

HUB&Spoke기반 비동기 메시징 방식의 EAI 모니터링 방안

서용석*

*고려대학교 컴퓨터정보통신대학원 소프트웨어공학과
e-mail:dollmaster2@naver.com

A Method to Monitoring of HUB&Spoke-based Asynchronous Messaging on EAI

Yong-Seok Seo*

*Dept of Software Engineering,
Graduate School of Computer & Information Technology,
Korea University

요 약

전사적 어플리케이션 통합(EAI)은 메시지 전달, 메시지 라우팅, 메시지 변환 등 시스템 간 데이터 및 어플리케이션 통합을 위한 기능을 제공하고 있다. 하지만 EAI솔루션은 Spoke 시스템 및 EAI HUB의 시스템 안정성과 메시지 전달 신뢰도 확보 방안에 대한 표준이 제공되고 있지 않다.^[1]

본 논문에서는 구간별, 유형별로 발생하는 시스템 및 어플리케이션 모니터링과 업무 데이터가 EAI를 거쳐 전달되는 과정에 대한 인터페이스 모니터링을 종합적으로 관리하는 방법으로 EAI 솔루션에 대한 상태 파악과 업무 인터페이스 데이터의 진행 상황을 실시간으로 확인 가능하도록 구축 한 사례를 들어 EAI의 안정된 운영과 신뢰도를 향상시킬 수 있는 방안을 제시한다.

1. 서론

EAI(Enterprise Application Integration)란 네트워크 프로토콜이나 데이터베이스, 운영체제 등과 상관없이 기업 내부의 다양한 시스템과 어플리케이션 및 데이터를 유기적으로 연동하여 이를 통합 관리, 사용할 수 있는 환경을 구축 할 수 있도록 해주는 구현 기술이다.

EAI는 가트너 그룹이 발표 한 이래 1990년대 말경부터 국내에 도입이 되기 시작했고, 포스코, 삼성전자, LG전자 등을 시작으로 전사적 어플리케이션 통합, 연계 방법으로 널리 쓰이게 되었다. 현재는 대형 제조업체뿐만 아니라 정부/공공, 금융, 정보통신 등 전 사업군으로 확장되어 이미 상당부분 구축이 완료 되었지만, 금융권 EAI차세대 수요가 줄을 잇고 있어 여전히 매력적인 기술이라 할 수 있다.

이러한 EAI솔루션으로는 IBM, BEA, VITRIA등의 여러 제품들이 있고, 본문에서는 세계적으로 높은 점유율을 가지고 있는 IBM사의 Websphere제품군의 MOM(Message Oriented Middleware)기반 미들웨어를 도입한 사례를 들도록 하겠다.

한번 도입되면 전사적으로 중요한 기능을 하게 되는 EAI특성상 24시간 365일 무 중단으로 운영 되어야 하며, 시스템 장애로 인하여 다른 정상적인 시스템에 영향을 주어서는 안 되기에 이를 관리할 수 있는 방법에 대한 요구는 지속되어 왔다.^[2]

본 논문에서는 각종 장애 상황에 따라 신속한 대응이 이루어 지도록 모니터링 시스템을 구축 했던 사례를 들어 EAI의 안정된 운영과 신뢰도를 향상시킬 수 있는 방안을 제시한다.

2. EAI 구축 유형

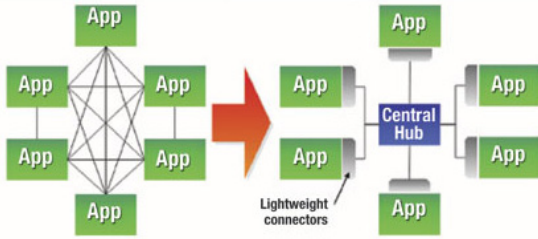
EAI 구축 방식을 크게 나누자면 HUB&Spoke방식과 BUS방식으로 나뉘는데 이에 따른 차이는 <표 1>에 기술되어 있듯 EAI HUB를 중심으로 모든 데이터가 처리되는 HUB&Spoke방식과 BUS를 두고 Adapter가 중심이 되어 처리하는 BUS방식으로 나뉘게 된다. BUS방식에서는 허브 대신 BUS라는 데이터 통로를 통해 전달된 데이터를 adapter가 처리하고, 이에 따른 장애 발생시 adapter 매니저가 분석하여, 백업 후 재처리 하도록 되어 있다.

