构件化 MIS 系统通用开发框架研究①

Common MIS Development Framework Based on Components

李成严 左修玉 (哈尔滨理工大学 计算机学院 黑龙江 哈尔滨 150080)

摘 要: 传统开发框架存在维护性差、重用度低等问题。本文在分析信息系统开发中存在的共通业务后，提出了基于软构件技术的 MIS 系统通用开发框架。应用聚类分析方法对构件进行了识别，并对构件按通用性进行了层次划分。通过软总线将各构件接口连接组成开发框架。利用所提出的通用开发框架构建了一个简单的系统，证明了该框架具有耦合度低、可扩展性强等特点。

关键词: 软构件 管理信息系统 开发框架

在管理信息系统(management information system, MIS)的开发、维护过程中，存在大量的共通之处，如：数据库访问接口、系统权限管理、查询窗口、报表等。如何将这些共性问题抽象出来作为 MIS 系统开发的通用框架，这一问题得到了业内人士的广泛关注。Grant[1]提出重用框架可以提高软件质量和生产效率，并且探讨了如何利用设计模式通过 UML 来设计通用框架；Maya 等[2]针对 MIS 系统的开发提出通过层次化开发、重用通用框架来提高系统的灵活性；David[3] 定义了应用程序构造箱(application buildbox)的概念以及针对部署事务操作提出了标志事务系统(Labeled Transition System,LTS)，并在此基础上建立了通用的构件部署框架。这些框架的研究与应用的确提高了软件的重用度，然而这类框架各部分之间由于耦合度较高，从而降低了框架的可维护性。

1 引言

软构件技术作为解决软件危机、提高软件生产率和质量的一种现实可行的途径，是近些年来研究的热点。构件技术往往着眼于系统上层业务的开发，但系统通用框架应该更多关注于底层业务逻辑的处理。若将构件技术引入到系统通用框架的研究中，则会有助于解决上述框架所遇到的问题。

文献[4]中提出了基于元模型(metamodel)来构建面向对象领域的开发框架。Kostadin[5]基于通用组件架构(Common Component Architecture,CCA)探讨了支持分布式构件的框架。但上述文中并没有结合具体的业务领域，使得框架实用性不强。

目前国际上流行的构件技术有国际组织 OMC 的 CORBA，SUN 公司的 EJB 技术还有微软公司的 COM/DCOM/.NET 技术。而.net 技术凭借其提供的统一开发平台和对 Windows 组件的良好集成，在实际的企业级开发中得到了广泛的应用。基于此，本文旨在运用软构件这一成熟技术，面向 MIS 系统开发这一特定领域，实现基于.net 平台的 MIS 系统开发的通用框架。

本文结构如下：在第 1 部分，针对企业信息系统开发这一领域，在分析其中涉及的基本业务流程之后，对构件进行了识别和划分，得出了通用开发框架的总体结构；第 2 部分，对开发框架的软总线以及构件接口部分进行了实现；第 3 部分，通过实例对提出的通用开发框架进行了验证，最后给出结论。

2 MIS系统通用开发框架的总体结构

2.1 构件的识别

首先对 MIS 系统所涉及的通用功能进行用例分析。图 1 显示了管理员，普通用户，开发维护人员以及其它系统这四个角色与 MIS 系统交互的情况。管理员对系统进行用户管理、权限管理等维护性操作；普通用户则对个人信息以及自己所负责的业务模块进行

① 收稿时间:2008-10-25

管理；开发人员对系统进行一些系统的二次开发操作；其它系统通过提供的接口与 MIS 系统交互。

应用软构件技术的首要任务是对构件的识别。关于构件的识别方法上述文献中提到有多种，这里选取聚类分析的方法。聚类分析识别构件方法是以需求模型作为样本点的数据来源，并给出样本点关联值的计算方法。它的基本过程是以需求模型中的每个用例组成的集合作为样本点集合 X，首先将每个样本点 X_i 自成一类，确定任意两个样本点 X_i 与 X_j 间的关联值 R_{ij}；选取一个合理的最小关联值 R_{min} 作为将两个样本点放入一个聚集的依据，如果 $R_{ij}>=R_{min}$，则将样本点 X_i 和样本点 X_j 放入一个聚集；如果 X_i 和 X_j 属于一个聚集，且 X_i 和 X_k 属于一个聚集，那么将 X_i，X_j，X_k 放入一个聚集。

