青鸟网上构件库系统的访问控制机制*

邹 炜, 孙家骕, 孙艳春

(北京大学 计算机科学技术系,北京 100871)
E-mail: {zouwei,sjs,sunyc}@cs.pku.edu.cn
http://www.pku.edu.cn

摘要: 青鸟网上构件库系统通过对可复用构件的管理,支持软件企业进行基于构件的软件开发.系统在提高开放性的同时,往往会带来安全性以及知识产权方面的问题.为了解决这些问题,在青鸟网上构件库系统中采用基于角色的访问控制机制,并将构件描述信息与构件实体区别对待.为构件库系统及其所管理的构件定义了用户、角色、权限和角色继承,满足了安全性、知识产权管理等多种需求,同时也保证了系统的效率和对复用的支持.

关 键 词: 软件复用;基于构件的软件开发;构件库系统;访问控制;基于角色的访问控制
中图法分类号: TP311 文献标识码: A

随着软件复用技术的迅速发展,基于构件的软件开发(component-based software development,简称 CBSD)因其对软件复用的有效支持而逐渐得到认可.CBSD 通过组装的方式,在软件开发过程的各个阶段复用现有的相关构件,以建造大型软件系统.该方法可以降低软件开发成本,加快系统开发的速度,减轻大型系统的维护负担[1].基于构件进行软件开发必须具备两个基本条件,即软件生命周期各阶段所对应的大量构件以及一个提供有效构件管理的复用环境.在 CBSD 中,构件库系统是一个至关重要的部分,它可以减小构件的复用代价,是实施软件复用的必备设施.

青鸟网上构件库系统(Jade Bird Web component library system,简称 JBWCL)以因特网为基础设施,支持对构件的管理、存储、检索等.作为一个开放式的系统,其安全性和软件知识产权问题尤为重要.本文介绍了 JBWCL 中的访问控制机制.第 1 节概述 JBWCL 及其对访问控制的需求.第 2 节简介基于角色的访问控制.第 3 节探讨了 JBWCL 中的访问控制.第 4 节为本文内容的总结.

1 青鸟网上构件库系统概述

1.1 JBWCL

作为一个支持复用的软件开发环境,青鸟 III 型系统的关键在于构件的有效管理和查询.其核心部分"青鸟构件库系统"通过支持对可复用构件的描述、管理、存储和检索,满足了 CBSD 的需要[2].

JBCL 是一个基于局域网的客户/服务器方式的构件库系统,适用于软件企业内部的构件存储和管理.随着企业规模的逐渐扩大、资源共享日益增多以及提供的服务种类不断增加,CBSD 的构件生产、复用和管理不再局限于一个软件企业内部,这就要求提高构件库系统的开放性,在更大范围内支持公共构件的存储和复用.另一方面,由于局域网构件库系统假设所有用户都使用同一个局域网,属于同一个企业或组织,彼此相互信任,所以 JBCL 的访问控制机制比较简单.系统开放性以及访问控制支持的不足,在一定程度上降低了系统的有效性.

* 收稿日期: 2000-03-23; 修改日期: 2000-07-18
基金项目: 国家"九五"科技攻关项目(96-729);国家 863 高科技发展计划资助项目(863-306-02-02-01)
作者简介: 邹炜(1973－),女,江西南昌人,博士生,主要研究领域为软件工程,数据库技术;孙家骕(1946－),男,吉林扶余人,教授,博士生导师,主要研究领域为程序设计语言,系统软件,软件工程;孙艳春(1970－),女,辽宁沈阳人,博士,讲师,主要研究领域为软件工程,CSCW.

鉴于 Internet 和 World Wide Web 技术的日益普及和飞速发展,我们将构件共享、提供服务与 Internet 底层结构结合在一起,以 JBCL 为基础设计实现了 JBWCL.JBWCL 借助因特网覆盖广阔的特点,使系统的应用超越地理位置的约束,从而有效地扩大了信息规模、拓展了系统的使用范围.

