

基于 ZOPE 的内容管理系统

The Content Management System Based on ZOPE

郭力子 (江苏信息学院 214023)

摘要:简述了内容管理系统的特点。介绍了开放源码 Web 服务器 ZOPE 的架构和特点。给出了应用 ZOPE 的 CMF 框架开发的内容管理系统实例以及在开发中需要注意的共性问题。结论是使用 ZOPE 构建中小企业的内容管理系统是值得考虑的解决方案之一。

关键词:内容管理 ZOPE 开放源代码

1 内容管理系统

内容管理系统(Content Management System),简称 CMS,目前广泛应用于出版、医疗、保险、金融以及各种信息密集型的企事业单位的网站,已成为当前电子商务中的新热点。

CMS 的功效和价值主要表现在两方面:对外,CMS 可以帮助企业建立统一的对外协作平台,加强与业务合作伙伴、客户的沟通和联系,为客户提供更加个性化的服务。对内,CMS 可以帮助企业打破信息孤岛,理顺业务流程,加强团队之间的协作,使企业拥有的“核心知识”得到充分的利用。

CMS 本身就由“内容”和“管理”两部分组成,所谓“内容”,就是具有价值的能够容易(数字化)、安全(权限控制)、快速(元数据管理)地进行共享和利用的信息资产。它不是某种单独的创新技术,而是许多先进技术的综合应用,它涵盖企业内联网(Intranet)、因特网(Internet)和企业外联网(Extranet)应用,大大突破了传统信息流管理软件、办公自动化软件以及文档管理软件的应用范围、使用效果和商业价值。

具体而言,CMS 应该满足的技术要求如下:

(1) 功能方面。包括数据检索、数据存储、格式转换、多语种支持、安全管理、页面创建、个性化定制、同步和复制等。

(2) 服务方面。包括流媒体服务、多媒体内容管理、电子数据交换、元数据和标引等。

(3) 内容集成方面。包括 workflow 管理、数字版权保护、数据挖掘等。

在当今信息管理成为企业的一个主要核心竞争力之际,CMS 的巨大市场需求已成为新一轮软件市场竞争的热点。目前市场已有的主导产品有 IBM 公司的 IBM DB2 Content Manager、Interwoven 公司的内容管理平台 Interwoven 6、微软公司的 Content Management Server 等。

应用 CMS 可以提升企业的信息管理水平,提高企业的核心竞争力。然而,对于一般的中小企业,面对的窘境就是难于采用昂贵的商品化 CMS。一个可替代的解决方案就是考虑使用现成的开源框架(open source framework),Zope 就是其中一种优秀的开源框架。

2 关于 Zope

Zope 是一款于 1998 年发布的面向对象的开放源代码 Web 服务器,自身提供了一个非常友好和便利的 CMS 开发框架。作为一种框架,它涵盖了很多 Web 应用程序开发者需要处理的底层细节,比如数据的持续性,数据的完整性,数据访问控制等等,这样就可以让你集中精力在解决问题上。比起其它的语言或框架,Zope 可以让你充分利用 Zope 提供的服务来更快速的构建 web 应用程序。

2.1 Zope 的结构与层次

作为一个面向对象的 WEB 服务器,Zope 的架构可分为四个层次:

