

基于 Schema 的 Web 框架设计

潘仙张, 米根锁

(兰州交通大学电子与信息工程学院, 兰州 730070)

摘要: 现有 Web 框架没有充分考虑面向 pojo 对象的编程和 XML 到关系数据库的映射, 导致系统中 pojo 对象与数据库关系表和其他对象之间的紧耦合。针对该问题提出一种基于 Schema 的开发框架, 该框架考虑数据库关系表与 DTD 的依赖, 集中且合理地布置系统中的 pojo 代码, 方便 service 对象调用 DAO 对象, 是一种易于扩展和维护的 Web 框架。

关键词: 软件框架; 模式; Schema 模型

Web Framework Design Based on Schema

PAN Xian-zhang, MI Gen-suo

(School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070)

【Abstract】 Existing Web framework does not take fully consideration of pojo object-oriented programming and the mapping of XML to relational databases, that resulting in tight coupling between the database table and other object with pojo. Aiming at this problem, this paper presents a development framework based on Schema, which considers to the database table reliance on DTD, layouts of the system code pojo intensively and rational, and convenient for service object to call DAO object. This framework is easy to expand and maintain.

【Key words】 software framework; pattern; Schema model

XML 能实现平台无关性, 即与数据存储平台和表示数据模型无关, 是一种很好的交换格式。利用面向对象编程的优势, 本文给出面向 pojo 对象的编程思想, 用 XML Schema 模型表述 pojo 对象, 利用 ant 实现从对象到关系表的自动转换, 解除对象与关系表之间的紧耦合。

1 目前流行的 Web 框架^[1]设计及其缺点

表示层中的输入对象设计以关系表中的字段为依据。控制层负责对表示层中所请求服务的转发。模型层负责表示层中的业务服务以及对数据库的访问, 数据库访问层的对象以及业务逻辑层的对象通常基于数据库表字段设计。整个 Web 框架设计以操作数据库关系表为中心, 框架的显示层、控制层、数据层设计的对象以 sql 直接操作关系表为基础。以此框架进行开发时, 先要设计导入数据库表, 再依据数据库关系表设计调试各个层中的对象。当有需求变动数据库关系表时, 会变动该框架各个层的对象, 修改的变动很大, 不易于系统维护与升级。

2 本文框架设计研究的方法论

2.1 XSchema 的定义^[2]与 XSchema 模型的优点

在给出 XSchema 定义前, 假设 E 是一系列元素名字, A 是一系列属性名, t 是一系列不可分数据类型(如 string, integer, date, ID, IDREF 等)。下文给出 XSchema 的定义: 一个 XSchema 是五元组 $X(E, A, M, P, r)$, 其中, E 是 E 中元素的有限集合; A 是一个元素名字 $e \in E$ 到多个属性名 $a \in A$ 的映射; M 是一个元素名字 $e \in E$ 到元素类型定义的映射, $M(e) = a$, a 是一个正则表达式 $a ::= \mu | t | a+a | a,a | a^*$, μ 代表空元素, $t \in t$, “,” 和 “*” 分别代表连接和闭包; P 是一个属性名字到它的属性类型定义的映射 $P(a) = b$, b 是一个四元组 (t, n, d, f) , t 是不可分的数据类型(如 string, integer, date 等), n 为 “?” (可空)或 “-?” (不可空), d 和 f 是与 μ 的有限集, 如果 t 是

IDREF 或 IDREFS, 则 t 还需要指定目标类型, 并用 “—” 指出; $r \in E$ 是根元素有限集合。本框架用 hibernate 中的 Schema 文件来描述 pojo 对象, 该 Schema 的作用是把 pojo 对象中的属性与数据库关系表中的字段对象对应, 直接从 pojo 对象生成数据库的表。框架的显示层、控制层、模型层的对象通常基于 pojo 对象, 数据库关系表的生成也基于 pojo 对象。当数据库关系表需求有变动时, 只要升级 pojo 对象, 并利用 ant 自动依据 pojo 对象重新生成数据库关系表, 提高了开发效率与可维护性。显示层对象、控制层对象、业务逻辑层对象通常可以保持不变。以数据库关系表为中心的设计思想转到以 pojo 为中心设计思想。