通过对 MIS 系统基本业务流程分析，得出 MIS 系统基本用例如图 1。业务用例之间的关系主要有 3 种：包含(include)、扩展(extend)和泛化(generalize-tion)。通常情况下，可以设定泛化关系 $R_{ij}=3$；扩展关系 $R_{ij}=2$；包含关系 $R_{ij}=1$；没有关系的 $R_{ij}=0$。R_{ij} 值越大，表明两个用例的关系越紧密，适合放在一个构件内部。这里只将具有泛化关系的用例合并到一个构件内，因此设定 R_{min} 为 3。具体的通过聚类分析进行构件识别的过程如图 1。

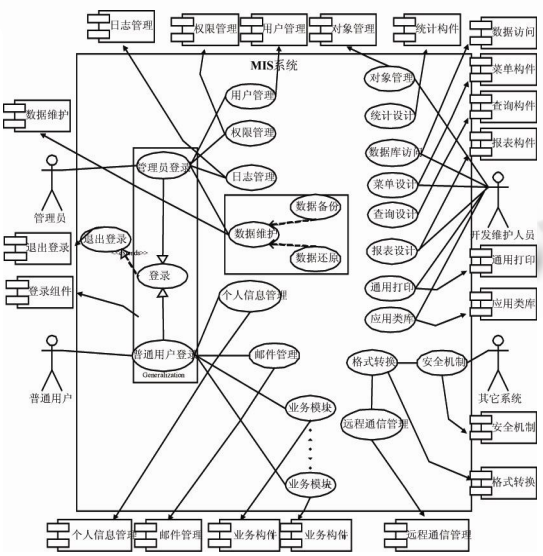

图 1　MIS 系统基本业务用例及构件识别图

2.2 构件的划分

接下来对 1.1 得到的构件按照其重用的范围进行层次划分，如图 2：

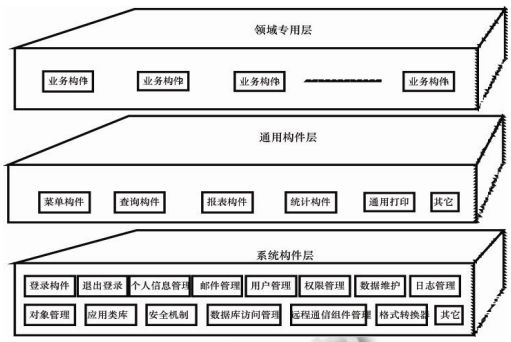

图 2　MIS 系统通用开发框架总体结构示意图

(1)MIS 系统构件层：是指整个 MIS 系统支撑和运行环境都使用的构件，如登录构件、邮件管理、用户管理、权限管理等。

(2)通用构件层：是指在各应用领域中具有较大复用价值的功能性构件，如报表构件、查询构件、统计构件等。

(3)领域专用层：也可以称为业务应用层，是指针对某个特定领域中所开发的构件，在其领域中有一定的复用价值。它一般粒度比较大，要求较高的抽象级别。如以某制造企业 MIS 系统为例，该层会包括决策管理、质量管理、销售管理、采购管理、生产管理、人力资源管理、物流管理、财务管理、办公自动化等构件。

最后由软总线通过各构件的标准接口将它们连接起来，即得到 MIS 系统通用开发框架的总体结构。下一部分将详细介绍软总线与构件接口的设计与实现。

3　MIS通用开发框架的组建

3.1 软总线的设计

这里针对软总线要解决的 4 个问题：模块间的数据传输协议；对模块的调度策略；对软总线的管理策略；对软总线和构件的接口策略。现设计以下 4 个功能模块：

(1)通信模块

该功能模块使用一条虚拟的数据传输线实现软构件库中的各模块的调用、控制和管理。所以通信功能模块的关键是模块间数据传输协议的选择和设计。传输协议要选择通用、标准的数据传输协议进行数据交换，如 TCP/IP 协议。

