

# 基于 J2EE 平台的信息化系统代码生成<sup>①</sup>

黄小锋<sup>1</sup>, 张晶<sup>2</sup>, 王宇飞<sup>2</sup>, 李春阳<sup>3</sup>

<sup>1</sup>(中国水电顾问集团国际工程有限公司, 北京 100101)

<sup>2</sup>(北京中电普华信息技术有限公司, 北京 100192)

<sup>3</sup>(国网信息通信产业集团有限公司, 北京 100031)

**摘要:** 企业信息系统在企业的经营管理中发挥着至关重要的作用, 然而, 在企业信息系统的开发过程中, 存在着开发效率低下、系统架构不一致、开发周期长等问题. 针对上述问题, 设计了一种基于 J2EE 平台的信息化系统代码生成工具, 开发人员只需要进行模型的可视化设计, 由代码生成工具完成代码的生成. 目前该代码生成工具已应用于实际的开发平台中, 实践表明, 代码生成工具的使用可以统一企业信息系统的开发架构, 大幅度减少开发人员编写重复性代码的工作量, 实现设计成果的复用和共享, 提高企业信息系统开发的质量和效率.

**关键词:** 业务模型; 界面场景; 代码生成; 信息化系统; J2EE

## Code Generation Tool Based on J2EE Architecture

HUANG Xiao-Feng<sup>1</sup>, ZHANG Jing<sup>2</sup>, WANG Yu-Fei<sup>2</sup>, LI Chun-Yang<sup>3</sup>

<sup>1</sup>(Hydro China International Engineering Co. Ltd., Beijing 100101, China)

<sup>2</sup>(Beijing China Power Information Technology Co. Ltd., Beijing 100192, China)

<sup>3</sup>(State Grid Information & Telecommunication Industry Group Co. Ltd., Beijing 100031, China)

**Abstract:** Enterprise information system plays a crucial role in business management. But in the development process of enterprise information systems, there are still many issues such as low efficiency, inconsistent system architecture, long development cycle and so on. A code generation tool based on J2EE architecture was designed to solve these problems. With the tool, system developers only need to design models in the visual designer and the tool will complete the code generation. At present, the code generation tool has been applied to the actual software development platform. Practice shows that the use of the code generation tool can unify systems architecture, reduce the development of repetitive code, reuse and share the design, improve the quality and efficiency of enterprise information system development.

**Key words:** business model; interfere scene; code generation; information system; J2EE

随着信息化的不断发展, 基于 B/S 架构的分层体系结构成为企业应用开发的首选. 分层体系结构的应用, 降低了系统的耦合性, 使系统具有良好的可拓展性、可维护性和稳定的系统质量, 同时, 可以提高软件的可重用性, 节省项目的开发时间.

J2EE 架构是目前较为流行的企业信息化系统开发框架, 已经成为基于 Web 的电子商务和企业级应用的标准<sup>[1]</sup>. 基于 J2EE 平台, 涌现出了一批成熟的框架, 如 Struts、Spring、Hibernate 等, 这些框架在不同的层次上对应用开发进行支撑, 降低了企业信息系统的开发难度, 保证了项目的质量. 然而框架的使用也带来

一些问题, 如学习成本高, 配置复杂等, 使用这些框架开发企业应用, 对于不同模块而言, 各层次代码结构有很大的相似性, 开发人员需要编写大量重复的代码, 影响了软件开发速度. 现有的一些工具也只能在显示层和数据访问层为用户减少代码量, 而且生成的显示层和数据访问层代码各自独立, 不是一套完整的代码. 针对这些问题, 本文设计了一个基于 J2EE 平台的信息化系统代码生成工具, 该工具实现了一种基于业务模型和场景的代码生成方法, 开发人员只需要对业务功能进行建模, 对界面场景进行配置, 由代码生成工具完成代码生成, 实现完整的业务功能.

