

柔性 workflow 建模引入 NVP 的可靠性优化研究^①

万杰龙

(同济大学 计算机科学与技术系, 上海 201804)

摘要: 针对当前大型企业的工作流管理系统普遍存在建模能力不充分、可靠性不足等问题, 提出一种基于 N 版本程序设计(N-Version Programming)容错策略的柔性 workflow 建模方法, 该方法相比于传统 workflow 模型, 有效应用 N 版本程序设计容错模式, 为 workflow 关键活动节点增加容错机制, 从而显著提升 workflow 管理系统的可靠性. 同时为满足模型应用的现实需求, 详细分析 N 版本程序设计的可靠性优化, 最后通过模型实例, 说明该建模方法在模型描述能力和系统可靠性方面的优势.

关键词: workflow 模型; 容错策略; N 版本程序设计; 可靠性优化

Optimizing the Reliability of Flexible Workflow Based on the N-Version Programming

WAN Jie-Long

(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)

Abstract: To improve the description ability and reliability of the workflow management system used in large enterprises, this paper proposes a flexible workflow model based on the N-version programming. Compared with the traditional workflow model, the model applies the N-version programming method on the key node in the workflow to significantly enhance the reliability of the workflow management system. Furthermore, how to maximize the reliability of the N-version programming method is described to meet the practical needs of the model application. Finally, this paper introduces the realized workflow model and the corresponding system in detail. The experiment result shows that the new modeling method is high efficient and satisfactory.

Key words: workflow model; Fault-tolerant strategy; N-version programming; reliability optimization

workflow 这一概念自 20 世纪 90 年代提出以来, 一直是计算机应用领域的研究热点, 广受业界学者关注. 1993 年, 国际 workflow 管理联盟(Workflow Management Coalition, WfMC)的成立标志着 workflow 技术开始进入相对成熟的阶段. 为了实现不同 workflow 产品之间的互操作, WfMC 在 workflow 管理系统的相关术语、体系结构及应用编程接口等方面制定了一系列的标准, 并给出 workflow 定义^[1,2]: workflow 是指整个或部分经营过程在计算机支持下的全自动或半自动化.

workflow 技术发展迅速, 已广泛应用于企业业务过程建模和过程优化, 为业务过程实现提供了强有力的支持^[3]. 然而随着企业信息化建设的深入和 Internet 的

发展, 现有 workflow 管理系统的不足也逐渐显现, 主要可以分为以下几种类型^[4]: (1)模型内容有限, 不能满足流程多样性的要求; (2)模型缺乏对于分布式环境的支持; (3)模型可靠性不高; (4)模型衍生软件易使用性、互操作性不佳.

workflow 管理系统在可靠性和系统柔性上所暴露的问题, 其关键在于 workflow 模型本身, workflow 模型是对业务过程的抽象表示, 相对于 workflow 产品的实现技术和发展速度, workflow 建模理论的研究较为滞后, 目前已有的建模方法主要针对提高模型柔性, 而忽略对于系统可靠性的提升. 在建模过程中, 并没有引入容错策略或其它能够提高系统可靠性的机制, workflow 的执

^① 收稿时间:2012-04-16;收到修改稿时间:2012-05-28

行基于每一个活动都能够正常完成这一假设,该逻辑无法应对当前日益复杂多变的信息环境。

N 版本程序设计^[5](N-Version Programming, NVP) 是由 Avizenis 和 Chen 于 1977 年首次提出的,是一种利用冗余机制建立容错软件的模式。N 个功能等价的软件共同执行同一个任务,多数一致的执行结果作为该任务的正确结果输出。在传统的软件开发过程中,利用 N 版本程序设计技术会极大地增加软件的实现和维护代价,因而只有极少数的关键系统会采用冗余技术来增强系统可靠性。随着 Web 服务技术和 workflow 系统的出现,很多独立的服务提供商都会提供许多功能等价或相似的 Web 服务,这使得 N 版本程序设计模式的容错机制实现变得可行。然而需要注意的是:在 workflow 模型中应用 N 版本程序设计模式提高系统可靠性的同时,也会使得 workflow 模型的执行代价增大,因此有必要确保在一定的代价约束条件下使 workflow 的可靠性达到最大。

本文基于软件容错理论,通过引入 N 版本程序设计容错模式,提出一种新型的建模方法,旨在提高 workflow 模型的系统可靠性。在传统的工作流建模基础上,为业务流程关键节点增加容错机制,以此提升 workflow 系统的可靠性。随后为满足模型应用的现实需求,详细分析 N 版本程序设计的可靠性优化。最后给出一个大规模征询活动的过程模型实例,验证该建模方法在模型描述能力与系统可靠性上的成果。

