基于故障树分析的软件安全性测试研究

赵跃华，朱媛媛
（江苏大学 计算机科学与通信工程学院，江苏 镇江 212013）

摘　要：针对软件安全性测试的本质特征在于快速降低由于软件失效而导致系统事故的风险，结合基于 Bayesian统计理论的测试方法，建立一套构建安全性测试剖面，并由此产生测试用例的测试方法。该方法运用故障树分析技术，对各模块发生故障对系统安全性的影响进行分析，找出影响较大的关键性模块，然后利用分析结果构建安全性测试剖面。最后给出了测试停止的标准。通过对例子的分析可知，本方法在快速降低软件事故风险方面比现有软件测试方法更有效。

关键词：安全性；故障树分析；安全度；安全性测试剖面
中图分类号：TP311　　**文献标志码**：A　　**文章编号**：1001-3695(2013)06-1760-04
doi:10.3969/j.issn.1001-3695.2013.06.041

Research on software safety testing based on fault tree analysis

ZHAO Yue-hua, ZHU Yuan-yuan
(*School of Computer Science & Telecommunication Engineering, Jiangsu University, Zhenjiang Jiangsu* 212013, *China*)

Abstract: Considering that the object of software safety testing is to reduce the risk of accident resulted by software failures sharply, combining with the test method based on Bayesian statistical theory, this paper presented a software safety testing method based on the safety testing profile. It used the technology of fault tree analysis to analyze the fault probability of all sections of the software to find the key sections that would be influence the safety. Based on that result, safety testing profile was constructed, as well as described the testing cases generating method in detail. Finally, it gave the software safety testing stopping rule. Analysis of the example indicates that the method provided above is effective to reduce the software accident risk.

Key words: safety; fault tree analysis; safety degree; safety testing profile

随着计算机系统和计算机软件发展的日新月异，计算机软件已经应用到很多重要的领域，如航空航天、国防、交通运输、核电能源和医疗卫生等系统中。一旦这些系统失效将会导致生命财产的重大损失以及环境可能遭受严重的破坏，此类系统称为安全关键系统[1]。安全关键系统对软件的可靠性和安全性有很高的要求，因此，对于软件系统的可靠性和安全性测评就势在必行。

为了更好地对安全关键软件进行测评，Parnas、Howden 等人[2,3]提出了基于经典统计假设理论的测试方法，为安全关键软件的可靠性测评奠定了取样理论基础；Littlewood、Miller 等人[4,5]则从另一方面提出了基于 Bayesian 统计理论的测试方法。利用这些方法对可靠性进行测评，其评估结果具有很高的可信度。因此，可靠性测评已经有一套完善的理论。相对而言，安全性测评就远不如可靠性的。在工程实践中，人们也往往把软件安全性测试作为可靠性测试来实施，或者着重于测试一些例外情况[6]，而忽略软件安全性本身的性质。

软件安全性与软件可靠性一样，是与软件失效密切相关的。软件安全性的概念最早是由美国加州的 Leveson[7] 教授提出的。MIL-STD-882C《系统安全性大纲》中对安全性的定义是，安全性是指避免危险条件发生，保证己方人员、设施、财产、环境等免于遭受灾难性事故或重大损失的能力。通过以上定义可以发现，软件安全性更加关注那些引起灾难性事故发生的失效，是为了避免灾难性事故的发生。因此，软件安全性测试的目标就是通过测试发现软件中存在的缺陷，并进行修正，以快速降低由于软件失效而导致系统事故的风险。所以，如果能够分析出系统的危险所在，并加强对这些部分的测试，既可以最大程度地保持系统的安全性又可以整体提高测试的效率。

本文正是基于这样的思想，提出利用故障树分析技术对各模块发生故障对系统安全性的影响进行分析，找出安全关键模块，然后对其依据安全度设计测试用例进行安全性测试。

1　基于 Bayesian 统计理论的软件测试

基于 Bayesian 统计理论的软件测试方法[8]是一种软件验证测试的方法，它也是统计测试的一种。在测试之前根据给定软件的可靠性和安全性指标，计算出所需的测试用例数，然后基于软件运行剖面进行抽样，获得所需的测试用例进行测试，并且在测试过程中无失效发生。

