

Proseminar
Website-Management-Systeme
Zope/CMF

Andreas M. Weiner

Technische Universität Kaiserslautern
Fachbereich Informatik
AG Softwaretechnik
Betreuer: Dipl. Inf. Christian Stenzel

Inhaltsverzeichnis

1	Einleitung	2
1.1	Die Stärken von Zope/CMF	2
2	Die Architektur von Zope	2
2.1	Veröffentlichung von Objekten	2
2.2	Akquisition	3
2.3	Through-The-Web Management	4
2.4	Versionierung und Undo-Mechanismus	4
2.5	Benutzerverwaltung und Sicherheitsaspekte	4
2.6	Weitere Zope Services	5
3	Erstellen von Webapplikationen mit Zope	5
3.1	Eine kurze Einführung in DTML	5
4	Das Content-Management Framework (CMF)	6
4.1	CMF Inhalte	6
4.2	CMF Dienste	6
4.2.1	Membership Service	6
4.2.2	Workflow Service	6
4.2.3	Security Service	7
4.2.4	Look and Feel Service	7
4.2.5	Search Service	7
5	Fazit	7

1 Einleitung

Web-Content-Managementsysteme haben sich inzwischen als Werkzeuge zur Beschaffung, Erzeugung, Aufbereitung, Verwaltung und Präsentation von Inhalten im World Wide Web etabliert. [8]

Zope (**Z** **O**bject **P**ublishing **E**nvironment) ist ein *Open-Source Webapplikationsserver*, der hauptsächlich in der objektorientierten Programmiersprache Python¹ implementiert wurde.

Begriffsklärung: Webapplikation

Unter dem Begriff einer *Webapplikation* versteht man ganz allgemein eine Anwendung mit graphischer Oberfläche, die im World Wide Web läuft: typischerweise entweder gestaltet mit reinem HTML, WML für mobile Geräte oder vielleicht mit einem Java-Applet für anspruchsvollere Clients. [1] Der Benutzer bzw. der Browser kann normalerweise nicht unterscheiden, ob der von ihm kontaktierte Webserver eine statisch-erzeugte Website oder eine Webapplikation zur Verfügung stellt. Im Gegensatz zu einer statischen Website werden die "Seiten" einer Webapplikation dynamisch erzeugt. [4]

Zope alleine ist *kein* funktionsfähiges Web-Content-Managementsystem (im folgenden mit *WCMS* abgekürzt). Das *Zope Programmgerüst*² stellt eine transaktionsbasierte Objektdatenbank (ZODB) zur Verfügung, die nicht nur Inhalte und Daten, sondern auch dynamische HTML Templates, Python-Skripte und Verbindungen zu relationalen Datenbank-Management-Systemen (RDBMS) speichern kann. Auf diesem Fundament lassen sich zusammen mit dem *CMF* (**C**ontent **M**anagement **F**ramework) *WCMS* erstellen, die an die Bedürfnisse des Kunden angepasst sind.

1.1 Die Stärken von Zope/CMF

Im Gegensatz zu vielen anderen WCMS ist Zope/CMF kostenlos unter einer Open-Source Lizenz (Zope Public License) verfügbar und umfasst alle nötigen Bestandteile, um Webapplikationen wie z.B. ein *WCMS* zu erstellen. Um eine Webapplikation mit Zope zu entwickeln, benötigt der Benutzer lediglich einen Webbrowser mit dem er Zugang zum *ZMI* (Zope Management Interface), der zentralen Entwicklungsumgebung von Zope, erhält. Zope zeichnet sich durch ein flexibles und erweiterbares Sicherheitsframework aus, das leicht in bestehende Authentifizierungssysteme wie LDAP³ eingebunden werden kann. Durch die Verfügbarkeit einer flexiblen Benutzerverwaltung und Rechtevergabe auf unterster Ebene, sowie durch Versionierungs-, Undo- und Historytools eignet sich Zope bestens für den Einsatz im Mehrbenutzerbetrieb.

