

J2EE 代码自动生成研究^①

Research on Technology of Automatic J2EE Code Generating

盛 刚 韩莉莉 (曲阜师范大学运筹与管理学院 山东日照 276826)

摘要:在开发基于 J2EE 的 B/S 应用系统的过程中,由于客户业务和技术两方面的原因,使开发过程中所做的重复性的工作比较多,并且基本上都是采用复制-粘贴形式的软件复用方式,导致了容易出错、修改复杂、维护困难等许多问题。总结了这些问题之后,提出一种代码自动生成的想法,即根据给定的能够代表客户业务的信息,由程序自动生成 J2EE 应用程序的各部分内容,而程序员所做的工作就是维护这些信息。

关键词:组件 自动生成 软件复用

1 引言

就目前来说,J2EE 技术是一门比较流行的技术,在电子政务、电子商务、网络办公等方面大显身手,而且经过多年的发展,现在的 J2EE 技术也比较成熟,分层技术是 J2EE 最大的优点。在许多解决方案中都采用了 J2EE 技术^{[1][2]}。

尽管 J2EE 有其与生俱来的优点,然而我们在实际开发中仍然发现了许多问题。我们发现,由于客户业务的特点,许多开发工作从技术上讲都是相同的。大部分客户业务都是对某项数据进行简单的增加、修改、删除、浏览等工作,这些业务对应到编程工作中就是对数据库进行插入、修改、删除、查询等操作。这样就导致重复性的工作非常多。程序员在开发工作中,就采用最原始的程序复用技术,不停地复制-粘贴,几乎不用做太大的修改。这种程序复用技术不但工作量大、效率低,而且还很容易出错、不利于以后修改。

从这一点出发,我们就提出了这样一个想法:建立一个 J2EE 代码自动生成系统,根据能够代表客户业务的信息由程序自动生成组成 J2EE 应用系统的各部分代码,以避免以上所提到的问题。其实,软件生产自动化^[3]也是今后软件开发的一种重要的工程方法。

2 必要性

首先,从程序员编码的角度进行分析。在进行系

统开发的过程中,各种文件的每一行、每个字都是程序员手工敲上去的,当然,有很多也是复制-粘贴过来的。工作量大小不必说,还比较容易出错。而且,由于业务之间差异性不大,所以,基本上都是在进行重复劳动,根本没有创造性而言。

其次,从测试人员角度看。

目前,测试人员的工作也是比较繁重的,而且测试这样的系统技术性并不强。不但要对程序进行测试,还要对业务进行测试。对每个 JSP 页面的每个输入框都要输入各种类型的值,以确保该页面进行了有效性校验。这样的问题在应用系统中层出不穷,而且还比较重要,如果没有测试的话,可能会造成输入的数据对具体业务而言为非法数据,也可能引起数据库错误,进而导致业务流程不能继续下去。另外还要对业务进行测试,看整个业务流程是否符合客户要求。

如果使用自动生成系统,业务组件都是由程序自动生成的,如果发现了一处错误,修改生成程序即可,其他类似的错误也随之消除了。这样,测试成本可以大幅度降低。测试人员也可以将主要精力放在对业务流程的测试上。

最后,从企业管理角度看。

现在的面向 J2EE 的三层结构的 B/S 模式架构已经比较成熟了,已经有很多基于此技术的成功产品。技术的成熟,意味着“暴利”阶段一去不复返了,只能

① 基金项目:曲阜师范大学青年科研基金(XJ04057)

经过工艺的改进降低成本,同时提高系统的适应性和质量。还有,就是在一些时间紧、任务急的项目中,如果在传统的开发中只能是增加人员、加班加点。而如果使用自动生成系统,则完全可以避免这种情况。有了自动生成系统,完全可以在短时间能做出合格的软件产品。

因此,通过以上几方面的分析,不论从程序员编码的角度,测试人员测试的角度,还是从企业管理角度都可以看出,现在有必要做出一个自动生成系统来。这样,业务组件生成实现自动化,其他的一些过程可以实现部分自动化,把一部分规律性强、属于重复劳动的工作交给程序完成。自动化的好处不但省时,而且可重复使用,这样,软件开发的效率将极大提高,而且极少会出现错误。

3 可行性

首先,从业务角度来看。对程序员而言,大部分客户业务无非就是对数据库进行简单的增加、修改、删除、查询等四项操作,不论是 JSP 文件,还是 Java 文件,都是基于这四项操作的。几乎数据库中的每一个表都对应增加、修改、删除、查询四个 JSP 文件和一个 Java Bean,这个 Java Bean 中有对应的增加、修改、删除、查询四个方法。

其次,从技术角度看。传统的 J2EE 应用系统所用到的技术主要是 J2EE, J2EE 技术目前已经是较成熟的技术,通过不断的学习和实践,我们对 J2EE 技术的理解已经比较深刻。完全有能力做出一个自动生成系统来。

因此,通过以上两方面的分析,不论从业务角度还是从 J2EE 技术角度都可以看出,现在是完全可以做出一个自动生成系统来的,而且不会有太大的风险。

4 模型

先看一下传统的系统模型图,见图 1。通过图 1 我们可以看出,该系统的数据库、JSP 文件、Java Bean 都是直接以客户业务为基础的,一旦客户业务发生变化,就会直接导致这三方面的变化。也就是说,整个传统的 J2EE 应用系统没有一个统一的入口,系统中的任何一个文件和数据库的表都允许程序员直接修改,而且也只能这样做。

