

数字化校园信息门户的设计研究

方蔚涛 杨丹 李珩 胡月 汪显伟

(重庆大学虎溪校区网络信息中心 重庆 400030)

摘要 本文就数字化校园信息门户建设,提出了一套较为完整、系统的理论方法。对门户建设中各种关键技术的重要环节进行了详细的分析,提出了门户设计需要遵循的规范和门户的工作原理。

关键词 信息门户, portal, portlet

Design and Reserch of Portal in Digital Campus

FANG Wei-Tao YANG Dan LI Heng HU Yue WANG Xian-Wei

(Department of Information Center, Chongqing University, Chongqing 400030)

Abstract This article gives a complete, systematic and practical method for portal system construction. It utilizes the key technologies in portal system and gives detailed explanations on key problems; also it gives a standard which a portal system should obey and what is the principle.

Keywords Portal, Portlet

1 引言

在URP(大学资源计划)项目中,实施范围越来越广,其他相关的子系统,比如教务子系统、一卡通子系统、办公子系统等众多相关的系统也在URP应用内逐步实施。这些系统分散性应用在学校中各个部门,没有一个统一的整体满足校区应用的一体化的要求,为此,目前URP项目采用EAI来整合所有系统,将各个系统间的“隔阂”消除。当系统整合后,提供给用户使用的有统一的入口,这个入口就是Portal。当然,针对协同应用,相应也会给不同的角色留有一个入口,共享工作中产生的所有信息,解决信息及时传输、信息共享、协同处理业务过程中产生的问题。

2 门户的整体架构

2.1 门户建设的目标

门户为各应用系统提供简单统一的访问点,此外,它还提供了许多有价值的附加功能,例如安全性、搜索、协作和工作流。门户提供了集成的内容和应用,以及统一的协作工作环境。完整的门户解决方案提供工具和用户界面,用于访问信息和应用程序,进行个性化管理和选择内容,让用户随时随地、安全、方便地访问完成他们任务所需的所有东西。门户平台的构建需完成三件事:为各种用户、设备和定制选项提供信息访问;业务流程的集成和自动化;构建、连接和管理应用程序。

2.2 门户的框架结构

事实上,门户建设首先应考虑应用整合,使得用户可以在一个入口进入。因此,门户(Portal)可以被称为“All service from one port”,如图1所示。

信息门户采用一个黑匣子将内部的信息源(数据)和外部的信息源(数据)联系起来。通过一个通道呈现给用户,这个用户可能是内部的用户也可能是外部的用户,用户操作时向

内写数据或从内部查阅数据,为此内部和外部的协同应用也随之完成。

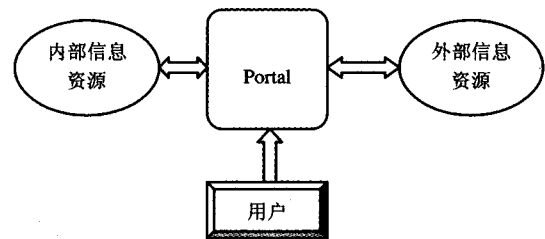


图1

完整的门户体系提供一个可扩展的框架,用于校内应用程序、内容、人员和流程的交互。自我服务特性让最终用户为门户定制和组织他们自己的视图,管理他们自己的档案,发布文档,和他们的同事共享文档。此外,门户提供附加的服务,例如单点登录、安全性、Web内容发布、搜索、协作服务、移动设备的支持和站点分析等。第三代门户技术的提出,拓展了门户的功能与应用范围,因此,构建完善的门户框架还应该包括委托管理、级联页面布局、门户联盟、知识管理和高级个性化,并附有普及计算功能,例如智能通知、脱机浏览和数据同步。

3 门户服务器设计

3.1 设计思想

在高校的数字化校园建设中,我们使用Portal的概念来整合校园内的各种服务。我们知道,Portal其实是一种基于Web的应用程序,它一般对应于多种信息来源提供了个性化(personalization)、单点登录(single sign on)和内容集成(content aggregation)等服务,主要是实现信息系统的展现层,将各种不同系统的信息内容放在一个网页中进行统一展现。

