中国图书馆学报(双月刊)2000 年第 2 期
ZHONGGUO TUSHUGUANXUEBAO

●张玉峰 文燕平

软件资源的组织方法研究*

摘 要 软件资源的组织管理方法与数据资源不同。软件对象具有完整性、抽象性、可复用性、连接和组配性及标准化等特性。软件资源的组织,可采用体系分类、组面分类的方法,也可采用超文本方法,但各有利弊。应尽力寻求多种方法结合的道路,以取长补短。表 1。参考文献 15。

关键词 软件资源 组织方法 体系分类法 组面分类法 超文本法
分类号 G254

ABSTRACT The organization methods of software resources are different form those of data resources. Software objects have the characteristics of integrity, abstractness, reusability, connectivity, combinability and standardization. For software organization, there are such ways as hierarchical classifi cation, facet classification and hypertext linking. We should combine different ways to achieve the best results. 1 tab. 15 refs.

KEY WORDS Software resources. Organization methods. Hierarchical classification. Facet classification. Hypertext.
CLASS NUMBER G254

1 引言

随着软件工程技术和其他信息技术的发展和应用,软件不断更新换代,已构成庞大的企业资产。面对海量的软件资源,如何对它们进行有效的组织、管理和利用呢?面向对象技术提供了模拟人类思维方式的全新的认知方法。它的核心是数据抽象,即用概念的划分与概括和概念的分析与综合的科学方法,来抽象、描述和组织数据对象,对于开发数据密集型领域的应用软件,取得了巨大的成就。但面向对象方法对于软件功能元素的表达和组织不够得力。

可复用构件技术在人们的艰苦探索中诞生,所谓构件是对软件元素(包括数据、功能、系统模型、设计开发过程等)的概括,而分类抽象方法是其核心。它揭示了软件资源内部结构的基本规律。张琪玉教授指出:"将事物概念纳入知识分类体系(事物和学科的分类体系),是对千差万别的事物作系统研究的重要方法,是对各种事物之间的区别和联系从本质上、原理上进行揭示的重要手段,对情报的系统化具有重大价值。"

基于各种技术、条件的成熟,软件工程已摆脱机械的线性组织、线性开发的模式,采用各种科学的描述方法,合理高效地组织和管理软件资源。笔者认为这是新型软件工程技术的基本问题,也是知识信息管理科学的新课题。

软件本身既包含静态的数据、对象和知识,更多的包含动态的行为、控制、关系、通讯、方法和策略,还具有灵活的互操作性,例如浏览、检索、连接、数据交换等。因此,软件资源的组织管理方法与数据资源的组织管理方法是有区别的,正处于百花齐放的探索阶段。本文从当前的软件工程新技术的环境出发,侧重讨论软件资源的分类组织方法和超文本方法及其模型。

2 软件对象及其特性

本文中的软件对象与面向对象技术中的"对象"含义不同。面向对象技术中的"对象",主要描述真实世界中自然的数据对象,是由数据和数据的有关操作组成的实体。而软件对象,主要描述客观事物的活动及其运动规律,称为功能,是计算机等领域专家们精心制作的人工品,是理论与经验的智慧结晶,属于知识范畴。

软件对象的概念随着软件工程等技术的发展而逐步广义化、高层次化,它是指所有用于构造和维护软件系统的单元。可复用构件技术中将提供一定功能或服务的可复用的软件单元称为软件构件,揭示了软件的最有生命力的核心部分。软件对象一般指描述计算处理、数

* 本文系湖北省软科学研究课题成果之一。课题编号:97192410。

据管理、处理对象、连接与约束、控制原则的软件产品乃至软件子系统, 也可以是软件开发过程的描述, 例如需求说明、分析说明和设计说明等, 进而还可以是高层次的抽象信息, 例如软件框架、模型、专家的经验知识、应用领域知识等。

软件对象之间存在着静态和动态关系。静态关系主要指各种语义相关关系和结构关系。动态关系包括过程调用、消息传递(如异步通告消息和异步请求消息)、方法引用、数据共享、远程过程调用等。

