构件化软件系统安全性分析技术研究

万永超 周兴社 董云卫
(西北工业大学计算机学院 西安 710129)

摘 要 复杂系统的软件安全性分析中存在众多的含糊表述和不确定性问题,主观评价和模糊集理论即为处理此类问题的有效方法。首先对复杂软件涉及到的安全性要素进行模糊数学化表述,按照构件化的思想,对单个构件/模块的安全度进行分析,进而利用 Dempster-Shafer 证据理论对整个软件系统进行安全度综合,实现对软件系统的安全性分析和评估,最终按照 DO-178B 标准给出软件的安全等级,并通过实例加以说明。

关键词 构件化软件,模糊集,主观评价,安全度,综合
中图法分类号 TP391 **文献标识码** A

Study on Component-based Software Safety Analysis

WAN Yong-chao ZHOU Xing-she DONG Yun-wei
(Department of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China)

Abstract Many obscure expressions and uncertainties exit during the process of safety analysis for complicated safety-critical software, while the theory of fuzzy sets and subjective evaluation is an effective methodology to deal with these problems. We presented the fuzzy expressions of the software safety factors, then analyzed the safety score of single component. After that, we synthesised the safety score of subsystem and system quantitatively by using the fuzzy operations and evidential reasoning approach. Finally, an example was presented to demonstrate the proposed software analysis and synthesis method.

Keywords Component-based software, Fuzzy sets, Subjective method, Safety score, Synthesis

随着计算机技术的飞速发展,软件已经逐渐应用到各种安全关键领域,例如航空航天、核电、国防、工业控制、交通运输、金融等。这类软件通常都规模庞大、结构复杂,对可靠性和安全性有着极高的要求。长期以来,由于此类系统的软件失效和安全性问题,造成生命财产损失,甚至给环境造成严重灾难的事例层出不穷[1],这就使得软件的安全性分析、评估和保障变得尤其重要。

但由于众多复杂因素对该类软件安全性的影响,以及因素之间的相互关系都难以准确、客观地描述出来,导致人们很难掌握进行软件安全性评估所需要的准确信息。主观评价方法则在一定程度上可以有效弥补人们不能很好地掌握模糊现象而带来的信息缺乏。该方法基于开发人员长期的实践经验和总结,基于对模糊现象的主观判断,是软件安全性评估中不可或缺的组成部分。而模糊集理论[2]作为主观评价方法的重要手段,已经得到一定的研究和应用。

本文基于模糊集理论和系统模块化/构件化分层的思想,提出对于单个构件的安全度评估方法,利用 D-S 证据理论[3,4]对其进行安全度综合,最终得出子系统以及整个软件系统的安全度评估结果,并通过实例加以说明。

1 构件化软件系统

近年来,在中间件技术的基础上,结合软件复用和面向对象的思想,模块化软件设计和开发作为一种提高软件生产率和软件质量的有效途径,受到了高度重视,在实践中得到广泛应用,也将是未来软件开发的主流方向。其中,基于构件的软件开发技术(简称 CBSE)[5]尤其得到了广泛发展。

CBSE 技术通过软件成分的重用,提供了一种自底向上的、使用标准软件构件构造系统的有效途径。其设计和开发可以从系统的总体架构入手,将系统逐级分解为子系统和各个构件,最终表示为各个构件和构件之间的交互关系,在高层抽象上指导和验证构件的组装过程,极大地便利了软件开发过程,提高了开发效率,开发过程如图 1 所示。

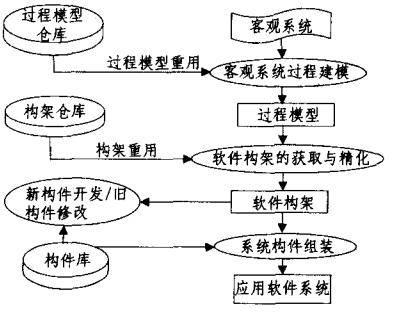

图 1 构件化软件开发

同样地,在软件的安全性分析中也可以借鉴此思想,即先

到稿日期:2009-06-03 返修日期:2009-09-01 本文受国家自然科学基金(60736017),国家 863 高技术研究发展计划基金项目(2007AA010304)资助。

