工作流重构技术研究

田 珂[1] 朱清新[2] 向培素[3]

(电子科技大学计算机学院 成都 610054)[1] (西南民族学院电气信息工程学院 成都 610041)[2]

摘 要 先进的工作流技术与传统的企业管理信息系统相结合,日益成为提高企业信息化的一个重要手段。目前的工作流是基于模型驱动的,定义一个完整的模型是相当复杂和费时的;而且,实际业务流程同流程模型之间必然存在差异。本文介绍了工作流网,工作流日志的概念;提出了一种基于日志包含的信息来重构业务流程模型的算法,该算法还能处理日志中的干扰信息和有效地度量流程模型和实际业务流程之间的差异。

关键词 工作流事件,工作流日志,排序关系,重构算法

Study of Refactoring for Workflow

TIAN Ke[1] ZHU Qing-Xin[2] XIANG Pei-Su[3]

(College of Computer, VESTC of China, Chengdu 610054)[1]

(School of Electric Information and Engineering, South-wet Nation College, Chengdu 610041)[2]

Abstract It is increasingly important for Enterprise to combine the advanced workflow techniques and legacy information system. Current workflow management systems are driven by process models. It is often complicated and time-consuming to define a workflow process model and typically, there are discrepancies between the actual workflow processes and the processes as perceived by the management. Workflow log contains information about all workflow events. We assume that each event refers to one task being executed for a single case. The algorithm presented in this paper induces the actual workflow model through the runtime workflow logs. It can also deal with noise and measuring the discrepancies between prescriptive models and actual process executions.

Keywords Workflow event, Workflow log, Ordering relation, Refactoring algorithm

1 介绍

在过去的十几年中,工作流技术在业界和学术界都得到了长足的发展。自1993年工作流管理联盟成立,制定了相关的一系列标准,工作流技术更是在EIS中得到了广泛的应用。但从目前的情况看,工作流技术往往并未发挥出预期作用;这其中固然有流程参与人员主观上的原因,但就技术层面而言,当前的工作流信息管理系统也存在一个开环控制、匮乏反馈的问题。工作流模型定义往往是由一些顾问,管理人员和专家等来完成的,其初始的模型定义中不可避免地存在着不完整性,主观性等问题。此外,在工作流的实施过程中,包含系统配置以及相关参与人员培训等工作,都有可能引入另外一些不确定因素。正如IT界的一句名言:魔鬼存在于细节之中。如果不能处理好这些不利和不确定性因素,就可能导致工作流技术应用失败。本文提出的工作流重构技术的初衷就是希望通过收集处于生产状态的业务流程所产生的工作数据,再对这些数据进行分析,以达到重构流程模型并度量流程模型和实际业务流程之间差异的目的。通过这种重构技术,帮助企业实行业务流调整、流程梳理与优化,以利于企业管理战略的实现。

2 基本概念

工作流技术的一个重要目标就是尽可能有效地处理在企业中存在的各种业务流程。这些流程一般说来都是以一定顺序执行的任务。对业务流程进行建模,不但需要定义需要执行的任务,并且要定义任务执行的先后顺序。这样的模型就是一个'工作流图'或者是'工作流路由定义'。在流程模型中,路由元素用来表述流程中所存在的顺序执行,条件,并行和反复路由。

2.1 工作流网

在Petri网的基础上,Aalst提出了工作流网(WF-Net)的概念。工作流网定义了一个流程实例的生命周期,并规定任何对流程处理没有影响的任务都不应包含在工作流网中。工作流的所有节点都应当在从开始到结束的路径之中。

2.2 工作流日志

T为一个工作流任务的集合,$\sigma \in T^*$是一个工作流回溯,$W \in P(T^*)$就是一个工作流日志。要基于工作流日志来重构工作流模型,首先需要分析任务之间的因果关系。例如:如果一个任务总是在另一个任务后面出现,就认为在两者之间存在着因果关系。

