

# 웹 서비스 기반 e-비즈니스 응용 프로그램 통합 프레임워크

## (A Web Services based e-Business Application Integration Framework)

이 성 득 <sup>†</sup>            한 동 수 <sup>\*\*</sup>  
(Sung-Doke Lee)    (Dong-Soo Han)

**요약** 본 논문은 인터넷에 연결된 여러 형태의 플랫폼 상에 장착되어 있는 다양한 응용 프로그램 통합을 지원하는 e-비즈니스 응용 프로그램 통합(eAI) 프레임워크를 제안한다. 연결된 응용 프로그램은 프레임워크를 구성하고 있는 워크플로우 시스템에 의해서 구동되고 조정되면서 특정 비즈니스 목적을 달성하게 된다. 프레임워크 구성을 위해서 5개의 하위 프레임워크 구성 모듈이 도출되었으며 도출된 각 모듈의 기능과 역할이 정의되었다. 도출된 5개의 하위 모듈은 비즈니스 프로세스 설계 툴, eAI 플랫폼, 비즈니스 프로세스 변환 모듈, UDDI 연결 모듈, 그리고 워크플로우 시스템을 포함한다. 제안된 프레임워크 환경에서 기업 내·외부 응용 프로그램들은 방화벽에 구애되지 않고 손쉽게 통합될 수 있다. 본 논문에서는 제안된 시스템의 구현을 위한 워크플로우 시스템의 확장에 대해서 비교적 자세하게 기술하였으며, 구현된 eAI 프레임워크를 사용한 응용 프로그램 구현을 통하여 제안된 프레임워크의 유용성을 확인하였다. 완전한 기능을 갖춘 eAI 솔루션은 이 프레임워크에 추가적인 기능을 점진적으로 추가함으로써 구현 가능하다.

**키워드** : eAI 프레임워크, 웹서비스, BPMT, eAI 플랫폼, 워크플로우 시스템

**Abstract** This paper proposes a compact eAI framework for the integration of various types of applications deployed on different platforms in the Internet. The applications are connected and invoked to achieve a business goal by the coordination of the workflow system in the framework. For the construction of the framework, five sub-modules are elicited and the functions and roles of each module are defined. The elicited five sub-modules include business process modeling tool, eAI platform, business processes transform module, UDDI connection module, and workflow system. In the framework, intra and inter organizational applications can be integrated together across firewalls. In this paper, the extension of a workflow system to implement the framework is also described in detail and the usefulness of the framework is ascertained by implementing an application process within the framework. A full-fledged eAI solution can be constructed by gradually adding supplementary functions within this framework.

**Key words** : eAI framework, Web Services, BPMT, eAI platform, workflow system

### 1. 서론

인터넷 및 웹 응용 서비스 프로그램의 확산은 기업의 비즈니스 환경을 급격하게 변화시키고 있다. 기업들은 다양한 e-비즈니스의 창출을 통해서 보다 나은 고객 서비스의 제공과 기업의 생산성 향상을 시도하고 있다. 이

러한 요구사항에 부합하며 제시된 기업 정보화 지원 솔루션 중 하나가 e-비즈니스 응용 프로그램 통합(e-Business Application Integration, 이하 eAI)이다. 기존에도 다양한 형태의 응용프로그램 통합이 추구되었지만 대부분의 경우 기업 내부의 응용 프로그램들을 통합하는 수준이었다[1,2]. eAI는 기존의 EAI(Enterprise Application Integration)와 B2Bi(Business to Business Integration)에 머무르지 않고 기업 외부의 응용 프로그램까지 포함한다. 즉 eAI는 기업 내·외부의 서로 다른 플랫폼 상에 산재되어 있는 복수의 응용 프로그램 및 비즈니스 프로세스를 효율적으로 통합해 기업 정보화의 효과를 극대

<sup>†</sup> 정 회 원 : 한국정보통신대학교 공학부 교수  
sdlee@icu.ac.kr

<sup>\*\*</sup> 총신회원 : 한국정보통신대학교 공학부 교수  
dshan@icu.ac.kr

논문접수 : 2003년 10월 27일

심사완료 : 2005년 8월 12일

화하는 것을 지원하는 기술이라고 정의할 수 있다.

eAI 솔루션을 구성하는 다양한 방식이 존재할 수 있지만 많은 전문가들은 주요 요소로 eAI 플랫폼, 어댑터, 데이터 브로커, 워크플로우 시스템의 4가지 소프트웨어를 들고 있다[1]. eAI 플랫폼은 데이터의 안전한 전달과 시스템으로서의 안정성, 성능 등을 보장하는 기반 소프트웨어 모듈에 해당하고, 어댑터는 해당 소프트웨어와 eAI 플랫폼 사이에 위치해 데이터 중개와 응용 프로그램 연동의 인터페이스를 담당하는 모듈이다. 데이터 브로커는 데이터 포맷 및 수준에 따른 다양한 변환을 자동화하는 모듈이며, 워크플로우 시스템은 미리 정의한 업무 처리에 따라서 응용 프로그램 간 데이터 전달 및 연동을 처리하는 소프트웨어 모듈이다. 이러한 eAI 솔루션의 구성을 감안해 볼 때, eAI 프레임워크는 워크플로우 기술의 자연스러운 연계 기술로 볼 수 있다[4].

최근까지 응용 프로그램 통합은 개별 기업 수준에 맞게 포인트 투 포인트 방식으로 해결되어 왔다. 하지만 시스템 수가 증가함에 따라 이런 방식의 통합은 유지보수가 까다롭고 복잡하다는 한계점을 노출하고 있다. 이러한 한계는 중앙의 응용 프로그램 허브를 통해서 시스템을 통합하는 허브-허브-스포크(Hub-and-Spoke) 방식[5,11]의 통합을 통해서 극복할 수 있다. 모든 통합 메시지가 허브를 통해 스포크로 분배하는 구조는 각 참여 시스템들에게 중앙 허브에 대한 어댑터만을 요구하기 때문에, 기존 시스템의 변경을 최소화하면서 전체 응용 프로그램을 통합이 가능하게 하며, 새롭게 추가되는 시스템과의 통합에도 유연한 인프라를 제공하는 것이다.

본 논문에서는 XML(eXtensible Markup Language) 및 웹 서비스에 기반한 메시지 서비스 방식의 응용 프로그램 통합 프레임워크를 제안한다. 특히, 허브-허브-스포크 방식의 응용 프로그램 통합 방식을 확대하여 통합 구조를 웹 서비스와의 접목을 통해 극대화 한다. 즉 어댑터와 데이터 교환에 있어서 표준을 도입함으로써 통합 개방형 프레임워크를 추구하는 것이다. 웹 서비스는 개방형 표준 기술인 XML, UDDI(Universal Description, Discovery and Integration), 그리고 WSDL(Web Services Description Language) 등을 기반으로 하여 서비스 공급자와 서비스 브로커 그리고 서비스 수요자를 유기적으로 결합시켜 주는데, 허브-스포크 방식의 통합에서는 허브를 비롯하여 스포크도 경우에 따라서 서비스 공급자 역할과 서비스 수요자 역할을 병행하며 상호 연동하게 된다.

본 논문에서는 eAI 프레임워크의 구성요소로 웹 서비스에 기반한 eAI 플랫폼, eAI 프로세스 정의도구(BPMT), 비즈니스 프로세스 상호변환 모듈, 표준 등록·저장소 상호연계 모듈, 프로세스 관리용 워크플로우 시

스템을 도출하고 이들 각각의 기능 및 역할을 정의하였다. 그리고 eAI 프레임워크 구성에 핵심적 요소인 워크플로우 시스템의 연결 방식을 고안하였다.

설계된 프레임워크 상에서 웹 서비스를 포함하는 다수의 응용 프로그램 들은 특정 비즈니스 프로세스를 구성하기 위하여 다양한 프로토콜로 데이터를 상호 송수신하면서 통합된다. 또한 설계된 프레임워크는 시스템의 이식성과 유연성을 갖추어서 임의의 메시징 시스템 및 워크플로우 시스템과도 손쉽게 연결될 수 있는 수단을 제공하고 새로운 웹 서비스 표준이나 기능 확장에 손쉽게 대응하게 된다. 한편 본 논문에서는 제안한 eAI 프레임워크의 프로토타입을 구현하고, 구현된 eAI 프레임워크의 프로토타입에서 웹 서비스와 연결되는 응용 프로세스 구현을 통해서 제안된 시스템의 유용성도 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서는 공개된 eAI 솔루션에 관하여 분류하고 설명한다. 3장에서는 본 논문에서 제안하는 eAI 프레임워크의 설계 목표 및 방안에 대해서 기술하고, 4장에서는 3장의 목표 및 방안을 기반으로 eAI 프레임워크의 구성을 제안하고, eAI 프레임워크를 구성하는 구성요소 각각에 대하여 기술한다. 5장에서는 제안한 eAI 프레임워크의 유용성을 확인하기 위하여 프로토타입의 구현 및 평가를 하였고, 6장에서 결론을 맺는다.

## 2. 관련 연구

현재 대부분의 eAI 솔루션들은 기존의 미들웨어 솔루션에서 출발하여, 비즈니스 프로세스 통합과 보안 기능 등을 통합하여 개발된 것들이다. 따라서 대부분의 웹 응용 서버나 미들웨어들이 eAI 솔루션으로 소개되고 있는 것이 현실이며, 대부분의 미들웨어 업체들은 자신들의 기존 솔루션을 업그레이드하면서 eAI 시장에 진출하고 있다. 이 때문에 같은 eAI 솔루션들이라 하더라도, 어느 부분의 통합에 초점을 맞추고 있는가에 따라 솔루션의 특징이 다른 것이 일반적이다.

DataMirror(<http://www.datamirror.com>), BMC(<http://www.bmc.com>), oracle(<http://www.oracle.com>), SmartDB(<http://www.smartdb.com>), D2K(<http://www.d2k.com>) 등의 DBMS(Database Management System), Data-Warehousing 벤더들은 데이터 중심의 통합방식을 선택하고 있다. 이들은 데이터 소스의 변환을 통해 통합이 이루어지는 형태로 응용 프로그램 간의 직접적 통합보다는, 통합된 메타데이터를 통해 통합이 이루어지도록 지원하고 있다. 또한 자동적인 변환보다는 데이터의 수동적 변환에 의존하는 통합 방식을 주로 사용하고 있으며 XML 기술을 중심으로 한 데이터 통합 방식을 주로 지원하고 있다.

BEA System(<http://www.bea.com>), IBM(<http://www.ibm.com>), Sun(<http://www.sun.com>) 등은 메시지 중심의 응용 프로그램 통합을 지원하고 있다. 이들이 지원하는 방식은 응용 프로그램 명령어들의 상호 변환을 통해 통합하는 방식으로 응용들은 이들이 제공하는 미들웨어에 의해서 상호 연동되는 방식을 지원하고 있다.

Attachmate(<http://www.attachmate.com>), Broadvision(<http://www.broadvision.com>), CNT(<http://www.cnt.com>) 등은 액세스(access) 중심의 통합 방식을 지원하고 있다. 이들은 외부 시스템의 데이터나 인터페이스를 하나 또는 그 이상 기업의 기존 시스템과 통합하는 방식으로서 e-비즈니스를 위한 기업 상호간 연동에 많이 사용되고 있다. 기업들은 상호간의 응용 프로그램을 연동할 필요가 없이 통합 서버를 통해 포털이나 외부 인터페이스와 연동되고, 이는 전체 응용 프로그램들의 작동에 통합적으로 영향을 미치는 방식이다. 본 논문에서 제안하는 프레임워크는 메시지 중심으로 운영된다는 측면에서는 메시지 중심의 응용 프로그램 통합 방식으로 분류되는데, 접근 중심의 통합 방식도 지원할 수 있다.