万永超(1986-),男,硕士生,主要研究方向为嵌入式系统及应用,E-mail:happywan1986@gail.com;周兴社(1955-),教授,博士生导师,主要研究方向为嵌入式计算、传感器网络与普适计算;董云卫(1968-),男,教授,博士生导师,主要研究方向为嵌入式软件设计与验证。

对软件模块/构件的安全性做出分析和评估，最终通过组合操作和系统综合实现对整体软件系统的安全性评估。

2 软件安全性的模糊描述

在大型模块化/构件化设计和开发的软件安全性评估领域中，由于相关因素众多，各个构件或者子系统的安全程度大多只会给出自然语言性质的模糊描述，并且信息残缺，难以获取。例如分析人员通常会用一些术语描述各种失效事件，比如“较危险”、“高风险”等，这样就不可避免地导致软件安全性的评估有着一定的不确定性。因此，要建立较为精确完整的失效事件的数学模型，对大型软件的安全性加以详细和定量分析是极其困难的。

目前对于软件的安全性描述，大多基于等级划分的思想，采用各种自然语言变量对其进行模糊的定性化描述[4]。显然，这样的描述和判断对于高可靠的安全关键领域的复杂软件安全性分析有着非常明显和严重的不足。本文首先对这种不精确的、二义性的语言描述进行模糊数学化。

定义 1[6] 给定论域 U 到[0,1]闭区间的任一映射 $\mu_A(U)$，确定 U 的模糊子集 $A=\{(u,\mu_A(u))\mid u\in U\}$，则称 μ_A 为模糊子集 A 的隶属函数，$\mu_A\{u\}$ 称之为 u 对于 A 的隶属度。论域之上的模糊子集 A 由隶属函数 μ_A 表征。

例如，$U=\{1,2,3,\cdots,n-1,n\}$ 代表一个自然语言分类等级的集合（如“Catastrophic”等），则这些语言变量就可用如下模糊子集予以表示：

$$Catastrophic=\{1/0,2/0,\cdots,n-1/0.75,n/1.0\}$$

式中，每一个分子代表分类，分母代表该模糊集合的隶属度。或者可用 $\mu_{Catastrophic}=\{0,\cdots,0,0.75,1.0\}$ 表示。

定义 2[6] 设集合 $U=\{u_1,u_2,\cdots,u_n\}$，$A,B\subseteq U$，则对于集合 A,B 的操作有

1）补运算：$\mu_{\overline{A}}=(\mu_{\overline{A}}^j)_{1\times n}$

式中，$\mu_{\overline{A}}^j=1-\mu_A^j$，$j=1,2,\cdots,n$。

2）交运算：$\mu_{A\cap B}=(\mu_{A\cap B}^j)_{1\times n}$

式中，$\mu_{A\cap B}^j=\min(\mu_A^j,\mu_B^j)$，$j=1,2,\cdots,n$

3）并运算：$\mu_{A\cup B}=(\mu_{A\cup B}^j)_{1\times n}$

$\mu_{A\cup B}^j=\max(\mu_A^j,\mu_B^j)$，$j=1,2,\cdots,n$。

4）笛卡尔积（直积）：$\mu_{A\times B}=(\mu_{A\times B}^{ij})_{n\times n}$

式中，$\mu_{A\times B}^{ij}=\min(\mu_A^i,\mu_B^j)$。

5）组合运算：给定 A 和 B 的笛卡尔积的隶属函数，则 B 的隶属函数为

$$\mu_B=\mu_{A\cdot A\times B}=(\mu_B^j)_{1\times n}$$

而 $\mu_B^j=\max(\min(\mu_A^1,\mu_{A\times B}^{1j}),\cdots,\min(\mu_A^n,\mu_{A\times B}^{nj}))$，$j=1,2,\cdots,n$。

单个构件的失效模式可由失效率 FR、危险后果严重度 CS 和失效后果发生概率 FCP 进行描述[7,8]。FR 表征的是在一定时期内失效发生的频率，CS 为可能发生危险后果的严重性，FCP 则定义了失效发生后导致严重后果的概率。上述参数均可分级评估以及模糊化，如表 1－表 3 所列。值得指出的是，为了表述简单化，表中值多取 0。并且由于各种模糊描述间存在交集，表中每列隶属度之和将可能略大于 1。