1.1　基于 Bayesian 统计理论的软件测试方法

在软件的验收测试中，规定测试不应该发现失效（否则拒绝该软件）。所以，在执行测试之前就要确定无失效测试用例数。统计取样测试方法，都是假定软件的操作失效概率为 p，且每次操作都满足贝努利（Bernoulli）实验的统计独立性，认为在 n 次测试中，出现 R 次软件失效的概率为二项分布，即

$$P(R=r|p)=C_n^r p^r(1-p)^{n-r} \tag{1}$$

由于软件失效数遵从式（1）所示的贝努利分布，则其失效

收稿日期：2012-10-10；**修回日期**：2012-11-24

作者简介：赵跃华（1958-），男，江苏苏州人，教授，博士，主要研究方向为信息安全（zhaoyh@ujs.edu.cn）；朱媛媛（1989-），女，江西赣州人，硕士研究生，主要研究方向为信息安全.

概率的概率密度函数的先验分布应为 Beta 分布,即

$$f(p)=\frac{1}{B(a,b)}p^{a-1}(1-p)^{b-1} \tag{2}$$

若测试用了 n 个测试用例,其中发生 r 个失效。那么失效概率密度函数的后验分布为

$$f(p|r,n,a,b)=\frac{1}{B(a+r,b+n-r)}p^{a+r-1}(1-p)^{b+n-r-1} \tag{3}$$

对于无先验知识的情况,有 $a=b=1$,则有

$$f(p|r,n,1,1)=\frac{1}{B(1+r,1+n-r)}p^{r}(1-p)^{n-r} \tag{4}$$

对于给定的安全性指标(p_0,C),当一次测试中不存在测试失效时的测试用例数 N 为满足式(5)中 n 的最小整数:

$$f(p\leqslant p_0)=\int_0^{p0}\frac{1}{B(1,1+n)}(1-p)^n d_p\geqslant C \tag{5}$$

由式(5)得

$$n=\left\lceil\frac{\ln(1-C)}{\ln(1-p_0)}-1\right\rceil$$

为了达到给定的安全性指标(p_0,C),需要无失效地测试 N 个测试用例;如果在测试过程中出现失效,则需要排除错误,并重新选取测试用例进行测试。

1.2　基于 Bayesian 统计理论的软件测试方法的不足

对于给定的一组安全性测试指标(p_0,C),利用式(5)计算所需的无失效测试用例数,如表 1 所示。安全关键软件的安全性要求都是很高的,一般失效率在 $10^{-7}\sim10^{-9}$。要达到这个失效率所需的测试用例数将非常大,这给工程实践带来很大的负担。为了解决这个问题,针对软件安全性测试的本质特征在于降低由于软件失效而导致系统事故的风险,可以采用故障树分析技术找出系统中安全性影响较大的模块,针对这些模块进行测试,以提高测试针对性和测试效率。

表 1　无失效测试用例数

p_0	C		
	0.90	0.99	0.999
10^{-5}	$2.3\times e^5$	$4.6\times e^5$	$6.9\times e^5$
10^{-6}	$2.3\times e^6$	$4.6\times e^6$	$6.9\times e^6$
10^{-7}	$2.3\times e^7$	$4.6\times e^7$	$6.9\times e^7$
10^{-8}	$2.3\times e^8$	$4.6\times e^8$	$6.9\times e^8$

2　安全性测试的故障树模型分析

故障树分析法主要用于分析大型复杂系统的可靠性及安全性,它被公认为是目前对复杂系统可靠性、安全性进行定性分析的一种有效方法。

2.1　建模思想

在建立故障树模型之前,要对规格需求说明书进行详细的分析,通过功能危险性分析技术(FHA)找出系统中所有可能出现的各种不安全状态(或者关键失效状态),列出软件关键安全故障事件表。然后对表中每一个故障事件建立一棵故障树。

