

---

# XBRL 서비스 품질 향상을 위한 의미제약 표현 방법 분석

## Analysis of Representation Methods for Semantic Constraints to Enhance the Quality of XBRL Services

---

김형도  
한양사이버대학교 경영학부

HyoungDo Kim(hdkim@hycu.ac.kr)

---

### 요약

XBRL은 재무 보고를 비롯한 다양한 비즈니스 보고에서 활용되고 있는 XML 기반 언어로서, 응용 분야의 개념들과 이들간의 관계를 정의하여 유연하게 활용할 수 있다. 이러한 비즈니스 보고 프로세스에서, 보고 내용의 일치성과 완전성을 문법적이고 의미적인 수준에서 검증하는 것은 XBRL 문서를 제출하는 측이나, 제출받아 이용하는 측 모두에게 매우 중요하다. 이와 같은 의미제약을 응용 프로그램 코드로 직접 표현하고 검증하는 것이 기본적인 방법이지만, 이것은 의미제약의 표현과 변경, 그리고 공유를 어렵게 한다. XML 문서의 의미제약을 일반적으로 표현할 수 있는 XSLT, Schematron 등의 언어는 XML을 이용한 의미제약의 명시적인 표현과 공유가 가능하나, XBRL 의미제약을 효율적이고 효과적으로 표현하기 어렵다. 이 논문에서는 현재 활발히 논의가 진행되고 있는 XBRL Formula의 의미제약 표현 방법에 대하여 검토하고, 사례 분석을 통해서 표현 능력과 문제점을 토의하고자 한다. 구체적인 분석 사례로서는 재무보고 분야의 재무제표를 이용한다.

■ 중심어 : | 비즈니스 보고 | 엑스비알엘 | 의미제약 | 서비스 품질 |

### Abstract

XBRL is an XML-based language, actively used for diverse business reporting applications including financial reporting. It can be flexibly applied to each application by defining concepts and their relationships existing within the application. In these business reporting processes, it is very important for senders and receivers to validate the consistency and completeness of reporting contents in the syntactic and semantic levels. The basic method is to directly represent and validate the semantic constraints using application program codes. However, the method makes it difficult to represent, change, share semantic constraints. While XML constraint languages for XML documents such as XSLT and Schematron support explicit representation and sharing of semantic constraints, they are limited in the efficiency and effectiveness of representing XBRL semantic constraints. This paper reviews XBRL formula, actively being discussed recently for standardization, and discusses the representation capability and limitations through a case analysis, which applies XBRL formula to business documents in the area of financial reporting.

■ keyword : | Business Reporting | XBRL | Semantic Constraint | Service Quality |

---

\* 본 논문은 2006년 한양사이버대학교 교내연구비 지원으로 연구되었음 (HYC-2006-3).

## I. 서론

XBRL[1-4]은 재무 보고를 비롯한 다양한 비즈니스 보고에서 광범위하게 활용되고 있는 XML 기반 언어로서, 응용분야별 개념들과 이들간의 관계를 정의하여 활용하도록 지원한다. 이러한 비즈니스 보고 처리 과정에서, 보고 내용의 일치성과 완전성을 문법적이고 의미적인 수준에서 검증하는 것은 XBRL 문서를 제출하는 측이나, 제출받아 이용하는 측 모두에게 매우 중요하다. 문법적인 수준에서는 XML Schema[5]와 같은 스키마 언어를 이용해서 검증이 가능하나, 의미적인 수준에서의 제약을 표현하고 적용하는 것은 현실적으로 매우 어렵다. 일반적으로 의미제약을 응용 프로그램 코드로 직접 표현하고 검증하는 것이 기본적인 방법이지만, 이것은 의미제약의 표현과 변경, 그리고 공유를 어렵게 한다. XML 문서의 의미제약을 일반적으로 표현할 수 있는 XSLT[6], Schematron[7] 등의 XML 언어를 이용하면 의미제약의 명시적인 표현과 공유가 가능하다. 그러나, 이러한 방법으로는 XBRL 의미제약을 효율적이고 효과적으로 표현하기 어렵다.

XBRL에서 제약을 표현하는 기본적인 방법으로는 계산과 정의의 관계를 이용한다. 관계는 링크로서 표현되는데, 계산(Calculation) 링크를 이용하여 개념들간의 계산 관계상의 제약을 규정할 수 있다. 이러한 계산 링크 만으로는 동일한 문맥상의 덧셈만이 가능할 뿐이며, 다차원 공간상의 복잡한 계산이 불가능하다. 또한 정의(Definition) 링크를 이용하여 개념들간의 의존성을 규정할 수 있으나, 단순한 제약에 한정된 것이다. XBRL Dimensions[8]을 이용하여 특정 사실과 연관된 세그먼트(Segment)나 시나리오(Scenario)의 내용에 대하여 단순하게 제약을 가할 수도 있다. 이상과 같이 기존의 XBRL만으로는 일반적인 비즈니스 규칙이나 제약을 표현하는데 한계가 있다.

이를 극복하기 위해서 등장한 것이 XBRL Formula[9]이다. 이 논문에서는 현재 활발히 표준화 논의가 진행되고 있는 XBRL Formula의 의미제약 표현 방법에 대하여 심도있게 검토하고, 사례 분석을 통해서 표현 능력과 문제점을 분석하고자 한다. 구체적인 분석

사례로서는 재무보고 분야의 재무제표가 이용된다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 XBRL을 비롯한 관련 연구에 대하여 소개하고, 3장에서는 XBRL Formula를 중심으로 의미제약 표현방법을 재무보고 분야의 재무제표를 이용하여 구체적으로 검토한다. 4장에서는 XBRL Formula의 표현 능력과 문제점에 대하여 토의한다. 마지막으로 5장에서는 결론 및 앞으로의 연구방향을 제시한다.

