ebXML 핵심 컴포넌트 기반 비즈니스 문서 생성 기술

ebXML Core Component-based Business Document Generation Technology

하영국(Young-Guk Ha)*, 손주찬(Joo-Chan Sohn)**, 손덕주(Deok-Joo Son)***

초 록

ebXML은 e비즈니스를 위한 글로벌 표준으로서 그 동안 기업간 협력의 가장 큰 장애물로 여겨졌던 정보 시스템의 이질성을 극복할 수 있는 대안을 제시하는 기술이다. ebXML에서는 전체적인 프레임워크를 구성하는 주요 요소로서 거래 절차를 기술하는 비즈니스 프로세스, 거래에 이용되는 공통의 용어와 그 의미를 정의하고 있는 핵심 컴포넌트, 거래에 필요한 정보를 저장하고 검색하는 레지스트리, 거래 문서의 전송을 위한 메시징 서비스 등을 정의하고 있다. 이러한 구조를 기반으로 ebXML에서의 거래 절차는 비즈니스 프로세스에서 요구하고 있는 다양한 비즈니스 문서의 교환에 의해서 수행되며, 이때 필요한 비즈니스 문서의 모델은 핵심 컴포넌트를 기반으로 저작된다. 본 논문에서는 우선 ebXML의 기술 구조 및 관련 기술 동향에 대하여 살펴보고, ebXML 핵심 컴포넌트 및 이를 기반으로 비즈니스 프로세스의 수행에 필요한 비즈니스 문서를 생성하는 기술에 대하여 분석 한다.

ABSTRACT

ebXML(Electronic Business XML) is a global eBusiness framework standard which provides solutions to interoperability problem between heterogeneous eBusiness systems. ebXML framework includes Business Process Specification, Core Components, Registry/Repository and Messaging Service as its major components. Based on this framework, ebXML business transactions are executed by exchanging appropriate business documents as specified in Business Process Specification. And required business documents are generated by assembling Core Components at the Business Process design time. In this paper, we present technical architecture of ebXML framework and Core Component technology, and then explain design and implementation issues of ebXML business document generation tool.

키워드 : ebXML, 핵심 컴포넌트, 비즈니스 문서, e비즈니스, 기업간 비즈니스

ebXML, Core Component, Business Document, eBusiness, B2B

* 한국전자통신연구원 비즈니스지식처리연구팀 선임연구원

** 한국전자통신연구원 비즈니스지식처리연구팀 팀장

***한국전자통신연구원 인터넷컴퓨팅연구부 책임연구원

1. 서 론

e비즈니스 프레임워크는 e비즈니스 거래를 위해 필요한 거래 당사자간의 상호 운용성과 호환성을 확보하고 전자 거래를 자동화하는 표준을 정의한다. 특히 글로벌 e비즈니스 프레임워크는 전산업과 전세계의 비즈니스 모델에 관계없이 전자 거래를 할 수 있는 기술적인 표준 제정에 초점을 맞추고 있으며, ebXML(Electronic Business XML)은 이러한 목적을 위해 UN/CEFACT와 OASIS가 주축이 되어 인터넷 및 XML 기술을 기반으로 제정한(현재도 작업중인) 전자 거래 프레임워크 표준이다.

ebXML 프레임워크는 거래 당사자들 간의 다양하고 가변적인 전자 거래를 지원하기 위한 비즈니스 프로세스 명세(BPSS, Business Process Specification Schema) 를 정의하고 있는데 이 명세를 기반으로 UML(Unified Modeling Language)과 같은 표준화된 모델링 기법을 통하여 만들어진 비즈니스 프로세스를 컴퓨터가 처리하여 정해진 절차에 따라 거래를 수행하게 된다. 전자 거래란 다양한 전자 문서를 메시징 서비스를 이용하여 거래 당사자간에 주고 받음으로써 이루어지며, 결국 원하는 업무를 수행하는데 필요한 비즈니스 문서(예를 들어 '상품카탈로그', '구매요구서', '구매요구응답서' 등)와 이러한 문서들을 특정 절차에 따라 사용하는 방법 등을 정의한 것이 비즈니스 프로세스이다.

글로벌 비즈니스를 수행하기 위해 요구되는 비즈니스 문서들은 매우 다양하고 복잡하며 지정학상의 특징, 상품의 종류, 거래상의 역할이나 비즈니스 도메인 등에 의존적인 성격을 갖는다(이러한 특정 비즈니스 환경을 비즈니스 컨텍스트라 함). 예를 들어 거래하는 당사자들의 소속 국가가 다른 경우 비즈니스 문서에서 사용되는 화폐의 단위나 환율이 달라지게 되며, 같은 개념을 갖는 용어일지라도 이종 업계에서는 상이한 표현을 쓸 수 있을 것이다. 또한 서로 다른 비즈니스 프로세스에서 동일하거나 유사한 비즈니스 문서를 사용하는 경우, 새로운 문서를 생성하기 보다는 이미 저작된 문서나 잘 정의된 문서의 구성 항목을 재활용 한다면 매우 효과적일 수 있다.

ebXML에서는 이러한 비즈니스 수행상의 문제점을 해결하고 컴포넌트의 재활용성을 극대화하기 위하여 비즈니스 문서를 구성하는 재활용 가능한 빌딩 블록으로서 핵심 컴포넌트(CC, Core Component)를 정의하고 있으며, 핵심 컴포넌트는 의미(Semantic)에 의존적이지 않은 하나의 특정한 개념(Concept)을 표현한다(예를 들어 'PostalAddress'라는 핵심 컴포넌트는 '주소'라는 개념을 표현하지만 상황에 따라 '발신인 주소', '수신인 주소', '상점 주소' 등과 같이 다양한 의미로 비즈니스 문서상에 나타날 수 있음).

본 논문에서는 ebXML 표준에서 정의하고 있는 핵심 컴포넌트를 기반으로 비즈니스 문서를 생성하는 기술에 대하여 살펴보기로 한다. 2장에서는 ebXML 프레임워크에 대하여 좀더 자세히 알아보고, 3장에서는 핵심 컴포넌트와 비즈니스 정보 엔티티(BIE, Business Information Entity)에 대하여 살펴보기로 한다. 4장에서는 CC를 기반으로 컨텍스트 구동

방식(Context-driven)의 비즈니스 문서 저작 기술에 대하여 알아보고 5장에서는 ebXML 핵심 컴포넌트 표준을 기반으로 하는 문서 생성기의 구현에 대해 설명한다.

2. ebXML 프레임워크

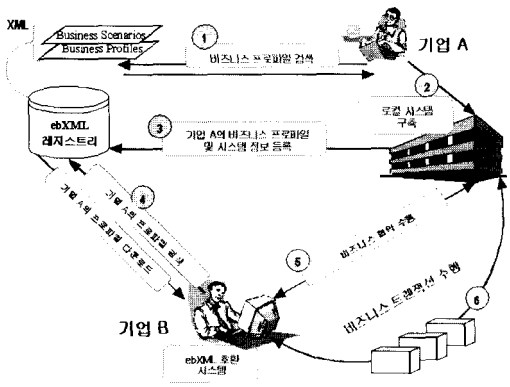
기업간의 전자 거래(B2B)가 붐을 조성하면서 가장 문제가 되었던 점은 전자 거래를 위한 표준의 부재이다. 다시 말해서, 특정 산업 내의 몇몇 선진 업체들을 중심으로 비즈니스 관련 문서뿐만 아니라 처리 절차 및 메시지 교환 방법 등에서 각기 다른 방식으로 구현되어 운영되고 있다는 것이다. 또한 최근 인터넷 사용의 폭발적인 증가와 더불어 글로벌 e비즈니스의 필요성이 절실히 요구되고 있는 가운데, 다양한 국제 비즈니스 환경에 대응하기 위해서는 거래 채널이 생길 때 마다 새로운 거래 당사자들과의 거래를 위하여 끊임 없이 새로운 인터페이스를 개발해야 하는 비효율성이 증대될 것이다. ebXML 프레임워크는 바로 이러한 문제점을 해결할 수 있는 방법을 제시하고 있다.

1.1 ebXML 시나리오

<그림 1>은 두 거래 당사자(기업 A 및 B) 간의 B2B 비즈니스 거래에 대한 개념적인 모델을 보여준다[4].

