

Die eigentliche Stärke von Zope besteht jedoch in der Erweiterungsfähigkeit um zusätzliche Module für unterschiedlichste Zwecke. Auf einige dieser Module („Produkte“) werden wir im Folgenden noch genauer eingehen. Die für unsere Zwecke bedeutendste Erweiterung ist Plone, ein Content-Management-System, das neben einigen erweiterten Typen wie Internet-Links oder auch News bereits ein tragfähiges Grundgerüst für eine dynamische und mehrsprachige Internetpräsenz liefert. Für Windows und MacOS X stehen unter <http://plone.org/downloads> vollständige Installationspakete bereit, die sowohl Zope als auch Plone beinhalten und bereits

nach Minuten den Betrieb eines ausgewachsenen Portals ermöglichen. Für andere Betriebssysteme wie Linux, *BSD oder Solaris sind die üblichen Tarballs verfügbar. Eine Migration zwischen unterschiedlichen Betriebssystemen ist ebenfalls möglich. In unserem Fall wurde ein Testsystem unter MacOS X betrieben und dieses dann auf ein größeres Linux-System (Gentoo-Linux) übertragen.

Erste Anpassungen

Nach der Installation präsentiert sich die Webseite in einem modernen weiß-grauen Design. Das Layout der Seiten wird komplett über CSS und per CSS positionierte DIV's umgesetzt – die generierten Seiten sind also konform zum W3C-Standard XHTML. Natürlich möchte man das Layout zuerst einmal den eigenen Bedürfnissen anpassen. Die Konfiguration erfolgt dabei weitest gehend über eine Management-Konsole im Browser. Mittels dieser lassen sich die verwendeten Style-Sheets und damit die dargestellten Farben, Schriftarten usw. anpassen, eigene Logos ablegen oder auch der Titel des gesamten Portals – in unserem Fall der Name des Lehrstuhls – festlegen. Besondere Bedeutung haben auch einige Templates: Mit ihnen lässt sich das gesamte Layout der generierten Seiten anpassen. Diese Templates beinhalten dabei nicht nur HTML-Code sondern auch so genannte TAL-Statements, die vor der Seitenauslieferung ausgeführt werden. So lassen sich beispielsweise wechselnde Hintergrundfarben für Tabellenzeilen oder auch so genannte Breadcrumbs – Brotkrumen, die einem den Weg zurück zum Ausgangspunkt (Der Startseite) weisen – einbinden.

Üblicherweise verfügt ein Plone-Portal auf der linken Seite über unterschiedliche Kästen, die so genannten Portlets. Das oberste dieser Portlets dient im Allgemeinen der Navigation, stellt also eine Baumstruktur der gesamten Site dar. Weitere häufig verwendete Portlets sind News, Kalender oder auch eine Eingabemaske für den Login registrierter Benutzer. Die Anordnung der Portlets auf der Seite lässt sich mittels der Templates ebenfalls frei konfigurieren.

Mit diesen Mechanismen erhält das Portal einen ersten individuellen Look und kann vor allen Dingen an das übliche Farb- und Schriftschema des Betreibers angepasst werden.

Inhalte verwalten

Ohne sinnvolle Inhalte taugt das beste Portal nichts. Wie also bringt man Inhalt in ein Plone-Portal? Plone kennt dazu verschiedene Objekttypen (Datei, Internet-Link, Bild, Ordner), die, entsprechende Berechtigung des jeweiligen Benutzers vorausgesetzt, an jeder Stelle des Portals hinzugefügt werden können. Jedes Objekt kann dabei neben einer id (entspricht der jeweiligen URL) mindestens mit einem Titel sowie einer kurzen Beschreibung versehen werden. Um sicher zu stellen, dass das Portal nicht durch versehentliche Falscheingaben oder ähnliches „entstellt“ wird, muss ein solches Objekt nach seiner Erzeugung zusätzlich in einem weiteren Schritt veröffentlicht werden. Erst dann ist es auch für die Allgemeinheit sichtbar. Die Vergabe von Sperrfristen, also beispielsweise die Veröffentlichung erst am übernächsten Vormittag, ist dabei ebenfalls möglich. Durch Anpassung der entsprechenden Workflows des Portals lässt sich auch umsetzen, dass ein neu erstelltes Objekt defaultmäßig "privat" – also auch für andere Portalmitglieder nicht sichtbar – ist.