2 Die Architektur von Zope

Im Gegensatz zu anderen dateibasierten Web-Templating-Systemen wie ASP oder PHP ist Zope von Grund auf objektorientiert. Mit Zope ist es möglich Daten, Anwendungslogik und Präsentation von einem Punkt aus zu verwalten. Zope unterstützt und entlastet den Entwickler bei wichtigen Details der Entwicklung von Webapplikationen wie z.B. der persistenten Speicherung und Integrität der Daten sowie bei der Zugriffskontrolle auf die Daten. Damit wird erreicht, dass sich der Entwickler auf das eigentliche Problem konzentrieren kann. Die Anwendungslogik wird primär in der Programmiersprache Python implementiert. Desweiteren bietet Zope zwei Möglichkeiten zur Erstellung von Text, HTML und XML Templates an: *DTML* (**D**ocument **T**emplate **M**arkup **L**anguage) und *ZPT* (**Z**ope **P**age **T**emplates).

2.1 Veröffentlichung von Objekten

Zope wurde mit der Erkenntnis entwickelt, dass das Web von Grund auf objektorientiert ist. Die Objektstruktur von Zope ist hierarchisch. URLs werden auf Objekte in der Zope-Umgebung abgebildet.

¹Für eine Einführung in Python siehe: [5]

²Ein Programmgerüst ist ein erweiterbares und anpassbares System von Klassen, das für einen allgemeinen, übergeordneten Aufgabenbereich eine Kernfunktionalität mit entsprechenden Bausteinen bereitstellt.[14]

³LDAP ist eine Software zur Integration verschiedener Verzeichnisdienste unterschiedlicher Hersteller. Für mehr Informationen siehe: [12] und [13]

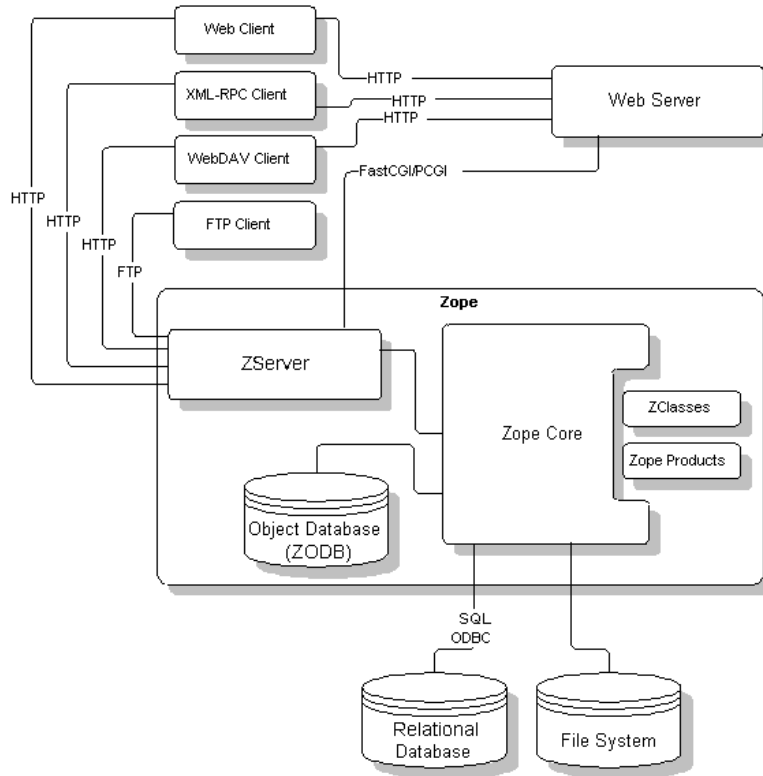


Abbildung 1: Die Archtiketur von Zope [4]

Eine URL ist somit ein eindeutiger Pfad zu einem Objekt in der Zope-Umgebung, an das mit Hilfe des HTTP-Protokolls Nachrichten gesendet bzw. von dem Nachrichten empfangen werden können. Der Pfad zu einem Objekt besteht aus den *IDs* der Folder-Objekte und der *ID* des Objektes, die durch das / Zeichen voneinander getrennt werden. Gibt es z.B. im Root-Verzeichnis des Servers ein Folder-Objekt "Bob", dann ist sein Pfad: /Bob. Wäre "Bob" ein Unterverzeichnis von "Uncles", dann wäre sein Pfad: /Uncles/Bob

Die wesentlichen Schritte, die der Zope-Server durchführt, um ein Objekt zu veröffentlichen sind:

1. Der Webbrowser sendet eine Anfrage an den Zope Server. Die Anfrage wird durch eine URL der Form `protocol://host:port/path?querystring` spezifiziert.
Ein Beispiel dafür wäre: `http://www.zope.org:8080/Resources?batch_start=100`
2. Zope spaltet die URL in ihre Einzelbestandteile "host", "port", "path" und "query string" auf.
In unserem Beispiel sind das: `http://www.zope.org`, `8080`, `/Resources` und `?batch_start=100`
3. Zope sucht in seiner Objektdatenbank nach dem Objekt, zu dem der Pfad `/Resources` führt.
4. Zope "führt" das Objekt mit dem `query string` als Parameter aus. Der Zustand des Objektes verändert sich in Abhängigkeit des übergeben Parameters.
5. Liefert die "Ausführung" des Objektes einen Rückgabewert, so wird dieser zurück an den Browser gesendet. Normalerweise liefert ein typisches Zope-Objekt eine HTML-Seite oder ein Bild zurück.
6. Die Daten werden vom Browser interpretiert und dem Benutzer angezeigt

2.2 Akquisition

Zope verwendet für die Festlegung der Eigenschaften von Objekten nicht nur die klassische Beziehung "Objekt A ist ein Objekt B", sondern außerdem die sogenannte Akquisition. Am ehesten vergleichbar ist

diese Technik mit den Gültigkeitsbereichen von Variablen in Programmiersprachen, die die Definition von Blöcken erlauben. Dort sind Variablen in allen im aktuellen Gültigkeitsbereich definierten Blöcken bekannt, nicht aber außerhalb.

Akquisition sorgt für die Vererbung von Eigenschaften der einer URL übergeordneten Ordner. Zum Beispiel kann man im Root-Ordner eines Angebots den Autor und den Release-Stand als Eigenschaften definieren, die alle untergeordneten Objekte verwenden können. Für einzelne Unterordner lassen sich separate Einstellungen vornehmen. Nicht nur Dateninhalte, sondern auch besondere Fähigkeiten werden vererbt: ist zum Beispiel ein Ordner in der Lage, Mail zu versenden, sind es auch alle Objekte, die hier beziehungsweise in Unterverzeichnissen angesiedelt sind. [6]

Beispiel

Sei eine DTML-Methode *acq_test* im Root-Verzeichnis / des Webservers gegeben durch:

```
<html>
  <body>
    <p>Ich wurde im <dtml-var id> Ordner aufgerufen!</p>
  </body>
</html>
```

Aufruf: `http://mywebserver:8080/acq_test`

Ausgabe: Ich wurde im / Ordner aufgerufen!

Aufruf: `http://mywebserver:8080/test/acq_test`

Ausgabe: Ich wurde im *test* Ordner aufgerufen

Der Akquisition ist es zu verdanken, dass die DTML-Methode auch im 2. Fall korrekt ausgeführt wird, obwohl sich die DTML-Methode *acq_test* nicht unmittelbar im Verzeichnis *test* befindet.

2.3 Through-The-Web Management

Sämtliche Managementaufgaben und die gesamte Anwendungsentwicklung in Zope erfolgen mit Hilfe des *ZMI* (**Z**ope **M**anagement **I**nterface), auf welches mit einem gewöhnlichen Webbrowser zugegriffen werden kann. Das ZMI stellt das Zope Objektsystem auf ähnliche Weise dar, wie z.B. der Microsoft Explorer oder der Konqueror des KDE das Dateisystem. Objekte können auch hier in der Objekthierarchie verschoben und manipuliert werden. (Es können z.B. neue Python-Skripte erstellt und getestet werden).

2.4 Versionierung und Undo-Mechanismus

Zope verfügt über ein hervorragendes Versionsmanagementsystem. Ein Benutzer kann sogenannte *Sessions* eröffnen, innerhalb denen er Änderungen an seiner Objektstruktur durchführen kann. Diese Änderungen sind für andere Benutzer per se nicht sichtbar. Die einzelnen Aktionen können durch den *Undo*-Mechanismus jederzeit wieder rückgängig gemacht werden. Wenn dann das gewünschte Aussehen und Verhalten der Objekte erreicht ist, kann der Sitzungsleiter diese abschließen und damit auf einmal seine Änderungen gültig machen, ohne einen *Staging-Server*⁴ zu verwenden und ohne den Zope-Server neu starten zu müssen.

