

분산 DB 환경에서 동기화 이동에이전트

국 윤 규[†] · 김 운 용^{††} · 정 계 동^{†††} · 김 영 철^{††††} · 최 영 근^{†††††}

요 약

본 논문에서는 분산 환경에서 e-비즈니스 프로세스들 간에 SFT(Slow Frequency Transaction) 데이터를 처리하는 동기화 에이전트 시스템을 설계 및 구현한다. 이 동기화 에이전트 시스템은 분산 프로세스들의 상호운용성 보장과 클라이언트/서버 구조의 문제를 해결하기 위해 이동에이전트를 이용하며, 분산된 이종 시스템 플랫폼상의 e-비즈니스 프로세스들의 데이터 동기화를 위한 정책을 제시한다. 제안된 동기화 정책은 유니캐스팅 전송 상에서 시스템 우선순위와 타임스탬프의 트랜잭션을 기본으로 한다.

A synchronized mobile agent in Distributed Database Environments

Youn-Gyou Kook[†] · Woon-Yong Kim^{††} · Gye-Dong Jung^{†††}
R. Yung-Chul Kim^{††††} · Yung-Keun Choi^{†††††}

ABSTRACT

This paper design and implements a synchronized agent mechanism that be able to synchronize SFT (Slow Frequency Transaction) data among e-business processes in distributed environment. This synchronized agent system is adopted to use mobile agent to guarantee the interoperability of the distributed processes and to solve some problems at the client/server architecture and proposes a policy to synchronize data of e-business processes on the any platforms of distributed heterogeneous systems. The proposed synchronization policy is based on the time-stamp transaction with the system priority on the unicasting transportation.

키워드 : 동기화(synchronization), 이동에이전트(mobile agents), 분산데이터베이스(distributed database), 분산시스템(distributed system)

1. 서 론

컴퓨터와 정보 기술의 발달은 사회 환경을 급속하게 변화시키며 다양한 온라인과 오프라인 비즈니스를 위하여 통합 e-비즈니스 솔루션을 구축하여 생산성 향상과 경쟁력을 강화시키고 있다. B2C 형태의 전자상거래는 B2B로 확장되어 e-트레이딩의 일반화로 e-비즈니스에서 발생한 정보 공유와 활용이 중요한 이슈가 되고 있다. 인터넷과 e-비즈니스 솔루션들간의 상호운용 등을 위한 정보 공유 필요성이 확대되어 그에 따라 데이터 동기화가 필요하다[3, 7].

인터넷과 e-비즈니스 솔루션 환경은 이종 시스템으로 구성된 분산 환경으로서 데이터 동기화를 위해서는 독립성과 자율성을 보장해야 한다. 궁극적으로 데이터 동기화는 분산 운용되는 플랫폼에 독립적이고, 데이터 갱신과 삭제에 따른 자율성이 보장되며, 분산 환경에 따른 네트워크 문제를 고려해야 한다. 이러한 문제점들을 해결하기 위해, 네트워크 부

하와 트래픽을 감소시키며 원격지에서 수행할 실행코드와 데이터를 받아 로컬 시스템에서 업무를 수행하는 이동 에이전트를 이용하여 해결할 수 있다[1, 4, 8, 11].

따라서 본 논문에서는 분산된 이종 시스템 플랫폼상의 e-비즈니스 프로세스들의 동기화를 위해 동기화 정책을 제시하며, 동기화 정책을 바탕으로 이동에이전트를 이용하여 각 e-비즈니스 프로세스들의 상호운용성을 보장하고 네트워크 문제를 고려한 동기화 에이전트 시스템에 대하여 기술한다.

본 논문의 동기화 에이전트 시스템은 FFT(Fast Frequency Transaction) 환경이 아닌 SFT(Slow Frequency Transaction) 데이터 동기화를 위한 이동에이전트와 에이전트 서버로 구성하며, 동기화 정책에 따라 동기화를 실행한다. 에이전트는 목적지 에이전트 서버의 이름과 주소를 식별하여 이동하며, 에이전트 관리자와 모듈 관리자들과 통신을 한다. 동기화를 위해서 제시된 동기화 정책은 트랜잭션의 타임스탬프에 기반하며, 트랜잭션 완료 및 복귀와 충돌 회피를 위한 시스템 우선순위 등을 고려한다.

본 논문은 2장에서 이동 에이전트와 동기화에 관하여 살펴보고, 3장에서 동기화 에이전트 시스템 및 동기화 정책을 설명한다. 4장에서 구현 결과를 분석하고, 마지막으로 결론

† 준 회 원 : 광운대학교 대학원 컴퓨터학과
 †† 준 회 원 : 광운대학교 정보통신연구원
 ††† 정 회 원 : 광운대학교 정보과학교육원
 †††† 정 회 원 : 홍익대학교 컴퓨터정보통신학과 교수
 ††††† 정 회 원 : 광운대학교 컴퓨터과학과 교수
 논문접수 : 2003년 4월 14일, 심사완료 : 2003년 10월 1일

으로 구성된다.

2. 관련 연구

2.1 이동에이전트

이동에이전트 시스템은 '네트워크로 연결된 독립성이 높은 다수의 소프트웨어 프로세서(에이전트)가 대등한 관계를 유지하며 하나의 시스템처럼 기능을 수행하는 시스템'으로 정의할 수 있다. 이동에이전트는 주어진 작업을 완수하기 위하여 다른 서버에 전송되는 프로세스로서 소스 서버와 독립적이며 자율적으로 진행하는데, 새로운 에이전트를 생성하여 다른 에이전트 서버에 전송할 수 있거나 다른 에이전트와 상호작용 할 수 있다[6]. 이동에이전트가 목적지 에이전트 서버에 도착하여 인증 처리 후에 실행을 시작하고, 작업 완료한 이동에이전트는 전송한 소스 서버나 목적지 에이전트 서버에 그 결과를 전달한다.

