

.NET 与 XDE 融合的构件化信息系统^①

舒远仲, 缪晓峰, 刘 雷

(南昌航空大学 信息工程学院, 南昌 330063)

摘 要: 目前信息系统在各个领域已得到广泛运用, 随着应用环境及用户需求的不断变化, 也相应的要求开发的产品具备高力度的可移植、可扩展功能。针对传统信息系统开发方式的弊端, 主要提出.NET 平台的分层领域构件开发框架, 在 Rational XDE 与.NET 平台融合环境下, 利用其同步开发功能, 对系统构件进行建模、提取、封装设计。并以某高职院校学生管理系统为开发实例, 阐述了部分模块及构件的主要设计步骤, 最后分析了其优点, 实例证明了在这种开发方式下系统具备可移植、可灵活扩展的功能。

关键词: 分层构件; 系统开发; 模型代码同步; 统一建模语言; 程序集

Component Information System Based on Integration of .NET and XDE

SHU Yuan-Zhong, MIAO Xiao-Feng, LIU Lei

(School of Information Engineering, Nanchang Hangkong University, Nanchang 330063, China)

Abstract: At present, the information system has been widely used in various fields, along with the constantly changing of the application environment and the needs of the user, developed products with a high intensity of portable and scalable function are required to respond to that. In accordance with the drawbacks of traditional information systems development methods, this paper mainly puts forward the .NET platform development framework based on hierarchical component. In the integrated platform of Rational XDE and .NET, with the function of Synchronous development, the components can be designed with the operations of modeling, extraction, and encapsulation. Also, taking student management information system of vocational colleges as a development example, this paper describes the main steps of the development approach of part of modules and analyzes their advantages. The example verifies that the system with the development methods described has a portable and scalable function.

Key words: hierarchical component; system development; model-code synchronization; UML; assembly

1 MIS开发概况

随着信息技术的不断发展, 特别是互联网技术的普遍化, 校园信息管理系统的应用已成为提高竞争力的一个重要因素。但是, 随着应用环境的变更以及用户需求的不断变化, 对信息系统也提出了更多新的要求。一方面, 传统开发方式下的信息系统结构和功能固定, 在新环境下的适用度与可拓展度相对较低, 开发的产品针对性强、通用性差; 另一方面, 用户的需求也是不断变化和发展的, 成型产品的可升级性差。

因此, 重复的开发浪费了大量人力、物力和财力资源。

信息系统的开发方法历经两代技术跨越: 面向过程(Procedure-Oriented), 包括面向功能和面向数据流, 主要是通过若干程序语句来实现某个功能的思想完成系统的开发; 面向对象(Object-Oriented), 体现功能与数据抽象方法的统一, 在系统开发中采用封装、继承、抽象等设计方法, 从问题域开始到面向对象分析、面向对象设计再到面向对象编程, 缩短了开发周期。随着软件工程领域软件复用^[1]思想的提出, 信息系统的

^① 基金项目:国家自然科学基金项目(609773096);航空基金(2010Z056007)

收稿时间:2011-09-07;收到修改稿时间:2011-10-02

构件化^[2]开发也得到快速的发展,一些文献中也提出了新的开发思路。文献[3]主要采用在基于 CORBA 技术构件下对 MIS 的设计与开发,从 MIS 的通用功能模块中抽取典型的两个 CORBA 对象进行分析和设计,详细介绍了利用 Delphi 可视化开发环境下开发 CORBA 构件的方法和步骤,并对其中构件入库作了详细的描述,提出了构件化信息管理系统实现方案;文献[4]提出基于 COM 构件技术的高校教务管理系统的开发过程模型,设计并实现了管理系统中的数据库结构、功能模块划分和系统流程设计,并部分实现了高校教务管理子系统。其它一些文献也提出了不同思路的开发方法,如 Maya 等^[5]提出通过层次化开发、重用通用框架来提高信息系统的灵活性;David^[6]定义了应用程序构造箱的概念以及提出了标志事务系统结构,并在此基础上建立了通用的构件部署框架等等。