所以,在我们数字化校园的整体设计中,我们通过Portal

方蔚涛 博士生,从事数据挖掘等方面的研究;杨丹 教授,博士生导师,主要研究领域为企业信息化;李珩 硕士生,主要研究领域为企业应用集成;胡月 博士生,主要研究领域为数字图像处理;汪显伟 硕士生,主要研究领域为网络信息安全。

来作为校园网用户访问校园内各种信息的门户网站, Portal 后面连接的大量服务资源提供了 Portal 的内容, Portal 代表各种应用向用户提供服务。同时, Portal 通过与 CertServer 服务器的交互负责用户的登录与信息安全等问题。一个普通的信息展示页面是由顶部的操作导航栏, 中间的内容显示区和底部的信息栏组成。

- 页面顶部的操作导航栏显示了整个应用的图标、标题以及其他一些基本的操作信息, 例如增删定制的面, 为每一个定制页面具体选择页面的内容, 修改网页的颜色风格和帮助信息等。

- 页面中间的内容显示区 (Portlet Main Content) 是整个 Portal 页面的主体部分, 一般由一个或多个 Portlet Window 组成。每个 Portlet Window 就是一个个小的矩形框架, 其中包含 Portlet Title (如天气预报)、Portlet Control 控制按钮 (最小化按钮、最大化按钮等) 和 Portlet Fragment (每个 Portlet 的具体内容) 几部分。

页面底部的信息栏显示一些与具体 Portal 相关的辅助信息, 如图 2 所示。

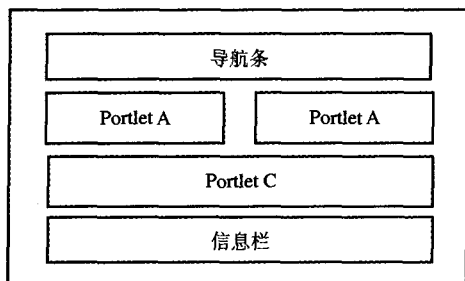


图 2

3.2 门户个性化的设计

个性化是 Portal 提供的一个基础而重要的功能。个性化的服务提供使得用户可以自定义页面的内容、布局和显示风格, 使用户最关注的内容得以最优体现, 提高了用户的访问效率, 增强了用户的满意度和归属感。

首先, 用户可以定制自己的页面, 每个页面可以有自己独特的名称, 或者表示一个专门的栏目, 按照自己的意愿将各个 Portlet 分门别类放入每个页面, 以及控制每个页面的显示次序等。例如, 用户可以定制一个名为“我的首页”的页面, 在这个页面里面存放他最常用的一些服务的 Portlet, 并且让这个页面在他登录后就显示在他面前, 这样就能达到提高用户工作效率的用途。

另外, 用户还能选择自己的页面布局, 例如活页式或菜单式, 以及可以选择每个页面显示内容区周围是否有边框包围, 以及每个 Portlet Window 之间的距离, 对于一个页面来说, 用户还可以定制页面中的其他元素。

3.3 门户组件的功能设计

门户组件 (Portlet) 是用户在 Portal 页面上可以看到的可视化组件, 表现为一个由 Portlet 容器管理的 Portlet Window, Portlet 基于 Java 技术, 它接收 Portal 容器传来的请求进行处理然后显示动态的内容, 具体结构如图 3 所示。每一个 Portlet 显示的内容被称为 Portlet Fragment, 这种 Fragment 遵循 Portlet 规范所规定的显示规则 (例如在 html 页面中, 不能产生 <html>、</html> 的标识等), 并且与其他 Portlet 的 Fragment 内容一起构成了 Portal 页面的主体内容。一个 Portlet 可以非常简单, 也可以进行非常复杂的应用。例如, 它既可以只简单地给用户提供一个计算器、电子词典之类的简单

功能, 也能向用户提供工作流、电子邮箱、搜索服务等复杂应用, 这完全由 Portlet 本身应用或者与后台程序交互的功能决定。

JCP 组织提出的 JSR168 规范定义了组件的实现标准。每个组件对外表现为一个小窗口, 有自己的默认样式和窗口状态。如图 3, 组件有自己的标题, 浏览状态下支持编辑、关闭、上移、下移、最大最小化功能, 编辑状态下支持返回和关闭功能。从各种数据来源提取的信息以组件内容的形式呈现在门户页面中。

