

文章编号:1000-582X(2003)03-0102-04

# 一类 DSS 方法库的可重用体系结构\*

李勇,肖智,陈玲

(重庆大学工 经济与工商管理学院,重庆 400044)

**摘要:** 在实践中,开发 DSS 系统往往是一个庞大的工程,系统开发费用的控制、开发效率的提高、软件质量的保证、软件开发一致性及软件可靠性的保证等显得非常困难,这就要求有一个完善的方法库的有效支撑。从方法库的可重用性出发来解决上述问题,将方法的可重用性分成了程序代码设计和模型构造这两个层面上的可重用性,使用面向对象的 OOM 和 OOP 方法,根据具体算法本身的逻辑结构,建立了方法库中方法的程序代码的可重用结构,给出了方法库的‘类层次’结构;同时通过关系数据模型来管理方法从而达到了在模型库中重用方法的目的。

**关键词:** 决策支持系统,面向对象,方法类库,类层次结构

**中图分类号:** F270.7

**文献标识码:** A

决策支持系统(Decision Support System, DSS)是以管理科学、运筹学、控制论和行为科学为基础,以计算机技术、模拟技术和信息技术为手段,面对半结构化的决策问题,支持决策活动的具有智能作用的人-机系统。DSS 的设计是模型驱动的,这与 MIS 的数据驱动设计思想有着很大的区别,所以模型库及模型库管理系统是 DSS 软件的核心。而方法是模型得以实现的一个基础,方法指基本算法,如:数学方法、数理统计方法、经济数学方法、优化方法等。因此,完善的 DSS 系统,要求有一个完善的方法库的有效支撑。从软件实现的角度,结合方法库中的方法在 DSS 中的作用来分析,方法具有更高的可重用性、稳定性及可靠性要求。典型的 DSS 系统结构图如图 1 所示:

在 DSS 中,方法的可重用体系可以细分为方法之间的程序代码的可重用性和在模型构造中重用方法这两个层面。用 OO 方法来实施方法库的构造和管理,在方法库中使用 OOM 和 OOP 技术实现方法的程序代码的重用,而通过关系数据模型来管理方法从而达到在模型库中重用方法的目的,形成一个完善的 DSS 可重用体系结构<sup>[1]</sup>。

面向对象方法(Object Oriented Method, OO 方法)是近年来在软件工程和计算机程序设计领域中逐

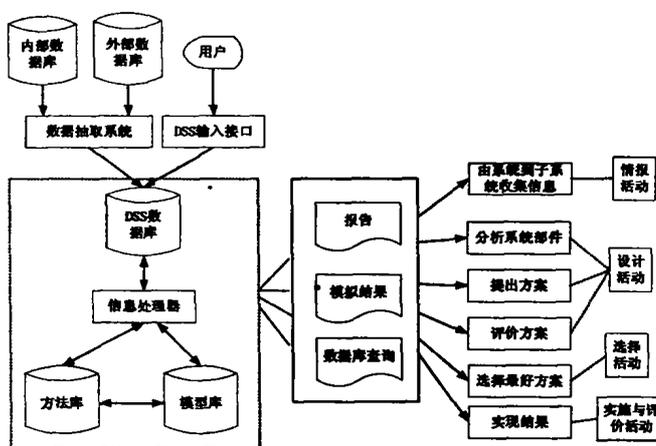


图 1 DSS 系统结构图

渐成熟起来的一种先进方法,该方法以对象和类为基本概念,以“继承、封装、多态等”为基本特征,以提高软件可重用性、稳定性、可靠性、易设计和易编程性、易维护性等为基本出发点,发展出了一整套完整的理论和方法<sup>[2]</sup>。对象所固有的封装性和信息隐藏等机理,使得对象内部的实现与外界隔离,具有较强的独立性,提供了比较理想的模块化机制和可重用的软件成分,它所实现的可重用性是自然的和准确的。相关的 OO 软件开发工具也随着 OO 技术而逐渐成熟,如:C++、Delphi、Powerbuilder、Java 等。

