



78-80,62

面向对象 软件开发 结构化法

计算机科学1994Vol. 21No. 6

面向对象开发方法与结构化方法的比较研究

周广声 李卫国

TP311.S2

(上海海运学院计算机科学系 上海200135)

摘要 This paper discusses the object-oriented system development method, meanwhile, describes the Object-oriented characteristics of structured methods. Finally, gives an approach transiting from the structure method to the Object-oriented method.

九十年代,面向对象的技术已开始为许多软件开发者和用户所理解和接受。不少软件都同时出现了面向对象的倾向,其中包括程序设计语言、用户界面、数据库和操作系统。面向对象的技术与七十年代的结构化程序设计一样,其目的都是改进软件的结构、维护及使用,但前者确实有助于控制软件开发中不断增加的复杂性和费用,从根本上改变程序员的工作方法,加快了生产应用程序的速度。另外,面向对象技术还能增强应用程序的功能,使端点用户可以通过各种计算机平台,实现对现有的数据类型和扩展数据类型的访问。

目前已有了一些面向对象的开发工具,开始用来简化面向对象的设计、编码和调试,而且出现了用于构造面向对象用户界面的开发环境,使得交互式应用程序可从预定义的类库中快速生成,大大地加快了开发速度。

结构化分析方法是目前系统分析的主要方法之一,从本质上说,面向对象的分析方法也可以说属于结构化分析方法,只不过两者在分析过程中的侧重点不同。本文就面向对象的开发方法与结构化开发方法进行系统的比较研究,并提出了向面向对象的系统开发方法过渡的途径。

一、面向对象的方法

表1列出了四种面向对象的方法:HOOD、OOS-D、OOSA 和 OOA。表中的 R 代表需求分析;A 代表分析;S 代表规范说明;D 代表设计;I 代表实现。下面对这些方法逐一进行讨论。

1. HOOD (Hierarchical Object-Oriented Design, 分层式面向对象设计)

HOOD 在面向对象(OO)和各种特性方面有着很好的表现。尽管对早期分析阶段支持较少,但引入结构化分析和设计之后,HOOD 仍旧明显地支持对象建模,HOOD 以层次方式对对象建模,其中父对象和子对象之间有继承的特性。HOOD 着重于对象接口说明和封装。系统的对象网络靠消息来通信,它的建立依赖于事件/消息。在此 HOOD 使用了 Book 的行为对象和被动对象。

HOOD 支持类的概念,但是继承性不太明显,重用也就乏力。该方法对设计阶段有较好的支持,并且提供了向 Ada 语言转换的功能。总的说来,HOOD 拥有 OO 的许多特性。但由于它是实时系统的设计方法,因此数据说明和继承机制没有予以足够重视。

2. OOSD (Object-Oriented System Development, 面向对象的系统开发)

OOSD 方法假设在分析阶段已产生了明确的对象,它为对象的类和继承的管理提供详细的说明。对象之间的通信是靠消息/事件,该方法对接口描述和封装说明是详尽的。系统模型被建立成一组顺序执行的层次表(或使用 Yourdon 结构表说明)或建立成进程的异步网络。

OOSD 对分析没有做任何工作,其覆盖范围只局限于设计阶段。

图1是 OOSD 的一个应用模型,图中,Cinema、Film、Booking、Viedo-copy 作为对象来描述而 Allocation Control、Put-results 作为方法来描述。

表1

方法	抽象	类	继承	封装	覆盖范围 (R-A-S-D-I)
HOOD	Y	Y	部分的	Y	D-I
OOSD	Y	Y	Y	Y	S-D-I
OOSA	Y	部分的	—	—	R-A- - -
OOA	Y	Y	Y	—	R-A- - -

计算机科学

5

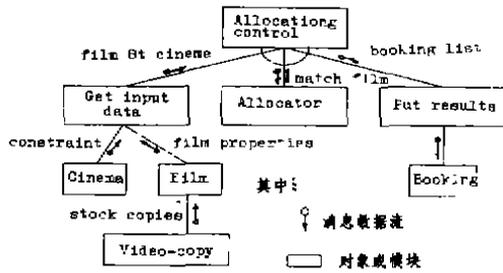


图1 OOSD应用模型

3. OOSA (Object-oriented System Analysis, 面向对象的系统分析)

该方法由 Shaler 和 Mellor 提出,它对对象的标识和分析给出了许多启发式的观点,有助于数据的抽象和对象模型的描述。OOSA 把祖先系归属到数据模型中,它的许多可取之处是不同于数据库中的 E-R 模型。该方法通过子类建立了一个对象关系网络,为每一个对象构建一状态转换说明,用数据流建立功能模型。