Besondere Bedeutung kommt jedoch dem ebenfalls bereits in der Grundkonfiguration enthaltenem Objekttypen "Dokument" zu. Hiermit lassen sich kurze Notizen oder Gedankenblitze aber auch Pressemitteilungen oder ganze Texte verfassen. Als mögliches Format steht dafür neben einfachem Text und HTML auch "structured text" zur Verfügung: Deutlich einfacher als HTML und dennoch mit der Möglichkeit, Aufzählungen, Links, grundlegende Textformatierungen und Fußnoten einzubetten. Dadurch wird der Aufwand für das Verfassen der meisten Inhalte deutlich geringer und die Autoren können sich auf das Wesentliche – den Inhalt – konzentrieren.

Erweiterungen

Wie oben beschrieben liegt die eigentliche Stärke des Gespanns von Zope und Plone in der Erweiterbarkeit. Viele Open-Source-Projekte befassen sich mit der Entwicklung von Erweiterungen für die unterschiedlichsten Zwecke. Einige dieser Erweiterungen (Produkte)

kommen auch auf unserem Portal zum Einsatz und sollen daher kurz vorgestellt werden. Die Entsprechenden Download-Links geben wir dabei nicht explizit an, eine kurze Online-Recherche führt aber im Regelfall schnell zum Ziel.

Eine wichtige Anforderung an das Portal bestand in der Unterstützung von Projekt- und Arbeitsgruppen durch die Möglichkeit zur Ablage von Arbeitsergebnissen und zur gruppeninternen Kommunikation. Hierfür kommen die Produkte „Workgroup“ und „CMFBoard“ zum Einsatz. Das Produkt Workgroup stellt dabei einen Arbeitsbereich zur Verfügung, der nur für Gruppenmitglieder einsehbar ist, und ermöglicht das Versenden von Mails an alle Mitglieder. Das Produkt CMFBoard ermöglicht das Einrichten von Diskussionsforen, wie sie nahezu überall im Internet anzutreffen sind, und unterstützt so die Projektarbeit. Derartige Bereiche für Arbeitsgruppen unterstützen in unserem Fall Arbeits- und Projektgruppen sowohl in der Forschung als auch in der Lehre.

Eine Online-Präsenz an einer deutschen Universität muss, um den Kontakt mit ausländischen Forschern erst zu ermöglichen, zumindest zweisprachig aufgebaut sein. Für Unternehmen mit internationalem Anspruch dürfte ähnliches gelten. Hierfür existieren ebenfalls diverse Erweiterungen für Plone. In unserem Fall kommen dabei die Produkte „I18NFolder“, „I18NLayer“ sowie das „PloneLanguageTool“ zum Einsatz. Bereits damit lassen sich sowohl Ordner als auch alle dateiähnlichen Objekte (Bild, Datei, Dokument usw.) internationalisieren. So wird beispielsweise von einer Pressemitteilung sowohl eine englische als auch eine deutschsprachige Version erstellt. Beide Versionen sind dann unter der selben URL verfügbar, die jeweils anzuzeigende bzw. auszuliefernde Version wird vom Portal anhand der Browser-Einstellungen identifiziert. Besucher können zudem die Sprache manuell auswählen.

Um den möglichen Leistungsumfang eines Systems wie Plone auch auszunutzen kommt das Produkt „CMFDynamicPage“ zum Einsatz, mit dem sich Seiten generieren lassen, die einem Datenbank-Report sehr ähnlich sind. DynamicPages werden bei uns verwendet, um beispielsweise in einem Ordner alle Unterordner sowie die 15 neuesten Objekte unterhalb dieses Ordners anzuzeigen. Ein weiterer Verwendungszweck besteht im Auflisten aller Veröffentlichungen, zu denen ein bestimmter Autor beigetragen hat oder die einem bestimmten Forschungsschwerpunkt zugerechnet werden.

Daneben kommen noch einige weitere Produkte zum Einsatz: Der WYSIWYG-Editor „Epoz“, „PloneExternalFile“ zum Ablegen großer Dateien im Filesystem anstatt in der Datenbank sowie „SimpleBlog“ und „Zwiki“ - erwartungsgemäß zum Erstellen von Blogs und Wiki-Seiten.