## II. 관련연구

### 1. XBRL

XML이 등장하면서, 이를 재무보고 분야에 적용하고자 시작된 것이 바로 XBRL(eXtensible Business Reporting Language)이다. 초기에는 XFRML(eXtensible Financial Reporting Markup Language)로 불렸으나, 곧바로 XBRL로 이름이 바뀌었다. 이것은 XBRL이 대차대조표, 손익계산서와 같은 재무보고 문서들을 표현하기 위한 것에 한정되지 않고, 광범위한 기업 비즈니스 보고에서 활용될 수 있음을 강조하고자 한 것이다. 오늘날 XBRL은 외부 보고를 위한 FR(Financial Reporting)과 내부 보고를 위한 GL(Global Ledger)이라는 두 가지의 큰 흐름을 중심으로 빠르게 발전하고 있다.

HTML과 같은 마크업 언어를 정의할 수 있는 메타언어인 XML은 다양한 응용분야에서 교환하고자 하는 의미를 쉽게 표현하고 활용할 수 있도록 지원한다. 이런 이유로 수많은 XML 언어들(이 활발히 사용되면서, 정보의 표현과 교환에서는 모두 XML을 사용하고 있다고 해도 과언이 아니다. 그런데 XBRL에는 일반적인 XML 언어들과 크게 다른 점이 두 가지가 있다. 그 중 한 가지는 스키마의 구성 체계에 대한 것이다. 일반적인 XML 언어에서는 표준으로 정의된 언어 스키마를 준수하여 사례를 정의하고 검증하는 체계로 운용된다. 이에 반하여, XBRL에서는 재무보고의 다양성을 효과적으로 소화할 수 있도록 표준으로 정의된 언어 스키마를 최소화하고, 이를 다양하게 확장할 수 있게 허용한다. 표준

으로 정의한 XBRL 스키마에서는 아이템(Item)과 튜플(Tuple)이라고 하는 두 가지 핵심 개념(Concept)을 정의하고 있다. 또 하나의 다른 점은 개념들간의 관계를 XLink로 표현하는 것인데, 이것은 계산 관계의 표현과 활용이나 하나의 콘텐츠를 다양한 형태로 가공하는 등과 같은 재무보고 분야의 필요성으로부터 출발한 것이다. 개념들간에는 정의(Definition), 프리젠테이션(Presentation), 계산(Calculation)이라는 3가지의 링크가 사용될 수 있으며, 개념을 이해하는데 필요한 라벨(Label) 자원이나 참조(Reference) 자원으로서의 링크도 가능하다. XBRL에서는 정의된 개념들을 택사노미(Taxonomy, 분류체계)라고 하며, 여기에 관련된 링크들을 합하여 DTS(Discoverable Taxonomy Set)이라 부르고 있다.

일반적인 비즈니스 보고 프로세스는 기업운영에서 내부보고를 거쳐 외부보고, 그리고 투자분석에까지 이르게 된다. 이 중에서 초기부터 XBRL이 활발하게 적용되고 있는 분야로는 외부보고에서 투자분석으로 이어지는 과정에서 이루어지는 규제신고나 재무보고가 대표적이다. 한국이나 미국 등과 같이 국가별로 GAAP(Generally Accepted Accounting Principle) 택사노미[10][11]의 표준화를 추진하고 있고, 국제적으로도 IFRS-GP 택사노미[12]나 GRI G3 택사노미[13] 등이 추진되고 있다. 이와 같이 다양한 분야에서 그 가치가 입증되면서 XBRL이 급속하게 보급되고 있지만, GAAP의 정비와 변화에 따른 신속한 택사노미 변경과 관리, 국제적 기준과 국가별 기준의 조정 등 해결해야 할 문제도 많이 있다.

내부적인 보고와 관련된 XBRL 표준화는 최근에서야 XBRL-GL(Global Ledger)라는 이름으로 활발히 진행되고 있다. 이것은 기업 내부에 존재하는 운영이나 회계 등의 데이터베이스에 존재하는 모든 상세 및 요약 데이터에 관한 것으로, 전형적인 ERP 시스템에 존재하는 파일들을 표현하기 위한 것이라고도 할 수 있다. 기본적으로 이것은 외부 보고와 연관해서 내부 데이터로 추적해 들어갈 수 있는 연결고리를 제공할 수 있다. XBRL-GL은 총계정원장(General Ledger)을 XML로 표현하고자 하는 것 이상을 의미하며, 각종 비재무 테

이터를 포괄하는 비즈니스 데이터의 저장, 교환, 통합, 추적 등에 유용하게 사용될 수 있다.

최근 XBRL을 비즈니스 인텔리전스의 새로운 수단으로 보는 관점이 설득력을 얻고 있다[5]. XBRL-GL 보고서들을 모아서 다차원의 데이터웨어하우스(Data Warehouse, DW)를 구축하고, 이를 바탕으로 내부 및 외부의 XBRL-FR 보고서를 생성하는 관점이 바로 이것이다. 물론 DW를 이용하여 OLAP(Online Analytical Processing)과 군집 분석, 분류, 연관관계 탐색 등의 지능적 분석이 가능하다. 이러한 관점에서 XBRL을 보완하기 위해 등장한 것이 XBRL Dimensions[8]이다. 원래 XBRL 사례문서에서는 문맥정보를 표현하기 위해서 context를 사용하는데, 여기에 포함된 segment와 scenario에는 자유롭게 정보를 기술할 수 있도록 허용하고 있다. XBRL Dimensions에서는 이 두 가지 요소의 내용을 규격화하여 일정한 방식으로 차원 정보를 XBRL 사례문서에 기술할 수 있도록 하고 있다. 이와 같이 XBRL-GL을 기반으로 기업 내부의 실시간 정보 교환과 지능적 활용이 가능해지면, 실시간기업(Real-Time Enterprise, RTE)이라는 이상으로 보다 전진하게 될 것이다.

