경영정보학연구
제14권 제1호
2004년 3월

ebXML 비즈니스 프로세스 명세를 위한 의미 제약의 모델링과 검증*

김 종 우**, 김 형 도***

Modeling and Validation of Semantic Constraints for ebXML Business Process Specifications

Jong Woo Kim, Hyoung Do Kim

As a part of ebXML (Electronic Business using eXtensible Markup Language) framework, BPSS (Business Process Specification Schema) has been provided to support the direct specification of the set of elements required to configure a runtime system in order to execute a set of ebXML business transactions. The BPSS is available in two stand-alone representations, a UML version and an XML version. Due to the limitations of UML notations and XML syntax, however, current ebXML BPSS specification fails to specify formal semantic constraints completely. In this study, we propose a constraint classification scheme for the BPSS specification and describe how to formally represent those semantic constraints using OCL (Object Constraint Language). As a way to validate a Business Process Specification (BPS) with the formal semantic constraints, we suggest a rule-based approach to represent the formal constraints and demonstrate its detailed mechanism for applying the rule-based constraints to the BPS with a prototype implementation.

Keywords : ebXML, UML (Unified Modeling Language), constraints modeling, model validation, rule-based language

* 본 연구는 2003년 한양대학교 교내 연구비 지원으로 연구되었음.
** 한양대학교 경영학부
*** 한양사이버대학교 경영정보학과

Ⅰ. 서 론

ebXML(Electronic Business using eXtensible Markup Language)은 모듈화된 전자 상거래 프레임워크를 제공하기 위한 명세(specification)의 집합이다[ebXML, 2002a; ebXML, 2002b]. ebXML의 목표는 기업의 규모나 지역적 위치에 상관없이 XML 기반의 메시지 교환을 통해서 상거래를 할 수 있는 글로벌 전자 상거래 환경을 제공하는 것이다. ebXML의 일부로서 제시된 ebXML Business Process Specification Schema(이하 BPSS로 표기)는 기업간의 비즈니스 트랜잭션을 모델링하기 위한 표현 방법이다. ebXML BPSS를 따라서 명세된 비즈니스 프로세스는 B2B 전자상거래를 위한 비즈니스 서비스 인터페이스 구성의 입력이 된다.

OASIS와 UN/CEFACT에 의해 표준화된 ebXML BPSS는 UML 버전과 XML 버전 두 가지가 제시되고 있다[ebXML, 2002a]. UML 버전의 경우는 단순한 UML 클래스 다이어그램 형태로, ebXML BPSS 내의 모델링 요소들과 그들간의 관계를 표현하고, ebXML BPSS에 따른 비즈니스 프로세스 명세 과정을 개념적인 수준에서 수행할 수 있도록 지원하기 위해서 개발되었다. XML 버전의 경우는 ebXML BPSS에 따른 최종적인 XML 형태의 비즈니스 프로세스 명세를 위해 제시되었으며, 따라서 모든 ebXML 비즈니스 프로세스 명세는 궁극적으로 XML 버전의 ebXML BPSS 형태로 작성되어야 한다.

UML버전의 ebXML BPSS가 BPSS 내의 모델링 요소와 그들간의 관계를 명확히 표현하기 위해서 제시되었지만, 클래스 다이어그램 수준에서의 의미적 정형화만 제공할 뿐, 모델링 요소들에 대한 명확한 정의는 부족한 형태이다. UML의 경우도 UML 모델링 요소들의 정형화된 의미의 표현을 위해서 UML 메타 모델[1]이 UML의 클래스 다이어그램과 OCL(Object Constraint Language)를 통해서 표현되어 있다[OMG, 1999; Rumbaugh et al., 1999]. 이와 마찬가지로, UML 버전의 ebXML BPSS에도 궁극적으로 OCL을 활용한 ebXML BPSS 모델링 요소들간의 제약들의 정형화된 표현이 필요하다. 또한 이러한 모델링 요소들의 제약에 대한 정형화된 표현은 ebXML BPSS로 작성된 개별 비즈니스 프로세스 명세(Business Process Specification, 이하 BPS로 표기)의 정확성(accuracy), 완전성(completeness)을 검증하는데도 활용될 수 있다.

본 논문에서는 ebXML BPSS의 모델링 요소들간의 제약들을 체계적으로 정리하기 위한 분류체계를 제시하고 이러한 제약식의 정형화를 위한 OCL 기반의 제약 표현을 제시한다. 또한 이를 활용한 BPS의 검증 방안을 규칙기반 언어인 CLIPS(C Language Integrated Production System) 프로토타이핑을 통해서 제시한다[CLIPS, 2002]. 본 논문의 구성은 다음과 같다. Ⅱ장에서는 관련 연구로, ebXML과 ebXML BPSS, OCL에 대하여 소개한다. Ⅲ장에서는 ebXML BPSS 제약의 분류체계와 OCL기반 BPSS의 제약 표현을 소개한다. Ⅳ장에서는 이러한 제약 표현을 활용한 규칙기반 BPS 검증 방안을 제시한다. Ⅴ장에서는 ebXML BPSS 제약의 정형 표현의 의의와 다른 접근법과의 비교 등에 대하여 논의한다. Ⅵ장에서는 결론을 제시한다.

Ⅱ. 관련 연구

2.1 ebXML과 BPSS

UN/CEFACT와 OASIS에 의해서 표준화가 진

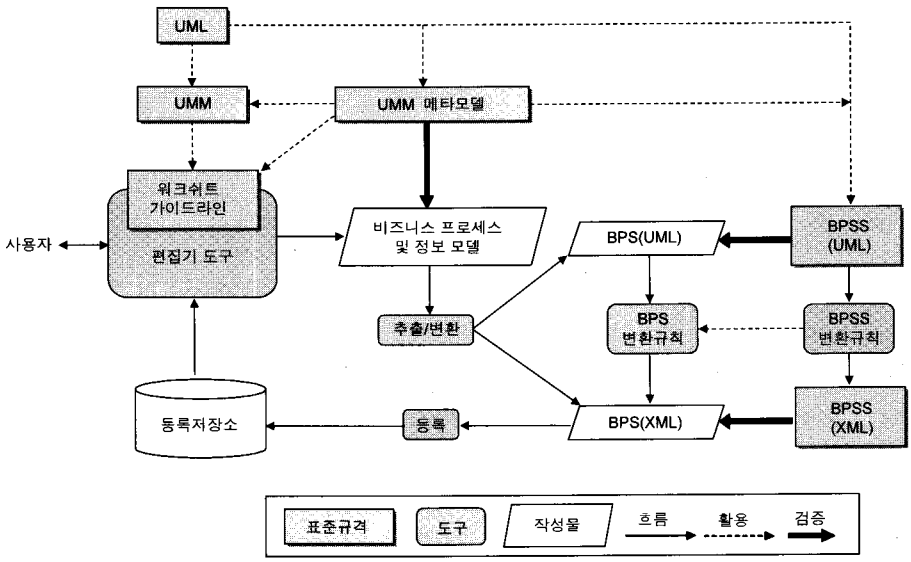
1) UML 메타모델은 UML구조, 모델링 요소들(클래스, 인스턴스, 데이터 타입 등), 다이어그램들에 포함된 의미(semantic)에 대한 반정형적(semi-formal) 정의를 제공함. 도식적 다이어그램, 정형언어(OCL), 자연어 등을 함께 사용하여 표현하고 있음[OMG, 1999].

행되고 있는 ebXML은 산업 분야, 기업 규모, 또는 지역/국가에 무관하게 XML을 사용하여 비즈니스 데이터를 교환하기 위한 전세계적인 프레임워크를 제공한다. 이 프레임워크는 EDI 분야의 오랜 경험을 반영하여 업무 프로세스 모델, 문법 중립적인 핵심 컴포넌트, 협력 규약 프로필과 합의, 분산된 등록기/저장소 등을 XML 메시지와 결합한 점을 특징으로 하고 있다. ebXML을 활용하여 기업들은 업무 메시지를 교환하고, 거래를 수립하고, 공통의 용어로 표현된 데이터를 전송하고, 업무 프로세스를 등록/저장하는 표준적인 기업간 전자상거래 방안을 사용할 수 있게 되었다.

ebXML BPSS에서는 어떤 기업이 거래 기업과의 상호작용을 편리하게 하기 위하여 공유된 역할(Role), 관계(Relationship), 의무사항(Responsibility) 등을 어떻게 수행할 것인지를 상세히 정의한 것이 기업간 거래에서의 비즈니스 프로세스이다[ebXML, 2002b]. 전통적인 기업간 전자상거래에서 많이 사용된 EDI는 비즈니스 프로세스를 충분히 고려하거나 개선하지 않고, 단순히 기존의 문서만을 표준화하여 사용하였기 때문에, 문서를 합리화하고 단순화하는 것이 불가능하였다. 최근에는 xCBL[xCBL.org, 2002], UBL[OASIS, 2002], OAGIS BOD[OAGI, 2002] 등과 같이 XML을 기반으로 하는 문서 표준화 활동도 활발한데, 재활용의 수준을 문서에서 시나리오(비즈니스 프로세스)로 확대하여, 시나리오를 정의하는 과정에서 문서에 대한 검토가 자연스럽게 이루어질 수 있도록 지원하며, 시나리오 수준에서 기업간 거래를 실행하고 관리될 수 있는 방안이 필요하다.

ebXML을 사용하여 거래하고자 하는 기업은 자신의 거래 시나리오를 모델링하여 등록저장소에 등록하고, 거래 상대방과 이 시나리오를 사용하여 거래를 수행할 수 있다. 필요에 따라서는 자신의 상황에 적합한 시나리오를 등록저장소로부터 선택하여 그대로 활용하거나 일부분을 변경하여 거래를 수행할 수도 있다. 주문관리 등과 같이 공통적으로 사용될 수 있는 거래를 비즈니스 프로세스로 모델링하여 재활용하는 것도 매우 중요하다. 국가적인 차원이나 특정한 산업분야 차원에서 기업간 시나리오들을 체계적으로 정의하여 관리할 수 있다면 거래를 수립하기 위한 시간과 비용을 절감하는 효과를 거둘 수 있게 될 것이다. XML로 명확히 정의된 시나리오에 따라서 기업간 거래를 실행하고 관리하는 기능이 지원될 수 있기 때문에, 비즈니스 시나리오를 프로그램 코드에 구현하여 수행하는 기존의 B2B 시스템들(EDI시스템, XML/EDI시스템 등등)이 변화에 유연하게 대응하기 어려웠던 점을 극복할 수 있다.