目前构件化系统开发已成为解决“软件危机”的一种主流开发技术,针对特定的系统功能采用的开发方式有利有弊。本文在目前已有信息系统的研究与开发基础上,运用一种更为快捷的构件开发方式,在 Rational XDE 环境下建立集成信息系统的 UML 模型,利用 CASE 工具的模型与代码同步的功能,生成面向 .NET 平台的程序,在构件提取分析后对系统的部分开发做详细的说明,最后分析了其开发方式的优点,以实例开发步骤说明在应用领域此开发方式的优越性。

2 开发框架

2.1 构件化 MIS 开发过程分析

构件化信息系统开发的优点之一就是可以对构件进行复用化操作。复杂的信息系统体系结构可采用横向分块、纵向分层的设计方式实现。横向分块是将信息系统实现的功能分为若干块,每一块既是独立的一个子系统又是大系统的组成部分,可以针对不同的用户需求进行模块化更改。纵向分层为系统的复杂结构和构件重用提供了一个解决方式,在分层信息系统的各级组织中,子系统与相应的模块通过构件接口进行链接映射。一方面,系统的各个模块使用标准化、模型化、通用化的设计,可完全实现系统的构件化组装,使其具备可扩展性、移植性、可配置性和可维护性的特点。另一方面,在系统的层次化开发基础上再使用构件设计,可减少模块间的重叠,对于类似的管理信息系统的开发,只需要针对该院校的业务特性,对某

些局部的构件进行更改就可完成系统的迅速更新。其模型如图 1 所示。

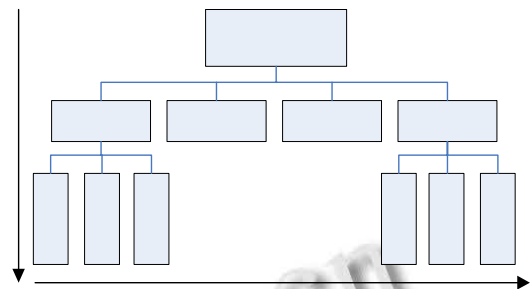


图 1 MIS 构件化过程模型

2.2 开发框架

信息系统是稳定性和可变性相统一的实体,这个特征决定了在构建构件化信息系统时必须采用“标准化”的领域业务构件和能够快速适应变化的开发框架相结合的方式^[7]。在稳定性方面,信息系统的模型及功能方面有很多相似之处,尽管不同的业务针对信息系统的功能需求各异,但在一定的领域内,所处理的信息结构是相对稳定的,变化的只是内容。信息系统所涉及到的业务种类也是有限的,其处理过程也具有相似性,不同的只是其运作和控制方式上的差异。通过系统的功能性分析,根据高职院校的系统特征,进行领域分析,可以将系统的构件分为三类:通用领域构件、专用领域构件、领域共性构件。如图 2 所示,即为一种可行的构件化系统开发框架。

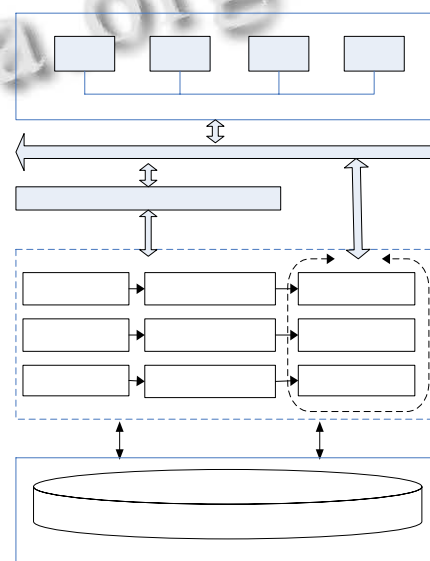


图 2 构件化 MIS 开发框架

利用上述框架可快速的建立系统，框架的核心部分在于领域构件库，根据构件对系统领域不同功能的处理，将构件进行划分：

(1) 通用领域构件：信息系统基本通用功能的构成成份，如基本的数据打印、用户界面、数据访问元素等，它们存在于各类系统应用中。

(2) 专用领域构件：信息系统特有功能的构成成份。根据特定业务流程的需要将初始信息转换为结果信息，实现对特定信息的处理与转换过程，是一个系统的重点部分。首先对系统的核心业务进行分类，然后进行抽象和构件化就形成了可组装的专用业务构件。如：食品质量管理构件、药品管理构件等，主要用于一些专用领域。

