

단일 글로벌 전자상거래 시장을 만들기 위한 컴포넌트와 컨텍스트 기반의 전자문서 설계 및 구현

김 완 평*

Design and Implementation of a Component and Context based Business Message for Creating a Single Global Electronic Market

Wan-Pyong Kim*

Abstract

ebXML (Electronic Business using eXtensible Markup Language), sponsored by UN/CEFACT and OASIS, is a modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet. Using ebXML, companies now have a standard method to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes. It is the mission of ebXML that provides an open XML-based infrastructure enabling the global use of electronic business information in an interoperable, secure and consistent manner by all parties for creating a single global electronic market.

This paper briefly overviews the concept of core component, context, assembly rule and context rule. Then, it designs by standard specifications of ebXML core component commonly used in an industry and among industries, and assembles business messages by using XML schema. Therefore, it suggests the mechanism which effectively exchanges business messages among the trading partners.

This paper designs core component by using only three business messages of retail industry : orders, dispatch report, sales report.

Keywords : XML/EDI, ebXML, Core Component, context, assembly rule, XML Schema

※ 본 연구는 한국전자거래진흥원의 ebXML 과제로 수행되었음.
* (주)데이콤/경영혁신추진단 기술개발팀 과장 (Wan100@chollian.net)

1. 서론

1.1 연구의 배경과 목적

업종간 공통으로 재사용 가능한, 표준 전자문서와 공통 비즈니스 프로세스 같은, 핵심 컴포넌트의 개발 활동은 전 세계적 단일 전자상거래 시장 형성이라는 ebXML비전을 현실화하는 작업으로 전자상거래 표준화를 통하여 업체별 중복 개발비용의 절감과 업종간 공통 비즈니스 프로세스 개발관리 및 이기종 이질 업무의 상호연동성을 제공할 수 있어서 업무 협업(business collaboration)을 꾀할 수 있다[14].

전자문서를 개발할 경우, 각 업종별로 유사한 비즈니스 프로세스에서 공통적으로 사용되는 전자문서가 도출된다. 이를 각 업종별로 또는 업종내의 각 업체별로 표준을 따르지 않고 각각 개발하여 사용한다면, 유사한 문서의 중복으로 인한 시스템간 연동의 어려움과 시스템간 연동을 위한 불필요한 비용의 소모가 발생한다. 본 연구에서는 이를 최소화하기 위하여, 차세대 전자상거래 표준 프레임워크(ebXML)의 적용을 통하여, 컴포넌트와 컨텍스트 기반의 전자문서를 설계하고 구현하는 개발 프로세스를 제안한다.

1.2 연구의 범위 및 방법

본 연구는 인터넷 기반의 B2B, G2B 전자상거래의 활성화가 본격적으로 이루어지는 현시점에서 전자상거래 시스템의 핵심을 이루고 있는 XML 전자문서를 재사용 할 수 있는 컴포넌트와 컨텍스트 기반의 전자문서를 설계하고 구현하는 개발 프로세스를 제시한다. 다시 말하면, 비즈니스 프로세스 상에서 발생하는 전자문서를 개발하고, 거래 상대방간에 주고받기 위해서 ebXML 프레임워크 내에서는 업종별, 또는 업종내의 각 업체별로 공통적으로 사용하는 요소들을 핵심 컴포넌트로 설계하고, 이를 전자문서 스키마로 조립하여

사용함으로써 운영자의 최소한의 간섭으로 전자문서를 효율적으로 주고받을 수 있는 메커니즘을 제공하고 있다.

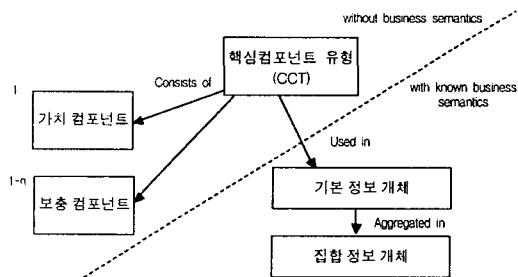
본 연구에서는 현재 유통부문에서 사용중인 EDI문서 중에서 주문서, 발송통지서, 판매현황 보고서 3종만을 대상으로 핵심 컴포넌트를 설계한다. 2001년 5월 ebXML 비엔나 회의에서 결정된 핵심 컴포넌트 관련 문서를 일부 인용하거나 참고하여 설계하였다[15-20].

2. 이론적 배경

2.1 핵심 컴포넌트 라이브러리의 구성요소

2.1.1 핵심 컴포넌트(Core Component)

핵심 컴포넌트 모든 비즈니스 메시지를 만들기 위해서 활용되는 하나의 의미적 빌딩 블록이다. 다시 말해, 특정 비즈니스 컨텍스트에 의존하지 않고 다양한 산업분야 및 환경에서 공통적으로 사용 가능한 빌딩 블록을 의미한다. 핵심 컴포넌트는 EDIFACT 문서를 만들기 위한 항목을 발췌해 놓은 EDIFACT MIG와 같은 역할을 하는 것으로 모든 문서, 모든 업종에서 공통으로 사용될 항목들을 발췌해놓은 것이다. 핵심 컴포넌트에는 핵심 컴포넌트 유형(Core Component Type), 기본 정보 개체(Basic Core Component), 집합 정보 개체(Aggregate Core Component) 등의 요소가 있다.



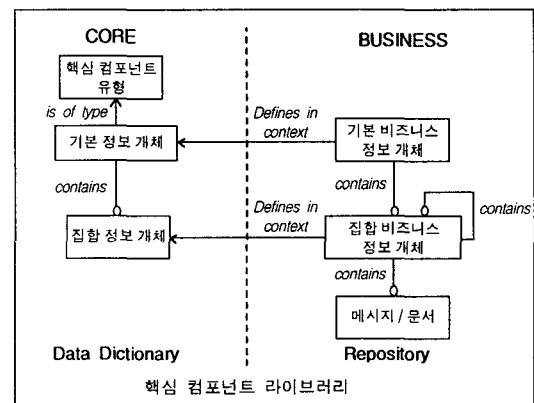
(그림 1) 핵심 컴포넌트

(그림 1)에서는 핵심 컴포넌트 유형과 핵심 컴포넌트 유형을 구성하고 있는 가치 컴포넌트와 보충 컴포넌트, 그리고 핵심 컴포넌트를 이용해서 만들어지는 정보 개체들의 관계를 보이고 있다. 위 그림에서 사선을 기준으로 위쪽은 비즈니스 의미를 포함하고 있지 않은 요소들이며, 아래쪽은 핵심 컴포넌트 유형을 이용하여 비즈니스 의미를 적용한 요소들을 나타내고 있다.