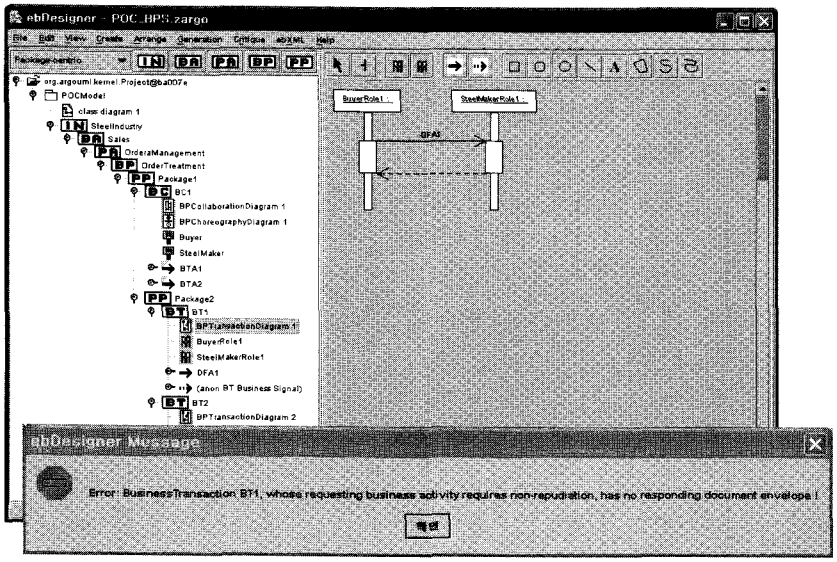
ebXML 프레임워크에서는 거래 시나리오와 문서를 정의하기 위해서는 체계적인 분석이 필요한데, 이러한 분석 작업에 UMM(UN/CEFACT Modeling Methodology)의 사용이 권장된다. UMM은 비즈니스 프로세스와 정보(문서)의 분석과 정의를 위한 방법론(절차 모델)이다[ebXML, 2001b]. UMM 메타 모델은 UML을 확장하여 ebXML 프레임워크 내에서 비즈니스 프로세스를 분석할 때 도출되어야 하는 모든 정보들의 타입을 제시하고 있다[ebXML, 2001b]. BPSS는 UMM 메타 모델의 부분집합으로서, 비즈니스 실행 환경의 측면에서(즉, 비즈니스 프로세스를 실제 실행하기 위해서 반드시 필요한 내용만을 규정하는 측면에서) 기업간 협업을 정의할 수 있도록 지원하기 위한 것이다. ebXML 프레임워크는 표준 객체지향 모델링 언어인 UML을 기본 모델링 언어로 활용하고 있다. UMM, UMM 메타모델, BPSS, 모두 UML을 활용하거나 이를 기초로 설명되고 있다. 예를 들어, BPSS는 <그림 1>과 같이 UML과 XML 스키마 두 가지의 독립적인 표현 방식을 제공한다. BPSS 변환규칙의 집합도 제공되는데, 이것은 UML 버전의 ebXML BPSS로부터 XML버전의 BPSS로의 변환을 정의한다. 또한 이에 근거한 BPS 변환규칙은



<그림 1> ebXML BPS 생성 과정

UML 버전의 BPS와 XML 버전의 BPS간의 변환을 정의한다. BPS의 작성은 크게 3가지 방법이 가능하다. 즉, ebXML 비즈니스 프로세스 분석 워크쉬트와 가이드라인에 기반한 편집기 도구를 사용하는 방법, XML 버전의 BPSS를 사용하는 방법, UML 버전의 BPSS를 사용하는 방법이다. ebXML 비즈니스 프로세스 분석 워크쉬트와 가이드라인은 UMM을 따라 비즈니스 프로세스와 정보 모델을 생성하는 과정을 지원하기 위해서 만들어졌다[ebXML, 2001b]. 이러한 편집기 도구를 활용하는 경우, 비즈니스 프로세스와 정보 모델을 작성하고, 이 중 BPS에 해당하는 부분을 추출하여 XML형태의 BPS로 변환하여 등록저장소에 등록하여 사용한다. 이러한 편집기 도구로 대표적인 것이 ebDesinger[Kim, 2002a]이다. BPS 작성을 위한 다른 방법은 UML 버전의 BPSS 또는 XML 버전의 BPSS를 활용하여 직접 BPS를 생성하는 것이다. 어떠한 과정을 통해서 작성되든지, 궁극적으로 등록저장소에 저장되는 BPS 형태는 XML 형태이다. XML 버전의 BPS와 UML 버전의 BPS가 상호변환이 가능하므로, 본 논문에서 UML 기반의 제약 검증을 중심으로 다루도록 한다.

BPSS를 준수하여 정의된 BPS는 비즈니스 프로세스에 필요한 업무 문서의 구조를 정의하거나 기업의 기술적인 능력을 정의하는데 참조되며, 실제로 비즈니스 프로세스를 실행하는 과정에서는 비즈니스 프로세스 엔진을 구동하는 시나리오 역할을 수행하게 된다. 따라서 특정한 비즈니스 프로세스와 관련하여 실존하는 의미 제약을 완전하고 일치되게 정의할 수 있는가 하는 것이 BPS를 (재)활용하는데 있어서 매우 중요하다. 현재의 BPSS 공식 문서에서는 BPSS 주요 개념들의 의미에 관한 설명이 서술적 문장들로 제공되고 있으며, XML기반의 BPSS에서는 XML Schema를 이용하여 명세에 관한 제약을 정의하고 있으나, 비즈니스 프로세스에 대한 의미 제약을 모두 표현하기에는 매우 부족하다. 예를 들면, 어떤 요소가 존재하면 또 다른 요소가 존재해야 한다는 것과 같은 의미 제약의 표현은 XML Schema를 이용한다고 하더라도 표현이 불가능하다. 이러한 문제점을 개선하기 위해서는 XML을 읽어 들여 분석하는 전용 분석 프로그램을 작성하거나, Schematron



<그림 2> ebDesigner에서의 의미제약 검증

[Jelliffe, 2002]이나 XCSL(XML Constraint Specification Language)[Ramalho, 2001]과 같은 XML 문서에 대한 의미 제약을 표현하는 언어를 이용할 수 있다. 사용자가 직접 XML로 BPS를 작성하거나 수정한 경우에는 이러한 점검 과정이 반드시 필요한데, 현재 제약을 표현하기 위한 표준 언어가 존재하지 않기 때문에 여러가지 어려움이 있다.

개념적인 수준에서 의미 제약을 표현하여 처리할 수 있다면 여러 가지 장점을 갖게 되는데, 일반적으로 작업중인 BPS 모델에 대한 의미 제약을 검증할 수 있고, 모델링 작업을 안내할 수 있으며, 이후에 자동으로 생성된 모델에 대한 의미 제약의 적용이 편리하다. 개념적 수준에서의 의미 제약을 자동으로 생성된 모델에 대한 의미 제약으로 변환하여 적용할 수도 있다. <그림 2>는 대화형 그래픽 인터페이스를 통하여 비즈니스 프로세스를 정의할 수 있도록 지원하는 ebXML 편집기 도구인 ebDesigner[Kim, 2002a]에서 BPS를 생성하는 중에 모델의 완전성을 검증하는 과정에서 의미 제약을 만족하지 못하는 경우의 메시지를 보여주고 있다. 이와 같이 모델링 지원 도구에서 모델의 완전성을 검증하기 위해서는 일반적으로 도구 개발자가 자체적으로 작성한 프로그램을 이용하게 된다. 이 방법은 프로그램에 의해서 검증이 진행되기 때문에 구현하기가 어렵고, 검증의 범위와 내용을 정확히 파악할 수 없고, 표준이 변화할 경우 프로그램을 수정해야 하기 때문에 유연성이 부족하다.