우선 A라는 기업은 인터넷을 통하여 비즈니스를 하기 위하여 ebXML 레지스트리[6]에 접근하여 현재 등록되어 있는 비즈니스 프로세스 명세(BPS, Business Process Specification)[1] 및 거래 상대방과의 협업을 위한 프로파일(CPP, Collaboration Protocol Profile)[3]을 검색해 본다(<그림 1>의 ① 단계).

ebXML을 사용하여 비즈니스를 수행하기로 결정한 회사 A는 자체적으로 ebXML 표준 명세를 따르는 응용 프로그램을 구현하거나 상용 제품을 구매하여 자사의 시스템에 설치한 후(<그림 1>의 ② 단계), 자사의 시스템에 대한 구현 세부 사항 및 CPP 등에 대한



<그림 3> ebXML 레지스트리 전체 시스템 구조

정보를 레지스트리에 등록한다(〈그림 1〉의 ③ 단계). CPP에는 회사 A에서 지원하고 있는 비즈니스 프로세스뿐만 아니라 ebXML 표준 지원 능력과 제약 사항 등을 명시하고 있다.

기업 B 역시 인터넷을 이용한 비즈니스를 수행하기 위하여 ebXML 레지스트리를 검색하여 회사 A의 CPP를 발견하고(〈그림 1〉의 ④ 단계), 회사 A와 거래하기 위하여 ebXML을 이용한 비즈니스에 참여하고 싶다는 요청을 회사 A에 보내고, 최종적으로 동의한 두 회사는 협업을 위한 비즈니스 협약[3](CPA, Collaboration Protocol Agreement)을 맺는다(〈그림 1〉의 ⑤ 단계). 이후 두 회사는 ebXML을 이용하여 비즈니스 거래를 수행한다(〈그림 1〉의 ⑥ 단계).

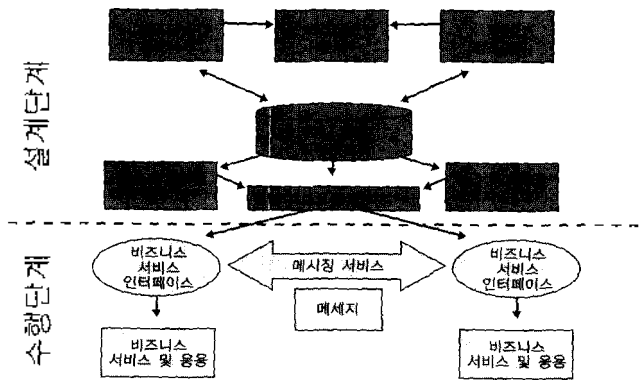
1.2 ebXML 구성 요소

전 세계를 하나의 e비즈니스 프레임워크로 통일하겠다는 대규모 표준화 작업을 위하여 UN/CEFACT와 OASIS에서는 비즈니스 프로세스(Business Process), 핵심 컴포넌트(Core Components), 협업 프로토콜(Collaboration Protocol), 메시징(Messaging), 레지스트리(Registry and Repository) 및 적합성(Implementation, Interoperability and Conformance) 분야로 나누어 작업반을 구성하고 병렬적으로 상호 협력하여 표준화 작업을 진행하였다(〈표 1〉 참조). 또한 정기적으로 개최하는 국제 회의에서 POC(Proof of Concept)를 통하여 각각의 작업반이 진행하고 있는 내용의 자체 검증과 다른 작업반과의 상호운용성에 대한 검증을 동시에 수행하였다.

〈그림 2〉의 ebXML 기술 구조도에서 보듯이 기술 작업반들의 작업 내용과 그 결과물이 바로 ebXML의 핵심 구성 요소를 이루고 있음을 알 수 있다.

1.3 ebXML 기술 동향

이번 절에서는 해외 선진 업체 및 국내의 ebXML 관련 기술 동향에 대하여 살펴보기로 한다.



〈그림 2〉 ebXML 구조

〈표 1〉 ebXML 주요 구성 요소 및 관련 작업반

구성요소	내 용	결과물(표준 명세)
비즈니스 프로세스 (UN/CEFACT ebTWG BP Project Team)	Business Process: 거래 절차에 대한 내용을 표준화된 방법(UML)으로 모델링하고 이를 자동으로 인식하여 거래가 이루어지도록 함	Business Process Specification Schema v1.01 [1]
핵심 컴포넌트 (UN/CEFACT eBT WG CC Project Team)	Core Component: 전자 거래에서 상호 교환되는 메시지(비즈니스 문서)를 구성하는 기본 단위들을 공유하여 재활용할 수 있도록 함	Core Components Technical Spec v1.8[2]
협업 프로토콜(OASIS ebXML CPPA TC)	Collaboration Protocol Profile & Agreement: 거래하고자 하는 거래 당사자에 대한 각종 정보와 협업을 위한 프로파일 모델을 정의함	Collaboration-Protocol Profile and Agreement Spec v2.0[3]
메시징 서비스(OASIS ebXML MS TC)	Messaging Service: 상호 교환될 메시지를 플랫폼에 상관없이 안전하게 전송할 수 있도록 함	Message Service Spec v2.0 [7]
레지스트리(OASIS ebXML RR TC)	Registry & Repository: 거래에 필요한 각종 정보들을 저장 공유하고 표준화된 방식으로 접근하도록 기본 모델을 제공함	Registry Information Model v2.0[5] & Registry Services Spec v2.0[6]
적합성(OASIS ebXML IIC TC)	Implementation, Interoperability & Conformance: e비즈니스 솔루션 개발 업체에게 표준 명세에 부합하고 서로 상호운용이 가능한 시스템을 개발하도록 함	Message Service Impl. Guidelines v0.2 & ebXML Test Framework v0.1[8]

■ Sun Microsystems

Sun은 자신들이 개발한 Java 플랫폼이 XML이나 ebXML과 같은 개방형 표준과 쉽게 결합될 수 있는 기술임을 내세우며 ebXML 표준을 지원하는 다양한 Java 기반의 API[17]를 제공하고 있다. 또한 각 ebXML 작업반에서도 활발히 활동하고 있으며 경쟁업체가 제안한 표준일지라도 Sun의 기본 방침과 뜻을 같이하는 표준은 자사의 기술과 접목하는 작업을 추진하고 있다.

■ IBM

IBM은 H/W뿐만 아니라 엔터프라이즈를 대상으로 개방형 표준으로 구성된 e비즈니스를 위한 플랫폼 시장에 많은 투자를 하고 있다. 특히 ebXML에 있어서 중심적인 역할을 하고 있는 CPP 및 CPA의 초안을 IBM에서 제시하였으며 메시징 서비스에 있어서 봉투 역할을 하는 SOAP(Simple Object Access Protocol)[15]은 Microsoft와 함께 강력하게 지원하고 있는 부분이다.

■ Microsoft

Microsoft는 자사의 Biztalk 서버[18]를 XML 기반의 e비즈니스 솔루션으로 소개하였다. Biztalk은 ebXML의 메시징 서비스에 해당하는 부분으로서 ebXML의 전체적인 기능과는 비교할 수 없는 수준이지만 이미 관련 시장에서 많은 고객을 확보하고 있는 상태이다. 또한 Biztalk은 자사의 e비즈니스 플랫폼인 .NET 플랫폼에 포함되어 진화될 것이다.

■ Fujitsu

Fujitsu는 ebXML 표준화 작업 초기부터 참여하여 꾸준히 ebXML에 대한 투자를 늘려왔다. 최근 OASIS의 공식 후원 업체로 등록하며 아시아권에서는 ebXML의 선두 기업의

위치에 서있는 Fujitsu는 분산 환경을 지원하고 있는 Iterstage[19]라는 B2B 플랫폼에 ebXML 뿐만 아니라 J2EE(Java 2 Enterprise Edition), RosettaNet[20], UDDI(Universal Description, Discovery and Integration)[21] 등과 같은 차세대 인터넷 기술을 접목하려는 야심을 가지고 있다.

■ 국내 동향

국내에서는 지난 10여년간 EDI(Electronic Document Interchange)기반의 전자 거래의 중심 역할을 수행해왔던 KTNET이 ebXML 표준화 작업 초기부터 참여하고 있으며, 국제적으로도 최초로 ebXML 레지스트리를 구현하여 한국전자거래진흥원의 국가 중앙등록저장소[22]로 활용되고 있다. 또한 최근에는 국내의 많은 e비즈니스 및 XML 솔루션 업체들이 ebXML 기술 개발에 박차를 가하고 있다.

