

报表自动生成模型^①

冯文堂¹ 王纪梅²

(1. 山东济南润丰农村合作银行 济南 250001; 2. 山东协和职业技术学院 济南 250109)

摘要: 报表功能可以说是所有数据库应用系统最基本、最重要的需求。本文通过对几种报表生成模型的研究, 提出了一种基于 MDA 的报表自动生成模型, 并以 ASP.NET 为目标平台验证环境, 证明了该模型的正确性。

关键词: 报表; MDA; FMP; 报表模型; 代码自动生成

Report Automatic Generation Model

FENG Wen-Tang¹, WANG Ji-Mei² (1. Shandong Jinan Runfeng Rural Cooperative Bank, Jinan 250001, China; 2. Shandong Xiehe Vocational and Technical College, Jinan 250109, China)

Abstract: Report function is the most basic and important need of the database application system. On the basis of studying plenty of report generation model, this paper puts forward a kind of automatic generation model based on MDA. ASP.NET is used as a target platform in the experiment which shows that this approach is correct.

Keywords: report; MDA; FMP; report model; codes automatic generation

1 介绍

目前市场上存在各种的报表生成工具, 但存在以下问题: 行业针对性强, 难以满足各种 MIS 的特殊要求; 可移植性差; 无法支持分布式环境下多数据源的查询。

模型转换是模型驱动开发过程中的一个关键步骤, 首先通过抽象分别建立源模型和目标模型的元模型, 对比得到的两个形式化元模型的抽象组成元素及其语义, 定义它们之间映射规则^[1,2]。

本文在分析了几种报表生成模型的基础上, 提出了一套基于模型驱动的报表自动生成模型的开发方法。

2 几种报表生成模型的分析

2.1 结构控制模型

该模型^[3]基于应用生成的思想, 采用了结构控制技术, 为用户提供了报表描述的用户定义语言, 并提供了建立报表数据结构和用户级制作数据窗口对象的工具。通过报表管理器统一解释用户定义语言, 自动

生成和校验报表数据, 并由报表管理器统一打印报表。

对于二维表的数据处理问题采用这样的方法。建立该报表的数据结构, 并将该结构存放的数据视为一矩阵, 表示为: $P = [p_{ij}]_{m \times n}$

另外还要建立与该报表相对应的报表控制结构, 录入控制数据, 可视此结构存放的控制数据为一矩阵, 称此矩阵为结构控制矩阵, 表示为: $A = [a_{ij}]_{m \times n}$

A 矩阵元素所对应行的属性和列的属性进行详细描述, 并精确描述了与之相对应报表(P 矩阵) 中的预确定元素的 SQL 语句。

2.2 基于 oo 的报表知识库模型

该模型^[4]从对象的角度来分析报表的属性和操作, 将报表类定义为一组具有相同操作的报表对象的集合。记报表对象的集合为 T, 报表类数据结构集为 P, 报表模型可以定义为:

$$TReport ::= \langle Tname, Inh, P, Coi, Itf \rangle$$

$$P = (d_1, d_2, \dots, d_n)$$

式中, Tname 是类标识, Inh 是 Coi 是操作集

① 收稿时间:2010-03-02;收到修改稿时间:2010-04-15

的具体实现, Itf 为类的对外接口, 接受用户的输入参数。P 包含了用户的全部需求 R, 即 R 为 P 的 Public 属性, $R \in P$

$R = \{d1, d2, d3, d4, d5, d6, d7, d8\}, R \in P$;

$d1 = \{fkKind, RptName, HeadCaption, BottomCaption, Page, Direct\}$

$d2 = \{obkind, obPointer, (x1, y1), (x2, y2)\}$

$d3 = \{TableName\} \quad l = 1, 2, 3 \dots (1-5)$

$d4 = \{(x1, y1), (x2, y2), Text, Font, TableName, FieldName\}$

$d5 = \{(x1, y1), (x2, y2), Text, Font, FormulaStr, ReplID, StatType\}$

$d6 = \{(x1, y1), (x2, y2), Text, Font, ReplID, Direct\}$

$d7 = \{(x1, y1), (x2, y2), LineWidth\}$

$d8 = \{FixedSize, Size, Name, Color, Style\}$

式中 d1 为报表模式子类, d2 为当选对象子类, d3 数据源子类, d4 数据文本子类, d5 公式子类, d6 静态文本子类, d7 线子类, d8 字体子类, 每一种子类都有一些相对应的实例属性。