表 1 失效率 FR 的模糊化

语言变量	层级分类						
μ_L	1	2	3	4	5	6	7
Highly frequent	0	0	0	0	0	0.75	1
Frequent	0	0	0	0	0.25	0	0.25
Reasonably Frequent	0	0	0	0.75	0	0.25	0
Average	0	0	0.5	0	0.5	0	0
Reasonably low	0	0	0.25	0.25	0.75	0	0
Low	0.25	1	0.75	0	0	0	0
Very low	1	0.25	0	0	0	0	0

表 2 危险后果严重度 CS 的模糊化

语言变量	层级分类						
μ_C	1	2	3	4	5	6	7
Negligible	0	0	0	0	0	0.75	1
Marginal	0	0	0	0.25	1	0.25	0
Critical	0	0.25	1	0.75	0	0	0
Catastrophic	1	0.75	0	0	0	0	0

表 3 失效后果发生概率 FCP 的模糊化

语言变量	层级分类						
μ_E	1	2	3	4	5	6	7
Highly unlikely	1	0.75	0	0	0	0	0
Unlikely	0.25	0	0.75	0	0	0	0
Reasonably unlikely	0	0.25	0.25	0.5	0	0	0
Likely	0	0	0.5	0	0.5	0	0
Reasonably likely	0	0.5	0	0.75	0.25	0.25	0
Highly likely	0	0	0	0	0.75	0.5	0.25
Definite	0	0	0	0.25	0	0.75	1

而该构件的安全度（Safety Score）可用这些参数模糊集（分别用 L,C,E 表示）的操作组合加以描述[9]：

$$S=C\circ E\times L$$

$$\mu_S=\mu_C\circ\mu_E\times\mu_L=(\mu_S^1,\cdots,\mu_S^j,\cdots)$$

式中，$\circ$ 代表组合操作，而 $\times$ 代表笛卡尔积。μ_S^j 为安全度隶属于第 j 类的程度。S 和 μ_S 是某一失效模式下危险因素导致的安全度等级的一种模糊描述。

自然语言变量的安全性表达，即最终需要确定的软件系统的安全等级，按照其隶属度可以分为 5 类，并对其进行模糊化处理，得到表 4。根据美国民航标准 DO-178B 的规定，软件的安全等级按照发生故障后导致的后果严重程度可以分为 5 级（A 级到 E 级逐次降低，见表 4）。

表 4 软件安全等级的模糊化

软件安全等级	语言变量	层级分类						
	μ_S	1	2	3	4	5	6	7
A	Catastrophic	0	0	0	0	0	0.75	1
B	Hazardous	0	0	0	0.5	1	0.25	0
C	Major	0	0.25	0.75	0.5	0	0	0
D	Minor	0.5	0.5	0	0	0	0	0
E	No effect	0.5	0.25	0.25	0	0	0	0

利用最优拟合法，计算安全度 S_i 和每一个安全性表达之间的闵科夫斯基距离，即两个模糊集合的欧几里得距离（以安全等级为 A 的模糊语言变量“Catastrophic”为例）：

$$d_{i1}(S_i,\text{“}atastrophic\text{”})=\left[\sum_{k=1}^{7}(\mu_{S_i}^k-\mu_{Catastrophic}^k)^2\right]^{1/2}$$

式中，d_{ij} 表示其对该模糊语言变量的确信程度，数值越小，说明模糊安全度（安全等级）S_i 距离第 j 个模糊语言变量越近。特殊情况下，当 $d_{ij}=0$ 时，表示 S_i 正好属于第 j 个模糊语言变量的表达。记 d_{iJ}（$J=1,2,3,4$）为与 S_i 最小的距离值，则可令 α_{i1}，α_{i2}，α_{i3} 和 α_{i4} 代表相对距离的倒数：

$$\alpha_{ij}=\frac{1}{d_{ij}/d_{iJ}}\quad(j=1,2,3,4)$$

将 α_{ij}（$j=1,2,3,4$）标准化如下：

$$\beta_{ij}=\frac{\alpha_{ij}}{\sum_{m=1}^{4}\alpha_{im}}$$

式中，$j=1,2,3,4$。

假设给定集合 $H=\{H_1,H_2,\cdots,H_j,\cdots,H_N\}$ 代表一系列自然语言变量的集合，用于安全性表达和评估，H_j 代表第 j 个语言变量，N 为变量的总数目，则最终对于该失效模式 i 的安全性评价可以表示为：