1.2 JBWCL中的访问控制

系统开放性的增强会在一定程度上影响安全性和软件知识产权.为了同时解决这 3 个方面的问题,JBWCL 的访问控制机制需要满足以下需求:

· 从安全性角度来看,构件库系统是一个多用户的环境,必须对用户的访问进行控制,其关键在于控制每一次访问的访问者、访问对象以及操作内容.

· 从知识产权角度来看,应将构件分为两个层次:构件描述层和构件实体层.为了维护知识产权,需要对构件实体层进行严格的访问控制,而为了推广复用则需要开放构件描述层.

· 从数据一致性角度来看,系统中的各种操作相互影响、彼此制约,必须综合考虑、协调制定整体策略,以防止对数据完整性和一致性的破坏.

· 从复用角度来看,构件库系统的根本目的在于支持复用者便利地查询、获取可复用构件.

· 从管理维护角度来看,构件库系统是一个开放的构件共享环境,在保证安全性和数据完整性的同时,应尽可能地降低系统管理负担、提高系统效率.

· 从用户特性角度来看,构件库系统的用户群庞大且动态变化,系统管理员相对较少,需要系统本身提供有力的支持,以减少管理的错误和代价.

· 从资源特性角度来看,构件库系统需要维护的构件数量巨大,其中的数据对象基于作业实现共享.就每一种作业而言,其内涵变化较少.

尽管现有的 Web 服务器提供了一定的权限管理功能,但并不能满足 JBWCL 的需求,必须采用专门的访问控制机制.

2 基于角色的访问控制

存在两种传统的访问控制机制:自主型访问控制(discretionary access control,简称 DAC)和强制型访问控制(mandatory access control,简称 MAC).DAC 允许用户自主地将访问权限授予其他用户,只能控制直接访问而无法控制间接访问.它以降低资源安全性为代价提供了较大的灵活性,适用于一般的商业机构和民间组织,而不能满足 JBWCL 的需求.

作为更强有力的访问控制手段,MAC 由授权机构为主体和客体分别定义固定的访问属性,用户无权进行修改.主体权限反映了信任程度,客体权限则与其所含信息的敏感度一致,通过二者的比较来判断访问的合法性[3].MAC 适用于军方的多极安全机制,灵活性较差,也无法满足 JBWCL 的需求.

基于角色的访问控制(role-based access control,简称 RBAC)是一种非自主的访问控制机制,支持对特定安全策略进行集中管理.其非自主性表现在用户并不“拥有”所访问的对象,换言之,用户并不能任意地将自己拥有的访问权限授予其他用户[4~6].RBAC 是对 DAC 和 MAC 的改进,基于用户在系统中所起的作用设置其访问权限.与 DAC 相比,RBAC 以非自主性取代自主性,提高了系统安全性.与 MAC 相比,RBAC 以基于角色的控制取代基于用户的控制,提高了系统的灵活性[7].

RBAC 采用与企业组织结构一致的方式进行安全管理,如图 1 所示.其基本思想是:在用户与角色之间建立多对多关联,为每个用户分配一个或多个角色;在角色与权限之间建立多对多关联,为每个角色分配一种或多种操作权限;同时,通过角色将用户与权限相关联,即当用户拥有的一个角色与某权限相关联时,用户拥有该权限[5].

