

# 有关大型信息系统的面向对象开发方法研究

王 涛 (浙江财经学院信息系 杭州 310012)

**摘要:**近年来,由于面向对象系统开发方法在软件复用性方面的优势,这一方法已经逐渐成为计算机软件系统开发方法的主流。但现有的面向对象软件工程文献大量地是进行有关理论和概念的讨论,少量文献涉及以小型系统为背景的面向对象系统开发方法的讨论。本文以大型信息系统为背景,结合作者多年来开发大型信息系统的实践经验,对基于大型信息系统的面向对象系统开发方法进行了一些研究。

**关键词:**面向对象 大型信息系统 OOA OOD

## 1. 问题提出

本世纪70年代,软件开发理论发生了一次大的变革,诞生了软件工程概念即所谓结构化系统开发方法。这种基于模型的结构化软件工程方法蕴涵着一个重要的概念:事务分离原则(Principle of Separation of Concerns),即分析阶段建立系统的逻辑模型,设计阶段建立系统的物理模型。二十年来,软件工程思想对整个计算机科学和计算机工程领域产生了重大影响。

应用基于模型的结构化软件工程方法开发大型信息系统所面临的主要问题是开发时间长,维护难度大,由此而导致系统开发和维护的成本难以控制,甚至最终使得系统无法维持。有经验的软件系统开发人员都知道,最困扰他们的问题是软件系统的维护,包括系统开发过程中的维护和系统运行过程中的维护。系统维护的焦点就是软件的复用性问题,复用包括:系统分析和系统设计过程中文档的复用、模块(程序)的复用,等等。面向对象的软件工程方法较好地解决了软件复用性问题。但现有的面向对象软件工程文献大量的的是进行有关理论和概念的讨论,少量文献涉及以小型信息系统为背景的面向对象系统开发方法的讨论。而当今大型信息系统开发需求日益增长,所以有必要进行基于大型信息系统的面向对象开发方法研究。

## 2. 面向对象的系统分析(OOA)

与结构化系统开发方法相似,面向对象的系统开发方法也要经过系统分析、系统设计、系统实施三个主要阶段。对于大型信息系统,在开发初期(即系统分析开始以前)需要进行总体方案分析(或称较详尽的可行性分析),这是因为:

(1)对于复杂的大型信息系统,开发初期无法把握系统的目标、规模等问题,必须进行总体方案分析加以明确;

(2)没有独立的总体方案分析阶段,就有可能忽视大型信息系统各组成部分(或称子系统)之间的关系,因此

而导致系统效率降低;

(3)在独立存在的总体方案分析阶段,由于初步明确了大型信息系统的目标和规模,则可制定较合理的系统投资计划,减少系统开发早期投资的盲目性。

大型信息系统总体方案分析阶段的主要工作内容有:

- ①初步的系统调查;
- ②分析和确定系统的目标;
- ③分析系统的基本功能以及系统的组成部分(或称子系统);
- ④分析系统的环境并确定系统的界面(范围);
- ⑤分析和描述系统各组成部分(或称子系统)的关系;
- ⑥估算系统的数据存储总量;
- ⑦拟定计算机硬件及系统软件的配置草案;
- ⑧系统目标的可行性分析;
- ⑨制定系统开发的总进度表;
- ⑩拟定系统开发的投资计划、开发人员安排计划以及系统维护人员培训计划。

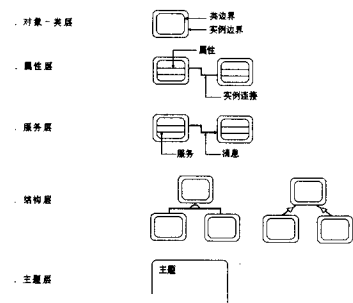


图1 OOA模型的结构

在大型信息系统的总体方案分析阶段,对于系统各组成部分(子系统)的划分及其关系的描述,结构化自顶向下的方法仍不失为一种行之有效的方法用面向对象方法的术语即问题论域(Problem Domain)的主题(Subject)分解。面向对象分析(Object-Oriented Analysis, OOA)模型一般分为5个层次(如图1所示):

