

분산 이기종 환경에서의 메시지미들웨어(MOM)기반 시스템 통합방안 연구

김종배*, 송재영**, 류성열***

요약

전산구조와 기술이 분산환경으로 옮겨가고 있고, 인수 합병 및 프로세스 아웃소싱 등의 증가로 인해, 혹은 조직내의 다양한 시스템이 신규 개발 또는 증설됨에 따라 이기종 플랫폼간의 상호연계미비, 유지보수의 난이함, 데이터 중복성과 일관성 결여 등의 문제점들이 발생되면서 EAI 환경의 도입에 대한 요구가 증가하고 있으나, 많은 비용과 솔루션 선정의 어려움으로 그 적용이 쉽지 않다. 따라서 본 연구에서는 분산시스템 환경에서 이기종 간의 데이터 및 응용프로그램 통합을 위한 효율적인 대안으로 메시지미들웨어를 적용한 시스템통합방안을 제시하였다. 본 논문에서 밝힌 시스템간의 메시지 미들웨어를 이용한 데이터 통합방안은 비용과 성능 측면에서 소규모 시스템간의 인터페이스를 구축할 수 있는 효율적 대안이 될 것으로 기대한다.

Methods to System Integration in Distributed Heterogeneous Environments

Jong-Bae Kim*, Jae-Young Song**, Sung-Yul Rhew***

Abstract

Computing infrastructures and technologies are moving into the distributed environments. Due to increase of M&A, and outsourcing processes, or increase or development of various systems in an organizations, there are various problems rising such as difficulties in maintenance and repairment, repetition or inconsistency of data, and lacks of interconnection between different platforms. All these factors add up to more needs to adopt EAI environments. However it has not been easy because of difficulties in financing and selecting an adequate solutions.

This study presents a method to integrate systems adopting message middleware as an efficient alternative for integration of applications and data between different models under distributed system environments. We expect that the integration method presented in this study, adopting message middleware between systems, will be an efficient alternative to build up interface between small systems in terms of expense and efficiency.

Key words : 메시지 미들웨어, 이기종, 시스템통합, EAI, MOM

1. 서론

전산구조(Computing Infrastructure)와 기술이 분산시스템으로 옮겨가고 있고, 인수·합병 및 프로세스 아웃소싱 등이 일반화 되고 있다. 또한, 기업이나 기관들의 정보시스템 구축에 대한 투자 확대 흐름에 힘입어, 다양한 시스템이 신규 구축되거나 증설되어 왔다. 이러한 현상은 지금에 와서 새로운 문제를 발생시키고 있는데, 이기종 플랫폼간의 상호연계미비, 유지보수의 난이함, 데이터 중복성과 일관성 결여 등의 문제점 등이 그것이다[1],[3].

한편, 최근에 정보시스템 환경이 다양한 업무적 변화가능성에 능동적으로 대처할 수 있는 유연성이

결여되어 있다는 판단에 따라, 그 대안으로 기업어플리케이션통합(EAI: Enterprise Application Integration) 솔루션이 고려되고 있지만, 많은 비용과 솔루션 선정의 어려움으로 인해 쉽게 도입을 결정하기 힘든 실정이다[6].

본 연구에서 사례로 선정한 중앙고용정보원과 근로복지공단 역시, 지난 1999년 부터 고용보험 적용정수업무를 수행하고 있으나, 두 기관에서 업무를 나누어 담당하고 있어 이들 사이의 실시간 업무연계가 중요한 이슈였다. 이러한 문제들에 대한 대안으로 본 연구에서는 두 기관사이의 통합시스템구축에 메시지 미들웨어(MOM: Message-Oriented Middleware)를 적용하여, 이기종 분산 환경의 시스템을 연

* 제일저자(First Author) : 김종배

접수일 : 2005년 7월 15일, 완료일 : 2005년 7월 25일

* 송실대학교대학원 컴퓨터학과

kjb@e-enterprise.co.kr

** 노동부 정보화기획팀

*** 송실대학교 컴퓨터학부 교수

계하는 방안을 제시하고자 한다.

이를 위해 미들웨어의 다양한 유형 및 기능을 살펴보고, 그 중에서도 메시지 미들웨어의 특징등 기술요소를 검토한뒤, 이를 실제 적용, 평가함으로써 시스템간의 메시지 미들웨어를 사용한 데이터 통합 방안이 향후 EAI 시스템환경 구축의 일반적인 대안이 될 수 있음을 보이고자 한다.

