

# E-로지스틱스 시스템에서 메시지 전송 에이전트의 이단계 완료규약 활용

안인순\*, 김경창\*\*, 이해원\*\*\*, 김인수\*\*\*, 김혜규\*\*\*

\*안동과학대학 정보처리학과

\*\* 홍익대학교 컴퓨터공학과

\*\*\*한국전자통신연구원 우정기술연구부

## Applying Two Phase Commit Protocol in Message Transport Agent for e-로지스틱스 System

Ihn-Soon Ahn\*, Kyung-Chang Kim\*\*, Hae-Won Lee\*\*\*, In-Soo Kim\*\*\*, Hye-Kyu Kim\*\*\*

\* Dept. of Information Processing, Andong Science College

E-mail : \*aisoon@andong-c.ac.kr

\*\* Dept. of Computer Science, Hongik Univ

E-mail : \*\*kckim@cs.hongik.ac.kr,

\*\*\*ETRI Postal Technology Development Department

E-mail : \*\*\*hewlee@etri.re.kr

### 요 약

e-로지스틱스 시스템은 인터넷을 기반으로 한 가상 기업 물류 활동 및 서비스 체계를 구축한 시스템을 말한다. 이 시스템은 대상 지역의 무제한 광역화를 허용하고, 높은 수준의 물류 체계 유연성 및 효율성을 요구하며, 다자간 물류정보의 실시간 통합환경을 요구한다. 또한 무엇보다도 고객 서비스의 지식 정보화가 요구되는 시스템이라 할 수 있다. e-로지스틱스 프레임워크 개발에 있어서 메시징 모델은 매우 중요하다. 현재 SOAP 기반 XML 형태의 메시징 모델이 개발 중이다.

본 논문에서는 e-로지스틱스 시스템의 서비스를 위하여 메시지 전송 에이전트가 서버와의 메시지 교환 시 발생할 수 있는 고장에 대하여 어떻게 효율적으로 대처할 수 있게 하는가에 대해 설명한다.

### 1. 서론

21세기는 정보와 네트워크 혁명의 대중화를 통한 지식정보화 사회가 구현되고 있다. 인터넷의 출현은 지식정보화 사회로 가는 하나의 예로써 충분하다. 인터넷은 인간에게 광범위한 정보의 세계의 필요한 정보를 제공함으로써 생활환경의 변화를 가져왔다.

또한 상업적으로는 인터넷을 통한 전자상거래가 활성화되어 기존 상거래시장에 커다란 영향을 주게 되었다. 그러나 현재의 전자상거래 비즈니스 모델에 대한 전자상거래에 커다란 영향을 미치게 되어 계속적

인 연구가 되어야 할 부분이다.

e-로지스틱스(Logistics) 시스템은 인터넷을 기반으로 한 가상 기업 물류 활동 및 서비스 체계를 구축한 시스템을 말한다. 이 시스템은 실시간으로 고객이 원하는 물류 정보에 대한 서비스를 지식 정보화한 것을 말한다.

또한 서로 다른 정보 시스템간의 데이터, 어플리케이션, 비즈니스 프로세스를 통합을 말한다. 또한 공급망의 가시화를 제공한다. 물류정보의 흐름에 대한 실시간 모니터링, 예외상황 발생 시 지능화된 경고 및 조치를 취한다. 그리고 최적화된 물류 계획을 수립하고 물류 계획의 동적 재조정을 가능하게 한다.

e-로지스틱스 시스템을 연구 개발해야 하는 필요성은 사회적 측면과 문화적 측면, 경제적 측면으로 나

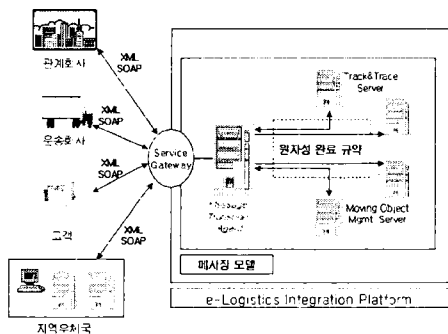
본 연구는 한국전자통신연구원 연구과제(과제번호:2001-S-157) 연구비에 의해 연구되고 있음.

높 수 있다. 사회적 측면은 물류 운송 사업의 수익성 확보 및 수익 증대를 위한 물류 인프라의 개선하고 e-Business의 활성화에 따른 물류 시장의 확대를 지원할 수 있는 통합 물류 시스템이 필요하다. 문화적 측면에서는 인터넷을 통한 물류 공급자들의 실시간 협업 및 비즈니스 프로세스의 공유가 필요하다. 또한 경제적 측면에서는 우정사업본부의 보고서에 따르면 국내 우편사업의 경우 연간 100억원 정도의 운송비용 절감을 기대할 수 있다. 그리고 물류 기업의 경쟁력 강화를 위해 생산원가 절감 및 효과적인 물류 관리를 가져 올 수 있다.