1 NVP容错模式增强系统可靠性

传统的工作流建模方法存在诸多弊端:(1)整个建模过程工作量十分巨大,且无法对突发事件作出灵活的调整;(2)对用户的要求较高,用户必须缜密地考虑复杂的业务逻辑关系,合理地建立整个业务流程模型;(3)必须在工作流建模阶段确定所有业务活动的执行细节,而实际应用中业务流程具有复杂性和多变性。因此在工作流建模中有必要引入能够反映复杂实际应用场景的机制提高模型描述能力,文献[6]引入协调理论与反馈机制的思想,扩充实现 workflow 建模的构成元素,一定程度上提高了模型本身的描述能力,但该方法未能有效地考虑到 workflow 执行过程中的系统可靠性问题,对于 workflow 中业务活动的执行仍假定为能够且必须完成。文献[7]采用 UML 作为建模语言,并给出了从 UML 活动图到 workflow 建模的转换框架,但没有形式化规则的支持。由此,可以发现目前对于 workflow 的研究

主要集中在模型、方法和框架上,而缺乏对于容错策略的支持,很难保证 workflow 执行的可靠性。

本文针对上述建模方法存在的问题,引入软件容错理论的思想,利用 N 版本程序设计容错机制,实现 workflow 中关键业务节点的容错保障,workflow 模型元素遵循协调理论与反馈机制,以保持对于业务环境的模型描述能力。N 版本程序设计容错模式不仅能有效保证系统关键节点的执行,并且能够同时提高 workflow 执行的整体柔性。

workflow 柔性(Flexibility)指的是 workflow 能够快速、低成本地响应外界变化的程度^[8]。从系统的观点来看,Slack 和 Nigel 根据柔性的范畴与响应特征将给出的柔性的经典定义包含了范围柔性(Range flexibility)和响应柔性(Responsiveness flexibility)2 个维度^[9]。范围柔性是指 workflow 系统在响应外界变化时所能相应变动的状态范围。响应柔性是指 workflow 系统在响应外界变化进行相应转换时,所需要的时间与经济成本。

在 workflow 执行的过程中,即便存在部分活动发生失效,仍然能够通过容错机制恢复正常执行直至整个 workflow 完成。在 workflow 建模过程中引入容错策略,能够大幅度提高 workflow 系统执行的柔性:(1)有效保证 workflow 的范围柔性,在 workflow 应对外界变化做出相应更改时,容错策略确保 workflow 执行过程中的各个业务活动本身的可靠性与可用性;(2)增强 workflow 的响应柔性,使用合适的容错策略能够降低整个 workflow 模型在进行相应转换时所需要的时间与费用。

1.1 基于工作流的可靠性预测模型

本文的 workflow 实现基于 BPEL(Business Process Execution Language, 业务流程执行语言),BPEL 采用流程结构表示业务中的处理逻辑与执行算法,用于自动化业务流程。对于 BPEL 工作流的描述采用业务流程图,流程图中的每一个节点称为一个活动,具体分为基本活动与结构活动^[10]。基本活动主要由 receive、reply 和 invoke 活动表示消息间的传递。结构活动主要由 switch、pick 活动表示选择关系,while 用于反复执行一个活动,直至条件满足。基本活动通过结构活动合成复杂的执行算法,表达业务流程中的具体设计。

依据 workflow 本身的性质,预定义流程的 workflow 建模,其实质是一个随机型离散事件系统。在 workflow 建模完成之后,通过业务流程执行进行离散事件模拟,可以得到 workflow 模型在一次执行过程中每个活

动的平均执行次数. 工作流的可靠性预测采用层次方法: 假设某个使用 BPEL 表示的工作流模型是由 n 个表示调用其它 Web 服务的 invoke 活动组合而成. 在无失效情况下, 每个活动的平均执行次数用 V_i 表示, 每个活动的可靠性为 R_i , 根据基于层次方法的可靠性估计模型^[11]得到工作流模型的可靠性 R :

$$R = \prod_{i=1}^n R_i^{V_i}$$

1.2 基于 NVP 的可靠性优化

N 版本程序设计是一种提高软件系统可靠性的冗余屏蔽技术. 使用 N 版本程序设计容错模式能够提升工作流的可靠性, 弊端在于使用该容错机制会对工作流执行时的代价开销产生影响, 所以为了更好的现实应用, 需要在可接受的工作流执行代价的限制下使用 N 版本程序设计进行可靠性优化, 由此讨论在一定代价约束下的可靠性最大化: 假定工作流由 N 个活动组成, 记为 $\{A_1, A_2, \dots, A_n\}$. 其中每个活动的候选服务集合为 $S_i (i=1, 2, \dots, N)$. 同时为不失一般性, 每一个服务集合 S_i 的个数应分别定义, 即 $|S_i|=N_i, S_i=\{S_{i1}, S_{i2}, \dots, S_{iN_i}\}$. 对于每一个活动的候选服务, r_{ij} 表示其可靠性, C_{ij} 表示其代价.