이동에이전트 시스템은 플랫폼에 독립적이며 자율성을 지니고 있으므로 다양한 이종 시스템에 적용될 수 있으며, 상황에 따라 동적으로 변화가 가능하므로 유연성이 뛰어나 클라이언트/서버 시스템의 결점을 해결할 수 있는 분산처리 시스템의 대안으로 인식되고 있다. 따라서 이동 에이전트는 전자 상거래를 비롯하여 네트워크 관리, 정보 검색, 정보 보안, 시스템 통합 등 많은 분야에서 이동에이전트를 적용하기 위한 연구가 진행되고 있으며, 특히 분산환경에서 이동에이전트를 이용하여 데이터에 접근하고자 하는 연구가 활발히 진행되고 있다[2].

분산 이종 데이터에서 멀티미디어 데이터를 검색하기 위한 연구, B2B를 구축하기 위해 이동에이전트를 이용하는 연구, 웹에서 분산된 데이터 접근을 위한 연구, 분산된 기업에 적용할 수 있는 이동에이전트에 대하여 쿼리 패턴에 관한 연구 등이 있으며[1, 3, 8, 9], 본 논문에서는 이동에이전트를 이용하여 분산된 데이터들의 동기화에 관한 연구와 구현 예를 기술한다.

2.2 동기화 정책

분산 환경에서 이종 시스템간의 데이터 동기화를 위한 기술은 데이터 동기화 절차와 충돌 해결 방법에 의해 이종 시스템간의 데이터 일관성을 유지한다[3, 5, 10, 12]. 동기화 요구시, 변경된 데이터의 트랜잭션 처리 방법은 다음과 같이 구분할 수 있다. 첫 번째로 최초 데이터가 변경된 시스템의 트랜잭션을 먼저 처리한 후, 변경이 요구되는 시스템들에게 트랜잭션 처리를 요구하는 방법과 두 번째로 최초 데이터가 변경된 시스템의 트랜잭션을 임시 상태로 처리한 후, 데이터 변경이 요구되는 시스템들의 트랜잭션을 처리한 후 마지막으로 최초 변경된 시스템의 임시 상태인 트랜잭션을 영구적으로 처리하는 방법이다. 또한 데이터 변경이 요구되는 시점에서 모든 트랜잭션을 임시상태에서 영구적

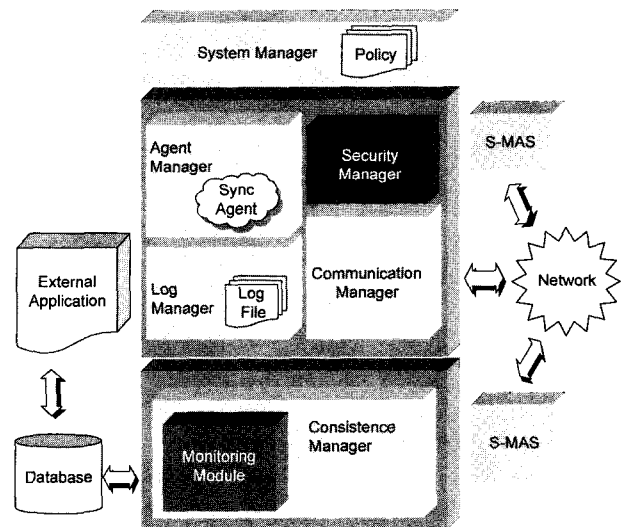
으로 일시에 변경하는 일괄처리 방법이 있다.

또한 데이터 동기화 트랜잭션 처리에 있어서 충돌 발생 여부를 감지해야 한다. 데이터 충돌에는 두 시스템이 동일한 튜플에 서로 다른 데이터 값을 삽입하는 삽입 충돌, 한 시스템은 튜플을 삭제하고 다른 시스템은 튜플을 갱신하는 삭제 충돌, 두 시스템이 동일한 데이터 필드에 서로 다른 데이터 값을 갱신하는 갱신 충돌이 있다. 따라서 본 논문에서는 이동에이전트의 이동 경로에 따라 유니캐스팅 전송을 하면서 트랜잭션 처리를 하는 선처리 기법을 이용하며, 충돌 탐지 및 해결을 위하여 타임스탬프와 시스템 우선순위를 이용한다. 또한 분산 환경에서 데이터 변경의 발생 빈도에 따라 데이터 동기화의 성능이 결정되므로, 본 논문에서는 SFT(Slow Frequency Transaction) 데이터에 대한 동기화 에이전트 시스템을 제안한다.

3. 동기화 이동 에이전트 시스템

3.1 시스템 구성 요소

동기화 에이전트 시스템의 특징은 플랫폼에 독립적이고, 이동이 원활하며, 자율적으로 동기화를 진행하며, (그림 1)에서와 같이 시스템 관리, 정책 조정 및 관리, 에이전트 관리 부분으로 구성된다.



(그림 1) 동기화 에이전트 시스템 구성요소

3.1.1 시스템 관리 모듈

시스템 관리에서는 동기화를 하고자 하는 목적지 시스템의 이동경로와 시스템 중요도에 따른 우선순위를 설정하고, 데이터에 접근하기 위한 기초적인 정보를 명시한다. 시스템 관리자는 동기화 에이전트의 이동경로를 위해 동기화를 하고자 하는 목적지 에이전트 서버들의 위치 정보와 SFT 데이터의 변경이 자주 발생하는 시스템을 위주로 우선순위를 설정해야 한다. 동기화 에이전트는 이동경로를 통해 유니캐

스팅 전송방법으로 이주를 하며, 트랜잭션 타임스탬프가 일치되어 데이터 충돌이 감지 될 경우에는 시스템 우선순위에 의해 트랜잭션을 처리한다. 마지막으로 동기화 에이전트 시스템이 데이터에 접근하기 위한 기초적인 정보를 명시해야 한다. 본 시스템이 Java로 이루어져 있기 때문에 데이터베이스에 접근하기 위하여 시스템 관리자는 JDBC(Java Database Connectivity)를 설정해야 한다. JDBC는 관계형 데이터베이스에 접근하고 조작하는 Java 표준 명세로서, 데이터베이스의 제공업체에 따라 JDBC를 설정해야 하며 데이터베이스간의 동기화가 이루어질 테이블의 정보를 명시해야 한다. 그리고 이동에이전트의 전송에 따른 보안관련 정책 설정은 에이전트 서버의 인증에 국한되며, 본 논문의 연구 방향의 일환으로서 지속적인 연구가 필요하다.

