

# EAI 시스템 테스트 방법에 관한 연구

임성준\*, 구자경\*, 신석규\*

\*한국정보통신기술협회

e-mail : {joons, jkkoo, skshin}@tta.or.kr

## A Study on method of EAI solution testing

SungJoon, Lim\*, JaKyung, Koo\*, SeokKyoo, Shin\*

Telecommunications Technology Association

### 요 약

최근 부각되고 있는 기업 업무 프로세스 통합을 위해 EAI(Enterprise Application Integration) 시스템을 도입하고 있다. 하지만 아직 EAI 시스템의 품질을 보증하기 위한 테스트 방법이 제공되고 있지 않는 실정이다. 본 논문에서는 EAI의 기능을 이루는 요소별로 분리하고, 각각 분리된 기능별로 테스트를 수행하는 방법을 제시하고자 한다.

### 1. 서론

최근 들어 외부 기업과의 비즈니스 연계의 필요성이 더욱 부각되고 있으며, 더불어 기업 내부 또는 기업간에 사용할 수 있는 다양한 어플리케이션들을 기업의 비즈니스 프로세스 안에서 통합할 필요성이 증대되고 있다[1]. EAI는 기업의 통합을 위한 이기종 기업 어플리케이션과 데이터, 비즈니스 프로세스 간의 상호운용성 향상을 위한 표준 아키텍처 기술 등을 의미하며 기업의 민첩한 비즈니스를 가능케 하는 시스템이다[2].

나아가서 기업 내부의 어플리케이션 통합 뿐만 아니라 타 기업과의 연계를 의미하는 B2Bi (Business to Business integration)의 개념을 포함한 eAI (e-Business Application Integration)으로 패러다임이 전환되고 있다. 이러한 움직임에 따라 많은 기업들이 통합에 관심을 가지고 있으며, EAI 구축 수요가 증대되고 있다. 그러나 아직까지 구축과정에서 품질을 보증하기 위한 테스트 절차 혹은 구축 이후에 테스트를 위한 정형화된 방법론은 제공되고 있지 않는 실정이다. 이는 EAI구축 사례에 대한 참고할만한 정보가 많지 않고 아직까지 성공적인 프로젝트 수행을 위한 가이드가 없기 때문에 해석해 볼 수

있다.

따라서 본 논문에서는 EAI 시스템의 품질 향상을 위한 테스트 방법에 관한 일반적이고 단순화된 모델을 제시하고자 하였으며, 이는 기존에 구축된 EAI 시스템 혹은 구축과정에 있는 프로젝트에서 기초적인 테스트 프레임워크로 사용될 수 있을 것이다.

### 2. EAI의 개념과 구조

EAI란 단일 기업 내 이기종 어플리케이션을 통합하는 것으로 새로운 미들웨어를 이용하여 비즈니스 프로세스를 중심으로 기업 내의 각종 어플리케이션을 통합하는 과정 및 관련 기술을 의미한다[3]. 일반적으로 EAI에서 제공하는 기능들은 (표 1)과 같이 데이터브로커, 메시지 버스, 어댑터, 비즈니스 프로세스 관리로 나뉘볼 수 있다.

본 논문에서는 EAI를 구축중이거나 기 구축된 시스템 즉 EAI시스템으로 지칭하기로 한다. 더불어 EAI를 구성하는 요소들에 대한 명칭의 통일이 이뤄지지 않았고 아직까지 표준화되지 않은 다양한 형태의 EAI가 공존하는 현실을 반영하여 EAI의 구성요소를 비즈니스 프로세스 관리, 데이터 브로커, 어댑터, 어플리케이션, 메시지 버스의 4개 요소로 나누었

다. 이는 EAI가 기능적 요구사항을 충분히 반영하였는지를 테스트해 보기 위한 기본적인 기능이다.

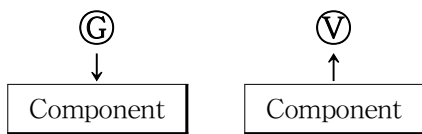
요 소	분 류
데이터 브로커	서로 다른 형태의 데이터 통합을 위한 데이터 포맷과 코드 교환
메시지 버스	표준화된 데이터를 다른 어플리케이션으로 전송
어댑터	다른 어플리케이션과 메시징 통로간 데이터 입출력
비즈니스 프로세스 관리	각 업무 시스템 및 어플리케이션 상호간의 데이터 교환과 각 업무에 대한 흐름을 어떤 시점 혹은 어떤 이벤트에 따라서 어디로 업무가 진행되어야 하는지를 정의하고 운용