再看现系统模型图,见图 2,可以发现现在客户业务

层之上,又增加了“客户业务表示”一层,在客户业务表示层之上才是原有的三部分。该“客户业务表示”层是对客户业务的某种形式化的表示,完全与客户业务相一致,可以通过程序对其进行操作。而目前的客户业务是通过 Word 文档等非形式化表示实现的,不能通过程序进行处理,只能通过程序员人工处理。这正是新旧系统所表现出来的重要区别。

图 2 所表示的模型正是我们将来在自动生成系统中所采用的模型。增加的这一个客户业务表示层可以说是整个自动生成系统的关键,正是增加了这一层,才会有诸多好处。这也是计算机技术常用的方法,通常增加一个中间层或增加一个中间接口就会解决很多问题。



图 1 传统的系统模型图

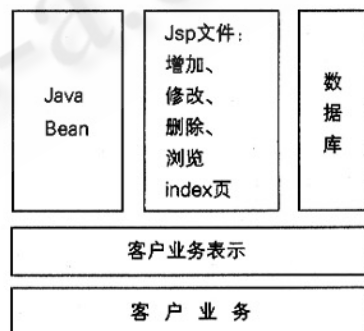


图 2 现系统模型图

这样一来,就给整个应用系统增加了一个统一入口,对系统的任何修改都要通过此处而且也只能修改此处,程序员再也不用挨个文件修改,而且程序员也没有权利对任何一个具体文件进行修改,即使修改了,也不会起任何作用。

5 总体设计

我们要建立的 J2EE 自动生成系统就是以给定信息为基础,由程序自动生成整个 J2EE 应用系统的各部分构件^[4],包括 JSP、Javascript、Java Bean 等,然后再根据具体的客户业务对个别部分进行调整。

根据开发客户系统所做的工作和整个开发的流程,我们把该系统进一步划分为以下几个子系统:

(1) 组件生成子系统。根据代表客户业务的信息生成与该业务相对应的 JSP、Javascript、Java Bean,生成的这几部分都是最小的部分,不能再分的。在本系统中,我们称这些不能再分的部分为组件。

(2) 组件组装子系统。根据由组件生成子系统所生成的各种组件进行组装,形成最后的 J2EE 应用系统。

(3) 组件提交子系统。在整个系统完成或者是对某些业务的修改完成后,负责将生成的或修改过的 J2EE 应用系统的相应组件部署到应用系统中去。

(4) 组件部署子系统。在组件提交后,就要把这些组件部署到应用程序服务器上去,该子系统可以实现组件的异地部署和自动部署。

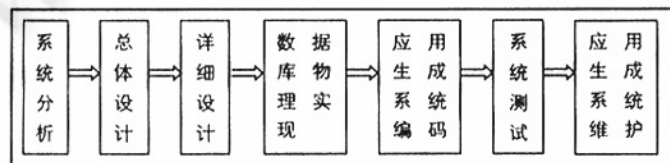


图 3 新的生产流程图

其中,组件生成和组件组装是主要的两个部分,之所以要区分这两部分,主要是考虑到有些客户业务比较复杂,如果不分开的话,那对程序员的要求可能要高一些,尤其是客户业务方面,而且实现起来也比较困难。而分开后,组件生成部分基本上不需要懂业务,只要按照详细设计报告进行操作即可,这部分对操作人员的要求不高;组件组装部分需要懂业务,需要把握业务流程,负责将各个 JSP 根据业务组合起来。还有一个方面就是,考虑到自动生成系统自身的可扩展性。在进行组装时,可能还要加入很多其他内容,比如工作流系统。分开后,可以灵活地根据客户业务的不同而增加一些不同的内容。

本系统在整个软件生产周期中的作用:

6 结束语

本系统包括了一系列的生成器:JSP 生成器、Javascript 生成器、Java Bean 生成器、SQL 语句生成器等等。但是我们的目标决不仅仅局限于一个生成器,一个小程序,我们将通过努力把把这个系统做强、做大,能成为真正的企业级的应用系统,为我们的系统开发提供一条快速高效的通道。该生成系统所具有的优点,可以总结为以下几方面:

第一,从技术角度讲,该生成系统在面向对象技术和组件技术的基础上最大程度地实现了程序复用,使软件开发实现了半自动化,由程序生成程序,是比面向对象技术和组件技术更加有效的一种程序复用方式。

第二,从程序员角度讲,有该生成系统的帮助,我们就能够快速而且高质量地完成开发或修改工作,而不再进行原始的复制-粘贴工作了。运用该系统进行应用系统开发,能够明显地缩短开发周期,尤其是在工期短、任务紧的情况下,更是如此。

现在,由生成系统所生成的应用系统只是简单的 JSP + Java Bean,没有运用特别的架构。而目前有很多成熟的架构,如 Struts^[5],JavaServer, Faces^[6]等,应用得都比较广泛。而生成系统现在还不能生成以这些架构为基础的应用系统。

参考文献

- 姚睿、彭舰、周明康,基于 J2EE 的学习管理系统[J],计算机应用,2003,23(9):118-120。
- 于万钧、刘大有、杨博,基于 J2EE 的 PDM 系统产品生命周期[J],计算机应用,2004,24(4):140-143。
- 杨笑清,软件工程技术发展思索[J],软件学报,2005,16(1):1-7。
- 杨笑清、梅宏、李克勤,软件复用与软件构件技术[J],电子学报,1999,27(2):68-75。
- Apache Software Foundation. Introducing Struts [EB/OL].
<http://struts.apache.org/faqs/kickstart.html>
- Sun Microsystems. J2EE JavaServer Faces [EB/OL].
<http://java.sun.com/j2ee/javaserverfaces/>