2.3 工作流事件及排序关系

一个工作流日志就是一个事件序列集合。一个事件可以用一个流程实例和一个任务标识符来表述,$e=(c,t)$。工作流事件e就是任务t相对于一个给定的流程实例c的执行。假设A、B是事件,w是工作流日志,下面定义基于工作流日志的四种排序关系:

(1)$A>B$ 当且仅当在日志中有事件跟随记录,可以得出B是直接跟在A的后面;

田 珂 博士研究生,主要研究方向:电信网络管理、工作流技术、电子政务;**朱清新** 博士生导师,主要研究方向:算法设计系统仿真、最优控制与搜索;**向培素** 硕士,主要研究方向:电信网络管理、工作流技术、电子政务。

(2)$A\rightarrow B$　当且仅当 $A>B$ 而 $B>A$ 是不存在的；

(3)$A\#B$　当且仅当 $A>B$ 和 $B>A$ 都不存在；

(4)$A//B$　当且仅当 $A>B$ 和 $B>A$ 都存在的；

关系 $A\rightarrow B$ 可以认为是依赖关系(B 直接依赖于 A)，$A\#B$ 是非并行关系(例如：在两者之间没有直接的依赖关系，而且也不存在着并行)，$A//B$ 是并行关系(在实际业务中往往体现为并发)。在本文的方法中，依赖关系用来联接事件，并行关系用来检查分支和合并的类型是 AND(与)还是 OR(或)。

2.4　完备的工作流日志

若 $PN=(P,T,F)$是一个正确工作流网，W 是 PN 的一个工作流日志，且 $W\in P(T^*)$，$P(T^*)$为 T^* 的一个幂集。每一个 $\sigma\in W$ 的 σ 都对应为 PN 网的一个点火序列，也对应 PN 网的某一个状态。W 为 PN 的一个完备工作流日志的条件是：对于 PN 网的任何工作流日志，$t\in T$，当 $\sigma\in W$ 时，都有 $t\in\sigma$。一个正确 WF 网的工作流日志仅需包含已被暴露出来的流程行为。

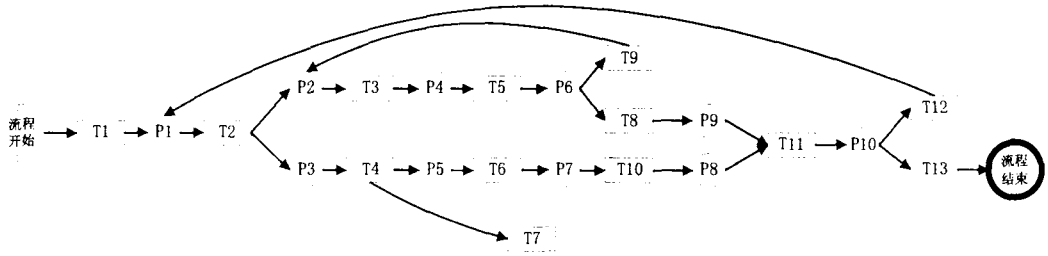

图 1　一个工作流的 Petri 网模型

Petri 网可以用来定义流程实例的路由，变迁代表任务，库所代表流程事件之间的因果依赖关系。事实上，一个库所可视作一个条件，该条件用作表示任务前后的与或关系。在任何时候，一个库所都包含着 0 个或多个令牌，变迁是 Petri 网的主动元素，它们根据点火原则来改变网络状态。

工作流网从功能，组织，信息和操作层面上对实际业务流进行了抽象，尽管其很简单，但解释能力很强。本文中，我们讨论的工作流网局限于正确的工作流网。

一个工作流是正确的，必须满足下面的条件：1)该工作流是可以确保终止的，若有多个结束节点，当第一个结束节点到达时，工作流模型终止；2)在终止节点上不包含任何令牌；3)没有死任务，能够通过路由来执行任何一个任务；4)工作流网中不能有死锁和活锁的情况出现。正确性只是任何一个业务流程运行应满足的最基本因素，正确的工作流网是对业务流程协作管理进行建模的基石。