2.2 Object Constraint Language

Object Constraint Language(OCL)는 제약식을 표현하기 위한 정형 언어(formal language)이다[OMG, 1999; Rumbaugh et al., 1999]. UML에서는 OCL을 UML의 모델링 요소들의 의미들을 정형화된 형태로 정의하기 위해서 사용한다. 즉, UML 다이어그램만으로는 제대로 다룰 수 없는 제약들을 정형화된 형태로 표현하기 위해 OCL을 사용한다. 또한 UML을 이용한 시스템 분석시에 분석가가 특정 응용 분야의 제약을 표현하는데 사용할 수도 있다. OCL은 일반적인 정형 언어가 수학적 배경 지식이 없으면 이해하기 어렵고, 일반적인 시스템 분석가가 사용하기 어려

운 단점을 극복하기 위해서 IBM에서 비즈니스 모델링 언어로 개발되었다. UML에서 OCL은 클래스 모델의 클래스와 타입의 불변조건(invariants, 항상 모든 인스턴스에 대해서 참으로 유지되어야 하는 제약조건)을 명세하거나, 2) 스트레오타입의 타입 불변조건을 명세하거나, 3) 오퍼레이션과 메소드의 사전 조건(precondition), 사후 조건(postcondision)을 표현하거나, 4) Guard 조건[2]을 기술하거나, 5) 네비게이션 언어로 사용하거나, 6) 오퍼레이션의 제약을 표현하는데 사용할 수 있다. 다음은 몇 가지 OCL 문장의 예제이다.

```
-- 사람의 나이는 0보다 커야한다.
  context Person inv:
     self.age > 0

-- 결혼한 사람은 18세 이상이어야 한다.
   context Person inv:
       self.wife → notEmpty implies
                   self.wife.age >= 18 and
       self.husband → notEmpty implies
                   self.husband.age >= 18[3]

-- 생일이 되면, 기존의 나이에 1살은 더한다.
   context Person.birthdayHappens() post:
       age = age@pre + 1

-- 회사의 직원 중에 forename이 'Jack'인 직원이
   적어도 한 명은 존재해야 한다.
   context Company inv:
       self.employee → exists(p | p.forename
                       = 'Jack')
```

'--'는 OCL에서 주석문을 의미하며, 첫 번째 제약식은 객체 Person의 속성 age에 대한 제약이고, 두 번째 제약식은 객체 Person이 가지는 연관관계에 대한 제약 조건이다. 두 제약식 모두에 적힌 'inv'은 invariant의 약자로, 두 제약식이 불변조건임을 표시한다. 세 번째 제약식은 객체 Person이 가지는 오퍼레이션 birthdayHappens에 대한 제약식이다. 이 제약식의 'post'는 이 제약 조건이 오퍼레이션 수행 후에 만족시켜야 하는 사후 조건임을 의미한다. 네 번째 제약식은 객체 Company에 대한 불변 제약식의 예로 OCL 내에서 exists, forAll 등이 사용될 수 있음을 보여주고 있다.

Ⅲ. OCL 기반의 ebXML BPSS 제약 표현

현재 가용한 최신의 ebXML BPSS 버전은 2002년 6월에 발표된 1.05 버전이다. 현재 ebXML BPSS는 UML 형태와 XML 형태가 함께 제시되고 있다[ebXML, 2002a]. 본 연구에서는 UML 버전의 ebXML을 기반으로 설명하도록 한다. BPSS의 모델링 구성요소(element) 명세는 크게 4가지(Business Collaborations, Business Transactions, Document Flow, Choreography)로 구분하여 제시되고 있으며 각각에 속한 구체적인 구성요소는 다음 <표 1>과 같다.

현재 1.05 버전에서는 ebXML BPSS 각 구성요소의 의미적 제약들이 서술적으로 표현되어 있다. 하지만 이전 버전인 Version 1.03에서는 각 구성요소별로 상위클래스, 태그값, 연관성, WellFormedness Rules(이하 WFR로 표기)로 구분하여 각 구성요소를 설명하고 있다. BPSS에서는 각 구성요소에 연관된 11개의 WFR(Version 1.03 기준)은 <표 1>에 표시되어 있으며, 구체적인 내용은 <부록 1>의 WFR 1~11에 있다. 이외에도 Transaction과 Collaboration에 대한 WFR 17개가 별도로 제시되고 있다(이것은 최근 버전

2) 상태변화(transition)가 발생하기 위해서 만족시켜야 하는 조건[OMG, 1999].

3) OCL에서 collections(sets, bags, sequences 등)는 사전에 정의된 타입으로, 다수의 사전에 정의된 property들을 가짐. Collection의 property 호출을 표시할 때 '→'를 사용함[OMG, 1999]. 참고로, OCL에서 한 객체 인스턴스가 가지는 연관관계 인스턴스들은 Collection으로 다루어짐.

과 이전 버전에서 동일함). Transaction과 Collaboration에 대한 WFR은 <부록 1>의 WFR 12~28에 해당한다. <표 1>, <부록 1>의 WFR의 번호는 ebXML BPSS 공식 문서에서 제시된 순서에 따라 본 연구에서 순차적으로 할당하였다.

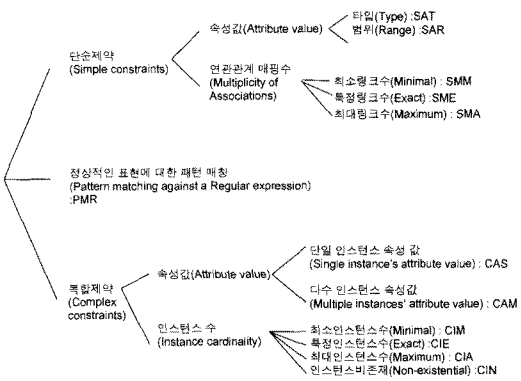
<표 1> ebXML BPSS의 UML 모델링 구성요소

구 분	구 성 요 소	Wellfomedness Rules 번호
Business Collaborations	MultiPartyCollaboration	1
	BusinessPartnerRole	2
	Performs	3
	AuthorizedRole	4, 5
	BinaryCollaboration	-
	BusinessActivity	-
	BusinessTransactionActivity	-
	CollaborationActivity	6
Business Transactions	BusinessTransaction	-
	BusinessAction	-
	RequestingBusinessActivity	-
	RespondingBusinessActivity	-
Document Flow	DocumentSecurity	-
	DocumentEnvelope	7, 8
	BusinessDocument	-
	Attachment	-
Choreography	BusinessState	-
	Transition	9
	Start	-
	CompletionState	-
	Success	10
	Failure	11
	Fork	-
	Join	-

ebXML BPSS 공식문서에는 28개의 WFR 형태로 BPSS의 의미적 제약들을 설명하고 있으나, <부록 1>에서 볼 수 있듯이 자연어에 기반한 설명 자체가 정형화되어 있지 않아서 해석상의 모호성을 일으킬 수 있으며, 기업에서 BPSS를 사용하여 생성한 Business Process Specification (BPS)이 의미적 제약에 맞는지 검증하는데도 활용될 수 없다. 따라서 본 연구에서는 이러한 WFR을 정형제약언어인 OCL을 활용하여 표현하고 검증하도록 한다.

3.1 제약식의 구분

일반적인 XML의 제약조건을 Jacinto et al.의 연구에서 1) 도메인 범위 점검, 2) 두 개의 구성요소 또는 속성간의 의존성, 3) 정상적인 표현에 대한 패턴 매칭, 4) 복합 제약식(complex constraints)으로 구분하였다[Jacinto et al., 2002]. 하지만 ebXML BPSS 내의 WFR들은 대부분 복합 제약식에 해당하므로, 본 연구에서는 이를 보다 세분하여 <그림 3>와 같이 구분하도록 한다.



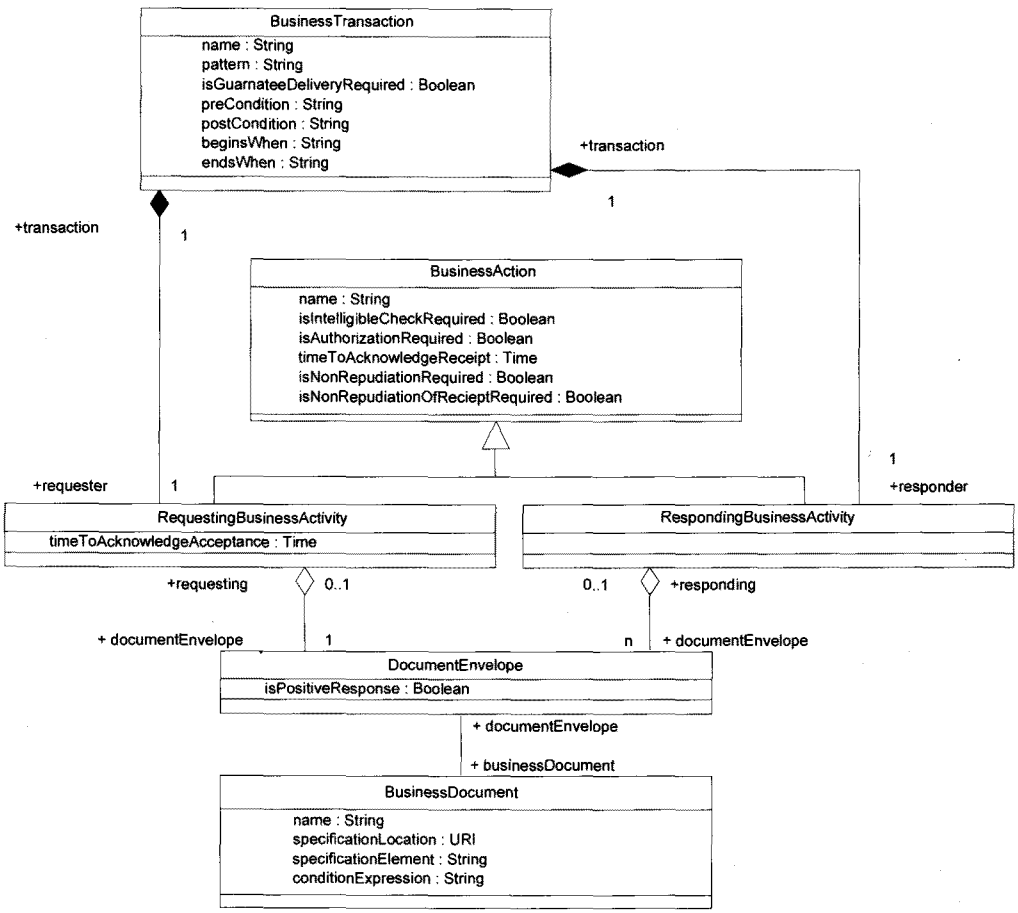
<그림 3> ebXML BPSS 제약식의 분류

3.2 단순제약

단순제약은 한 속성값 또는 한 연관관계의 매핑수에 국한된 제약 조건을 의미한다. 속성값에 대한 제약으로는 속성의 자료형 제약(Simple:AttributeValue:Type; SAT)과 속성값의 범위 제약(Simple:AttributeValue:Range; SAR)이 포함된

다. 연관관계의 매핑수 제약은 연관관계에 참여하는 인스턴스(UML 용어로는 링크) 수와 관련된 제약으로, 최소 링크 수 제약(Simple:MultiplicityofAssociation:Minimal; SMM), 정확히 명시된 숫자의 링크를 가져야 하는 특정 링크 수 제약(Simple:MultiplicityofAssociation:Exact; SME), 최대 링크수 제약(Simple:MultiplicityofAssociation:mAximum; SMA)으로 구분된다.