2.5 Benutzerverwaltung und Sicherheitsaspekte

Zope bietet vier wesentliche Komponenten für das Sicherheitsmanagement: *Benutzer*, *Rollen*, *Rechte* und *Akquisition*. *Benutzer* (besser: Benutzerobjekte) dienen der Authentifizierung gegenüber dem System (in der Regel eine Nutzer-/Passwortabfrage). *Rollen* bilden das Bindeglied zwischen den Benutzern und Rechten, sie entsprechen in etwa dem, was andere Systeme durch Gruppen realisieren. Die Autorisierung für Operationen legen die *Rechte* fest. Objekte können auch Rechte erzeugen, die in den Zope-Standardobjekten

⁴Auf einem Staging-Server werden neue oder veränderte Webseiten temporär gespeichert und getestet, bevor sie veröffentlicht werden.

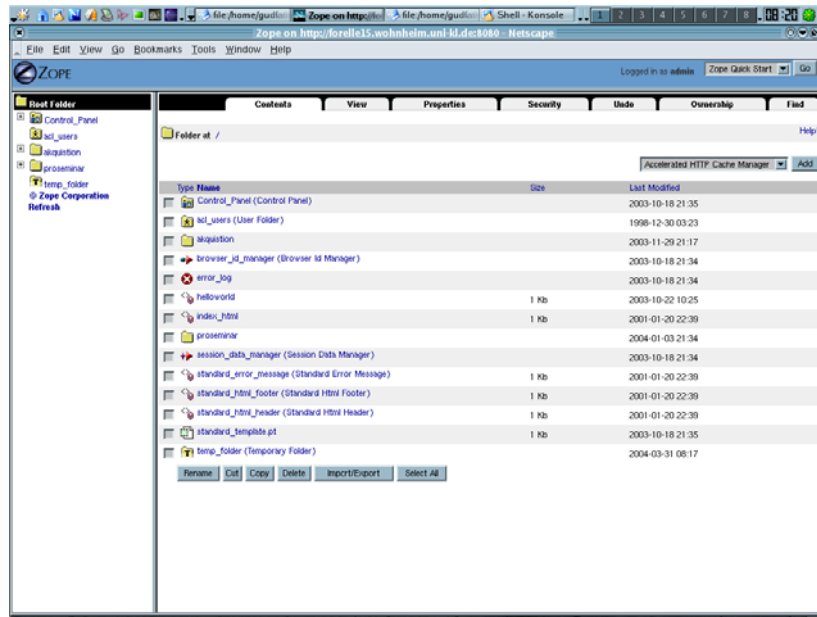


Abbildung 2: Das Zope Management Interface

nicht vorgesehen sind. Schließlich sorgt die *Akquisition* dafür, dass sich Rechte in der URL-Hierarchie auch nach “unten” durchsetzen. Auf diese Weise ist es möglich, die Benutzerverwaltung in einem weiter “oben” liegenden Ordner anzusiedeln und so den gesamten darunter liegenden Baum zu kontrollieren. [6]

2.6 Weitere Zope Services

Mit dem *ZCatalog*-Service verfügt Zope über einen schnellen und flexiblen Mechanismus zur Indizierung, Archivierung und zum Durchsuchen von Inhalten. Durch die Verwendung des *Caching Service* kann die Antwortzeit von umfangreichen Skripten durch temporäres Zwischenspeichern der Ergebnisse erheblich verringert werden. Als Add-ons sind Schnittstellen zum Zugriff auf alle gängigen relationalen Datenbanksysteme wie z.B. Oracle, Sybase, DB2 und MySQL verfügbar. Das *Site Error Log*-Objekt stellt Fehler- und Debugginginformationen in Echtzeit zur Verfügung. Desweiteren verfügt Zope über einen *Virtual Host Service*, einen *Sessioning Service* und einen *Internationalization Service*.

3 Erstellen von Webapplikationen mit Zope

3.1 Eine kurze Einführung in DTML

Zope bietet mit der serverseitigen *DTML* (**D**ocument **T**emplate **M**arkup **L**anguage) eine einfache Möglichkeit dynamische Webseiten zu erstellen. Jedes Ordnerobjekt kann DTML-Methoden enthalten. Gibt es z.B. einen Ordner `meinOrdner`, der eine DTML-Methode `MeineMethode` enthält, dann ist `MeineMethode` eine Methode des Ordnerobjektes `meinOrdner`. Kann Zope DTML Variablen bzw. Methoden im aktuellen Objekt nicht finden, so wird automatisch in deren Container-Objekten nachgesehen (in deren Elternverzeichnis). Dadurch ist es Objekten möglich, Inhalte und Verhalten von deren Elternobjekten zu akquirieren.