3.1.2 정책 조정 및 관리 모듈

정책에 대한 조정 및 관리는 동기화가 이루어질 시스템들간에 정책들을 조정하고, 동기화 에이전트 시스템에서 발생하는 모든 이벤트를 기록한다. 통신 관리자는 이동에이전트들의 이동경로에 해당되는 목적지 시스템들의 상태를 파악하며, 이동에이전트의 이주보장 기능을 지원한다. 또한 로그 관리자는 동기화 에이전트 시스템에서 발생하는 이동에이전트의 접근과 인증, 데이터 변경에 따른 에이전트 활성화 및 전송 등의 이벤트에 대한 로그를 기록한다. 보안 관리자는 원격지에서 전송된 이동에이전트의 인증을 통해 시스템의 접근에 따른 다른 등급 권한을 주기 위하여 보안 및 인증 부분이 설계되어야 하지만, 본 논문에서는 기본적인 보안 문제 해결만을 위해 Java에서 제공되는 보안 관리자를 구성하여 이동에이전트의 인증을 처리한다.

3.1.3 에이전트 관리 모듈

동기화 에이전트 시스템의 설정된 정책을 관리하고, 원격지에서 전송되는 에이전트의 인증 처리 및 데이터 동기화에 관련된 에이전트들을 관리한다. 에이전트 관리자는 원격지에서 전송된 이동에이전트의 인증을 보안 관리자에게 요구하며, 동기화가 필요한 데이터의 충돌을 피하기 위하여 일관성 관리자와의 통신을 통하여 시스템 접근여부에 대한 결정을 한다. 또한 데이터 변경으로 인한 일관성 관리자의 요청에 따라 주어진 이동경로를 통해 동기화 에이전트를 전송하고, 시스템에서 발생한 이벤트에 대한 기록을 로그 관리자에게 요청한다. 일관성 관리자는 데이터에 관한 이벤트가 발생하면 에이전트 관리자에게 이벤트 발생에 대한 정보를 전달하며 동기화 에이전트 활성을 요청하며, 전송된 동기화 에이전트에게서 요구받은 데이터의 충돌 여부를 판단하여 데이터베이스 접근 여부를 결정하고, 이에 대한 결과를 에이전트 관리자에게 전달한다. 데이터 충돌의 경우에는 시스템 우선순위 정책에 따라 높은 우선순위의 데이터를 선택하게 된다. 동기화 에이전트는 데이터 변경에 따른

이벤트 발생으로 일관성 관리자가 정보를 전달하면 에이전트 관리자에 의해 활성화되어 이동경로에 따라 목적지 시스템으로 전송된다. 또한 목적지 시스템에서 인증을 받고 데이터 충돌 여부를 검사 받으며, 동기화를 실행한 이후 그 결과 값을 획득하여 이동경로에 따라 이동한다. 다음 <표 1>은 시스템 구성요소들의 역할을 정리한 것이다.

<표 1> 시스템 구성요소들의 역할

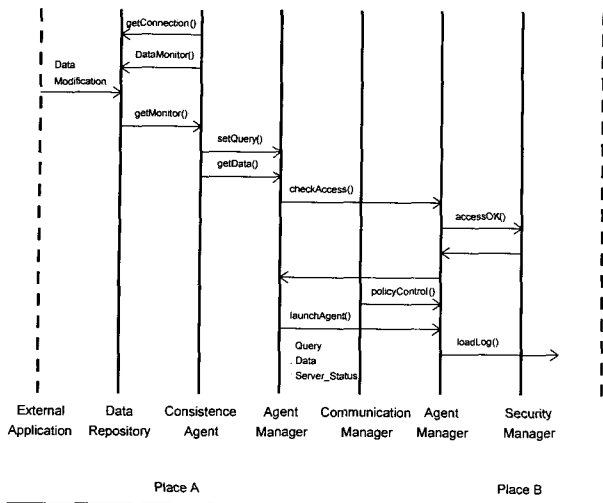
시스템 구성 요소		Role(역할)
시스템 관리 모듈	시스템 관리자	<ul style="list-style-type: none"> • 동기화 에이전트의 이동경로 설정 (Sync Agent Itinerary setting) • Policy 설정(System Priority Information Setting) • JDBC Information 설정
	통신 관리자	<ul style="list-style-type: none"> • 다음 이동경로의 system 상태를 시스템 관리자로부터 획득하여 Check • Sync Agent의 이주보장기능 지원
정책 조정 및 관리 모듈	로그 관리자	<ul style="list-style-type: none"> • 발생 이벤트에 대한 로그 기록 : 에이전트 관리자(에이전트 활성화/비활성), 통신 관리자(에이전트 이주), 보안 관리자(에이전트 인증)
	보안 관리자	<ul style="list-style-type: none"> • 원격지 Sync Sgent의 System 접근 권한 인증
	에이전트 관리자	<ul style="list-style-type: none"> • 보안 관리자에게 원격지 Sync Agent에 대한 인증요청 • Sync Sgent의 Life Dycle 조정 기능 • 동기화 에이전트 활성화/ 비활성화 처리 • System Log 기록
에이전트 관리 모듈	일관성 관리자	<ul style="list-style-type: none"> • Transction 유지/처리 기능 • 데이터 충돌시 정책 반영/조정 기능 • Database Event 감시기능(Monitoring Module : 데이터의 갱신/삽입/삭제)
	동기화 에이전트	<ul style="list-style-type: none"> • 데이터 동기화 작업 수행 • 결과값을 획득하여 유지
	에이전트 관리자	<ul style="list-style-type: none"> • 보안 관리자에게 원격지 Sync Agent에 대한 인증요청 • Sync Sgent의 Life Dycle 조정 기능 • 동기화 에이전트 활성화/ 비활성화 처리 • System Log 기록

3.2 시스템 설계

동기화 에이전트 시스템은 데이터 변경에 따른 에이전트 활성화와 전송을 위한 이주 메커니즘, 동기화를 위한 목적지 시스템으로의 이동 후에 에이전트의 인증과 실행 및 데이터 충돌 방지 등을 위한 동기화 메커니즘으로 구분된다.

