

# 面向动态演化的复杂网构软件系统可靠性分析方法

宋敏<sup>1</sup>, 鞠阳<sup>2</sup>, 韦正现<sup>2,3\*</sup>

1. 北京外国语大学信息技术中心, 北京 100089
2. 中国船舶工业系统工程研究院, 北京 100094
3. 青岛海洋科学与技术试点国家实验室, 青岛 2660011

**摘要** 网构软件是Internet上各种软件实体以各种协同方式实现跨网络的互连、互通和协作的软件系统, 动态演化是其基本特征, 演化过程中会导致软件系统失效, 面向动态演化的网构软件可靠性分析成为新的技术难题。首先分析构件接口交互之间的协议关系和网构软件动态演化错误传播可达性, 提出通过错误传播可达图来界定构件动态演化错误的影响范围, 并建立错误传播可达图生成过程模型; 其次, 为有效解决Markov状态空间“爆炸”的问题, 利用错误传播可达图来生成Markov状态转移图, 提出基于Markov的网构软件动态演化可靠性分析方法; 最后通过案例分析来验证该方法的有效性。

**关键词** 网构软件; 动态演化; 系统可靠性; 错误传播

网构软件(internetware)是一种基于Internet平台的新型软件形态, 是Internet上各种软件实体(构件、服务或Agent等, 统称为构件)在开放的环境下以各种协同方式实现跨网络的互连、互通和协作的软件联盟, 在复杂、开放和动态的环境下为用户提供高效服务。它具有自主适应性、协同性、反应性、演化性和多态性等形态特征, 为用户提供7×24 h、不断演化更新、智能化的服务, 同时能感知环境的动态变化, 并按照功能指标、性能指标和可信性指标等进行在线动态演化<sup>[1-2]</sup>。动态演化是网构软件的一个基本特征, 是软件运行时刻、在没有(或尽量少)用户干预的前提下进行演化, 要求在不中断网构软件所提供服务前提下, 将软件实体的运

行实例从当前流程模式(源模式)动态地迁移到新流程模式下(目标模式)继续执行, 在网构软件动态演化过程中, 可能会出现新的错误(称为动态演化错误), 这些错误会随着构件之间的接口交互进行传播, 导致网构软件失效<sup>[3-4]</sup>, 使网构软件的可靠性发生改变<sup>[5-6]</sup>。

基于Markov理论进行网构软件可靠性分析是目前研究的热点之一<sup>[7-10]</sup>, 在采用Markov理论对网构软件可靠性进行分析时, 当状态空间较大时会连带产生“爆炸”现象。针对网构软件动态演化的特征, 首先从组成网构软件的构件接口交互之间的协议关系角度, 分析不同协议关系下网构软件动态演化错误传播的可达特性, 通过错误传播可达图来界定构件动态演化错误的

收稿日期: 2018-12-20, 修回日期: 2019-01-21

基金项目: 国家自然科学基金项目(61502037, 61772152); 国防科技创新特区项目(18-H863-05-ZT-001-007-02)

作者简介: 宋敏, 工程师, 研究方向为软件可靠性、数据库和软件体系结构, 电子信箱: songmin@bfsu.edu.cn; 韦正现(通信作者), 研究员, 研究方向系统工程、体系工程和人工智能等, 电子信箱: weizhengxian@sina.com

引用格式: 宋敏, 鞠阳, 韦正现. 面向动态演化的复杂网构软件系统可靠性分析方法[J]. 科技导报, 2019, 37(13): 98-103; doi: 10.3981/j.issn.1000-7857.2019.13.015

影响范围,建立了错误传播可达图生成过程模型;然后,基于Markov模型可以动态反映系统从启动到经过一系列的中间状态再到失效状态的转移过程,同时为了有效解决Markov状态空间“爆炸”的问题,利用错误传播可达图生成Markov状态转移图,提出了基于Markov的网构软件动态演化可靠性分析方法,为有效分析网构软件动态演化下的系统可靠性提供了方法支持;最后,通过案例分析来验证该方法的有效性。