2. 관련 연구

2.1 미들웨어의 유형 및 기능

미들웨어의 정의에는 여러 가지가 있지만 기본적으로는 '둘 혹은 그 이상의 소프트웨어 시스템간의 통신(Communication)을 수월하게 해 주는 모든 형태의 소프트웨어'라고 볼 수 있다. 미들웨어는 하나의 엔티티(Entity: 어플리케이션 또는 데이터베이스)를 다른 엔티티와 통신할 수 있게 해 준다. 미들웨어는 소스(Source)와 타겟시스템(Target System)의 복잡함을 숨겨준다. 그 때문에 개발자들은 로우레벨(Low-Level)의 API들과 네트워크 프로토콜(Protocol)들, 그리고 그것들간의 정보 공유에 집중하는 것으로부터 자유로워질 수 있다[2],[7].

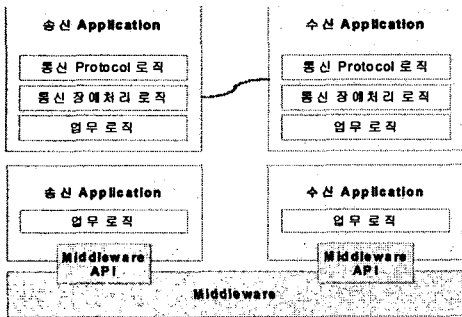


그림 1. 전형적인 Custom 송수신 Application과 Middleware Application의 차이

미들웨어를 유형(Category)으로 분류하는 것은 점점 어려워지고 있다. 그렇지만 다양한 형태의 미들웨어는 각기 특정한 형태의 문제들을 해결한다 [4],[5],[8].

RPCs(Remote Procedure Calls)는 미들웨어 중 가장 오래된 형태이다. 기본적으로 RPCs는 개발자들에게 하나의 프로그램에서 함수를 호출하여 다른 원격 컴퓨터의 프로그램에서 실행되는 함수를 구현할 수 있는 능력을 제공한다. RPCs의 장점은 매커니즘(Mechanism)의 단순함과 쉽게 프로그램 할 수 있다

는 점이다. RPCs의 가장 큰 단점은 커다란 처리능력(Processing Power)과 많은 네트워크 통신을 요구한다는 점이다.

메시지 기반 미들웨어(MOM: Message Oriented Middleware)는 메시징(Messaging)을 이용하여 RPCs의 일부 단점을 보완하기 위해 만들어졌다. 전통적인 MOM은 전형적으로 P2P(Point To Point)간 정보를 이동시키는 데에 메시지를 사용하는 큐잉 방식의 소프트웨어(Queuing Software)이다. MOM이 어플리케이션들간의 통신에 메시지 개념을 사용하고 있기 때문에 직접적으로 미들웨어 매커니즘과 어플리케이션간의 결합은 요구되지 않는다. MOM은 비동기(Asynchronous) 패러다임을 사용한다. 이는 서로 다른 어플리케이션으로부터 분리되어 어플리케이션들이 독립적으로 기능할 수 있게 해 준다. 비동기 패러다임은 개발자들과 사용자들에게 많은 편리함을 제공한다. 비동기 모델이 동기식 보다 좀 더 복잡하더라도 수행되는 도중에 어플리케이션이 블로킹(Blocking)되지는 않기 때문이다. 게다가 MOM은 여러가지 복잡한 매커니즘을 통해 어플리케이션 간의 메시지가 배달되는 것을 보증할 수 있다.

분산객체(Distributed objects)들은 어플리케이션 상호간의 통신을 수월하게 해 주기 때문에 미들웨어라고 볼 수 있다. 그러나 이는 또한 어플리케이션 개발의 매커니즘이기도 하다. 분산객체들은 서로 통신하기 위해 표준 인터페이스(Interface)와 프로토콜(Protocol)을 사용하는 작은 어플리케이션들이라고 할 수 있다.

