计算机科学1997Vol.24№.4

基于领域的O-O软件重用工具的研究与开发

Research and Development of Domain-based O-O Software Reuse Tool

毛新军　齐治昌

（国防科技大学计算机系　长沙 410073）

摘　要　Domain-based Object-Oriented software reuse tool is developed under the environment of PC Windows to manage the reusable software components. The paper introduces the tool's characteristics, demonstrates the tool's logical architecture, analyses the tool's implementation ideas, methods and technique details, and presents a software development process model which is based on the tool and supports software reuse.

关键词　Software reuse　Software component　Process model

基于领域的面向对象软件重用工具（简称OORT）是我们在PC Windows环境下开发的可重用软部件管理工具。OORT按应用领域来组织、管理可重用软部件，支持多个应用领域软部件库的建立和维护；支持系统重用，OORT收集了OO软件开发过程中四类可重用软部件信息：(1)应用领域范围内的项目(project)信息，(2)OOA信息，(3)OOD信息，(4)OOP信息；工具提供多种软部件检索设施，用户不仅可以浏览和导航软部件库中的内容，而且还允许以查询方式检索源代码软部件；OORT引进了领域字典的概念，并提供一组设施来定义领域关键术语，建立领域术语间的各种语义关系，解决了关键字描述的一词多义、多词同义问题；并能基于领域字典实现最大范围软部件检索以及进行匹配度评估；OORT提供的工具集确保了系统的一致性和完整性，软部件库是开放式的；工具的层次式逻辑体系结构使得系统易于集成和扩充；为了纪录可重用软部件的各种信息，加强对软部件的理解，OORT支持软部件文档信息的建立和维护；在Windows环境下开发，具有较高的软件平台和友好、一致的用户界面。

1. 逻辑体系结构

OORT的逻辑体系结构如图1所示。系统以存储库为核心，由逻辑上相关的四个层次组成。(1)存储库：由领域字典、数据库和软部件库组成。(2)库管理系统：负责管理和维护存储库，提供一组基本库操作（如加入、删除、修改、查询等）作为外层工具的接口。(3)工具箱：提供一组工具（如匹配度评估、软部件修改、软部件浏览器等），这些工具基于库管理系统提供的基本库操作，充当用户和库管理系统的中介，协助用户完成各种软件重用活动。(4)界面系统：实现用户和工具间的信息交互。OORT的这种层次式逻辑体系结构，使得工具结构清晰，易于实现和扩充。外界用户不必具体了解库系统的结构和相关操作；任何基于库管理系统提供的公共接口的工具都易集成到该系统中。

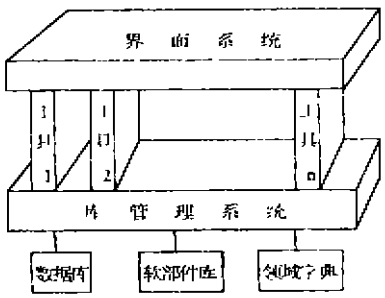

图1　OORT的逻辑体系结构

2. 实现思想、方法和技术细节

2.1　存储组织

OORT以应用领域为单位来组织、管理可重用软部件。整个存储库由多个应用领域存储库组成，分别支持各个应用领域可重用软部件的存储和检索。每个应用领域存储库包括三类信息库：(1)领域字典：定义领域关键术语，建立领域术语间的语义关系并收集语义相似度/相关度信息；(2)数据库：存储可重用软部件的内部特性信息（查询信息），以支持软

部件查询；(3)软部件库：存储 OO 软件开发过程中的各种可重用软部件实体(包括项目软部件实体，OOA、OOD、OOP 软部件实体)，每个可重用软部件实体由三个部分组成：可重用软部件、软部件文档和软部件外部特性信息。

2.2 软部件分类

软部件的分类直接决定了对软部件的检索和定位，好的分类法应支持快速、有效的查询。OORT 采用二种分类法来组织存储库信息：枚举分类和面分类。对于应用领域存储库，我们采用枚举分类法，枚举因子对应于应用领域名。对于应用领域目录库中的各个目录项，我们采用面分类法。我们对 OO 源代码软部件的特性进行了抽象，设计了一个三元组的面分类方案：〈Class Name，Operate，Operate On〉。其中，Class Name：类名。如 integer stack，library 等；Operate：类提供的操作。如 push，delete，create 等；Operate On：操作对象。如 integer，real，windows 等。