<sup>①</sup> 收稿时间:2015-04-10;收到修改稿时间:2015-06-15

## 1 架构选择及信息系统分析

### 1.1 基于开源框架的 J2EE 架构

J2EE 是一个分层体系结构<sup>[2]</sup>, 一般划分为四个层次, 包括显示层、控制层、业务逻辑层和数据访问层<sup>[3]</sup>. 显示层提供与用户交互的界面; 控制层响应用户请求; 业务逻辑层处理各种业务逻辑和计算; 数据访问层负责存放和管理应用的持久性业务数据.

代码生成工具采用开源框架 Spring 和 Hibernate 作为 J2EE 体系架构的一种轻量级实现. 其中, Spring 负责响应用户请求及整个应用的配置和管理, Hibernate 作为持久层框架, 处理数据库的 CRUD 操作. 分层设计使项目结构清晰, 有利于项目的开发维护. 同时, 代码生成工具根据每层的特点生成代码, 降低了实现的复杂度.

### 1.2 业务模型和场景分析

企业信息系统中业务数据的存储方式主要是关系型数据库<sup>[4,5]</sup>. 在信息系统的设计中, 一张数据表对应着一个业务实体. 业务实体之间的一对一、一对多、多对多等关系转化成数据库表之间的外键关系. 企业信息系统中的业务操作主要是对业务实体的增删改查, 对应数据库中单表或关联表的增删改查操作.

企业信息系统的展现页面具有简洁、易操作的特点, 一般通过图表和树来展现数据. 代码生成工具根据企业信息系统的展现特点, 设计了单表场景、主从表场景、树场景、Tab 场景等一系列界面场景, 并创建了相应的场景模型. 用户通过对场景模型的配置, 实现代码的快速生成.

## 2 代码生成工具设计

### 2.1 总体设计

#### 2.1.1 总体架构设计

界面场景定义系统界面的样式和布局, 业务模型定义业务实体, 通过配置向导将业务实体填充到界面场景来完成一个业务功能的定义. 本代码生成工具对于场景模型库中的场景模型, 可以匹配不同的业务实体实现业务数据的增删改查功能, 开发人员无需编写任何代码. 用户可以自定义界面场景, 对场景模型库进行扩充, 以适配更多类型页面的代码生成.

代码生成工具的总体架构如图 1 所示.

代码生成工具由业务模型编辑器、场景模型库、代码生成配置向导、模型校验器、模型解析引擎、代

码生成引擎、模板库等 7 部分组成.

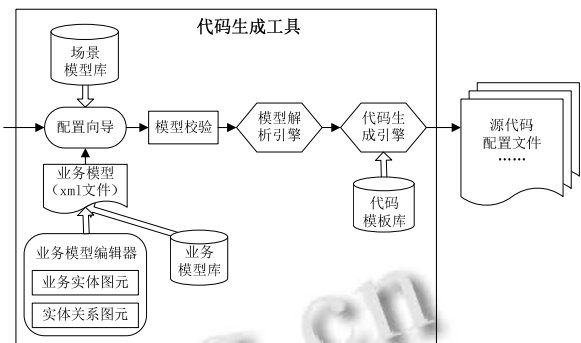


图 1 代码生成架构图

①业务模型编辑器, 通过图形化的操作界面, 完成业务实体和实体关系的创建和修改.

②场景模型库, 内置了单表场景、主从表场景、树场景等常用的界面场景,

③模型校验器用于验证模型及用户的配置是否符合规范.

④模型解析引擎对场景模型和业务模型进行解析和匹配, 为代码生成组装数据.

⑤代码生成引擎根据用户的配置生成前后端可执行代码.

⑥代码生成配置向导用于界面场景和业务实体的选择和配置, 实现了界面场景和业务实体的绑定.

⑦代码模板库用于存放描述场景信息的 XML 文件和前后端代码模板, 模板包含可变部分和不变部分. 不变部分即代码片段, 可变部分由用户在向导中的配置来决定.

