

에러복구 자동화를 위한 협조 트랜잭션 의존성 분석 메카니즘

김경숙, 정병수
경희대학교, 전자계산공학과
kskim@jupiter.kyunghee.ac.kr jeong@khu.ac.kr

A Dependency Analysis Mechanism For Error Recovery Automation in Coordinative Transaction Model

Kyoung-Sook Kim⁰ Byeong-Soo Jeong
Dept. of Computer Engineering, Kyung-Hee University

요 약

최근에 ACID를 특징으로 하는 기존의 전통적인 데이터베이스 트랜잭션 모델을 확장함으로써 데이터베이스 트랜잭션 모델의 응용분야를 다양하게 확대시키고자 하는 고급 트랜잭션 모델 (Advance Transaction Models) 들에 관한 많은 연구가 진행되고 있다. 본 논문에서는 다양한 유형의 트랜잭션들로 구성된 사무업무 프로세스 (Business Process)의 처리를 자동화하는데 효과적으로 활용될 수 있는 새로운 유형의 고급 트랜잭션 모델로서 협조 트랜잭션 모델을 정의하고, 본 데이터베이스 협조 트랜잭션 모델에서의 에러 복구 자동화를 구현하기 위한 협조 트랜잭션의 의존성 분석 메카니즘을 제안한다.

1. 서론

컴퓨터 기술과 전자통신 기술의 급진적인 발전과 이들 간의 기술적 수렴은 바로 전자적인 작업환경 (Electronic Workplace)이라고 하는 새롭고도 매우 효율적인 상호 작용 지원 수단 및 방법을 잉태하였다. 전자적인 작업환경이란 전(全) 조직체적 통합 시스템으로 정보 처리 활동과 정보 통신 활동의 통합을 통해 조직체 구성원들간의 상호 활동 및 관계를 정의하고 지원하는 개선된 형태의 조직 활동 지원 수단 및 방법이다[1]. 특히, 기존의 전통적인 데이터 처리 활동과는 달리 이러한 전자적 작업환경에서는 초대형집적회로 설계 분야나 CAD/CAM 프로젝트 분야, 사무업무 자동화 분야, 그리고 소프트웨어 개발 분야 등과 같은 다양한 유형의 공동작업이나 협업 기술 응용 분야들을 통합적으로 지원될 수 있어야 한다. 또한, 이러한 전자적 작업환경을 구현하기 위해서는 데이터베이스를 기반으로 하는 트랜잭션 처리 기술의 안정적 지원을 반드시 필요로 한다. 그러나 전통적인 데이터베이스 트랜잭션 처리 기술은 이러한 다양한 유형의 응용 분야들을 효율적으로 지원하는 데 기술적 한계를 가지고 있다. 즉, ACID (atomicity, consistency, isolation, durability)를 특성으로 하는 기존의 전통적인 트랜잭션 처리 모델은 장시간을 요하는 트랜잭션 (Long-lived Transaction Activity) 이나 종료가 불명확한 트랜잭션 (Open-ended Transaction Activity) 그리고 상호 협업을 요하는 트랜잭션 (Cooperative Transaction Activity) 등을 특징으로 하는 초대형집적회로 설계 분야나 CAD/CAM 프로젝트 분야, 사무업무 자동화 분야, 그리고 소프트웨어 개발 분야 등과 같은 새로운 유형의 데이터베이스 트랜잭션 응용 분야에 대해서 일관성 있는 서비스를 제공해줄 수 없다[4].

따라서, 전통적인 트랜잭션 처리 모델이 갖는 이러한 이론적/기술적 한계를 극복하고자 최근에 많은 수의 개선된 고급 트랜잭션 모델 (Advanced Transaction Models) 들이 제안되어 왔다. Nested Transaction Model, Multi-Transaction Model, STDL (structured transaction definition language), SAGAS, Split-and-Join Transaction Model, Flex Transaction Model, ATM (activity transaction model), ACTA, ConTract Model, Cooperative Transaction Model 등이 현재까지 제안되어 온 대표적인 고급 트랜잭션 모델의 예이다[5,6,8].

본 논문에서는 데이터베이스 트랜잭션 처리 기술을 사무업무 자동화 분야에 적용할 수 있도록 하기 위해서 기존의 전통적인 트랜잭션 모델을 확장하여 새로운 고급 트랜잭션 모델을 정의하는데 그 목적을 두고자 한다. 즉, 여러 가지 다양한 유형의 트랜잭션들로 구성되는 사무업무 프로세스 (Business Process)의 처리를 자동화하기 위한 이론적 기반이 되는 새로운 고급 트랜잭션 모델을 제안하고, 이를 협조 트랜잭션 모델이라고 정의하였다[2,3]. 협조 트랜잭션 모델은 일련의 단위업무 트랜잭션들로 구성되는 사무업무 프로세스를 정의하고 분석하는데 이용할 수 있을 뿐만 아니라 각 단위업무 트랜잭션들의 실행과 그들 간의 실행 순서 및 흐름을 제어하는 협조 트랜잭션 처리 모니터 시스템을 설계하고 구현하기 위한 이론적 기초가 된다. 그리고 본 논문에서 제안된 협조 트랜잭션 모델은 트랜잭션 제어 넷 (Transaction Control Net) 이라는 모델로 정의하는데, 이 모델은 그래픽적 방법과 정형적 방법으로 표현될 수 있도록 하였다.