2.3 软部件描述

为了对软部件进行分类、存储、检索和理解，必须对软部件的特性加以描述。OORT 从三个方面描述软部件的特性。

1)内部特性描述。即对上述面分类方案中的各个面进行描述。OORT 的内部特性描述采用关键字表示法：用名词短语描述类名，动词短语描述软部件提供的操作。这种约束用 BNF 文法表示为：

Class Name→NP

Operate→Verb

OperateOn→NP

NP→Adjective NP|Noun

OORT 根据上述约束对用户软部件内部特性描述进行语法分析，对需求描述进行理解。

2)外部特性描述。OO 软件开发过程中的各软部件并非相互独立，而是互为联系的。软部件的外部特性集中体现在软部件的这种关系上。OORT 从两个方面描述软部件的外部特性信息：(1)横向关系(层次关系)，OO 软件开发过程中相邻阶段开发的软部件间的关系。如 OOA 软部件与其相关的 OOD 软部件间的关系。(2)纵向关系(继承及调用关系)，OO 软件开发过程中同一阶段开发的软部件间的关系。OORT 仅描述 OOP 阶段各类间的继承及调用关系。

OORT 的软部件外部特性描述采用超文本(Hyper Text)表示法。OORT 软部件外部特性的描述和建立，使得用户可以极为方便地了解软部件库中的内容。为 OO 软件开发提供一个自底向上的开发环境。

3)详细描述。采用非形式化、自然语言的形式对软部件的特性作出更为详细、准确的描述，记录软部件可重用性信息，形成软部件文档。

2.4 领域字典

关键字描述虽然简单、易于实现，但存在严重的一词多义、多词同义问题。此外，客观事物是相互联系的，这种联系表现在泛化与特化关系以及整体与部分关系。OO 软件开发的一个重要特点就是试图揭示客观事物间的上述联系来促进软件开发。客观事物间的上述联系对软件重用工具提出了更高的要求：能否根据用户提出的泛化对象(如笔)，检索出与之相关的特化例子(如铅笔、钢笔等)，如果用户需要一个具有整体特征的对象(如树)，工具能否从软部件库中检索出对应于该对象具有部分特征的例子(如树干、树枝等)。

OORT 引进领域字典来解决上述问题。OORT 的领域字典定义领域关键术语，引进映射函数 f：term word→domain concept 建立领域关键术语与领域概念间的一一映射，通过四元组〈T，T，R，V〉建立领域术语间的语义关系并收集语义相似度/相关度信息。其中 T 为术语世界，R 为术语间的语义关系且 R＝{同义，泛化与特化关系，整体与部分关系}，V 为描述关联术语的语义相似度/相关度值。OORT 基于领域字典，对用户软部件描述作出解释，对用户需求描述进行理解。领域概念是指应用领域的对象、行为和特征，如栈，删除等等。对于某个领域概念 A，存在多种术语描述 $\alpha=t_1,\cdots,t_n$，假如我们选择集合 α 中术语 t_i 作为描述领域概念 A 的关键术语，那么就可通过映射函数 f 建立术语 t_i 与领域概念 A 间的映射，通过四元组$(t_j,t_i,$同义$,v_j)$建立术语 t_j 与 t_i 间的同义关系$(1\leqslant i,j\leqslant n\quad j\neq i)$。此外，客观事物间的泛化和特化、整体和部分的关系必然体现在描述它们的术语语义上，因而存在一组术语 $\beta=T_1,\cdots,T_m$ 与 t_i 相关联，通过四元组(T_j,t_i,R_j,V_j)建立术语 T_i 与 $t_i(1\leqslant i,j\leqslant m)$间的泛化和特化、整体和部分的关系。这样，如果用户用术语 t_j 描述所需的软部件，则 OORT 通过术语 t_j 与术语 t_i 之间的同义关系，将该术语解释为领域概念 A，进而从软部件库中检索出软部件 A。同样，假如用户用术语 T_j 描述所需的软部件，则 OORT 不仅能从软部件库中检索出与 T_j 所对应的领域概念相关的软部件，还能通过术语 T_i 与术语 t_i 间存在的泛化和特化、整体和部分

