



编者按:北京万方数据股份有限公司为发展我国“网络资源与建设”,在理论和实践上推动网络资源与建设的发展与应用,以进一步加快图书情报技术网络化进程愿与本刊合作,协办本栏目的工作,为此编辑部代表广大读者对北京万方数据股份有限公司支持我国图书情报领域计算机网络资源与建设发展的举措,表示衷心的感谢!

高校数字化校园应用软件构建平台研究

许鑫 苏新宁

沈风亭

(南京大学信息管理系 南京 210093) (南大百合网络科技有限公司 南京 210093)

【摘要】 由于高校业务的繁多,需求的多变,致使高校的管理信息系统建设一直困难重重。本文试图从软件体系结构模式的角度着眼,首先构建一个在 J2EE 平台上基于 MVC 模式的应用软件开发框架,然后在此基础上设计和实现了一个高校应用软件构建平台。

【关键词】 开发框架 构建平台 管理信息系统 数字化校园 **【分类号】** G43

Research on the Application Software Construct Platform in Digital Campus

Xu Xin Su Xinning

(Department of Information Management, Nanjing University, Nanjing 210093, China)

Shen Fengting

(Lily Network & Technology Co., Ltd., NJU, Nanjing 210093, China)

【Abstract】 Because of the complexity of the operation and the levity of the requirement in high school, it is very difficult to build the Management Information Systems. This paper firstly attends the software architecture pattern, then constructs an application software development framework which is build upon the MVC pattern based on the J2EE, at last describes the design and implementation of the application software construct platform.

【Keywords】 Development framework Construct platform MIS Digital campus

1 引言

60 年代的软件危机使得人们开始重视软件工程的研究。起初,人们把软件设计的重点放在数据结构和算法的选择上,随着软件系统规模越来越大、越来越复杂,软件系统的结构和规格说明显得越来越重要。软件危机的程度日益加剧,现有的软件工程方法对此显得力不从心。对于大型复杂的软件系统来说,对总体的系统结构设计比起对计算的算法和数据结构的选择已经变得明显重要得多,而现在高校的数字化校园系统的建设正面临着这一问题。

人们现在逐渐意识到软件体系结构的重要性,并认

为对软件体系结构的系统深入的研究将会成为提高软件生产率和解决软件维护性问题的最有希望的途径^[1],那么究竟如何快速高质的实现应用系统?如何随着需求快速变更部署?不少专家和开发一线的工作者们提出了诸如框架、构件库、业务中间件、构建平台等解决方案。本文作者根据自己多年的数字化校园开发经验,从 MVC 模式的应用软件开发框架和高校业务构建平台两个层面分析研究了高校应用软件开发过程中面临的问题并给出了行之有效的解决方案。

2 基于 MVC 模式的应用软件开发框架

2.1 MVC 模式

模式分成三种类型:体系结构模式、设计模式、惯用

法。每一种类型都由具有相似规模或抽象程度的模式组成。体系结构模式 (Architectural Pattern) 表示软件系统的基本结构化组织图式, 提供一套与定义的子系统, 子系统的功能, 及用于组织它们之间关系的规则和指南^[8]。它可作为具体软件体系结构的模板, 是开发一个软件系统时的基本设计决策。设计模式 (Design Pattern) 提供用于细化软件系统的子系统或组件, 或它们之间关系的图式。它在规模上比体系结构模式小, 但又独立于特定的编程语言或编程风范, 提供分解更复杂的服务或组件的结构, 它的应用对子系统的体系结构有较大影响^[3,4]。惯用法 (Idiom) 处理特定设计问题的实现, 是具体针对一种编程语言的低层模式。惯用法描述如何使用给定语言的特征来实现组件的特殊方面或它们之间的关系。

模型 - 视图 - 控制器 (Model - View - Controller, MVC) 体系结构模式将一个交互式应用程序分为三个组件^[5]。模型包含应用问题的核心数据、逻辑关系和计算功能, 它封装了所需的数据, 提供完成问题处理的操作过程, 还为视图获取显示数据提供访问其数据的操作; 视图向用户显示信息; 控制器以事件触发方式处理用户输入, 并为每个输入事件提供了相应的操作服务。视图和控制器共同构成了用户接口。

