

大型软件密集型系统的可管理性研究

宗平¹, 郭剑²

(1. 南京邮电大学 海外教育学院, 江苏 南京 210046;
2. 南京邮电大学 计算机学院, 江苏 南京 210046)

摘要:随着软件密集型系统的大量建设,相应的管理任务难度日益凸显。文中针对此类问题将系统体系(SoS)的思想应用到系统的管理之中,对可管理性进行描述和明确了系统的管理任务,提出了一种提高系统可管理性的方法。通过简化系统体系的层次,运用分层管理与增加智能管理组件,设计了一种通用的智能管理组件架构,通过对系统可管理性定量分析方法的讨论,提出按照不同的管理目标和不同的组成部分来分别加权,说明定量分析方法的正确性和有效性。

关键词:软件密集型系统;系统体系;可管理性;管理组件;定量分析

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2012)09-0099-04

Study of Manageability for Software-intensive Systems

ZONG Ping¹, GUO Jian²

(1. College of Overseas Education, Nanjing University of Posts & Telecommunications, Nanjing 210046, China;
2. College of Computer, Nanjing University of Posts & Telecommunications, Nanjing 210046, China)

Abstract: With the development of a large number of software-intensive systems, the corresponding difficulty for management tasks become increasingly prominent. It applies the system of systems (SoS) to the management of systems, describes the systems' manageability and identify systems' management tasks clearly, proposes an approach to improve system manageability. Then give a design scheme for a generic architecture in the component management with the hierarchical management and component management. Finally the analysis methods are discussed to illustrate the quantitative analysis method is correct and effective.

Key words: software-intensive systems; system of systems; manageability; management component; quantitative analysis

0 引言

近年来,随着技术的进步和各领域对系统要求的不断发展,很多领域的应用很难用单一的系统来支撑,往往是由一些不同阶段形成、采用不同技术、为不同所有者所有、提供不同功能的独立系统,通过一定的关联手段进行协调与组合,从而达到为复杂的业务提供自动化、端到端的服务支撑^[1,2]。这些系统通常具有规模庞大、结构复杂、功能综合等众多因素,因此对这些大型软件密集型系统的管理难度亦日益增大。

1 可管理性

在进行系统分析、综合及模型化,通常采用基于确定的数学模型的方法^[3],但这只适用于确定的且内部可见的系统。然而在构建大型软件密集型系统时就不

可避免地遇到一些自治系统或不确定的系统,导致需要研究其分析、综合及模型化的新方法,一个不可管理控制的系统是没有价值的。

1.1 系统体系

建立一个复杂系统并不是完全开发一个全新的系统,一种有效的途径是能够根据新系统目标,在原有的遗存系统上继承和发展,把现有的旧系统与新开发的系统组合在一起。这样的软件密集型系统中存在着大量的、管理相对独立的自治系统,此时的目标系统就具有了系统体系(SoS)的基本特性^[4]。

来自于系统工程理论的系统体系概念在国际上得到计算机软件领域的普遍重视。很多学者从各自的角度提出“系统体系”的定义,但大都是围绕自治构件的存在和参与所产生的性质来描述系统体系的定义,比较典型的就是 Maier 等人描述的系统体系基本特征,即操作方面独立、管理方面独立、演进发展、呈现行为和地理分布等五种性质^[5,6]。

系统体系所倡导的是一种灵活多变的体系结构,其功能随着需要不断地进行改变,不断兼容新的技

收稿日期:2012-01-04;修回日期:2012-04-11

基金项目:江苏省科技支撑项目(BE2009157)

作者简介:宗平(1956-),男,博士,教授,研究方向为计算机网络、软件工程等;郭剑(1987-),男,硕士研究生,研究方向为软件工程。

术^[7]。当前系统体系发展的难点在于如何管理系统,使其长期稳定和可靠运行。原来适用于单个系统的管理维护理论与方法就不再适用于系统体系环境了。由于系统的组成部分不断演化,大量的呈现行为会出现,如果不加以管理控制,就不能准确地达到系统的目标。另外系统的整体性能(包括可管理性)评估也不能简单地按照子系统性能指标推断,而需要适用系统体系的可管理性测度方法。

1.2 组成可管理性

一般认为可管理的系统主要有以下四方面特征,反映出不同于单一系统的可管理性特征。

第一是内部可见,内部的可见又可分为状态可见和结构可见,主要是通过输出信号对系统状态进行观测。

第二是性能可控,即通过控制过程能够实现控制目的,利用输入控制信号对系统状态以及输出实现控制,而且由于许多自治系统的存在,对这些系统组成部分只能对其影响,而不能完全对其进行控制。