3. ebXML 핵심 컴포넌트

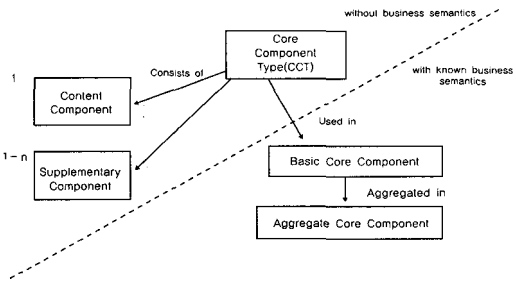
앞서 ebXML 기술 구조 전반에 대하여 살펴보았다. 본 장에서는 ebXML 구성 요소 중에서 특히 비즈니스 문서 생성에 필요한 핵심 컴포넌트에 대하여 좀더 자세히 알아보기로 한다.

3.1 핵심 컴포넌트의 개념

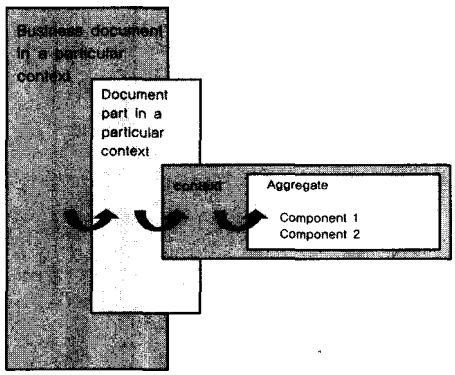
핵심 컴포넌트는 비즈니스 문서를 구성하기 위해 재사용 가능한 빌딩 블록(Building Block)의 개념을 제공한다[11].

핵심 컴포넌트[2]는 〈그림 3〉과 같이 핵심 컴포넌트 형(CCT, CC Type), 원소성(Atomic)을 갖는 기본 핵심 컴포넌트(BCC, Basic CC) 및 기본 핵심 컴포넌트의 집합으로 이루어진 집계 핵심 컴포넌트(ACC, Aggregate CC)로 분류할 수 있다.

핵심 컴포넌트 형은 핵심 컴포넌트의 실제 내용을 저장하지만 특별한 의미는 없는 내용 컴포넌트(Content Component)와 별도의 의미를 내포하는 보충 컴포넌트(Supplementary Component)로 이루어지며, 단일 비즈니스 개념을 표현하는 기본 핵심 컴포넌트는 핵심 컴포넌트 형으로부터 생성되어 복합적인 개념을 표현하는 집계 핵심 컴포넌트를 구성하는 데 이용된다.



〈그림 3〉 핵심 컴포넌트 구성[2]



〈그림 4〉 핵심 컴포넌트 개념[2]

〈그림 4〉와 같이 ebXML 비즈니스 문서는 특정 컨텍스트하에서의 기본 핵심 컴포넌트 및 집계 핵심 컴포넌트의 집합에 의해 구성됨을 알 수 있다.

3.2 핵심 컴포넌트와 레지스트리

e비즈니스에 필요한 각종 정보들을 저장 및 관리하고 표준화된 방식을 통하여 접근하도록 하는 ebXML 레지스트리(및 리파지토리)는 ebXML 프레임워크를 구성하는 중심적인 요소라 할 수 있다. 핵심 컴포넌트 또한 레지스트리에 등록되고 레지스트리로부터 검색되어 비즈니스 문서를 생성하는데 사용된다[5][6].

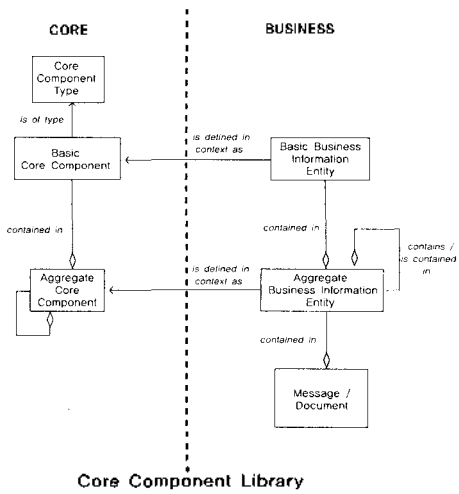
핵심 컴포넌트는 XML 스카마[14]로 정의되어 UUID(Universal Unique ID)[12]를 갖는 레지스트리 객체로서 등록되어 관리된다. 〈그림 5〉는 날짜와 시간을 표현하는 핵심 컴포넌트를 XML 스키마로 정의한 예(DateAndTime.Details) 이다.

```
<schema xmlns="http://www.w3.org/2001/XMLSchema">

    <!-- Core Component Type: Date_Type -->
    <simpleType name="Date_Type">
        <restriction base="date"/>
    </simpleType>

    <!-- Core Component Type: Text_Type -->
    <simpleType name="Text_Type">
        <restriction base="string"/>
    </simpleType>


    <!-- Aggregate Core Component: DateAndTime_Details -->
    <complexType name="DateAndTime_Details">
        <sequence>
            <element name="Date_Details" type="Date_Details"/>
            <element name="Time_Details" type="Time_Details"/>
        </sequence>
    </complexType>

    <!-- Aggregate Core Component: Date_Details -->
    <complexType name="Date_Details">
        <simpleContent>
            <extension base="Date_Type">
                <attribute name="Date_Format_Text" type="Text_Type"/>
            </extension>
        </simpleContent>
    </complexType>

    <!   Aggregate Core Component  Time_Details -->
    <complexType name="Time_Details">
            ...
    </complexType>
            ...


</schema>
```

〈그림 5〉 핵심 컴포넌트 정의

3.3 비즈니스 정보 엔티티

비즈니스 정보 엔티티(BIE, Business Information Entity)는 유일한 비즈니즈적인 의미(Semantic)를 갖는 비즈니스 정보의 단위로서 핵심 컴포넌트가 실제로 특정 비즈니스에서 사용될 때 BIE로 정의된다[2]. 다시 말해서 BIE는 핵심 컴포넌트에 특정 비즈니스 컨텍스트가 적용된 결과이다(〈그림 6〉 참조). 비즈니스 정보 엔티티는 기본 비즈니스 정보 엔티티(BBIE, Basic BIE) 및 집계 비즈니스 정보 엔티티(ABIE, Aggregate BIE)로



〈그림 6〉 비즈니스 정보 엔티티[2]

구성되는데. BBIE 및 ABIE를 조합하여 비즈니스에 필요한 문서 모델을 생성하게 된다.

3.4 핵심 컴포넌트 카탈로그

ebXML 프레임워크에서 명시하고 있듯이 모든 핵심 컴포넌트는 ebXML 레지스트리에 등록되고 레지스트리에 연결된 리파지토리에 저장되어야 한다. 그러나 중소규모의 조직이나 기업에서 적시에 이러한 ebXML 명세를 준수하는 시스템을 구축하여 표준 레지스트리에 접근하기란 쉬운 일이 아니다.

따라서 레지스트리에서 지원하는 전체 핵심 컴포넌트에 대하여 자유롭게 배포할 수 있는 카탈로그 형태로 관련 정보를 작성하여 관리하는 것이 중요하다. 핵심 컴포넌트 카탈로그(CC Catalogue)[2]는 레지스트리에 등록된 각 컴포넌트가 UML 객체[16]로서 어떻게 리포지토리에 저장되어 있는지 알려주는 상세한 정보[9]를 제공해야 한다(〈그림 7〉 참조).

핵심 컴포넌트 카탈로그는 핵심 컴포넌트 라이브러리(CC Library)를 구성하는 한 부분이며 핵심 컴포넌트 라이브러리 전체는 다음과 같은 부분으로 구성되어 있다.

- *Core Component Types*
- *Core Component Catalogue*
  - Basic Core Components
  - Basic BIEs
  - Aggregate Core Components
- *Catalogue of Aggregate BIEs*

3.5 비즈니스 컨텍스트

비즈니스 컨텍스트(Business Context)는 특정 비즈니스 상황에 대한 형식적 표현으로서 컨텍스트 카테고리(〈표 2〉)와 그에 따른 컨텍스트 값(각 카테고리 별로 레지스트리에 등록된 Classification Scheme[5]에서 정의된 값)의 쌍으로 이루어지며 핵심 컴포넌트를

UID	Dicitionary Entry Name	CCT Used	Basic or Aggregate	Definition	Remarks	Object Class	Property Term	Representation Term	Business Terms	Core Component Children
000024	Address. Type Code	Type Code	Basic	The type of the address	For example a business address or a home address. Not the Role of the address	address	Type	Code		
000147	Base Charge Price Quantity	Quantity Type	Basic	The Base quantity of the charge/price unit amount	For example for a charge of $.5/day for 10 days, the charge base quantity is 1 day	Base Charge Price	Quantity	Quantity		

〈그림 7〉 핵심 컴포넌트 카탈로그의 예[2]

한정(Qualifying)하거나 정제(Refine)하여 BIE를 정의하는 메커니즘을 제공한다[2].

