

构件库技术的研究与发展*

潘颖 赵俊峰 谢冰

(北京大学计算机科学技术系软件工程研究所 北京100871)

The Research and Development of the Technologies in Component Library

PAN Ying ZHAO Jun-Feng XIE Bing

(Department of Computer Science and Technology, Institute of Software Engineering, Beijing University, Beijing 100871)

Abstract The retrieval and management of components play important roles on the Component Based Software Development (CBSD), which is an effective way to support software reuse. Reusable component library is one of the foundation establishments in it. It is well able to provide component representation, classification, certification, storage, retrieval, evaluation and feedback, interoperable libraries, etc. This paper is a summarization on the technologies in component library. It introduces their current status, presents four component libraries, and outlines the future trends.

Keywords Software reuse, Software component, Component repositories, Reuse library

1. 引言

软件工程的目的是致力于提高软件生产效率和软件质量,摆脱手工作坊式的开发方式。如果每个应用软件系统的开发都从头开始,其中必然存在大量的重复劳动。软件复用是一条提高软件生产效率和软件质量的切实可行的解决方案,其出发点是应用系统的开发以已有的工作为基础,充分利用已有系统的开发中所积累的知识经验进行新的开发。这样软件开发的重点就可以集中于应用系统中的特有构成成分上。软件构件只有在数量上达到了一定的规模才能真正满足软件复用和基于构件的软件开发(CBSD)的需求,因此必须有一个强有力的工具来对这些数量庞大的软件构件进行管理。构件库作为一种支持软件复用的基础设施,它提供对软件构件进行描述、分类、存储和检索等功能。

构件库提供的构件管理的工具从使用者的角度可以分为两类:构件库工作人员(系统管理维护人员)使用的构件库管理维护工具,和普通用户使用的查询检索工具^[1]。构件库系统必须有以下基本功能^[2]:存储构件、提供一种机制发现构件、获取构件和维护构件。对构件库技术的研究也主要围绕这几个方面进行。

在软件技术的发展过程中,出现过多种不同形式的构件库,主要包括:

(1)子程序库和类库:高级程序语言编译器的开发商提供给用户的开发工具中一般都带有若干子程序库,例如C语言的库函数。类库是OOPL中不可缺少的组成部分,例如Inprise的VCL、微软的MFC、SUN的Java类库等。一些面向某个领域的通用子程序库也发挥了很大的作用,例如,数学程序库就是非常成功的子程序库。它们是最早出现的构件库形式。

(2)领域专用构件库:领域专用构件库用于存储和管理某一特定领域构件的构件库。例如,Ada软件库^[3]存储的是可复用的Ada源代码。

(3)软件资产库:80年代末90年代初研究人员认为软件开发过程中分析、设计、编码、测试阶段一切有复用价值的软件

成分都可以称为构件,构件也被称为软件资产^[4],软件资产库用于存储这些广义范围的软件构件。软件资产库是十余年来构件库领域的研究重点,学界和业界对此做了大量的研究和实践工作,取得了相当成果。

(4)构件检索系统:检索构件和获得构件是构件库的重要功能,构件检索系统是支持可复用软件构件分类和检索的自动化工具^[1]。构件搜索引擎是近年来CMU/SEI的Seacord等人研究和开发的一种构件检索系统^[5],能够在Internet上自动寻找和搜集JavaBeans、ActiveX、CORBA、EJB构件,获得构件的URL等相关信息,并为之建立索引。

2. 构件库技术的研究

目前,与构件库相关的技术研究主要包括构件的描述与分类、验证、存储、检索、评估与反馈以及构件库的互操作等。一般来说,构件从入库到被用户复用需要经过如图1所示的流程。