28개의 WFR에 대한 제약 구분은 <부록 1>에 4번째 열에 표시되어 있다. <그림 4>는 ebXML BPSS내의 Business Transaction과 관련된 클래스와 연관관계를 포함하는 클래스 다이어그램이다. WFR 27은 <그림 4>의 객체 DocumentEnvelope와 RespondingBusinessActivity, RequestingBusinessActivity 간의 연관관계의 매핑수에 대한 두 개의 제약이다. 즉, WFR 27은 RespondingBusinessActivity는 0 또는 1개의 문서 흐름을 가져야 하며(SMA 제약), RequestingBusinessActivity에 관련된 문서는 하나이어야 함을 명시한 제약식이다(SME 제약). 이러한 제약식을 OCL로 표현하면 다음과 같다. 아래 OCL 표현에서 'size'는 OCL 내에 기정의된 함수로 연관성에 참여하는 객체 인스턴스 수를 반환하는 함수이다. 단순제약의 경우는 BPSS의 UML 클래스다이어그램 내에서 표현되며 상대적으로 단순하다. 단순제약식의 경우 XML 버전의 BPSS는



<그림 4> Business Transaction 관련 클래스 다이어그램(Source: [ebXML, 2002a])

XML 파서에 의해서 검증이 가능하다.

[WFR 27-1]
content DocumentEnvelope **inv:**
self.respondingBusinessActivity → size <= 1

[WFR 27-2]
content RequestingBusinessActivity **inv:**
self.documentEnvelope → size = 1

3.3 정상적인 표현에 대한 패턴 매칭 제약

정상적인 표현에 대한 패턴 매칭 제약은 속성값의 표현을 특정 표기 기법을 맞추어서 작성해야 하는 제약을 의미한다. 예를 들어, ebXML BPSS에서 시간에 대한 표기법이 이러한 예가 된다. 즉, 수령일자, 확인일자 등에 대한 명세가 "P5D"(현재부터 5일 이내), "19940508/P1Y6M"(1994년 5월 8일부터 1년 6개월간) 형태로 표현되는데, 이는 UN/CEFACT에서 개발한 시간, 일자 표현을 위한 표준인 ISO-8601에 따른 표기이다[UN/CEFACT, 1998]. 따라서 시간에 대한 명세가 이러한 표기법에 맞도록 기재되었는지를 검증하는 것이 필요하다. 패턴 매칭 제약은 간단한 경우를 제외하고는 UML 클래스 다이어그램 내에서 표현이 불가능하고, OCL을 이용하여 명세가 가능하다. 이러한 패턴 매칭 제약은 별도의 프로그램 형태로 작성되어 XML 파싱 과정에서 검증된다.

3.4 복합제약

복합제약은 두 개 이상의 속성값 또는 두 개 이상의 객체 인스턴스의 상태가 관련된 제약을 의미한다. 크게 속성값에 대한 제약과 객체 인스턴스 수에 대한 제약으로 구분된다. 속성값에 대한 제약은 다시 한 객체 인스턴스 내의 두 개 이상의 속성값이 관련된 제약(단일 인스턴스 속성값 복합제약, Complex:AttributeValue:SingleInstance, CAS)과 두 개 이상의 객체 인스턴스의 속성값이 관련된 다수 인스턴스 속성값 복합제약(Complex: AttributeValue:MultipleInstances; CAM)으로 구분된다.

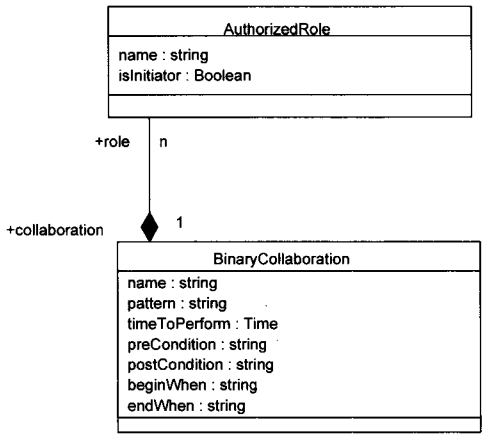
인스턴스 수 제약은 객체 인스턴스 수와 관련된 제약으로 최소 인스턴스 수 제약(Complex: InstanceCardinality:Minimal; CIM), 정확히 명시된 숫자값을 가져야 하는 특정 인스턴스 수 제약(Complex:InstanceCardinality:Exact; CIE), 최대 인스턴스 수 제약(Complex:InstanceCardinality:mAximum; CIA), 특정한 상황에서는 인스턴스가 존재해서는 안 되는 인스턴스 비존재 제약(Complex:InstanceCardinality:Non-existencial; CIN)으로 구분된다. 적어도 하나의 인스턴스가 존재해야 한다는 형태의 존재성 제약은 최소 인스턴스 수 제약(CIM)에 포함된다.

3.4.1 복합제약: 속성값 제약

WFR 14와 WFR 28는 복합제약 중 속성값 제약의 예이다. WFR 14는 RequestingBusinessActivity에서 확인수령일자가 확인승인일자보다 먼저이어야 한다는 제약조건을 표현하고 있다(<그림 4> 참조). 이 경우 동일한 객체 RequestingBusesssActivity 인스턴스의 두 속성 timeToAcknowledgeAcceptance, timeToAcknowledgeReceipt간에 관계에 대한 제약이므로, CAS(Complex:AttributeValue:SingleInstance)에 해당한다. WFR 14를 OCL로 표현하면 다음과 같다.

[WFR 14]
content RequestingBusinessActivity **inv:**
timeToAcknowledgeAcceptance <> '' **and**
timeToAcknowledgeReceipt <> ''
implies
timeToAcknowledgeAcceptance >
timeToAcknowledgeReceipt

WFR 28은 BinaryCollaboration에서 하나의 Business Partner의 역할(Role)이 initiating과 responding이 동시에 될 수는 없음을 표현하는 제약식이다(<그림 5> 참조). 이 경우 제약식의 만족여부를 검토하기 위해서는 하나의 Binary-Collaboaration 객체 인스턴스와 연관 관계를 가지고 있는 AutorizedRole 객체 인스턴스들 중에 속성 'name'값이 같으면서 동시에 속성 'isInitiator'값이 같은 두 인스턴스가 존재하는지를 점검해야 한다. 따라서 이 제약은 다수 인스턴스 속성값 제약(CAM)에 해당한다. WFR 28를 OCL로 표현하면 다음과 같다.



<그림 5> BinaryCollaboration 관련 클래스 다이어그램(Source: [ebXML, 2002a])

[WFR 28]
content BinaryCollaboration **inv:**
self.authorizedRole->forAll(p1 :
authorizedRoles, p2 : authorizedRoles |
p1.isInitiator = true, p2.isInitiator =
false **implies** p1.name != p2.name)

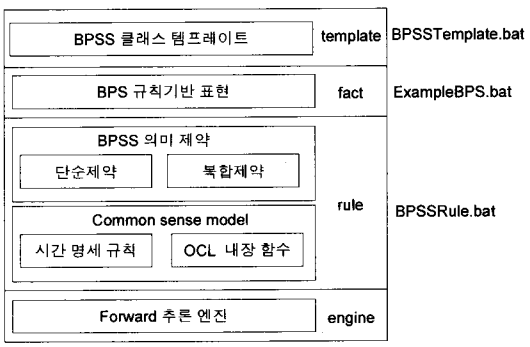
3.4.2 복합제약: 인스턴스 수

복합제약 중 인스턴스 수에 대한 제약들은 속성값이나 객체 인스턴스 존재여부에 따라 객체 또는 연관관계 인스턴스의 존재 여부 또는 숫자를 제한하는 제약조건들이다. 예를 들어 WFR 19는 이행거부불가(nonRepudiation) 조건이 필요한 RequestingBusinessActivity의 경우에는 이에 대응되는 RespondingBusinessAcitivity가 적어도 하나의 문서를 포함하여야 한다는 제약조건이다(<그림 4> 참조). 이 경우는 적어도 하나의 연관관계 인스턴스를 가져야 하고, 이를 검토하기 위해서 참조되어야 하는 객체 인스턴스의 수가 두 개 이상이므로, 복합제약 중 최소 인스턴스 수 제약(Complex:InstanceCardinality:Minimum; CIM)에 해당한다. WFR 19를 OCL로 표시하면 다음과 같다.

