

一种基于工作流的软件可靠性评估方法

李妍,方少波,么强

(华南理工大学 计算机科学与工程学院,广东 广州 510000)

摘要:随着工作流系统在企业运营中担负越来越重要的角色,工作流可靠性研究的重要性也日渐增加。文中提出了一套完整的工作流系统可靠性评估方法,帮助企业检验工作流系统的可靠性。首先,文中对工作流系统的体系结构进行分析,其中重点分析了工作流系统的流程及活动模式,提出一个以流程为基础的工作流可靠性评估方法。文中采用基于马尔可夫链的流程可靠性算法,将活动中的可靠性数据整合,计算流程的可靠性数据。在此基础上,基于 Additive 模型,根据流程的使用率将多流程的可靠性信息进行整合,最后求出工作流系统的可靠性数据。根据该方法,设计并实现了工作流系统可靠性评估软件,并通过在 .NET 工作流平台上的实验,验证了该方法的可靠性预测效果是令人满意的。

关键词:软件可靠性;基于体系结构的软件可靠性;马尔可夫算法;工作流系统;流程可靠性;工作流可靠性评估

中图分类号:TP391.4

文献标识码:A

文章编号:1673-629X(2014)03-0034-05

doi:10.3969/j.issn.1673-629X.2014.03.009

A Method of Workflow-based Software Reliability Assessment

LI Yan, FANG Shao-bo, YAO Qiang

(College of Computer Science and Engineering, South China University of Technology,
Guangzhou 510000, China)

Abstract: As the workflow systems take an increasingly important role in business, the study of workflow reliability becomes more and more important. A method of process-based workflow software reliability assessment is presented to help enterprise for testing the reliability of workflow system. First of all, analyze workflow system architecture, which focuses on workflow processes and activities patterns. On this basis, a process-based workflow system reliability evaluation method is put forward using the Markov algorithm to calculate the processing reliability data by integrating the reliable data of activities. Then, based on the Additive model, integrate the reliability information of the multi-process and calculate the reliability of the entire workflow system. Based on this method, design and implement the workflow-based reliability evaluation software. Experimental results verify the effectiveness of this process-based reliability algorithm through .NET platform.

Key words: software reliability; architecture-based software reliability; Markov algorithm; workflow system; process reliability; workflow reliability assessment

1 概述

软件可靠性定义为,在给定时间内,特定环境下软件无错运行的概率^[1]。软件可靠性建模方法可以分为两种类型:基于时域的软件可靠性模型建模以及基于体系结构的可靠性模型建模。随着计算机技术的发展,软件系统演变得更加复杂以及多样,像近年来出现的手机软件系统、云计算系统、社交网络平台等。因此,仅仅依靠简单的可靠性模型来研究软件可靠性是不够的。软件可靠性分析需要以系统的体系结构为核心来研究,如此方能更好地评估软件可靠性信息。

随着企业规模以及业务的发展,流程和信息技术越来越成为企业管理范畴的重要课题。工作流管理系统是一个软件系统,它能够完成业务流程的定义和管理,按照预先定义的执行步骤和业务规则调度流程实例,在业务活动或任务之间合理地分配资源^[2]。工作流系统,属于基于构件系统的一个分支。但是,与传统的基于构件系统不同,工作流系统是以流程为基础组织在一起的,并且企业的流程是动态搭建的,持续变动的流程使得系统的规模结构动态增长,这为工作流可靠性研究带来一定难度。相对其他体系结构的可靠性

收稿日期:2013-05-16

修回日期:2013-08-24

网络出版时间:2014-01-08

基金项目:国家自然科学基金与中国民用航空总局联合资助项目(60776816);广东省产学研项目(x2jsE8130430)

作者简介:李妍(1987-),女,硕士研究生,研究方向为软件可靠性;导师:梅登华,副教授,研究方向为软件可靠性、嵌入式软件。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140108.0810.003.html>