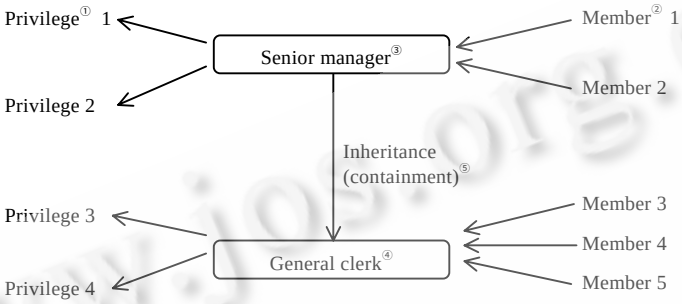
RBAC 中引入了一个新概念,即角色继承,通过支持角色之间的继承关系(包含关系),从而使角色形成一个层次结构[4].若角色甲继承(包含)角色乙,则甲拥有乙的所有权限.RBAC 采用继承机制实现角色层次结构,符合人的自然思维方式和企业的自然组织结构.例如,在一个企业中,高层管理员往往拥有普通职员的所有权限,通

常的做法是为普通职员分配其拥有的全部权限(权限 3、权限 4),而只为高层管理员分配其特有的那些权限(权限 1、权限 2),缺省则认为他拥有普通职员的全部权限,如图 2 所示.与此对应,在 RBAC 系统中可定义两个角色:高层管理员角色和普通职员角色,前者继承(包含)后者.

①用户,②角色,③权限.

Fig.1 Relation among user, role and privilege

图 1 用户、角色与权限的关系



①权限,②成员,③高层管理员,④普通职员,⑤继承(包含).

Fig.2 Role inheritance

图 2 角色继承

RBAC 遵循以下原则:

(1) 最小权限原则:用户只享有足以完成其职责的权限,不允许拥有超出此范围的任何权限.

(2) 责任分离原则:目的在于为不同角色进行任务和相关权限的划分,以便有效地防止用户相互勾结.依责任的互斥程度,可以分为两种形式:

(2a) 静态责任分离(static separation of duty constraint,简称 SSD):不能将某两个角色分配给同一个用户,例如银行系统中的出纳员角色和审计员角色;

(2b) 动态责任分离(dynamic separation of duty constraint,简称 DSD):允许将某两个角色分配给同一个用户,但是不允许任何一个用户在同一个事务处理过程中担任这两个角色,例如出纳员角色和帐户持有人角色.

(3) 角色集最大值限定原则.一个角色的成员个数不允许超出规定的最大值[4,8].

3 JBWCL 的访问控制机制

构件库系统由系统管理部分和构件部分共同构成,显然,其访问控制也应该分为两个方面,一方面是对系统管理部分的访问控制(简称系统访问控制),另一方面是对构件的访问控制.二者均采用基于角色的访问控制策略,但是,由于其访问的性质不同、所控制的用户范围和所访问的资源各异,因此,在具体进行访问控制时需要区别对待.

构件访问控制包括构件查询访问控制(构件检索与下载)和构件管理访问控制(构件的增加、删除、修改等).构件查询是与复用者相关的操作;而构件管理则是与构件提交者和系统管理员相关的操作,与复用者无关.构件管理既不能够对所有用户开放,也不需要进行多级访问控制.构件管理访问控制与系统访问控制有相似性,因此在第 3.2 节中将二者合并进行介绍.

3.1 构件库系统的访问控制模型

JBWCL 采用 RBAC 进行访问控制.基于角色的构件库访问控制(role-based component library access control,简称 RBCLAC)模型构造如下:

· 用户(users):JBWCL 的所有访问者,即 CBSD 涉及到的所有人员,包括构件的开发者、构件的复用者以及构件的管理者.

·角色(roles):JBWCL 中的任务集合,代表一组允许执行的操作和/或操作的对象.用户对构件库的访问表现为对库中某些对象进行操作,一个用户的角色界定了他究竟能够执行哪些操作.

· 权限(privileges):用户可以进行的操作以及操作的对象.关于权限有两种定义:一种是将权限定义为一个操作以及该操作访问的一组数据,即在权限内部进行数据访问控制,这一策略要求在设计时将操作和数据进行绑定;另一种方式是将权限定义为操作,而不包括其访问的数据,需要单独定义一套规则来管理用户对资源的访问.JBWCL 选择第 1 种策略,在权限内部进行操作与数据之间的绑定,很好地解决了安全问题.