在故障树建立的过程中,本文以 FHA 找出的故障事件作为顶事件,通过分析寻找出导致顶事件故障发生的所有可能的直接原因,这些原因又被称之为中间事件。顶事件与中间事件之间用“与”门或者“或”门等进行连接。接着分析寻找每一个中间事件发生的所有可能原因,以此类推,直至追踪到最后一级基本事件,也即底事件。软件故障树分析的最底层取决于分析的要求,原则上可以深入到程序的编码或语句。为了与软件统计测试进行结合,本文采用各功能模块失效作为底事件进行故障树建模。

2.2　模型建立

假设软件系统由若干相互独立的功能模块组成,各软件功能模块的事件状态为两态:工作或失效。如存在软件系统 $S=\{m_1,m_2,m_3,m_4,m_5,m_6,m_7\}$。

假设以某一个系统危险失效作为顶事件,通过层层分析可以得到软件系统故障树模型如图 1 所示,其中,G_i 表示中间事件。

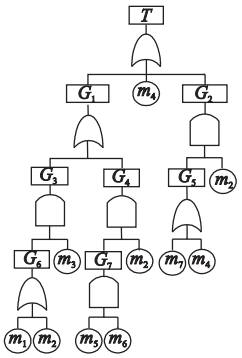

图1　软件系统故障树模型

为了对故障树进行安全性分析,找出各功能模块失效对顶事件的影响,需要求出故障树的最小割集。目前对故障树最小割集进行求解的方法主要有上行法、下行法和计算机算法。利用下行法求解最小割集的过程如图 2 所示。根据 AB + A = A 原则,可以得出最小割集有 $K_1=\{m_4\}$,$K_2=\{m_3,m_1\}$,$K_3=\{m_3,m_2\}$,$K_4=\{m_2,m_5,m_6\}$,$K_5=\{m_7,m_2\}$,共五个。其中,K_1 为一阶割集,K_2、K_3、K_5 为二阶割集,K_4 为三阶割集。

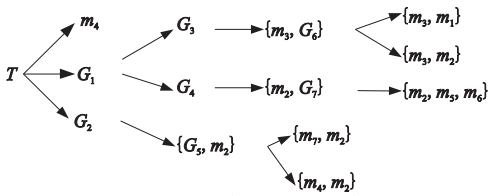

图2　下行法求解最小割集

2.3　故障树模型分析

首先分析最小割集中各功能模块失效对顶事件的影响,也即模块的安全度。

由于模块 m_4 独立构成一个最小割集,其模块失效将会直接导致系统故障,因此,属于安全关键模块,其安全度最高。

从故障树中可以看出,越低阶的故障事件与顶事件的联系越直接,影响越大;同一个故障事件出现在不同的割集中,说明那个事件可以通过不同的途径导致顶事件发生,它对顶事件的影响也比较大。表 2 就是通过这种定性分析得出各功能模块安全度次序的。在所有的系统故障树最小割集中,假设割集的个数是 k,割集 E_r 中含有 m_r 个事件,则基本事件 X_i 的安全度可以用公式:$I_k(i)=\frac{1}{k}\sum_{r=1}^{k}\frac{1}{m_r(X_i\in E_r)}$来计算 ,其中 $m_r(X_i\in E_r)$表示对 $X_i\in E_r$ 的割集进行计算,不属于的割集不加入计算。

表 2　功能模块安全度次序表

模块	在一阶割集中出现次数	在二阶割集中出现次数	在三阶割集中出现次数	安全度次序
m_1		1		4
m_2		2	1	2
m_3		2		3
m_4	1			1
m_5			1	5
m_6			1	5
m_7		1		4