3 模型驱动的报表生成模型

如图 1 所示。该模型建立在 FMP 模型^[5]方法的基础上, 利用了 MDA 中模型转换的思想, 可以满足用户动态添加新报表的要求, 开放性较强。

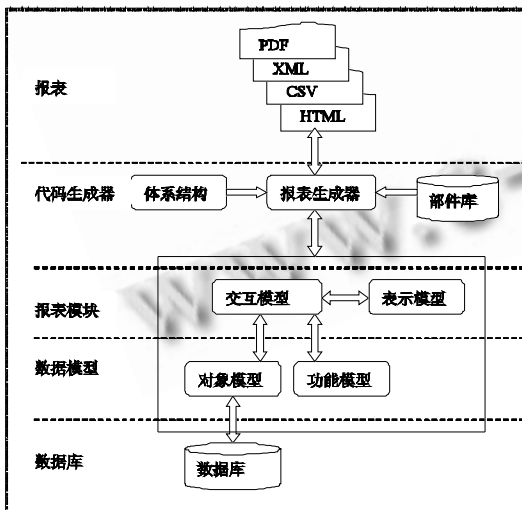


图 1 基于 MDA 的报表生成模型

基于 MDA 的报表生成模型, 主要包括: 数据模型, 报表模块, 代码生成器三部分。

3.1 数据模型

数据模型主要有功能模型和对象模型两部分组成。功能模型主要使用扩展用例图来进行表达描述, 并通过分析需求, 确定界面功能、界面框架、界面之间的关系以及界面对内在模型(对象模型和交互模型)的需求。对象模型是界面的数据模型主要使用对象关系描述应用领域的内部数据, 它从静态的角度描述交互系统, 为功能模型和交互模型提供基础支持。一般说来, 对象成员的基本数据类型有数值、字符串、枚举, 而对象模型全面扩展了对象的数据成员描述特性, 以满足界面描述的需要。为了满足对报表页面定制的要求, 需要对一般的对象模型进行扩展。

对象模型的扩充主要包括如下: 为提供界面上交互单元的信息, 对属性扩充了 UI 类型、缺省值、标签和计量单位, 并扩展了数据类型的描述; 为表现 UI 间的相关性和关联性提出了导出关系和联动关系; 为满足用户的心理和视觉特性提供了属性分组的描述; 使用对象间的关系表示界面的导航。

扩展的属性描述形式如下:

$\langle Attribute \rangle := \langle AccessType \rangle \langle DataType \rangle (\langle AttributeName \rangle \{ = \langle DefaultValue \rangle \}) \{ \langle ValueRange \rangle \} \{ \langle Unit \rangle \} \{ \langle DataSource \rangle \} \{ \langle UIType \rangle \} \{ \langle Label \rangle \}$

其中, $\langle Attribute \rangle$ 表示属性; $\langle AccessType \rangle$ 表示运行时的可见性; $\langle DataType \rangle$ 表示数据类型; $\langle AttributeName \rangle$ 为属性名; $\langle DefaultValue \rangle$ 为属性的缺省值; $\langle ValueRange \rangle$ 为取值范围; $\langle Unit \rangle$ 为计量单位; $\langle DataSource \rangle$ 为值来源; $\langle UIType \rangle$ 为可视对象类型; $\langle Label \rangle$ 是可视对象所附带的标签。

3.2 报表模块

报表模块框架包含了交互模型和表示模型两部分。该模块实现了用户对报表内容和报表格式的定制, 为以后的报表工具代码的生成提供全部的参数信息。

交互模型可看成由三元组 $\langle V(G), E(G), \phi G \rangle$ 表示的有向图, 其中 $V(G)$ 是交互对象集合, $E(G)$ 是交互关系集合, ϕG 是从交互关系集合 E 到交互对象有序偶集合上的函数。