(3) 领域共性构件：是信息系统所属领域的共性构成成份，它们存在于该领域的各个信息系统中。如高职院校子系统领域的学生信息注册构件、学生寝室管理构件等等。

为了方便信息系统的设计，必须对系统进行所有功能的领域分析^[8]。领域分析是一种标识出某一类问题的知识并将其明确而详尽地表达出来的过程，其目的是为这一类问题领域的描述和解决方法提供支持。利用领域分析，可以最大限度地复用已有的系统模块和应用程序。要设计重用性好的构件，就必须对要重用的领域有足够的了解，通过对领域知识的分析发掘可重用的成分，标识出构件，并对构件的属性、方法进行设计，对构件功能进行必要的说明。通过信息系统领域分析能够发现该领域的共性，构造出各种可重用的构件，使得这些构件能够在开发该领域的系统时得到最大限度的利用。除了设计前主要的领域分析之外，构件化 MIS 开发框架的建立还需解决以下两点问题：

(1) 构件开发标准。首先，标准的构件开发使公共模型保持一致，让交互操作成为可能，也使具体的构件接口规约一致，有利于实现有效快速的组装。其次，标准可用来确保整体结构的一致性，以便在更大的组装和互操作情形下指定构件的位置。

(2) 组装平台。一方面，基于微软平台.NET 框架的普遍性；另一方面，.NET 的开发环境可以根据服务器构件程序集清单了解服务器构件里的类及其方法、属性、接口及事件等信息，方便对构件的组装与扩展。

3 系统设计

下面以某高职院校集成信息系统中学生信息管理系统为设计开发实例，详细的阐述在面向.NET 的 XDE 环境下系统建立用例图、类图、构件图的主要方法和步骤。实例主要包括学生管理模块，不涉及教务管理方面，其主要模块有以下四个部分：招生管理、学籍管理、班级管理、寝室管理。

3.1 系统用例图描述

系统的用例图主要从系统使用者角度考虑系统功能的静态及动态行为描述^[9]，同时支持系统的需求分析，使得各个角色的操作一目了然。学生信息管理系统的参与者主要有四个：

(1) 系统管理员：主要工作是对系统进行管理与维护。例如可对寝室管理模块的考察内容进行修改，学生学籍数据库的更新等等。

(2) 教学秘书：对录取的学生进行专业、班级的划分，并分配辅导员到各个班级。

(3) 辅导员：主要是对学生的管理进行操作。例如对学生的寝室划分、寝室评比结果录入等等。

(4) 学生：可登陆系统注册，查看录取的相关信息。

系统的用例图如图 3 所示

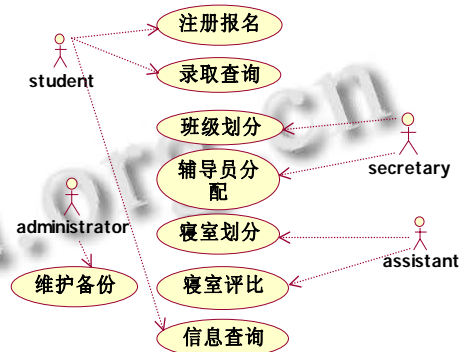


图 3 系统用例图

3.2 系统类图描述

类图主要用来描述系统的静态结构，定义了每个对象的详细属性与具体操作，它只要包括三个部分：类、用户接口、联系。属性描述类性质的实例所具有的值；操作实现类的服务功能，它可以被本类的对象请求执行；用户接口就是用户与系统交互的界面，也可以用对象类表示；联系代表对象类之间的关系，主要有关联、聚合、泛化、依赖等关系。根据对象类的角色性质，学生信息管理系统相应的

划分为以下四个类：“学生”类、“辅导员”类、“教学秘书”类、“系统管理员”类。“学生”类主要对登陆注册报名的学生的信息处理，它的属性有“姓名”、“性别”、“年龄”、“出生年月”等，对这些信息服务的操作主要有系统登陆、信息注册、录取查询等；“辅导员”类、“教学秘书”类和“系统管理员”类同上类似，不再累述。系统类图如图 4 所示：