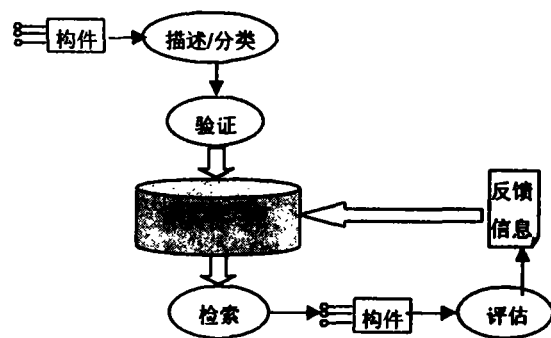


图1 构件管理流程

2.1 构件的描述与分类

构件库中的构件是来自软件生命周期各个阶段的可复用产品,具有多形态、多层次以及多种表示形式等特点。对单个构件的复用可以划分为构件的识别与检索、构件理解和适应性修改三部分,复用公式为:

*本文获得863项目(编号:2001AA113070)的支持。潘颖 博士研究生,主要研究方向为软件复用和软件工程。赵俊峰 博士研究生,主要研究方向为软件复用和软件工程。谢冰 博士后,主要研究方向为软件工程,软件开发环境。

复用成本=检索成本+理解成本+修改成本^[4] (1)

要减小复用成本,使复用者能较正确地理解构件,更好地复用软件构件,就需要对构件进行描述和分类。构件在入库前需要提供以下信息(称为构件的元数据^[7]):①构件的特性、安装、验证和使用的完整手册;②构件的摘要信息;③构件的分类信息;④实际要复用的部分;⑤构件的测试计划、目标、脚本和预期的结果;⑥相关构件及其关系(见文[8])。

从构件的表示出发,W. Frakes 将现有的构件分类方法划分为人工智能方法、超文本方法和信息科学方法三类。其中,信息科学方法在实际构件库的项目中应用较为成功。信息科学方法又分为基于受控词汇表(如枚举、刻面)分类和基于不受控词汇表(如关键词)分类两种形式。其中关键词分类、枚举分类、属性/值分类、刻面分类是构件库中较为常见的分类方法^[9]。

关键词分类 每个构件用一组与之相关的关键词编目。一般而言关键词的取值是不受控的词汇表,由于缺乏上下文语境,导致检索的效率和精确度也得不到保证。

枚举分类 通常将一个被关注的领域划分为不相交的子领域,依次构成层次结构^[10]。此方法有一定的不足,很难解决同一个构件属于不同子领域的二义性问题,不易于随领域的演变而改变。

属性-值分类 构件是根据一组固定的属性和对应的值来描述的。属性的值域是无限的确定空间(Infinite and Definite Value Space)^[11]。

刻面分类 由一组描述构件本质特征的刻面所组成^[11],每个刻面从不同的视角对构件库中的构件进行精确的分类,每个刻面具有一组术语(关键词),术语之间有类层次关系而形成结构化的术语空间。术语空间是可以演变的有限的不定空间(Finite and Indefinite Term Space)^[6]。刻面分类方法的使用存在着不少困难,一方面,刻面和术语空间的选择和确定将直接影响到此分类模式的基本结构,因此它们的选择和建立是一项艰苦的工作;另一方面,刻面术语并没有确定的、为大家所公认的标准,难以建立较为统一的术语空间,为构件库的互操作带来巨大的困难。

在学术界,对构件的描述和分类的研究非常多,几乎所有的研究都认为对构件的描述与分类应该是多样化的,构件库需要支持多分类模式,不能把对构件的描述局限在一种体制上,例如,Mary Shaw 提出构件规约需要提供对属性种类和分类模式的不断演化的支持^[12]。

2.2 构件的验证与存储

构件库需要对每个推荐入库的构件进行定性和定量的评价,主要目的是为了在一定程度上表明构件能满足的需求,预测构件将带来的收益等^[1],以达到保证库中构件的质量和规范性,此过程称为构件验证。目前构件库基本上都没有提供自动验证的机制,一般通过构件库管理员手工确认提交的构件是否是合格构件,或者将验证作为外部支持体系。