最底层:对象数据库。Zope 的对象数据库 ZODB 实现了一个面向对象的文件数据库。下层对象能够继承上层对象的方法和属性。实现了用户和权限功能。

实现了数据库操作的 API。ZODB 就是一组模块来实现数据库功能,来供 Python 程序调用。ZODB 也实现了版本管理功能。

能实现。

到目前为止,Zope.org 站点中已经有了大大超过 500 种的可用于应用程序的 Product。这些产品包括

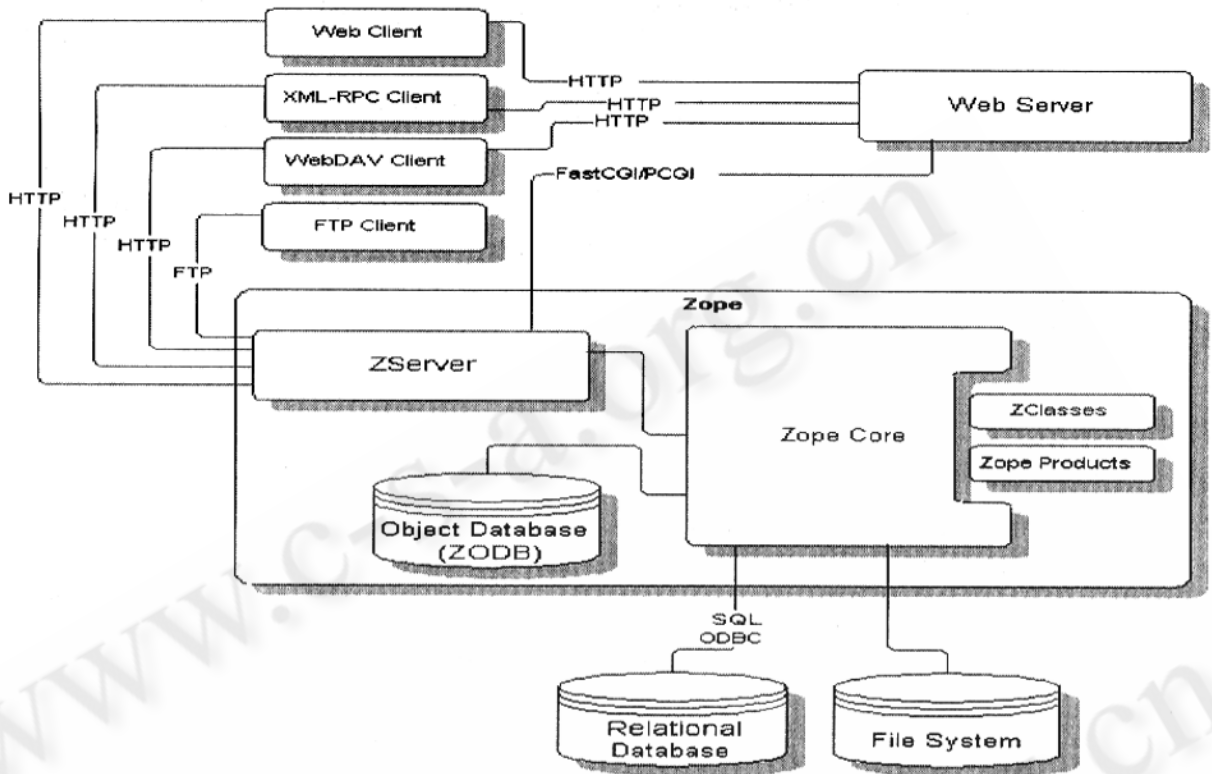


图 1 Zope 架构图

第二层:服务器。Zope 实现了一个 Web 服务器, Ftp 服务器, Webdev 服务器。实现了 ZMI (Zope 管理界面), 一个 Web 形式的对 ZODB 的操作。并提供了 Ftp 和 Webdev 方式的数据存取。Zope 还可以和现有的 Web server 结合在一起, 比如 Apache 或者 Microsoft IIS, 以及其它支持通用网关接口 (CGI) 的 Web 服务器。

第三层:开发。Product (产品) 是 Zope 的一种开发方式。每个 Product 都有自己特有的文件类型和相应的编程 API。用表现、程序、和数据、权限、管理界面、帮助等组成一种具有完整功能的特定类型的网站, 如 weblog、forum 等。也有些 Product 只是为了完善或增加 Zope 的某一方面的功能。如 xml、user folder、formulator 等。可以说 Product 是在 Zope 的平台上开发的。也可以说 Product 是 Zope 的一个组成部分, 因为 Zope 的编程的底层接口都对 Product 开放, Product 也要实现和 Zope 类似的 API、ZMI 的功能, Product 也要和 ZODB 直接打交道。Zope 能实现的功能, Product 都

Weblog, 比如添加论坛、数据处理、XML 工具, 以及电子商务程序等等。

第四层:除 Product 外的几种网站开发方式。这几种开放方式都是 Zope 的一种 Product。DTML (动态文本模型语言)。DTML 是 Zope 最早实现的表层的网站开发方式, 采用标签的方式来实现在文本中插入动态内容。DTML 有自己的变量赋值、流程控制等程序语言功能的实现。更复杂的功能采用在双引号中调用 Python 的方式来实现。DTML 能够直接存取 Zope 命名空间中的各个对象和变量 (在最新的 Zope3 中已取消了对 DTML 的支持)。

Python script (Python 脚本)。Python script 的出现是为了实现将 DTML 中的复杂逻辑部分剥离。Python 通过绑定 context 等环境变量和 Zope 的命名空间交流。Python script 有三种方式调用 Zope 功能: 通过调用 Zope 的库的方式; 使用 Zope 的 API; 将传进来的参数转为调用 ZMI 页面的参数, 来实现和 ZMI 操作相

同的功能。

TAL(模板属性语言)。TAL 是为了适应在 html 和 xml 内部实现动态内容的需求而出现的。它把动态语言以标签的属性的方式书写。这样,当使用各种 html 和 xml 编辑工具的时候,不会出现不应有的信息。它也能调用 Python 语言。

