基于随机 Petri 网的软件可靠性分析①

Reliability Analysis of Software Based on Stochastic Petri Net

朱连章 李妍琛 (中国石油大学(华东) 山东东营 257061)

摘要:文章介绍了一种基于随机 Petri 网、在软件体系结构设计阶段对构件化软件进行可靠性早期分析的方法。该方法建立起的模型可以很好的描述软件系统的动态变化过程,并可以得到软件系统处于各个状态的瞬时及稳态概率,为分析系统运行一定时间后的可靠性情况提供了有利的手段。

关键词:随机 Petri 网 软件可靠性 软件体系结构 瞬时可靠度

1 引言

构件技术[1]是一种适应软件开发高效要求、集中体现了软件复用的思想的软件实现技术和方法。相对于以往的软件开发,构件技术更强调和侧重于软件体系结构(Software Architecture,SA)的设计。在 SA 设计阶段对软件进行可靠性分析能够尽早地发现软件系统中可能存在的问题,更加有助于提高软件的质量以及开发效率[2][3]。这也符合尽可能在软件开发的上游阶段对软件进行可靠性评估的思想。

但长期以来,在软件工程实践中,人们对整个软件系统的可靠性分析和评估工作大多忽略了软件体系结构的重要性。

本文通过分析软件体系结构的特征和可靠性因素,从故障建模和故障率分析的角度出发,提出了基于随机 Petri 网(Stochastic Petri Net,SPN)的构件化软件可靠性分析方法。该方法既可以动态跟踪软件系统的可靠性,又有利于降低软件可靠性描述与分析的复杂度,提高评价和预测可靠性的精确度。

2 随机 Petri 网

Petri 网是一种网状信息流模型,它不仅能用图形符号表示事件的原因和结果之间的关系,而且能表示系统的动态行为,为复杂系统的集成化建模、分析和评价提供了良好环境。Petri 网的结构元素包括位置(Place)、变迁(Transition)和弧(Arc),分别由圆圈、粗短线及圆圈中的黑点表示。位置用于描述可能的系统局部状态条件或状况,变迁用于描述修改系统状态的事件,弧规定了局部状态和事件之间的关系。事件引发局部状态的转换。每一条弧有一个对应的权值,称为弧权(Weight)。

在 Petri 网模型中,标记(Token)包含在位置中。随着事件的发生,标记可以按照弧的方向流动到不同的位置,从而动态的描述了系统的不同状态。如果一个位置描述一个条件,它能包含一个标记或不包含标记,当一个标记出现在这个位置中,条件为真,否则为假。如果一个位置定义一个状况,在这个位置中的标记个数用于规定这个状况。

一个 Petri 网模型的动态行为是由它的实施规则(Firing rule)规定的。如果一个变迁所有的输入位置至少包含一个标记,那么这个变迁可能实施(相联系的事件可能发生)。对这种情况这个变迁称为可实施。一个可实施变迁的实施导致从它所有输入位置中都清除一个标记,在它的每一个输出位置中产生一个标记。当使用大于 1 的弧权时,在变迁每一个输入位置中都要包含至少等于连接弧权的标记个数,它才可实施;这个变迁的实施,要根据相连接的弧权在它每一个输出位置中产生相应标记个数。

由于变迁的实施使标记在位置中流动,因此不同时刻标记在各个位置中的分布不同,这种不同的分布称为标识(Marking)。标识就相当于系统所处的状态

① 基金项目:中国石油大学(华东)2006 年研究生创新基金资助项目(项目编号:S2006 - 19)

(State)。

SPN 将时钟概念考虑进去,在 Petri 网的基础上对其中每个变迁引入一个满足指数分布的实施速率(firing rate)。SPN 以研究系统模型的组织结构和动态行为为目标,着眼于系统中可能发生的各种状态变化以及变化之间的关系[4][5],已成为性能评价领域一种有力的系统建模和分析工具。它既有助于定性理解被建模系统的动态行为,也能定量计算各种性能指标,为系统结构和参数的选择提供依据。

3 利用 SPN 对软件系统进行可靠性分析

对于由寿命分布均服从指数分布的构件组成的软件系统,可以先将系统分解成若干子系统;再用 SPN 对每个子系统进行建模;然后利用与 SPN 同构的马尔可夫链(Markov Chain,MC)对子系统在整个任务时间的可靠性进行评估;最后,按照子系统间的相互关系,应用组合分析的方法(本文仅考虑了基本的串并联方法)分析得到整个系统的可靠性结果。

