016zope.qxd 02.02.2001 17:42 Uhr Seite 16

The free Web application server Zope in use

VILLAGE GOSSIP ON THE NET

JOACHIM WERNER

A year ago a little-known niche product, Zope has now become a serious competitor for PHP, Perl, and even for Java, in the field of Web applications. But many outsiders are now wondering, not without some justification, what's it all about, this "weird hybrid", which is so hard to classify into one of the usual categories such as "Web server", "application server" or "scripting language".

If you ask true "Zopistas" what Zope really is, they sometimes just start to stutter, otherwise they launch into a very long lecture. The author is in fact himself infected by "Zopitis", but nevertheless we will attempt together on the following pages to understand the Zope phenomenon using a sample application. Whether you the reader allow yourself to catch the bug at the end or prefer to stay with Perl, PHP or Java, is for you to decide. But first, a little look back.

In the beginning was an aeroplane

The year is 1996. The scene is a return flight from a training course on CGI-based programming, taken over on behalf of a colleague at short notice. Jim Fulton, the chief programmer of the small American software firm Digital Creations, started to ruminate. What he had acquired in a few days of self study for the course on Web programming simply did not fit into his "world view." The link between CGI scripts and HTML pages is more than a little tenuous, and advanced functionalities are hard to convert. He was especially bothered by the fact that the path via CGI scripts is not "transaction-secure": When for example just a single step in a log-on procedure fails, it is very time-consuming to cancel the whole log-on. There must be another way: above all **object-oriented**, and naturally the whole thing should be written in Fulton's grassroots programming language **Python**.

That is how the idea was born. But it then took until autumn 1998 before Zope received its present name and - at the advice of the financial management of Digital Creations - was let loose on the world as open source software.

”Villages onto the Net” Initiative

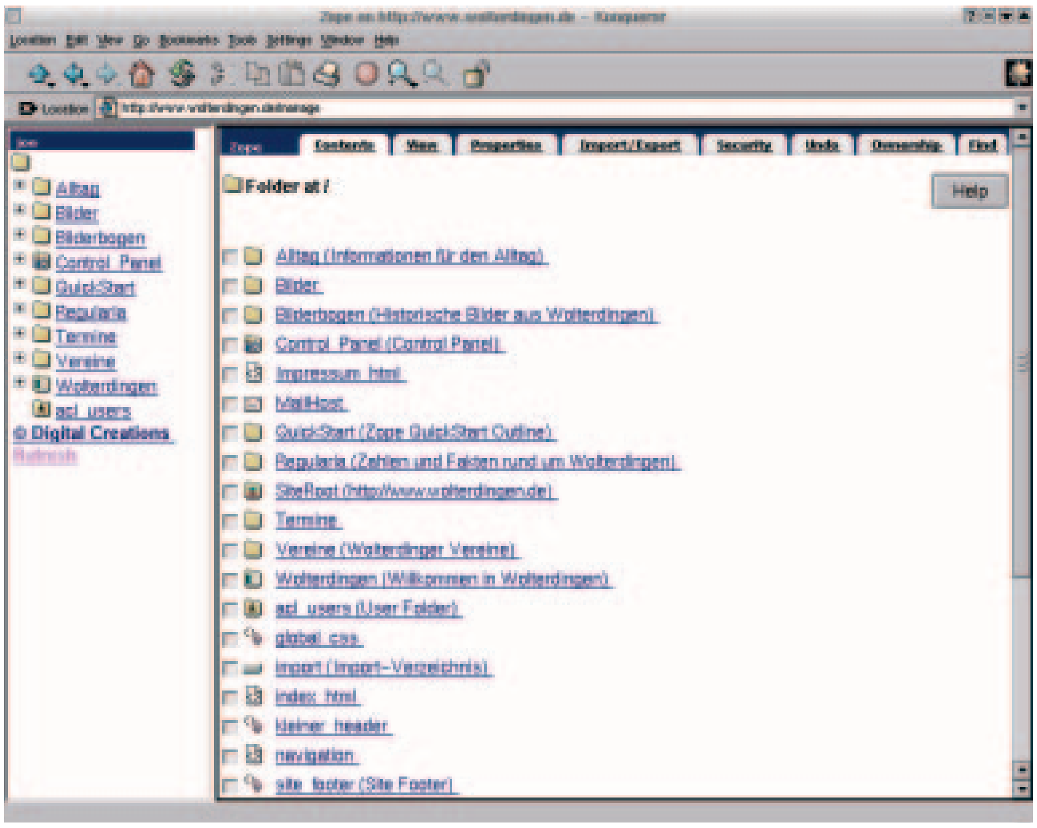
Let’s get back to the present, to a little village in Bavaria called Wolterdingen, which is not quite as out in the sticks as one might think, because the village council passed a resolution that the future could no longer be ignored and an Internet presence was necessary. Now they’ve all got one. And as the people of Wolterdingen wanted to get a bit ahead of the competition from neighbouring villages, it had to be a ”dynamic” Web presence. All the societies and clubs in the village should be able to publish articles themselves or announce events on the Web.

