

# 기업간 비즈니스 프로세스 등록저장소를 위한 메타데이터 온톨로지 설계

김종우<sup>†</sup> · 김형도<sup>\*\*</sup> · 윤정희<sup>\*\*\*</sup> · 정현철<sup>\*\*\*\*</sup>

## 요 약

B2B 등록저장소는 기업의 프로파일, 비즈니스 문서, 비즈니스 프로세스, 제품 서비스 등의 B2B 관련 비즈니스 정보들을 저장하고 거래 파트너 또는 잠재 거래 파트너들에게 정보를 제공하는 시스템이다. 본 논문에서는 이 중 B2B 비즈니스 프로세스의 등록저장소의 설계에 초점을 맞추어, 이를 지원하기 위한 메타데이터 온톨로지를 제시한다. 현재 ebXML BPSS (Business Process Specification Schema), WSBPEL (Web Service Business Process Execution Language), BPMN (Business Process Modeling Notation) 등의 다수의 비즈니스 프로세스 정의 언어들이 시장에서 경쟁하고 있다. 이러한 상황을 고려하여, 다수의 다른 프레임워크에서 표현된 비즈니스 프로세스를 등록저장하기 위해서 제시된 메타데이터 온톨로지는 3개 계층, 공통 메타데이터, 언어별 메타데이터, 상호연관관계 메타데이터로 구성된다. 제시된 메타데이터 온톨로지의 유용성을 보이기 위해서, ebXML BPSS와 WSBPEL로 표현된 비즈니스 프로세스의 온톨로지 내에 저장할 예제로 제시한다. 또한 제시된 메타데이터 온톨로지의 구현을 위해서 ebXML 등록저장소 정보모델로의 매핑 방안을 제시한다.

키워드 : 비즈니스 프로세스, B2B 등록저장소, 메타데이터 온톨로지

## Metadata Ontology Design for B2B Business Process Registries

Jong Woo Kim<sup>†</sup> · Hyoung Do Kim<sup>\*\*</sup> · Jung Hee Yun<sup>\*\*\*</sup> · Hyun Chul Jung<sup>\*\*\*\*</sup>

## ABSTRACT

B2B registries are information systems to register B2B related business information such as companies' profiles, business documents, business processes, and services and to provide query facilities to find information about potential business partners. Focusing on the design of the registry for B2B business processes, in this paper, a metadata ontology is designed to register B2B business processes. In practice, there are several competitive business process definition languages such as ebXML BPSS (Business Process Specification Schema), WSBPEL (Web Service Business Process Execution Language), BPMN (Business Process Modeling Notation), and so on. In order to register heterogeneous business processes based on different representation frameworks, the proposed metadata ontology consists of three layers, common metadata, language-specific metadata, and interrelationship metadata. To show the usefulness of the proposed metadata ontology, two examples which are represented by ebXML BPSS and WSBPEL respectively are described in order to show how the proposed metadata ontology is used to registry B2B business processes. To implement the proposed metadata ontology using ebXML registry, metadata mapping scheme to ebRIM (ebXML Registry Information Model) is also suggested.

Key Words : Business Process, B2B Registry, Metadata Ontology

## 1. 서 론

기업간의 전자거래를 위해서는 거래 기업 또는 잠재적 거래 기업들이 비즈니스를 수행하기 위해 필요한 정보를 등록하고 검색할 수 있는 공간이 필요한데, 이를 B2B 등록저장소라 한다[1-3]. 이러한 등록저장소에 등록, 공유될 필요가 있는 주요 기업간 거래 정보로는 거래 파트너 프로파일, 전

자 문서 양식, 비즈니스 프로세스 정의, 서비스 정보 등이 있다. 이 중 비즈니스 프로세스 정의는 비즈니스 프로세스가 가지는 복잡성과 동적인 특성으로 인해서, 표현의 복잡성과 다양성이 존재한다. 또한 현실적으로 ebXML BPSS (Business Process Specification Schema), WSBPEL (Web Service Business Process Execution Language), BPMN (Business Process Modeling Notation), WSCL (Web Services Conversation Language), WSCI (Web Service Choreography Interface), Microsoft사의 XLANG 등 다양한 비즈니스 프로세스 정의 언어가 존재하고 경쟁하고 있다 [4-10]. 따라서, 비즈니스 프로세스 정보를 등록저장소에 저

<sup>†</sup> 종신회원 : 한양대학교 경영학부 부교수(교신저자)

<sup>\*\*</sup> 정 회 원 : 한양사이버대학교 경영학부 교수

<sup>\*\*\*</sup> 정 회 원 : 한국정보사회진흥원 ITA표준팀 선임연구원

<sup>\*\*\*\*</sup> 정 회 원 : 한국정보사회진흥원 선임연구원

논문접수 : 2007년 3월 15일, 심사완료 : 2007년 6월 1일

장하고 재활용해야 하는 필요성이 존재함에도 불구하고, 비즈니스 프로세스 표현의 복잡성과 표현의 다양성은 비즈니스 프로세스의 등록 및 관리를 어렵게 하고 있다. 또한, 현재 등록저장소에 비즈니스 프로세스를 전체 파일 단위로 관리하고 있기 때문에, 비즈니스 프로세스 구성요소 수준에서의 재활용이 불가능하고, 비즈니스 프로세스 정의에 포함된 내용을 기반한 의미적인 검색이 불가능한 형편이다. 따라서, 본 연구에서는 다양한 표현 방식을 갖는 비즈니스 프로세스 정의들을 통합적으로 등록, 저장하고 검색할 수 있는 B2B 등록저장소 메타데이터 온톨로지를 제시하고자 한다. 또한 이러한 메타데이터 온톨로지를 ebXML 등록저장소 표준에 맞추어 개발하기 위한 매핑 규칙을 제시하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련 연구 검토로, 기존 B2B 비즈니스 프로세스 표현 방법들과 B2B 등록저장소에 대하여 검토한다. 3장에서는 본 논문에서 제시하고자 하는 메타데이터 온톨로지를 ebXML BPSS와 WSBPEL로 표현된 비즈니스 프로세스 예제를 중심으로 소개한다. 4장에서는 제시된 메타데이터 온톨로지의 ebRIM (ebXML Registry Information Model)로의 매핑 방안을 제시한다. 5장에서는 결론을 제시한다.

## 2. 관련 연구

### 2.1 기업간 비즈니스 프로세스의 표현 방법

비즈니스 프로세스는 조직의 미션 달성을 지원하기 위한 여러 단계에 걸쳐서 이루어지는 활동들을 의미한다[11]. 특히, 기업간 비즈니스 프로세스는 기업간에 진행되는 비즈니스 로직을 의미한다. 비즈니스 프로세스를 정의하기 위해서는 단위업무(Business Transaction)에 대한 정의, 메시지 또는 정보에 대한 교환 흐름을 정의하는 문서 흐름(Document Flow) 정의 외에도 단위 업무들에 대한 업무 흐름 및 순서 등을 정의하기 위한 조율(Choreography) 등에 대한 정의가 필요하다. 비즈니스 프로세스를 표현하는 방법에는 도식적인 표현 방법과 XML 기반의 표현 방법이 있다. 객체지향 표준 모델링 방법론인 UML (Unified Modeling Language)을 사용하여 기업간 비즈니스 프로세스를 도식적으로 표현하는 경우가 많이 있었으나, 최근에는 BPMN (Business Process Modeling Notation)의 사용이 증가하고 있다[4, 9, 12-16]. XML 기반의 비즈니스 프로세스 기술 방법에는 ebXML의 BPSS (Business Process Specification Schema), 웹 서비스 진영의 WSBPEL (Web Service Business Process Execution Language), WSCI (Web Service Choreography Interface), WS-CDL (Web Service Choreography Description Language) 등이 있다[5-9, 17].