研究,基于工作流的可靠性研究并不是很成熟,发展较慢。现有的工作流系统可靠性研究分为两类:第一种是从流程业务角度进行可靠性分析,通过基于路径或多目标规划算法获得流程业务可靠性,但这些方法均是理论上的研究,并不适用于复杂流程^[3-4]。第二种主要从面向服务层面对流程进行分析,通过流程将业务组装,计算出整个服务的可靠性。但该类方法主要采用基于路径的方式,通过将流程不断分解来进行计算,当活动模式互相交叉时,该类算法并不能分解流程。同时,流程分解算法也加大了系统实现的难度和复杂性^[5-7]。相比以上两种算法,文中提出的基于Markov 矩阵的流程可靠性评估方法,仅需要分析流程的一步转移率,降低了评估算法实现难度,提高了算法对流程分析的灵活性,并且适用于流程活动模式交叉时的情况,从而能更好地适应于实际的流程可靠性分析及评估。

2 流程结构分析

工作流管理系统,主要由工作流引擎、流程定义工具、监控及辅助工具构成。首先,用户通过流程定义工具定义流程,已定义好的流程通过工作流引擎进行运转,在运转过程由监控及辅助工具进行辅助。文中主要研究工作流管理系统中最核心的部分,即流程的运转时工作流系统的可靠性。

流程,是工作流系统的核心,主要由活动及活动间的转移构成。通过对活动及活动转移方式进行细致分析,可以为构建流程可靠性算法打好基础^[8]。

2.1 活动节点分类

活动是流程中的基本单位。基于工作流管理系统框架,文中将活动分为三类:标志节点、任务活动节点以及逻辑活动节点^[9]。具体分类如表1所示。

表1 活动节点分类

活动节点分类	活动节点
标志节点	起始活动、结束活动
任务活动	人工型活动、自动应用
过程逻辑活动	选择/条件活动、并行活动、循环活动、触发活动

2.2 活动失效分析

当活动不能提供用户所需要的服务时,将这种情况定义为活动失效。文中通过离散失效概率来定义及描述活动失效情况,计算公式为:

$$P_{failure} = \text{失效次数} / \text{总执行次数} \tag{1}$$

该活动的可靠性 P_i 为:

$$P_i = 1 - P_{failure} \tag{2}$$

不同类型的活动节点失效情况不同,下文将对每一类活动的失效情况进行分析。

2.2.1 标志节点

起始与结束活动仅仅是从逻辑意义上代表整个工作流的开始,而不涉及任何功能及服务的提供,因此该活动的失效概率及可靠性为:

$$P_{failure} = 0, P_i = 1 \tag{3}$$

2.2.2 任务活动

任务活动,是向整个流程提供功能及服务的基本单元,因此活动内存在程序错误并会引发软件失效。任务活动分为新建立活动及复用活动。新建立活动需要由流程专家定义活动的失效概率。而复用活动的失效概率则为:

$$\text{活动失效概率} = \text{活动失效次数} / \text{活动执行次数} \tag{4}$$

由于任务活动需要通过工作流引擎进行流转,所以也需要考虑工作流引擎的失效情况。任务活动实际失效概率及可靠性为:

$$P_{failure} = \text{活动失效概率} \times \text{工作流引擎失效概率} \tag{5}$$

$$P_i = 1 - \text{活动失效概率} \times \text{工作流引擎失效概率} \tag{6}$$

2.2.3 过程逻辑活动

对于选择、并行、循环等过程逻辑活动,活动主要功能为控制流程的流转。因此,活动的实际失效概率及可靠性为:

$$P_{failure} = \text{工作流引擎失效概率} \tag{7}$$

$$P_i = 1 - \text{工作流引擎失效概率} \tag{8}$$

2.3 活动转移方式

由于流程的多样性,活动转移方式非常复杂,WorkflowPatterns 一文中总结了 20 种工作流模式,这 20 种模式可以分为以下 6 个主要类别:基础控制流模式、分支及聚合模式、结构化模式、多实例模式、基于状态的模式、结束模式^[10]。

但是,在 20 种模式中,发现并不是所有模式都与流程的结构及活动转移方式相关。并且,这 20 种模式将分支及聚合的情况分开说明,并不能完整地展示流程分支情况^[5]。

因此,在工作流 20 种模式的基础上,对活动转移模式进行进一步的分析,将活动转移方式分为以下 6 种类型:

2.3.1 顺序模式

顺序模式中,活动根据活动排列的次序进行转移。活动需要在前面活动执行完毕的情况下才会进行。因此,当顺序模式中的其中一个活动失效,整个顺序路径均会失效。转移概率为:

$$P_i(i,j) = 1 \tag{9}$$

2.3.2 单选模式

单选模式中,活动根据选择的判断条件进行跳转。

在多条选择子路径中,单选模式只允许选择一条子路径执行。设 $P_i(\text{select},j)$ 为路径的选择概率。容易得知:

$$\sum_{j=1}^{j \leq n} P_i(\text{select},j) = 1$$

(10)

2.3.3 循环模式

循环模式中,循环里的活动,当满足循环条件时,会重复执行,直到循环条件不满足时为止。

2.3.4 并行模式

并行模式中,任务会并发地执行。当并行任务中的任一任务失败时,整个并行路径均会失效。转移概率为:

$$P_i(i,j) = 1$$

(11)

2.3.5 多选模式

多选模式中,用户在选择子路径时能够挑选多条路径执行,被选择的路径会并发执行。当多选子路径中的其中一个活动失效时,整个多选路径也会失效。

2.3.6 触发模式

触发模式中,触发活动后的所有子路径均会执行。但与并行模式不同,只要子路径中其中一条执行完毕,整个触发过程就完成,系统不会获取其他子路径活动的执行结果。因此,在触发子路径中,只要一条子路径没有失效,则整个触发路径均不会失效。

3 基于流程的可靠性计算方法

3.1 Markov 算法分析

Cheung 方法提出的 Markov 可靠性计算方法,主要通过路径的 N 步失效率求和的方式,将路径中的构件可靠性信息进行整合^[11-12]。但是,通过概率论知识及证明,发现 Cheung 所提出的基于构件的 Markov 可靠性算法,并不能完全支持工作流的转移模式,如表 2。

表 2 Markov 算法对转移模式的支持分析

支持 Markov 算法	不支持 Markov 算法
顺序模式	并行模式
单选模式	多选模式
循环模式	触发模式

因此,需要在 Cheung 方法的基础上,改进 Markov 算法,使之能支持工作流的所有转移模式。

3.2 基于 Markov 的流程可靠性计算算法

由上文可知,马尔可夫算法并不支持并行、多选及触发模式。此节将提出一套完整的流程可靠性计算算法,使之适用于所有工作流模式。

3.2.1 顺序、单选、循环模式

1) 构建马尔可夫状态空间。
首先,将流程中的每个活动,作为马尔可夫状态空间上的一个状态。并在此基础上增加 Failure 状态,代

表活动失效状态。

2) 构建马尔可夫矩阵。

需要根据活动的失效情况及转移概率将流程转化为马尔可夫矩阵。设 $P(i,j)$ 为状态的一步转移概率,设马尔可夫状态空间有 N 个状态,则可以构造 $N \times N$ 的矩阵。设 $P_{\text{markov}}(i,j)$ 为状态 i 到状态 j 的马尔可夫转移过程,设活动 i 到活动 j 的转移概率为 $P_i(i,j)$,活动 i 的可靠性为 P_i ,则可知:

■ 活动到活动状态的转移:

$$P_{\text{markov}}(i,j) = P_i(i,j)P_i$$

(12)

■ 活动到 Failure 状态的转移:

$$P_{\text{markov}}(i,\text{failure}) = 1 - P_i$$

(13)

通过构造马尔可夫矩阵 P_{markov} ,将活动转移信息及失效信息统一在同一个模型中,为流程可靠性计算打好基础。

3) 马尔可夫矩阵计算。

根据马尔可夫算法,求得流程的 N 步转移矩阵 $P_{\text{markov}}(n)$:

$$P_{\text{markov}}(n) = P_{\text{markov}}^n$$

(14)

设流程起点坐标为 i ,Failure 状态坐标为 j ,流程的 N 步失效率为 $P_{\text{failure}}(n)$ 。那么,流程的 N 步失效率为:

$$P_{\text{failure}}(n) = P_{\text{markov}}^n(i,j)$$

(15)