2.2 J2EE 平台下的 MVC 模式

MVC 模式是 J2EE 应用程序开发中被广泛使用的一种体系结构, 它将传统的输入、处理和输出模型转化为图形显示的用户交互模型。J2EE 平台上, 模型层负责表达和访问商业数据, 执行商业逻辑和操作, 同时控制层也可以访问其功能函数以完成相关的任务。视图层负责显示模型层的内容, 它从该层取得数据并指定这些数据如何被显示出来, 它也会将用户的输入传送给控制器。控制层负责定义应用程序的行为, 它可以分派用户的请求并选择恰当的视图用于显示, 也可以解释用户的输入并将它们映射为模型层可执行的操作。

J2EE 下的 MVC 模式层次比较高, 具体到模型层、视图层和控制层这些类型的组件设计的时候, 需要综合所有开发者对这类设计问题所在领域的知识和见解来描述特定问题被验证的解决方案, 换句话说, 为了具体的实现, 需要在 J2EE 下分析设计各类设计模式。在 SUN 公司的网站上有 J2EE 模式完整的介绍, J2EE 技术体系中某个模式和各种模式之间的关系, 迄今定型的模式共 15 种, 分布在 J2EE 的不同层次上, 相互间通过一定的方式随着层次间的交互而交互^[9]。J2EE 是一种框架软件, 提供了相应领域的各个问题的解决方法, 提供了一组 API, 供企业级计算使用^[6,7]。J2EE 还隐含了框架与设计模

式之间的密切联系。框架是骨架, 设计模式是细化, 设计模式是支撑框架的重要组件。本质上设计一个良好可伸缩的应用软件前期工作就是设计一个框架结构。这是一类在 J2EE 技术体系内的框架, 基于 J2EE 技术可以设计许多适合更具体的应用框架, 这样可以更方便的开发, 更高效的重用。

2.3 Lily Framework——一个 J2EE 平台下的应用软件开发框架

(1) Lily Framework 的设计

框架 (Framework) 是一个试图实例化说明的部分完整的软件 (子) 系统。它为一个 (子) 系统族定义体系结构并提供创建它们的基本构造块, 它也定义具体功能特性需要改进的地方^[2]。应用框架可抽取特定领域中的共性问题, 并部分或者全部的加以实现。在进行应用软件开发的时候, 利用框架只需要集中精力完成系统的业务逻辑设计, 它是对于一个软件系统的全部或部分的复用设计。本质上 J2EE (Java Two Enterprise Edition) 是一种框架结构, 它被应用于特定的企业领域, 预先定义了一套在领域内的不变的架构, 如 JTA、JDBC 和 EJB, 可以用来处理企业的数据库操作和事务要求, 提供了系统的、有极强伸缩性和可扩展性的安全解决方案^[10]。

Lily Framework 是在南京大学数字化校园建设的应用系统开发中逐步设计和完善的一个基于 J2EE 平台的应用软件开发框架。最先搭建于南京大学继续教育学院教务系统的开发中, 其后该框架又应用于南京大学海外教育学院教务系统、南京大学研究生院综合管理系统和南京大学人事信息管理系统, 在应用中不断的完善, Lily Framework 的设计有如下一些思路:

①Model 部分: 模型包含应用程序的核心功能, 封装了应用程序的状态, 它对视图或控制器一无所知。Lily Framework 中将其分为两个部分, 一部分是 JavaBean 类似的实体对象, 它对应系统中真实存在的对象, 或者对象之间的关系; 另外一部分是处理对象的商业逻辑。

②View 部分: 视图提供模型的表示, 它是应用程序的外观。它可以访问模型的读方法, 但不能访问写方法, 它对控制器一无所知。当更改模型时, 视图应得到通知。Lily Framework 在显示层主要使用 JSP + TagLib 来完成, 完全脱离 JSP 部分去实现界面太过抽象且不利于实现, 应该提高 JSP 的重用性, 做好前期的规划。现在提供的 4 类验证器 (RequiredFieldValidator, CompareValidator, RangeValidator, REValidator) 能满足用户各种需求。

③Controller 部分: 控制器是应用系统处理具体流程和导向的核心部分。它把模型对象给出的信息转换成视图可以理解的形式, 并且处理系统流程的走向。Lily Framework 中该部分被弱化成一个控制整个系统的 XML 文件, 这部分仍然分为模块 Module 和动作 Action, Module 将对应着应用系统的功能模块, 而 Action 对应具体功能的操作。这样的对应关系可以比较明确, 容易管理。

④Database 部分: 数据库是整个应用系统最核心的部分, 数据库设计的合理与否将直接影响程序。良好的数据库设计应该是尽量合

理,符合第三范式,类型简单化,且仅有一对一关系,一对多关系(主从/父子关系),多对多关系。在数据库的SQL语句的执行,使用通用的数据库操控类,同时要求SQL语句使用框架中定义的格式。