于是,对图 2 求出的五个最小割集,可计算各模块失效(基本事件)的安全度:

$$I_k(m_1)=\frac{1}{k}\sum_{r=1}^{k}\frac{1}{m_r(X_i\in E_r)}=\frac{1}{5}(\frac{1}{2})=\frac{1}{10}$$

$$I_k(m_2)=\frac{1}{k}\sum_{r=1}^{k}\frac{1}{m_r(X_i\in E_r)}=\frac{1}{5}(\frac{1}{2}+\frac{1}{3}+\frac{1}{2})=\frac{4}{15}$$

$$I_k(m_3)=\frac{1}{5}(\frac{1}{2}+\frac{1}{2})=\frac{1}{5}$$

$$I_k(m_4)=\frac{1}{5}(1)=\frac{1}{5},I_k(m_5)=\frac{1}{5}(\frac{1}{3})=\frac{1}{15}$$

$$I_k(m_6)=\frac{1}{5}(\frac{1}{3})=\frac{1}{15},I_k(m_7)=\frac{1}{5}(\frac{1}{2})=\frac{1}{10}$$

对计算结果进行排序得:

$$I_k(m_2)>I_k(m_3)=I_k(m_4)>I_k(m_1)=I_k(m_7)>I_k(m_5)=I_k(m_6) \tag{6}$$

式(6)与表 2 安全度次序比较可以发现,除了模块 m_4,其他结果都是一致的。这是由于,模块 m_4 单独构成一个最小割集,它的安全度最高,所以不能用这种方法直接计算。因此,需要对计算的结果进行调整,使它满足安全度定性分析。

在一棵故障树中,将最大的安全度值赋予一阶割集的模块,并把一阶割集的安全度值赋给具有最大安全度值的模块。这样做虽然会与分析的结果有所不同,但是整体上还是符合的。在这个例子中 $I_k(m_4)=\frac{4}{15}$、$I_k(m_2)=\frac{1}{5}$。

通过以上方法,能够确定系统中的安全重要模块以及它们对系统安全性的影响程度(安全度),从而可以利用这些安全性定性分析的结果来构造安全性测试剖面,进行安全性测试。

3 软件安全性测试

软件安全性测试的目的是将引起软件危险事故的失效降低到足够水平。因此,不仅要考虑软件失效的频率,还要考虑它们对安全性的影响。在安全性测试时,应该把测试资源应用于那些对安全性影响较大的功能模块的运行、操作上,只有这样,所发现的缺陷对降低风险事故的贡献才大,才能提高测试的效率。利用故障树分析方法找出了系统中的安全重要模块,并且给出了各模块的安全度。接着,利用这些模块来构建软件安全性测试剖面进行测试。

在故障树中,底事件代表对应模块的失效,由于软件开发的高内聚和低耦合性,软件各个模块的功能趋于单一,因此,软件的一个模块失效也即对应这一个功能失效。因此可以利用模块的功能失效以及失效的影响程度(安全度)来建立重要功能集合 PI,$PI=\{PI_i\mid PI_i=\langle PH_i,I_k(i)\rangle,i=1,2,\cdots,n\}$。其中:$PI_i$ 表示第 i 个重要功能元素;PH_i 表示第 i 个功能失效;$I_k(i)$ 表示第 i 个失效对应的安全度;n 表示所有安全相关性功能失效的个数,即故障分析得出的基本事件的个数。第 2 章中的例子得出的重要功能集合为

$PI=\{\langle PH_1,0.1\rangle,\langle PH_2,0.2\rangle,\langle PH_3,0.2\rangle,\langle PH_4,0.27\rangle,\langle PH_5,0.065\rangle,\langle PH_6,0.065\rangle,\langle PH_7,0.1\rangle\}$

3.1 安全性测试功能剖面的生成

重要功能集合就是可能的功能失效以及功能失效对系统安全性影响的一个集合,其中所描述的功能是软件系统功能的一部分,也是统计测试功能剖面的一部分。因此,需要利用重要功能集合对统计测试的功能剖面进行调整。

假设用统计测试运行剖面生成方法生成了软件系统的所有功能剖面和运行剖面。安全性测试中只对重要功能集合中的功能进行测试,将它们从原功能剖面中抽取出来,并调整每个功能的概率值,从而可生成安全性测试的功能剖面。