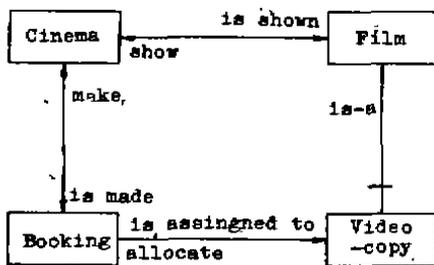


图2 OOSA对象模型

OOSA 通过连接数据模型和行为,描述功能数据流及其对象的状态转换模型,以组合为行为-数据模型。该方法缺乏对继承的支持,类是可支持的,但是仅仅支持对象属性的继承,而不支持服务(行为)继承,也不明确支持重用。该方法只限于前一阶段(分析),对设计阶段没有定义。图2是一个 OOSA 对象模型的实例。

4. OOA (Object-Oriented Analysis, 面向对象的分析)

OOA 覆盖了所有面向对象的概念,具有分类和继承机制,由于它只是一种分析方法,因而有关设计阶段的观点很薄弱,在一个大系统中分组描述对象的结构层次,有助于数据的抽象。对象之间有三种连接方法:关系连接、类继承及消息的传送。Coad 和

Yourdon 把最后的规范说明的复杂性分布在不同层次:主题、结构、属性、服务的标识上。该方法在封装和对象接口方面不象 HOOD 和 OOSD 那样做得详细;但总的来说,很适合大多数的 OO 标准规范。图3 是一个 OOA 对象模型。

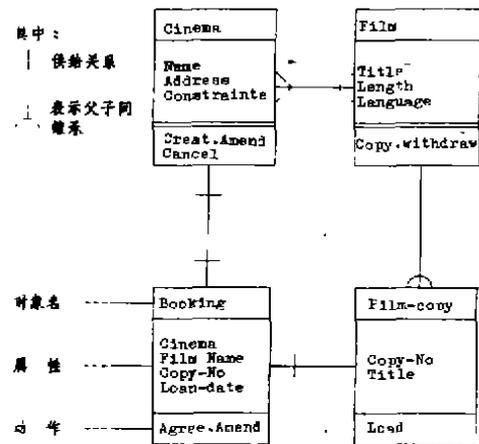


图3 OOA对象模型

二、系统开发方法的面向对象特性

表2列出了四种系统开发方法:信息工程法(IE)、结构化分析/设计法(SASD)、杰克逊系统开发法(JSD)和尼杰森信息分析法(NIAM),表3对这四种系统开发方法的面向对象特性进行了比较。下面就这些方法逐一进行讨论。

表2

方法	功能 进程	数据 关系	事件 次序	覆盖 范围 R-A-S-D-I	应用领域
IE	Y	Y	Y	R-A-S-D	信息系统
SASD	Y	N	Y	A-S-D	信息系统
JSD	N	Y	Y	A-S-D-I	信息/ 实时系统
NIAM	Y	Y	N	A-S-D	信息系统

表3 四种系统开发方法的面向对象特性

方法	对象模型	数据/行为	封装	类型/实例	类
IE	可能	N	N	Y	N
SASD	Y	N	N	N	N
JSD	Y	Y	Y	Y	N
NIAM	可能	可能	N	Y	Y

1. IE (Information Engineering, 信息工程方法)

数据模型是 IE 的重要组成部分, 着重于系统成分的数据的对象模型。功能说明使用了进程的依赖性和动作表, 它与数据模型分离, 而不提倡使用公共数据和控制说明。本方法提供了对实体的功能调用。状态转换表明确地把动作与实体联系起来, 具有 OO 说明的部分特征。这里有类—实例的概念, IE 着重而向对象的事务处理的概念模型。数据模型由实体和关系组成, 给出一个系统的静态网络说明。由于分析过程与数据分离及功能的分解, 因此意味着 IE 不能作为真正的面向对象的方法。

