

웹서비스를 이용한 Collaborative B2B 아키텍처 설계*

김태운
경성대학교 산업공학과
(twkim@ks.ac.kr)

김승완
경성대학교 산업공학과
(mksw25@hanmail.net)

한용호
부산외국어대학교 정보시스템학과
(yhhan@taejo.pufs.ac.kr)

본 논문은 웹서비스 프로토콜을 기반으로 B2B 애플리케이션을 위한 collaborative 아키텍처의 디자인을 목적으로 하며, 시스템의 가능성을 제시하기 위해서 이 아키텍처를 기초로 하여 개발한 시스템의 사례를 소개하고 있다. 상이한 비즈니스 프로세스가 B2B 환경에서 인터페이스가 되어져야 하므로 협업의 기능은 B2B 구현을 위해서 필요한 분야이다. 본 시스템의 개발 및 구현을 위한 플랫폼으로는 웹서비스, XML 및 ASP.NET을 이용하였다. 웹서비스는 애플리케이션간의 상호작용을 위해 체계적이고 확장 가능한 프레임워크를 제공한다. 웹서비스 프레임워크는 통신프로토콜, 서비스 기술 및 서비스 제공의 세부분으로 구성되는데 SOAP, WSDL, UDDI가 각각 여기에 해당한다. ASP.NET은 .NET의 상위 계층에 위치한 서비스 집합과 컴포넌트에 맞게 활용되어진다. B2B 환경에서 시스템 구현을 위한 상품의 범주와 이에 대한 상세한 설명의 서비스를 위해 웹서비스에서는 SQL 서버 데이터 베이스를 이용하였다.

논문접수일 : 2003년 1월

게재확정일 : 2003년 5월

교신저자 : 김태운

1. 서론

정보통신 기술은 공급 채널을 따라서 상호 작용하는 업무 기능들을 연결하여, 다수의 독립적인 회사들을 하나의 경쟁력 있는 공급 체인 시스템으로 변화시키고 있다. 정확한 정보의 가치가 증가하고 통신 속도가 향상됨에 따라서, 이제 선도적인 기업은 정보 통신 기술을 생산성 향상과 생산 주기 단축을 위한 관리 도구로서 뿐 아니라, 경쟁력 있는 조직 문화와 의사소통 구조를 활성화 시킬 수 있는 핵심 요소로 활용해 나가

고 있다.

최근에 B2B와 e-비즈니스의 발전으로 기업의 비즈니스 프로세스가 기업내부 뿐만아니라 기업 외부와 연동되어 구현되는 것이 필수적 요구되고 있다(오세영, 2002). 그러나 이러한 글로벌한 구현도 일차적으로는 기업 내부 프로세스가 원활히 구현되어져야 하며, 이를 기반으로 기업간 국제간에 비즈니스 프로세스가 상호 연결되고 처리되어져야 한다.

그간에 소프트웨어의 재 사용성과 개발자간의 협동 및 협조가 소프트웨어의 주요 이슈였으며,

* 본 연구는 2001년도 정보통신부의 대학기초연구지원사업으로 수행되었음.

이를 위해서 컴포넌트 기술이 각광을 받아왔다. 현재 많이 사용되는 컴포넌트 기술로는 COM (Common Object Model)/DCOM (Distributed Component Object Model), OMG (Object Management Group), Java Beans 등이 존재하나 이러한 기술들은 상호간에 호환되지 않는 문제점이 있다. 또한 이러한 컴포넌트 기술은 지금의 가장 보편적인 플랫폼인 웹과 호환성을 직접 갖지 못한다. 이에 반해서 XML (eXtensible Markup Language)은 플랫폼과 프로그램 언어에 독립적이고 확장가능하며 그 자체만으로도 데이터와 메시지 전달을 하는데 표현수단으로 이용 가능하다.

특히 IT관련 기업은 물론이고 전통적인 제조업이라 할지라도 B2B (Business-to-Business)나 e-Marketplace의 활용이 불가피하게 요구되고 있으므로 서로 환경이 다른 기업간에 문서, 데이터 및 정보들을 교환하기 위해서는 XML에 대한 요구가 급증하고 있다. 이러한 XML에 기초하여 근래에 새로운 웹서비스 개념이 도입되어 전자상거래 분야에서 이용이 시도되고 있으며 여러 기업간에 협업 상거래를 위한 프레임워크로 제안되고 발전되어 오고 있다 (Banerjee et al., 2000).

본 논문에서는 기업들 간에 서로 상이한 프로세스가 상호 인터페이스가 가능하도록 Collaborative한 웹서비스 환경에서 B2B를 위한 시스템을 설계하고 이를 구현하기 위한 아키텍처를 제안하고자 한다. 이를 위해서 웹서비스와 e-비즈니스 및 e-Marketplace, XML에 대한 최신 기술현황을 조사 분석하고 이를 바탕으로 구현을 위한 시스템의 구성요소와 개발환경을 설정하였다. 시스템 구현을 위하여 XML표준을 기반으로 하여 B2B가 실현가능한 시스템을 상호운용성이

용이한 웹서비스를 이용하여 설계하고, 부서간 기업간에 글로벌하게 비즈니스 프로세스가 인터페이스가 되는 프레임워크를 설계하고 ASP.NET 환경에서 구현하였다.

2. 관련 연구 조사 분석

2.1 웹서비스

웹서비스는 아직 진화 초기단계이므로 명확한 정의가 내려져 있지 않고, 정의 주체에 따라 웹서비스의 범위나 기능이 조금씩 다르다. 가트너 그룹은 웹서비스와 기존의 EAI (Enterprise Application Integration), Application Server, EIP (Enterprise Information Portal), 각종 인터넷 툴 등의 기능을 포괄하는 개념의 다이내믹 e-비즈니스웹 (DBW: Dynamic Business Web)를 제안하고 있는 IT사업자들의 개념을 받아들이고 있다 (Gartner, 2002). MS사와 IBM 등의 대형 IT벤더들이 웹서비스라는 용어를 쓴 것은 최근이지만 HP는 e스피크 (e-Speak)를 지난 95년부터 준비해왔고, IBM의 웹서비스 역시 기존 웹서버인 웹스피어 (WebSphere)를 기반으로 하고 있다.

