

城市规划档案信息系统的架构优化分析与设计

The Optimized Architecture Analysis and Design for Documentary Information System of City Planning

李晓华 (河北省城乡规划设计研究院 石家庄 050021)

摘要:以城市规划档案信息系统为例给出了用组件技术构建信息系统的框架和模板。通过分析二层软件架构风格与多层软件架构风格的优势与不足,使两者相互取长补短,完美地融合在一起,应用于城市规划档案信息系统。对二层软件架构风格和多层软件架构风格进行垂直再分层为表现层、校验层、城市规划档案业务逻辑层、数据存取层、资源层、数据层。对按层分配功能的方案进行了优化分析和设计论证。优化的分层功能分配方案既有利于系统的设计、实现、维护和复用,又使得系统保持了高性能。

关键词:软件架构 分层 EJB 功能分配 组件(构件)

1 引言

在组件体系结构技术出现以前,信息管理系统无论是采用二层软件架构风格,还是采用多层软件架构风格,或采用二层架构与多层架构相结合的风格进行构建,系统的表现逻辑、校验逻辑、业务逻辑、数据存取逻辑和系统所使用的资源通常是柔和在一起的,导致软件的模块性与封装性很差,从而使得系统存在不易于设计、不易于实现、不易于维护、过强地依赖于运行平台、不易于移植、所开发组件的可复用性差、不易于适应计算机技术的更新和业务逻辑的变化等缺陷。

2 城市规划档案信息系统的软件架构优化分析

城市规划档案信息系统的用户分为两大类:一类是档案管理人员,另一类是城市规划人员。

2.1 档案管理人员子系统的软件架构优化分析

城市规划档案管理人员使用系统进行大量档案数据的录入和编辑,以便对档案信息进行统一管理,这就要求档案管理人员端的程序要具有强大、高效的与用户和数据库直接交互的能力,也就是说档案管理人员所面对的子系统应包含具有强大的与用户交互能力的窗口控件和具有强大的与数据库交互能力的窗口组件。档案管理人员端的程序要做到高效、频繁地与用户和数据库进行交互,就应包含表现逻辑、校验逻辑、城市规划档案业务逻辑、档案数据存取逻辑和必要的资源等,即档案管理人员所面对的子系统为“客户端/数据库服务器”二层架构风格。客户端程序只部署在档案管理人员的计算机上,数量不多,即使城市规划档案业务逻辑发生变化也容易维护。

其实,业务逻辑既可以放在客户端,也可以放在数据库上。把业务逻辑放在客户端的优点之一是便于快捷地进行业务处理,优点之二是便于使客户端程序从一种数据库移植到另一种数据库,可移植的原因是程序使用 ODBC 接口屏蔽了不同数据库厂家实现的差异。把业务逻辑放在客户端的缺点是当业务逻辑发生变化时,如果客户端数量众多,使得系统维护不方便。对安全性要求较高的行业来说,把业务逻辑放在客户端的另一个缺点是存在潜在的不安全因素。把业务逻辑放在数据库上的优点是安全和易维护,缺点之一是当数据库服务器面对大规模并行访问时,数据库服务器负载过重,容易成为整个系统的瓶颈,缺点之二是如果业务逻辑是用扩展 SQL 实现的,则当系统从一种数据库移植到另一种数据库时,业务逻辑需要重写,从而导致系统的可移植性降低。对安全性要求较高的应用,如电子金融、电子商务等,可以把最核心、最关键的

业务逻辑放在数据库上,其他一般的业务逻辑可以放在客户端,这样可以平衡系统的负载,提高系统的整体性能。

校验逻辑既可以放在客户端,也可以放在数据库上。数据类型是否正确、数据格式是否符合要求、数据值是否超出范围等校验逻辑可以放在客户端。关键字段的数据值是否为空、是否有重值出现等校验逻辑可以放在数据库上,因为数据库管理系统非常擅长做这些校验工作。

城市规划档案信息系统的二层架构风格子系统的客户端程序用 PowerBuilder 实现,PowerBuilder 是面向对象和面向组件的符合 DCOM 规范的数据库客户端开发工具,它有着独特的数据窗口组件技术,数据窗口可以完成数据库中的数据与程序中的数据的双向映射,也就是说数据窗口组件封装了数据库中的数据与程序中的数据之间的双向映射关系,程序可以透明地使用数据窗口组件与数据库进行交互。数据库采用 MS SQL Server 关系型数据库。