The villagers asked enquired whether something like that could be done. What they learned at their next village meeting made their ears ring: a couple of confirmed Linux hackers in the village swore by a self-built solution in Perl, naturally under Linux and with the Apache Web server. Also, an SQL database as backend was needed, either MySQL or PostgreSQL, which is something the two geeks were not totally agreed on. Then there was the IT consultant, who works during the week at a large software house in Munich. He swears totally by Java. Plus a reasonable application server with ”EJB”, ”RMI” and naturally ”JDBC”. Without a database, of course, here again nothing would happen, but it should be a ”proper” one, from Oracle, or at least Sybase. The local PC dealer would have just liked to point out that it would also be possible to create really great Web sites with Windows and Active Server Pages too, but he didn’t get a word in edgeways.

In the furthest corner of the hall however, a cluster had formed around a young woman who had brought her notebook with her. On the screen could be seen a Web site which came close to the proposals of the villagers. ”How did you get that so fast?”, asked the village council leader incredulously. When he heard the reply, his jaw dropped: ”I downloaded this Zope from the Internet yesterday evening. There is a module called Squishdot. I just had to install it and adapt the colours and the logo a bit. And the Web server is already there!”

Zope: The all-in-one solution

Let us leave the Wolterdingers and take a look at the ”Feature List” of Zope (details can be found in the box ”Zope at a glance”): Zope comes, unlike most other Web-tools, as an ”all-in-one” solution. The binary distribution even comes with an adapted version of the programming language Python, on which Zope is largely based (a few performance-critical parts are coded in C). The so-called *ZODB*, the Zope Object Database, serves as database. As the name says, this is a truly object-oriented database, with which objects and their status variables can be easily serialised and stored. For SQL fans a Python-based test database is included, which although not suited to larger



Folder at /

Altag (Informationen für den Alltag)
Bilder
Bilderbogen (Historische Bilder aus Wolterdingen)
Control_Panel (Control Panel)
Impressum_html
MailHost
QuickStart (Zope QuickStart Outline)
Regularia (Zahlen und Fakten rund um Wolterdingen)
SiteRoot (http://www.wolterdingen.de)
Termine
Vereine (Wolterdinger Vereine)
Wolterdingen (Willkommen in Wolterdingen)
acl_users (User Folder)
global_css
import (Import-Verzeichnis)
index_html
kleiner_header
navigation
site_footer (Site Footer)

The Management-Interface: All about Wolterdingen on the Net at a glance

projects, serves well when it comes to experimenting and is fully integrated in Zope.