<표 1> EAI 처리방식 비교^[3]

통합유형	특징	장점
Hub&Spoke방식	Application사이에 미들웨어(Hub)를 두어 처리하는 방법 허브시스템을 통해 데이터를 전송하는 중앙집중 방식	모든 데이터가 Hub를 통해 전송되는 구조. 전송보장, 중앙 모니터링 가능 Spoke 추가시 매넨터만 설치하면 확장 가능
Messaging Bus방식	Application사이에 미들웨어(Bus)를 두어 처리하는 방법 adapter가 Spoke와 Bus를 연결하는 구조	데이터 전송로 역할을 수행하는 버스를 통해 병목현상 및 실패 최소화

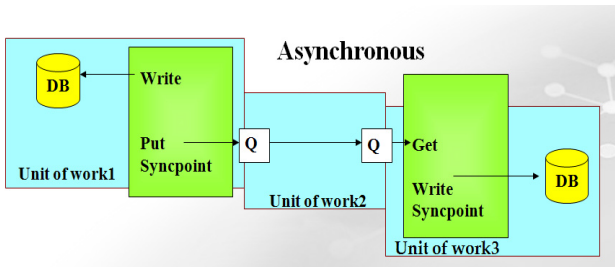
HUB&Spoke방식의 경우 모든 데이터는 Spoke시스템의 다수의 어플리케이션에서 HUB를 거쳐 가공되거나 분산되어 전달되도록 설계되어 있다. Spoke시스템과 HUB시스템은 MOM기반 미

들웨어인 MQ(Message Queue)를 이용하여 연결된다. 이러한 중앙 집중식 연결은 (그림 1)과 같이 어플리케이션들의 단일 경로를 만들어 링크들 간 생기는 n^2 의 링크를 n 으로 줄여주며, Spoke는 독립적 자원을 가지고 HUB가 중재하기 때문에 타 Spoke에 방해 되지 않게 특정 Spoke application 들을 수정하거나 삭제하기 쉽다.^[4]



(그림 1) HUB&Spoke방식

HUB는 message transformation, validation, routing 등의 기능을 필요로 하며 비동기 메시징(Asynchronous Messaging)방식을 기반으로 동작한다. 비동기 메시징 방식은 업무 혹은 work를 하나의 transaction 단위로 처리하는 Synchronous 방식과 달리 Message Queue를 기준으로 work를 나누어 처리하는 방식을 말한다. (그림 2)는 DB동기화를 위해 비동기 메시징 방식을 사용한 예시 이다. 데이터의 흐름은 DB에서 큐로 전송, 큐에서 큐로 전송, 큐에서 DB로 반영 등으로 각각의 transaction이 나누어 처리하게 된다.



(그림 2) 비동기 메시징 방식의 예

HUB&Spoke방식은 중앙 집중식으로 모든 데이터가 HUB로 집중되므로 모니터링이 편리하지만 HUB시스템에 막대한 부하를 주게 되고, HUB시스템에 장애가 발생하면 EAI로 구성된 전체 시스템이 제 기능을 할 수 없게 될 정도로 중요도가 높아진다. 또한 장애가 발생 하였을 경우 어느 단계에 어떠한 문제가 발생하였는지 파악하기 힘든 단점이 있다. 이러한 단점을 극복하기 위해 모니터링 및 장애처리 방안에 대한 연구가 계속되고 있다.

3. EAI 모니터링 관련 연구

EAI관련 솔루션이나 EAI구축 프로젝트에서 사용되어 왔던 모니터링 방법은 어플리케이션 관리 기능을 가진 솔루션이 모

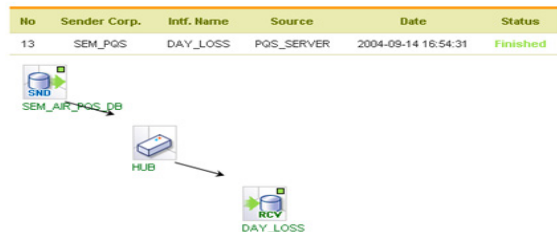
니터링 및 장애처리 기능을 수행하는 것이다. 어플리케이션 관리 기능은 분산 환경에서 수행되는 중요한 어플리케이션들이 중단 없이 수행되기 위해 필요한 필수 기능으로서 HUB에서 어플리케이션의 수행 상태를 감시하고 어플리케이션의 이상 상태 발생 시 원격에서 조치할 수 있는 기능을 제공한다.^[5]