통합 구조에 따라 eAI 시스템을 분류해 볼 수도 있다 [11]. 현재 알려진 eAI 솔루션의 아키텍처는 허브-스포크(Hub and Spoke), 분산 네트워크(Distributed Network), 정보 버스(Information Bus), 그리고 포인트-포인트(Point to Point) 방식 등이 있다.

허브-스포크 방식은 중앙의 응용 허브를 통해 시스템을 통합한다. 모든 통합 메시지는 허브를 거쳐 스포크로 분배된다. 분산 네트워크 방식은 각 응용 프로그램의 미들웨어 간 통신을 위한 것으로 소스 시스템에서 대상 시스템으로 메시지를 전송하고 대상 시스템에서 독립적으로 메시지를 처리한다. 즉 허브-스포크 방식에서 스포크의 기능이 크게 강화되는 것이다. 정보 버스 방식은 메시지 버스를 통해 메시지를 송수신하는 것으로 소스 시스템에서 메시지를 공개하고 타깃 시스템은 이를 획득한다. 포인트-포인트 방식은 통합 대상 시스템간에 직접 일대일로 인터페이스를 통해서 연결한다.

웹 서비스에 기반한 eAI 시스템에 관한 연구 개발은 다양한 형태로 시도되고 있지만 본 논문에서는 쉽게 이해되는 프레임워크에 기반하여 체계적으로 접근한다는 점에서 기존의 방식과 구분된다. 또한 eAI 프레임워크를 구성하는 워크플로우 시스템의 역할 및 기능을 명확하게 도출하고 기술한 것도 워크플로우 시스템의 기능과 역할을 다소 모호하게 기술하는 기존의 시스템들과 구분된다. 마지막으로 본 논문에서 제안하는 eAI 프레임워크는 완성된 형태가 아닌 프로토타입 형태로도 웹 서비스와 연동되는 응용 프로그램 개발이 가능함을 보여줌으로써 점차적으로 오랜 기간에 걸쳐 시스템을 완

성하는 토대를 구축해 준다는 점에서도 기존의 eAI 구성 방식과 구분된다.

### 3. 설계 목표 및 방안

본 장에서는 웹 서비스(<http://www.w3.org/2002/ws/>)에 기반한 eAI 프레임워크의 설계 및 개발을 하는데 있어서 요구되는 사항들과 그 접근 방법에 관해서 기술한다. 웹 서비스에 기반한 eAI 프레임워크의 기본적인 요구 사항은 웹 서비스 환경에서 기업 내·외부 응용 프로그램 및 시스템을 이음새 없이 통합할 수 있는 개발 및 실행 환경을 제공하는 것으로 사용자는 제공된 프레임워크 안에서 기업 내·외부 응용 프로그램을 별다른 구분 없이 연결하여 새로운 비즈니스 프로세스를 손쉽게 설계, 구현하는 것이다. 이를 위해서 프레임워크는 웹 서비스 환경에서 다양한 기간 시스템 및 응용 프로그램의 연동 및 통합이 필요한 비즈니스 프로세스를 설계하고 실행할 수 있는 방법 및 도구를 지원함과 더불어 통합을 지원하는 핵심적인 모듈을 제공하게 된다. 이 밖에도 SOAP/HTTP, JAVA/JMS 등의 다양한 전송 프로토콜 지원, 표준화의 진전에 따른 모듈의 확장성 확보, 워크플로우 표준 API 준수, 그리고 기존의 개발 및 운영 중인 워크플로우 관리 시스템의 구성 요소에 미치는 영향 최소화 등을 목표로 하여 시스템을 설계하였다. 세부 목표 및 방안 각각에 대하여 보다 상세한 설명은 다음과 같다.

- 웹 서비스 표준 지원(Supporting Web Services Standard)
  - 본 논문에서 제시되는 eAI 프레임워크는 웹 서비스 환경에 기반하여 시스템을 구성함으로써 기업 외부 서비스 시스템 및 응용 프로그램과의 효과적인 통합 수단을 제공하는 목표를 가지고 있다. 이를 위해서 제안된 프레임워크는 웹 서비스 환경에서 제공되는 개발 환경을 이용함과 동시에 웹 서비스 표준 사항을 준수하는 연동 모듈들을 장착시킨다. 구체적으로 프레임워크 안에서 개발되는 모듈은 XML, SOAP, WSDL, UDDI 등의 웹 서비스 표준을 완벽하게 지원할 수 있도록 구현한다.
- 다양한 전송 프로토콜 지원(Supporting Various Transmission Protocol)
  - 인터넷 상에 산재한 응용 시스템 및 기간 업무 시스템은 많은 경우 다양한 프로그래밍 언어와 운영환경에서 작동하고 있으며, 각 시스템간 전송 프로토콜 또한 다양한 방식을 취하고 있다. 따라서 본 프레임워크에서는 HTTP, CORBA, EJB, .Net 등 다양한 전송 프로토콜을 사용하는 응용 프로그램들의 연동을 지원하기 위해 웹 서비스에 기반한 메시징 서비스 연동 방식을 채택한다. 이러한 환경 구현을 위해 웹 서비스 게이트웨이[6] 내에 프로토콜 교환 모듈을 구현하고, 이를 이용해서 다양한 전송

프로토콜을 지원하도록 한다.

· 이식성(Portability) - eAI 프레임워크는 자주 워크플로우 시스템 및 메시징 서비스 시스템과 긴밀하게 연결되어 구성된다. 이 과정에서 자칫 개발된 eAI 프레임워크가 특정 워크플로우 시스템 또는 메시징 서비스 시스템에 의존적일 수 있다. 본 논문에서 추구하는 이식성은 프레임워크가 특정 시스템에 의존적이지 않고 다양한 시스템에 플러그인 될 수 있음을 의미한다. 특히 이식성은 eAI 프레임워크가 기존에 설치 운영 중인 워크플로우 시스템 등의 수용을 위해서는 반드시 필요한 중요한 요소이기도 하다. 이식성은 개념적으로 워크플로우 시스템 및 메시징 서비스 시스템도 프레임워크 상에서 통합되는 하나의 응용 시스템으로 간주함으로써 실현 가능하다. 구체적으로는 워크플로우 시스템 및 메시징 서비스 시스템에 종속적인 부분을 별도의 요소로 분리하여 구성하고 표준 API와 특정 시스템의 고유한 기능을 지원하는 부분을 구분하여 구성함으로써 가능하다.

유연성(Flexibility) - 현재 웹 서비스 표준은 다양한 개념을 적용해 가면서 계속해서 표준안이 변경되거나 기능이 추가되고 있다. 프레임워크의 유연성이란 이러한 표준의 변화 및 기능 추가에 쉽게 대처 가능한 구조를 의미한다. 이를 위해서 표준과 관련된 데이터를 처리하는 요소를 별도로 구성하고 관련 서비스를 이곳에 집중시킴으로써 새로운 표준이나 기능 추가에 용이하게 대처한다.

이 같은 설계 목표 달성을 통해서 참고문헌에서 주장한 응용 프로그램 통합에 요청되는 요구 사항[1,3,12]을 작간접적으로 만족시킬 수 있다.

#### 4. eAI 프레임워크

3장에서 기술한 바와 같이 본 논문에서 제안하는 eAI 프레임워크는 웹 서비스의 구조를 반영하여 구성된다. 구성된 프레임워크 상에서 eAI 서비스 제공자는 eAI 프로세스 정의 도구(e-Business Process Modeling Tool, 이하 e-BPMT)를 이용하여 서비스를 정의하고 적절한 포맷으로 변환한 후 표준 등록·저장소(UDDI Repository and ebXML Reg/Repository)에 등록한다. 표준 등록·저장소에 새롭게 정의된 서비스의 등록을 지원하기 위해서 프레임워크는 표준 등록·저장소 상호연계 모듈(UDDI/ebXML Repository Interaction Module)을 포함한다. eAI 서비스 사용자들 또한 표준 등록·저장소 상호연계 모듈을 통하여 표준 등록·저장소를 검색하여 원하는 서비스를 찾아내며, 사용자 응용 프로그램을 통해 웹 서비스에 연결하여 해당 서비스를 호출하게 된다. 호출된 서비스는 워크플로우 시스템에 의해 수행되는데 수행 과정에서 다양한 응용 프로그램의 통합이 요청되면 비즈니스 프로세스 상호변환 모듈(Business Process Transform Module)을 이용하여 서로 다른 데이터 포맷의 한계를 극복하게 된다. 이러한 일련의 과정은 많은 경우 eAI 플랫폼을 허브로 이용하여 처리된다. 그림 1은 이와 같은 과정을 지원하기 위하여 본 논문에서 제안하는 eAI 프레임워크의 구성을 보여주고 있다.

eAI 프레임워크는 워크플로우 시스템을 포함하여 공공 영역(Public Area)과 클라이언트 영역(Client Area) 들에 공통적으로 제공할 수 있는 모듈들로 구성한다. 이 모듈 들을 그 기능 및 역할 관점에서 분류해 보면 다음과 같은 다섯 개의 하위 모듈로 세분해 볼 수 있다.

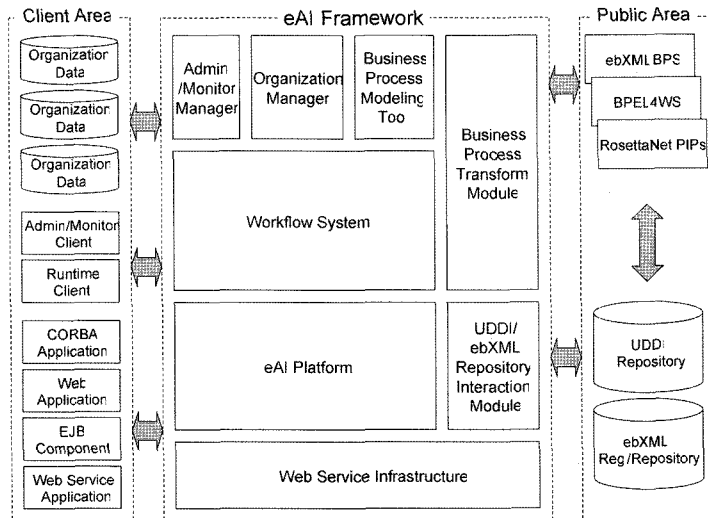


그림 1 eAI 프레임워크 구성도

- ▶ 웹 서비스에 기반한 eAI 플랫폼(eAI Platform)
- ▶ 비즈니스 프로세스 정의 도구(e-BPMT)
- ▶ 비즈니스 프로세스 상호변환 모듈(Business Process Transform Module)
- ▶ 표준 등록 저장소 상호연계 모듈(UDDI/ebXML Repository Interaction Module)
- ▶ 프로세스 관리용 워크플로우 시스템(Workflow System) 프레임워크 구성에서 중요한 부분 중 하나인 조직도 관리기 (Organization Manager) 및 시스템에 대한 관리/모니터 관리(Admin/Monitor Manager) 모듈 등은 워크플로우 시스템에 포함되는 것으로 간주하여 구체적인 기술을 하지 않았다. 다음은 제안한 eAI 프레임워크의 구성을 웹 서비스에 맞추도록 구체적인 세부 모듈을 설계하고 그 내용을 기술한다.

**4.1 웹 서비스에 기반한 eAI 플랫폼**

eAI 플랫폼은 e-BPMT를 통하여 정의된 비즈니스 프로세스가 웹 서비스 환경에서 워크플로우 시스템을 통한 구동 및 운영을 지원하는 소프트웨어 환경이다. 그림 2는 eAI 플랫폼의 구성도 및 플랫폼과 연계된 모듈들을 보여주고 있다.