(2)调度模块

调度模块主要是针对计算机操作系统通过软总线

实现对软构件的调用、安装和卸载，完成对软构件库的管理。通信模块通过数据通道与管理模块相连接(各模块均通过数据通道与管理模块相连接)，进行数据交换。

(3)管理模块

该模块重点解决计算机操作系统与软件组装开发平台及构件库间进行通信和数据交换、数据传递的合理分配和控制使用软总线的问题。调度模块和软件总线管理模块可采用统一的标准语言开发，如.net 平台支持的语言。

(4)接口模块

此模块主要是用来解决软总线和软构件库的接口问题。要实现计算机操作系统对软构件库的管理和控制、软构件之间的互相通信联络、数据和信息的传送，必须首先解决软总线和这些构件接口的问题。接口界面模块采用标准的建模语言 UML 开发，开发时充分考虑各种开发语言所开发的应用软件模块的接口情况，以便于实现操作系统对各种类型模块的管理。

软总线的总体框架如图 3 所示：

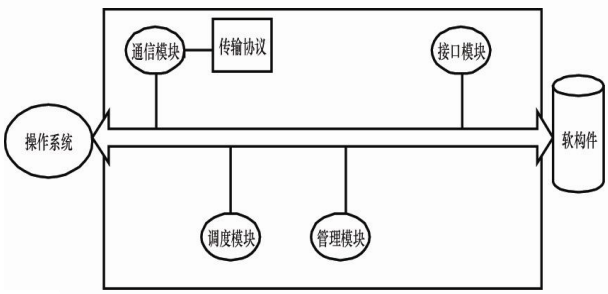

图 3　软总线的总体框架

3.2 构件接口逻辑的设计

构件接口由一组相关的操作组成，因此，设计构件接口的主要工作是设计出一组可准确表达构件功能的相关操作。

构件对应着业务用例，而用例内部实现通常采用活动图来描述，因此构件的接口定义应该能够表达出整个活动图描述的业务流程。对活动图中包含的各种图元进行分析之后，可以发现将每个"活动"图元影射为一个操作较为简单、有效，最后将识别出的所有操作合并为构件的一个接口。此外，如果存在若干个活动必须满足一个事务完整性要求，最好将它们合并为一个操作，这样可以降低构件的实现难度。设计构件接口可按如下方法进行：(1)针对业务用例列出其中涉及的每个操作。(2)将业务用例的每个操作影射为构件接口中的操作，并且消除命名冲突。(3)将必须满足一个事务性要求的若干操作合并为一个操作。(4)为每个构件定义一个接口，将构件所有的操作放入到该接口中。

以用户管理构件为例，它的接口识别过程如图 4 所示：

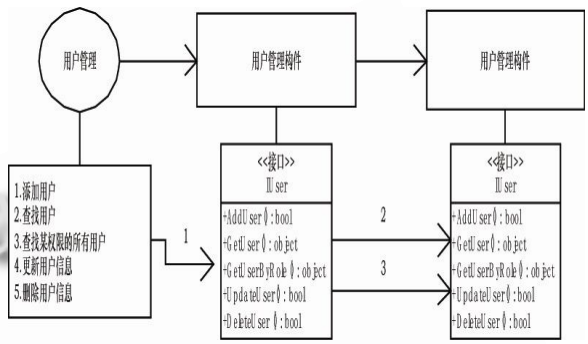

图 4　识别用户管理构件接口

该接口具体的代码实现如下(以.net 平台上的 C# 语言描述)：

```
/// <summary>
/// Interface to the User DAL
/// </summary>
public interface IUser{
    /// <summary>
    /// Add a new user
    /// </summary>
    /// <param name="user"> the user information to be added</param>
    /// <returns> true when delete successfully,false when not </returns>
    Bool AddUser(UserInfo user);
    /// <summary>
    /// Search users by roleId
    /// </summary>
    /// <param name="roleId"> roleId to search
for</param>
    /// <returns>Interface to Model Collection Generic of the results</returns>
    IList<UserInfo> GetUsersByRole(string roleId);
```

4　实例

应用该框架进行系统开发的基本步骤如下：

(1)使用聚类分析方法针对具体的业务用例得出业务构件，组成框架的领域专用层。

(2)对框架中原有的系统构件层和通用构件层进行适当的选择或扩充。

(3)通过软总线对上述所得构件进行配置组装得到最终的系统。

下面以某库存管理子系统为例介绍本通用开发框架的具体应用过程。首先分析该系统涉及的业务用例，该过程中所涉及角色为仓库管理员，业务用例有：出库、入库、盘点等。各用例之间的关系如图 5 所示。利用 1.1 中提到的聚类分析方法可以得到出库、入库、盘点、库存、单据等构件。