当前的应用需求和软件技术要求软件对象应具有以下主要特征: (1) 完整性: 应集合所有的同类相关事务的处理, 可以独立执行; (2) 抽象性: 能表达事物的本质属性, 可实现不同层次的抽象; (3) 可复用性: 能满足人们"一次开发, 多次使用"的要求, 从而大大提高程序开发和效率, 从最早的结构化程序设计中的公共子程序, 到面向对象技术中的可复用对象, 再到面向构件的程序开发中的构件, 都体现了软件对象的可复用性; (4) 连接和组配性: 支持良好的动态与静态连接, 便于用户的复用、卸装、组配和替换, 以提高软件的可重组性; (5) 标准化: 对不同层次、不同形式的表达, 遵循统一的开放标准, 便于实现对象之间的互操作, 从而提高软件系统的开放性。

3 软件资源的分类组织方法

分类是人们的一种思维活动, 是根据事物的异同来区分事物、聚集事物的方法。用分类方法组织软件资源, 能充分表达软件资源的知识体系, 便于软件资源的有效管理和复用。目前, 在各个领域的软件库系统中应用较为广泛的有体系分类法和组面分类法。

3.1 体系分类法(System Classification)

体系分类法又称枚举分类法, 其基本思想是对某一个领域选用一个主题作为分类标准, 运用概念划分与概括方法, 按知识分类的逻辑次序, 从总到分, 层层划分, 逐级展开, 形成一个严格的等级结构(层次结构或树型结构)。同位类(同层类目)之间是一种并列关系, 类目不重复, 不交叉, 并只对应于一个上位类。上位类与下位类之间是一种隶属关系。列类方法(类表)是枚举式的。

软件是描述应用领域的业务活动(即功能)的, 这些业务活动具有较强的层层分解的规律性, 选择业务功能作为分类标准来分类软件是科学的。一些软件对象表示高层次的业务功能(如子系统功能), 可将相同性质的软件归为一类, 该类对象包含一系列有序的互不相同的子任务, 可将表示各种子任务的软件, 划分为若干子类, 直至表达基本功能的软件对象类。同理也适于其他的表示框架、模型等抽象信息的软件对象。

用体系分类法组织的软件库结构, 是一种严格的树型结构, 每一个软件类就是树的一个结点, 这个类下又分出许多小类, 形成这个结点的子结点。父子结点之间的是隶属关系, 父子行为之间是实现关系。由于体系分类是一种严格的概念划分, 因而子类必须穷尽父类, 每个子类结点只能有一个父结点, 且同一级结点(同位类)按功能之间的逻辑次序排列。

例如, 对企业领域的应用软件, 使用体系分类法得到的分类结构如下:

(1) 文档管理: 建立, 排序, 增加, 修改, 删除, 数据查询, 数据集成;

(2) 计算: 帐务处理, 劳资计算, 材料核算, 成本核算, 固定资产核算, 利润核算;

(3) 计划: 计划初算, 计划调整, 测算, 计划制定, 实施;

(4) 统计: 数据收集, 分类统计, 汇总统计;

(5) 决策: 数据挖掘, 信息集成, 综合分析, 决策制定与修改;

(6) 报表处理: 数据收集, 制表, 报表显示, 报表打印;

(7) 知识管理: 知识获取, 知识表示, 知识库管理;

(8) 智能处理: 知识检索, 知识推理, 模式识别, 自然语言处理。

由于软件的灵活复杂的动态特性, 较难处理严格的隶属关系, 例如查询、打印等功能, 可为较多的其他业务功能使用, 不能归于某一上位类, 可以将这些公共子对象单独成类, 并建立与其他对象之间的参照连接关系。一些同一级的软件对象(同位类)之间存在着"使用"关系, 也需建立相应的连接关系。

体系分类法对软件领域进行清晰的高度结构化的划分, 能帮助用户理解软件对象之间的本质关系, 提供向上和向下的自然查找方法。这种分类体系结构的最大特点是继承性, 便于软件信息的共享; 其次, 结构的准确性和严密性, 使得对象的归类易于实现; 而且结构的层次反映了软件对象的抽象层次, 从高层至低层由概括到详细, 便于用户进行扩检和缩检, 提高检全率和检准率。但另一方面, 与其它分类方法比较起来, 该方法过于严格, 需要手工分类, 自动化分类方法正处于探索之中。面向对象技术和构件技术, 解决了大型类层次结构的实现问题, 可以动态维护和扩充。目前, 一些软件系统暂用面向对象技术中的对象模型实现, 尚不能满足软件的全部要求, 而公用对象请求代理结构(CORBA)和 JAVA 为之带来了希望。

3.2　组面分类法(Facet Classification)

