文章编号:1671-4598(2017)08-0005-03 DOI:10.16526/j.cnki.11—4762/tp.2017.08.002 中图分类号:TP311 文献标识码:A

开放网络环境下软件安全性测试技术研究

谢巧玲
(西安文理学院 信息工程学院,西安 710068)

摘要:在开放网络环境下软件容易受到攻击,导致软件故障,需要进行安全性测试,针对无监督类测试方法开销较大和复杂度较高的问题,提出一种基于半监督自适应学习算法的软件安全性测试方法;首先采用模糊度量原理构建软件安全测试的半监督学习数学模型,分析软件产生安全性故障的数组特征,然后通过软件故障的熵特征分布方法进行软件的可靠性度量,在开放式网络环境下建立软件可靠性云决策模型,实现安全性测试和故障定位;最后通过仿真实验进行性能验证,结果表明,采用该方法进行软件安全性测试,对软件故障定位的准确度较高,测试的实时性较好,保障了软件的安全可靠运行。

关键词:开放网络环境;软件;测试;安全;半监督学习

Research on Software Security Testing Technology in Open Network Environment

Xie Qiaoling
(School of Information Engineering, Xi′an University, Xi′an 710068, China)

Abstract: The vulnerable in the open network environment software, lead to software failure, the need for safety testing, the testing method of overhead non supervisory large and complex problems, put forward a kind of software security testing methods based on semi supervised adaptive learning algorithm. First, a semi supervised learning model of fuzzy measure principle construction of software security testing, security feature array fault analysis software, then the software reliability measurement by the method of entropy feature of software fault distribution, the establishment of software reliability of cloud decision model in open network environment, security test and fault location. Finally, through simulation experiments verify the performance, results show that using the method of software security testing of software fault location accuracy, real-time test well, guarantee the safe and reliable operation of the software.

Keywords: open network environment; software; testing; security; semi supervised learning

0 引言

随着信息处理技术的不断推广和数字化技术的深化发展,大规模的软件技术应用迅速普及,为了满足网络用户对大数据传输和处理的需要,需要通过软件数据分析和信息传输调度。伴随着网络大数据信息传输的迅速增长,软件作为信息处理的重要工具,是通过一定的编程和算法设计并实现一定功能的程序代码,软件在信息处理、工业控制、单片机控制、计算机控制、网络通信以及人工智能等领域都表现出卓越的应用价值,人类越来越离不开软件进行相应的过程控制,软件的可靠性和安全性在一定程度上决定了软件的可移植性和寿命周期,研究软件的安全性测试技术,对改善软件性能,提高软件应用的普适性方面具有重要意义[1]。

软件的安全性测试主要是实现软件的故障定位识别、Bug挖掘和错误代码纠正等功能,软件测试的目的是保障软件运行的可靠性和稳定性,通过高质量且有效的测试实例进行软件分析,进而提高软件嵌入式系统和相关应用产品的可靠度,对软件安全性测试的原理是进行软件代码运行故障的特征提取和分析研究,通过良好的测试实例分析软件缺陷,得出缺陷报告,为软件设计者提供设计参考。当前,对软件的安全性测试方法主要采用的是监督学习下的无监督类学习方法[2],通过对软件故障特征的信息素定位分析,得到软件分布的虚拟信息资源,结合学习算法进行自适应故障定位和资源信息匹配,根据这一测试原理,取得了一定的研究成果,其中,文献[3]中提出一种面向安全性分析的嵌入式软件测试方法,基于聚类技术和专家检验技术进行软件的故障点聚类分析,采用没有监督的训练算法进行故障属性归类,实现软件安全性测试,具有一定的安全检测性能,但该方法在故障聚类中受到不确定扰动和不规则代码因素等影响,对软件代码之间的耦合差异性识别精度不高;文献[4]采用基于最小点覆盖的控制平面跨层生存性设计进行软件安全性定义与测试,建立二维空间的四叉树模型,实现对软件的可靠性、可用性评估,但该方法存在的问题是故障漏检率较高;文献[5]中提出一种基于最坏分离的联合分辨率判别分析的软件可靠性度量与安全测试技术,结合故障树分析方法进行安全性联合分辨,该方法采用的是无监督类学习方法,在开放式网络环境下存在开销较大和复杂度较高等问题。

