

# 基于 SOA 架构软件服务可靠性评价方法研究

张 亚

(江苏自动化研究所,江苏 连云港 222061)

**摘 要:**针对面向服务架构软件可靠性领域存在的问题,从服务功能性属性和非功能性属性可靠性度量的角度出发,给出了服务的可靠性定义描述,描述了服务在其注册、查找、绑定组合等各个阶段的失效过程,分析了服务各阶段存在的可靠性影响因素,根据服务在不同阶段的失效过程构建了各个阶段的可靠性模型,并集成各阶段模型和服务功能性属性模型,给出单个服务的可靠性模型。通过对组合服务的流程编排进行结构分类,提出了基于流程的组合服务可靠性计算模型。

**关键词:**SOA 架构;服务可靠性;流程组合;可靠性模型;可靠性预测

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2015)04-0067-04

doi:10.3969/j.issn.1673-629X.2015.04.016

## Research on Service-reliability Evaluation Method Based on SOA

ZHANG Ya

(CSIC Software Test Center, Lianyungang 222061, China)

**Abstract:** In order to solve the problems of service-reliability in Service Oriented Architecture (SOA), from the viewpoint of reliability measurement of the service functional properties and non-functional properties, give the definition of service reliability, present the failure processes of service publication, service discovery, service composition, service binding and service execution in SOA, and analyze the influence factors of reliability in each stage, reliability models in different stage are built according to different failure processes. Then, the staged reliability models and service function properties model are integrated to form the reliability model of single service. At last, according to the characteristic of business processes, propose the reliability computing model of composition service based on process.

**Key words:** SOA; service-reliability; process composition; reliability model; reliability prediction

## 0 引 言

面向服务的体系架构(Service Oriented Architecture, SOA)一般被看作一个组件模型,SOA 将系统中的不同组件通过确定的接口/协议进行组装集成,通过业务需求驱动的流程编排进行服务的动态组合,通过服务总线进行服务的管理和远程调用,进而实现特定的任务,SOA 在更高的层次上实现了软件复用和封装。

SOA 是新一代的软件架构思想,同面向对象、模型驱动设计方式的出现一样,给软件测试和可靠性评价带来了新挑战。随着面向服务架构新技术的推广和应用,SOA 架构软件的质量备受关注,可靠性作为服务质量的一个重要属性,服务的可靠性等性能问题逐渐成为研究热点。如何根据需求界定服务的可靠性、如何对服务的可靠性数据进行量化处理、如何对服务

组合进行可靠性评价等挑战性问题仍然急需解决。文献[1-2]研究了组合服务的关键技术和可靠性预测问题;文献[3-4]研究了服务的可靠性计算模型和改善应用;文献[5]研究了操作剖面对服务可靠性的影响。目前 SOA 架构软件可靠性的研究正在如火如荼地进行。文中将针对简单服务和组合服务的可靠性问题,重点围绕服务可靠性定义及度量方式、服务可靠性度量方法和可靠性模型深入探讨,结合 SOA 架构软件服务组合中的典型流程结构,提出 SOA 架构软件服务可靠性的评价模型。

## 1 面向服务架构软件可靠性定义

根据结构化信息标准促进组织(OASIS)提出的 WS-Reliability 标准<sup>[6]</sup>,WS-Reliability(Web 服务的可靠性)可以定义为通过应用层上定义的可靠性消息

传递的 SOAP 协议,实现在存在脆弱性的不可靠网络环境中进行可靠的消息传递。

面向服务架构软件是一种特殊的系统,传统的软件可靠性评价方法和模型在一定程度上能够应用于这种新型架构。其中的原子服务是对软件模块的一种封装,依然能够采用传统的软件可靠性模型。然而,面向服务架构软件的服务通常分散分布,彼此之间采用一定的消息传递机制进行通信,具有高复用性。服务的动态组合通过将简单服务按照业务流程进行编排,组合成新服务。单个原子服务可能并不是同一个开发者开发的,而是由不同的服务提供者提供的。服务信息只能通过服务的 WSDL 文件来获取,而 WSDL 文件对服务的描述信息相对简单,这些特点最终导致 SOA 不适用传统软件可靠性模型。