## 2. 의미제약의 표현

오늘날 의미를 검증하는 가장 일반적인 방법은 응용의 일부분으로 검증 기능 프로그램을 구현하는 것이다. 이것은 선언적인 문법 검증이 불가능한 경우로, 여러 가지 단점을 동반한다. 프로그래머들은 검증 코드를 작성해야 하므로, 검증 대상 문서의 유형에 친숙해져야 한다. 검증 코드는 XML 데이터를 접근하는 코드들과 뒤섞이게 되고, 결국 검증에 집중할 수 없도록 만든다. 프로그램 언어에 종속적으로 검증 코드가 작성되어 이식성이 낮고, 재사용 및 관리도 어렵다. 이러한 모든 요인들이 의미 검증을 어렵게 하고, 중요한 비즈니스 지식을 비가시적으로 만들게 된다. 프로그램에 의한 검증의 유일한 장점은 처리 성능의 향상이라고 할 수 있는데, 여러 단점들이 이러한 장점을 압도한다.

XML기반 언어에서는 XML Schema[5]를 이용하여 XML 문서에 관한 제약을 정의할 수 있으나, 문서에 대

한 의미계약을 모두 표현하기에는 매우 부족하다. 예를 들면, 어떤 요소가 존재하면 또 다른 요소가 존재해야 한다는 것과 같은 의미계약의 표현은 XML Schema를 이용한다고 하더라도 표현이 불가능하다. 이러한 목적을 달성하기 위해서는 해당 문서를 읽어 들여 분석하는 전용 프로그램을 작성하거나, XSLT와 같은 XML 문서 변환 언어를 이용하거나, XML 문서에 대한 의미계약 표현 언어를 이용할 수 있다.

XSLT[6]는 입력 XML 문서를 출력 XML 문서로 변환하기 위한 언어로서, XML 문서의 검증을 위해서도 사용되고 있다. 어떤 엘리먼트에 적용되는 템플릿 규칙을 작성할 수 있으며, XSLT 함수를 이용하여 템플릿에 검증 코드를 삽입할 수 있다. XSLT의 출력은 입력 문서에 대한 진단 메시지를 포함하는 보고서 형태이다. XSLT는 프로그램에 의한 방법보다 장점이 있는데, 선언적인 언어여서 처리 코드를 작성해야 하는 부담을 제거할 수 있다는 것이 가장 중요하다. 그러나 이것은 제약언어가 아니라 변환 언어이기 때문에 몇 가지 단점을 가지고 있다. XSLT의 풍부함이 역으로 처리 최적화를 제약한다. 이에 반해서 전용 제약 언어는 표현 재활용이 용이하며, 제약 위반을 해결하는 방안의 수립 등과 같은 계약의 분석도 가능하다. XSLT는 이러한 분석이 불가능하거나 구현하기 어려우며, 문서간 또는 문서와 데이터베이스간 검증도 어렵다.

Schematron[7]은 XPath 기반의 대표적 제약언어로서, 불리언 논리를 이용하여 XML 문서의 계약을 표현할 수 있다. Schematron의 제약은 전용 검증 엔진에 의해서 평가되거나, XSLT 템플릿으로 변환되어 XSLT 처리기에 의해서 실행될 수 있다. Schematron이 일견 XSLT와 비슷하지만, 변환을 지원하기 위해 필요한 요소가 생략되어 XSLT보다 범위가 훨씬 좁은 언어이다. 따라서 언어를 사용하기 쉽고, 분석 도구를 작성하기 쉽다. XCMML[14], XCSL[15], xlinkit[16] 등과 같은 언어도 표현력이나 제공하는 기능이 Schematron과 거의 유사하나, 광범위하게 사용되지는 못하고 있다.

이상과 같은 여러 의미계약 표현 방법을 특성에 따라서 함께 비교해보면 [표 1]과 같이 정리할 수 있다. XBRL Formula도 XML 제약언어의 일종으로 볼 수 있

다. 이러한 표현 방법들은 XBRL 사례 문서에도 적용이 가능하나, 비즈니스 보고 분야의 계약을 표현하는 능력이나 효율성 측면에서 한계가 있기 때문에, 전용 제약 언어인 XBRL Formula의 표준화가 급속히 진행되고 있다.

표 1. 의미계약 표현 방법 비교

특성	방법	프로 그램	XML Schema	XSLT	XML 계약언어
선언적표현		불가능	가능	가능	가능
계약표현능력		높음	낮음	보통	높음
공유및재활용		낮음	높음	높음	높음
처리성능		우수	보통	보통	보통
계약분석능력		없음	낮음	낮음	높음
주요용도		구현	스키마 검증	문서 변환	계약표현

### III. XBRL 의미계약 사례분석

이번 장에서는 재무보고 분야의 재무제표를 이용하여 의미계약 표현방법을 구체적으로 검토한다. 무엇보다도, 최근에 표준화 작업이 활발히 진행되고 있는 XBRL Formula를 중심으로 사례를 분석한다.