예를 들어 미국에 속해있는 컴퓨터 부품(DRAM) 도매업체의 '판매자 주소'를 표현하는 BIE는 'PostalAddress.Details'라는 ACC를 사용하며, 다음의 비즈니스 컨텍스트에 의해 정제될 수 있다.

Geopolitical = 'United States'
Industry Classification = 'Wholesalers'
Product Classification = 'DRAM'

4. 비즈니스 문서 저작 기술

이제까지 ebXML의 기술 구조와 핵심 컴포넌트 및 비즈니스 정보 엔티티 그리고 비즈니스 컨텍스트의 개념에 대하여 살펴보았다. 이번 장에서는 이러한 기본적인 지식을 바탕으로 ebXML에서 사용되는 비즈니스 문서를 저작하는 기술에 대하여 본격적으로 설명하기로 한다.

4.1 ebXML 비즈니스 문서 저작 절차

ebXML 비즈니스 문서의 저작은 앞서 〈그림 2〉에서 보듯이 비즈니스 프로세스를 모델링하는 디자인 단계(Design Time)에서 이루어진다. 정확하게 말해서 비즈니스 문서의 저작이란 핵심 컴포넌트를 조립하고 비즈니스 컨텍스트에 의해 정제하여, 실행 단계(Runtime)에서 비즈니스 프로세스를 수행하며 요구되는 메시지 인스턴스를 생성하기 위한 문서 모델을 만드는 것이다.

전체적인 비즈니스 문서 저작 과정을 요약 정리해 보면 다음과 같다[2][11].

① 비즈니스 프로세스 검색(BP Discovery) 원하는 거래를 정확히 표현하고 있는 비즈니스 프로세스가 있는지 레지스트리를 검색한다. 만일 정확히 일치하거나 유사한 것이 존재한다면 그것을 재사용

〈표 2〉 인증된 컨텍스트 카테고리

카테고리	설 명	비 고
Business Process	비즈니스 절차	Common BP 카탈로그 참조
Product Classification	제품 분류	UNSPSC 등
Industry Classification	산업 분류	NAICS 등
Geopolitical	지정학적 분류	ISO3166 등
Official Constraints	법적인 제약 사항	관련법 및 정부의 정책
Business Process Role	비즈니스 절차상의 역할	Common BP 카탈로그 참조
Supporting Role	비즈니스 파트너 이외의 역할	Third party
System Capabilities	비즈니스 시스템관련 제약 사항	S/W 및 H/W

하고 그렇지 않다면 새로운 비즈니스 프로세스를 모델링해야 한다.

② 컨텍스트 카테고리 결정 (Identifying Context Categories)
수행하려는 거래를 위한 BIE 및 비즈니스 문서를 생성하기 위하여 필요한 비즈니스 카테고리(<표 2> 참조)를 결정한다.

③ BIE 검색(BIE Discovery)
새로운 비즈니스 프로세스를 모델링하는 과정에서 거래에 필요한 비즈니스 문서를 모델링하게 된다. 이때 문서를 구성할 BIE를 레지스트리로부터 검색하는데 만일 정확히 일치하거나 유사한 것이 존재한다면 그것을 재사용하고 그렇지 않다면 새로운 BIE를 생성하고 레지스트리에 등록 요청한다.

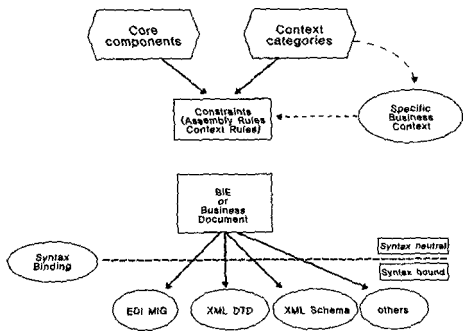
④ CC 검색(CC Discovery)
새로운 BIE를 생성하기 위하여 레지스트리로부터 핵심 컴포넌트를 검색한다. 만일 정확히 일치하거나 유사한 것이 존재한다면 그것을 재사용하고 그렇지 않다면 새로운 핵심 컴포넌트를 생성하고 레지스트리에 등록 요청한다

⑤ 문서 생성 규칙 작성
필요한 BIE 및 CC를 검색했다면 비즈니스 문서를 생성하기 위하여 필요한 조립 규칙(Assembly Rules)과 컨텍스트 규칙(Context Rules)을 작성한다. 조립 규칙과 문서 생성 규칙은 제약 언어(Constraint Language)를 이용하여 정의된다.

⑥ 비즈니스 문서 생성
비즈니스 컨텍스트와 CC 및 문서 생성 규칙을 기반으로 비즈니스 문서 모델을 생성한다.

4.2 컨텍스트 구동방식의 문서 생성

컨텍스트 메커니즘을 이용하여 비즈니스 문서를 생성하는 방식을 컨텍스트 구동(Context-driven) 문서 생성 기법[11]이라 한다. 이러한 메커니즘에 있어서 <그림 8>에서 보여주는 것처럼 핵심 컴포넌트와 비즈니스 컨텍스트 외에도 조립 규칙 및 컨텍스트 규칙과 같은 문서 생성 규칙이 필요하며, 다음에 살펴볼 제약 언어에 의하여 정의된다. 문서 생성 규칙에 의해 완성된 비즈니스 문서는 문법 중립적인(Syntax Neutral) 형태로 출력되며, 최종적으로 문법 연결(Syntax Binding) 과정을 통하여 다양한 형태의 문법으로 변환된다.



<그림 8> 컨텍스트 구동 메커니즘[2]

4.3 컨텍스트 제약 언어

컨텍스트 제약 언어[2][10](Context Constraints Language)는 특정 비즈니스 컨텍스트하에서 핵심 컴포넌트에 어떻게 비즈니스적 의미가 적용되는가를 표현하며 다음과 같은 두 가지 기능적인 규칙을 정의하는데 이용된다.

■ 조립 규칙(Assembly Rules)

조립 규칙[2][10]은 각각의 핵심 컴포넌트나 BIE를 ABIE 또는 비즈니스 문서로 조립(Assemble)하기 위한 규칙을 정의한다. 문서 조립에 요구되는 핵심 컴포넌트는 조립 규칙 내에서 참조되는 UUID를 통해서 레지스트리로부터 검색되어 사용된다.

■ 컨텍스트 규칙(Context Rules)

컨텍스트 규칙[2][10]은 조립 규칙에 의하여 조립된 결과를 비즈니스 컨텍스트에 의해 정제(Refine)한다. 정제 과정은 핵심 컴포넌트에 새로운 의미(Additional Semantics)를 부여하거나 기존의 의미를 확장(Extension) 또는 제한(Restriction)하는 것으로 이루어진다.

이와 같이 조립 과정과 정제 과정을 분리하는 것은 문서 조립 도구 구현상의 편의성을 배려한 것이기도 하지만, 표준 조립(Standard Assembly)을 생성하고 사용자들로 하여금 각자의 비즈니스에 적합한 정제를 수행하도록 하는데 목적이 있다.

컨텍스트 제약 언어는 XML 문법에 의해 표현되며 XML DTD[10]로 정의된다. 따라서 조립 규칙과 컨텍스트 규칙은 모두 XML 인스턴스로서 생성된다. 〈표 3〉은 컨텍스트 제약 언어를 요약 정리한 것이다.

4.4 문법 연결

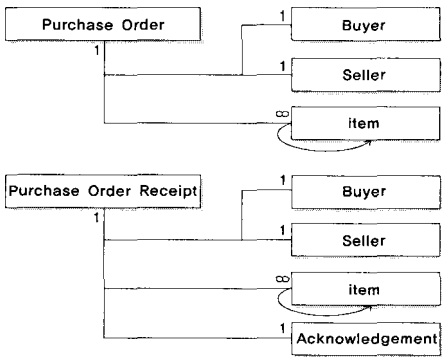
문법 연결 과정(Syntax Binding)은 특정 문법에 종속적이지 않은 형태로 생성된 문서를 e비즈니스 응용 시스템에서 필요로 하는 특정 문법(XML 스키마, XML DTD 또는 EDI Message Implementation Guide 등)으로 변환하는 작업을 의미한다[2].

4.5 비즈니스 문서 저작의 예

이번 절에서는 지금까지 알아본 내용을 기반으로 ebXML 비즈니스 문서 모델을 생성하는 구체적인 예를 살펴보기로 한다.

예제에서 생성하려는 비즈니스 문서는 '구매요구서(Purchase Order)'와 '구매요구응답서(Purchase Order Receipt)'이다.