정보 개체는 크게 기본 정보 개체와 집합 정보 개체로 나눌 수 있으며, 정보 개체에 비즈니스 의미 부여 여부에 따라 기본 비즈니스 정보 개체 및 집합 비즈니스 정보 개체로 나눌 수 있으며, (그림 2)에서는 핵심 컴포넌트와 정보 개체, 그리고 비즈니스 의미를 부여한 비즈니스 정보 개체와의 관계를 설명하고 있다.

2.1.2 컨텍스트(Context)

특정한 비즈니스 상황 및 환경을 분류, 적용시키기 위한 메커니즘이다. 예를 들어, 하나의 회사가 자재를 구매할 경우 구매자로서, 생산품을 소비자에게 판매할 경우에는 판매자로서, 또는 다른 회사에 생산품을 납품할 경우 납품 자로서의 역할을 갖게되며 각각의 역할에 따라서 비즈니스 전자문서에서 표현하여야 할 항목이 달라지게 된다. 따라서, 핵심 컴포넌트를 이용한 전자문서 개발에서는 정의된 핵심 컴포넌트를 이용하여 비즈니스 정보 개체를 만들 때 컨텍스트를 적용하여 비즈니스 정보 개체의 이름을 정의하고, 사용할 항목을 결정하게 된다. 컨텍스트의 분류의 종류에는 지역, 산업, 비즈니스 프로세스, 제품, 역할, 서비스 수준, 비즈니스 목적, 업종별 시장 등이 있다.



(그림 2) 핵심 컴포넌트 라이브러리

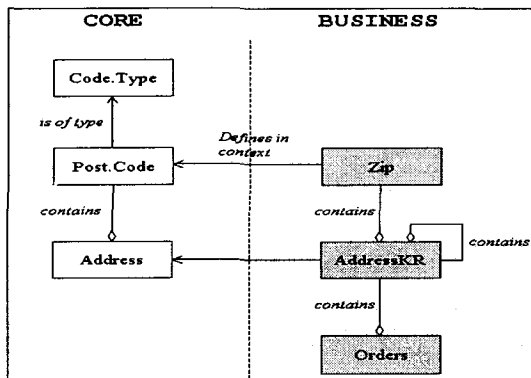
2.1.3 비즈니스 정보 개체

핵심 컴포넌트에 실제 비즈니스 의미가 적용된다는 것은 비즈니스 문서를 만들기 위해서 사용될 요소, 즉 비즈니스 정보 개체를 정의한다고 할 수 있다. 비즈니스 정보 개체란 하나의 유일한 비즈니스 의미적 정의를 갖는 비즈니스 데이터를 말한다. 이는 EDIFACT MIG의 NAD, DTM 등과 같은 항목을 의미한다고 할 수 있다. 단, 실제 비즈니스 문서를 작성할 경우 비즈니스 정보 개체와는 다른 이름으로 사용할 수 있다.

핵심 컴포넌트 라이브러리를 나타내고 있는 (그림 2)의 CORE 부분에는 핵심 컴포넌트 유형을 이용하여 정의한 기본 정보 개체와 이들의 집합체인 집합 정보 개체가 있으며, 이는 핵심 컴포넌트 사전에 해당하는 요소들이다. BUSINESS 부분에는 CORE 부분의 기본 정보 개체와 집합 정보 개체에 업종별, 지역별 등의 비즈니스 컨텍스트를 적용한 기본 비즈니스 정보 개체와 집합 정보 개체가 있으며, 이러한 요소들을 이용해서 주문서, 송장 등과 같은 비즈니스 전자문서를 생성하게 되는 것이며, 생성된 전자문서는 등록저장소에 저장된다.

핵심 컴포넌트 라이브러리의 구성요소를 실제 CORE 부분과 BUSINESS 부분의 예를 들어 실체화하여 보이면 (그림 3)과 같다. (그림 3)에서 왼쪽 Core 부분의 Code.Type, Post.Code, Address는 핵심 컴포넌트 유형, 기본 정보 개체

그리고 집합 정보 개체에 해당한다. Post.Code는 Code.Type이라는 핵심 컴포넌트 유형을 이용하여 우편번호를 정의하기 위한 기본 정보 개체이며, 이 Post.Code는 주소를 나타내기 위한 Address라는 집합 정보 개체의 구성 요소로 사용된다. 그리고 오른쪽 Business 부분의 Zip, AddressKR, Orders는 자신의 비즈니스 프로세스에 맞는 핵심 컴포넌트들을 참조하여 문서조립 규칙과 컨텍스트 규칙을 적용한 결과로 생성된 기본 비즈니스 정보 개체, 집합 비즈니스 정보 개체 그리고 전자문서에 해당된다. Post.Code 기본 정보 개체를 이 예제에서는 Zip이라는 이름으로 사용되며, Address라는 집합 정보 개체는 지역적 컨텍스트가 적용되어 Address 개체에서 정의하고 있는 여러 항목 중에서 한국의 주소를 표현하기 위해 필요한 항목만을 사용하여 AddressKR이라는 이름으로 주문서에서 사용된다.



(그림 3) 핵심 컴포넌트 라이브러리 실체화

기본 정보 개체와 집합 정보 개체를 EDI 환경에서 사용 중인 전자문서 설계 지침서인 MIG를 가지고 얘기하자면, 집합 정보 개체는 SG(Segment Group)이나 NAD, TAX, FTX 등 세그먼트 또는, 세그먼트내의 복합자료 항목(Composite Data Element)에 해당된다고 할 수 있고, 기본 정보 개체는 각 세그먼트 또는 복합자료 항목이 가지고 있는 단순자료 항목(Component Data

Element)에 해당된다고 할 수 있다.

2.2 핵심 컴포넌트 유형

2.2.1 핵심 컴포넌트의 표현 유형

핵심 컴포넌트 유형은 그 자체로는 비즈니스 의미를 가지고 있지 않은 핵심 컴포넌트들이다. 이 핵심 컴포넌트 유형들이 비즈니스 컨텍스트에서 재사용될 경우 그것들은 기본 정보 개체들이 된다. (그림 3)을 보면, 코드(Code.Type)는 그 자체로는 어떠한 비즈니스적인 의미를 가지고 있지 않는 요소이다.