按照上述的设计思路设计出 Lily Framework 的总体框图,如图1所示。在此基础上结合高校的业务流程特点,按照B/S方式下应用系统开发的常用模式,并在数据库设计中遵循中华人民共和国教育部2002年底公布的《教育管理信息化标准第1部分:学校管理信息标准》,快速高质量地完成了一批高校管理信息系统。

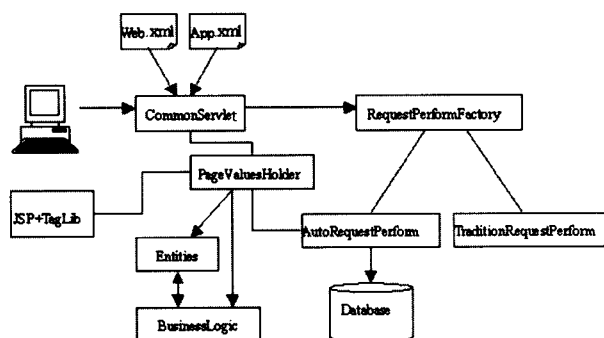


图1 Lily Framework 的总体框图

(2) Lily Framework 的实现

前面描述了 Lily Framework 的 MVC 机理和总体设计框图,并简要介绍了一些核心功能,对于 Lily Framework 的实现可以以 B/S 结构的应用软件系统为例,图2就是一个典型的 Web 请求处理示意图。

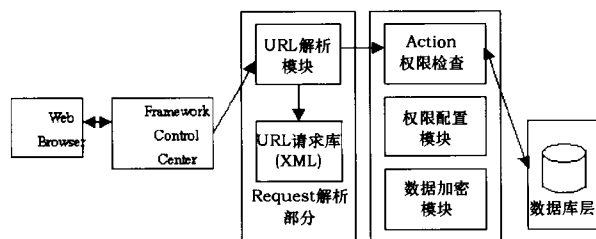


图2 Lily Framework 的 Web 请求处理示意图

其中 Request 解析部分的 XML 文件除了可直接生成 SQL 以外,还可以规定相应的名字和参数,然后在数据库系统中写相应的存储过程,不管前者还是后者都能很方便的增删改相应的数据。另外一些动态查询,Lily Framework 的处理方式是不经过 XML 解析,直接拼接后对应到基础类。权限管理要求每一个页面请求所对应的 Action 都唯一的与模块 (Module) 和组 (Datagroup) 关联,当解析出请求的 Action 后,连接数据库判断当前用户是否有模块操作权限和数据权限,通过 Check 进行后续实体操作,反之则拒绝服务,将信息返回用户。Lily Framework 中的控制器对应到实际的开发中归结为三大任务:写 XML 文件、写基础类、写存储过程。XML 解析后的 SQL 语句以及基础类和存储过程对应的数据库都是和 MVC 中相应的模型相对应。

使用 Lily Framework 时,Request 的解析部分对应总体框图里的 app.xml,它是整个系统中最核心的配置文件,app.xml

文件的名称可在 Servlet 的配置文件 web.xml 文件中进行配置,demo 中命名为 schema.xml。其中:

①app 是整个系统的根节点,代表着 WebApplication。

②module 节点对应 WebApplication 功能模块,其属性 id 是给与这个功能模块的唯一标识,在整个 xml 文档中是唯一不重复的。roles 属性标识该功能模块能操作的角色。

③action 节点对应该模块的一个具体功能操作,其属性 id 是给与这个具体功能操作的唯一标识,在这个 module 节点内可以包含多个 action 子节点,其中 action 的 id 属性值在 module 节点内唯一不重复。roles 属性标识该具体功能操作能操作的角色。

④每个 action 包含 req, res, homePage 节点,req 是描述客户端发送的请求数据的说明,在该框架的现行版本中没有进行处理,因为使用了页面的数据验证器生成了进行数据验证的 JavaScript 脚本,这样可基本满足数据的完整和正确性。

⑤res 节点是整个 app 文档的关键节点,是实现整个系统功能的最重要部分,其包含 autoProcess, autoSql 等子节点,可以容纳多个这样的子节点。

⑥homePage 节点属性 url 值为该 action 操作所对应的 JSP 页面,作为必须值不能缺省。

实现中类实例及数据库自动化技术包括:

①简单 JavaBean 的自动初始化设计,为了充分利用 HttpServletRequest 中的 ParameterMap,只需要建立这样的 javaBean 实体类,其中属性名称跟 ParameterMap 中的参数名称保持一致。

