

# 面向对象方法的无缝性研究

Research on Seamless Object-Oriented Methodology

余永红 徐洁磐 马 恕

(南京大学计算机科学与技术系 南京210093)(中央财经大学信息管理系 北京100008)

**Abstract** This paper discusses the importance of seamlessness of object-oriented methods, it mainly analyzes the seams existing in object-oriented methodology, and presents the methods which can pieces up the seams. The solution of seamlessness problem can make a great progress in leading object-oriented methodology to its practical applications.

**Keywords** Object-oriented analysis, Object-oriented design, Object-oriented implementation, Seamlessness

## 1 引言

软件系统从分析、设计到实现都使用统一的表示范式,以实现由一个阶段向另一个阶段的无缝性转换是软件系统开发人员的一个共同需求。尽管面向对象技术的流行产生了许多的面向对象方法<sup>[1~5]</sup>,这些面向对象方法理论上可满足软件系统的分析、设计到实现都使用统一的表示范式并实现由一个阶段向另一个阶段的无缝性转换,然而实际情况并非如此,由于面向对象方法是一种面向问题域的方法,因此把采用该方法分析、设计的结果转向计算机的实现较为困难,使得面向对象方法在大多数情况下只能实现从面向对象分析到面向对象设计的平滑过渡即告结束而无法贯通整个软件生命周期。面向对象方法无论在概念、实现方法和实现工具等方面都存在一定的差异,特别是在从面向对象设计到实现过渡中存在明显的裂缝,这种存在于面向对象方法中的裂缝,极大地阻碍了面向对象方法在实际中的广泛应用。本文主要讨论面向对象方法的无缝性问题并提出解决的办法。

## 2 面向对象方法中裂缝问题的分析

一个完整的现代软件系统都包括持久性数据、挥发性数据与界面数据三部分。虽然理论上面向对象方法可满足软件系统的分析、设计到实现的无缝性转换,但实际情况并非如此,下面从面向对象方法无缝性的三个方面分别进行讨论。

(1)概念上的无缝性:是指从分析、设计到实现阶段中所有涉及的概念都是完全一致的,保证了转换的无缝性。理论上所有面向对象的实现方法所涉及的面向对象概念都是相同的,但由于在系统面向对象模型

中存在不同性质的数据,实际中对这些不同性质数据的面向对象实现需要采用不同的实现方法和实现工具,因此会引起许多概念上如下图所示的差异:

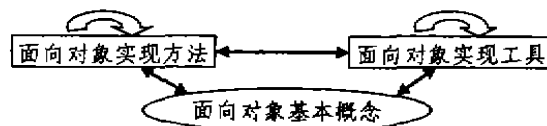


图1 面向对象概念之间可能的差异

图1中的差异具体表现为:这些不同的实现方法之间对面向对象概念的理解可能不同,因此实现方法间在概念上有差异;这些不同的实现工具所支持的面向对象概念可能不同,因此引起实现工具间概念差异;选择的面向对象实现方法可能与支持这种实现方法的实现工具间存在概念上的差异;所选择的面向对象实现方法本身与基本面向对象概念可能存在理解上的差异;所选择的面向对象实现工具所支持的概念与基本面向对象概念可能存在理解上的差异。上述诸多概念上的差异使面向对象分析、设计、实现整个过程失去一致性,进而无法实现整个面向对象方法的概念无缝性。

(2)纵向无缝性:是指从分析到设计,再从设计到实现之间的贯穿整个生命周期的平滑过渡,从一个阶段到另一个阶段的平滑过渡保证了转换的无缝性。目前面向对象方法一般可以实现面向对象分析到面向对象设计的纵向无缝性,但在从面向对象设计过渡到面向对象实现时存在问题,一个完整反映客观世界的软件系统模型中包括持久类、挥发类和界面类,目前面向对象方法只实现了挥发类的面向对象实现的纵向无缝性,但没有考虑对持久类和界面类的面向对象实现的

纵向无缝性实现。由于持久类、挥发类与界面类的实现方式是不同的,特别是对需要长期保存的持久类的实现,这是难以用面向对象程序设计语言实现的。

因此一般认为面向对象方法只能实现从 OOA 到 OOD 的无缝性过渡即告结束,而无法实现从 OOD 到 OOI 的无缝性过渡,因而无法做到整个软件生命周期内的无缝性。面向对象方法纵向裂缝问题在实际应用中造成了通过 OOA 与 OOD 所构筑的 OO 模型在实际实现时不能发挥充分作用,妨碍了面向对象方法在实际软件开发中的应用。