2. SASD (Structured Analysis/Structured Design, 结构化分析/设计)

SASD 运用自顶向下的原则进行功能分解, 通过数据流连接的处理网络来分析整个系统。该方法采用基本功能的聚集及耦合的原则。功能聚集指将具有相同目标的动作分为一组, 而耦合指两个系统成分之间的联系。作为 SASD, 要求其功能要高内聚 (越具有共同的目标越好) 和低耦合 (系统成分之间独立性越高越好)。数据流图把系统表示为功能通信网, 且可以嵌套使用。该方法一点也不支持 OO 的观点, 数据与处理是分离的。图4是 SASD 方法的数据流程图 (DFD)

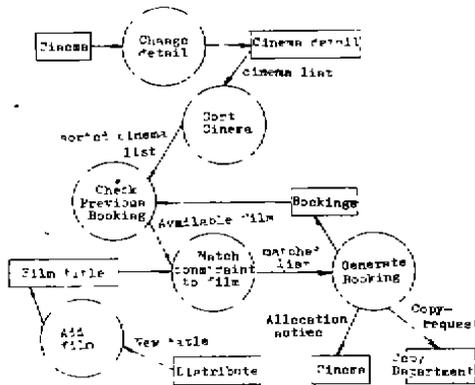


图4 SASD 数据流图 (DFD)

3. JSD (Jackson System Development, 杰克逊系统开发方法)

JSD 是诸多现存软件开发方法之一, 其显著特点是它所产生的软件之高可靠性和易维护性。JSD 的系统模型建立在并发通信的进程网上, 进程的联系通过实体之间通信而实现, 具有类—实例的概念。JSD 不支持类和继承, 系统的控制机制是根据与实体相关联的动作的时序建立的。最新的研究重点是研究包含数据及动作的对象模型的数据分析方法。

JSD 的系统规范说明图显示了一个通信的进程网, 它与对象模型类似。由于 JSD 着重于实体的生命周期, 所以它与 OO 方法很相似, 虽然它并不全部支持 OO 概念, 但它们在方法上有着共同之处。图5是 JSD 规范说明图。

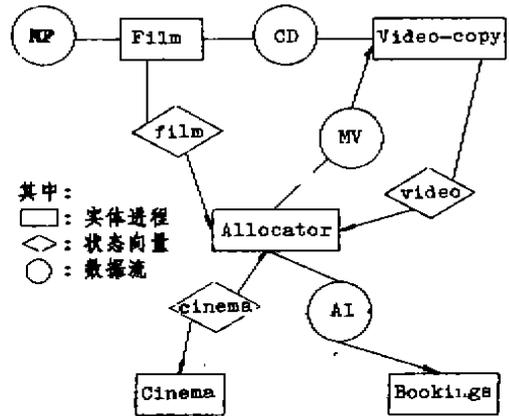


图5 JSD 规范说明图

4. NIAM (Nijssen's Information Analysis Method, 尼杰森信息分析方法)

NIAM 是一种概念模型方法, 在分析的早期阶段集中进行数据规范说明。它基于 ANSI/SPARC 规范, 支持概念模型的数据抽象, 因而支持面向对象的运用。NIAM 在过程分析中加入了语义限制, 从而形成语义模型。它支持类与实例的概念, 就如同类由实体、子类型支持一样。所以说 NIAM 具有 OO 处理过程的一些特点, 但是 NIAM 不支持继承的概念。NIAM 着重于数据模型中与关系紧密结合的带条件限制的处理, 从而大大减少了 NIAM 的 OO 特性。

三、向面向对象的系统开发方法过渡

面向对象开发原则已被用于诸如规范说明、可维护性和软件重用这些难题上。有人认为使用一种特殊的系统开发方法, 用 OO 实现技巧来实现并非难事; OO 设计可从任意规范说明来派生, 这种观点值得争议, 从上述对不同的 OO 方法和非 OO 方法所产生的规范说明的研究可以看到, 这是不实际的。然而数据模型和 OO 规范说明可以结合起来考虑, 可利用一种从结构化方法到面向对象的移植途径来实现。