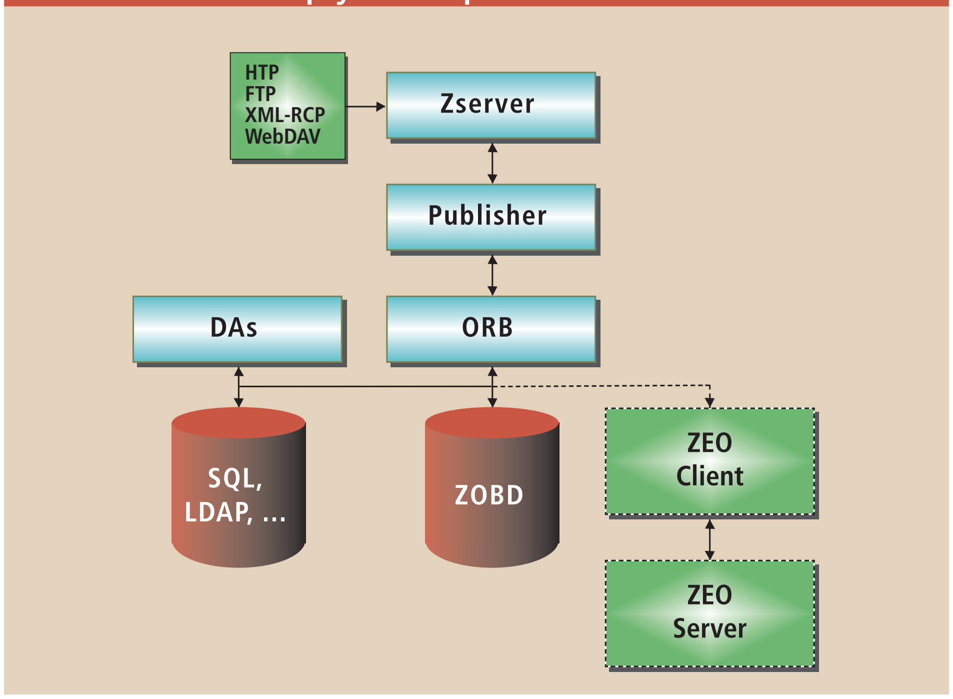
As already mentioned, the Web server is included in the form of *Zserver*, an expanded version of the python server *Medusa*. As well as the Web protocol HTTP (including the *PUT* command to upload web pages) the ZServer also understands FTP even the Web DAV protocol which is already supported as standard under Windows in the form of the so called **webfolder**, though scarcely known in Linux. This makes it childsplay to load an existing web page, or a folder with graphics, into the Zope Object database. The best is yet to come, though: Zope has a simple and fast Web front-end, by means of which almost all functions of the platform can be administered.

Zope Architecture

To install under Linux simply download the binary distribution from the Zope server www.zope.org and execute the *install* script in the folder which is created after unpacking. Then Zope is started with the *start* script and then, if nothing has been altered in the basic settings, can be contacted at http://localhost:8080. Naturally the start can also later be automated via a script. Apart from the binary distribution one can also select the source distribution, which has the parts of Zope programmed in C in source code. Source installation is also governed by a script and if Python and GNU C-Compilers have been correctly pre-installed there is no problem running under Linux. The source distribution is platform-independent and should be able to run on all commercial Unix systems (including MacOS X) and of course the various BSDs. Naturally one of the simplest ways to obtain Zope is in the common Linux distributions. But the versions which come with SuSE or RedHat Linux, for example, are sometimes look a bit past their best and thus cannot be unreservedly recommended.

For the sake of completeness the Windows binaries are worth a mention, which could also be of interest for die-hard Linux users, because many people who have been using Linux on their server for a long time may still be running Windows on their notebook. Since the Zope applications are completely platform-independent, there is nothing wrong with running a small test server on the Windows computer and then simply exporting the results to the Linux server.

Simplyfied Zope-Architecture



One platform, many ways

Back to Wolterdingen, the village with the big plans for the Web: After the Wolterdingers have settled on Zope, a few more decisions have to be made. Because for Zope the same is true as is often said of Perl: There is more than one way to get things done.