②数据库 SQL 语句参数的自动设置和 RowSet (ResultSet) 到 JavaBean 对象的自动创建,能够实现这样的功能需要对 SQL 语句进行编码的规则要求,比如

```
SELECT name, age FROM student WHERE id = $ id
```

这里参数使用 \$ 开头的字符串,同时 id 这个名称是 Map 参数表中的键值;返回数据集的列名称必须跟生成对象的属性名称一致。关于 RowSet 接口,在框架中使用了 Sun 公司的实现,CachedRowSet 类,这个实例是可以断线存在的,可在数据库的 Statement 和 Connection 关闭后仍然保持数据。

(3) Lily Framework 的应用

Lily Framework 是在南京大学数字化校园建设的应用系统开发中逐步设计和完善的一个基于 J2EE 平台的应用软件开发框架,是一个用于高校管理信息系统领域快速开发的应用软件开发框架。从图3可以看到,Lily Framework 充分利用了 MVC 模式分析,并进一步组合扩展了 J2EE 的相关各类组件,结合应用软件开发领域的需求加以实现,提供了满足需求的相应功能。在 Lily Framework 的基础之上快速构建应用软件业务平台,如高校的教学教务系统,人事系统、科研系统、后勤系统以及其他各类应用系统,这些应用软件在构建时可以通过调用 Lily Framework 中的一些已实现的预定义类。而这一框架也是按照 MVC 的模式构建的,比如页面的显示、报表的打印等功能都可以归为视图类的组件实现,所以,Lily Framework 本质上亦可归为一个高校信息化建设领

域的业务中间件。

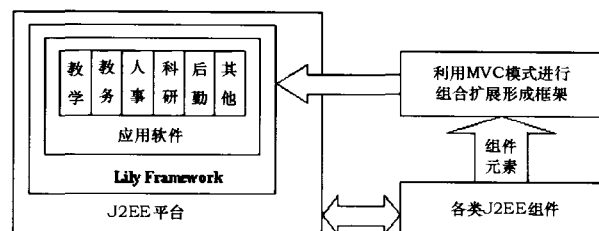


图3 Lily Framework 应用示意图

Lily Framework 已经经过了数个项目的测试与检验,其间也作了不少改动,但是设计思想和核心技术并没有变。实践表明,采用这种方法开发应用软件能扩大软件代码的复用率,提高软件代码的复用质量,降低应用软件开发工作量,从而,提高了软件生产率。具体而言使用此框架后比同等规模的系统开发节省了超过50%的代码量,相应也就节省了接近50%的人月,测试中发现的bug数量也仅为过去开发的系统的1/3。

3 高校应用软件构建平台

3.1 从开发框架到构建平台

虽然应用软件的开发框架可以极大的解决数字化校园中的应用开发面临的问题,但是随着使用发现其也存在一些弱点。首先是层次较低,开发框架主要着眼于可复用类和对象的复用,而这些复用是代码级的复用;其次是粒度较小,开发框架中主要实现的是一些通用功能或者常规操作,没有比较大粒度的构件复用;与前两者对应的是此类开发框架的功能较弱,无法适应日益庞大日益复杂的数字化校园系统建设;最后就是使用人员的定位问题,使用开发框架的基本上是了解熟悉此类框架的软件开发专业技术人员,而对于懂一些技术的业务人员而言,无法满足他们随着需求变更修改系统或者自主开发小应用的需要。现实需要在开发框架上能有所提升,此时更高层次的应用软件构建平台的进入了数字化校园建设者们和维护者们的视野。

应用软件构建平台主要是为了适应学校信息化的各个阶段需求所提供的业务支撑平台,该平台的目标定位是支持用户通过统一的业务平台快速构建学校完整的业务流程,实现学校内部、学校之间的业务整合、流程整合和数据整合,最终立体化的整个“数字化校园”体系。应用软件构建平台是“数字化校园”体系架构重要的组成部分,它使开发人员能快速、高效地构建业务功能模块。开发人员只需开发应用组件,将应用组件插入框架之中即可构建应用程序。

一般而言,应用软件构建平台是基于开放或共享标

准的实现面向产品化、实用性的组件库系统,并具有开放性、可扩展性;支持异构环境中的框架、组件的互联和通信由 Web Services 实现;实现新旧系统的兼容性;遵循重要组件标准,具有透明本地化、平台无关性特点;系统的配置、数据交换基于 XML 和 Java 的标准化格式;支持个性化信息服务定制和菜单重构。

