

## 工作流中事务处理--小组阅读报告

### ◆ 论文题目汇总：

吴晓彤	Modeling Long-Lived Health Care Workflow Transactions
许鑫禹	Combining dynamic Workflow and transactional semantics using a pattern-based approach
罗璇	Petri Net Modeling and Verification of Transactional Workflows
汪欣言	Transaction in Document Approval Process Workflow
王成成	Application Research of Transaction Properties In Migrating Workflow
税泽敏	Execution Recovery in Transactional Composite Service

### ◆ 问题描述：

- 1.如何使用基于模型的方法来结合动态工作流和事务性语义；
- 2.了解事务模型与工作流模型的关系；
- 3.分析事务工作流的概念及需要满足的四个基本属性；
- 4.利用 petri 网对事务工作流进行建模与验证；
- 5.有关医疗问题与文件审批问题的事务模型建立。

### ◆ 现实背景：

工作流技术在商业建模中的使用一直都很广泛，近年来在非传统的应用中，事务模型被使用的越来越多。网络服务是模块化和自述性的软件应用，这就会导致当根据用户需求添加新的网络服务的不可靠性以及不灵活性，因此有效结合动态工作流和事务性语义来确保网络服务的可靠和灵活显得十分有必要。但与此同时，在多样化和松散耦合的环境中，web 组合服务在实际执行过程中会发生某些服务失效，为保证用户的设备在端到端的工作流的服务质量，需要完善的事务处理与恢复机制。

### ◆ 研究现状分析：

#### 1.如何使用基于模型的方法来结合动态工作流和事务性语义：

将所有网络服务抽象成一个简单的事务并找到相应实体；然后根据这些实体间的描述来规定实体间的约束；通过给这些实体规定一个入口（in）和出口（out），并将一些可并行处理的实体聚合在二者之间。

## 2.了解事务模型与 workflow 模型的关系：

提出事务模型是 workflow 的一个子集这样的观点。

## 3.分析事务 workflow 的概念及需要满足的四个基本属性：

首先，事务实现的四个属性为：原子性、一致性、独立性、耐用性。将事务属性添加到数据库系统中增加了系统的可靠性和稳定性，然而，随着动态、异构、自治以及分布式复杂应用的发展，传统的事务模型很难再满足这些需求，因此人们设计了一系列更高级的事务模型，随着这些先进的事务模型的出现，事务 workflow 也随之发展而来。而随着 workflow 应用的不断复杂，存在与 workflow 系统中的分布和异构决定了 workflow 事务的属性以及确保了这些特征的难度。

## 4.利用 petri 网对事务 workflow 进行建模与验证：

用 petri 网对事务 workflow 建模，并且满足 LTL 公式：

大方向有两种情况

1、transactional properties：比如 pivot 和 retrievable

假设 petri 网结构可保证一致的行为，我们用可被事务 workflow 检查的 LTL 公式表达我们所需的行为。

2、transactional dependencies：比如 alternative 和 cancellation

每个限制（structural 和 semantic）被表达为 LTL 公式。

分为四种处理情况：

1、Retriable dependency

假设一个活动 A，失败的时候可以重新激活直到成功。

2、Pivot activity

假设一个活动 A，当它被激活，则以后都不会再激活。

3、Cancellation dependency

假设活动 A 和 A'是对称的，A 被取消，则 A'被激活，A 和 A'应该是并行的。

4、Alternative dependency

对于 Alternative dependency 有两种恢复方法：

A 为修改前，A'为 A 的修改，如果 A 失败则 A' 被触发。

1、Forward recovery：A'总在 A 后。

2、Backward recovery dependency: A'总在 A 前。

## 5.有关医疗问题与文件审批问题的事务模型建立:

### 医疗问题:

引用 Liu 的理论, 规定一系列的任务。

- ① **Forcible:** 最终会成功。
- ② **Critical:** 如果事务管理系统提交, 则必须被成功提交。
- ③ **Compensatable:** 当 workflow 废弃时, 事件可以未完成。
- ④ **Undo-not-required:** 任务提交时, 事件处理被废弃, 不需要被逆转。

规定完整性准则:

sequence, parallel split, exclusive choice, simple merge, multi-choice, synchronising merge, and arbitrary cycles,

规定 TWC 协议:

- ① 一个 TWF 可以提交, 如果 workflow 中所有 critical 任务已经提交了。
- ② 一个 TWF 可以抛弃, 如果所有 undo-required 任务已经偿还。
- ③ 一个 TWF 不可以抛弃, 如果有一个 critical non-compensatable 任务已经被执行。
- ④ 一个 TWF 可以被称作是强制性的, 如果所有有 critical 任务在 workflow 中都是带有强制性的。

规定准则:

- ① 序列: 当 A 完成任务时能使得 B 开始执行, 则称两个活动 A 和 B 是有序的。
- ② 并发: 任务 A 从一个单一的执行路径中分离, 并进入多个并发执行的路径。
- ③ 排除选择: 当从多个可选执行路径中选择一个路径时, 任务 A 指向执行路径。
- ④ 简单合并: 任务 A 有很多执行路径, 但是只有一条路径时活跃的, 这时将它们合并为一条简单执行路径。
- ⑤ 多选: 任务 A 可以从一个简单执行路径分离为多条可以并发执行的路径。
- ⑥ 同步合并: 当一些可能并发执行的路径汇入一条路径时, 任务 A 合并多条执行路径, 并在接下来的执行中等待所有可能活跃的入口。
- ⑦ 任意循环: 一系列能够形成循环的活动不断地执行。

### 文件审批问题:

① 对 BS 模式下的 DAWF 模型进行详细的了解。从对文件审批流程图的解释入手, 解释整个流程中的各活动、活动的执行顺序与控制关系及数据交流, 再分别从客户端与服务端的角度阐释文件审批的流程。

② 分析 workflow 模型与事务模型之间的关系与区别, 提出事务模型是 workflow 的一个子集这样的观点。

③ 提出 DAWF 中事务处理的规则, 设计了 DAWF 的事务处理模式。

④ 对 DAWF 的数据库进行进一步的设计, 画出 ER 图并进行阐述。

## ◆ 结论和展望：

六篇论文都对提出的相应问题作出了恰当的解决，在阅读的过程中，通过对论文的单  
理解与一些资料查阅，我们小组的同学对工作流中事务处理这一领域有了较为深入的认识。  
很多学者在这方面的研究都旨在提高对工作流事务的理解层次，再以此为基础进行更加实际  
的设计，不难看出事务处理模型正在不断的被提出并优化，在解决企业工程、IT 体系及审  
批等问题上都做出了极大的优化贡献。这也必将成为未来处理事务的一大良好趋势。