XBRL에서 계약을 표현하는 기본적인 방법으로는 계산과 정의 관계를 이용한다. XBRL에서 관계는 링크로서 표현되는데, 계산(Calculation) 링크를 이용하여 개념들간의 계산 관계상의 계약을 규정할 수 있다. 대차대조표에서 자산(Assets)은 유동자산(Current Assets)과 비유동자산(Non-current Assets)의 합과 일치해야 한다. 이 계약은 [그림 1]과 같이 계산 링크를 이용하여 표현될 수 있다. 그러나, 이러한 계산 링크만으로는 동일한 문맥상의 덧셈만이 가능할 뿐이며, 다차원 공간상의 복잡한 계산이 불가능하다.

정의(Definition) 링크를 이용하여 개념들간의 의존성을 규정할 수 있으나, 계산 링크와 마찬가지로 단순한 계약에 한정된다. XBRL Dimensions[8]을 이용하여 특정 사실과 연관된 세그먼트(segment)나 시나리오(scenario)의 내용에 대하여 단순하게 계약을 가할 수도 있다. [그림 2]는 'cl6'이라는 컨텍스트가 지역차원(RegionDim)에서 'Korea'이고, 제품차원(ProductDim)에서 'RaspberryWin'이라는 세그먼트에 해당됨을 규정

하고 있다. [그림 3]과 같이 이들 차원과 구성요소는 아  
이템으로, 그리고 이들간의 관계는 정의 링크로 규정되  
며, 'Sales' 아이템의 세그먼트에 적용될 때 제약으로서  
작동하게 된다. 여기서 하이퍼큐브  
'hc\_Product\_x\_Region'은 차원 'dim\_Product'와  
'dim\_Region'으로 구성되며, 전자는 'm\_TotalProduct',  
'm\_RaspberryWine'으로, 후자는 'm\_TotalRegion',  
'm\_Korea'로 각각 세분화된다.

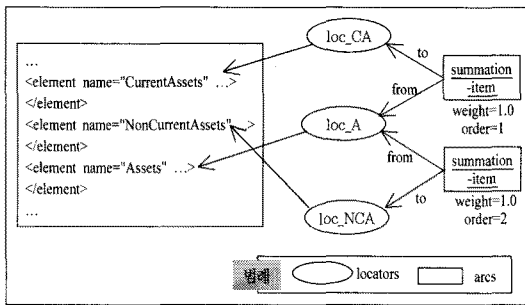


그림 1. 계산 링크를 이용한 계산 관계상의 제약 사례

```
<context id='c16'>
  <entity>
    <identifier scheme='http://nic.net'...
    <identifier>
    <segment>
      <xbrldi:explicitMember dimension='tax.dim_Region'>m_Korea
      <xbrldi:explicitMember>
      <xbrldi:explicitMember dimension='tax.dim_Product'>m_RaspberryWine
      <xbrldi:explicitMember>
    </segment>
  </entity>
  <period><forever></period>
</context>
```

그림 2. 세그먼트에 대한 차원 적용 사례

XBRL 사례 문서를 정의하는 XML 스키마에서는 허  
용된 값이나 구조에 대한 제약만을 표현할 수 있으며,  
요소간의 관련성에 대한 제약을 표현하는 것은 불가능  
하다.

이상과 같이 기존의 XBRL만으로는 일반적인 비즈니  
스 규칙이나 제약을 표현하는데 한계가 있다. 이러한  
한계를 극복하기 위한 노력중의 하나가 XBRL  
Formula이다. XBRL Formula에서는 출력 사례문서에  
존재할 아이템들을 반환하거나, 입력 사례문서상의 일  
치성(Consistency), 존재(Existence), 값(Value) 등에

대한 검증이 가능하다. XBRL Formula는 Formula 링  
크베이스에 표현되는데, 여기에서는 formula,  
assertion, variable, filter, precondition 등의 새로운 종  
류의 자원을 이용하게 되며, [그림 4]와 같이 이들간을  
링크로 연결하여 공식과 제약을 표현하게 된다. 제약을  
나타내는 assertion은 formula와 연결되며, formula는  
공식에서 사용하는 변수들과 연결되고, 각각의 변수들  
은 필터와 연결되어 XBRL 사례문서의 데이터와 연결  
되는 구조를 갖는다. formula에는 precondition을 설정  
하여 계산의 사전조건을 설정할 수 있다.