组面分类法使用概念的分析与综合的逻辑方法,将主题概念多向成类,从不同专指度、不同角度较全面地表达事物。其基本思想可以从下面三个方面来说明。

(1) 组面的获得。领域分析专家就某一领域进行分析,预先得到能充分表示该领域的具有代表性的重要术语,即分类对象的若干属性或特征,称之为"组面"。组面与组面之间尽可能做到不相关,组面数目依不同的领域而异。

(2) 组面的排列。将得到的组面按照组面所反映领域对象的重要性从左至右进行排列。

(3) 组面的赋值。领域内包含诸多领域对象,将每一个领域对象依据给出的组面进行分解,得到相应的关键词,这些关键词即各个组面的"值"。

组面分类法,允许对同一类事物用多种属性作为分类标准进行划分,分类标准必须是事物的本质属性或有一定检索意义的特征。用组面分类法分类软件对象,主要问题是分类标准的选择和分面数目的确定。

一些专家的组面分类方案如下。CASE 软件开发者选择三个组面: 库信息(Repository)、功能(Functionality)、应用领域(Domain)。Sindre 在 REBOOT 构件库中定义的四个组面为: 抽象(Abstraction)、操作(Operation)、操作对象(Operateson)、依赖关系(Dependencies)。国内青鸟工程 JBCL 中的五组面为: 应用环境(Application Environment)、应用领域(Application Domain)、功能(Functionality)、抽象层次(Level of Abstraction)、表示方法(Representation)。Prieto-Diaz 提出的六组面为: 功能(Function)、对象(Object)、介质(Media)、系统类型(System Types)、功能领域(Function Area)和应用环境(Setting)。

软件对象的分类表示应独立于软件的实现,选用较高层次(如概念层或逻辑层)的描述。组面应反映软件开发者和复用者对软件对象的认识分析和要求,以保证结构体系的质量和软件的最大重用。笔者认为软件对象的以下属性较为重要,可以作为组面选择的候选对象,如功能描述、操作对象、所属系统类型、应用领域、抽象层次和表示方法(如C,Ada,OO,CASE)。具体的组面分类方案随软件领域的实际需要而定。

组面分类法是根据对象的某个或某几个特征来查找对象的,这就要求在建软件库时对库中的对象要尽可能地从多个方面进行详细描述。对象描述的越详细,角度越多,查找对象的准确度就越高,误检和漏检的可能性就越低。但是,要注意防止出现不必要的组面,以避免给分类和使用带来麻烦。

软件的组成分类模式,可用所选定的组面及其组面值组成的二维表来表示。例如UNIX 软件对象的三组面分类模式,如表 1 所示。

表 1　UNIX 软件的一种组面分类方案

操作	操作对象	所属系统
Get	File-names	Line-editor
Put	Identifiers	Text-formatter
Update	Line-number	
Append	Character	
Check	Number	
Detect	Expression	
Locate	Entry	
Search	Declaration	
Evaluate	Line	
Compare	pattern	
Make		
Build		
Start		

从上面的叙述可以总结出组面分类法的以下优点:

(1) 可扩充性好。它是一种动态的、便于扩充的分类体系,既可对组面进行增删、修改,也可向每一个组面增加、更新新的术语。对每个组面的修改不会影响其他组面。

(2) 组面分类灵活。它建立在领域分析的基础上,不同的应用领域可以用不同的组面进行分类,体现了自由分类的思想。

(3) 提供多种查询途径。组面分类表较全面地表达软件对象的主题内容,用户可以根据自己的需要选择若干组面进行查询。

(4) 易于实现。由于采用的是表格形式,可以用关系数据库很方便地实现。

但是,组面分类法并非完美无缺,它没有解决同义词问题及词的语义模糊问题。对同义词不进行处理容易把不相关的软件对象检索出来。与同义词相关的一个问题是"近似匹配",即检索到的软件对象与所需软件对象极为相近但不完全匹配。Prieto-Diaz 提出利用"概念图"对组面值之间的相似性进行定量描述。其基本原理是对近似匹配的软件对象与所需的软件对象根据组面值逐一进行相似性计算,所得结果为两者之间的"相关距离",并分配一个数字来表示,人们根据这些数字就可以进行软件对象的相关性排序。关于词的语义模糊性问题,具体表现在: 对某一个具体的组面值来说,如果仅考虑较少的可能的含意,则会降低检全率;如果考虑另外的无关的含意,则会影响检准率。为此,可以建立一个可控词表,来规范用于软件对象分类的词,检索式只能由词表中的词组成。