对于活动 A_i 及其候选服务集合 $S_i=\{S_{i1}, S_{i2}, \dots, S_{iN_i}\}$, 令 M_i 是 S_i 的子集且 $|M_i| \geq 1$. 若子集 M_i 的个数为 1, 则该活动只有一个候选服务进行执行; 若子集 M_i 的个数大于 1, 则该活动由多个候选服务采用 N 版本程序设计容错模式执行.

工作流的活动集合为 $\{A_1, A_2, \dots, A_n\}$, 活动 A_i 的候选服务集合为 $S_i (i=1, 2, \dots, N)$, 利用 N 版本程序容错机制的工作流可靠性优化问题是指寻找该工作流的满足代价约束的建模方案 P , 使得该建模方案的可靠性最大, 即:

$$\begin{aligned} & \max \prod_{i=1}^n R_i^{V_i} \\ & s.t. \sum_{i=1}^N \sum_{j \in M_i} C_{ij} \leq B \end{aligned}$$

其中 $|M_i| \geq 1, M_i \subseteq S_i$.

根据文献[12], 该冗余优化问题是 NP-hard 问题, 为了得到满足代价约束的工作流可靠性最优化方案, 单纯地通过枚举所有可能的方案寻找满足约束条件的解将产生难以承受的计算开销. 由此工作流建模用户需要使用合适的优化搜索算法寻找最优方案.

2 基于NVP的工作流建模实例

为了在工作流模型中显式地说明业务活动节点是否使用 N 版本程序设计容错机制, 在工作流建模的流程图描述中新增如图 1 所示的表示方法: 若工作流模型中的活动节点周围有正八边形虚线框环绕(标注 FT-N), 则该活动节点使用了 N 版本程序设计容错机制. 反之则该节点活动未使用任何容错机制;

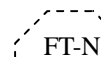


图 1 使用 NVP 容错策略

图 2 给出了一个大规模征询活动的工作流模型实例(例如: 地方政府或者职能部门向社会征集某方面的解决方案、可行建议等), 大规模征询活动主要包括: 征集内容的表述准备、发布征询、接收各方结果、征询机构内部评选、最后根据评选确定最终结果.

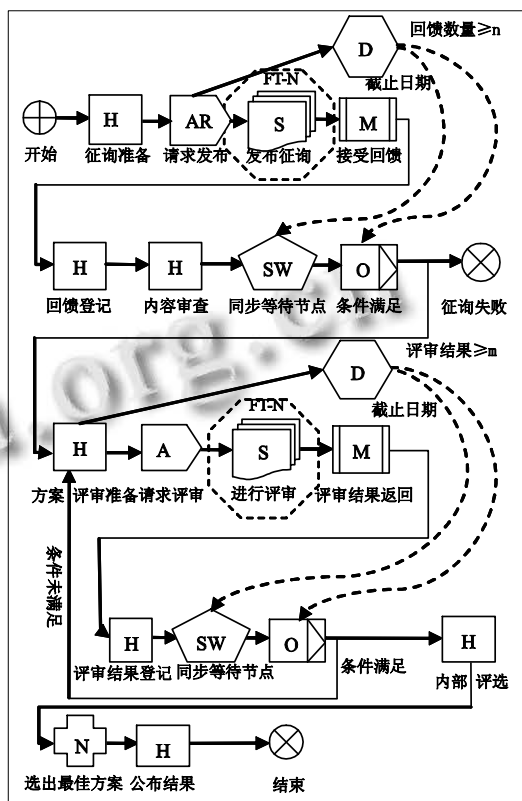


图 2 大规模征询实例的工作流模型

根据上述工作流模型的流程图表述, 使用 Oracle BPEL Process Manager 作为工作流建模工具, 并进行该模型实例的模拟仿真实验, 针对工作流模型在系统

可靠性方面的表现进行验证. workflow 模型的运行 PC 环境为双核 Intel Core i5 CPU、内存 1G×2 双通道, 操作系统 Windows XP SP2.