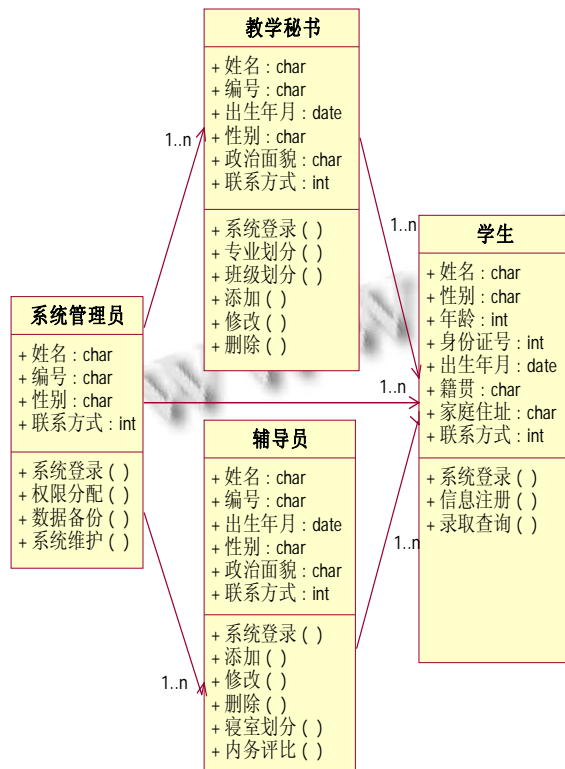


图 4 系统角色类图

3.3 构件抽取分析

构件的抽取由用例驱动，用例描述系统的需求，类图则描述实现用例的类之间的关系，构件则是类的抽象。高职院校学生管理系统的功能各异，为了更好的提取构件，我们按照 2.2 节在领域分析后提出的构件层次进行划分，主要分成三类：通用领域构件、专用领域构件、领域共性构件。结合复用的原则，对系统进行整体的构件抽取，由于系统不包含特殊业务的处理，这里我们不考虑专用领域构件。

- 通用领域构件有：界面设置构件，信息查询构件、数据分页显示构件、数据录入构件、报表工具构件、权限设置构件、操作日志构件、数据维护构件等；
- 领域共性构件有：整体窗口构件、数据库连接

构件、用户信息管理构件、学生信息管理构件、班级管理构件、评比管理构件等。

基于 UML 的构件抽取是一个复杂的过程，需要考虑的因素比较多，根据不同的业务需要，应用的知识也不相同。为了抽取更加合理的构件，一般需要多次迭代，同时也需要团体协调工作流程。所以，在构件的抽取过程中，用例图和类图的建立至关重要，根据用例图与类图的设计将构件的粒度进行细致的划分及抽取。

4 系统实现

系统构件化设计主要目的就是提高系统的可移植、可拓展功能，在设计构件时考虑多个因素(1)复用性(2)封装性(3)粒度大小(4)集成性(5)接口连接机制。下面以学生信息管理构件为例，阐述模型与代码同步过程中具体开发方法。

4.1 代码同步与逆向工程

Rational XDE 与 .NET 平台融合之后，一方面可以在 .NET 平台建立系统模型，另一方面可以通过模型与代码的同步功能，实现代码的转换。通过 3.1 和 3.2 中对系统的分析与设计，可以通过 Synchronize 和 Generate Code 操作生成系统所需源代码。在这种开发方式下，由于系统模型与代码同步，代码与系统设计的用户界面同步，使得整体的开发更具可视化，开发效率也相应的提高。建模工具 Rational XDE 另一个强大的功能就是从编码进入建模工作，可以通过编码实现设计的细节，简称为逆向工程^[10]。在 3.1 和 3.2 小节，对学生管理系统用例和类的分析之后，加入页面层的模型分析，实现页面层与系统设计层两者模型接口的连接。

利用 XDE 这个强大的功能，以学生信息管理页面为实例，从建立 WEB 应用程序开始，在 VS.NET 中添加新的 ASP.NET Web Application 的 C# 项目，添加 LoginPage、ErrorPage 页面和 TaskList、Button、DropDownList、TextBox、Label 控件，然后利用 VS.NET 可视化编辑工具绘制页面与控件，接下来通过 XDE 逆向工程建立 UML 模型，XDE 捕捉到它的结构，如图 5 所示，即学生信息管理页面的反向工程类图。它包含学生信息的添加、删除操作，图 5 是以页面形式体现出来的，因此包含页面内容的确定、取消等操作。