### 1 构件交互协议关系

构件C由接口规约和内部规约组成。接口规约可以分为两类,一类是服务接口(provided/service/public interface) $I_p$ ,它是构件C提供功能(服务)的接口;另一类是请求接口(required/entry interface) $I_e$ ,它是构件C需要的功能(服务)接口。构件如图1所示。

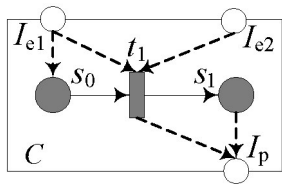


图1 构件内部示意

Fig. 1 Internal relationship of component

在图1中,只有请求接口 $I_{e1}$ 或 $I_{e2}$ 得到满足,构件C才能实现功能或提供服务并通过接口 $I_p$ 对外提供功能,如果 $I_{e1}$ 和 $I_{e2}$ 具有时序、逻辑等关系,才能导致 $I_p$ ,则对于 $I_p$ 接口来说, $I_{e1}$ 和 $I_{e2}$ 就具有了协议关系,主要包括以下关系。

- 1) 对应关系:构件C满足请求接口 $I_{e1}$ 才可以触发或提供服务接口 $I_p$ ,记为 $\theta I_{e1} \rightarrow I_p$ ,称为 $\theta$ 关系。
- 2) 选择关系:构件C需要满足请求接口 $I_{e1}$ 和 $I_{e2}$ 的其中之一或全部,才能触发或提供服务接口 $I_p$ ,记为 $I_{e1} \sqcup I_{e2} \rightarrow I_p$ ,称为 $\sqcup$ 关系。
- 3) 互斥关系:构件C在请求接口 $I_{e1}$ 和 $I_{e2}$ 之间, $I_{e1}$ 被导入则 $I_{e2}$ 不能被导入,或者 $I_{e2}$ 被导入则 $I_{e1}$ 不能被导入,才触发或提供服务接口 $I_p$ ,记为 $I_{e1} \oplus I_{e2} \rightarrow I_p$ ,称为 $\oplus$ 关系。
- 4) 并发关系:构件C需要请求接口 $I_{e1}$ 和 $I_{e2}$ 都同时得到满足,才能触发或提供服务接口 $I_p$ ,记为 $I_{e1} \sqcap I_{e2} \rightarrow I_p$ ,称为 $\sqcap$ 关系。
- 5) 顺序关系:构件C只有请求接口 $I_{e1}$ 被导入并处

理完成以后, $I_{e2}$ 才能被导入并进行处理,而触发或提供服务接口 $I_p$ ,记为 $I_{e1} \infty I_{e2} \rightarrow p$ ,称为 $\infty$ 关系。

- 6) 产生关系:构件C的服务端口 $I_p$ 不需要任何请求端口就可以提供或引发,记为 $0\tau \rightarrow I_p$ ,称为 $\tau$ 关系;结束关系:构件C的请求端口 $I_e$ 都不会引发或提供任何服务端口,记为 $\sigma I_e \rightarrow 0$ ,称为 $\sigma$ 关系。

构件要完成一定的功能或提供相应的服务,这是由构件内部功能域的活动和状态来实现,形成构件内部规约。在图1中,请求接口 $I_e$ 可以作为构件内部状态或活动的输入,同时服务接口 $I_p$ 可以由构件内部活动或状态中提供。同样在不同的构件接口交互之间,当一个构件的请求接口 $I_e$ ,需要由其他构件的一个或多个服务接口触发或提供时,则构件之间的接口交互也就具有了协议关系。