从前文描述的工作流建模的优化机理中可以发现, 是否使用 N 版本程序设计容错机制对于工作流的柔性以及可靠性有着重要的影响, 由此, 采用压力测试比较使用 N 版本程序设计容错机制的工作流模型与未使用该容错机制的工作流模型, 检验 N 版本程序设计容错机制对于工作流模型的作用.

压力测试参数				
开始日期:	Tue Feb 14 22:37:41 CST 2012			
实际调用持续时间:	17031 ms			
并行线程数:	200 个线程			
循环数:	5 循环			
各调用间的延迟是否恒定?	1000 ms			
清除统计信息?	是			
BPDL 服务器配置				
启动程序线程 (最小值):	默认值			
启动程序线程 (最大值):	默认值			
启动程序线程 (负载因子):	默认值			
Worker 线程 (最小值):	默认值			
Worker 线程 (最大值):	默认值			
Worker 线程 (负载因子):	默认值			
调用聚集报告				
计数	平均值	最小值	最大值	速率
1000	2260.1 ms	0 ms	6312 ms	58.71 i/sec

图 3 使用 NVP 容错机制的工作流测试报告

压力测试参数				
开始日期:	Tue Feb 14 20:51:54 CST 2012			
实际调用持续时间:	16582 ms			
并行线程数:	200 个线程			
循环数:	5 循环			
各调用间的延迟是否恒定?	1000 ms			
清除统计信息?	是			
BPDL 服务器配置				
启动程序线程 (最小值):	默认值			
启动程序线程 (最大值):	默认值			
启动程序线程 (负载因子):	默认值			
Worker 线程 (最小值):	默认值			
Worker 线程 (最大值):	默认值			
Worker 线程 (负载因子):	默认值			
调用聚集报告				
计数	平均值	最小值	最大值	速率
967	2235.7 ms	0 ms	7123 ms	63.27 i/sec

图 4 未使用 NVP 容错机制的工作流测试报告

比较上述 2 份测试报告可以发现: 未使用容错机制的工作流模型在大规模并发线程数达到 1000 时, 只成功运行其中 967 组, 而使用容错机制的工作流模型成功运行所有的 1000 组线程. 由此可以表明容错策略机制的引入对于提升执行大规模事务的工作流的可靠性有着显著作用.

3 结语

目前, 制约 workflow 系统发展的主要原因在于 workflow 管理系统在可靠性以及柔性方面存在着不足, 而大部分的相关研究针对于提升系统柔性, 对于系统可靠性的关注不够. 本文提出一种基于 N 版本程序设计容

错策略的工作流建模方法, 相比于传统 workflow 模型, 应用 N 版本程序设计容错模式, 有效保证 workflow 关键活动节点的执行, 从而提升整个 workflow 流程的可靠性. 对于大规模征询活动的工作流模型实例进行分析以及仿真实验的验证表明, 本文所提出的建模方法能够有效提高在高负载情况下 workflow 执行的可靠性.

参考文献

- 1 WfMC. Workflow Management Coalition Technology and Glossary(WfMC-TC-1011). Technical Report Workflow Management Coalition, Brussels, 1996.
- 2 WfMC. The Workflow Reference Model. WfMC-001003, 1998.
- 3 罗海滨, 范玉顺, 吴澄. 工作流技术综述. 软件学报, 2000, 11(7):899-907.
- 4 Wang WY, Sha JC, Tan DF. Comparison of Petri net based and UML activity diagram based workflow modeling. Journal of System Simulation, 2006, 18(2):504-507, 510.
- 5 Ashraei N, Berman O, Cutler M. Optimal design of large software-systems using N-version programming. IEEE Trans. on Reliability, 1994, 43(2):344-350.
- 6 范玉顺, 吴澄. 一种提高系统柔性的工作流建模方法研究. 软件学报, 2002, 13(4):833-839.
- 7 Jiang P, Mair Q, Newman J. Using UML to Design Distributed Collaborative Workflows: from UML to XPDL. IEEE Computer Society, 2003.
- 8 Huang LH, Ge YL, Fu XL. Conceptual Framework of the Flexibility of Business Process. Systems Engineering-Theory & Practice, 1999(10):63-68.
- 9 Nigel S. Manufacturing Systems Flexibility an Assessment Procedure. Computer Integrated Manufacturing Systems, 1988, 1(1):25-31.
- 10 Zhong DH, Qi ZC, Xu XS. Optimizing the Reliability of Web Services Composition Based on the N-Version Programming. Computer Engineering & Science, 2008.
- 11 Carey MJ, DeWitt DJ. Of Objects and Databases: A Decade of Turmoil. Proc of VLDB'96. 1996:3-14.
- 12 OMG. Persistent State Service Specification version 2.0. 2001.