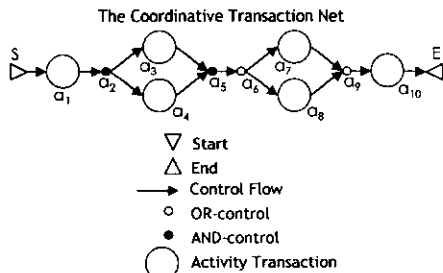
기존의 전통적인 사무업무 프로세스는 사용자 중심의 단위 업무들로 구성된 일련의 업무 프로시저를 정의하고 처리하므로 그의 주요 응용 영역은 주로 사무 양식 처리 중심의 사무업무를 바탕으로 하고 있다. 하지만 협조 트랜잭션 모델에서 대상으로 하는 고급의 사무업무 프로세스는 사용자의 개입이 필요없는 프로그램 또는 트랜잭션들로 정의되므로 데이터베이스 트랜잭션 처리 모니터를 중심으로 하는 응용 영역을 바탕으로 한다. 따라서, 협조 트랜잭션 모델에 의한 사무업무 프로세스는 기존의 전통적인 사무업무 프로세스와는 달리 시스템 측면의 많은 문제들을 해결해야 한다[1]. 그 중의 중요한 연구 이슈는 단위업무 트랜잭션의 에러 복구와 보상 트랜잭션의 정의 및 처리를 위한 메커니즘과 동시에 수행되어야 하는 여러 단위 트랜잭션 업무들의 데이터 공유로 인한 트랜잭션 처리의 비일관성 문제 등이 있다. 본 논문에서는 이와 같은 협조 트랜잭션 모델에서 직면하고 있는 여러 문제들 중에서 가장 중요한 에러 복구 자동화 문제를 해결하기 위한 트랜잭션 의존성 분석 메커니즘을 제안하였다[7,9].

본 논문의 구성은 다음과 같다. 제2장에서는 협조 트랜잭션 모델을 그래픽적 및 정형적으로 표현하기 위하여 트랜잭션 제어 넷을 정의하며, 제3장에서는 협조 트랜잭션 모델의 단위업무 트랜잭션들간의 실행 제어 및 흐름에 대한 분석을 통한 상호 의존성 분석 메커니즘을 제안하였다. 끝으로, 본 논문에서 제안한 협조 트랜잭션 의존성 분석 메커니즘이 트랜잭션 에러 복구 자동화 문제점들을 해결하는데 어떻게 이용되는가를 간단히 기술하였다.

2. 협조 트랜잭션 모델

본 장에서는 협조 트랜잭션 모델을 정의하고 분석하기 위해 이용되는 협조 트랜잭션 제어 넷 (Coordinative Transaction Net) 모델을 그래픽적 표현 방법과 정형적인 방법으로 정의하고, 이를 바탕으로 단위업무 트랜잭션 (Activity Transaction)들간의 제어 의존성을 분석하게 된다. 트랜잭션 제어 넷은 사무업무 프로세스의 개념을 일련의 관련된 단위업무 트랜잭션의 집합으로 정의하며, 이들간에는 처리순서상의 선후관계가 존재하게 되고, 이는 일련의 집합으로 표현된다. 트랜잭션 제어 넷은 그래픽적으로 프로세스, 트랜잭션, 선후관계를 나타내는 실행흐름 (Control Flow)을 표현한다[9].

2.1 협조 트랜잭션 제어 넷



<그림 1> 협조 트랜잭션 제어 넷의 구성 요소

<그림 1>에서 나타내었듯이 협조 트랜잭션 제어 넷은 큰 원으로 표현되는 일련의 단위업무 트랜잭션과 작고 빈 원으로 표현되는 OR, 채워진 원으로 표현되는 AND, 그리고 이러한 노드들을 연결하는 화살표(Arc)로 구성된다.

화살표(Arc)는 실선(Solid)과 점선(Dashed)으로 표현되는데 이들은 각각 노드들 간의 선후관계 및 자료저장소와의 입/출력을 표현한다.

3.2 트랜잭션 제어 넷의 정형적 표현

트랜잭션 제어 넷에 대한 정형적 표현은 4 개의 구성요소인 프로세스, 트랜잭션, 선후관계(Precedence)와 자료저장소로 구성되며, A 를 일련의 트랜잭션들의 집합이라고 하고, R 을 일련의 자료 저장소의 집합이라 할 때 정형적인 표현과 정의는 다음과 같다.

