软件构件的可信保证研究

郭树行　兰雨晴　金茂忠
（北京航空航天大学软件工程研究所　北京 100083）

摘　要　近年来，可信构件的研究逐渐引起软件工程领域的重视。可信构件研究与应用的目标是为了给基于构件的软件工程(CBSE)提供坚实的基础，而方法就是通过扩展与完善可信重用的软件构件(可信构件)库。构件的可信来源于可信保障技术的应用，如：契约设计的使用、正确性的数学证明、软件测试、详细的代码走查、基于度量的评估、实际项目的验证、严格的变更管理等。本文通过分析可信构件研究的若干领域，总结出构件可信性的3个角度，探讨了可信构件研究的不足之处，并分析其原因。作为总结，给出了可信构件领域研究需要解决的若干问题。

关键词　可信构件，可信构件质量模型，构件验证，形式化开发

Some Issues about Trusted Components Research

GUO Shu-Hang　LAN Yu-Qing　JIN Mao-Zhong
(Software Engineering Institute, Beinang University, Beijing 100083)

Abstract　Trusted component is emerging as one of the key research areas in software engineering recently. The research and application aimed at providing a solid foundation for the software industry through extensive libraries of trusted reusable software components, where trust is built from a combination of approaches: use of Design by Contract; mathematical proofs of correctness; testing; wide public scrutiny; evaluation through metrics; exhaustive validation in practical projects; rigorous change management. The concept of trusted component is brought up based on analyzing and comparing the several quality model definitions about software component. Based on summing up the activities about trusted components, two categories of study about it are extracted out, and the advancements of researches of it are subsequently introduced from three aspects. Additionally, some disadvantages of study on it are discussed, and the causes are explained at the same time. Finally, it is concluded with some significantly promising problems about research on trusted component.

Keywords　Trusted component, Quality model of trusted component, Components certification, Formal development

1　可信构件的起源与发展

基于可信构件的软件复用作为一种提供软件生产率和软件质量的有效途径，是解决软件危机的必然选择，已经成为目前软件工程领域研究重要热点之一[5,13]。在安全关键应用中使用第三方软件构件(COTS)，并不像常规应用那样普及。相对于电子部件，在软件领域缺乏一种质量评估与保证的方法，使得软件开发者很难将可信性“打造”进软件构件。与此同时，电子设备已经有了一系列度量电子部件的质量指标[32]，如可靠性、性能、应用领域等，然而关于软件构件的度量指标尚没有形成一致意见，实际中软件构件的演化能力和环境适应性也比较差。

可信构件作为可重用的软件元素，往往伴随着用户对高质量的强烈期望，而且必须提供一系列令人信服的参数。构件的可信性包含3个方面的因素：从技术角度有契约设计、充分测试、合适的编程语言、形式化证明等；从管理的角度有CMM等系统过程；从社会的角度是用户对构件开发者的信任程度。“可信构件”这一观点，在1998年IEEE的文献中作为面向对象技术提出。一些相关研究学者，于1998年在澳大利亚墨尔本Monash大学召开的“TOOLS PACIFIC 1998”会议上成立了第一个可信构件研究小组，正式将可信构件研究引入到软件工程领域。

1.1　可信构件的发展现状

从软件过程的角度，可信构件的复用包含3个阶段，即可信构件的生产、管理与复用。可信构件的复用技术(基于构件的组装开发技术)发展相对比较成熟，形成了以体系结构为核心的构件组装过程[10]。形成这一现象的主要原因[2,4]，在研究方面，现代软件工程思想，特别是对复用技术的强调；在软件工业角度，支持用构件来构造GUI、数据库和应用部件的实际可用技术及理论的成功；在构件模型角度，出现了支持主流开发技术的构件模型，如CORBA，COM与EJB开发者的推动；从软件界，对象技术的广泛使用，提供了建造和使用构件的概念基础[31]和使用工具。

然而对于可信构件的生产技术，即面向构件的可信保证，目前尚没有实用、成熟的技术用来生产完整的可信构件[2]。到目前为止，主要的构件可信保证技术包括：①契约设计。该方法从软件设计的角度，从软件设计开始就确保软件的可信性，根据契约将可信作为需要进行质量保证的属性参数的集合，依托目标可信参数进行软件需求分析、设计、实现、测试及文档化等；②形式化验证。依托契约设计的约束规则，进行形式化的验证技术，出现了诸如DEEM[20]、DEPEND[21]、HIDE[22]、OpenSESAME[23]等技术或工具；③面向对象技术的应用[25]与

