

# 基于领域工程的构件化可扩展管理信息系统<sup>①</sup>

周锦程<sup>1</sup> 王丹<sup>1</sup> 余泉<sup>1</sup> 夏开建<sup>2</sup> 洪华军<sup>3</sup> (1. 黔南民族师范学院 数学系 贵州 都匀 558000; 2. 常熟理工学院 计算机科学与工程学院 江苏 苏州 215500; 3. 中国船舶科学研究中心 江苏 无锡 214082)

**摘要:** 为使管理信息系统(MIS)适应不断变化的用户需求,在基于领域工程的软件开发方法和构件技术基础上,对 MIS 的领域分析、领域设计和领域构件的实现进行详细研究;并在基于 Struts、Spring 和 Hibernate 三种组合框架支持下,利用领域模型的分层结构思想,提出了一种具有数据抽象与细粒化、松耦合层次风格相结合的软件体系结构,在此结构上结合构件技术和代码自动生成技术解决了 MIS 因环境或服务变化引起的系统需求变化,为 MIS 及相关系统因需求变化引起的领域对象及相关业务操作变化提供了一种合理的技术解决方案。

**关键词:** 可扩展; 管理信息系统; 领域工程; 软件体系结构; 代码自动生成技术

## Extensible Management Information System Based on Domain Engineering and Component

ZHOU Jin-Cheng<sup>1</sup>, WANG Dan<sup>1</sup>, YU Quan<sup>1</sup>, XIA Kai-Jian<sup>2</sup>, HONG Hua-Jun<sup>3</sup>

(1. Department of Mathematics, Qiannan Normal College for Nationalities, Duyun 558000, China; 2.

Computer Science and Engineering College, Changshu Institute of Technology, Suzhou 215500, China; 3.

China Ship Scientific Research Center, Wuxi 214082, China)

**Abstract:** In order to meet MIS's changing user-requests, based on the domain engineering software development methods and component technology, we deeply study about the MIS's domain analysis, domain design and the implementation of the domain components; Based on three kinds of composite frame Struts, Spring, and Hibernate support, use the idea of domain model's stratification structure, proposed a kind of software architecture which has the data abstraction and the sliming and coupled loosely level style, In this architecture use component technology and the code automatic production technology to solve the requirements change of information system caused by the environment and the service change, provide a reasonable technology solutions for MIS and related systems due to the changes in requirements caused by domain object and the associated changes in business operations.

**Keywords:** expansion; MIS; domain engineering; software architecture; automatic code generation technology

近年来,信息技术的快速发展加快了企业信息化的进程。在企业信息化进程中, MIS 的开发普遍存在效率低、工作进程难度难以控制、周期长、成本高、可维护性和适应性差,不能很好地满足用户频繁的业务变化需求等问题。在企业级业务应用的 MIS 的使用

过程中,因环境和服务的变化引起的 MIS 局限性特别明显,客户往往要求在不修改系统源代码的情况下, MIS 能对业务需求的扩展或变动提供支持,而不需要增加企业重复开发的负担。这就要求 MIS 本身能够实现快速的扩展、响应和适应用户业务需求的变化。

<sup>①</sup> 收稿时间:2010-04-14;收到修改稿时间:2010-05-19

## 1 相关技术

### 1.1 于构件的软件开发

集软件复用、分布式对象计算、CASE 和企业级应用程序开发等技术于一体的基于构件的软件开发(CBD, Component Based Software Development)是以软件构架为组装蓝图,以可复用软件构件为组装模块,支持组装式复用,是提高软件生产效率和软件产品质量的有效途径<sup>[1]</sup>。构件技术是面向对象技术的发展,只要遵循构件技术的规范,就可以用自己方便的语言去实现可复用的构件,用户可以在这些构件的基础上,根据需要编制自己的应用程序。