설명의 편의를 위하여 구매요구서 및 구매요구응답서의 모델은 〈그림 9〉와 같은 단순



〈그림 9〉 예제 문서 구조

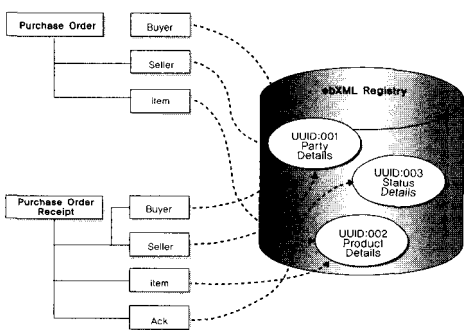
〈표 3〉 컨텍스트 제약 언어

Construct	Component Construct	Description
Assembly	Assemble @version	조립 규칙의 최상위 construct
Assembly	CreateGroup CreateBIE @name	CC로부터 조립될 문서나 BIE(하나의 Assemble은 하나의 BIE 또는 문서임)
CreateGroup	CreateGroup CreateBIE UseBIE Annotation @type	BIE의 그룹(ABIE)을 생성 @type은 'choice' 또는 'sequence'를 지정
CreateBIE	Type Name MinOccurs MaxOccurs Rename Condition Annotation @location	CC로부터 BIE를 생성 Name은 BIE의 명칭, Type은 BIE의 클래스, MinOccurs와 MaxOccurs는 cardinality를 지정 @location은 참조할 CC의 UUID를 지정
Name		생성될 BIE의 명칭
Type		생성될 BIE의 클래스
Rename	@from @to	해당 BIE에 포함된 child의 명칭 변경(@from a @to)
ContextRules	Rule @version	컨텍스트 규칙의 최상위 construct
Rule	Condition Taxonomy @apply @order	@apply에 지정된 조립 결과 노드(XPath)에 적용될 정체 규칙 @order에 적용 순서 명시
Taxonomy	@ref @context	계층 구조를 갖는 분류 체계(Classification Scheme)
Condition	Action Condition Occurs @test	@test에 지정된 boolean 표현식(XPath)이 true면 Action을 수행
Action	Add Subtract Occurs Condition Comment Rename @applyTo	Rule이 적용되는 방법을 지정 Rule은 @applyTo에 지정된 조립 결과의 노드(XPath)에 적용됨
Add	MinOccurs MaxOccurs CreateGroup BIE Attribute Annotation @before @after	새로운 BIE나 Attribute를 추가함 @before 어떤 노드 앞에 추가할 것인지 지정(XPath) @after 어떤 노드 뒤에 추가할 것인지 지정(XPath)
Subtract	BIE Attribute	기존의 BIE나 Attribute를 제거함
Occurs	BIE MinOccurs MaxOccurs	BIE나 Attribute의 cardinality를 지정
BIE	Name Type Attribute Annotation	BIE를 정의 Attribute는 BIE에 포함될 attribute 목록
Attribute	Name Type Use Value	Attribute를 정의 Use는 'required', 'optional', 'fixed' 또는 'default'Value는 fixed나 default 값

UseBIE	CreateGroup Use BIE Annotation @name	참조된 CC로부터 재사용할 child 노드를 지정 @name은 재사용할 BIE의 명칭
Comment		Action 주석문
MinOccurs		Cardinality를 지정
MaxOccurs		Cardinality를 지정
Annotation	Documentation Appinfo	BIE에 주석 첨가
Documentation		문서화 정보
Appinfo		응용 명세 정보

한 구조를 갖는다고 가정한다(각 문서를 구성하는 Buyer, Seller, Item, Acknowledgement 등의 항목들은 각각 특정한 비즈니스 의미를 갖는 BIE임).

우선 구매요구서는 구매자(Buyer), 판매자(Seller) 및 상품(Item) 정보로 구성되며, 구매자 정보와 판매자 정보는 구매요구서당 한 번씩 나오는 정보이며 그룹화되어 표현될 수 있다. 또한 구매하고자 하는 상품은 여러 가지가 있을 수 있으므로 상품 정보는 문서상에 여러 번 나타날 수 있다. 구매요구응답서는 구매자, 판매자, 상품 정보가 포함되는 점에서 기본적으로 구매요구서와 동일하나 구매요구에 대한 하나의 응답(Acknowledgement) 정보가 포함된다는 것이 다르다.



〈그림 10〉 예제 문서와 핵심 컴포넌트

상기의 문서 구조는 실세계에서 일반적으로 사용하는 구매 문서 양식을 참조하여 모델링한 것이다. 이것을 ebXML 전자 거래에서 사용하는 비즈니스 문서화 하기 위해서는 레지스트리에 접근해서 각 항목에 부합하는 핵심 컴포넌트를 검색하고 발견된 핵심 컴포넌트를 참고하여(UUID를 이용) 각 항목에 대한 기본적인 조립 규칙을 작성해야 한다(〈그림 10〉 참조).

〈그림 11〉은 핵심 컴포넌트를 기반으로 기본적인 조립 규칙을 XML 문서로 작성한 것이다(조립 규칙 문서 작성시에는 〈표 3〉에 나열되어 있는 제약 언어를 사용함).

조립 규칙 문서의 예에서 보듯이 각각의 CreateBIE 항목은 사용하고자 하는 핵심 컴포넌트의 UUID(편의상 24bit의 숫자로 표기)를 참조하고 있다. 실제로 문서 조립 도구를 이용하여 문서 조립 시에 이 UUID를 기반으로 레지스트리를 검색해서 비즈니스 문서 모델 생성에 사용되는 핵심 컴포넌트를 참조하게 된다. 〈그림 12〉는 집계 핵심 컴포넌트(ACC)인 'Party.Details'를 XML 스키마로 정의한 내용의 일부이다(ACC인 Party.Details

```
<?xml version="1.0"?>
<!DOCTYPE Assembly SYSTEM "assembly.dtd">
<Assembly version="1.8">
  <Assemble name="PurchaseOrder">
    <CreateGroup>
      <CreateBIE location="urn:uuid:001" id="Buyer">
        <Name>Buyer</Name>
        <Type>Party.Details</Type>
        <CreateGroup>
          <UseBIE name="Party.Identification.Details"/>
          <UseBIE name="Party.PostalAddress.Details">
            <CreateGroup>
              <CreateGroup type="choice">
                <UseBIE
                  name="PostalAddress.BuildingName.Text"/>
                <UseBIE
                  name="PostalAddress.Building.Identifier"/>
              </CreateGroup>
              <UseBIE name="PostalAddress.StreetName.Text"/>
              <UseBIE name="PostalAddress.Region.Identifier"/>
              <UseBIE name="PostalAddress.State.Identifier"/>
              <UseBIE name="PostalAddress.Country.Identifier"/>
            </CreateGroup>
          </UseBIE>
        </CreateGroup>
      </CreateBIE>
      <CreateBIE location="urn:uuid:001" id="Seller">
        <Name>Seller</Name>
        <Type>Party.Details</Type>
      </CreateBIE>
    </CreateGroup>
    <CreateBIE location="urn:uuid:002" id="Item">
      <Name>Item</Name>
      <Type>Product Details</Type>
      <MinOccurs>1</MinOccurs>
      <MaxOccurs>unbounded</MaxOccurs>
    </CreateBIE>
  </Assemble>
  <Assemble name="PurchaseOrderReceipt">
    <CreateGroup>
      <CreateBIE idref="Seller"></CreateBIE>
      <CreateBIE idref="Buyer"></CreateBIE>
    </CreateGroup>
    <CreateBIE idref="Item"></CreateBIE>
    <CreateBIE location="urn:uuid:003">
      <Name>Acknowledgment</Name>
      <Type>Status.Details</Type>
    </CreateBIE>
  </Assemble>
</Assembly>
```

〈그림 11〉 조립 규칙 문서

는 컴포넌트 내부에 또다른 BCC 또는 ACC 들을 포함하고 있음). 참고로 핵심 컴포넌트 명칭에서 ACC는 접미사로 '.Details'를 CCT는 '.Type'을 사용한다[2].

