

# EAI의 C.M의 Ledger File을 통한 데이터 전달보증 처리에 관한 연구

조승용\*

전남대학교 전산학과

e-mail : [csytop1@hanmail.net](mailto:csytop1@hanmail.net)

## A Study on assure data delivery using C.M(Certificate Message) Ledger

Seung Yong Cho

*Dept,of Computer Science Chonnam National University*

### 요 약

EAI(Enterprise Application Integration : 전사적 애플리케이션 통합)는 기업간 또는 기업 내부의 다양하고 이질적인 시스템과 애플리케이션 그리고 비즈니스 프로세서를 통합 하는것[1]이며, 본 논문에서는 기존의 데이터 처리에서 발생하는 문제점인 수신측 장애시 데이터의 손실이 발생하는 것을 방지하기 위한 방법으로 C.M(Certificate Message)방식을 제안하였으며 이를 증명하기 위하여 Ledger File에 Confirm 되지않은 메시지를 보관하여 수신측의 장애 이거나 네트워크 장애에도 Data 손실을 방지하는, Review Ledger File 방식을 적용한 C API 프로그램으로 구현하여 증명함으로써 수신측 상태가 정상적으로 된 경우에 송신측의 Ledger File 다시 읽어서 재송신하여 그 어떤 장애 상황에도 Data 손실됨을 방지하여 실시간 트랜잭션의 처리에 획기적인 방법을 제시하였고, 기존 EAI 시스템 개발시에 문제점으로 대두된 표준화된 개발 방법이 부재하여 개발 및 유지보수에 많은 비용이 발생한 것에 대하여 본 논문은 표준화 방법을 통하여 해결책을 제시하였고, 또한 Ledger File 및 데이터에 대한 모니터링 방법을 Cobol API 프로그램을 통하여 구현하여 실시간 장애 모니터링 방법을 제시하였다. 마지막으로 결론에서 앞으로의 발전방향에 대하여 논의한다.

### 1. 서론

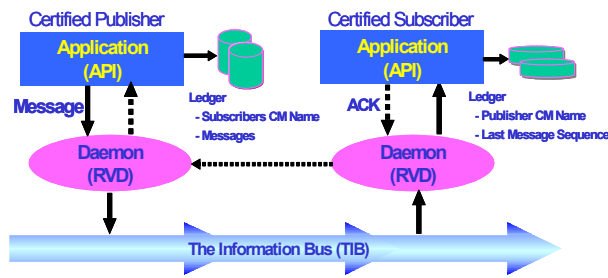
전사적 애플리케이션 통합 (EAI:Enterprise Application Integration)이란 기업 내부의 다양한 시스템과 애플리케이션을 통합하는 것이며, MOM(Message Oriented Middleware)은 메시징에 초점을 맞춘 미들웨어이다.[2] EAI 시스템 구축시에 기존의 문제점인 수신측 및 네트워크 장애시에 전달되고 있던 데이터의 전달을 보증할수 없었던 단점[3]을 본 논문에서는Review Ledger File의 개념을 도입하여 수신측 및 네트워크 장애시 에도 안정적인 Data의 전달을 수행 하기 위하여 Cobol Program을 통한 Ledger File을 구현하여 송신측에서 Confirm을 받지 못한 Data를 저장하여 수신측 상태가 정상적으로 바뀌면 송신측 Ledger File에 보관하고 있던 데이터를 재송신하여 어떤

장애상황에도 Data의 전달을 보증함으로써 향후 EAI 구축시에 필수적인 요소인 데이터의 전달보증을 통한 전체적인 트랜잭션 처리에 문제가 없게 하였다. 본 논문에서는 이를 실제 프로그램 구축을 통하여 실현함으로써 기존 방식의 문제점을 해결함으로써 향후 EAI 시스템 구축시 많은 도움이 될 것이다. 그리고 마지막으로 결론에서 향후과제를 논의한다.