第三是功能可见,即全面地掌握系统组成部分的功能,有序地组织各部分的功能与资源,利用有利的系统呈现行为为具体的业务需求提供有效支撑。

第四是演化可控,对于系统组成部分不断出现新技术、新需求和新条件,可以对演进方向进行控制,保证系统的总体目标与性能不受影响。

这里通过一个“网上订餐系统”来具体说明系统的管理任务。网上订餐系统是一个包含多个自治系统的集成环境,初步具有一些系统体系的特征,有一定的典型性。

网上订餐系统的业务流程大致为:

(1) 顾客通过网络提交订餐信息,个人地址信息;

(2) 餐厅核实顾客个人信息以及订单信息;

(3) 顾客通过网上付费系统付费;

(4) 餐厅通过付费系统确认顾客付费信息,制作相应套餐;

(5) 套餐制作完成,送餐员通过地图系统所得到的线路信息将套餐送往指定地点;

(6) 短信系统在套餐制作完成后一定时间内发出短信提示顾客做好取餐准备;

(7) 顾客收到套餐,网上付费系统与餐厅的银行账户进行结算,交易终结。

整个业务流程中涉及到了餐厅订餐系统、网上付费系统、地图系统、短信系统,每一个系统都从属不同的拥有者,均属于自治系统,在整个订餐流程中提供相应的服务。其中短信系统就有明显的时延要求,不能因为各种各样的故障(例如网络的原因)而导致短信提示未能及时到达顾客的手机。地图系统所提供的路

线应提供最优的路线来减少配送时间,这就是对地图系统的计算性能的要求。网上付费系统也应运行在一个安全的环境之中,来保证顾客、餐厅的账户安全,这就是对网上付费系统的安全性的要求。上述的这些要求全部需要进行管理,保证餐厅的订餐业务达到预定目标。

2 提高系统的可管理性

影响系统管理性的因素大致来源于三方面:系统结构、系统规划设计和系统组成部分的来源。其中系统组成部分来源往往根据系统的整体目标来选定,选择的自主度较小,故此因素不能用于后期对系统可管理性的提升。通过改变系统的规划设计来提升可管理性对于集成大量的系统这样任务来说,已经是不可行的。所以,文中提出的改进系统可管理性的方法是通过系统结构的重新审视,并通过增加管理组件来调整和优化系统结构。

运营管理独立的自治系统一般都具有自己内部的管理系统或是管理组件,然而软件密集型系统需要完成一个既定的目标,则需要从系统体系层面进行管理,这样一个外部的管理系统或是管理组件就不可或缺。对系统的结构调整策略应是适应不断变化的外部或内部环境,期望延长系统的使用寿命。在对系统体系背景下系统进行规划设计时就应预留出相应的管理接口,减小系统组成部件之间的耦合度。

文中给出一种面向管理组件的分层框架,该框架的内容说明如下:

(1) 分层管理(基础设施,服务的执行,客户的资料)。

网络的层次结构在许多实际系统中有着广泛的应用,如具有社会关系结构的社会经济系统、具有纯技术结构的通信网络系统,以及具有复杂社会-技术结构的软件密集型系统^[8]。SoS 层次就是在系统体系环境下针对软件架构提出的技术方法,将大型系统从上至下地分为六层^[9]: enterprise、family of systems、systems、subsystems、components 和 parts。

在进行系统管理的时候可以根据系统体系所划分的层次进行管理层次的划分,文中将 SoS 层次的六层对应到管理上的三层,如图 1 所示。最底层将元件和组成部分与子系统划归为同一层,元件和组成部分的管理应由其自身完成,子系统层面已经有自己独立的功能和系统目标。第二层是系统层,系统层面包含了许多自治系统,拥有了管理的独立性和自主演化性,这也是传统的系统管理所关注的一个层次。第三层就是系统体系层次,其管理的是突然出现的、非线性的呈现行为,对组成部分不断的演化和整个系统外部环境进

行感知和反馈。这样精简系统体系的六层结构,可以适当忽视一些系统内部的通信,将主要关注点放在系统之间的互操作性上,并将系统体系的整体呈现行为作为系统的最终管理目标,可以增加系统的可管理性。