设整个流程的失效率为 P_{Failure} ,流程图活动数大于 3,则流程的失效概率为:

$$P_{\text{Failure}} = P(i,j) + P^2(i,j) + \cdots + P^n(i,j), n \geq 3$$

(16)

$$P_{\text{Failure}} \approx (I - P)^{-1}(i,j)$$

(17)

通过该算法,求得流程的失效率 P_{Failure} ,从而计算出整个流程的可靠性数值。

$$P_{\text{reliability}} = 1 - P_{\text{Failure}}$$

(18)

3.2.2 并行、多选、触发模式

由于并行、多选、触发模式马尔可夫算法并不支持,所以并不能直接构造马尔可夫矩阵。文中算法通过将并行、多选、触发模式的子流程进行分解计算,求得流程的可靠性。

1) 子路径分解。

第一步,根据活动的类型,通过识别并行、多选、触发活动,识别出对应的子路径;

第二步,对并行、多选、触发部分子路径进行分解,将每个子路径单独分离;

第三步,当子路径的活动数量大于 2 时,对子路径构造马尔可夫矩阵;

第四步,根据马尔可夫矩阵计算子路径的可靠性。

2) 子路径可靠性整合。

通过对子路径进行马尔可夫矩阵运算,获得每个子路径的可靠性数据。设子路径的可靠性为 P_{sub1} ,

$P_{sub2}, \dots, P_{subn}$, 通过将子路径的可靠性整合, 计算出并行、多选、触发模式部分的可靠性。计算算法如下所示:

■ 并行模式。
设并行模式部分的可靠性为 $P_{parallel}$, 可知:

$$P_{parallel} = P_{sub1}, P_{sub2}, \dots, P_{subn} \tag{19}$$

■ 多选模式。
设多选模式部分的可靠性为 P_{multi} , 路径选择的概率为 P_{si} , 可知:

$$P_{multi} = 1 - (1 - P_{s1})P_{sub1}(1 - P_{s2})P_{sub2} \dots (1 - P_{sn})P_{subn} \tag{20}$$

■ 触发模式。
设触发模式部分的可靠性为 $P_{discrim}$, 路径选择的概率为 P_{si} , 可知:

$$P_{discrim} = 1 - (1 - P_{sub1})(1 - P_{sub2}) \dots (1 - P_{subn}) \tag{21}$$

3) 主流程可靠性计算。
将并行、多选、触发模式部分分别用活动替代, 构造主流程。根据子路径整合求得的可靠性作为该活动的可靠性, 在此基础上对主流程构造马尔可夫矩阵, 计算整个流程的可靠性。

4 子流程组合构造 workflow 系统可靠性

workflow 系统可靠性计算方法主要借鉴基于构件的 Additive 模型的思想, 将多个流程的可靠性信息进行组合, 求出整个 workflow 系统的可靠性信息。基于构件的 Additive 模型, 它的核心思想是系统的失效强度是由子系统的失效强度叠加构成, 因此系统的失效等于子系统失效之和。该方法主要适用于由低耦合子系统构成的软件系统。workflow 系统, 虽然会存在不同流程中调用相同的活动, 但是流程与流程之间的关联度并不大。所以, 可以借鉴 Additive 模型, 将多个流程的可靠性叠加起来, 求出整个 workflow 系统的可靠性^[13]。

Additive 模型仅仅是对于各个子系统失效情况的简单叠加, 而并未考虑各个子系统的使用情况。在此基础上, 考虑了各个流程的实际使用频率, 以此为基础

构建整个 workflow 系统的软件可靠性模型:

workflow 系统可靠性 = 子流程 1 利用率 × 子流程 1 可靠性 + 子流程 2 利用率 × 子流程 2 可靠性 + ... + 子流程 N 利用率 × 子流程 N 可靠性

(22)

5 实验设计和结果

上文提出了一种基于流程的 workflow 可靠性计算及评估方法。在此基础上, 设计并实现了 workflow 可靠性评估软件。通过对比软件的计算结果以及 .NET workflow 平台上的流程运转可靠性情况, 验证了文中方法的正确性。