EAI장애 처리 관련 연구는 주로 EAI솔루션에 대한 모니터링 위주가 되어 왔고, 기타 발생 할 수 있는 시스템의 H/W 상태 및 네트워크 상태를 감시하기 위한 기능은 타 솔루션에 의존하는 기타 영역으로 치부되어 왔다. 하지만 EAI솔루션의 상태와 관련 없는 CPU점유율, 네트워크 장애, disk 장애등으로 발생할 수 있는 시스템 장애도 EAI 장애 상황으로 이어질 가능성이 있다. 또한 EAI솔루션이나 시스템 장애가 발생 하지 않았더라도 EAI인터페이스에서의 처리 지연 혹은 HUB 및 각 업무 프로그램 상에서 발생하게 되는 오류는 별도의 모니터링 시스템에서 관리하거나 직접 에러로그를 확인하는 방법으로 처리해야만 했다. 대부분의 EAI 구성에서 모니터링 방법은 EAI솔루션 모니터링, 시스템 모니터링, 인터페이스 모니터링 등으로 나뉘어 관리되거나 특정 모니터링에만 치중하는 모습을 보이게 되었다. 따라서 유형별로 나뉘어져 관리되고 있는 모니터링 방법에 대한 통합적 관리 요구가 지속되고 있는 실정이다.

4. 제안하는 HUB&Spoke방식 모니터링 방안

어떠한 시스템이라도 장애가 발생할 수 있는 여지는 있다. 이를 미연에 방지 하는 것이 최선이지만 그 이후의 가장 최선의 방안은 지속적인 모니터링이다. 본문에서는 EAI시스템의 모니터링 영역을 인터페이스영역과 시스템영역으로 나누었다. EAI구성으로 연계, 통합되는 각 업무별로 ID를 부여하여 ID별 인터페이스를 정의 하고, 해당 업무에서에서 발생한 데이터가 목적지까지 전달되는 과정을 인터페이스영역으로 정의하고, EAI 시스템의 하드웨어 및 소프트웨어에 대한 상태는 시스템영역으로 정의한다. 따라서 모니터링 툴도 인터페이스 모니터링 툴과 시스템 모니터링 툴로 나뉘어 동작하도록 설계되어 있다. 인터페이스 모니터링 툴은 실제 업무 데이터가 EAI인터페이스를 따라 이동 될 때의 실시간 데이터 상태를 (그림 3)과 같이 모니터링 화면으로 조회 할 수 있도록 하는 툴이다.

: Detail Monitoring



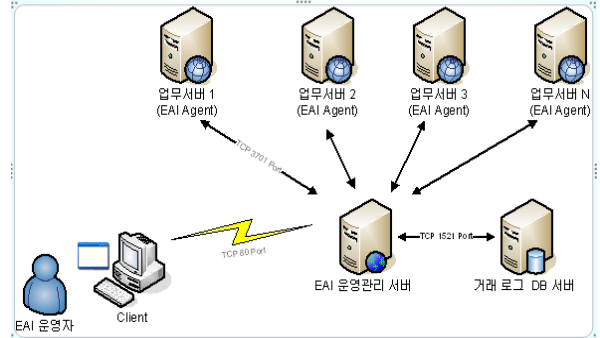
(그림 3) 인터페이스 모니터링 화면

시스템 모니터링 툴은 EAI 인터페이스가 적용되어 있는 시스템의 하드웨어, 소프트웨어 이상 유/무를 체크하여 (그림 4)와 같이 모니터링 화면에서 확인 할 수 있는 툴이다.



(그림 4) 시스템 모니터링 화면

EAI 운영관리 서버로 전송한다. EAI 운영관리 서버로 종합된 정보는 운영자가 모니터링 화면에서 쉽게 확인 가능하며 장애 발생 시 모니터링 상태의 색상 변화와 에러로그 확인 기능으로 신속한 장애 파악이 가능하다.