2.2 Zope 的特点

Zope 采用一种比 GPL 更灵活的 ZPL 协议发布,所有人都可以基于源码开发自用或销售,并且没有任何义务,Zope 可以在开放源代码许可证条件下自由分发,不同于那些昂贵的商业应用服务器。

Zope 是一套完整的平台。它包含了开发应用程序所需的全部组件。它的内容管理器配合 Zope 面向对象的设计,将内容的管理和权限管理系统合成一体。Zope 提供精细的、可扩展的权限框架。它的内容管理权限设置粒度可以细至每一个对象或文件。有一个非常容易使用的界面来增加、删除对象及修改对象的权限。

Zope 允许并鼓励第三方开发者打包和分发应用程序。因此,Zope 已经有了很多可以立即使用的产品组件(Product)。大多数组件都是自由并开放源代码的。Zope 拥有一大批社区开发者。

Zope 创建的应用程序可以直接通过 Zope 企业对象(ZEO)进行扩展。通过 ZEO,可以在多台计算机中部署 Zope 应用程序,而不需要修改代码。

Zope 允许开发者只使用浏览器就可以创建 Web 应用程序。比如:Internet Explorer, Mozilla, Netscape 等浏览器都可以支持 Zope 的管理界面(ZMI)。Zope 还可以通过使用统一的 web 界面让其他的开发者安全的同时进行开发。其它应用服务器很少支持这个特性。

Zope 可以让开发团队高效协同开发。协同环境可以让用户不会相互干扰,Zope 使用 Undo, Versions, History, 以及其它工具来帮助人们一起工作,并且可以从错误中恢复过来。而其它大多数应用服务器不支持这些特性。

Zope 可以运行在大多数计算机操作系统平台中: Linux, Windows NT/2000/XP, Solaris, 和 Mac OS X。Zope 甚至可以运行在 Windows 98/ME 中。而其它大多数应用服务器做不到这一点。

Zope 是用 Python 语言开发的。可以通过 Python 语言进行平滑地扩展。Python 是一种完全面向对象的高效易学的语言,可以促进快速开发。Python 中的许多功能库可以直接用于创建你的应用程序。而其它一些应用服务器却很难做到这一点。

3 使用 Zope 的 CFM 实现内容管理系统

针对用户的特殊需求,Zope 还提供了二级开发平台——现成的内容管理框架 CFM (Content Management Frame)。可以使用 CFM 来建立、管理网站,开发新功能。它是一个用来建立定制 CMS 的开发框架。CFM 位于 Zope 核心应用程序服务器系统(Zope Core application server system)之上。它提供了许多功能强大的工具和服务,如基于群的内容管理、基于许可的工作流系统以及用户化框架。CFM 对 zope 的用户管理、文档管理及版本控制、Product 进行了大量的扩展来实现更加高端的功能。

另外,CFM 也是对 Product 开发方式的改进。在 Product 中,表现、逻辑和数据被一起混合在 Product 包中,能够定制的部分只有 ZMI 中有限的几个选项,如果要定制更多就只能深入 Product 内部了。CFM 用皮肤(Skin,这里的皮肤不是只能改变字体和颜色,而是能改变版式等很多部分)、服务、文件类型的方式实现了表现、内容、数据的分离。每一个部分都可以定制,并且每一部分的改变不影响其它部分。

当前基于 Zope 的最有影响的内容管理系统 Plone 就是在 CFM 平台上开发的。

作为一个框架,CFM 可以让用户自主开发 CMS。但在开发中,笔者发现有些共性的问题需要引起重视:

3.1 对 ZODB 存储方式的扩展

Zope 是目前少数有影响的使用对象数据库作为后台支撑的 Web 服务器。我们知道用来管理结构化数据的关系数据库从理论到技术上经历了数年发展,已经相当成熟,而非结构化数据的复杂程度远远高于结构化数据。在当今的信息社会中,企业和个人都有着各种各样的“非结构化数据”,大量的 Word、Excel、Powerpoint 文档以及图象、电子办公文件、视频和音频等多媒体数据,还有每天都收到的大量邮件。面对这些“非结构化数据”,常规的关系数据库是难以胜任

