

기업간 비즈니스 프로세스의 통합적 등록저장을 위한 메타데이터 스키마 설계

Metadata Schema Design for Integrated Registry of B2B Business Processes

김종우(Jong Woo Kim)*, 김형도(Hyoung Do Kim)**

초 록

B2B 등록저장소는 효율적인 기업간 전자거래의 지원을 위해서 거래 기업 또는 잠재적 거래 기업들이 거래 관계를 맺는데 필요한 정보들을 등록하고 검색할 수 있는 공간을 제공한다. 비즈니스 프로세스는 B2B 등록저장소의 주요한 콘텐츠 중에 하나인데, 비즈니스 프로세스가 가지는 복잡성과 동적인 특성으로 인해서 표현의 복잡성이 높다. 또한 현재 ebXML BPSS, WSBPEL, BPMN 등 다양한 표현 방식들이 경쟁하고 있다. 본 논문에서는 B2B 등록저장소에 다양한 표현 방식으로 표현된 비즈니스 프로세스 정의를 체계적으로 등록하기 위한 메타데이터 스키마를 제시한다. 제시된 메타데이터 스키마는 다양한 형식으로 표현된 비즈니스 프로세스 정의를 등록할 수 있도록 확장성을 가지며, 재활용의 수준을 전체 비즈니스 프로세스 수준에서 비즈니스 프로세스의 구성요소 수준으로 확대하도록 설계되었다. 제시된 메타데이터 스키마의 유용성을 확인하기 위하여, 대표적인 XML 기반의 비즈니스 프로세스 정의 언어인 ebXML BPSS와 WSBPEL로 표현된 비즈니스 프로세스의 메타데이터 추출을 예시적으로 설명한다.

ABSTRACT

B2B registries provide spaces to register and retrieve information which is necessary to support B2B transactions among business partners or potential business partners. Business process specifications are one of important contents in B2B registries, and there is high complexity of representation due to complex and dynamic characteristics of business processes. Also, currently there exist several competing specification frameworks such as ebXML BPSS, WSBPEL, BPMN, and so on. This paper proposes a metadata schema to register business process specifications which are represented by different specification frameworks. The proposed schema has extensibility to register business process specifications which are represented by various different specification frameworks. Also, it extends reuse level from whole business specification processes to their components. To show the usefulness of the proposed schema, this paper demonstrates metadata extraction from business process specifications which are represented by two representative XML-based business process specification languages, ebXML BPSS and WSBPEL.

키워드 : 비즈니스 프로세스 정의, B2B 등록저장소, 메타데이터 스키마
Business Process Specification, B2B Registry, Metadata Schema

* 한양대학교 경영학부

** 한양사이버대학교 경영학부

1. 서 론

B2B 등록저장소는 효율적인 기업간 전자거래의 지원을 위해서 거래 기업 또는 잠재적 거래 기업들이 거래 관계를 맺는데 필요한 정보들을 등록하고 검색할 수 있는 공간이다[4,9,10]. 등록저장소에는 거래 기업의 프로파일, 전자 문서 양식, 비즈니스 프로세스 정의, 서비스 정보 등이 저장된다. 이중 비즈니스 프로세스 정의는 비즈니스 프로세스가 가지는 복잡성과 동적인 특성으로 인해서, 표현의 복잡성이 높고 재활용이 용이하지 못한 한계를 가지고 있다. 등록저장소 내에 비즈니스 프로세스 저장과 관련해서 존재하는 또 다른 문제점 중에 하나는 비즈니스 프로세스 표현과 관련된 문제이다. 현재 비즈니스 프로세스를 표현하는 데 사용되는 표준으로는 ebXML BPSS (Business Process Specification Schema), WSBPEL (Web Service Business Process Execution Language), BPMN (Business Process Modeling Notation) 등이 다양하게 사용된다[3,6,11,12,16,17,18]. 또한 비즈니스 프로세스를 표현하는 수준도 ebXML BPSS나 WSBPEL 등은 XML 형태의 실행 수준(Executable Level)으로 상세하나, BPMN 이나 UMM (UN/CEFACT Modeling Methodology)의 워크시트들에서는 이 보다는 상위 수준에서 비즈니스 프로세스를 표현하고 다루고 있다.

비즈니스 프로세스 정의를 B2B 등록저장소에 저장하고 검색하기 위해서는 다양한 형태로 표현된, 또한 다양한 수준에서 표현된 비즈니스 프로세스를 어떻게 등록할 지에 대한 메타데이터 스키마의 정의가 우선적으로 필요하다[1]. 하지만, 현재 활용되고 있는 B2B 등록저장소에서는 이러한 비즈니스 프로세스 정의의 다양성과 복잡성을 반영한 메타데이터 스키마를 제공하고 있지 못하다. 본 논문에서는 B2B 등록저장소에 비즈니스 프로세스

정의를 체계적으로 저장하기 위한 메타데이터 스키마를 제시하고, 제시된 메타데이터 스키마의 활용가능성을 살펴보고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대한 검토로 B2B 비즈니스 프로세스의 표현 프레임워크에 대한 검토와 B2B 등록저장소에 대하여 살펴보도록 한다. 3장에서는 본 논문에서 제시하는 메타데이터 스키마의 설계 내용들과 ebXML BPSS, WSBPEL로 표현된 비즈니스 프로세스 정의의 등록을 예시적으로 살펴본다. 4장에서는 제시된 메타데이터 스키마와 관련된 토의사항들에 대하여 언급한다. 5장에서는 결론과 향후 연구 과제를 제시한다.

2. 관련 연구

2.1 B2B 비즈니스 프로세스 표현

기업간 비즈니스 프로세스는 기업간에 진행되는 전자상거래 처리절차를 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. 기업간 비즈니스 프로세스를 표현하는 방법에는 도식적인 표현 방법과 XML 기반의 표현 방법이 존재한다. 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[2,11,12,17,19]. XML 기반의 비즈니스 프로세스 기술 방법에는 ebXML의 BPSS (Business Process Specification Schema), 웹 서

비스 진영의 WSBPEL (Web Service Business Process Execution Language), WSCI (Web Service Choreography Interface), WS-CDL (Web Service Choreography Description Language) 등이 있다 [6,11,12,16,17,19].

이 중 XML 기반의 ebXML BPSS와 웹 서비스 진영의 WSBPEL이 실행가능한(Executable) 비즈니스 프로세스 기술 언어로 주로 활용되고 있다 [11,12]. 초기에 ebXML의 BPSS는 UN/CEFACT에서 작성되어 관리되었다. 2003년 8월 버전 1.0.9를 발표한 이후 OASIS 내에 새로 생성된 ebXML BP TC (Business Process Technical Committee)에서 규약을 발전시키고 유지하고 있다. 현재, 2006년 4월에 Committee 명세로 승인된 버전 2.0.3이 최근 버전이다. BPSS는 기본적으로 ebXML 프레임워크 내에서 ebXML 다른 표준들(ebXML MS (Message Service), ebXML CPPA (Collaboration Protocol Profile and Agreement), ebXML RIM (Registry Information Model), ebXML RS(Registry Services))과 연계되어 기업간 전자거래를 이루어 갈 때 비즈니스 트랜잭션에 대한 정의와 트랜잭션의 순서, 관계 및 제약 조건, 참여 기업들의 역할 등을 명시하고 있다 [5,9,10,11]. 버전 2.0에서부터는 웹 서비스로 구현된 비즈니스 트랜잭션을 전체 비즈니스 프로세스의 단위 업무로 수용할 수 있도록 제시하고 있다[11].

WSBPEL은 WSDL, UDDI, SOAP 기반의 웹 서비스 내에서 비즈니스 프로세스를 정의하고 실행하기 위해서 개발되었다[6,12]. BEA, IBM, Microsoft, SAP AG, Siebel Systems 등의 주도로 개발되었으며, Microsoft사에서 제시되었던 XLANG과 IBM에 의해 제시되었던 WSFL (Web Service Flow Language)을 결합한 비즈니스 프로세스 모델링 언어이다. 버전 1.1이 2003년 5월에 발표되었으며, 2.0 버전의 Committee 초안이 계속 보완되고 있으며 현재 참조 가능한 버전은 2006년 5월에 개발된 버

전이다. WSBPEL을 사용하여 WSDL을 통해서 명시된 웹 서비스 오퍼레이션들이 어떠한 순서로 진행되는지를 표현하고 실행할 수 있다. WSBPEL 이외에 비즈니스 프로세스 모델링과 관련해서 웹 서비스 진영에서 주목할 필요가 있는 표준으로 WS-CDL (Web Services Choreography Description Language)이 있다[17]. WS-CDL 규약은 W3C의 WS Choreography WG (Working Group)을 통해 작성되어 관리되고 있다. WS Choreography WG는 2006년 12월 WS-CDL 버전 1.0을 W3C Candidate Recommendation으로 발표하고 이에 대한 검토를 공개적으로 받고 있는 상태이다. WS-CDL은 비즈니스 프로세스 실행을 위한 언어는 아니며, 기술(description)을 위한 언어이다. 또한 WSBPEL에서는 기업간 거래에 참여하는 한 기업의 입장에서 비즈니스 프로세스를 표현하는데 반해서, WS-CDL에서는 거래 참여기업 중 한 기업의 관점이 아닌 중간적인 관점에서 거래에 참여하는 참가자들이 메시지를 주고받는 상호작용에 대한 처리 순서, 조건 및 제약 사항에 대해 기술할 수 있도록 해준다.

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 등록저장소라 한다. 이러한 등록저장소는 거래를 하려는 상대방들을 발견하고 이 상대의 비즈니스 정보를 얻을 수 있을 뿐만 아니라, 자신의 정보를 다른 잠재적 거래 상대에게 알릴 수 있는 장소가 되기 때문에 기업간 전자상거래에서 핵심적인 부분이다.

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[9,10]. 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[6]. UDDI는 전세계 서비스에 대한 명세, 발견, 통합 등에 대한 정보를 담고 있는 서비스 등록 저장소로, 현재 웹 서비스의 등록저장소 표준으로 사용되고 있다.

B2B 등록저장소 내의 주요한 콘텐츠 중의 하나인 비즈니스 프로세스 정의(Business Process Specification, BPS)의 저장과 관련하여 현재 다음과 같은 한계점이 있다. REMKO[13]와 같이 현재 사용하고 있는 B2B 등록저장소에서는 저장의 단위가 전체 비즈니스 프로세스로 되어 있으며, 비즈니스 프로세스 내의 구성요소 수준의 재활용을 제공하고 있지 못하다. 또한 다양한 비즈니스 프로세스 표현에 대한 고려가 되어 있지 않아서, 비즈니스 프로세스 등록 시에 등록자에 의해서 제공되는 메타 데이터(비즈니스 프로세스 명, 등록날짜, 등록자, 분류 정보 정도) 이외에 비즈니스 프로세스 내에 포함된 의미나 내용을 기반으로 한 비즈니스 프로세스 검색 기능이 구현되어 있지 못하다.