트랜잭션 미들웨어는 많은 서로 다른 자원들 간에 정보를 이동시키고 메소드(Method)를 공유하는 것을 조정하는 데에 꽤 훌륭한 역할을 수행하고 있다. 그러나 그것들이 채택하고 있는 트랜잭션 패러다임이 메소드를 공유하는 데에는 훌륭한 매커니즘이지만, EAI의 진정한 목표인 단순한 정보 공유에 이르러서는 그리 효과적이지 못하다. 왜냐하면 대부분 트랜잭션 미들웨어는 견고(Tight)하게 결합되어 시스템간의 부하를 가중시키기 때문이다. 게다가 어플리케이션의 소스도 트랜잭션 미들웨어를 위해 변경되어야 하기 때문이다.

2.2 미들웨어 통신모델 (Communication Models)

개념적인 통신모델은 정보가 시스템을 통하여 어떤 방식으로 흘러가는지를 표현한다. 반대로 물리적 통신모델은 그것을 적용하고 있는 기술과 실제로 어떻게 흘러가는지를 표현한다. 미들웨어는 P2P(Point-to-Point)뿐 아니라 M2M (Many-to-Many, One-to-Many포함)설정으로도 동작한다. Point-to-Point 미

들웨어가 다른 종류의 미들웨어에 비하여 단점으로 지적되는 사항은 둘 이상의 어플리케이션 간의 결함을 지원하지 않는다는 점이다[5],[9].

MOM 제품(예를 들어, MQ 시리즈 등)이나 RPCs(예를 들어, DCE 등)와 같은 Point-to-Point 미들웨어들의 목적은 주로 오직 소스와 타겟어플리케이션만 관계된 Point-to-Point 솔루션을 제공하는 것이다. 비록 요즘에는 둘 이상의 어플리케이션을 전통적인 Point-to-Point 미들웨어로 연결할 수 있지만 이렇게 하는 것은 그리 좋은 방법이 아니다. 둘 이상의 어플리케이션을 다룰 때는 너무 복잡한 것들이 관련된다. 둘 이상의 어플리케이션을 Point-to-Point 미들웨어로 서로 연결시키기 위해서는 모든 어플리케이션이 연관된 Point-to-Point 결합이 필요하게 된다.

대부분의 EAI를 요구하는 환경이, 많은 어플리케이션들의 연결을 요구하기 때문에 이런 Point-to-Point 솔루션의 일반적인 개념을 이해한다면 Point-to-Point 미들웨어는 효과적인 솔루션이 아니라는 것을 알 수 있다. 더욱이 이런 시나리오에서 정보를 공유하기 위해서는 어플리케이션이 연결되어야 하고, 정보는 공유된 중앙 서버를 통하여 중개되어야 한다. 다시 말하면 정보를 공유하기 위해서는 메시지브로커(Message Broker)가 필요하다.

Many-to-Many 통신모델은 많은 어플리케이션과 많은 다른 어플리케이션을 연결한다. 이것은 EAI에 가장 적합하며 미들웨어의 경향이기도 하다. 또한 EAI를 요구하는 환경에 유연성과 응용성을 제공하는 가장 강력한 논리적 모델이다.

Many-to-Many 통신모델은 메시지브로커(Message Broker), 트랜잭션 미들웨어(어플리케이션서버, TP Monitor 등), 그리고 분산객체들과 같은 많은 예가 있다. 기본적으로 둘 이상의 소스와 타겟어플리케이션을 한번에 다룰 수 있는 모든 종류의 미들웨어가 이 모델을 지원할 수 있다. Point-to-Point 모델이 그것의 단순함에 장점을 가지는 것과 마찬가지로 Many-to-Many 모델의 단점은 너무 많은 시스템들 함께 연결하는데에 따르는 복잡함이다[7],[9].

3. 메시지 미들웨어의 적용

3.1 현행 정보연계 아키텍처

본 연구에서 사례로 선정된 고용보험 업무의 특징은 다음과 같다.

고용보험은 실업예방, 고용촉진, 근로자의 직업능력의 개발·향상은 물론 생활에 필요한 급여를 지급

하여 실직근로자의 생활안정 및 재취업을 지원하는 사회보장제도로 근로자의 생활안정, 원활한 구직활동을 위해 신속·정확한 고용정보를 제공하는 고용안정사업, 직업능력개발사업, 실업급여 사업을 수행하고 있다.