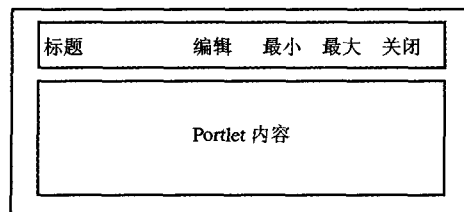


图 3

Portlet 是 Portal 中最基本的元素, 它负责每个服务的具体展示。一个 Portlet 通常代表了一个服务。以财务的 Portlet 为例, Portlet 负责页面的初始显示, 然后读取由 Portal 发送过来的用户的交互信息 (例如工资查询), 接着它负责同财务处的服务器系统进行查询请求, 然后接收财务处服务器的返回信息, 再返回给 Portal 服务器集成显示。

3.4 门户的安全设计

一般来说, 在 Portal 的安全措施研究中涉及到认证、授权和日志记录等。

在认证机制上, 一般采用门户与单一登录相结合的方式, 实现所有信息系统的 SSO, 即通过一次用户名密码验证, 便可在不同信息系统间漫游。

在用户通过认证进入 Portal 系统之后, 系统必须确保用户拥有足够的权限来执行所需的操作, 这就是授权。用户在访问系统数据或功能时可以具有几个级别的权限, 这些权限是管理员根据个人所获许可而授予用户访问系统内信息的。授权通常是通过系统中的访问控制列表 (ACL) 来实现, ACL 包括权限列表, 系统可以根据这些权限判断用户对目标是否能进行读写以及执行等不同操作。访问控制措施可以根据多种标准来实现, 例如基于用户身份、基于角色、基于时间或者基于事务。

Portal 在确定用户身份之后, 会向认证服务器请求该用户的角色和访问控制表, 由于一个访问后台服务的 Portlet 会对应于后台系统的一个服务, 因此根据认证服务器返回的信息就可以决定该授权用户可以操作哪些 Portlet。Portal 的集成模块和页面管理模块也根据这些信息决定用户可以看见的 Portlet 列表。

为了保证 Portal 的安全性与提供审计信息, 类似于认证的日志记录, 还需要为 Portal 服务程序的关键性操作 (例如用户登录、退出、修改配置等) 使用 Log4j 方式提供日志记录信息。

3.5 门户的信息整合设计

门户的信息整合将提供内容管理、内容整合、数据整合、应用整合、流程整合等功能。利用整合技术建设高校信息门户、整合信息资源的最终目标为:

- 将分散、异构的信息资源集成在一起, 使得所有数字化校园的用户可以共享使用。
- 对分散、异构的内容进行整合, 实现统一采集、统一编

辑、统一校对、统一发布。

- 对分散异构的数据进行整合,形成统一的可以方便查询与管理的虚拟数据库,方便不同用户对数据的高效率使用,辅助管理者决策,提高决策的正确率和及时性。

- 对不同的应用系统实现业务逻辑级的集成,使得各个应用系统可以从流程层面实现互联互通。

高校的信息源有三类:结构化数据、非结构化数据和应用系统。结构化数据指数据库、软件或设备输出规范格式数据等;非结构化数据指文件、图片、多媒体等;应用系统包括高校内部和外部的应用系统。结构化数据、非结构化数据和应用系统分别通过各种接口接入信息门户,经过信息整合的内容整合、数据整合和软件整合后,最终以个性化形式展现给用户。

在以上信息整合的过程中,应用系统的集成是其中最关键且复杂的一个环节,我们根据实际情况,提出以下设计思想:

应用系统首先按照逻辑/表示分开的原则分析自己的结构,表示层的结构通过 Portlet 形式安装在统一信息门户系统中,应用系统自身保留对业务逻辑处理的结构,并提供给接口统一信息门户系统访问。当用户使用统一信息门户系统中的某一 Portlet 时,统一信息门户系统会将用户对 Portlet 的操作反映给应用系统的业务处理接口,处理完毕后,会返回给统一信息门户系统,Portlet 则把这些结果反映在页面上。