Web 服务的可靠性应该既包括软件可用性的非功能性度量,也包括服务的功能性信息是否满足用户需求的度量。服务的可靠性信息还包括服务的容错性、时效性、互操作性等,然而,用户可能不关心或无法获取这些可靠性信息。文中借鉴已有的服务可靠性度量方法,结合服务可靠性的定义,从测试者的角度出发,将服务可靠性定位为两方面的属性:功能性属性和非功能性属性。功能性属性通过软件的正确性来度量,而非功能性属性通过软件的可用性来度量。服务可靠性可以进一步定位为:服务请求者根据需求在具有相同功能属性(Functional Property, FP)却具有不同非功能属性(Non-Functional Property, NFP)的候选服务中选取并正确执行所需服务的概率。

## 2 面向服务架构软件功能性属性可靠性模型分析

随机过程类的软件可靠性模型一般分为 Markov 过程模型和非齐次 Poisson 过程模型两大类<sup>[7]</sup>。Markov 过程模型的适用条件为:在软件未进行改动的区间内,假定错误出现率为一个常数,且该数随着错误数的减少而下降<sup>[8]</sup>。另外,非齐次 Poisson 过程模型在一定条件下可以近似为:将排错过程中的累计错误数看作时间的函数。

面向服务架构软件的功能性属性的提高是一个不断排错的可靠性增长过程,现有的软件可靠性增长模型多数基于排除缺陷的同时不会引入新缺陷的假设。但由于软件复杂性与人为因素的干扰,这种假设条件通常不成立。软件可靠性增长模型中较重要的为非齐次泊松过程类(NHPP)模型,其中应用较广泛的是 G-O 模型。但是,G-O 模型也存在类似的问题,其假设错误的发现和排除过程也过于理想化,而且未考虑测试过程中各种复杂因素对可靠性的影响,不符合实际

应用,在简化可靠性模型的同时,也导致模型的应用性较差,评估结果的可信度不足。因此,在面向服务的功能性属性验证过程中,必须在可靠性建模过程中充分考虑假设条件的合理性。实际执行测试时,测试人员所发现的故障数会随着时间的推移,呈逐渐下降的趋势,且越到后期越难发现故障。另外,开发人员在解决故障的过程中,引入新的故障是必然的结果,所以软件的故障总数也会不断增加。

在面向服务架构软件服务功能属性可靠性模型中,将期望故障数、故障检测率、软件预期故障数相关联,将测试覆盖率与不完美排错这两种重要的影响因素相结合<sup>[9-12]</sup>。假设面向服务架构软件的实际运行剖面 and 可靠性测试执行剖面相同,软件的故障检测过程服从 NHPP 过程,且软件中的初始错误个数  $N_0$  为固定不变的常数,一次排除一个错误,某些错误在其他错误被排除之前,不可能被查出,随着测试工作的深入,软件错误被逐渐发现和排除,剩余错误的数量和类别随之变化,因此可以将错误排除率看作测试时间的函数。排错过程中会引入新的错误,错误引入率随测试的进行发生变化,也可以看作测试时间的函数。通过以上分析,建立基于非理想排错过程的可靠性模型改进假设条件为:

(1) 软件错误检出率与剩余错误数成正比,表示为随时间变化的函数  $D(t)$ ,  $0 < D(t) \leq 1$ ;

(2) 错误引入率与已经检出的错误数成正比,表示为随时间变化的函数  $I(t)$ ,  $0 < I(t) \leq 1$ ;

(3) 每次软件的失效是由一个软件错误引起,当失效发生后,错误排除需要时间,错误排除率表示为随时间变化的函数  $\varphi(t)$ ,  $0 < \varphi(t) \leq 1$ ;

(4) 累积错误数的期望函数  $[N(t), t \geq 0]$  是一个独立的单调递增函数,且  $N(0) = 0$ ,服从 Poisson 分布,且  $\lim_{t \rightarrow \infty} N(t) = A$ ,其中  $A$  为  $N_0$  和新引入的错误数之和。

假设 Poisson 分布的期望函数为  $w(t)$ ,依据条件(1),可得软件可靠性建模方程。

$$\frac{dw(t)}{dt} = D(t)(Z(t) - \tau(t)) \quad (1)$$

其中,  $\tau(t)$  为软件运行至  $t$  时刻已解决的错误数;故障检出率  $D(t)$  表示为错误排除率  $\varphi(t)$ 、错误引入率  $I(t)$  的差和测试覆盖率  $C(t)$  的乘积,即  $D(t) = C(t) * (\varphi(t) - I(t))$ 。