郭树行　博士研究生，主要研究领域为需求工程、可信软件过程；**兰雨晴**　博士生，主要研究领域为软件测试、软件项目管理；**金茂忠**　教授，博士生导师，主要研究领域为软件测试、软件开发环境。

可重用库设计规则的使用[15]；④详细的代码走查。在实践中，往往被证明是非常有效的；⑤充分测试。利用契约设计的优势，把测试焦点定位在构件的重用性；⑥有效度量。跟踪构件可信属性参数的迁移变化，使其处于可控范围之内。

1.2 可信构件的主要研究方向

可信构件作为具有严格的质量保证的可重用构件[1]，其目标就是使软件开发建立在坚实的基础之上，有效解决软件生产率和软件质量的矛盾，即软件危机。基于构件的软件开发范型，必须将软件重用技术、软件质量保证技术相融合，可信构件的研究与发展也自然成为软件工业发展的潜力所在。到目前为止，关于构件的可信性保证技术的研究方面，可以通过二次直角正交的方法进行划分，即管理与技术、事先与事后。

第一类划分方法从产品和过程角度出发考虑对构件可信性有影响的因素。许多研究学者认为软件工程问题，从根本上来说，是管理问题，技术是次要的，因而从本质来讲，形成了面向软件过程的管理技术，如软件能力成熟度模型(CMM)、ISO 验证、六 Sigma 等。相对应，另外一些研究学者更多的研究侧重于构件的可信保证技术，如增强程序设计语言的可靠性、开发工具支持技术、测试技术、形式化开发方法、设计模式等。

第二类划分方法从软件生产阶段和初始软件版本出现后的阶段，研究其所应包含的可信性保证技术，形成事先与事后两种。事先保证技术有助降低软件设计的缺陷，如形式化的设计方法、软件建模设计属于此类。事后技术往往应用到软件的初始版本设计完成之后，用于验证和弥补事先技术的保证构件可信性的不足。

然而，关于构件可信性保证技术也不能绝对完全按照二次直角正交的方法进行唯一划分，如关于软件配置管理属于技术和管理两个范畴，契约设计也跨越了事先事后两个阶段[24,25]。总的来说，构件可信性的保证技术给出了一个基本可行划分方法：二次直角正交法。

2 可信构件的研究现状

2.1 可信构件的定义

软件工程的基本目标是生产具有正确性、可用性及开销合宜(合算性)的产品。正确性意指软件产品达到预期功能的程度；可用性意指软件基本结构、实现及文档达到用户可用的程度；开销合宜意指软件开发、运行的整个开销满足用户的需求。以上目标的实现不论在理论上还是在实践中均存在很多问题有待解决，制约了对过程、过程模型及工程方法的选取。分析传统产业的发展，其基本模式均是符合标准的零部件(构件)生产以及基于标准构件的产品生产(组装)，其中构件是核心和基础，“复用”是必须的手段。实践表明，这种模式是软件开发工程化、软件生产工业化的必由之路[35]。因此，软件产业的发展并形成规模经济，标准构件的生产和构件的复用是关键因素。实现软件复用的关键因素(技术和非技术因素)主要包括：软件构件技术(software component technology)、领域工程(domain engineering)、软件构架(software architecture)、软件再工程(software reengineering)、开放系统(open system)、软件过程(software process)、CASE 技术等，以及各种非技术因素，且各种因素是相互联系、相互影响的[36]。

不可否认，软件工程在过去 40 多年的发展对于软件质量产生了非常有益的影响，然而面对日益增长的软件产业发展的需求，软件质量保证技术仍有很多不足之处。软件复用已经发展得相对成熟，然而可信构件的质量保证技术发展到了关键阶段[1]。下面给出可信构件的定义。

定义 1(可信构件[1]) 是一种软件重用元素，执行指定的操作处理并且属性具备质量保证。“可信”概念在 1998 年 IEEE 文献中引入，但是这里的可信概念和可信计算中“安全可信”不同，后者属于构件所必需提供的质量属性的一种[3,12]。在可信构件中，“可信”包含了所有质量因素。