交互对象包括数据对象、汇集(数据对象的集合)、查询条件对象、控制参数对象、角色、用例、部件等。数据对象来源于数据模型, 不同之处在于, 它与界面显示直接相关。查询条件对象一定是可见对象, 一般访问模式是用户输入, 特别是对于数据检索而言。如

果查询条件对象是个数据对象,那么它大多是来源于数据库的对象,其输入方式有别于一般对象;如果是个汇集,那么它一般是只读的,它的事件可能会触发其他操作。控制参数对象同样属于参与加工的数据对象,是数据处理的前置条件,它的作用在于给关联的对象或汇集传递控制参数。角色的定义不同于 UML,它是界面中的用例操纵者,只可以是人,不可以是外部系统。

交互关系表达了交互对象之间的关系,它强调对象与用例间的交互行为和这种行为对于界面中对象间关系的影响。包括以下几种:关联关系、消息发送、方法调用、参与、界面导航。

表示模型的功能是定义界面的可视外观,描述可视成分的组成和成分之间的关系,主要处理界面元素的可视形式显示和界面布局问题。

3.3 代码生成器

在基于 MDA 的应用程序自动生成中,从 PSMs 到代码,甚至从 PIMs 到代码的映射都是由代码生成器实现的,代码生成器是用来完成代码生成的核心组件。

3.3.1 报表生成器

报表生成器的设计是生成目标代码的关键一步,它的设计与具体的平台及生成的目标代码相关。以生成 ASP.NET 网页为例,实现一个报表生成器。

(1) 读取 xml 文档并验证

读取 xml 文档,可以采用两种应用程序接口:DOM 和 SAX 接口。把 XML 文档读入内存建立 DOM 树时,可以用 DTD、XDR 或 XSD 验证文档的正确性。

表 1 映射转换关系

FMP(XML 文档)	ASP.NET	说明
体系结构模型	Global 模块,工程信息文件	
对象模型	业务处理模块	对交互、表示模型提供支持
交互模型	后台处理代码和前台展示对象	用例,方法调用,界面导航分别映射为控制对象,对方法的调用
功能模型	用树形表示的主菜单	
表示模型	页面组成元素的前台展示	位置、大小、颜色等

(2) 确定映射转换关系

映射转换关系主要包括两个方面:XML 文档包含了对象、交互、展示等不同的模型信息;模型中具体的组成元素与目标语言之间的映射关系。很显然,第二方面的映射是关键核心的部分(参看表 1)。

3.3.2 报表生成算法

目标代码的生成算法主要包括报表框架的生成、报表界面元素的生成和报表打印及预览程序的生成。

1) 报表框架的生成算法

该算法用来生成用户报表界面的最上层的算法,它的依据就是根据表示模型(界面模板)提供的信息来生成用户报表框架,报表界面元素,报表界面元素布局等。

```

Begin InterfaceGenerate( 界面模板对象
UITemplateObj)
{  OpenCurrentDoc(UITemplateObj);
  //打开当前界面模板对象对应的文档
  输出报表的表头,标题部分;
  AreaNodeList=GetCurrentDocAreaNodeList
(CurrentDoc);
  //获得当前界面模板文档的节点对象
  While(AreaNodeList)
  {   AreaNodeObj=AreaNodeList.GetCurrent
tAreaNode();
    //获得当前的区域接点
    UiObjList=AreaNodeObj.GetObjList();
    AreaNode.UiPlaceMent(UiObjList);
    //调用界面布局算法[60]
    If(AreaNodeObj.DataType=="汇集" and
AreaNodeObj.placementType=="对象集合")
    //数据类型是汇集展示形式是对象集合
    {   ObjGroupGenerat(UiObjList);
      //调用汇集对象初始化算法  }
    While(UiObjList)
    {   UiObj= UiObjList.GetCurrentObj();
      //获得当前界面对象
      CreateUiComponent(UiObj);
      //生成目标代码下的报表对象源代码
      UiObjList.MoveNext;  }
    AreaNodeList.MoveNext();
    //链表指针下移 }