依据条件(2),软件运行至  $t$  时刻的期望错误数  $Z(t)$  可表示为故障引入率  $I(t)$ 、故障排除数  $\tau(t)$  的乘积与初始故障数  $N_0$  之和。

$$Z(t) = N_0 + I(t)\tau(t) \quad (2)$$

其中,  $\tau(0) = 0, Z(0) = N_0$ 。

依据条件(3),假设单位时间内故障排除数与单位时间内期望发现的故障数成正比,比率为故障排除率  $\varphi(t)$ ,可得

$$\frac{d\tau(t)}{dt} = \varphi(t) \frac{dw(t)}{dt} \quad (3)$$

依据条件(4),可以得累积故障数为  $n$  时的分布概率。

$$P\{N(t) = n\} = \frac{(w(t))^n}{n!} \exp(-w(t)), n = 0, 1, 2, \dots \quad (4)$$

用  $w(t)$  表示在  $t$  时刻发现的累积故障数,初始故障数为  $N_0$ ,  $\tau(t)$  为  $t$  时刻累积排除的故障数,求解方程组可测期望故障数  $w(t)$ 。模型的正确性验证可以采用最小二乘估计和最大似然估计的方法,通过 MSE 来度量模型的拟合效果, MSE 越小,拟合效果越好。

### 3 面向服务架构软件非功能性属性可靠性模型分析

服务的过程流程可以分为服务注册、服务查找、服务绑定组合、服务执行 4 个步骤<sup>[13]</sup>,服务的可用与否与服务的整个生命周期都紧密相关。然而,服务过程的任何一步出现错误,都会引起服务的失效,进而导致服务的不可用。

#### (1) 服务注册的可靠性模型。

服务可以看作是一个功能的封装,一般较常用的是服务描述语言 WSDL 来定义服务接口和接口绑定信息,服务的接口信息包括服务对应的操作及其输入参数、输出参数和容错处理信息。服务接口绑定信息描述定义服务使用者与提供者进行信息交互的方式,以及服务的请求格式和响应格式,另外可以定义一组前提条件、后置条件或服务质量的描述。服务发布时可能出现服务描述格式的错误或描述的内容与功能不一致,这些错误将导致后续整个流程中服务匹配错误。分析可知,服务描述错误出现的概率大小与服务提供者的业务水平有着直接关系。假设在给定的时间区间  $[0, t]$  内,服务提供者共注册  $N$  个服务,其中匹配错误的服务个数为  $n$ ,则该服务发布的可靠性  $R_{\text{Pub}}$  可以表示为

$$R_{\text{Pub}} = 1 - \frac{n}{N} \quad (5)$$

#### (2) 服务发现阶段的可靠性模型。

当服务请求者需要调用某个服务时,首先提交服务需求请求, UDDI 根据请求者提供的服务需求在已注册服务列表中查找满足需求的服务,此过程中可能由于请求服务的描述错误或服务的请求超时等原因,导致找到一个错误的服务,或者请求失败未找到服务,

服务描述错误引起的失效与服务发布阶段错误相同,属于同一类错误,在此不重复考虑<sup>[14-15]</sup>。UDDI 对于服务请求按照一定的处理等待机制进行排序处理,一般采用优先级高者先服务或先到先服务,当发起的服务请求较多时,如果 UDDI 正在处理某个请求,处于忙碌状态,那么,后续的请求将会形成一个等待队列。此时,如果请求服务等待的时间超过设定的阈值,则发生超时失效导致查找失败,引起服务发现错误。分析可知,单位时间内服务发现成功的服务数符合 Poisson 分布的特点,则参数为  $\lambda$  的 Poisson 分布记为  $P(k; \lambda)$ 。

$$R_{\text{Dis}} = P(\xi = k) = \frac{\lambda^k}{k!} e^{-\lambda}, k = 0, 1, 2, \dots \quad (6)$$

#### (3) 服务绑定与调用阶段的可靠性模型。

UDDI 在查找完服务请求描述后,向服务请求者返回检索到服务的描述信息,将抽象的服务需求绑定到某个具体服务。在绑定过程中,可能出现由于接口地址或安全协议等问题引起的绑定失效。假设服务绑定失效的概率为  $P_{\text{SX}}$ ; UDDI 查找返回的满足查询要求的服务列表个数为  $N$ , 其中第  $i$  个服务发生绑定失效的概率为  $P_{\text{Bind}, i}$ , 则服务绑定的可靠性可以表示为

$$R_{\text{Bind}} = \prod_{i=1}^N (1 - P_{\text{bind}, i}) * (1 - P_{\text{SX}}) = (1 - P_{\text{SX}}) * \prod_{i=1}^N (1 - P_{\text{bind}, i}) \quad (7)$$