安全性测试功能剖面:$\{FP_i\mid FP_i=\langle F_i,P_{Fi}\rangle,i=1,2,\cdots,nf\}$。其中:$FP_i$ 是第 i 个功能剖面元素;F_i 是第 i 个重要功能;P_{Fi} 是第 i 个重要功能发生的概率;nf 是当前系统模式下重要功能的个数。具体做法如图 3 所示。

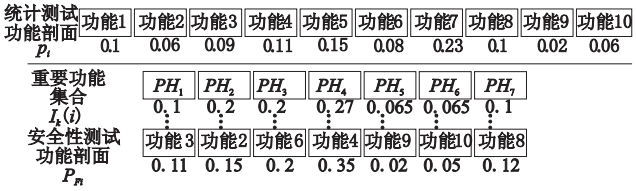

图3 安全性测试功能剖面生成

假设已知软件统计测试的功能剖面、功能及对应的概率如图 3 第一行数据所示。通过故障树分析得出的重要功能集合对应统计测试功能剖面的功能 3、2、6 等七个功能。安全性测试功能剖面中重要功能的概率计算过程如下:

a)计算 $P_{Fi}=I_k(i)\times p_i$,这是一个中间量,例子中的 $P_{F1}=I_k(1)\times p_1=0.009$,$P_{F2}=0.012$,$P_{F3}=0.016$,$P_{F4}=0.03$,$P_{F5}=0.0013$,$P_{F6}=0.0039$,$P_{F7}=0.01$,$\sum P_{Fi}=0.0822$。

b)计算重要功能发生概率 $P_{Fi}=\frac{P_{Fi}}{\sum P_{Fi}}$,$P_{F1}=\frac{P_{F1}}{\sum P_{Fi}}=\frac{0.009}{0.082}=0.11$,$P_{F2}=0.15$,$P_{F3}=0.2$,$P_{F4}=0.35$,$P_{F5}=0.02$,$P_{F6}=0.05$,$P_{F7}=0.12$,$\sum P_{Fi}=1$。

将重要性功能从统计测试功能剖面中抽取出来时,使用重要性抽样原理[9]。抽取出来的功能概率和为 $Pc=p_3+p_2+p_6+p_4+p_9+p_8+p_{10}=0.52$,而原功能剖面的概率总和为 1,则测试的加速因子 $\wedge(x)=Pc<1$。在抽样测试的过程中,不仅提高了对安全重要功能的测试,而且对每个重要功能的测试力度还根据它的重要性进行了调整。

3.2 安全性测试剖面的生成

软件安全性测试功能剖面中的每个功能的完成,是通过输入一系列输入变量值来实现的。规定这些输入变量具体取值区间,在取值区间内把它们看做是独立的。运行剖面需要分别描述各功能输入变量的取值区间,即区间的上下限,和各输入变量在其取值区间内的概率分布。该概率分布是条件概率分布,是指在功能被选中的条件下,该功能各输入变量取值的概率分布。根据输入变量和它的概率分布来构造运行剖面。对于每一个功能需要构造运行剖面,之后对不同功能下的相同的运行剖面需要进行合并,构成最终的软件的安全性测试剖面。安全性测试剖面就是一组运行及它们的概率组合。

安全性测试剖面:$OP=\{op_i\mid op_i=\langle \text{opertion}_i,p_i\rangle,i=1,2,\cdots,n\}$,其中对任意的 i、j 有 $\text{operation}_i\cap\text{operation}_j=\varnothing,i\neq j$,$\sum_{i=1}^{n}p_i=1$。$\text{operation}_i$ 表示运行,p_i 表示对应运行的发生概率。安全性测试剖面生成的具体做法如图 4 所示。这组运行的概率和 $\sum_{i=1}^{n}p_i=1$。