· 角色继承(role inherits:roles×roles):角色之间的继承关系.若两个角色具有继承关系,则子角色具有父角色的全部属性.

· 角色-权限关联(roles×privileges):角色与权限之间的多对多关联.每个角色可享有多种权限,从而对构件库系统进行不同的操作;同时,每一种权限可属于多个角色.

·用户-角色关联(users×roles):用户与角色之间的多对多关联.JBCWL 为用户分配角色,用户通过自己所扮演的角色来享有对构件库的访问权限.每个用户可同时扮演多个角色,以多种不同的身份对构件库进行操作;并且允许多个用户扮演同一个角色.

· 静态责任分离(SSD:roles×roles):角色之间的静态责任分离关系.具有 SSD 关系的两个角色是互斥的,其责任完全冲突,两个角色必须绝对分离,不允许同时分配给任何一个用户.

· 动态责任分离(DSD:roles×roles):角色之间的动态责任分离关系.若两个角色具有 DSD 关系,则它们的责任在某种程度上发生冲突.也就是说,在静态时可以允许共存,而在任何一次访问的动态过程中,两种角色互斥.

· 角色集最大值限定(cardinality):cardinality(r)代表角色 r 所允许的最大成员数.有些角色可以由任意多个用户来扮演,而另外一些角色,尤其是对构件库影响和破坏能力比较大的角色,其用户数应该得到控制.

· 最小权限原则:构件库系统的所有用户均不拥有超出其职责范围的任何权限.禁止一切非法的、超越权限之外的访问[4,9~12].

3.2 系统访问控制以及构件管理访问控制

本节介绍了对系统和构件管理的访问控制.二者策略一致,均采用基于角色的方式.

· 用户:系统管理员以及构件提交者,不包括构件复用者.

· 角色:定义角色的过程基于对构件库系统运作方式的全面分析.例如,在系统中可以定义如下角色:构件提交者、构件验证者、构件描述信息管理者、构件实体管理者、刻面管理者、用户信息管理者、超级管理员等等.各个角色具有不同的权限,相互协作,共同维护着构件库系统的正常运作.每个用户都可以担任其中之一或者兼任多职.

·权限:对权限的设置和划分由操作的对象决定.JBWCL 构件管理涉及构件的所有内容,包括构件的一般性描述、形式化描述、分类描述信息以及作为复用实际内容的构件实体;而系统访问的对象则包括用户信息、访问权限等.相应地,设置权限如下:维护构件基本属性、维护构件规约、维护刻面及其术语空间、维护构件实体、维护用户个人信息、分配访问权限等.

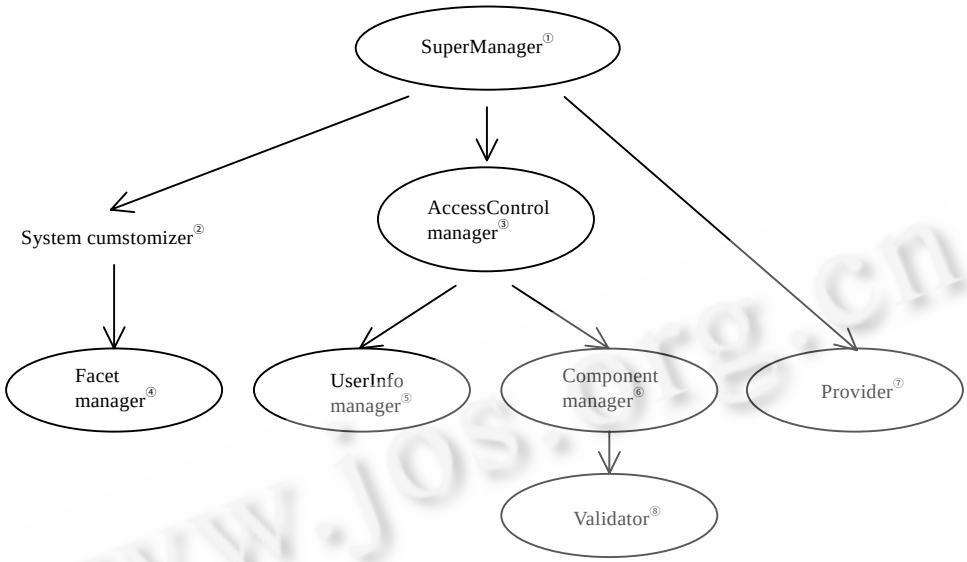
· 角色继承:JBWCL 的访问控制支持角色之间的继承关系.图 3 给出了部分角色之间的继承关系.例如,角色“系统定制者”继承角色“刻面管理者”,即系统定制者除了自身的特定权限之外,还拥有刻面管理者的所有权限.