In the first place, Zope can be deployed "out-of-the-box" as a pure content management system. Via the Web interface, easily called up by attaching /manage onto the appropriate URL for the website, new managers and users, sub-domains ("Folders") and documents can be made and images and HTML pages be set up. If errors occur at this time, it's not a big problem. Firstly, one can make changes in a so-called *version*. These can then be seen by those who are also logged onto the version. The visitor to the Web site, however, does not see them until the version is released. Secondly, all actions can be reversed until the administrator "packs" the database. This is when, depending on the setting, any objects no longer used, whose modification was more than ten days ago, are deleted. Some objects even offer a *History* function, with which one still has access to all old revisions of the document or the method. Then, for example, one can make last week's version into the latest one again, because since then a bug had slipped in.

But many users, particularly those without a programming background, will miss some comfort-features of professional content management systems, such as more complex release workflows or an inbuilt *WYSIWYG* editor.
Web pages must either be processed in what are usually somewhat small "text area" boxes in the browser or created locally with an editor and then uploaded via FTP or WebDAV. Under Windows many people swear by Web editors like GoLive.

The next step is to provide the pages with a consistent *standard_html_header* and *standard_html_footer*. This makes it very easy to achieve a consistent design for all the pages. The navigation menu for the Web site is usually also packed into the header and contains the necessary code to generate on each page perhaps a menu or a *directory* line (*Home > Clubs > Rabbit breeders*).

Headers and footers, together with other dynamic page elements, are integrated as tags according to the table "<dtml-var Name>" in the pages. This scripting notation, which is in principle similar to PHP, is called DTML (Dynamic Template Mark-up Language). Originally this was only intended to include pre-produced modules written in Python in the sites and to realise simple logic with if-elements and loops. In the meantime, though, DTML has become an alm ost complete programming language with, admittedly, often highly individual syntax.

Acquisition: If you haven't got one, get one!

One particularly useful, but also at first confusing, characteristic of Zope is the so-called "acquisition" of methods and objects, which needs a bit more explanation: Put simply, all elements of a Zope site are in fact objects. A folder for example is an object container, which can contain other objects. A DTML method may look like a simple HTML document, containing DTML tags, but it behaves towards the folder in which it is located, like a method which can be applied to it. A simple example would be that of a folder with the title "Clubs" and the ID *clubs_folder* on our Wolterdingen Web site, which contains a method *view*:

```
<dtml-var standard html header>
<h1>
Welcome to the
<dtml-var title or id> site
        </h1>
        <dtml-var standard_html_footer>
```

If one now goes with the browser to http://www.wolterdingen.de/vereins_folder/view , the header and footer ensure that the standard layout of the Wolterdingen site is used and the following text can be read:

Welcome to the Club Page!

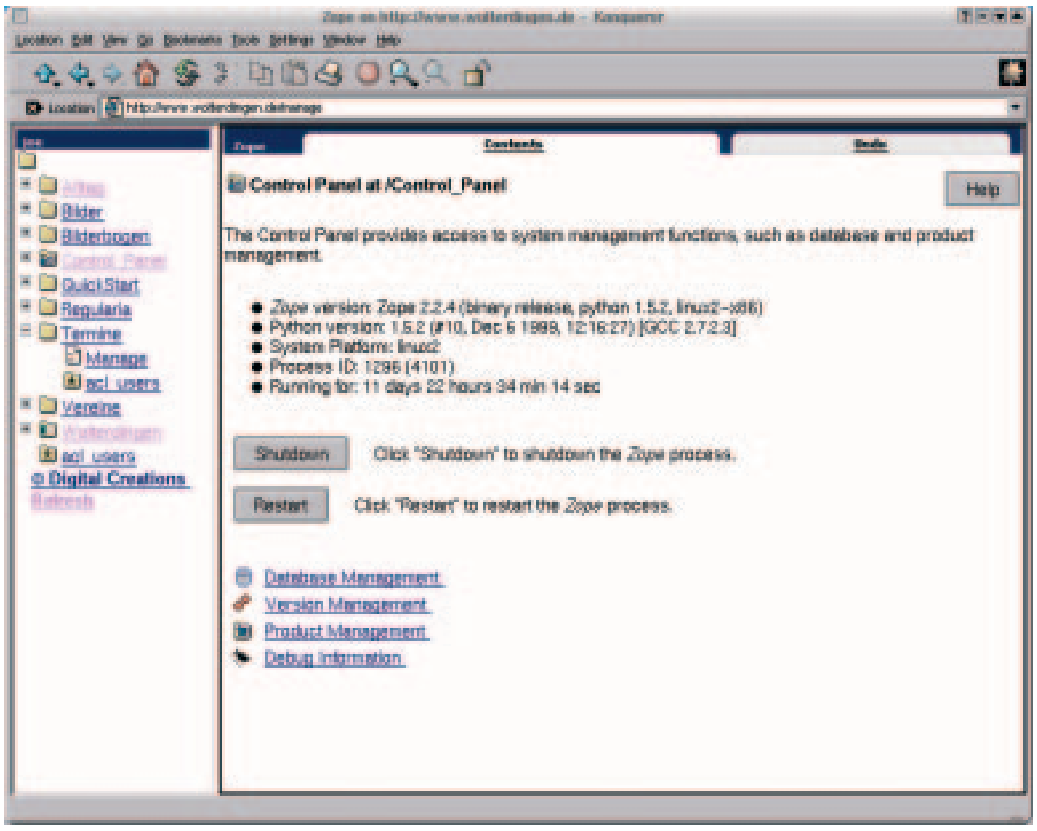
If the "Clubs" folder had not defined a title, *<dtml-var title_or_id>* would automatically use the folder-ID (the "file name" of the folder, as it were), so this would be *clubs_folder*.