3.2.1 동기화 에이전트 이주 메커니즘

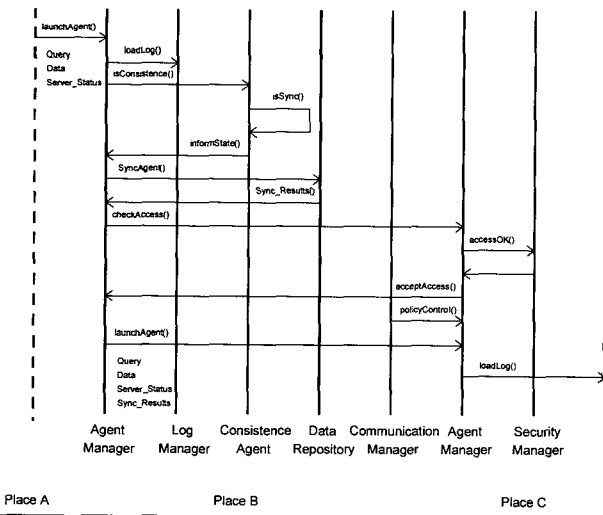
외부 애플리케이션에 의한 데이터 변경을 감지한 일관성 관리자는 에이전트 관리자에게 이벤트 발생에 대한 정보를 전달하며 동기화를 요구한다. 에이전트 관리자는 주어진 이동경로를 통해 동기화 에이전트를 전송하고, 시스템에서 발생한 이벤트에 대한 기록을 로그 관리자에게 요청한다. 목적지 시스템에서 보안 관리자에 의해 동기화 에이전트의 접근 허용을 인증 받고, 통신 관리자에게 정책을 전송한다. 동기화 에이전트는 데이터 변경에 따른 데이터 갱신, 삽입, 삭제 정보와 소스 시스템의 상태 정보 등과 함께 이주한다. (그림 2)는 이러한 동기화 에이전트 이주 메커니즘을 보여 준다.



(그림 2) 동기화 에이전트 이주 메커니즘

3.2.2 데이터 동기화 메커니즘

동기화 요구를 받은 목적지 에이전트 관리자는 요구에 대한 기록을 로그 관리자에게 요청하며, 일관성 관리자에게 데이터 충돌 여부에 대한 결과를 요구한다. 일관성 관리자는 데이터 충돌 여부 상태에 따라 동기화 에이전트의 실행을 결정하고, 그 결과를 에이전트 관리자에게 전달한다. 데이터 동기화를 실행한 에이전트 관리자는 다음 목적지 시스템에게 접근 인증을 부여받고 동기화 데이터, 서버 상태 및 동기화 결과를 함께 전송한다. (그림 3)은 데이터 동기화 메커니즘을 보여 주며, 데이터 동기화 기법은 다음 3.3 절에서 설명한다.



(그림 3) 데이터 동기화 메커니즘

3.3 데이터 동기화 기법

본 시스템의 동기화 정책은 트랜잭션의 타임스탬프에 기반하며, 트랜잭션 완료 및 복귀와 충돌 회피를 위한 시스템 우선순위 등을 고려한다. 외부 애플리케이션의 데이터 변경

에 따라, 일관성 관리자는 감지된 데이터 변경 이벤트에 대하여 에이전트 관리자에게 결과를 전송하고 동기화를 요구한다. 동기화 에이전트의 이동경로 및 트랜잭션 타임스탬프, 시스템 우선순위를 정의하면 다음과 같다.

$$N = \{S_1, S_2, S_3, \dots, S_n\} \quad n < \infty, S \text{ is a system node,}$$

N is a family of system nodes

$$TS = \{TS_1, TS_2, TS_3, \dots, TS_n\} \quad n < \infty,$$

TS is a transaction timestamp

$$P = \{P_1, P_2, P_3, \dots, P_n\} \quad n < \infty, P \text{ is a system priority}$$

$$T = \{T_1, T_2, T_3, \dots, T_n\} \quad n < \infty, T \text{ is a transactions of } N$$

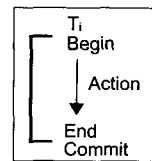
$$T_i = \{B, A, E, C\}, \text{ where } T_i \in S$$

B is a begin of transaction

A is a action of transaction

E is a end of transaction

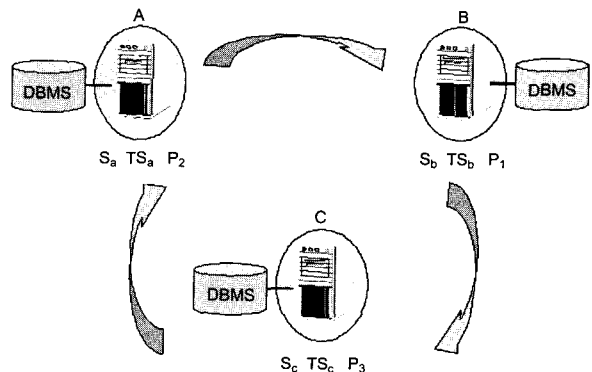
C is a commit of transaction



N 은 시스템 노드이고, TS 는 데이터 변경이 발생된 트랜잭션 타임스탬프, P 는 SFT 데이터 변경이 자주 발생하는 시스템을 위주로 우선순위가 설정된다. 트랜잭션 타임스탬프의 값에 따라 가장 최근에 발생한 트랜잭션을 처리하며, 트랜잭션 타임스탬프의 값이 일치하면 시스템 우선순위에 의하여 트랜잭션이 처리된다. 이를 도식화하면 다음과 같다.

- if $TS_1 > TS_2$ then $T = TS(1, 2) \Leftrightarrow Transaction(1)$
- if $TS_1 = TS_2$ then $T'' = TS(1|2) \Leftrightarrow P(1, 2)$

지금부터는 (그림 4)와 같은 간단한 동기화 에이전트 시스템을 적용 예로서 이용한다. 시스템 노드는 $\{S_a, S_b, S_c\}$ 로 하고, 트랜잭션 타임스탬프는 $\{TS_a, TS_b, TS_c\}$, 시스템 우선순위는 $\{P_1, P_2, P_3\}$ 이다. 시스템 우선순위는 S_b 의 P_1 , S_a 의 P_2 , S_c 의 P_3 순이다.