### 1.2 域工程

领域工程是识别应用系统的共同特征和可变特征,对这些特征进行抽象,形成领域分析模型,依据领域分析模型产生领域中一类应用系统共同具有的框架,即特定领域的软件构件,并以此为基础,识别、开发和组织可复用构件<sup>[2]</sup>。一方面,领域工程产生的可复用信息来源于领域中现有的系统,体现了领域中系统的本质需求;另一方面,领域工程将关于领域的知识转化为领域中系统共同的规约、设计和构架,使得可复用信息的范围扩大到抽象级别较高的分析和设计阶段。因此领域工程为软件开发提供更广泛的可复用性。领域工程可分为以下三个主要阶段:领域分析、领域设计和领域实现<sup>[3]</sup>。

(1) 领域分析:该阶段的主要目标是获得领域模型。通过分析领域中系统的需求,识别领域中被广泛共享的共性及可变的需求,从而建立领域模型。

(2) 领域设计:该阶段的目标是获得特定领域的软件体系结构。软件体系结构是对领域模型表示的需求给出解决方案,它不是单个系统的表示,而是能够适应领域中多个系统需求的一个高层次的设计。由于领域模型中的领域需求具有一定的变化性,软件体系结构也要相应地具有变化性。

(3) 领域实现:该阶段根据领域模型和软件体系结构,实现领域构架和可重用构件。

### 1.3 件体系结构

作为控制软件复杂性、提高软件系统质量、支持软件开发和复用的重要手段之一,软件体系结构自提出以来,日益受到软件研究者和实践者的关注,并发展成为软件工程的一个重要的研究领域<sup>[4]</sup>。在软件开发中,认识和理解体系结构可以使系统的高层次关系

得到全面表达和深刻理解。在体系结构全局思想的指导下任何系统的维护、扩充和升级,不会因修改和扩充而破坏整体的完整和一致性。

## 2 MIS领域对象特征分析

### 2.1 领域分析

领域分析的目的是支持软件重用,产生描述特定领域中各系统共性的领域模型。领域模型描述共性需求的同时,要表现领域内应用系统的需求变化性。其变化性主要有两方面:一是需求的可选性,指同一领域中,不同的应用系统在需求上的差异;二是需求自身的变化性,指应用系统中的需求因封装了不同细节,在不同情况下表现不同属性或行为。

对大量 MIS 分析后发现, MIS 基本上是以数据为中心的,其主要功能都围绕着数据进行,如数据定义、数据录入、数据处理、数据检索、统计分析、报表、数据传输等。各类业务逻辑组成有如下特征:

(1) 数据的组织方式有固定模式:在 O-R 模型中,无论多么复杂的业务逻辑都可以由实体之间一对一、一对多、多对多三种关系来表示, MIS 中任意一个功能模块所操作的领域对象值是关系数据库中一张或多张表中的数据,因此数据的组织方式有一定规律可循。

(2) 业务逻辑=领域对象+领域服务:无论多么复杂的业务逻辑都可描述成对特定领域对象组织特定的领域服务的过程。

(3) 对领域对象值模式有规律可循:一些操作无论何种业务逻辑都会用到,如记录操作:创建、插入、修改、删除、索引、检索;数据库操作:查询、数据定义、数据操纵、数据控制、完整性约束、事务处理等。

### 2.2 基于领域模型的 MIS 体系结构设计

在应用程序中,对于每个活动中涉及到的技术和逻辑,程序必须保持简单,否则就会很难理解,因此,将一个复杂的程序切分成层,开发每一层中内聚的设计以提供层的低耦合。在项目开发中,对框架的选择很重要,使用成熟的框架,会减少重复开发的工作量、缩短开发时间、降低开发成本,增强程序的可维护型和扩展性。对于多变的需求,就要引入灵活多变的架构,领域设计要根据领域模型派生出满足领域需求的具有高度的灵活性、鲁棒性和可扩展性,能快速响应