$$S_{(i)}=\{(H_j,\beta_{ij})\mid j=1,\cdots,N\}$$

式中，β_{ij} 代表该评价的置信度，且满足 $\beta_{ij}\geqslant 0$，$\sum_{j=1}^{N}\beta_{ij}\leqslant 1$。当 $\sum_{j=1}^{N}\beta_{ij}=1$ 时，称 $S_{(i)}$ 为安全评价；当 $\sum_{j=1}^{N}\beta_{ij}<1$ 时，称 $S_{(i)}$ 为不安全评价。本文中一律认定为安全评价。如果安全度 S_i 完全近似于第 j 个安全表达，则 β_{ij} 为 1 并且其他的 β_{ij} 为 0。该评价的置信度随着安全性分析人员主观评价的次数和人数会发生变化，随着数学方法的利用而更加减小软件安全性估计值（安全等级）与实际值的差别。

按照系统分解的思想，整个系统由多个子系统构成，而子系统由多个构件组成。因此，可先描述单个构件的各种失效模式的安全度，然后评估该构件安全度，再通过证据合成推理的方法，计算出整个软件的安全度，以及给出对应于各个安全等级的概率表示，实现对子系统进而整个系统的安全性评估。

3 构件化软件系统的安全性分析

前面已经简单提到，基于构件的软件系统可以利用分层的评估过程对其进行安全性分析和评估。这种评估方法基于处理不确定度量问题的 Dempster-Shafer(D-S)理论。该理论随着证据的不断累计可以对假设集的不确定度问题做出良好的模拟和处理[6]。在软件系统中，单个构件的安全度 $S_{(i)}$ 的计算结果可以理解为一种假设，并且具有置信度 β_{ij}。多个构件的安全度可以基于不同的证据而加以合成和推理，最终计算出子系统和整个系统的安全度。

3.1 D-S 理论

D-S 证据理论由 Dempster 于 1967 年提出，后来由 Shafer 对其进行推广和完善。该理论是一种不确定性的证据推理方法，允许把整个问题和证据分解为若干个子问题和子证据。在对其子问题做出相应处理后，利用 Dempster 合成法则，可以得到整个问题的解。D-S 证据理论能较好地处理具有模糊和不确定信息的合成问题，通过它可以对软件各个构件的安全性进行证据数据融合，可以得到更好的量测结果。

定义 3[3] 给定某命题的各种独立的可能假设构成的有限集合 $\Theta=(a_1,a_2,\cdots,a_N)$，则可称 Θ 为该命题的一个识别框架（Frame of discernment）。Θ 中的所有可能集合用幂集合 2^{Θ} 来表示。

定义 4[4] 给定某命题 Θ，若有函数 $m:2^{\Theta}\to[0,1]$，且满足 $m(\phi)=0$，$\sum_{X\subseteq\Theta}m(X)=1$，可称 m 为 Θ 上的基本概率分配函数(BPA)；而 $\forall X\subseteq\Theta$，$m(X)$ 称为 X 的基本置信度或者 Mass 函数 $m(X)$，代表的是证据支持命题 X 发生的程度，为该命题不确定性的载体。假若 $X\subseteq\Theta$ 并且 $m(X)>0$，则称 X 为证据的焦元(Focal Element)，所有焦元的集合称之为核(Core)。

定理 1[4] 给定两个不同证据的 mass 函数 m_1，m_2，根据 D-S 合成规则，则可以得到复合 mass 函数，即，

$$m(A)=m_1\oplus m_2=\begin{cases}0, & A=\phi\\ \left(\sum_{B\cap C=A}m_1(B)m_2(C)\right)/1-k, & A\neq\phi\end{cases}$$

式中，$k=\sum_{B\cap C=\phi}m_1(B)m_2(C)$，$k<1$。

上述运算称之为 m_1 和 m_2 的“直积”，反映证据的联合作用。只要证据之间不冲突，就可以利用该合成法则计算出一个联合的置信度函数。k 代表证据间的冲突程度，k 越大，说明证据间的冲突越大。若 $k=1$，则认为 m_1 和 m_2 完全矛盾，此时公式无法使用。系数 $1/1-k$ 称之为归一化因子，其作用是在证据合成时避免将非 0 的概率赋给空集 ϕ，完全抛弃证据间的冲突性并把所有与其相关的概率分配于空集。