本文中，我们通过工作流日志来重构业务流程模型。如上所述，工作流日志是一个工作流事件序列集合，可能包含上千个流程实例的信息。对于某个定义完备的工作流，可以通过流程实例标识符将该工作流日志按其实例进行分割。对于一个流程模型而言，不同实例之间的事件之间没有因果关系。因此，将工作流日志转换成工作流事件序列的集合，不会造成任何的信息损失。工作流日志最终可以表述为一种简单的任务标识队列，例如：{$T1,T2,T4,T3,T5,T9,T6,T3,T5,T10,T8,T11,T12,T2,T4,T7,T3,T5,T8,T11,T13$}。

3　重构算法的形式化规约

本文的重构方法基于上述的四种排序关系。算法的形式化规约假定下列完备信息：

(1)日志是完整的(例如：如果一个任务可直接跟在另一个任务后面，日志中就至少包含这种情况的一个记录)；

(2)假设在日志中没有干扰信息(例如：任何在日志中记录的信息都是正确的)。在实际情况下，日志很少是完整的，并且总是包含着噪声(我们借用一个信息论词汇来表示在日志中记录的错误信息)。因此，在现场情况下，根据日志来判断两个事件 A 和 B 之间的关系是 $A\rightarrow B$ 还是 $A\#B$ 其实较为困难。例如：因果关系 $A\rightarrow B$ 意味着：在两个事件 A 和 B 之间，在日志中只能跟踪到有 B 直接跟在 A 后面的记录(即有 $A>B$ 存在)，而在日志中应没有 A 直接跟在 B 后面的记录。但在噪声环境中，一个偶然的错误日志就可能推翻原本正确的结论。

基于上述原因，希望可以找出一种启发式的流程重构算法，它对噪声不敏感，也不要求日志的完整性。该算法中包含三个部分：(1)依赖-频率表的构造；(2)依赖-频率图的推导；(3)从图和表中重构工作流网。

3.1　构建依赖-频率表

对于一个工作流任务 A，可以从流程日志中提炼出下列信息：

(1)任务 A 的总发生次数 (记为 $\#A$)；(2)任务 A 直接在 B 后面的总发生次数(记为 $\#B<A$)；(3)任务 B 直接跟随在 A 后面的总发生次数(记为 $\#A>B$)；(4)局部依赖关系强度(记做 $A\rightarrow^{L}B$)，标记任务 A 和 B 之间直接依赖关系的强弱；(5)全局依赖关系强度(记做 $A\rightarrow B$)。

其中(4)可以记为：$A\rightarrow^{L}B=(\#A>B-\#B>A)/(\#A>B+\#B>A>A+1)$。

在定义(4)中，仅利用了相邻任务之间的关系。根据定义(4)，如果事件 B 直接跟在事件 A 的次数为 5，而事件 A 从未跟在事件 B 的后面，则 $A\rightarrow^{L}B=5/6=0.8333$。对于该值，我们尚不能完全确信 A 和 B 之间依赖关系的存在(特别是考虑存在干扰数据的情况)；但如果 $\#A>B$ 的值是 50，而 $\#B>A$ 是 0，那么 $A\rightarrow^{L}B=50/51=0.98$。对于该值，就能够确信 A 和 B 之间依赖关系的存在。即便存在噪声，导致 $\#B>A$ 为 1，$A\rightarrow^{L}B=49/52=0.942$，我们仍能确信这个依赖关系的存在。

定义(5)是对工作流事件依赖关系进行全局度量，不仅考虑直接相邻工作流事件之间的关系，还考虑非直接相邻工作流事件之间的关系。该定义基于下面的假设：如果任务 A 发生后不久，任务 B 也发生，那么可认为 A 是导致 B 发生的原因。另一方面，如果存在 B 在 A 的前面发生的记录，那么认为 A 是导致 B 发生原因的理由就不充足。