需求变化的软件体系结构。结合 MIS 的领域需求和 CBD 技术以及当前 J2EE 技术下优秀的轻量级开源组合框架 SSH(Struts+Spring+Hibernate)，将一个完整构件纵向分解为多个层次的更细粒度的构件，并选择 Struts 来架构构件表示层与控制层；用 Spring 来架构构件业务逻辑层，处理业务的控制及事务管理；用 Hibernate 架构构件数据访问层和领域层，通过 O-R Mapping(对象关系映射)将领域对象同数据库表对应，各层具体说明如下：

(1) 表示层：提供用户界面的显示和与用户所进行的交互操作。包括管理用户的请求和响应；提供一个控制器负责调用业务逻辑的处理方法以及其它处理；组装一个可以被呈现在视图(View)中的模型(Model)；执行表示层的校验等。

(2) 控制层：Struts 的大部分功能工作在此层，它将用户输入的表单数据包装成 Form-Bean，根据用户请求的路径，通过 ActionServlet 检查用户请求是否合法，若合法就将 Form-Bean 传递到相应的 Action，相应的 Action 再调用业务层构件的处理接口，完成业务操作后返回结果给表示层构件。

(3) 业务逻辑层：进行软件核心业务逻辑的处理。包括处理业务逻辑和进行业务判断；事务管理；通过使用接口和其它层构件交互；管理业务层构件对象的依赖性，实现一个业务的上下文，以便于表示层构件可以调用业务层构件的服务等。

(4) 数据访问层：实现数据的插入、删除、更新和查询等。

(5) 领域模型层：用于保存持久化数据，降低业务逻辑层的复杂度，使其可以只关心业务处理，而相应数据的持久化操作则交给领域模型层来进行。这样实现了业务逻辑与数据的持久化操作的分离，减轻了业务逻辑开发的工作量和复杂程度。

(6) 数据库层：该层是系统的底层，是架构在各类网络通信硬件和软件之上的，是系统的后台数据库，存放系统的业务数据以及系统的支持数据。可以采用各种 RDBMS。

基于构件技术的分层体系结构使整个系统处于有效的松耦合状态，便于开发的构件细粒化，能有效提高代码质量。将这些构件通过框架无缝地整合起来，应用到项目开发中。这种架构模式将领域对象所涉及的操作分层，实现有效的解耦，松散耦合的层次关系

也便于利用代码生成技术对每一层生成相应的代码。系统体系结构如图 1 所示。

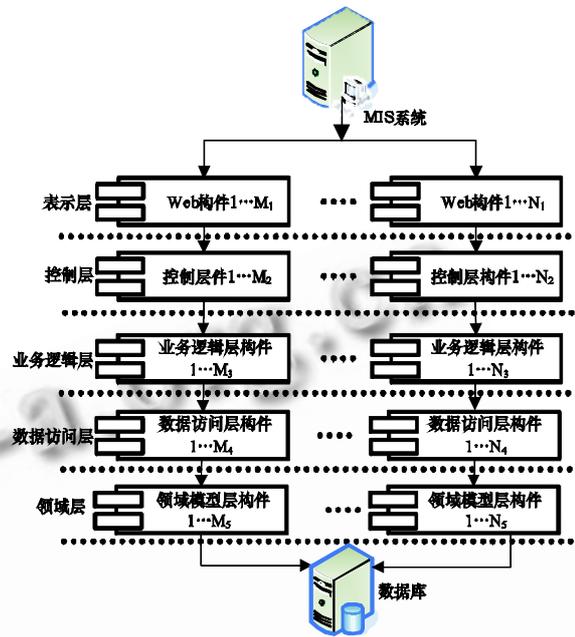


图 1 基于构件技术的系统分层体系结构

### 2.3 领域实现

领域实现的主要目标是依据领域模型开发 MIS 领域特定的构件与构架，实现用户自行扩展 MIS 的新功能，而不需要额外开发成本。完成以上功能须在应用中把建立领域模型的任务交给用户，让其根据系统实现支持自定义领域模型属性的可扩展领域模型实体。主要包括两个方面的工作，即领域对象的描述和由描述自动化生成相应领域的实体对象及业务功能。