<표 1>은 핵심 컴포넌트의 표현 유형과 각각의 표현 유형에 대하여 설명하고 있다.

<표 1> 핵심 컴포넌트의 표현 유형

표현 유형	설 명
Amount	통화에 지정된 통화단위의 개수
Code	간결함 또는 언어 독립성을 위한 문자열로 특성을 지닌 정의된 값이나 텍스트를 표현하거나 대치한다
Date	특정한 연월이 포함된 날짜
DateAndTime	시간진행에서의 특정한 지점
Identifier	같은 스키마 내에서 한 객체 인스턴스를 식별하고 고유하게 구별하기 위한 문자열
Indicator	On/Off, True/False와 같은 상태 지시 값
Measure	측정 단위를 명시한다
Name	사람, 장소, 물건, 개념 등을 확정적으로 명명하는 단어나 구절
Quantity	수량의 단위를 명시한다
Rate	비율을 표시한다(예 : 킬로미터/리터)
Text	한 언어의 여러 단어로 구성된 문자열

2.2.2 핵심 컴포넌트의 명명 규칙

핵심 컴포넌트의 명명 규칙은 다음과 같다.

- ① 사전 항목명은 유일해야하며 영어로 작성된다.
- ② 모든 이름은 대문자로 시작된다.

- ③ 특별히 복수형의 의미를 갖지않는한 단수형 명사를 사용한다.
- ④ 사전 항목명은 객체 클래스, 속성명, 표현명으로 이루어진다.
- ⑤ 사전 항목명은 반드시 표현명을 정의한다.
- ⑥ 객체 클래스, 속성명, 표현명은 "." 으로 구분한다. 예) Party.Country.Code
- ⑦ 객체 클래스, 속성명은 여러 단어로 구성될 수 있으며, 단어와 단어 사이의 공백은 사용하지 않음을 권고한다.
- ⑧ 사전 항목명은 중복된 단어가 연속적으로

나타나는 것을 허용하지 않는다.

- ⑨ 핵심 컴포넌트의 표현명은 "Type"을 사용한다. 예) Address.Code.Type
- ⑩ 기본 정보 개체의 표현명은 핵심 컴포넌트 유형을 참조하여 사용한다. 예) Address. Post.Code
- ⑪ 집합 정보 개체의 표현명은 "Details"를 사용한다. 예) Party.Name.Details

2.3 집합 정보 개체의 정의

집합 정보 개체의 정의 사례를 <표 2>에서 보

<표 2> 집합 정보 개체의 정의

UID	Aggregate Information Entity	Embeddded Entity	Core Component Type	Basic or Aggregate Type	Definition	Remarks
0001	Party			Aggregate	파티 정보	
0002		Party.Type.Code	Code.Type	Basic	파티 유형 코드	BY = 구매자 DP = 납품처 SU = 공급자
0003		Party.Type.Text	Text.Type	Basic	파티 유형 설명	
0004		Party.Name.Details	Details	Aggregate	이름 세부 사항	
0005		Party.Address.Details	Details	Aggregate	주소 세부 사항	
0006		Party.Identifier	Identifier.Type	Basic	파티 식별자	
0007	Address	Party.Organization.Details	Details	Aggregate	조직 세부 사항	
0008				Aggregate	조직 세부 사항	
0009		Address.Type.code	Code.Type	Basic	주소 타입 코드	
0010		Address.Street.Name.Text	Text.Type	Basic	번지, 통/반, 동, 건물명	
0011		Address.City.Name.Text	Text.Type	Basic	도시 이름	
0012		Address.Province.Name.Text	Text.Type	Basic	특별시, 광역시, 도	
0013		Address.Post.code	Code.Type	Basic	우편번호	
0014		Address.Country.Code	Code.Type	Basic	국가 코드	
0015		Address.Country.TExt	Text.Type	Basic	국가 이름	
0016	Organization	Organization.Name	Name.Type	Aggregate	조직 세부 사항	
0017				Basic	조직 이름	
0018		Orgqanization.President.Name	Name.Type	Basic	대표자 이름	
0019		Orgqanization.Status.Text	Text.Type	Basic	사업자 업태	
0020		Orgqanization.Classification.Text	Text.Type	Basic	사업자 업종	
0021		Orgqanization.Identifier	Identifier.Type	Basic	사업자등록번호	
0022		Orgqanization.Country.Code	Code.Type	Basic	조직 국가 코드	

여주고 있다.

2.4 기본 정보 개체의 정의

전자문서 내에서 사용될 우편번호를 위한 정보 개체를 정의한다면, Code.Type 핵심 컴포넌트를 사용하여 우편번호라는 비즈니스 의미를 부여한 Post.Code라는 기본 정보 개체를 정의할 수 있다. Post.Code 기본 정보 개체는 주소를 표현하는 집합 정보 개체에 포함되어 사용될 것이다. 기본 정보 개체의 이름은 “기본 정보 개체의 이름. 핵심 컴포넌트 표현 유형 이름”으로 명명된다. 기본 정보 개체 정의의 예는 <표 3>과 같다.

2.4 핵심 컴포넌트 라이브러리의 기본 구조

ebXML 표준스펙과 같은 핵심 컴포넌트 라이브러리의 기본구조를 <표 4>에서 보여주고 있다.

- ① UID : 집합 정보 개체 식별을 위한 ID.
UID는 향후 국제 또는 한국표준의 UID 부

여 방식을 따른다.