#### 2.1.2 代码生成流程

利用代码生成工具生成代码, 主要包括 4 个步骤:

①根据业务需求创建业务模型;

②在代码生成向导中进行场景配置, 包括场景的类型、场景的名称、代码的存放路径等, 并选择场景需要用到业务实体;

③配置界面的展现形式, 即配置业务实体属性展现采用的控件;

④由代码生成工具对业务模型和场景模板进行解析、验证, 生成各个层次的代码.

由代码生成工具生成的代码无需修改, 可以作为一个业务功能的完整实现. 该代码生成工具有良好的可扩展性和灵活性, 用户可根据需要新增模型和代码

模板,从而不断丰富代码生成工具的基础信息库。

## 2.2 业务模型设计

### 2.2.1 业务模型定义

业务模型 OM 是一个二元组,描述为  $OM = \langle \{ BusinessEntity \}, \{ Association \} \rangle$ , 其中,  $BusinessEntity$  为业务实体,  $Association$  为实体关系。

定义 1. 业务实体  $BusinessEntity$  定义为  $BusinessEntity = \langle \{ Property \}, \{ Operation \} \rangle$ , 其中  $Property$  表示实体属性,  $Operation$  表示对实体的业务操作。实体属性信息根据数据库中表的字段进行设计,除了包括字段长度、字段类型等常规属性外,还包括校验和界面显示信息,用于对数据进行验证和配置数据在页面的显示形式。

定义 2. 实体关系  $Association$  定义为  $Association = \langle one2one, one2many, many2one, many2many \rangle$ , 表示实体间一对一、一对多、多对一及多对多的关系。

### 2.2.2 业务模型的生成

代码生成工具提供了可视化建模、业务模型导入和数据库逆向建模三种创建业务模型的方式。代码生成工具提供可视化编辑器,通过拖曳表格和关系连线完成业务模型的创建。业务模型导入指将公共模型库中共享的业务模型导入到代码生成工具中实现模型的重用。数据库逆向建模解析关系数据库中的数据模型,并完成到业务模型的转换。数据库逆向建模的步骤如下:

- ① 读取数据源配置信息,包括数据库连接字符串、用户名和密码,进行数据库连接;
- ② 查询并列出的所有数据库表,由用户选择要导入的表;
- ③ 根据用户选择的表,创建业务实体;
- ④ 查询表的字段名称、类型、长度、是否为空等属性,创建业务实体的属性;
- ⑤ 查询表的外键信息,创建业务实体的关系。

模型导入后,在可视化编辑器中对实体进行编辑,添加属性校验器和属性编辑器。代码生成工具中内置了多种属性校验器,包括长度校验、日期校验、URL 校验、正则表达式校验等,实现运行时服务端数据校验功能。属性编辑器控制数据在页面的显示样式,如多行文本、密码框、下拉框,多选列表等。

代码生成工具提供了业务模型库来保存业务实体,实现对业务实体的复用。在业务模型库中内置了组

织、人员、角色等企业信息系统通用的业务实体,简化了系统设计。随着代码生成工具的不断使用,业务模型库的规模将不断扩大,模型的复用率和代码的开发速率将会得到提升。

### 2.3 项目代码结构

为简化代码生成,使项目结构清晰、易于管理和维护,代码生成工具对生成代码的结构和文件命名进行了规范性定义,如表 1 所示。

表 1 代码结构及命名规范

层次	文件及命名
显示层	index.jsp、weblet.js、XXXView.js、XXXViewController.js
控制层	XXXController
业务逻辑层	IXXXBizc、XXXBizc
数据访问层	IXXXDao、XXXDao、POJO、Xml

显示层主要由 JSP 文件和 JS 文件组成。为了解决大规模 JS 脚本难以组织和维护的问题,在显示层引入了 MVC 开发模式,一个功能模块的显示层代码被封装为一个 Weblet 小程序,Weblet 小程序是 Web 应用的最小逻辑单元,它将页面文件、脚本文件和资源文件按照一定的目录结构管理。表 2 展示了工程管理模块的 Weblet 结构。