为了解决传统方法在软件安全性测试方面存在的问题,本文提出一种基于半监督自适应学习算法的软件安全性测试方法。首先采用模糊度量原理构建软件安全测试的半监督学习数学模型,分析软件产生安全性故障的数组特征,然后通过软件故障的熵特征分布方法进行软件的可靠性度量,在开放式网络环境下建立软件可靠性云决策模型,实现安全性测试和故障定位。最后通过仿真测试进行性能验证,得出有效性结论。

收稿日期:2016-12-16; **修回日期**:2017-03-15。
基金项目:西安市科技计划项目(CXY1531WL39)。
作者简介:谢巧玲(1979-),女,陕西安康人,硕士,讲师,主要从事软件开发、软件测试方向的研究。

1 软件安全测试原理与指标体系分析

在开放式网络环境下软件质量受到软件的代码可扩展性、防御攻击性以及软件的可移植性等方面因素的影响，软件安全测评技术就是通过对软件的可靠性评估，挖掘软件存储的BUG和代码价值属性，通过产品的内部属性分析，构建软件产品的质量监督机制，实现对软件质量的内部和外部属性的监督和测量。当前主流的软件测试模型有McCall模型、Boehm模型和ISO/IEC 9126模型，这三种模型通过对软件可靠性、程序语言复杂度的定量分析和评估，为软件的应用部门、移植部门和维护部门提供有用的数据参考，降低在开放式网络环境下软件容易受到攻击的缺陷和交易风险，通过软件的安全性测试，完善软件的质量评价机制。软件的安全性测试分为三个层面，分别为代码调试层、虚拟信息资源管理层和物理资源管理层[6]，对软件运行代码进行安全建设管理，结合专家数据库信息系统进行资源信息部署和虚拟应用资源的分类识别，构建服务器资源，在应用软件层对软件产品进行定向评估和质量评价，分析软件产生安全性故障的数组特征，然后通过软件故障的熵特征分布方法进行软件的可靠性度量，开放式网络环境下软件安全性测试的三层结构模型如图1所示。

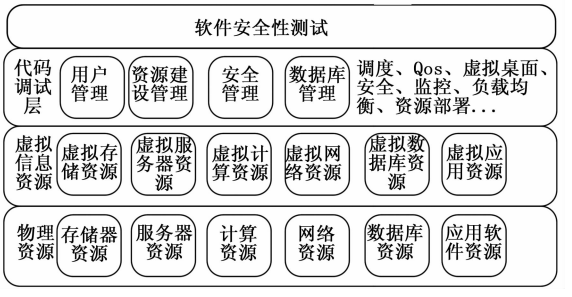

图1 开放式网络环境下软件安全性测试的三层结构模型

分析图1得知，软件的安全性测试与软件开发过程紧密相关，通过ISO/IEC 9126模型分析[7]，软件安全性测试由内容组成包括了有用性、可靠性、可维护性、可移植性，通过多个层次的量化分析，在软件测试中，需要对软件运行的争取性、可靠性、效率、完整性和可用性进行定量评估，采用三层测试结果，得到软件安全性测试的指标体系描述如图2所示。通过对软件安全性测试的指标体系建模，通过对软件的底层设计和顶层质量评估，在Boehm模型中进行量化分析，以满足客户提出的标准和要求。

2 软件安全性测试数学模型

在上述进行了软件安全性测试的指标体系分析的基础上，进行安全性测试数学建模，建立软件安全测试的半监督学习数学模型，采用模糊度量原理构建软件安全测试的半监督学习数学模型[8]，首先给出软件质量评价的量度模型体系如图3所示。

在开放式网络环境下，采用一个四叉树模型表示软件分布特征的数据结构，用三元组 $SC=(Ex, En, He)$ 用来描述软件在数学空间内的属性数值特征，假设集合 O 定义在 n 维论域 μ 上的软件可移植性指标参量，则 O 的四叉树定义为：$\mu=[d_{1\mu}:d'_{1\mu}]\times[d_{2\mu}:d'_{2\mu}]\times\cdots[d_{n\mu}:d'_{n\mu}]$。建立一个表示软件BUG的概念的可度量粒度，通过对软件质量的一致性评估，通过可靠性度量模型将 O 和 μ 存储下来。在软件测试的三层网

