

陈 远

信息系统构件的标引问题研究

摘 要 实施信息系统构件标引的不利因素有:构件无严格分类标准、具有多态性、是“黑盒”型的、属性复杂、标识用语不规范,等等。信息系统构件特征分为内在的、外在的和不规则的,可供标引的款项很多。信息系统构件标引的原则是:适度抽象、逐步规范、尽量包容、易于理解及方便增减。参考文献5。

关键词 信息系统 构件标引 标引款项 标引原则
分类号 G254

ABSTRACT In the paper, the author analyzes factors unfavorable to the indexing of components of information systems, characteristics of these components and principles for the indexing. 5 refs.

KEY WORDS Information system. Indexing of components. Principle.

CLASS NUMBER G254

为了提高信息系统开发的质量与效率,采用构件来“集成”信息系统的优势是毫无疑问的。目前任何一个信息系统的开发都不同程度地采用了这项技术。这里所说的构件是指可以用于多个信息系统的、具有可复用性质的系统成分,其内容兼有独立性、完整性、可标识性、一般性、适应性、可靠性、标准化的特点。伴随构件开发水平的提高,构件的种类与数量越来越多,能否在开发一个新系统时准确地判断、找到合适的构件并避免误用是一件颇为困难的事。这就需要构件进行分类、存贮和管理,建立构件的检索系统。而这项工作的前提则必须全面、深刻、简明地描述构件的特性,即对构件进行必要深度的标引。

1 构件标引的困难

构件的特性描述较之文献的特性描述要复杂困难得多,原因在于:

(1) 构件无严格的分类标准。不同的角度有不同

的分类:构件有专门生产商提供的,如有名的 Microsoft、IBM、Component Ware Consortium 公司和组织,也有用户自行开发的;有遵循一定标准的,如 ActiveX、JavaBeans、CORBA 规范,也有非标准化的。还可根据构件适用范围分为通用基本构件(适用于各种信息系统)、领域共性构件(是信息系统所属领域的共性组成成分)、应用专用构件(是每个信息系统特有组成成分)等。

(2) 构件的层次规模不一。按不同的适用阶段,分别有基于信息系统分析、设计、测试、维护等阶段的构件。还有不同抽象层次的构件,从产品的概念、系统模型和规约,到程序源码甚至目标码。有不同粒度的,从系统构架、框架、类簇到代码片断。也有不同形式的,像黑盒构造块、生成器、模板、设计模式等。

(3) 构件具有多态特性。构件往往可用于不同的环境,或在同一环境中扮演不同的角色。构件是不能脱离具体的系统而独立生存的,与环境中其它构件应

参考文献

- 1 Verona, Eva. The function of the main entry in the alphabetical catalogue. in: ICCP report. London:1963:145 - 157
- 2 Harrison, j., Laslett, P., ed. Anglo-American Conference on the Mechanization of Library Services, University of Oxford, 1966 The Brasenose conference on the automation of library. London: Mansell: 24 - 32
- 3 Lubetzky, Seymour. Code of cataloguing rules: author and ti-

tle entry: an unfinished draft. Chicago: American Library Association, 1962. p. ix

- 4 American Library Association, et al. Anglo-American rules. Edited by Mikchael Gorman and Paul W. Windler. 2nd ed. Chicago: ALA, 1978: 2

赵慧勤 北京大学信息管理系硕士研究生。通讯地址: 北京大学。邮编 100871。(来稿时间:2000-04-18)

有可互操作性,交互时构件或许仅一部分性能可见或一部分操作可用。因此,一个构件含有不同的交互配置界面,不同的交互协议,不仅需要标识构件提供什么,还要标识其它交互方给予什么,而这些参数都具有动态性质。

(4) 构件本身是“黑盒”型的。提供使用时,仅有构件的外部描述是不够的,还必须反映构件的固有特征,例如它所描述的对象、功能及应用场合等。构件的准确理解无法像文献资料一样以配备原文系统予以辅助,因为“黑盒”封装本就是构件设计的目标,撇开容量顾虑,即使提供原件,其代码或符号系统解读起来也非常困难,加之(3)所述的动态特征,它的功能也不能孤立地确定。