근로복지공단은 6개 지역본부 및 40개 지사에서 실업자 생활안정지원 사업과 임금채권보장 그리고 고용보험 적용 징수 업무를 담당하고 있다. 정보기술환경은 분산 시스템 구조를 채택하여 3개 지역본부에 정보시스템을 위치적으로 분산된 오픈 시스템이다. 어플리케이션 아키텍처는 3Tier-클라이언트 서버 구조를 채택하여 트랜잭션 미들웨어(Transaction Processing Middle ware)로 텍시도(Tuxedo)를 사용하고 있으며 파워빌더(Sybase PowerBuilder)/ASE(Sybase사의 DBMS)를 사용하고 있다.

산업인력공단 중앙고용정보원은 국민의 능력과 적성에 맞는 직업 선택과 고용안정을 지원하기 위하여 각종 심리검사개발과 상담기법을 개발하고, 취업, 고용보험, 실업대책, 직업훈련 등 노동시장관련 정보를 관리하고 제공하며, 직업정보 및 전망, 노동시장 동향 등을 조사 연구하며 신속·정확한 고용정보의 제공을 통하여 실업자에게 일자리를 찾아주는데 핵심적 역할을 하고 있으며 고용안정종합정보망인 워크넷(Work-Net)과 같은 고용보험 전산망을 운영하고 있다. 정보시스템 환경은 주로 센터에 집중되어 있으며 오픈 시스템에 웹(WEB)환경의 서비스가 많은 구조를 가지고 있다.

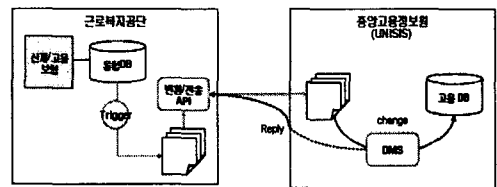


그림 2. 현행 정보연계 아키텍처

주요한 업무 데이터 흐름은 근로복지공단 3개 지역센터의 데이터베이스(DBMS)에 저장되는 업무 데이터가 중앙고용정보원의 통합 고용정보 데이터베이스에 실시간으로 이동하는 유형(One way - Online)이 대부분의 인터페이스 형태이다.

기존의 고용보험시스템은 파일전송(File Transfer)에 의한 간접적 정보연계방식과 단방향 정보연계 체계로, 특히 전달 보증 체계가 미흡하였고, 이에 따라 중앙고용정보원과 근로복지공단 사이의 실시간 업무연계문제가 지속적인 이슈로 제기되었다.

이러한 문제들에 대한 대안으로 본 연구에서는 두

기관사이의 통합시스템구축에 메시지 미들웨어(MOM: Message-Oriented Middleware)를 적용하여, 이기종 분산 환경의 시스템을 연계하는 방안을 제시하였다.

3.2 MOM 적용 아키텍처

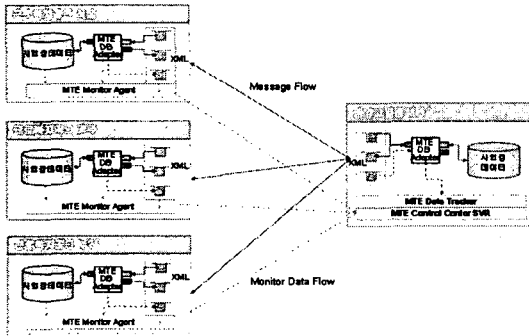


그림 3. MOM 적용 SW 아키텍처

MOM을 적용한 고용보험 연계시스템의 특징은 다음과 같다.

첫째, 메시지 미들웨어 기반 Point-to-Point통신 모델의 적용이다. 신 고용보험시스템은 대부분 데이터의 흐름이 근로복지공단 서버에서 중앙고용정보원 서버로 이동하는 특성이 있으며, 센터간의 통신은 필요 없는 구조적 특징을 갖고 있기 때문에 P2P 통신모델을 사용하여 상대적으로 간단히 구축하면서, 메시지 미들웨어를 적용하여 정보 전달의 안정성 및 시스템간의 독립성을 확보하였다.