3. 비즈니스 프로세스 메타데이터 스키마

3.1 설계 원칙

본 연구에서 제시된 B2B 비즈니스 프로세스 등

록저장소 메타데이터 스키마를 설계하는데 있어서의 주요 설계 원칙은 다음과 같다.

- 비즈니스 프로세스 재사용의 단위를 비즈니스 프로세스뿐만 아니라 비즈니스 프로세스 구성요소 수준으로 한다.

비즈니스 프로세스 정의가 역할, 문서 흐름, 단위 업무, 업무 흐름 등을 포함하는 복합적인 객체이므로, 재활용의 수준을 비즈니스 프로세스의 구성요소 수준으로 상세화함으로써 등록저장된 내용의 재사용을 증진할 수 있다. 예를 들어, ebXML BPSS의 경우, BPSS로 기술된 비즈니스 프로세스 정의뿐만 아니라, 비즈니스 프로세스 내에 포함되어 있는 비즈니스 트랜잭션, 비즈니스 협업 단위의 재사용이 가능하다.

- 메타데이터 스키마를 특정 표현 프레임워크에 독립적으로 설계한다.

다양한 표현 프레임워크로 표현된 비즈니스 프로세스를 등록, 저장하기 위해서는 메타데이터 스키마가 특정 표현 프레임워크에 독립적인 형태를 가지는 것이 바람직하다. 또한 메타데이터 스키마 자체는 최대한 단순한 형태가 되도록 한다.

- 등록 대상에 대한 확장 가능성을 제공한다.

등록저장소에 저장 대상을 비즈니스 프로세스 구성요소 수준으로 상세화할 뿐만 아니라, 등록 대상이 되는 구성요소를 등록저장소 설계자가 결정할 수 있도록 한다. 이것은 등록저장소의 목적이 따라서, 재활용의 수준이 달라질 수 있도록 하기 위함이다.

- ebXML, 웹서비스 진영의 비즈니스 프로세스를 모두 등록저장할 수 있도록 지원한다.

현재 가장 많이 사용되고 있는 ebXML BPSS와 웹서비스 진영의 WSBPEL로 표현된 비즈니스 프로세스 정의를 등록, 저장할 수 있도록 지원한다. 또한 ebXML 진영에서 비즈니스 프로세스 분석, 설계 방법론으로 추천하는 UMM 워크시트로 표

현된 정보들도 등록저장할 수 있도록 지원하며, 최근 비즈니스 프로세스의 도식적 표현 방법으로 많이 활용되고 있는 BPMN도 지원할 수 있도록 한다.

- 타 프레임워크 내에서 등록저장된 구성요소의 연관성을 지원한다.

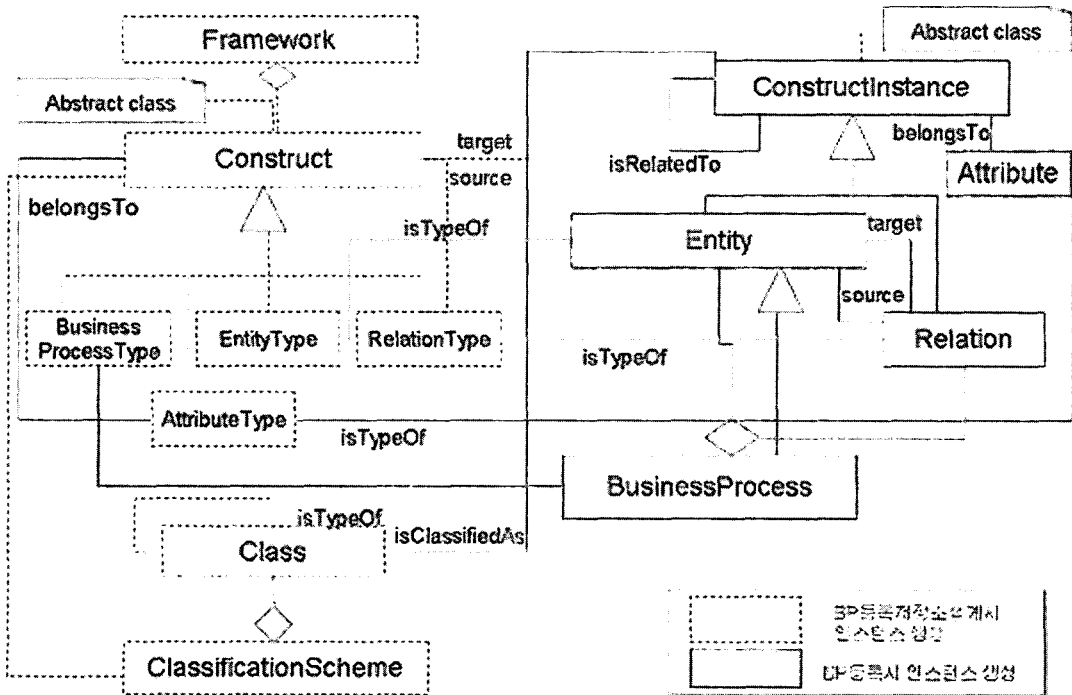
동일한 비즈니스 프로세스가 다른 표현 프레임워크(예를 들어, BPSS, BPMN)에서 표현된 경우, 동일한 비즈니스 프로세스임을 표현할 수 있도록 하고 이들 연관 관계가 등록저장소에서 관리될 수 있도록 한다. 즉, 등록저장소에서 비즈니스 프로세스 검색 시에 서로간의 연결을 제공한다.

3.2 메타데이터 스키마의 설계

본 연구에서 제시된 메타데이터 스키마는 <그림 1>과 같다. 메타데이터 스키마의 클래스는 크게 등

록저장소 설계 시에 인스턴스가 생성되는 클래스 (<그림 1>에서 점선으로 표시)와 개별 비즈니스 프로세스가 등록 시에 인스턴스가 생성되는 객체 (<그림 1>에서 실선으로 표시)로 구분된다. 등록저장소 설계 시에 비즈니스 프로세스 표현 프레임워크(Framework) 내에 비즈니스 프로세스를 표현하기 위한 구성요소(Construct)들은 Business ProcessType, EntityType, RelationType, AttributeType으로 등록된다. 예를 들어 ebXML BPSS의 경우, 비즈니스 프로세스를 구성하는 비즈니스 트랜잭션, 비즈니스 협업 등이 EntityType으로 정의되고 이들간의 관계가 RelationType으로 정의되고 비즈니스 프로세스 구성요소들을 설명하는 속성들이 AttributeType으로 등록된다. 구체적인 예제는 3.3절과 3.4절에서 제시하도록 한다.

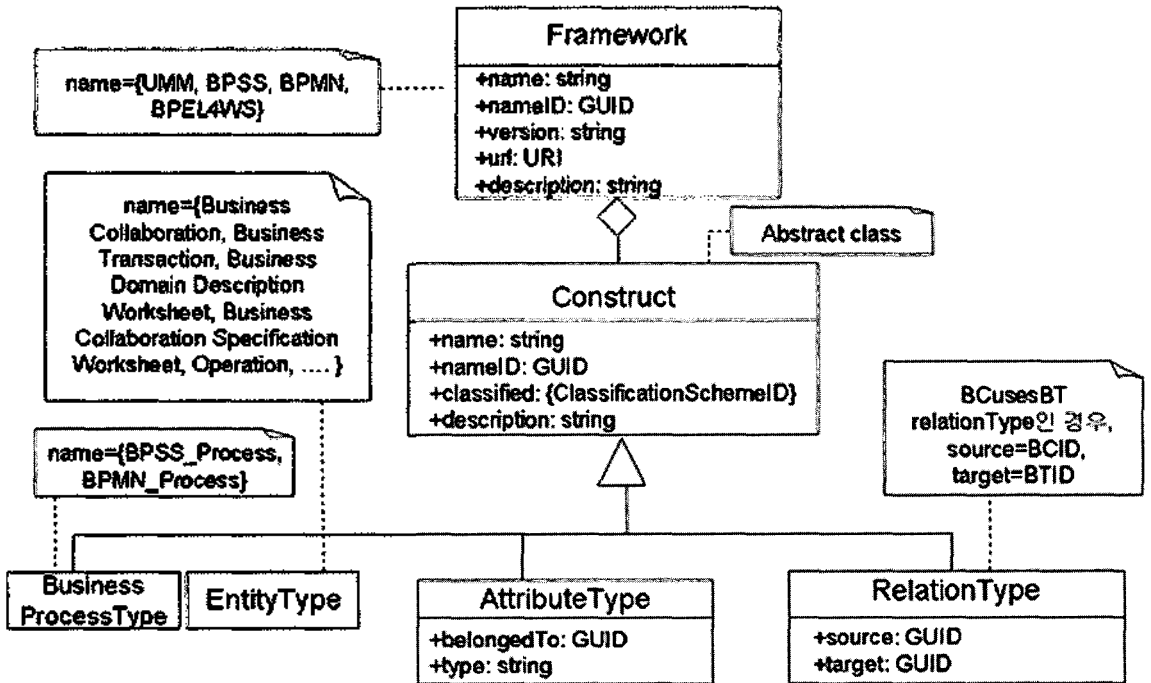
등록저장소 설계 시점에서 등록해야 되는 클레



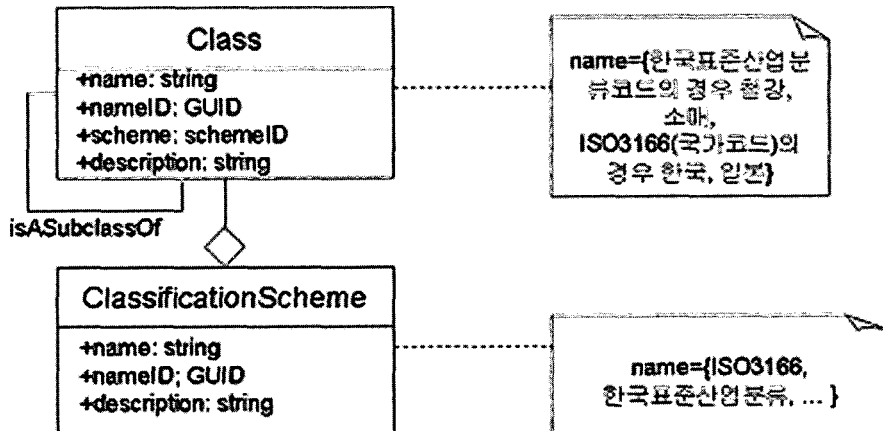
<그림 1> 메타데이터 스키마 클래스 다이어그램

스 중에는 ClassificationScheme과 Class가 포함된다. ClassificationScheme은 비즈니스 프로세스 Entity, Relation, Attribute를 분류하기 위한 분류체계를 의미한다. Class는 이러한 분류체계 내에 포함되는