(표 1) EAI 구성요소

### 3. EAI 시스템 테스트

일반적으로 자동화되어 있는 테스트의 경우 시스템을 통해 출력되는 결과 값과 시나리오를 통해 유추해낸 값을 서로 비교해 봄으로써 프로그램이 올바르게 작동하는지를 알 수 있다. 입력 값을 동일하게 설정하고 테스트를 위한 시나리오에 적용한 출력과 실제 프로그램을 통해 산출되는 결과 값이 일치하지 않을 경우 해당 프로그램이 올바르게 작동하지 않고 있음을 가정해 볼 수 있다.

테스트의 대상이 되는 단위는 테스트의 특성에 따라 여러 종류가 있을 수 있으며 본 논문에서는 입출력 인터페이스를 가지고 있는 컴포넌트(Component)를 하나의 단위로 설정한다.

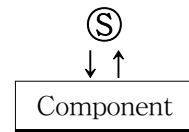


(그림 1) 데이터 입출력

(그림 1)에서 G는 데이터 입력자(Generator)를, V는 데이터 검증자(Verifier)를 의미한다. 데이터 입력은 다양한 방법으로 이뤄질 수 있다. 스크립트 파일이나 데이터베이스로부터 읽어 들인 데이터를 입력할 수도 있고, 다른 프로그램이나 시스템으로부터 얻은 데이터를 입력할 수도 있다. 또한 랜덤(random)하게 생성하거나 특정 알고리즘을 이용하여 제조한 데이터를 이용하여 입력할 수도 있다.

데이터 검증은 테스트 시나리오에 따라 기대한 값과 실제 출력되는 데이터를 비교해 보는 과정이다. 하나의 어플리케이션이 여러 컴포넌트의 유기적인 결합으로 이뤄졌다고 가정할 때, 출력 값을 또 다른 컴포넌트의 테스트를 위한 입력 값으로 사용할 수 있다. 따라서 테스트를 하고자 하는 컴포넌트의 데이터 검증 결과는 다른 컴포넌트의 테스트에도 영향을 미치게 된다.

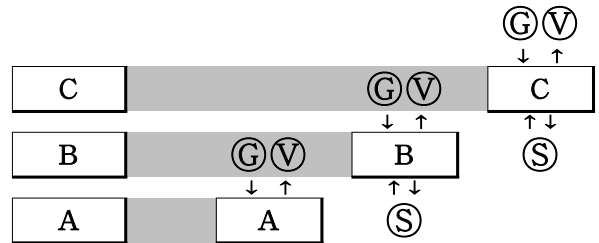
따라서 테스트를 거쳐 검증 받은 컴포넌트와 이를 이용한 다른 컴포넌트의 테스트 과정은 (그림 2)와 같이 나타낼 수 있다.



(그림 2) 테스트 컴포넌트

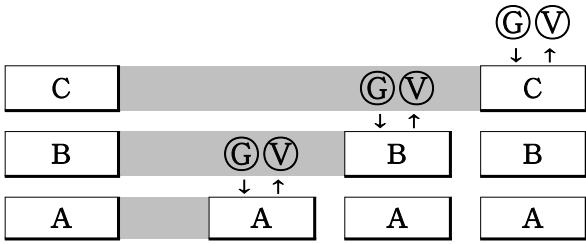
S(Stub)는 테스트를 받고 있는 컴포넌트로부터 데이터를 받아서 다시 컴포넌트에게 유효한 데이터를 입력하는 역할을 한다.

각각의 컴포넌트가 레이어의 형태로 어플리케이션을 구성하고 있다고 가정할 경우 단계별로 레이어를 테스트하여 전체적인 테스트를 가능하게 한다.