5.1 实验 1

实验 1 采用企业的接受订单 workflow 流程进行测试, 流程图如图 1 所示^[14]。

其中, 单选活动转入常规发货的概率为 0.9, 而转入通宵发货的概率为 0.1。对于每个活动的可靠性如表 3 所示。

表 3 接受订单流程活动失效率

活动编号	活动名称	可靠性
1	接受订单	0.97
2	寄送发票	0.99
3	接受付款	0.98
4	填写订单	0.97
5	常规发货	0.99
6	通宵发货	0.98
7	结束订货	0.99

通过在工作流可靠性评估工具软件中搭建流程, 可计算出整个流程的可靠性数值 $P_{reliability} = 0.89379$ 。通过将流程实际可靠性与系统计算的可靠性数值进行对比, 获得了评估软件计算结果的误差率。结果如表 4 所示。

根据测试发现, 随着测试次数的增加, 误差率有了显著的降低, 并且测试的数据的误差率均小于 1%。

5.2 实验 2

实验 2 依旧采用实验 1 的企业接受订单流程。该

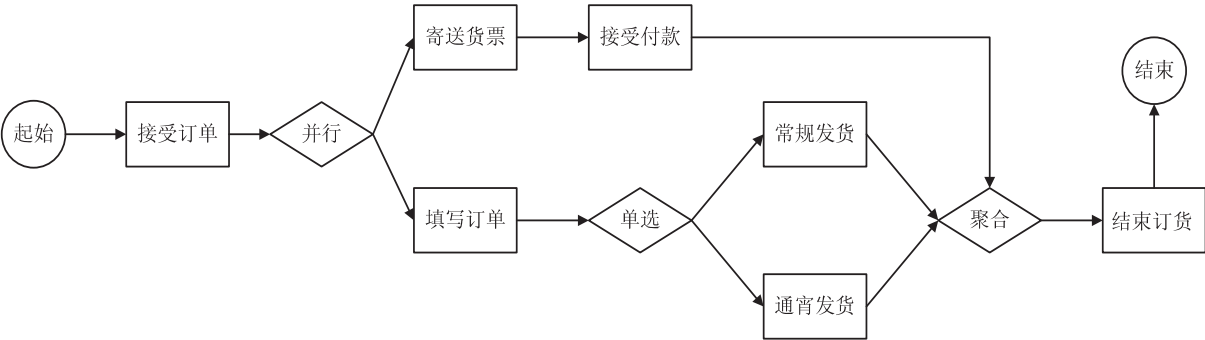


图 1 接受订单 workflow 流程图

表 4 实验 1 结果比较

测试 次数	失效 次数	流程实际 可靠性	误差率 /%	测试 次数	失效 次数	流程实际 可靠性	误差率 /%
1 000	112	0.888	0.65	6 000	642	0.893	0.08
2 000	209	0.895 5	0.19	7 000	736	0.894 86	0.12
3 000	338	0.887 3	0.7	8 000	850	0.893 75	0.004 4
4 000	447	0.888 25	0.62	9 000	967	0.892 6	0.14
5 000	532	0.893 6	0.02	10 000	1 084	0.891 6	0.2

实验通过.NET workflow 平台,对流程中的每个活动设置不同的失效率。通过程序控制,活动在设置的概率下,弹出错误信息。实验采用了 10 组典型数据进行测试,从而观测当活动失效率不同时,对于计算误差的影响。测试数据集流程执行次数为 1 000 次。测试结果如表 5 所示。

表 5 实验 2 测试数据集

数据 集	流程计算 可靠性	流程实际 可靠性	误差率 /%	数据 集	流程计算 可靠性	流程实际 可靠性	误差率 /%
1	0.795 43	0.788	0.8	6	0.941 48	0.954	1.36
2	0.589 8	0.578	1.9	7	0.160 88	0.164	1.9
3	0.615 89	0.624	1.3	8	0.989 45	0.987	0.25
4	0.415 23	0.42	1.14	9	0.972 84	0.976	0.32
5	0.868 3	0.859	1.05	10	0.954 31	0.941	1.41