3.2 基于 D-S 理论的构件化软件安全度综合

以下介绍应用 D-S 证据理论对软件系统安全性分析的不确定度问题进行处理的算法。

3.2.1 软件安全度的综合算法

假设第 k 个构件有 L_k 个失效模式，定义 c_k 为构件 k，而 e_{ki} 为构件 k 所相关的失效模式 i，则该构件 k 的失效模式的集合可以定义为：

$$E_k=(e_{k1},\cdots,e_{ki},\cdots,e_{kL_i})$$

令 λ_{ki} 为构件 k 的失效模式 i 在软件评估中的相对权重，并且 $0\leqslant\lambda_{ki}\leqslant 1$。该参数用来将给定的单个因素的评估置信度转化为基本概率分配函数表示，可用如下公式表示：

$$\lambda_{ki}=\alpha_k\frac{\zeta_k^i}{\zeta_k^I}$$

$$\prod_{j=1}^{4}(1-\alpha_k\frac{\zeta_k^i}{\zeta_k^I})\leqslant\delta$$

式中，δ 为最终某一种安全性评价的置信度值，而 ζ_k^i 代表第 j 个失效模式的相对权重；ζ_k^I 代表第 k 个构件中所有失效模式权重的最大值；α_k 代表该构件中最重要因素所代表的重要性的优先系数。

假设 $m_{ki}^i=m(H_j/e_{ki})$（$m_{ki}^i\leqslant 1$）为实值，代表了一个基本分配函数，这个函数用以表示获得的第 i 个失效模式的安全度评估值支持所做假设（第 k 个构件的安全度符合假设 H_j）的程度。然后，m_{ki}^i 可用下式计算：

$$m_{ki}^i=\lambda_{ki}\beta_{ij}$$

式中，β_{ij} 的值已经由第 k 个构件予以给定。

因为 $0\leqslant\lambda_{ki}\leqslant 1$，并且 $\sum_{j=1}^{N}\beta_{ij}=1$，则有 $\sum_{j=1}^{N}m_{ki}^i\leqslant 1$。设 $m_{ki}^H=m(H/e_{ki})$ 是假设 H 的基本概率分配函数，则有 $m_{ki}^H=1-\sum_{j=1}^{N}m_{ki}^i$。那么，构件 c_k 的安全度相关的失效模式 E_k 的基本概率分配矩阵 $m(c_k/E_k)$ 可以用下式计算：

$$M(c_k/E_k)=\begin{matrix}\\ \\ \\ \\ \\ \end{matrix}\begin{matrix}\begin{pmatrix}m_{k1}^1 & \cdots & m_{k1}^i & \cdots & m_{k1}^N & m_{k1}^H\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ m_{ki}^1 & \cdots & m_{ki}^i & \cdots & m_{ki}^N & m_{ki}^H\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ m_{kL_k}^1 & \cdots & m_{kL_k}^i & \cdots & m_{kL_k}^N & m_{kL_k}^H\end{pmatrix}\begin{matrix}\{e_{k1}\}\\ \cdots\\ \{e_{ki}\}\\ \cdots\\ \{e_{kL_k}\}\end{matrix}\end{matrix}$$

假设 $m_{c_k}^i$ 是第 k 个构件的安全度被评估为 H_j 的置信度，那么，$m_{c_k}^i$ 就可以通过综合在 $M(c_k/E_k)$ 中列出的基本分配函数，并通过下面描述的证据推理的综合算法来加以综合。

假设 ψ 为安全度表达 H 的子集，即 $\psi\subseteq H$。分别定义构件 k 的失效模式集合 E_k 的子集 $e_{l_{k(i)}}$，组合概率分配函数 $m_{k(i)}^{\psi}$ 如下：

$$e_{l_{k(i)}}=\{e_k^1,\cdots e_k^i\},1\leqslant i\leqslant L_k$$

$m^{\psi}_{k(i)}=m(\psi_{l_k(i)})$

则该分层综合算法[10]的步骤如下：

$$\{H_j\}:m^j_{l_k(i+1)}=K_{l_k(i+1)}\{m^j_{k(i)}\,m^j_{k,i+1}+m^j_{k(i)}\,m^H_{k,i+1}+m^H_{k(i)}\,m^j_{k,i+1}\},j=1,\cdots,N$$

$$\{H\}:m^H_{l_k(i+1)}=K_{l_k(i+1)}\,m^H_{k(i)}\,m^H_{k,i+1}$$

$$K_{l_k(i+1)}=[1-\sum_{\tau=1}^{N}\sum_{\substack{j=1\\ j\neq\tau}}^{N}m^{\tau}_{l_k(i)}\,m^j_{k,i+1}]^{-1}$$