2.2 城市规划人员子系统的软件架构优化分析

城市规划人员使用子系统可以并发地进行档案信息查询和浏览,但无权修改和编辑档案信息。这要求城市规划人员所面对的子系统要具有良好的城市规划档案查询业务逻辑和表现逻辑,而不必具有很强的数据编辑能力,且系统要易于维护。城市规划人员客户端计算机较多,要使城市规划人员所面对的子系统易于维护,就不能把城市规划档案查询业务逻辑、表现逻辑、校验逻辑、档案信息存取逻辑和资源等放在城市规划人员计算机上,这些逻辑也不能全部放在数据库服务器上,而应放在城市规划档案管理应用服务器上。多层架构风格的分布式 J2EE 组件体系结构恰好能满足上述需求,城市规划人员计算机上运行通用的浏览器,表现逻辑和校验逻辑放在 Web 组件上,城市规划档案业务逻辑、档案信息存取逻辑放在 EJB 组件上,资源放在 JNDI 上。只需维护 Web 组件和 EJB 组件,城市规划设计人员客户端保持不变,且表现逻辑、校验逻辑、城市规划档案业务逻辑、档案信息存取逻辑是分开维护的,如果只是表现逻辑或校验逻辑改变,则只需维护 Web 组件,如果只是城市规划档案业务逻辑或档案信息存取逻辑改变,则只需维护 EJB 组件,如果只是资源改变,则只需维护 JNDI 即可。

多层架构子系统与二层架构子系统共享同一个数据库。按 J2EE 规范来构建城市规划档案查询子系统。用 UML 建立城市规划档案查询子系统的模型,使

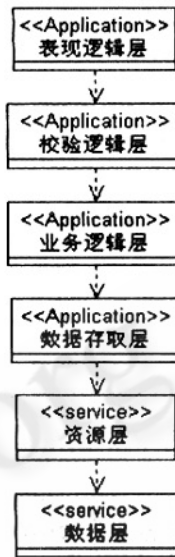


图 1 分层软件架构风格

用 Jbuilder 实现所建模型,用 WebLogic 作为城市规划档案 J2EE 应用服务器(包含 Web Server、EJB Server、JNDI Server、JMS、JTA 等)。

2.3 多层架构风格与二层架构风格的优势互补

二层架构风格与多层架构风格相结合,可以做到取长补短。二层架构风格的优势是客户端能直接与数据库进行交互,能够同时承载表现逻辑、校验逻辑、业务逻辑、数据存取逻辑和资源层等,二层架构客户端程序允许用户进行大批量数据的录入、查询、编辑、修改、删除,具有很强的与用户和数据库直接交互的能力。二层架构风格的缺点是如果客户端数量众多,当业务逻辑、表现逻辑、校验逻辑或数据存取逻辑发生变化时,每一个客户端程序都要进行维护,而且业务逻辑直接放置在客户端也是不安全的。多层架构风格的优点是表现逻辑、校验逻辑、业务逻辑、数据存取逻辑和资源层等放在 Web 组件、EJB 组件、JNDI 上,当表现逻辑、校验逻辑、业务逻辑、数据存取逻辑发生变化时,只需维护相应的 Web 组件和 EJB 组件即可,数量众多的客户端(浏览器)保持不变,多层架构风格的优点正好弥补二层架构风格的缺点。多层架构风格的缺点是客户端(浏览器)不直接与数据库相连,没有直接与数据库通讯的组件,客户端与数据库的通讯是以 Web 组件和 EJB 组件为中转站的,这就使得客户端不便快捷地进行大批量的数据处理,而二层架构风格的优点正好可以弥补多层架构风格的缺点。

3 管理系统的软件架构优化设计

分层软件架构风格功能分配的 UML 类示意图如图 1 所示,其中表现逻辑层的主要成份是 GUI,用于用户与系统的交互和数据的输入输出;校验逻辑层的作用是对用户输入系统的数据进行有效性检验,以保证进入到系统的数据符合约束规则;业务逻辑层主要用于进行业务处理;数据存取层负责程序中的数据与数据库中的数据的双向映射,以便程序访问数据库;资源

层的内容主要是用于共享的资源,如:ODBC、JNDI、JD-BC、等;数据层是城市规划档案数据库。

3.1 档案管理人员子系统的分层软件架构优化设计

图 1 中的前五层功能分布在档案管理人员使用的客户机上,数据层部署在数据库服务器上。

表现逻辑层:主要包括:窗口、菜单、数据窗口控件、按钮等 GUI 组件,负责系统与用户的交互。校验逻辑层:分为两部分,一部分是指客户端的数据窗口控件对用户录入的数据进行检查,例如:检查数据类型是否正确,是否超出了数据范围等,另一部分校验在数据层进行,例如:检查数据是否为空值,是否有重值出现等。不论是哪一部分校验,若数据有错,则在客户端报告给用户,以便修正错误。第一部分校验由数据窗口控件负责向用户报告错误及如何修改,第二部分校验由数据窗口控件负责把数据库服务器传送到客户端的错误信息翻译给用户,且告诉用户如何改正错误。城市规划档案业务逻辑层:内容包含了对数据进行操作的业