构件库中存储的构件包括构件的实际拷贝(称为构件实体)和构件的描述信息。一般而言,特别是随着 Internet 技术的发展,构件库中不需要存放构件的实际拷贝^[1]。多数构件库采用的是构件描述信息和构件实体相分离的存储策略,构件描述信息通过数据库管理系统进行存储,每个构件的实体用一个独立文件保存。这种方法的优点是构件库系统运行的负荷较低,同时由于构件实体可以通过网络分布到各个节点上,从而使得系统的开放性得到提高。此外,这种策略还有效地保

护了构件的知识产权。但是,采用这种方法,构件实体的安全性和可用性得不到保证,备份工作较为复杂。此外,有些基于 Internet 的构件库采用 html 文件存储构件描述信息,如欧洲 ACTS 基金资助的 SCREEN 项目^[13]。

2.3 构件的检索

从复用公式(1)可以得知有效的构件检索机制能降低构件复用成本,因此,构件检索是一项非常关键的构件库技术。多年来研究人员对此做了大量的工作,构件检索方法可以归结为三类:基于外部索引的检索、基于内部静态索引的检索和基于内部动态索引的检索^[14]。

基于外部索引的检索 如常见的关键词检索、刻面检索和基于属性的检索等。这类检索大多采用控制词典、属性等外部索引对构件进行检索。几乎所有的研究都认为提供自动化支持是十分必需的,自动索引、分层浏览、查询条件的简易规约和自动生成有助于提高效率、增加复用机会和提高复用质量。W. Frakes 的研究实验表明各种分类法基础上的检索在辅助不同用户的理解上并无太大差异,需要考虑在同一个系统中支持多种基于分类法的检索方法的组合使用^[15]。

基于内部静态索引的检索 根据构件自身的结构元素进行检索构件,其中构件规约的语法和语义匹配、结构匹配技术等是主要方法。构件规约是构件描述信息的特殊组成部分,用形式化的方法描述了构件功能的语法和语义,基于构件规约的检索包括语法匹配和语义匹配两方面的内容。到目前为止,规约的语法匹配已经研究得比较充分,语义匹配也形成了一个一般性的方法,但是,构件规约的语义表达能力非常有限并且语义表达形式也十分复杂。而且,语义匹配过程中的自动定理证明机制将导致系统的响应速度急剧降低,如何改进自动定理证明机制从而降低匹配的时间代价是近年来语义匹配的研究重点^[16]。

基于内部动态索引的检索 利用构件的可执行特性(如:构件的输入与输出空间)进行检索,基于行为的检索是目前这类检索中较常见的方法。构件行为一般定义为构件对输入信息的动态输出响应^[17]。Podgurski 和 Pierce 提出了行为检索的基本原理^[18],Mili 提出了使用网格结构做为行为检索的基础^[19]。一般来说,行为检索只适合面向对象类型的构件,采用 z 或者 object-z 语言标记对象的行为。由于实际应用的复杂性,基于行为的检索目前仅停留在研究阶段。

2.4 构件的评估与反馈

构件复用者在检索和选取适当的构件,构件库的管理者在管理和维护构件及构件库时,需要一套辅助进行分析决策支持的机制。一方面,构件评估与反馈机制有助于用户从检索到的构件中选取最符合自己需求的构件,如何为用户提供多样的检索机制,如何帮助用户正确评估和选取所需的构件,提高选取的正确性和高效性,如何协助构件库的管理员管理构件、调整构件库中的构件组织结构等,都要求构件库系统能提供一套辅助分析决策支持的机制^[20]。另一方面,对构件的度量结果也需要一套辅助分析决策支持的机制,如何对后期度量处理,如何将后期度量与前期度量的结果相比较,如何将度量结果反映给用户,以及如何辅助用户利用度量值和反馈信息进行构件的评价和选取,也成了一个亟待解决的问题。