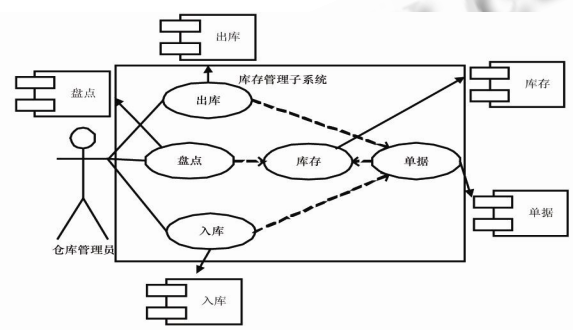

图 5　库存管理子系统业务用例示意图

由于单据在该领域中有较大的复用价值，可以将其放入通用构件层。所以这里选择登录构件、退出登录构件、用户管理构件和权限管理构件组成系统构件层，选择菜单构件和单据构件作为通用构件层，所得各层的构件列表如图 6 所示：

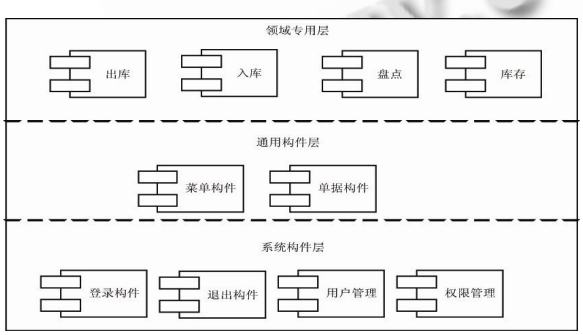

图 6　库存管理子系统各层选用构件列表

接下来按照 2.2 设计各构件接口，如就权限构件而言，其接口的组成部分有：(AddRole, GetMenuBy－Role,DeleteRole,UpdateRole..)。然后对得到的构件进行配置来组装库存管理子系统，如：由用户构件 GetRoleByID 接口得到相应的权限传递给权限构件 GetMenuByRole 接口，从而得到可以显示给该用户的菜单列表，过程如图 7 所示：

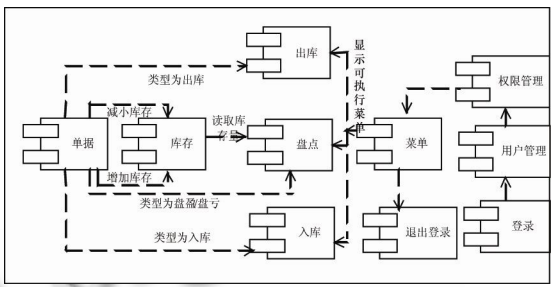

图 7　库存管理子系统构件配置组装示意图

上述实例中，利用已有通用开发框架，使得开发可以专注于业务构件，从而提高了开发的效率。同时根据需求可以合理选择框架的组成部分，可以有效降低框架的复杂性。框架的各部分之间通过软总线通信，降低了各部分之间的耦合度同时也提高了扩展性。下一步的工作是继续丰富该通用开发框架的内容，扩大它的应用范围。

5　结束语

本文针对 MIS 系统开发过程中存在的共通业务特征，利用软构件技术，提出了面向 MIS 系统开发的构件化通用开发框架。从而解决了传统开发框架存在的耦合度高，可扩展性差等缺点。

参考文献

1 Grant L. Designing Component-Based Frameworks. Communications of the ACM, 1999,42(10):38－45.

2 Maya D,Roel J,Wieringa. A requirements engineering framework for cross-organizational ERP systems. Requirements Eng, 2006,11:194－204.

3 David L, Scott F. A Formal Framework for Component Deployment. ACM, 2006,10:325－343.

4 Gaya J, John T, Lin P. Component Agent Framework for domain-Experts (CAFnE) Toolkit. ACM, 2006, 5:1465－1466.

5 Kostadin D, Ashwin D, Steven P. CCALoop: Scalable Design of a Distributed Component Framework. ACM, 2007,7:213－214.