- ② Aggregate Information Entity : 집합 정보 개체의 이름. 예) Party, Name, Address
- ③ Embedded Entity : 집합정보 개체가 포함하고있는 기본 정보 개체들을 기술.
예) • Address.Type : Address 집합 정보 개체에 속하는 Type 기본 정보 개체
• Address.City : Address 집합 정보 개체에 속하는 City 기본 정보 개체
• Address.Street : Address 집합 정보 개체에 속하는 Street 기본 정보 개체
- ④ Core Component Type : 기본 정보 개체가 사용하는 핵심 컴포넌트 유형.
- ⑤ Basic or Aggregate Type : 핵심 컴포넌트가 기본 정보 개체인지 집합 정보 개체를 구분. 예) • Basic : 기본 정보 개체
• Aggregate : 집합 정보 개체
- ⑥ Definition : 핵심 컴포넌트에 대한 의미 설명
- ⑦ Remarks : 핵심 컴포넌트에 관련된 예제나 참조

<표 3> 기본 정보 개체의 정의

기본 정보 개체 이름	사용된 핵심 컴포넌트	설 명
Party.Identifier	Identifier.Type	Party를 구별할 수 있는 ID
Party.Name	Name.Type	Party 이름
Tax.Amount	Amount.Type	세금 총합계
Tax.Description.Text	Text.Type	세금 종류를 나타내는 문자열
Tax.Category.Code	Code.Type	세금의 분류를 나타내는 코드
Packaging.Quantity	Quantity.Type	포장 수량
Person.Birth.Date	Date.Type	생년월일
Currency.Exchange.Rate	Rate.Type	환 율

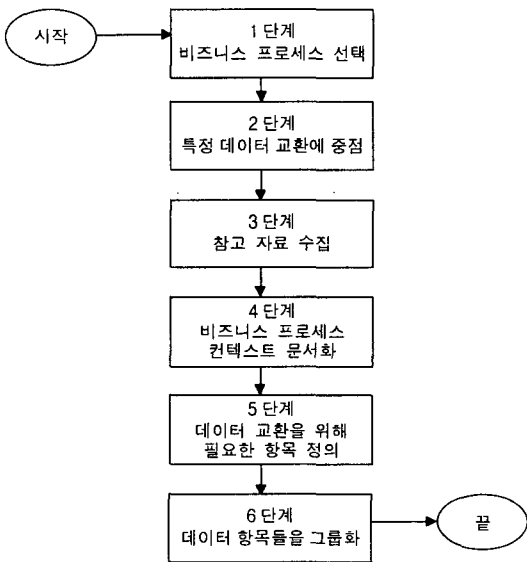
<표 4> 핵심 컴포넌트 라이브러리의 기본 구조

UID	Aggregate Information	Embedded Entity	Core Component Type	Basic or Aggregate Type	Definition	Remarks

3. 핵심 컴포넌트 정의 절차

3.1 준비 단계

핵심 컴포넌트를 정의하기 위해서 비즈니스 프로세스와 비즈니스 컨텍스트 선택에 있어서 그에 맞는 데이터 교환관련 핵심 컴포넌트 요소를 정의하는 과정이다. 준비단계에서 정의된 핵심 컴포넌트는 발견 및 분석과정을 통해서 등록자 장소에 등록되어지게 된다.



(그림 4) 핵심 컴포넌트 정의 단계

(그림 4)의 핵심 컴포넌트 정의 단계중 5단계까지는 기본 정보 개체의 정의 절차에 해당한다. 6단계는 집합 정보 정의 단계로써 2.3절에서 설명하고 있다.

1 단계 : 특정 도메인에서 광범위한 정보컨텐츠를 제공하는 비즈니스 프로세스를 선택한다.
예) 주문 프로세스, 지불 프로세스, 결제 프로세스

2 단계 : 비즈니스 프로세스상에서 발생하는 데이

터 중에서 반복적으로 재사용되는 핵심 비즈니스 정보를 담고있는 데이터 교환에 중점을 둔다. 예) 주문서, 송장, 대금 결제정보

3 단계 : 비즈니스 프로세스상에서 주고받은 전자문서와 관련된 비즈니스 정보 및 관련 세부사항을 수집한다. EDIFACT 문서개발을 위한 MIG(Message Implementation Guide), PIPs(Partner Interface Processes), 업체별 Flat File 등의 유사한 산출물을 참고한다. 본 연구의 사례에서는 유통업종에서 사용중인 MIG와 업체별 Flat File을 기준으로 핵심 컴포넌트 라이브러리를 구성한다.

4 단계 : 분석중인 비즈니스 프로세스의 컨텍스트를 문서화한다. 컨텍스트의 분류는 다음과 같다

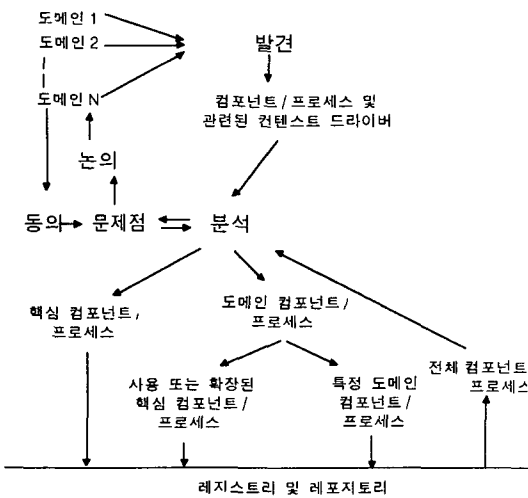
- 비즈니스 프로세스 컨텍스트
- 상품 분류 컨텍스트
- 업종 분류 컨텍스트
- 지역적 컨텍스트
- 비즈니스 프로세스 역할 컨텍스트
- 역할 컨텍스트
- 시스템 수준 컨텍스트

5 단계 : 데이터 교환을 위해서 필요한 데이터 항목을 선정한다. 선정된 데이터 항목에 알맞은 핵심 컴포넌트 유형을 적용한다.

6 단계 : 5단계에서 정의된 데이터 항목을 데이터 교환을 위한 항목별로 그룹화하여 집합 정보 개체를 정의한다. 집합 정보 개체는 2가지 이상의 기본 정보 개체 또는 집합 정보개체를 포함한다. 각각의 집합 정보개체는 자체의 비즈니스 의미론적 정의를 가진다.

3.2 발견(Discovery) 및 분석(Analysis) 단계

핵심 컴포넌트 정의 준비 단계에 의해서 정의된 핵심 컴포넌트를 등록저장소나 도메인별 핵심 컴포넌트로부터 발견하고 분석하여 추가로 필요한 핵심 컴포넌트를 선정하여 등록저장소에 등록하는 과정을 (그림 5)에서 보여주고 있다.