根据测试数据发现,对于活动可靠性较高的流程,系统计算的误差率较为准确,而对于活动可靠性较低的流程,计算误差率则比较大。计算误差率小于 2%。

6 结束语

文中提出了一套完整的基于流程的工作流可靠性评估算法。文中的创新点在于从活动的类型、失效情况以及活动转移方式进行分析,并通过改进马尔可夫算法,使之能满足工作流的活动转移方式。与现有基于路径的工作流可靠性评估算法相比,文中采用马尔可夫算法,在实现建模时仅需要考虑流程的一步转移概率,并不需要陷入复杂的流程分解与计算之中,更可以解决活动流转模式互相嵌套的问题。基于该算法,实现了基于工作流的可靠性评估软件,并通过实验对

比,验证了工作流可靠性评估算法的正确性。

参考文献:

[1] Michael R. 软件可靠性工程手册[M]. 北京:电子工业出版社,1997.

[2] 侯志松,余周,冯启高. workflow 管理系统开发实录[M]. 北京:中国铁道出版社,2010.

[3] 林晖,赵泽超,张优云. workflow 系统模型的建立及其可靠性的评价[J]. 机械工程学报,2001,37(2):48-54.

[4] 王晓宏,孙壮志. 计算机协同设计中工作流可靠性的研究[J]. 计算机工程与应用,2003,39(3):47-49.

[5] Coppolino L, Romano L, Mazzocca N, et al. Web services workflow reliability estimation through reliability patterns[C]//Proc of third international conference on security and privacy in communications networks and the workshops. France:IEEE Conference Publications,2007.

[6] Zhao Shi,Lu Xiaoming,Zhou Xianzhong. A reliability model for Web services from the consumers' perspective[C]//Proc of CSSS. Nanjing:IEEE Conference Publications,2011.

[7] Grassi V,Patella S. Reliability prediction for service-oriented computing environments[J]. Internet Computing, 2006, 10(3):43-49.

[8] van der Aalst W,van Hee K,王建民,等. workflow 管理-模型、方法和系统[M]. 北京:清华大学出版社,2004.

[9] 范玉顺. 集成化企业建模方法与系统[M]. 北京:中国电力出版社,2007.

[10] van der Aalst W M P,ter Hofstede A H M,Kiepuszewski B. Workflow patterns[J]. Distributed and parallel database, 2003,14(1):5-51.

[11] Cheung R C. A user-oriented software reliability model[J]. Software engineering,1980,6(2):118-125.

[12] Reussner R H,Schmidt H W,Poernomo I H. Reliability prediction for component-based software architectures[J]. The journal of systems and software,2003,66(3):241-252.

[13] Xie M,Wohlin C. An additive reliability model for the analysis of modular software failure data[C]//Proc of software reliability engineering. Toulouse:IEEE Conference Publications, 1995.

[14] 张春艳. 基于 Web 服务的支持虚拟企业 workflow 技术的研究[D]. 福州:福州大学,2006.

(上接第 33 页)

[8] 丁杰. 新型高效协作式移动无线传感器网络技术研究[D]. 北京:北京邮电大学,2012.

[9] 肖明. 无线传感器网络中基于分簇的虚拟 MIMO 传输策略的研究[D]. 合肥:中国科学技术大学,2011.

[10] Heinzelman W B,Chandrakasan A P,Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks[J]. IEEE transactions on wireless communica-

tions,2002,1(4):660-670.

[11] 刘卫平. 地面战场无线传感侦察系统中的盲信号处理问题研究[D]. 南京:南京邮电大学,2013.

[12] Li X. Blind channel estimation and equalization in wireless sensor networks based on correlations among sensors[J]. IEEE transactions on signal processing,2005,53(4):1511-1519.

一种基于工作流的软件可靠性评估方法

作者：[李妍](#)，[方少波](#)，[么强](#)，[LI Yan](#)，[FANG Shao-bo](#)，[YAO Qiang](#)

作者单位：[华南理工大学 计算机科学与工程学院, 广东 广州, 510000](#)

刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

ISTIC

年，卷(期)：2014(3)

本文链接：http://d.wanfangdata.com.cn/Periodical_wjtz201403009.aspx