### 2 网构软件动态演化错误传播可达性分析

Uokhale等<sup>[11]</sup>指出,在对网构软件进行可靠性分析时,需要获得2方面信息:(1) 构件 $C_i$ 的内部错误行为;(2) 构件间接口交互 $I_j$ 的错误行为。由于网构软件的某个构件发生动态演化时(演化源),可能会导致该构件出现新的错误,这些错误会随构件之间接口交互进行传播,导致网构软件服务出现故障和失效。因此,在分析面向动态演化的网构软件可靠性时,还需要获得3个方面的信息:(1) 演化操作导致构件 $C_i$ 的错误行为;(2) 构件间接口交互变化的错误行为;(3) 演化源状态变化通过接口交互传播,导致的错误行为。

上述的这5种错误事件都可能导致网构软件失效。网构软件中错误事件在不同构件之间通过接口交互传播,构件之间的接口交互关系分为直接交互关系和间接交互关系<sup>[12]</sup>,构件之间的错误传播关系也分为直接传播关系和间接传播关系两类。不同构件之间的接口交互协议关系,对于错误传播可达性的影响也有所不同,以下分别进行讨论。

#### 2.1 具有 $\sqcap, \infty, \sqcup, \oplus$ 关系的构件演化的错误传播情况

具有 $\sqcap, \infty, \sqcup, \oplus$ 关系的构件修改,如图2所示。图2中如果 $C_3$ 产生了修改,并且出现了错误,则错误传播可达性的情况如下。

- 1) 如果有 $I_{e1} @ I_{e2} @ I_{e3} \rightarrow I_p, @ \in \{\sqcap, \infty\}$ ,则构件 $C_3$ 的错误会传播至构件 $C_4$ ,或者会引发或触发 $C_4$ 出现新错误,

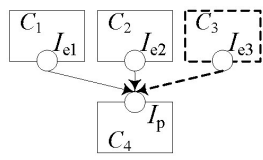


图2 具有 $\cap, \infty, \sqcup, \oplus$ 的错误传播

Fig. 2 Error propagation on  $\cap, \infty, \sqcup, \oplus$

即对于构件  $C_4$  而言,  $C_3$  的错误传播是可达的。

2)  $I_{e1} @ I_{e2} @ I_{e3} \rightarrow I_p, @ \in \{\sqcup, \oplus\}$ , 则构件  $C_3$  的错误可能不会传播至构件  $C_4$ , 或者不会引发或触发  $C_4$  出现新错误, 即对于构件  $C_4$  而言,  $C_3$  的错误传播是不可达的。

3) 如果  $I_{e1}, I_{e2}$  和  $I_{e3}$  到  $I_p$  的协议关系是前二种情况的组合, 则需要综合分析错误传播是否可达。

### 2.2 具有 $\theta$ 关系的构件演化的错误传播情况

对具有  $\theta$  关系的两个接口之间交互的构件进行修改时(图3), 如果有  $\theta I_{e2} \rightarrow I_{p3}$ , 修改了  $C_2$ , 并且出现了错误事件, 则构件  $C_2$  的错误会传播至构件  $C_3$ , 或者会引发或触发  $C_3$  出现新错误, 即对于构件  $C_3$  而言,  $C_2$  的错误传播是可达的。

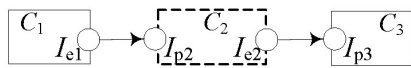


图3 具有  $\theta$  的错误传播

Fig. 3 Error propagation on  $\theta$

构件之间错误传播关系具有方向性, 因此, 网构软件的错误传播可以表示成一个有向图  $SA:=G(C, T)$ ,  $C$  表示构件集合,  $T$  表示错误传播关系集合, 另外这里假设该有向图不存在环。因此需要建立错误传播可达图来分析构件发生动态演化错误时的影响范围。错误传播可达图生成过程如图4所示。