2.2 Web 系统设计原则

图 1 描述了 Web 系统分析与开发评估迭代。

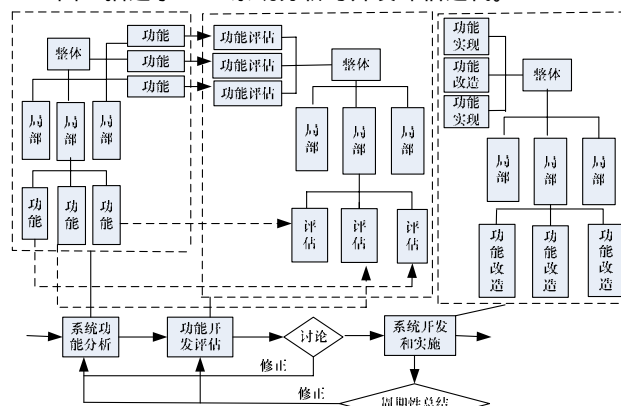


图 1 系统分析与开发评估迭代

作者简介: 潘仙张(1981 -), 男, 硕士, 主研方向: 软件体系结构, 计算机应用技术; 米根锁, 教授

收稿日期: 2008-12-01 **E-mail:** arcxianzhang@163.com

对 Web 系统进行系统分析, 确定 Web 平台应具备的功能以及此类功能达到的目标。系统分析和开发评估依据以整体把握局部、以局部丰富整体的原则。系统分析和开发评估是一个迭代式过程, 需要反复修正和完善。

2.3 系统功能分解目标

按系统体系结构^[3]的要求, 对系统的功能进行合理和各种粒度的分解。依据以整体把握局部、以局部丰富整体的原则。在分层分布式体系结构中, 整体性指基础性的功能, 局部性指应用性的功能。系统功能分解最终将系统功能按照整体性和局部性的特点进行分解。从功能类型中子功能的数目上来看, 系统功能分解最终得到一个倒金子塔状的功能结构。在某一个功能类型内部也存在整合和局部的概念。即金字塔是嵌套的。参考系统功能依据, 对 Web 系统平台的系统功能做出合理的层次划分, 得出如下层次结构: (1) 展现层包含人机界面、综合报表、综合图形、信息服务等功能模块。(2) 应用层包含调用控制、视图表示、OLAP、数据挖掘、运行考核等功能模块。(3) 数据层包含数据交换、抽取组合、CIS 接口、专用接口、规约接口、Web Service、Cim/xml 导入导出、文件、模型中心、实时数据中心、历史数据中心、图形中心、版本控制、数据资源管理等功能。(4) 模型层包含模型转化、模型建模、模型导入导出、模型变更通知、建模工具、建模管理、图模工具等功能。(5) 系统层包含消息服务、日志服务、主备服务、文件服务、在线帮助、网络管理、安全管理、数据库系统、系统管理、进程管理、基于 CORBAL 总线等功能。(6) OS 层包含 Windows, HP-Uinx, Aix, Linux。(7) 硬件层包含 SUN, HP, IBM。此系统功能层次结构很好地解释了系统各层次中的功能在系统中的地位及其开发的先后顺序。该系统各层类似 TCP/IP 7 层结构中各层自上而下的依赖关系, 在层次中也存在类似的自上而下的功能依赖。

3 框架设计

3.1 显示层(V层)

显示层^[4]的作用是提供直观、简便易用的界面在网络上与客户互动, 是信息系统获取或输出信息的接口和前台, 客户可以方便地提出要求。得到所需信息在编写浏览器端的 HTML 代码时, 基于浏览器代码尽可能复用的原则, 采用 XSLT 模板解析 XML 生成 HTML 在浏览器显示的设计思想, 在具体实现时, 应在 jsp 页面中指定该 jsp 页面的后台服务器相应的 Servlet。当 URL 请求该 jsp 页面时, 上述控制器会在后台调用相应 XSLT 与 XML 代码生成登陆页面的 HTML 代码发给浏览器显示。对整个 Web 业务逻辑的分析, 在总体上把整个业务的显示页面分为数据的插入页面、数据的详情页面、数据的更新页面, 数据的查询页面。为每个页面编写 AJAX 代码时, 把相应的 AJAX 放在一个 JS 文件里, 在服务器程序中根据要生成的具体的 HTML 页面把相应的 JS 文件指定到相应的 XML 文件里, 此类 XML 文件再与相应的 XSLT^[5]文件生成 HTML。AJAX 集中放在一起, 如果以后有业务变动时, 只要变动该 JS 文件中的 AJAX 代码, 而其他内容, 如服务器代码与该页面的 XSLT 文件一般不变。把大部分 XSLT 文件用到的公共显示效果单独组织成一个 XSLT^[5]文件, 如日历(Button.xslt)、按钮(CalendarTemplate.xslt), 而其他要用到日历和按钮的 XSLT 文件则引用这些 Button.xslt 和 CalendarTemplate.xslt。需要升级该按钮时, 只要变动此 Button.xslt 文件即可, 其他引用该 XSLT 的所有文件一般保持不变。

3.2 控制层(C层)