기업들은 자사의 모든 전산시스템을 웹서비스로 구현함으로써 B2B, B2C, 고객서비스 등 각종 업무를 비즈니스 서비스로 만들 수 있으며, 개방형 표준기술 채택으로 플랫폼 등에 구애받지 않고 시스템 간 연동이나 원활한 업무처리가 가능해지는 것이다.

이런 업무환경을 구현하기 위해서는 무엇보다 기업의 분산컴퓨팅환경의 통합작업이 우선 이뤄져야 한다. 현재 기업정보시스템은 메인프레임

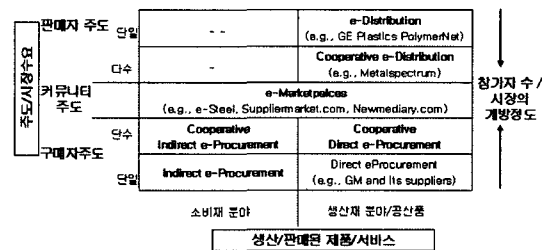
중심의 레거시시스템, 자체 개발한 클라이언트 서버시스템, 전사적 자원관리 (ERP), 고객관계 관리 (CRM)등의 패키지 소프트웨어, 그리고 웹 애플리케이션까지 다양한 운영시스템과 다양한 플랫폼이 혼재돼 있다. 이런 환경에서 애플리케이션 커뮤니케이션과 실시간 데이터 업데이트와 공유를 위해서는 EAI와 같은 새로운 애플리케이션이 요구된다. EAI솔루션은 각종 이기종 환경에서도 유연한 연동 및 데이터의 무결성을 보장하며, 고객이 단 한번의 ID입력만으로도 모든 관련 서비스를 이용할 수 있는 싱글 사인온 서비스를 웹 환경에서 구현할 수 있다. 특히 C-커머스 (Collaborative-Commerce)를 위해서는 사내 애플리케이션과 데이터베이스의 통합과 실시간 업데이트뿐만 아니라 방화벽 외부의 정보시스템과의 실시간 커뮤니케이션과 데이터 전환 및 통합이 요구된다.

결국 웹서비스는 유연성 (flexibility), 상호운용성 (interoperability), 보안성 (security)이 보장되어야 한다 (Cornes et al., 2002). 일반적으로 제안된 웹서비스 필수조건으로 '인터넷을 통해 제공되고, 인터넷 표준을 지원하며, 비즈니스 로직을 포함하는 서비스로, 객체기술에 기반을 둔 컴포넌트의 집합체'라는 개념이 제시 되었다 (Chappell et al., 2002).

웹서비스는 HTTP, TCP/IP 등의 표준뿐만 아니라, 차세대 인터넷 표준인 XML, SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), UDDI (Universal Description Discovery Interface)등을 지원한다. 이는 웹서비스가 플랫폼에 독립적이며, 상호운용성이 높은 서비스를 가능케 한다는 의미다.

2.2 e-비즈니스 및 e-Marketplace

기업간의 전자상거래인 B2B는 지난 몇 년간의 초기단계를 거쳐서 이제는 본격적으로 적용을 시도해 가는 단계에 이르고 있다. 이러한 B2B는 시장이 판매자 주도인가 소비자주도인가 아니면 복수의 커뮤니티가 주도하느냐와 생산 판매하는 제품/서비스가 소비재 생산재 여부와 시장의 개방정도에 따라서 e-Distribution, e-Procurement, e-Marketplace 등으로 구분되며, 기업의 업종과 특성에 따라서 해당되는 업무영역을 따르게 되는 이들 각각에 해당되는 내용은 아래 <그림 1>과 같다.



<그림 1> 시장수요, 제품/서비스 유형 및 시장참가자 수에 따른 B2B의 유형

e-Marketplace의 유형은 그 동안에 수직적 e-Marketplace에서 수평적 e-Marketplace로 발전해 오고 다시 수직과 수평이 복합된 통합적인 형태로 발전되어 오고 있다. 수직적 e-Marketplace에서 구매자 위주 마켓플레이스의 특징은 아래와 같다.

- 판매자가 구매자의 서버에 접속하여 판매활동을 추진
- 구매자가 대기업인 경우 적합한 형태
- 이전의 Extranet이 여기에 해당
- GE의 TPN/POST가 대표적인 경우이며, GE

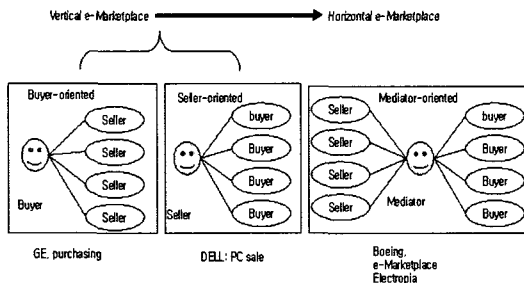
에서는 구매부서를 인터넷상에서 on-line화해서 추진

수직적 e-Marketplace에서 판매자 위주 마켓플레이스의 특징은 아래와 같다.

- 구매자가 판매자의 서버에 접속하여 구매활동을 추진
- 판매자가 대기업인 경우 적합한 형태
- DELL에서는 인터넷을 통한 PC 판매를 시행함
- Make-to-order에 의해서 유연생산 시스템을 도입하고 적정재고량을 유지

수평적 e-Marketplace에서는 하나의 판매자나 하나의 구매자 아닌 다수의 판매/구매자가 존재하며 이들 중간에 있는 중개인이 거래를 중개하는 형태를 취하는 것으로 이전에 한국의 무역 종합상사가 off-line상에서의 전형적인 이 형태에 속하며 특징은 아래와 같다.