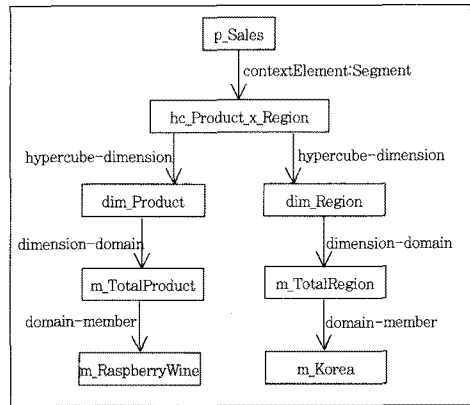


그림 3. 'Sales' 아이템의 세그먼트에 대한 차원 제약 사례

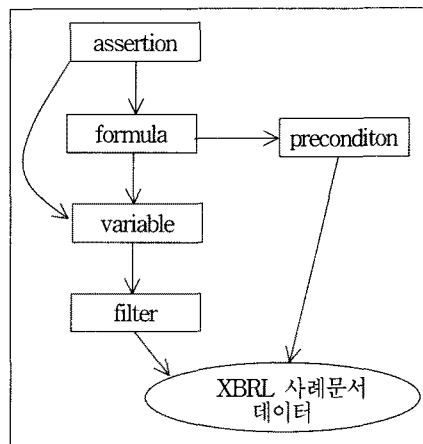


그림 4. XBRL Formula의 자원간 연결 구조

대차대조표 상의 부채(liabilities)와 자본(equity)을  
더한 자산/assets)의 값을 출력하고자 하는 경우를 가

지고 분석하고자 한다. 부채와 자본은 [그림 5]와 같이 정의된 아이탬들로서 monetaryItemType을 따른다.

```

01. <xsd:schema ...>
02. <xsd:annotation>
03. <xsd:appinfo>
04. <link:linkbaseRef
05.   xlink:href="14010-computeAssets-formula.xml"
06.   xlink:type="simple" xlink:arcrole="http://
07.     www.w3.org/1999/xlink/properties/linkbase" />
08. </xsd:appinfo>
09. </xsd:annotation>
10. <xsd:import
11.   namespace="http://www.xbrl.org/2003/instance"
12.   schemaLocation="http://www.xbrl.org/2003/xbrl-
13.     instance-2003-12-31.xsd"/>
14. <xsd:element name="assets" id="assets"
15.   xbrli:periodType="instant"
16.   type="xbrli:monetaryItemType"
17.   substitutionGroup="xbrli:item"
18.   abstract="false" nillable="true" />
19. <xsd:element name="liabilities" id="liabilities"
20.   xbrli:periodType="instant"
21.   type="xbrli:monetaryItemType"
22.   substitutionGroup="xbrli:item"
23.   abstract="false" nillable="true" />
24. <xsd:element name="equity" id="equity"
25.   xbrli:periodType="instant"
26.   type="xbrli:monetaryItemType"
27.   substitutionGroup="xbrli:item"
28.   abstract="false" nillable="true" />
29. </xsd:schema>
    
```

그림 5. XBRL 아이탬들

XBRL Formula의 특정 formula에서 이들을 연결하여 출력을 계산하는 것은 [그림 6]과 같이 표현될 수 있다. 이 그림에서 4행에서 14행까지는 formula를 정의하고 있다. 구체적으로 5행에서는 value 속성을 이용하여 계산공식을 규정하고 있다. 즉, 'v:liabilities', 'v:equity'를 더한 값을 구하는 공식인 것이다. 15행에서 20행까지는 2개의 factVariable을 선언하며, 21행에서 32행까지는 이들 각각에 대한 conceptName 필터를 정의한다. 이 필터는 XBRL 개념의 이름을 가지고 사례문서의 사실들을 처리한다. 33행에서 40행까지는 formula로부터 종속되는 factVariable로의 variableArc를 정의하고 있으며, 41행에서 52행까지는 factVariable로부터 필터를 연결하는 variableFilterArc를 정의하고 있다.

```

01. <link:linkbase ...>
02. <generic:link xlink:type="extended"
03.   xlink:role="http://www.xbrl.org/2003/role/link">
04.   <formula:formula xlink:type="resource"
05.     xlink:label="formula" value="$v:liabilities + $v:equity"
06.     source="v:liabilities" aspectModel="dimensional"
07.     implicitFiltering="true">
08.     <formula:decimals>0</formula:decimals>
09.     <formula:aspects>
10.       <formula:concept>
11.         <formula:qname>c:assets</formula:qname>
12.       </formula:concept>
13.     </formula:aspects>
14.   </formula:formula>
15.   <variable:factVariable xlink:type="resource"
16.     xlink:label="variable_liabilities"
17.     bindAsSequence="false" />
18.   <variable:factVariable xlink:type="resource"
19.     xlink:label="variable_equity"
20.     bindAsSequence="false" />
21.   <cf:conceptName xlink:type="resource"
22.     xlink:label="filter_liabilities">
23.     <cf:concept>
24.       <cf:qname>c:liabilities</cf:qname>
25.     </cf:concept>
26.   </cf:conceptName>
27.   <cf:conceptName xlink:type="resource"
28.     xlink:label="filter_equity">
29.     <cf:concept>
30.       <cf:qname>c:equity</cf:qname>
31.     </cf:concept>
32.   </cf:conceptName>
33.   <variable:variableArc xlink:type="arc"
34.     xlink:arcrole="http://xbrl.org/arcrole/2003/variable-set"
35.     name="v:liabilities" xlink:from="formula"
36.     xlink:to="variable_liabilities" order="1.0" />
37.   <variable:variableArc xlink:type="arc"
38.     xlink:arcrole="http://xbrl.org/arcrole/2003/variable-set"
39.     name="v:equity" xlink:from="formula"
40.     xlink:to="variable_equity" order="2.0" />
41.   <variable:variableFilterArc xlink:type="arc"
42.     xlink:arcrole="
43.       http://xbrl.org/arcrole/2003/variable-filter"
44.     complement="false" cover="true"
45.     xlink:from="variable_liabilities"
46.     xlink:to="filter_liabilities" order="2.0" />
47.   <variable:variableFilterArc xlink:type="arc"
48.     xlink:arcrole="
49.       http://xbrl.org/arcrole/2003/variable-filter"
50.     complement="false" cover="true"
51.     xlink:from="variable_equity" xlink:to="filter_equity"
52.     order="3.0" />
53. </generic:link>
54. </link:linkbase>
    
```