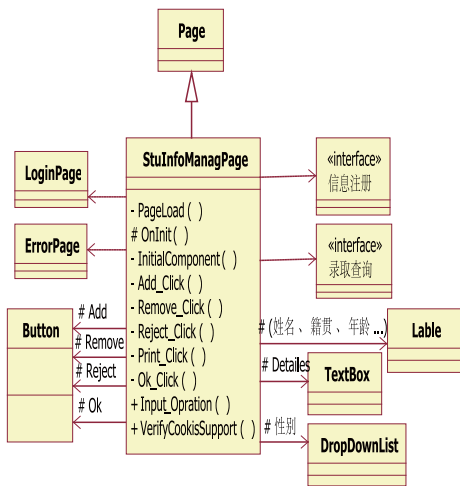


图 5 学生信息管理页面的逆向工程类图

在建立逆向工程模型之后，可以对 UI 的需求进行分析并绘图，此时编码与设计都前后展开了，可以对类做进一步的细化工作，包括接口的依赖关系和类之间的关联等，实现页面层与模型设计层的交互，开发更快捷化。

4.2 数据库模型建立

在学生管理系统中，对数据库的分析、设计和实现均采用构件化设计，利用 XDE 建立数据库表、关系和存储过程。数据库的模型建立之后，XDE 会自动处理这些过程：生成数据定义语言(DDL)脚本，或者直接创建或更新 SQL 数据库。首先，需要标注每个表或存储程序，让它们驻留在新的数据库中，XDE 通过使用关系箭头表明出来，如图 6 所示，即数据库单元模型图：

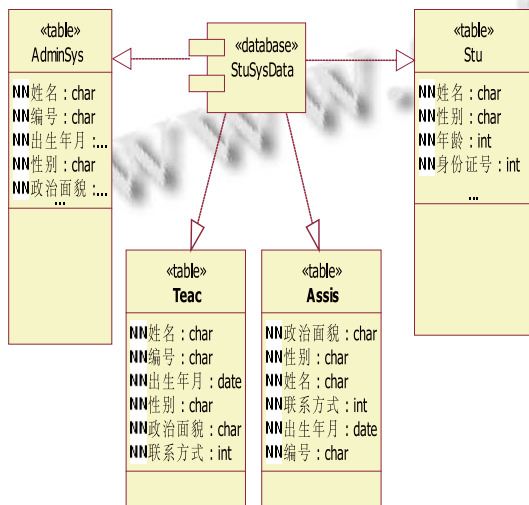


图 6 数据库单元模型图

然后，从数据模型中生成该数据库，编写数据库的存储过程，通过使用 VS.NET 的工具生成 DataSet 类，在 VS.NET 中使用 Add New Item 对话框来添加一个新的数据集文件，TaskData.xsd。该文件允许对 DataSet 类进行可视化编辑，并且最后可以添加新的数据集来对 DataSet 类进行可视化编辑，可以拖放 SQL 单元，同时 VS.NET 将自动生成一个 DataSet 类来保存存储过程的结果，随后可以在项目中创建并使用 TaskData 对象。存储过程关键代码，如下所示：

```
CREATE PROCEDURE "dbo"."GetStudent"@stu
dentID char(8)
```

```
AS
If (@studentID is null) or (@studentID='')
select * from student
Else
select * from student where (studentID=@stude
ntID)
Go
```

```
CREATE PROCEDURE "dbo"."GetStudentTask"
@studentIDList TEXT
```

```
AS
If (@studentIDList is null) or (@studentIDList like
'')
Select * from studentTask
Else
Select * from studentTask where (@studentIDList
like (''+studentID+'%'))
Go
```

4.3 构件封装

在完成所有领域构件的开发之后，需要将构件进行封装，再通过接口类将构件组装成可应用的系统。NET 框架应用程序的主要构造块就是程序集，它构成了基于.NET 的应用程序的部署、版本控制、重用、激活范围和安全权限的单元，因此可以将构件封装成程序集，它可以是动态链接库（DLL）形式或者独立的应用程序（EXE）形式，可以包含多个实体模块，每个模块可包含多个构件。如图 7 所示，程序集 A 包含两个构件“信息统计构件”和“信息查询构件”组成一个单一的信息模块，而程序集 B 包含三个构件“学生管理构件”、“班级管理构件”、“用户管理构件”组成一个单一的管理模块。使用这样的封装方式，可以迅速的