[WFR 19]
content RespondingBusinessAcitivity **inv:**
self.BusinessTransaction.Requesting
BusinessActivity.isNonRepudiationRequired
= 'true'
implies
self.DocumentEnvelope → size > 0

Ⅳ. 규칙기반 제약 검증

정형화된 제약식의 표현은 BPSS로 작성된 BPS의 검증을 위해서 활용될 수 있다. 본 절에서는 정형화된 제약식을 바탕으로 BPS를 검증



<그림 6> 규칙언어 기반 BPS 검증

하기 위한 규칙기반 언어인 CLIPS를 활용한 방안을 예제를 중심으로 제시한다. <그림 6>은 BPS를 검증하기 위한 기본 구조이다. 먼저 eb-

```
; CLIPS Program for ebXML BPSS Validation
; Class Templates

(deftemplate BusinessTransaction                          ①
   (slot id)
   (slot name)
   (slot pattern)
   (slot isGuarnateeDeliveryRequired)
   (slot preCondition)
   (slot postCondition)
   (slot besignsWhen)
   (slot endsWhen)
)

(deftemplate RequestingBusinessActivity                   ②
   (slot id)
   (slot name)
   (slot isIntelligibleCheckRequired)
   (slot isAuthorizationRequired)
   (slot timeToAcknowledgeReceipt)
   (slot isNonRepudiationRequired)
   (slot isNonRepudiationOfRecieptRequired)
   (slot timeToAcknowledgeAcceptance)
)

(deftemplate RespondingBusinessActivity                   ③
   (slot id)
   (slot name)
   (slot isIntelligibleCheckRequired)
   (slot isAuthorizationRequired)
   (slot timeToAcknowledgeReceipt)
   (slot isNonRepudiationRequired)
   (slot isNonRepudiationOfRecieptRequired)
)

(deftemplate DocumentEnvelope                             ④
   (slot id)
   (slot isPositiveResponse)
)

(deftemplate BusinessDocument                             ⑤
   (slot id)
   (slot name)
   (slot specificationLocation)
   (slot specificationElement)
   (slot conditionExpression)
)
```

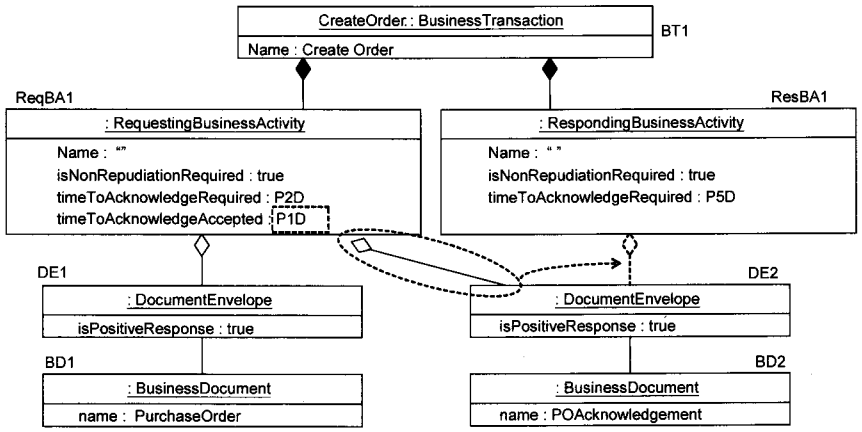
<그림 7> BPSS의 클래스 템프레이트 예제

XML BPSS의 UML 클래스 다이어그램에 존재하는 클래스들이 CLIPS의 템프레이트(template)로 사전에 정의된다(본 연구의 예제에서는 "BPSS Template.bat"라는 파일로 정의됨). 예를 들어, <그림 4>에 표시된 클래스 중 일부에 대한 CLIPS 표현은 <그림 7>과 같다.

```
; CLIPS Program for ebXML BPSS Validation
; Association Templates

;BusinessTransaction-to-RequestingBusinessActivity        ①
(deftemplate BT-ReqBA
   (slot transaction)
   (slot requester)
)

;BusinessTransaction-to-RespondingBusinessActivity        ②
(deftemplate BT-ResBA
   (slot transaction)
   (slot responder)
)

;RequestingBusinessActivity-to-DocumentEnvelope           ③
(deftemplate ReqBA-DE
   (slot ReqBA)
   (slot DE)
)

;RespondingBusinessActivity-to-DocumentEnvelope           ④
(deftemplate ResBA-DE
   (slot ResBA)
   (slot DE)
)

;DocumentEnvelope-to-BusinessDocument                     ⑤
(deftemplate DE-BD
   (slot DE)
   (slot BD)
)
```

<그림 8> BPSS의 연관관계 템프레이트 예제

<그림 7>의 ①, ②, ③, ④, ⑤는 각각 클래스 BusinessTransaction, RequestingBusinessActivity, RespondingBusinessActivity, DocumentEnvelope, BusinessDocument에 해당하는 템프레



<그림 9> BPS 예제(UML 표현)

이트이다. 또한 <그림 8>의 ①, ②, ③, ④, ⑤는 <그림 4>에 포함된 5개의 연관관계 BT-ReqBA (BusinessTransaction과 RequestingBusinessActivity 간), BT-ResBA(BusinessTransaction과 RespondingBusinessActivity 간), ReqBA-DE(RequestingBusinessActivity와 DocumentEnvelope DocumentEnvelope 간), DE-BD(DocumentEnvelope와 BusinessDocument 간)에 해당하는 템프레이트이다.

특정 기업간의 비즈니스 트랜잭션을 표현하는 하나의 BPS는 이러한 템프레이트에 사실(fact)로 표현된다. <그림 9>는 ebXML BPSS 공식 문서에 포함된 예제 BPS의 UML 표현 중 일부분인데, 본 논문에서 예제로 활용하도록 한다. 실제로 본 논문의 목적상 <그림 9> 내에는 두 가지 오류가 포함되도록 수정되었는데, RequestingBusiness Activity ReqBA1의 속성 timeToAcknowledge Accepted가 "P3D"로 기재되어야 할 것이 BPS 설계자의 실수로 "P1D"로 기재되었고, Responding BusinessActivity ResBA1으로 연결되어야 할 DocumentEnvelope DE2가 ReqBA1으로 잘못 연결되어있다. <그림 10>은 이 BPS를 XML 형태로 표현한 것이다. <그림 10>의 XML 표현 내에도 <그림 9>와 마찬가지로 두가지 오류가 그대로 포함되어 있다. 실제로 도식적으로 표현된 UML 형태의 BPS에 비해서 문법위주의 XML 형태의 BPS 표현에서 오류를 찾아내기가 더 어렵다. <그림 9>의 클래스 다이어그램에 포함된 객체 인스턴스들과 연관관계의 인스턴스들이 검증을 위해서 <그림 11>에서의 같이 CLIPS 사실들의 집합으로 매핑된다. <그림 11>의 위 쪽 ①에는 Business Transaction, RequestingBusinessActivity, RespondingBusinessActivity, DocumentEnvolope, BusinessDocument 객체 인스턴스들에 대한 정의이고, 아래 쪽 ②에는 이들 객체 인스턴스들간의 연관관계 인스턴스들이 정의되어 있다. BPS를 <그림 11>에서와 같이 사실들의 집합으로 매핑할 때, 각 객체 인스턴스의 ID를 시스템이 자동적으로 할당하게 된다. 이것은 XML 버전의 BPS에서는 객체간의 중첩된 구조로 ID없이도 연관관계를 알 수 있으나, UML 버전에서는 이러한 중첩 구조가 아니므로, 이를 ID를 통해서 표현하도록 한다.

제약식의 규칙 표현을 위해서는 'Common sense model'이 필요하다. 여기서 'Common sense model'은 OCL 내장 함수들과 시간 명세 규칙을 가지고 있다. OCL 자체의 정의에는 다수의 내장함수들이 존재한다. 예를 들어 WFR 27의 기술에 활용된 연관관계에 참여하는 인스턴스의 수를 세는 'size'와 같은 함수가 이러한

예이다. 또한 시간 명세 규칙은 ebXML이 따르고 있는 ISO-8601의 시간표현 규칙을 의미한다 [UN/CEFACT, 1998]. 이러한 내장함수와 규칙들은 특정 BPS에 따라 결정되는 것이 아니라, 모든 BPS에 적용되는 공통 모델에 해당한다.

```
.....
<BusinessTransaction name = "CreateOrder" >
    <RequestingBusinessActivity name = ""
        isNonRepudiationRequired = "true"
        timeToAcknowledgeReceipt = "P2D"
        timeToAcknowledgeAcceptance = "P1D" >
    <DocumentEnvelope isPositiveResponse = "ture"
        businessDocument = "PurchaseOrder" />
    <DocumentEnvelope isPositiveResponse = "ture"
        businessDocument = "POAcknowledgement" />
    </RequestingBusinessActivity>
    <RespondingBusinessActivity name = ""
        isNonRepudiationRequired = "true"
        timeToAcknowledgeReceipt= "P5D" >
    </RespondingBusinessActivity>
</BusinessTransaction>
....
```

"P3D"

<그림 10> BPS 예제(XML 표현)

앞 장에서 제시된 OCL로 표현된 ebXML에 대한 제약식들은 CLIPS의 규칙 형태로 표현된다. <그림 12>는 앞 절에서 OCL로 표현한 4개의 규칙에 대한 CLIPS 표현이다. <그림 12>의 ①은 WFR 14에 대한 것으로 조건절에 사용된 previous 함수는 시간명세 규칙의 일부로, 두개의 ISO 8601 형식으로 작성된 시간을 비교해서 앞에 것이 빠르면 true를, 그렇지 않으면 false를 반환하는 함수이다. 마찬가지로 ②는 WFR 28에 대한 CLIPS 규칙 표현이고, ③은 복합제약: 최소 인스턴스 수 제약(CIM)인 WFR 19에 대한 CLIPS 규칙 표현이다. WFR 19는 이행거부불가(non-Repudiation) 조건이 필요한 RequestingBusinessActivity의 경우에는 이에 대응되는 RespondingBusinessAcitivity가 적어도 하나의 문서를 포함하여야 한다는 제약조건이다. <그림 12>의 ④는 WFR 27의 CLIPS 표현이다. Ⅲ장에서 WFR 27은 두 개의 OCL 제약식으로 표현되었는데, 첫번째 제약식에 해당하는 CLIPS 표현이 규칙 checkWFR27-1이다. 규칙의 내용은 앞에서 언급했듯이 특정 DocumentEnvelope 인스턴스와 연관관계를 갖는 RespondingBusinessActivity 인스턴스가 1개 이하이어야 한다는 것이다. WFR27의 두 번째 제약식은 RequestingBusiness Activity와 DocumentEnvelope간의 연관관계에 관한 것이다. 즉, 특정 RequestingBusinessActivity와 연관성을 갖는 DocumentEnvolope 인스턴스의 수가 정확하게 1이어야 한다는 규칙이다. 이 두번째 규칙은 <그림 12>에서 보면 두 개의 규칙 checkWFR27-2-1, checkWFR27-2-2로 표현되어있다. 첫 번째 것은 특정 RequestBusinessActivity에 대하여 적어도 하나의 연관관계를 갖는 DocumentEnvelope이 존재해야 한다는 규칙이고, 두 번째 규칙은 연관관계를 갖는 DocumentEnvelope이 존재한다면 그 개수는 1개이어야 한다는 것을 표현하고 있다. SMM(특정 링크 수 제약)이나 CIE(특정 인스턴스 수 제약)와 같이 특정 숫자와 동등함을 요구하는 제약은 CLIPS로 변환 시에 두 개의 부등식 제약들(즉, '작거나 같다'와 '크거나 같다'는 제약)로 매핑된다.