### 2. 관련 연구

기존의 방식에는 송신측에서 송신한 데이터를 보관할 방법을 가지고 있지 않아서 수신측 프로세서나 네트워크가 중단되는 경우 그 중단 기간에 발생한 데이터를 보관할 방법이 없어서 송신한 데이터에 대하여 전달을 보장 받을수 없게 되어서 전체 트랜잭션 처리 및 데이터 정합에 문제가 되어서 심각한 문제점

으로 대두되어 왔었다. 그리하여 본 논문에서는 송신 데이터를 순서대로 보관하여 수신측 프로세서 및 네트워크가 정상적으로 되었을 경우에 다시 송신측의 Ledger File에 있던 데이터를 재송신하여 수신측에서 그 어떤 장애상황이 발생하여도 데이터의 전달보증을 받을수 있는 것이 C.M (Certificate Message)이다. 즉 C.M 방식에서는 Ledger File에 데이터를 보관하고 있다가 수신측의 장애가 정상적으로 복구될 경우에 송신한 데이터의 Sequence을 가지고 있어서 순서대로 다시 Data을 보낼수 있게 되어서, 메시지가 손실되는 경우가 없이 재전송할 수 있게 되기 때문에 데이터의 안정성을 보장할수 있게 된다. 본 논문에서는 이러한 데이터의 전달보증을 증명하기 위하여 Review Ledger File 개념을 도입하였고 Cobol API Program을 통하여 구현 증명 하였으며, 또한 데이터의 결과를 추적하여 장애시 신속한 문제해결이 가능하도록 데이터 추적 프로그램을 구축함 으로서 실시간 장애처리 방법을 제안하였다.



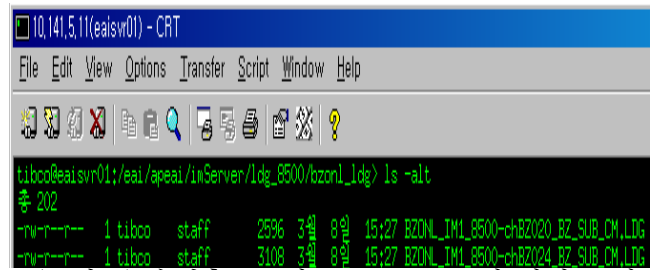
(그림 1) Certificate Message 개념도

위의 그림과 같이 C.M 방식에서는 영구히 메시지의 신뢰성을 보장하기 위한 전송방식으로 저장장소인 Memory 또는 File을 사용하며 [4], 송신측에서는 Data를 보내기 전에 저장소에다 메시지를 보관하고 수신측의 데이터 손실시 Ledger File에 있는 Data를 재전송하여 수신측에서 데이터를 받게 되면 Confirm Message를 송신측에 전송하고, 송신측은 Ledger File에 있는 보낸 데이터를 삭제하여 중복 송신을 방지한다.

### 3장 EAI Ledger File 모니터링 구현 예제

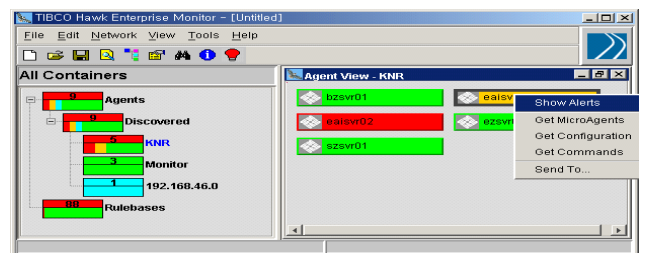
EAI연계 방식중 CM(Certificate Message)통신을 할 경우 Ledger라는 물리적 파일을 사용하여 정보 교환에 필요한 정보 및 DATA등을 저장하여, 수신측에서 정보를 정상적으로 받지 못한 경우에 데이터를 재전송을 하여 데이터의 전달을 보장하게 되므로 송신측의 Ledger File의 크기를 모니터링하여 크기가 계속 증가시에 수신측에서 Confirm Message를 주지 못하여 송신측 Ledger File의 크기가 증가 한 것 이므로 즉시 장애에 대하여 복구조치를 하고,

다시 데이터를 재송신하여 수신측에서 정상적으로 데이터를 처리할 수 있게 하는 모니터링 방법 구현 하였다.