在当前许多结构化方法中, 仅仅 JSD 具有 OO 模型的大多数特征, 尽管它不支持类和继承。数据模型方法运用了适用于数据结构的规则; 就象 NIAM (下转第62页)

具作为后盾。其次,对过程模型本身的研究尚不成熟。因此,目前的工作一方面集中于对过程的理解,研究过程的体系结构、过程建模方法和支持,以及过程本身的开发、评价和改进。软件工程环境方面,主要还是考虑实现部分环境,完成有限的集成、有限的设施,并力求与别的环境进行通讯,来支持过程指导和过程的半自动化运作,并通过与过程的模拟运作相结合完成过程的改进。

参考资料

- [1] C. J. Tully, Prospects for Future Environments, Introd. to Panel Session, In Proc. of the 9th Intl. Conf. on SE, Monterey, California, Apr. 1987
- [2] E. Fedchak, An Introduction to Software Engineering Environments, In Proc. of the COMPSAC 1986, Chicago, Illinois, USA, Oct. 1986
- [3] M. M. Lehman, Process Models, Process Programs, Programming Support, In Proc. of the 9th Intl. Conf. on Software Engineering, Monterey, California, Apr. 1987
- [4] J. D. Zucker, ALF: Accueil de logiciel futur, In Software Engineering Environments, Volume 3
- [5] A. Legait et al., MASP: A Model for Assisted Software Processes
- [6] N. H. Madhavji, The Process Cycle, Software Engineering Journal, Sep., 1991
- [7] P. H. Feiler et al., Software Process Development and Enactment, Concepts and Definitions, In Proc. of 2nd Intl. Conf. on Software Process, 1993
- [8] J. Lonchamp, A Structured Conceptual and Terminological Framework for Software Process Engineering, In Proc. of 2nd Intl. Conf. on Software Process, 1993
- [9] M. H. Penedo, W. Riddle, Process-sensitive SEE Architecture Workshop Summary, Software Engineering Notes vol 18 no 3
- [10] 麦中凡、张莉,集成CASE的集成模型,软件学报,1994,2。
- [11] 麦中凡、熊璋,兆程序设计与软件过程驱动的软件开发,第三届抗恶劣环境计算机学术会议,1994,5,洛阳。
- [12] Y. Sugiyama et al., OPM: An Object Process Modeling Environment, In Proc. of the 5th Intl. Software Process Workshop, 1990
- [13] Y. Sugiyama et al., Describing Working Environments in OPM, In Proc. of the 5th Intl. Software Process Workshop, 1990
- [14] An Overview of the ALF Project, In Proc. of the 5th Intl. Software Process Workshop, 1990
- [15] Peiwei Mi et al., Process Integration in CASE Environments, IEEE Software, 1992, 3
- [16] R. N. Taylor, et al., Foundations in the ARCADIA Environment Architecture, Software Engineering Notes, 13(5), Feb. 1989
- [17] M. Dowson, ISTAR and the Contractual Approach, In Proc. of 9th ICSE, USA, 1989

(上接第80页)

的语义数据模型,它是一种很有前途的方法。由Coad和Yourdon方法创建的OO规范说明的派生,说明了该方法的进化是有实现可能的。

向面向对象的过渡,主要依赖于系统开发者确信该方法的好处。由于没有一种OO方法包含明确的步骤或模型用于重用系统开发,所以对象模型并不能完全解决重用问题。

面向对象的软件体系结构将在九十年代占据主导地位,向这种新型体系结构的过渡正在进行,现在已出现了面向对象的语言、数据库、界面、操作系统及开发环境,而且新的数据类型、分布式处理、多媒体应用程序以及端点用户计算正在有力地推动着面向对象软件的实现。

主要参考文献

- [1] 马茜等,面向对象的软件设计基础,北京科海培训中心
- [2] Booch, G., Object-Oriented Development, IEEE Trans. Soft. Eng., Vol. 12, No. 2, 1986 pp211-221
- [3] Jackson, M., System Development, Prentice Hall Int. UK, 1983
- [4] Cameron, JR., JSP and JSD, the Jackson Approach to Software Development, IEEE Computer, Society Press, USA, 1988
- [5] A. G. Sureliffe, Object-oriented System Development, Survey of Structured Method, Information and Software Technology, Vol. 33 No. 6, July/Aug. 1991