특정 BPS가 의미적으로 정확하게 작성되었는지 점검하기 위해서, <그림 6>에서와 같이 클래스 템프레이트, 해당 BPS의 사실 집합, BPSS 의미 제약, common sense model들을 기반으로 전진추론(forward inference)을 수행한다. 만일 사실 형태로 표현된 특정 BPS 내에 의미적 오류가 포함되어 있는 경우에 의미 제약 규칙들에 의해 오류가 찾아진다. 예를 들어, <그림 7> <그림8>의 템프레이트, <그림 11>의 예제 BPS, <그림 12>의 규칙(BPSS 의미 제약과 common sense model 포함)들이 각각 "BPSSTemplate.bat", "ExampleBPS.bat", "BPSSRule.bat"로 저장되어 있다면, <그림 13>(a)는 CLIPS 내에서 예제 BPS를 점검하기 위한 명령어들이다. 이러한 명령어들을 순차적으로 실행하면, 예제 BPS에 존재하는

```
; Example BPS (Define Facts)

(deffacts BusinessTransactions
   (BusinessTransaction (id BT1) (name CreateOrder))
)

(deffacts RequestingBusinessActivities
   (RequestingBusinessActivity (id ReqBA1)
        (name CreateOrderRequesting)
        (isNonRepudiationOfRecieptRequired true)
        (timeToAcknowledgeReceipt P2D)
        (timeToAcknowledgeAcceptance P1D)  ;P3D
    )
)

(deffacts RespondingBusinessActivities
   (RespondingBusinessActivity (id ResBA1)
        (name CreateOrderResponding)
        (isNonRepudiationOfRecieptRequired true)
        (timeToAcknowledgeReceipt 5) ;P5D
   )
)

(deffacts DocumentEnvelopes
   (DocumentEnvelope (id DE1)
        (isPositiveResponse true))
   (DocumentEnvelope (id DE2)
        (isPositiveResponse true))
)

(deffacts BusinessDocuments
   (BusinessDocument (id BD1)
        (name PurchaseOrder))
   (BusinessDocument (id DE2)
        (name PO_Acknowledgement))
)

(deffacts BT-ResBAs
    (BT-ResBA (transaction BT1) (responder ResBA1))
)
(deffacts BT-ReqBAs
    (BT-ReqBA (transaction BT1) (requester ReqBA1))
)

(deffacts ResBA-DEs
;   (ResBA-DE (ResBA BA1) (DE DE2))
)

(deffacts ReqBA-DEs
   (ReqBA-DE (ReqBA ReqBA1) (DE DE1))
   (ReqBA-DE (ReqBA ReqBA1) (DE DE2))
)

(deffacts DE-BDs
   (DE-BD (DE DE1) (BD BD1))
   (DE-BD (DE DE2) (BD BD2))
)
```

① ②

<그림 11> BPS 예제(CLIPS)

```
; WFR 14
(defrule checkWFR14
   (RequestingBusinessActivity (name ?name)
      (timeToAcknowledgeReceipt ?TimeToReceipt)
      (timeToAcknowledgeAcceptance ?TimeToAcceptance))
      (previous ?TimeToReceipt ?TimeToAcceptance))
   =>
   (printout outfile "WFR14 Validation Error ! :
            RequestingBusinessActivity" ?name crlf)
)

;WFR 28
(defrule checkWFR28
   (BinaryCollaboration (id ?BC))
   (AuthorizedRole (id ?Arid1) (name ?ARname1)
                   (isInitiator true))
   (AuthorizedRole (id ?ARid2) (name ?ARname2)
                   (isInitiator false))
   (BC-AR (BC ?BC) (AR ?ARid1))
   (BC-AR (BC ?BC) (AR ?ARid2))
   (test (eq ?ARname1 ?ARname2))
   =>
   (printout outfile "WFR28 Validation Error ! :
   BinaryCollaboration" ?BCid "has the same
   initiator and terminator " crlf)
)

(defrule checkWFR19
    (RespondingBusinessActivity (id ?ResBA))
    (BT-ResBA (transaction ?BT) (responder ?ResBA))
    (BT-ReqBA (transaction ?BT) (requester ?ReqBA))
    (RequestingBusinessActivity (id ?ReqBA)
        (isNonRepudiationOfRecieptRequired true))
    (not (exists (ResBA-DE (ResBA ?ResBA))))
   =>
    (printout outfile "WFR19 Validation Error ! :
     RespondingBusinessActivity" ?ResBA "must have a
     DocumentEnvelope." crlf)
)

;WFR 27
(defrule checkWFR27-1
   (ResBA-DE (ResBA ?ResBA1) (DE ?DE))
   (ResBA-DE (ResBA ?ResBA2) (DE ?DE))
   (test  (neq ?ResBA1 ?ResBA2))
   =>
   (printout outfile "WFR27-1 Validation Error ! :
    Document Envelope" ?DE "has more than two
    RespondingBusinessActivity" ?ResBA1  ", " ?ResBA2 crlf)
)
(defrule checkWFR27-2-1
   (RequestingBusinessActivity   (id ?ReqBA1))
   (not (exists (ReqBA-DE (ReqBA ?ReqBA1))))
   =>
    (printout outfile "WFR27-2 Validation Error ! :
      RequestingBusinessActivity" ?ReqBA1
      "does not have DocumentEnvelope"  crlf)
)
(defrule checkWFR27-2-2
   (ReqBA-DE (ReqBA ?ReqBA) (DE ?DE1))
   (ReqBA-DE (ReqBA ?ReqBA) (DE ?DE2))
   (test  (neq ?DE1 ?DE2))
   =>
   (printout outfile "WFR27-2 Validation Error ! :
    RequestingBusinessActivity" ?ReqBA "has more than
    two  DocumentEnvelope" ?DE1 ", " ?DE2  crlf)
)
```

① ② ③ ④

<그림 12> 제약식의 CLIPS 표현

오류들에 대한 메시지는 <그림 13>(b)와 같이 "check.txt" 파일 내에 출력된다. 이러한 정보를 참고하여 BPS 설계자는 예제 BPS의 두 가지 오류를 발견하고 수정할 수 있다.

```
(load "BPSSTemplate.bat")
(load "ExampleBPS.bat")
(load "BPSSRule.bat")
(reset)
(open "check.txt" outfile "w")
(run)
(close outfile)
```

(a) CLIPS 내에서 제약 점검 실행

```
WFR27-2 Validation Error ! :
  RequestingBusinessActivity ReqBA1 has more
  than two DocumentEnvelope DE1,  DE2
WFR19 Validation Error ! :
  RespondingBusinessActivity ResBA1 must
  have a DocumentEnvelope.
WFR14 Validation Error ! :
  RequestingBusinessActivity
  CreateOrderRequesting TimeToReceipt P2D is
  not previous TimeToAcceptance P1D
```

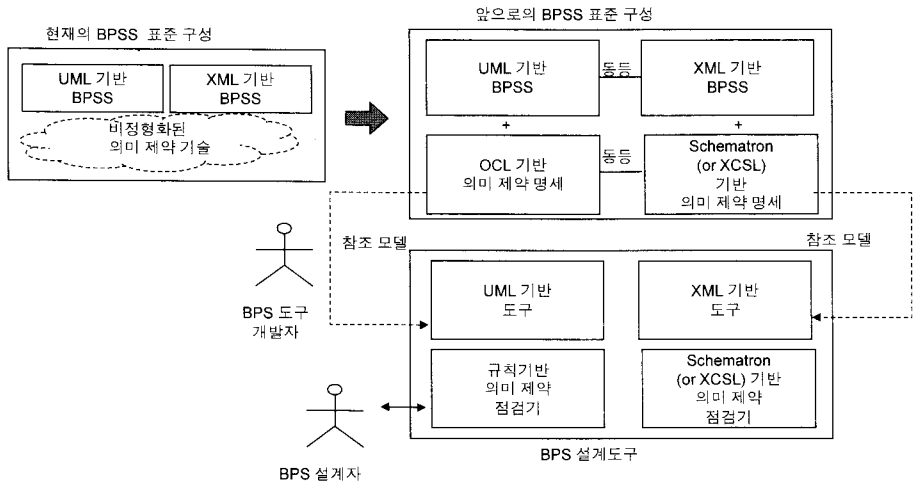
(b) 제약 점검 결과("check.txt")