그림에서 각 세부 구성 모듈에 관한 설명은 다음과 같다.

- **웹 서비스 게이트웨이(Web Service Gateway):** 웹 서비스는 기업에 있어서 효율적인 프로세스 통합을 위한 기법으로 인식되고 있다. 웹 서비스를 구축하고자 할 경우 서비스의 범위를 확장시킬 필요가 있으며 이러한 목적을 위한 필요한 구성요소가 웹 서비스 게이트웨이이다. 웹 서비스 게이트웨이를 통해서 타 시

스템 및 외부 시스템에게 서비스를 제공할 수 있다. 또한 단순한 서비스 제공뿐만 아니라 외부 시스템과의 게이트웨이 역할을 하기 때문에 내부적으로 여러 가지 응용 서비스를 제공할 수 있는 기반 역할이 가능하다.

- **메시지 서비스 핸들러(Message Service Handler: MSH):** 메시지 서비스 핸들러는 웹 서비스를 기반으로 협업하고자 하는 비즈니스 프로세스에 따라 문서 및 메시지를 전달하고자 할 때 워크플로우 시스템 및 기간 시스템과의 연동을 위해 메시지 전송을 위한 서비스를 제공한다. 이 구성요소의 핵심 기능은 XML 메시지의 송수신에 있다. 실질적인 송수신 메커니즘은 웹 서비스 표준 사양에 나와있기 때문에 SOAP/WSDL 전송 프로토콜을 이용하고, 웹 서비스 지원 어댑터를 이용하여 연동하고자 하는 파트너마다 다른 프로토콜을 사용하더라도 쉽게 적용할 수 있도록 설계하였다. 송신 부분은 목적지와 전송할 내용(메시지)를 받아서 이미 약속된 프로토콜을 이용해서 메시지를 전송하는 것으로 기능을 마치게 되며, 수신 부분은 연동에 사용되는 전송 프로토콜에 대해서 메시지의 전달을 받아 들일 수 있어야 하기 때문에 적절한 포트를 열고 수신 모드로 항상 대기하고 있어야 한다.
- **표준 등록 저장소 상호연계 모듈(UDDI/ebXML Repository Interaction Module):** ebXML 등록·저장소(ebXML Reg/Repository) 및 UDDI 저장소(UDDI Repository) 연동의 기능을 연동 모듈과 같이 적용하기 위해서 XML 저장소(JAXR)는 Java API 사양을 사용한다. JAXR은 저장소에 접근 편의를 위한 API로서, 등록

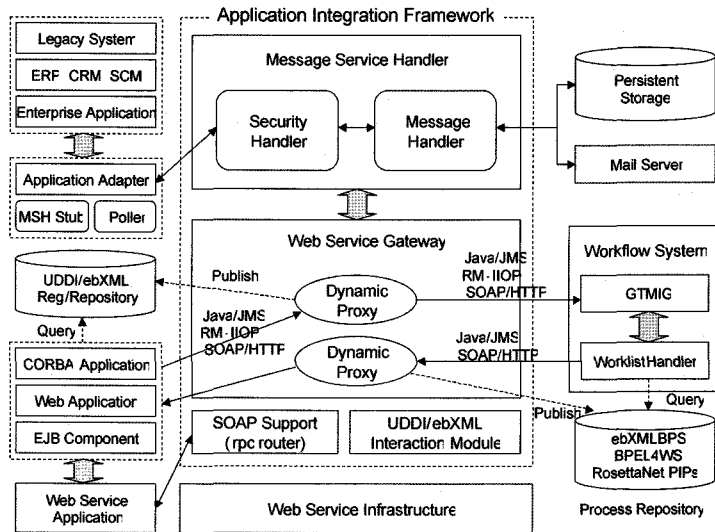


그림 2 eAI 플랫폼 구성도

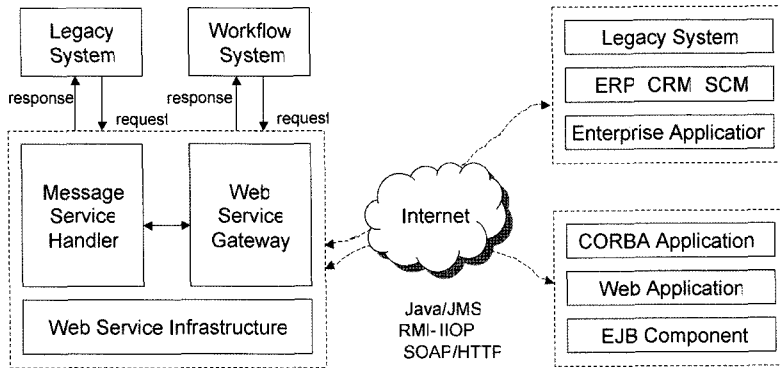


그림 3 기간 시스템 연동 개념도

(publishing), 검색(querying), 편집(editing) 기능들을 지원한다. JAXR은 특히 XML 웹 서비스 중에서 B2B 응용에 특화되었고, 복잡한 콘텐츠 조회와 publish/subscribe XML 메시지를 지원한다. 또한 다른 타입의 저장소 접근도 지원을 하며 ebXML 저장소 접근 API 역시 지원한다. 이후 UDDI ver.3 지원 및 ebXML 등록/저장소 ver.3 까지 지원할 수 있도록 구현한다.

- **변환기(Router):** 요청된 메시지를 웹 서비스 형태로 변환하거나 또는 웹 서비스를 메시지 형태로 변화해주는 역할을 지원함으로써 유기적이고 효율적인 웹 서비스 변환이 이루어진다.
- **웹 서비스 인프라스트럭처(Web Service Infrastructure):** 웹 서비스 인프라스트럭처는 웹 서비스를 지원하는 환경으로서 JDK, 웹 서버, JSP/Servlet 엔진 등을 포함하는 일련의 개발 및 웹 서비스 지원환경을 의미한다. 본 논문에서는 채택한 웹 서비스 인프라스트럭처는 JDK 1.4.2 와 Tomcat 4.1.8 이다. 이런 개발 환경 이외에도 웹 서버와 JSP 엔진을 포함하는 웹 서비스 개발환경으로 상용화된 제품으로는 IBM WebSphere 서버, BEA WebLogic 서버, Sun One 서버 등이 있다. 한편, 기간 시스템(Legacy System) 간 서비스 통합이 이루어지는 기본적인 시나리오를 살펴보면 다음과 같다. 먼저 워크플로우 시스템을 포함한 기간 업무 시스템에서 MSH에게 특정 메시지 송·수신 서비스를 요청한다. MSH는 메시지 송·수신을 요청한 대상에게 서비스 하기 위해 header, body, payload로 구성된 메시지를 SOAP/WSDL 메시지로 포장하여 요청된 메시지를 전달한다. 서비스를 요청 또는 전달하는 대상은 웹 서비스에 기반한 메세징 서비스를 받기 위해 인터페이스를 필요로 하며 이를 통해 MSH와 연동할 수 있다. 메시지를 웹 서비스 형태로 전환하기 위해서는 앞에서 기술한 웹 서비스를 기반으로 하는 웹 서비스 게이트웨이를 활용한다. 그림 3은 이러한 일련의 과정에 대한 연동 개념

도를 보여주고 있다.

그림 3에서 기업 내의 기간 업무 시스템 간 데이터 송·수신은 MSH 또는 웹 서비스 게이트웨이를 통해서 이루어지며 기업 밖의 응용 시스템은 주소 웹 서비스 게이트웨이를 통해서 연결되는 구조이다. 이러한 통합은 MSH 및 웹 서비스 게이트웨이에 응용 프로그램 형태로 연결된 로컬 워크플로우 시스템을 중심으로 내부 기간 시스템 및 외부의 응용 프로그램을 연결하는 형태로 구현된다. 로컬 워크플로우 시스템과의 연결은 SOAP/WSDL로 구성된 메시지로 이루어지며 이러한 메시지 서비스를 제공 받기 위해서는 WSDL 표준 인터페이스를 구현한 메시지 스템(Stub) 인터페이스를 이용한 연결 요청이 필요하다. 외부 시스템과의 연동에는 각 시스템에서 요구하는 전송 프로토콜을 지원하는 웹 서비스 게이트웨이를 통해 서비스를 이용하는 통로를 제공한다.

그림 4는 웹 서비스 게이트웨이, MSH가 포함된 eAI 플랫폼 상에서 인터넷 상의 흩어진 응용 프로그램, 기간 시스템, 그리고 워크플로우 시스템 들의 연결 관계를 보여주고 있다.

그림 4에서와 같이 웹 서비스 게이트웨이, 그리고 MSH는 웹 서버 위에서 하나의 웹 서비스 서버로 작동하면서 메세징 서비스, 그리고 웹 서비스 게이트웨이를 이용한 다양한 프로토콜을 통한 서비스 요청을 지원한다. 워크플로우 시스템을 웹 서비스 게이트웨이에 적용하기 위해서는 워크플로우 서비스를 웹 서비스 게이트웨이에 웹 서비스화 해서 장착시키는 과정이 필요하다. 그리고 장착된 서비스는 메시지 서비스 핸들러를 통해서 어플리케이션과의 연동 서비스가 이루어 지거나 바로 외부 시스템에 공개되어서 서비스를 지원하게 된다. 서비스의 공개는 표준등록 저장소에 의해서 공개되며 외부 시스템에서 검색하는 과정을 통해 이루어 진다.

#### 4.2 비즈니스 프로세스 정의도구(e-BPMT)

eAI를 지원하는 비즈니스 프로세스 정의도구(eBPMT)

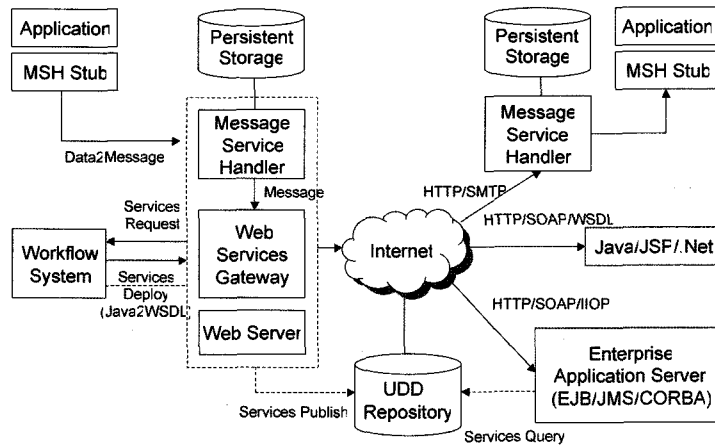


그림 4 시스템 연동 모듈을 통한 데이터 전송 개념도

는 eAI 프레임워크 구성에 반드시 필요한 중요한 요소 중 하나이다. 비즈니스 프로세스 설계자 및 개발자는 이것을 사용하여 웹 서비스를 포함하는 기업 내·외부의 다양한 응용 프로그램을 연결하여 원하는 새로운 비즈니스 프로세스를 설계, 구현 그리고 테스트하게 된다. 따라서 e-BPMT는 워크플로우 시스템과 연결된 기존의 BPMT와 유사하게 프로세스 설계를 위한 캔버스 및 드로잉 환경을 제공함과 더불어 응용 프로그램 통합 기능 및 프로그램 개발 작업대로 활용할 수 있는 보강된 기능을 가지게 된다. 비즈니스 프로세스 설계 기능에 덧붙여 e-BPMT가 가지게 되는 추가적인 기능을 열거하면 다음과 같다.