### 2.2 B2B 등록저장소

e-비즈니스 거래 파트너들이 비즈니스를 수행하기 위해서 필요한 거래 파트너 프로파일, 비즈니스 프로세스, 서비스 정보 등을 저장하고 검색할 수 있는 서비스를 제공하는

장소를 B2B 등록저장소라 한다. 이러한 등록저장소는 거래를 하려는 잠재적 거래 기업들을 발견하고 이 파트너의 비즈니스 정보를 얻을 수 있을 뿐만 아니라, 자신의 정보를 다른 잠재적 거래 파트너에게 알릴 수 있는 장소가 되기 때문에 기업간 전자상거래에서 핵심적인 부분이다[18].

ebXML 프레임워크에서는 “등록저장소”라는 컴포넌트를 표준화하여 서로간에 정보를 공유할 수 있도록 기술기반구조를 제시하고 있다. 이를 위해서 등록저장소 정보 저장을 위한 ebRIM (ebXML Registry Information Model), 등록저장소 정보 검색을 위한 ebRS (ebXML Registry Service and Protocol)가 표준으로 제시되고 있다[2, 3]. 웹서비스의 경우는 서비스 명세 표준인 WSDL (Web Service Description Language)과 함께, 서비스 등록 및 발견(discovery) 표준인 UDDI (Universal Description, Discovery, and Integration)를 사용하고 있다[19]. UDDI는 전세계 서비스에 대한 명세, 발견, 통합 등에 대한 정보를 담고 있는 서비스 등록저장소로, 현재 웹 서비스의 등록저장소 표준으로 사용되고 있다. 이는 2000년부터 마이크로소프트, IBM, 아리바의 주도로 개발되었으며, UDDI 표준에 대한 개발과 관리는 OASIS의 UDDI 기술위원회에서 이루어지고 있다. SOA (Service Oriented Architecture)가 새로운 분산 컴퓨팅 패러다임으로 발전하면서, 서비스의 등록과 탐색은 매우 중요한 주제로 인식되고 있으며, 기존의 UDDI의 확장이나 온톨로지를 사용해서 의미적인 요소를 통한 검색을 지원하기 위한 노력들이 최근에 이루어지고 있다[18, 20-22]. 하지만 아직까지는 비즈니스 프로세스 등록, 저장을 위한 등록저장소 메타데이터에 대한 논의는 부족한 형편이다.

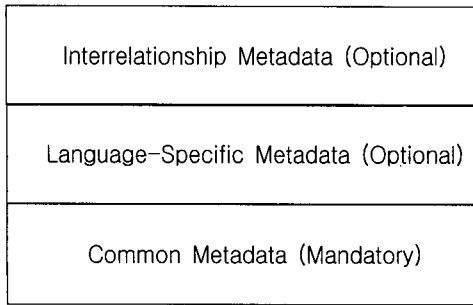
B2B 등록저장소 내의 주요한 콘텐츠 중의 하나인 비즈니스 프로세스 정의(Business Process Specification, SPS)의 저장과 관련해서 현재 다음과 같은 문제점이 존재한다. 첫째 B2B 비즈니스 프로세스는 다양한 표현 방식으로 존재하고 있다. 따라서 B2B 등록저장소는 ebXML BPSS, WSBPEL, BPMN 등의 다양한 형태로 표현된 비즈니스 프로세스 정의를 모두 저장할 수 있어야 한다. 둘째, REMKO(www.remko.or.kr)와 같이 현재 사용되고 있는 B2B 등록저장소에서는 재활용의 대상이 전체 비즈니스 프로세스로 제한되어 있으나, 비즈니스 프로세스가 복합적인 객체이므로, 재활용의 대상을 비즈니스 프로세스의 구성요소인 단위업무, 비즈니스 협업, 역할, 문서 등으로 확장하는 것이 필요하다.

## 3. 기업간 비즈니스 프로세스 메타데이터 온톨로지

### 3.1 메타데이터 온톨로지 개요

본 논문에서 제시하는 기업간 비즈니스 프로세스 메타데이터 온톨로지는 (그림 1)과 같이 크게 3가지 종류의 메타데이터를 기술할 수 있도록 지원한다.

공통 메타데이터(Common Metadata)는 비즈니스 프로세스 명세 언어에 공통적으로 적용될 수 있는 가장 기본이 되는 메타데이터이며, 언어별 메타데이터(Language-Specific



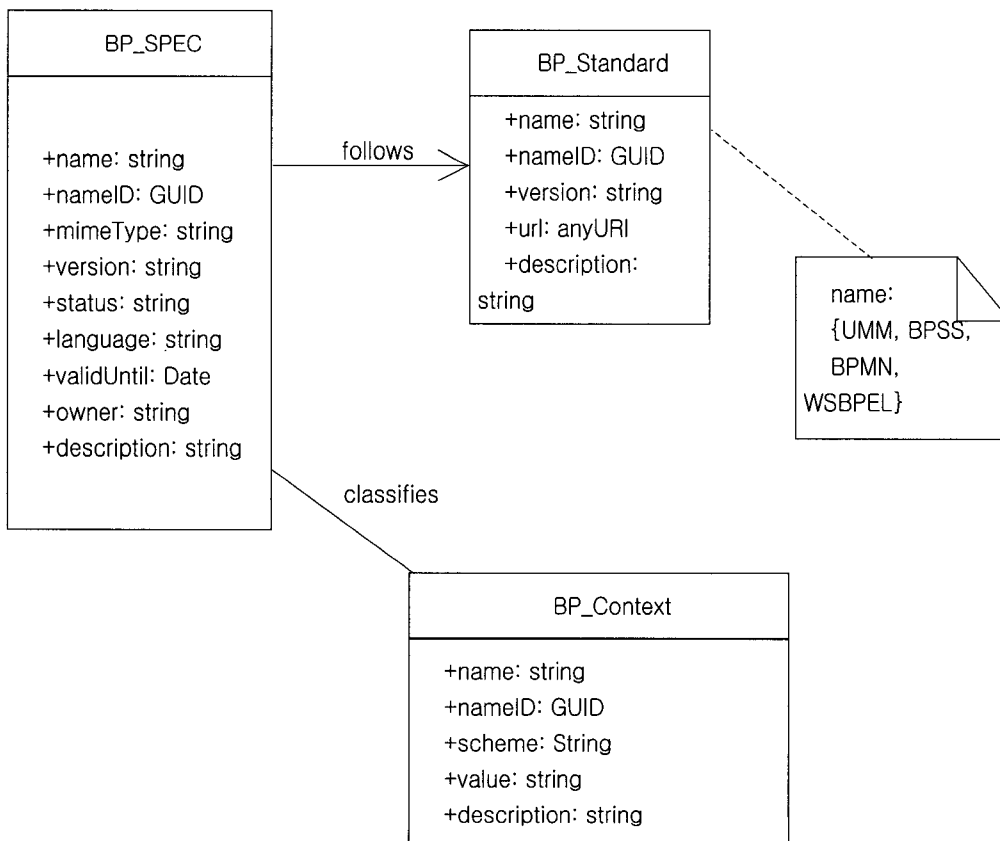
(그림 1) 비즈니스 프로세스 메타데이터 온톨로지 3계층

Metadata)는 BPSS, WSBPEL, BPMN, UMM (UN/CEFACT Modeling Methodology) 등과 같은 특정 명세 언어나 방법론에 따라서 기술이 가능한 메타데이터에 대한 부분이다. 마지막으로, 상호 연관관계에 관한 메타데이터(Interrelationship Metadata)는 작성된 비즈니스 프로세스 정의간의 상호관계를 규정하기 위한 메타데이터를 의미한다. 공통 메타데이터만이 의무사항(Mandatory)이며, 나머지는 모두 조건적으로 사용된다. 즉, 언어별 메타데이터나 상호 연관관계에 대한 메타데이터는 등록저장소 설계자가 선택적으로 사용할 수 있다. 이렇게 조건적으로 규정함으로써 언어별 메타데이터나 상호 연관관계에 관한 메타데이터를 추가로 작성하기 위한 노력의 정도를 등록저장소 설계자가 목적에 맞추어 적절하게 조정할 수 있도록 허용한다. 따라서 비즈니스 프로세스

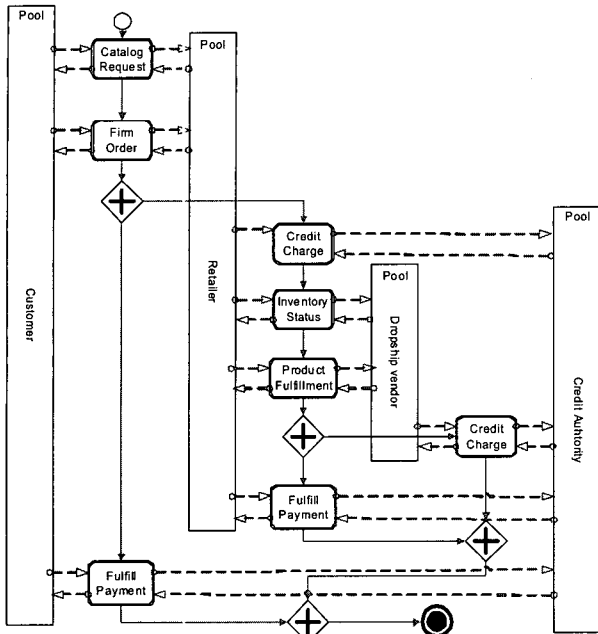
스 메타데이터 등록의 순서는 공통 메타데이터, 언어별 메타데이터, 상호 연관관계에 관한 메타데이터 순으로 이루어져야 한다.