4 虎溪信息门户构建的技术实现

4.1 门户的体系结构

虎溪信息门户(Portal)的基本体系结构如图 4 所示,Portal Web Application 处理客户的请求,从客户的当前页中提取出门户组件(Portlets),然后调用组件容器来获得每一个组件的内容。Portal 通过组件容器的 Invoker API 来访问组件容器。这些 API 是组件容器的主要调用接口,它们为 Portal 提供了一些基于请求的方法来调用门户组件。容器的使用者(即 Portal)必须实现组件容器的 Container Provider SPI (Service Provider Interface)回调接口,来为组件容器提供与 Portal 相关的信息。最后,组件容器通过 Portlet API 调用所有的组件。

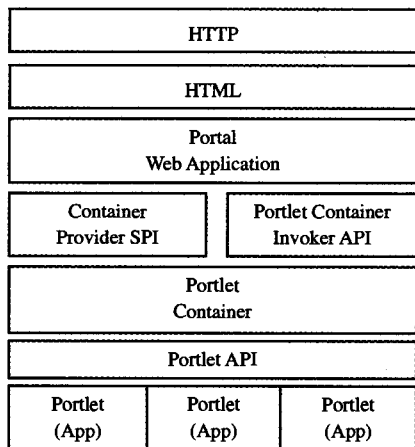


图 4

在此体系中引入了“容器服务”的概念,包含五个方面:

(1)信息提供者(Information Provider)。为组件容器提供关于 Portal 及其框架的信息。通过该接口只能获得一些已知的或存在 Portal 中的信息。这些信息包括带导航状态(navigational state)的 URL 生成、组件上下文(context)、组件

模式(mode)和窗口状态(window state)控制。

(2)工厂管理者(Factory Manager)。定义了如何通过工厂获得一个实际应用的接口。

(3)日志服务(Log Service)。定义了输出日志的方法。

(4)配置服务(Config Service)。定义了如何获得配置值的接口。

(5)属性管理者(Property Manager)。该服务让 Portal 可以获得 JSR 168 规范中定义的属性的值。

4.2 门户技术构成

4.2.1 门户采用的规范

门户的开发遵循 JSR168 规范和 WSRP 标准(Web Service for Remote Portlets),这两个同时出现的标准都发布了开放的源码,它们都完成了在相应的规范中定义的所有必要功能,都把能很好地互相协作作为它们共同的目标。

JSR168(Java Specification Request, Portlet Specification V1.0)用来提供不同的门户和门户组件之间的互通性。只要开发的门户组件遵循 JSR168,则可以在所有遵循 JSR168 的门户上部署运行。

WSRP 是 Web Service 的一种新的商业应用,一种新的标准,主要用来简化 Portal 对于各种资源或者程序整合的复杂度,可以避免编程带来的整合麻烦和问题。而且门户管理员可以从海量的 WSRP 服务中选择需要的功能用以整合到目前所用的 Portal 中。它有三种角色:1.生产者—提供组件;2.消费者—使用组件;3.终端用户—最终用户。它的特点在于生产者将消费者所需要的信息通过 WSRP 返回给消费者,这些信息是相对标记片断,例如 HTML、XHTML 等,可以直接嵌入用户的页面中,而不用像 Web Service 一样开发用户端接口。

实现这个规范,门户可以跟各式各样的数据源打交道,彻底终结信息孤岛的窘境。

4.2.2 Struts and Tiles

虎溪信息门户通过扩展 struts 的基本类 ActionServlet 生成自己的 MainServlet 类,而所有的 HTTP 或 WAP 请求都要提交给这个类来处理,以保证能传递给合适的 PortletAction 来处理。

Tiles 框架为创建页面提供了一种模板机制,它能将网页的布局和内容分离。它允许先创建模板,然后在运行时动态地将内容插入到模板中。Tiles 框架包含以下内容:

- Tiles 标签库
- Tiles 组件的配置文件
- TilesPlugIn 插件

4.2.3 Session EJBs, Spring, and Hibernate

通过使用 EJB,门户系统分别可以在 Web 服务器、EJB 服务器、数据库服务器三层中实现集群。系统中的 EJB 通过 ejb.xml 文件自动生成。

虎溪信息门户使用 JAAS 来完成用户认证安全管理,当一个用户登录后,它的安全属性可以在 Servlet 和 EJB 层中沿用。具体讲,远程 EJB 可以沿用安全检查及权限属性,本地的 EJB 是为其它 EJB 提供业务逻辑服务的,不能被远程调用所以也不必做此类检查;安全原则也派生到 POJO 实现中,而这此实现其实是远程 EJB 的基础类。