- 다수의 구매자와 다수의 판매자들 사이에 거래를 중개하는 서버 존재
- 보잉사에는 전 세계에 있는 보잉 항공기에 대해서 각 항공사와 여러 유지/보수 회사들 간의 업무를 중개하는 역할을 수행
- 현재 추진중인 업종별 e-Marketplace도 여기



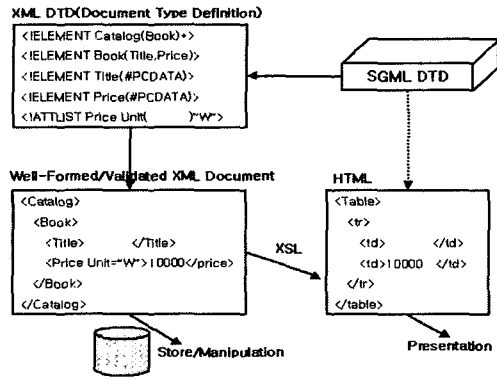
<그림 2> e-Marketplace의 유형

에 해당하는 것으로서 다수의 판매자와 구매자가 인터넷상에서 만날 수 있는 공간을 제공하는 것으로 B2B 전문회사가 여기에 해당하며 전자부품 분야의 Electropia도 여기에 해당하는데 이상에서 설명한 여러 유형이 <그림 2>에 나타나 있다.

2.3 XML 및 ebXML

HTML은 기본적으로 자료나 저작물을 인터넷 화면에 표현하는 방법을 정의하기 위한 마크업 언어로서 데이터의 공유에는 부적절하다. <그림 3>에서 볼 수 있듯이, 상품 정보를 테이블 형태로 출력할 경우, <table>... </table>로 그 형태를 정의하고, <tr>...</tr>로 줄을 구분하여 한 제품 정보를 구성하며, <td>...</td>로 제품명, 가격 등의 정보를 열로 구분하여 표현한다. 이러한 정보는 사람이 분석하여 의미를 파악할 수 있으나, 태그들을 바탕으로 컴퓨터가 그 의미를 해석하는 것은 매우 어렵다.

이러한 HTML의 문제점을 해결하기 위한 가장 간단한 방법은, 특정한 HTML 페이지의 테이블을 구성하는 열 등을 특별하게 해석하는 것이다. 상품 정보 검색 엔진들이 이러한 방법을 사용하고 있으나, 각 페이지별 특성을 추출해서 프로그램으로 작성해야 하며, 구조가 변화된 페이지들을 검색하여 프로그램을 수정해야 하는 등 변화에 취약하다. 또 다른 방법으로는 비표준 태그들을 사용하여 상품 정보를 제공하는 것이다. 이러한 비표준 태그들은 브라우저에서 무시되며, 스크립트와 같은 프로그램에 의해서 특별히 처리된다. 그러나 이런 방법은 여전히 문제가 있다. 즉 여러 사이트에서 다른 형식의 비표준 데이터를 받게 될 경우 각각을 따로 처리해야 하는 문



<그림 3> XML, HTML 및 SGML 연계

제가 생기게 된다.

HTML의 한계점을 근본적으로 해결하기 위한 방안으로, XML을 사용하는 기술이 속속 실현되고 있다. XML은, 간단히 표현하면, HTML과 같은 마크업 언어를 정의하여 특정 목적에 맞게 사용할 수 있도록 지원한다. 즉, 데이터의 의미와 구조를 표현하기 위한 태그들을 정의할 수 있어서, XML문서를 수신한 브라우저는 데이터의 의미를 이해하고 어떻게 처리해야 할지를 알 수 있다. 일례로, 상품 카타로그를 사용자에게 전달하기 위해서는 제품명과 가격을 `<Product Name="TV" Price =10000 />`와 같은 XML 표현을 사용할 수 있다. XML은 <그림 3>에서와 같이, 데이터 구조에 대한 정보를 DTD (Document Type Definition)로 기술할 수 있다 (Hunter et al, 2002). 따라서 이 메타 정보를 이용하여 문서 내 데이터의 구조적 완전성을 검사할 수 있다. XSL (XML Style Sheet)을 사용한 XML 문서의 구조적 변환도 가능해서, XML 정보를 웹브라우저로 검색하기 위해서 HTML 문서로 변환하는 것이 가능하다. 한편으로 XML을 발전시켜서 확장형 규칙 표식언어 (eXtensible Rule Markup

Language; XXML)을 제안하고 (이재규 외, 2002) 이를 이용하여 웹 페이지에 내재된 암묵적 규칙의 식별, 구조적인 규칙으로의 변환, 사람과 에이전트간의 지식공유를 가능하게 하며 이를 통해 지식 기반 시스템과 지식관리 시스템을 통합할 수 있는 새로운 개념도 최근에 소개되었다.

또한 XML은 다른 형식으로도 변환하여 사용할 수 있는데, <표 1>은 XML, SGML (Standard Generalized Markup Language), HTML을 상호간에 서로 비교하여 보여주고 있다.

ebXML은 UN/CEFACT와 OASIS에 의해서 XML을 전자상거래분야에 활용하기 위한 국제적인 공동연구의 결과이다 (UN/CEFACT and OASIS, 2002). 이러한 시도의 목적은 전 세계적인 XML의 실행을 표준화 하는데 있어서 기술적인 기반을 식별하고 조사하기 위함이다. 이 목적은 일관성 있는 일정한 방법으로 응용 프로그램 간, 응용 프로그램과 사용자 간, 또는 사용자와 응용 프로그램 간 환경에서 전자 거래 교환을 위한 XML 기반의 개방형 기술 프레임워크를 제공하여 단일한 전 세계적 시장을 탄생시키는 것이다. 하나의 전 세계적 전자상거래 시장이 탄생되기 위해서는 단일한 집합의 ebXML 기술규격들이 제정되어야 하며 이는 ebXML을 따르는 거래 상대방의 응용 프로그램과 상호 연동이 가능해야 한다. 또한, 공인된 EDI (Electronic Data Interchange) 표준에서 ebXML으로의 변환 과정을 지원하고 XML 업무 표준을 개발하는 동시에, 상호 연동성 및 효율성이 극대화되어야 한다.

ebXML은 기존의 국제 표준에 기반하고 있으며, 새로운 국제 표준이 되기 위하여 국제적으로 승인된 표준 기관에 제출될 계획이다. ebXML 시도에서 성공을 위한 핵심 포인트는 최대한도로 실행 가능 범위에서 W3C의 XML 규격과 이에