而问题论域、主题、对象三者之间的关系则如图2所示。

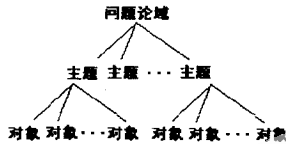


图2 问题论域、主题、对象三者之间的关系

大型信息系统各组成部分的层次关系的描述往往无法用图2的三层关系来完成。对于大型信息系统各组成部分的描述可以将图2变化为图3。

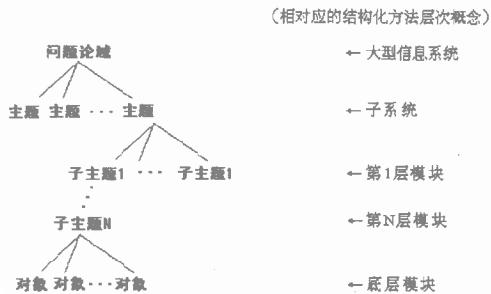


图3 大型信息系统各组成部分的层次关系

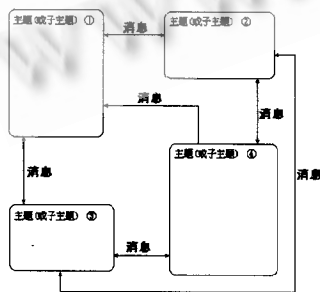


图4 同层主题(子主题  $i, i=1,2,\dots,N$ )之间的关系

应用结构化软件工程方法,我们可以描述大型信息系统各组成部分较为复杂的层次关系(如图3所示)。同层主题(子主题  $i, i=1,2,\dots,N$ )之间的关系则可用OOA的方法描述(如图4所示)。

根据大型信息系统的总体方案分析阶段确定的系统目标、规模、各组成部分及其关系、运行环境等内容,我们便可以进入大型信息系统的面向对象分析(OOA)阶段。在大型信息系统适当地分层后,OOA阶段的主要工作就可以界定在第N层子主题范围(如图3所示),即对一系列第N层子主题的每个对象及其关系进行详尽的描述,OOA内容如图1所示。在第N层子主题发现和标识合适的对象的方法如图5所示,其中LIA是基于语言的信息分析(Linguistic-based Information Analysis),三视图模型是3种结构化系统分析工具:数据流程图、实体-关系图和状态-迁移图。图5描述了应用LIA和三视图模型发现并且确定第N层子主题的对象,从而建立OOA模型。如图1所示,OOA模型由对象-类、属性、服务、结构和主题5个层次组成,属性和服务封装在对象里,结构描述对象与类的关系,若干对象-类及其结构组成一个主题,应用问题论域则由若干主题组成。由于属性(数据)和服务(功能)封装在对象(模块)内,“对象”在OOA中的独立性比“模块”在结构化模型中的独立性更强。如OOA中的一个对象的数据变化而引起的功能变化并不影响其他对象的数据或功能的变化;而结构化模型的数据流程图中的某数据(存储)的变化会引起与其相关的各处理逻辑(功能)的变化。所以,OOA中“对象”的复用性要比结构化模型中“模块”的复用性好得多。

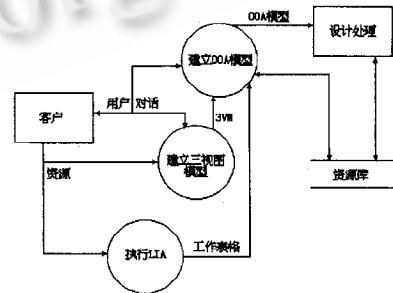


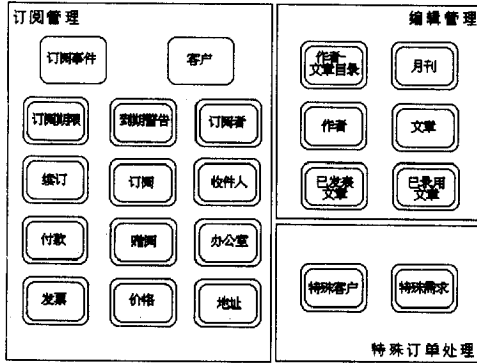
图5 发现对象技术概要

在这里给出一个“杂志管理系统”的例子来说明OOA模型的结构(见下列图示)。

杂志管理系统

.主题层(三个主题:订阅管理、编辑管理、特殊订单

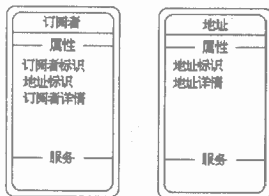
处理)