분류 노드들을 의미한다. <그림 2>와 <그림 3>에서는 등록저장소 설계 시점 클래스의 속성들과 속성의 데이터 타입에 대한 상세한 정보를 보여준다. 등록저장소 설계 시점에서 등록된 비즈니스 프



<그림 2> 등록저장소 설계 시점 클래스(1)



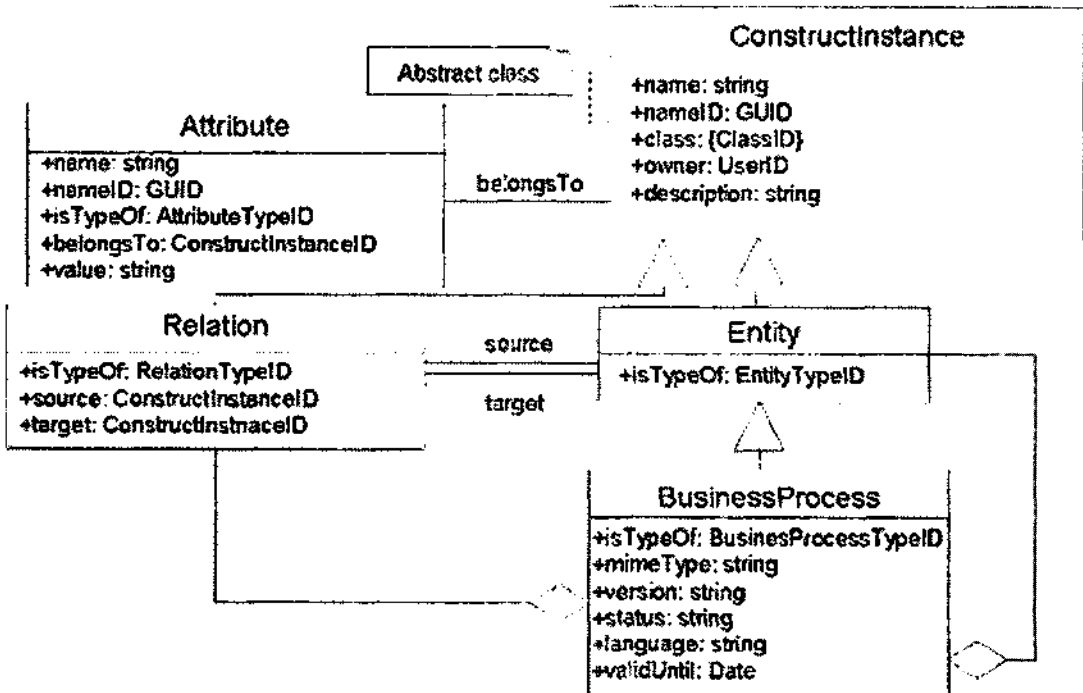
<그림 3> 등록저장소 설계 시점 클래스(2)

로세스 표현 프레임워크 내의 비즈니스 프로세스와 그 구성요소들은 추후 개별 비즈니스 프로세스 인스턴스 등록 시에 등록저장소에 등록된다. 개별 프로세스 인스턴스 등록 시에 사용되는 클래스들은 BusinessProcess, Entity, Relation, Attribute 등이다 (<그림 1>, <그림 4> 참조). 즉, 하나의 비즈니스 프로세스 인스턴스를 등록하면, 하나의 BusinessProcess 인스턴스가 생성되고 이 프로세스를 구성하는 구성요소들인 Entity, Relation, Attribute 인스턴스들이 저장된다. <그림 4>에는 프로세스 등록 시점에 인스턴스들이 생성되는 클래스들의 속성들을 보여준다.

3.3 ebXML BPSS 비즈니스 프로세스의 등록

3.3.1 등록저장소 설계 시점 클래스 등록

BPSS를 구성하고 있는 주요 개념은, 거래에 참여하는 참여자들의 역할(Role)과 비즈니스 협업(Business Collaboration, 이하 BC)을 포함한다. BC는 다시 업무들의 순서를 정의하는 코레오그래피(Choreography)와 단위 업무를 나타내는 비즈니스 트랜잭션(Business Transaction, 이하 BT), BT 내의 문서 흐름(Document Flow)으로 이루어진다. 코레오그래프는 BC 내에서의 BT 또는 BC들의 실행 순서를 표현하며, UML 또는 BPMN으로 표현될 수 있다. BT는 6가지 패턴 중의 하나를 가지게 된다. 이 6가지에는 Commercial Transaction, Information Distribution, Notification, Query/Response,



<그림 4> 비즈니스 프로세스 등록 시점 클래스

Request/Confirm, Request/Response가 포함된다[11].

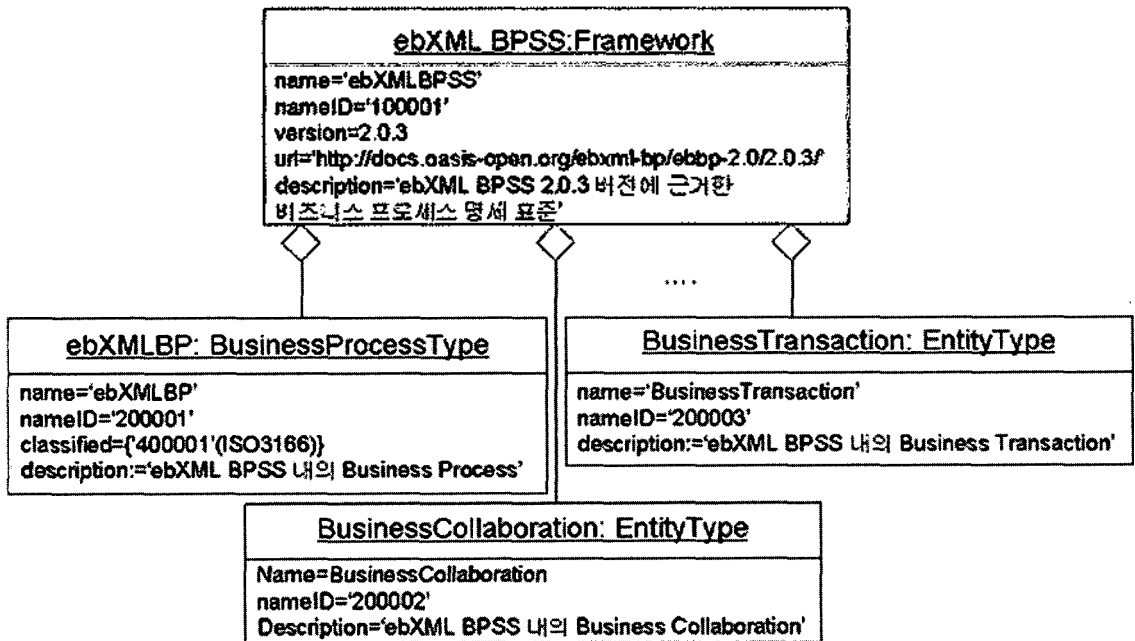
〈그림 5〉는 등록저장소 설계 시점에 BPSS 프레임워크를 등록하기 위해서 생성된 객체 인스턴스의 일부이다. "ebXML BPSS"는 Framework 클래스의 인스턴스로 정의가 되었고 ebXML BPSS로 정의된 비즈니스 프로세스를 추후 등록하기 위한 BusinessProcess 클래스와 비즈니스 프로세스를 구성하는 요소인 BusinessTransaction과 BusinessCollaboration이 EntityType으로 정의되었다. 〈그림 6〉도 등록저장소 설계 시점에 생성되는 객체 인스턴스의 일부를 보여준다. 〈그림 6〉에서는 BusinessTransaction과 BusinessCollaboration의 관계인 "BCUsesBT"가 RelationType 클래스의 인스턴스로 정의되었다. 또한 BusinessTransaction의 속성으로 BusinessTransactionPattern이 정의되는데, 이 BusinessTransactionPattern이 가지는 값은 앞에서 언급한 BT 패턴명을 가지게 된다. 예시적으로 ebXML BPSS를 위한 등록저장소 설계 시점 생성

객체들은 전체 목록은 〈표 1〉과 같다.

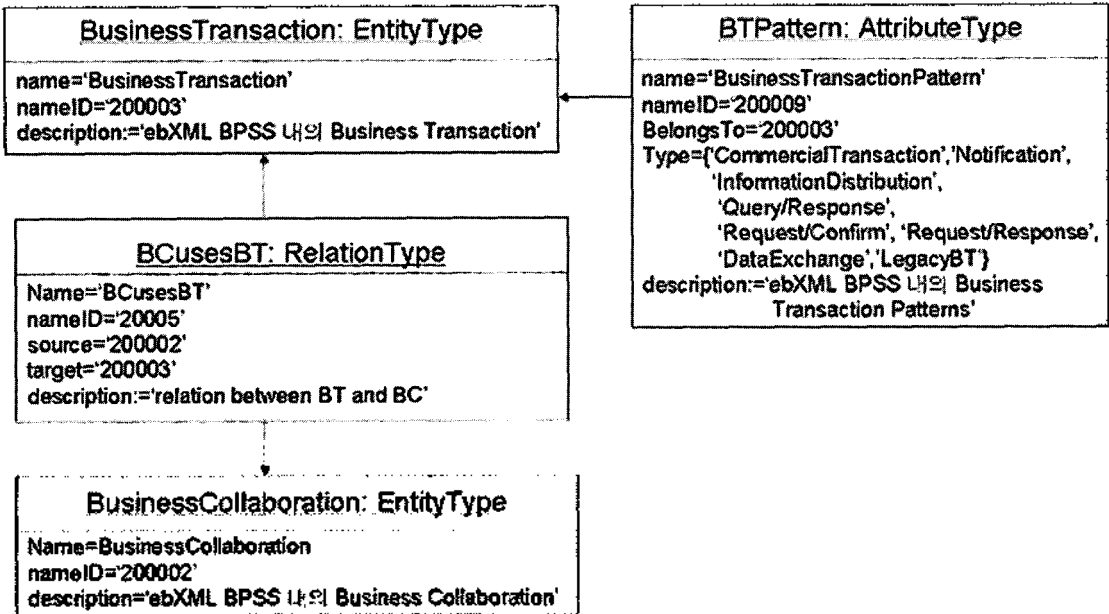
3.3.2 비즈니스 프로세스 등록 시점 클래스 등록

비즈니스 프로세스 등록 시점에는 BusinessProcess, Entity, Relation, Attribute 클래스의 인스턴스들이 생성된다. Entity 인스턴스가 가지는 isTypeOf 속성값은 EntityType 인스턴스인 BusinessTransaction, BusinessCollaboration, BusinessDocument의 nameID(〈표 1〉의 경우, '200002', '200003', '200004' 중 하나)이고 Relation 인스턴스가 가지는 isTypeOf 속성값은 RelationType 인스턴스인 BPIncludesBC, BCUsesBT, requesting, responding의 nameID이다.