· 静态责任分离:例如,在构件提交者与构件验证者这两个角色之间满足 SSD,具体表现为不允许一个用户既是构件提交者和又是构件验证者.

· SSD:Provider×validator.

· 角色集最大值限定:举例来说,系统规定超级管理员角色集的最大值为 1,只允许有一个超级管理员,即 Cardinality(SuperManager)=1.

· 最小权限原则:所有用户都只拥有其角色所赋予的权利.例如,一个刻面管理者只能够维护刻面及其术语空间,不能进行任何其他操作.



①超级管理员,②系统定制者,③构件访问控制管理者,④刻面管理者,
⑤用户信息管理者,⑥构件管理者,⑦构件制作者,⑧构件验证者.

Fig.3 Role inheritance in component library system

图 3 构件库系统的角色继承

3.3 构件查询访问控制

构件是复用的核心,也是构件库系统的核心.构件的知识产权是软件企业的宝贵资产.维护构件的知识产权是软件企业参与竞争、保持优势、开拓并占领市场的重要手段,从构件知识产权角度考虑,应对构件复用加以限制.而另一方面,查询构件的目的在于检索、选择合适的构件进行复用.为了支持和推广复用,应该将构件全面开放.如何解决这个矛盾,成为关于构件查询访问控制的一个核心问题.

作为支持构件复用的基础设施,构件库系统应当提供一套二者兼顾的控制机制,在大力提高构件复用程度和复用范围的同时,严格维护系统安全性和构件知识产权.为此,将构件查询目标划分为两个层次:构件描述信息层和构件实体层.构件描述信息包括构件的基本属性、关键词、形式化表示、分类信息等各种描述信息,而构件实体才是真正可复用的内容.

为了有效地支持复用,允许复用者任意查询构件描述信息,以便使复用者全面了解构件库系统中储存了哪些构件,并最终选择自己所需要的构件;同时,从安全性以及构件产权的角度考虑,对直接用于复用的构件实体的下载进行严格控制,杜绝构件的非法使用.与构件的两个层次相对应,将构件查询访问控制划分如下:对所有用户开放构件描述层访问,而对构件实体访问则要依据安全级别进行严格控制.下面介绍构件实体的访问控制模型.

· 用户:构件库系统的所有用户,包括系统管理员、构件提交者以及构件复用者.

· 角色:依据安全级别进行划分.每个角色代表一个特定的安全级别,拥有一组特定的权限.

· 权限:对构件实体的访问控制中只有一种操作,即获取构件实体.划分权限的依据为操作对象的分组,每个权限代表对一组构件实体的查询操作.

· 角色继承:角色之间存在继承关系.角色之间依据其对应的安全级别形成一个层次关系.每个角色的权限由自身拥有的权限和继承而来的权限两部分组成.

· 最小权限原则:每个用户只拥有其角色所赋予的权利,任何一个用户都不能访问超越自己权限级别之外的构件实体.