类及对象层(举例)



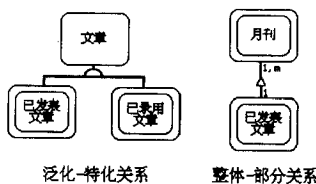
属性层(举例)



服务层(举例)



结构层(举例)



结构层主要描述了类与对象的关系。由上图“泛化

-特化关系”可知,类包含了所属对象的一些共性,因此,“特化对象”可以继承“泛化类”的共性。结构层从另一层面反映了面向对象方法软件复用性的特点。对象之间的关系由“消息”来描述,标识和描述对象之间通信的工具是事件-响应对象交互(EROI)图(如图6所示),它是由电信系统中使用的 Fence-Post 图演变来的。

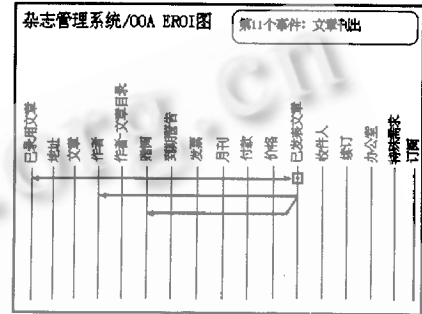


图6 杂志管理系统/OOA EROI图(举例)

某些对象可能在不同的主题中同时出现,根据复用原则,我们在系统优化过程中,将这些对象唯一地确定在某一主题内,系统的其他主题则复用(共享)这些对象。在系统分析阶段通过上述5个层次结构及其关系(EROI图)的描述,较为完整地建立了OOA模型。

3. 面向对象的系统设计(OOD)

有了OOA模型这一基础,大型信息系统的开发便可进入系统设计阶段(Object-Oriented Design, OOD)。OOD的特点之一是直接在OOA模型的基础上进行设计层面的细化工作。OOD的主要工作是对OOA模型(5个层次)在问题论域部分、人机交互部分、任务管理部分和数据库管理部分这四个层面进行详细设计(如图7所示)。

(1)OOD的问题论域部分。OOD的问题论域部分是在主题层描述所包含的对象之间的关系。现今开发的大型信息系统一般以广域网作为其物理环境(如图8所示),应用问题论域的划分往往会以不同的站点作为其第一层主题。因此,大型信息系统OOD的问题论域部分需要分别讨论和描述各层主题之间的关系及最底层主题内对象之间的关系。对于以广域网作为其物理环境的大型信息系统,讨论和描述第一层主题(各站点)之间的关系则显得尤为重要。广域网各站点之间的关系主要有两个层面:数据层面和通信层面。广域网各站点之间数据层面的关系表现为各站点之间所传递的数据的内容(即各站点有关对象之间所传递的“消息”);通信层面的关系表