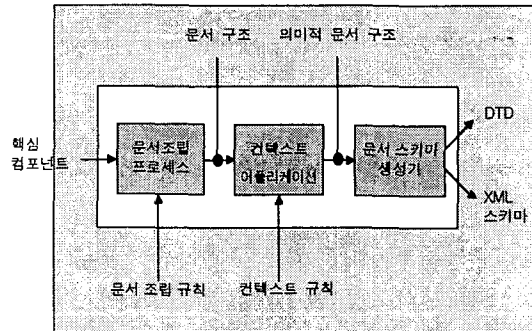


(그림 5) 핵심 컴포넌트 발견 및 분석 과정

4. 컴포넌트와 컨텍스트 기반의 전자 문서 개발 과정

핵심 컴포넌트를 이용하여 개발된 전자문서를 거래 상대방간의 상호운용성을 유지하면서 교환하기 위해서는 XML 전자문서 개발을 위해서 핵심 컴포넌트에 어떠한 제약사항이 적용되었는지 정의해줄 필요가 있다. 이렇게 핵심 컴포넌트 사용에 가해지는 제약들은 크게 핵심 컴포넌트를 조합하는 규칙과 여기에 의미적, 구조적 변화를 가해 정보 개체를 만들어내는 컨텍스트 규칙으로 나눌 수 있다.

문서조립 규칙과 컨텍스트 규칙이 적용되어 문서를 개발하는 과정은 (그림 6)과 같다.



(그림 6) XML 문서 스키마 생성과정

XML 문서 스키마의 생성과정은 다음과 같다.

- ① 핵심 컴포넌트를 선택한다.
- ② 핵심 컴포넌트를 이용할 문서조립 규칙을 작성한다.
- ③ 컨텍스트를 적용할 컨텍스트 규칙을 작성한다.
- ④ 핵심 컴포넌트와 문서조립 규칙 문서조립 프로세스에 전달한다
- ⑤ 문서조립 프로세스는 핵심 컴포넌트를 문서조립 규칙에 맞게 조립하여 문서구조를 생성한다.
- ⑥ 생성된 문서 구조와 컨텍스트 규칙을 컨텍스트 어플리케이션을 전달한다.
- ⑦ 컨텍스트 어플리케이션으로부터 컨텍스트가 적용된 의미적 문서 구조가 생성된다.
- ⑧ 생성된 의미적 문서구조는 문서 스키마 생성기로 전달된다.
- ⑨ 문서 스키마 생성기는 XML DTD 또는 XML 스키마를 생성한다.

4.1 문서 조립 규칙(Document assembly rule)

문서조립은 저장소로부터 핵심 컴포넌트가 추출되고, 스키마 모델로 만들기 위해 사용되는 규칙에 근거하는 프로세스이다. 이것은 적절한 때,

관련된 컨텍스트 규칙의 어플리케이션 적용 후에 XML스키마 만들기 위해 사용되고, 또한 비즈니스 문서내용을 유효화하는데 사용된다. 예를 들면 구매주문 스키마는 양측 당사자(구매자, 판매자), 일련의 품목으로 구성된다. 구매주문은

저장소에서 발견되는 핵심 컴포넌트들로 구성되어야 하고 구매 주문은 핵심 컴포넌트가 아니다. 문서조립 규칙과 컨텍스트 규칙은 구문론적으로 공통점이 많기 때문에 <표 5>에서 하나의 표로 설명한다.

<표 5> 문서 조립 규칙 및 컨텍스트 규칙 구분

명칭(Names)	유형(Type)	발생빈도	기본값	설 명(Description)
Assembly				
Assembly	complex	+		조립된 핵심 컴포넌트 리스트
@name	string	선 택		조립된 문서 스키마 집합 명칭
@Version	string	선 택		조립 규칙 문서 버전
Assemble				
CreateElement	complex	+		핵심 컴포넌트 리스트
CreateGroup	complex	*		요소들의 그룹 생성
@name	string	필 수		조립되는 문서 스키마 명칭
CreateGroup				
@type	enum	기 본	Sequence	생성되는 그룹 유형(단 허용되는 값은 sequence와 choice다)
CreateGroup	complex	*		요소들 그룹 생성
CreateElement	complex	*		요소 생성
UseElement	complex	*		생성될 요소 자식들 중 지명된 요소만 사용
Annotation	complex	*		주석 삽입
CreateElement				
Type	string	선 택		생성될 요소 유형
MinOccurs	string	선 택		생성된 요소 최소 발생횟수
Maxoccurs	string	선 택		생성된 요소 최대 발생 횟수(정수 이외에) 단 가능한 값은 unbounded 임
@id	ID	필 수		생성된 요소 ID
@idref	IDREF	선 택		다른 생성된 요소 ID로 참조
Name	string	필 수		조립될 요소 명칭
@location	UUID URI	필 수		조립될 요소위치(즉 레지스트리 참조)
Rename	EMPTY	선 택		조립된 요소 명칭
Annotation	complex	*		주석삽입
Rename				
@from	string	필 수		명칭 변경될 자식요소의 본래 명칭
@to	string	필 수		명칭 변경될 자식요소의 신규 명칭
ContextRules				
Rule	complex	+		적용되는 규칙 리스트
@version	string	선 택		ContextRule 문서 버전

Rule				
명칭(Names)	유형(Type)	발생빈도	기본값	설 명(Description)
@Apply	enum	기 본	Exact	아래 참조
Condition	complex	필 수		규칙(Rule)이 적용되어야 시점
Action	Complex	+		규칙(Rule) 적용될 때 수행할 행위(Action)
@order	Integer	기 본	0	규칙(Rule)들이 적용될 순서를 지시. 순서에서 높은 값을 갖는 규칙(Rule)이 우선 수행
Taxonomy	EMPTY	+		계층적인 조건이 적용된 규칙 내에서 사용된 분류 리스트
Taxonomy				
@ref	URI	필 수		분류(Taxonomy)에 대한 포인터
Condition				
@Test	string	필 수		규칙(Rule)이 적용되어야할지 테스트하기 위한 Boolean 표현, XPath 구문[XPATH]을 사용
Action				
@ApplyTo	string	필 수		행위가 적용될 노드(Node)
Add/Subtract Occurs	complex	+		컨텐츠 모델에 대한 변경 리스트
UseElement				
Name	string	필 수		사용될 요소 명칭
Annotation	string	*		주석삽입
Comment				
	string	선 택		규칙(rules) 문서가 위치하는 곳에서 설명을 기록. 결과문서 산출물에 영향이 없음
Type				
	string	선 택		산출물 유형
Minoccurs				
	string	선 택		산출물에 발생(occurrence) 최소 횟수
Maxoccurs				
	string	선 택		산출물에 발생(occurrence) 최대 횟수

4.1.2 문서 조립 규칙의 XML문서

문서 조립 규칙 XML 사례에서는 유통업종 주 문서를 개발하기 위한 핵심 컴포넌트 조립 규칙이다. 조립 규칙에 대한 개략적인 설명은 다음과 같다.