- UMM 방법론(UN/CEFACT Modeling Methodology)에 따른 프로세스 설계 지원
- 예외상황 명세 및 처리
- 프로세스 시뮬레이션 및 검증

다음 절에서는 이들 기능 각각에 대하여 좀 더 자세히 기술하기로 한다.

4.2.1 UMM 방법론에 따른 프로세스 설계 지원

UMM 방법론은 UN/CEFACT(<http://www.unece.org/cefact/>)가 제시한 e-Biz 구축 방법론으로서 UMM 방법론을 통해 정의되는 비즈니스 프로세스는 객체지향적 언어이면서 시스템에 적용이 빠르고 용이한 UML 형태로 표현된다. UML 형태로 표현된 비즈니스 프로세스는 XML 형태로의 상호 변환을 통해서 손쉽게 표준등록 저장소에 등록되고 참조될 수 있다. 따라서 비즈니스 프로세스 설계자가 UMM 방법론에 기반하여 비즈니스 프로세스를 설계하는 경우에는 프로세스 설계 과정에서 생성되는 산출물을 체계적으로 생성하고 관리하는 것이 용이하다. 또한 표준화된 모델링 방법론의 채택으로 이 방법론에 따라 얻어진 기존의 연구 성과들을 손쉽게 이

용할 수 있는 장점이 있다.

e-BPMT는 프로세스 설계자가 UMM 방법론에 의하여 체계적인 절차에 따라 모델링 작업을 진행할 수 있도록 지원하며 설계 과정에서 얻어지는 산출물을 생성, 저장, 관리할 수 있도록 지원한다. 또한 설계된 프로세스에 대하여 과학적으로 분석하고 분석 결과를 반영할 수 있도록 한다. 즉, e-BPMT는 UMM 방법론에 기반한 비즈니스 프로세스 모델링을 지원하기 위해 프로세스 모델을 구축함에 있어서 분석, 설계 부분을 UMM 기반의 분석과 모델링을 통하여 자동화하고 표준화된 양식의 산출물을 생성한다. 모든 프로세스 정보는 XML 기반의 비즈니스 프로세스 명세서 형태로 저장되고 관리된다. 정의된 비즈니스 프로세스는 4.3절에서 설명하는 프로세스 상호 변환 모듈을 통해서 ebXML BPS, RosetaNet PIP, BPEL4WS 등의 다양한 형태로 변환되어 활용된다.

UMM 방법론에 따른 모델링 과정은 비즈니스 프로세스 설계 단계와 구현 단계로 구성된다. 비즈니스 프로세스 설계 단계는 다시 3가지 단계로 구성되는데 개념적 디자인, 논리적 디자인, 물리적 디자인 등이 그것이다. 구현 단계에서는 디자인 된 비즈니스 프로세스를 구성 요소로 변환하는 작업이 진행된다. e-BPMT는 이 과정에서 유스케이스 다이어그램, 클래스 다이어그램, 시퀀스 다이어그램 등을 생성하고, 설계된 프로세스를 수행할 워크플로우 엔진과의 프로토폴에 해당된다고 볼 수 있는 BOM 구성(BOM Structure)과 프로세스 다이어그램도 생성하게 된다. 그림 5는 e-BPMT 및 관련 산출물들의 모습을 보여주고 있다.

4.2.2 예외상황 명세 및 처리

인터넷 환경에서 웹 서비스를 포함하는 기업 내·외부의 다양한 응용 프로그램을 연결하여 생성된 비즈니스

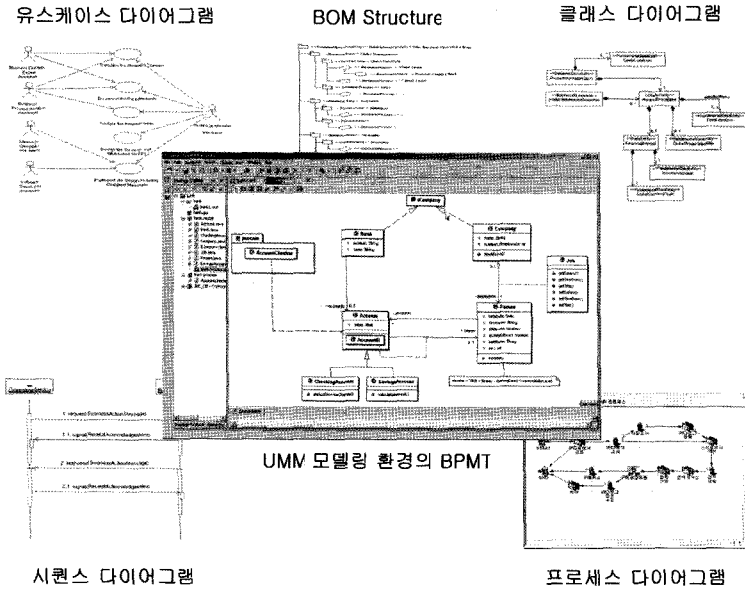


그림 5 e-BPMT 및 관련 산출물

프로세스가 실제로 실행되는 경우에는 eAI 기반 시스템이 예기치 못하고 대처할 수 없는 많은 예외 상황에 직면할 수 있다. 이러한 예외 상황은 자칫 해당 비즈니스 프로세스를 운영하는 기업에 심각한 손실을 가져올 수 있고, 그 역할이 중요한 비즈니스 프로세스의 경우에는 더 많은 주의가 요청된다. 비즈니스 프로세스 설계자는 비즈니스 프로세스 설계 시 이와 같이 기반 시스템에 의해서 처리되지 못하는 예외적 상황을 염두에 두고 그러한 예외적 상황에 적절하게 대처하는 안전한 비즈니스 프로세스를 설계하여야 한다. 따라서 e-BPMT가 안전한 비즈니스 프로세스 설계를 지원하기 위해서는 예외 상황 명시 기능 및 해당 처리 루틴의 연결을 명시할 수 있는 기능 등을 갖추어야 한다[7,8]. 명시된 예외 상황은 4.3절에서 기술하는 워크플로우 시스템에 전달되고 워크플로우 시스템은 프로세스 실행 중 명시된 예외 상황의 발생을 감시하고 발생된 예외 상황에 대해서는 해당 처리 루틴을 호출하여 수행한 뒤, 다시 비즈니스 프로세스 설계자가 명시한 바에 의해서 비즈니스 프로세스의 적절한 곳으로 제어를 넘겨주게 된다.

e-BPMT 상에서 비즈니스 프로세스 설계자는 프로세스 설계 시 단위업무 또는 블록 단위로 예외상황 발생 감시 영역을 설정하고 전체 조건(pre-condition)과 후속조건(post-condition)을 통해서 예외 상황을 명시한 뒤 e-BPMT가 지원하는 예외처리 모델 중 하나를 선택하게 된다. 그림 6은 e-BPMT가 지원하는 각각의 예외처리 모델의 차이를 제어흐름 관점에서 보여주고 있다.

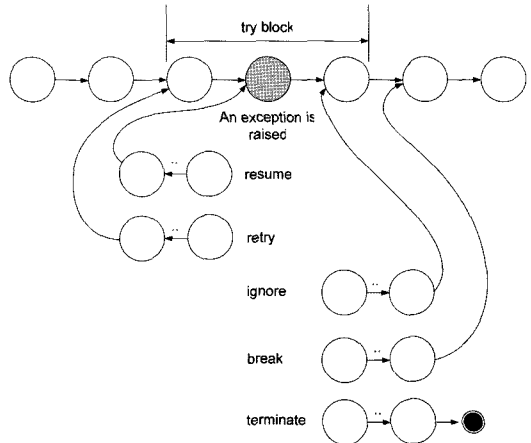


그림 6 예외처리 모델

- ignore 모델: 예외 처리 후 제어 흐름이 다음 단위업무 무로 이동
- retry 모델: 예외 처리 후 제어 흐름이 예외 상황발생 감시 영역의 최초의 단위업무로 이동
- resume 모델: 예외 처리 후 제어 흐름이 예외 상황발생 시킨 단위업무로 이동
- break 모델: 예외 처리 후 제어 흐름이 예외 상황발생 감시 영역을 벗어난 지점으로 이동
- terminate 모델: 예외 처리 후 예외상황을 발생시킨 프로세스 단계 종료



4.2.3 프로세스 시뮬레이션 및 검증

e-BPMT 상에서 설계되는 비즈니스 프로세스는 자주 많은 기업 내·외부의 응용 프로그램의 통합을 도모하면서 구성되는 만큼 복잡하고, 그 규모도 큰 것이 보통이다. 이와 같이 복잡하고 대규모 비즈니스 프로세스를 안전하게 개발하기 위해서는 비즈니스 프로세스 설계자 및 개발자는 설계 과정에서부터 설계된 비즈니스 프로세스가 올바른지 비즈니스 프로세스 실행 전에 프로세스를 검증하는 절차가 반드시 필요하다[9,18].

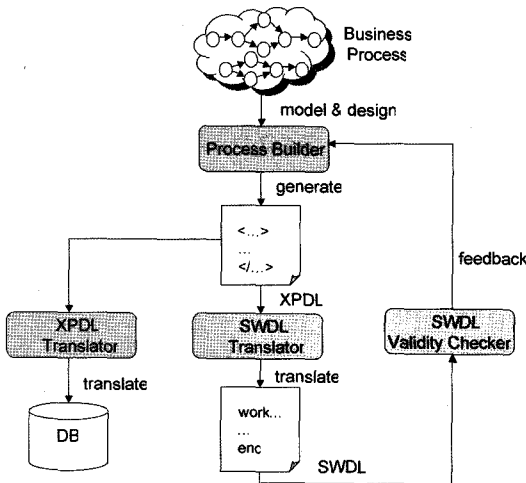


그림 7 프로세스 오류검사 및 피드백 과정

e-BPMT는 비즈니스 프로세스 시뮬레이션 기능과 비즈니스 프로세스 용 검증자(Correctness Checker)를 갖추고 이러한 검증 절차를 지원하고 있다. 비즈니스 프로세스 시뮬레이션 기능을 통해서 설계자는 단위 업무 별로 임의의 입력 데이터와 출력 데이터를 생성하면서 비즈니스 프로세스의 전체 또는 부분의 동작 상황을 점검할 수 있게 된다. 이 과정에서 설계자는 예외상황을 발생시켜 발생한 예외상황이 올바르게 처리되고 있는지도 점검하기도 한다.

검증자는 명시된 비즈니스 프로세스에 내재된 오류를 찾아 주는 기능으로 현재 e-BPMT 는 생성된 비즈니스 프로세스를 SWDL형태로 변환한 후 접근 충돌(Access Conflict) 분석을 수행하고 발견하지 못한 예외처리를 찾아서 설계자에서 알려주는 기능을 가지고 있다. 이와 같은 기능 지원을 통해서 e-BPMT 는 설계자가 올바른 비즈니스 프로세스를 설계하는 것을 도와준다. 그림 7은 설계된 프로세스가 SWDL로 변환되고 분석되어 그 결과가 다시 설계자에게 전달되는 일련의 과정을 보여주고 있다.

4.3 비즈니스 프로세스 상호변환 모듈

비즈니스 프로세스가 웹 환경에서 적절히 활용되기 위해서는 표준을 포함한 다양한 형태로의 변환이 필요하다. 비즈니스 프로세스 상호변환 모듈은 UMM 방법론에 의거하여 모델링 된 비즈니스 프로세스 모델을 ebXML 비즈니스 프로세스 명세서(http://www.ebxml.org)로 변환하는 작업을 기본적으로 수행하며, 이후 WSDL 및 BPEL4WS[3] 그리고 RosettaNet PIPs 등과의 상호 변환을 실시한다(http://www.rosettanet.org). 4.2절에서 소개된 e-BPMT를 사용하여 생성된 비즈니스 프로세스가 널리 쓰이고 다른 표준으로 일반 공표된 프로세스를 eAI 프레임워크 상에서 수용하기 위해서는 이 모듈은 반드시 필요하다. 그림 8은 본 프레임워크가 제공하는 비즈니스 프로세스 표준 간 상호변환 서비스 및 절차를 보여주고 있다.