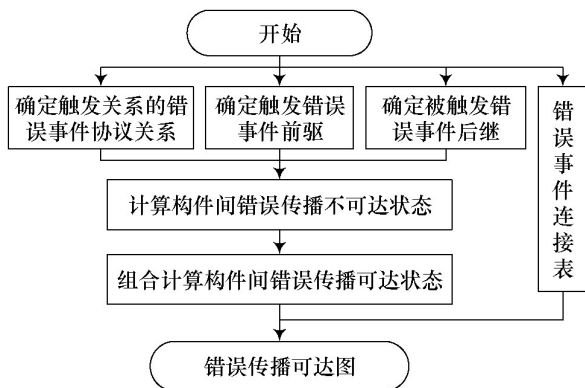


图4 错误传播可达图生成过程模型

Fig. 4 Generation model of error propagation reachability graph

在错误传播可达图生成过程中, 首先, 通过识别有向图中构件内部时序对错误事件进行排序, 获得构件间不可同时到达的错误事件组; 其次, 对构件间的可到达错误事件建立笛卡尔积, 获得网构软件系统的所有可达错误事件组; 最后, 根据错误事件连接表获得系统错误传播可达图。错误传播可达图为面向动态演化的网构软件失效提供了定性分析方法, 需进一步定量分析面向动态演化的网构软件可靠性, 要求将错误传播可达图转换成 Markov 状态转移图, 并分析计算网构软件的可靠性。

### 3 面向动态演化的网构软件可靠性分析

Markov 模型可以动态地反映系统从启动到经过一系列的中间状态再到失效状态的转移过程, 既能反映网构软件各构件间的静态关系, 又能反映网构软件整体的动态变化, 用于分析计算网构软件动态演化的可靠性来说具有较高精度。

#### 3.1 Markov 过程模型

Markov 过程是一种特殊的随机过程, 设  $\{X(t), t \geq 0\}$  是取值在  $E=\{0, 1, 2, \dots\}$  或  $E=\{0, 1, 2, \dots, N\}$  离散状态空间的一个随机过程, 若对任意自然数  $n$  及任意  $n$  个时刻点  $0 \leq t_1 < \dots < t_n$ , 均有

$$P\{X(t_n)=i_n | X(t_1)=i_1, X(t_2)=i_2, \dots, X(t_{n-1})=i_{n-1}\} = P\{X(t_{n-1})=i_{n-1}\}, \quad i_1, i_2, \dots, i_n \in E \quad (1)$$

则称  $\{X(t), t \geq 0\}$  为离散状态空间  $E$  上的连续时间 Markov 过程。该公式可解释为: 当给定时刻  $t_{n-1}$  过程  $\{X(t), t \geq 0\}$  处于某个状态的情况下, 过程在  $t_{n-1}$  以后发展的概率规律与过程在  $t_{n-1}$  以前的历史无关。简单来讲: 当给定过程现在所处的状态, 则过程将来发展的概率规律与过程的历史无关。

若对任意  $t, u \geq 0$ , 均有

$$P\{x(t+u)=j | X(u)=i\} = p_{ij}(t) \quad (i, j \in E)。$$

与  $u$  无关, 则称 Markov 过程  $\{X(t), t \geq 0\}$  是时齐的。对固定的  $i, j \in E$ , 函数  $p_{ij}(t)$  称为转移概率函数。矩阵  $P(t) = [p_{ij}(t)]$  称为转移概率矩阵。该公式则表示 Markov 过程的转移概率仅与时差  $t$  有关, 而与起始时刻的位置  $u$  无关。

假定 Markov 过程  $\{X(t), t \geq 0\}$  的转移概率函数  $p_{ij}(t)$

满足

$$\lim_{t \rightarrow 0} p_{ij}(t) = \delta_{ij} = \begin{cases} 1, & i=j \\ 0, & i \neq j \end{cases} \quad (2)$$

对于转移概率函数,显然有以下性质

$$\begin{cases} p_{ij}(t) \geq 0 \\ \sum_{j \in E} p_{ij}(t) = 1 \\ \sum_{k \in E} p_{ik}(u) p_{kj}(v) = p_{ij}(u+v) \end{cases} \quad (3)$$