ebXML로 표현된 비즈니스 프로세스의 등록 예제로 ebXML BPSS 표준 기술 명세(버전 2.0.3)에서 제시된 Dropship 예제를 사용하도록 한다[11]. DropShip 예제에서는 고객(customer), 판매자



〈그림 5〉 BPSS 등록저장소 설계 시점 클래스(1)



〈그림 6〉 BPSS 등록저장소 설계 시점 클래스(2)

〈표 1〉 BPSS 등록을 위한 초기 객체 생성

Type	nameID	name	description
BP	200001	BusinessProcess	ebXML BPSS 내의 Business Process
Entity	200002	BusinessCollaboration	ebXML BPSS 내의 Business Collaboration
Entity	200003	BusinessTransaction	ebXML BPSS 내의 Business Transaction
Entity	200004	BusinessDocument	ebXML BPSS 내의 Business Document
Relation	200005	BPIncludesBC	ebXML BPSS 내의 Business Process 내의 Business Collaboration 포함관계
Relation	200006	BCUsesBT	ebXML BPSS 내의 Business Collaboration 내에서 Business Transaction 사용 관계
Relation	200007	requesting	ebXML BPSS 내의 Business Transaction 에서의 requesting document 관계
Relation	200008	responding	ebXML BPSS 내의 Transaction 내에서 responding document 관계
Attribute	200009	BTPattern	Business Transaction Pattern. 'Commercial Transaction', 'Notification', 'Information Distribution', 'Query/Response', 'Request/Confirm', 'Request/Response', 'DataExchange', 'LegacyBT' 중에 한 값을 가짐

(retailer), 벤더(vendor), 신용기관(credit authority)이 참여한다. DropShip 예제에서는 고객이 주문을 하고, 주문처리(fulfillment)는 벤더가 담당한다. 신용기관은 지불을 확인하고, 지불을 처리한다. <그림 7>, <그림 8>은 Dropship 예제의 비즈니스 프로세스 정의의 일부이다. <그림 7>을 살펴보면, 비즈니스 프

로세스의 이름은 "PurchaseCluster"로 정의되어있고 버전은 "1"로 정의되어있다. <그림 7>에는 두 개의 비즈니스 문서인 "Invoice"와 "InvoiceResponse"에 대한 정의가 함께 포함되어있다. <그림 7> 중간 부분에는 "Data:Invoice"라는 이름의 DataExchange 엘리먼트가 보이는데, 이는 비즈니스 트랜잭션이며, 이

```

<ProcessSpecification
xmlns="http://docs.oasis-open.org/ebxml-bp/ebbp-2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xi="http://www.w3.org/2001/XInclude"
name="PurchasingCluster" nameID="PC23"
uuid="urn:purchasingcluster" specificationVersion="2"
instanceVersion="1">
<!--xi:include href="signals-package-2.0.3.xml" parse="xml"
xpointer="xpointer(/ProcessSpecification/Package[1])"/-->
<BusinessDocument name="Invoice" nameID="bd-invoice">
<!--Shows use of externalDocumentDefRef optional attribute-->
<Specification type="schema"
location="ubl-1.0-SBS-cs/xpaths/xml/XPath/Invoice-XPath.xml"
targetNamespace="urn:oasis:names:specification:ubl:schema:xsd:Invoice-1.0"
name="Invoice" nameID="invoice32"
externalDocumentDefRef="urn:oasis:names:tc:ubl:xpath:Invoice-1.0:sbs-1.0"/>
</BusinessDocument>
<BusinessDocument name="InvoiceResponse"
nameID="bd-invoiceResponse">
<Specification type="schema"
location="http://purchasingcluster.com/InvoiceResponse.xsd"
name="InvoiceResponse" nameID="invoice33"/>
</BusinessDocument>
<DataExchange name="Data:Invoice" nameID="data-invoice">
<RequestingRole name="DInitiator" nameID="DInitiator1"/>
<RespondingRole name="DResponder" nameID="DResponder1"/>
<RequestingBusinessActivity name="ReqBA:SendInvoice"
nameID="debareq-invoice"
timeToAcknowledgeReceipt="PT6H"
timeToAcknowledgeAcceptance="PT12H">
<DocumentEnvelope name="DE:ProcessInvoice"
nameID="data-de-invoice" businessDocumentRef="bd-invoice"/>
</RequestingBusinessActivity>
<RespondingBusinessActivity name="ResBA:ReceiveInvoice"
nameID="debares-invoice">
<DocumentEnvelope name="DE:ProcessInvoiceResponse"
nameID="data-de-invoiceResponse"
businessDocumentRef="bd-invoiceResponse"/>
</RespondingBusinessActivity>
</DataExchange>
.....
</ProcessSpecification>
    
```

<그림 7> Dropship 예제의 비즈니스 프로세스 정의

비즈니스 트랜잭션의 패턴이 DataExchange이다.
 "Data:Invoice" 비즈니스 트랜잭션은 Requesting

Activity에서 "Invoice" 문서를 보내고, Responding
 Activity에서는 "InvoiceResponse" 문서를 받는 것으

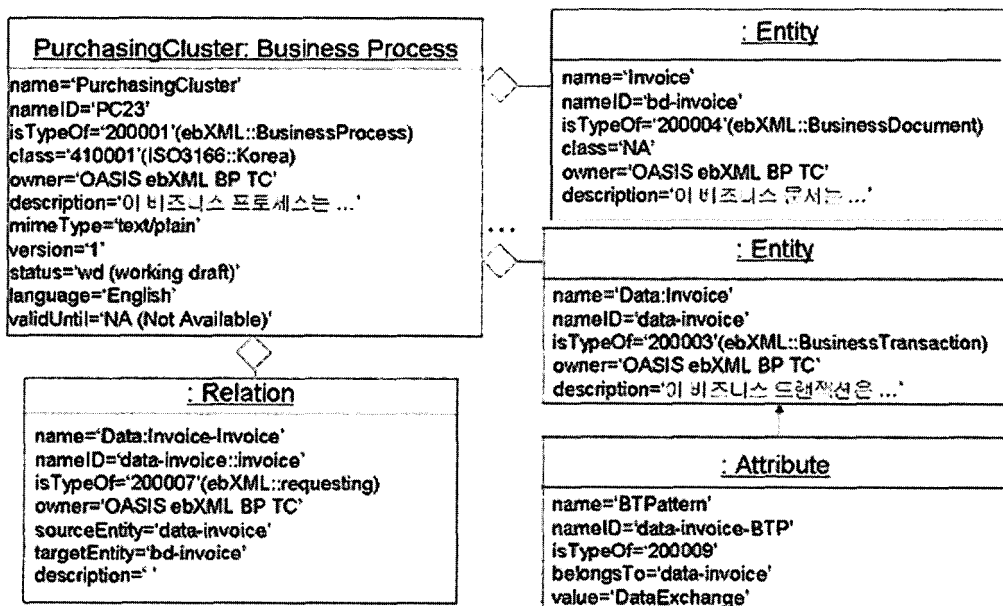
```

<!--...-->
<QueryResponse name="Catalog Request" nameID="ID100" isGuaranteedDeliveryRequired="false">
  <RequestingRole name="QRinitiator" nameID="QRinitiator1"/>
  <RespondingRole name="QRresponder" nameID="QRresponder1"/>
  <RequestingBusinessActivity name="requestCatalog" nameID="ID101">
    <DocumentEnvelope name="Catalog Request" nameID="ID102" businessDocumentRef="ID100"/>
  </RequestingBusinessActivity>
  <RespondingBusinessActivity name="sendCatalog" nameID="ID103">
    <DocumentEnvelope name="Catalog Response" nameID="ID104" isPositiveResponse="true"
      businessDocumentRef="ID1001"/>
  </RespondingBusinessActivity>
</QueryResponse>

<BusinessCollaboration name="BC" nameID="BC100">
  <Role name="Buyer" nameID="ID7902847"/>
  <Role name="Supplier" nameID="ID7902028"/>
  <TimeToPerform duration="P1D" type="design"></TimeToPerform>
  <BusinessTransactionActivity name="Catalog Request" nameID="ID100300"
    businessTransactionRef="ID100" hasLegalIntent="false">
    <TimeToPerform duration="P1D"/>
    <Performs currentRoleRef="ID7902847" performsRoleRef="QRinitiator1"/>
    <Performs currentRoleRef="ID7902028" performsRoleRef="QRresponder1"/>
  </BusinessTransactionActivity>
  <!-- start and completion omitted -->
</BusinessCollaboration>

<!--...-->
    
```

<그림 8> Catalog Request 비즈니스 협업 예제



<그림 9> BPSS 비즈니스 프로세스 등록 시점 객체 예제(1)

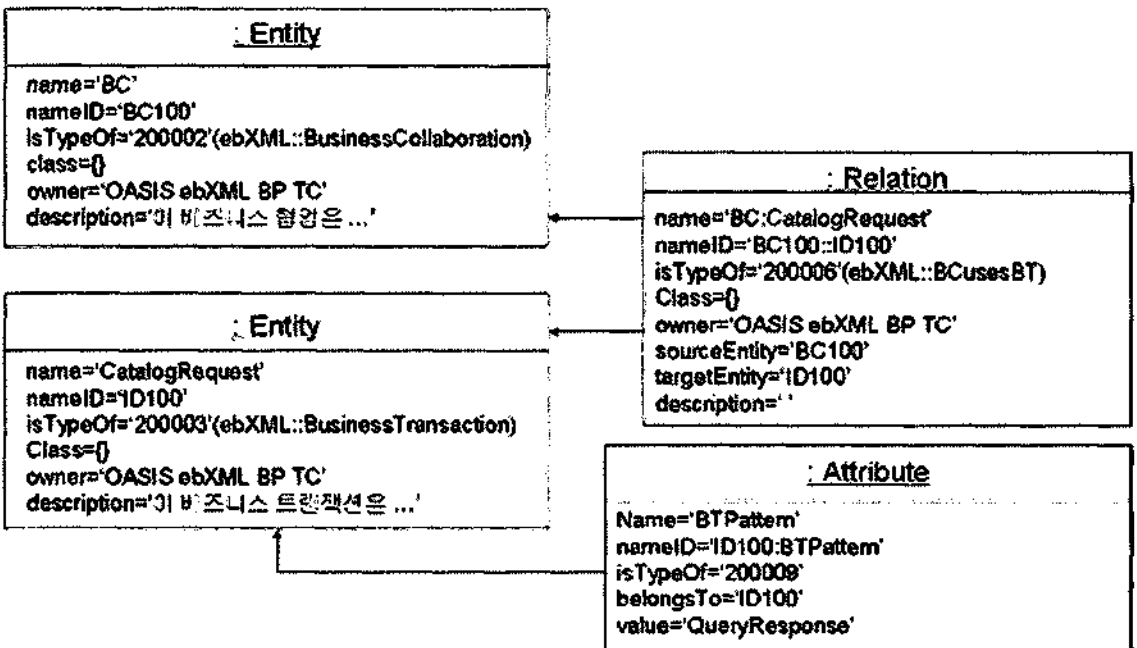
로 정의되어 있다.