3.1 软件系统的分解[6][7]

设软件系统由 n 个子系统(由 E_i 表示)组成,当且仅当 n 个子系统全部正常工作时,系统才正常工作,则称该软件系统是由 n 个子系统组成的串联型软件系统,其组成如图 1(a)。设第 i 个子系统的可靠度为 $R_i(t)$,则整个系统的可靠度 $R_s(t)=\prod_{i=1}^{n}R_i(t)$ (1)。若只要某一子系统正常工作系统就可以正常工作,则称该软件系统是由 n 个子系统组成的并联型软件系统,其组成如图 1(b),此时,整个系统的可靠度 $R_s(t)=1-\prod_{i=1}^{n}(1-R_i(t))$ (2)。

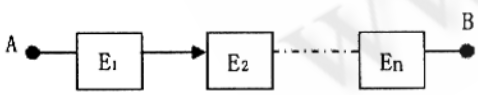


图 1 (a)

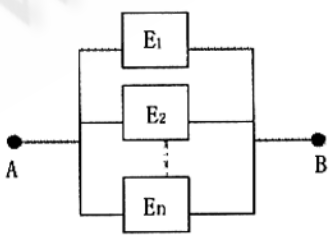


图 1 (b)

对于其它复杂的软件系统可采用递归的方法:每次依串联或并联的形式,对系统进行逐级分解,直至分解所得的每一部分均只含有单一的串联或并联形式为止;再利用串并联型软件系统的可靠性分析方法逐级回溯,便可分析出整个软件系统的可靠性。

3.2 软件子系统的 SPN 可靠性建模

在软件故障分析时,表征故障信息的 SPN 可以描述故障事件间的逻辑关系,如一因一果、竞争等,见表 1 所示。其中,位置可以表示构件的运行状态;变迁表征了系统状态的动态变化(即构件失效、软件缺陷等对系统运行状态的影响)及故障事件的传播。弧表征故障的传播方向。标记可以表征故障事件的发生,位置中标记数目的变化是由变迁实施引起的。

在相应的 SPN 模型中,变迁有三种构成:一是构件的故障率,用 λ_i 表示;二是构件的故障对相关构件的故障传递率,用 θ_i 表示;三是构件的修复率。我们可以用四种映射来描述构件之间的相互关系,同时也描述了软件失效的基本因素。

(1) 一对一映射。构件 A 的缺陷以故障率 λ_A 使 A 产生故障,A 的故障导致构件 B 产生故障,故障传递率用 θ_A 表示;若用 θ_{AB} 取代 θ_A 则表示是 A 的故障和 B 的缺陷共同使 B 产生故障,如图 2(a)所示。

(2) 一对多映射。如图 2(b)所示,A 的故障将以故障传递率 θ_A 同时使 B 和 C 产生故障;当然,A 对 B 和 C 的影响可能不是同时的,影响的程度也有所不同,可能是由于 A 的故障分别与 B 或 C 的缺陷共同导致 B 或 C 产生故障,如图 2(c)所示。

(3) 多对一映射。如图 3(a)所示构件 A 和构件 B 产生的故障共同以故障传递率 θ_{AB} 使构件 C 产生故障;也可能是 A 或 B 的故障分别与 C 的缺陷导致 C 产生故障,如图 3(b)所示;图 3(c)则表明 C 的故障由 A 与 B 的故障与 C 的缺陷共同造成。

(4) 多对多映射。可以将这种情况转化为一对多和多对一的映射加以分析。

构件间的具体依赖方式确定了弧权的大小。刚开始运行时系统中的缺陷还未被触发,因此各个位置中的标记数为零。只要设定的位置里标记数达到这些位置的容量,系统就将失效。

3.3 软件子系统的可靠性评估

(1) 结合子系统 SPN 模型的可达树确定子系统的失效状态集。

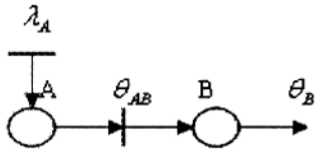


图 2 (a)

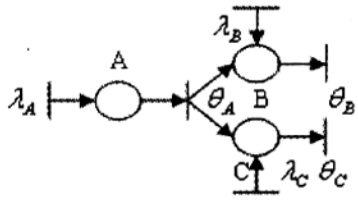


图 2 (b)