的。Zope 自带的 ZODB 可以非常从容的处理和存储这类数据。

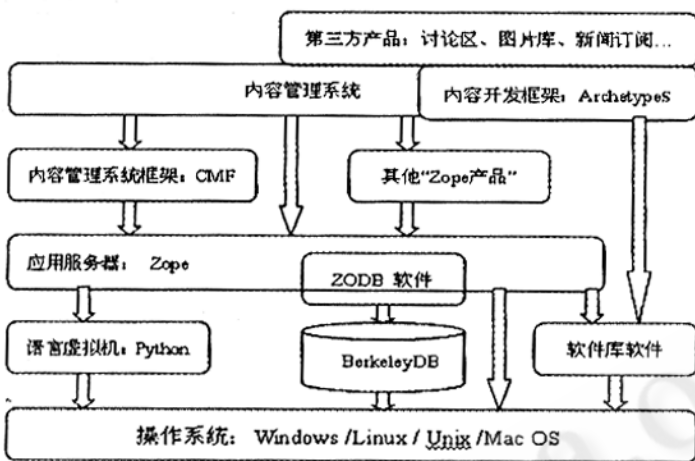


图 2 典型的使用 ZOPE 开发的内容管理系统的体系结构

目前版本的 ZODB 是储存在文件名为单一的 Data.fs 上的。如要避免单一的 ZODB 过大,减少 ZOPE 的负载。可采用 ZOPE 社区提供的 ExtFile / ExtImage, ExternalFile, Ape 等 Product, 将内容对象存储到指定的外部存储区,同时仍由 ZODB 组织和管理这些内容对象。使用 Ape 还可完成 ZODB 与 RDB 的连接。将内容对象储存到 Oracle, PostgreSQL, Sybase, MySQL 等成熟的关系数据库中去。

3.2 实现系统的国际化

在 ZOPE 中,ZOPE 是自动搜寻内建的 i18N (Internationalization) 文件来实现国际化的。该文件就是 ZOPE 的翻译对照文件(po 文件)。可以通过自己手工编辑 po 文件来实现软件的国际化。具体的翻译信息,存放在该文件中,文件应该保存在系统安装的某个产品(Product)的 i18n 目录下。重新启动 Zope,会自动起用新的翻译对照文件。

3.3 对 CMS 显示界面的调整

常规网站界面布局中,一般都包括左右两个面板区,包括日历、新闻、导航树等面板内容,通过 ZMI 可以灵活地配置、调整面板的显示。这在 ZOPE 中成为“皮肤”更换(Change Skin)一个皮肤决定了文档如何显示给最终用户,包括内容周围的图片和样式。一个皮肤聚集了相关的元素,把那片内容包装起来,以某种方式展现出来。要生成用户看到的样子,皮肤有很多元素,包括静态的,如图片;和动态的,如脚本等。ZOPE 社区

的 Product 也有已开发的工具供使用。一个功能较强的并且具有弹性的 Product 名为 CPSSkin。你可以选用已有皮肤来随意整合出你想要的新皮肤给你的 CMS。你可以仅做少许修改或者给出一些较大的改动。

3.4 搜索引擎插件

Zope 可提供快速,灵活的索引,采用 ZCatalog 对象搜索。ZCatalog 对 Zope 网站下所有类型的对象提供可高度定制的全文和域的索引。对象可以根据其内容或者属性索引,能提供结构化的搜索。搜索结果可以来自多种不同的数据源,除了基本的文档外,还可能来自邮件列表的 Email 消息对象,数据库里的记录。Zope 的缺省全文检索 ZCTextIndex 不支持中文全文检索,主要原因是无法区分中文的断词方法(英文采用空格断词)。现在已有上海润普公司为此开发的针对性的 Product: CJKSplitter 可以很好的解决这个问题。

4 结论

Zope 作为一款开源代码的 Web 服务器,又是一种 Web 应用构建的框架。它内带的对象数据库、文件系统和 CMF 框架,可以实现 CMS 的快速开发,同时 Zope 社区不断增加的数百种 Product 产品可以便捷地满足各种不同用户的需求。因此,基于 ZOPE 的 CMS 理应在中小企业实施信息化的解决方案中占有一席之地。

参考文献

- 1 The Zope Book (2.6 Edition), Amos Latteier, Michel Pelletier, Chris McDonough, Peter Sabaini, <http://www.zope.org> (Zope 官方网站)。
- 2 Zope Bible, Zope Bible, Michael R. Bernstein [M]. Scott Robertson, and the Codeit Development Team, Hungry Minds, Inc. [M]. 2002.
- 3 Learning Python, Mark Lutz [M]. David Ascher, O'Reilly & Associates, Inc. [M]. 1999.
- 4 Python 编程宝典, (美) H. M. Deitel, P. J. Deitel, J. P. Liperi, B. A. Wiedermann 著,周靖译 [M], 清华大学出版社, 2003 年 6 月。