〈그림 11〉의 조립 규칙에서 Buyer 항목은 Party.Details를 참조하고 있으나 Party.Details에 포함된 모든 CC를 사용하지는 않는다. UseBIE 구문을 사용해서 구체적으로 사용할 컴포넌트 만을 명시하고 있다. 또한 Buyer 항

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:uuid:001">
  <complexType name="Party.Details">
    <annotation>
      <documentation xml:lang="en">
        Details on an individual, a group or a body having a role
        in a business function
      </documentation>
    </annotation>
    <sequence>
      <element name="Party.Type.Details"
               type="Identification.Details" minOccurs="0">
      </element>
      <element name="Party.Identification.Details"
               type="Party.Identification.Details"
               maxOccurs="unbounded">
      </element>
      <element name="Party.Title.Details"
               type="Party.Title.Details"
               maxOccurs="unbounded">
      </element>
      <element name="Party.PostalAddress.Details"
               type="PostalAddress.Details"
               maxOccurs="unbounded">
      </element>
      <element name="Party.Location.Details"
               type="Location.Details"
               maxOccurs="unbounded">
      </element>
      <element name="Party.Communication.Details"
               type="Communication.Details"
               maxOccurs="unbounded">
      </element>

          ...

    </sequence>
  </complexType>
</schema>
```

〈그림 12〉 핵심 컴포넌트 Party.Details

목 생성 부분의 두번째 UseBIE 구문의 내부를 보면 더 세부적으로 Party.PostalAddress.Details 컴포넌트 내에서 구체적으로 사용할 컴포넌트만을 명시적으로 나열하고 있다.

조립 규칙을 작성한 후에는 조립된 문서를 특정 비즈니스 환경에 대하여 정제하기 위하여 비즈니스 컨텍스트를 결정하고 컨텍스트 규칙을 작성한다(컨텍스트 규칙 문서도 〈표 3〉에 나열되어 있는 제약 언어를 사용함). 편의상 본 예제에서 사용하는 비즈니스 컨텍스트는 'Geopolitical = US'로 국한한다.

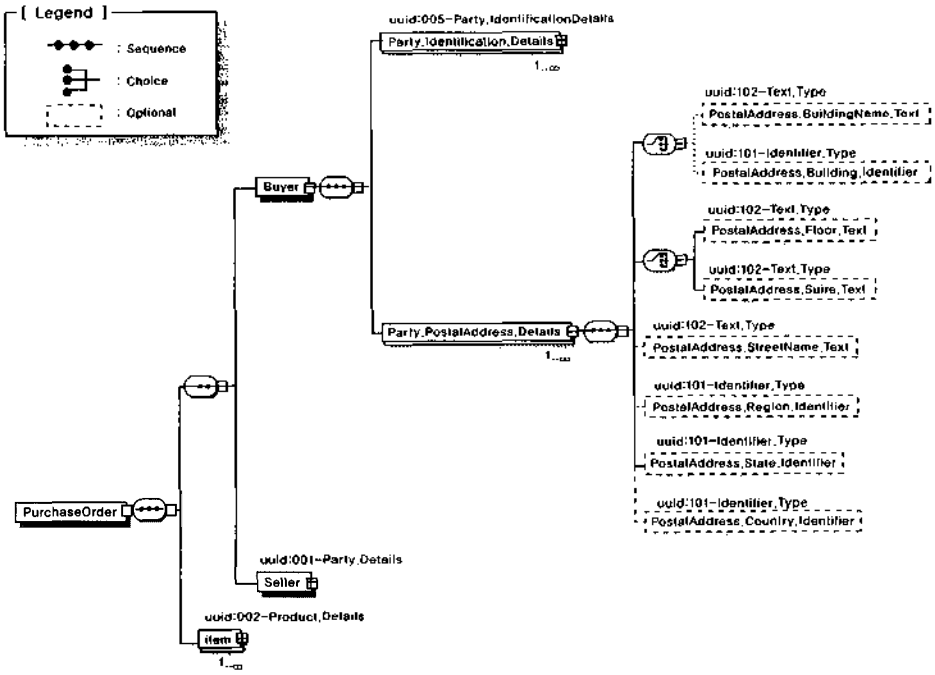
〈그림 13〉은 구매요구서를 위한 컨텍스트 규칙 문서이다. 본 컨텍스트 규칙은 단일 규칙 구문(Rule Construct)을 포함하고 있으나

```
<?xml version="1.0"?>
<!DOCTYPE ContextRules SYSTEM "contextrules.dtd">
<ContextRules id="USWholesalers">
  <Rule apply="hierarchical">
    <Taxonomy context="Geopolitical"
     ref="ebxml/classification/iso3166ClassificationScheme.xml"/>
    <Condition test="$Geopolitical='US'">
      <Action applyTo="//Buyer/Party.PostalAddress.Details">
        <Occurs>
          <BIE>
            <Name>PostalAddress.State.Identifier</Name>
          </BIE>
        </Occurs>
        <Add before="PostalAddress.StreetName.Text">
          <CreateGroup type="choice">
            <BIE>
              <Name>PostalAddress.Floor.Text</Name>
              <Type>Text.Type</Type>
            </BIE>
            <BIE>
              <Name>PostalAddress.Suite.Text</Name>
              <Type>Text.Type</Type>
            </BIE>
          </CreateGroup>
        </Add>
      </Condition>
    </Rule>
</ContextRules>
```

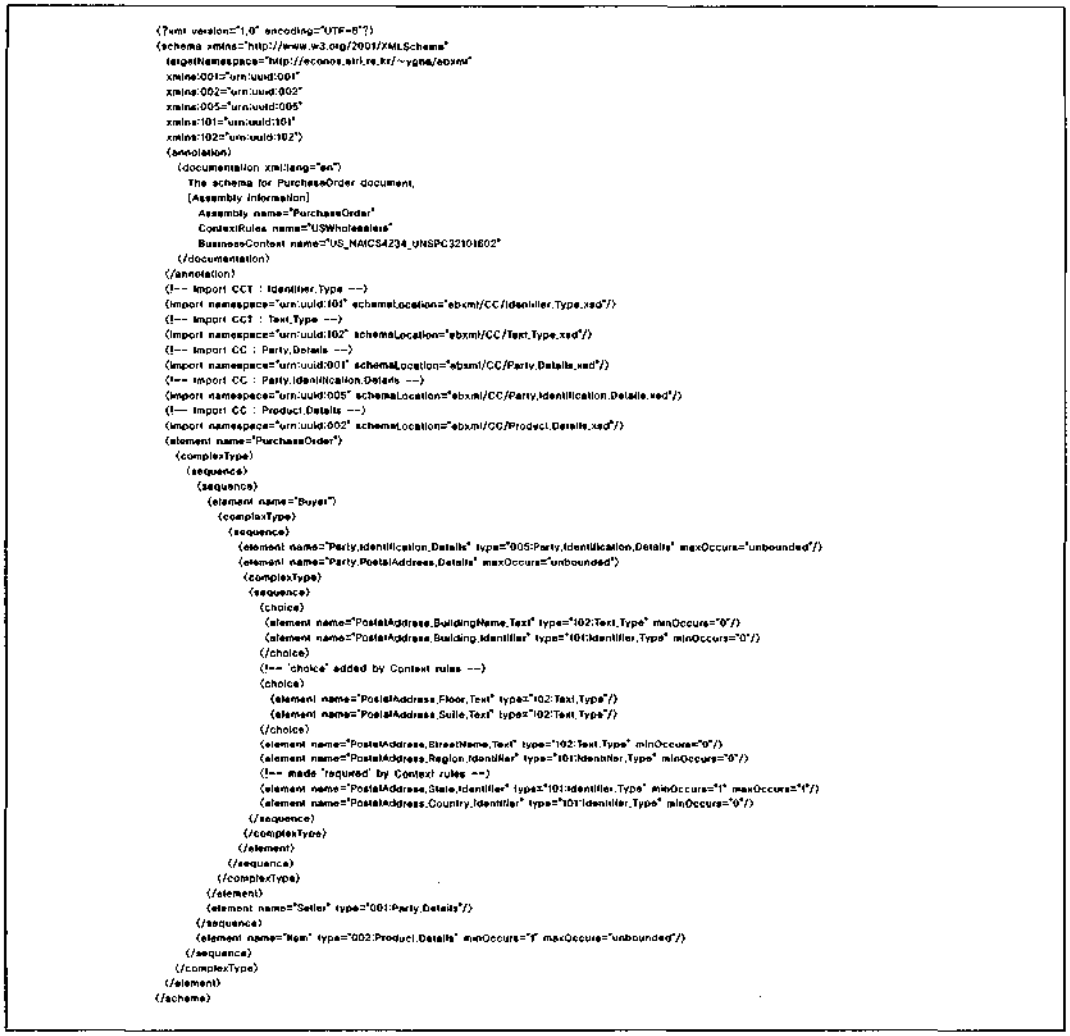
〈그림 13〉 컨텍스트 규칙 문서

여러 개의 규칙 구문이 포함되어 있는 경우에는 기본적으로 정의된 순서를 따르게 된다. 만일 적용 순서(@order Attribute)가 구체적으로 명시되어 있다면 명시된 순서를 반영한다.

예제의 컨텍스트 규칙은 비즈니스 컨텍스트의 Geopolitical 카테고리의 값이 'US'인 경우에 Action 구문에 명시된 내용을 수행하라는 의미이다. 예제에서 Action 구문은 조립 결과로 생성된 문서 모델에서 Buyer 항목에 포함된 PostalAddress 부분에 적용되는데, PostalAddress는 PostalAddress.Details 라는 CC를 참조하고 있음을 알 수 있다(〈그림 12〉 참조). Action 구문의 내용은 PostalAddress의 State 항목을 선택 항목에서 필수 항목으로



〈그림 14〉 완성된 구매요구서 모델(다이어그램)