<표 1> 트랜잭션 제어 넷 모델의 정형적 표현

$\Gamma = (\delta, \gamma, I, O)$
δ : precedence constraint among transactions
γ : repository input/output of transactions
I: initial input repository
O: final output repository

① I는 초기에 입력되는 자료저장소들의 유한 집합이며, 트랜잭션 제어 넷의 실행 전에 어떠한 외부의 프로세스에 의해서 로드(Load)되어야 한다고 가정한다.

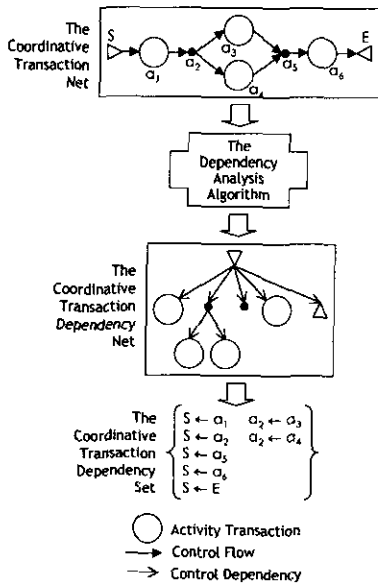
② O는 마지막으로 출력되는 자료저장소들의 유한 집합이며 트랜잭션 제어 넷의 실행 후에 어떠한 외부의 프로세스에 의해서 이용되는 정보를 포함하고 있다고 가정한다.

③ $\delta = \delta_1 \cup \delta_2$
여기서 $\delta_1: A \rightarrow \wp(A)$ 은 하나의 트랜잭션이 후행하는 트랜잭션들의 집합에 연결하는 관계를 나타내며 $\delta_2: A \rightarrow \wp(A)$ 은 하나의 트랜잭션을 선행하는 트랜잭션들의 집합에 연결하는 관계를 나타낸다.

④ $\gamma = \gamma_1 \cup \gamma_2$
여기서 $\gamma_1: A \rightarrow \wp(R)$ 은 하나의 트랜잭션을 후속하는 트랜잭션 집합들을 출력 자료저장소들의 집합과 연결하는 것 중 하나이며 $\gamma_2: A \rightarrow \wp(R)$ 은 하나의 트랜잭션을 선행하는 트랜잭션 집합들을 입력 자료저장소들의 집합과 연결하는 관계를 나타내는 것 중 하나이다.

3. 협조 트랜잭션 의존성 분석 메커니즘

본 장에서는 협조 트랜잭션 모델을 구성하고 있는 단위업무 트랜잭션들간의 제어 의존성을 분석하기 위한 트랜잭션 의존성 분석 메커니즘을 정의한다. <그림 2>는 본 의존성 분석 메커니즘의 개략적인 개념을 도식화한 것이다. 그림에서 나타내었듯이, 본 논문의 제어 의존성 분석 알고리즘은 협조 트랜잭션 제어 넷 (The Coordinative Transaction Net)으로 정의된 어느 한 협조 트랜잭션 모델상의 일련의 단위업무 트랜잭션들간에 존재한 제어의 흐름을 분석하여 협조 트랜잭션 제어 의존 넷(The Coordinative Transaction Dependency Net)을 생성한다. 생성된 제어 의존 넷을 바탕으로 협조 트랜잭션 처리 모니터에서 에러 복구를 자동화적으로 처리하기 위해 필요로 되는 일련의 제어 의존 정보(The Coordinative Transaction Dependency Set)를 생성하는 것이다[7].



<그림 2> 협조 트랜잭션 의존성 분석 메카니즘

① 협조 트랜잭션 의존 넷

협조 트랜잭션 의존 넷은 다음의 <표 2>와 같이 정형적인 방법으로 정의한다.

<표 2> 협업 트랜잭션 의존 넷의 정형적 표현

$\Omega = (\varphi, \xi, S, E)$ over a set A of activity-transactions and T of control-transition conditions

- $\varphi = \varphi_1 \cup \varphi_2$
- $\varphi_1: A \rightarrow \wp(A)$: A 에 제어의존되는 단위업무트랜잭션의 집합
- $\varphi_2: A \rightarrow \wp(A)$: A 가 제어의존하는 단위업무트랜잭션의 집합,
- $\xi = \xi_1 \cup \xi_2$
- ξ_1 : 제어선이 조건 집합, $\tau \in T$ on each arc, $(\varphi(\alpha), \alpha)$;
- ξ_2 : 제어선이 조건 집합, $\tau \in T$ on each arc, $(\alpha, \varphi_2(\alpha))$, where $\alpha \in A$;
- S is a finite set of initial control-transition conditions;
- E is a finite set of final control-transition conditions;