在流程实例中，任务 A 在任务 B 前面发生，N 是它们之

间工作流事件的总数，那么 A 和 B 之间的全局依赖关系的强度 $\$\ A\rightarrow B$ 可视为因子 δ 的指数函数 $(\delta)^n$，δ 取值范围在[0，1]之间。根据实验数据，δ 取 0.8 较好。当任务 B 直接发生在任务 A 的后面，若二者之间没有其它工作流事件，$N=0$，$\$\ A\rightarrow B$ 值最大取为 1；随着 N 增加，依赖关系的逐渐减弱。对所有流程实例的日志进行这样的处理，用得到的值除以 min($\#A$，$\#B$)，就可以得出工作流事件 A 和 B 之间的全局依赖关系的强度。

3.2 推导依赖-频率图

一般说来，没有必要为每一对工作流事件建立一个规则来判断二者之间是否存在因果关系。每一个非初始事件必然有一个导致其发生的事件；每一个非最终的事件必然要有一个依赖其的事件。利用这些有用信息，可大大缩小查找范围。对于两个工作流事件 X 和 Y 依赖关系(记为 $DE(X,Y)$)的启发式定义如下：

X 和 Y 是工作流事件，那么两者之间依赖关系记为：

$DE(X,Y)=((\$\ X\rightarrow^{L}Y)^2+(\$\ X\rightarrow^{L}Y)^2)/2$(规则 1)

其中，$A\rightarrow X$ 表示 $DE(A,X)$ 为最大的工作流事件是 X；$Y\rightarrow A$ 表示 $DE(Y,A)$ 为最大的工作流事件是 Y。在工作流日志上利用规则 1 可以导出图 2。我们比较下面的依赖-频率图，发现节点间的所有联接都同前面图 1 中的模型定义相吻合，所有节点间的联接都是正确的，且没有遗漏。

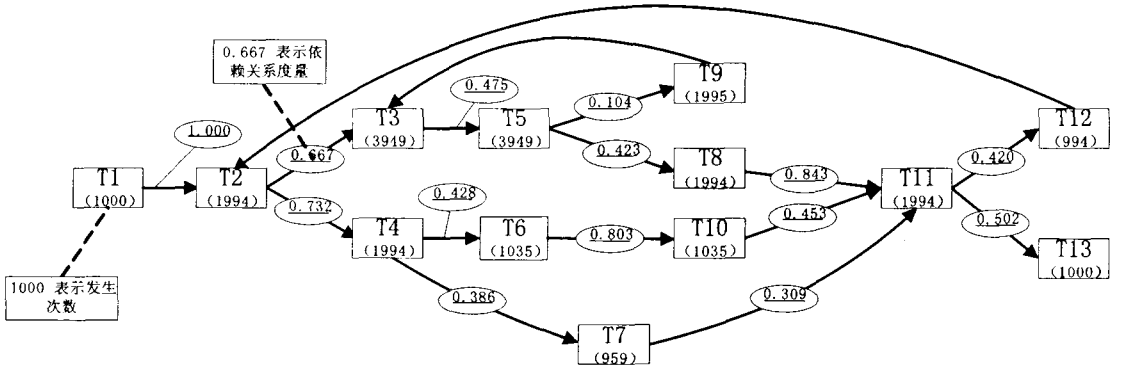

图 2 依赖-频率图

但是规则 1 还存在着不足，不能够识别出所有可能存在的依赖关系，如在图 3 中所列出的依赖关系。

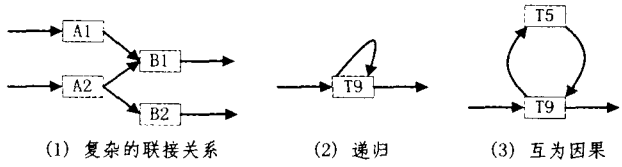

图 3 (1)复杂的联接关系；(2)递归；(3)互为因果

对于(1)中的复杂联接关系，应用规则 1，可能会丢失依赖关系。例如：如果在 $A1$ 上运用规则 1，导出 $A1\rightarrow B1$；在 $A2$ 上运用规则 1，得出 $A2\rightarrow B2$；同理，在 $B1$ 上可导出 $A1\rightarrow B1$，在 $B2$ 上可导出 $A2\rightarrow B2$。但依赖关系 $A2\rightarrow B1$ 没有被导出。基于这个原因，我们对规则 1 做一点修正。