在构件实体访问控制中,管理任务主要是分配或撤销系统中指定角色集合的成员.当构件库系统新增加一个用户时,管理员为其分配一个或多个角色,使其拥有对某些构件实体的访问权限.当某用户对应于构件实体的安全级别发生变化时,管理员将其从原有角色的成员中删除,并授予新的角色.最后,当需要从构件库系统中删除一个用户时,只需将其从所有角色的成员中删除即可.

4 结束语

JBWCL 采用因特网作为基础设施,借助因特网覆盖广阔的特点,使系统的应用超越地理位置的约束,提高了系统的开放性.为了保证系统的安全性和构件的知识产权,需要有一套健全的访问控制机制.在研究和比较了各种访问控制之后,我们发现,采用 RBAC 能够满足 JBWCL 的安全性和知识产权管理的多种需求.RBAC 是近年来在访问控制领域的一个研究热点.它支持安全性的 3 个原则:最小权限、责任分离和数据抽象,同时又基本上保证了系统的效率.

当前版本的 JBWCL 采用基于角色的访问控制机制,为构件库系统及其管理的构件定义了用户、角色、权限和角色继承,并且将构件描述信息层与构件实体层独立进行管理.JBWCL 已经在北京大学青鸟公司以及山东浪潮通用软件公司安装并成功试用,能够满足公司的安全性和知识产权管理的需求.

进一步的工作主要包括:① 继续在多个软件企业中试用 JBWCL,完善角色定义及权限分配;② 支持用户对 RBAC 的定制,从而提供更大的灵活性.

References:

[1] Brown, A.W., Wallnau, K.C. Engineering of component-based systems. In: Component-Based Software Engineering: Selected Papers from the Software Engineering Institute. Los Alamitos, CA: IEEE Computer Society Press, 1996. 7~15.

[2] Li, Ke-qin, Guo, Li-feng, Mei, Hong, *et al*. An overview of JB (Jade Bird) component library system JBCL. In: Chen, Jian, Li, Ming-shu, Mingins, C., *et al.*, eds. Proceedings of the 24th International Conference TOOLS Asia. Los Alamitors, CA: IEEE Computer Society Press, 1997. 261~267.

[3] DOD (U.S. Dept. of Defense). Trusted Computing System Evaluation Criteria. DOD 5200, 28-STD, 1985. http://www.fas.org/irp/nsa/rainbow/std001.htm.

[4] Ferraiolo, D.F., Kuhn, R.D. Role-Based access control. In: Proceedings of the 15th NIST-NSA National Computer Security Conference. Baltimore, MD: ACM Press, 1992. 13~16. http://www.itl.nist.gov/div893/projects/893-00-access.html.

[5] Ferraiolo, D.F., Cugini, J.A., Kuhn, R.D. Role based access control: features and motivations. In: Proceedings of the 11th Annual Computer Security Applications Conference. Los Alamitors, CA: IEEE Computer Society Press, 1995. 22~31.

[6] Sandhu, R., Coyne, E., Feinstein, H., *et al*. Role-Based access control models. IEEE Computer, 1996,29(2):38~47.

[7] Barkley, J. Comparing simple role based access control models and access control lists. In: Proceedings of the 2nd ACM Workshop on Role Based Access Control. Fairfax, Virginia: ACM Press, 1997. 127~132. http://dev.acm.org/pubs/citations/proceedings/commsec/266741/p127-barkley/.

[8] Mohammed, I., Dilts, D.M. Design for dynamic user-role-based security. Computers and Security, 1994,13(8):661~671.

[9] Gavrila, S.I., Barkley, J.F. Formal specification for role based access control user/role and role/role relationship management. In: Proceedings of the 3rd ACM Workshop on Role-Based Access Control. Gavrila, Barkley: ACM Press, 1998. 81~90.

[10] Ferraiolo, D.F., Barkley, J., Kuhn, R.D. A role based access control odel and reference implementation within a corporate Intranet. ACM Transactions on Information Systems Security, 1999,1(2):34~64.