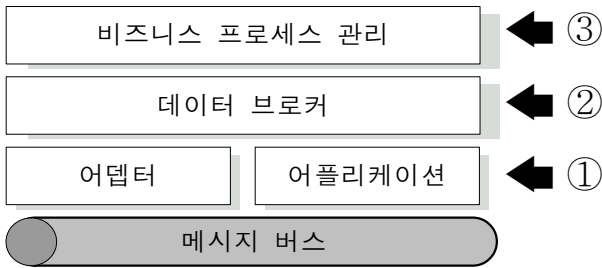
(그림 3) 레이어 개념을 이용한 단위 테스트

통합 테스트를 위해서는 이러한 레이어별로 이뤄진 테스트보다 많은 시간과 비용이 필요하다. 외적 의존성이 높은 최하위 레이어(A)를 먼저 테스트하고 상위로 올라가면서 테스트한다. 예를 들어 'A'레이어와 'B'레이어의 통합단계에서 이상이 발견되지 않았다가 'C'레이어를 통합 시킨 후 결함이 발견되었을 경우, 결함이 발생하는 부분은 두 가지 경우로 생각해 볼 수 있다. 하나는 'C'레이어 자체의 결함이고 다른 하나는 이미 테스트를 통해 무결성을 검증 받은 컴포넌트와 새로 결합되는 컴포넌트간의 상호작용의 오류이다.



(그림 4) 레이어 개념을 이용한 통합 테스트

EAI시스템은 서로 다른 분리된 시스템간의 연결을 가능케 하는 것이다. 이는 일반적인 범용 정보시스템인프라를 통해 이뤄진다. 앞서 설명한 EAI의 기능에 따른 분류를 레이어 개념으로 표현하면 (그림 5)와 같이 나타낼 수 있다.



(그림 5) EAI 시스템 테스트 순서

일반적인 메시지 버스(메시지 인프라스트럭처)는 전자적인 네트워크를 통하여 안전하고 믿을만한 통신을 가능케 한다. 각각의 어플리케이션들은 어댑터를 통해서 메시지버스로부터 데이터를 받거나 보내는 역할을 한다. 데이터 브로커의 기능은 서로 다른 어플리케이션에서 생성되는 데이터를 사용할 수 있도록 형태를 변환하는 역할을 한다. 메시지버스, 어댑터, 데이터 브로커가 서로 다른 어플리케이션간의 신뢰성 있는 데이터 통신의 기능을 수행하는 총체적인 기능을 '비즈니스 프로세스 관리'라고 정의한다.

이러한 EAI 시스템의 특성을 감안하여 레이어에 따른 테스트를 메시지 버스를 최하단으로 가정하고 상위 어댑터, 어플리케이션, 데이터 브로커, 비즈니스 프로세스 관리로 이어지는 'bottom-up'방식의 테스트를 생각해 볼 수 있다.

통합 테스트를 수행하기 전에 어플리케이션 자체는 올바르게 동작하고 있다는 것을 가정해야 한다. 이러한 가정에서 EAI 시스템의 테스트는 어플리케이션에 추가된 인터페이스와 어댑터등에 초점을 둔다.

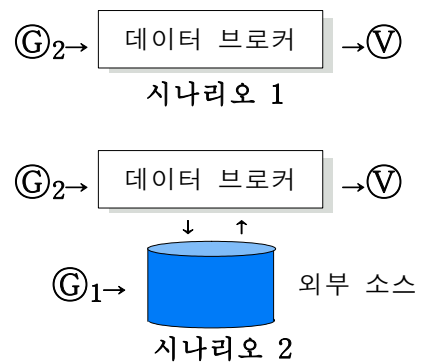
어댑터를 테스트하는데 있어 데이터 입력 방향

및 출력 방향에 따라 다음 (그림 6)과 같이 4가지 경우로 나눌 수 있다.



(그림 6) 어플리케이션과 어댑터 테스트

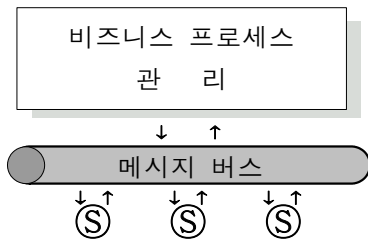
어플리케이션에서 제공하는 API를 통해 어댑터와의 연결이 가능하다면 (그림 6)의 '시나리오 1'과 같은 경우를 가정해 볼 수 있다. 그러나 테스트를 위한 데이터에 대해 직접적인 결과를 얻을 수 없다면 '시나리오 2'와 같이 어플리케이션에 직접적으로 접근하여 결과를 얻어낼 수 있다. '시나리오 3'의 경우 입력하는 데이터는 여러 가지 다양한 형태와 과정을 거칠 수 있다. 또한 데이터 입력을 위해 어댑터와 연결하는 API를 어플리케이션에서 지원해 주지 않는다면 '시나리오 4'와 같이 어플리케이션에 직접적으로 데이터를 입력 후 어댑터를 통해 출력되는 데이터를 검증하는 형태가 가능하다.