应用软件构建平台为应用开发提供一致的、健壮的、可扩展的、灵活的骨架,其可以提供两种服务接口以及各种具体的公共组件:一是对业务组件提供服务的接口,这些接口是基本固定的;二是提供公共组件具体实现的嵌入和管理服务,公共组件包括灵活报表、通用查询、打印、通用导入导出等等。

3.2 设计思路

在整个设计中,充分考虑了学校信息化的特征,主要体现在以下几个思路上:

(1) 数据驱动思路

学校的数据所呈现的层次是立体的,在这个立体结构中,各部分数据对象是相对稳定的,对象属性的特征也是相对稳定的,所以在充分保证数据完整性、稳定性的基础上,在平台实现中,实现了以业务数据驱动应用、以控制数据驱动流程处理、以变化数据驱动交换的技术。

(2) 分级管理思路

学校的应用具有分级分层的立体结构特征,通过分级授权,可以大幅提高系统的维护效率和使用效率。

(3) 组件化部署思路

在整个平台开发过程中,一直定位在用户可以自我管理、自维护。基于所提供的工具和组件,用户可以根据自身需求,通过配置快速构建自身应用。

(4) 模块化组装思路

基于数字化校园应用平台所构建的所有应用均采用模块化接口,均可以自由拆分、组合,是最大限度地支持用户实现学校应用集成。

(5) 标准化持续发展思路

通过和原厂商的合作(包括 SUN、IBM、微软),在系统中采用国际通用技术标准,如 XML、SOAP、HTTP、LDAP 等,保证平台软件的可持续升级。

应用软件构建平台采用组件化、模块化开发思路,在系统方面提供统一用户管理、统一信息发布、统一数据库机制,在业务方面提供 workflow 平台、业务建模和应用生成机制、报表查询工具、通用交换服务、通用文档管理等一系列实用的工具包,各组件的实现采用面向对象的机制,最大限度地提高了各组件的功能复用和代码复用程度、提高了系统的运行可靠性和功能一致性。

3.3 体系结构

在 J2EE 平台上如何设计我们上面的表述的高校应用软件构建平台呢? J2EE 使用 EJBServer 作为商业组件的部署环境,在 EJBServer 中提供了分布式计算环境中组件需要的服务,例如组件生命周期的管理、数据库连接的管理、分布式事务的支持、组件的命名服务等。J2EE 用于实现应用服务器有其优势,它可以利用 Java 语言自身具有的跨平台性、可移植性、对象特性、内存管理等方面的性能,为应用服务器的实现提供一个完整的底层框架。J2EE 中定义的各种服务,包括 JSP 和 Servlet 容器、EJB 容器、JDBC、JNDI(名字目录服务)、JTS/JTA(事务服务)、JMS(消息服务)等,也分别为应用服务器提供了各种支持。底层框架给出了许多的扩展点来鼓励可扩展性,组件被定义在适当的层次并提供一致的逻辑接口,从而着重于简单化和模块化,同时鼓励可重用性,组件的设计遵循标准模式和最佳实践,以保证性能和可伸缩性。图 4 就是一个应用软件构建平台所采用的体系结构。