<그림 13> 제약 점검 추론의 실행

V. 논의사항

5.1 제약 표현의 의의

앞에서 언급했듯이, 현재의 ebXML BPSS의 공식 문서에는 UML 버전과 XML 버전 두 가지가 함께 제시되고 있다. 하지만, BPSS 내에 포함된 비즈니스 프로세스를 표현하기 위한 구성요소들이 가지는 의미 제약들은 서술적으로 비정형된 형태로 제시되고 있다(<그림 14> 참조). 아직까지 이러한 의미 제약들을 정형화하려는 연구들은 시도되고 있지 못하다. 따라서 본 연구에서는 이러한 정형화의 필요성을 제기하고, UML 버전의 BPSS를 중심으로 의미적 제약식 표현과 활용에 대하여 연구하였다. 이와 병행하여 XML 기반의 BPSS에 대한 제약 표현과 활용에 대한 연구가 함께 필요할 것으로 보인다. 실제로, 현재 XML의 제약표현언어로 연구되고 있는 Schematron, XCSL(XML Constraint Specification Language), XML-Schemas 등을 활용한 제약의 표현 및 검증이 가능할 것으로 보이며, 5.2절에서 구체적으로 Schematron을 사용하는 방안을 간략히 제시하도록 한다. 하지만 XML 제약표현언어의 경우는 아직까지 표준화가 이루어지고 있지



<그림 14> UML기반 vs. XML기반 제약 표현

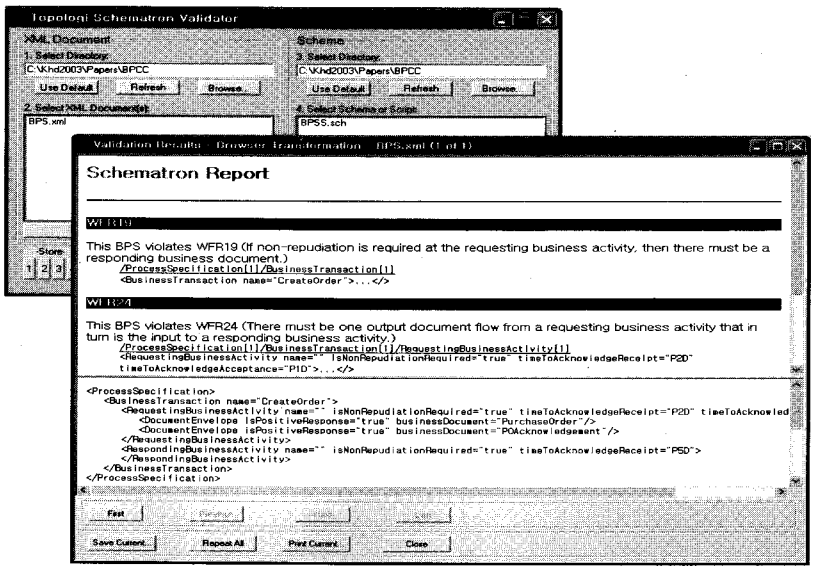
못해서, 표준화 추이에 맞추어 관련 연구를 수행하는 것이 바람직하다.

정형화된 제약 표현은 특정 기업간 거래를 위한 BPS 설계 시에 BPS 설계자가 BPS를 생성하는 과정을 지원하는데 활용될 수 있다. 즉, BPS 생성을 위해서 사용하는 BPS 설계도구들이 BPS 설계자가 작성하는 BPS 모델의 중간산출물이나 최종결과를 UML 기반의 BPS 또는 XML 기반의 BPS 형태로 내재적으로 표현하게 된다. OCL 기반의 제약 표현은 BPS 설계도구 개발자들에 의해서 CLIPS와 같은 규칙기반 언어로 변환되어, UML 기반 설계도구에 내장되어 BPS 설계 과정을 도와줄 수 있다(<그림 14> 참조). XML 기반 도구를 활용하는 경우는 Schematron 또는 XCSL 등 XML 제약표현언어로 명세된 제약식들을 각 제약언어별로 제시된 검증 방안이나 도구를 사용하여 오류 점검이 가능하다.

현재 WellFormedness Rule(WFR) 형태로 제시되고 있는 BPSS 내의 의미 제약들은 완전성이 보장되지 못한 상태이다. 즉, 현재의 28개 WFR이 BPSS 내의 모든 의미 제약들을 표현하고 있다고 보장할 수 없다. 따라서 본 연구와 같은 제약들의 정형화 노력을 통해서 WFR에 포함되지 않은 추가적인 의미 제약들의 도출이 가능할 것으로 보이며, 궁극적으로 BPSS의 의미적 완성도를 높일 수 있다. 또한 본 연구에서 제시된 제약식의 분류 체계는 이와 같은 추가적인 제약 도출에 도움을 줄 수 있으며, 제약식의 체계적인 관리에도 도움을 준다.

5.2 Schematron을 사용한 검증과 OCL 활용과 비교

본 절에서는 XML 제약표현언어 중 하나인 Schematron을 사용하는 경우, BPS 의미제약 점검이 어떻게 이루어질 수 있는 지를 살펴본다. <그림 15>는 Schematron을 기반으로 작동하는 공개 소프트웨어[Topologi, 2003]를 이용하여 BPS를 검증하는 작업을 보여준다. <그림 16>은 WFR19와 WFR24를 Schematron으로 표현한 것으로 BPSS.sch 파일 내용 중 일부분이다. 여기서는 이 의미제약 파일을 이용하여 BPS.xml 파일



<그림 15> Schematron을 이용한 XML 의미제약 검증

```
< schema xmlns = "http://www.ascc.net/xml/schematron" >
    < pattern name = "WFR19" >
        < rule context = "//BusinessTransaction" >
            < assert
test = "RequestingBusinessActivity/@isNonRepudiationRequired and RespondingBusinessActivity/
DocumentEnvelope" //
                This BPS violates WFR19 (If non-repudiation is required at the requesting
business activity, then there must be a responding business document.)
            </ assert >
        </ rule >
    </ pattern >
    < pattern name = "WFR24" >
        <rule
context = "//BusinessTransaction/RequestingBusinessActivity" >
            <assert test = "count(DocumentEnvelope)=1" >
                This BPS violates WFR24 (There must be one output document flow from
a requesting business activity that in turn is the input to a responding business activity.)
            </ assert >
        </ rule >
    </ pattern >
</ schema >
```

<그림 16> Schematron을 이용한 BPSS 의미제약 표현

(그 내용은 <그림 15>의 Validation Results 윈도의 하단 부분에 제시되어 있으며, <그림 10>의 내용을 포함)을 검증하고 있는데, 그 결과로 WFR19와 WFR24가 위반되었음을 알려준다. XML 제약표현언어를 이용하여 BPS를 검증하는 작업은 문법적, 구조적 수준에서 XML 문서에 대한 정확한 제약의 표현이 가능하나, 복잡한 의미제약의 경우 초보자가 이해하기 어렵고, 표현하기도 쉽지 않다. 그러나, 비즈니스 프로세스 실행을 위해서는 최종적으로 XML기

<표 2> 제약의 OCL 표현과 XML 의미제약 표현 비교

기 준	OCL 표현	XML 의미제약 표현
적용 대상	UML 기반 BPS	XML 기반 BPS
표준화	객체지향 모델링 표준 언어인 UML의 객체에 대한 제약 표현 표준 언어	Schematron, XCSL 등 의미제약 표현 언어가 다수 있으며, 표준화 필요(DSDL[Holman, 2002] 표준화 진행현황 참조)
표현력	개념적인 수준에서 모델의 의미제약 표현력과 이해 가능성이 우수함	문법적/구조적 수준에서 XML 문서에 대한 정확한 제약의 표현이 가능하나, 표현이 어렵고 복잡함
규칙 매칭	반복적(recursive) 추론에 의한 의미 제약의 검증이 가능	반복적인 추론의 표현과 실행이 매우 어려움
활용 시점	개념적 수준에서 UML기반의 BPS를 생성하는 과정에서 안내 역할을 할 수 있으며, 최종적으로 검증하여 XML기반 BPS를 자동 생성하고자 할 경우 적용	XML기반의 BPS를 직접 작성하거나, UML 기반의 BPS로부터 자동 생성된 XML기반 BPS를 수정한 경우 활용될 수 있음

반의 BPS가 생성되어야 하고, UML기반의 BPS로부터 XML기반의 BPS를 자동 생성한 경우라 하더라도 수정이 불가피한 경우가 있으며, XML에 익숙한 사용자는 직접 XML기반의 BPS를 작성할 수도 있기 때문에, XML 수준에서의 검증도 반드시 필요하다. 따라서 앞에서 제시된 개념적인 수준에서의 의미제약이 XML과 같은 문법적인 수준에서의 제약으로 자동적으로 변환되어 적용될 수 있다면 바람직하겠다. <표 2>는 의미제약을 OCL과 XML기반으로 표현할 경우의 특성을 비교한 것이다.

Ⅵ. 결 론

본 연구에서는 ebXML BPSS(Business Process Specification Schema) 내에 포함된 구성요소들의 의미 제약을 정형화하고, 이를 활용하여 BPSS에 맞추어 작성된 BPS Business Process Specification)의 정확성을 검증하기 위한 규칙 기반 방안을 제시하였다. 현재 ebXML BPSS에 대한 표준 명세가 OASIS와 UN/CEFACT에 의해서 수립되어 있으며 이를 바탕으로 국내외에서 상용화된 제품들도 개발되고 있다. 하지만, ebXML의 실질적인 확산을 위해서는 현재 BPSS 표준 명세가 앞으로도 계속 보완되는 것이 필요할 것으로 보이며, 그 중의 하나로 BPSS 구성요소들의 의미에 대한 보다 정형화된 표현이 필요할 것으로 보인다. 본 연구에서는 BPSS 내의 구성요소들에 포함된 의미 제약들의 분류 체계를 제시하였다. 제시된 제약식의 분류체계는 단순제약, 정상적인 표현에 대한 패턴 매칭, 복합제약, 3가지로 크게 분류되고, 단순제약과 복합제약을 더 상세하게 구분하여 12개의 유형으로 분류하였다. 또한 다양한 형태의 의미 제약식들이 OCL (Object Constraint Language)을 활용하여 정형화될 수 있음을 예시적으로 설명하였다. 이러한 정형화된 제약식들을 활용하여 개별 BPS의 정확성을 점검하기 위해서 구체적인 구현 방안으로 규칙기반 언어인 CLIPS를 활용한 구현 방안을 제시하였다. 현재 본 연구의 연장선상에서 OCL 기반 제약식 표현을 CLIPS와 같은 규칙기반언어로 자동 또는 반자동적으로 매핑하는 방안에 대한 연구가 수행되고 있다.