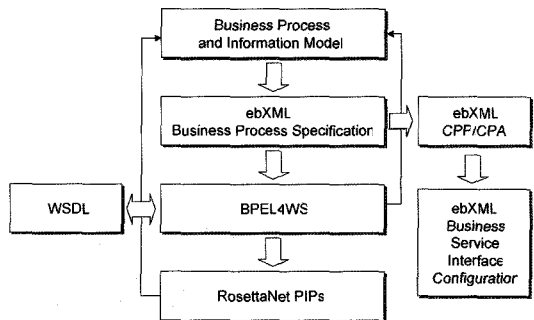


그림 8 비즈니스 프로세스 상호변환 및 절차

- UMM방법론에 의해 e-BPMT상에서 모델링 된 비즈니스 프로세스는 일단 그림에서와 같이 UMM 비즈니스 프로세스와 정보 모델로 정의된다. UML기반 언어로 정의되는 비즈니스 프로세스와 정보는 추출 및 변환을 거쳐 XML 베이스의 ebXML 비즈니스 프로세스 명세서와 문서 명세서로 변환된다. 이에 반하여 ebXML 비즈니스 프로세스 명세서로 정의된 업무 프로세스를 UMM 모델로 가져와서 활용하는 것도 가능하다.
- BPEL4WS는 비즈니스 프로세스의 상호관계를 기술하는 XML기반 업무 흐름 언어로서 BPEL4WS를 사용하면 기업 내부 프로세스 간의 상호 관계 이외에도 기업과 기업 프로세스 간의 상호관계도 기술 가능하다[13]. 이러한 비즈니스 프로세스는 협력 업체가 제공한 웹 서비스를 이용하고, 자신의 서비스를 웹 서비스 비즈니스 프로세스로서 사내의 다른 조직이나 협력업체에 신뢰할 수 있는 방식으로 제공한다. 따라서 BPEL4WS는 서로 다른 기술을 이용하는 사내 다른 조직이나 다른 회사와의 비즈니스 프로세스를 상호

운용하는 데 있어서 필요한 핵심적인 정보를 제공하는 수단이다. IBM에서 개발한 BPWS4J는 BPEL4WS 명세를 구현한 것으로 BPEL4WS 프로세스를 만드는 데 필요한 실행 엔진과 에디터를 포함하고 있다. 따라서 BPEL4WS 형태의 비즈니스 프로세스는 IBM BPWS4J 툴을 사용하여 손쉽게 정의할 수 있지만 e-BPMT상에서 정의된 비즈니스 프로세스를 BPEL4WS 형태로 바꾸기 위해서는 별도의 변환 모듈이 필요하다. 이 변환은 4.2절에서 변환하여 얻어진 ebXML 형태의 비즈니스 프로세스를 BPEL4WS 형태로 바꾸는 것으로 대체한다. 그림 9는 e-BPMT에서 작성된 비즈니스 프로세스 모델이 BPEL4WS 표현으로 변환된 모습을 보여주고 있다.

RosettaNet은 폭 넓은 산업과 개방적인 전자 상거래 표준을 만들고 구현하기 위해 모든 공급 체인 거래 상대방들과 구매자들의 이익을 제공하기 위한 공급 체인 거래 상대방들 간에 전자적인 비즈니스 인터페이스를 제공하여 개방적이며 일반적인 프로세서를 정의하는 표준이다. RosettaNet이 제시하는 전자상거래 표준화 작업을 위해서 필요한 필수 구성요소로는, 사전(Dictionaryes), RNIF(RosettaNet Implementation Framework), PIPs(Partner Interface Dictionary), 제품 및 업체 코드로 구분된다. e-BPMT상에서 정의된 비즈니스 프로세스가 RosettaNet에서 사용되기 위해서는 RosettaNet PIP로의 변환이 필요하고 이것은 해당 비즈니스 프로세스로부터 얻어진 BPEL4WS를 RosettaNet PIP 형태로 변환하는 모듈을 통하여 지원하게 된다. 또한 RosettaNet 상에서 정의된 비즈니스

프로세스는 이것을 e-BPMT 상의 비즈니스 프로세스 로 변환하는 모듈을 통하여 변환을 지원한다.

**4.4 표준 등록·저장소 상호연계 모듈**

e-BPMT 상에서 정의되고 ebXML 및 BPEL4WS 형태로 변환된 비즈니스 프로세스가 웹 서비스 환경에서 서비스되기 위해서는 ebXML 등록·저장소 및 UDDI 저장소(http://www.uddi.org)에 저장되어야 한다. ebXML 등록·저장소 및 UDDI 저장소에의 연동 기능은 e-BPMT 및 자바 플랫폼에서 운영하거나 저장소로 접근하기 위해서 XML 등록(JAXR)을 위한 Java API 사양을 사용하는 것도 가능하다. JAXR은 저장소에의 접근 편의를 지원하기 위한 API로서, 등록, 검색, 그리고 편집 기능을 지원한다. JAXR은 특히 XML 웹 서비스 중에서 B2B 응용에 특화되었고, 복잡한 컨텐츠 조회와 publish/subscribe XML 메시지를 지원한다. 또한 다른 타입의 저장소에 대한 접근 지원을 할 수 있으며 ebXML 저장소 접근 API지원도 가능하다. 본 상호연계 모듈의 ebXML 등록·저장소 및 UDDI 저장소에 연동 기능은 이러한 JAXR 사양에 기반하고 있다.

표준 등록·저장소 상호연계 모듈을 구현하기 위해서는 ebXML 저장소 에 등록되어있는 비즈니스 프로세스를 검색하고 접근해서 정보를 가져올 수 있도록 요구 및 응답 그리고 조회 API 구현이 필요하다. 이러한 기능의 구현에도 JAXR API를 사용한다.

표준 등록·저장소 상호연계 모듈 구현에는 웹 서버와 SOAP 서버와 같은 요소가 포함 된다. 그리고 응용 서버 및 EJB 연동을 위한 인터페이스도 포함된다. 현재 ICU/COWS 워크플로우 시스템에서는 DBMS HTTP,

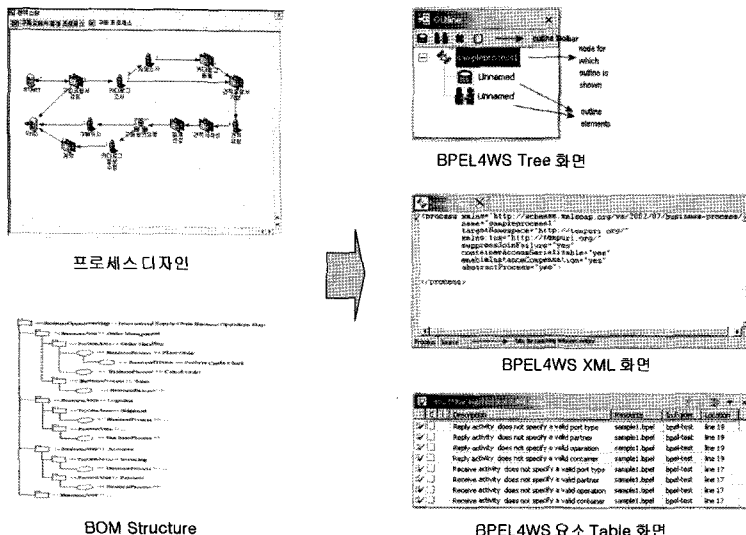


그림 9 BPEL4WS 표현으로 변환된 비즈니스 프로세스 모델

EJB, CORBA 와 관련된 접근 인터페이스가 구현되어 있으며 개발 환경도 제공하고 있다[10]. 따라서 표준 등록·저장소 연계 모듈은 기존에 구현된 인터페이스에 웹 서비스 표준 프로토콜을 지원하는 인터페이스를 추가하는 형태로 구현되었다.

4.5 워크플로우 시스템

그 동안 다양한 워크플로우 시스템이 다양한 목적으로 개발되어 왔다. 하지만 본 논문에서 제안된 eAI 프레임워크는 웹 서비스에 기반하고 있어 프레임워크 구성에 참여하는 워크플로우 시스템은 웹 서비스 방식에 적합하게 수정되어야 한다. 이를 위해서는 워크플로우 시스템이 웹 서비스 형태로 외부에 제공하는 표준적인 인터페이스 정의 및 웹 서비스 게이트웨이의 장치가 필요하다. 또한 기업간 비즈니스 통합을 위해서는 통합 프로세스를 설계할 수 있는 수단이 요구되는데, 이 부분은 BPMT에 해당되는 것으로 앞서 기술된 eAI 프로세스 정의도구에 해당된다. 다음은 종전의 워크플로우 시스템이 eAI 프레임워크에 사용될 때 고려할 사항 및 수정될 내용을 ICU/COWS 워크플로우 시스템[10]을 중심으로 기술한다.

4.5.1 워크플로우 시스템의 웹 서비스 화

기존 워크플로우 시스템의 웹 서비스화는 eAI 프레임워크 구현에 있어서 반드시 필요하다. 왜냐하면 허브 역할을 수행하게 되는 워크플로우 시스템이야말로 연동하고자 하는 기간 시스템들이 연동해야 할 유일한 대상이기 때문이다. 워크플로우 시스템의 웹 서비스 화를 위해서는 기존 워크플로우 시스템에 관한 분석이 필요하다. 그리고 이러한 분석을 기반으로 새롭게 확장할 부분과, 확장하는 과정에서는 가능하면 새롭게 필요한 모듈을 독립적인 모듈이 되도록 구성하는 것이 바람직하다. 독립된 모듈의 부착을 통한 방식은 당연하지만 기존 소스의 변경보다 소스간 독립성을 확보하는데 유리한 장점을 가지고 있다. 본 절에서는 새롭게 도입된 독립 모듈을 사용하여 기존의 워크플로우 시스템을 웹 서비스화 하는 방안을 소개한다.

4.5.1.1 기존 워크플로우 시스템 분석

기존 ICU/COWS 워크플로우 시스템은 CORBA 기반의 분산 객체 기술을 활용하여 개발되었다. 즉, ICU/COWS는 순수 자바로 구현되어 자바 가상 머신이 동작하는 모든 운영 체제에서 동작이 가능하기 때문에 대부분의 플랫폼을 지원한다고 볼 수 있다. 특히 시스템의 구조로 분산 구조를 채택함으로써 중앙집중식 구조에 대해서도 지원이 가능하다. 기존 워크플로우 시스템의 입장에서는 분산구조 또한 단순한 분산구조와 완전한 분산구조로 구별하는데, 완전한 분산 구조의 경우에는 시스템 구성요소가 클라이언트 사이트에 위치할 수 있

는 경우를 의미한다. 서버가 클라이언트에게 더욱 접근함으로써, 신뢰성이나 성능면에서 우월할 수 있다.

서버 구성의 핵심적인 모듈에는, 참여자와의 비동기적인 통신을 전달하는 워크리스트 핸들러(WorklistHandler)와 프로세스 객체나 능동 객체의 생성을 담당하는 TMIF(TMI Factory) 그리고 프로세스 생성 요청을 받아들여서 처리하는 GTMIG(GTMI Generator) 등이 포함된다. 기존 워크플로우 시스템에서는 단위업무 관리자를 TMI(Task Management Instance)라 부르고 프로세스 관리자를 GTMI(Global TMI)라고 부른다. 따라서 GTMI는 다수의 TMI를 거느리고 지정된 프로세스를 수행하게 되는 것이다. 이 과정에서 필요한 경우 워크리스트 핸들러를 통해 참여자에게 과업을 부과하기도 하는데, 이렇게 참여자에게 부과되는 작업에 대해 담당하는 TMI를 AssignableTMI라고 부른다. ApplicationTMI는 응용이 지정되어 있는 단위업무 관리자로 배정된 과업을 응용의 수행을 통해 달성하게 된다. 이 두 가지 단위업무 관리자가 기존의 워크플로우 시스템에서는 외부 기간 시스템과의 연동 포인트가 된다.