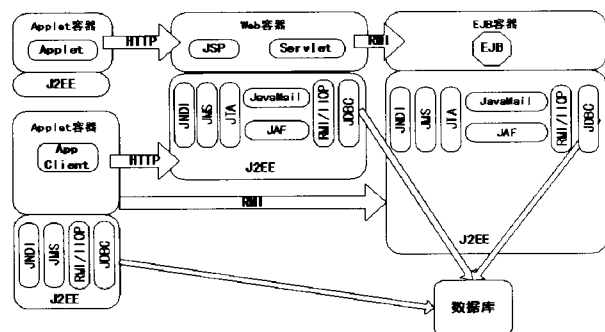


图4 应用软件构建平台体系结构

此处,MVC 模式的应用有了进一步的发展,如图 5 所示。视图层一般为浏览器上显示的页面,但现在构建平台加强了客户端的交互能力及对打印、报表的支持,保留

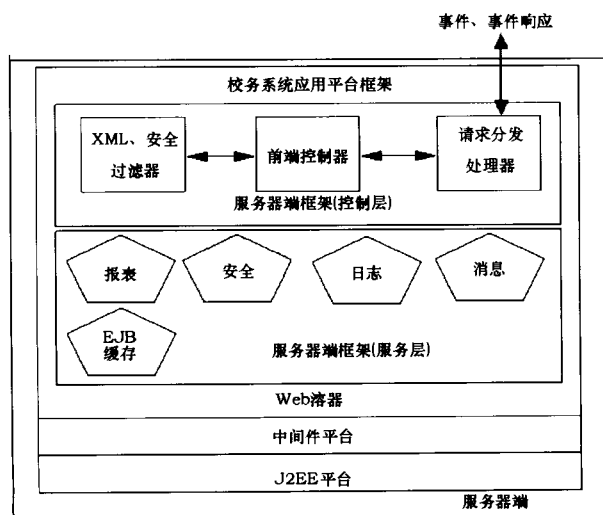


图5 应用软件构建平台水平层次结构

了用户对界面的操作习惯。构建平台将控制层封装成数字化校园系统应用平台框架,该框架的底层是 J2EE 平台,中间件平台提供的系统级服务提供事物、安全、持久性等基本 J2EE 服务;中间层是系统框架总体,并在设计分成两个主要部分:流程控制及系统服务。其中流程控制部分负责控制整个系统的处理流程,包括请求的过滤和分发等等;系统服务提供了许多应用所需的服务,如:安全、日志、交易、EJB 缓存等。

在模型层中采用 J2EE 提供的 EJB 技术将核心的业务封装成组件的形式,从而使业务层得到最大限度的重用,如图 6 所示。也就是说,在组件的数据交互接口不变的前提下,如果业务逻辑发生变化则只需要修改组件内部逻辑,实现了层次间的松耦合和业务逻辑的封装两个优点。

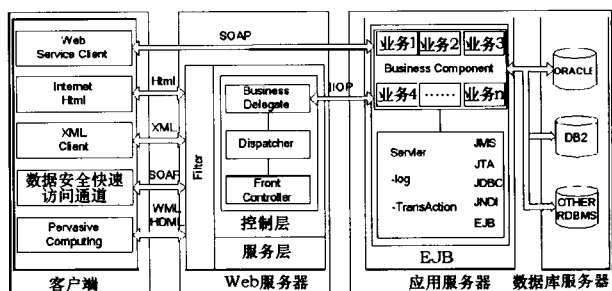


图6 应用软件构建平台垂直层次结构

分级的体系结构有助于实现体系结构的目标。例如,可以通过机器增加和减少处理能力来实现可伸缩性;可靠性则可以通过将处理能力分布到多个机器上来实现;互操作性的实现则是通过采用通用的网络和进程间的通信机制。这种级之间的隔离可以将某一级的改变对其他级的影响降到最小,由此提高了可扩展性和可重用性。图 7 就是一个典型的应用系统分级模型。

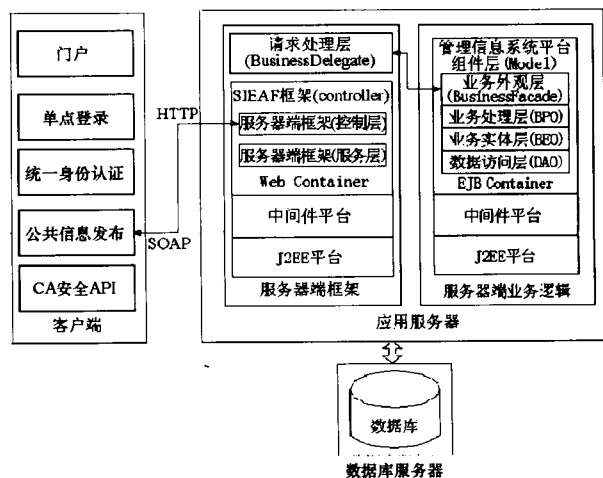


图7 应用软件构建平台分级模型

3.4 组成部分

应用软件构建平台提供了各种各样组件和引擎,这些组件和引擎可以很方便的组合使用,以安全组件为例,安全组件通过确认各方的身份和信息及过程的完整性来保证应用的可信,为构架提供认证和授权的支持,用户认证确保用户确实是其声明的身份。每一个经过认证的用户被赋予相应的信任状和许可,授权使系统可以基于认证用户的安全信任状识别、区分并决定访问控制,可为跨越教育系统的各个应用层实现统一的认证和授权提供一套通用的基础架构。

比较重要的引擎包括:

(1) 服务引擎:服务引擎是建立使用共享、可复用、分布式应用组件的重要工具,是我们数字化校园应用架构调用服务的入口。服务可以用各种方法实现,如工作流、Web Services 和异步服务(JMS)等,服务提供者对使用者完全透明。