现为各站点首先需创建有关“通信对象”,规定各站点之间的信息传递方式(如通信设备、网络通信协议、数据结构等),由各站点的“通信对象”实现相互间的“消息”传递。

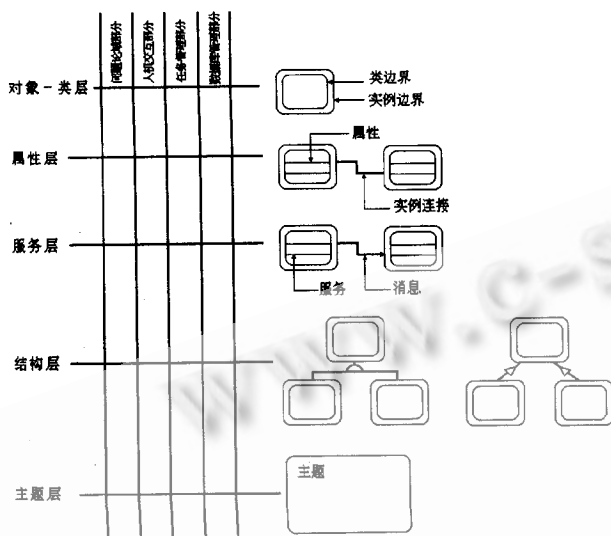


图7 OOD模型的总体结构

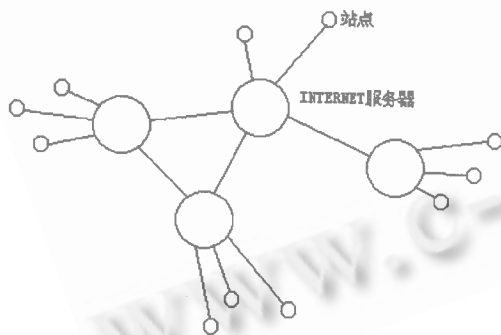


图8 基于INTERNET的大型信息系统示意图

(2)OOD的人机交互部分。OOD的人机交互部分工作主要是系统(对象级)用户界面设计,内容包括:菜单设计(菜单结构、菜单形式(文字的或图标的)等);窗口设计(操作设计、平面设计等);输入/输出设备(或界面)设计,等等。

(3)OOD的任务管理部分。OOD的任务管理部分的工作主要是建立一些新的类(或对象),这些类(或对象)负责处理操作系统级的并发问题、中断、调度以及其他与特定平台有关的问题。对于以广域网为其物理环境的大型信息系统,OOD的任务管理部分不仅要建立操作系统级的有关控制的类(或对象),还要建立广域网上通信级的有关控制的类(或对象),这些通信级的类(或对象)负责处理广域网上各站点信息的输入/输出、输入/输出数据(或文件)的格式及其格式转换、数据库引擎等问题。

(4)OOD的数据库管理部分。OOD的数据库管理部分的工作以OOA模型的属性层内容为基础,将数据库技术从系统的其他部分中分离出来,这部分内容包括:表(Table)、SQL及各种数据管理和备份工作,等等。对没有存储数据需求的系统,则可能不需要进行数据库管理部分的工作。

#### 4. 结束语

在系统实现阶段,今天的应用面向对象方法的系统开发者比当年(八十年代)面向对象理论的初创者要幸运得多,因为现在已经有了技术相当成熟的面向对象编程语言(或工具),如 Visual C、Visual Basic、PowerBuilder、Delphi,等等。大型信息系统的编程人员以OOA-OOD模型为基础,选择应用适当的面向对象编程工具,就可以高效率地完成系统编程工作,最终实现大型信息系统。

知名的面向对象理论学者 E. Yourdon 教授说得很好:“我们不应期望在第一次用面向对象方法开发系统时便取得很大收益。一般情况下,用面向对象方法开发系统的收益在第二、第三个项目以后才会体现出来。”Yourdon 教授在这里强调的是面向对象方法的最大好处是软件(包括 OOA、OOD 文档及程序)的可复用性。

对于大型信息系统而言,开发周期较长,涉及范围较广,开发过程中就存在大量的维护工作,系统实现后维护工作更是漫长而繁重。由于系统维护的成本和效率与软件复用性密切相关,因此用面向对象方法开发大型信息系统就系统维护这一个方面的好处便不言而喻。所以,应用面向对象方法开发大型信息系统将会成为这一领域的趋势。

#### 参考文献

- [1] P. Coad & E. Yourdon. Object - Oriented Analysis. Englewood Cliffs, NJ:Prentice Hall, 1990
- [2] P. Coad & E. Yourdon. Object - Oriented Design. Englewood Cliffs, NJ:Prentice Hall, 1991
- [3] 王涛. 基于广域网的管理信息系统开发方法研究. 管理工程学报, 1998, Vol. 12, No. 4: 29 - 34

(来稿时间:1999年9月)