若令  $p_j(t) = \{X(t)=j\}$ ,  $j \in E$ , 表示时刻  $t$  系统处于状态  $j$  的概率,则有

$$p_j(t) = \sum_{k \in E} p_k(0) p_{kj}(t) \quad (4)$$

### 3.2 基于Markov的网构软件动态演化可靠性分析方法

Markov 过程为网构软件可靠性的求解提供了理论基础。采用 Markov 模型进行网构软件动态演化的可靠性计算,需要建立 Markov 状态转移图。在错误传播可达图中,若错误传播可达图中包含导致网构软件失效的活动与状态,则将该状态作为网构软件失效状态,这样就可以利用错误传播可达图来生成 Markov 状态转移图。下面开始基于 Markov 的网构软件动态演化可靠性分析过程。

1) 根据状态转移图,求出转移概率矩阵  $A = [a_{ij}]$ 。

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} \quad (5)$$

其中,  $a_{ii} = -\sum_{i \neq j} a_{ij}$ 。对于已定义的过程,求出  $p_{ij}(\Delta t) = a_{ij} \Delta t + o(\Delta t)$ ,  $i \neq j, i, j \in E$ 。

如果网构软件系统为不可修系统,则转移概率矩阵  $A$ ,进一步可以得到:  $a_{ij} = 0, i > j$  且  $a_{ij} = -\sum_{j=i+1}^K a_{ij}$ 。

2) 求  $p_j(t) = P\{X(t)=j\}$ ,  $j \in E$ 。解微分方程组

$$[p'_0(t), p'_1(t), \dots, p'_n(t)] = [p_0(t), p_1(t), \dots, p_n(t)]A \quad (6)$$

该模型的初始条件为

$$\begin{cases} p_0(0) = 1, \\ p_i(0) = 0, \end{cases} \quad i = 1, 2, \dots, n \quad (7)$$

具体解法可采用拉普拉斯变换,将上面的微分方程组转换为拉普拉斯方程组,解出线性方程组后,再作

反拉普拉斯变换求得  $p_n(t)$  的解析表达式,该值对应于网构软件系统失效发生的概率,即  $t$  时刻的失效概率。

3) 求网构软件系统可靠度。解如下方程组:

$$[p'_0(t), p'_1(t), \dots, p'_n(t)] = [p_0(t), p_1(t), \dots, p_n(t)]B \quad (8)$$

其中,  $B$  是转移概率矩阵  $A$  左上角  $K+1$  行、 $K+1$  列组成的子矩阵。则网构软件系统的可靠度为  $R(t) = \sum_{j \in W} p_j(t)$ 。

## 4 实例分析

为了验证所提方法的有效性,以校园网数字图书馆系统为案例进行试验分析,假设数字图书馆分为借还书构件、学生管理构件和图书管理构件 3 部分组成,这 3 个构件之间都有接口交互。案例分析过程中,假设条件如下。

1) 网构软件的各个构件发生动态演化错误的时间服从指数分布。

2) 所有的构件仅包含两种状态:正常状态、错误状态。初始时刻,系统完好,各构件均正常工作。

3) 在时间间隔内至多只有一个构件发生一次动态演化并产生一个动态演化错误。

当构件发生动态演化错误时,该错误以一定的概率在构件之间传递错误,错误传递以及导致的软件失效的 Markov 状态转换如图 5 所示。

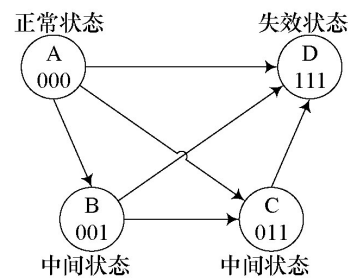


图 5 Markov 状态转换

Fig. 5 Markov state transition

假设图书馆系统软件中所有构件的失效时间服从参数为  $\lambda$  的指数分布,并且初始时刻系统中所有的构件均为正常,当某个构件发生动态演化错误时,其状态转移路径为:  $A \rightarrow D, A \rightarrow B \rightarrow D, A \rightarrow C \rightarrow D, A \rightarrow B \rightarrow C \rightarrow D$ 。Markov 对应的微分方程如下:

$$P'(t) = AP(t) \quad (9)$$

$$\text{其中, } P'(t) = \begin{bmatrix} P'_0(t) \\ P'_1(t) \\ P'_2(t) \\ P'_3(t) \end{bmatrix}, P(t) = \begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix},$$

$$A = \begin{bmatrix} -3\lambda & \lambda & \lambda & \lambda \\ 0 & -2\lambda & \lambda & \lambda \\ 0 & 0 & -\lambda & \lambda \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$P_i(t)$ 表示系统在时刻 $t$ 处于状态 $i$ 的概率, $P'_i(t)$ 为 $P_i(t)$ 的导数。以 $P(0)=(1, 0, 0, 0)^T$ 为初始值求解方程,得到的解析解为:

$$P(t) = (e^{-3\lambda t}, e^{-2\lambda t}, -e^{-3\lambda t}, e^{-\lambda t}, e^{-2\lambda t}, 1 - e^{-\lambda t})^T。$$

则系统在时刻 $t$ 的可靠度为: $R(t) = P_0(t) + P_1(t) + P_2(t) = e^{-\lambda t}$ 。其系统可靠性随时间的变化如图6所示。

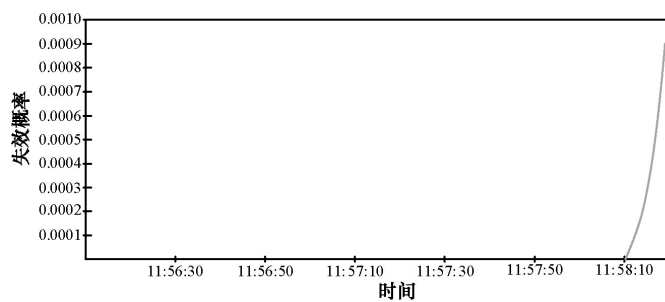


图6 系统失效概率的时间变化

Fig. 6 System Failure rate with time

图6中的系统失效随时间变化的概率图能够较为客观反映了该软件在发生动态演化错误时的可靠性情况。通过分析网构软件中各个构件的接口交互协议关系,以及动态演化错误传播可达性,描述错误发生的时间顺序和传递关系,同时通过错误传播可达图构建Markov状态转换能够降低状态的数量,这样更有效地对动态演化的网构软件可靠性进行分析。

## 5 结论

基于Markov理论进行网构软件可靠性分析是目前研究的热点之一。在采用Markov理论对网构软件可靠性进行分析时,当状态空间较大时会连带产生“爆炸”现象。当网构软件发生动态演化会出现新的动态演化错误,这些错误会随着构件之间的接口交互进行传播,

导致网构软件失效,增加了网构软件可靠性分析的难度。从复杂网构软件系统中构件接口交互之间具有不同的协议关系入手,分析网构软件动态演化错误传播可达性,并建立错误传播可达图生成过程模型;然后将错误传播可达图转换为Markov状态转移图,这样有利于降低Markov状态的数量,有效地解决Markov状态空间“爆炸”的问题;在此基础上,提出基于Markov的网构软件动态演化可靠性分析方法,为实现复杂网构软件系统动态演化的可靠性计算提供了技术基础。