表 2 工程管理的 Weblet 结构

```

projectManager
  -resource
    -css
    -images
  -scripts
  -weblet.js
  -views
    -MainView.js
    -MainViewController.js
    -DetailView.js
    -DetailViewController.js
  -index.jsp
    
```

每个 Weblet 有一个 index.jsp 文件作为入口,页面的定义和交互采用 JS。在视图 View 中渲染组件,在视图控制器 Controller 中编写组件交互行为,通过 RESTFUL 风格的服务和控制层交互。组件定义与交互行为的分离简化了前端模板的复杂度,提高了代码生成的效率。

在业务逻辑层,抽取出通用的操作逻辑,如 hibernateDao 注入,增删改查等形成基类

BizCDefaultImpl, 在基类中采用泛型的方式注入实体类, 并预留缺省增删改查操作的扩展事件. 代码生成时, 各功能模块的 Bizc 继承基类, 在很大程度上简化了业务逻辑层模板的定义, 优化了代码结构. BizCDefaultImpl 的代码结构如表 3 所示.

表 3 BizCDefaultImpl 代码结构

```
public abstract class BizCDefaultImpl<T, PK extends Serializable>
implements IBizC<T, PK> {
    protected Class<T> persistentClass;
    //hibernateDao注入
    @Autowired
    protected IHibernateDao hibernateDao;
    @SuppressWarnings("unchecked")
    public BizCDefaultImpl() {
        this.persistentClass = (Class<T>)
ReflectionUtils.getSuperClassGenericType(getClass(), 0);
    }
    //新增方法
    final public T add(T be) {
        T result = null;
        if (beforeAdd(be)) {
            hibernateDao.saveOrUpdateObject(be);
            afterAdd(be);
        }
        return result;
    }
    .....
    /**缺省操作的扩展事件*/
    protected abstract boolean beforeAdd(T be);
    protected abstract void afterAdd(T be);
    .....
}
```

### 2.4 代码修改

代码生成由代码生成引擎根据业务模型来生成代码. 当业务功能需要调整时, 用户只需要修改业务模型, 由代码生成引擎完成对代码的重构. 代码生成引擎对用户扩展的内容进行标识和记录, 形成操作队列, 当用户修改完模型进行代码生成时, 代码生成引擎自动将队列中的用户扩展应用到模型中, 完成新的代码生成, 不会破坏用户对原模型的自定义扩展. 图 2 展示了业务模型修改时的代码重构过程.

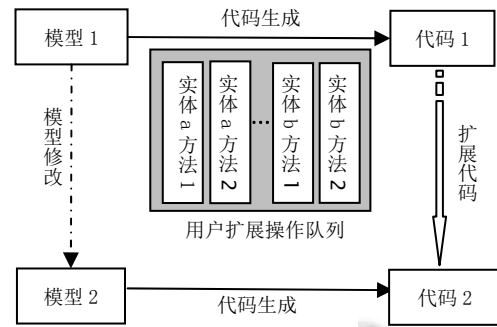


图 2 业务模型修改代码重构图

## 3 代码生成工具实现

### 3.1 利于模板生成代码

代码生成工具采用模板技术, 模板将代码中不变的框架结构、处理逻辑与可变的数据相分离, 实现代码的动态生成<sup>[6]</sup>. 相对于其它模板技术, Freemarker 具有功能强大、报错信息友好、内置大量常用功能、宏定义方便等优点<sup>[7]</sup>, 业务逻辑层代码生成使用 Freemarker 模板.

使用 Freemarker 生成 Bizc 的步骤如表 4 所示.