(2) 工作流引擎:在 MIS 系统中,对那些基于状态变化的业务,采用工作流来描述业务过程,使业务流程动态重构和自动化;工作流引擎是 WMS 的核心软件组件。它的功能包括:解释过程定义,创建过程实例并控制其执行,调度各项活动,为用户工作表添加工作项,通过应用程序接口(Application Program Interface, API)调用应用程序,提供监督和管理功能等。工作流执行子系统可以包括多个工作流引擎,不同工作流引擎通过协作共同执行工作流。

(3) 消息服务引擎:消息服务引擎包括 JMS、E-mail、Forum、Bulletin,其中邮件组件提供标准的 SMTP 邮件服务,包括支持多个接收方、CC、BCC 和附件。基于 Java Mail API 实现 Send Mail 服务。消息组件基于 Java JMS API 为应用间的异步消息交换提供可靠、灵活的服务。

(4) 实体引擎:基本实体对应于实际数据库表结构,视图实体对应于数据库视图。采用实体引擎,以 XML 数据定义对象到关系实体的映射(O/R Mapping),实现对关系数据的存取。

(5) 报表引擎:提供一致的方式,开发系统中的各种业务报表;报表组件为 Java 应用提供报表的功能。这一组件支持 JDBC 数据源,并能本地生成 Pdf 和 Excel 格式的报表。

(6) 日志引擎:为了监视代码中变量的变化情况,周期性的记录到文件中供其他应用进行统计分析和跟踪代码运行时轨迹,作为日后审计的依据;担当集成开发环境中的调试器的作用,向文件或控制台打印代码的调试信息。数字化校园系统提供日志引擎组件,日志组件提供对应用和组件进行错误诊断、审计/监视的统一机制。

构建平台还包括工作调度引擎(实现即时提醒等任务定时调度功能)、打印引擎(主要负责在打印控制器的控制下将接收到的打印内容转印到打印纸上)等。

构建平台包括一项很重要的机制——异常处理。系

统开发一种错误消息处理机制、并向用户提供有用的、清楚的、有意义的信息。一旦事务失败,服务器就会向系统发出一个用于帮助用户发现并修复绑连的数据库错误信息。系统通过异常处理功能来获取这种异常信息,并修复出现的故障。

3.5 平台特点

应用软件构建平台坚持保持数据的完整性和稳定性,以数据为中心,完全基于 XML 实现数据驱动,支持学校应用之间的数据整合。

其灵活的流程定义功能,支持跨平台应用的业务流程整合和面向数据的不同层次授权,包括数据对象(数据表)的授权,数据属性(数据字段)的授权管理。而且用户所见即所得,一方面保证应用扩展的简单快捷,一方面也支持用户在模板中实现的自定义扩展和自定义界面。

由于采用面向对象、组件开发等技术,应用软件构建平台同时也是一个信息集成平台,具有二次开发能力和与其它系统集成能力。其提供的开放式接口,符合国际标准接口,可用于和学校原有应用系统的应用集成,建立与学校外部系统的信息交换标准接口,并预留与未来发展系统之间的接口。

一系列完善的机制保证高校应用软件构建平台的通用和强大,包括提供可靠灵活的安全和管理机制,提供专用灵活的权限控制工具,提供丰富的运行监视机制(可对系统性能、模式运行、用户操作等监视),提供完善的消息处理机制(处理范围包括流程通知、系统运行监控信息、数据变更通知以及在线用户沟通,同时用户能够根据预先设置的规则,主动提示相应用户进行任务处理,在使用过程中,若有新的事务到达,系统也能主动给出等明显、及时的提示)等。

4 结 语

高校的数字化校园建设,归根结底是平台基础上的应用与资源建设,如何快速高效的建设各类管理信息系统成为困扰各个高校的一大难题。在实际的系统开发中,我们首先提炼出共性的软件开发框架,然后随着研究的深入和应用的拓展,不断地完善框架基础上的构件、引擎等复用机制,逐步形成了基于高校的应用软件构建平台。当然,现在的实现机制主要还是以复用性程度比较高的功能性模块为主,涉及到个性化或者快速定制方面还有很多不足,这也是我们需要在应用软件构建平台后期开发中花大气力完善的内容。

(下转第36页)

件配置推荐:

CPU:主频 500MHZ 以上

内存:128M 以上

硬盘:10G 以上

②软件需求

操作系统:Windows 2000 + SP3 或 SP4 补丁。

浏览器:IE5.0 或以上版本。

IPv6 协议栈的补丁:tpipv6-001205-sp3-ie6.zip

(2)安装 SP3 或 SP4 补丁

(3)安装 IPv6 补丁

将 tpipv6-001205-sp3-ie6.zip 解压到某个目录下,在 setup 的子目录下,打开文件 hotfix.inf,修改其中的参数 (NTServicePackVersion)。若补丁为 SP3,则参数为 NTServicePackVersion = 768;若补丁为 SP4,则参数为 NTServicePackVersion = 1024。