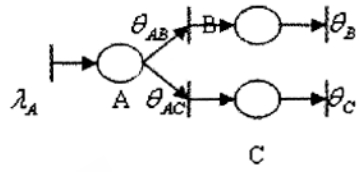


图 2 (c)

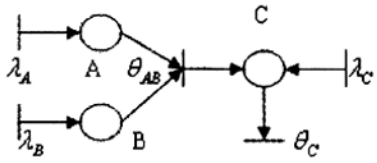


图 3 (a)

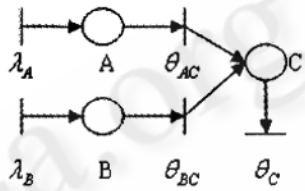


图 3 (b)

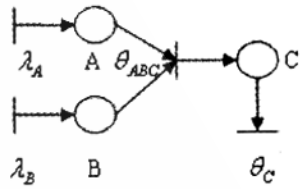


图 3 (c)

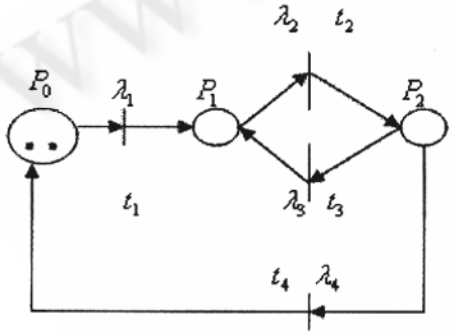


图 4 (a)

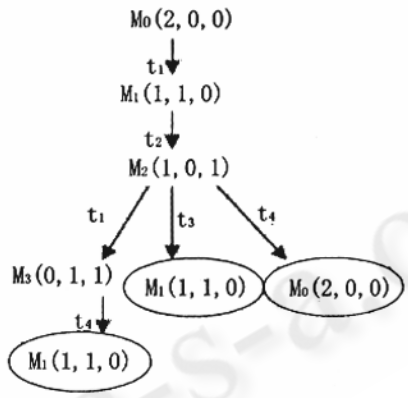


图 4 (b)

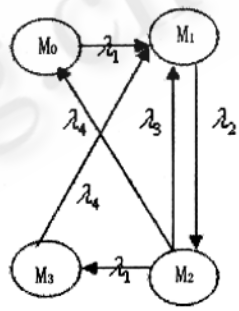


图 4 (c)

(2) 将可达树中相同标识合并,然后将每条弧上标注的实施变迁用该变迁的平均实施速率替换即可得到同构的 MC。MC 的每一个状态表示了一个特定的工作条件,包括正在运行的构件的数量、有多少是有故障的以及系统使用构件的逻辑等。

(3) 构造 MC 的 $n \times n$ 阶一步转移矩阵 $A = [a_{i,j}]$,其中 $1 \leqslant i,j \leqslant n$。

(4) 分析 MC 中各个可达状态标识,它们的瞬时概率及稳态分布分别由方程组(3)(4)确定。其中,$P[M_i](t) = x_i(t)$ 表示子系统在任意时刻 t 处于某状态的概率,即各可达状态标识的瞬时概率;$P[M_i] = y_i$ 表示 MC 中各个可达状态标识的平稳分布,$Y = (y_1, y_2, \cdots, y_n)$。

$$\begin{cases} (x'_1(t), x'_2(t), \cdots x'_n(t)) = (x_1(t), x_2(t), \cdots x_n(t))A \\ \sum x_i(t) = 1, 1 \leqslant i \leqslant n \end{cases} \quad (3)$$

$$\begin{cases} YA = 0 \\ \sum y_i = 1, 1 \leqslant i \leqslant n \end{cases} \quad (4)$$

(5) 根据失效状态集中失效状态的瞬时概率得出子系统失效的概率,从而得到子系统的瞬时可靠度:$R(t) = 1 - \sum P[M_i](t)$ (5),其中 $P[M_i](t)$ 为失效状态集中的元素。

(6) 根据得到的子系统 SPN 模型和 MC 以及可靠度确定子系统的其他相关可靠性指标。

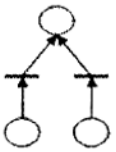
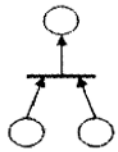
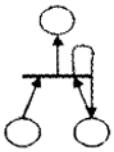
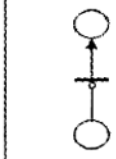
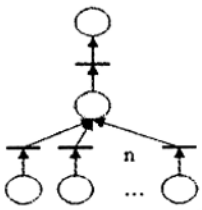
Petri 网模型					
因果关系	至少有一个输入事件发生时，输出事件就发生	仅当所有输入事件同时发生时，输出事件才发生	当条件事件发生时，输入事件引出输出事件	输出事件是输入事件的对立事件	n个输入事件中只要有m(m≤n)个发生，输出事件就发生