〈그림 8〉은 DropShip 예제의 일부분으로, Catalog Request를 정의하기 위한 비즈니스 트랜잭션과 비즈니스 협업의 정의를 보여준다. 첫번째 엘리먼트인 QueryResponse는 "Catalog Request" 비즈니스 트랜잭션을 QueryResponse 패턴으로 정의하고 있다. 두번째 엘리먼트인 BusinessCollaboration에서는 이 비즈니스 트랜잭션을 사용하는 비즈니스 협업을 정의하고 있다.

〈그림 9〉은 Dropship 예제에 대한 BusinessProcess 인스턴스 "PurchaseCluster"와 비즈니스 문서 "Invoice"에 해당하는 Entity 인스턴스, "Data:Invoice"에 해당하는 Entity 인스턴스를 보여준다. 각각의 isTypeOf 속성을 통해서 이들이 각각, BusinessDocument, BusinessTransaction임을 알 수 있

다. 또한 〈그림 9〉에 포함된 Relation 인스턴스는 "Data:Invoice" 비즈니스 트랜잭션과 "Invoice" 비즈니스 문서간의 관계 requesting을 표현하고 있다. 또한 Attribute 인스턴스는 해당 비즈니스 트랜잭션의 패턴이 "DataExchange"임을 value 값을 통해서 표현하고 있다.

〈그림 10〉은 〈그림 8〉의 Catalog Request와 관련된 BusinessTransaction과 BusinessCollaboration에 대한 객체 인스턴스들을 보여준다. 즉, "BC" BusinessCollaboration이 "CatalogRequest"(nameID = 'ID100')을 "uses"하는데 이를 Relation 인스턴스 "BC:CatalogRequest"를 이용해서 표현하고 있다. 또한 Attribute 객체 인스턴스를 이용하여 "CatalogRequest" BusinessTransaction의 패턴이 'QueryResponse'임을 표시하고 있다.



〈그림 10〉 BPSS 비즈니스 프로세스 등록 시점 객체 예제(2)

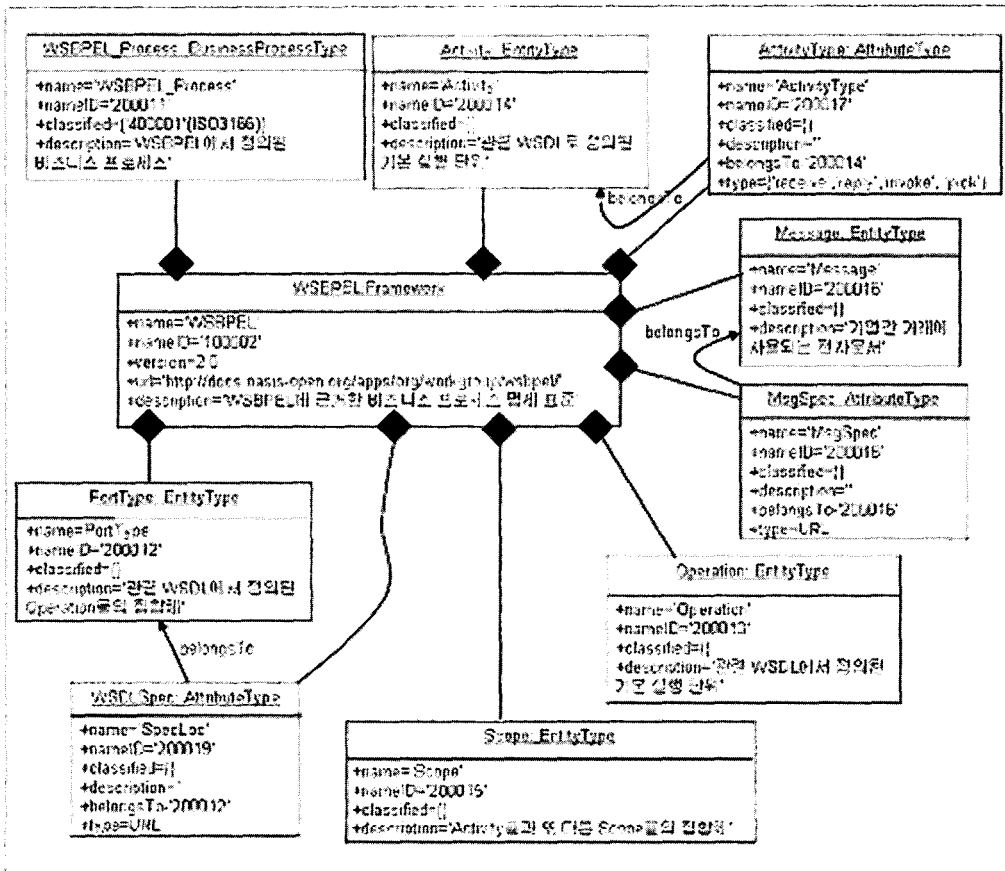
3.4 WSBPEL 비즈니스 프로세스의 등록

3.4.1 등록저장소 설계 시점 클래스 등록

WSBPEL은 웹 서비스에서 제공되는 서비스들을 묶어서 새로운 복합 서비스인 프로세스로 정의할 수 있도록 지원한다[12]. WSBPEL에서 프로세스의 각 단계들을 활동(activity)이라고 부르는데, WSBPEL에서 제공하는 기본 활동(primitive activity)에는 invoke, receive, reply, wait, assign, throw, terminate, empty 등이 있다. 여러 개의 서비스를 연결해서 새로운 서비스를 만들기 위해서, WSBPEL 프로세스는 다른 서비스를 호출하거나, 클라이언

트(서비스의 사용자)로부터의 요청을 받을 수 있도록 되어있다. 다른 서비스를 호출하기 위한 활동이 invoke이고, 클라이언트로부터 요청을 받고 응답하기 위한 활동이 receive와 reply이다.

WSBPEL 프로세스의 등록이 가능하도록 하기 위해서 등록저장소 설계 시점에 생성해야 할 객체로는 <그림 11>의 개체(Entity)/속성(Attribute) 유형들과 <그림 12>의 관계(Relation) 유형들이 있다. 먼저 WSBPEL이라는 프레임워크에는 WSBPEL_Process라고 하는 비즈니스 프로세스 유형이 있으며, Message, Operation, PortType, Activity, Scope 라는 다섯 개의 개체 유형이 있다. 이들은



<그림 11> WSBPEL 프로세스를 등록하기 위한 개체 유형과 속성 유형

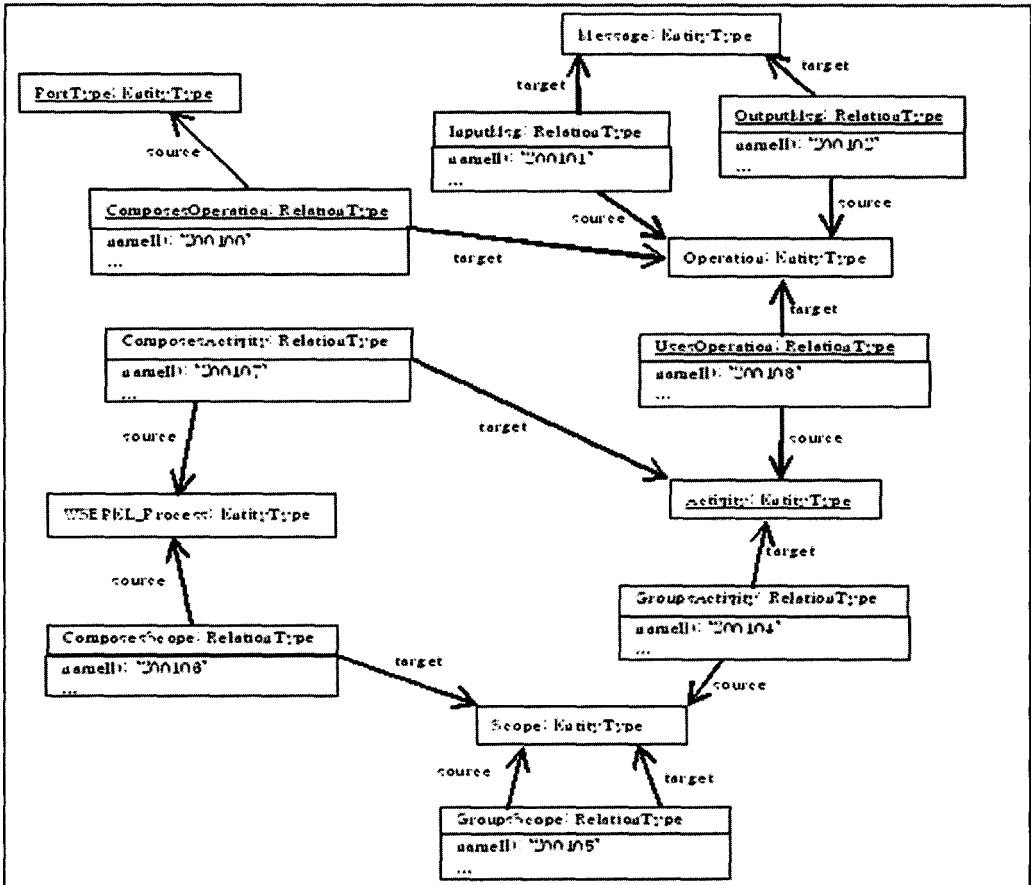
각각 다음과 같이 정의된다.