表 4 BizC 代码生成

```
Configuration conf = new Configuration();
conf.setClassicCompatible(true);
conf.setClassForTemplateLoading(this.getClass(), "/template");
conf.setObjectWrapper(new DefaultObjectWrapper());
bizcTemplate = conf.getTemplate("Bizc.ftl", "UTF-8");
String filePath = bizcs.get(i).getBizc() + "Bizc.java";
Retainer r = new Retainer(filePath);
r.readOldFileContents();
Writer out = new BufferedWriter(new FileWriter(new File(filePath)));
bizcTemplate.process(root, out);
```

### 3.2 POJO 及 XML 生成

持久层采用 hibernate 作为 O/R mapping 框架, 为了简化代码生成处理逻辑及复杂的模板设计, 在 POJO 和 XML 配置文件生成时, 使用了 Hibernate Tools 工具. Hibernate Tools 是 JBoss 推出的一个 Eclipse 综合开发工具插件, 该插件提供了映射文件编辑器、Hibernate 控制台、Hibernate 配置文件创建导航以及代码生成器等工具来简化 Hibernate 开发<sup>[8]</sup>. 该插件提供的代码生成功能是一个独立的可视化向导, 不能直接集成到代码生成工具中, 因此在使用时, 提取工具的核心包 hibernate-tools.jar, 通过源码调用的方式实现代码生成. 代码生成首先将业务模型中的业务实体和关

系转换成 Hibernate 中的表及关系，然后调用 org.hibernate.tool.hbm2x.POJOExporter 和 org.hibernate.tool.hbm2x.HibernateMappingExporter 完成 POJO 和 XML 配置文件的生成。

### 3.3 业务模型设计器实现

业务模型设计器，基于 GEF 图形编辑框架开发，以插件方式集成到 Eclipse 平台。GEF 允许开发人员以图形化的方式展示和编辑模型，采用 MVC(Model-View-Control)模式解除用户界面、行为和表示之间的耦合，具有良好的扩展性<sup>[9]</sup>。图3展示了业务实体的 GEF 实现类图。

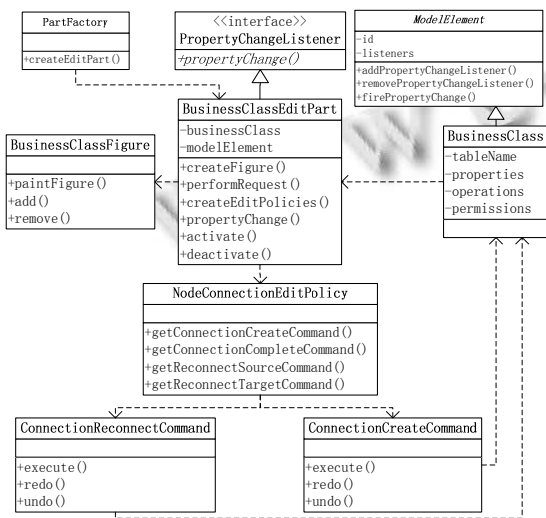


图3 业务实体的 GEF 类图

其中，BusinessClassFigure 是业务实体的视图 (View)，继承了 org.eclipse.draw2d.Figure 类，使用 Draw2d 图形库来绘制图形，视图的显示用 AbstractEditPartViewer，它负责将视图安装到编辑器。

BusinessClass 是业务实体模型(Model)，是一个简单的可持久化的实体，通过 PropertyChangeSupport 类型的成员变量 listener 触发事件，来通知控制器 BusinessClassEditPart 模型的变化。

BusinessClassEditPart 控制器监听模型的变化，根据变化修改视图，同时，当用户编辑视图时，把编辑的结果反映到对应得模型中。GEF 通过请求编辑策略与命令模式来分担控制器的负担，编辑策略用于解析用户编辑请求，命令用于修改模型。NodeConnectionEditPolicy 编辑策略用来处理业务实体关系连线的创建、删除及重定位等请求，当编辑策略

接收到请求后，分析请求信息，根据分析结果选择创建 ConnectionCreateCommand 命令或 ConnectionReconnectCommand 命令，完成连线的创建重练等功能。