S_n : System Node
 TS_n : Transaction Timestamp
 P_i : Priority of System

(그림 4) 동기화 에이전트 시스템의 이동경로, 트랜잭션 타임스탬프, 시스템 우선순위

$$N = \{S_a, S_b, S_c\}$$

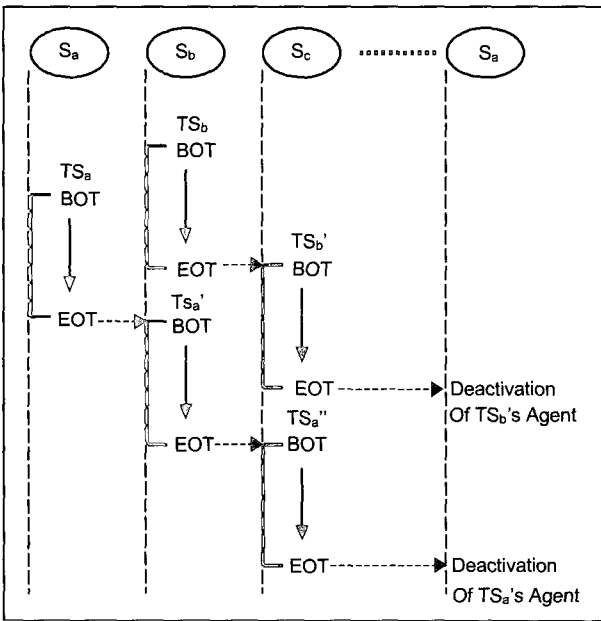
$$TS = \{TS_a, TS_b, TS_c\}$$

$$P = \{P_1, P_2, P_3\}$$

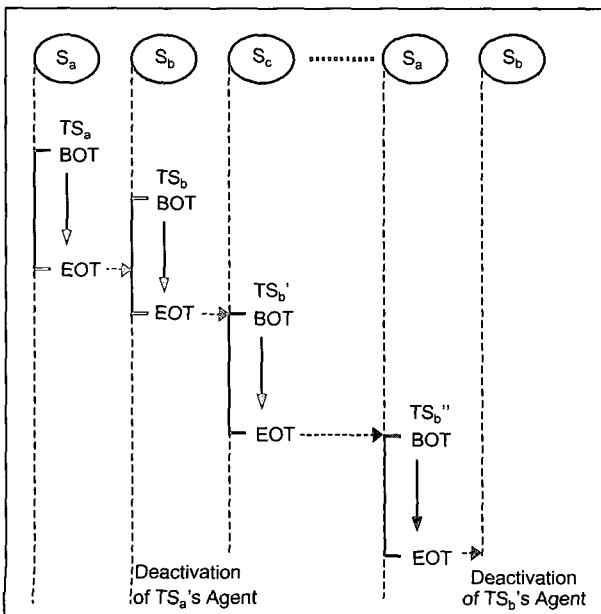
동기화 에이전트 시스템에서 모든 노드가 정상 동작할 경우와 어떤 노드가 비정상 동작할 경우를 고려하여 처리된 트랜잭션을 살펴보면 다음과 같다. 먼저 모든 시스템 노드가 정상 동작할 경우의 예이다.

$$\forall N, S_n > 0$$

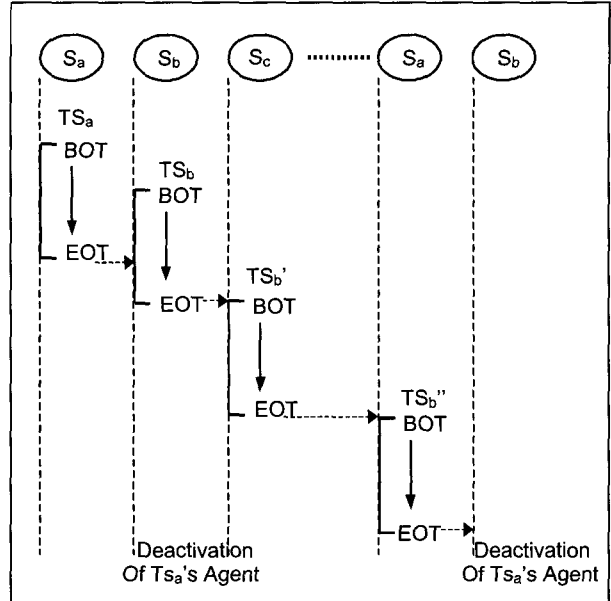
$$T_1 = TS(a, b) \Leftrightarrow Transaction(a)$$



$$T_2 = TS(b, a) \Leftrightarrow Transaction(b)$$



$$T_3 = TS(a | b) \Leftrightarrow P(a, b) \Leftrightarrow Transaction(b)$$

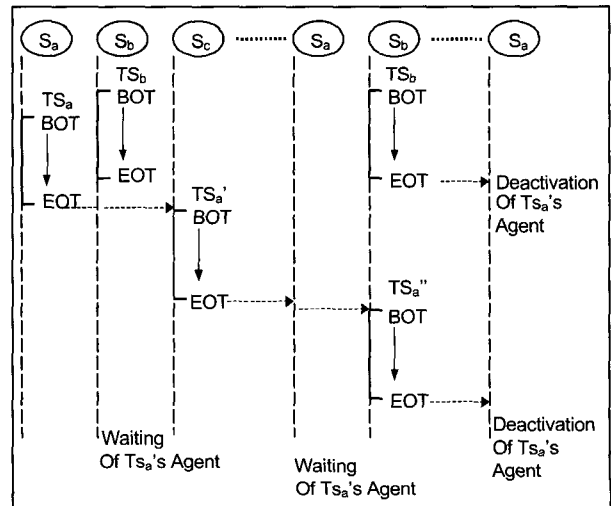


트랜잭션 타임스탬프의 값에 따라 동기화를 위한 트랜잭션 처리는 T_1, T_2, T_3 의 경우가 있다. S_a 와 S_b 에서 외부 애플리케이션에 의한 데이터 변경 이벤트가 발생하여 동기화를 하고자할 때, S_a 의 트랜잭션 타임스탬프가 높으면 TS_a 의 트랜잭션을 처리하고, S_b 가 높으면 TS_b 의 트랜잭션을 처리한다. TS_a 와 TS_b 가 일치하면 S_a 와 S_b 의 시스템 우선순위를 고려하여, (그림 4)에서 주어진 바와 같이 S_b 가 S_a 보다 우선하므로 TS_b 의 트랜잭션을 처리한다.