需求的地方,需要 PowerBuilder 程序员编写大量的代码,这一层除了要实现城市规划档案业务逻辑外,还要负责数据的持久性管理和事务管理,这一点与 EJB 不同。数据存取层:数据窗口组件负责完成客户端程序中的数据与数据库中的数据的双向映射,档案业务逻辑层借助数据存取层透明地在数据库中存取数据,数据存取层是连接业务逻辑层与数据库的桥梁。资源层:包括二层架构子系统所使用的资源,如 ODBC 等,资源层不是归本系统所专有,其他程序也可共享使用这些资源,所以资源层的可复用程度较高。数据层:包括数据库管理系统和城市规划档案数据库,这一层通常部署在一台单独的机器上作为数据库服务器,与客户机协作实现城市规划档案信息的分布式计算。数据层的主要功能是响应客户机的请求,对数据进行存储、排序、检索等。城市规划档案信息系统的二层架构子系统中的总体规划档案信息管理程序所涉及到的 UML 构件图如图 2 所示。

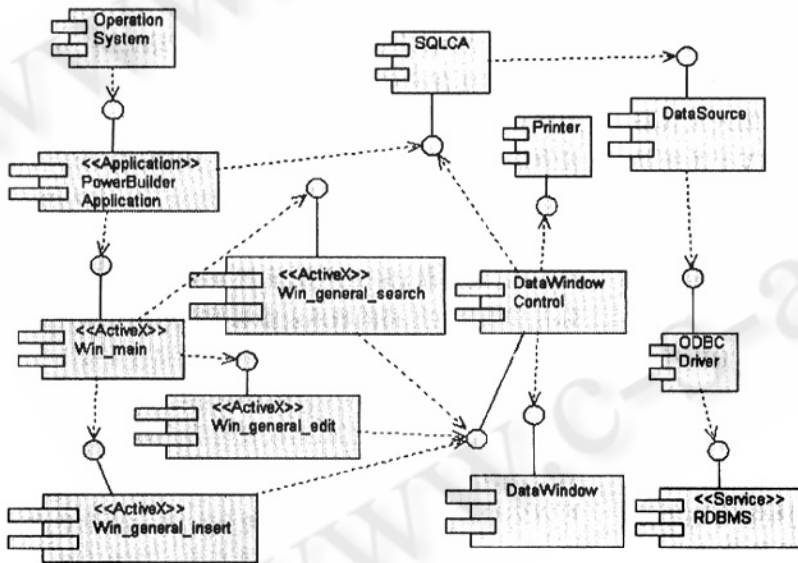


图 2 总体规划档案信息管理程序所涉及到的 UML 构件图

务流程,例如:数据录入、编辑、汇总、统计、借阅、查询、打印等,这部分工作由数据窗口控件、窗口、按钮等构件的消息处理函数完成。PowerBuilder 的消息处理机制的特点是:控件既是消息生产者,同时又是消息监听者(消息消费者),这点与 JAVA 不同。城市规划档案业务逻辑层是系统的核心部分,是满足用户的功能性

详细规划、专项规划、市政工程规划、市政工程设计、建筑设计、可行性研究等档案信息管理程序所涉及到的构件图与总体规划档案信息管理程序所涉及到的构件图相似。

3.2 城市规划人员子系统的分层软件架构优化设计

城市规划人员所面对的多层架构风格子系统把图 1 中的前五层功能部署在城市规划档案信息应用服务器上。

(1) 表现逻辑层。HTML、JSP 和 Servlet 组件负责建立系统与用户交互的界面,允许用户向系统提出请求,并把系统对请求的响应结果反馈给用户,有时也包含 Java Applet,以增强视觉渲染效果。

(2) 校验逻辑层。通常分布在 JSP、Servlet 和 JavaBean 等 Web 组件上,验证用户所提交的数据是否为空值、数据类型是否正确、是否为合法用户等。除了验证工作外,JSP、Servlet 和 JavaBean 还负责把用户所提交的数据传递给 EJB 或数据库,且把 EJB 或数据库中