<표 1> XML, SGML, HTML 비교

기준	XML	SGML	HTML
상호 관계	SGML을 단순화한 것으로 사례 정의 가능	최초의 마크업 표준으로 사례 정의 가능	SGML로 정의된 한 사례
용도	전자상거래	문서화	프리젠테이션
확장성	가능(DTD정의 가능)	가능(DTD 정의 가능)	불가능(비표준 확장은 가능)
복잡도	다소 복잡	매우 복잡	간단
브라우저	현재 IE 5.0 지원	전용	범용
표준화 현황	XML 1.0 권고안 1998년 2월 XSL, XLL 표준화 진행 중	1986년 ISO8879 표준	1998년 HTML 4.0 권고안
데이터 공유	가능	가능	불가능
스타일 처리	XSL	DSSSL	CSS
전망	웹 변화의 기본 틀 표준화 주도 기업 경쟁력의 기준	CALS 문서화	XML의 한 사례로서 정의된 XHTML

연관된 Web 기술 규격에 충실한가 하는 점이다. 비록 이러한 규격이 최적의 기술 솔루션을 제공하지 못하더라도 업무적이고 기술적인 커뮤니티에 의한 ebXML의 수용은 XML과 연관되어 있다. ebXML을 전자상거래에 적용한 사례로서 최형림 (최형림 외, 2002)은 기업간 전자거래상에서 기존의 검색 시스템과의 비교를 통해 ebXML 프레임워크에서 효율적인 거래파트너 검색을 위한 XML 기반의 검색 시스템을 개발하여 구현한 경우를 보여주고 있다.

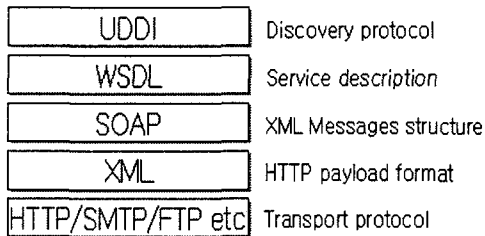
3. Collaborative B2B 아키텍처 모델링

중전에 분산환경에서 사용되어온 컴포넌트 기술들은 상호간에 호환이 되지 않아서 다른 프로젝트에 사용하기 위해서는 새로운 클래스를 만들거나 아니면 사용이 불가능한 문제점들을 가지고

있고 특히 웹과 호환성을 직접적으로 가지지 못하였다. 이러한 문제점을 극복하기 위하여 플랫폼과 프로그램 언어에 독립적이고 확장가능하며 그 자체만으로도 데이터와 메시지 전달을 가능하게 하는 표현수단인 XML을 바탕으로 한 웹서비스를 이용하고자 한다.

웹서비스는 웹 인터페이스를 통한 기능의 하나이며, 이는 HTTP와 같은 기존의 표준 인터넷 프로토콜을 통해서 이용이 가능하다. 즉, 어떤 클라이언트도 인터넷을 사용해서 서버에 접속하는 RPC (Remote Procedure Call) 같은 프로토콜 사용이 가능하고 XML로 응답을 받을 수 있다는 것을 뜻한다. 메시지가 클라이언트와 서버에서 전송될 때 SOAP이라고 불리는 특별한 XML로 인코딩된다. SOAP의 장점은 HTTP로 보내진다는 것이다. 대부분의 방화벽은 인터넷을 통한 HTTP 트래픽을 최종 사용자에게 전송하는 것을 허용하기 때문에 서로 상이한 컴포넌트간에 통신

이 가능하고 DCOM, CORBA (Common Object Request Broker Architecture), RMI (Remote Method Invocation) 같은 한 가지의 기술에만 의존할 필요없이 시스템을 분산 처리할 수 있게 해준다 (Cerami, 2002). 예를 들어, 전자상거래 애플리케이션을 만드는 경우 고객의 쇼핑카트에서 정확한 운송료를 계산하는 웹 서비스를 사용할 수 있다. 새로 운송료를 계산하는 계산기를 만들어 설치하고 컴포넌트를 설정하는 것 대신에 간단히 SOAP 메시지를 통해서 이미 그 기능이 있는 시스템에 접속하면 된다. 최종 사용자에게는 접속하는 모든 웹 사이트에서 개별적으로 구현하는 것보다는 운송료에 대해서 일괄적이고 정확한 정보를 제공해 주기 때문에 매우 큰 장점으로 작용할 수 있다. 이러한 웹서비스를 구성하는 서비스 스택 구조는 <그림 4>와 같다.



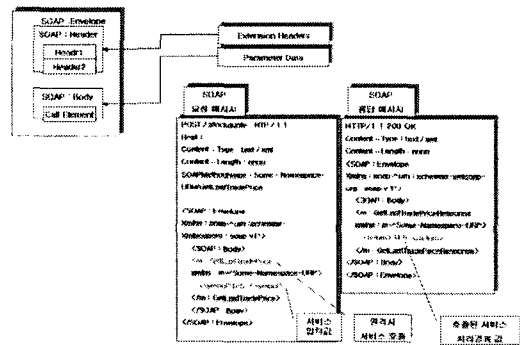
<그림 4> 웹서비스 스택 구조

3.1 시스템 구성요소

3.1.1 SOAP

SOAP은 MS사와 IBM사에 의해 급속하게 보급되고 있는 표준으로, 클라이언트의 작업 요청과 시스템의 응답을 XML 문서를 이용해서 표현하고 전송 프로토콜로는 HTTP를 사용하는데

SOAP 프로토콜의 구조는 <그림 5>와 같다. SOAP의 구조는 Envelope, Header, Body로 구성되어 있는데 Envelope는 외부의 컨테이너이고 Header는 라우팅, 처리 및 전달관련 정보를 서술하는 부분인데 선택적으로 사용가능하고, Body가 실제로 메시지의 내용을 서술하는 곳으로 이는 반드시 필요한 부분이다. 즉, SOAP은 HTTP와 XML의 결합으로서 분산 환경 하에서 정보의 상호교환을 가능케 하는 프로토콜이다.