본 논문에서는 e-로지스틱스 시스템에서 메시지 전송 에이전트(Message Transport Agent : MTA)[1]의 역할과 각각의 서버와 메시지 교환 시 발생할 수 있는 고장에 대해 어떻게 대처하는 가를 설명한다. 2절에서는 e-로지스틱스 시스템에서 메시지 모델 설계에 대하여 설명하고, 3절에서는 MTA가 각각의 서버들과 메시지 교환 시 발생할 수 있는 고장에 대해 기존 분산 데이터베이스 시스템에서 사용하는 이단계 완료규약을 어떻게 활용하여 대처하는 가를 보인다. 4절에서는 향후연구과제와 결론을 맺는다.

## 2. e-로지스틱스 시스템의 메시지 모델 설계

메시징 모델 개발은 e-로지스틱스 통합 플랫폼에서 개발을 요하는 과제로서 기존 우편 업무 환경과 향후 재편될 우편 업무 환경을 고려한 우편 물류 데이터를 안전하고 신뢰성 있게 전달 할 수 있는 SOAP 기반 XML 형태의 메시징 기법 연구와 메시지 교환 모델 개발에 중점을 두고 있다[3].



(그림 1) e-로지스틱스 시스템의 메시지의 흐름

메시징 모델은 사용자의 질의가 요청되면 서비스 게이트웨이(Service Gateway)을 통하여 MTA에게 전

달된다. MTA는 요청에 따라 요청에 관련된 서버들에게 질의를 보낸다. 실질적인 우편물이 자동 처리되어, 어느 집중국을 거쳐 어떤 배송 차량을 통해 도착지까지 운송되고 배달되는지는 Track & Trace 서버에게 보내지고, 그 배송 차량에는 다른 어떤 우편물이 함께 운송되고 있는지 등의 정보는 Moving Object 관리 서버에게 보내는 등 요청에 대한 서버의 역할을 분류했다.

따라서 서버들은 자신들의 데이터베이스를 참조하여 질의에 대한 응답을 하게된다. 이렇게 실시간으로 질의 처리할 해줌으로 고객의 욕구 충족은 물론 고객의 신뢰를 개선시키고 이를 바탕으로 고객과의 장기적 유대관계를 형성함으로써 보다 높은 경쟁력을 갖춘 우체국으로 성장하는데 도움이 될 것이다.

그러기 위해서는 메시징 모델에서 MTA와 Track & Trace 서버와 Moving Object 관리 서버 등, 여러 서버의 데이터 분석과 메시지 전송 시 발생할 수 있는 고장에 대해 MTA와 서버들의 불일치성을 방지할 수 있는 완료규약의 연구가 필요하다(그림 1).

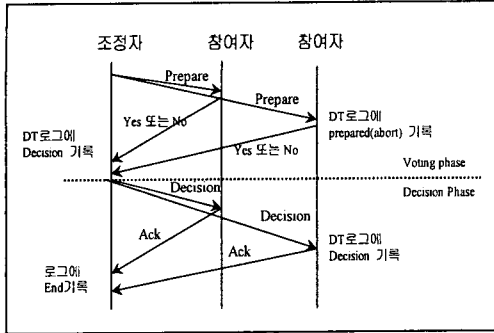
## 3. 메시지 전송 에이전트의 이단계 완료 규약 활용

MTA와 각 서버들 간의 메시지 교환 시 발생할 수 있는 고장은 MTA와 각 서버들의 결정을 불일치하게 만들 수 있다. 따라서 이단계 완료규약을 응용하여 이러한 상황을 없애고, 메시지 교환을 원활하게 할 수 있게 한다.

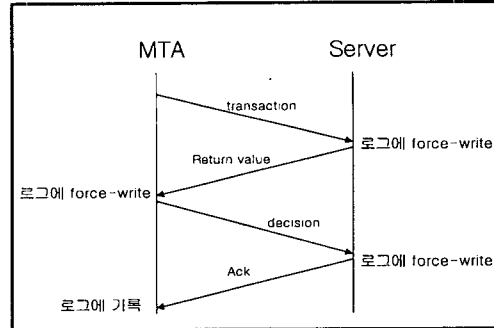
다음은 이단계 완료규약에 대하여 설명하고, 이단계 완료 규약을 e-로지스틱스 통합 플랫폼에 응용하므로써 메시징 모델 개발 시 MTA와 각각의 서버들의 불일치성을 극복하게 만든다.