(5) 构件的属性灵活复杂。标识构件是为了分类、检索及正确使用构件。全面、多角度地标识构件是必需的,也是很困难的,例如有关构件的测试报告、应用记录和修改需求等属性就不容易标识。另外,构件有时隐含了一些在新系统的使用条件下将发生、发作的错误,因此构件具有修改、淘汰、升级问题,它的生存期处理与文献的老化问题不大一样。构件的复杂属性中有不少多变的成分。

(6) 构件标识用语难以规约。还没有一种公认适用的规范语言来描述构件特性,有关构件的术语空间无限不定,难于维护,在构件提供者、复用库组织和使用者之间建立和维护共同合适的词汇表需要足够的技巧和耐心,构件属性中有不少术语具有同义词及词的语义模糊问题。另如一个构件在计算域的语义是“在数组中找最小值”,而在应用域的语义是“在学生成绩单中找最低分”。用计算域的语义描述构件对跨领域的运用非常重要,但多数使用者却更习惯于从应用域的角度去查找构件,构件的通用性使得用应用域语义进行描述过于具体,可能会导致片面和繁琐的缺陷。

2 构件标引的网罗性

信息收录的广泛全面和检索的迅速准确,是对检索系统的基本要求。构件特征著录的详略,是决定检索工具所提供的信息量是否丰富的一个因素,而对构件主题分析所达到的深度,即标引深度,也是决定检索工具质量的一个重要因素。构件特征分内在的、外在的和不规则的,归纳起来,一般有如下可著录的款项:

(1) 名称。构件的名称有自然语言的,也有专用符号代码的。某些构件除了通用名称外,还可能有别

名和译名等。

(2) 支持者。构件的属主、维护人、技术支持人常常是不同的,如一个构件属于某公司机构,是某某人开发的,由某组织进行管理。

(3) 时效。反映构件的稳定性和生命期限。刚开始利用构件时会有较大的风险,多次利用后,构件日趋成熟,风险减小。有些构件自提交日起,会逐步老化、淘汰。

(4) 来源。提供构件的出处,如商业供货的、军方供应的、政府供应的,还是特定系统内部流通的。

(5) 应用级别。构件可用于信息系统开发的不同阶段,构件本身的类型决定它是应用于设计、运行还是集成阶段。

(6) 应用范围。构件应用的目标领域与场所。要指明构件所有一般和专门的应用环境通常是很困难的,必须说明那些最具代表性和概括性的应用领域。

(7) 构件类型。构件种类划分有多种观点,当一个构件的“构件类型”属性值为“分析”或“设计”时,就不适合以“编程语言”来区分它。

(8) 功能。定义构件所完成的一项或多项服务和操作的性能,功能描述是抽象还是具体,应视构件本身特性、使用方式和应用领域而定。

(9) 技术。包括构件采用的理论、方法、标准、协议、语言、介质、对象等,以及其约束条件。

(10) 互操作性。尽管构件本身设计成尽可能独立,但在一个系统或应用环境中,构件总是会和其它构件本相互作用,这种调用或连接界面必须明确定义。

(11) 兼容性。构件相互作用的标准多样,不同系统的构件并不匹配。即使统一了一些简单的标准,诸如应用于构件间互操作的 ASC 文本,它保证了构件间的通信和协作,但不能保证构件的兼容性,可进行一些必要说明。

(12) 移植性。定义构件运行和可被移植的平台(如硬件、操作系统、通讯系统等等),一种代码构件需要专门的译码器,一个可执行构件要指定操作系统。

(13) 可修改性。许多专家不提倡对构件进行修正,这样会失去其可靠性,但部分构件提供给使用者调整的余地,以适应不同应用的需要,必要的修改方法应详细说明。

(14) 同类与隶属。标示能解决同类问题,功能相同和相近的其它有名或流行的构件,提供类似同义词或近义词的参照体系。与构件有继承等关联的其它

构件时常也有必要标明。

(15)重要线索。对构件使用情况、版本更新、隐含错误进行简要登记。

不同类型的构件需要标引的款目差别很大。以上是具有共性和普遍性的描述构件特性的标识项。然而,任何信息只要经过处理就会产生误差,尽可能缩短认知距离是选择标引项目和用语的宗旨。