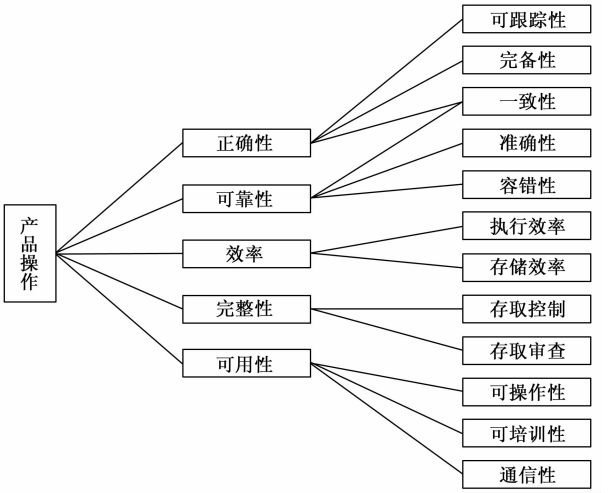

图2 软件安全性测试的指标体系

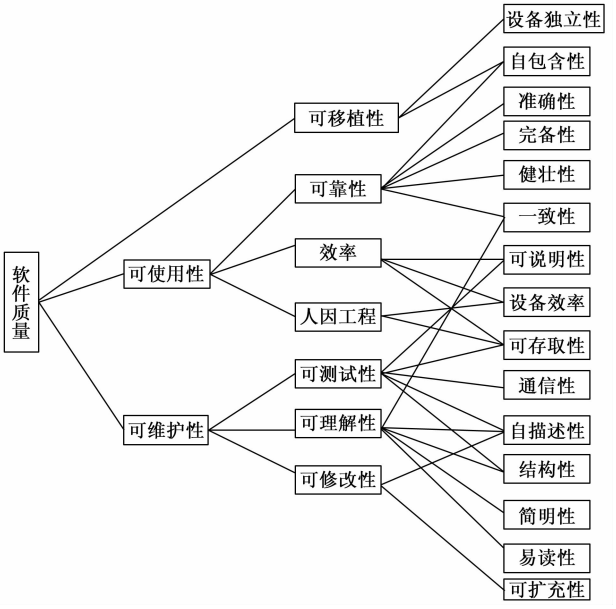

图3 软件质量评价的量度模型体系

络模型的每一级，确定软件可靠性的度量质量集合被分解为 2^n 个子集，在开放网络环境下，由于软件受到攻击等干扰，这里并通过区间划分来反映度量等级[9]，假设 $n=2$，输入：软件安全性度量的 $n-1$ 个度量区间。

输出：$SC_i(Ex_i, En_i, He_i), i=1,2,3,\cdots,n$。

1) $Ex_i=(C^i_{\max}+C^i_{\min})/2$;(当 $i=1$ 时，令 $En_1=0$;)

2) $En_i=(Ex_i-Ex_{i-1})/3$;(当 $i=1$ 时，令 $En_1=(Ex_2-Ex_1)/3$;)

$He_i=\alpha$。(其中 α 为一个常数)

假设软件的可靠性分为若干个等级，分布表示为4个子集 $\mu_{d_{1L}d_{2R}}$，$\mu_{d_{1R}d_{2R}}$，$\mu_{d_{1L}d_{2L}}$，$\mu_{d_{1R}d_{2L}}$。根据软件的功能性、可靠性、易用性等特征，假设软件的初始度量等级 $d_{1mid}=(d_{1\mu}+d'_{1\mu})/2$，$d_{2mid}=(d_{2\mu}+d'_{2\mu})/2$，定义软件的二级度量指标描述如下[10-11]：

$$O_{d_{1R}d_{2R}}=\{o\in O: o_{d_1}>d_{1mid}, o_{d_2}>d_{2mid}\} \quad (1)$$

$$O_{d_{1L}d_{2R}}=\{o\in O: o_{d_1}>d_{1mid}, o_{d_2}>d_{2mid}\} \quad (2)$$

$$O_{d_{1L}d_{2L}} = \{o \in O: o_{d_1} \leqslant d_{1mid}, o_{d_2} \leqslant d_{2mid}\} \quad (3)$$

$$O_{d_{1R}d_{2L}} = \{o \in O: o_{d_1} > d_{1mid}, o_{d_2} \geqslant d_{2mid}\} \quad (4)$$