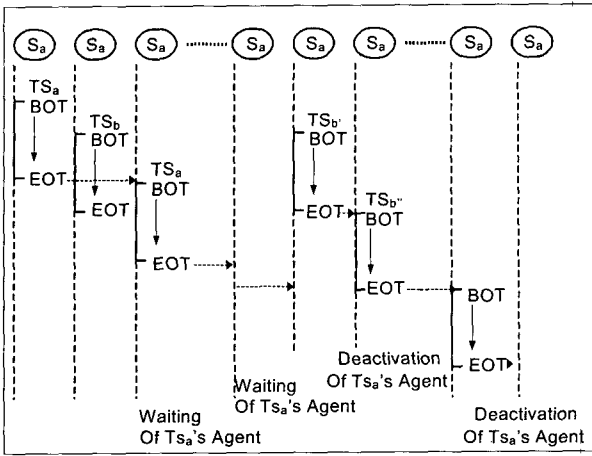
다음은 이동경로에 있는 시스템 노드 중에서 어떤 노드가 비정상 동작을 할 경우의 예이다.

$$\exists N, S_n = 0$$

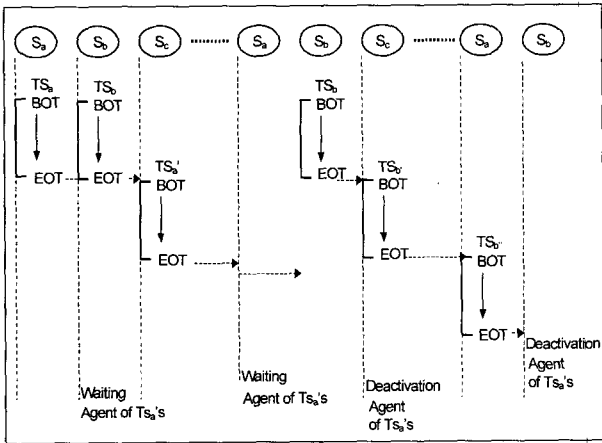
$$T_4 = TS(a, b) \Leftrightarrow Transaction(a)$$



$$T_5 = TS(b, a) \Leftrightarrow Transaction(b)$$



$$T_6 = TS(a | b) \Leftrightarrow P(a, b) \Leftrightarrow Transaction(b)$$



이동경로에 있는 시스템 노드의 비정상 동작에 의한 트랜잭션 처리는 T4, T5, T6 등의 경우가 있다. Sb가 비정상 동작하면 Sa의 트랜잭션이 진행되며, 트랜잭션 완료가 되지 않은 Sb로 인해 TSa의 동기화 에이전트는 대기하게 된다. Sb가 정상 동작하면 트랜잭션 타임스탬프의 값에 의하여 트랜잭션을 처리한다.

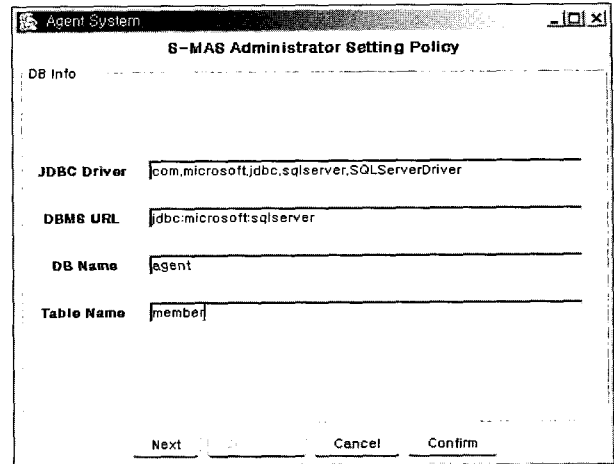
4. 구현 결과

4.1 동기화 정책 설정

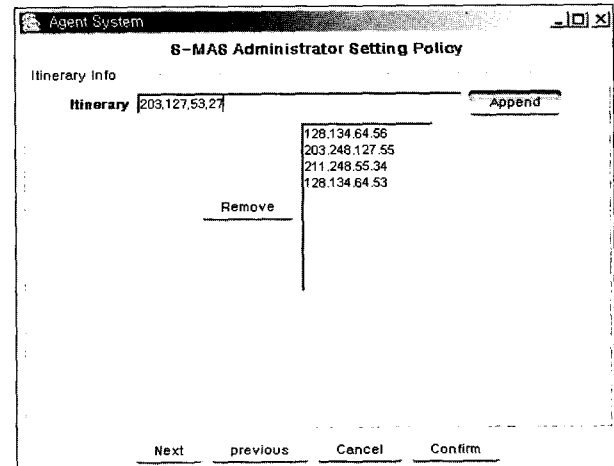
동기화 에이전트 시스템에서 정책은 데이터베이스 정보와 이동경로 정보, 시스템 우선순위 정보 등을 설정하도록 3부분으로 구성되어 있다.

첫째, 데이터베이스 정보는 본 시스템이 Java로 구현되어 있기 때문에, 각 에이전트 시스템에서 데이터베이스에 접근하기 위한 기본 정보인 JDBC, URL 정보와 동기화가 필요한 DB정보와 테이블 정보를 설정하며, 이는 (그림 5)와 같다. 둘째, 이동경로 정보는 동기화 에이전트가 이동하게 될 경로로서 목적지 시스템들에게 동기화 에이전트가 이주하

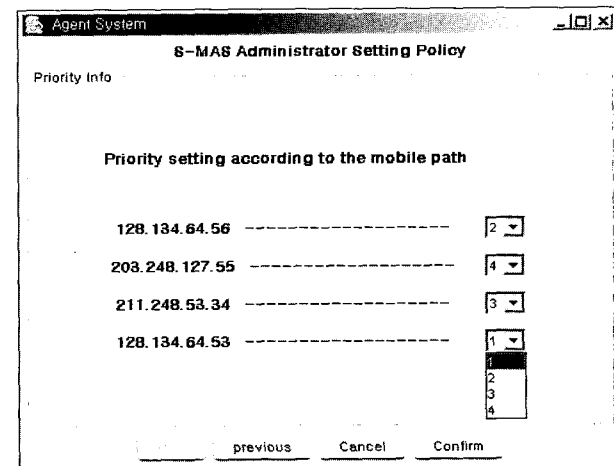
는 유니캐스팅 전송을 위한 정보를 설정하며, 이는 (그림 6)과 같다.