- Message: 기업간 거래에서 사용될 수 있는 전자문서를 표현하기 위한 개체 유형이다. 전자문서로는 표준 전자문서 설계 규칙을 적용하여 작성된 비즈니스 문서를 사용해야 한다.
- Operation: 포트타입에서 제공되는 오퍼레이션에 관한 정보를 표현하기 위한 개체 유형이다. 메시지 입출력 인터페이스로서 정의된다.
- PortType: 추상적인 오퍼레이션의 집합인 포트타입에 관한 정보를 표현하기 위한 개체 유형이다. 이렇게 WSDL에서 정의된 포트타입의 오퍼레이션들을 연결하여 WSBPEL 프

로세스가 구성되게 된다.

- Activity: 오퍼레이션과 관련된 기본적인 활동을 표현하기 위한 개체 유형이다. receive, reply, invoke, pick과 같이 직접적으로 operation과 관련된 활동만을 대상으로 하며, 기타 컨트롤을 위한 활동들은 제외된다.
- Scope: 중첩된 방식으로 복합적인 활동을 정의할 수 있도록 지원하는 범위에 관한 정보를 표현하기 위한 개체 유형이다. 활동들을 그룹화하는 것으로, 활동들과 함께 또 다른 범위들을 포함할 수 있다.

개체 유형 중에서 특별한 속성 유형이 필요한



〈그림 12〉 WSBPEL 프로세스를 등록하기 위한 관계 유형

경우로는 Activity, Message, PortType이 있다. Activity에서는 receive, reply, invoke, pick 등의 활동 유형을 기술하기 위한 속성이 필요하기 때문에 ActivityType이라고 하는 속성 유형을 추가하였다. Message와 PortType의 경우에는 각각이 정의된 명세서를 기술하는 것이 필요하기 때문에 MsgSpec, WSDLSpec이라는 속성 유형을 각각 추가하였다.

WSBPEL 프로세스를 등록하기 위해서 필요한 관계 유형으로는 InputMsg, OutputMsg, ComposesOperation, UsesOperation, ComposesActivity, GroupsActivity, ComposesScope, GroupsScope 등 8가지가 있다. 이들과 앞에서 정의된 개체 유형과의 관계는 <그림 12>와 같다. 관계 유형 각각의 의미는 다음과 같다.

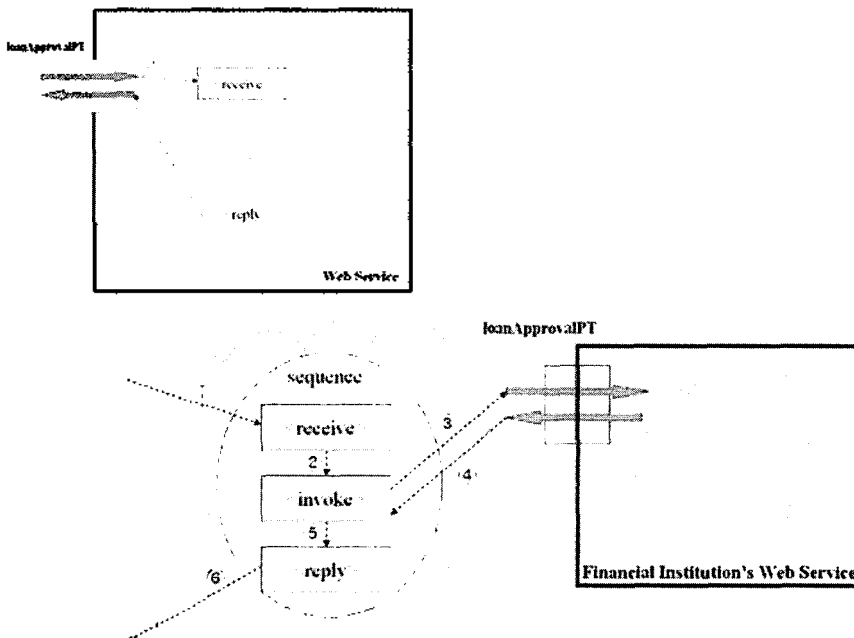
- InputMsg: 오퍼레이션의 입력 메시지를 지정
- OutputMsg: 오퍼레이션의 출력 메시지를 지정
- ComposesOperation: 오퍼레이션들의 집합체인

포트타입에 오퍼레이션이 부속됨을 표현

- UsesOperation: 특정한 활동에서 특정한 오퍼레이션을 이용함을 표현
- ComposesActivity: 활동들과 범위들의 집합체인 WSBPEL-Process에 활동이 부속됨을 표현
- GroupsActivity: 특정한 활동이 특정한 범위에 포함됨을 표현
- ComposesScope: 활동들과 범위들의 집합체인 WSBPEL-Process에 범위가 부속됨을 표현
- GroupsScope: 특정한 범위가 또 다른 범위에 포함됨을 표현

3.4.2 비즈니스 프로세스 등록 시점 클래스 등록

WSBPEL의 비즈니스 프로세스는 등록시점에 <그림 4>에 포함된 클래스를 사용하여 등록된다. 간단한 대출 승인 프로세스 WSBPEL 예제를 사용하

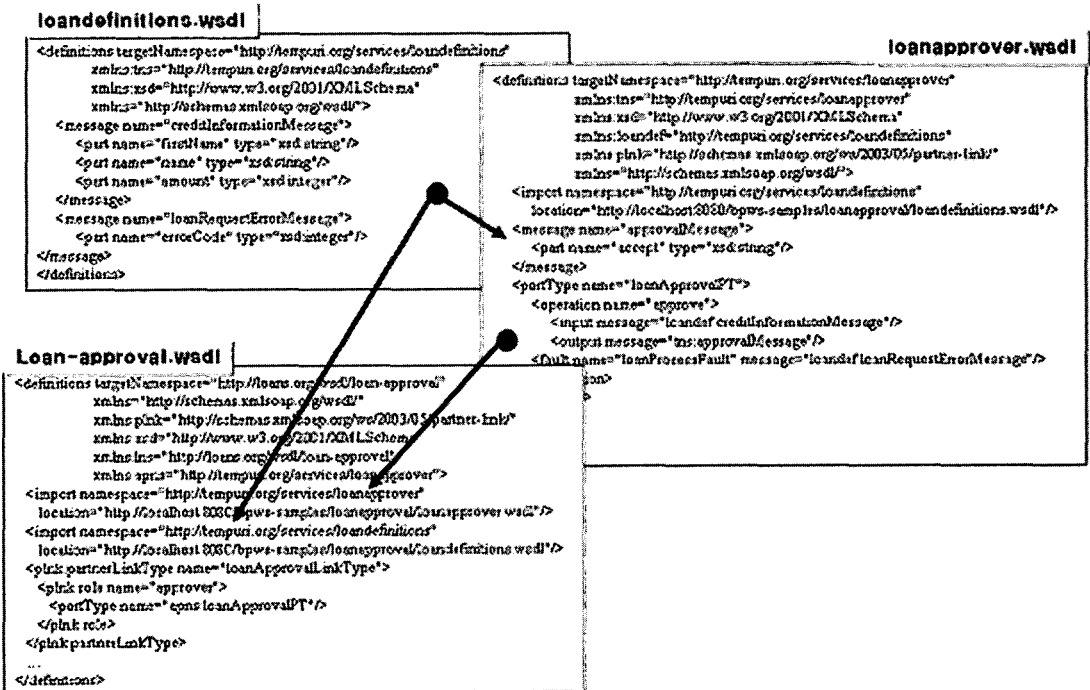


<그림 13> 대출 승인 프로세스의 외부적/내부적 모습

여 등록 시점 클래스 생성을 설명하도록 한다. 이 프로세스는 한 고객이 대출을 요청하면, 이 요청이 금융 기업에서 처리되어, 고객에게 대출이 승인되었는지를 통보하는 복합 서비스다. <그림 13>은 고객의 입장에서 대출 승인 프로세스의 외부적 모습과 내부적인 모습을 보여준다. 즉, <그림 13>의 상단의 박스가 프로세스의 외부적 모습이다. loanApprovalPT라고 하는 포트타입(Port Type)의 오퍼레이션(operation)을 통해서 요청을 하고 그 결과를 이 오퍼레이션을 통해서 받게 된다. 내부적으로 이 프로세스는, 고객이 대출을 요청하면 이를 금융 기업이 받아서 관련 금융 기관의 웹 서비스를 호출하고 그 결과를 받아서 고객에게 응답을 하게 된다. 여기서 한 가지 주의할 점은 이 금융 기관의 웹 서비스에서 제공되는 포트타입도 금융 기업의 포트타입인 loanApprovalPT와 동일하다는

것이다. 고객으로부터 대출 승인을 요청 받고 관련 금융 기관의 웹 서비스를 호출하고 고객에게 결과를 응답하는 활동들이 각각 receive, invoke, reply로 정의되고 또한 이러한 활동들이 순차적으로 일어나는 sequence 활동으로 표현된다.

WSBPEL의 프로세스 정의는 WSDL과 밀접하게 관련되어 있다. 위의 대출 승인 프로세스를 정의하기 위해서는 해당 금융 기업의 WSDL과 관련 금융 기관의 WSDL이 필요하다. 우선적으로 금융 기업 및 기관들간에 교환되는 대출 정보를 규정하는 표준이 정의되어 있다고 가정하며, 그 내용은 <그림 14>의 loandefinitions.wsdl과 같다고 하자. loandefinitions.wsdl에 보면 두 개의 메시지 creditInformationMessage와 loanRequestErrorMessage가 정의되어 있다. creditInformationMessage의 경우, firstName, name, amount로 구성되어 있고,



<그림 14> 예제 wsdl 파일

loanRequestErrorMessage는 errorCode로 이루어져 있다.

관련 금융 기관의 대출 승인 서비스의 WSDL은 <그림 14>의 loanapprover.wsdl과 같이 정의된다. loanapprover.wsdl에는 하나의 포트타입(portType) loanApprovalPT가 있고 여기에 포함된 오퍼레이션 approve를 통해서 대출 여부를 판단한다. 입력 메시지는 앞의 loandefinitions.wsdl에서 정의된 것을 동일하게 사용한다. <그림 14>의 loan-approval.wsdl은 금융 기업의 대출 심사를 위한 WSDL 파일이다.

WSDL 파일 내에는 partnerLinkType에 대한 정의가 있는데, 이는 역할(role)과 포트타입(portType)을 연결한다.