以上各式分别表示的是软件的故障密度、测试覆盖率、平均失效间隔时间、平均恢复时间等 4 个二级度量质量，同理，对于 $n=3$ 时，软件的安全性度量指标可以分解为 8 个子集：$\mu_{d_{1L}d_{2R}d_{3L}}$，$\mu_{d_{1L}d_{2R}d_{3R}}$，$\mu_{d_{1R}d_{2R}d_{3L}}$，$\mu_{d_{1L}d_{2R}d_{3R}}$，$\mu_{d_{1L}d_{2L}d_{3L}}$，$\mu_{d_{1L}d_{2L}d_{3R}}$，$\mu_{d_{1R}d_{2L}d_{3L}}$，$\mu_{d_{1L}d_{2L}d_{3R}}$。

对于 n 维数据，设权重集为 V，根据软件可靠性的依从性知 $V=\{w_1,w_2,w_3,w_4\}$，且 $w_1+w_2+w_3+w_4=1$，由此得到软件修复的有效性分解的集可以表示为：$\mu_{d_{1\alpha}d_{2\alpha}\cdots d_{n\alpha}}$，$\alpha \in \{L,R\}$。

通过上述设计，建立软件安全性测试数学模型，结合软件产生安全性故障的数组特征，通过故障定位实现软件安全性测试研究。

3　软件安全性测试的半监督学习故障定位实现

结合上述设计的软件安全性测试的数学模型，基于半监督自适应学习算法进行软件安全性测试，通过软件故障的熵特征分布方法进行软件的可靠性度量[12-13]，根据环形复杂度度量方法，得到半监督学习下软件的测试复杂度为 $(b+1)w$，其中 b 为二叉树的深度，v 为隶属度，c 为常数。通过模糊隶属度分析，得到软件故障分布的熵特征形式化表示为：$U \to [0,1]$，$\forall x \in U, x \to SC(x)$，故障修复时间为 $O(m^2)$。不确定性度量的复杂度为 $O((b+1)w+m^2)$。通过半监督自适应学习方法评估软件测试的故障修复率 FPR，失效密度 FNR，误检率 $Error$：

$$FPR = \frac{B}{A+B} \quad (5)$$

$$FNR = \frac{C}{D+C} \quad (6)$$

$$Error = \frac{B+C}{A+B+C+D} \quad (7)$$

以上这些参数满足软件的质量许可要求时，软件的质量效果较好，综上分析，通过软件故障的熵特征分布估计方法，得到软件故障定位的计算式描述为[14]：

$$POF = \frac{\sum_{i=1}^{TC} M_o(C_i)}{\sum_{i=1}^{TC}[M_n(C_i) \times DC(C_i)]} \quad (8)$$

$$M_d(C_i) = M_n(C_i) + M_o(C_i) \quad (9)$$

式中，$M_o(C_i)$ 为易恢复性指标提下 $C_i(i=1,2,\ldots,n)$ 中的修复有效性，$DC(C_i)$ 为 $C_i(i=1,2,\ldots,n)$ 的所有子类数，$M_n(C_i)$ 为 $C_i(i=1,2,\ldots,n)$ 的平均宕机时间。