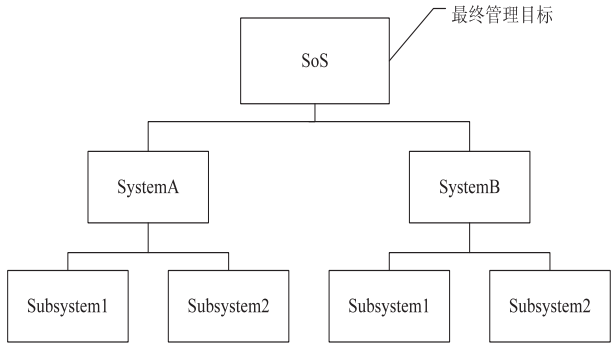


图1 SoS 管理层次

根据划分的三个管理层次,设置相应粒度的管理系统元件(管理组件)来实现管理任务和目标。子系统层次更多地关注所提供的服务、系统的运算性能等,并负责采集相应的系统数据,系统层次则关注网络性能、信息安全性等整体性目标,完成系统之间的交互操作。

(2)增加管理组件。

在不能进行集中控制的情况下,选择增加一系列相关的智能管理组件对系统进行管理是一种十分有效的途径^[10]。这些管理组件可以通过一个反馈的回路或者通过一个决策机制来进行对目标系统的控制,当然这个决策机制要建立在系统运行环境和系统信息之上,或者建立在预先研究好的系统模型之上,并且这些管理组件可以通过感知到信息进行新的参数化和升级。管理组件组成架构如图2所示。

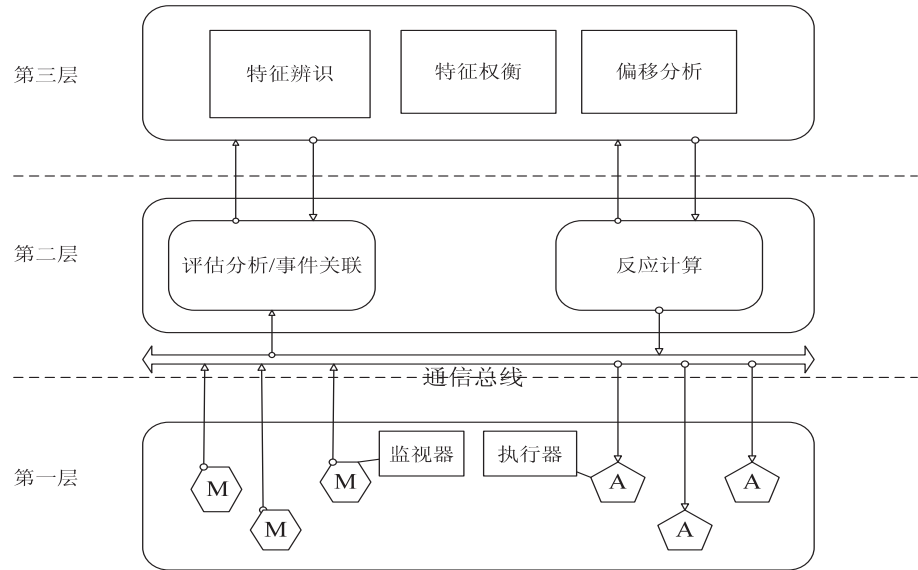


图2 管理组件总体架构

图2所描述的这种管理组件可实现系统的闭环管理,其整体架构主要分为三层。

第一层主要由一些分布在系统各处的监视器以及执行器构成,监视器对系统的各种配置以及系统特定目标(功能是否正确,性能,安全性,可靠性)的相关数据进行采集并对数据进行过滤等预处理,执行器是一些动作机制的执行端。

第二层主要包含评估分析/事件关联、动作计算这些功能模块,数据总线负责收集信息的传输。评估分析/事件关联进行全局的关联与分析。如果一个错误产生并被监视器所发现,进行一系列的分析计算之后,通过存在于监视器周围的执行器执行相应的动作。当然无论是评估分析/事件关联和动作计算都是根据第三层的规则指导进行的。

第三层主要包含特征辨识、特征权衡、偏移分析等功能模块,起主要作用的是一些分析的算法和规则。这些算法以及规则是人为制定和输入的。然而在系统体系环境中,由于自治系统的存在,这些算法及规则模块就应更加智能,能够在运行过程中感知系统环境的变化以增加适应性,感知的任务则是需要监视器模块来执行。这样的管理组件可根据系统组成部分的运行变化实现可知可控的演进。