WSDL 파일들을 기초로 해서 비즈니스 프로세스가 정의된다. WSBPEL 파일에는 WSDL의 파트너링크타입(partnerLinkType)을 따르는 파트너링크(partnerLink)들을 정의하게 된다. 이 예제에서는 두 개의 파트너링크 assessor와 approver가 존재한다. 파트너링크에서는 파트너링크타입

```

simpleloanprocessing.bpel
<process name="loanApprovalProcess"
  targetNamespace="http://acme.com/simpleloanprocessing"
  xmlns="http://schemas.xmlsoap.org/ws/2002/07/business-process/"
  xmlns:ins="http://loans.org/wsd/loan-approval"
  xmlns:loandef="http://tempuri.org/services/loandefinitions"
  xmlns:apns="http://tempuri.org/services/loanapprover">
  <partnerLinks>
    <partnerLink name="assessor"
      partnerLinkType="ins:loanApprovalLinkType"
      myRole="approver"/>
    <partnerLink name="approver"
      partnerLinkType="ins:loanApprovalLinkType"
      partnerRole="approver"/>
  </partnerLinks>
  <variables>
    <variable name="request" messageType="loandef:creditInformationMessage"/>
    <variable name="approvalInfo" messageType="apns:approvalMessage"/>
  </variables>
  <sequence>
    <receive name="receive1" partnerLink="assessor"
      portType="apns:loanApprovalPT" operation="approve"
      variable="request">
      </receive>
    <invoke name="invokeapprover" partnerLink="approver"
      portType="apns:loanApprovalPT" operation="approve"
      inputVariable="request" outputVariable="approvalInfo">
      </invoke>
    <reply name="reply1" partnerLink="assessor"
      portType="apns:loanApprovalPT" operation="approve"
      variable="approvalInfo">
      </reply>
    </sequence>
</process>

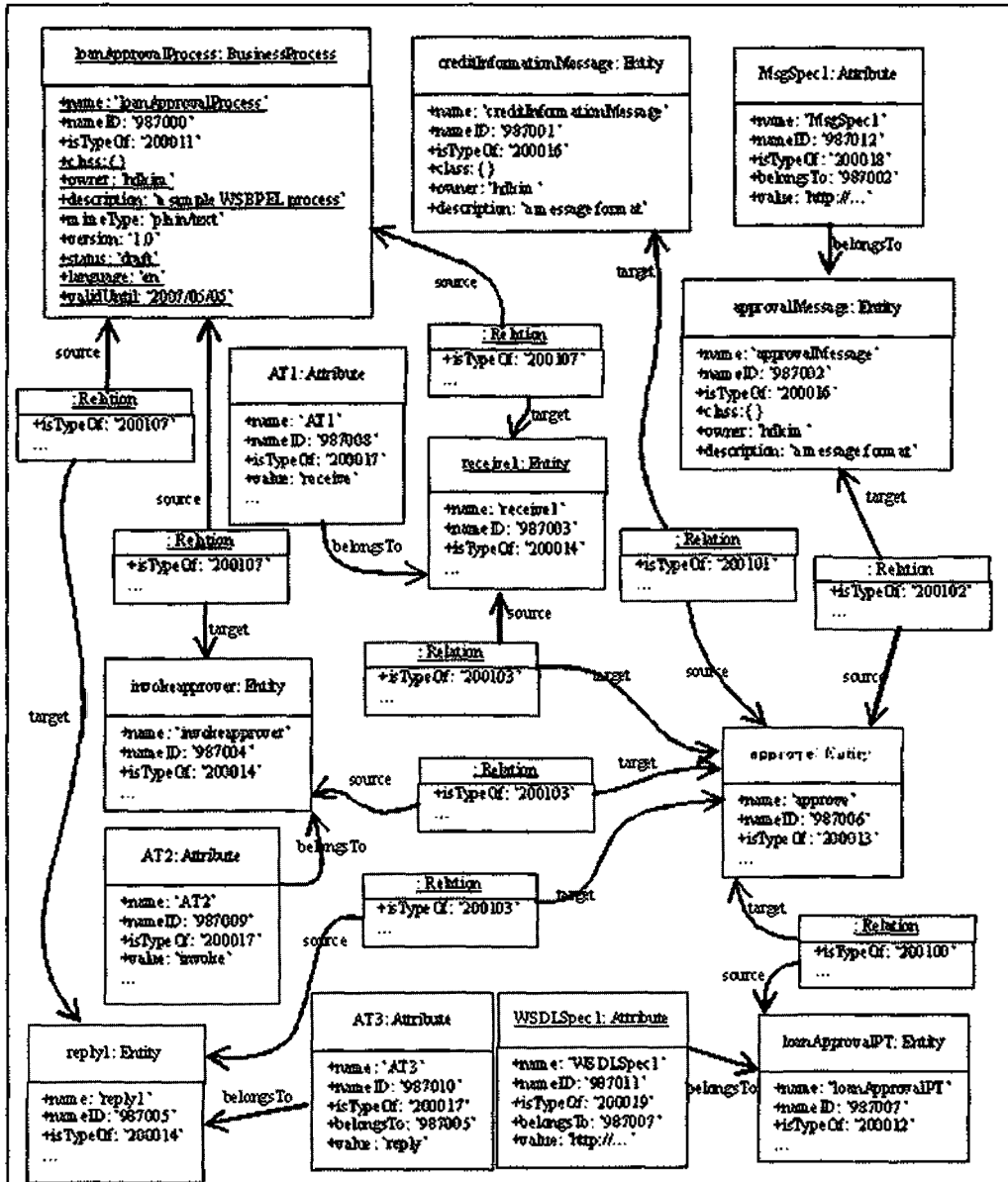
```

<그림 15> loanApprovalProcess의 정의

(partnerLinkType)에서 자신의 역할과 파트너의 역할을 myRole, partnerRole 속성으로 정의한다. 다음으로, WSBPEL 활동들과 WSDL 메시지의 데이터를 접근하도록 정의하기 위한 variable들이 정의된다. 마지막으로 프로세스 내의 활동들과 이러한 활

동들의 선후관계가 정의된다. loanApprovalProcess를 등록하고자 할 경우 그 내용은 <그림 15>와 같이 표현된다.

앞에서 소개한 loanApprovalProcess를 등록저장소에 등록하고자 할 경우 그 내용은 <그림 16>과 같



<그림 16> loanApprovalProcess 등록 사항

다. BusinessProcess 클래스 인스턴스로 'loanApprovalProcess' 클래스를 생성하고, 이것이 WSBPEL_Process 타입(isTypeOf값이 '200011', <그림 11> 참조)임과 기타 사항들을 기술한다. 이 비즈니스 프로세스에는 <그림 16>과 같이 여러 ConstructInstance 객체들이 모두 포함된다. 그리고 각종 관계들로 이들 객체들이 연결된다. 예를 들어, 'receive1'은 'Activity' 객체 유형(isTypeOf값이 '200014', <그림 11> 참조)이고, 'loanApprovalProcess'와 '200107' 관계 유형(즉, ComposesActivity를 표현하는 관계 유형, <그림 12> 참조)의 인스턴스로 연결된다. 또한 'receive1'에 속하는 속성을 보면 'ATI'이 있는데, 이것은 200017이라고 하는 속성 유형인 ActivityType을 기술하기 위한 속성으로서 (<그림 11> 참조), 그 값은 'receive'를 가지고 있다.

4. 토의사항

3장에서는 B2B 등록저장소에 비즈니스 프로세스 등록을 위한 메타데이터 스키마를 제시하고 ebXML, BPSS와 WSBPEL을 중심으로 유용성을 살펴보았다. 이 절에서는 제시한 메타데이터 스키마의 특징과 관련 이슈들을 살펴보도록 한다.

(1) 비즈니스 프로세스 재활용 단위의 확장

기존의 B2B 등록저장소에서 비즈니스 프로세스의 등록, 저장의 단위는 전체 비즈니스 프로세스로 한정되어 있었다. 이것은 비즈니스 프로세스를 구성하고 있는 구성요소에 대한 부분적인 재활용을 충분히 지원하지 못하는 것을 의미하고, 비즈니스 프로세스 내에 포함된 내용을 기반으로 한 검색의 제공이 어렵다는 것이다. 예를 들어, 비즈니스 프로세스를 구성하는 비즈니스 트랜잭션별 또는 비즈니스 협업 수준의 재사용이 어렵고 “구매요청서

를 사용하는 비즈니스 프로세스를 찾아라”는 식의 내용 기반의 질의를 처리하기가 쉽지 않다. 이와 같은 한계를 극복하기 위해서는 비즈니스 프로세스 구성요소들과 이들 간의 관계에 대한 등록이 필요하다. 본 연구에서 제시한 메타데이터 스키마는 비즈니스 프로세스와 구성요소들, 그들의 관계와 속성들을 등록하고 이를 통해서 검색할 수 있는 비즈니스 프로세스 등록저장소 메타데이터 스키마를 제시하고 있다.

(2) 비즈니스 표현 프레임워크 중립성

제시된 메타데이터 스키마는 기존에 다양한 B2B 비즈니스 프로세스 표현 프레임워크들이 경쟁하고 있으며, 계속 발전하고 있는 상황을 반영하여 비즈니스 표현 프레임워크에 종속적인 부분을 포함하지 않고, 비즈니스 프로세스 표현 프레임워크, 비즈니스 프로세스, 비즈니스 프로세스 구성요소(객체 유형), 이들 간의 관계(관계 유형), 이들의 속성(속성 유형)을 중심으로 비즈니스 프로세스의 내용을 등록할 수 있도록 설계되었다. 이와 같은 표현 프레임워크에 대한 중립성은 다양한 표현 프레임워크를 포함할 수 있는 기초를 제공하며, 또한 발전하는 표현 프레임워크들의 특성을 지속적으로 반영하는 것이 가능하도록 한다.

(3) 등록 대상의 확장가능성

제시한 메타데이터 스키마를 사용하여 등록저장소를 구축하기 위해서는, 등록저장소 설계 시점에 등록저장소 관리자가 등록저장소 내에서 지원하는 비즈니스 프로세스 표현 프레임워크들과 등록 대상이 되는 비즈니스 프로세스의 구성요소들과 이들의 관계를 메타데이터 스키마를 활용하여 등록하도록 되어 있다. 이렇게 등록된 내용을 기초로 추후에 비즈니스 프로세스 인스턴스를 등록하는 사용자들은 구성요소 단위의 재활용, 검색, 브라우징

동이 가능하다. 이와 같은 등록저장소 설계 시점 등록 객체와 비즈니스 프로세스 등록 시점 객체의 구분은 등록 대상의 확장가능성과 유연성을 제공한다. 즉, 등록저장소의 목적에 따라서 등록저장소 설계자가 지원되는 표현 프레임워크들과 등록 대상 비즈니스 프로세스 구성요소들의 범위를 정할 수 있다.