### 3.2 공통 메타데이터

공통 메타데이터의 구성은 (그림 2)와 같다. 비즈니스 프로세스 전체에 대한 정보를 담고 있는 BP\_SPEC은 특정 비즈니스 프로세스 표현 표준(BP\_Standard)을 따르고, 분류 정보(BP\_Context)를 가지고 있다. 즉, BP\_SPEC 객체 인스턴스는 비즈니스 프로세스 인스턴스에 대한 개요적인 정보들을 포함하는데, 이에에는 비즈니스 프로세스의 이름(name), 프로세스 인스턴스 ID(nameID), 프로세스 인스턴스가 저장된 파일 타입(mimeType), 버전(version), 상태(status), 기술 언어(language), 유효일자(validUntil), 소유자(owner), 설명(description) 등을 포함한다. BP\_Standard 객체 인스턴스는 비즈니스 프로세스 표현 언어 표준을 의미하여, BPSS, WSBPEL 등이 대표적인 예가 된다. BP\_Standard에는 표준의 이름(name), ID(nameID), 버전(version), 표준을 참조할 수 있는 URL(url), 설명(description)이 포함된다. 비즈니스 프로세스 인스턴스는 여러 분류 체계 아래에서 분류될 수 있는데, BP\_Context에 이러한 정보들이 포함된다. BP\_Context 인스턴스는 이름(name), ID(nameID), 사용된 분류 체계(scheme), 분류 체계 내에서 범주(value), 설명을 가진다. 예를 들어, 국가코드 표준인 ISO3166 분류체계에서 한국으로 분류된 경우, scheme값이 ISO3166 분류체계에 대한 ID가 할당되고,



(그림 2) 공통 메타데이터의 구성



(그림 3) Dropship 예제의 BPMN 표현(출처 [6])

value에 'KR'이 할당된다.

제시된 메타데이터 온톨로지의 유용성을 보이기 위해서, 본 논문에서는 ebXML BPSS와 WSBPEL로 표현된 비즈니스 프로세스 정의의 예제를 사용하여 설명하도록 한다. ebXML BPSS로 표현된 비즈니스 프로세스 정의 예제로는 BPSS 기술 명세(버전 2.0.1)에서 사용한 DropShip 예제를 사용하도록 한다[6]. DropShip 예제에서는 고객(customer), 판매자(retailer), 벤더(vendor), 신용기관(credit authority)이 참여한다. DropShip 예제에서 고객이 주문을 하고, 주문처리(fulfillment)는 벤더가 담당한다. 신용기관은 지불을 확인하고, 지불을 처리한다. (그림 3)은 Dropship 비즈니스 프로세스에 대한 BPMN 표현이다. (그림 3)을 보면, 4개의 역할, Customer, Retailer, Dropship vendor, Credit Authority가 비즈니스 프로세스에 참여하며, 그들간에 일어나는 활동들이 어떤 순서로 발생하는지를 알 수 있다.

(그림 4)은 DropShip 예제의 비즈니스 프로세스 정의의 일부이다. (그림 4)의 ①②③④로 표시된 부분으로부터, 공통 메타데이터가 추출된다. 즉, ①번 박스로부터, BP\_SPEC의 owner 값과 버전 정보가 얻어지고, ②번 박스로부터 BP\_SPEC의 status, name, language 정보가 얻어진다. ③번 박스 내의 정보로부터 BP\_SPEC의 description 정보가 얻어진

<code>&lt;?xml version="1.0" encoding="UTF-8"?&gt;</code>		
<code>&lt;!--</code>		
Metadata: Owner: ebxmlbp (OASIS ebXML Business Process TC) Product: ebxmlbp (aka ebBP) Product Version: 2.0.1	①	
Artifact Type: Instance		
Stage: wd (Committee Draft Candidate) Descriptive Name: DropShip Revision: r07 Language: er (English)	②	
Form: xml (instance) Date: 20050721 (21 July 2005) Note: This is a conformant instance example only and is included in the ebBP v2.0.1 technical specification (Spec). -->		
<!-- Author J. Dean E.P. Hemopo <jdean@ebxml.co.nz> Researcher of ebXML, Massey University, Palmerston North, New Zealand (Aotearoa).		③
This ebBP schema-conformant v2.0.1 XML instance document is intended to portray ALL the UBL v1.0 documents in a mock trading relationship. This is only one interpretation of those documents in use. This XML instance document is from the perspective of a Dropship Vendor, in a Consumer/Dropship Vendor/Supplier(s) trade relationship. Stock is not held by the Dropship Vendor therefore JIT supply principles are applied.		
-->		
<ProcessSpecification xmlns="http://docs.oasis-open.org/ebxmlbp/ebbp-2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xi="http://www.w3.org/2001/XInclude" xsi:schemaLocation="http://docs.oasis-open.org/ebxmlbp/ebbp-2.0 http://docs.oasis-open.org/ebxmlbp/ebbp-2.0" name="SAMPLE" nameID="IDSample" uuid="urn:www.oasis-open.org/committees/ebxmlbp/UBLInstance/2.1" specificationVersion="2" instanceVersion="2">		④

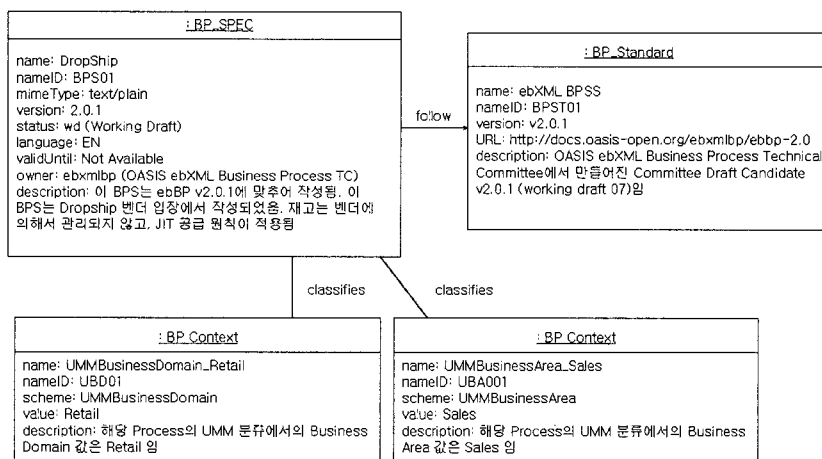
(그림 4) DropShip 예제 BPSS의 일부

다. (그림 5)는 DropShip 예제의 공통 메타데이터에 대한 UML 객체 다이어그램이다. (그림 5)의 BP\_SPEC 객체 인스턴스는 이름이 DropShip이고, ID는 BPS01, 파일의 mimeType은 text/plain이고, 버전은 2.0.1, 상태는 “working draft,” 기술 언어는 “EN(영어)”임을 알 수 있다. ④번 박스의 값은 BP\_Standard 내의 URL 속성값과 일치하는 BP\_Standard 인스턴스와 연결을 제공한다. 즉, 해당 비즈니스 프로세스 인스턴스는 ebXML BPSS를 따르고, 버전이 v2.0.1임을 (그림 5)의 BP\_Standard 객체 인스턴스를 통해서 알 수 있다. BP\_Context는 해당 비즈니스 프로세스의 분류값을 가지게 된다. (그림 5)에서는 예시적으로 두 개의 BP\_Context 객체 인스턴스가 해당 비즈니스 프로세스와 연결되어있다. 왼쪽의