(그림 5) 동기화 정책설정 - DB 정보



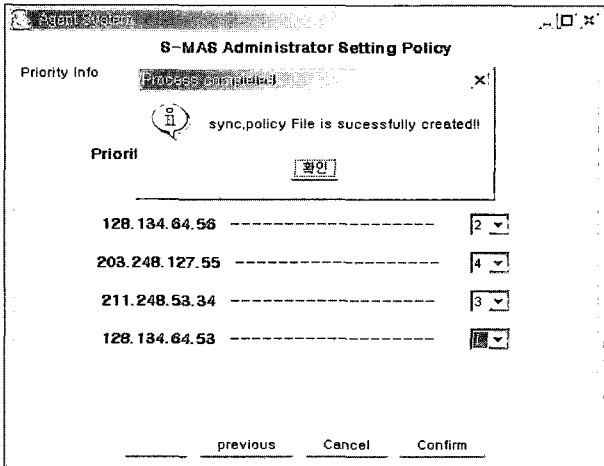
(그림 6) 동기화 정책설정 - 이동경로 정보



(그림 7) 동기화 정책설정 - 시스템 우선순위 정보 1

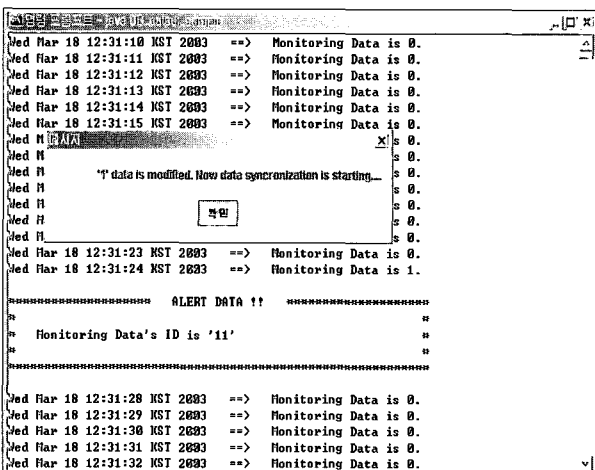
셋째, 시스템 우선순위 정보는 이동경로와 무관하게 SFT

데이터의 변경이 자주 발생하는 시스템을 위주로 높은 우선순위로 결정하게 된다. 동기화 정책에 의해 타임스탬프 기반으로 데이터 일관성을 유지하며, 동일한 타임스탬프 값으로 인한 데이터 충돌이 발생하면 시스템 우선순위에 의해 데이터 값의 유효처리 여부를 판단한다. (그림 7), (그림 8)에서 시스템 우선순위를 설정하며, 우선순위 값은 중복될 수 없고 유일한 값을 가져야 한다.



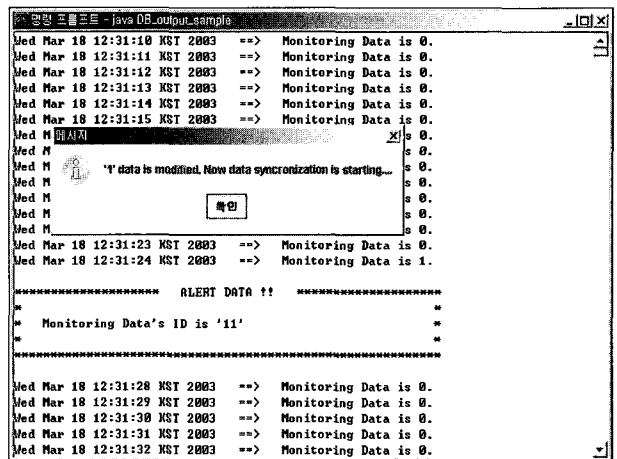
(그림 8) 동기화 정책설정 - 시스템 우선순위 정보 2

4.2 동기화 에이전트의 구동



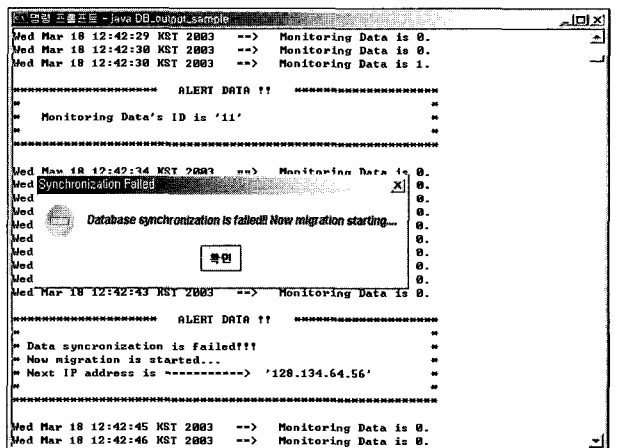
(그림 9) 동기화 에이전트 시스템의 동기화 요구 실행

다음은 동기화 에이전트 시스템의 구동에 의하여 데이터 변경 발생에 대한 일관성 관리자의 감지활동을 보여준다. (그림 9)에서처럼, 데이터 변경이 감지되면, 에이전트 관리자에게 데이터 변경을 알리고 목적지 시스템으로의 동기화를 요구한다. 정책 설정에 따른 이동경로의 목적지 시스템에게 변경된 데이터의 동기화를 추진하고 데이터 동기화가 성공적으로 처리되었을 경우에는 (그림 10)과 같은 동기화 완료 메시지를 보여준다.



(그림 10) 동기화 에이전트 시스템의 동기화 완료

네트워크 단절 등과 같은 이동경로 상의 목적지 시스템의 비정상적인 동작으로 인하여 동기화에 실패하였을 경우에는 (그림 11)과 같은 동기화 실패 메시지를 보여주며 다음 이동경로로의 동기화를 추진한다. 실패한 목적지 시스템은 후에 다시 동기화를 추진한다.