式中，$i=1,\cdots,L_k-1$。

3.2.2 安全度计算的逐层综合

在上面所示的算法中，可以很简单地证明如下结论：

对于任意的 $\psi\subseteq H$，$m^{\psi}_{k(i)}$ 都是 ψ 的总体概率分配函数，由 E_k 以及 $m^{\psi}_{l_k(L_k)}=0$ 所确定。因此，某软件子系统第 k 个构件的安全度就会以一定的置信度($m^j_{c_k}$)而被评估为 H_j。这样的评估可以通过综合给定相关失效模式的安全度加以实现，用下式表示：

$$S_{(c_k)}=\{(m^j_{c_k},H_j),j=1,\cdots,N\}$$

同样地，可以计算得到每一个构件的安全度评估值。进一步的问题是提出对该构件所在的子系统的安全度进行综合。假设在软件系统第 l 个子系统中存在着 L_i 个构件，则子系统 l 的构件集合可以定义为：

$$F_l=(c_{l1},\cdots,c_{lk},\cdots,c_{lL_i})$$

某软件子系统第 k 个构件的安全度以一定的置信度($m^j_{c_k}$)而被评估为 H_j，那么这些构件的评估结果就可以认为是第 l 个子系统的安全度被评估为 $H_j(j=1,\cdots,N)$ 的证据，通过证据推理理论就可以得出第 l 个子系统的安全度。问题转化为如何从 $m^j_{c_k}$ 获取和计算 $m^j_{s_l}(j=1,\cdots,N;k=1,\cdots,L_i)$。要解决上述问题，只需要将 c_{lk} 视之为 e_{ki}，而将 $m^j_{c_k}$ 和 $m^j_{s_l}$ 当作 β_{ij} 和 $m^j_{e_k}$ 即可。则第 l 个软件子系统的安全性可以用下式加以评估：

$$S_{(s_l)}=\{(m^j_{s_l},H_j),j=1,\cdots,N\}$$

同样的道理，将 m^j 作为整个系统安全度评估为 $H_j(j=1,\cdots,N)$ 时的置信度，其值可以通过 $m^j_{s_l}(j=1,\cdots,N;l=1,2,\cdots,s_l)$ 给出，式中的 s_l 是整个软件系统的子系统总个数，并使用证据推理的算法加以计算得出。由此，最终整个系统的安全度为：

$$S_{(s)}=\{(m^j,H_j),j=1,\cdots,N\}$$

计算出系统总体的安全度之后，可将其映射到 DO-178B 所规定的安全性表达(5 种软件等级)中，最终即可实现对软件系统的安全性评估。

4 实例分析

以某三余度无人机飞控系统[15]为实例，其系统架构图如图 2 所示。

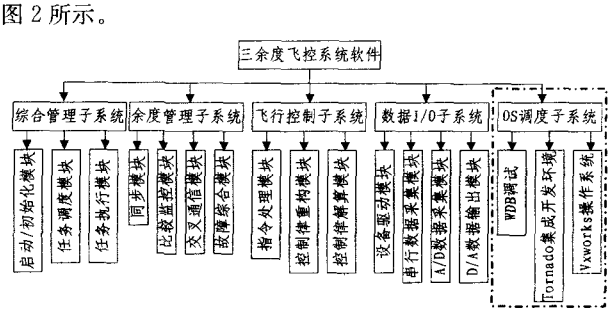

图 2 某三余度无人机飞控系统架构

该飞控系统由 5 个子系统构成，每一子系统又由数个模块构成。因此，对于该软件系统的安全性分析可以从最底层的模块开始，由其失效模式的各种参数推算出模块的安全度，从系统架构级别向上综合，最终计算出整个系统的安全度。对该例中的各个模块按照图中顺序加以编号，并将 OS 调度子系统的安全度计算略去。