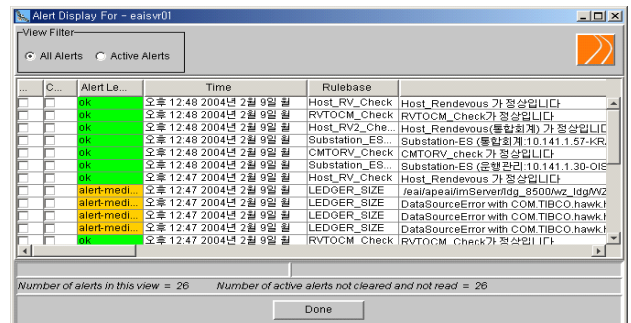
(그림 2) 송신측 IM의 Ledger File 의 파일 크기

[eaisvr01:]/eai/apeai/imServer/ldg\_8500/bzoni\_ldg/BZONL\_IM3\_8500-chBZ020\_BZ\_SUB\_CM.LDG 송신측 IM 의 Ledger File에 보관된 파일 크기이다.



(그림 3) 장애시에 표시되는 경고 메시지 창

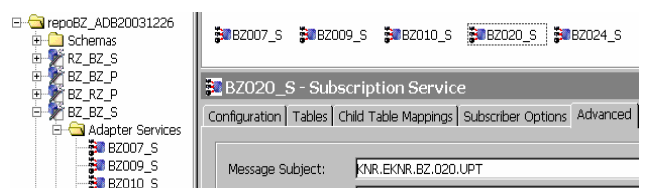
위의 내용은 Eaisvr01서버에서 장애상황을 감지하여 SMS를 통하여 경고 메시지를 표시하며 그 내용을 분석하여 즉각적인 장애복구가 가능하게 하였다.



(그림 4) 경고창에서 레저 File Full 에러 표시

[eaisvr01]/eai/apeai/imServer/ldg\_8500/wz\_ldg/WZONL\_IM1\_6500-chWZ006\_PUB\_CM.LDG 의 파일 Size가 1024Byte를 넘어서 장애가 발생함.

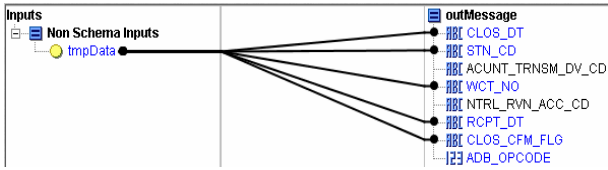
### 4장 C.M 방식을 통한 데이터 전달보증을 위한 구현 예제



(그림 5) 수신측 ADB 프로세서 내용

위의 수신측 ADB 프로세서이며 Subject Name

은 KNR.EKNR.BZ.020.UPT이며, 송신 테이블에 변경사항(insert)이 발생하면 송신측 테이블의 데이터를 읽어서 수신측으로 송신한다.



(그림 6) I.M 을 통한 송.수신 데이터 매핑  
I.M은 이기종 간의 데이터 필드를 매핑하여 수신측으로 보내어 처리하는 메시지 브로커이다.