(3)实现方法间的无缝性:是指不同实现方法之间通信和接口的一致性,不同实现方法之间通信和接口的一致可保证实现方法间的无缝性。虽然目前面向对象方法中只讨论了用面向对象程序设计语言来实现面向对象模型中的挥发类,且实际上对面向对象模型中的持久类与界面类也都有相应的实现方法,但一个面向对象模型用三种不同的实现方法来处理,且这三种不同的处理之间存在不同的通信和接口,因此目前面向对象方法也难以做到实现方法间的无缝性,也影响了面向对象方法在实际中的应用。

既然存在上述裂缝就必然丧失面向对象方法无缝性的某些好处,而为获得这些失去的好处又必需进行弥补,因此需要应用程序不得不进行许多额外的工作。

### 3 面向对象方法裂缝的弥补

基于上述分析,下面从三个方面具体介绍面向对象裂缝的弥补方法:

**面向对象方法概念裂缝的弥补** 对于概念上裂缝的弥补,一般采用简化模型或经过相应概念的转换间接地实现概念上的一致,如对于封装概念,Smalltalk 和 Eiffel 语言可以较好地支持,而 C++ 语言则支持不够;对于合成概念,C++ 语言也不支持,但可以通过定义复合对象来实现;如用关系数据库来实现面向对象持久类,则需要解决面向对象模型向关系模型转换的问题。但这种经概念转换而间接实现的概念无缝性毕竟不是直接完全的概念无缝性,因此会丢失许多原有概念的语义,最好的解决方法是在整个软件开发生命周期中采用一种无论从深度和广度都完全统一的概念表示方法,这样可避免因概念上的裂缝而产生的影响。

**面向对象方法纵向裂缝的弥补** 一般系统分析与设计结果的系统模型中包含有持久类、挥发类与界面类,因此面向对象实现包含三种意义的实现,一是持久类的实现,二是挥发类的实现,三是界面类的实现,若能解决三者的实现问题便可解决面向对象方法纵向裂缝的问题。

(1)持久类的实现方式:一般面向对象数据模型中

持久类的实现大都基于数据库系统,因此实现面向对象模型持久类的方式有二种:

①面向对象数据库的实现方式:它的一个很重要的特征就是将数据定义语言与数据操纵语言融为一体,成为数据库程序设计语言,是面向对象程序的程序设计语言与数据管理能力的结合。面向对象数据库管理系统遵从与面向对象分析与设计相同的概念和方法,从一个阶段向另一个阶段的过度是自然、直接而连贯的,其间并没有出现很大的变化,因此这种实现方式实现了一种理想的从面向对象分析到面向对象设计,再到面向对象实现纵向无缝性表示,应是一种首选的实现方案。

②关系数据库的实现方式:其用户众多、应用基础深厚,人们对关系数据库的信赖程度较高。目前,一种较为可行的方案是用关系数据库系统来实现面向对象模型中的持久类。但关系数据模型和面向对象数据模型之间存在很大的不同,用关系数据库来实现面向对象数据模型需要解决面向对象数据模型向关系数据模型的转换问题,因此这种实现方式又可以具体分为:

·转换工具方案:是将 OO 模型通过一种工具转换成关系模型,这种方案较为实用,对不熟悉 OODB 者尤为适合,目前这种工具较为著名的如 Rational 公司的 ROSE。

·关系数据库的对象化方案:近年来,众多厂商均在关系模型之上增加面向对象功能从而出现了对象-关系数据库,其著名系统如 ORACLE8。

·转换程序方案:对于小型的系统可以采用较为简单的方法,即编制一个由 OO 模型到关系模型的转换程序,此种方法灵活实用,能适应多种不同差异的需求。

(2)挥发类的实现方式:面向对象挥发类具有过程抽象、数据抽象、封装、继承和三种组织方法中的对象与属性、类与成员、整体与部分特征,同时由于挥发类的表现方式大都是程序中的内存结构,因此决定了其实现方式非常适合用程序设计语言通过编程来实现。由于面向对象语言(如 C++, Smalltalk 和 Eiffel 等)直接支持过程抽象、数据抽象、封装、继承和三种组织方法中的对象与属性、类与成员两种,虽然不显式地支持整体与部分,但可通过复合对象的表示来间接地支持,因此面向对象程序设计语言可用来直接进行面向对象模型中挥发类的实现。