3 构件标引的原则

根据信息系统构件的特征,在标引时应该掌握一些基本原则。

(1)适度抽象。具有较高功能抽象的构件也会有更加宽广的应用范围,穷举构件的功能效应和应用参数是不太现实的。尽管构件使用者倾向于直接从应用域角度去理解构件,但大多数构件使用者具有一定的系统开发知识,所以适当的抽象化描述不会影响构件的使用。

(2)逐步规范。构件开发的标准化趋势给标引的规范带来了希望,有望确定公认的标引款目和合适的标引用语。标引的规范化有利于分类和检索算法的实现,但构件不同于一般的产品,具有复杂动态特性,因此,其标引的规范化还只能是局部的。只有在构件技术成熟普及后,标引的规范才能真正实现。

(3)尽量包容。不同类型、不同系统构件供标识的属性不同;不同粒度、不同层次的构件对同一属性的定义也会不同,例如代码级别的构件会有输入或输出参数,而文档类的构件却无此项说明。标引既要保证全面概括,又不能过细或过于专指。

(4)易于理解。对信息系统构件的组织和管理是为了在开发新系统时可以直接利用,提高开发质量与效益。为了避免构件的误用与“埋没”,标引款目应界定分明,容易理解,防止造成语义模糊和二义性现象,还要分析构件使用者的业务范围,专业素质及习惯做法,以帮助属性标识的选择。

(5)方便增减。构件技术正在发展之中,专门独立的构件管理检索系统还很少见。因此,可标识的构件的内外属性变化性大,标引时应充分考虑这些因素,增加一定的灵活性,留有修改余地,例如,设置一些参考项目等。

4 结束语

信息系统构件是一种精神产品,它的复杂性大大

超出了人类能制造的物质产品的复杂性。以构件集成系统的思想早已出现,但远远未达到大规模、大批量采用构件生产质量稳定的信息系统的目标。除了构件本身技术的原因外,非技术因素也不容忽视,包括对构件的有效管理还很不够,这方面的研究随着构件技术的发展正在日益深入。

由于构件的特殊性和复杂性,不能保证标引附加的描述信息能唯一地确定每个构件,也不能保证这些信息与使用者的理解完全一致,例如一个构件功能描述为“商品销售”,而使用者试图寻找一个功能为“售货”的构件。

为了解决这些矛盾,在设计构件检索系统时可以采取一些相应措施,如增加更多的刻面与构件属性;根据构件内容预先给出一组关键词(一般应由构件开发者给出);建立同义词和近义词对照表;记录用户使用构件库的经验,据此改进构件检索系统;对每个构件给出一段文字描述,进行简要介绍;提供较强的人机对话功能;引导使用者不断修正自己的需求表达等等。

参考文献

- 1 Sherif Yacoub, Hany Ammar, and Ali Mili. Characterizing a Software Component. 1999 International Workshop on CBSE.
- 2 Ivica Crnkovic, Magnus Larsson. Managing Standard Components in Large Software Systems. 1999 International Workshop on CBSE.
- 3 常继传,郭立峰,马黎.可复用软件构件的表示和检索.计算机科学,1999(5):45~49
- 4 刘东生,宗晓梁,周容国等.可重用构件的合成技术.计算机应用研究,1999(7):52~54
- 5 邵维忠,杨芙清.面向对象的系统分析.北京:清华大学出版社,1998

注 本文系国家自然科学基金项目(79900012)研究论文。

陈 远 武汉大学传播与信息管理学院副教授,发表论文 20 余篇,参编著作 3 部,参与 4 项国家自然科学基金和社会科学基金项目研究。通讯地址:武汉大学传播与信息管理学院。邮编 430072。

(来稿时间:2000-08-24)