Listener	Last Confirmed Message	Total Unacknowledged Messages
BZ020 BZ PUB CMQ8500	679985	20

(그림 7) Ledger File 에 쌓인 건수 (20개)

위 파일에서 처럼 만약 수신측의 프로세서의 장애로 인하여 송신한 데이터가 수신측의 D.B에 정상적으로 Insert를 하지 못한 경우에 즉 송신측에서 Confirm을 받지 못한 데이터가 Total Unacknowledged Messages 20으로 표시된 것이며 IBM 의 HOST의 CM Agent에서 프로세서 기동이 중지되었거나 I.M프로세서의 이상 또는 수신측 ADB 프로세서의 중지 등으로 인하여 송신측 Ledger File에 데이터 가 쌓인 것이며, C.M Agent 프로세서가 재기동 되거나 수신측에서 프로세서가 정상적으로 수신가능하면 다시 Ledger File을 읽어서 재송신하여 데이터의 전달을 보증하는 방법을 구현하였다.

```

SDFS DA K2 K2 PAG 0 SIO 3754 CPU 17/ 13 LINE 1-3 (3)
COMMAND INPUT ==>>
SCROLL ==>>
NP JOBNAME STEPNAME PROCSTEP OWNER C POS REAL PAGING SIO CPU%
TIBRVDKR TIBRVDKR STEP01 TIBU IN 5107 0.00 0.00 0.05
TIBKNRA TIBKNRA TIBSMT TIBU NS 1750 0.00 0.00 0.00
TIBHOSR TIBHOSR HOSTUNIX TIBU IN 2110 0.00 0.00 0.02
    
```

(그림 8) Host C.M 프로세서 기동(TIBHOSR)

위에서 TIBHOSR가 CM Agent 인데 이 프로세서가 Down되었을 경우에 Ledger File에 쌓인다.

```

Menu Options View Utilities Compilers Help
DSLIST - Data Sets Matching TIBCO.* (K1 V2P)
Command ==>>
Tracks %Used
TIBCO_BDBA001B_LEDGER1 300 1
TIBCO_BDBA016B_LEDGER1 300 1
TIBCO_BDBA101B_LEDGER1
TIBCO_BDBA116B_LEDGER1
TIBCO_CICSLIST 15 20
TIBCO_CM_TIBHOST_LEDGER1 1500 1
TIBCO_CM_TIBHOST2_LEDGER1 450 1
TIBCO_CM_TIBHOST3_LEDGER1 450 2
    
```

(그림 9) Host Ledger File의 크기 ( 2%)

위는 Ledger File Named 이 TIBCO.CM.TIBHOST3.LEDGER1의 사용량인 %Used가 2%가 되어서

Ledger File에 쌓인 데이터를 분석 하기위하여 파일을 PC로 다운로드 한후에 아래의 내용을 분석한다.

New Seq	Ack	Sequence	Message
679985		( DATA="BDRG128B012005022801010000025316820050307" )	
679987		( DATA="BDRG128B012005022801010000025346820050307" )	
679988		( DATA="BDRG128B012005022801010000025366820050307" )	
679989		( DATA="BDRG128B012005022801010000025386820050307" )	
679990		( DATA="BDRG128B012005022801010000025406820050307" )	
679991		( DATA="BDRG128B012005022801010000025426820050307" )	
679992		( DATA="BDRG128B012005022801010000025446820050307" )	
679993		( DATA="BDRG128B012005022801010000025466820050307" )	
679994		( DATA="BDRG128B012005022801010000025486820050307" )	
679995		( DATA="BDRG128B012005022801010000025506820050307" )	
679996		( DATA="BDRG128B012005022801010000025526820050307" )	
679997		( DATA="BDRG128B012005022801010000025546820050307" )	
679998		( DATA="BDRG128B012005022801010000025566820050307" )	
679999		( DATA="BDRG128B012005022801010000025586820050307" )	
680000		( DATA="BDRG128B012005022801010000025606820050307" )	
680001		( DATA="BDRG128B012005022801010000025626820050307" )	
680002		( DATA="BDRG128B012005022801010000025646820050307" )	
680003		( DATA="BDRG128B012005022801010000025666820050307" )	
680004		( DATA="BDRG128B012005022801010000025686820050307" )	
680005		( DATA="BDRG128B012005030101010000025948820050307" )	

(그림 10) Ledger File의 내용 표시

위의 파일을 분석하여 보면 LastAcked Msg #: 679985는 마지막으로 confirm message을 받은 것이고 TotPending Msg 20은 수신측으로부터 confirm message을 받지 못한 데이터 건수가 20개 임을 표시한 것 이며 이를 Ledger File에 보관한 것이다.이곳에 쌓인 data을 다시 Polling해서 재송신 하기 위한 Review Ledger Program이 필요하므로 다음의 CM Review Ledger Program으로 구현 증명 하였다.