둘째, 어댑터를 사용한 XML 형태의 메시지 송수신환경 구축이다. 메시지 미들웨어와 데이터베이스 사이에 적용된 DB Adapter는 DB와 MOM 사이의 트랜잭션을 관리하여 데이터추출 및 로딩 시에 데이터 정확성을 보장하며 GUI 개발 환경을 제공하고 메시지를 XML로 만들어 주므로, 타 시스템간 메시지 교환의 유연성을 확보하여 주며 향후 EAI 시스템으로 확장하기 위한 인프라를 제공한다. 적용된 어댑터는 대상시스템으로부터 데이터를 추출하여 전송매체, 즉 MOM까지 전달하는 역할을 한다.

셋째, 어댑터 개발환경 및 시스템 모니터링 환경의 구축이다. 어댑터가 제공하는 데이터 트래킹 기능을 적용하여 인터페이스 발생에 대한 추적이 가능하게 되었으며 장애발생시 즉시 담당자에게 통지할 수 있는 기반을 구축하였고, 또한 구축된 모니터링 DB 로부터 전송에 대한 통계를 제공할 수 있게 되었다.

현재 정보시스템 환경에서 어플리케이션이나 데이

터베이스와 같은 자원을 연계시키도록 해주는 토대 기술은 '미들웨어'로 알려져 있다. 그러나 대부분의 미들웨어는 데이터 수준의 통합만을 수행할 수 있다. 미들웨어는 IT 소스의 통합을 지원하는 통신 프레임워크만을 제공할 뿐 이러한 자원들이 서로 완벽하게 통신하도록 보장하는 것은 아니다. 통합 프로젝트에 진정한 가치를 더해주는 EAI 제품은 미들웨어를 이용한 데이터 수준 통합 이상의 역할을 수행해야 한다. 소프트웨어를 통해 부분적으로 자동화(또는 지원)될 수 있는 의미론적 통합 수준이 높을수록 그 소프트웨어의 가치도 높아진다. EAI 제품 중 중요한 기능은 패키지형 자원 인터페이스이다. '커넥터' 또는 '어댑터'로도 알려져 있는 이들 인터페이스는 어플리케이션과 데이터베이스를 통해 정보를 획득해야 하는 개발 상의 부담을 완화해준다.

현재 대부분의 EAI 프로젝트는 어플리케이션간에 커스텀 포인트 투 포인트 커넥션을 이용하고 있다. 즉, 2개의 어플리케이션이 서로 메시지를 주고받을 수 있도록 개발자들이 이들 어플리케이션간에 로우 레벨 통신 코드를 작성하는 것이다. 이는 일부 프로젝트의 경우 단기적으로 보면 가장 비용효율적인 방법임이 입증됐다. 예를 들어 명확하게 정의된 2개의 어플리케이션 로직 간에 가능한 한 빨리 통합이 이루어져야 할 경우 EAI 제품을 이용하면 EAI 제품의 아키텍처 및 틀에 익숙해지기까지 커스텀 코딩 방식보다 시간이 더 오래 걸릴 것이다. 그러나 이러한 전통적인 커스텀 방식은 어플리케이션이나 시스템, 또는 사업에 변화가 생길 때마다 코드를 재 작성해야 하는 등 커넥션 코딩에 어려움이 따르고, 시간이 지나면서 이러한 어플리케이션과 커넥션의 수가 늘어나면 비즈니스 시스템을 결합하는 코드가 서로 뒤섞여 손댈 수도 없는 지경에 이르게 될 것이다. 따라서, 진행 중인 통합 프로젝트가 복잡한 것이거나 시간이 경과하면서 관리해야 하는 인터페이스의 수가 수십 개로 늘어나면 커스텀 개발 통합 솔루션보다는 EAI 제품에 투자하는 것이 중장기적으로 볼 때 보다 비용 효율적인 방법이다.

3.3 적용 결과 테스트

다음으로 본 연구에서는 중앙고용정보원의 고용보험시스템과 근로복지공단간 정보연계를 위해 메시지 미들웨어(MQ)의 설치 및 운영에 따라 기존 운영 시스템에 미치는 영향을 평가하고 문제점을 사전에 도출하여 대응하기 위해 근로복지공단의 실 운영환경에 메시지 미들웨어를 설치하여 스트레스 테스트(Stress Test)를 수행하여 영향 평가를 수행하였다.