(4) 다양한 비즈니스 프로세스 표현 프레임워크의 수용

본 연구에서 제시한 메타데이터 스키마는 개체-관계 속성의 보편화된 데이터 모델링 프레임워크에 기반하고 있어서 다양한 비즈니스 프로세스 표현 프레임워크의 수용이 가능하다. 본 논문에서는 가장 보편적으로 활용되는 XML 기반의 B2B 비즈니스 프로세스 표현 프레임워크인 ebXML BPSS와 WSBPEL의 예제만을 제시하였으나, 실제로는 도식적 비즈니스 프로세스 표현 프레임워크인 BPMN과 UMM으로 생성된 워크시트들을 등록저장하기 위한 메타데이터 스키마의 활용 방안 에 대한 연구도 함께 진행이 되었다[1]. 따라서 제시된 메타데이터 스키마를 통해서 XML 기반의 비즈니스 프로세스 표현 프레임워크뿐만 아니라 도식적 비즈니스 프로세스 표현 프레임워크도 지원할 수 있음을 확인하였다.

(5) 타 프레임워크 내 등록저장된 구성요소간의 연관성의 지원

동일 표현 프레임워크 내의 구성요소들간의 관계에 대한 관리에 대해서는 3장의 ebXML BPSS와 WSBPEL 예제에서 다루어졌다. 예를 들어, ebXML BPSS의 구성요소인 비즈니스 협업과 비즈니스 트랜잭션간에는 'BCUsesBT' 라는 관계 유형을 통해서 이들간의 관계가 등록저장소 내에서 관리된다. 하지만 관계 유형(RelationType) 클래스

의 활용은 동일 프레임워크 내로 한정되지는 않는다. 이것은 다른 표현 프레임워크로 표현된 구성요소들간의 관계도 등록하여 관리될 수 있음을 의미한다. 예를 들어, 동일한 비즈니스 프로세스에 대한 ebXML BPSS 표현과 BPMN의 도식적 표현이 존재하고 이들간에 "isEqualTo" 관계를 관리하고자 한다면, 이 관계 유형을 등록저장소 설계 시점에 생성함으로써 관리가 가능하다.

(6) 메타데이터 스키마 기반의 등록저장소 구현

제시된 메타데이터 스키마를 기반으로 등록저장소를 구현하는 방안으로는, 제시된 메타데이터 스키마를 기초로 등록저장소의 기능을 하나하나 개발하는 방법이 있을 수 있으나, 이는 많은 노력과 시간이 소모되는 방안으로 바람직해 보이지 않는다. 이보다는 기존의 등록저장소 정보 모델인 ebRIM이나 UDDI를 기초로 하여[8,10], 본 연구에서 제시된 메타데이터 스키마를 이들 정보 모델에 매핑하는 것이 더 적절한 방안으로 생각된다. 즉, 제시된 메타데이터 스키마를 이들 정보 모델로 매핑하여, 이들 정보 모델을 지원하는 기개발된 등록저장소 시스템들을 활용하여 구현하는 것이 더 효율적인 방안이 될 것으로 보인다. 따라서 본 연구의 연장선 상에서 ebRIM으로 매핑에 대한 연구가 진행되었다[1]. 실제 서비스가 될 수 있는 등록저장소 시스템의 구현을 위해서는 스키마의 매핑뿐만 아니라, 비즈니스 프로세스 내용을 입력 받아, 이를 매핑 규칙에 따라서 변환하여 저장하는 기능의 추가적인 구현이 필요하다. 또한, 제시된 메타데이터 스키마를 통한 질의(query)를 내부 저장 정보 모델에 대한 질의 집합으로 재구성하여 질의를 수행하고 이 질의의 결과를 다시 제시된 메타데이터 스키마에 맞추어 전환하여 제시하는 기능의 구현이 필요하다.

(7) 등록저장 편의성의 제공

본 연구에서 제시된 메타데이터 등록저장소가 비즈니스 프로세스의 구성요소 수준에서 재활용을 제공하는 장점이 있지만, 단일 비즈니스 프로세스 등록 시에 각 구성요소를 하나하나 비즈니스 프로세스 등록자가 등록해야 한다면, 제시된 메타데이터 스키마의 유용성은 크게 훼손될 것이다. 하지만 다행스럽게도 ebXML, BPSS나 WSBPEL로 표현된 비즈니스 프로세스의 경우는 XML 기반이므로, 이들 구성요소의 자동적인 추출이 가능할 것으로 보인다. 물론 이를 위해서는 등록저장소 시스템 내에 이들을 추출하는 기능이 추가적으로 개발되어야 할 것이다. 하지만, BPMN과 같은 도식적 비즈니스 프로세스 표현 프레임워크의 경우는 자동적인 구성요소의 추출이 더 어려울 것으로 보인다. 따라서 이를 위해서는 추가적인 연구가 필요할 것으로 보인다.

5. 결 론

본 연구에서는 다양한 표현 방법으로 표현된 기업간 비즈니스 프로세스를 B2B 등록저장소에 등록하고 검색하기 위한 메타데이터 스키마를 제시하였다. 메타데이터 스키마는 등록저장소 설계 시점 생성 객체와 개별 비즈니스 프로세스 등록 시

에 생성 객체를 포함한다. 제안한 메타데이터 스키마는 특정 비즈니스 프로세스 표현 프레임워크에 독립적인 형태를 가지고 있으며, 재활용의 수준을 전체 비즈니스 프로세스에서 더욱 상세화하여 비즈니스 프로세스의 주요 구성요소 수준으로 할 수 있도록 설계되었다. 제안된 메타데이터 스키마가 다양한 비즈니스 프로세스 표현 방안들을 지원할 수 있는 지를 확인하기 위해서, 대표적인 XML 기반의 비즈니스 프로세스 표현 언어인 ebXML, BPSS와 WSBPEL로 표현된 비즈니스 프로세스 예제들을 중심으로 설명하였다.

추후 연구 과제는 다음과 같다. 첫째, 제시된 메타 모델을 근간으로 실제로 작동하는 B2B 비즈니스 프로세스 등록저장소를 구현하여 관련 종사자들이 유용하게 활용할 수 있도록 하는 것이 필요하다. 특히, 재활용 수준을 전체 비즈니스 프로세스가 아닌, 비즈니스 프로세스를 구성하는 구성요소들로 상세화하기 위해서는 비즈니스 프로세스 정의로부터 자동적으로 구성요소들을 추출하여 등록하는 기능이 필요하다. 둘째로, 실제 등록저장소의 유용성을 높이기 위해서는 등록저장소 내에 다수의 비즈니스 프로세스들이 축적되는 것이 필수적인데, 기업들이 자발적으로 비즈니스 프로세스를 공유할 수 있도록 하는 유도 방안의 도출과 실행이 필요하다.

참 고 문 헌

- [1] 김종우, 김형도, 김호철, 박병용, 이영곤. 비즈니스 프로세스 등록 저장을 위한 메타 모델 설계. 연구보고서, 한국전자거래진흥원, 11월, 2006.
- [2] 김형도, 김종우. "UML 기반의 기업간 비즈니스 프로세스 명세 모델링," *Journal of Information Technology Applications & Management*, 제13권, 제4호, 2006. 12, pp.71-88.
- [3] BPM. "Business Process Modeling Notation (Version 1.0)," *Business Process Management Initiative*, May, 2004.
- [4] Dogac, A. et al. "Enhancing ebXML Registries to Make them OWL Aware," *Distributed and Parallel Database*, Vol. 18, 2005, pp. 9-36.
- [5] OASIS, Collaborative-Protocol Profile and Agreement Specification (Version 2.0), OASIS, September, 2002.
- [6] OASIS, Business Process Execution Language for Web Service (Version 1.1), OASIS, May, 2003.
- [7] OASIS, UDDI (Version 3.0.2), OASIS, 2004.
- [8] OASIS, Using BPEL4WS in a UDDI Registry, Technical Note, OASIS, 2004.
- [9] OASIS, ebXML Registry Service and Protocols (Version 3.0), OASIS, May, 2005.
- [10] OASIS, ebXML Registry Information Model (Version 3.0), OASIS, May, 2005.
- [11] OASIS, ebXML Business Process Specification Schema Technical Specification (version 2.0.1), OASIS, July, 2005.
- [12] OASIS, "Web Service Business Process Execution Language (Version 2.0)," OASIS, August, 2005.
- [13] REMKO, <http://www.remko.or.kr>, 2006.
- [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] W3C, "Web Services Choreography Interface (WSCI) 1.0," W3C, <http://www.w3.org/TR/wsci/>, 2002.
- [17] W3C, "Web Services Choreography Description Language (WS-CDL) Version 1.0," W3C, 2005.
- [18] WfMC, "XML Process Definition Language Version 1.0," WfMC, <http://www.wfmc.org/standards/XPDL.htm>, 2005.
- [19] Weerawarana, S. and Curbera, F., "Business Process with BPEL4WS: Understanding BPEL4WS," IBM Corporation, 2002.
- [20] White, S.A., "Introduction to BPMN," IBM Corporation, _ .

저 자 소개



김종우 (E-mail: kjw@hanyang.ac.kr)
 1989. 서울대학교 수학과 (학사)
 1991. 한국과학기술원 경영과학과 (석사)
 1995. 한국과학기술원 산업경영학과 (박사)
 1995~1996. 한국과학기술원 산업경영연구소 Post Doc.
 1996~2003. 충남대학교 통계학과 전임강사, 조교수, 부교수
 1999.~2000. University of Illinois at Urbana-Champaign 방문연구원
 2003~현재 한양대학교 경영학부 부교수
 관심분야 전자상거래, 상품 추천 기술, 데이터 마이닝, 지능정보시스템 의사결정지원시스템, 비즈니스 프로세스 모델링과 통합



김형도 (E-mail: hdkim@hycu.ac.kr)
 1985. 서울대학교 산업공학과 (학사)
 1987. 한국과학기술원 경영과학과 (석사)
 1992. 한국과학기술원 경영과학과 (박사)
 1993~1999. (주)데이콤 EC인터넷 연구/기술 팀장
 2000~2002. 아주대학교 정보통신전문대학원 교수
 2002~2006. 전자상거래표준화통합포럼 전자문서기술위원회 부위원장
 2004~2006. ebXML 전문위원회 위원장
 2006~현재 한국전자거래진흥원 비즈니스 프로세스 워킹그룹 위원장
 2003~현재 한양사이버대학교 경영학부 교수
 관심분야 XML, 전자상거래, 비즈니스 프로세스 디지털 워터마킹, IT 아키텍처, 객체지향모델링, 데이터 마이닝 등