基于上述定义，本文认为可信构件从使用的角度必须满足 4 个条件：可以被其他的软件元素所使用；提供正式使用描述且充分满足软件设计开发者所用；要具备通用性(不限定被某些特定类型的软件元素所使用)；构件属性达到系统质量目标。

2.2 可信构件的质量模型

构件的质量模型是评估构件产品的标准，然而它并不面向生产软件构件的过程。目前关于构件质量模型研究已取得初步成果。2003 年澳大利亚 Bertrand Meyer 在波兰国际软件工程会议上提出了可信构件的 ABCDE 模型[1]；在国内，2003 年 10 月科技部批准成立了软件构件标准工作组，并于 2004 年参考 ISO/IEC 9126[26-29]关于产品质量模型的基础上，给出了软件构件两部分模型：a)内部质量和外部质量模型；b)使用质量模型。这些质量模型的出现为世界范围内进一步研究可信构件质量模型标准奠定了基础。

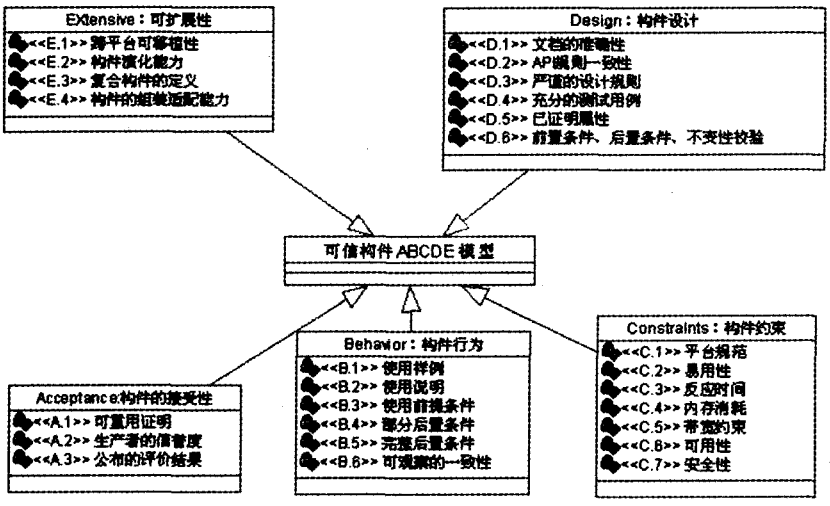

图 1 可信构件的质量模型

定义 2(可信构件的 ABCDE 模型[1]) 根据构件质量特性的特点将其分成正交的 5 个类别,构件评估时不存在统一的度量标准刻度。可信构件的 ABCDE 模型中:A,构件的接受性,从非技术角度提供构件可重用的特征;B,构件行为,提出成熟构件应具备的关键属性;C,构建约束,包含了对构件性能上的考虑;D,构件设计,从构件开发者角度关注构件内部质量特性,给出构件设计规则;E,构件扩展性,从构件使用者的角度提出构件的重用特征,不仅关注构件的重用,还关注构件演化兼容性。构件质量模型框架如图 1 所示。

定义 3(构件的内部与外部质量模型[30]) 将软件构件质量属性划分为 7 个特性(功能性、可靠性、易用性、效率、维护性、可移植性和可复用性),并进一步细分为若干子特性(图 2)。这些子特性可用内部或者外部度量来度量。在软件构件的 7 个质量特性中,由于前 6 个也是一般软件所应具有的,因此可以将它们看作软件构件的基本质量特性范畴,可复用性即是软件构件区别于一般软件的特殊的质量特性。