그림 6. XBRL Formula 제약표현 사례

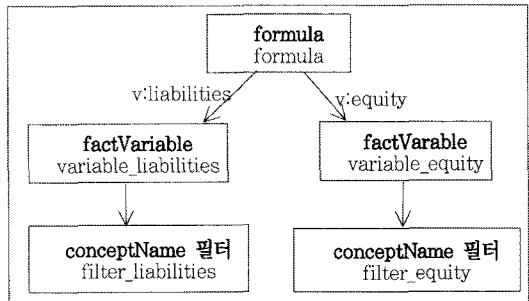


그림 7. Formula와 리소스간 관계

이상의 내용을 그림으로 정리하면 [그림 7]과 같이 표현할 수 있다. [그림 8]과 같은 XBRL 사례문서에 대하여 [그림 6]의 XBRL Formula를 적용하면, [그림 9]와 같은 결과를 얻게 된다. 즉 부채와 자본을 더한 자산(assets) 아이টে임을 얻게 되는 것이다.

formula에 대한 사전제약은 precondition을 통해서 표현할 수 있다. [그림 10]에서 정의되는 formula는 v:fact가 0보다 작은 조건을 나타내는 5행에서 6행까지의 precondition과 연결되어 있으며, 이 사전조건이 만족될 경우에만 formula를 실행하여 그 값을 반환한다.

```

01. <xbrl ...>
02. <link:schemaRef xlink:type="simple"
03.   xlink:arcrole=
04.   "http://www.w3.org/1999/xlink/properties/linkbase"
05.   xlink:href="14010-computeAssets-concepts.xsd" />
06. <context id="context">
07. <entity>
08.   <identifier scheme=
09.     "http://xbrl.org/entity/identification/scheme">01
10.   </identifier>
11. </entity>
12. <period>
13.   <instant>2007-12-31</instant>
14. </period>
15. </context>
16. <unit id="unit">
17.   <measure>iso4217:USD</measure>
18. </unit>
19. <bs:liabilities decimals="0" contextRef="context"
20.   unitRef="unit">600</bs:liabilities>
21. <bs:equity decimals="0" contextRef="context"
22.   unitRef="unit">400</bs:equity>
23. </xbrl>
    
```

그림 8. XBRL 사례문서

```

01. <xbrl ...>
02. <link:schemaRef xlink:type="simple"
03.   xlink:arcrole=
04.   "http://www.w3.org/1999/xlink/properties/linkbase"
05.   xlink:href="14010-computeAssets-concepts.xsd" />
06. <context id="context">
07. <entity>
08.   <identifier scheme=
09.     "http://xbrl.org/entity/identification/scheme">01
10.   </identifier>
11. </entity>
12. <period>
13.   <instant>2007-12-31</instant>
14. </period>
15. </context>
16. <unit id="unit">
17.   <measure>iso4217:USD</measure>
18. </unit>
19. <bs:assets precision="0" contextRef="context"
20.   unitRef="unit">1000
21. </bs:assets>
22. </xbrl>
    
```

그림 9. XBRL Formula 처리 결과

값에 관한 제약은 [그림 11]과 같이 valueAssertion을 이용하여 표현하는데, 4행에서 test 속성의 값으로

“\$v:NetIncomes le \$v:GrossIncomes”라고 설정하여 ‘v:NetIncomes’가 ‘v:GrossIncomes’보다 적어야 함을 규정하고 있다.

```

01. <formula:formula xlink:type="resource"
02.   xlink:label="formula" aspectModel="dimensional"
03.   implicitFiltering="true" source="v:fact"
04.   value="$v:fact" />
05. <variable:precondition xlink:type="resource"
06.   xlink:label="precondition" test="$v:fact lt 0" />
07. <generic:arc xlink:type="arc" xlink:arcrole=
08.   "http://xbrl.org/arcrole/2008/variable-set-precondition"
09.   xlink:from="formula" xlink:to="precondition"
10.   order="1.0" />
11. <variable:factVariable xlink:type="resource"
12.   xlink:label="variable_fact" bindAsSequence="false" />
13. <cf:conceptDataType xlink:type="resource"
14.   xlink:label="filter_dataType" strict="false">
15. <cf:type>
16.   <cf:qname>xbrli:monetaryItemType</cf:qname>
17. </cf:type>
18. </cf:conceptDataType>
19. <variable:variableArc xlink:type="arc" xlink:arcrole=
20.   "http://xbrl.org/arcrole/2008/variable-set"
21.   xlink:from="formula" xlink:to="variable_fact"
22.   order="1.0" name="v:fact" />
23. <variable:variableFilterArc xlink:type="arc" xlink:arcrole=
24.   "http://xbrl.org/arcrole/2008/variable-filter"
25.   complement="false" cover="true"
26.   xlink:from="variable_fact" xlink:to="filter_dataType"
27.   order="1.0" />
    
```

그림 10. Formula 사전조건 설정

```

01. <va:valueAssertion xlink:type="resource"
02.   xlink:label="assertion" id="assertion"
03.   aspectModel="dimensional" implicitFiltering="true"
04.   test="$v:NetIncomes le $v:GrossIncomes" />
05. <variable:factVariable xlink:type="resource"
06.   xlink:label="GrossIncomes" bindAsSequence="false" />
07. <variable:factVariable xlink:type="resource"
08.   xlink:label="NetIncomes" bindAsSequence="false" />
09. <variable:variableArc xlink:type="arc" xlink:arcrole=
10.   "http://xbrl.org/arcrole/2008/variable-set"
11.   xlink:from="assertion" xlink:to="GrossIncomes"
12.   order="1.0" name="v:GrossIncomes" />
13. <variable:variableArc xlink:type="arc" xlink:arcrole=
14.   "http://xbrl.org/arcrole/2008/variable-set"
15.   xlink:from="assertion" xlink:to="NetIncomes"
16.   order="2.0" name="v:NetIncomes" />
17. <cf:conceptName xlink:type="resource"
18.   xlink:label="GrossIncomesFilter">
19. <cf:concept>
20.   <cf:qname>concept:GrossIncomes</cf:qname>
21. </cf:concept>
22. </cf:conceptName>
23. <cf:conceptName xlink:type="resource"
24.   xlink:label="NetIncomesFilter">
25. <cf:concept>
26.   <cf:qname>concept:NetIncomes</cf:qname>
27. </cf:concept>
28. </cf:conceptName>
29. <variable:variableFilterArc xlink:type="arc"
30.   xlink:arcrole="http://xbrl.org/arcrole/2008/variable-filter"
31.   xlink:from="GrossIncomes"
32.   xlink:to="GrossIncomesFilter"
33.   complement="false" cover="true" order="1.0" />
34. <variable:variableFilterArc xlink:type="arc"
35.   xlink:arcrole="http://xbrl.org/arcrole/2008/variable-filter"
36.   xlink:from="NetIncomes" xlink:to="NetIncomesFilter"
37.   complement="false" cover="true" order="1.0" />
    
```