控制器的生成使用工厂模式，一个模型对应一个控制器，PartFactory 工厂类负责根据模型创建与之对应的控制器。

### 4 代码生成工具应用

下面以企业信息系统中常用的部门人员管理为例，介绍代码生成工具的使用方法。部门人员管理包含两个业务实体：部门和人员，部门和人员之间的关系是一对多的关系。部门信息包括 id、部门名称、部门类型，人员信息包括 id、姓名、电话、入职日期、所属部门。

①通过拖曳方式在业务模型设计器中完成业务模型的创建，代码生成工具自动为每个业务实体添加通用的增删改查等方法，简化用户操作。创建的业务模型如图4所示。

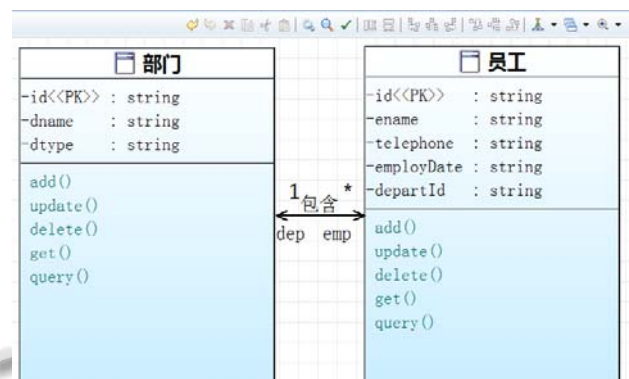


图4 业务模型设计图

②代码生成工具根据业务模型生成后端代码及数据库脚本。

③在代码生成配置向导中选择界面场景，本例采用“主从表”场景，为界面场景配置业务模型作为数据源，代码生成工具根据场景配置生成前端代码。

代码生成工具已集成到国家电网统一应用开发平台中，从2013年至今，共有87个项目使用代码生成工具进行开发，包括基建管控、统一电力市场、电力生产管理 PMS2.0 等百万级代码行项目。目前基建管控项目和统一电力市场项目已经上线试运行。由代码生成工具自动生成代码行数与系统总代码行数对比如表

5 所示.

表 5 自动生成代码行与总代码行对比

信息系统	自动生成代码行	总代码行
基建管控项目	125 万	150 万
统一电力市场项目	260 万	300 万
电力生成管理 PMS2.0	680 万	800 万

实践证明,使用代码生成器能够减少开发人员 80%以上的编码量,提高开发效率.

## 5 结语

本文介绍了一种基于 J2EE 平台的信息系统代码生成工具的设计和实现,该工具提供了可视化的业务建模工具和场景配置向导,开发人员只需要对业务逻辑进行分析建模,对界面场景进行配置,由代码生成工具实现代码生成.实践证明,使用代码生成工具能够大幅度减少开发人员的编码量,提高开发效率;统一开发架构,规范代码结构,提高代码质量.

### 参考文献

1 Berry CA. J2EE Design Pattern Applied. USA Wrox Press,

2003: 14-20.

- 徐长梅.基于数据驱动操作模式的 J2EE 应用自动化生成方法.长沙大学学报,2006,20(5):1-2.
- 谢鹏.基于 J2EE 架构的远程教学平台的研究与应用[硕士学位论文].西安:西安科技大学,2008.
- 张静,孔芳,杨季文.一个数据模型驱动的代码生成工具的设计与实现.计算机应用与软件,2010,23(11):151-153.
- Jorges S, Margaria T, Steffen B. Assuring property conformance of code generators via model checking. Formal Aspects of Computing, 2011, 23(5): 589-606.
- 刘于新,姚凯学,许道云.基于模板的.Net 三层架构的代码自动生成.计算机技术与发展,2012,22(8),13-16.
- 刘铁行,及俊川,任玉平.基于 SSH2 与 Apache Shiro 整合的代码生成器的研究.科研信息化技术与应用,2013,4(4):82-88.
- 陈天河等编著.Hibernate 项目开发宝典.北京:电子工业出版社,2006.
- 张鹏,姜昊,许力.Eclipse 插件开发学习笔记.北京:电子工业出版社,2008.