Similarly to other web servers, incidentally, Zope searches, automatically for a method "index_html" when http://www.wolterdingen.de/clubs_folder/ is entered,and displays this (in Zope-Speak one would say that Zope "renders" the *index_html* method. The fact that *index.html* in Zope is called *index_html*, is incedentally due to the fact that the dot in Python (and most other object-oriented languages) is used as a separation mark between object and method. But now Zope can also easily render files with a dot, which is important for example in the case of images ("logo.gif"), because many browsers would not otherwise co-operate.

Now let's place e.g. the rabbit breeders' club and the gardening club under the clubs folder. Now, at last, acquisition comes into play: http://www.wolterdingen.de/clubs_folder/gardening/view now displays

Welcome to the Gardening club page!

without the *view* method having been copied into the corresponding folder! Zope simply searches in the next highest folder for a corresponding method and uses this. This means you can do fantastic stuff: The Gardening Club can now use the standard



Control Panel at /Control_Panel

The Control Panel provides access to system management functions, such as database and product management.

- Zope version: Zope 2.2.4 (binary release, python 1.5.2, linux2-x86)
- Python version: 1.5.2 (#10, Dec 6 1999, 12:16:27) [GCC 2.7.2.3]
- System Platform: linux2
- Process ID: 1296 (4101)
- Running for: 11 days 22 hours 34 min 14 sec

Click "Shutdown" to shutdown the Zope process.

Click "Restart" to restart the Zope process.

Database Management
Version Management
Product Management
Debug Information

The Control-Panel: Zope can be administered completely via the Web

016zope.qxd 02.02.2001 17:42 Uhr Seite 20

header and -footer. At some point, the committee decides that instead of the standard layout for Wolterdingen, it would prefer to have a couple of green ivy twines along the side. Nothing could be easier: Just copy the *standard_html_header* using *copy & paste* onto the Web interface in the Club folder and make the corresponding changes there. Then all you need do is upload the necessary graphics and the whole thing is done. At the next call up the *view* method will now use the header of the Gardening Club, as soon as it has been called up from its folder.