#### 2.3.1 建立可扩展领域对象

每个领域对象主要包括以下部分：名称、说明、属性、联系等信息。其中领域对象属性不仅包括了名称、说明、类型，长度、是否允许为空等基本特征，还包括了在信息显示列表中是否显示该属性，显示该属性的顺序，以及该属性与其它表的关联情况等信息。图 2 为新增领域对象界面构件，用户可通过“增加字段”按钮来增加领域对象的属性，通过“删除”按钮来减少领域对象的属性，通过选择关联表中的属性来实现领域对象之间的联系等。

领域对象模型建立后，我们可以向领域对象模型中插入相应的数据供用户使用，图 3 为领域对象列表显示主界面构件。



图 2 添加领域对象“通用件”界面构件



图 3 领域对象列表显示主界面构件

### 2.3.2 键技术

| 领域对象属性表      |                   | 领域对象描述表 |              |
|--------------|-------------------|---------|--------------|
| 序号           | NUMBER(19,0) <pk> | 序号      | NUMBER(19,0) |
| 属性名          | VARCHAR2(30)      | 领域对象名   | VARCHAR2(30) |
| 属性描述         | VARCHAR2(60)      | 领域对象信息  | VARCHAR2(30) |
| 属性类型         | VARCHAR2(30)      | 标志      | INT(1)       |
| 所属领域对象编号     | NUMBER(19,0) <fk> |         |              |
| 是否是文件        | SMALLINT          |         |              |
| 是否为空         | SMALLINT          |         |              |
| 是否在显示列表界面中显示 | SMALLINT          |         |              |
| 显示顺序         | SMALLINT          |         |              |
| 是否唯一         | SMALLINT          |         |              |
| 关联领域对象       | VARCHAR2(30)      |         |              |
| 关联属性         | NUMBER(19,0)      |         |              |

图 4 领域模型在关系数据库中的表示

领域模型在关系数据库中可用如图 4 所示的领域对象描述表和领域对象属性描述表来表示。其中，领

域模型描述表中的“标志”用来标识该表是否是用户自扩展的领域对象，领域对象属性表中“是否是文件”用来描述该属性是否用文件附件的形式表示，“关联领域对象”和“关联属性”用来描述该是否与其他领域对象中的属性相关联。

各层构件的实现将采用 J2EE 中 SSH 框架的优势并结合代码生成技术实现，使系统结构清晰，降低开发的复杂性，提高开发的效率。

(1) 表示层构件。主要用于显示数据，负责处理用户界面事件及与外部系统间的数据接口，实现用户界面的功能。如“添加、更新领域对象界面构件”，“领域对象显示列表主界面构件”，“领域对象详细信息查看界面构件”等。采用 JSP 页面和 Struts 标签库、CSS、JavaScript 等技术来完成构件的实现，根据客户创建的领域对象的信息来生成符合客户要求的页面，通过控制构件实现与低层次的业务逻辑构件的连接组装。

(2) 控制构件，也叫控制连接件，定义控制逻辑。控制构件根据需要来调用业务逻辑构件完成各种处理方法，并将结果返回相应的表示层构件显示。

(3) 在业务逻辑构件中，根据需求的定义制定好各种业务逻辑，并调用相关的数据访问构件完成数据库操作。此层包括了数据信息查询、修改、删除、添加、批量导入导出等功能中的相关业务逻辑构件。为 MIS 提供业务对象、业务操作及业务流程的定义及处理功能。下面以“保存领域对象的业务逻辑构件”和“保存领域对象中输入数据构件的业务逻辑构件”为例来说明业务逻辑层构件的实现方法。