在逻辑上写一个控制类^[4], 该控制类接受浏览器端的请求, 并根据浏览器端的请求生成相应业务逻辑服务对象, 图 2 描述了具体的 UML 图, 其中, 本文的 Servlet 层在整个 Web 中的设计如下: public class WebBaseServlet extends MultiActionController。各个浏览器页面的 Servlet 通常继承该 WebBaseServlet, 而每个控制类一般根据覆盖 LoadPage()方法来实现每个 jsp 页面的不同请求服务。该控制类的目的是进行服务转发, 当上述 jsp 页面由于某种原因在处理业务逻辑变动时, 可以只变动该控制层中的 LoadPage()方法, 而其他代码一般可以保持不变。

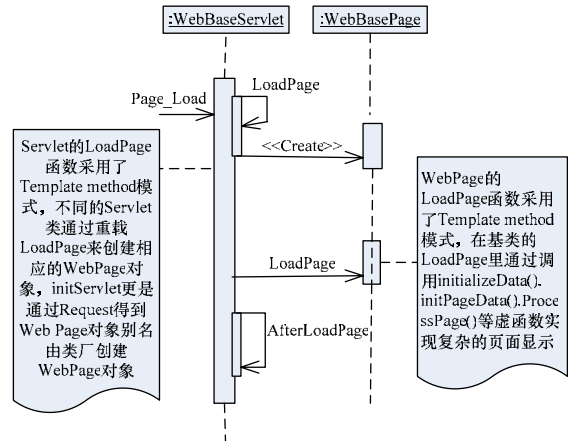


图 2 浏览器端与 Servlet 的交互

3.3 模型层(M层)

本框架将模型层^[4]进一步划分成数据访问层与业务逻辑层。数据访问层的主要作用是提供一个性能良好、使用可靠、开放和易于扩展的支撑环境。数据访问层分为访问共享数据库和访问分析数据库, 实现对基础数据和分析数据的分散管理。共享数据库的数据来源于企业内部资源的数据信息和通过用户界面获得的客户信息。即存储提供动态实现整个集成系统所需的产品信息和客户信息的提取查询等信息交互功能数据。分析数据库将相关客户信息和产品信息进行数据处理, 为上层决策提供依据。在组织数据层访问层的代码方面主要是扩展 HibernateDaoSupport 中的类库。抽象基类 BaseData List 的主要作用是对 Data(数据访问层)的包装, 具有自注册功能, 包括数据库连接、连接字符串、应用程序组、应用程序名和用户自定义属性。Data 主要负责操作 pojo 对象, 并定义 hibernate 中 Schema 文件的属性与数据表关系表中的字段对应关系以及在 Schema 中组织 pojo 各个对象之间的关系。需要访问数据库时, 通常继承该 BaseData List 来进一步封装数据访问层的类。在本框架里提供了基本数据访问层的类如下: UpdateData List 类继承 BaseData List 类, 它具有实现数据的各种显示, 生成对数据库更新的数据集, 处理数据类型例外、表信息、字段转换表、字段事件列表、约束集合、应用程序组、资源管理器、资源文件名、资源名, 和禁用页面等功能。XmlBaseData List 类继承 BaseData List 类, 它只提供浏览数据功能, 不提供数据提交功能的服务。最主要的是 XmlData List 类继承 UpdateData List 类, 对上层业务逻辑层提供统一简单的数据访问层接口, 定义了每个具体的 jsp 页面对数据库记录操作的 Insert, Update, Detail, List, Delete 5 种模式, 因此, 本文 XmlData List 类提供了 FillUpdateTables(), FillListTables(), SetInsertXmlData(), SetUpdateXmlData(),

SetDetailXmlData(), SetListXmlData()方法来支持上述 5 种操作模式。Data List 工具类根据 AOP 思想,对 XML 数据的处理以及将 DataTable 中的记录转化成 XML 时,用 DataList SetUtil 类来处理,并通过代理模式把 DataListSetUtil 类织入到本框架中的 Page 服务类。在数据转化格式有变动时,只要改变 DataListSetUtil 类即可。

在实现 jsp 页面的业务逻辑服务时,用 Page 服务类,其中,抽象类 WebBasePage 具有对 Page 进行包装的功能、自注册功能以及 Page 和相关对象、页面地址、页面风格等方面的功能服务。在本框中提供的基本 Page 服务类如下: WebBaseGeneratorPage 类继承 WebBasePage 类,含有 DataList Set 的页面的功能; WebBaseXmlPage 类继承 WebBase GeneratorPage 类,具有 XML 解析功能的页面; WebXml HttpPage 类继承 WebBaseXmlPage 类,具有数据提交功能。数据提交时序见图 3。