The administrator of the Wolterdingen site does not, of course, want every club Web master making changes willy-nilly on the site. So she can make a so-called *User Folders*, defining users for the folder, in which they are located, and all sub-folders and objects stored therein. This procedure allows roles to be defined for any kind of user, e.g. *Club member*, *Committee* or *Guest*. These roles can then be very finely-tuned to be linked with individual rights. So a committee can be given the right to add network users and delete them for their club. The User Folders can by the way also be replaced by

Zope service providers:

beehive elektronische Medien GmbH: http://www.beehive.de iuveno - Smart Communication AG: http://www.iuveno.de Lightwerk - Premium Internet Solutions: http://www.lightwerk.de

A selection of Zope products and modules:	
Database adapters (DAs) and Z Object Database Implementations	
Zope Interbase Storage	ZODB-Implementation for Interbase from Borland
Zope MySQL Database Adapter	MySQL DA
DBMaker Database Adapter	DBMaker DA
Ultraseek DA	DA for the search engine Ultraseek
Zope ODBC Database Adapter	ODBC-DA for Windows
Z Solid Database Adapter	DA for the commercial DB from Solid
ZPoPyDA+A10	latest PostgreSQL-DA
ZRadius	DA for Radius protocol
Zope Sybase DA	Sybase DA
Berkeley Storage	ZODB-Implementation for the Berkeley DB
GV Interbase Database Adapter	Interbase DA
ZopeLDAP	Adapter for querying LDAP from Zope
SAPDB-DA	SAP-DB Database support (still in development)
User Folder Implementations	
Generic User Folder	general implementation, which can be used with various DAs
NTUserFolder	User Folder for NT Domains
smb User Folder	Imports SMB/Samba-User into Zope
MySQL user folder	User Folder for MySQL
LDAP User Folder	Authenticates Zope-User against an LDAP directory
Zope Applications	
Squishdot	Web Log a la Slashdot
Portal Toolkit	Trial toolkit for community portals
Zwiki Web	Zope version of the web community tool Wiki Wiki Web
ZopeGUM (Zope Grand	Groupware tool a la MS Outlook Unified Messenger)
Other	
ZpdfDocument	generates PDF documents from Zope
DTML-Tex Product	generates Tex documents from Zope
Etailer	E-Commerce Shop
Zope Cascading StyleSheets	Style Sheets easily processed via forms
Zope Internet Explorer Editor Demo	HTML WYSIWYG-Editor; sadly, only for MS Explorer
zCommerce	E-Commerce Shop
ZDP-Tools	
Active Images	
GDChart Product	
Emarket	E-Commerce Shop
Local File System	Access to the file system via Zope (e.g. for uploads/ downloads over the Web)
Site Access 2	enables virtual hosting with Zope
ZSwGenerator	generates flash code from Zope
PHP Object	generates PHP code from Zope

plug-ins and users imported automatically from an LDAP directory or from a Samba server.

Applications out of the construction set

Apart from the aforementioned plug-ins, the Zope website contains a multiplicity of more or less useful expansion products for Zope. A large group of these are the database adapters, about which more later. Also, there is a large number of smaller modules for creating charts, advertising banners, navigation menus and other elements. But what interests the people of Wolterdingen is the ready-made "Zope Applications", of which **Squishdot** is only the best known. With Squishdot it is possible to make so-called "Web Logs", a la Slashdot, without the trouble of programming. The Web master only has to complete a few configuration forms and upload the necessary graphic elements, and there it is: an individualised Web Log with complete functionality. In the next few months many more interesting Zope applications are planned: Groupware tools following the example of Microsoft Outlook, toolkits to construct portals for the Internet and/or for company Intranets or a "proper" content management system, which can be used without HTML- or DTML-know-how and more besides.

ZClasses - A (Z-)class for all seasons

If we take a closer look at the Wolterdingen Club pages, one wish comes to the fore: Wouldn't it be nice if there was a simple model, a "Template", with which the Web master could simply add a new club with all the necessary elements, such as their own calendar for events or a member directory? No problem: For this Zope has *ZClasses* - Zope expansions, programmable via the web-front-end, which can then be simply selected from a menu like the products which can be downloaded from the Zope website. In most cases a corresponding form then has to be completed, for example to specify the administrator for the new club or to make the first layout adjustments.

Anyone who is now wondering how to find his data again in these ZClasses without SQL, can find the answer in the *ZCatalog*, a search engine integrated into in Zope. It is fairly powerful and fast, with automatic index-generation, but which, at least in the older Zope versions, still had a few little bugs such as failing to correctly interpret German umlauts.