\* 收稿日期:2002-11-26

作者简介:李勇(1967-),男,四川德阳人,重庆大学讲师,技术经济学管理在职博士,研究方向:商务智能,信息系统与决策支持系统。

### 1 方法程序代码的面向对象可重用性分析——类层次结构分析

方法库由方法程序库和方法字典组成,方法程序库负责方法库中方法程序代码的存储,方法字典对这些方法进行登录和索引<sup>[3]</sup>。

面向对象的软件技术中有两种方法可重用对象类:一种方法是创建该类的实例,从而直接使用它;另一种方法是使用继承机制,使子类可以重用父类的数据结构和程序代码,而且可以方便地修改和扩充。可以将方法库中的所有要实现的算法视为一个系统,它由若干对象以及对象之间的关系组成。从算法本身的逻辑流程出发,将算法分解或合并成具有一定功能的类对象,这些类对象通过继承以及其他的OO重用机制形成方法类库,实现方法的程序代码的可重用性<sup>[4]</sup>。

选择算法中的基本数学方法、统计方法和运筹优化方法,研究这三类方法中所包含的各种算法本身的逻辑流程和算法之间的逻辑关系,抽象、合并、分解出具体的算法类对象,从而构造出方法库中的算法类层次结构。在对方法库的类层次结构运用OOM技术建模时,采用UML(统一建模语言)来进行描述。

#### 1)类层次结构图

在图2中仅体现各类的上下级层次关系,对于各类之间的相互关系在图3中展示。

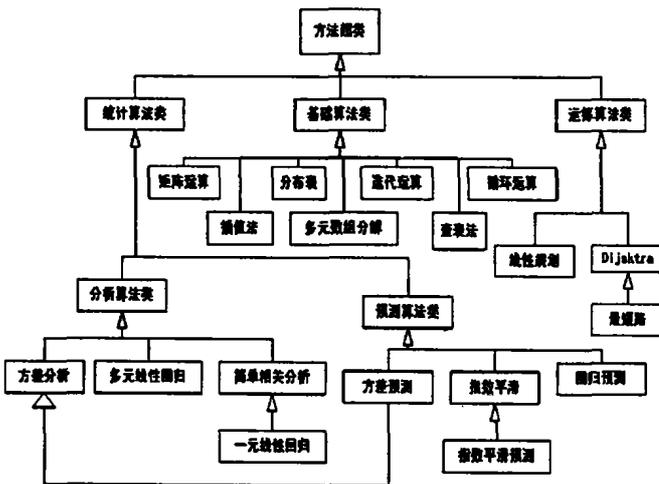


图2 方法库的类层次结构图

#### 2)类框图

在研究类与类之间的关系时,舍去具体算法类的公共父类,仅将具体算法类作为算法系统中的类对象(类层次结构中的‘类’在算法系统中称之为‘类对象’,在DSS中称之为‘方法’),分析它们之间的关系