그림 11. 값에 관한 제약

존재에 관한 제약은 [그림 12]와 같이 existenceAssertion을 이용하여 표현되는데, 10행에서 15행까지 concept 필터를 사용하여 'NetIncomes' 아이탬이 존재해야 함을 규정하고 있다.

```

01. <ea:existenceAssertion xlink:type="resource"
02. xlink-label="assertion" id="assertion"
03. aspectModel="dimensional" implicitFiltering="true" />
04. <variable:factVariable xlink:type="resource"
05. xlink-label="NetIncomes" bindAsSequence="false" />
06. <variable:variableArc xlink:type="arc" xlink:arcrole=
07. "http://xbrl.org/arcrole/2008/variable-set"
08. xlink:from="assertion" xlink:to="NetIncomes"
09. order="2.0" name="v:NetIncomes" />
10. <cf:conceptName xlink:type="resource"
11. xlink-label="NetIncomesFilter">
12. <cf:concept>
13. <cf:qname>concept:NetIncomes</cf:qname>
14. </cf:concept>
15. </cf:conceptName>
16. <variable:variableFilterArc xlink:type="arc"
17. xlink:arcrole="http://xbrl.org/arcrole/2008/variable-filter"
18. xlink:from="NetIncomes" xlink:to="NetIncomesFilter"
19. complement="false" cover="true" order="1.0" />

```

그림 12. 존재에 관한 제약

일치성에 관한 제약은 [그림 13]과 같이 consistencyAssertion을 이용하여 표현되는데, 1행에서 10행까지는 'assertion'이라는 일치성 제약에서 허용되는 오차의 한계는 0.05%임을 proportionalAcceptanceRatius 속성으로 지정하고 있으며, 11행에서 21행까지는 'v:CapitalInvested'와 'v:RiskFactor'를 더한 값이 'RiskyCapital'과 같아야 함을 formula로 규정하고 있다.

#### IV. 토의

기존의 XBRL 문서에서 제약을 표현할 수 있는 방법으로는 계산(Calculation) 링크를 이용하여 개념들간의 계산 관계상의 제약을 정의하거나, 정의(Definition) 링크를 이용하여 개념간의 의존 관계를 규정하거나, XML 스키마를 이용하여 개념에 허용된 값이나 튜플의 구조를 제한하거나, XBRL Dimensions을 이용하여 특정 사실과 연관된 세그먼트(segment)나 시나리오(scenario)의 내용에 대하여 제약을 가할 수 있다. 이러한 제약들은 보고서의 품질을 부분적으로 향상시킬 수 있을 뿐이다. 규제 상황 속에서 필요한 일반적인 비즈

니스 규칙의 표현과 활용은 불가능하다.

```

01. <ca:consistencyAssertion xlink:type="resource"
02. xlink-label="assertion" id="assertion" strict="false"
03. proportionalAcceptanceRadius="v:ToleranceRatio" />
04. <variable:parameter xlink:type="resource"
05. xlink-label="toleranceRatio" name="v:ToleranceRatio"
06. select="0.05" />
07. <variable:variableArc xlink:type="arc"
08. xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set"
09. xlink:from="assertion" xlink:to="toleranceRatio"
10. order="1.0" name="v:ToleranceRatio" />
11. <formula:formula xlink:type="resource"
12. xlink-label="formula" aspectModel="dimensional"
13. implicitFiltering="true"
14. value="$v:CapitalInvested * $v:RiskFactor"
15. source="formula:uncovered">
16. <formula:aspects>
17. <formula:concept>
18. <formula:qname>RiskyCapital</formula:qname>
19. </formula:concept>
20. </formula:aspects>
21. </formula:formula>
22. <generic:arc xlink:type="arc"
23. xlink:arcrole="http://xbrl.org/arcrole/2008/
24. consistency-assertion-formula"
25. xlink:from="assertion" xlink:to="formula"
26. order="1.0" />
27. <variable:factVariable xlink:type="resource"
28. xlink-label="capitalInvested"
29. bindAsSequence="false" />
30. <variable:factVariable xlink:type="resource"
31. xlink-label="riskFactor" bindAsSequence="false" />
32. <variable:variableArc xlink:type="arc"
33. xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set"
34. xlink:from="formula" xlink:to="capitalInvested"
35. order="1.0" name="v:CapitalInvested" />
36. <variable:variableArc xlink:type="arc"
37. xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set"
38. xlink:from="formula" xlink:to="riskFactor" order="2.0"
39. name="v:RiskFactor" />
40. <cf:conceptName xlink:type="resource"
41. xlink-label="capitalInvestedFilter">
42. <cf:concept>
43. <cf:qname>concept:CapitalInvested</cf:qname>
44. </cf:concept>
45. </cf:conceptName>
46. <cf:conceptName xlink:type="resource"
47. xlink-label="riskFactorFilter">
48. <cf:concept>
49. <cf:qname>concept:RiskFactor</cf:qname>
50. </cf:concept>
51. </cf:conceptName>
52. <uf:singleMeasure xlink:type="resource"
53. xlink-label="pureUnitFilter">
54. <uf:measure>
55. <uf:qname>xbrl:pure</uf:qname>
56. </uf:measure>
57. </uf:singleMeasure>
58. <variable:variableFilterArc xlink:type="arc"
59. xlink:arcrole="http://xbrl.org/
60. arcrole/2008/variable-filter"
61. xlink:from="capitalInvested"
62. xlink:to="capitalInvestedFilter" complement="false"
63. cover="true" order="1.0" />
64. <variable:variableFilterArc xlink:type="arc"
65. xlink:arcrole="http://xbrl.org/
66. arcrole/2008/variable-filter"
67. xlink:from="riskFactor" xlink:to="riskFactorFilter"
68. complement="false" cover="true" order="1.0" />
69. <variable:variableFilterArc xlink:type="arc"
70. xlink:arcrole="http://xbrl.org/
71. arcrole/2008/variable-filter"
72. xlink:from="riskFactor" xlink:to="pureUnitFilter"
73. complement="false" cover="true" order="2.0" />