(1)综合管理子系统

按照主观评价的专家评判法进行该子系统的安全度评估，可以得到初始化启动模块的各项模糊参数数据：

$L_{11}=\{1/0,2/0,\cdots,6/0.75,7/1.0\}$

$C_{11}=\{1/0,2/0.25,3/1,4/0.75,5/0,6/0,7/0\}$

$E_{11}=\{1/0,2/0.25,3/0.75,4/0,5/0,6/0,7/0\}$

根据定义 2 中的模糊集运算法则，计算得到初始化启动模块安全度为：

$S_{11}=\{1/0,2/0.25,3/0.75,4/0,5/0,6/0,7/0\}$

进一步计算得到该模块的安全性等级：

$S_{(s_{11})}=\{(0.1150,'Catastrophic'),(0.1222,'Hazardous'),(0.3403,'Major'),(0.1819,'Minor'),(0.2406,'Noeffect'),\}$

同理，可以得到其余模块的安全度分别为：

$S_{(s_{12})}=\{(0.1378,'Catastrophic'),(0.4785,'Hazardous'),(0.1473,'Major'),(0.1213,'Minor'),(0.1151,'Noeffect'),$

$S_{(s_{13})}=\{(0.1195,'Catastrophic'),(0.5740,'Hazardous'),(0.1277,'Major'),(0.1557,'Minor'),(0.0231,'Noeffect'),$

通过综合算法，计算可得：

$S_1=\{(0.0412,'Catastrophic'),(0.7307,'Hazardous'),(0.1394,'Major'),(0.0748,'Minor'),(0.0139,'Noeffect')\}$

(2)余度管理子系统

按照主观专家评判法给定数据可计算出：

$S_2=\{(0.1342,'Catastrophic'),(0.4516,'Hazardous'),(0.2103,'Major'),(0.1623,'Minor'),(0.0416,'Noeffect')\}$

(3)飞行控制子系统

$S_3=\{(0.1219,'Catastrophic'),(0.4375,'Hazardous'),(0.2028,'Major'),(0.1027,'Minor'),(0.1351,'Noeffect')\}$

(4)数据 I/O 子系统

$S_4=\{(0.2543,'Catastrophic'),(0.3875,'Hazardous'),(0.1241,'Major'),(0.1273,'Minor'),(0.1068,'Noeffect')\}$

最终的软件系统的安全度为：

$S=\{(0.0030,'Catastrophic'),(0.9813,'Hazardous'),(0.0129,'Major'),(0.0028,'Minor'),(0.0000,'Noeffect')\}$

由此即可评定出该软件系统的安全等级为 B 级，其置信度为 98.13%，可以提供决策参考。

结束语 复杂软件的安全性评估相关因素众多，难以评估。本文首先对模糊语言表述进行数学化，基于系统分层评估的主观评价方法，通过 D-S 证据理论予以计算综合，利用数

(下转第 161 页)

簇误判为离群簇，误报率相应也就越高。图3中(c)表明每阶段簇个数受Q变化影响不大。随着数据的不断流入，生成的簇个数整体呈上升趋势。

结束语 数据流离群检测不同于静态数据集的离群检测，因检测对象具有动态性、不可复读性、数据量大等特点而成为离群检测的一个难点，引起了许多研究人员的关注。本文探讨了混合属性数据流离群检测问题，提出了基于衰减模型的聚类特征结构，用以近似估计数据流的分布状况。同时提出了数据流中簇的离群因子定义，通过计算特定簇的离群因子，可得到离群簇集合，作为结果提交给用户，合理区分了离群簇与数据进化初始阶段，改进了数据流离群检测的质量。实验结果表明，本文提出的算法是有效的，检测性能优于其它相关算法。进一步提高算法的实现效率，将其扩展到更一般的数据流模型，以及与数据流中其他相关的数据挖掘算法进行结合，是下一步的研究方向。

参 考 文 献

[1] Aggarwal C C, Han Jia-wei, Wang Jian-yong, et al. A Framework for Clustering Evolving Data Streams[C]// Proceedings of the 29th International Conference on Very Large Data Bases. Berlin, 2003: 81-92

[2] Aggarwal C C, Han Jia-wei, Wang Jian-yong, et al. A Framework for Projected Clustering of High Dimensional Data Streams[C]// Proceedings of the 30th International Conference on Very Large Data Bases. Toronto, 2004: 852-863

[3] Cao Feng, Ester M, Qian Wei-ning, et al. Density-based Clustering over an Evolving Data Stream with Noise[C]// Proceedings of the 6th SIAM International Conference on Data Mining. Bethesda, 2006: 326-337