Eigene Objekttypen

Für einige Einsatzzwecke reichen jedoch die öffentlich verfügbaren Objekttypen nicht aus. In unserem Fall waren daher die zusätzlichen Objekttypen "Student", "Veranstaltung", "Veranstaltungsteilnahme" und "Publikation" nötig. Solche noch nicht verfügbaren Objekttypen lassen sich mit dem Produkt "Archetypes" schnell und komfortabel selbst erstellen. Die dazu nötigen (minimalen) Kenntnisse der Programmiersprache Python lassen sich schnell aneignen, auch wenn die Dokumentation zu Archetypes an einigen Stellen durchaus noch ausbaufähig ist.

Eigene Objekttypen werden mit Archetypes auf Basis eines "Schemas" erstellt: Neben einem Grundgerüst müssen lediglich die nötigen Felder und die gewünschten Eingabewidgets definiert werden. Für den Objekttypen "Student" sind dies Name (String), Vorname (String), Matrikelnummer (6 Ziffern) sowie die Benutzerkennung (String) und eine optionale Telefonnummer (String). Neben der einfachen Definition von Feldern lassen sich natürlich weitere Automationsschritte durchführen – in unserem Fall etwa das automatische Anlegen eines Benutzers nach dem Anlegen eines Studentenobjektes – dazu werden lediglich den eigenen Bedürfnissen entsprechende Python-Methoden in die Typdefinitionen eingefügt. Zur weiteren Dokumentation sei auf die öffentlich verfügbaren Quellen verwiesen (s.u.).

Werden Studentenobjekte noch lediglich für die interne Verwaltung in der Lehre benötigt, so kommt beim Objekttyp "Publikation" hinzu, dass einige Felder (z.B. eine Kurzzusammenfassung eines Artikels, das "Abstract") internationalisiert werden müssen. Dazu ließen sich auch die oben genannten Mechanismen nutzen. Das würde jedoch dazu führen,

dass alle Informationen mehrfach eingegeben werden müssten. Bei einer Publikation sind jedoch Angaben wie Autoren, Titel oder Veröffentlichungsdatum in allen Sprachen gleich. Aus diesem Grund haben wir hier das Produkt I18NArchetypes eingesetzt. Damit ist es möglich, innerhalb des Schemas zu einem Objekttyp einzelne Felder als "internationalisierbar" zu definieren – etwa die Kurzzusammenfassung. Dadurch wird der Aufwand für die Internationalisierung eigener Objekttypen auf das wirklich notwendige Maß beschränkt.

Eigene Skripte

Als Herausgeber des Open-Source-Jahrbuches war es für uns notwendig, eigene Skripte für den Bestellvorgang auszuführen. Dabei sollen die Bestelldaten nach der Eingabe in einem Webformular auf Vollständigkeit und Plausibilität überprüft werden und der Benutzer bei Bedarf um Korrektur der Eingaben gebeten werden. Sind die Daten vollständig und plausibel, so soll eine Mail mit den entsprechenden Angaben verschickt werden. Derartige Vorgänge lassen sich mit Hilfe von Python-Skripten und Page-Templates sowie dem in Plone enthaltenen Produkt "CMFFormController" ebenfalls umsetzen, erfordern jedoch eine etwas tiefer gehende Einarbeitung in die Funktionsweise von Plone bzw. Zope.

Link-Struktur beibehalten

Absolut unverzichtbar für eine Migration wie die hier beschriebene ist es oftmals, dass bestehende URLs weiterhin ihre Gültigkeit behalten. Da in unserem Fall mit der Migration eine teilweise Umstellung der Ordnerstruktur einherging, mussten einige URLs umgeleitet werden. Das Produkt "ExoRedirect" leistet dies in Form regulärer Ausdrücke, die ähnlich wie die für Apache gebräuchlichen Rewrite-Rules eingesetzt werden können. Wo es möglich war, wurden die statischen Inhalte der alten Internetpräsenz in das neue Portal übernommen und sind so mittels der genannten Redirects weiterhin erreichbar.

In einigen Fällen erwies sich jedoch ein solches Vorgehen als eindeutig zu aufwändig oder gar unmöglich. Um den Zugriff auf die entsprechenden Inhalte jedoch auch weiterhin zu ermöglichen, entschlossen wir uns dazu, die gesamte alte Website innerhalb eines Ordners "/oldstatic" abzulegen. Das Produkt "Ape" ermöglicht es dabei, eine gesamte im Dateisystem liegende Ordnerstruktur in die hinter dem Portal liegende Datenbank zu "mounten". Damit dies wie gewünscht funktioniert, musste der Dateibaum jedoch im Vorhinein mit einigen Skripten bearbeitet werden – Zope bzw. Plone weigert sich ansonsten vehement, Webseiten auszuliefern, die nicht XHTML-konform sind. Im gleichen Atemzug wurde zudem in jede HTML-Seite des Ordners "/oldstatic" ein Hinweis eingefügt, dass die angezeigte Seite Teil der alten Präsenz ist, die nicht weiter gepflegt wird.