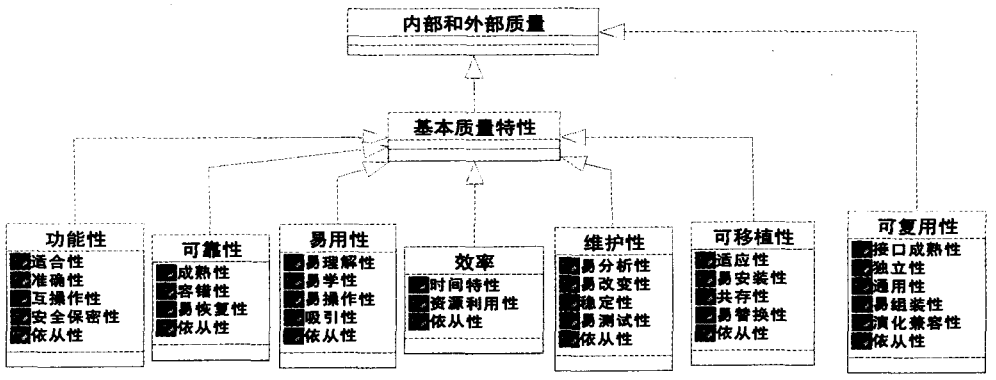

图 2 构件的内部与外部质量模型

定义 4(构件使用质量模型[30]) 使用质量的属性分类为 5 个特性:有效性、生产率、安全性、满意度和可信度(图 3)。

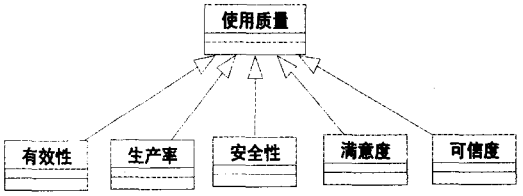

图 3 构件的使用质量模型

使用质量是从构件使用者角度所表现出来的构件质量特征,它的获得依赖于取得必需的外部质量,而外部质量则依赖于取得必需的内部质量(图 2)。度量通常在 3 个层次上都是需要的,因为满足内部度量准则的要求并不常常足以确保符合外部度量准则,而满足子特性的外部度量准则也不是常常足以保证符合使用质量准则。

比较上述不同构件质量模型可以发现,模型定义的角度不同,但是其目标都是为了提供构件可信性验证的标准。定义 2 给出了一个综合构件内部、外部、使用质量的可信质量模型框架,从 5 个角度正交划分可信构件应具备的属性,但是没有明确给出现有模型与实际软件质量特性、子特性之间的相互关系,也没有提出在此基础上进行有效定性与定量评估的有效方法。定义 3、4 分别从构件的内部与外部、使用质量等特性的角度给出了构件的质量模型,给出了现有质量模型与 ISO/IEC 9126 中质量特性之间的继承关系,同时指出构件的使用质量模型和内外部质量模型之间存在依赖关系,但是没有明确给出构件使用质量属性与内外部质量特性之间的约束关系,构件的使用质量特性之间也存在约束关系[16,17]。

因此在此基础上为了有效地对可信构件的质量属性进行综合评估,可以将构件属性间进行正交划分,给出构件属性与 ISO/IEC 9126 中质量特性之间的关系,利用有关质量特性综合评估构件属性可信度,利用构件属性的可信度综合评估每一个划分区间(构件维)的可信度。

2.3 可信构件研究的不同思路

可信构件已经成为解决软件生产率与软件质量矛盾的突破点。生产高质量的软件构件,即构件的可信保证也就成为研究的热点。在实际的软件开发领域,近年来出现了很多软件构件类型,从面向对象的类库(C++、JAVA、Eiffel)到 ActiceX 控件、COM 与 DCOM 对象、EJB 组件、.NET 组件。这些软件构件技术的进展较大程度上得益于质量保证技术,尤其是商业构件的构件验证技术。在构件的可信保证研究领域,近年来形式化方法方面已经取得很大进展[6~9,18,20~23],尤其是程序证明技术。这些研究成果的取得,使得规模适度的软件构件进行充分性证明成为可能。构件开发作为形式化开发的理想应用领域,软件复用技术与形式化开发方法的融合无疑成为研究者的目标[19]。

关于可信构件的研究可以分成两种思路:一种是面向可以获得较大研究进展的长期路线,一种是面向目前软件工程领域目前发展需要的短期思路。短期的研究路线立足于现在已有的构件类型、商业构件或开源构件。构件质量保证水平从本质上受到了客观的限制,由于商业构件的源代码透明,不可能通过程序证明的方法。在这种情况下,构件验证就成为研究的焦点,目标就是在已有可信构件质量模型基础上尽可能给出构件属性质量的综合评估方法。

与之对应,长期研究思路目标是为构件提供经过充分正确性证明的构件属性。这一目标的实现必须开始于粒度较小的原子构件。在获得可信原子构件之后,再进一步研究复合构件的可信性,逐步复合最终得到目标可信复合构件。总的来说,就是要通过正确性证明生产构件,通过证明构件满足了指定的属性目标。其中构件契约包含了构件接口的前置条件、后置条件与类别不变性,对于构件正确性证明起到了重要作用。构件契约[14]目前已经应用到分析设计、获得软件构件正确性、增强易读性、构件库管理、契约驱动的软件测试与调试。关于契约的理论与实践的发展,使得把契约作为构件属性成为可能,用以表征软件抽象及其要进行正确性证明的实现。