目前,国内外学术界在构件库系统中的辅助分析决策支持方面进行了深入研究,例如,REBOOT 系统评估工具尽可能多地提供给用户关于构件的质量信息,包括一定程度的比

较信息,通过 Kiviat 图显示结果,使用户更易于理解构件,用户能根据他所感兴趣的方面定制评估模型^[21]。JBCLMS 实现了基于数据仓库的反馈机制,采用多维数据模型,支持面向主题的、集成的、带有时间属性、包含历史数据的数据组织,提供对反馈信息的多维数据分析^[22]。前者使用的是文本类型存储用户灵活多样的反馈信息,没有较好的量化的比较信息。后者采用的是数据仓库技术,可以对用户的反馈信息进行量化,提供柱状图的可视化分析,但是不能提供足够的评估内容和反馈信息。

2.5 构件库的互操作

构件库互操作的目标是解决不同构件库之间数据共享的问题。目前主要的解决策略是建立一个构件库所共同遵循的数据模型。RIG 的一个技术委员会提出了一个数据模型 BIDM,并在此基础上提出了 BIDM 的超集—UDM,定义了的支持互操作性的库之间交换软件构件所需要的信息^[6, 23]。UDM 提供了一个标准的数据模型作为构件库的无缝互操作的基础,各个库可以将它作为自己数据模型的中间表示,用来和其它库交换使用符合各自数据模型的数据。BIDM 定义了为了实现互操作,复用库交换软件构件时所需的信息的最小集。同时 RIG 指出,UDM 并不是定义可互操作数据模型的最终结果,还需要 UDM 的一个超集提供比 UDM 更加强化的软件构件交换能力。

3. 实例研究与比较

构件库作为软件复用的基础设施,成为近十几年来软件复用的研究热点和重点,国内外学术界对其进行了比较深入的研究。当前,已经出现了不少构件库系统和原型,它们都基于一些共同的观念,同时,它们之间存在一定的差异,主要表现在各个构件库采用的技术、系统的侧重目标,以及处理的构件类型和形态等方面。为了进一步理解构件库技术的研究现状,以下简要介绍四种构件库系统。

3.1 REBOOT 构件库系统

REBOOT (Reuse Based on Object Oriented Techniques) 是欧盟信息技术计划 ESPRIT 中的一个项目,目标是研究、开

发以及推广复用驱动的和面向对象的软件技术。REBOOT 系统是90年代初期国际上比较著名的构件库系统,由一个存储构件的复用库以及一组支持构件生产、考查、分类、选择、评估和适配的复用工具组成^[21]。有下列技术特色:

- 采用剖面分类作为主要分类方法,定义了 Abstraction、Operations、Operates on、Dependencies^[24]四个剖面来描述系统所存储的面向对象的源代码构件。

- 提供构件的详细描述和测试信息,用户可以通过这些信息来理解构件。此外,对每一个构件设置一个开放论坛,提供给用户报告他们的复用经验和碰到的问题。

- 开发了一个基于“要素—标准—度量”模型标准的度量体系^[25],定义了两个质量与可复用性的模型来评估构件的质量和可复用性。复用者可以通过主观评估与模型的静态计算相结合来评估构件。

- 要求用户在使用适配工具提取构件时,提供构件的复用历史,包括用户的满意程度、提取代价、修改代价、节省的努力、批评意见、构件的修改需求等方面的反馈。

3.2 青鸟构件库管理系统(JBCLMS)

JBCLMS 是国家“九五”重点科技攻关项目——“软件工程环境(青鸟 CASE)工业化生产技术及系统(JB/SEIMS)的研究开发”中的核心组成部分之一。(本文作者曾参与青鸟构件库管理系统的研发工作)。JBCLMS 是一个基于 Internet 的软件资产库管理系统,具有如下特点:

- 可以存储和管理广义的构件,采用数据库管理系统存储构件描述信息;文件系统存储构件实体。

- 提供关键词、剖面、规约等多种检索方式。JBCLMS 选取使用环境、应用领域、功能、层次、表示方法等作为基本剖面,以满足宽谱系构件的描述需求。

- 采用“用户—人员组—构件子库—构件”四级模式来实现用户在构件库中的职责和权限。

- 提供基于数据仓库技术的构件反馈机制,为用户选取构件提供辅助分析。

JBCLMS 的体系结构如图2所示。

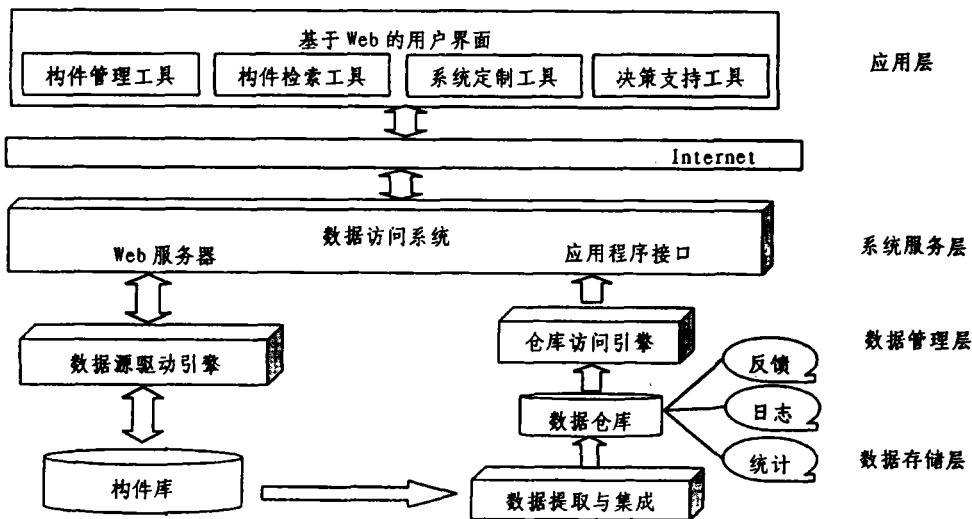


图2 JBCLMS 体系结构

3.3 Agora 构件搜索引擎

Agora 是 CMU SEI 开发的一个构件搜索引擎^[5]。SEI 的一些研究人员认为传统的构件库是一种大型的集中式系统,库的可访问性和可扩展性都有限。随着 JavaBeans、ActiveX、

CORBA 等商业构件标准的建立与发展,在 Internet 的各个节点上分布着大量可复用的标准构件,因此,应该研究一种在 Internet 上搜索这些构件的工具^[5]。

Agora 通过对 Web 页面的搜索来获取构件 URL 等相关

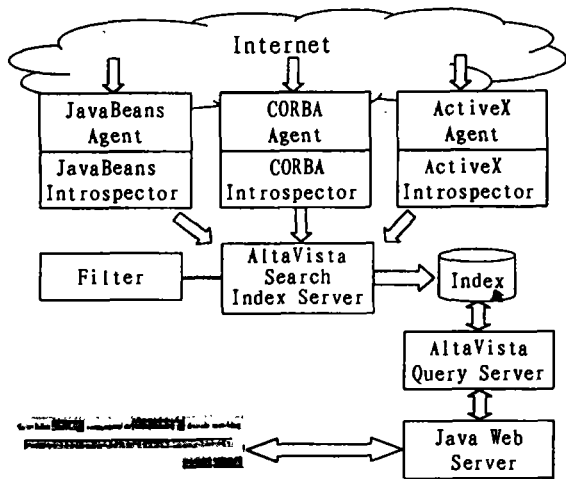


图3 Agora 的体系结构^[5]