(그림 5) 시스템 모니터링 구성도

4.1 인터페이스 모니터링

업무 인터페이스별로 발생한 데이터는 Message Queue 방식의 미들웨어를 통해 전송되게 되는데, 각 EAI 시스템은 크게 송신/HUB/수신으로 나뉘게 되고 처음 메시지를 생성하는 송신 노드를 기준으로 메시지의 헤더 영역에 모니터링 정보를 담아 업무 데이터와 함께 전송한다. 각 노드에서는 메시지의 헤더 영역의 정보를 바탕으로 이벤트 메시지를 생성하여 모니터링 서버로 전송한다. 이후 각 노드를 거칠 때 마다 노드의 시스템 정보와 상태를 추가하여 이벤트 메시지를 EAI 운영관리서버로 전송한다. 운영관리 서버에서는 각 단계별 이벤트 메시지들을 종합하여 인터페이스의 상태를 처리중, 완료, 에러 등으로 나누고, 각 노드별 이벤트 발생시간을 표시 해 준다. 에러가 발생했을 경우 에러내용을 표시하여 장애 진단이 가능하도록 한다. 다음은 인터페이스 모니터링 툴의 동작 원리를 설명한다.

- (1) HUB 및 Spoke 시스템의 adapter, broker 등 각각의 노드로 분리하여 각 노드에서 데이터의 전송이 일어 날 때 마다 이벤트 메시지를 발생시켜 EAI 운영관리 서버로 전송한다.
- (2) EAI 운영관리 서버는 각 노드에서 발생한 모니터링 정보를 DB에 저장.
- (3) DB에 저장된 이벤트 정보를 이용하여 웹 화면을 통해 인터페이스 진행 상황 및 현재 상태를 조회.

4.2 시스템 모니터링

시스템 모니터링 툴은 업무나 인터페이스 처리 상황과는 상관없이 각 노드 시스템들이 정상적으로 동작하고 있는지에 대한 상태를 모니터링 한다. 데이터를 전송 해 주는 Message Queue 방식의 미들웨어의 오브젝트 상태, 중요 프로세스의 동작 여부를 일정 간격으로 체크하여 그 결과를 (그림 5)와 같이

다음은 시스템 모니터링 툴의 동작 원리를 설명한다.

- (1) Spoke 시스템에 설치 된 각각의 Agent가 일정 간격으로 서버 및 프로세스 등의 자원 모니터링 정보를 EAI 운영관리 서버로 전송.
- (2) EAI 운영관리 서버는 정보를 DB에 저장하고, Agent와의 통신 상태를 체크한다.
- (3) EAI 운영관리 서버는 DB 정보를 모니터링 화면에 제공 하고, 설정된 Alarm Level에 따라 경고메일 혹은 SMS 메시지를 보낸다.

4.3. EAI 통합모니터링 적용

EAI 운영관리 서버에 집중된 인터페이스 정보와 시스템 정보는 업무 처리 상태와 시스템 상태의 파악을 용이하게 해 준다. 예를 들어 DB 변경내역을 목적지 DB에 반영하는 batch성 업무 인터페이스에서 송신측 DB의 변경내역을 Message로 만들어 Queue를 이용해 HUB를 통해 전송 했을 때, 수신측 DB에 반영하는 과정에서 query 에러 혹은 key 중복 에러 등으로 실패 할 수 있다. 비동기 메시징 방식에서는 송신과 HUB transaction이 이미 commit 상태이기 때문에 에러에 대한 rollback이나 자동 재처리가 어렵게 된다. 인터페이스 모니터링을 사용하면 송신노드, HUB노드는 정상처리 되었으나 수신 노드에 에러가 발생하였고, 대략적인 에러 내용을 확인 할 수 있다. 다른 예로 앞서 설명한 batch성 업무에서 수신 시스템의 DB가 비정상 종료 되었거나 네트워크 이상으로 메시지 전송이 불가능한 경우 인터페이스 모니터링에서는 단지 인터페이스가 진행중이거나 실패했음을 알 수 있을 뿐 정확한 장애 원인 파악은 사용자가 직접 확인해 한다. 이러한 어플리케이션 및 시스템 장애는 시스템 모니터링으로 확인 할 수 있기 때문에 사용자는 EAI 인터페이스 진행 상태와 장애 발생을 EAI 통합모니터링 화면에서 확인 가능하다.