(3)界面类的实现方式:界面类是一种特殊的挥发类,其实现不同于一般挥发类。过去用的界面设计和实现工具主要是 Basic、Pascal 和 C 等第三代语言,使用这些工具进行界面设计完全是以编写程序代码和过程

性的方式实现的,故很难满足界面类实现的要求。现代软件系统的界面设计大多采用计算机第四代语言,如 Visual Basic、Power Builder、SQL Windows、Delphi 和跨平台语言 Java 等,这些面向对象的应用程序开发系

统是事件和消息驱动型的,提供了丰富的系统预定义 GUI 类和类函数,开发人员可方便地使用这些类和类函数创建应用对象。除此以外,开发人员还可以创建自己的类,使它符合自己的特殊需要。

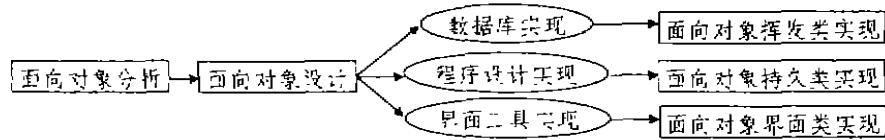


图2 组合实现面向对象方法纵向无缝性的方式

上述三种面向对象的实现方式可以解决面向对象方法纵向裂缝的问题,而且从图2中可以看出,可以采用组合实现方式实现从分析、设计到实现的面向对象方法的纵向无缝性,实际中我们也采用这种组合实现方式进行了许多实用系统的软件开发,因此不存在一般认为的面向对象方法只能实现从 OOA 到 OOD 的平滑过渡即告结束,而无法贯通整个软件生命周期的问题。

**面向对象方法实现方法间裂缝的弥补** 虽然面向对象方法可以组合实现方式实现由分析、设计到实现的纵向无缝性,但对一个面向对象数据模型同时采用三种不同的实现方式和工具,可能因这些不同的实现方式和工具间或多或少地存在差异,毕竟会造成实现方式和工具间面向对象概念的不一致与实现异域间接口的困难,因此,为实现面向对象方法真正意义上的无缝性,还需要解决不同实现方法和工具间存在的裂缝问题,对面向对象方法实现方法间裂缝的弥补主要是消除不同实现方法间通信和接口的困难。

为此,较为理想的解决方法是基于某种统一的、可同时实现面向对象持久类、挥发类和界面类的集成系统。这种集成的面向对象实现系统既需要有面向对象数据库系统的功能,又需要支持面向对象的程序设计,同时还必须支持系统界面的生成,以完全实现面向对象方法的无缝性,见图3。

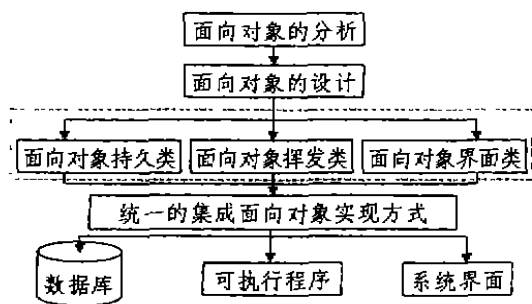


图3 统一的面向对象实现方式模型

从对三种面向对象不同实现方式和工具的具体分析中,我们可以找到一些解决方法:

(1)从对基于面向对象数据库管理系统的面向对象持久类实现方式中可以了解到,由于面向对象数据库管理系统一般有三种类型的实现模式:一是对关系数据库管理系统进行面向对象扩展,二是开发纯面向对象数据库管理系统,三是对面向对象程序设计语言的持久化,其中面向对象数据库管理系统的基于面向对象程序设计语言持久化的实现模式,为一个系统既支持持久类的实现又支持挥发类的实现奠定了基础。

(2)从对基于面向对象程序设计语言的面向对象挥发类实现方式中可以了解到,由于面向对象程序设计语言与面向对象挥发类具有许多相同的特性,因此面向对象挥发类的实现一般都是采用面向对象程序设计语言编程实现。