4 超文本方法

分类方法应用人类的抽象思维方式, 从知识分类的角度组织软件资源, 实际上, 软件对象之间还存在着其他多种语义相关和互操作关系, 并且不同类型的分布式软件资源所采用的组织方法各异(异构), 如何充分利用软件资源内外的各种关联, 实现最大程度的软件重用、资源共享呢? 一些专家试用超文本方法, 取得了可喜的成果。

超文本是能描述和管理任意复杂结构的信息资源的新技术。它应用人类的抽象思维、形象思维、经验思维等多种方式, 将信息资源组织为一种信息网络结构, 由结点和链接组成。

用超文本方法组织软件资源, 结点表示软件对象或其抽象信息, 链接表达结点之间的关系, 例如调用、服务、通讯、数据库存取、部件交换等动态实现关系, 以及分类、语义相关等静态逻辑关系。结点之间的链接可依需要任意设置, 但是链接太复杂反而影响软件对象的检索和重用, 应加以适当控制。

超文本方法能提供强有力的多途径浏览功能, 可以帮助使用者充分了解软件库的内容, 为提高浏览效率, 应配置合理的"导航"机制。

5 结束语

以上的分类方法和超文本方法, 从不同方面描述了软件对象的内容及其联系。没有一种方法能充分满足用户的需求。体系分类法具有结构化和继承特性, 但需手工分类; 组面分类法具有多维性和易实现性, 但实现信息量大, 有重复信息; 超文本方法具有动态性和灵活性, 但查找速度慢。

因此, 在组织软件资源时, 应尽可能采用多种方法, 取长补短, 分清主次。类层次结构是所有数据模型的核心部分, 可以作为所有资源组织的公用元模型, 有利于标准化。以功能本质特征组织的分类结构, 可作为软件库的元模型或数据模型, 以提高软件的可控性、可复用性和共享性。组面分类法组织的二维倒排表, 可作为软件库的扩展结构, 以提高软件库的检索效率。超文本结构用于非结构化的对象连接, 作为辅助结构, 与其他方法组合使用。随着构件技术、数据仓库、人工智能等新技术的不断发展, 还需探索其他组织方法, 如智能方法、多维组合方法。

参考文献

1 Tomas Isakowitz and Robert J. Kauffman, "Supporting Search for Reusable Software Objects", IEEE Transactions on Software Engineering, Vol 22, No. 6, June 1996
2 William B. Frakes and Thomas P. Pole, "An Empirical Study of Representation Methods for Resuable Software Components", IEEE Transaction on Software Engineering, Vol 20, No. 1, Aug1994
3 W. B. Frakes and P. Gandel, "Representing reusable software", Inform. Software Technol, Vol 32, pp. 47-54, 1990
4 Ruben Prieto-Diaz, "Implementing Faceted Classification", Communication of ACM, Vol 34 No. 5, page 88-97, May 1991
5 Guttorm Sindre, Reidar Conradi and Even-Andre Karlsson, "The REBOOT Approach to Software Reuse", J. System Software, 1995, 30: 201-212
6 Li Keqin, Guo Lifeng, Mei Hong, and Yang Fuqing, "An Overview of JB (Jade Bird) Component Library System JBCL", Proc TOOLS Asia '97, Beijing, China, Sep. 1997
7 张琪玉编 情报语言学基础: 修订第二版, 武汉: 武汉大学出版社, 1997
8 软件构件和软件构架技术 计算机世界, 1999-03-15: C7
9 软件复用及其相关技术 计算机世界, 1999-03-15: C5
10 基于构件库及构件组合的软件重用 计算机工程, 1998 (11)
11 曲艳霞等 面向对象重用库管理中分类查询机制的设计 哈尔滨工业大学学报, 1995(5)
12 余江 基于知识的可重用构件查询系统 KCQ. 软件学报, 1994(10)
13 张学群 CASE 和软件重用性 微机发展, 1994, 4(2)
14 马宇飞 软件重用项目中的若干问题的对策 计算机科学, 1993, 20(4)
15 耿刚勇等 基于构件的应用软件系统的体系结构及其开发模型 计算机研究与发展, 1998(7)

张玉峰 武汉大学大众传播与知识信息管理学院副教授, 出版专著两部, 发表论文多篇, 完成多项科研项目。通讯地址: 武汉市。邮编 430072。

文燕平 武汉大学大众传播与知识信息管理学院硕士研究生。通讯地址同上。

(来稿时间: 1999-08-27。编发者: 李万健)