规则 1(修正版)：对于一个任务 A，若 X 是使 $DE(A,X)$ 为最大(M)的事件，当且仅当 $DE(A,Y)>0.95*M$，可以得出 $A\rightarrow Y$。此外，假设 X 是 $DE(X,A)$ 值为最大(M)的事件，那么当且仅当 $DE(Y,A)>0.95*M$，我们就可以得出 $Y\rightarrow A$。在这里引入一个因子，其值 0.95，这个因子的引入是基于日志中的有噪声和并行流程情况的考虑。该因子取值的调整意义不大，0.95 的缺省值就很好了。

对于图 3 中另外两种关系：(2)递归和(3)互为因果。递归关系可以通过在日志中观察 $\#T9>T9$ 关系的高频出现，再结合 $DE(T9,T9)$ 的值，通常情况下 $DE(A,A)$ 的值约为 0，而且 $>A$ 也接近 0。

对于 $T9$ 和 $T5$ 之间互为因果关系的情况，将会导致在日志中出现模式{$T5,T4,T9,T5,T9,T6,T5,T8$}。对于这种互为因果的情况，理论上可以通过观察到高且相等的频率 $T5>T9$ 和 $T9<T5$ 出现，同时在依赖关系强度 $DE(T5,T9)$ 和 $DE(T9,T5)$ 值都接近 0；但是同样的日志记录在 $T5$ 和 $T9$ 是事件 X 的并发分支时也会出现，故修订后的规则 1 对于(3)中互为因果的情况尚不能处理。

3.3 从依赖-频率图中生成工作流网

从工作流日志中得到了相应的依赖-频率图，但分支、汇聚的类型在图中还没有得到体现。根据依赖-频率表的信息，再结合依赖-频率图中的节点频率，最终可以判断汇聚/分支的类型。例如：要检测从任务 A 到任务 B 和 C 的分支类型是“与”还是“或”，我们可以查看依赖-频率表中的 $B>C$ 和 $B<C$ 的值。若是“与”分支，也即是存在着关系 $B//C$，那么 $B>C$ 和 $B<C$ 这两种情况都会出现。若是“或”分支，那么 $B>C$ 和 $C>B$ 的情况都不会出现。

查看图 2 中节点的频率，可以推出 $T2$ 是一个“与”分支(因为 $\#T4=\#T2$)，$T5$ 是一个“或”分支(因为 $\#T5=\#T8+\#T9$)。类似的情况，还有 $T4$ 和 $T11$。在 $T11$ 上的汇聚判断有一点复杂：它表现为一个在 $T7$ 和 $T10$ 之间的一个“或”类型汇聚，和一个 $T8$ 的“与”类型汇聚($\#T7+\#T10=\#T8=\#T11$)。

通过重构算法，可以导出工作流事件之间的依赖关系，附之以频率检查，就能够决定分支/汇聚的类型。在这些收集的信息基础上，理论上就完全可以重构完整的业务流程模型。

结论 在本文中，介绍了工作流重构技术对于保障工作

一种乐观嵌套工作流事务模型

董云卫[1] 郝克刚[2]

(西北工业大学计算机学院 西安 710072)[1] (西北大学计算机科学系 西安 710069)[2]

摘 要 为解决目前工作流事务管理的不足和存在的问题,本文基于多数据版本、三阶段执行的并发控制理论提出了一种乐观嵌套工作流事务模型,该事务模型借用时间戳的概念,通过对不同事务中活动类型的分类,较好地解决了长执行事务和协同事务的可靠性和正确性问题,提高了工作流处理的效率。乐观嵌套事务模型把嵌套事务、工作流模型和并发控制协议有机地结合在一起,定义了较为完整的事务操作原语及其语义。本文还给出了乐观事务模型到工作流模型的映射,使得事务工作流执行过程中,其操作原语和乐观事务模型的操作原语是一致的,工作流活动的转移控制与乐观嵌套事务模型的子嵌套事务的生成过程及其表示方式也是一致的。