- 구매자(Buyer)와 판매자(Seller)는 핵심 컴포넌트 Party를 이용하여 Buyer와 Seller라는 이름으로 엘리먼트를 생성한다.
- 핵심 컴포넌트 Party에 포함된 기본 정보

개체 중에서 사용할 항목은 PartyTypeCode, PartyTypeText, Organization, Address 이다.

- Organization과 Address를 생성하는 경우에는 id에 따라 사용하는 엘리먼트가 다르다.
- CARREFOUR의 경우는 전 세계에 지점을 갖고 있으므로 Address 정보표현을 미국식으로 표현하고 있다. Address를 AddressUS로, Address/Province를 State로 이름을 변경한다.

```

<?xml version = "1.0"?>
<!-- 유통 주문서 조립 규칙 -->
<Assembly version = "1.0">
  <Assemble name = "PurchaseOrder" id = "PO" >
    <CreateGroup>
      <CreateElement id = "Buyer" location = "UUID">
        <Name> Buyer </Name>
        <Type> Party </Type>
        <CreateGroup>
          <UseElement name = "PartyTypeCode">
            <UseElement name = "PartyTypeText">
              <UseElement name = "Organization">
                <CreateGroup id = "UKPDC02">
                  <UseElement name = "Registration-
                    Number" />
                  <UseElement name = "PresidentName"/>
                </CreateGroup>
              </UseElement>
            </CreateGroup>
          </UseElement>
          <UseElement name = "Address">
            <CreateGroup id = "UKPDC02">
              <UseElement name = "Type" />
              <UseElement name = "Street" />
              <UseElement name = "City" />
              <UseElement name = "Province" />
              <UseElement name = "PostCode" />
            </CreateGroup>
          </UseElement>
        </CreateGroup>
      </CreateElement>
      <Condition test = "$ServiceUser = 'CARRE-
        FOUR'">
        <Rename from = "Address" to = "Address-
          US"/>
        <Rename from = "Address/Provice" to =
          "State"/>
        <Rename from = "address/PostCode" to =
          "ZIP"/>
      </Condition>
    </CreateElement>
    <CreateElement id = "Seller" location = "UUID">
      <Name> Seller </Name>
      <Type> PartyType </Type>
      <CreateGroup>
        <UseElement name = "PartyTypeCode">
          <UseElement name = "PartyTypeText">
            <UseElement name = "Organization">
              <CreateGroup id = "UKPDC02">
                <UseElement name = "Registration-
                  Number" />
                <UseElement name = "PresidentName" />
              </CreateGroup>
            </UseElement>
          </CreateGroup>
        </CreateElement>
      </CreateGroup>
    </CreateElement>
  </Assemble>
</Assembly>

```

~~~~~ 중략 ~~~~~

## 4.2 컨텍스트 규칙(Context rule)

컨텍스트 규칙은 핵심 컴포넌트를 특정 컨텍스트에 맞게 의미적으로나 구조적으로 재정의 하여 비즈니스 정보 개체로 만든다. 비즈니스 프로세스가 발생될 때, 컨텍스트는 컨텍스트 카테고리(분류)와 그와 관련된 값으로 식별된다. 예를 들면 자동차 제조업자가 화학약품 제조업체로부터 페인트를 구매하고자 할때, <표 6>과 같은 컨텍스트 값으로 구성된다.

<표 6> 컨텍스트

| 컨텍스트 분류   | 값      |
|-----------|--------|
| 프로세스      | 조 달    |
| 상품 분류     | 페인트    |
| 지역(구매자)   | 프랑스    |
| 지역(판매자)   | 미 국    |
| 산업분야(구매자) | 없음(일반) |
| 산업분야(판매자) | 소매업    |

### 4.2.1 컨텍스트 규칙의 DTD

```

<?xml version = "1.0" encoding = "euc-kr"?>
<!ELEMENT ContextRules (Rule+)>
<!ATTLIST ContextRules
  version CDATA # IMPLIED
  id ID # IMPLIED
  idref IDREF # IMPLIED
>

<!ELEMENT Rule (Taxonomy+, Condition+)>
<!ATTLIST Rule
  apply (exact|hierarchical) "exact"
  order CDATA # IMPLIED
  id ID # IMPLIED
  idref IDREF # IMPLIED
>

<!ELEMENT Taxonomy EMPTY>
<!-- ref는 URI 형태로 표시되어야 한다 -->
<!ATTLIST Taxonomy
  context CDATA # REQUIRED
  ref CDATA # REQUIRED
  id ID # IMPLIED
  idref IDREF # IMPLIED
>

```



```

<?xml version = "1.0"?>
<!-- 유통 주문서 컨텍스트 규칙 -->
<!DOCTYPE ContextRules SYSTEM "contextrules.
dtd">
<ContextRules id = "DacomService" >
  <Rule apply = "hierarchical">
    <Taxonomy context = "Geopolitical"
      ref = "http://ebxml.org/classification/
        ISO3166"/>
    <Taxonomy context = "Industry"
      ref = "http://ebxml.org/classification/industry/
        retail"/>
    <Condition test = "$Geopolitical = 'Korea'">
      <Action applyTo = "//Buyer/Organization">
        <Add after = "@id = 'UKPDC02'">
          <CreateGroup type = "FollowedBy">
            <Element >
              <Name> BuyerID </Name>
              <Type> string </Type>
            </Element>
            <Element >
              <Name> BuyerLocationnCode
                </Name>
              <Type> string </Type>
            </Element>
          </CreateGroup>
        </Add>
      </Action>
      <Action applyTo = "//Buyer/Address">
        <Condition
          test = "$ServiceUser = 'CARREFOUR'">
          <Occurs>
            <Element >
              <Name> State </Name>
              <Name> ZIP </Name>
            </Element>
          </Occurs>
        </Condition>
      </Action>
    </Condition>
  </Rule>
</ContextRules>

```

### 5. 핵심 컴포넌트 스키마

본 연구에서 유통부문의 3개 문서를 대상으로 발췌한 핵심 컴포넌트를 핵심 컴포넌트 사전의 구조뿐만 아니라 XML 스키마 문서로 표현한 사례이다. 이들 핵심 컴포넌트 스키마는 (그림 6)

의 XML 문서 스키마 생성과정에서 나타나는 문서 조립 프로세스의 입력이 되는 문서들이다.

#### 5.1 Orders.xsd

Orders.xsd는 한국 유통부문 핵심 컴포넌트를 이용하여 개발된 주문서 스키마를 정의하고 있다.