的语义关系检索出软部件 A。

根据关键字描述的语法约束，领域字典收集了三类领域术语：名词、动词和形容词。OORT 对领域术语间的各种关系（同义关系、泛化和特化关系、整体和部分关系）作同一处理。通过领域分析建立领域字典。

2.5 软部件存储

软部件的存储机制应支持软部件的有效检索，软部件库不仅存储可重用软部件，还应存储软部件的外部特性信息。

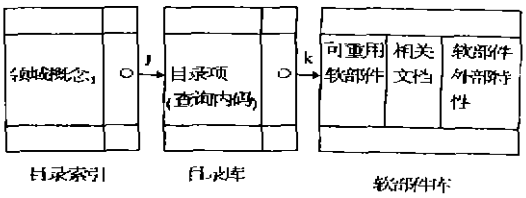

图2 目录索引、目录库和软部件库间的关系

OORT 的软部件存储主要完成以下几项工作。按关键字描述的语法约束对用户软部件描述作语法分析，识别领域术语，将不在领域字典中的领域术语加入到领域字典中。OORT 基于领域字典对领域术语进行解释，形成工具可以识别的查询内码。建立该查询内码项的目录索引并基于目录索引将查询内码项存储于数据库中。OORT 的数据库由两类目录库组成，按类名组织的目录库和按操作名组织的目录库；它们各有一个目录索引。目录索引按领域概念来组织，它指出了与领域概念相关的查询内码项在目录库中的具体位置和数目，目录库中的每个查询内码项用领域概念描述了软部件的内部特性，指明了该软部件在软部件库中的存储位置（见图2）。其中，类模板目录库按 Class Name 面进行分类、组织查询内码项，操作目录库按 Operate 面进行分类、组织查询内码项。OORT 根据类名描述中的中心名词所对应的领域概念，在类目录库的目录索引中建立该查询内码项的目录索引并基于该目录索引找出该查询内码项在类目录库中的存储位置，将查询内码项存储于类目录库中；同样，OORT 根据操作名描述中的中心动词所对应的领域概念，在操作目录库的目录索引中建立该查询内码项的目录索引并根据该目录索引找出该查询内码项在操作目录库中的存储位置，将查询内码项存储于操作目录库中。OORT 根据用户对软部件外部特性的描述，建立软部件的外部特性信息。OORT 将可重用软部件与其文档、外部特性信息进行包装，形成可重用软部件实体存储于软部件库中。

2.6 软部件外部特性的建立和修改以及软部件的浏览

OORT 软部件库中记录了各软部件的外部特性信息。OORT 软部件外部特性描述采用超文本（hypertext）表示法。它由一系列的结点和边组成，结点对应于软部件实体，边则反映了这些结点间的关系（横向关系、纵向关系）。超文本允许用户从一个结点通过连接结点的边到达另一结点，从而对这些结点进行查询、检索、浏览和重用。

OORT 引进结点栈来实现软部件浏览。结点栈中的每个项是指向软部件库中软部件实体的指针。OORT 的结点栈不仅记录了当前浏览软部件的信息，从而可以查看、重用、删除和修改该软部件，并基于该软部件的外部特性信息作进一步地浏览；而且还记录了软部件浏览的历史路径信息，使得用户可以极为方便地返回到最近浏览的结点。结点栈的栈顶结点始终是用户当前浏览结点。OORT 根据栈顶结点的外部特性信息，提供与该结点相关联的软部件，使得用户可以进一步向前浏览。OORT 通过弹出结点栈的栈顶结点，使得次栈顶结点成为栈顶结点，返回到最近浏览的结点。

2.7 软部件查询

工具提供了多种软部件的查询方式：OORT 既可基于用户对类名和操作、操作对象名的描述，检索出具有该类名和操作、操作对象名特征的软部件，亦可基于用户对操作、操作对象名的描述，检索出具有该功能的软部件。