#### (4) 服务执行阶段的可靠性模型。

服务的执行过程中调用的各个服务通过网络进行服务的调度和通信,在服务调度的过程中,需要考虑网络脆弱性对服务执行可靠性的影响,网络拥塞或者不同带宽引起的服务信息通信失败或者通信错误等情况都会导致整个服务的可靠性较低,因此在服务执行阶段,网络可靠性是服务可靠性的一个重要影响因素<sup>[16]</sup>。假设网络连接失效服从 Poisson 过程,使用  $i, j$  表示网络中相互连接的两个节点,使用  $L(i, j)$  表示网络的连接,  $D(i, j)$  表示节点间的传输信息量,  $S(i, j)$  表示带宽,则两网络节点间传输信息耗时  $T(i, j) = D(i, j) / S(i, j)$ 。由此,网络连接  $L(i, j)$  的可靠度  $R_{\text{Link}}$  为

$$R_{\text{Link}} = \exp(-\lambda_{i, j} T(i, j)) \quad (8)$$

通过上面的分析,根据服务属性对服务的可靠性进行建模,将可用性可靠度分为请求阶段和执行阶段的可靠性维度,请求阶段可进一步分为服务发布阶段、服务发现阶段和服务绑定阶段,对四个阶段分别进行评估,最后综合出一个值,提供给用户,进行服务选择的参考。服务的可用性可靠度模型如下所示。

$$R_{\text{U}} = R_{\text{Pub}} * R_{\text{Dis}} * R_{\text{Bind}} * R_{\text{Link}}$$

其中,  $R_{\text{Pub}}$  为服务发布阶段可靠性;  $R_{\text{Dis}}$  为服务发



现阶段可靠性;  $R_{\text{Bind}}$  为服务绑定阶段可靠性;  $R_{\text{Link}}$  为服务执行阶段可靠性。

#### 4 面向服务架构软件可靠性模型

由上面的分析可知,服务的可靠性可以分为功能性属性可靠性和非功能性属性可靠性,文中将服务可靠性定义为正确性、容错性、测试性、可达性以及互用性的综合表现,那么服务可靠性可以定义为一个服务被正确执行的概率,即服务的准确性和服务的可用性的集成。

$$R_S = R_A * R_U$$

其中,  $R_S$  为服务的整体可靠度;  $R_A$  为服务的准确性可靠度;  $R_U$  为服务的可用性可靠度。

根据第三节软件非功能性属性的描述,给出服务可用性可靠度模型。

$$R_U = (1 - \frac{n}{N}) * \frac{\lambda^k}{k!} e^{-\lambda} * (1 - R_{\text{Sec}}) \prod_{i=1}^N (1 - R_{i,\text{Bind}}) * \exp(-\lambda_{i,j} T(i,j)) \quad (9)$$

整理可得

$$R_U = \prod_{i=1}^N (1 - R_{i,\text{Bind}}) * \exp(-\lambda_{i,j} T(i,j)) * \left( \frac{\lambda^k}{k!} e^{-\lambda} - \frac{n\lambda^k}{Nk!} e^{-\lambda} - \frac{R_{\text{Sec}}\lambda^k}{k!} e^{-\lambda} + \frac{nR_{\text{Sec}}\lambda^k}{Nk!} e^{-\lambda} \right) \quad (10)$$

其中,  $N$  为服务总数;  $n$  表示服务描述不全或不匹配的服务个数;  $T(i,j)$  为节点  $i$  和  $j$  之间传输信息所用的时间,  $T(i,j) = D(i,j) / S(i,j)$ ,  $D(i,j)$  为两节点之间传输的信息量,  $S(i,j)$  为传输信息所用的时间;  $R_{i,\text{Bind}}$  为第  $i$  个服务发生绑定失效的概率;  $R_{\text{Sec}}$  为其他问题如协议使用引起的服务绑定失效概率;  $\lambda_{i,j}$  为节点  $i$  和  $j$  之间网络连接的失效率。

结合第二节服务功能性属性的可靠性分析,根据非齐次泊松分布的性质,服务的准确性可靠度模型为:

$$R_{(x|t)} = P\{N(t+x) - N(t) = 0\} = \exp(m(t) - m(t+x)) \quad (11)$$

将服务的功能性属性可靠性和非功能性属性可靠性有效集成,可得单个服务的可靠度量模型为

$$R_S = R_U * R_{(x|t)} = \prod_{i=1}^N (1 - R_{i,\text{Bind}}) * \exp(-\lambda_{i,j} T(i,j)) * \left( \frac{\lambda^k}{k!} e^{-\lambda} - \frac{n\lambda^k}{Nk!} e^{-\lambda} - \frac{R_{\text{Sec}}\lambda^k}{k!} e^{-\lambda} + \frac{nR_{\text{Sec}}\lambda^k}{Nk!} e^{-\lambda} \right) * \exp(m(t) - m(t+x)) \quad (12)$$