```

<?xml version = "1.0" encoding = "EUC-KR"?>
<schema targetNamespace = "http://www.ebxml.or.
kr/schema/Retail"
  xmlns = "http://www.w3.org/2000/10/XMLSchema"
  xmlns : qrsc = "http://www.ebxml.or.kr/schema/
    Retail">
  <annotation>
    <documentation> 한국 유통부문 구매자
      </documentation>
  </annotation>
  <include schemaLocation = "Party.xsd"/>
  <include schemaLocation = "Address.xsd"/>
  <include schemaLocation = "Organization.
    xsd"/>
  <include schemaLocation = "ContactPoint.
    xsd"/>
  <complexType name = "BuyerParty.Details">
    <complexContent>
      <extension base = "retcc : Party.Details">
        <sequence>
          <element name = "Address" type = "ret
            c : Address.Details" minOccurs = "0"/>
          <element name = "Organization" type =
            "retcc : Address.Details" minOccurs =
            "0"/>
          <element name = "Contact" type = "retc
            c : Contact.Details" minOccurs = "0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>

```

#### 5.2 Party.xsd

핵심 컴포넌트 사전의 집합 정보 개체인 Party 항목이 Party.xsd의 문서로 표현된다. Party.xsd는 한국 유통부문 Party 핵심 컴포넌트의 스키마를 정의하고 있다.

```

<?xml version = "1.0" encoding = "EUC-KR"?>
<schema targetNamespace = "http://www.ebxml.or.kr/schema/Retail" xmlns = "http://www.w3.org/2000/10/
XMLSchema" xmlns : retcc = "http://www.ebxml.or.kr/schema/Retail">
  <annotation>
    <documentation> 한국 유통부문 Party 핵심 컴포넌트 </documentation>
  </annotation>
  <include schemaLocation = "CoreComponentType.xsd"/>
  <include schemaLocation = "Organisztion.xsd"/>
  <complexType name = "Party">
    <sequence>
      <element name = "PartyTypeCode" minOccurs="0">
        <complexType>
          <sequence>
            <element name = "CodeContent">
              <simpleType>
                <restriction base = "string">
                  <enumeration value = "BY">
                    <annotation>
                      <documentation> 구매업체 </documentation>
                    </annotation>
                  </enumeration>
                  <enumeration value = "DP">
                    <annotation>
                      <documentation> 납품업체 </documentation>
                    </annotation>
                  </enumeration>
                  <enumeration value = "SU">
                    <annotation>
                      <documentation> 공급업체 </documentation>
                    </annotation>
                  </enumeration>
                </restriction>
              </simpleType>
            </element>
            <element name = "CodeList.Identifier" type = "string" minOccurs = "0"/>
            <element name = "CodeList.Agency.Identifier" type = "string" minOccurs = "0"/>
            <element name = "CodeListVersion.Code" type = "string" minOccurs = "0"/>
            <element name = "Name" type = "string"
          </sequence>
        </complexType>
      </element>
      <element name = "PartyIdentifier" type = "retcc : Identifier.Type" minOccurs = "0">
        <annotation>
          <documentation> 당사자 식별자 </documentation>
        </annotation>
      </element>
      <element name = "DescriptionText" type = "retcc : Text.Type" minOccurs = "0"/>
      <element name = "Address" type = "retcc : Address.Details" minOccurs = "0"/>
      <element name = "Organization" type = "retcc : Organisation.Details" minOccurs = "0"/>
    </sequence>
  </complexType>
</schema>

```

## 5.3 Address.xsd

Address.xsd는 한국 유통부문 Address 핵심

컴포넌트의 스키마를 정의하고 있다.

```

<?xml version = "1.0" encoding = "EUC-KR"?>
<schema targetNamespace = "http://www.ebxml.or.kr/schema/Retail" xmlns = "http://www.w3.org/2000/10/
XMLSchema" xmlns : retcc = "http://www.ebxml.or.kr/schema/Retail">
  <annotation>
    <documentation> 한국 유통부문 Address 핵심 컴포넌트 </documentation>
  </annotation>
  <include schemaLocation = "CoreComponentType.xsd"/>
  <simpleType name = "Post.Code">
    <restriction base = "string">
      <pattern value = "3-3"/>
    </restriction>
  </simpleType>
  <complexType name = "Address.Details">
    <sequence>
      <element name = "AddressTypeCode" minOccurs = "0">
        <complexType>
          <sequence>
            <element name = "Content" minOccurs = "0">
              <simpleType>
                <restriction base = "string">
                  <enumeration value = "1">
                    <annotation>
                      <documentation> 가정집주소 </documentation>
                    </annotation>
                  </enumeration>
                  <enumeration value = "2">
                    <annotation>
                      <documentation> 연락처주소 </documentation>
                    </annotation>
                  </enumeration>
                  <enumeration value = "3">
                    <annotation>
                      <documentation> 도착지주소 </documentation>
                    </annotation>
                  </enumeration>
                  <enumeration value = "4">
                    <annotation>
                      <documentation> 출발지주소 </documentation>
                    </annotation>
                  </enumeration>
                  <enumeration value = "5">
                    <annotation>
                      <documentation> 사업장주소 </documentation>
                    </annotation>
                  </enumeration>
                </restriction>
              </simpleType>
            </element>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>

```

```

        </enumeration>
      </restriction>
    </simpleType>
  </element>
  <element name = "CodeList.Identifier" type = "string" minOccurs = "0"/>
  <element name = "CodeList.Agency.Identifier" type = "string" minOccurs = "0"/>
  <element name = "CodeList.Version.Code" type = "string" minOccurs = "0"/>
  <element name = "Name" type = "string" minOccurs = "0"/>
</sequence>
</complexType>
</element>

```

~~~~~ 요약 ~~~~~

5.4 Organization.xsd

Organization.xsd는 한국 유통부문 Organization 핵심 컴포넌트의 스키마를 정의하고 있다.