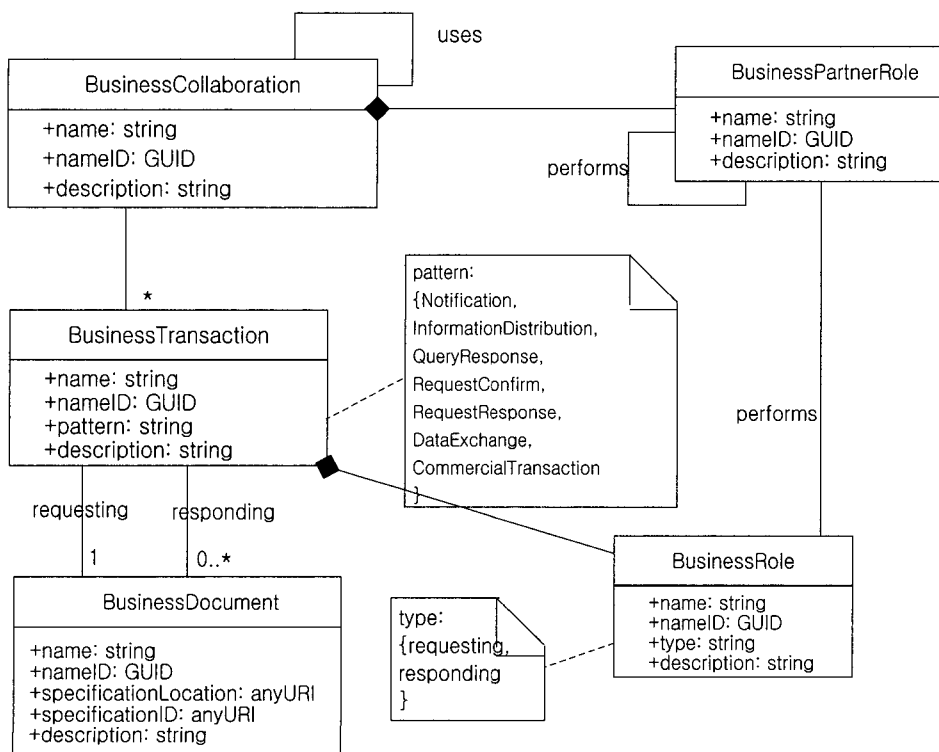
객체 인스턴스는 UMM의 분류체계에서 Business Domain 값으로 해당 비즈니스 프로세스가 “Retail” 값을 가짐을 보여 주고 있으며, 오른쪽의 BP\_Context 인스턴스는 해당 비즈니스 프로세스의 Business Area가 “Sales”임을 보여준다.

### 3.3 명세 언어별 메타데이터

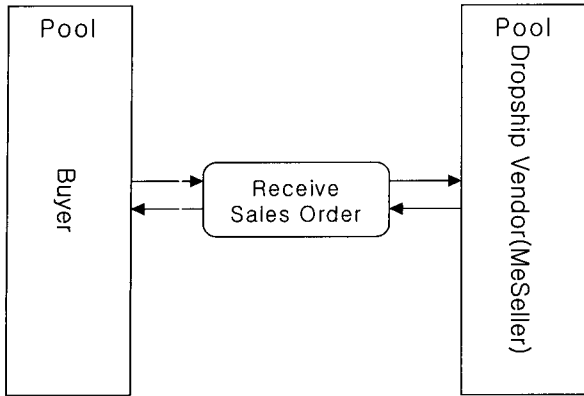
언어별 메타데이터는 명세 언어별로 각각 기술할 수 있는 메타데이터를 말한다. 본 연구에서는 ebXML BPSS, WSBPEL, BPMN, UMM, 4 가지 언어에 대한 메타데이터 온톨로지를 규정하였다(실제로 UMM은 모델링 언어가 아니라 방법론이고, 따라서 UMM을 따라서 작성된 산출물(UMM 워크시트)을 저장하기 위한 온톨로지를 의미한다). (그림 6)은 ebXML



(그림 5) DropShip 예제에서의 공통 메타데이터



(그림 6) ebXML BPSS를 위한 메타데이터의 구성

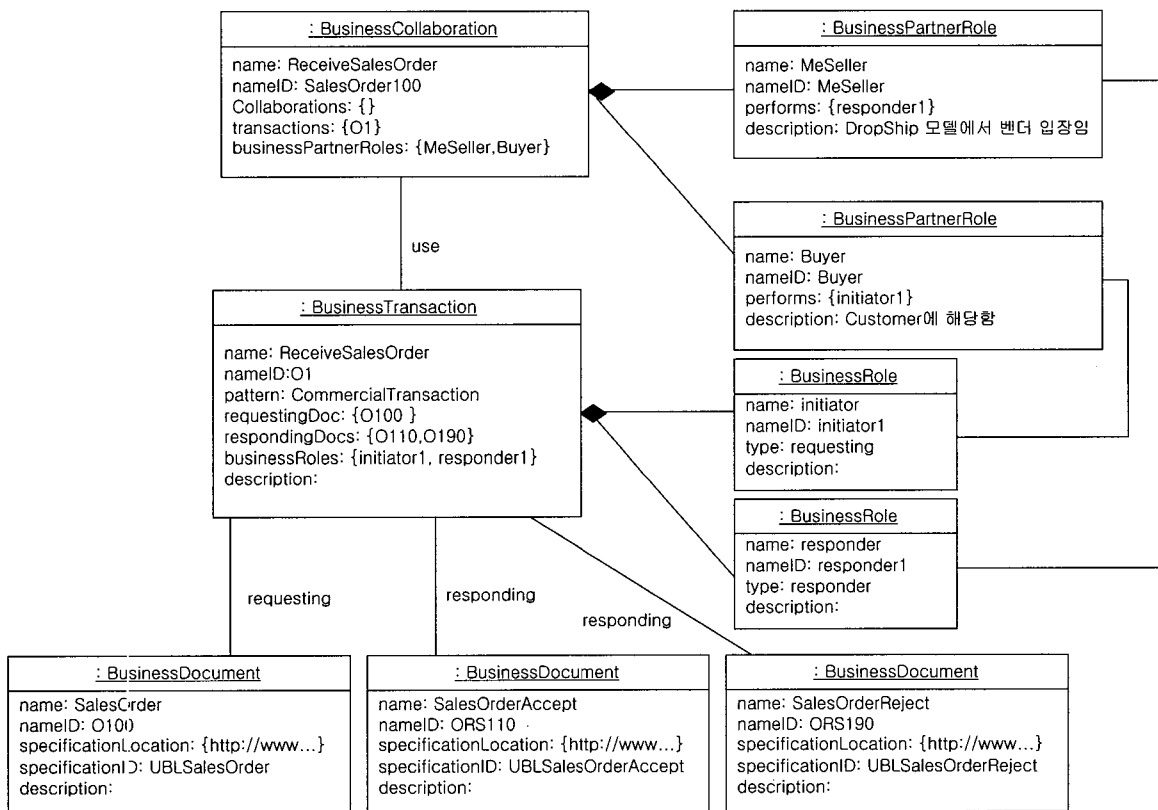


(그림 7) Receive Sales Order에 대한 BPMN 표현

BPSS로 작성된 비즈니스 프로세스를 위한 메타데이터 온톨로지이다. 비즈니스 협업(BusinessCollaboration), 비즈니스 트랜잭션(BusinessTransaction), 비즈니스 문서(BusinessDocument), 비즈니스 파트너 역할(BusinessPartnerRole), 비즈니스 역할(BusinessRole) 등 5개의 클래스로 구성된다. ebXML BPSS에서는 비즈니스 프로세스를 거래에 참여하는 참여자 등의 역할(Role)과 비즈니스 협업(Business Collaboration)으로 표현한다. 비즈니스 협업은 다시 업무의 순서를 정의하는 코레오그래피(Choreography)와 단위 업무를 나타내는 비즈니스 트랜잭션(Business Transaction), 비즈니스 트랜잭션 내의 문서 흐름(Document Flow)으로 이루어진다. 따라서, (그

림 6)의 각 클래스들은 이들에 대한 내용을 담기 위한 클래스들이다. ebXML BPSS에서 비즈니스 트랜잭션은 7가지 패턴으로 구분되며, 이 정보는 BusinessTransaction 내에 pattern 필드에 기술된다. 비즈니스 트랜잭션은 거래 요청시에 제공되는 문서와 응답으로 받는 문서를 가지는데, 이 정보가 BusinessTransaction과 BusinessDocument간의 연관관계인 'requesting'과 'responding'으로 기술된다.