<그림 5> SOAP 구조

SOAP은 XML뿐만 아니라 물론 과거 CORBA와 같은 통합기술, IBM의 메시징 기술 개발로부터 얻은 경험을 기초로 발전해 가고있다. SOAP의 인터페이스는 XML로 표현되며 HTTP와 e-mail과 같은 범용 프로토콜을 이용하여 인터넷으로 전송되기 때문에 싸고 간편하게 애플리케이션 소프트웨어들을 통합할 수 있으며 보다 간단한 컴퓨터 환경을 구축할 수 있다. SOAP을 이용하기 위해서는 약간의 프로그래밍 기술 능력이 요구되며 아직 해결해야 할 문제가 남아있는 것은 사실이지만 SOAP은 e-비즈니스 통합을 위해 중요한 도구가 될 것으로 전망된다.

3.1.2 WSDL

WSDL은 웹서비스의 IDL (Interface Definition Language) 버전이며 특정 웹서비스의 방법과 프로토콜, 데이터 포맷들을 더욱 상세하게 정의하는 일종의 스크립트이다. WSDL은 XML 포맷으로 구성되고 HTTP를 통해서 전달될 수 있으며 인터페이스를 정의하는 IDL에 해당한다. 즉, 특정 서비스가 어떤 방법과 속성을 가지며 어떤 인자로 호출해야 하고 어떤 방식의 리턴 값을 제공하는지를 알려주는 것이다. 이 내용을 알게 되면 클라이언트는 알게 된 인터페이스 규약에 맞추어 호출하고 서비스를 사용할 수 있게 된다.

3.1.3 UDDI

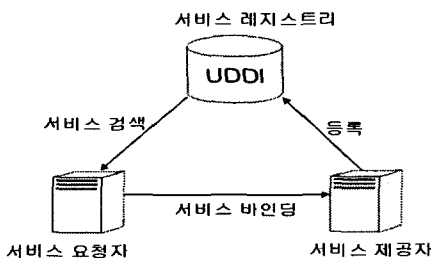
UDDI는 웹서비스에 관한 정보를 등록/검색하기 위한 분산형 웹기반 등록기를 말한다. 즉, UDDI의 목표는 웹 상에서 이용 가능한 전자서비스들에 대하여 분산형 웹 기반 글로벌 레지스터리를 만드는 것이다. 실제, 이것은 전자서비스들에 대하여 전자 전화번호부를 제공하는 것과 같다. UDDI를 개발하게 된 직접적인 동기는 서로 연관된 이질적인 시스템, 인터페이스, 프로토콜들을 인식하지 않은 상황에서 중개 소프트웨어와 중개자들을 통해 이용자들이 서비스들을 탐색하

고 상호 작용할 수 있는 전자 마켓플레이스의 개념을 구현하는 데 있으며 기본구성은 <그림 6>에 나타나 있다.

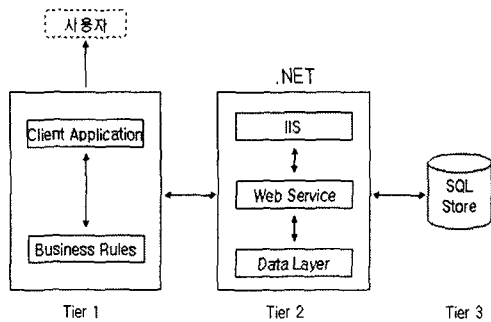
웹서비스 아키텍처에서 서비스 제공자는 서비스 레지스터리를 통해서 제공하는 서비스에 대한 기술을 제공하고 서비스 소비자 (고객)는 필요로 하는 서비스를 찾기 위하여 서비스 레지스터리를 검색하는데 서비스 소비자는 사람일 수도 있고 프로그램이 대신 수행 할 수도 있다 (Snell et al., 2002).

3.1.4 ASP.NET

본 연구에서 구현을 위하여 개발툴로서 ASP.NET을 사용하였는데 이 ASP.NET은 .NET 프레임웍의 맨 위의 계층을 차지하는 컴포넌트와 서비스의 집합이라고 할 수 있다 (Anderson et al., 2002). 웹서비스를 개발하기 위해 ASP.NET을 사용함으로써 개발자는 프레임웍으로 통합된 확장 클래스 라이브러리를 사용할 수 있다. 이렇게 함으로써 SOAP 메시지로 작업을 하고 WSDL 파일을 처리하는 데 필요한 모든 기초 기능들을 사용할 수 있게 된다. 또한 ASP.NET은 웹서비스 애플리케이션을 설정하고 실행하는데 필요한 작업들에 대해 부담을 덜어주는 런타임 환경을 많이 제공해 준다. ASP.NET은 .NET 프레임웍에서 웹서비스를 개발하는 방법이다. 다음 <그림 7>은 논리적인 다중 계층 애플리케이션 디자인을 웹서비스로서 여러 대의 컴퓨터에 분산시키는 모습을 보여준다. 폼을 갖춘 데이터세트 (typed dataset)가 HTTP 계층을 통해 비즈니스 규칙 계층으로 반환되고, 클라이언트 애플리케이션이 그 데이터세트를 사용하여 사용자에게 데이터를 표현하게 된다.



<그림 6> 웹서비스 아키텍처 기본 구성



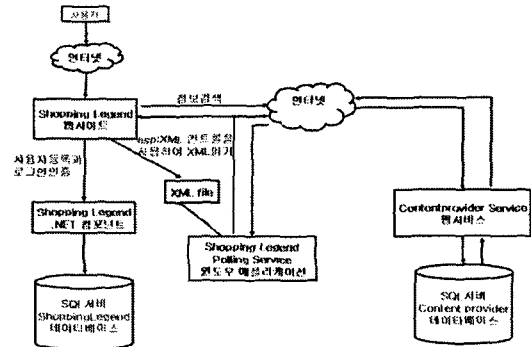
<그림 7> ASP.NET 기반 다중 애플리케이션

ASP.NET 웹서비스는 어떠한 사용자 인터페이스도 공개되지 않는다는 점을 제외하고 ASP.NET 웹폼 애플리케이션과 같은 개념으로 실행된다. 대신에 WSDL과 SOAP 인터페이스의 메소드를 통해서 일반적인 기능들을 공개한다. 또한 컴파일된 .NET 코드로 구성되어 있고, IIS에 있으면서 IIS에 있는 다른 웹 애플리케이션에게 하듯이 시스템 관리자에게 웹서비스를 보호하고 모니터링하는 데 필요한 모든 옵션을 제공한다(Bell et al., 2002).