그림 10은 기존 워크플로우 시스템의 구성을 다이어그램 형태로 보여주고 있다. 그 중에서 점선으로 분리된 영역이 웹 서비스화의 과정에서 확장이 필요한 부분이다. 워크플로우 시스템 입장에서는 연동 대상이 되는 기간 시스템도 하나의 응용으로 간주된다. 다만 단순한 호출로 충분한 단순 응용과는 달리 상호운영 관계를 형성해야 한다는 점에서 구별된다. 이러한 상호운영을 위해서 WfMC에서는 Interface4라는 표준 안을 제안하고 있는데 Interface 4의 XML 버전이 XPDL이다.

XPDL에서는 상호연동을 하는 대상들이 주고 받을 콘텐츠에 대한 사양을 정의하고 있다. 반면에 본 논문에서는 XPDL과는 다르게 상호연동 과정에서 주고 받게 될 콘텐츠 보다는 상호연동이 이루어지는 절차의 구현

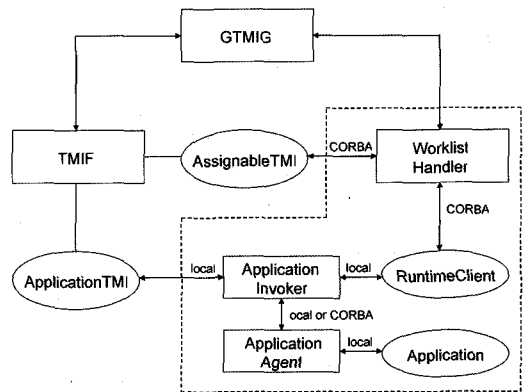


그림 10 기존 워크플로우 시스템의 구성

에 초점을 맞추고 있다. 왜냐하면 콘텐츠에 대해서는 실제로 연동하게 될 대상들 사이에서의 협의를 요구하기에 사전 작업에 큰 의미를 둘 수 없지만 상호연동이 이루어지는 절차나 방식은 시스템적으로 사전에 구현하여 적용시키는 것에서 의미를 찾을 수 있기 때문이다. 그리고 향후의 적용 과정에서는 이미 마련 되어 있는 연동 체계 하에서 주고 받을 콘텐츠에 관한 협의만 이루어지면 가능하다.

4.5.1.2 기업 내·외부 시스템간 연동방식

그림 11에서와 같이 기존 워크플로우 시스템에서는 외부 응용과 연동하는 방식으로 응용 에이전트 방식과 상호운영 방식을 채택하고 있다. 응용 에이전트 방식의 경우에는 단순한 호출로 충분한 단순 응용과의 연동에 주로 이용된다. 기간 시스템과의 연동의 경우에도 클라이언트 응용을 작성하여 응용 에이전트를 통해 호출할 수도 있다. 그러나 이 경우에는 제한적인 연동만이 가능하다. 완전한 연동을 위해서는 쌍방간에 콘텐츠를 주고 받을 수 있는 상호운영 방식이 적합하다.

본 논문에서 제안하는 eAI 지원 프레임워크는 웹 서비스를 기반으로 하기 때문에 기존의 상호운영 방식의 확장으로 볼 수 있다. 또한 역으로 웹 서비스를 통해서 기존의 상호운영 방식을 손쉽게 구현하는 것으로 볼 수도 있다.

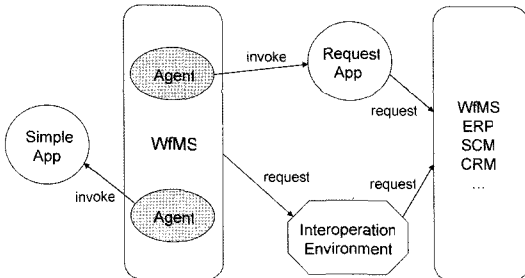


그림 11 기존 워크플로우 시스템의 응용 연동 방식

워크플로우 시스템의 웹 서비스화를 위해서는 워크플로우 시스템에 외부에 제공해야 할 인터페이스 부분에 대한 파악이 전제되어야 한다. 현재는 다양한 형태의 인터페이스 표준들이 존재하기 때문에 이를 기반으로 일반적인 인터페이스들을 손쉽게 파악할 수 있다. 본 논문에서는 WfMC에서 제안하는 다섯 가지 인터페이스를 바탕으로 OMG 그룹의 IDL 파일을 분석하여 원하는 인터페이스들을 추출한다.

4.5.1.3 워크플로우 시스템의 웹 서비스 구조

워크플로우 시스템의 웹 서비스 지원에는 두 가지 관점이 있을 수 있다. 하나는 워크플로우 시스템이 서비스 제공자가 되는 것이고 다른 하나는 워크플로우 시스템

이 서비스 요청자가 되는 것이다. 즉 서비스 제공자의 입장에서는 워크플로우 시스템의 일부 인터페이스를 웹 서비스 형태로 외부에 제공해야 하고, 서비스 요청자의 입장에서는 기간 시스템이 웹 서비스 형태로 공개한 인터페이스를 호출해야 하는 것이다. 본 논문에서는 워크플로우 시스템의 확장에 있어서 서비스 제공자로서의 역할에 비중을 두고 접근한다. 독립적으로 추가되는 모듈에 있어서도 대부분의 경우 웹 서비스로서 공개되어야 하는 인터페이스들에 대한 구현부를 포함한다.

그림 12는 워크플로우 시스템의 웹 서비스 지원 구조를 보여주고 있다. 인적 자원과의 통신을 전달하는 WorklistHandler가 웹 서비스의 지원을 담당하고 있음이 나타나 있는데, 이는 외부의 기간시스템을 외부의 참여자와 동일하게 취급함을 의미하는 것이다. 즉 WfMC 표준안에서의 자원 가운데 그 종류가 "system"에 해당되는 경우로 취급함을 의미한다. WfMC 표준안에서는 WorklistHandler의 통신 대상이 단순한 인적 자원에 국한되는 것이 아니라 외부의 자원을 포괄적으로 수용하는 것을 의미하고 있다.

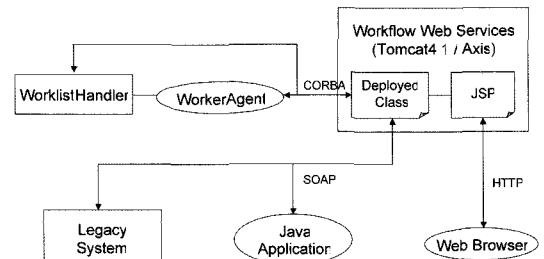


그림 12 워크플로우 시스템의 웹 서비스 지원

즉 WorklistHandler는 각 기간 시스템 별로 적절한 WorkerAgent를 지정하여 통신을 위한 창구로 활용하게 되는데 이 WorkerAgent와의 연동 인터페이스들이 자바로 구현되어 웹 서비스 서버에 배치된다. 외부의 서비스 요청자가 호출하는 웹 서비스는 Deployed Class의 절차에 매핑이 되고 배치된 클래스의 해당 절차는 CORBA 프로토콜을 통해 WorkerAgent와 통신함으로써 실제 서비스 루틴의 수행을 요청하게 되는 것이다. JSP 모듈은 웹 브라우저를 사용하는 서비스 요청자를 위해 추가되었다. JSP와 Deployed Class 사이에는 내부 호출(local call)이 가능하다. 따라서 외부의 요청자가 Deployed Class에게 서비스를 요청하는 방식에는 SOAP 프로토콜을 이용하는 직접 요청과 HTTP 프로토콜을 이용하는 간접 요청 방식을 지원한다.

그림 13은 .NET를 포함한 워크플로우 시스템의 웹 서비스 지원 구조를 보여주고 있다. 다만 자바 이외의

언어 플랫폼 환경에 대한 지원 방안이 추가되어 있다는 점에서 그림 12와 차이가 있다. 구체적으로는 새롭게 널리 쓰이고 있는 .NET 프레임워크를 연동 대상으로 포함하고 있다. 그 이외의 플랫폼 환경 또한 이와 유사한 형태가 될 것이다.

.NET 플랫폼이 추가되면서 조금 더 복잡한 시스템 구성이 되었지만 사실상 ASP 환경의 추가로 귀결된다. 물론 이 경우에도 가장 일반적인 연동 방식은 .NET 어플리케이션이 SOAP 프로토콜을 통해 직업 Deployed Class에 접근하는 것이다. 웹 브라우저를 이용하는 서비스 요청자의 경우에는 환경에 따라 JSP나 ASP를 통해 간접적으로 접근하게 된다. 이 경우 ASP와는 HTTP 프로토콜을 이용하게 되고, ASP는 SOAP 프로토콜을 이용하여 Deployed Class에 접근하게 된다.

한편, .NET 플랫폼을 위한 웹 서비스 지원 모듈을 개발하는 과정에서 자바 이외의 언어 환경에서의 웹 서비스를 위해서는 추가적인 작업이 요구된다. 대부분의 기본 타입 데이터의 경우에는 쌍방간의 지원이 이루어지기 때문에 별도 등록이 요구되지 않지만 기본 타입을 기반으로 파생되는 복합 데이터나 클래스의 경우에는 서비스 요청자의 언어 환경에 따라 별도의 코딩 작업이 요구되는 것이다.

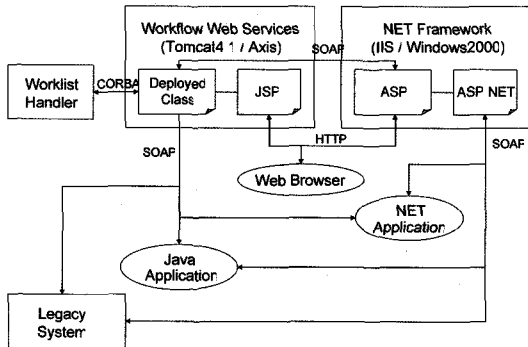


그림 13 워크플로우 시스템의 웹 서비스 제공(.NET 포함)

코딩 작업의 지원을 위한 스텝 파일의 생성을 위해서는 스텝 파일의 대상이 되는 데이터들을 일일이 등록해 주어야 하는 번거로움이 발생한다. 특히 등록 대상 데이터나 클래스의 공용 절차에서 쓰이는 매개변수나 반환 값에 쓰이는 데이터 타입에 대해서도 이러한 상황은 연쇄적으로 발생한다. 이런 연쇄적인 파생을 최소화 하기 위해서는 서비스 함수의 매개변수나 반환 값에 대해 가급적 단순 타입을 이용하고 불가피하게 복잡한 타입을 이용해야 할 경우에는 그 복잡한 타입에 대해서도 가급적 단순 타입이 이용되도록 하는 노력이 필요하다. 이런 연유로 기존 인터페이스 상에 나타나는 몇몇 클래스에

대해서는 포장 클래스를 새롭게 정의해야 하며, 이런 작업으로 인해 워크플로우 시스템의 웹 서비스화 지원 모듈의 규모는 당초 예상했던 것보다 커지게 된다.