Wird nun eine URL aufgerufen, die nicht innerhalb des Portals existiert, so generiert Plone eine 404-Fehlermeldung. Eigene Erweiterungen überprüfen daraufhin, ob ein entsprechendes Dokument innerhalb der oldstatic-Sektion vorhanden ist und fügen bei Bedarf einen entsprechenden Hinweis nebst Link in die Fehlermeldung ein. Auf eine automatische Weiterleitung haben wir verzichtet, um die neue Präsenz möglichst publik zu machen.

Enterprise

Die Kombination aus Zope und Plone ist beileibe kein bloßes Spielzeug: Nasa (siehe Online-links am Ende) und Lufthansa setzen es für einige besondere Webseiten ebenso ein wie das Marriott New York Marquis oder die Bank Crédit Municipal de Paris. Wenn die Anforderungen an ein solches System jedoch steigen, dann reichen die internen Mechanismen für Datenablage und Webserving unter Umständen nicht mehr aus. Diese Aufgaben lassen sich allerdings auch mit externen, leistungsfähigeren Komponenten durchführen. So kann beispielsweise anstatt des internen Zservers ein Apache zum Einsatz kommen oder als Datenbank mit entsprechenden Konnektoren etwa MySQL, SAPDB oder auch Oracle. Auf derartige Erweiterungen verzichten wir in unserem Fall jedoch bisher weitestgehend.

Fazit

Den Anforderungen eines kleinen Unternehmens oder einer Universitätsabteilung wird das

Gespann aus Zope und Plone durchaus gerecht. Schnell und unkompliziert kommt man zu einem eigenen Webserver, dessen Funktionalität weit über das (immer noch weit verbreitete) Ausliefern statischer, vielleicht sogar manuell erstellter HTML-Seiten hinausgeht. Das Layout lässt sich mit ein wenig Wissen über CSS und XHTML unkompliziert den eigenen Wünschen und Erfordernissen anpassen und dank der offenen Struktur stehen in Open-Source-Projekten verschiedenste Erweiterungen zur Verfügung, die auch den einen oder anderen ausgefallenen Wunsch erfüllen. Dank der Betriebssystemunabhängigkeit der verwendeten Programmiersprache Python lässt sich Plone sowohl auf einem Windows-Arbeitsgruppenserver als Intranet-Lösung nutzen als auch auf einem Linux-Rechner, der das externe Internet mit einer ansprechenden internationalisierten Präsenz bedient. Die Nutzer brauchen dabei keinerlei Kenntnisse von XHTML, FTP und ähnlichem – sie können das System einfach benutzen.

Die von uns verwendeten Erweiterungen waren im Regelfall gut dokumentiert und wenn doch einmal ein Problem auftauchte, dann hatte die sehr lebendige Entwickler- und Anwendergemeinde meist bereits eine Antwort darauf.

Den Anforderungen, die große Zeitschriften oder Fernsehsender an ein CMS stellen, werden von Plone vielleicht nicht erfüllt. Es ist jedoch wesentlich mächtiger als viele "selbstgebaute" LAMP-Systeme. Die hier beschriebene Webseite ist zu erreichen unter <http://ig.cs.tu-berlin.de>. Die Autoren stellen außerdem allen Interessierten auf Nachfrage gern die von ihnen angepassten Dateien zur Verfügung.

Links:

<http://zope.org>

<http://plone.org>

<http://http://plone.org/about/sites/>

Plone-Sites:

<http://www.jpl.nasa.gov/>

<http://mars.telascience.org/home>

<http://www.creditmunicipal.fr/>

Die Autoren sind tätig am Lehrstuhl für Informatik und Gesellschaft der TU Berlin und forschen unter anderem im Bereich Open-Source und Mobile Computing.

<http://ig.cs.tu-berlin.de/ma/mb/ap>

<http://ig.cs.tu-berlin.de/ma/fp/ap>