### 3.1 이단계 완료규약

이단계 완료 규약은 조정자와 참여자들간의 메시지 교환을 통하여 규약에 참여하는 모든 사이트가 동일한 결정에 도달하게 하는 것이다[6]. 이 규약은 선출 단계(voting phase)와 결정단계(decision phase)로 구성된다(그림 2). 선출단계는 조정자가 참여자들에게 완료규약을 수행할 준비가 되어있나 질의하는 단계이고 결정단계는 조정자가 참여자들에게 규약을 완료하거나 철회시키는 단계이다. 규약의 수행과정은 다음과 같다.



(그림 2) 2PC 규약의 수행과정



(그림 3) MTA 와 서버들간의 메시지 교환과정

- [단계1] 조정자는 선출단계에서 Prepare 메시지를 모든 참여자에게 보낸다.
- [단계2] 참여자들은 Prepare 메시지를 받으면 트랜잭션을 완료할 것인지 아닌지 여부를 결정하여 DT(Distributed Transaction) 로그에 prepared 또는 abort를 기록하고 조정자에게 Yes 또는 No 메시지를 보낸다. DT 로그는 분산된 트랜잭션의 정보를 기록하며 조정자와 참여자 모두 가지고 있다. 또한 DT 로그의 내용은 사이트 고장이 발생하는 경우에도 살아남기 위해 안전 장치에 유지된다.
- [단계3] 조정자는 참여자들이 보내온 Yes 또는 No 메시지에 따라 참여자들을 완료 또는 철회할 것인지 결정한다. 만약 모든 참여자가 Yes 메시지를 보냈다면, 모든 참여자들의 완료를 결정하기 위해 DT 로그에 commit를 기록하고, Commit 메시지를 모든 참여자에게 보낸다. 그렇지 않은 경우는 DT 로그에 abort를 기록하고 Yes 메시지를 보낸 참여자들에게 Abort 메시지를 보낸다.
- [단계4] 참여자는 결정 메시지(Commit 또는 Abort)를 받으면, 결정에 따라서 DT 로그에 abort 또는 commit를 기록하고 조정자에게 Ack 메시지를 보낸다.

### 3.2 MTA의 이단계 완료규약의 활용

MTA은 이단계 완료규약에서의 조정자 역할을 하게되고 각각의 서버들은 참여자 역할을 하게된다. 이 단계 완료규약과는 다른 환경이므로 이단계 완료규약을 응용하여 우편 물류 데이터의 신뢰성을 향상시킬 수 있게 하였다. 다음은 MTA와 서버들간의 메시지 교환의 일관성을 유지하기 위한 완료규약 수행과정을 설명한다(그림 3).

- [단계1] MTA는 로그에 transaction을 기록하고, 각 Server들에게 transaction을 보낸다.
- [단계2] 서버들은 로그에 transaction, 반환 값(return value)을 기록하고 안전장치로 옮긴 후 반환 값을 MTA에게 보낸다. 반환 값이 error인 경우는 로그에 abort를 기록하고 안전장치로 옮기고 트랜잭션을 abort 시킨다.
- [단계3] MTA는 서버들의 반환 값을 확인하고 서버들의 반환 값 중 하나라도 error가 있으면 로그에 abort를 기록하고 안전장치로 옮긴 후 abort 메시지를 error를 보내지 않은 서버들에게 보내고, 그렇지 않은 경우는 로그에 commit를 기록하고 안전장치에 옮긴 후 모든 서버들에게 commit 메시지를 보낸다.
- [단계4] 서버들은 MTA의 결정에 따라 트랜잭션을 로그에 commit 또는 abort를 기록하고 안전장치로 옮긴 후, 완료하거나 철회를 결정하고 Ack 메시지를 MTA에게 보낸다.
- [단계5] MTA는 모든 Ack 메시지를 받으면 트랜잭션을 완성하고, 서비스 게이트웨이(Service Gateway)를 통해 사용자에게 질의의 응답을 보낸다.

MTA와 서버들 간의 메시지 교환 시 고장으로 인해 메시지가 도착하지 않으면 메시지를 계속 기다려야 하는 경우가 발생한다. 이러한 경우를 피하기 위해 timeout 행위를 이용한다. timeout 행위는 다음과 같이 [단계2], [단계3], [단계4], [단계5]에서 발생할 수 있다. 그러나 MTA와 서버들의 timeout 행위는 단순히 회복될 때까지 기다린다. 이것은 시스템의 정확성과 견고성을 유지하기 위함이다.

MTA와 서버들이 고장이 발생하여 회복될 때 자신들이 결정은 다음과 같이 처리한다.

- ▷ MTA가 트랜잭션을 보내기 전에 고장이 발생  
: 로그의 기록을 참조하여 트랜잭션을 보낸다.
- ▷ 서버가 트랜잭션을 처리하고 반환 값을 보내기 전에 고장발생  
: 서버들은 고장에서 회복되면 로그 기록을 참조하여 반환 값을 보낸다.
- ▷ MTA가 결정 메시지를 보내기 전에 고장이 발생  
: 조정자는 회복되면 로그 기록을 보고 결정 메시지를 보낸다.
- ▷ 서버들은 결정 메시지를 받은 후 고장이 발생  
: 회복되면 MTA에게 Ack 메시지를 보낸다.