DTML Kommandos werden in Form von Tags geschrieben die mit `dtml-` beginnen. Dynamische Inhalte werden dadurch erzeugt, dass man HTML und DTML Tags miteinander kombiniert.

Beispiele

Folgende DTML Anweisung sorgt dafür, dass der Inhalt der Datei `texttinsert` in die aktuelle Datei kopiert wird:

```
<dtml-var texttinsert>
```

Der `dtml-in` Tag bietet die Möglichkeit über Listen zu iterieren, ganz egal ob es sich dabei um einen Teil des Verzeichnisbaums, ein Python-Skript oder um eine SQL Abfrage handelt. Folgendes Codebeispiel iteriert über das Ergebnis des Skriptes `actors` und schreibt den Inhalt in eine HTML Tabelle:

```
<table border="1">
  <th>Name</th>
  <dtml-in actors>
    <tr><td>
      <dtml-var sequence-item>
    </td></tr>
  </dtml-in>
</table>
```

4 Das Content-Management Framework (CMF)

Wie bereits weiter oben ausgeführt wurde, ist Zope kein *WCMS*. Zope erlaubt es, verschiedene Objekte zu verwalten und im Web zu veröffentlichen. Ein *WCMS* dient aber nicht nur der Veröffentlichung von Inhalten. Der Managementbereich wird von Zope nur sehr rudimentär abgedeckt.

Das *CMF* (**C**ontent **M**anagement **F**ramework) ist ein *Zope Product*⁵, das den Zope Server um *WCMS*-Fähigkeiten erweitert. Das CMF gestattet es dem Entwickler *WCMS* zu entwickeln, die an die Erfordernisse und Wünsche des Kunden angepasst sind. Mit dem CMF ist es möglich, Portale oder Websites für das Intranet zu erzeugen.

4.1 CMF Inhalte

Eine zentrale Idee des CMF ist es, dass alle digitale "Werte" wie z.B. Artikel, Berichte, Graphiken und Klänge als Inhaltsobjekte behandelt werden. Diese haben alle gewisse gemeinsame Eigenschaften. Inhaltsobjekte unterliegen gewissen Sicherheitsrichtlinien, sind Teil eines oder mehrerer Workflows und verfügen über ein anpassbares "Look and Feel". Das CMF stellt Standardtools zur Verwaltung und Bearbeitung von Inhalten zu Verfügung. Die Inhalte können entweder mit einem Webbrowser über das Web oder mit Desktoptools bearbeitet und verwaltet werden. CMF fördert *Distributed Authorship*. Das bedeutet, dass Besucher der Site, in Abhängigkeit von deren Zugangsberechtigungen, Inhalte erstellen und veröffentlichen können. Dabei werden alle Zugriffe auf Inhalte durch Sicherheitsmechanismen und Workflowregeln überwacht.

4.2 CMF Dienste

Die CMF Dienste bilden den zweiten wichtigen Bestandteil des *CMF* und stellen zentralisiert Managementfunktionalitäten zur Verfügung.

Hier nun ein Überblick über die wichtigsten *CMF Dienste*.⁶:

4.2.1 Membership Service

Der *Membership Service* kontrolliert, wie ein Benutzer Mitglied einer Site werden kann, verwaltet die Benutzerdaten und ist für die Authentifizierung zuständig.

4.2.2 Workflow Service

Der *Workflow Service* erlaubt es, Regeln zu definieren, nach denen Inhalte entwickelt und veröffentlicht werden können. Zum Beispiel kann einen neuer Artikel von einem Autor geschrieben werden, aber nur vom Redakteur zur Veröffentlichung freigegeben werden.

⁵Ein Zope-Product ist ein Python-Paket, das Erweiterungen für den Zope-Server zur Verfügung stellt.