关键词 乐观嵌套事务模型,工作流事务,扩展信牌驱动分布式工作流计算模型

An Optimistic Nested Transaction Model in Workflow

DONG Yun-Wei[1] HAO Ke-Gang[2]

(School of Computer Science, Northwest Polytechnic University, Xi'an710072)[1]

(Department of Computer Science, Northwest University, Xi'an710069)[2]

Abstract An optimistic nested transaction workflow model is presented to solve some problem which is existed among workflow transaction management in this paper. The optimistic nested transaction workflow model adopts many technologies, such as multi-data version, three-phase concurrent control protocol and timestamp. It sorts workflow activities to assure reliability and correctness of long-running transaction and/or collaborative transaction, and to get a high level efficiency of system ability of computing. The optimistic nested transaction workflow mode defines completely primitives and semantic transaction operator. A mapping is also presented from optimistic nested transaction workflow mode to Extend Xinpai driven Distributed Workflow Model in this paper. The optimistic nested transaction workflow model is consistent with Extend Xinpai driven Distributed Workflow Model in workflow process running-time and building-time.

Keywords Optimistic nested transaction model, Workflow transaction, Extend xinpai driven distributed workflow model

1 前言

事务处理的概念起源于数据库系统,在以数据为中心的系统中,事务作为操作的基本单元必须满足 ACID 属性,即原子性(Atomicity)、一致性(Consistency)、孤立性(Isolation)和持久性(Durability)。而工作流事务管理是以过程为中心,保证工作流应用的可靠性和正确性。早期工作流建模方法是通过放松事务的基本特性来实现的,这样的工作流事务模型通常称为扩展事务模型或高级事务模型。

在工作流事务模型设计中,要考虑的问题一般要分两个层次:一是实现每一个操作的正确性和可靠性,这是数据库系统中关注的主题;二是实现每一个活动中许多操作之间流的管理。因此,表示过程活动执行的事务与传统的数据库事务具有显著不同的特点是[1~4]:

1)长时间执行:活动事务的执行时间长,事务执行的周期和工作量在许多情况下难以估计,可能几天、几月,甚至数年,称为长执行事务(long running transaction);

2)协同性:工作的事务中的活动执行的结果互相影响,并且,这些活动可能分布在异构环境中不同的系统中,称为协同事务(Collaboration transaction)。

3)可见性:中间和最终结果为多个合作者间共享和交换,是工作流事务必需的。它常常要求未提交的数据提前开放,并同时保证数据的一致性和正确性。

在工作流应用中,人们为了满足工作流事务管理的需要,

董云卫 博士,副教授,研究兴趣:工作流技术、面向方面的开发方法、软件测试技术。

流技术成功推广应用的重要意义。通过引入工作流网,工作流日志,工作流事件排序关系等相关概念,提出了一种基于工作流日志工作流重构算法。该算法能够处理干扰数据,因而能适应于大多数实际的工作流。重构算法包括:构建依赖-频率表;推导依赖-频率图;工作流网构建三个部分。在实验中,该算法能够对大多数业务流程的进行重构。当然,该算法仍有局限性,特别在识别工作流模型中复杂模式时,仍存在不完整性,有待进一步的研究。

参 考 文 献

1 Baldan P. Modelling concurrent computations: From contextual Petri nets to graph grammars:[Ph. D. thesis. TD-1/100]. Dipartimento di Infomatica University of Pisa, 2000

2 Weijters T, van der Aalst W M P. Process Mining: Discovering Workflow Models from Event-Based Data. In: Krse, B. et. al, eds. Proc. 13th Belgium-Netherlands Conf. on Artificial Intelligence (BNAIC01), 25-26 October 2001, Amsterdam, The Netherlands, 2001. 283~290

3 van der Aalst W M P, Desel J, Oberweis A, eds. Business Process Management: Models. Techniques, and Empirical Studies, volume 1806 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2000

4 Herbst J. A Machine Learning Approach to Workflow Management. In: 11th European Conf. on Machine Learning, volume 1810 of Lecture Notes in Computer Science, Springer, Berlin, Germany, 2000. 183~194