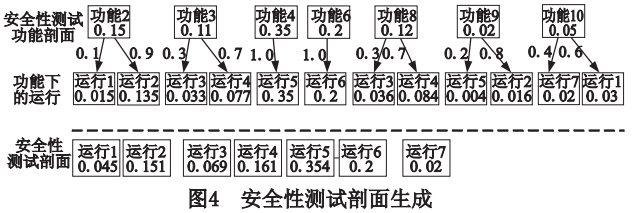

图4 安全性测试剖面生成

3.3 安全性测试用例的生成

软件安全性统计测试用例的产生包括两方面:a)依据安全性测试剖面选择相应的运行;b)产生触发软件运行的相应的输入变量的具体取值。

抽取运行的过程如下:将安全性测试剖面中所有运行 op_i 发生的概率 p_i 求前 j 项和 S_j,$S_j=\sum_{i=1}^{j}p_i$,形成一个数列$\{S_j\}$,其中 $j=1,2,\cdots,n$,n 为安全性测试剖面中运行总数,规定 $S_0=0$,并有 $S_1=p_1$,$S_n=1.0$,$S_j-S_{j-1}=p_j$。任给一个随机数 $\eta\in(0,1.0)$,观察 η 落在哪个区间,若 η 满足 $S_{j-1}<\eta\leqslant S_j$,则该随机数 η 与 p_j 这个概率值对应,那么这次随机抽到的运行为 operation$_j$。

要进行第二次抽样来确定运行中每个输入变量将取到的具体取值。由于输入变量的取值类型可以是离散的,也可以是连续的,在随机抽样时要分别考虑这两种类型输入变量的抽样方法。连续性的输入变量要在其取值区间内,依概率密度函数抽样;离散型的要在其可取值集合内依概率分布抽样。

通过对运行和各输入变量取值两个步骤的抽样,就生成了一个测试数据。每个输入变量的测试数据构成一个输入向量,用于测试被选定的功能。不断重复上述步骤,直到生成所需数量的测试数据为止。

4 安全性测试的停止标准

对于给定的安全性测试指标,可以计算出无失效测试所需的测试用例数,并且在 3.1 节中,给出了安全性测试的加速因子$\wedge(x)=Pc$,下面计算安全性测试所需的测试用例数。

依据统计测试剖面产生 n 个测试用例,其中发生 r 个失效。根据安全性测试运行剖面抽样需要抽取 n/Pc 个测试用例才能导致这 r 个失效。那么 p 的后验分布为

$$f(p\mid r,n/Pc,1,1)=\frac{1}{B(1+r,1+n/Pc-r)}p^{r}(1-p)^{n/Pc-r}$$

假设给定的安全性指标为(p_0,C),根据$f(p\leqslant p_0)=\int_0^{p_0}\frac{1}{B(1,1+n/Pc)}(1-p)^{n/Pc}d_p\geqslant C$。可以计算出安全性测试所需的测试用例数,$n=\left\lceil Pc\frac{\ln(1-C)}{\ln(1-p_0)}-1\right\rceil$。

由于 $Pc<1$,例子中给出的 Pc 值为 0.52,也即安全关键性功能占总的功能剖面的比例为 0.52。当给定的安全性指标(p_0,C)为$(10^{-8},0.99)$时,基于 Bayesian 统计理论的软件测试方法所需的测试用例数为 $4.6\times e^8$;安全性测试所需的测试用例数为 $2.39\times e^8$,安全性测试用例数约为原来的 1/2。

安全性测试所需测试用例数将小于基于 Bayesian 统计理论的软件测试所需的用例数。如果需要测试的安全性相关功能是极小概率事件,这样的方法会显得更加有效。

5 结束语

本文对现有的安全关键软件的测评方法进行研究,分析了它们在安全性测试中存在测试用例数过大的局限,提出了一种利用故障树的分析结果对软件统计测试功能剖面进行调整,生成安全性测试剖面的过程,利用该测试剖面生成测试用例进行安全性测试。这样便能加大安全相关的功能在测试中的测试力度,以更好地满足安全性测试的要求。通过例子分析可以发现,采用这种方法进行软件测试可以有效地减少测试用例量,提高测试效率。本方法不仅可以提高对安全性相关功能的测试,而且对安全性影响大的功能测试也会更加充分。

参考文献:

[1] 杨仕平,桑楠,熊光泽. 安全关键软件的防危性测评技术研究[J]. 计算机学报,2004,27(4):442-450.