```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://econos.snu.ac.kr/~ygha/ebxml"
  xmlns:001="urn:uuid:001"
  xmlns:002="urn:uuid:002"
  xmlns:005="urn:uuid:005"
  xmlns:101="urn:uuid:101"
  xmlns:102="urn:uuid:102">
  <annotation>
    <documentation xml:lang="en">
      The schema for PurchaseOrder document.
      [Assembly information]
        Assembly name="PurchaseOrder"
        ContextRules name="USWholesales"
        BusinessContext name="US_NAICS4Z34_UNSPC32101602"
    </documentation>
  </annotation>
  <!-- import CCT : Identifier.Type -->
  <import namespace="urn:uuid:101" schemaLocation="ebxml/CC/Identifier.Type.xsd"/>
  <!-- import CCT : Text.Type -->
  <import namespace="urn:uuid:102" schemaLocation="ebxml/CC/Text.Type.xsd"/>
  <!-- import CC : Party.Details -->
  <import namespace="urn:uuid:001" schemaLocation="ebxml/CC/Party.Details.xsd"/>
  <!-- import CC : Party.IdentificationDetails -->
  <import namespace="urn:uuid:005" schemaLocation="ebxml/CC/Party.Identification.Details.xsd"/>
  <!-- import CC : Product.Details -->
  <import namespace="urn:uuid:002" schemaLocation="ebxml/CC/Product.Details.xsd"/>
  <element name="PurchaseOrder">
    <complexType>
      <sequence>
        <sequence>
          <element name="Buyer">
            <complexType>
              <sequence>
                <element name="Party.Identification.Details" type="005:Party.Identification.Details" maxOccurs="unbounded"/>
                <element name="Party.PostalAddress.Details" maxOccurs="unbounded">
                 <complexType>
                  <sequence>
                    <choice>
                     <element name="PostalAddress.BuildingName.Text" type="102:Text.Type" minOccurs="0"/>
                     <element name="PostalAddress.Building.Identifier" type="101:Identifier.Type" minOccurs="0"/>
                    </choice>
                    <!-- 'choice' added by Context rules -->
                    <choice>
                      <element name="PostalAddress.Floor.Text" type="102:Text.Type"/>
                      <element name="PostalAddress.Suite.Text" type="102:Text.Type"/>
                    </choice>
                    <element name="PostalAddress.StreetName.Text" type="102:Text.Type" minOccurs="0"/>
                    <element name="PostalAddress.Region.Identifier" type="101:Identifier.Type" minOccurs="0"/>
                    <!-- made 'required' by Context rules -->
                    <element name="PostalAddress.State.Identifier" type="101:Identifier.Type" minOccurs="1" maxOccurs="1"/>
                    <element name="PostalAddress.Country.Identifier" type="101:Identifier.Type" minOccurs="0"/>
                  </sequence>
                 </complexType>
                </element>
              </sequence>
            </complexType>
          </element>
          <element name="Seller" type="001:Party.Details"/>
        </sequence>
        <element name="Item" type="002:Product.Details" minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>
</schema>
```

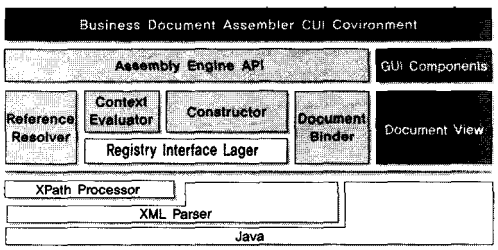
〈그림 15〉 구매요구서 모델(XML 스키마)

변경하라는 것과 StreetName 항목 앞에 Floor 또는 Suite 항목 선택 사항을 삽입하라는 의미이다(컨텍스트 규칙에서 사용하는 모든 Condition 조건 표현식과 문서 항목에 대한 참조는 XPath[13] 표현식을 이용하여 표기됨). 〈그림 14〉는 조립 결과에 대하여 컨텍스트 규칙을 적용하여 정제한 후 완성된 구매요구서 모델을 다이어그램으로 도식화한 것으로서 문서의 각 항목을 표현하는 BIE와 그것이 참조하는 CC를 나타낸다. 〈그림 15〉는 최종적으로 XML 스키마로 문법 연결을 수행한 결과를 보여준다.

5. ebXML 문서 생성기의 구현

5.1 문서 생성기의 구조

본 논문에서 구현한 문서 생성기의 구조는 〈그림 16〉과 같이 크게 문서 조립 엔진(Assembly Engine) 모듈과 사용자 인터페이스(User Interface) 모듈로 나뉘어 진다. 문서 조립 엔진 모듈은 다시 문서 구성기(Constructor), 컨텍스트 연산기(Context Evaluator), 문법 연결기(Document Binder) 및 엔진 API로 구성되며, 사용자 인터페이스 모듈은 문서 표시기(Document View), 기본 GUI 컴포넌트 및 GUI 기반 통합 사용자 환경으로 이루어 진다. 〈그림 17〉은 본 논문에서 구현한 문서 생성기의 통합 사용자 환경의 실행 모습을 보여준다.



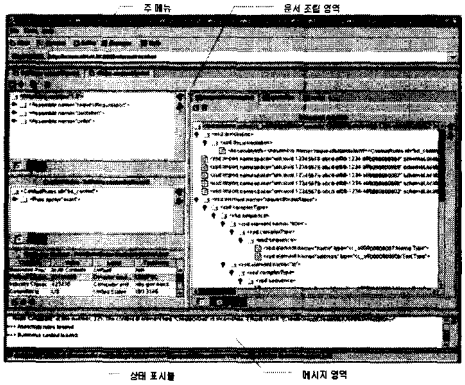
〈그림 16〉 문서 생성기의 구조

5.2 문서 조립 엔진

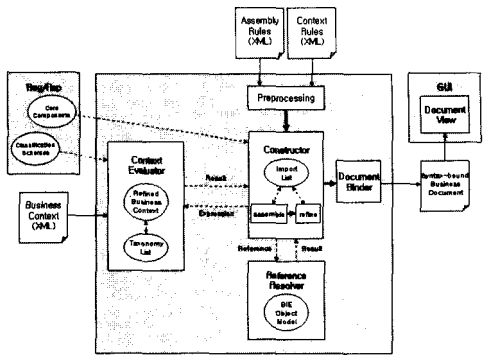
문서 조립 엔진은 ebXML 문서 생성기의 핵심 구성 요소로서 그 세부 구조는 〈그림 18〉에서 보는 바와 같다. 문서 조립 엔진은 문서 조립 규칙, 컨텍스트 규칙 및 비즈니스 컨텍스트를 입력으로 받아 레지스트리를 검색해서 필요한 핵심 컴포넌트를 참조하여 이를 기반으로 자동으로 비즈니스 문서를 생성해 주는 역할을 수행한다.

입력된 조립 규칙 및 컨텍스트 규칙은 우선 각각 전처리(Preprocessing) 과정을 통해서 내부 규칙간의 상호 참조 및 수행시 우선 순위를 결정한다. 문서 구성기는 전처리된 조립 규칙을 해석하고 레지스트리로부터 필요한 핵심 컴포넌트 검색하여 이를 기반으로 비즈니스 문서의 구조를 생성한다. 또한 컨텍스트 규칙을 해석하여 지정된 비즈니스 컨텍스트에 맞도록 생성된 문서의 구조를 변경(새로운 Element나 Attribute를 추가 또는 삭제)하거나 특정 문서의 항목의 Cardinality 변경 또는 새로운 명칭을 부여하는 등의 작업을 수행하게 된다. 문서 조립 규칙 또는 컨텍스트 규칙 내부에서 문서 항목을 구성하는 핵심 컴포넌트는 해당 컴포넌트의 고유 명칭, 각각에 부여된 UUID[21] 또는 URL(Universal Resource Location)을 통해 참조된다.