```

```

//对整个界面模板区域划分
调用界面布局算法;
输出报表表尾部分;
调用打印预览程序的生成算法;

```

End

2) 报表界面元素的生成算法

该算法是用户报表界面基本元素生成的算法，主要用来生成具体的报表界面控件的。这是一个递归算法，因为，有的控件还有子控件。例如 Frame, Tabstrip 等等，这类控件里面还有其它的自控件所以必须利用递归的方式来生成。

```

Void CreatUiComp(元素指针*pUiObjComp 对
象指针*pANode,代码容器指针 *pCode)

```

```

{ if (pUiObjComp==NULL) then return;
//如果当前的对象指针指向空则返回

```

```

if ( "界面元素类型" ="Group Box" ||
"RootArea" || "Grid")

```

```

{ //如果是控件组需要生成其孩子控件,因此
需要递归调用函数,表示模型里面把区//域的//根节
点定义为:"RootArea" 与源代码里面的 frame 相对
应。

```

```

AddUiComponentHead(pUiObjComp,pANo
de,pCode);

```

```

//生成当前控件头定义的源代码
for(当前元素孩子链表头节点 to 尾节点)
{ pChildObject=pUiObjCompGetNext(pos);
//获得当前孩子节点并且指针下移
CreatUiComp(pChildObject,pANode,pCode)

```

```

//递归调用控件生成算法 }

```

```

AddUiComponentTail(pUiObjComp,pANode
,pCode);

```

```

//孩子节点生成完成后生成当前控件尾定义的源
代码}

```

```

else
//如果是标准控件 UiObj 对象时,需要做的工作。

```

```

{AddUiComponentTail(pUiObjComp,pANod
e,pCode)

```

```

//直接生成标准控件,该类控件没有孩子节
点。}}

```

3) 报表打印预览程序

打印预览模块是报表系统的又一个核心模块，它是整个系统的输出部分。目前在.NET 平台下，实现了一个打印预览类，它实现报表打印预览的基本过程是：先定义好整个打印页面的布局，然后再定义各个打印区(Rectangle 对象)，包括打印区的尺寸、需要打印的文本、图形、图像等，最后调用系统提供的 DrawString(), DrawRectangle()方法来输出相应的文本、图形、图像即可。