(3)从对基于面向对象界面开发工具的面向对象界面类实现方式中可以了解到,面向对象界面类的实现大都采用支持面向对象的界面开发工具实现,而面向对象的界面开发工具也可以采用面向对象程序设计语言来支持界面类的编程实现。

上述三种实现方式都可以基于某种面向对象的程序设计语言来实现,为建立一种统一的集成面向对象实现系统奠定了基础。一般一个统一集成的面向对象实现系统可按下述策略来构造:

(1)选择某种适合于进行持久化的面向对象程序设计语言,增加其对持久性数据的管理部分,用来构筑一个面向对象数据库系统以实现面向对象数据模型中的持久类管理。

(2)面向对象程序设计语言可以有效地进行面向对象挥发类的实现。

(3)选择的面向对象程序设计语言也必需适合于进行面向对象界面类的编程实现。

这种面向对象实现系统由于采用一种统一集成的方式实现面向对象模型中的所有内容,因此不会出现不同实现方法和工具间的裂缝,是解决不同面向对象

实现方法和工具间裂缝的有效方法。虽然这类系统目前并不常见,但这种系统无疑是目前面向对象实现中极为需要的工具,实际上已经存在这类集成的面向对象实现系统, ObjectStore 就是一例,它是 OODB 与 C++ 的统一结合体,又有定义多种界面的能力。

总之要全面实现面向对象方法的无缝性,则应保证软件生命周期所有阶段涉及的面向对象的基本概念应完全一致,只有这样才能实现从分析到设计,再到实现的自然和连贯性,而不至于因为概念上的差异而导致较大的变化,概念的一致保证了转换的无缝性,另外在实现方法和实现工具的选择上应注意尽可能地选择一种可贯穿整个软件生命周期的软件开发工具,以实现从一个阶段向另一个阶段的无缝性转换。

#### 参 考 文 献

1 Coad P, Yourdon E. Object-Oriented Analysis. 2nd edi-

tion, Prentice-Hall, 1991

2 Coad P, Yourdon E. Object-Oriented Design. Prentice-Hall, 1991

3 Jacobson I. Object-Oriented Software Engineering, A Use Case Driven Approach. New York: Adison-Wesley Publishing Company, 1992

4 Boach G. Object-Oriented Analysis and Design with Applications. Benjamin/Cummings Publishing Company, 1994

5 Rumbaugh J. Object-Oriented Modeling and Design[M]. Prentice-Hall, 1991

6 Pressman R S. Software Engineering, A Practitioner's Approach. McGraw-Hill, 1997

7 Norman R J. Object-Oriented Analysis and Design. 清华大学出版社, 1997

8 Rao B A. Object-Oriented Database: Technology, Application, Products. McGraw-Hill, 1994

## 第八届全国 Petri 网学术年会征文通知

由中国计算机学会 Petri 网专业委员会主办的第八届全国 Petri 网学术年会将于 2001 年 8 月下旬在太原召开(太原重型机械学院承办)。会议将对 Petri 网理论及其在各个领域中的应用开展广泛深入的讨论。现发出征文通知如下:

### 一、征文范围

Petri 网理论研究;

Petri 网工具开发;

Petri 网在计算机科学技术、自动化技术及其它各个领域中的应用研究;相关的并发模型研究。

### 二、征文要求

1 凡在正式刊物或其它学术会议上发表过的论文不再征用。

2 投送论文须给出中、英文标题、摘要、关键词;无论录用与否概不退稿,请作者自留底稿。

3 会议论文集以《计算机科学》增刊出版,被录用的论文要求用 Word 排版、绘图,寄出的软盘应与稿件的内容一致。

4 投送论文请写明作者姓名、通信地址(包括邮政编码)、电话、E-mail 以便联系。

### 三、重要日期

截稿日期:2001 年 3 月 10 日

通知录用或修改日期:2001 年 4 月 20 日

稿件(软盘)返回日期:2001 年 5 月 20 日

### 四、论文请寄

姚淑珍 邮 编:100083 通信地址:北京航空航天大学 602 号信箱 电 话:010-82317602(办公室)

中国计算机学会 Petri 网专业委员会

2000 年 11 月