OORT 的软部件查询主要完成以下几项工作。按关键字描述的语法约束对用户需求描述作简单的语法分析，识别出领域术语；OORT 基于领域字典对用户需求描述作出解释，将用户的关键字描述转换成领域概念描述，形成工具可以识别的查询内码；选择查询目录库，根据目录索引，查找软部件，OORT 根据用户需求描述中的中心术语所对应的领域概念，在相应的目录索引中找出查询内码项在目录库中的存储位置和查询范围，进而在该目录项范围内检索软部件，判断各目录项的软部件描述是否完全或部分地满足用户的需求，如是，则对该软部件同用户需求描述间的匹配程度进行评估，并根据该目录项的索引从软部件库中提取出相应的可重用软部件实体，送到软部件浏览窗口，供用户查看和重用，以及根据该软部件的外部特性信息作进一步地浏览。OORT 的上述查询方式，可以有效地减少查

询对象的数目和响应用户所需的时间，提高工具的查询效率。

3. 支持重用的软件开发过程模型

基于OORT支持重用的软件开发过程模型如图3所示。该模型将软件开发过程、软件重用过程和

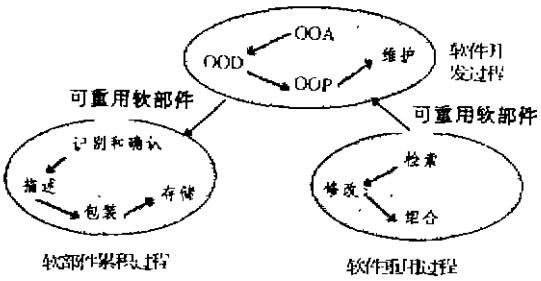

图3 支持重用的软件开发过程模型

软部件累积过程三者紧密地结合起来，提供了与软件重用相关的七个活动。

·识别和确认：基于领域分析，确认软件开发人员提供的可重用软部件的合法性，识别和获取软部件的可重用性特征，防止非法和可重用性低下的软部件加入到软部件库中；

·描述：对即将加入到软部件库中的软部件的特性（包括内部特性和外部特性）加以描述，以支持工具对它的存储和检索以及用户对它的理解；

·包装：提供一个单独的软部件通常是不完备的，为了加强用户对软部件的理解，还必须提供有关文档和软部件外部特性信息，并将它们与软部件包装在一起，构成一个可重用软部件实体；

·分类和存储：将经上述处理的软部件进行分类，以一种易于检索的方式存储于软部件库中；

·检索：用户根据自己的需求从软部件库中检索，选择相关的可重用软部件；

·修改：检索到的软部件通常很少能同用户的需求完全匹配，用户必须对可重用软部件进行适当的修改，以完全满足用户的需求；

·组合：用户对所选择的软部件进行组合，以形成复杂的软件系统。

基于上述模型，我们在引进OORT的软件开发过程中应遵循以下三个原则：

·在软件开发的各个阶段充分地利用已有的软部件资源，减少开发新产品的数目；

·任一阶段的工作都是基于前一阶段的工作（自顶向下）和相应下一阶段已有的可重用软部件（自底向上）寻求问题的解，充分发挥重用的潜力；

·适时开发高质量的可重用软部件，丰富软部件库的内容。

结束语 OORT软部件管理工具的研究和开发，为我们对软件重用的研究作了有益的探讨，为进一步研究奠定了基础。OORT能较好地支持软件开发过程中的各种软件重用活动，为用户提供了一个系统、有规划的软件重用环境。进一步的工作包括：1）累积软部件，丰富软部件库的内容；2）研究如何利用现有的数据库技术、知识库技术和信息检索系统来支持软部件库的建立、管理和维护；3）开发支持软部件形式化描述和检索的软部件管理工具。

参考文献

[1] T. G. Lewis, CASE: Computer-Aided Software Engineering

[2] James W. Hooper et al., Software Reuse Guidelines & Methods

[3] Even-Andre Karlsson et al., Classification of Object-Oriented Component for Reuse

[4] 毛新军、齐治昌，软件重用研究和应用，计算机科学，No. 4, 1994

下期主要内容预告

人类自然语言不确定性及粗集理论的表示
智能多媒体表现系统：研究现状与发展趋势
语音翻译系统技术分析
虚拟现实建模语言：现状与展望
ODP系统中的联邦交易模型
一种分布式多媒体系统模型
基于CORBA的多数据库系统
知识发现的若干问题及应用研究
Fusion方法导论
Client/Server结构系统中的处理分布