4 报表的设计及生成实例

该部分描述的是在 AUI 环境中自动生成管理系统应用程序中报表框架代码例子。它的生成共有以下几个步骤。

首先，获得用户需求。

其次，根据用户需求在功能模型上建立用例模型图，从上层明确地描述了该系统的用户需求。如图 2 所示。

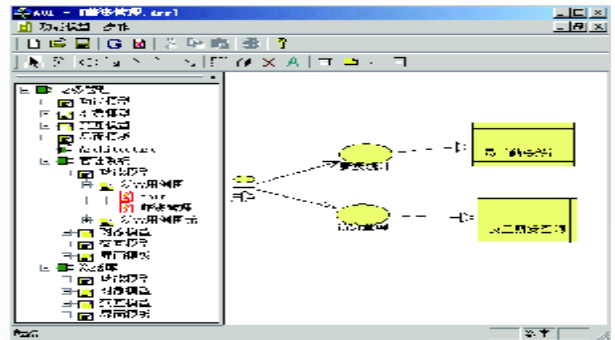


图 2 管理系统功能模型

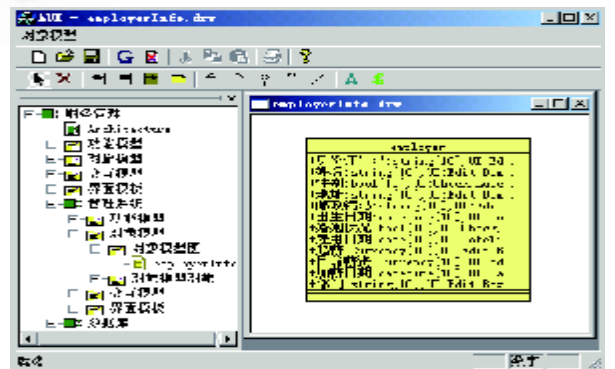


图 3 管理系统的对象模型

再次，创建对象模型和交互模型图。在对象模型图中有一个 employer 对象，该对象有身份证号码，

姓名,性别,地址等几个属性组成。在交互模型图中,有一个员工信息.薪资统计汇集对象。如图3所示。

第四,创建员工薪资统计报表页面,并进行相应的参数设置,为报表代码的生成提供全部的参数信息。如图4,5所示。

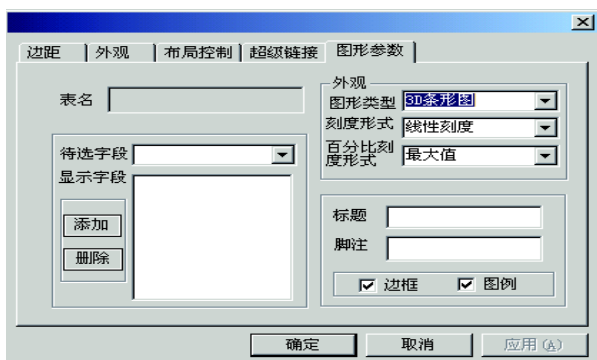


图4 图形展示的参数设置

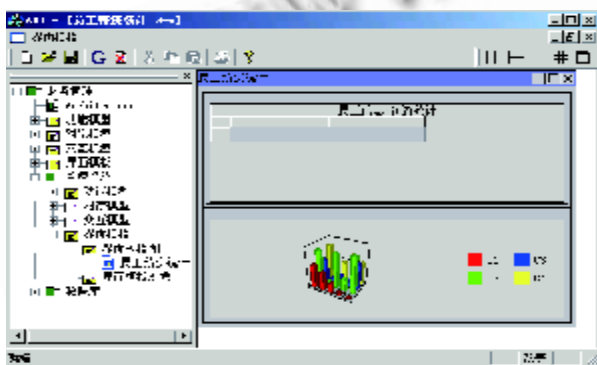


图5 报表的界面模板图



图6 报表页面

最后,选择目标代码语言,本例选择的是asp.net,执行代码自动生成,即可生成系统中的报表源代码。其中员工薪资统计报表页面的运行结果如图6所示。

由于篇幅有限我们在这儿只给介绍了一个简单应用中报表的生成。

5 总结

本文介绍了与报表相关的一些技术,总结分析了一些具有代表性的传统报表工具的报表模型。在此基础上提出了一个基于MDA的报表自动生成模型,对此模型作了详细的分析论述,并在基于FMP模型的AUI环境中构建了相应的报表自动生成环境。

在今后的工作中需要进一步解决以下的问题:①解决对数据模型描述中的数据类型支持力度不足的问题,目前支持少数几种数据类型。②就其代码的重用性来讲还存在着不足,目前单独模型的设计还不能被下一个要自动生成的目标工程所引用③目前系统只支持C#语言的报表代码的生成,其他语言的代码生成有待完善。④需要进一步改善系统的功能,使报表界面的图形表达方式更完善,改善系统的算法。

参考文献

- 1 Caplat G, Sourrouille JL. Model Mapping Using Formalism Extensions Software. IEEE March-April 2005,22(2):44—51.
- 2 Caplat G, Sourrouille JL. Model Mapping in MDA. Workshop in Software Model Engineering (WISME2002), 2002.
- 3 廉东本,孙德松,王汉军.报表生成系统的研究.小型微型计算机系统,1997,18(9):65—68.
- 4 袁裕.可视化动态报表系统的UML建模研究与实现.计算机系统应用,2001,10(11):39—41.
- 5 万建成,孙彬.支持用户界面自动生成的界面模型.计算机工程与应用,2003,39(18):114—118.