实例 1 用户在添加领域对象界面构件中单击“保存”按钮后触发控制层构件并转到相应的保存领域对象的业务逻辑构件中，该业务逻辑构件所做的工作主要算法流程。

Step1: 判断数据库中是否已有该领域模型存在(在领域对象表中判断)，如果不存在，则执行 Step2，否则返回到添加领域对象界面。

Step2: 向领域对象表中添加领域对象名，领域对象信息及是否为用户扩展的领域对象的标志值；向领域对象属性表中添加领域对象的组成属性及相关信息，执行 Step3；

Step3: 自动生成该领域对象所对应的领域层构件，如领域模型实体类 (.java 文件，该文件含该领域

模型中的属性声明和相应的 `get` 与 `set` 方法)并自动编译成 `.class` 文件, 自动生成该领域对象实体中 `Hibernate` 所对应的 `.hbm.xml` 文件构件; 将新增加的领域对象和领域层的业务对象加入到 `Spring` 的配置文件 `applicationContext.xml` 中, 同时通过 `.hbm.xml` 文件和 `applicationContext.xml` 文件在数据库中自动创建该领域模型所对应的表; 自动修改 `Sturts` 中的对应动态表单(`.java` 文件, 含属性声明和属性所对应的 `get` 与 `set` 方法)并自动编译。

实例 2 保存领域对象中输入数据构件的业务逻辑构件的主要工作流程。

通过例 1 中 `Step3` 自动生成的动态表单类, 从中得到领域对象对应的属性名, 属性类型, 属性所对应的 `get` 与 `set` 方法, 读取添加领域对象输入数据页面构件中的数据(借助对应动态表单), 通过领域模型实体类找到该领域对象所涉及的数据库中的表, 从领域对象属性描述表中读取某属性的类型并对特殊类型如大文本, 文件, 图片等做相应处理, 调用数据访问构件中的保存数据方法将数据入库。

(4) 数据访问构件, 实现操作数据库的功能, 为高层次的业务逻辑构件提供服务。主要包括领域对象描述和领域对象属性描述中所涉及的与数据库访问相关的接口及其实现。

(5) 领域层构件。此层主要由所扩展的领域模型的实体域类组成, 它是与关系数据库直接发生关系的层, 将领域实体同关系数据库联系起来, 主要包括领域对象描述和领域对象属性描述两个实体类及对应的 `.hbm.xml` 文件。

(6) 数据库层。本系统用 `Oracle` 关系数据库来支持数据存储。

我们将各个不同层次不同功能的原子构件通过不同的连接方式组装成大粒度的复合构件, 再通过复合构件集成最终完成整个 `MIS` 系统的开发。通过以上分析, `MIS` 的体系结构层次清晰, 各层构件间具有高内聚、低耦合特性和良好的可复用性。

### 3 结束语

科学合理的软件体系结构与基于特定领域的构件复用能够显著提高软件开发效率与软件质量。本文所描述的可扩展 `MIS` 系统的实现方法在某领域的军用 `MIS` 的开发中得到了实践, 经过实践检验, 该系统具有以下优势: (1) 用户可根据自己需求快速扩展系统的功能; (2) 系统结构清晰, 便于系统的维护管理; (3) 利用代码生产技术可以减少开发人员的工作量, 减少维护周期, 增加系统使用周期, 并维护了代码风格的一致性与用户界面的一致性。

#### 参考文献

- 1 梅宏, 陈锋, 冯耀东, 杨杰. ABC: 基于体系结构、面向构件的软件开发方法. 软件学报, 2003, 14(4): 721 - 732.
- 2 Prieto-Diaz, Domain R. analysis: an introduction. Software Engineering Notes, 1990, 15(2): 47 - 54.
- 3 王千祥, 吴琼, 李克勤, 等. 一种面向对象的领域工程方法. 软件学报, 2002, 13(10): 1977 - 1983.
- 4 梅宏, 申峻嵘. 软件体系结构研究进展. 软件学报, 2006, 17(6): 1257 - 1275.