4.5.1.4 워크플로우 시스템의 웹 서비스 모듈 설계

그림 14는 워크플로우 시스템의 웹 서비스 모듈에 대한 클래스 다이어그램을 보여주고 있다. 앞 절에서 언급했던 것처럼 포장(Wrap) 클래스들은 모두 이중 언어 플랫폼의 서비스 요청자들을 지원하기 위한 포장 객체 클래스들로서 포장을 뗀 부분이 워크플로우 상에서의 클래스 이름이 된다. 주로 WorklistHandler 패키지의 클래스들인데 이는 그림 상에서 워크플로우가 실질적으로는 WorklistHandler라는 것과 무관하지 않다. 서비스 이용 객체는 결국 워크플로우 입장에서는 외부의 자원으로 간주된다.

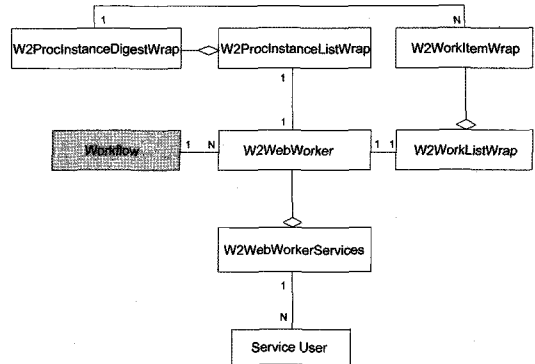


그림 14 워크플로우 시스템의 웹 서비스 모듈에 대한 클래스 다이어그램

워크플로우 시스템에서 외부에 공개하는 웹 서비스는 W2WebWorkerServices 클래스이다. 즉 웹 서비스를 위해서 웹 서비스 엔진에 배치 되는 기본 클래스가 바로 W2WebWorkerServices인 것이며, 나머지 클래스들은 W2WebWorkerServices의 공용 메소드의 반환 값이거나 매개변수로서 파생되어 배치되는 클래스들이다. 실질적으로는 W2WebWorker가 서비스 요청자 별로 배치되어 서비스를 전달한다. 따라서 W2WebWorkerServices의 역할은 서비스 요청에 대해 적절한 W2WebWorker를 추출하여 해당 메소드를 호출하고 그 처리 결과를 서비스 요청자에게 반환하게 된다.

W2WebWorker는 내부적으로 작업 리스트와 프로세스 리스트를 보유하고 있다. 작업 리스트는 서비스 요청자가 외부 자원으로 수행해야 할 작업들의 리스트를 의미하는데, 불가피하게 요구되는 클래스로 여겨졌기에 포장 객체를 도입하게 되었다. 프로세스 리스트는 서비스 요청자가 생성을 요청한 프로세스 인스턴스들의 리

스트를 의미하는데, 이 또한 포장 객체의 도입이 요구된다.

만약 같은 자바 환경의 서비스 요청자만을 대상으로 하는 경우에는 훨씬 덜 복잡한 클래스 다이어그램을 구성할 수 있다. 왜냐하면 모든 포장 클래스들이 제거되기 때문이고 아무리 복잡한 데이터라 할지라도 바이트 배열로 표현하여 주고받는 것이 가능하기 때문이다.

4.5.2 외부 웹 서비스 호출

워크플로우 시스템은 외부의 자원에게 작업의 수행을 지시하거나 직접 응용 프로그램을 호출하여 작업을 수행하게 된다. 이러한 기본적인 단위업무 관리자로부터 나머지 단위업무 관리자들은 파생되어 구현된다. 대부분의 파생 단위업무 관리자들은 업무 흐름 제어를 목적으로 도입되는 경우가 보통이다. 이처럼 워크플로우에서 지정된 프로세스를 수행함에 있어서 외부의 응용 프로그램 호출은 주요한 수단이다. 따라서 다양한 형태의 응용 프로그램 연결을 지원할 필요가 있으며, 본 논문에서 제안한 eAI 프레임워크는 웹 서비스를 기반으로 하는 연동 방식을 제안했기 때문에 외부의 웹 서비스 또한 응용 프로그램의 일종으로 포함된다.

그림 15는 워크플로우 시스템이 서비스 요청자로서 외부의 웹 서비스를 요청하는 구조를 보여주고 있다. 외부의 웹 서비스를 요청은 기본적으로 기존 워크플로우 시스템의 응용 프로그램 호출 방식으로 처리한다. 응용 프로그램을 호출하고자 하는 객체는 Application Invoker가 되며, 호출해야 하는 응용 프로그램 정보에 의거하여 Application Agent를 생성하고 호출을 담당하는 것이다. 이것을 구현하기 위해서는 웹 서비스 호출을 전담하는 별도의 Application Agent의 추가가 요구된다. 예를 들면 우선 서비스 제공자의 플랫폼으로 에이전트가 구현될 필요가 있다. 이는 워크플로우 시스템의 웹 서비스 지원의 경우처럼 데이터 타입의 불일치 해소와 연관되는 것이다. 따라서 단순히 웹 서비스에 등록하는 수준을 넘어서 웹 서비스 요청이 가능할 수 있도록 사전 작업이 필요하다.

이 밖에도 요구되는 작업이 몇 가지 있다. 우선 웹 서비스를 응용 프로그램으로서 워크플로우 시스템에 등록하는 작업이 요구된다. 이렇게 등록된 내용이 Application Agent를 생성하는 근거가 될 것이기 때문에, 여기에는 해당 Application Agent가 요구하는 정보가 반영되어야 한다. 아울러 프로세스 설계자에게 편리한 웹 서비스 등록 환경도 제공되어야 한다. 즉 웹 서비스 등록을 담당하는 도구가 직접 UDDI 레지스터리와 연동하여 검색 기능을 대행할 필요가 있다는 것이다.

또한 웹 서비스의 등록이 이루어 졌을 때, 응용 프로그램 형태로 호출하기 위해서는 후속 작업이 요구된다. 최소한 WSDL의 스텝을 가지고 어떤 형태로든 구동 용

용 프로그램이 작성되어야 한다. 이 작업의 과정에서도 앞서 언급했던 데이터 타입의 불일치 상황이 발생할 수 있다.

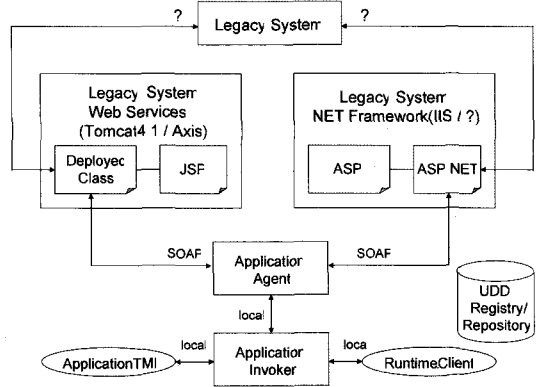


그림 15 워크플로우 시스템의 외부 웹 서비스 요청

5. 구현 및 평가

본 논문에서 제안한 eAI 프레임워크의 유용성을 확인하기 위하여 eAI 프레임워크의 프로토타입을 구현하고, 협업 물류 서비스라는 예제 시나리오를 eAI 프레임워크 프로토타입에 적용하여 그 결과를 평가화면으로 나타내었다.

5.1 구현 환경

제안한 eAI 프레임워크 프로토타입은 순수 자바로 구현되었다. 구체적으로는 JDK 1.3 이상의 환경에서 구현되었다. 작업 환경은 MS 윈도우 환경과 선 솔라리스 환경으로 서버는 솔라리스 환경에서 구현하였고 클라이언트는 윈도우 환경에서 구현하였다. 데이터베이스로는 오라클을 이용하였는데, 8.x 이상의 제품이면 실행 가능하다. HTTP 서버 및 JSP 엔진으로는 톰캣(Tomcat) 4.x 이상의 환경에서 구현하였다. 마지막으로 웹 서비스 서버로는 액시스(Axis)를 이용하였는데 1.x 이상의 환경에서 구현하였다.

5.2 협업 물류 서비스 시나리오

협업 물류 서비스에 관한 시나리오를 그림 16에 나타내었는데 전자조달 프로세스의 일부분을 추출하여 구성한 것이다. 점선으로 연결되어 있는 부분은 하위 프로세스 형태로 처리됨을 의미한다. 따라서 본 시나리오를 위해서 설계된 프로세스는 모두 세 가지이다. 본 프로세스는 구매요청서가 입력이며, 구매계약을 체결한 후 그 결과를 서비스 요청자에게 전달하는 것으로 완료된다. 견적서 요청하는 부분과 계약 체결하는 부분에서 외부 시스템과의 연동이 발생할 수 있다. 하지만 본 구현에서는 가상의 외부 업체를 설정하고 웹 서비스를 요청하는 형

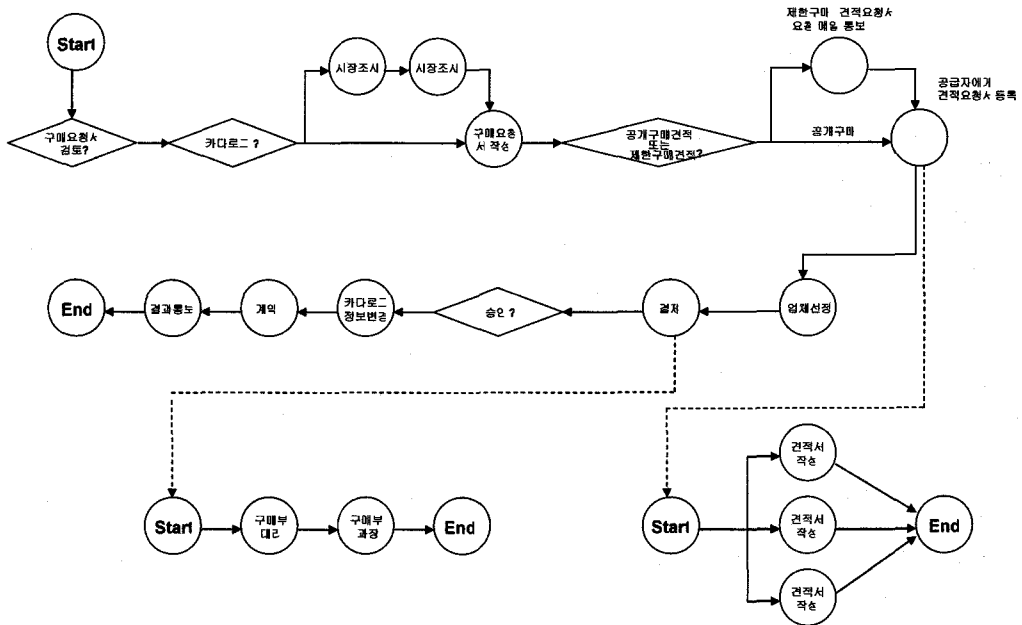


그림 16 협업 물류 서비스 시나리오

태로 처리하였다. 한편, 개발 절차 및 웹 서비스 모듈 내역에 관한 내용은 참고문헌 [21]에 상세하게 기술되어 있다.

5.3 평가 결과

5.2절의 협업 물류 서비스에 관한 예제 시나리오를 본 논문에서 제안한 eAI 프레임워크에 적용하였을 때 (a)의 구입 요청서 화면으로부터 (h)의 계약서 화면에 이르기까지 처리되는 각각의 과정 결과 및 평가 화면을 그림 17에 나타내었다. 처리 과정 화면에서 보여지는 바와 같이 요청서부터 계약까지의 과정이 기업 내·외부 응용 프로그램 연결을 포함한 상당히 복잡한 과정을 거침에도 불구하고 본 논문에서 제안되고 구현된 시스템을 사용하는 경우에는 몇 개의 간단한 UI 화면을 만드는 것으로 구현되는 것을 확인하였다.

