

ZOPE



Web Content Management mal ganz anders eingesetzt.



Überblick



- Was ist Zope
- Netzadministration – Theorie und Praxis
- Wunschliste des Netzadministrators
- Wie Zope dazu passt
- Besonderheiten von Zope
- Zusammenfassung

Was ist Zope?



From the Zope web site:

<http://www.zope.org/>

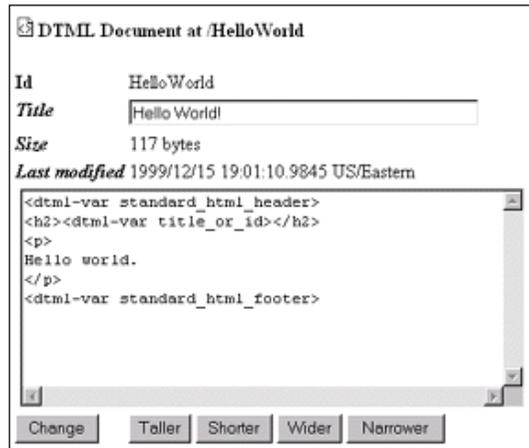


The Leading Open Source Application Server

Zope leads the Open Source application server market because it is the most flexible solution in existence for creating and maintaining large web presences.

You can create a maintainable infrastructure that will grow with your needs using Zope's standards-based tool set. Zope gives you the power to create a site that uniquely addresses your business needs. Better yet, it's free!

Was Zoep wirklich ist:



Permission	Roles		
	Anonymous	Manager	Owner
Acquire permission settings?			
<input checked="" type="checkbox"/> Access contents information	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Change permissions	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Delete objects	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Manage properties	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Manage users	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Undo changes	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> View	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> View management screens	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

- OO und script Entwicklungsumgebung
- Integrierte Objektdatenbank
- Flexible Zugangskontrolle
- Web als Nutzerinterface
- Eingebaute Server fuer Protokolle wie FTP und HTTP
- Schnittstellen zu externen System, z.B. LDAP, SQL u.a.
- Open Source
- Die mitgelieferten Klassen sind auf Webapplikationen spezialisiert.

Diskurs: Netzadministration I



Heutige Netzwerk Management Systeme:

- Geteilt in Netzwerk- und Element-Management
- Detailliertes Datenmodell
- Hauptanwendungen: Ueberblick, Status, Statistik
- Konfiguration ist nur unzureichend abgedeckt –
Netzelemente zum Teil, dazugehoerende Dienste
(z.B. RADIUS & DNS) garnicht.

Netzadministration II



Netzwerk Management in realen Systemen:

- oft mit script-basierten Systemen,
- die haeufig auf freien tools wie z.B. RCS, PERL, LDAP und ssh basieren.

Konfiguration:

- aendert sich mit dem Netzwerk,
- waechst haeufig unkontrolliert mit dem Netzwerk.

Die Kluft



Es bestehen gravierende Unterschiede zwischen

- der theoretischen Kunst des
„Netzwerk Managements“

und

- der realen Praxis der Netzkonfiguration.

Wunschliste I



- Zugangskontrolle auf ‚Objektebene‘ und ‚funktionsorientierte Rollen‘ (z.B. ‚kann neue Benutzer hinzufuegen, aber nicht frei in den Nutzerdaten schreiben).
- Kompromiss zwischen
 - Komplexitaet bei der Verwaltung von Rechten
 - Flexibilitaet
- Zugangskontrolle fuer ganze ‚Hierarchien‘ von Objekten

Wunschliste II



- Mehrere (beinahe) identische Konfigurationen sollen effizient verwaltet werden.
- Flexible Abgabe von Verantwortung (delegation).
- ‚Ausnahmen‘ muessen moeglich sein – auch nachtraeglich.

Wunschliste III



- Wer hat was, wann und warum geändert
- Mehrere Änderungen erst testen und dann als eine Transaktion übernehmen.
- Möglichkeit zur Rückkehr zu alten Konfigurationsständen.
- Kommentare und Notizen in die Daten einfügen.