[11] Luihi Giuri. Role-Based access control for the Web using Java. In: Proceedings of the 4th ACM Workshop on Role Based Access Control. Fairfax, Virginia: ACM Press, 1999. 11~18.

[12] Kuhn, R.D. Mutual exclusion of roles as a means of implementing separation of duty in role-based access control systems. In: Proceedings of the 2nd ACM Workshop on Role Based Access Control. Fairfax, Virginia: ACM Press, 1997. 23~30.

Access Control in Jade Bird Web Component Library System*

ZOU Wei, SUN Jia-su, SUN Yan-chun

(*Department of Computer Science and Technology, Beijing University, Beijing* 100871, *China*)

E-mail: {zouwei,sjs,sunyc}@cs.pku.edu.cn

http://www.pku.edu.cn

Abstract: Jade Bird web component library system (JBWCL) can support the management of software components, thereby facilitating component-based software development in software enterprises. However, the system brings the problem of security and copyright, at the same time it improves openness. To solve these problems, the role-based access control is employed in JBWCL, and the component entity is separated from its description in this paper. The user, role, phrivilege and role hierarchy for the system and those components stored in it are defined. The mechanism meets the requirement of security and copyright, meanwhile ensure the efficiency of the system and the support to reuse.

Key words: software reuse; component-based software development; component library system; access control; role-based access control

* Received March 23, 2000; accepted July 18, 2000

Supported by the Key Sci-Tech Project of the National 'Ninth Five-Year-Plan' of China under Grant No.96-729; the National High Technology Development 863 Program of China under Grant No.863-306-02-02-01

敬 告 作 者

《软件学报》创刊以来,蒙国内外学术界厚爱,收到许多高质量的稿件,其中不少在发表后读者反映良好,认为本刊保持了较高的学术水平.但也有一些稿件因不符合本刊的要求而未能通过审稿.为了帮助广大作者尽快地把他们的优秀研究成果发表在我刊上,特此列举一些审稿过程中经常遇到的问题,请作者投稿时尽量予以避免,以利大作的发表.

1. 读书偶有所得,即匆忙成文,未曾注意该领域或该研究课题国内外近年来的发展情况,不引用和不比较最近文献中的同类结果,有的甚至完全不列参考文献.

2. 做了一个软件系统,详尽描述该系统的各个方面,如像工作报告,但采用的基本上是成熟技术,未与国内外同类系统比较,没有指出该系统在技术上哪几点比别人先进,为什么先进.一般来说,技术上没有创新的软件系统是没有发表价值的.

3. 提出一个新的算法,认为该算法优越,但既未从数学上证明比现有的其他算法好(例如降低复杂性),也没有用实验数据来进行对比,难以令人信服.

4. 提出一个大型软件系统的总体设想,但很粗糙,而且还没有(哪怕是部分的)实现,很难证明该设想是现实的、可行的、先进的.

5. 介绍一个现有的软件开发方法,或一个现有软件产品的结构(非作者本人开发,往往是引进的,或公司产品),甚至某一软件的使用方法.本刊不登载高级科普文章,不支持在论文中引进广告色彩.

6. 提出对软件开发或软件产业的某种观点,泛泛而论,技术含量少.本刊目前暂不开办软件论坛,只发表学术文章,但也欢迎材料丰富,反映现代软件理论或技术发展,并含有作者精辟见解的某一领域的综述文章.

7. 介绍作者做的把软件技术应用于某个领域的工作,但其中软件技术含量太少,甚至微不足道,大部分内容是其他专业领域的技术细节,这类文章宜改投其他专业刊物.

8. 其主要内容已经在其他正式学术刊物上或在正式出版物中发表过的文章,一稿多投的文章,经退稿后未作本质修改换名重投的文章.

本刊热情欢迎国内外科技界对《软件学报》踊跃投稿.为了和大家一起办好本刊,特提出以上各点敬告作者.并且欢迎广大作者和读者对本刊的各个方面,尤其是对论文的质量多多提出批评建议.