⁶Für einen Überblick über alle *CMF Dienste* siehe: [11]

4.2.3 Security Service

Das CMF bietet eine einheitliche Sicherheitsarchitektur, die alle Funktionen der Site kontrolliert. Damit können für verschiedene Inhaltsobjekte unterschiedliche Zugriffsrechte vergeben werden. Der *Security Service* sorgt dafür, dass nur der Benutzer ein Inhaltsobjekt verändern kann, der auch die dafür nötigen Zugriffsrechte besitzt.

4.2.4 Look and Feel Service

CMF verfügt über einen einheitlichen *Look and Feel Service*, der für ein konsistentes Look and Feel sorgt und die Site für verschiedene Benutzer personalisiert. Inhalte mit unterschiedlichem Look and Feel können wiederverwendet werden.

4.2.5 Search Service

Das CMF verfügt über einen mächtigen und schnellen *Search Service*. Damit können Inhalte durch Volltextsuche oder auf Basis von Metadaten⁷ gefunden werden. Mit dem *Search Service* können Berichte für den internen Gebrauch erstellt oder Suchfunktionen für die Besucher der Site zur Verfügung gestellt werden.

5 Fazit

Zope zeichnet sich durch seine durchgängige objektorientierte Arbeitsweise aus. Durch die Bereitstellung eines Sicherheitsframeworks auf unterster Ebene und verschiedener Tools zur Versionierung eignet sich Zope bestens für den Mehrbenutzerbetrieb. Durch die Verfügbarkeit von Datenbank-Konnektoren für alle gängigen Datenbank-Management-Systeme und der Unterstützung von LDAP ist für die Wiederverwendung bestehender Datenbestände gesorgt. Für die Erstellung von Webapplikationen ist Zope bestens geeignet.

Mit dem CMF lassen sich kleinere Portale ohne große Mühe erstellen. Die Möglichkeiten des CMF sind aber dennoch begrenzt. Für größere Websites sollte auf das WCMS-Framework *Plone*⁸, das auf Zope/CMF aufbaut, zurückgegriffen werden.

⁷Unter Metadaten ("Daten über Daten") versteht man strukturierte Daten, mit deren Hilfe eine Informationsressource beschrieben und dadurch besser auffindbar gemacht wird.[15]

⁸Für mehr Informationen zu Plone siehe: <http://www.plone.org>

Literatur

- [1] Fabian, T. (Hrsg.), Kuhn, S., Langham, M., et al.:
Portale und Webapplikationen mit Apache Frameworks;
Software & Support Verlag
- [2] **What is Zope?;**
<http://www.zope.org/WhatIsZope>
- [3] Brian Llodoy:
An Introduction to Zope;
<http://www.zope.org/Resources/ZopeIntro/>
- [4] **The Zope Book;**
<http://www.zope.org/Documentation/Books/ZopeBook/current/>
- [5] Guido van Rossum, Fred L. Drake, Jr:
Python Tutorial;
<http://www.python.org/doc/current/tut/tut.html>
- [6] Frank Tegtmeier:
Objekthierarchie - Open-Source-Webpublishing mit Zope;
iX 8/1999
- [7] Tom Riedl:
Objektorientierte Webentwicklung mit Zope;
Linux-Magazin 6/1999
- [8] Stefan Jablonkski, Christian Meiler:
Web-Content-Managementsysteme;
Informatik Spektrum Band 25 Heft 2 April 2002
- [9] Adam Field:
Zope Content Management Framework (Part 2 of 2);
http://www.cmswatch.com/Features/ProductWatch/FeaturedProduct/?feature_id=64
- [10] **The Content Management Framework (CMF);**
<http://cmf.zope.org/Members/beehive/ZWACKChap5.html>
- [11] Zope Corporation:
The Zope Content Management Framework Documentation;
<http://cmf.zope.org>
- [12] Network Working Group:
Lightweight Directory Access Protocol (LDAP) RFC 1777;
<http://www.ietf.org/rfc/rfc1777.txt>
- [13] Peter Wächtler:
LDAP - die neue Auskunft;
Linux-Magazin 9/1998
- [14] Arnd Poetzsch-Heffter:
Konzepte objektorientierter Programmierung;
Springer-Verlag, 2000
- [15] Niedersächsische Staats- und Universitätsbibliothek Göttingen:
Was sind Metadaten?;
<http://www2.sub.uni-goettingen.de/intrometa.html>