3.2 시스템 아키텍처

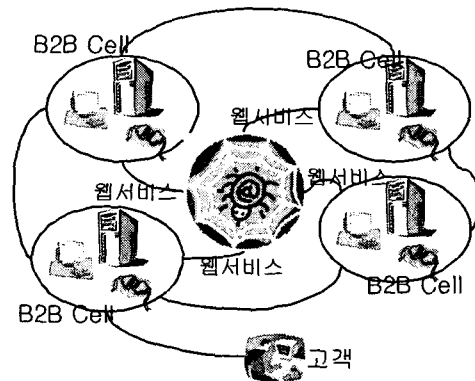
ASP.NET은 이러한 틀을 웹서비스 인터페이스를 기반으로 동적으로 WSDL 파일을 만드는 데 사용한다. 따라서 웹서비스가 개발되거나 바뀔 때 순수 WSDL 파일을 업데이트 하거나 바꿀 필요가 없다. 또한 클라이언트 접속 코드를 만들 필요없이 웹 브라우저를 통해서 웹서비스의 테스트를 빠르게 할 수 있도록 테스트 페이지를 만들어준다. 이렇게 SOAP과 WSDL등의 표준으로 구성되어 있는 웹서비스와 이와 연관된 웹 애플리케이션 서버를 구축하는데 ASP.NET을 활용하여 Collaborative B2B 아키텍처를 디자인 하였

다. 다음 <그림 8>은 웹서비스를 이용한 Collaborative B2B 아키텍처를 보여준다.



<그림 8> 웹 서비스 기반 Collaborative B2B 시스템 구성도

이러한 ASP.NET에 기초하여 웹서비스를 이용한 B2B 아키텍처가 <그림 9>에 나타나 있다. 각각의 셀 (B2B Cell)은 그 자체가 고객이 되면서 동시에 서비스가 가능한 서버가 될 수도 있고 그 셀 내에서는 비즈니스 프로세스가 독립적으로 수행되지만 서버 모드를 통해서 필요한 협업이 일어날 수 있다.



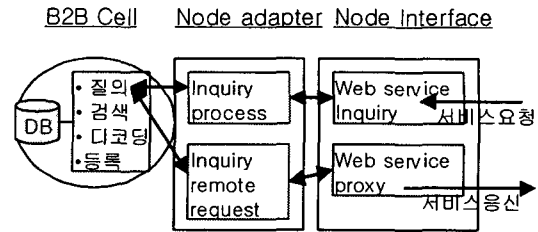
<그림 9> 웹서비스 기반 B2B Cell 구성도

기본적으로 각 셀에서는 질의 (Inquiry), 응답 (Response), 검색 (Retrieval), 디코딩 (Decoding), 서비스 등록 등의 기능이 자동으로 수행되며 그 셀은 근본적으로 서로 상이한 특성을 가지고 있으며 시스템 설계, 데이터 구조, 데이터 저장, 사용자 인터페이스 측면에서 각각의 특성으로 되어있다. 왜냐하면 B2B를 구성하는 회사나 벤더들이 자기 나름의 조직구조와 비즈니스 프로세스를 가지지만 웹서비스를 통해서 다른 기업과 협업이 가능하다.

이 구조에서는 대표적인 서버가 따로 없으며 인터넷으로 연결된 분산환경에서 웹서비스 프로토콜을 따라서 상호 작용하고 반응하게 된다. 또한 여기서는 각 셀이 서로 상이하므로 각각 저장된 정보를 상호 교환함으로써 분산된 기능들이 작동을 하게 된다. 따라서 표준화된 데이터 형식이 필요하게 되고 이를 위해서 XML이 이용될 수 있다. 각 셀은 분산된 환경에서 인터넷으로 상호 연결되어있으므로 각각의 파트너 셀과 정보를 교환함으로써 분산된 기능을 작동하게 된다. 따라서 웹서비스가 여기에 가장 적합한 대안이다.

아래 <그림 10>은 외부 고객, B2B 셀, 노드 어댑터, 인터페이스간의 관계를 보여준다. 외부 고객이 서비스에 대해서 하나의 질의를 하게 되면 웹서비스 질의를 통해서 노드 어댑터에 질의 프로세스를 요청하고 이것이 B2B셀의 DB 검색을 통해서 응답을 찾고 원격요청 컴포넌트를 통해서 인터페이스에 있는 proxy 객체가 웹서비스를 호출해서 회신을 한다. 즉 원격의 웹서비스를 불러오기 위해서 SOAP 요청 메시지가 서버측으로 보내지고 서버측의 노드 어댑터에 있는 질의 프로세스 컴포넌트의 서비스나 메소드를 호출함

으로써 질의에 대한 결과를 고객의 노드로 회신한다.



<그림 10> 웹서비스 노드간의 연결

이러한 과정에서 데이터교환을 위해서 DataSet 클래스가 사용되며, 이는 XML 표준인 XSD 스키마로 표현된다. 본 연구에서 구현대상으로 삼은 어떤 부품을 거래하기 위해서 필요한 필드는 다음과 같다.

- 부품의 이름/ID
- 카테고리
- 가격/수량
- 납기, 납품장소

노드 인터페이스에서 부품구매 질의에 대한 웹서비스의 반응을 카테고리과 가격의 두 인자의 관점에서 보면 다음과 같다.

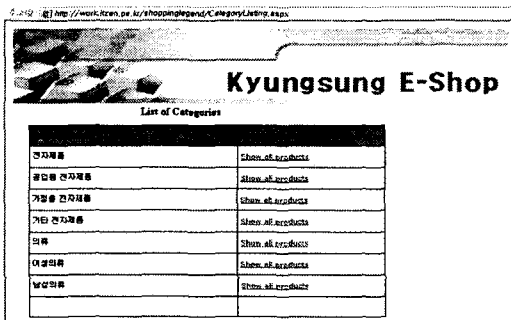
```
public B2BServiceDataSet
    GetServiceDataSetByCategory (String
        categoryCode);
public B2BServiceDataSet
    GetServiceDataSetByLocation (String
        locationCode);
```

어댑터는 셀 시스템 내의 관계형 데이터베이스를 검색해서 부품에 대한 레코드를 포함하는 DataSet을 만들어 준다. 이 어댑터는 ASP.NET 페이지로 나타나며 웹서비스의 요청에 반응해서 DataSet을 가져와서 웹서버 페이지나 웹 브라우저

저에 제공한다. 브라우저 상에서는 테이블 형태를 표현하기 위해서는 DataGrid 같은 객체를 사용 가능하다.