这些管理组件并不是单一地进行部署,而是针对整个系统做框架部署,在足够大的网络位置进行安放、接收和处理相应的管理任务,并通过一些分布式的协调协议负责它们的同步。当这些组件发现服务提供单元功能出现异常,它就会协调其它组件来解决问题,寻找替代的服务提供单元并进行相应的系统环境配置。这样的管理组件就能帮助整个系统体系创建一个动态的和面向服务的系统。

3 可管理性的测度

软件系统是在现代软件工程思想及方法的指导下设计并部署起来的,单纯的定性分析存在着片面性,其也应如工程产品一样具有一些具体的量化指标^[11]。对可管理性做定量分析,需要具体的量化指标,以了解系统的状态,指导系统的建立以及后期的维护演进。

3.1 管理的目标

管理的目标分为功能性目标和非功能性目标。

●系统的功能性目标主要包括:

(1)系统目标,这是系统建立的动机;

(2)系统信息,主要为系统当前的状态信息、配置信息等;

(3)系统功能的灵活性和适应性;

(4)系统外观,系统的外观又包括界面以及功能;

(5)系统功能对使用者的呈现性,良好的呈现性能让使用者不需要了解系统的内部细节,出现问题则由开发者或者管理者来解决。

●系统的非功能性目标有:

(1)系统性能;

(2)可靠性;

(3)健壮性;

(4)可控性;

(5)可维护性;

(6)安全性;

(7)可用性。

对上述这些目标的可管理性主要体现在对这些要求的实现程度以及可管理性受这些因素的影响。管理目标可以进行控制或者影响,不能控制与影响的则可以进行观察和测度。

3.2 管理目标的加权

无论在自然界还是现实社会中,不难发现,几乎每一个系统,在其组成要素中,总有一个或几个要素占有非常重要的位置。如果改变或者去掉它们,这个系统将会在诸如结构、稳定性,甚至生存上都会有很大的问题。这样的系统组成部分在系统组成部分中的加权值就应相对比较大小。

在系统体系的环境下,对系统呈现行为的影响力也是管理目标重要性的重要衡量标准之一。呈现属性是由许多自治实体的局部活动和相邻交互的累积效果产生的^[12]。因此,一个呈现属性是一个系统中的特性,该特性是不能在单一的独立代理组件上或者在小数目组件上被局部化的。有时该属性也被称作全局属性。它们是系统体系中不可避免的。呈现属性在其效应上可以是有利的、有害的或者是中立的,呈现是系统体系成功和失败的关键机制^[13]。在产生系统呈现行为的过程中,不同的系统组成部分(应用,服务等)所起到的作用是不同的,根据作用的不同也应予以不同权值。如在“网上订餐系统”的示例中,地图系统的性能对于整个送餐效率的影响十分大,但是这还不是致命的,付费系统的安全性对于餐厅拥有者才是至关重要的。

除了不同的系统组成部分(子系统或是自治系统)在整个系统可管理性测度中应体现不同的权值以外,系统组成部分的各种目标,包括功能性目标和非功能性目标都应该赋予不同的权值。通常互联网可以认为是一个系统体系,因此文中参照网络管理的基本策

略^[14],对于普通网络节点比较看重其功能的灵活性和可用性并予以较高的权值,然而对其系统性能要求就相对较低。但是在一些处于网络关键位置的节点,例如一个网络的边界节点的可靠性以及系统性能就应该予以较高的权重,其灵活性就是相对次要的要求。这些对系统不同管理目标的权值赋予是一个演进过程,需要在实践中不断优化完善。根据上述两个角度对系统的可管理目标的加权测度,可以更有力地对软件进行控制和管理,合理地组织和分配资源,还可以有效地对遗存系统进行价值评估,考虑其能否向系统体系环境进行迁移。

4 结束语

文中在对软件密集型系统的可管理性进行分析的基础上,明确了管理任务和目标,指出系统体系背景下系统管理的技术途径,采用分层管理以及设置系统管理组件,设计了一种通用的管理组件架构,给出了改进系统可管理性的技术方法,并讨论了可管理性的定量分析途径。在后期工作中需要在实际部署的系统进行方案及效果的进一步验证。