[4] 倪巍伟，陆介平，陈耿，等. 基于k均值分区的数据流离群点检测算法[J]. 计算机研究与发展，2006，43 (9)：1639-1643

[5] 杨宜东，孙志挥，朱玉全，等. 基于动态网格的数据流离群点快速检测算法[J]. 软件学报，2006，17(8)：1796-1803

[6] 俞研，郭山清，黄皓. 基于数据流的异常入侵检测[J]. 计算机科学，2007，34(5)：66-71

[7] 周晓云，孙志挥，张柏礼，等. 高维类别属性数据流离群点快速检测算法[J]. 软件学报，2007，18(4)：933-942

[8] Jiang Sheng-Yi, Song Xiao-Yu. A Clustering-based Method for Unsupervised Intrusion Detections[J]. Pattern Recognition Letters, 2006, 27(5): 802-810

[9] 杨春宇，周杰. 一种混合属性数据流聚类算法[J]. 计算机学报，2007，30(8)：1364-1371

[10] He Zeng-you, Xu Xiao-fei, Huang Zhe-xue, et al. FP-Outlier: Frequent Pattern Based Outlier Detection[J]. Computer Science and Information System, 2005, 2(1): 103-118

[11] Aggarwal C, Yu P. An Effective and Efficient Algorithm for High-dimensional Outlier Detection[J]. The VLDB Journal, 2005, 14(2): 211-221

(上接第126页)

学方法尽量减少安全性分析的随意性和不确定性，实现了软件安全性的分析和评估，并给出软件安全等级的置信度水平。随着分析人员的估计次数和人数的增加，置信度水平会更接近于实际的软件等级。

但是，该方法未对软件安全性的保障技术以及构件失效概率等问题进行探讨。下一步需要结合具体应用背景，对复杂软件的安全性分析做更深入研究，并对计算模型和综合算法做一定程度的优化和改进。

参 考 文 献

[1] John C K. Safety-Critical System: Challenges and Directions [C]// Proceedings of the 24th International Conference on Software Engineering. May 2002: 547-550

[2] Wang J. A Subjective Methodology for Safety Analysis of Safety Requirements Specifications [J]. IEEE Transactions on Fuzzy Systems, 1997, 5(3): 418-430

[3] Dempster A P. A generalization of Bayesian inference(with discussion)[J]. Journal of the Royal Statistical Society Series B, 1968, 30(2): 205-247

[4] Shafer G. A Mathematical Theory of Evidence[M]. Princeton: Princeton University Press, 1976

[5] Atkinson C, Bunse C, Gross H-G, et al. Component-based Software Development for Embedded Systems[M]. Berlin Heidelberg, Germany: Springer-Verlag, 2005

[6] Schmucker K J. Fuzzy sets, Natural Language Computations and Risk Analysis [M]. Rockville, MD: Computer Science Press, 1984

[7] DO-178B. Software Considerations in Airborne Systems and Equipment Certification[S]. RTCA/EUROCAE, December 1992

[8] MIL-STD-882C. System Safety Program Requirements[S]. Department of Defense. USA Military Standard, 1993

[9] Karwowski M. Potential Applications of Fuzzy Sets in Industrial Safety Engineering [J]. Fuzzy Sets and Systems, 1986, 19: 105-120

[10] Liu J, Yang J B, Wang J, et al. Safety analysis and synthesis using fuzzy rule-based evidential reasoning approach [C] // the 2003 UK Workshop on Computational Intelligence. University of Bristol, September 2003

[11] Zadeh L A. Fuzzy Sets, Information and Control[M]. 1965: 338-353

[12] Herrera F, Martinez L. A 2-tuple fuzzy linguistic representation model for computing with words [J]. IEEE Transactions on Fuzzy Systems, 2000, 8(6)

[13] Bowles J B, Pelaez C E. Fuzzy logic Prioritization of failures in a system failure mode, effects and criticality analysis[J]. Reliability Engineering and System Safety, 1995, 50: 203-213

[14] Anderson L. The theory of possibility and fuzzy sets: new ideas for risk analysis and decision making[M]. Swedish Council for Building Research, 1988: 165-167

[15] Zimmerman H J. Fuzzy Set Theory and Its Application[M]. Norwell, MA: Kluwer, 1991

[16] 张锦. 三余度飞控计算机系统软件的研究与设计[D]. 西安：西北工业大学，2006：7-16