综上分析，得到本文设计的软件测试算法描述为：

输入：$SC_i(Ex_i,En_i,He_i),(i=1,2,\cdots,p)$ 及对应权重值：$w_1,w_2,\cdots,w_p$。

输出：软件质量评价的安全度量云 $SC(Ex, En, He)$，并执行[15]：

1) $Ex = \sum_{i=1}^{P} WiEx_i$；

2) $En = \sqrt{\sum_{i=1}^{P} Wi(En_i)^2}$；

3) $He = \sum_{i=1}^{P} WiHe_i$；

4) 重复(1)到(3)，算出所有的度量指标 S_i；

5) 当满足阈值 $MAX(S_i)$ 时，$G=i$。

本文设计的基于半监督自适应学习算法的软件安全性测试流程如图 4 所示。

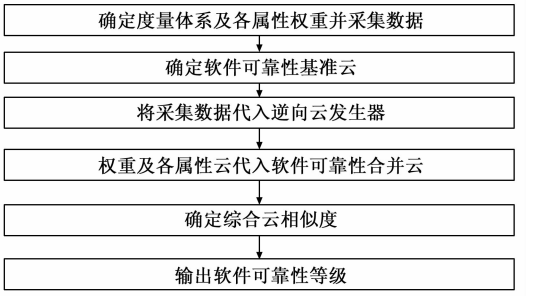

图 4　软件安全性测试流程

4　测试实验分析

为了验证本文方法在实现开放式网络环境下软件安全性测试中的应用性能，进行仿真实验分析，实验使用 Matlab7 作为仿真实验平台，网络环境建立在开放式的 Hadoop 0.20.2 平台基础上，采用 Eclipse 集成开发环境进行软件平台调试，软件的质量评价体系分为不安全、低安全、中安全、高安全和绝对安全等 5 个质量等级[16-20]，根据前期的数据采样分析和实践调试，得到不同安全等级描述的软件安全性度量数字特征见表 1。

表 1　软件安全性度量的数字特征

软件安全度量等级	Ex	En	He
Ⅰ(不安全)	0	0.022	0.05
Ⅱ(低安全)	0.31	0.043	0.05
Ⅲ(中安全)	0.45	0.062	0.05
Ⅳ(高安全)	0.68	0.096	0.05
Ⅴ(绝对安全)	1	0.1	0.05

根据表 1 给出的实验指标体系，采用本文方法进行某大型测控软甲的安全性测试，得到各级指标体系下的测试数据结果见表 2。

表 2　某大型测控软件的安全性测试数据

一级度量	权系数	二级度量	结果
成熟性	0.33	失效解决	0.675
		失效间隔	0.156
		避免死机	0.56t
		故障密度	0.674
		恢复时间	0.843
		宕机时间	0.865
		抵御误操作	0.945
		依从性	0.845
容错性	0.25	故障排除	0.986
		代码结构	0.897
		扩展度	0.965
		自包含性	0.968
		健壮性	0.987
		抗攻击性	0.947
移植性	0.84	结构性	0.976

(下转第 19 页)

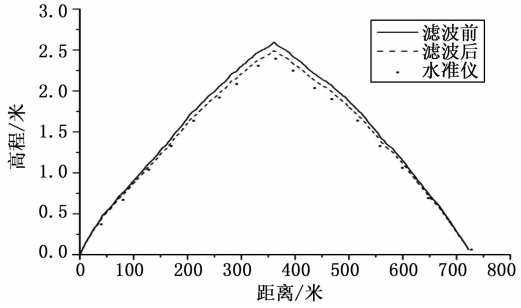

图 13　某跨江大桥斜拉桥线型

实际测量中不同工况下的频域特点。为使用其他辅助手段提高测量精度提供了理论基础。使用切比雪夫低通滤波器，去除掉噪声信号，可以有效的提高测量精度。在目前靠改进硬件设计来减小测量运载体的振动噪声误差在客观上困难越来越多，成本越来越高的情况下，使用数字信号处理的手段分析和解决光纤陀螺线形测量系统在实际检测中遇到的振动所带来的误差，效率高且易于实现。

参考文献：

[1] JTG/T J21－2011，公路桥梁承载能力检测评定规程［M］．北京：人民交通出版社，2011.

[2] 杨小森，闫维明，陈彦江，等．基于倾角仪的桥梁挠度测试方法研究［J］．土木工程学报，2010，43（S）：105－111.

[3] 李　盛，胡文彬，杨　燕，等．基于光纤陀螺的大跨桥梁连续线形检测技术研究［J］．桥梁建设，2014（5）：69－74.

[4] 吴光强．汽车理论．第 2 版［M］．北京：人民交通出版社，2014.

[5] Alhasan A，White D J，De Brabanterb K. Continuous wavelet analysis of pavement profiles［J］. Automation in Construction，2016，63，134－143.

[6] 程建华，李明月，时俊宇，等．船用光纤陀螺小波实时滤波算法的设计与实现［J］．传感器与微系统，2011（7）：104－107，110.

[7] 魏明果．实用小波分析［M］．北京：北京理工大学出版社，2005.

[8] Raul Ruiz de la Hermosa Gonzalez－Carrato. Pattern recognition by wavelet transforms using macro fiber composite transducers［J］. Mechanical Systems and Signal Processing，2014，48.