Datenmodelle I



Detaillierte Modelle: SNMP MIBs, LDAP/DEN, PIBs

Alternative: Textbasierte Konfigurationsfiles, z.B. *mail*, *bind*, Cisco-routers

<i>Feature</i>	Text	ASN.1 Model
<i>Revision Control</i>	Lots of generic tools available (RCS, CVS, SCCS ...)	Needs to be supported by model (examples?) or by special tools (examples?)
<i>Embedding external data</i>	Some generic tools available (preprocessors), lots of specialized tools/scripts.	
<i>How to change</i>	Lots of generic tools – editors, sed, awk, perl	Some generic tools
<i>Add to or change model</i>	Usually no problem, only few specialized tools need to be adapted	Extension of model usually difficult – more tools need to be adapted
<i>Storing Comments</i>	No problem	Usually not possible.

Datenmodelle II



- Komplexitaet und Variabilitaet von Konfigurationsdaten scheint mit dem Netzwerk-layer anzusteigen.
- Zusammenfassung:
 - Kein klarer Sieger zwischen ‘durchstrukturierten Daten’ und einfachem Text..
 - Ein Loesung in der Mitte scheint sinnvoll
 - In jedem Fall muessen Anwendungen einfach adaptierbar sein
- Idee: **‘Model on demand’**

Nichts ist wirklich neu ...



- Ganz aehnliche Probleme gibt es auch in der Softwareentwicklung und bei Web-Anwendungen.

- Zope bietet fuer fast alles eine eingebaute Loesung
 - Versionskontrolle
 - Zugriffskontrolle
 - Scripts
 - Abgabe von Verantwortung
 - ...

Zope: ‚Acquisition‘



- Alle Zope Objekte in der Datenbank werden von einem uebergeordneten Objekt referenziert – aehnlich der Beziehung ‘Verzeichniss’ und ‘Datei’ bei Files.
- Wenn die Klasse des Objects ein bestimmtes Attribut nicht bietet, wird in den ‘Verzeichniss-objekten’ rekursiv gesucht.
- Dieser ‘Acquisition’ Mechanismus wird ueberall in Zope verwandt um templates sichtbar zu machen. Wenn ein Objekt nichts anderes spezifiziert, werden die Daten aus dem Objektbaum darueber uebernommen.

Versionsmanagement



- Alle alten Versionen eines Objekts bleiben in der Datenbank
- Auch vom Nutzer definierte Klassen erben diese Eigenschaft
- “Version” Objekte erlauben ‘Seitenaeste’ im Versionsbaum
- Objekte an denen gearbeitet wird sind gesperrt.

Zope: Zugangskontrolle



- Der Acquisition-mechanismus wirkt auch auf Daten zur Zugangskontrolle
- Differenzierte Rechte koennen vergeben werden – herunter bis auf ‘methoden’ Ebene
- Ein Satz von Rechten kann in ‘Roles’ gebuendelt werden, die wiederum Nutzern zugewiesen werden.
- Ausfuehrbare Objekte koennen eigene Privilegien haben (ueber die des aufrufenden Nutzers hinaus).
- Nutzer werden ebenfalls durch Objekte repraesentiert, die durch ‘acquisition’ fuer andere Objekte sichtbar werden. Ein Nutzerobjekt ist daher nur innerhalb seines Unterbaums gueltig!

Zope: 'Model on Demand'



- Integration von 'Text' und 'Datenmodellen'
 - Zope bietet alle ueblichen Funktionen zum 'Textprocessing'
 - Eigene Klassen koennen leicht definiert werden
- Migration ist einfach, da neu definierte Klassen leicht in existierende Textdaten eingebunden werden koennen.
- Tolerantes OO-modell von Python
- Schnittstelle zu externen Systemen – SQL, LDAP, PERL ...

Zusammenfassung



- Zope erfuehlt die meissten Anforderungen an ein System um Netzkonfigurationen zu verwalten – besser als viele ‚Netzwerk Management Systeme‘
- Noch offen: Verwaltung von Abhaengigkeiten, wie z.B. ‚make‘ es bietet.
- Einen Versuch ist es auf alle Faelle wert:
<http://www.zope.org/>
Selbst wenn Sie ‚nur‘ Webseiten erstellen wollen ☺