假定网络环境满足服务信息的正常传输,在服务通信的过程中,不会出现网络拥塞或者带宽受限引起

服务失效的问题。则单个服务的可靠度量模型为

$$R_S = \left( \frac{\lambda^k}{k!} e^{-\lambda} - \frac{n\lambda^k}{Nk!} e^{-\lambda} - \frac{R_{\text{Sec}}\lambda^k}{k!} e^{-\lambda} + \frac{nR_{\text{Sec}}\lambda^k}{Nk!} e^{-\lambda} \right) * \prod_{i=1}^N (1 - R_{i,\text{Bind}}) * \exp(m(t) - m(t+x)) \quad (13)$$

#### 5 结束语

文中介绍了面向服务架构的特点,从服务功能性属性和非功能性属性可靠性度量的角度出发,将服务的准确性和可用性集成起来,给出了服务可靠性的定义,通过对服务整个生命周期中的服务注册、查找、绑定组合以及服务执行等各阶段的分析,给出了各个阶段的可靠性模型,将服务的可靠性分解到全生命周期中,综合服务过程中可能出现的各种失效情形,集成各阶段模型和服务功能性属性模型,给出单个服务的可靠性模型。其中,网络环境对信息传输无影响假设条件属于理想化条件,实际的服务过程可能会受到网络拓扑结构、网络带宽、网络信息传输量等因素的影响,网络脆弱性和可靠性问题对整个服务的可靠性有着至关重要的影响,后续研究需要考虑。

#### 参考文献:

- [1] 任志宏. Web 服务复合的若干关键问题研究[D]. 北京:中国科学院软件研究所,2003.
- [2] 钟读杭. Web 服务组合的可靠性预测研究[D]. 长沙:国防科学技术大学,2007.
- [3] 苏志勇,周颖,李必信. 面向用户的 Web 服务可靠性计算模型[J]. 东南大学学报:自然科学版,2008,38(4):605-610.
- [4] 马江. 服务可靠性度量及改善的研究[D]. 上海:上海交通大学,2009.
- [5] 梁胜杰,张志华. 操作剖面对软件可靠性评估可信度的影响研究[J]. 计算机工程与应用,2006,42(14):102-104.
- [6] Evans C, Chappell D, Bunting D, et al. Web Services Reliability (WS-Reliability) version1.0[EB/OL]. 2003. <http://www.hitachi.co.jp/Prod/comp/softl/wsrn>.
- [7] 孙志安,裴晓黎. 软件可靠性工程[M]. 北京:北京航空航天大学出版社,2009.
- [8] 蔡开元. 软件可靠性工程基础[M]. 北京:清华大学出版社,1995.
- [9] Chang Yen-Chang, Lie Ching-Ti. A generalized JM model with applications to imperfect debugging in software reliability[J]. Applied Mathematical Modeling, 2009, 33: 3578-3588.
- [10] Shyur Huan-Jyh. A stochastic software reliability model with imperfect-debugging and change-point[J]. Journal of System and Software, 2003, 66: 135-141.
- [11] Pham H, Zhang Xuemei. NHPP software reliability and cost

ta), Statlog(Heart), 作为实验数据。从中随机选取一部分点作为训练样本点, 余下的点作为测试样本点。所有实验都是在个人电脑上的 Matlab2009 上进行的。

实验得到的分类的准确度如表 1 所示。

表 1 分类准确度

数据集	SVM	LSSVM	WCS-LSSVM
	准确度/%	准确度/%	准确度/%
Ripley	90.80	90.70	91.40
Australian	85.50	85.00	86.52
German	72.71	75.26	75.26
heart	83.89	84.44	85.00

实验结果表明, 新提出的算法比支持向量机和最小二乘支持向量机的分类准确度要高。最小二乘支持向量机可以视为基于类内离散度的最小二乘支持向量机模型的一个特例(当  $\beta = 0$  时)。

4 结束语

文中通过将类内离散度和最小二乘支持向量机结合得到了新的模型, 并给出了它的数学模型。利用新的模型进行分类试验, 结果表明分类准确度得到了提高。当然类内离散度矩阵有一个缺点, 就是当样本个数少于样本维数时, 类内离散度矩阵奇异, 但是在绝大多数情况下样本数量是要大于样本维数的, 另外还可以进行降维处理。文中实验研究也仅局限在仿真和对比, 在应用方面还需努力。

参考文献:

[1] Cortes C, Vapnik V. Support-vector networks[J]. Machine Learning, 1995, 20(3): 273-297.