得到类框图(图3)。

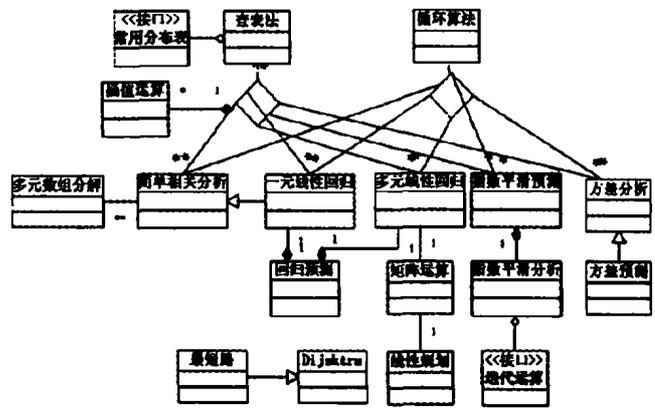


图3 方法类框图

#### 3)程序代码示例

使用JAVA语言开发和构造算法类的程序代码及其结构,现以统计算法类中的一部分算法为例加以说明<sup>[5]</sup>。

```

◆ Public class Method(方法超类) {
    protectedString MethNam; //方法名称
    public void setMethNam() { //方法命名
        MethNam = "方法超类";
    }
    public String toString() { //显示方法名称
        return getMethNam();
    }
}

◆ Public class StatArgl(统计算法类) extends Method(方法超类)
    public void setMethNam() { //替换直接父类的方法命名
        MethNam = super. setMethNam() + ". 统计算法类";
    }
}

◆ Public class Anal(分析算法类) extends StatArgl(统计算法类) {
    public void setMethNam() {
        MethNam = super. setMethNam(.) + ". 分析算法类";
    }
}

◆ Public class SimRelAna(简单相关分析) extends AnalArg(分析算法类) {
    private double R; //简单相关系数
    private xSum, ySum, xySum, xxSum, yySum, up, subDown, down;
    private int n;
    private String grade; //显示相关程度的字符串
}

```

变量

```
private double[][] rmat;
SimRelAna (double [][] mat) { //简单相关分析类的构造函数}
    public String getGrade () { //相关程度的显示}
        //其他细节省略...
```

```
◆ Public class SimLineReg (一元线性回归) extends
SimRelAna(简单相关分析){
protected double[][] rmat, double[][] newY;
protected double[] wch = new double [7];
private double chR; //相关系数临界查表值
SimLineReg (double [][] matr) { //一元线性回归分析类的构造函数}
//其他细节省略...
```

## 2 关系数据库管理方法对象的数据结构

在方法库中是使用方法字典来进行方法的索引和解释,方法字典使用关系数据模型来实现。在该关系数据库中不含有具体的方法代码,仅含有方法的一些索引值;它既支持方法库管理系统又支持模型库管理系统中的‘方法对象’管理模块<sup>[6]</sup>。

1) 具体有如下 4 个关系数据结构(表 1—表 4)。

(1) 方法基本情况表(方法编号 MethodID, 名称 Methodnm, 所属类简码 ClassID, 特征简述 Brief, 类别代码 SortID)

Keyword: 方法编号 (MethodID)

(2) 方法类别表(类别代码 SortID, 用途简述 ABrief)

Keyword: 类别代码 (SortID)

(3) 方法类对象基本情况表(类简码 ClassID, 类名称 Classnm, 代码存放地址 Savadr, 继承关系 InherFrom, 成员函数 InFun)

Keyword: 类简码 (ClassID)

(4) 成员函数表(成员函数名 InFun, 输入参数 InPara, 输出参数 OutPara, 所属类简码 ClassID, 函数简述 FunBrief)

Keyword: 成员函数名 (InFun)

2) 数据表内容如下:(不完全列举仅举例说明)

表 1 方法基本情况表

| 方法编号     | 名 称         | 类简码         | 特征简述        | 类别代码   |
|----------|-------------|-------------|-------------|--------|
| MethodID | Methodnm    | ClassID     | Brief       | SortID |
| Char 5   | Varchar(20) | Varchar(10) | Varchar(40) | Char 2 |
| 00010    | 简单相关分析      | SRA         | 两变量之间的线性相关性 | aa     |
| 00011    | 一元线性回归      | SLR         | 两变量之间的线性方程  | aa     |
| 00012    | 回归预测        | RE          | 利用回归方程来预测   | bb     |
| 00013    | 线性插值        | LI          | 线性插值方法      | ee     |
| 00014    | 分布查表法       | RC          | 各种分布值及检验的查找 | ee     |
| 00015    | 矩阵运算        | MC          | 矩阵的加减乘转置及求逆 | ff     |

表 2 方法用途表

| 类别代码 (SortID) | 用途简述 (ABrief) |
|---------------|---------------|
| Char 2        | Varchar(20)   |
| Aa            | 分析            |
| Bb            | 预测            |
| Cc            | 规划            |
| Dd            | 网络及图          |
| Ee            | 辅助方法          |
| Ff            | 基础运算          |