(그림 7)은 DropShip 예제에 포함된 비즈니스 협업인 "Receive Sales Order"에 대한 BPMN 다이어그램이다. "Receive Sales Order" 비즈니스 협업을 중심으로 한 ebXML BPSS 메타데이터 객체 인스턴스들은 (그림 8)과 같다. (그림 8)에는 한 개의 BusinessCollaboration 객체 "ReceiveSalesOrder"와 연관된 두 개의 BusinessPartnerRole, "MeSeller", "Buyer"가 존재한다. 또한 BusinessTransaction "ReceiveSalesOrder"가 연관된다. 이 BusinessTransaction은 두 개의 BusinessRole, "initiator1", "responder1"과 연관되며, 하나의 requesting 문서인 "SalesOrder"와 두 개의 responding 문서인 "SalesOrderAccept", "SalesOrderReject"와 연관되어 있다. 또한 BusinessPartnerRole "MeSeller"와 "Buyer"는 각각 "responder1", "initiator1"의 BusinessRole을 수행한다. 주목할 것은 (그림 8)의 내용들이 XML 형태로 작성된 BPS (Business Process Specification)로부터 자동적으로 생성이 가능하다는 것이다. 예를 들어, (그림 8)의 "ReceiveSalesOrder" BusinessCollaboration의 경우, 각 속성값 name, nameID, collaborations, transactions, businessPartnerRole 등은 BPS의 해당 부분에서 추출이



(그림 8) ebXML BPSS 메타데이터 예제

```

<BusinessCollaboration name="Receive Sales Order" nameID="SalesOrder100" pattern="">
  <Role name="Buyer" nameID="Buyer100"/>
  <Role name="MeSeller" nameID="MeSeller100"/>
  <TimeToPerform duration="P2D"/>
  <Start name="Start Receive Sales Order" nameID="StartReceiveSalesOrder">
    <ToLink toBusinessStateRef="BTA100"/>
  </Start>
  <BusinessTransactionActivity name="BTA100" nameID="BTA100"
    businessTransactionRef="O1" hasLegalIntent="true">
    <TimeToPerform duration="PT24H"/>
    <Performs currentRoleRef="Buyer" performsRoleRef="initiator1"/>
    <Performs currentRoleRef="MeSeller" performsRoleRef="responder1"/>
  </BusinessTransactionActivity>
  .....
</BusinessCollaboration>
    
```

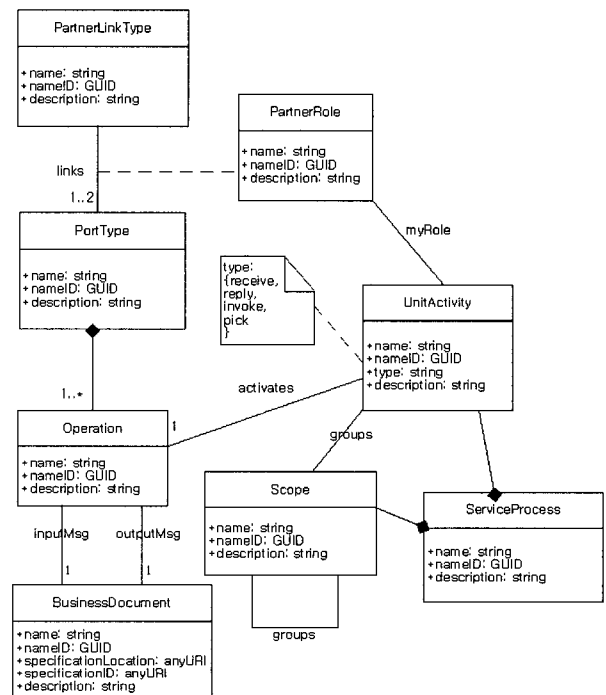
(그림 9) "Receive Sales Order"의 BPS 일부

가능하다. (그림 9)는 ebXML BPS의 Business Collaboration, "ReceiveSaleOrder"에 대한 부분 중 일부분이다. 그림에서 첫 번째 박스에서 BusinessCollaboration 객체의 name, nameID 정보를 추출할 수 있다. 두 번째, 세 번째 박스로부터 BusinessPartnerRoles를 추출할 수 있으며, 네번째 박스로부터 사용하는 BusinessTransaction의 nameID, "O1"을 추출할 수 있다. 또한 (그림 9)의 다섯 번째 박스에서 BusinessPartnerRole, Buyer, MeSeller와 BusinessRole, initiator1과 responder1의 연결을 찾을 수 있다. 비즈니스 트랜잭션 "O1"과 연관된 비즈니스 문서, 비즈니스 역할에 대한 정보들도 유사한 방법으로 XML로 표현된 BPS로부터 추출이 가능하다.

(그림 10)은 WSBPEL을 위한 메타데이터 온톨로지를 보여준다. WSBPEL을 위한 메타데이터는 PartnerLinkType, PortType, PartnerRole, UnitActivity, Operation, Scope, Service, BusinessDocument이라는 8개의 클래스를 사용하여 표현한다. WSBPEL에서는 웹서비스에서 제공되는 서비스들을 묶어서 새로운 복합 서비스인 프로세스로 정의할 수 있도록 지원한다. 이렇게 정의된 프로세스는 다른 웹서비스와 마찬가지로 WSDL의 PortType의 집합으로 표현된다. PortType은 외부 사용자에게 제공하는 오퍼레이션(Operation)을 요청할 수 있는 "출입구" 역할을 한다. 오퍼레이션은 입력 메시지와 출력 메시지를 가지는데, 이러한 메시지를 BusinessDocument로 표현한다. WSBPEL 프로세스의 각 단계들을 활동(activity)라고 부른다. 여러 개의 서비스를 연결해서 새로운 서비스를 만들기 위해서, WSBPEL 프로세스는 다른 서비스를 호출하거나, 클라이언트(서비스의 사용자)로부터의 요청을 받을 수 있도록 되어있다. 다른 서비스를 호출하기 위한 활동이 invoke이고, 클라이언트로부터 요청을 받고, 응답하기 위한 활동이 receive와 reply이다. 실제로 WSBPEL에서는 invoke, receive, reply, pick 이외에도, 한 곳의 데이터를 다른 곳으로 복사하는 assign, 오류처리를 위한 throw, 분기 및 선택적 실행을 위한 sequence, switch, while 등이 있으나, 직접적으로

operation과 관련된 활동만을 등록하기 위해서 UnitActivity라는 이름으로 객체를 정의하였다. 즉, 하나의 비즈니스 프로세스((그림 10)에서 ServiceProcess)는 다수의 UnitActivity들로 이루어진다.

WSBPEL로 표현된 비즈니스 프로세스 정의의 메타데이터 저장을 설명하기 위해서 WSBPEL 명세(버전 2.0)에서 사용한 주문처리 예제의 일부를 사용하도록 한다[7]. (그림 11), (그림 12)은 각각 주문처리 예제의 WSDL, BPEL 파일의 일부이다.