(4)运行 hotfix.exe,系统重启后,安装 IPv6 协议栈

(5)IPv6 网络配置

Windows 下的 IPv6 的配置是基于命令行的形式,其主要的命令就是 IPv6。因为配置在系统重启后会自动消失,建议编写专门的配置批处理文件,放在开机启动文件夹中。

(6)浏览器的配置

在安装 IPv6 补丁时,wininet.dll 文件被替代为支持 IPv6 的版本,因此,当补丁安装后,浏览器可自动支持 IPv6 网页的浏览,无须任何改动。

4 基于 NSTL-IPv6 网络平台的文献信息服务系统的实现及关键技术

即使 IPv6 网络平台能够提供更多的终端接入,更高的安全保证和质量保证,但是如果没有应用,也就没有任何意义。NSTL-IPv6 城域网在建设之初就充分考虑到

这个问题,强调应用的发掘,强调从提供丰富的资源着手,开发新一代文献信息服务系统,这正是 NSTL-IPv6 城域网的生命力所在。

目前,NSTL-IPv6 网络平台上已经建成的文献信息服务系统包括信息检索、专家咨询、期刊目次浏览等内容,拥有的文献量达 300 多万种。整套应用系统采用 Java、C++ 语言开发,后台数据库为 TRIP 全文数据库。

5 结 论

NSTL-IPv6 文献信息服务平台从信息服务领域挖掘 IPv6 应用,希望能够通过一定用户群体的实际使用,创造培育 IPv6 用户的途径与经验,为 IPv6 公用网的发展和商业化贡献一份力量。

虽然目前 IPv6 的杀手级应用还没有出现,但是互联网上必将出现越来越多新颖而精彩的服务,而 IPv6 将是实现这些服务的关键。

参考文献:

- 1 雷震洲. 下一代服务与 IPv6. 现代电信科技,2004(1):2-4
- 2 CERNET IPv6 试验床. IPv6 相关文档. http://www.ipv6.net.edu.cn/2000_7/reference/REF/index.html. (Accessed Apr. 10, 2005)
- 3 Microsoft Corporation. IPv6/IPv4 Coexistence and Migration. Microsoft @ Windows@ Server 2003 Technical Article; Aug. 2002
- 4 乐德广,吴孙桃. NAT-PT 技术及其在 FreeBSD 系统中的实现. 电信科学,2003(2):20-23
- 5 王志淳,吴峰. IPv6 关键技术的分析. 中国有线电视,2003(23):13-16
- 6 张鸿,钱华林. IPv6 对域名系统的需求及其解决方法的研究. 微电子学与计算机,2003(1):35-38

(作者 E-mail: wangli@istic.ac.cn)

(上接第 53 页)

参考文献:

- 1 Mary, G. David. Software Architecture: Perspectives on An Emerging Discipline, Prentice Hall, Inc., Simon & Schuster Company, 1996
- 2 N. Medvidovic, R. N. Taylor. A classification and comparison framework for software architecture description languages. IEEE Trans Software Engineering, 26(1):483-491, 2000
- 3 R. N. Taylor, N. Medvidovic, K. M. Anderson, D. L. Dubrow. A component and message-based architectural style for GUI software. IEEE trans Software Engineering, 22(6):390-406, 1996
- 4 M. W. Denise, V. M. David. Component independence for software system reliability. Proc Quality Week, Europe (QWE98), Belgium: 78-85, 1998
- 5 C. Hofmeister, R. Nord, D. Soni. Applied Software Architecture. ADDISON-WESLEY, 1999:4-60

- 6 J. Engel. Programming for the Java Virtual Machine. Addison Wesley:24-28, 1999
- 7 T. Lindholm, F. Yellin. The Java Virtual Machine Specification (2e). Addison Wesley:78-82, 1999
- 8 P. C. Clements, From Subroutines to Subsystems: Component Based Software Development. American Programmer, vol. 8, No. 11, November 1995
- 9 The Java virtual Machine Specification. <http://java.sun.com/docs/books/vmspec/2nd-edition/html/VMSpecTOC.doc.html> (Accessed Dec. 18, 2004)
- 10 使用 J2EE 设计面向服务的体系结构框架. <http://www-900.ibm.com/developerWorks/cn/webservices/ws-designsoa/> (Accessed Dec. 18, 2004)

(作者 E-mail: xuxin@njin.edu.cn)