문서를 구성하는 각각의 항목들은 문서 생성 규칙들 내부에서 고유한 경로를 가지며 이는 XPath[13] 표현식을 통해 표현된다. 그러나 이러한 문서 생성 규칙내의 논리적인 경로와 실제 생성되는 문서 스키마에 나타나는 Element의 물리적인 경로는 서로 상이하며, 이들간의 매핑이 필요하게 되는데 참조처리기는 이러한 논리적 경로와 물리적 경로간의 변환 기능을 제공하는 모듈이다. 예를 들어 4.5의 예제 문서인 구매요구서(PurchaseOrder) 생성 규칙내의 구매자(Buyer) 항목을 나타내는 XPath 표현식은 "/PurchaseOrder/Buyer"



〈그림 17〉 문서 생성기의 사용자 환경



〈그림 18〉 문서 조립 엔진

로 나타낼 수 있다. 그러나 이 항목이 실제로 문서 스키마로 생성될 때 다음과 같은 XPath 표현식으로 나타나게 된다

/schema/element[@name = "PurchaseOrder"]/complexType/sequence/element[@name = "Buyer"]

문서 생성 규칙내의 컨텍스트 조건문 또한 XPath 표현식을 이용하여 표현한다. 예를 들어 어떤 비즈니스 문서를 위한 컨텍스트 규칙내에 "거래 국가가 한국 또는 일본이고, 구매하려는 물건이 16M DRAM인 경우 'Samsung' 또는 'NEC' 중에 선택할 수 있는 구매 업체 항목을 추가"하라는 의미의 규칙이 있다고 하면 다음과 같은 컨텍스트 조건절을 갖게 된다.

〈Condition test = "($Geopolitical = 'KR' or $Geopolitical = 'JP') and $Product Classification = '16M DRAM'"〉

컨텍스트 연산기는 이러한 컨텍스트 조건절을 연산하여 조건이 만족하는지 여부를 결정하는 모듈로서 그 연산 결과에 따라 문서 조립 엔진은 조건문 내부에 정의된 수행문(Action statement)을 처리할지 여부를 판단하게 된다. 문서 문법 연결기는 문서 구성기가 문서 생성 규칙들을 통해 생성한 문서 객체 모델(Document Object Model) 형태의 최종 결과물을 사용자가 지정한 특정 문법(XML 스키마, DTD, EDI Message Implementation Guide 등)을 갖는 문서 파일 형태로 출력해 주는 역할을 수행한다.

6. 결 론

본 논문에서는 ebXML 프레임워크 및 이를 기반으로 글로벌 e비즈니스를 수행하기 위한 ebXML 핵심 컴포넌트 기반의 비즈니스 문서 생성 기술에 대하여 살펴보고, 이를 기반으로 구현된 ebXML 문서 생성 시스템의 구조에 대하여 설명하였다.

현재 기업간의 비즈니스 환경은 이미 인터넷을 떼어 놓고는 생각할 수 없게 되었으며, 또한 기업 내의 정보 시스템을 아무리 잘 갖추어놓는다 할지라도 다른 기업과의 협력과 거래를 원활히 할 수 없다면 국제 경쟁에서 뒤처질 수 밖에 없는 상황이 되었다. 이러한 사실은 차세대 e비즈니스 모델로서 주목 받고 있는 c커머스(Collaborative Commerce)의 개념과도 그 맥을 같이하는 것으로서 앞으로의 e비즈니스 시스템 구축에 있어서 거스를 수 없는 추세라 하겠다.

이에 ebXML은 e비즈니스를 위한 글로벌 표준을 통해서 그 동안 기업간의 협력(B2B)에서 가장 큰 장애물로 여겨졌던 정보 시스템의 이질성을 극복할 수 있는 대안을 제시하는 기술로 기대되고 있으며, 향후 몇 년 이내에 글로벌 e비즈니스 표준으로 자리매김 할 것으로 예측되고 있다.

이미 우리나라에서도 한국전자거래진흥원을 중심으로 산업계 및 학계의 전문 위원들이 ebXML 표준화에 각 부문별로 활동하고 있으며 한국전자통신연구원을 중심으로 관련 기술 개발이 이루어지고 있다. 앞으로는 POC 참여 등과 같은 보다 적극적인 활동과 본 논문에서 제시한 것과 같은 ebXML 표준 구현 기술의 보급을 통해 ebXML 기반의 비즈니스 시스템을 활성화 한다면 우리나라가 차세대 글로벌 e비즈니스 기술 시장을 선도해 나갈 수 있을 것으로 생각된다.

참 고 문 헌

[1] eBTWG Business Process Project Team, "Business Process Specification Schema Version 1.01," UN/CEFACT, May. 2001.

[2] eBTWG Core Components Project Team, "Core Components Technical Specification Part I Version 1.8," UN/CEFACT, Feb. 2002.

[3] ebXML Collaboration Protocol Profile and Agreement Technical Committee, "Collaboration-Protocol Profile and Agreement Version 2.0," OASIS, Jun. 2002.

[4] ebXML Technical Architecture Project Team, "ebXML Technical Architecture Specification Version 1.04," OASIS, Feb. 2001.

[5] ebXML Registry and Repository Technical Committee, "Registry Information Model Version 2.0," Dec. 2001.

[6] ebXML Registry and Repository Technical Committee,"Registry Services Specification Version 2.0," Dec. 2001.

[7] ebXML Messaging Services Technical Committee, "Message Services Specification Version 2.0," Feb. 2002.

[8] ebXML Implementation, Interoperability and Conformance Technical Committee, "Message Service Implementation Guidelines Version 0.1 and ebXML Test Framework Version 0.2," 2002.

[9] ebXML Core Components Team, "Core

Components Structure Version 1.04," UN/CEFACT and OASIS, May. 2001.

[10] ebXML Core Components Team, "Document Assembly and Context Rules Version 1.04," UN/CEFACT and OASIS, May. 2001.

[11] Chappell, David A. et al., "Professional ebXML Foundations," Wrox Press, 2001.

[12] Open Group, "Universal Unique Identifier,"http://www.opengroup.org/onlinepubs/009629399/apdxa.htm

[13] Clark, James and DeRose, Steve, "XML Path Language Version 1.0," W3C, http://www.w3c.org/TR/xpath

[14] Fallside, David C. et al., "XML Schema," W3C, http://www.w3c.org/XML/Schema

[15] XML Protocol Working Group, "SOAP Version 1.2," W3C,http://www.w3c.org/2000/xp/Group/

[16] UML, "UML Home," www.uml.org

[17] Sun Microsystems, "Java Technology and XML,"http://java.sun.com/xml/index.html

[18] Microsoft, "Biztalk Server Technology," http://www.microsoft.com/technet/prodtechnol/biztalk/Default.asp

[19] Fujitsu, " Interstage V5," http://interstage.fujitsu.com/en/products/index.html

[20] RosettaNet, "RosettaNet Home," www.rosettanet.org

[21] UDDI, "UDDI Home," www.uddi.org

[22]한국전자거래진흥원, "ebXML 중앙등록저장소," http://www.ebxml.or.kr

저 자 소 개

하영국 (E-mail : ygha@etri.re.kr)
1993. 건국대학교 전산학과(학사)
1995. 건국대학교 대학원 전산학과(석사)
1995 ~ 1998. 한국전자통신연구원 부설 시스템공학연구소 연구원
1998 ~ 현재 한국전자통신연구원 비즈니스지식처리연구팀 선임연구원
관심 분야 : 전자상거래, 정보보호, 컴퓨터통신

손주찬 (E-mail : jcsohn@etri.re.kr)
1983 ~ 1988. 한국외국어대학교 서반아어과(학사)
1988 ~ 1990. 한국외국어대학교 경영정보학과(석사)
1990 ~ 1991. ㈜에스콰이어 전산실
1991 ~ 1998. 한국전자통신연구원 부설 시스템공학연구소
1998 ~ 현재 한국전자통신연구원 비즈니스지식처리연구팀장
관심 분야 : 전자상거래, 웹서비스, 시맨틱웹, 지능형 비즈니스

손덕주 (E-mail : djson@etri.re.kr)
1976. 서울대학교 수학교육과(학사)
1978. 한국과학기술원 전산학과(석사)
1978.3 ~ 현재 한국전자통신연구원 인터넷컴퓨팅연구부 책임연구원
관심 분야 : 이동컴퓨팅, 분산시스템소프트웨어, e-Business미들웨어, 웹 서비스 기술