4. 웹서비스를 이용한 시스템 구현

이상에서 제안된 시스템을 구현하기 위하여 기업간 전자상거래에서 컴퓨터 관련 부품이나 상품을 조달하는 과정을 대상으로 하여 웹서비스를 이용하여 .NET 환경에서 시스템을 구축하였다. 시스템 구성도에서 제시된 대로 기업 혹은 개인 고객은 인터넷을 통해서 UDDI 기능과 같은 웹서비스에서 제공되는 서버에 접속해서 구매 혹은 조달가능한 상품목록을 검색하고 주어진 조건에 따라서 거래를 진행하게 된다. <그림 11>은 사용자가 웹사이트에 접속하였을 때 서비스 정보 리스트를 보여주는 화면이다.



<그림 11> 웹사이트 ShoppingLegend의 서비스 카테고리

<그림 12>는 사용자가 웹서버에 있는 서비스 목록에서 검색하고자 하는 정보를 요청하였을 때 웹서버에서 웹서비스에 보내어지는 SOAP Request의 형식을 보여준다.

SOAP Request

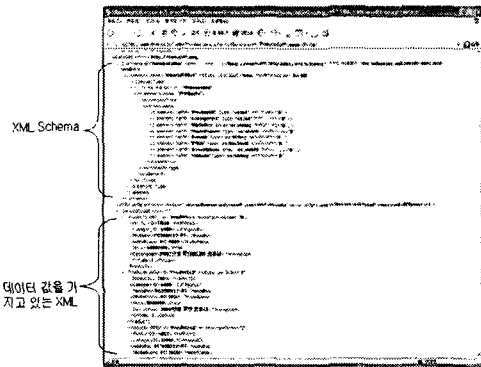
```
POST /ContentProviderService/ProductService.asmx HTTP/1.1
Host: work.litcen.pe.kr
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/ProductGet"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/XMLSchema-instance" xmlns:
  <soap:Body>
    <ProductGet xmlns="http://tempuri.org/">
      <CategoryID>int</CategoryID>
    </ProductGet>
  </soap:Body>
</soap:Envelope>
```

CategoryID가 1000인 Product 요청 int=1000

<그림 12> SOAP Request

데이터베이스를 위해서는 SQL 서버가 사용되었는데, 웹서비스가 DataSet 메소드를 호출하여 값을 구하고자 할 때 이 데이터베이스로부터 필요한 정보를 검색하고 값을 반환하게 된다. <그림 13>은 웹서비스가 DataSet메소드를 통해서 SQL 서버에서 가져온 데이터를 SOAP으로 인코딩해서 보내기 전에 웹서비스가 XML 형태로 저장하고 있는 모습을 보여준다.



<그림 13> 메소드 호출에 따른 처리결과

<그림 14>는 웹서비스에서 웹서버에 Response할 때 <그림 3>에 나타난 XML문을 SOAP으로 인코딩해서 Response할 때의 SOAP Response 의 형식을 보여준다.

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

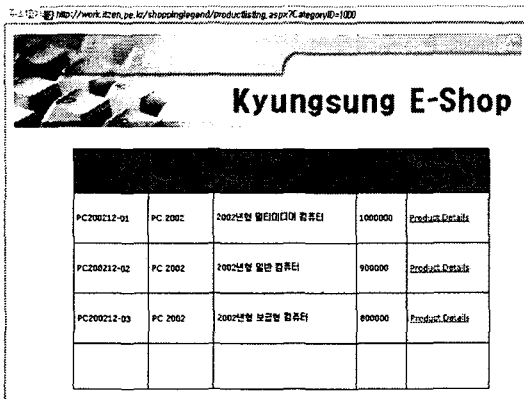
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xml:
  <soap:Body>
    <ProductsGetResponse xmlns="http://tempuri.org/">
      <ProductsGetResult>
        <xsd:schema>schema</xsd:schema>xml</ProductsGetResult>
      </ProductsGetResponse>
    </soap:Body>
  </soap:Envelope>
    