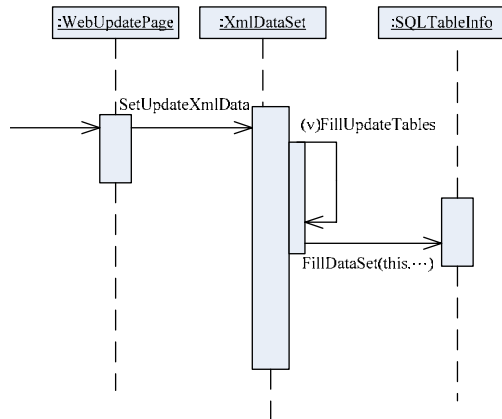


图 3 数据提交时序

3.4 基于本框架所设计系统的运行原理

系统一旦运行就创建 Application 的对象并根据 Web.config 的内容初始化 Application 对象的有关数据类型注册 Application[WebBasePage.ApplicationData]=AGlobal。创建 Session 对象:初始 FileGroup 的数据,生成 Transform 对象(该 ITransform 的类型由应用程序重载),初始化 UserInfo 对象,根据 UserInfo 对象初始化 UserRight 的登陆信息,初始化 UserRight 的 DataRight 对象(该 DataRight 的类型由应用程序

重载),初始化 UserRight 的功能权限信息并进一步初始化 Transform 对象的 HeadLeft 对象 Session[WebBasePage.SessionData]=AppSessionGlobal。基于本框架的系统运行原理见图 4。

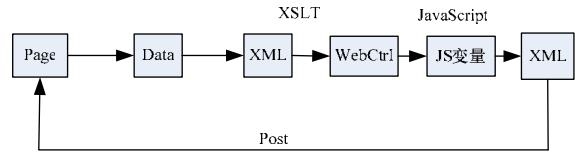


图 4 基于本框架的系统运行原理

4 基于本框架系统的分析

本框架采用 Schema 模型开发软件,转变了以关系表为中心的程序设计思想,体现了面向对象的思想,较好解决了由数据库关系表变动带来的代码维护问题,图 5 显示了当数据库关系表有变动时,本框架与目前流行的框架的修改情况,包括对数据库关系表、显示层、控制层、模型层中各个对象的修改情况。

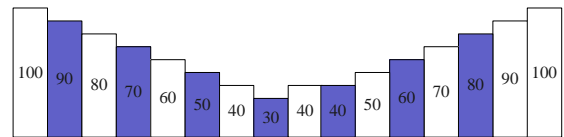


图 5 本框架与目前流行的框架的修改情况

在图 5 中,浅色长方形为目前流行框架的修改量;深色片断为本框架的修改量。可以看出,本框架降低了修改工作量,提高了软件可维护性。

参考文献

- [1] Fowler M. 企业应用架构模式[M]. 北京: 机械工业出版社, 2004.
- [2] 胥正川. 基于关系数据库的 XML 数据存储、更新和检索[D]. 上海: 复旦大学, 2003.
- [3] Foster I, Kesselman C, Fsudik G. A Security Architecture for Computational Grids[M]. [S. l.]: IEEE Computer Society, 2002.
- [4] Gamma E. 设计模式: 可复用面向对象软件的基础[M]. 北京: 机械工业出版社, 2000.
- [5] Khun Yee-Fung. XSLT 精要——从 XML 到 HTML[M]. 北京: 清华大学出版社, 2002.

编辑 陈 晖

(上接第 56 页)

的情况下,即所有被监控对象监控周期相同且每次均在同一时刻进行监控操作,当被监控节点数达到 50 时,集中式监控节点的 CPU 占用率峰值达到 97%,内存占用率峰值达到 35 MB。因此,CDOMS 最少可以支持同时监控 50 个数据库服务器的规模。考虑到在实际情况下,各个被监控服务器的监测时刻并不总是集中在一起,因此,CDOMS 实际能够处理的规模要远大于该值。

6 结束语

针对分布式多数据库系统的高可用问题,本文设计并实现了一个面向 Oracle 的集中式监控系统 CDOMS。该系统目前已经投入使用,能够同时对多台 Oracle 数据库服务器进行监控,可以根据实际需要不同的被监控节点定制不同的监控策略,对可能发生问题的故障进行主动报警,能够提供远程处理功能且运行稳定。

参考文献

- [1] 贾 焰,王志英. 分布式数据库技术[M]. 北京: 国防工业出版社, 2000.
- [2] Ernst B, Rasmussen H R, Schwinn U, et al. Enterprise DBA Part 1A: Architecture and Administration[Z]. Oracle Corporation, 1999.
- [3] Quest Corporation. Quest Central for Oracle: Optimize the Oracle Database[EB/OL]. (2008-01-01). http://www.quest.com/quest_central_for_oracle/.
- [4] 蒋凤珍. Oracle 数据库性能优化技术[J]. 计算机工程, 2004, 30 (增刊): 94-96.
- [5] 王 娜,宿红毅,白 琳,等. 数据库性能监控分析系统的设计与实现[J]. 计算机工程, 2005, 31(24): 105-107.
- [6] Cormode G, Garofalakis M. Sketching Streams Through the Net: Distributed Approximate Query Tracking[C]//Proc. of VLDB'05. Trondheim, Norway: [s. n.], 2005.

编辑 顾姣健