현재 본 논문에서 제시된 OCL 기반의 제약식의 표현은 UML 버전의 ebXML BPSS에 적합하도록 작성되었다. 이와 병행하여, XML 버전의 ebXML BPSS에 적합한 정형화된 의미 제약의 표현도 함께 필요할 것으로 보인다. 본 논문에서는 OCL로 표현된 의미 제약들이 XML 제약표현언어 중 하나인 Schematron을 활용하여 표현되고 검증에 활용될 수 있음을 예시적으로 보였다. 추후 연구로 OCL 기반의 제약 표현과 XML 제약표현언어의 표현간의 자동화된 변환에 대한 연구가 필요하다. 또한 본 연구에서 제시된 제약 표현 및 검증 방안을 활용한 지능형 BPS 설계도구의 개발이 필요하다.

〈참 고 문 헌〉

[1] CLIPS, *CLIPS Reference Manual (Version 6.20)*, http://www.ghg.net/clips/CLIPS.html, 2002.

[2] ebXML, Business Process Project Team, *ebXML Business Process Specification Schema (Version 1.03)*, UN/CEFACT and OASIS, May 2001a.

[3] eXML, *Business Process Project Team, Business Process Analysis and Guidelines v1.0*, UN/CEFACT and OASIS, May 2001b.

[4] ebXML, Business Process Project Team, *ebXML Business Process Specification Schema(Version 1.05)*, UN/CEFACT and OASIS, June 2002a.

[5] ebXML, "Enabling a Global Electronic Market," http://www.ebxml. org, 2002.

[6] Jacinto et al., Jacinto, Marta Henriques, Librelotto, Giovani Rubert, Ramalho, Jose Carlos Leite, Henrigues, Pedro Randgel, "Constraint Specification Languages: Comparing XCSL, Schematron, and XML-Schemas," *Proceedings of the XML Europe 2002*, Barcelona, May 2002.

[7] Holman, G.K. Holman, "ISO/IEC 19757 - DSDL (Document Schema Definition Languages," http://www.dsdl.org, 2002.

[8] Jelliffe, Rick Jelliffe, "The Schematron: An XML Structure Validation Language using Patterns in Trees," http://www.ascc.net/xml/resource/schematron/schematron.html, 2002.

[9] Kim, HyoungDo, "Conceptual Modeling and Specification Generation for B2B Business Process based on ebXML," *ACM SIGMOD Record*, Vol. 31, No. 1, March 2002.

[10] Kim, HyoungDo, "Model Management Support for B2B Electronic Commerce," Proceedings of the 2nd Asian eBiz Workshop, Seoul, August 2002.

[11] OAGI, Open Applications Group, http://www.openapplications.org/, 2002.

[12] OASIS, *OASIS Universal Business Language TC*, http://www.oasis- open.org/committees/ubl/, 2002.

[13] OMG, *OMG Unified Modeling Language Specification (Version 1.3)*, OMG, 1999.

[14] Rumbaugh, James, Jacobson, Ivar, Booch, Grady, *The Unified Modeling Language Reference Manual*, Wesley, 1999.

[15] Ramalho, Jose Carlos, "Constraining Content: Specification and Processing," *XML Europe 2001*, 2001.

[16] Topologi Pty. Ltd., "Schematron Validator," http://www.topologi.com/products/validator/index.html, 2003.

[17] UN/CEFACT, *Numerical Representation of Dates, Time, and Periods of Time (Recommendation No. 7, TRADE/WP.4/INF. 108, TD/B/FAL/INF. 108)*, UN/CEFACT, 1998.

[18] *UN/CEFACT Modeling Methodology (UMM)*, http://www.gefeg.com/tmwg/n090r10.htm, November 2001.

[19] Widhalm, Richard and Mueck, Thomas A., "Web Metadata Semantics - On the Road to Well Formed Topic Maps," *Proceedings of the Second International Conference on Web Information Systems Engineering*, Vol.2, Kyoto, December 2001.

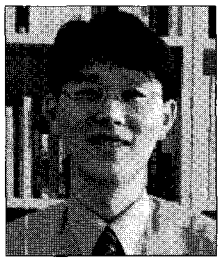
[20] xCBL.org, *XML Common Business Library*, http://www.xcbl.org/, 2002.

<부록 1> ebXML BPSS의 WFR(WellFormedness Rule)

구 분	WFR 번호	규칙의 내용	제약의 종류
MultiPartyCollaboration	1	All multiparty collaborations must be synthesized from binary collaborations	CAM
BusinessPartnerRole	2	A partner must not perform both roles in a given business activity	CAM
Performs	3	For every performs performing an AuthorizedRole there must be a Performs that performs the opposing AuthorizedRole, otherwise the MultiParty Collaboration is not complete	CIM
AuthorizedRole	4	An AuthorizedRole may not be both the requestor and the responder in a business transaction.	CAM
	5	An AuthorizedRole may not be both the initiator and the responder in a binary business transaction.	CAM
CollaborationActivity	6	A binary collaboration may not re-use itself.	CIN
DocumentEnvelop	7	A Document Envelope is associated with exactly one requesting and one responding activity.	CIE
	8	IsPositiveResponse is not a relevant parameter on a DocumentEnvelope sent by a requesting activity	CAM
Transition	9	A transition cannot enter and exit the same state	CIN
Success	10	Every Binary Collaboration should have at least one success	CIM
Failure	11	Every Binary Collaboration should have at least one failure	CIM
BusinessTransaction	12	If non-repudiation is required then the input or returned business document must be a tamper-proofed entity	CAM
	13	If authorization is required then the input business document and business signal must be an authenticated or a tamper proofed secure entity.	CAM
	14	The time to acknowledge receipt must be less than the time to acknowledge acceptance if both properties have values. timeToAcknowledgeReceipt < timeToAcknowledgeAcceptance	CAS
	15	If the time to acknowledge acceptance is null then the time to perform an activity must either be equal to or greater than the time to acknowledge receipt.	CAS
	16	The time to perform a transaction cannot be null if either the time to acknowledge receipt or the time to acknowledge acceptance is not null.	CAM
	17	If non-repudiation of receipt is required then the time to acknowledge receipt cannot be null	CAR
	18	The time to acknowledge receipt, time to acknowledge acceptance and time to perform cannot all be zero.	SAR
	19	If non-repudiation is required at the requesting business activity, then there must be a responding business document	CIM

구 분	WFR 번호	규칙의 내용	제약의 종류
RequestingBusiness Activity	20	There must be one input transition whose source state vertex is an initial pseudo state.	CIE
	21	There must be one output transition whose target state vertex is a final state specifying the state of the machine when the activity is successfully performed	CIE
	22	There must be one output transition whose target state vertex is a final state specifying the state of the machine when the activity is NOT successfully performed due to a process control exception	CIE
	23	There must be one output transition whose target state vertex is a final state specifying the state of the machine when the activity is NOT successfully performed due to a business process exception.	CIE
	24	There must be one output document flow from a requesting business activity that in turn is the input to a responding business activity.	CIE
	25	There must be zero or one output document flow from a responding business activity that in turn is the input to the requesting business activity	CIE
RespondingBusiness Activity	26	There must be one input transition from a document flow that in turn has one input transition from a requesting business activity	CIE
	27	There must be zero or one output transition to a document flow that in turn has an output transition to a requesting business activity.	SMA SME
BusinessCollaboration	28	A Business Partner Role cannot provide both the initiating and responding roles of the same business transaction activity	CAM

◆ 저자소개 ◆



김종우 (Kim, Jong Woo)
현재 한양대학교 경영학부 부교수로 재직 중이다. 서울대 수학과에서 이학사(1989), 한국과학기술원 경영과학과에서 공학석사(1991)를 취득하고 한국과학기술원 산업경영학과에서 공학박사(1995)를 취득하였다. 한국과학기술원 경영정보연구센터 연수연구원, University of Illinois at Urbana-Champaign 방문연구원, 충남대학교 통계학과 부교수로 근무한 경력이 있다. 주요 관심분야는 경영정보시스템, 의사결정지원시스템, 전자상거래, 데이터 마이닝 응용 등이다.

김형도 (Kim, Hyoung Do)
서울대학교 산업공학과 학사, 한국과학기술원 경영과학과 석사, 그리고 한국과학기술원 경영과학과에서 경영정보학으로 박사학위를 취득하였다. 현재 한양사이버대학교 부교수로 재직하고 있으며, 전자상거래 표준화 통합포럼(ECIF) 전자문서 기술위원회 부위원장, ebXML 전문위원회 칸텐츠 소위원회 위원장을 맡고 있다. ebXML, 웹 서비스, 시맨틱 웹 등 새로운 전자상거래 기반을 중심으로 연구개발, 교육 및 표준화 활동을 활발히 전개하고 있다. Decision Support Systems, ACM SIGMOD Record, IEICE Transactions on Information & Systems, Int'l Journal of Management Science 등에 다수의 논문을 게재 하였으며, 저서로는 "B2B 전자상거래 @ XML"와 "전자상거래원론"이 있다. 주요 관심분야는 XML기반의 전자상거래, B2B 워크플로우, 디지털 컨텐츠 워터마킹, 객체지향 모델링, 데이터 마이닝 등이다.

◈ 이 논문은 2003년 1월 14일 접수하여 2차 수정을 거쳐 2003년 8월 19일 게재확정되었습니다.