5.4 고찰

5.3절의 평가 결과에서 보듯이 eAI 프레임워크 프로토타입 상에 5.2절의 시나리오를 적용한 결과 구체적으로 다음과 같은 점에서 본 논문에서 제안한 eAI 프레임워크의 효과를 확인할 수 있었다.

- 첫째, 기업 내·외부 응용 프로그램의 구별이 없어지는 장점을 확인하였다. 기존에는 기업 외부의 응용 프로그램을 연결하기 위해서는 방화벽을 포함한 추가적인 문제를 고려하여야 하지만 이 방식은 그러한 문제를 제거하여주고 있다.
- 둘째, 각종 다양한 형태의 응용 프로그램들이 eAI 프

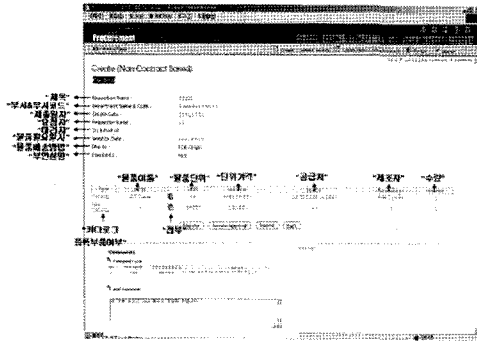
레이워크 안에서 별도의 작업 없이 시스템이 제공하는 몇 가지 API를 사용함으로써 손쉽게 연결되는 것을 확인하였다.

- 셋째, 이들 응용 프로그램들의 연결 과정이 워크플로우 시스템의 추적기능을 통해서 확인됨으로써 응용 시스템의 규모에 관계없이 문제가 되는 지점을 손쉽게 찾고 대처할 수 있음을 알게 되었다. 이상과 같은 관점에서 볼 때 제안된 시스템은 앞서 기술한 바와 같이 충분히 그 유용성이 있는 것으로 볼 수 있다.

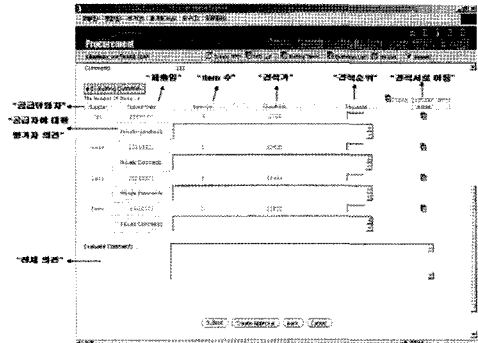
6. 결론

본 논문에서는 웹 서비스 표준 및 기술의 정비를 활용하면서 그리고 인터넷 환경에서 기업 내·외부 응용 프로그램 통합에 대한 요구에 부합하는 eAI 프레임워크를 워크플로우 시스템과 연계시켜 고안하고 설계하였다. 제안된 eAI 프레임워크는 eAI 플랫폼을 구성하는 웹 서비스 게이트웨이를 매개로 방화벽을 뛰어넘으면서 다양한 프로토콜로 외부 응용 프로그램과 연동할 수 있으며 MSH를 통하여 기존의 응용 프로그램들과도 연결될 수 있다.

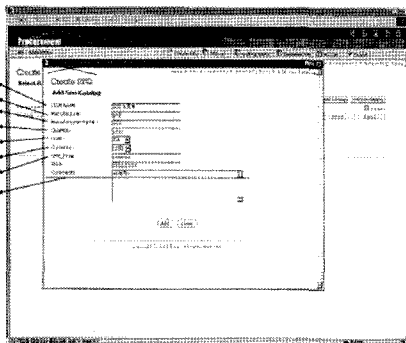
본 논문에서는 제안한 eAI 프레임워크의 프로토타입 구현에 관하여서는 워크플로우 시스템을 중심으로 비교적 자세하게 설명하였으며 웹 서비스와 연결되는 하나의 응용 프로세스를 제안한 eAI 프레임워크의 프로토타



(a) 구입 요청서 화면



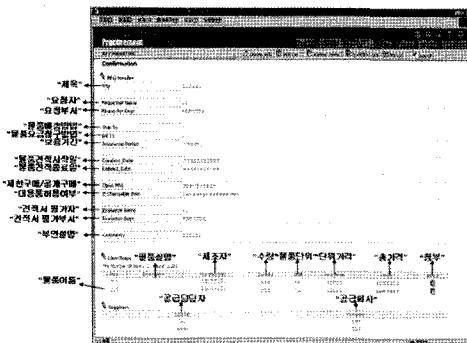
(e) 업체선정 화면



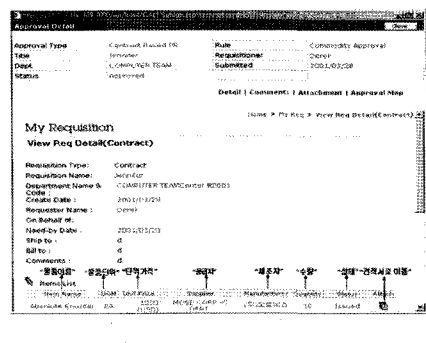
(b) 카타로그 화면



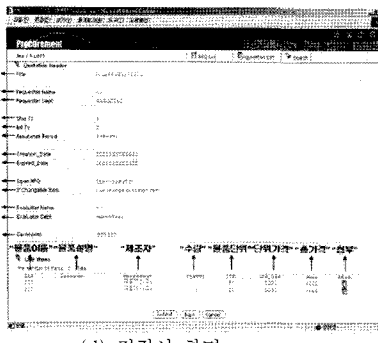
(f) 결제 화면 #1



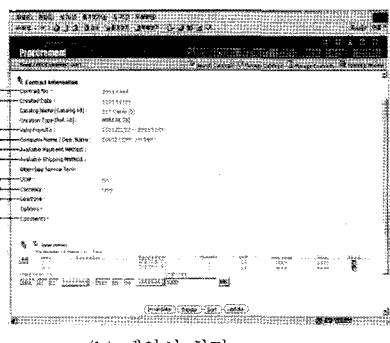
(c) 견적서 요청서 화면



(g) 결제 화면 #2



(d) 견적서 화면



(h) 계약서 화면

그림 17 서비스 처리 화면



입을 사용하여 구현함으로써 서두에서 제시한 시스템의 유용성도 확인하였다. 일반적으로 eAI 솔루션은 그 규모도 크고 복잡도도 높은 것으로 알려져 있다. 하지만 본 논문에서 제안한 eAI 프레임워크의 구성 방식은 주어진 웹 서비스 환경을 활용하고 몇 개의 핵심 모듈을 보강함으로써 기본적인 eAI 프레임워크의 구현이 가능함을 보여 주었다. 모든 기능이 잘 갖추어진 eAI 프레임워크를 만드는 데는 많은 노력과 시간이 필요하겠지만 제안된 프레임워크는 이와 같이 많은 기능이 갖추어진(full-fledged) 프레임워크 구성을 위한 토대로 활용하는 데 충분할 것으로 기대된다.

향후 웹 서비스 환경의 보편화와 더불어 제안된 프레임워크와 유사한 환경에서 이들 웹 서비스를 연결하여 새로운 비즈니스 프로세스를 구현하려는 시도가 많아질 것으로 예상된다. 이러한 시도에 효과적으로 대응하기 위해서는 본 논문에서 제안된 프레임워크를 eAI 프레임워크에 기반한 응용 프로그램 개발 환경으로 발전시켜야 할 것이다.

### 참 고 문 헌

- [1] Linthicum D., *Enterprise Application Integration*, Addison-Wesley, 2000.
- [2] Linthicum D., "Application servers and EAI," eAI Journal, 2000.7.
- [3] Linthicum D., "Application Integration Manifesto," eAI Journal, November 2003.
- [4] Schmidt, M.T., "The Evolution of Workflow Standard," IEEE Concurrency, Vol. 7, Issue 3, July-Sept. 1999.
- [5] J.J. Coyle, E.J. Bardi and R.A. Novack (1994) *Transportation*, Fourth Edition, New York: West Publishing Company, pp. 402.
- [6] Chandra Venkatapathy and Simon Holdsworth, "An introduction to Web Services Gateway," <http://www-106.ibm.com/developerworks/webservices/library/ws-gateway/>, May, 2002.
- [7] J. Shim and D. Han, "Exception Analysis of Structured Workflow Definition," The 20th IASTED International Multi-Conference, Applied Informatics, February 18-21, 2002 Innsbruck, Austria.
- [8] Y. Song and D. Han "Exception Specification and Handling in Workflow Systems," The Fifth Asia Pacific Web Conference, 23-25 April 2003, Xi'an China, also in Lecture Notes in Computer Science 2642, pp. 495-506, 2003.
- [9] M. Lee, D. Han and J. Shim "Set-Based Access Conflict Analysis for Structured Workflow Definition Language Information Processing Letters, Elsevier Science, Vol. 80, issue 4, pp. 189-194, Nov. 2001.
- [10] D. Han, J. Shim and C. Yu, "ICU/COWS: A Distributed Transactional Workflow System Supporting Multiple Workflow Types," IEICE Transactions of Information and Systems, Vol. E83-D, No. 7, July 2000.
- [11] P. Johannesson, B. Wangler, and P. Jayaweera, "Application and Process Integration-Concepts, Issues, and Research Directions," *Information Systems Engineering Symposium 2000*, eds. S. Brinkkemper, E. Lindencrona, and A. Sölvberg, Springer Verlag, 2000.
- [12] P. Johannesson and E. Perjons, "Design Principles for Application Integration," *12th Conference on Advanced Information Systems Engineering*, eds. B. Wangler and L. Bergman, Springer LNCS, 2000.
- [13] F. Curbera et al., "Business Process Execution Language for Web Services, Ver. 1.0," IBM, July 2002.; [www-106.ibm.com/developerworks/library/ws-bpel](http://www-106.ibm.com/developerworks/library/ws-bpel).
- [14] J.G. Hayes, E. Peyrovian, S. Sarin, M.T. Schmidt, K.D. Swenson, R. Weber, Workflow interoperability standards for the Internet, IEEE Internet Computing, Volume: 4 Issue: 3, May-June 2000, pp. 37-45.
- [15] Yan Zhang, Meilin Shi, Chunyu Miao, Zixin Zhuang and Xinxiang Chen, Workflow interoperability-Enabling E-business, Computer Supported Cooperative Work in Design, The Sixth International Conference on, 2001, pp. 403-408.
- [16] M. Kwak and D. Han, A Dynamic Interoperation Framework for Workflow Management Systems, Accepted to the 35<sup>th</sup> Hawaii International Conference on Systems Sciences (HICSS-35), Jan 6-9, 2002, Big Island, Hawaii.
- [17] M. Kwak, D. Han and J. Shim, "A Framework for Dynamic Workflow Interoperation and Enterprise Application Integration using Multi-Subprocess Task," IEEE RIDE 2002, San Jose, California.
- [18] J. Shim, D. Han and H. Kim "Communication Deadlock Detection of Inter-organizational Workflow Definition," Lecture Notes in Computer Science 2544, Springer Verlag, pp. 43-57, Dec. 2002.
- [19] WfMC, The Reference Model, Document number TC00-1003, Jan 1995.
- [20] OMG, Workflow Management Facility Specification, V1.2, 1997.
- [21] 한동수, 이정희, 장우혁, 송성대, 구중영, "협업 물류를 위한 기업 내부 시스템간의 연동방식 설계", 한국전자통신연구원 위탁연구 최종연구보고서, 2003년 12월.

이 성 독

정보과학회논문지 : 컴퓨팅의 실제  
제 11 권 제 5 호 참조

한 동 수

정보과학회논문지 : 컴퓨팅의 실제  
제 11 권 제 5 호 참조