对需要更新的构件进行更新，避免了重复的开发。

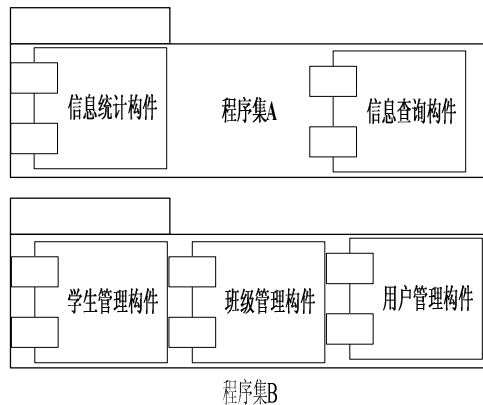


图 7 构件封装为程序集图

程序集是 .NET 中基本的封装单元。程序集中的元数据是一个全面、标准、完全的、描述程序集内所包含的内容的方式^[4]。元数据描述在程序集中有多种可用的类型(如类、接口、枚举、结构等)，以及包含它们的命名空间、每个类型的名称、可见性、它的基类、方法、支持的接口、每个方法的参数等。面向 .NET Framework 的编译器向所有的模块和程序集嵌入元数据,使 .NET 构件成为自描述构件,使构件交互更加简单,这种方式有助于构件的无缝集成。

构件化信息管理系统的开发主要优势在于每一个功能构件对外提供标准接口，客户调用相应的接口函数获得所需的功能。由于采用构件设计，系统的功能由一个个功能独立的构件“组装”而成，减少了系统之间的代码依赖性，当系统将来需要进行改动或升级时，只需对相应的构件进行改动，而且系统需要增加新的功能时，也只需针对新的需求设计新的应用构件程序，与原系统“组合”使用，这样就大大增强了系统的开放性，减少了系统的开发及维护工作量。由于组件的优点在于可重用性好，具有标准应用接口，因此在系统设计中应将功能尽可能地细化，使组件能完成某一项独立的功能，以充分体现组件的优越性，提高组件的重复利用率。

5 结语

本文在 .NET 与 XDE 平台融合环境下，通过建立系统 UML 模型，以实例化的开发证明了其设计方式的优越性，信息系统在可移植、可扩展和代码生成等进一步升级方面都具快捷化。它的主要设计思想是将构件开发与 UML 建模和 .NET 平台的系统设计相结合，使得系统建模层、页面层、代码层能够同步一致，一方面帮助开发人员进行高效率的建模设计和高质量的代码编写；另一方面还可以可视化地追踪软件运行逻辑，以序列图的形式展现该运行逻辑，能够更好的了解与测试系统的稳定性。

参考文献

- 1 梅宏.软件复用技术研究与应用.2002 年中国软件国际化论坛,2002:39-42.
- 2 杨芙清,梅宏,等.构件化软件设计与实现.北京:清华大学出版社,2008.210-216.
- 3 吴国庆.MIS 系统复用构件库的研究与实现.成都:电子科技大学,2006:30-42.
- 4 吴志亚.基于构件的高校系务管理系统设计与实现.上海:上海交通大学,2008.25-40.
- 5 Maya D, Roel J, Wieringa. A requirement engineering framework for crossorganizational ERP systems. Requirements Eng, 2006,11:194-204.
- 6 David L, Scott F. A Formal Framework for Component Deployment. ACM, 2006,10:325-343.
- 7 赵方圆,魏志强,等.基于构件的可重构软件开发平台技术研究.计算机工程与设计,2005,26(1):33-36.
- 8 王筠,郭莹,杨萍,等.领域需求差异分析方法与应用.计算机应用,2010,30(8):2177-2180.
- 9 Shoemaker ML. UML 实战教程-面向 .NET 开发人员.北京:清华大学出版社,2006.20-81.
- 10 周新宽,陈平,李青山,等.一种 UML 活动图的逆向恢复方法.计算机工程与应用,2006,42(17):4:26-28.
- 11 特罗尔森,王少葵,等.C#与 .NET3.0 高级程序设计(特别版).北京:人民邮电出版社,2008:6-21.