**Review Ledger COBOL Program 예제**

```

IF WS-TOTAL-MSGS > 0
MOVE C-NO TO CHECK-CONTINUE.
IF REVIEW-CNT >= +3
    
```

위의 COBOL PROGRAM에서 Ledger File을 3번 Polling 하여 Confirm 을 받지 못한 Data를 다시 읽어서 재전송 함으로써 데이터의 손실을 방지하여 데이터의 정확성을 보장하는 것이다.

```

[10/14/05 10:54:00] - CRT
File Edit View Options Transfer Script Window Help
[2005/03/08 10:54:00:122] msocint: C:\GetDataMsg\ICD\TA\142\ LOCAL_PUB_KNR_EKNR_BZ_020_MHP - CDATA="BDRG128B01200503010100000272488"
    
```

(그림 11) 데이터 내용 추적 화면

위 화면과 같이 해당 로그 파일을 통하여 각 시스템 별로 해당 message(KNR.EKNR.BZ.020.MAP)가 언제 발생하여 어느 시스템까지 데이터가 도달 했는지를 확인할 수 있어서 어떤 장애시 예도 신속한 조치가 가능하다.

**5장 EAI 구축시 표준화 방법**

EAI를 구축함에 EAI 표준화는 중요하다. 표준화에는 인터페이스와 실시간 모니터링 표준화[5]가 있다. 인터페이스 표준화는 기존의 다양한 인터페이스를 정리하고 명명규칙, 생성규칙등을 정리하여 표준화하는 것이며, 실시간 모니터링 표준화는 처리결과 및 처리상태를 실시간으로 모니터링 하는 것이며 본 논문에서는 데이터 내용 추적 프로그램(C API)을 통하여 구현 증명하였다.

5.1. 인터페이스 표준화 정의 및 구현 예제

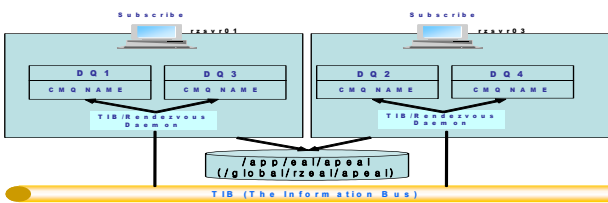
EAI는 인터페이스의 이름만으로도 각종 정보(송신시스템,수신시스템,업무명,업무형태,응답형태)을 알 수 있어야 한다.

- (1) EAI 인터페이스 표준화 명명 규칙 적용 예제
  - ADB : 시스템명+Serial # + (송신:P(Publisher))
  - (예제) BZ + 001\_ + PUB

5.2. 실시간 모니터링 표준화 정의 및 구현 예제

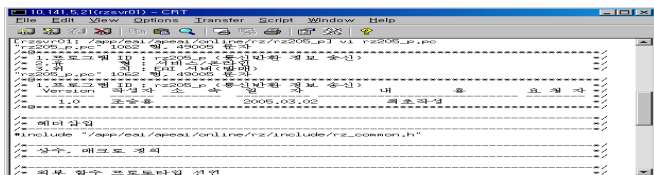
EAI 대상 시스템 및 관련 자원을 관리하기 위한 통합 모니터링 시스템과 인터페이스 정합성 관리를 위한 메시지 Tracking 시스템을 데이터 추적 프로그램으로 구현하여 해당 프로세서의 로그 추적 시스템을 통하여 구현하였고. 통합 운영관리 시스템(SMS)과 연계하여 EAI 담당자에게 SMS 장애 메시지 전송을 통해서 즉각적인 장애복구가 가능토록 했다.