```

Schema 부분이
삽입된다.

데이터 값을 가지고
있는 Xml부분이 들어
간다.

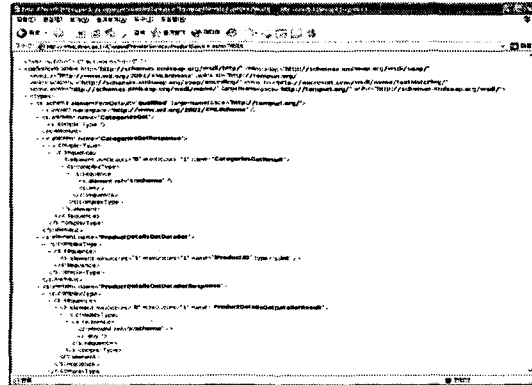
<그림 14> SOAP Response

<그림 15>는 SOAP을 디코딩해서 값을 가지고 있는 XML부분을 얻어서 이를 다시 HTML문으로 변환해서 웹브라우저에 보여주는 화면이다.



<그림 15> 웹사이트에서 사용자에게 보여지는 화면

<그림 16>은 웹서비스를 기술하는 WSDL을 보여주는데, 이는 개발자가 코딩할 필요없이 시스템이 자동으로 생성시키는 것으로, 여기에서는 ASP.NET환경에서 자동으로 생성되어져서 개발자에게 보여 지는 WSDL 화면이다.



<그림 16> 자동생성되는 WSDL 화면

5. 결론 및 추후 연구방향

본 논문에서는 기업들 간에 상이한 프로세스가 상호 인터페이스가 가능하도록 Collaborative한 웹서비스 환경에서 B2B 구현을 위한 시스템 아키텍처를 디자인 하고, 시스템의 가능성을 판단하기 위해서 이 웹서비스를 기초로 하여 구현한 시스템의 사례를 제시하고 있다. 이를 위해서 웹서비스와 e-비즈니스, XML등에 대한 최신 기술현황을 조사 분석하고 이를 바탕으로 구현을 위한 시스템의 구성요소와 개발환경을 설정하였다. 시스템 구현을 위하여 XML표준을 기반으로 한 웹서비스를 이용하였으며, 부서간 기업간에 글로벌하게 비즈니스 프로세스가 인터페이스가 되는 프레임워크를 설계하고 ASP.NET 환경에서 구현하였다.

본 연구를 통한 기대효과는 아래와 같다.

첫째는 플랫폼 독립성으로, 인터넷에 연결될 수 있는 어떤 애플리케이션이라도 SOAP 메시지를 인터넷상의 어떤 웹서비스에서도 주고받을 수

있다.

둘째는 다양한 커뮤니케이션 채널로서, 커뮤니케이션 메커니즘을 작동시키기 위해 인터넷에 의존하므로 사용하려는 솔루션이 방화벽 안에 있는 클라이언트의 데스크탑이나 미들-티어(middle-tier) 컴포넌트의 한 부분이든 간에 외부 웹서비스에 접속될 수 있다.

셋째는 기업 간 상호 이용성으로서, 사업 파트너와 공급자들 간에 비즈니스 시스템을 통합하고 자동으로 인터페이스가 가능하게 해 준다.

넷째는 기능의 재사용성으로, 소프트웨어를 개발하는데 시간이 걸리는 것을 줄일 수 있고, 컴포넌트를 만들 필요 없이 공개된 기능들을 사용하기만 하면 된다.

다섯째는 비즈니스의 확장성으로, 비즈니스를 서로 연관성 있게 확장하고 고객에게 더욱 쉽게 접근할 수 있도록 해주며 새로운 파트너가 참여할 때 전체 시스템 안에 쉽게 통합시킬 수 있다.

여섯째는 서버 중립성으로, 사용된 프로그래밍 언어와 서버에 독립적인데 이는 인터페이스가 표준을 기반으로 만들어졌고 메시지는 웹서비스와 클라이언트 간에 HTTP 위에 XML을 통해서 통신하기 때문이다.

추후 연구되어야 할 분야로서 기업들이 웹서비스를 이용하여 비즈니스를 수행하기 위해서는 먼저 보안문제가 가장 중요한 문제이다. 웹서비스가 구현되는 환경은 조직의 내부 뿐만 아니라 외부 조직과의 인터페이스가 기본으로 요구되며 웹의 공개성으로 인하여 글로벌한 연결이 불가피하다. 따라서 웹 서버내의 정보에 누구라도 접근이 가능하므로 이에 대한 보안이 강구되어야 한다. 또한 핵심 기술에 대한 인지도가 미비하고 아직 웹서비스가 완전히 표준화되어있지 않으

로 표준화 진행에 따른 변화에 대처능력이 필요하다.

참고문헌

- [1] 오세영, ".NET 2기 비전," *정보처리학회지*, 9권 4호(2002), 65-75.
- [2] 이재규, 손미애, 강주영, "확장형 규칙언어설계 원리 및 응용," *한국지능정보시스템학회논문지*, 8권 1호(2002), 141-157.
- [3] 최형립, 김현수, 최현덕, "ebXML 환경에서의 지식기반 검색 시스템," *한국지능정보학회 춘계학술대회*, 부산 Bexco, 5월(2002), 230-238.
- [4] 한동일, 김혜정, 이상수, "웹서비스 기술과 활용 전망," *정보처리학회지*, 9권4호(2002), 15-24.
- [5] Anderson, R., B. Francis, A. Homer, R. Howard, D. Sussman and K. Watson, *Professional ASP.NET 1.0 Special Edition*, Wrox, 2002.
- [6] Banerjee, A., A. Corera, Z. Greenvoss, A. Krowczyk, B. Maiani, C. Nagel, C. Peiris and T. Tangarathinam, *C# Web Services*, Wrox, 2000.
- [7] Bell, E., H. Fing, E. Soong, D. Zhang and S. Zhu. *Fundamentals of web applications using .NET and XML*, Prentice Hall PTR, 2002.
- [8] Cerami, E., *Web services*, O'Reilly, 2002.
- [9] Chappell, D., V. Chopra, I. Dubray and C. Evans, *Professional ebXML Foundations*, Wrox, 2002.
- [10] Cornes, O., C. Ullman, T. Libre and C. Goode, *Beginning ASP.NET using VB.NET*, Wrox, 2002.
- [11] Gartner, <http://www.gartner.com>, 2002.
- [12] Hunter, D., K. Cagle, C. Dix, R. Kovack, J. Pinnock and J. Rafter, *Beginning XML 2nd edition*, Wrox, 2002.

[13] Snell, J., D. Tidwell and P. Kulchenko,
Programming web services with SOAP,
O'Reilly, 2002.

[14] UN/CEFACT and OASIS, "ebXML: Creating
a Single Global Electronic Market,"
<http://www.ebxml.org>, 2002.

Abstract

Collaborative B2B architecture design using Web services

TaeWoon Kim* · SeungWan Kim* · YongHo Han**

This paper aims at the design of collaborative architecture for business to business (B2B) applications based on Web service protocol. As different business processes should be interfaced in the B2B environment, collaboration is important for the success of B2B implementation. For the development tools, XML, Web services and ASP.NET were adopted. Web services are emerging to provide a systematic and extensible framework for application-to-application interaction. The Web services framework is divided into three areas; communication protocols, service descriptions and Web discovery. Web services such as SOAP, WSDL and UDDI correspond to the three areas respectively. ASP.NET is utilized which corresponds to the component and service set located in the top layer of .NET. For the service of product category and product details, Web service architecture was implemented based upon the SQL server database.

Key words: Collaborative B2B, XML, SOAP, UDDI, WSDL, ASP.NET

* Dept. of Industrial Engineering, KyungSung University, Busan

** Dept. of Information System, Pusan University of Foreign Studies, Busan