信息,采用和普通搜索引擎相同的方式管理构件信息。除了利用 Spider 技术在 Web 页面上搜索构件外,Agora 也允许构件

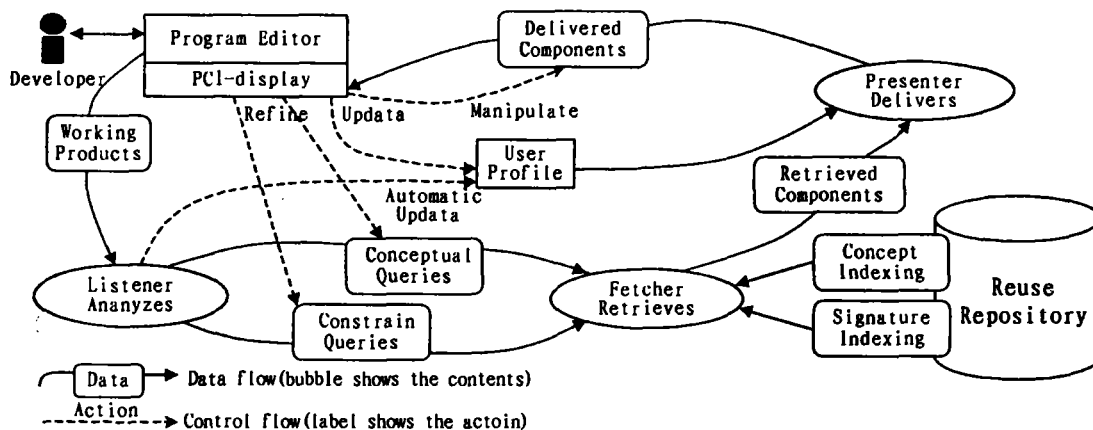


图4 CodeBroker 体系结构^[26]

系统通过三个 Agent (Listener、Fetcher、Presenter) 实现构件的主动查询。Listener 在后台实时监控用户在编辑器中的输入,通过对输入文本(注释、类名或方法名)的分析获得查询条件,交给 Fetcher 处理。Fetcher 在构件库中检索满足条件的构件通过 Presenter 在编辑器中动态显示出来,这样编程人员就可以及时而自动地获取所需要的构件。CodeBroker 的查询条件完全依赖于用户在编写程序过程中的文本输入,因此查准率较低,为此 CodeBroker 提供了一套构件过滤机制帮助用户快速删除不相关的构件。

目前,构件库技术在商业界和学术界都得到广泛的研究,并且已经开发了不少的构件库系统。除了以上介绍的四种构件库系统外,商业界有+1Reuse Repository、SALMS(软件资产库管理系统)、ASRR(自动软件复用库)、The Universal Repository(通用库)、AIRS(基于 AI 的软件复用库)、RLT(复用库工具集)和 HSTX 复用库等构件库系统,还有政府级构件库,如:DSRS(美国国防部软件库系统)、LID(构件库互操作示范工程)、I-CASE(计算机辅助软件工程集成环境)、MORE(面向多媒体的构件库)、SAIC/ASSET(面向软件工程的软件资产复用技术)、PAL(公共 Ada 库)、CAPS 软件可复用构件库和 DISA(Ada 库暨美国国防部信息系统代理复用库)等^[27]。

开发者将他们开发的构件直接在 Agora 服务器上注册,Agora 的体系结构如图3所示。

相对于传统的构件库,Agora 系统有如下特点:

- 节省了构件验证和构件库管理的成本,构件开发者无需提交构件,构件库的用户能够相对及时地得到构件的最新版本。

- 搜索的构件种类有限,由于构件模型的差异,对 Java Applet 和 ActiveX 的搜索效果比较好,而对其它种类的构件(如 CORBA、EJB)搜索效果较差。

- 构件的描述信息和分类信息不够充分。缺乏构件功能的语义描述信息,用户只能通过方法、事件或属性名称进行查询,不便于用户准确定位查询和使用构件。

- 库中构件的唯一性和存在性得不到保证。

3.4 CodeBroker

CodeBroker 是美国 Colorado 大学开发的一个构件库原型,其特色是构件库与源程序编辑工具实现无缝集成,为用户提供主动查询服务^[26]。CodeBroker 存储的构件是 Java 的类和方法,辅助 Java 程序员的开发,其体系结构如图4所示。