虎溪信息门户所有的业务逻辑都通过 Spring 管理的 POJO 来实现,POJO 类通过继承 PrincipalBean 类,来实现有关调用者的方法,所以可以远程调用。如:调用 getUserId() 可以得到当前用户的 ID;调用 getUser() 则返回当前用户的对象。EJB 再继承这类 POJO,实现远程调用。

虎溪信息门户所有的业务数据都通过 Hibernate 来实现并通过 POJO 来调用。在数据库方面,通过 Hibernate 的映射关系实现与各主流数据库的完全兼容。Hibernate 类是根据模式(model)类对应生成的。这样就可以在多层系统中允许模式类是可作序列化处理的,而 Hibernate 类则不必。系统生成的持久层方法有 add, update, delete, find, remove, count。同时生成协助类(Helper classes),可以用来调用持久层方法。协助类调用 Hibernate 的方法来对数据库进行更新操作,但是也可以改写配置文件,使用自己专用的类来完成,但这种类要求继承默认的持久层类。换言之,用户完全可以定制自己的持久层数据,可以是一个正统的数据库,或者是 LDAP 服务器或者其它。

4.2.4 SOAP, RMI, and Tunneling

客户程序可以通过传统的或者无线设备来访问门户。而开发者能够通过开放的 SOAP、RMI 以及客户隧道(tunnel)类来实现对门户的访问。

结束语 本文从信息门户的整体架构出发,提出了设计门户系统的技术手段。信息门户对各种资源进行了整合和管理,提供了统一的访问入口,从而改变了现有信息系统的访问模式。同时,信息门户还为用户提供了个性化的使用界面,使

以前面向单一应用的服务模式变为面向人的服务模式,为今后信息系统的发展提出了新的开发思路。门户的构建与设计是一个比较复杂的过程。灵活运用各种技术,充分了解企业的内部需求,才能构建出合理实用的信息门户。

参考文献

- 1 Terry Lee. Portlet 应用开发(JSR168)
- 2 Eamoi. Portal 架构
- 3 王映雪,沈陪华,陈怀楚. 清华大学信息系统结构发展策略. 中山大学学报, 2001
- 4 BEA Portal management schema
- 5 Martel C, Vignollet L. Educational Web portal baeced on pereonal
- 6 Java Community Process JSR 168. <http://www.jcp.org/about-Java/communityprocess/final/jsr168/>(Accessed Sept. 22, 2004)
- 7 Java Technology: Java 2 Platform, Enterprise Edition (J2EE). <http://java.sun.com/j2ee/>(Accessed Sept. 22, 2004)
- 8 Web Services Activity. <http://www.w3.org/2002/ws/>(Accessed Oct. 8, 2004)
- 9 Myerson J. Human-facing Web services. <http://www-900.ibm.com/developerWorks/cn/webservices/ws-adapt/part1/index-eng.shtml>(Accessed Oct. 8, 2004)
- 10 Lakos A. Personalised Library Portals and Organisational Change. <http://www.lib.uwaterloo.ca/~aalakos/index.html>. Mar, 2001(Accessed Oct. 11, 2004)
- 11 Apache JetSpeed. <http://jakarta.apache.org/jetspeed/site/index.html>(Accessed Oct. 8, 2004)

(上接第 97 页)

为基础,虽然同时考虑了结构和约束但是造成了大量的属性冗余。本文以 XML 键为基础,提出了一种新的转换算法,该算法既考虑到了约束关系,又可以根据第 4 节得到的极小函数依赖集转换过程中的实现规范化,避免了文[9]的分步操作,更重要的是减少了表中大量属性结点的冗余:

算法 2 基于 XML 键的 XML 模式到关系模式的规范化转换算法

i. 寻找 XML 键。利用算法 1 寻找 XML 键的方法得到 DTD 的所有键集合 $\{k_1, k_2, \dots, k_m\}$;

ii. 推理。利用定理 1 的推理规则得到函数依赖的完整集合 Σ' ;

iii. 消解。对由推理规则扩充得到的完整集合 Σ' 利用定理 2 的消解规则得到极小函数依赖集合 Σ , 该集合内包含的函数依赖形式的左部路径仅含键;

iv. 归纳。对于规则推理和消解后生成的极小函数依赖集 Σ , 扫描每一条函数依赖, 同时划分 Σ 为 $\Sigma_1, \Sigma_2, \dots, \Sigma_n$, ($n > m$), 分别对应左部路径的键是 $\{k_1\}, \{k_2\}, \dots, \{k_m\}, \{k_1 \dots k_j\}$ 形式的函数依赖, 其中 $\{k_1 \dots k_j\}$ 表示左部路径是由两个以上的键决定右部路径的形式;

v. 转换。对每个 Σ_i 进行转换得到以 XML 键约束的结点为名的关系表 R_i , 并把左部路径和右部路径中的每个元素结点 E 和属性 A 均映射为 R_i 中的一个属性列, 再将左部路径映射的属性标记为关系表 R_i 的键。

例 5 对于例 1 给出的 XML 模式最终生成满足 3NF 的关系模式, 如下所示:

```
Course(@cno, cname, credit, classhour, experimenthour);
Student(@sno, sname);
Guardian(@gID, gvalue);
Grade(@cno, @sno, grade);
GS(@gID, @sno);
```

虽然关系表 GS 中仅有两个键而没有任何约束的属性, 但如果没有这张表, 我们只能根据 Σ 中的路径还原到 DTD 结构, 而无法根据关系数据库中记录还原 XML 文档, 因为以 guardian 结点为根的子树在还原到 XML 文档中的时候无法确定归属于哪个 student 结点。

结论及将来工作 本文引用 DTD 和 XML 文档的定义, 在此基础上将 XML 模式与关系模式表示形式进行比较, 依据函数依赖的思想, 在 DTD 结构中提出了 XML 键的概念, 并给出算法可以找到所有的 XML 键; 再改进关于 XML 函数依赖的一组逻辑推理规则, 将其分为推理和消解两个部分, 利用定理 1 的推理规则扩充原有的函数依赖集, 弥补 XML 模式向关系模式转换过程中的语义丢失; 再利用定理 2 的消解规则消除重复冗余的函数依赖, 得到基于 XML 键的一组左部路径仅含键形式的极小函数依赖集; 最后以键为中心划分关系, 基于 XML 键得到保持语义依赖的、并且满足 3NF 规范化的关系模式。

本文转化算法的好处在于它是以 XML 键为中心划分属性, 最终可以得到非常简洁的关系模式; 同时将规范化在转换过程中进行, 避免了文档重复扫描从而降低了时间复杂度。该算法不仅对本文的例子成立, 实验证明对任何一个有效的 DTD 均成立。

本文只讨论了 XML 键在 XML 模式向关系模式规范化转换中的应用, 今后还可以做的工作是将 XML 键的概念应用到 DTD 模式规范化和索引的建立的工作中, 得到规范化的 XML 模式和优化的检索。

参考文献

- 1 Arenas M, Libkin L. Symp, A normal form for XML document, Principles of Database Systems. Madison, Wisconsin, USA: ACM Press, 2002. 85~96
- 2 Lee M L, Ling T W, Low W L. Designing functional dependencies for XML. VIII Confrence on Extending Database Technology(EDBT), Prague, 2002
- 3 王元元. 计算机科学中的逻辑学. 科学出版社, 1989
- 4 吕腾, 顾宁, 闫萍. XML 文档的范式. 小型微型计算机系统, 2004, 25(10)
- 5 王庆, 周俊梅, 等. XML 文档及其函数依赖到关系的映射. 软件学报, 2003, 14(7)
- 6 谈子敬, 施伯乐. 函数依赖和规范化在关系和 XML 间的传播. 软件学报, 2005, 16(4)
- 7 吕腾, 闫萍, 王真星. XML 的函数依赖. 小型微型计算机系统, 2005, 26(5)
- 8 萨师焯, 王珊. 数据库系统概论(第三版). 北京高等教育出版社, 2000
- 9 Tan ZiJing, Xu JianJun, Wang Wei, Shi Baile. Storing Normalized XML Documents in Normalized Relations, CIT'05, IEEE, 2005