(그림 10) WSBPEL을 위한 메타데이터의 구성

```

.....
<message name="POMessage">
  <part name="customerInfo" type="sns:customerInfo"/>
  <part name="purchaseOrder" type="sns:purchaseOrder"/>
</message>
<message name="InvMessage">
  <part name="IVC" type="sns:Invoice"/>
</message>
.....

<!-- portTypes supported by the purchase order process -->
<portType name="purchaseOrderPT">
  <operation name="sendPurchaseOrder">
    <input message="pos:POMessage"/>
    <output message="pos:InvMessage"/>
    <fault name="cannotCompleteOrder"
      message="pos:orderFaultType"/>
  </operation>
</portType>
.....
<plnk:partnerLinkType name="purchasingLT">
  <plnk:role name="purchaseService">
    <plnk:portType name="pos:purchaseOrderPT"/>
  </plnk:role>
</plnk:partnerLinkType>
<plnk:partnerLinkType name="invoicingLT">
  <plnk:role name="invoiceService">
    <plnk:portType name="pos:computePricePT"/>
  </plnk:role>
  <plnk:role name="invoiceRequester">
    <plnk:portType name="pos:invoiceCallbackPT"/>
  </plnk:role>
</plnk:partnerLinkType>
.....
</definitions>

```

(그림 11) 주문처리 예제의 WSDL 파일의 일부

```

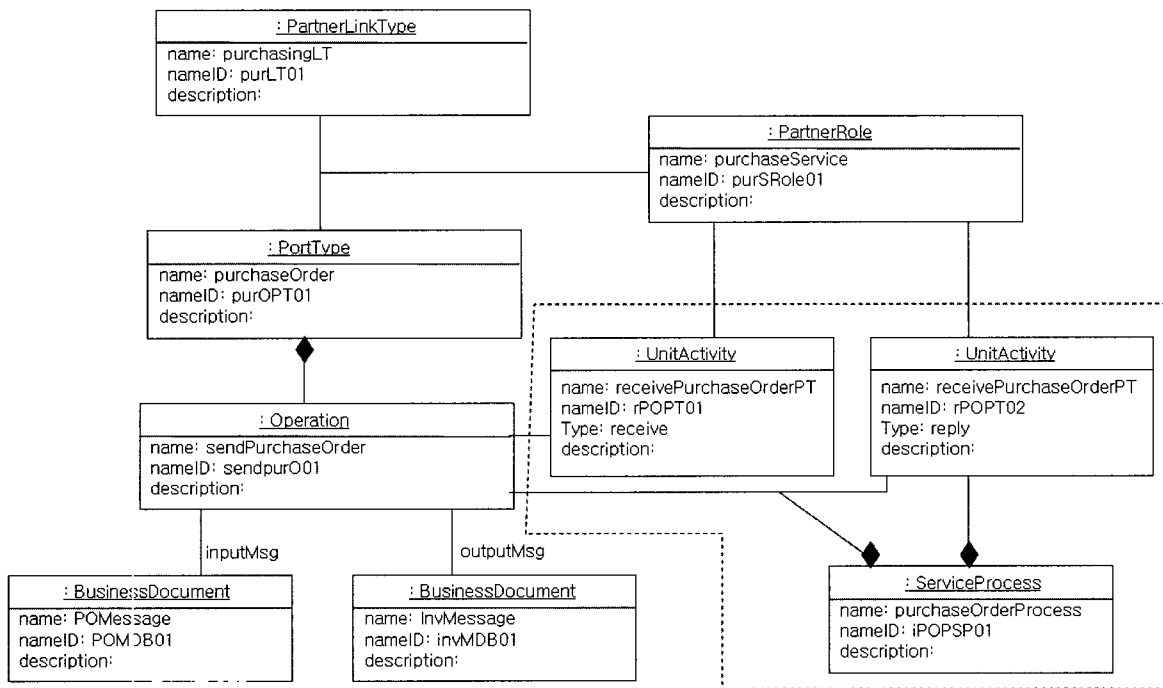
<process name="purchaseOrderProcess"
  targetNamespace="http://acme.com/ws-bp/purchase"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  xmlns:ins="http://manufacturing.org/wsd/purchase">

  <partnerLinks>
    <partnerLink name="purchasing"
      partnerLinkType="ins:purchasingLT"
      myRole="purchaseService"/>
    .....
  </partnerLinks>

  <sequence>
    <receive partnerLink="purchasing"
      portType="ins:purchaseOrderPT"
      operation="sendPurchaseOrder"
      variable="PO">
    </receive>
    <flow>
      <links>
        <link name="ship-to-invoice"/>
        <link name="ship-to-scheduling"/>
      </links>
      <sequence>
        <assign>
        .....
        </assign>
      </sequence>
    </flow>
    <reply partnerLink="purchasing"
      portType="ins:purchaseOrderPT"
      operation="sendPurchaseOrder"
      variable="Invoice"/>
    </reply>
  </sequence>
</process>

```

(그림 12) 주문처리 예제의 BPEL의 일부



(그림 13) WSBPEL 메타데이터 예제(점선 안은 BPEL 파일로부터 추출된 부분)



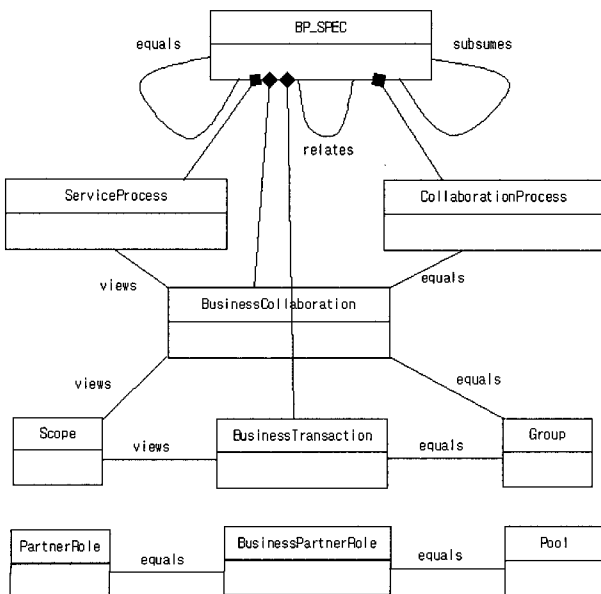
(그림 13)는 (그림 11)과 (그림 12)의 내용을 바탕으로 추출된 주문처리 예제의 메타데이터의 일부이다. WSDL 파일에는 메타데이터를 위한 클래스 중, PartnerLinkType, ParterRole, PortType, Operation, BusinessDocumentation에 대한 정보가 포함되어 있다. 예를 들어, (그림 11)의 <partnerLinkType> 엘리먼트 정보로부터, “purchasingLT”라는 이름의 PartnerLinkType 객체 인스턴스가 생성되고, “purchaseService”라는 이름의 PartnerRole이 생성된다. 또한 “purchasingLT” 객체와 PortType 객체 인스턴스인 “purchaseOrderPT”간의 관계가 생성된다. <portType> 엘리먼트의 정보로부터, PortType, operation 객체 인스턴스가 생성되고, 이들과 BusinessDocument 객체간의 관계가 생성된다. BusinessDocument의 상세한 내용은 <message> 엘리먼트 내에 포함되어 있다.

BPEL 파일 내에는 메타데이터 온톨로지의 UnitActivity, Scope, ServiceProcess 클래스에 해당하는 정보들을 포함하고 있다. (그림 13)의 점선 내의 객체 인스턴스들이 (그림 12)의 BPEL 파일에서 생성된 부분이다. <receive>와 <reply> 엘리먼트의 내용들로부터 두 개의 UnitActivity와 Operation “sendPurchaseOrder”와의 관계를 생성할 수 있다. 또한 <partnerLink> 엘리먼트 내용을 통해서 partnerLinkType “purchasingLT”와 해당 UnitActivity간의 관계들을 생성할 수 있다.

유사한 방법으로 BPMN, UMM로 표현된 비즈니스 프로세스를 위한 메타데이터 온톨로지가 정의되었다.