参考文献

- [1] 卢捍华,张顺颐,王亚石,等. 系统体系方法与软件系统功能形式分解[J]. 通信学报,2010(12):131-139.
- [2] 肖红梅,卢捍华,丁傲西. 大型电信应用软件系统中系统体系的综述[J]. 电信快报,2010(9):28-29.
- [3] 王景光. 信息系统建模与结构复杂性[M]. 北京:机械工业出版社,2011.
- [4] Brooks R T, Sage A P. System of Systems Integration and Test [J]. Information Knowledge and Systems Management, 2005, 5(4):261-280.
- [5] Maier M W. Architecting Principles for System of Systems [J]. Systems Engineering, 1998, 1(4):267-284.
- [6] Sage A P, Cuppan C D. On the Systems Engineering and Management of Systems of Systems and Federations of Systems [J]. Information Knowledge and Systems Management, 2001, 2(4):325-345.
- [7] Fisher D A. An Emergent Perspective on Interoperation in Systems of Systems[R]. Pittsburghers: Carnegie Mellon University, 2006.
- [8] 张志杰. 基于分层结构的管理信息系统架构设计[J]. 计算机技术与发展, 2010, 20(10):147-149.
- [9] Morganwalp J, Sage A P. A System of Systems Focused Enterprise Architecture Framework and an Associated Architecture Development Process[J]. Information Knowledge and Systems Management, 2003(3):1-19.
- [10] Baresi L, Baumgarten M, Mulvenna M, et al. Towards Perva-

停用词。然后采用 TF-IDF 方法对文本相关概念度进行计算,并进行语义标注,再结合文中方法计算概念相似度。对测试数据集采用支持向量机 SVMs (Support Vector Machines)^[14] 方法进行分类测试,并进行分类效果的比较。文中对两种权重表示的测试文本进行分类,对于不同的惩罚系数 c 进行实验比较,实验结果如图 1 所示:

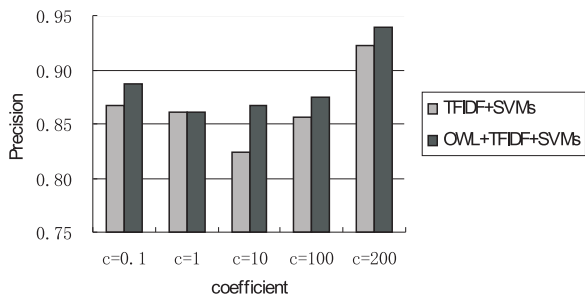


图 1 两种权重表示的分类比较

从图 1 可以得出结论,利用本体改进的概念相似度计算方法,在不同的惩罚系数下,文中方法比传统的 TF-IDF 权重在分类准确率方面有一定的提高。实验表明文中方法能充分利用 OWL 的语义特征来计算相关概念之间的相似度,因而可以为基于本体的文本分类和聚类方法提供参考。

4 结束语

文中提出了一种基于 OWL 语言描述的本体概念相似度计算方法,通过结合 OWL 属性特征与概念层次关系来计算概念相似度,并通过与传统的 TF-IDF 方法进行比较实验。

实验结果表明,文中提出的本体概念柔性相似度计算方法能够有效地体现概念的相似性排序,并可根据服务请求进行动态调整。通过对文本分类的实验对比,在结果的查准率方面有明显提高。后续研究将在现有探讨语义相似度的基础上,进一步分析本体描述语言所包含的语义特征与概念属性的权重问题等,这对于基于本体的文本分类和聚类问题研究有着积极的作用。

参考文献:

- [1] 薛中玉,李春梅,黄道雄. 基于文本挖掘的本体自动构建系统架构解析[J]. 计算机技术与发展,2011,21(1):100-103.
- [2] Paolucci M, Kawamura T. Semantic Matching of Web Services Capabilities[C]//Proceedings of the International Semantic Web Conference. Berlin/Heidelberg: Springer, 2002: 333-347.
- [3] Kuang Li, Deng Shuiguang, Li Ying, et al. Exploring Semantic Technologies in Service Matchmaking[C]//Proceedings of the Third IEEE European Conference on Web Services (ECOWS'05). [s. l.]: [s. n.], 2005: 226-234.
- [4] Lin D. An Information-theoretic Definition of Similarity[C]//Proceedings of the International Conference on Machine Learning. [s. l.]: Morgan Kaufmann, 1998: 296-304.
- [5] Gruber T R. Toward Principals for the Design of Ontologies Used for Knowledge Sharing[J]. International Journal of Human-computer Studies, 1993, 43(5/6): 907-928.
- [6] Studer R, Benjamins V R, Fensel D. Knowledge Engineering, Principles and Methods[J]. Data and Knowledge Engineering, 1998, 25(1-2): 161-197.
- [7] Maedche A, Staab S. Ontology Learning for the Semantic Web[J]. IEEE Intelligent Systems, 2001, 16(2): 72-79.
- [8] 潘谦红,王 炬. 基于属性论的文本相似度计算[J]. 计算机学报, 1999, 22(6): 651-655.
- [9] 张晓李,王西锋. 基于本体和相似图的概念语义相似度计算[J]. 计算机技术与发展, 2011, 21(8): 101-104.
- [10] Escalera A D, Moreno L E, Salichs M A. Road Traffic Sign Detection and Classification[J]. IEEE Transactions on Industrial Electronics, 1997, 44(6): 848-859.
- [11] 陈杰蒋,祖 华. 领域本体的概念相似度计算[J]. 计算机工程与应用, 2006, 42(33): 163-166.
- [12] 李 鹏,陶 兰,王弼佐. 一种改进的本体语义相似度计算及其应用[J]. 计算机工程与设计, 2007, 28(1): 227-229.
- [13] Salton G. Introduction to Modern Information Retrieval[M]. New York: McGraw Hill Book Company, 1983.
- [14] Joachims T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features[C]//Proceedings of the 10th European Conference on Machine Learning (ECML). [s. l.]: Springer, 1998: 2-7.

(上接第 102 页)

- sive Supervision for Autonomic Systems[C]//Proceedings of the IEEE Workshop on Distributed Intelligent Systems; Collective Intelligence and Its Applications (DIS'06). [s. l.]: IEEE Computer Society, 2006: 365-370.
- [11] 王明哲. 大型集成系统体系结构研究进展与挑战[J]. 系统工程理论与实践, 2008(80): 163-170.

- [12] 陈英武,姜 江. 关于体系及体系工程[J]. 国防科技, 2008, 29(5): 30-35.
- [13] 王元放,周宏仁,敬忠良. 系统的系统综述[J]. 系统仿真学报, 2007, 19(6): 1182-1185.
- [14] Keating C, Rogers R, Unal R. System of Systems Engineering[J]. Engineering Management Journal, 2003, 15(3): 36-45.

作者: 宗平, 郭剑
作者单位: 宗平(南京邮电大学 海外教育学院, 江苏 南京 210046), 郭剑(南京邮电大学 计算机学院, 江苏 南京 210046)
刊名: 计算机技术与发展
英文刊名: Computer Technology and Development
年, 卷(期): 2012(9)

参考文献(16条)

1. [Android 中文网 \(android.NET\) what is Android](#) 2007
2. [Palm Source, Inc Open Binder Version 1.0](#) 2005
3. [Chin E;Felt A P;Greenwood K Analyzing Inter-application Communication in Android](#) 2011
4. [苗忠良;宛斌 Android多媒体编程从初学到精通](#) 2011
5. [Vogel L Android Development Tutorial](#) 2011
6. [Albertchen Android IPC通讯机制源码分析](#) 2010
7. [周学海;余艳伟 Android安全技术加固\[期刊论文\]-计算机系统应用](#) 2011(10)
8. [宋杰;党李成;郭振朝 Android_OS手机平台的安全机制分析和应用研究\[期刊论文\]-计算机技术与发展](#) 2010(06)
9. [陈萍;吴波 Linux IPC机制安全性的研究与改进\[期刊论文\]-解放军理工大学学报](#) 2004(03)
10. [汪源;刘芳 Android的进程与单线程模型浅析\[期刊论文\]-仪器仪表用户](#) 2011(05)
11. [王文义;武华北 Linux中进程间信号通信机制的分析及其应用\[期刊论文\]-计算机工程与应用](#) 2005(03)
12. [Enck W;Ongtang M;McDaniel P Understanding Android Security\[外文期刊\]](#) 2009(01)
13. [Schmidt A D;Schmidt H G;Batyuk L Smartphone Malware Evolution Revisited:Android Next Target](#) 2009
14. [胡伟 Android系统架构及其驱动研究\[期刊论文\]-广州广播电视大学学报](#) 2010(04)
15. [杨丰盛 Android技术内幕](#) 2011
16. [李刚 疯狂Android讲义](#) 2011

本文链接: http://d.g.wanfangdata.com.cn/Periodical_wjtz201209027.aspx