## 参考文献(References)

- [1] 杨芙清. 软件工程技术发展思索[J]. 软件学报, 2005, 16(1): 1-7.  
Yang Fuqing. Thinking on the development of software engineering technology[J]. Journal of Software, 2005, 16(1): 1-7.
- [2] 吕建, 马晓星, 陶先平, 等. 网构软件的研究与进展[J]. 中国科学(信息科学), 2006, 36(10): 1037-1080.  
Lü Jian, Ma Xiaoxing, Tao Xianping, et al. Research and development of internetware[J]. Scientia Sinica Informationis, 2006, 36(10): 1037-1080.
- [3] 宋巍, 马晓星, 胡昊, 等. 过程感知信息系统中过程的动态演化[J]. 软件学报, 2011, 22(3): 417-438.  
Song Wei, Ma Xiaoxing, Hu Hao, et al. Dynamic evolution of processes in process-aware information system[J]. Journal of Software, 2011, 22(3): 417-438.
- [4] 宋敏, 韦正现, 印桂生. 面向数据流的网构软件服务动态演化分析[J]. 软件学报, 2013, 24(12): 2797-2813.  
Song Min, Wei Zhengxian, Yin Guisheng. Evolution analysis of data flow oriented internetware service[J]. Journal of Software, 2013, 24(12): 2797-2813.
- [5] Papazoglou M P. The challenges of service evolution[C]//International Conference on Advanced Information Systems Engineering. Berlin: Springer, 2008.
- [6] Andrikopoulos V, Benbernou S, Papazoglou M P. Managing the evolution of service specifications[C]//International Conference on Advanced Information Systems Engineering. Berlin: Springer-Verlag, 2008: 359-374.
- [7] 张策, 崔刚, 刘宏伟, 等. 构件软件可靠性过程技术[J]. 计算机学报, 2014, 37(8): 749-760.  
Zhang Ce, Cui Gang, Liu Hongwei, et al. Component-based software reliability process technologies[J]. Chinese Journal of Computers, 2014, 37(8): 749-760.
- [8] Hsu C J, Huang C Y. An adaptive reliability analysis using path testing for complex component based software systems[J]. IEEE Transactions on Reliability, 2011, 60(1): 158-170.

- [9] Pietrantuono K, Russo S, Trivedi K S. Software reliability and testing time allocation: An architecture-based approach[J]. IEEE Transactions on Software Engineering, 2010, 36(3): 323-337.
- [10] Crnkovic I, Sentilles S, Vulgarakis A, et al. A classification framework for software component models[J]. IEEE Transactions on Software Engineering, 2011, 37(5): 593-615.
- [11] Uokhale S S, Wong W E, Horgan J K, et al. An analytical approach to architecture based software performance and reliability prediction[J]. Performance Evaluation, 2004, 58(4): 391-412.
- [12] 印桂生, 宋敏, 韦正现, 等. 面向构件语义关系的软件体系结构演化分析[J]. 哈尔滨工程大学学报, 2011, 32(10): 1329-1335.
- Yin Guisheng, Song Min, Wei Zhengxian, et al. Analysis of software architecture evolution orienting semantic relationship between components[J]. Journal of Harbin Engineering University, 2011, 32(10): 1329-1335.

## A reliability analysis method of internetware oriented dynamic evolution

SONG Min<sup>1</sup>, JU Yang<sup>2</sup>, WEI Zhengxian<sup>2,3\*</sup>

1. Information Technology Center, Beijing Foreign Studies University, Beijing 100089, China
2. System Engineering Research Institute of China State Shipbuilding Corporation, Beijing 100094, China
3. Pilot National Laboratory for Marine Science and Technology, Qingdao 266011, China

**Abstract** Internetware is the software system in which various components in open and independent forms implement cross-network interconnections, intercommunications and collaborations in various collaborative methods. Dynamic evolution is an important characteristic of internetware and would lead to system failure when it occurs. So reliability analysis of internetware oriented dynamic evolution is one of the hot research topics. Firstly, the protocol relations of interaction at the interfaces between components and dynamic evolution error propagation reachability of internetware are analyzed, and a generation model of error propagation reachability graph is established. Secondly, in order to solve the problem of state space "explosion", a Markov state transition graph is generated from the error propagation reachability graph, and a Markov-based reliability analysis method for internetware oriented dynamic evolution is proposed. Finally, a case study demonstrates the effectiveness of the proposed method.

**Keywords** internetware software; dynamic evolution; system reliability; error propagation ●



(责任编辑 刘志远)