4. 发展趋势

构件库是支持基于构件的软件开发(CBSD)的基础设施。在构件复用的过程中需要一个实用有效的工具来支持构件的管理、存储和检索。对于开发人员来说一方面需要有一个构件提交的场所用以发布自己开发的构件,另一方面需要在软件开发的过程中尽快检索到自己所需的构件,这两方面都需要有一个强有力的构件库的支持。

构件库中构件的管理具有较大的灵活性,小规模企业内部的构件库可以是集中式的系统,并且可以在一定程度上促进企业软件复用。但是,由于构件数量的有限,软件开发的效率提高必然会受到很大的局限。只有众多软件企业联合,形成一个统一的软件构件库,采用基于网络的分布式管理方式,不同的构件库之间在物理上呈分布结构,在逻辑上形成一个整体,为不同领域构件提供一个发布、管理、检索、交易的场所,才能真正为软件开发效率带来质的提高。

随着 WWW 和 Internet 技术的发展,出现了 XML、Web Services^[28]、UDDI 等技术,“软件即服务”的思想使 Web 和 Internet 构成了新的计算环境和系统模型,为构件库技术的研究提供了新的技术手段和方案。复用技术和 Web 技术的不断发展为构件库技术指出了新的发展方向,还可以从以下几

(下转第156页)

应用系统本身的身份认证,以增强应用系统的安全性。而且实践证明该方案是有效的。

参考文献

- 1 Lampert L. Password authentication with insecure communication. *Communications of the ACM*, 1981, 24(11): 770~772
- 2 Haller N M. The S/Key one-time password system. In: *Proc. of the Internet Society Symposium on Network and Distributed System Security*, San Diego, CA, Feb. 1994
- 3 Rigney C, Rubens A, Simpson W, Willens S. Remote Authentication Dial In User Service(RADIUS). RFC 2138, April 1997
- 4 Rigney C. RADIUS Accounting. RFC 2139, April 1997
- 5 Steiner J, Neuman B C, Schiller J I. Kerberos: An authentication service for open network systems. In: *Proc. Winter USENIX Conference*, Dallas, TX, 1988. 191~202
- 6 Kohl J, Neuman C. The Kerberos Network Authentication Service(V5). RFC 1510, Sep. 1993
- 7 Tsudik G. Message authentication with one-way hash functions. In: *Proc. of IEEE Infocom' 92*, Florence, Italy, May 1992
- 8 Schneier B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, New York, 1994
- 9 Needham R, Schroeder M. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 1978, 21(12): 993~999
- 10 Davies D W, Price W L. *Security for Computer Networks*. John Wiley & Sons, second edition, 1989

(上接第93页)