(그림 7) 데이터 브로커 테스트

데이터 브로커의 시험은 보다 간단하여 데이터 입력 후 데이터 브로커를 거쳐 출력되는 데이터 검증의 일반적인 절차를 따른다. 여기서 데이터 입력은 단방향의 입력일 수도 있고 외부 소스에 의해 여러 형태의 복잡한 과정을 거치는 입력일 수도 있다.

마지막으로 비즈니스 프로세스 관리의 테스트는 많은 변수와 일정하지 않는 규칙들로 정형화된 테스트 시나리오를 설계하기가 어려우며 많은 논란의 여지가 있다.

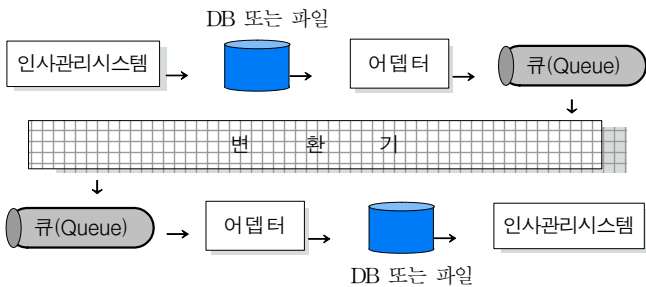


(그림 8) 비즈니스 프로세스 관리 테스트

단순화된 형태를 가정한다면 테스트를 통해 검증된 S(stub)를 이용하여 메시지 버스를 통해 비즈니스 프로세스의 통합된 형태로 테스트를 할 수 있을 것이다.

**4. 적용사례**

이러한 단순화된 방법론을 실제 적용하기까지는 많은 어려움이 따르는 것이 사실이다. EAI 개념이 아직까지 성숙된 단계에 이르지 못했으며, 고객의 요구에 따라 복잡하고 다양한 형태로 맞춤형 되어 구축되기 때문이다.



(그림 9) 실제 적용 사례

A사에서 구축한 EAI시스템에 적용하여 테스트를 수행해 본 결과 그 기능의 경계를 명확히 나눌 수는 없었으나 지금까지 제시한 단순화된 모델을 사용하여 체계적인 테스트 절차를 수립할 수 있었다. A사의 EAI시스템은 두 종류의 서로 다른 인사관리

데모시스템 간의 데이터 교환을 위하여 구축되었다.

테스트를 위한 기능별 구조는 (그림 9)와 같이 나타났다. 각각의 기능에 따른 테스트 방법을 적용하여 전체 통합된 기능 테스트를 가능하게 되었다. 그러나 실제 적용 과정에서 다음과 같은 한계점들이 발견되었다.

- EAI 시스템이 설치될 환경은 서로 다른 시스템들 간의 연동을 전제로 하기 때문에 테스트케이스, 테스트 결과 비교 등이 어렵다.
- EAI 솔루션은 여러 형태의 어플리케이션이나 패키지들을 통합한다. 일반적으로 이러한 어플리케이션의 내부 구조에 접근이 어렵다.
- 대부분의 비즈니스 프로세스는 시간에 종속적이다. 결과 측정을 위해서 비즈니스 프로세스가 진행되어야 할 경우 상당한 시간이 소요될 수도 있다.

**5. 결론**

빠르게 변화하는 비즈니스 환경과 더불어 EAI의 개념과 기본 기능 요소, 구현 방식도 지속적으로 변화하고 있다. 하지만 EAI는 ‘통합’이라는 기본적인 기능을 수행하기 위해 도입되며 본 논문에서는 EAI시스템을 기능적인 측면에서 몇 가지 모듈로 단순화 하고 각각에 관한 테스트 방법을 제시하였는데 의미가 있다. 실제 EAI시스템 테스트에 적용하여 본 결과 아직까지 단순화된 모델의 한계점을 지니고 있었으나 본 논문에서 제시한 방법은 보다 실용적인 테스트 프레임워크를 만들어 나가는 바탕이 될 것으로 기대한다.

**참고문헌**

[1] 이장석, 홍정기, 지정권, “비즈니스 민첩성 향상을 위한 EAI접근 방안 및 구현 전략”, 한국 정보 과학회지, Vol. 22, No. 7, pp.13-21, 2004. 7

[2] 이강일, “통합 영역의 분류에 따른 EAI 구축 방안”, 한국 정보 과학회지, Vol. 22, No. 7, pp.41-45, 2004. 7

[3] 정재원, “EAI 시스템의 도입에 영향을 미치는 요인에 관한 연구”, 한국외국어대학교, 석사학위논문, 2003