3 可信构件研究的不足之处

可信构件的重要性日益得到软件工业领域的认可。构件的可信保证作为未解决的关键问题，必须作为软件工程领域具有挑战性的研究目标之一。对于软件工程而言，无论其昨天还是今天，仍然没有能力进行工业化的软件生产。立足于可信构件的开发及其可重用构件的可信保证研究，目前面临的主要问题包括：

① 与通常的自上而下过程形式不同，基于构件的开发是一种自底向上的风格，必须提供使用有效的形式化证明技术，面向证明的开发方法必须应用到整个系统，同时也必须满足构件本身证明正确性的需求；

② 必须提供适当的构件可信建模技术，支持构件质量属性的模型检测，研究可信性的预测分析与构件建模方法交叉应用，其宗旨之一是将可信性的综合评估方法较系统地引入到构件建模环境中，减少基于构件的软件系统的设计缺陷，进一步保证和提高构件质量；

③ 完善目前的面向对象的契约设计规则，使其满足完整规格说明书的要求，必须被适配到支持可信构件形式化证明；

④ 必须提供统一可信构件的质量模型，为可信构件的验证提供统一标准，并且提供可信构件质量属性的综合评估方法；

⑤ 在可信原子构件的基础上研究可信复合构件的智能化生产技术等。

可信构件从本质上讲是对构件属性进行质量保证，是提供用户对构件信任程度的基础。可信构件的研究从工业化生产的角度分为3个范畴，即面向可信的构件元模型描述、构件属性的质量保证、构件的质量评估验证。总的来讲，基于构件的软件开发范型日趋成熟，支持构件开发软件过程的研究也取得不少研究成果[10]，然而可信构件本身的研究相对而言比较匮乏。本文认为，无论哪一种研究思路，必须解决构件元模型统一描述、给出统一构件质量模型、形式化开发方法与软件复用技术的有效融合，才能从根本上解决软件工业中生产率与软件质量的矛盾。

总结 系统构件化已经成为软件技术总体发展趋势之一[33]，可信构件的研究也自然成为具有重要意义的研究目标之一。可信构件的研究无论是面向软件技术还是过程管理，面向事先可信保证还是事后构件验证，其目标都是提高构件的可信度。可信构件的质量模型为构件可信验证提供标准，构件元模型为研究构件可信度与构件属性之间的关系奠定基础。构件可信度保证研究也必须建立在这两者基础上。

通过分析，构件可信性研究思路分为“短期”与“长期”两类阶段性研究，“短期”研究是“长期”研究的有益补充，“长期”研究也必然成为“短期”研究的最终目标。同时，在此基础上指出了可信构件研究的一些不足之处。本文认为，基于本体的可信构件的元模型描述、可信构件库的管理与检索、复核构件的智能设计、可信构件的契约设计是目前比较可行的一些研究课题。