个方面进行研究:构件信息的共享机制;基于 Web 的构件检索机制;分布构件库的一体化;构件表示的标准化、多样化;多库的协同检索;构件的反馈和后期度量。

参考文献

- 1 NATO Standard for Management of a Reusable Software Component Library, 1991
- 2 Sauer L D, Clay R L, Armstrong R. Meta-component architecture for software interoperability. *Software Methods and Tools*, 2000. SMT 2000. Proceedings. International Conference on, 2000. 75~84
- 3 Conn R. *The Ada Software Repository and the Defense Data Network: A Resource Handbook*, New York Zoetrope, 1987
- 4 NATO Standard for Development of Reusable Software, 1991
- 5 Seacord R C, Hissam S A, Wallnau K C. Agoro - a search engine for component. *IEEE Internet Computing*, Nov. /Dec. 1998. 62~70
- 6 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. 45~49
- 7 Cha J-E, Yang Y-J, Song M-S, Kim H-G. Design and implementation of component repository for supporting the component based development process, 2001 IEEE International Conference on Systems, Man, and Cybernetics. In: 2001 IEEE Intl. Conf. on, Volume; 2, 2001. 735~740
- 8 RIG Basic Interoperability Data Model (BIDM); [RPS-0001], 1993
- 9 Frakes W B, Pole T P. An Empirical Study of Representation Methods for Reusable Software Components. *IEEE Trans. On Software Engineering*, 1994, 20(8)
- 10 Sauer L D, Clay R L, Armstrong R. Meta-component architecture for software interoperability. *Software Methods and Tools*, 2000. SMT 2000. proceedings. International Conference on, 2000. 75~84
- 11 Prieto-Diaz R. Implementing Faceted Classification. *Communication of ACM*, 1991, 34(5): 88~97
- 12 Mary Shaw: Truth vs Knowledge: The Difference Between What a Component Does and What We Know It Does, Proceedings of 8th International Workshop on Software Specification and Design, 1996 pp. 181-185
- 13 Partners of the SCREEN Project: Service Creation Engineering Environment, Sema Group Telecom, France Telecom CNET, ACTS Ref.: AC227 SCREEN, 1999
- 14 Atkinson S. A Unifying Model for Retrieval from Reusable Software Libraries; [Technical Report No. 95-41]. the University of Queensland, 1995
- 15 Frakes W B, Pole T P. An Empirical Study of Representation Methods for Reusable Software Components. *IEEE Trans. On Software Engineering*, 1994, 20(8)
- 16 Fischer B. Specification-Based Browsing of Software Component Libraries. *Journal of Automated Software Engineering*, 2000, 7(2): 79~200
- 17 Atkinson S, Duke R. Behavioural Retrieval from Class Libraries. *Australian Computer Science Communications*, 1995, 17(1): 13~20
- 18 Podgurski A, Pierce L. Behaviour Sampling: A Technique for Automated Retrieval of Reusable Components. In: *Proc. of the 14th Intl. Conf. on Software Engineering*, 1992. 349~360
- 19 Mili A, Mili R, Mittermeir R. Storing and Retrieving Software Components: A Refinement Based System. In: *Proc. 16th ICSE*, IEEE Computer Society Press, 1994. 91~100
- 20 Kontio J. Case Study in Applying a Systematic Method for COTS Selection. In: *Proc. of the 18th Intl. Conf. on Software Engineering*. Berlin, Germany, Los Alamitos, CA: IEEE Computer Society Press, 1996. 1996. 201~209
- 21 Morel J M, Faget J. The REBOOT Environment. BULL. S. A. Rue Jean JAURES, F-78340 LESCLAYES-SOUS-BIOS, France
- 22 杨燕燕. 基于数据库技术的构件库系统研究. [博士研究生学位论文]. 北京大学计算机科学技术系, 1999
- 23 Reuse Library Interoperability Group. RIG Uniform Data Model for Reuse Libraries (UDM); [RPS-0002], 1994
- 24 Sorumgard L S, Sindre G, Stokke F. Experiences from Application of a Faceted Classification Scheme. In: *Proc. Reuse '93*, Lucca, Italy, 1993. IEEE CS Press, 1993
- 25 Chichester E-A. *Software Reuse: A Holistic Approach - Measuring the Effect of Reuse Chapter*. New York: Wiley, 1995. 113~180
- 26 Ye Yunwen. An Active and Adaptive Reuse Repository System. In: *Proc. of 34th Hawaii Intl. Conf. on System Sciences (HICSS-34)*, Software Technology Track, Maui, HI. IEEE Press, CD-ROM, 2001. 10
- 27 Guo J, Luqi. A Survey of Software Reuse Repositories. In: *Proc. of the IEEE Intl. Conf. and Workshop on the Engineering of Computer Based Systems (IEEE ECBS'2000)*, Edinburgh, Scotland, UK, 2000
- 28 Leymann F. *Web Services Flow Language (WSFL 1.0)*, IBM Software Group, 2001