근로복지공단의 센터 중 시스템 사양이 가장 낮은

단계로 운영 중인 지방의 한 센터를 대상으로, 정보 처리량이 가장 많은 시간대(Peak Time)의 트랜잭션(DB Transaction) 양을 고려한 정보연계 처리 후, 이에 따른 시스템 성능 부하 정도를 측정하였다. 성능 측정은 운영서버에 대한 테스트 진행으로 보다 정확한 메시지 미들웨어의 시스템 영향 평가를 위해 1,2차 테스트를 진행 하였으며, 어댑터의 데이터베이스 처리(DB Handling) 성능 향상을 위한 데이터베이스 튜닝(SQL 튜닝) 후 3차 성능 영향 평가를 실시하였다. 예상 스트레스 테스트 근거는 Peek Time시 1일 예상 Transaction : 165,400회(연계 대상 테이블에 대한 Insert, Update, Select에 대한 예상 Transaction 수), Busy Time Zone(10:30~11:30, 3~6시)에서 요구 되는 초당 Transaction 처리 건 수 약 12.25회(2,325 Byte)를 기준으로 수행하였다.

또한 성능 측정 방법은 HP-UX에서 제공하는 Glance 명령 사용하여 CPU 사용률, Memory 사용률, Disk 사용률(Multi Disk중 I/O 발생률이 가장 높은 Disk의 사용률)을 측정하였고, Top 명령을 사용하여 구동 중인 전체 프로세스 중 상위 40개 프로세스별 사용률 측정하였다. 전체 스트레스 테스트의 제약 조건은 실 운영환경이기 때문에 I/F Table에 연계 자료 insert는 별도 프로그램에 의해 구현 되었으며(실제는 DB Trigger가 동작), 동일시점, 동일환경이 아니기 때문에 시점에 따라 테스트 결과에 오차가 발생 할 수 있다. 3차례에 걸친 테스트 결과를 요약하여 다음과 같이 표로 정리하였다.

표 1. 3차례 테스트 결과 요약표

구분	테스트 내용	테스트 결과
1차 테스트	MQ 구동 후 시스템 사용량 측정	<ul style="list-style-type: none"> MQ구동 전과 비교시 시스템 사용량 차이가 없음 (CPU/Memory/Disk 51%/85%/23%)
	DB Transaction과 데이터 전송처리 중 시스템 사용량 측정	<ul style="list-style-type: none"> Memory 사용률 변화 : +1% 이하임 Disk 사용률 변화 : Queue 사용량에 따라 증가(Disk IO) CPU 사용률 변화 : +10~20% 증가함(약 60~70%)
2차 테스트	MQ 종료 후, DB Transaction만 수행 후 시스템 사용량 측정	<ul style="list-style-type: none"> Transaction 수행 자체에는 시스템 성능의 영향이 거의 없음(DB작업으로 분산 됨)
	데이터 전송 처리 중 시스템 사용량 측정	<ul style="list-style-type: none"> CPU 사용률 변화 : +10~20% 증가함(약 60~70%)
3차 테스트	SQL 튜닝(인덱스 생성)후, 1차 테스트 내용 수행	<ul style="list-style-type: none"> Memory 사용률 변화 : +1% 이하임 Disk 사용률 변화 : Queue 사용량에 따라 증가(Disk IO) CPU 사용률 변화 : +5~10% 증가함

결과를 분석해보면 메시지 미들웨어의 설치 또는 가동, 근로복지공단의 자체 DB Transaction 작업은 서버에 영향을 거의 주지 않고 있음을 보여주고 있다. 한편, 테스트 결과 발생한 대량의 I/F 테이블 정보를 메시지 미들웨어에 의해 전송하는 과정에서 CPU 부하가 5~10% 증가한다. 그러나, 기존의 전송 방식(I/F 파일 정보를 Tuxedo 모듈이 중앙고용정보원 Unisys에 전송)과 비교했을 때, 부하의 차이가 많지 않을 것으로 예상되며, 신 고용보험시스템 가동 이후 기존 방식은 사용하지 않게 되므로 시스템에 대한 영향은 현재와 많은 차이가 없을 것으로 판단 된다.

4. 결론

최근에 정보시스템 환경이 다양한 업무적 변화가 능성에 능동적으로 대응이 가능하기 위해 많은 프로젝트에서 EAI 솔루션이 도입되고 있지만 상대적으로 소규모 환경에 적용하기에는 비용적인 부담과 적용솔루션의 선정 어려움이 있어 왔다.