参 考 文 献

1 Meyer B. The grand challenge of trusted components. In: Software Engineering, 2003. Proceedings, 25th International Conference on, May 2003. 660～667
2 Meyer B, Mingins C, Schmidt H. Providing trusted components to the industry. Computer, 1998, 31(5): 104～105
3 Lindqvist U, Olovsson T, Jonsson E. An analysis of a secure system based on trusted components. Computer Assurance, 1996. COMPASS '96, 'Systems Integrity. Software Safety. Process Security'. In: Proceedings of the Eleventh Annual Conference on, June 1996. 213～223
4 Jezequel J. Trusted Components 2nd Workshop on Trusted Components. Technology of Object-Oriented Languages and Systems, 1999. In: Proceedings of, June 1999. 422～422
5 Heineman G T, Councill B, Flynt J S, et al. Component-based software engineering and the issue of trust. In: Proceedings of the 22nd international conference on Software engineering, June2000
6 Chung L, Nixon B, Yu E, et al. Nonfunctional Requirements in Software Engineering. Kluwer Academic Publishers, 2000
7 Cysneiros L M, Leite J C S P, Neto J M S. A Framework for Integrating Non-functional Requirements into Conceptual Models. In: RE'01, 2001
8 Cysneiros L M, Leite J C S P. Nonfunctional Requirements: From Elicitation to Conceptual Models. IEEE Transactions on Software Engineering, May 2004
9 Gross D, Yu E. From Non-Functional Requirements to Design Through Patterns. Requirements Engineering Journal, 2001, 6
10 Mei H, Chen F, Feng Y D, et al. ABC: An architecture based, component oriented approach to software development. Journal of Software, 2003, 14(4): 721～732
11 Roshandel R, Medvidovic N. Multi-View Software Component Modeling for Dependability. Computer Science Department University of Southern California Los Angeles, CA 90089-0781, USA
12 Howden W E, Huang Yudong. Software trustability. In: Software Reliability Engineering, 1994. In: Proceedings, 5th International Symposium on, Nov. 1994. 143～151
13 Meyer B. The Significance of Components in Software Development. http://www. sdmagazine. com/documents/s = 7207/sdm9911k/, consulted February 2003
14 Meyer B. Contracts for Components in Software Development. http://archive. eiffel. com/doc/manuals/technology/bmarticles/sd/contracts. html, consulted February 2003
15 Brown A. Large-Scale Component-Based Development. New Jersey: Prentice Hall, Inc, 2000
16 Nicola G. Understanding, building and using ontologies: A commentary to "using explicit ontologies in KBS develop2ment". International Journal of Human2Computer St, 1997, 46(2-3): 293～310
17 Fluit C, Sabou M, Harmelen F. Ontology2based information visualization. In: Proceedings of Visualising the Semantic Web (VSW 2002), Springer2Verlag, 2002. 546～554
18 Chao Pingyi, Chen Tsungte. Analysis of assembly through product configuration[J]. Computers in Industry, 2001, 44: 189～203
19 Meyer B. The next Software Breakthrough. IEEE Computer, July 1997, 30(7): 113～114. http:// archive. eiffel. com/doc/manuals/technology/bmarticles/sd/contracts. html, consulted February 2003
20 Bondavalli A, Mura I, Chiaradonna S, et al. DEEM: a tool for the dependability modeling and evaluation of multiple phased systems. Dependable Systems and Networks, 2000. DSN 2000. In: Proceedings International Conference on, June 2000. 231～236
21 Ries G, Kalbarczyk Z, Kraljevic T, et al. DEPEND: a simulation environment for system dependability modeling and evaluation. In: Computer Performance and Dependability Symposium, 1996, Proceedings of IEEE International, Sept. 1996. 54
22 PDCC Pisa. HIDE - High-level Integrated Design Environment for Dependability was carried out by FAU Erlangen. TU Budapest, Intecs Sistemi Pisa and MID GmbH Nuremberg
23 Walter M, Trinitis C, Karl W. OpenSESAME: an intuitive dependability modeling environment supporting inter-component dependencies. Dependable Computing, 2001. In: Proceedings. 2001 Pacific Rim International Symposium on, Dec. 2001. 76～83
24 Meyer B. Applying "Design by Contract". In: Rine D, ed. Object-Oriented Systems and Applications, IEEE Computer Press, 1994
25 Meyer B. Object-Oriented Software Construction. second edition. Prentice Hall, 1997
26 GB/T 16260.1-ISO/IEC 9126-1: 2003《软件工程-产品质量》第一部分《质量模型》
27 GB/T 16260.2-ISO/IEC 9126-2: 2003《软件工程-产品质量》第二部分《外部度量》
28 GB/T 16260.3-ISO/IEC 9126-3: 2003《软件工程-产品质量》第三部分《内部度量》
29 GB/T 16260.4-ISO/IEC 9126-4: 2004《软件工程-产品质量》第四部分《使用质量度量》
30 信息产业部软件构件标准工作组. 软件构件质量模型, V0.9, 2004.12
31 王良斌, 朱国进. 本体论与构件复用. 计算机工程与应用, 2004, 14: 53～56
32 江建慧. 可信性指标体系. 同济大学. http://www. plinux. org/teach/jhj/dep/
33 杨芙清. 软件工程发展思索. 软件学报, 2005(1)
34 杨芙清, 梅宏, 李克勤. 软件复用与软件构件技术. 电子学报, 1999, 27(2): 68～75
35 杨芙清. 软件复用及相关技术. 计算机科学, 1999, 26(5): 1～4