（上接第 7 页）

分析上述测试结果得知，采用本文方法进行软件安全性测试，软件的各项测试指标，满足设计要求，能有效保障软件的质量，图 5 给出了采用不同方法进行软件故障定位的误差对比，分析得知，采用本文方法进行软件安全性测试误差率较低，性能更优。

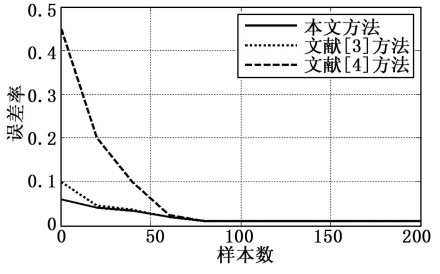

图 5　软件测试的误差对比

5　结束语

本文研究了在开放性网络环境中软件的安全性测试问题，提出一种基于半监督自适应学习算法的软件安全性测试方法。构建软件安全性测试的三层结构模型，分析评价软件安全质量的指标体系，构建软件质量故障分布的数组特征，通过软件故障的熵特征分布进行软件的可靠性度量，实现安全性测试和故障定位。研究得知，本文方法进行软件测试准确性好，误差较低，展示了较高的应用价值。

参考文献：

[1] 汤永新，刘增良．软件可信性度量模型研究进展［J］．计算机工程与应用，2010，46（27）：12－16.

[2] Zhou L M，Cai G Q，Yang J W，et al. Monte－Carlo simulation based on FTA in reliability analysis of Door System［A］. Proc of International Conference on Computer and Automation Engineering［C］. Piscataway：IEEE Press，2010，1（3）：713－717.

[3] 林永峰，陈　亮．面向安全性分析的嵌入式软件测试方法研究［J］．现代电子技术，2016，39（13）：80－83.

[4] 熊　余，董先存，李圆圆，等．软件定义光网络中基于最小点覆盖的控制平面跨层生存性设计［J］．电子与信息学报，2016，38（5）：1211－1218.

[5] 杨磊磊，陈松灿．最坏分离的联合分辨率判别分析［J］．软件学报，2015，26（6）：1386－1394.

[6] 郑长友，刘晓明，黄　松．基于蚁群算法的软件可靠性模型参数估计方法［J］．计算机应用，2012，32（4）：1147－1151.

[7] 刘玲艳，吴晓平，叶　清．云模型和混合 Petri 网相结合的系统可靠性评价［J］．火力与指挥控制，2011，36（8）：19－22.

[8] Suzukit，Kudo H. Two－dimensional non－separable block－lifting structure and its application to M－channel perfect reconstruction filter banks for lossy－to－lossless image coding［J］. IEEE Transactions on Image Processing，2015，24（12）：4943－4951.

[9] 谢洪安，李栋，苏旸，等．基于聚类分析的可信网络管理模型［J］．计算机应用，2016，36（9）：2447－2451.

[10] 杜小阳，龚川森，刘建辉，等．航空机载软件安全性测试技术研究［J］．科技创新与应用，2016，10（4）：5－15.

[11] 陈文康，赵光俊，王汝英．基于 B/S 结构的电力物联网应用软件开发［J］．电子设计工程，2016，24（22）：33－35.

[12] 林永峰，陈　亮．面向安全性分析的嵌入式软件测试方法研究［J］．现代电子技术，2016，39（13）：80－83.

[13] 姬忠孝，江国华．一种基于 FTA 和 FDG 的安全关键函数定位方法［J］．计算机与现代化，2016，10（4）：85－89.

[14] 张　放．计算机软件中安全漏洞检测技术初探［J］．山西青年，2016，20（6）：26－30.

[15] 王勇利．安全漏洞检测技术在计算机软件中应用研究［J］．数字技术与应用，2016，9（11）：210－215.

[16] 吴子杰．基于 Android 的可配置工业远程监控软件设计与实现［D］．南京：南京邮电大学，2016.

[17] 苏　欣．计算机软件中安全漏洞检测技术及其应用［J］．智能城市，2016，9（4）：110－111.

[18] 刘　丹．电子计算机联锁系统通信协议设计及安全性分析［J］．电子技术与软件工程，2016，6（15）：130－136.

[19] 甄　鹏．嵌入式软件开发模式与软件架构研究［J］．企业技术开发，2016，35（6）：64－64.

[20] 段海军，赵根学，陈　福，等．航空电子设备自动测试系统的软件架构设计［J］．计算机测量与控制，2016，24（9）：167－169.