본 논문에서 검토한 사례인 중앙고용정보원과 근로복지공단 사이의 사업장 정보연계시스템 연계방식은 EAI를 구축하기 위한 솔루션들 중에서 메시지미들웨어(MOM)와 어댑터(DB Adapter) 그리고 모니터링(Monitoring) 솔루션만을 이용하여 Point-to-Point 통신모델로 적용한 사례로 앞의 테스트 결과와 같이 +5~ 10% 정도의 부하로 비용과 성능 측면에서 소규모 시스템간의 인터페이스를 구축할 수 있는 구체적 대안이라고 하겠다.

시스템간의 연계 필요성은 업무환경의 변화에 맞추어 점차 증대되고 있다. 현재 많이 구현되고 있는 P2P 환경의 EAI 데이터 통합은 현재 환경에서의 업무적 필요성만을 반영하고 있다. 그러나 향후 시스템 요구사항이 타 기관연계나 대국민 서비스 등의 다양한 요구를 해결해야 하는 경우에 본 연구에서 제시한 메시지 미들웨어를 바탕으로 하여 메시지 브로커(Message Broker)의 검토가 추가적으로 필요하다.

메시지 브로커의 사용이 필요한 대부분의 업무적인 요건은 한 개의 송신시스템에서 여러 개의 수신 시스템으로 데이터가 전송되는 경우이다. 이런 요건이 발생하면 P2P 통신모델에서 Many-to-Many 통신 모델로의 전환을 고려하고, 비용 및 구현의 용이성, 관리의 용이성등을 고려하여 메시지 브로커의 사용을 검토하여야 한다. 향후 본 연구의 결과를 토대로

메시지 브로커를 도입한 아키텍처와 이에 따른 성능 영향을 연구할 필요가 있다.

참 고 문 헌

- [1] 김경민 외, “이종의 컴포넌트 미들웨어 프레임워크 간의 상호운용성을 위한 모델 설계”, 한국정보과학회 학술발표 논문집, pp.13-15, 2003.
- [2] 윤은영 외, “컴포넌트 기반 미들웨어 기술”, 정보처리 제8 권 제5호, 한국정보처리학회, 2001.
- [3] 홍정기, EAI 구현전략과 사례, 시사컴퓨터, 2002.
- [4] MATJAZ B.JURIC 외, PROFESSIONAL J2EE EAI, 정보문화사, 2002.
- [5] JP MORGENTHAL, XML과 JAVA를 이용한 엔터프라이즈 응용 통합, 영한출판사, 2002.
- [6] 남세진, “Intelligent 비즈니스 기술 : EAI 시스템에서 웹 서비스의 구현”, 한국전자거래학회 종합학술대회, 한국전자거래(CALS EC)학회, 2002.
- [7] IBM Redbooks, An EAI Solution Using Websphere Business Integration V4.1, Vervante, 2003.
- [8] Rowell, Michael, Understanding EAI, Macmillan Computer Pub, 2000.
- [9] Rosemary Rock-Evans, “Middleware the key to Distributed Computing”, Overn, 1995.

김 중 배



1996년 2월 : 서울시립대학교
경영학과 졸업(학사)
2002년 8월 : 숭실대학교 정보과학
대학원 정보산업학과 졸업(석사)
2004년 8월 : 숭실대학교 대학원
컴퓨터학과 박사과정(수료)

관심분야 : 정보보호, 소프트웨어 개발 방법론, 에이전트 시스템, BPM, 웹서비스

송 재 영



1988년 2월 : 단국대학교
경영학과 졸업(학사)
2002년 8월 : 숭실대학교 정보과학
대학원 졸업(석사)
2004년 8월 : 숭실대학교 대학원
컴퓨터학과 박사과정(수료)

1993년 12월 ~ : 노동부(현재 정보화추진단장)
관심분야 : 정보시스템 감리, 정보보호 등

류 성 열



1997년 2월 : 아주대학교 컴퓨터학부
(공학박사)
1997년 3월 ~ 1998년 3월 :
George Mason University 교환교수

1981년 3월 ~ 현재 : 숭실대학교 정보과학대학
컴퓨터학부 교수

관심분야 : 소프트웨어 유지보수/재사용, 소프트웨어 재공학/역공학, 정보보호 등