3.4 상호연관관계 메타데이터

B2B 등록저장소에 저장된 비즈니스 프로세스 정의들이 서로 연관관계를 가질 수 있다. 예를 들어, 동일한 비즈니스 프로세스에 대하여 ebXML BPSS로 표현한 비즈니스 프로세스 정의와 WSBPEL로 정의한 비즈니스 프로세스가 함께 등록저장소에 저장될 수 있다. 이러한 경우 이들 비즈니스 프로세스 정의들간의 관계, 또는 이들 비즈니스 프로세스 정의를 구성하는 구성요소간의 관계를 정의하여 관리할 필요가

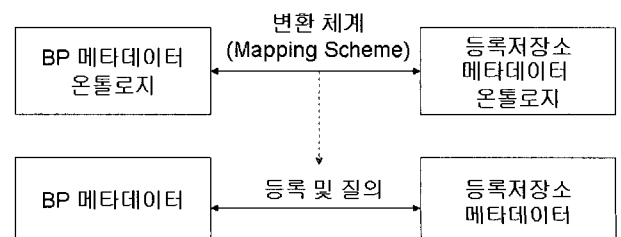


(그림 14) 상호연관관계 메타데이터의 구성

있다. B2B 등록저장소에 저장된 기업간 비즈니스 프로세스 명세간의 상호 연관 관계를 위한 메타데이터는 (그림 14)와 같이 정의되었다. 동일한 비즈니스 프로세스를 다른 표현 방식으로 표현한 경우 BP\_SPEC 인스턴스간의 “equals” 관계를 통해서, 이들이 동일한 프로세스에 대한 다른 표현임이 관리된다. 또한 비즈니스 프로세스 정의간에는 포함관계가 존재할 수 있는데, 이는 “subsumes” 관계를 통해서 관리된다. (그림 14)를 보면, WSBPEL에서의 “ServiceProcess”는 ebXML 비즈니스 협업의 “view”로 정의되어있다. 이것은 WSBPEL에서의 비즈니스 프로세스는 기업쌍방간의 협업 측면에서 비즈니스 프로세스를 표현하기 보다는 한 기업 입장에서의 비즈니스 프로세스를 중심으로 표현하기 때문이다. 따라서, WSBPEL에서의 비즈니스 프로세스(ServiceProcess)는 ebXML의 비즈니스 협업의 한 기업 입장에서의 관점인 것이다. ebXML의 비즈니스 트랜잭션은 ebXML의 비즈니스 프로세스를 표현하는 기본 단위로, 요청 활동(Requesting Activity)와 응답 활동(Responding Activity)으로 구성된다. 하지만, 이러한 단위가 WSBPEL에서는 존재하지 않는다. 그러나 필요에 따라서 이러한 활동들의 묶음을 Scope 형태로 정의할 수 있다. 또한 WSBPEL의 PartnerRole과 ebXML BPSS의 BusinessPartnerRole, 그리고 BPMN의 Pool이 동일한 의미를 가진다.

4. 메타데이터 온톨로지의 ebXML 등록저장소로의 매핑

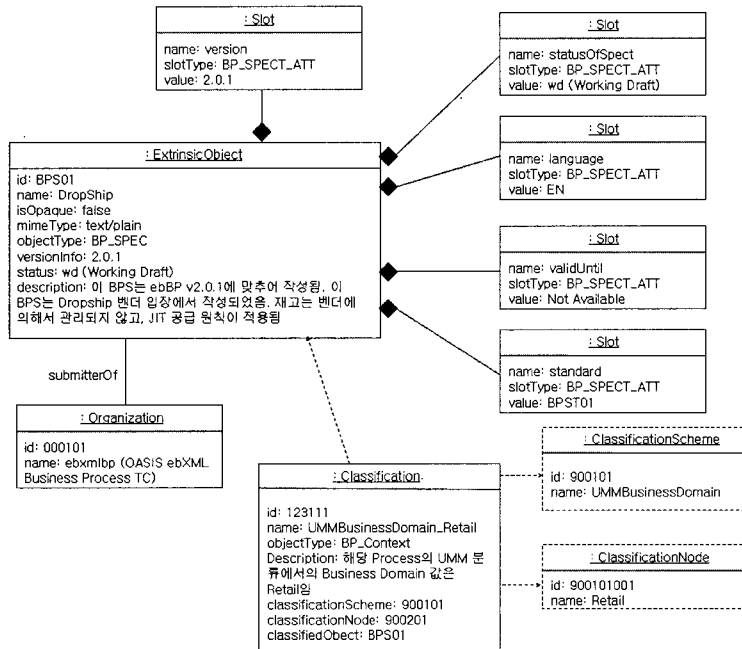
제시된 메타데이터 온톨로지가 일반적인 범용성을 갖추고 있지만, 제시된 메타데이터 온톨로지를 지원하는 B2B 등록저장소를 실제 활용가능한 시스템으로 구현하는 데는 많이 노력이 필요하다. 따라서, 제시한 메타데이터 온톨로지를 직접 구현하기 보다는 (그림 15)에서와 같이 실제 활용가능한 등록저장소 메타데이터 온톨로지로의 매핑을 통해서, 시스템을 구현하는 것이 더 현실적이다. 본 절에서는 ebXML 등록저장소의 메타데이터 온톨로지인 ebRIM (ebXML Registry Information Model)으로의 매핑을 제시하고자 한다. ebRIM은 RegistryObject, Classification, Slot 등의 클래스를 포함한다[2]. 변환 체계를 정의하는데 있어서 고려할 중요한 사항은 ebRIM에서 새로운 사용자 정의 클래스를 생성할 수 없다는 것이다. 따라서 RegistryObject 클래스의 objectType 속성을 이용하여 개별 객체를 특정한 클래스에 속함을 규정하고, 추가적으로 Slot 객체들을 생성하여 RegistryObject 객체의 클래스에 해당하는 속성들을 추가하였다.



(그림 15) 변환 체계의 역할

(그림 16)는 앞의 3.1.3절에서 제시되었던 DropShip 예제의 공통 메타데이터에 해당하는 BP\_SPEC 객체 인스턴스와 두 개의 BP\_Context 객체 인스턴스에 대한 매핑 결과이다 ((그림 5) 참조). 비즈니스 프로세스의 전체적인 메타 정보를 가진 BP\_SPEC은 1개의 ExtrinsicObject, 조직 정보(Organization), 5 개의 Slot으로 매핑된다. 예를 들어, 비즈니스 프로세스의 표현 언어를 나타내는 속성(BP\_SPEC 클

래스의 language 속성)은 'language' 명을 가지는 별도의 Slot으로 매핑되었다. 유사한 방법으로, 공통 메타데이터, 언어별 메타데이터, 상호연관관계 메타데이터 온톨로지를 구성하는 클래스 각각에 대하여 매핑 규칙이 정의되었다. 예를 들어, <표 1>은 BP\_SPEC 클래스에 대한 상세한 매핑 규칙이다.



(그림 16) 공통 메타데이터 매핑

<표 1> BP\_SPEC 관련 변환 규칙

B2B BP Metacata Ontology		eBRIM		설명
Class / Association	Attribute	Object Type	Attribute	
BP_SPEC		ExtrinsicObject		↔ 하나의 BP_SPEC 객체는 하나의 ExtrinsicObject 객체로 매핑 ↔ 이 객체의 objectType 속성 값은 'BP_SPEC'임
BP_SPEC	name	ExtrinsicObject	name	
BP_SPEC	nameID	ExtrinsicObject	id	
BP_SPEC	mimeType	ExtrinsicObject	mimeType	
BP_SPEC	version	Slot		↔ 하나의 Slot객체로 매핑 ↔ 이 객체의 slotType 속성값은 'BP_SPEC_ATT'임 ↔ 이 객체의 name 속성값은 'version'임
BP_SPEC	status	Slot		↔ 하나의 Slot객체로 매핑 ↔ 이 객체의 slotType 속성값은 'BP_SPEC_ATT'임 ↔ 이 객체의 name 속성값은 'statusOfSpec'임
BP_SPEC	language	Slot		↔ 하나의 Slot객체로 매핑 ↔ 이 객체의 slotType 속성값은 'BP_SPEC_ATT'임 ↔ 이 객체의 name 속성값은 'language'임
BP_SPEC	validUntil	Slot		↔ 하나의 Slot객체로 매핑 ↔ 이 객체의 slotType 속성값은 'BP_SPEC_ATT'임 ↔ 이 객체의 name 속성값은 'validUntil'임
BP_SPEC	owner	Organization		↔ 이 속성은 하나의 Organization 객체로 변환됨 ↔ 이 객체의 name은 owner 속성 값과 동일함 ↔ BP_SPEC 객체와 Organization 객체는 submitterOf 관계로 연결
BP_SPEC	description	ExtrinsicObject	description	
follows		Slot		↔ BP_SPEC 객체는 단 하나의 BP_Standard 객체를 준수하게 되는데, 이 관계는 Slot 객체로 표현 ↔ 이 객체의 slotType 속성값은 'BP_SPEC_ATT'임 ↔ 이 객체의 name 속성값은 'standard'임

## 5. 결 론

본 논문에서는 기업간 비즈니스 프로세스 등록저장소 구현을 위해서 필수적으로 필요한 비즈니스 프로세스 메타데이터 온톨로지를 제시하였다. 제시된 메타데이터 온톨로지는 ebXML BPSS, WSBPEL, BPMN, UMM으로 표현된 비즈니스 프로세스 정의를 포괄할 수 있도록 설계되었다. 비즈니스 프로세스 정의가 표현 형태와 무관하게 가지는 메타정보들을 별도로 추출하여, 공통 메타데이터로 정의하였고, 각 표현 방법에 종속적인 정보들을 구분하여 메타데이터로 정의하였다. 또한 비즈니스 프로세스 정의들간의 상호연관 관계를 정의하기 위한 메타데이터를 정의하였다.