Perl and Python in close harmony ...

Sometimes - often fairly quickly - one comes to a point with Zope's DTML scripts, when there is a keen desire for a "proper" programming language. Zope now has several options for this: Firstly, you can call up "External Methods" written in Python from Zope. These must simply be placed in the *Extensions* folder of the Zope installation and activated in the Web interface. On the other hand it has now become possible to make small *Python Scripts* via the Web-front-end, but their functionality is somewhat limited on the grounds of security. Both options will - perhaps even by the time this article comes out - also be realised for Perl. This means that in future Zope will capture the almost inexhaustible resources of the existing Perl code, such as the excellent Wwwlib, even if some Python fans will certainly turn aside with a shudder.

With external methods and scripts almost any additional functionality required can be integrated into Zope, because behind a small method there can also be a "Wrapper" for an entire program library (e.g. the PIL Library for image processing or the powerful XML-Tools from Python). But the royal road to Zope programming is found in the so-called "Zope products" - new Zope modules, written completely in Python. We have already looked at these from the point of view of the user: Squishdot is a Zope product. Because there are now very good instructions and aids such as complete code templates, programming Zope products is simpler than you think. Anyone who already has the corresponding know-how in Python - and that's not so hard to acquire - will often achieve an objective

A look into the Zope crystal ball

*It is pretty certain that the future for Zope will soon have no support for Java. In return Zope will soon be fully compatible with the new SOAP protocol, which although being promoted among others by Microsoft, is nevertheless an open and highly promising standard for distributed applications. Also, work is proceeding at fever pitch on improving the existing XML support. The first two fruits of this work are **HiperDOM** - a new XML-based template concept, which is comparable to the XMLC approach from Enhydra - and **Parsed XML**, the new XML rendering engine for Zope. Internally, there will at first be only some small changes. Porting onto **Python 2.0** and the ability of Zope to handle Unicode character sets, is as good as finished. This Unicode support is an important building block for Zope to have full multilingual capacity. At present there is some effort, mainly in Europe, to translate the web interface of Zope. A German version is already available. In the next step the project will also be transferred to application level to simplify the work on multilingual websites. Another interesting product is the **ZPatterns Framework**. Unfortunately ZPatterns have previously only been properly understood by an illustrious circle of Zopistas. The underlying concepts for reusability of functionalities and clean abstraction between data and views of this data are, however, ingenious. The majority of activity for the next few months is expected to be in Zope-based applications. For example work is proceeding at a feverish pace on a **Groupware system**, a **Content Management Framework** and a **Portal Toolkit**. Last but not least there are also various projects underway which have the aim of a sort of IDE (Integrated Development Environment) for Zope. The most ambitious of these was **ZopeStudio**, a development platform based on the new features of the Mozilla browser. Unfortunately, because of diverse technical problems, the project has been put on ice for the time being at the Mozilla end.*

The author

Joachim Werner is the father of a two-year-old son, who can unfortunately already say "mobile", but not yet "Zope". As founder and CEO of iuveno Smart Communication AG he sadly all too rarely gets round to looking at picture books with his son. His largest project at present is the total conversion of a university to a Zope-based content management system.

faster than with ZClasses and DTML. Python simply has clearer and shorter syntax and is easier to read. But this does mean that programming in teams over the web can only be realised to a limited extent. Zope versions or Undo are of course not available. Naturally, for this one can simply get help from tried and tested programmer tools like CVS.

Hello World! - integrating Zope

One point which we have only touched on in passing so far is the integration of Zope into existing system landscapes. The reason for this is simple. Because Zope comes with everything needed for the average web application, there is no need at the beginning to worry much about SQL databases or alternative Web servers. But in most cases it's not all plain sailing to start off. Often existing SQL databanks need to be brought onto the web. But Zope is also well-equipped to do this. For many commercial and most common Open-Source databases there are so-called database adapters (DAs), though their quality does vary.