제어 의존 넷은 단위업무 트랜잭션들간의 제어 의존 관계를 정의하는데 이용되는데, 이는 어느 한 단위 트랜잭션이 에러로 인해 실행이 실패되었을 때, 실패된 트랜잭션에 대한 에러 복구 또는 보상의 범위를 결정하는데 이용된다. <그림 2>에서 나타내었듯이 본 제어 의존 넷은 최종적으로 일련의 협조 트랜잭션 의존 데이터 집합으로 변형되어 데이터베이스 트랜잭션 처리 모니터가 에러 발생(transaction abort)시의 Rollback 또는 보상 범위를 자동적으로 생성하는데 이용된다.

② 제어 의존성 분석 알고리즘

본 알고리즘<그림 3>은 협조 트랜잭션 넷의 정형적 표현을 입력받아 일련의 분석 규칙들(*strongly control-dependent*, *immediate forward dominator*, *immediate backward dominator*, *Walk* 등, 본 논문에서는 제한된 페이지로 인해 자세히 기술하지 않았음)을 적용하여 정형적으로 표현되는 협조 트랜잭션 제어 의존 넷을 생성하는 역할을 한다. 본 알고리즘의 복잡도는 $O(n^2)$ 이다. 왜냐하면, 단위업무 트랜잭션들간의 *strongly control-dependent* 관계를 분석하는데 $O(n^2)$ 이고, For-loop 에 $O(n^2)$ 의 시간이 요구되기 때문이다.

INPUT : A coordinative transaction net;
OUTPUT : A coordinative transaction dependency net;

BEGIN

FOR all $x \in A$ and $y \in A$ in a CTN
IF x is strongly control dependent on y
// OR or Loop Constructs
// Get a dependent flow_
ADD x TO $\varphi(y)$; ADD y TO $\varphi(x)$;
// Get control-transition conditions between y and x .
ADD $\kappa(x)$ TO $\xi(y)$; ADD $\kappa(x)$ TO $\xi(x)$;

ELSE IF $fd(y) \in x$ AND $(fd(y) \alpha fd(ibd(y))$ OR $ibd(y) = \emptyset$)

// Get a dependent flow_
ADD x TO $\varphi(y)$; ADD y TO $\varphi(x)$;
// Get control-transition conditions between y and x
ADD $\kappa(x)$ TO $\xi(y)$; ADD $\kappa(x)$ TO $\xi(x)$;

FI

END.

END.

<그림 3> 협조 트랜잭션 의존성 분석 알고리즘

4. 결론

본 논문에서는 전자상거래의 활성화와 더불어 최근에 많은 주목을 받고 있는 사무업무 프로세스 처리의 자동화 분야에 데이터베이스 트랜잭션 처리 기술을 확대 적용하기 위한 이론적 모델을 제시하고자 협조 트랜잭션 모델이라는 새로운 유형의 고급 트랜잭션 모델을 제안하였다. 그리고 제안된 협조 트랜잭션 모델에서의 트랜잭션 에러 복구 및 보상을 자동화시키기 위한 트랜잭션 의존성 분석 메카니즘을 설계하였다.

5. 참고문헌

- [1] Clarence A. Ellis and Gary J. Nutt, "Office Information Systems and Computer Science", Computing Surveys, Vol. 12, No. 1, March 1980
- [2] Clarence A. Ellis, "Goal Based Models of Groupware", University of Colorado, Boulder, Colorado, USA
- [3] Kwang-Hoon Kim and Su-Ki Paik, "Actor-Oriented Workflow Model", The Second Cooperative Database Systems for Advanced Applications, Wollongong Australia, March 1999
- [4] A. Elmagarmid, editor. "Transaction Models for Advanced Database Applications," Morgan-Kaufmann, 1992
- [5] Georgakopoulos, D. and Hornick M. and Sheth A. "An Overview of Workflow Management: From Process Modelling to Workflow Automation Infrastructure," Distributed and Parallel Databases, 3(2):119-152, 1995
- [6] M. Kamath and K. Ramamritham. "Modeling, Correctness & Systems Issues in Supporting Advanced Database Applications using Workflow Management Systems," Technical Report TR 95-50, University of Massachusetts, Computer Science Dept., 1995
- [7] R. Obermarck, "Special Issue on TP Monitors and Distributed Transaction Management," Data Engineering, Vol. 17, No. 1, March 1994
- [8] C. Mohan, "Tutorial: Advanced Transaction Models - Survey and Critique," ACM SIGMOD International Conference on Management of Data, Minneapolis, May 1994
- [9] A.P. Buchmann, M.T. Ozsu, and Dimitrios G., "Towards a Transaction Management System for DOM," GTE Laboratories Incorporated Technical Report TR-0146-06-91-165, June 1991