이 논문에서 제시된 메타데이터 온톨로지와 ebRIM으로의 매핑 체계는 다양한 형태로 표현된 비즈니스 프로세스들을 단일 등록저장소에 저장하고 관리할 수 있는 기초를 제공하며, 기업간 비즈니스 프로세스 등록저장소를 구현하는데 있어서의 혼란을 크게 줄일 수 있다. 또한 등록저장소 내에서 비즈니스 프로세스의 재활용 단위를, 전체 프로세스 수준에서 더 세분화하여, 프로세스의 구성요소 수준에서의 재활용이 가능하도록 지원한다. 등록저장소 내에서 비즈니스 프로세스의 검색 시에 비즈니스 프로세스 내의 의미적 하위 개념들과 관련 문서 등을 활용하여 검색할 수 있도록 함으로써 의미적 검색을 위한 기반을 제공한다.

향후 연구방향으로는 ebXML 등록저장소로의 매핑 체계를 바탕으로 B2B 등록저장소를 구현하여 그 가능성을 구체적으로 검토하고, 등록저장소간의 비즈니스 프로세스 정보를 교환하거나 통합적 검색을 제공할 수 있도록, 메타데이터 온톨로지에 대한 표준화 작업을 추진하고 있다.

## 참 고 문 헌

- [1] A. Dogac, et al, "Enhancing ebXML Registries to Make them OWL Aware," Distributed and Parallel Database, Vol.18, pp.9-36, 2005.
- [2] OASIS, 'ebXML Registry Service and Protocols (Version 3.0)', OASIS, May, 2005.
- [3] OASIS, 'ebXML Registry Information Model (Version 3.0)', OASIS, May, 2005.
- [4] BPM, 'Business Process Modeling Notation (Version 1.0)', Business Process Management Initiative, May, 2004.
- [5] OASIS, 'Business Process Execution Language for Web Service (Version 1.1)', OASIS, May, 2003.
- [6] OASIS, 'ebXML Business Process Specification Schema Technical Specification (version 2.0.1)', OASIS, July, 2005.
- [7] OASIS, 'Web Service Business Process Execution Language (Version 2.0)', OASIS, August, 2005.
- [8] W3C, 'Web Services Choreography Interface (WSCI) 1.0', W3C, <http://www.w3.org/TR/wsci/>, 2002.
- [9] W3C, 'Web Services Choreography Description Language (WS-CDL) Version 1.0', W3C, 2005.
- [10] WfMC, 'XML Process Definition Language Version 1.0', WfMC, <http://www.wfmc.org/standards/XPDL.htm>, 2005.
- [11] B. Medjahed, B. Benatallah, A. Bouguettaya, A.H.H. Ngu, A.K. Elmagarmid, "Business-to-Business Interactions: Issues and Enabling Technologies," The VLDB Journal, Vol.12, pp.59-85, 2003.
- [12] H. Kim, "Conceptual Modeling and Specification Generation for B2B Business Process based on ebXML," SIGMOD Record, Vol.31, No.1, pp.37-42, March, 2002.
- [13] 김형도, 김종우, "UML 기반의 기업간 비즈니스 프로세스 명세 모델링," Journal of Information Technology Applications & Management, 제13권, 제4호, pp.71-88, 12월, 2006년.
- [14] UN/CEFACT, 'Business Process Analysis Worksheets & Guidelines (Version 1.0)', UN/CEFACT and OASIS, 2001.
- [15] UN/CEFACT, 'UN/CEFACT Modeling Methodology (UMM) User Guide', UN/CEFACT, 2003.
- [16] S.A. White, "Introduction to BPMN," IBM Corporation, ...
- [17] S. Weerawarana, and F. Curbera, "Business Process with BPEL4WS: Understanding BPEL4WS," IBM Corporation, 2002.
- [18] L. Zhang, T. Chao, H. Chang, J. Chung, "XML-Based Advanced UDDI Search Mechanism for B2B Integration," Electronic Commerce Research, Vol.3, No.1-2, pp.25-42, 2003.
- [19] OASIS, UDDI (Version 3.0.2), OASIS, 2004.
- [20] D. Bianchini, V.De Antonellis, B. Pernici, P. Plebani, "Ontology-based Methodology for e-service Discovery," Information Systems, Vol.31, pp.361-380, 2006.
- [21] A. Naumenko, S. Nikitin, V. Terziyan, J. Veijalainen, "Using UDDI for Publishing Metadata of the Semantic Web," Proceedings of IASW-2005, pp.141-159, 2005.
- [22] K. Lee, K. Lee, D. Lee, S. Lee, "Integrating Service Registries with OWL-S Ontologies," Lecture Notes in Computer Science, Vol.3683, pp.163-169, 2005.



**김종우**

e-mail : kjw@hanyang.ac.kr  
1989년 서울대학교 수학과 졸업(학사)  
1991년 한국과학기술원 경영과학과  
(공학석사)  
1995년 한국과학기술원 산업경영학과  
(공학박사)

1995년~1996년 한국과학기술원 첨단경영정보연구센터 Post Doc.  
1996년~2003년 충남대학교 통계학과 전임강사, 조교수, 부교수  
1999년~2000년 Visiting Scholar, University of Illinois at Urbana-Champaign

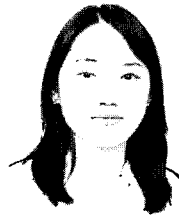
2003년~현재 한양대학교 경영학부 부교수  
관심분야: e-비즈니스, 상품/컨텐츠 추천기술, 데이터 마이닝, 비즈니스 프로세스 모델링과 통합 등



**김형도**

e-mail : hdkim@hycu.ac.kr  
1985년 서울대학교 산업공학과 (학사)  
1987년 한국과학기술원 경영과학과  
(석사)  
1992년 한국과학기술원 경영과학과  
(박사)

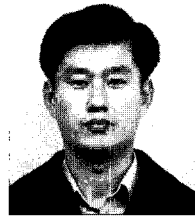
1993년~1999년 ㈜데이콤 EC인터넷 연구/기술 팀장  
2000년~2002년 아주대학교 정보통신전문대학원 교수  
2002년~2006년 ECIF 전자문서기술위원회 부위원장  
2004년~2006년 ebXML 전문위원회 위원장  
2006년~현재 한국전자거래진흥원 BP 워킹그룹 위원장  
2003년~현재 한양사이버대학교 경영학부 교수  
관심분야: XML, 전자상거래, 비즈니스 프로세스, IT 아키텍처, 데이터마이닝 등



**윤정희**

e-mail : yunjh@nia.or.kr  
1998년 KAIST 전산학과(학사)  
2000년 KAIST 전산학과(공학석사)  
2002년 KAIST 전산학과(박사수료)  
2002년~현재 한국정보사회진흥원  
ITA표준팀 선임연구원

관심분야: XML, 데이터베이스, e-비즈니스, 서비스 지향 아키텍처(SOA) 등



**정현철**

e-mail : junghc@nia.or.kr  
1990년 충남대학교 물리학과 졸업(학사)  
1996년 서울대학교 대학원 물리학과  
(이학석사)  
2000년~현재 한국정보사회진흥원  
선임연구원

관심분야: Web Technology(XML, Web Services, Semantic Web etc), e-Gov, CMMI, Project Management 등