[2] 刘文, 吴陈. 一种新的中文文本分类算法—One Class SVM-KNN 算法[J]. 计算机技术与发展, 2012, 22(5): 83-86.

[3] 王辉. 主成分分析及支持向量机在人脸识别中的应用[J]. 计算机技术与发展, 2006, 16(8): 24-26.

[4] 雷虎, 樊泽明. 基于 SVM 和 D-S 理论的三维人脸识别[J]. 计算机技术与发展, 2014, 24(6): 75-78.

[5] Scholkopf B, Smola A J, Williamson R C, et al. New support vector algorithms[J]. Neural Computation, 2000, 12(5): 1207-1245.

[6] Suykens J A K, Vandewalle J. Least squares support vector machine classifiers[J]. Neural Processing Letters, 1999, 9(3): 293-300.

[7] Lin Chunfu, Wang Shengde. Fuzzy support vector machines[J]. IEEE Transactions on Neural Network, 2002, 13(2): 464-471.

[8] Mangasarian O L, Musicant D R. Lagrangian support vector machines[J]. Journal of Machine Learning Research, 2001, 1: 161-177.

[9] Jayadeva, Khemchandani R, Chandra S. Twin support vector machines for pattern classification[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(5): 905-910.

[10] Zafeiriou S, Tefas A, Pitas I. Minimum class variance support vector machines[J]. IEEE Transactions on Image Processing, 2007, 16(10): 2551-2564.

[11] An W J, Liang M G. Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises[J]. Neurocomputing, 2013, 110: 101-110.

[12] 阎辉, 张学工, 李衍达. 支持向量机与最小二乘法的关系研究[J]. 清华大学学报: 自然科学版, 2001, 41(9): 77-80.

[13] 汪廷华, 陈峻婷. 核函数的选择研究综述[J]. 计算机工程与设计, 2012, 33(3): 1181-1186.

[14] 郭丽娟, 孙世宇, 段修生. 支持向量机及核函数研究[J]. 科学技术与工程, 2008, 8(2): 487-490.

[15] Mika S, Ratsch G, Weston J, et al. Fisher discriminant analysis with kernels[C]//Proceedings of the 1999 IEEE signal processing society workshop on neural networks for signal processing IX. Madison, WI: IEEE, 1999: 41-48.

[16] 郭啸, 魏延, 吴瑕. 基于混合核函数的支持向量机[J]. 重庆理工大学学报: 自然科学, 2011, 25(10): 66-70.

(上接第 70 页)

models with testing coverage[J]. European Journal of Operational Research, 2003, 145(2): 445-454.

[12] Huang C Y, Kuo S Y, Lyu M R. An assessment of testing-effort dependent software reliability growth models[J]. IEEE Transactions on Reliability, 2007, 56(2): 198-211.

[13] Mallick S, Kushwaha D S. LWSDM: layered Web service discovery mechanism[J]. Advanced in Information Sciences and Service Sciences, 2010, 2(3): 25-31.


[14] Canfora G, Penta M D, Esposito R, et al. A framework for QoS

-aware binding and re-binding of composite Web service[J]. Journal of Systems and Software, 2008, 81(10): 1754-1769.

[15] 谢景燕, 安金霞, 朱纪洪. 考虑不完美错峰情况的 NHPP 类软件可靠性增长模型[J]. 软件学报, 2010, 21(5): 942-949.

[16] Dai Yuanshun, Yang Bo, Dongarra J. Cloud service reliability modeling and analysis[EB/OL]. 2009. <http://www.netlib.org/utk/people/JackDongarra/PAPERS/Cloud-Shaun-Jack.pdf>.

基于SOA架构软件服务可靠性评价方法研究

作者：[张亚](#)，[ZHANG Ya](#)  
作者单位：[江苏自动化研究所, 江苏 连云港, 222061](#)  
刊名：[计算机技术与发展](#)  
英文刊名：[Computer Technology and Development](#)  
年，卷(期)：2015(4)

引用本文格式：[张亚](#). [ZHANG Ya](#) [基于SOA架构软件服务可靠性评价方法研究](#)[期刊论文]-[计算机技术与发展](#) 2015(4)