6장 D.Q(Distributed Queue)방식의 Load Balancing 개념 정의 및 구현 예제



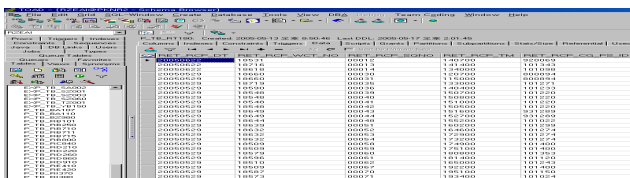
(그림 12) D.Q 구현 다이어그램

데이터가 여러 프로세스에 의해 처리가 되므로 Performance 향상되며(Load Balancing)[4] 프로세스 전체가 다운되지 않는 한 처리가 계속되므로 높은 안전성이 보장된다.



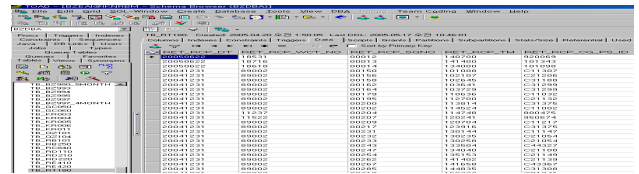
(그림 13)개발 표준 라이브러리 (송신측 C API)

위의 개발 소스에서 DQ방식을 통하여 2대의 서버에서 실행하여 스케줄러가 각 서버의 상태에 따라서 job을 분배하여 멀티프로세싱을 한다.



(그림 14) 송신측 P\_TABLE의 데이터

송신측에서 데이터를 insert 하면 P\_TABLE에 DATA를 넣고 Polling 후에 data를 보낸다



(그림 15) 수신측 TABLE의 수신한 데이터

위에서 수신측 DB(BZEI/BZEAI@IKNRBM)에 DATA가 들어왔음을 알수있다.

7 장 결론 및 연구 방향

위 논문에서 살펴보았듯이 기존 EAI 구축시에 문제점으로 대두 되었던 수신측 및 네트워크 장애시에 송신한 데이터의 전달을 보장하지 못하였으나 본 논문에서 이의 문제점을 개선하는 방법으로 C.M(Certificate Message)의 Ledger File을 통한 송신되지 못한 데이터를 재송신함으로써 Data의 전달을 보장하였고 이를 증명하기 위하여 Cobol API 프로그램 구현을 통하여 데이터의 정합성 보장 및 전체 transaction 처리에 문제가 없게 하였다. 그리고 EAI 구축시에 중요한 요소인 EAI 표준화를 통하여 향후의 추가 시스템을 확장할 경우에 표준화룰을 적용하여 신속한 개발 및 유지보수의 편리함을 가능하게 하였다. 마지막으로 앞으로의 연구과제는 Workflow 개념의 비즈니스 프로세서의 통합을 이루는 것이다.[6] 이를 통하여 비로소 완전한 전사적 통합(EAI)이 될 것이다. 그 예로 BPM(Business Process Management)을 통한 프로세스 관리의 자동화 [7,8]가 필수적인 요소로 등장 할 것이다.

참고 문헌

- [1] White Paper, 'Managing EAI:Framework or Specialist Tools?', Saint consulting Limi
- [2] Teresa Jones, 'Middleware Options', Butler Group, 2003.Feb.
- [3] Accenture.com, <http://www.accenture.com>.
- [4] tibco.com, <http://www.tibco.com>.
- [5] 정중훈 논문지 "능협사례를 통한 EAI 표준화 방안" p8~9 2004.
- [6] 전자상거래표준화통합포럼, "시스템 통합 절차 지침", 전자상거래표준화통합포럼, 2001.12.
- [7] A case study from IDC. Editing by Tony M. Brown, 'Banking on IBM to build an Integrated e-Infrastructure', The Business Integrator Journal, Winter 2001.
- [8] 디지털타임즈, <http://www.dt.co.kr>, 2003.12.22