(그림 11) 동기화 에이전트 시스템의 동기화 실패

5. 결 론

e-비즈니스 솔루션들간의 원활한 정보 활용과 상호운용 등을 위한 정보공유 필요성이 확대되어 감에 따라 데이터 동기화에 대한 연구가 활발히 진행되고 있다. 정보공유를 위해서는 이중 시스템으로 구성된 분산환경의 플랫폼에 독립적이어야 하고, 데이터 갱신과 삭제에 따른 자율성이 보장되어야 하며, 네트워크 문제를 고려해야한다. 본 논문에서는 다양한 e-비즈니스 프로세스들의 상호 운용성을 보장하고 네트워크 문제를 고려하면서 데이터 동기화를 할 수 있는 이동 에이전트를 이용한 동기화 에이전트 시스템에 대하여 기술하였다. 또한 유니캐스팅 전송 상에서 시스템 우선순위와 타임스탬프의 트랜잭션을 기본으로 한 동기화 정

책을 제시하였다.

본 논문에서 제안된 동기화 에이전트 시스템은 데이터 변경 발생 빈도가 적은 SFT(Slow Frequency Transaction) 데이터 동기화에 국한하였다. 분산 환경에서 데이터 변경의 발생 빈도에 따라 데이터 동기화의 성능이 결정되므로, 주식 시세를 반영하는 시스템처럼 데이터 변경이 잦은 FFT(Fast Frequency Transaction) 환경에는 적합하지 않으며, 데이터 변경이 적게 발생하는 SCM(Supply Chain Management) 등과 같은 환경에 적합하다.

향후 연구 과제는 임의로 결정된 시스템 우선순위를 정하는 방법에 대한 연구와 분산된 노드의 수가 증가함에 따라 발생하는 지연현상에 대한 보완 연구, 보안 관련 정책과 레거시 시스템과 독립적인 응용 솔루션간의 정보 공유를 위한 다각적인 연구가 필요하다.

참 고 문 헌

- [1] Balazs Goldschmidt, Zoltan Laszlo, "Mobile Agents in a Distributed Heterogeneous Database System," IEEE, 2002.
- [2] Giacomo Cabri, Letizia Leonardi, Franco Zambonilli, "Mobile-Agent Coordination Models for Internet Applications," IEEE, 2000.
- [3] Heng Li, etc, "A framework for developing a unified B2B e-trading construction marketplace," Automation in Construction, Vol.12, pp.201-211, Mar., 2003.
- [4] Informix Software, "Cloudscape White Paper : Cloudscape LUCID Application Synchronization," April, 2001.
- [5] Isao Kaji, Yongdong Tan, Kinji Mori, "Autonomous Data Synchronization in Heterogeneous Systems to Assure the Transaction," 1999.
- [6] James E. White, "Mobile Agents White Paper", 1998.
- [7] Kenny K. F. Lee, "The Five Disciplines of ERP Software Implementation," ICMIT, 2000.
- [8] Stavros Papastavrou, George Samaras, Evaggelia Pitoura, "Mobile Agent for World Wide Web Distributed Database Access," IEEE, 2000.
- [9] Todd Papaionnou, John Edwards, "Mobile Agent Technology Enabling The Virtual Enterprise : A Pattern for Database Query," Agent Based Manufacturing workshop, part of Agents 1998.
- [10] Uwe Hansmann, etc "SyncML Synchronizing and Managing Your Mobile Data," PH PTR, pp.11-16, 2003.
- [11] Wang Yan, Law Ken C. K., "A Mobile Agent based System for Distributed Database Access on the Internet," IEEE, 2000.
- [12] Y. M. Siu, etc "A SFH Spread Spectrum Synchronization Algorithm for Data Broadcasting," IEEE Transactions on Broadcasting, Vol.47, Mar., 2001.



국 윤 규

e-mail : ykkook@cs.kwangwoon.ac.kr
 1999년 광운대학교 전자계산학과(이학사)
 2001년 광운대학교 컴퓨터과학과(이학석사)
 2003년 광운대학교 컴퓨터과학과 박사
 과정 수료
 관심분야 : 컴포넌트 개발방법, 웹서비스, 분산시스템, 분산컴퓨팅기술, 이동에이전트, 상호운용성



김 운 용

e-mail : wykim@cs.kwangwoon.ac.kr
 1999년 광운대학교 정보과학기술대학원 (이학석사)
 2003년 광운대학교 컴퓨터과학과(공학박사)
 2003년~현재 광운대학교 정보통신연구원
 관심분야 : 객체지향프로그래밍 언어, 객체 모델링, 디자인패턴, 컴포넌트 개발 방법, 분산컴퓨팅기술, 유비쿼터스



정 계 동

e-mail : kdjung@daisy.kwangwoon.ac.kr
 1985년 광운대학교 전자계산학과(이학사)
 1992년 광운대학교 산업대학원 전자계산학과(이학석사)
 1999년 광운대학교 컴퓨터과학과(이학박사)
 1993년~현재 광운대학교 정보과학교육원
 관심분야 : 객체지향프로그래밍 언어, XML 분산시스템, 분산 컴퓨팅기술, 이동에이전트



김 영 철

e-mail : bob@hongik.ac.kr
 1985년 홍익대학교(이학사)
 1987년 광운대학교(이학석사)
 2000년 Illinois Institute of Technology (전산학박사)
 2000년~2001년 LG 산전 중앙연구소 Embedded 시스템 책임연구원/부장
 2003년 카네기멜로 대학교(Advanced Software Engineering 프로그램)
 2001년~현재 홍익대학교 컴퓨터정보통신학과 교수
 2003년~현재 유비젠(주) 고문
 관심분야 : Use Case 방법론, Design driven Testing, TMM(Testing Maturity models, 데이터 모델링, 이동에이전트, 상호운용성



최 영 근

e-mail : ygchoi@cs.kwangwoon.ac.kr
 1980년 서울대학교 수학교육과(이학사)
 1982년 서울대학교 계산통계학과(이학석사)
 1989년 서울대학교 계산통계학과(이학박사)
 1992년~2000년 광운대학교 전산정보원 원장
 1992년~현재 광운대학교 컴퓨터과학과 교수
 2002년~현재 광운대학교 교부연구처장
 관심분야 : 프로그래밍 언어, 병렬 프로그래밍언어, 객체지향 설계 및 분석, 분산 컴퓨팅기술, 이동에이전트, 상호운용성