的数据反馈给用户,即 Web 组件是连接用户与 EJB 或数据库的桥梁。在没有 EJB 时, JSP、Servlet 和 JavaBean 等 Web 组件还具有完成业务逻辑和数据存取逻辑的职能,这时的 Web 组件不但要负责表现逻辑、校验逻辑、业务逻辑、数据存取逻辑,还同时兼有数据持久性管理、访问数据库的事务管理和数据访问的安全性管理等职能,这种情况下的 Web 组件的职责显得很混乱,不利于程序的设计、实现和维护,而且影响系统的运行性能。

(3) 城市规划档案业务逻辑层。业务逻辑可以放在 Web 组件上或 EJB 组件上,但好的做法是把业务逻辑放在 EJB 组件上。通常用会话 EJB 封装业务逻辑,用实体 EJB 建立数据库与应用程序之间的数据的双向映射,会话 EJB 使用实体 EJB 完成业务逻辑,也可以用实体 EJB 同时封装业务逻辑和数据映射逻辑。用 EJB 把业务逻辑和数据存取逻辑从 Web 组件中分离出来, Web 组件负责表现逻辑和校验逻辑,任务边界定义清晰,模块性很好,使 Web 变得简洁,利于 Web 组件的设计、实现和维护。程序员在开发 EJB 时可把主要精力放在业务逻辑的设计与实现上,而数据的持久性、事务和安全性等管理工作交给 EJB 容器完成,可以减轻程序员的负担,分工明晰,有利于 EJB 组件的设计、实现、复用。若业务逻辑有变,则只需修改相应的 EJB 组件,系统的其它部分保持不变,有利于系统的维护。EJB 承载业务逻辑时的城市规划档案信息系统的系统响应速度和吞吐量比把业务逻辑放在 Web 组件上时高,且城市规划档案信息系统所面对的并行访问规模越大,这种系统性能优势就越明显,原因是 EJB 具有独特的生命周期回调机制,而 Web 组件没有,有关系统性能的详细分析将另文论述(高端 J2EE Web 应用的性能分析)。可以把 EJB 组件应用到需要相同功能的其它系统中去,而且系统中的 EJB 组件还可由实现了相同功能的其他 EJB 组件所置换,或有些通用的 EJB 组件可以直接从第三方购买,所以, EJB 组件具有很好的可复用性和灵活性。

(4) 城市规划档案数据存取层。优化的设计是把业务逻辑放在会话 EJB 上,数据存取逻辑放在实体 EJB 上,这样做的好处是数据存取逻辑和业务逻辑可以独立进行设计和维护。

(5) 资源层。主要是指 JNDI, JNDI 负责公共资源

的注册、查找和管理的工作。可向 JNDI 注册的内容很丰富,如: EJB 组件, JMS 连接工厂、数据库连接、事务管理对象等。J2EE 程序可用资源名称从 JNDI 中查找所要使用的资源对象,找到后就可以使用资源,待程序不再使用资源时,就把资源对象归还给 JNDI,以供其他 J2EE 程序使用 JNDI 的共享资源, JNDI 的有效作用范围是整个 J2EE 域。

(6) 城市规划档案数据层。符合 J2EE 规范的多层架构风格子系统与符合 DCOM 规范的二层架构风格子系统共用同一个数据层,即数据库管理系统和城市规划档案信息数据库。

4 结论

城市规划档案信息系统的优化架构采用了分层软件架构风格,所设计系统的特点是:模块性好,每一层可以做成一个或若干个构件;易于维护,只要层的接口不变,任意一层都可以单独维护,不影响其他层;能很好地支持复用,本系统的构件可以被其他系统使用,同时本系统的构件可以被实现了相同接口的其他构件所置换;自上层至下层,层与用户的交互性逐渐下降,而每层内的构件被上层内的构件复用的程度逐渐增强;因为不能越层访问,所以安全性较高。城市规划档案信息系统的架构优化分析与设计具有一定的代表性和普适性,可以作为框架或模板应用于电子商务、电子政务、电子金融等系统的架构分析与设计。

参考文献

- 1 Raphael Malveau, Thomas J. Mowbray. Software Architecture Bootcamp [M], 北京:电子工业出版社, 2003. 19-27.
- 2 Jim Arlow, Ila Neustadt. UML and the Unified Process—Practical Object Oriented Analysis & Design [M], 北京:机械工业出版社, 2003. 185-188.
- 3 Alan W. Brown 著, 赵文耘、张志等译, 大规模基于构件的软件开发 [M], 北京:机械工业出版社, 中信出版社, 2003. 137.
- 4 Marc T. Sewell, Laura M. Sewell 著, 将慧等译, 软件架构师职业导读 [M], 北京:机械工业出版社, 2003. 49-66.