MTA와 서버들이 교환하는 메시지 형태는 다음과 같은 정보들로 구성된다.

▷ 트랜잭션 메시지 형태

Tid	MTAid	Serverid	message(transaction)
-----	-------	----------	----------------------

Tid : 트랜잭션 식별자  
MTAid : MTA 식별자  
Serverid : 서버 식별자  
message : MTA가 서버에게 보내는 트랜잭션

▷ 반환 값(return value) 메시지 형태

Tid	MTAid	Serverid	message(return value)
-----	-------	----------	-----------------------

message : 서버가 MTA에게 보내는 transaction 처리의 결과 값

▷ 결정(decision) 메시지 형태

Tid	MTAid	Serverid	message(commit또는 abort)
-----	-------	----------	-------------------------

message : MTA가 서버에게 보내는 반환 값에 대한 결정(트랜잭션의 완료 또는 철회)

▷ Ack 메시지 형태

Tid	MTAid	Serverid	message(Ack)
-----	-------	----------	--------------

message : 서버가 MTA에게 보내는 결정에 대한 응답

MTA와 서버들이 저장하는 로그 레코드의 형태는 MTA와 서버들이 교환되는 메시지의 내용이 저장된

다. 또한 로그 레코드의 내용은 고장이 발생하는 경우를 대비하여 안전장치에 저장한다.

이와 같이 우리는 e-로지스틱스 시스템에서 메시지 전송 에이전트와 서버들간의 메시지 교환을 일관되게 하게 하기 위하여 분산 데이터베이스 시스템에서 사용하는 이단계 완료규약을 응용하여 수행하므로써 MTA와 각각의 서버들의 원자성을 가지게 할 수 있었다.

#### 4. 결 론

e-로지스틱스 시스템은 인터넷을 기반으로 한 가상 기업 물류 활동 및 서비스 체계를 구축한 시스템을 말한다. 이 시스템의 활용은 대상 지역의 무제한 광역화를 허용하고, 높은 수준의 물류 체계 유연성 및 효율성을 가질 수 있다. 또한 다자간 물류정보의 실시간 통합환경을 설정할 수 있는 장점을 가지고 있다.

본 논문에서는 e-로지스틱스 시스템에서 메시지 전송 에이전트(Message Transport Agent : MTA)와 서버들간의 메시지 교환을 일관되게 할 수 있도록 분산 데이터베이스 시스템에서 사용하는 이단계 완료규약을 변형하여 e-로지스틱스 시스템에 적합하게 적용하므로써 MTA와 서버들의 원자성을 보장할 수 있었다. 이것은 무엇보다도 시스템의 안전성에 비중을 두고 적용한 완료 규약이다.

향후연구과제로는 MTA와 서버들간의 메시지 교환을 향상시켜 고장이 발생하더라도 회복될 때까지 기다릴 필요 없이 완료 규약을 수행시킬 수 있는 새로운 완료규약의 연구가 필요하고, 현재 연구 진행중인 MTA와 서버들간의 메시징 모델을 SOAP 기반의 XML 형태로 구현해야 하겠다.

#### [참고문헌]

- [1] 천성광, 이해원, e-Logistics Platform MTA 요구 사항 정의서 ETRI 내부문서, 2001.10.
- [2] ebXML Message Service An In-depth Overview, Christopher Ferris  
Http://www.gca.org/papers/xml europe2001/papers/html/s09-3.html
- [3] Electronic business XML(ebXML) Requirements Specification Version1.0, http://www.ebxml.org
- [4] Message Service Specification v1.0 Transport, Routing & Packaging Team, 11 May 2001
- [5] XML messaging, Part1, Dirk Reinshagen  
http://www.javaworld.com/javaworld/jw-03-2001/j

w-0302-xmlmessaging\_p.html

[6] Lampson, B. Atomic Transactions. Distributed Systems: Architecture and Implementation - An Advanced Course, B. Lampson(Ed.), Lecture Notes in Computer Science, Volume 105, pp. 246-265, Springer-Verlag, 1981.

[7] J. Gray and A. Reuter, Transaction Processing: Concepts and Techniques. Morgan Kaufman, 1993.

[8] Skeen D, "Non-blocking Commit Protocols," Proc. of the ACM SIGMOD Int'l Conference on the Management of Data. pp. 133-142, May 1981.

[9] Mohan, C. and B. Lindsay, "Efficient Commit Protocols for the Tree of Processes Model of Distributed Transactions," Proc. of the 2nd ACM SIGACT/SICOPS Symposium on Principles of Distributed Computing, Aug. 1983.