호 교환되는 거래 전자문서 작성에 있어서 업종간 핵심 컴포넌트 사전 및 어셈블리 규칙을 고려하여 작성한다면 상호간의 시맨틱 연동성이 증가하여 시스템간 연동을 원활히 할 수 있게 된다. 본

```

<?xml version = "1.0" encoding = "EUC-KR"?>
<schema targetNamespace = "http://www.ebxml.or.kr/schema/ Retail" xmlns = "http://www.w3.org/2000/10
/XMLSchema" xmlns : retcc = "http://www.ebxml.or.kr/schema/Retail">
  <annotation>
    <documentation> 한국 유통부문 Organization 핵심 컴포넌트 </documentation>
  </annotation>
  <include schemaLocation = "CoreComponentType.xsd"/>
  <complexType name = "Organization.Details">
    <sequence>
      <element name = "OrganisationName" type = "retcc : Text. Type" minOccurs = "0"/>
      <element name = "PresidentName" type = "retcc : Text. Type" minOccurs = "0"/>
      <element name = "StatusText" type = "retcc : Text.Type" minOccurs = "0"/>
      <element name = "ClassificationText" type = "retcc : Text. Type" minOccurs = "0"/>
      <element name = "RegistrationCountryCode" type = "retcc : Code.Type" minOccurs = "0"/>
      <element name = "RegistrationNumber" type = "retcc : Identifier.Type" minOccurs = "0"/>
    </sequence>
  </complexType>
</schema>

```

6. 결 론

ebXML은 전자상거래 촉진을 위한 시맨틱 상호 연동성(semantic interoperability)을 추구하여 단일한 전 세계적 규모의 전자상거래 마켓을 구축하는 것을 목표로 한다. 따라서 거래 당사자간의 상

연구는 국제표준에 신속히 대처하고 국내 ebXML 확산에 따른 대응을 신속히 하는데 그 의의가 있다.

이 연구는 2001년 5월에 발행된 "ebXML_Guide to core component dictionary v.1.04, ebXML_Core component dictionary v.1.04, Con-

text and Re-Usability of Core Components V1.04, Document Assembly and Context Rules V1.04, Catalogue of Context Drivers V1.04" 연구와 ISO 11179/3 메타데이터 저장소 규격을 인용하여 ebXML 프레임워크상의 전자문서 설계규격 사례를 개발하는 연구를 개발하였다. 이를 위해서 유통업종 EDIFACT표준 문서 3종(주문서, 발송통지서, 판매현황보고서)만을 대상으로 인용규격을 준수하면서, 핵심 컴포넌트를 설계하고 등록저장소에 저장된 핵심 컴포넌트를 추출하여 스키마를 생성하기 위해 문서조합 및 컨텍스트 규칙을 적용하는 규칙기반 프로세스 작업을 진행하였다.

본 연구에서는 유통업종 일부 문서만을 대상으로 컴포넌트를 발췌하여 ebXML 전자문서 개발 과정을 보였다. 이 연구의 개발 사례의 일반화(generalization)를 통하여 학습이 진행되면서 여러 산업으로 핵심 컴포넌트를 고려한 ebXML 전자문서 개발의 신속한 확산을 기대한다.

참 고 문 헌

- [1] 산업자원부. (2002). 차세대 전자상거래 표준 프레임워크 백서: 2002 ebXML(Electronic business extensible markup language). 산업자원부.
- [2] 김형도. (2000). B2B 전자상거래@XM 배움터.
- [3] KS X 1517-3. (1996). 데이터 요소-제 3부: 데이터 요소의 기본 속성.
- [4] KS X 1517-4. (1996). 데이터 요소-제 4부: 데이터 요소의 정의를 위한 규칙과 지침
- [5] KS X 1517-5. (1997). 데이터 요소-제 5부: 데이터 요소를 위한 명명 및 식별원칙.
- [6] KS X 1517-6. (1997). 데이터 요소-제 6부: 데이터 요소의 등록.
- [7] ebXML. <http://www.ebxml.org>.
- [8] ebXML. (2001). Context and Re-Usability of Core Components V1.04.
- [9] ebXML. (2001). Document Assembly and Context Rules V1.04.
- [10] ebXML. (2001). Catalogue of Context Drivers V1.04.
- [11] ebXML. (2001). Guide to core component dictionary v.1.04.
- [12] ebXML. (2001). Core component dictionary v.1.04.
- [13] ebXML Registry Project Team. (2001). Registry Information Model v1.0. <http://www.ebxml.org/specs/ebRIM.pdf>.
- [14] ebXML Registry Project Team. (2001). Registry Service v1.0. <http://www.ebxml.org/specs/ebRS.pdf>.
- [15] ebXML Business Process Project Team. (2001). Business Process Specification Schema v1. 01. <http://www.ebxml.org/specs/ebBPSS.pdf>.
- [16] ebXML Initiative. (2002). Core Component Overview v1.05. <http://www.ebxml.org/specs/ebOVER.pdf>.
- [17] ebXML Trading-Partners Team. (2001). Collaboration Protocol Profile and Agreement Specification version 1.0. <http://www.ebxml.org/specs/ebCPP.pdf>.
- [18] ebXML Transport, Routing, and Packaging Project Team. (2001). Messaging Service Specification version 1.0. <http://www.ebxml.org/specs/ebMS.pdf>.
- [19] IBM. developerWorks Web Services Zone,

<http://www.ibm.com/developerWorks/webservices>.

- [20] ISO/IEC IS 11179, Information technology Specification and standardization of data Elements.
- part 1 : Framework for the specification and standardization of data element, December, 1999, Final draft
 - part 2 : Classification for data elements, February, 2000, Final draft.
 - part 3 : Basic attributes of data elements, 1994.
 - part 4 : Rules and guidelines for the formulation of data definitions, 1995.
 - part 5 : Naming and identification principles for data elements, 1995.
 - part 6 : Registration of data elements, 1997.
- [21] W3C. (2000). Simple Object Access Protocol version(SOAP) 1.1. <http://www.w3.org/TR/SOAP>.

■ 저자소개



김 완 평

Wan-Pyong Kim received the B.S. degree in computer science and statistics from Chonnam National University in 1985, and the M.S degree in computer science from Seoul National University in 1987. He is now in PH.D course in Department of Management Information Systems, Hankuk University of Foreign Studies. He obtained a (information processing) Professional Engineer license in 1997. From 1991 to 1992, he was employed as a software engineer in Samsung Advanced Institute of Technology. From 1998 to 2000, he was employed as a full-time instructor in Hanyang Womens College. Since Jan. 1993, he has been with DACOM Corporation as a senior manager in New-Biz Technology Team. His research interests are Internet business, XML/EDI, Knowledge Management, and Information Security.