表 1

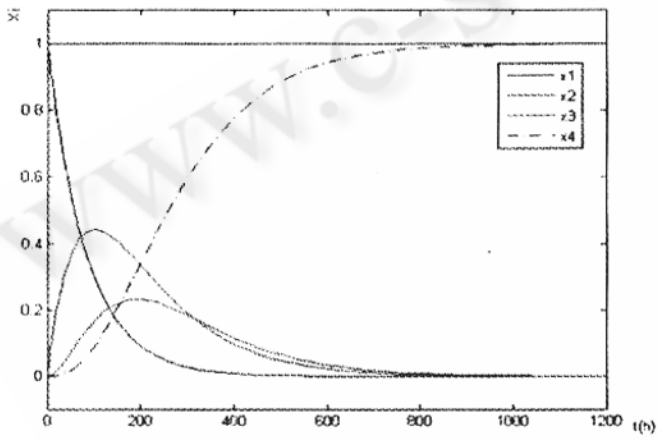


图 5

图 4(a) 可以看作是一个简单软件子系统的 SPN 可靠性模型，该子系统仅由两个构件 1、2 组成。位置 P_0 代表整个子系统的初始状态，P_1、P_2 分别代表两个构件的运行状态，弧权均为 1，位置容量 K = (2,1,1)。刚开始运行时子系统中的缺陷还未被触发，位置 P0 中的标记数为 2，位置 P1、P2 中的标记数均为 1。子系统运行一段时间后，构件 1 由于自身缺陷以故障率 λ_1 产生故障；修复构件 1 时，构件 1 的故障以故障传递率 λ_2 导致构件 2 产生故障；修复构件 2 时，构件 2 的故障以故障传递率 λ_3 导致构件 1 产生故障；对构件 1 无影响时，构件 2 的修复率为 λ_4。

图 4(b) 是 SPN 的可达树，图 4(c) 是对应的 MC。在此，根据需求，设 $\lambda_1 = 3\lambda, \lambda_2 = 2\lambda, \lambda_3 = 0, \lambda_4 = 0 (\lambda \neq 0)$，即构件 2 一旦发生故障便无法修复。以 M_3 为该子系统的失效状态，结合初始条件 $x_1(0) = 1, x_2(0) = 0, x_3(0) = 0, x_4(0) = 0$，根据公式 (3)(5) 求得这个软件子系统的瞬时可靠度为：$R(t) = 1 - P[M_3](t) = 9e^{-2\lambda t} - 2(4 + 3\lambda t)e^{-3\lambda t}$。当 $\lambda = 0.004h^{-1}$ 时，MC 各个可达状态标识的瞬时分布及变化趋势如图 5 所示，由此可以分析得到子系统的瞬时可靠度及稳态可靠度的走势。

4 结论

文章提出了一种基于随机 Petri 网的软件可靠性分析方法。着重介绍了构件化软件系统的分解、基于 SPN 的软件子系统的表达及获取系统可靠性分析结果的步骤等三个方面。这种方法可以清晰刻画出系统各个构件自身运行的情况及对整个系统可靠性的影响度。根据这些分析结果，可以选择合适的构件、优化软件体系结构；可以有针对性地编写测试用例并进行测试；还可以根据模型的变迁情况确定系统失效的路径，帮助对系统故障进行诊断等等。

从分析过程可以看出，构件数量的增加将会造成模型状态空间快速增长。虽然文中我们已经从模型的基本结构入手简化了系统的分析过程，减小了计算工作量，但为了快速求解，还应该在模型同构和压缩上做进一步研究。

参考文献

1 Clemens Szyperski, Domonik Grunlz, Stephan Mu - rer. Component Software: Beyond Object - Oriented Programming [M]. Second Edition, Publishing House of Electronics Indutsry, 2003.8.

2 毛晓光、邓勇进，基于构件软件的可靠性通用模型 [J]，软件学报，2004，(15).

3 张书杰等，基于构件软件系统集成测试的初步研究 [J]，北京工业大学学报，2004.6.

4 林闯，随机 Petri 网和系统性能评价 (第 2 版) [M]，北京：清华大学出版社，2005.4.