[2] PARNAS D L, Van SCHOUWEN A J, KWAN S P. Evaluation of safety-critical software[J]. Communication of ACM, 1990, 33(6): 636-648.

[3] HOWDEN W E. Good enough versus high assurance software testing and analysis methods[C]//Proc of the 3rd International High Assurance Systems Engineering Symposium. Washington DC: IEEE Computer Society, 1998: 166-175.

[4] LITTLEWOOD B, STRIGINI L. Assessment of ultra-high dependability for software-based systems [J]. Communications of the ACM, 1993, 36(11): 69-80.

[5] MILLER W M, MORELL L J, NOONANR E, *et al*. Estimating the probability of failure when testing reveals no failures [J]. IEEE Trans on Software Engineering, 1992, 18(1): 33-43.

[6] TANG D, HECHT H. A possible approach to assessing dependability for safety-critical software[C]//Proc of the 2nd Annual Conference of Computer Assurance. 1997.

[7] LEVESON N. Software safety: why, what and how[J]. ACM Computing Surveys, 1986, 18(2): 125-163.

[8] 覃志东,雷航,桑楠,等. 安全关键软件可靠性验证测试方法研究[J]. 航空学报,2005,6(3):334-339.

[9] 郭瑛,艾渤,钟章队. 重要性采样研究进展[J]. 信息与电子工程,2011,9(5):604-609.

(上接第 1759 页)

[7] 杨斌,张卫冬,张利欣,等. 基于 SOA 的物联网应用基础框架[J]. 计算机工程,2010,36(17):95-97.

[8] 蒋林岑,季一木. 物联网业务模型描述语言的研究与设计[J]. 计算机技术与发展,2012,22(2):249-253.

[9] 侯丽珊,金芝,吴步丹. 需求驱动的 Web 服务建模及其验证:一个基于本体的方法[J]. 中国科学 E 辑:信息科学,2006,36(10):1189-1219.

[10] 汤健,郑丽伟,金芝. 需求驱动的服务 agent 协作及其协商框架[J]. 模式识别与人工智能,2008,21(5):643-653.

[11] 田加正. 物联网环境下基于 QoS 的 Web 服务组合研究[D]. 北京:中国石油大学,2011.

[12] 许蕾,陈林,徐宝文. 用户需求驱动的 Web 服务测试[J]. 计算机学报,2011,34(6):1029-1040.

[13] 万长林,陈立民,王竹晓,等. 语义 Web 服务组合中的服务建模及规划算法[J]. 智能系统学报,2009,4(6):490-496.

[14] 李曼,王大治,杜小勇,等. 基于领域本体的 Web 服务动态组合[J]. 计算机学报,2005,28(4):644-650.

[15] 曹步清,李兵. 一种网络化软件的按需服务发现方法[J]. 计算机科学,2012,39(1):96-123.

[16] LIU Yu-tu, NGU A H, ZENG L Z. QoS computation and policing in dynamic Web service selection[C]//Proc of the 13th International World Wide Web Conference. New York: ACM Press, 2004: 66-73.

[17] TAO Fei, ZHAO Dong-ming, HU Ye-fa, *et al*. Resource service composition and its optimal selection based on particle swarm optimization in manufacturing grid system[J]. IEEE Trans on Industrial Informatics, 2008, 4(4): 315-327.

[18] 刘必欣. 动态 Web 服务组合关键技术研究[D]. 长沙:国防科学技术大学,2005.

[19] 龚小勇. 基于 QoS 的 Web 服务发现与组合方法研究[D]. 重庆:重庆大学,2008.