表 3 方法类对象基本情况表

| 类简码         | 类名称         | 代码存放地址                 | 继承关系        | 成员函数          |
|-------------|-------------|------------------------|-------------|---------------|
| Class ID    | Classnm     | Savadr                 | Inher From  | InFun         |
| Varchar(10) | Varchar(20) | Varchar(40)            | Varchar(10) | Varchar(30)   |
| SRA         | SimRelAna   | E:\Java\Math\Sra.class | TI          | SRA, getgrade |
| SLR         | SimLinReg   | E:\Java\math\Slr.class | SRA         | SLR, check    |

表 4 成员函数表(若函数名与类简码相同则表示为该类的构造函数)

| 成员函数名       | 输入参数及类型           | 输出参数及类型     | 所属类简码       | 函数简述          |
|-------------|-------------------|-------------|-------------|---------------|
| InFun       | InPara            | OutPara     | LAID        | FunBrief      |
| Varchar(30) | Varchar(30)       | Varchar(20) | Varchar(10) | Varchar(40)   |
| SRA         | Double[ ]<br>[ ]m | Void        | SRA         | 简单相关分析中求解相关系数 |
| getgrade    | null              | Sting grade | SRA         | 简单相关分析中得到相关程度 |

### 3 面向对象的模型库中重用方法库

模型库中的方法重用可以归结为方法的检索和提取以实现模型对象的生成和修改。生成一个模型时,模型库管理系统从用户问题出发根据方法索引号通过‘方法库管理系统’找到方法类以及所要求的输入参数从而生成所需模型;修改一个已建模型时只需要修改这个模型所包括的方法索引号,模型就可以进行相应改变,但方法本身的程序代码并没有改变。另一方面,对于DSS的系统管理员来说,在需要具体的模型或方法的程序代码时也可以通过方法索引编号方便地找到源代码,进行编辑。

### 4 结束语

从支持模型库的方法库的可重用性出发,将方法的可重用性分成了程序代码和方法对象两个层面上的可重用性,尝试着使用面向对象的OOP和OOM方法,

根据具体算法本身的逻辑,提出了具体的方法库的‘类层次’结构和用于管理方法的关系数据模型;来达到建立可重用体系结构的目的。

#### 参考文献:

- [1] 贾洞,杨晓峰. 面向对象技术与软件重用[J]. 计算机与现代化. 1997,1:4-7.
- [2] JMAES MARTIN. Principles Of Object-Oriented analysis and Design [M]. USA. New Jersey: PTR Prentice-Hall,1993.
- [3] 陈晓红. 决策支持系统理论和应用[M]. 北京:清华大学出版社,2000.
- [4] 黄明和,滕少华,黄育潜. 面向对象开发中的继承机制与可重用性[J]. 计算机与现代化. 1996,3:1-4.
- [5] Jacquie Barker. Java 面向对象编程指南[M]. 北京:电子工业出版社,2001.
- [6] 王青,张为民,蔡建平. 一种面向对象的库管理系统的模型[J]. 软件学报,1997,8(5):391-396.

## Structure of Recycle for Methods Base in DSS

LI Yong, XIAO Zhi, CHEN Ling

(College of Economy Business Administration, Chongqing University, Chongqing 400044, China)

**Abstract:** In practice, it is a huge project to exploit DSS and very difficult to control expense of exploiting DSS, raise efficiency, and to ensure software's quality and reliability. Then a perfect methods base is required. Recycling methods base can solve questions above. This kind of recycle contains two respects: recycle of program codes and recycle methods in building model. Based the logic structure of arithmetic and using the technology of OOM/OOP, the recycle of program codes of methods is established. The structure of methods class library's arrangement and their relation map are gotten, at the same time relation data model manage methods are used in order to recycle methods in models base.

**Key words:** Decision Support System; object oriented; method class library; the structure of classes arrangement

(责任编辑 刘道芬)