The DAs maintained by Digital Creations itself on behalf of Oracle and Sybase are especially stable. But the PostgreSQL and MySQL drivers leave very little to be desired. MySQL though, has only recently started to provide substantial support for roll-backs of half-finished transactions, and the corresponding DAs were not yet ready when this article was written. This means that when using MySQL you lose part of the terrific transaction functionalities of Zope. Apart from SQL there are other database interfaces supported by Zope one way or another, such as LDAP, the Lightweight Directory Access Protocol, or the free Berkeley DB. It is also true that practically all data sources for which there is a Python interface, can be integrated very easily via a small wrapper into Zope.

CORBA, the Common Request Broker Architecture, used e.g. in the GNOME Project, is not yet supported by Zope, but there have been plans to do so for years now. Zope has yet another "Goody" ready for this, which could in future be even more important than CORBA: Zope supports XML-RPC (XML Remote Procedure Calls). XML-RPC is a protocol which is actually very similar to the HTTP. It is just expanded by a few extra commands and sends in the "Body" of the message, instead of a query for a Web site or the corresponding reply from the Web server, method calls and their results. So one could for example very easily pass on a database query to a Zope server from another Zope server or any XML-RPC-Client at all. With XML-RPC Zope can thus practically be completely "remote controlled"! A spicy remark in passing: The new SOAP protocol, which Microsoft introduced in the framework of its ".NET" initiative - and which by the way is open and free to use, is based on XML-RPC. SOAP can also be integrated into Zope very easily. Which means that Zope servers could very soon play an interface role between Linux and the Microsoft world.

Apache? That works, too

Even if Zope comes with its own webserver, it can sometimes be useful to rely on the existing server infrastructure. Zope can also be operated in principle in two ways with most of the common Web servers: Either as CGI script with the modifications *Fast-CGI* and *Persistent CGI* or via a "Proxy" configuration, where the calls made to a specific address are simply passed on by the Web server to Zope's Zserver. The box "Zope and Apache" goes into more detail about how this proxy mode is used with the Apache webserver.

Zope Enterprise Objects: High availability for the poor ...

One last time, back to that village in Bavaria: Say the villagers of Wolterdingen were to stumble across a hot spring in their village. If "Wolterdingen Spa" was then driven crazy by enquiries from visitors, a Zope server which they operate might soon become too small. But there is a remedy for this, too: With the "Zope Enterprise Objects" (ZEO) distributed clusters of Zope servers can be constructed. And www.zope.org is one such cluster. ZEO now allows the operation of, in theory, as many Zope servers as required on a central Zope database server. The Zope servers also come with their own cache memory and can intercept any brief interruptions of contact to the central server. Zope-father Jim Fulton is already working on designing the database server to be redundant. Then Zope could finally become a player in the super-league of scaleable "High-Availability" solutions.

And where are the Java Beans?

To become a "proper" application server, Zope obviously now only lacks support for Java Beans, or Java anyway. But this is not going to happen so fast, since the people at Digital Creations are certain that anyone having worked with Python will now only work with Java for web solutions at gunpoint. This is quite simply because since Python scripts and modules in Zope can be tested immediately after saving without compilation so it is far more productive than with Java.

To finish, there is still the answer to the question which you have probably been asking all along: "Does the little village of Wolterdingen actually exist?" - Yes, it does. And the good people of Wolterdingen have also decided on a Zope-based Web presence, which has turned out very nicely and with which they are very happy, so far. Whether the village meeting did actually proceed exactly as described by the author or if he dreamed a bit of it up, shall remain a secret. ■

Info

[1] The Zope-Homepage (Downloads, Manuals): http://www.zope.org [2] "Zope Newbies" (daily news about Zope): http://weblogs.userland.com/zopeNewbies/ [3] EuroZope (the European Zope Community): http://www.eurozope.org [4] Zope Documentation Project: http://zdp.zope.org [5] Zope book from Digital Creations (published by O'Reilly); Download at: http://www.zope.org/Members/michel/ZB/ [6] beehive elektronische medien GmbH (pub.):Zope; The Open Source Web Application Server, dpunkt.verlag, planned to appear in April 2001, ISBN 3-932588-93-2

■