EAI장애 통보 방법은 EAI운영관리 서버로 종합된 모니터링 정보를 EAI통합모니터링 웹페이지를 통해 확인하거나, 메일, SMS등의 장애 통보 수단을 활용 할 수도 있다. <표 2>는 EAI 모니터링 방법에 따라 장애 테스트를 실시하였을 때 감지 가능 여부를 나타내고 있다. EAI통합 모니터링 방법은 인터페이스 모니터링과 시스템 모니터링의 기능을 모두 갖추고 있기 때문에 장애 감지 기능이 뛰어나함을 알 수 있다.

<표 2> 모니터링 방법 비교

테스트 방법		모니터링 방법		
구분	시나리오	인터페이스 모니터링	시스템 모니터링	EAI 통합 모니터링
기능 테스트	S1- MQ Queue 장애 시 alert 발생	X	O	O
	S2- MQ Channel 장애 시 alert 발생	X	O	O
	S3- Adapter 장애 시 alert 발생	X	O	O
시스템 장애 테스트	S4- network 장애 시 alert 발생	X	O	O
	S5- EAI운영관리 Server 종료	X	X	X
	S6- 특정 업무서버 종료 감지	X	O	O
인터페이스 모니터링 테스트	S7- Web UI를 통해 인터페이스 현황	O	X	O
	S8- 인터페이스 시작일, 종료일, 수신 업무 등 세부 정보 조회	O	X	O
	S9- 실패 인터페이스에서 실패노드를 클릭하여 원인 확인	O	X	O

5. 결론

EAI시스템이 차지하는 중요도는 규모에 비례하여 높아지는 만큼 장애에 대한 모니터링 및 처리 방안은 필수적이다. 또한 대부분의 EAI솔루션들은 어플리케이션 통합, 연계 등의 기본 기능에 충실하기 때문에 상대적으로 불안 할 수밖에 없는 모니터링 부분을 충족시키기 위한 여러 어플리케이션들이 개발되고 있다. 기존에 구축되어 운영되고 있는 EAI프로젝트들의 경우 주로 EAI운영관리 틀에서 EAI솔루션들에 대한 모니터링만이 강조되어 업무 처리 현황이나 시스템 하드웨어 및 네트워크 상태에 대한 상황을 알기 어려웠다는 단점이 존재했다. 본문에서 제시한 모니터링 방법은 기존에 널리 사용되던 부분별 모니터링 방법에 비해 각 업무 및 시스템 자원에 대한 모니터링을 종합적으로 처리 할 수 있다는 장점을 가진다.

본문에서는 EAI벤더사들이 보장하고 있는 데이터 정합성, 시스템 안정성에 대한 사용자들의 의혹에 대해 적극적으로 대처 할 수 있도록 정확한 모니터링 방법을 도입함으로써 보다 안정적이고 신뢰성 높은 EAI시스템 구축 방법을 제시했다. 앞으로 EAI 이후 BPM, SOA등의 차세대 기술에서도 모니터링 분야에서 사용 가능 하도록 연구해 나갈 예정이다.

참고문헌

[1] 홍재우 , “EAI에서 수신 시스템 장애 발생시 HUB 시스템 안정성 및 메시지 전달의 신뢰도 향상” , 서강대학교 정보통신대학원, pp.6, 2005
 [2] 권기정, “EAI시스템 장애처리 및 안정성 향상 연구”, 건

국대학교 정보통신대학원, pp.1, 2007

[3] 안병보, “EAI에서 BUS기반 비동기 메시징 방식의 신뢰도 향상 방안” , 서강대학교 정보통신대학원, pp.1, 2003

[4] Loek Bakker, “Goodbye Hub-and-Spoke, Hello ESB? Integration Architecture With BizTalk 2004” , <http://dotnet.sys-con.com/node/121831>, 2005

[5] 윤학수, “EAI 기술을 이용한 OSS/BSS 시스템 통합 전략” , TIBCO Software Inc. Korea, pp.3. 2001