```

그림 13. 일치성에 관한 제약 사례



일반적인 XML 문서의 제약조건을 Jacinto 등의 연구 [17]에서는 도메인 범위 점검, 두 개의 구성요소 또는 속성간의 의존성, 정상적인 표현에 대한 패턴 매칭, 복합 제약 등으로 구분하고 있다. XBRL Formula에서는 formula로 공식을 표현할 수 있고, assertion을 이용해서 값, 존재, 일치성에 대한 제약을 표현할 수 있어서, 이러한 제약조건을 모두 표현할 수 있다. 3장의 사례 분석에서 볼 수 있듯이, XBRL Formula를 이용하여 XBRL에서 필요한 의미제약을 효율적이고 효과적으로 표현할 수 있다.

하지만, 제약을 표현하는데 있어서 XBRL Formula가 전혀 문제가 없는 것은 아니다. 이 논문에서는 크게 세 가지 문제점을 지적하고자 한다.

첫 번째는 다수의 XBRL 사례문서 간에 존재할 수 있는 제약을 표현하지 못한다는 점이다. 예를 들면, 대차대조표와 손익계산서를 바탕으로 작성된 재무비율이 포함된 문서의 타당성을 검증하기 위해서는 이 문서와 함께 대차대조표 및 손익계산서가 필요하다. 이렇게 3개의 문서를 참조하여 제약을 표현하고 처리할 수 있어야 한다. 이와 같이 여러 회사의 문서가 결합되어 사용되는 상황이 점점 더 많아지고 있다는 점에서 이러한 다중문서에 적용되는 제약의 필요성은 점점 더 중요해지고 있다.

이에 대한 단순한 해결책으로는 통합된 택사노미와 통합된 사례문서를 만들어서 기존의 방법대로 XBRL을 적용하는 것이다. 물론 통합된 택사노미에는 새로운 제약을 추가하여 표현한다. 하지만 이 해결책은 원래의 사례문서들을 변형하게 되고, 다수의 통합 택사노미를 양산한다는 점에서 바람직하지 않다.

다수의 사례문서와 택사노미를 그대로 사용하면서 해결하기 위해서는 xlinkit[16]의 접근법이 유효하다고 판단된다. 이 제약언어에서는 분산된 문서들의 일치성을 표현하기 위하여 집합을 기반으로 둔 규칙언어를 제공하고 있다. [그림 14]는 광고 문서의 모든 'Adverts' 엘리먼트에 대하여 카탈로그 문서에 하나의 'Products' 엘리먼트가 있어서 그 이름이 일치하여야 함을 규정하고 있다. 이와 같이 xlinkit는 기본적으로 XLink와 XPath를 사용하고 있어서, 직접 XBRL에 적용하는 것도 어느

정도 도움이 될 것이며 (XBRL Dimensions 등과 같이 XBRL만의 특수한 요구사항으로 인하여 한계가 있음), XBRL Formula 표준화에 반영하는 것도 용이할 것으로 판단된다.

```

01. <consistencyrule id="r1">
02.   <forall var="a" in="//Adverts">
03.     <exists var="p" in="//Catalog/Products">
04.       <equal op1="$a/ProductName/test()"
05.         op2="$p/Name/test()" />
06.     </exists>
07.   </forall>
08. </consistencyrule>
    
```

그림 14. xlinkit 규칙 표현 사례 [16]

두 번째 문제점으로는 다수의 규칙들이 연결고리를 이루면서 추론하는 복잡한 과정을 표현하는 것이 불가능하다는 점이다. 단순히 사례문서에 존재하는 엘리먼트들 간의 관계만을 제약으로 표현하고 검증한다면, 진정한 의미제약을 적용하고 있다고 할 수 없을 것이다. 예를 들어 어떤 문서로부터 파생된 모든 문서에 적용되는 특정한 제약이 있다고 하면, 이 문서로부터 일차적으로 파생된 문서들뿐만 아니라 반복적으로 파생된 모든 문서들을 파악할 수 있어야 이 제약을 제대로 적용한 것이 된다. 이를 해결하기 위해서는 formula, variable, precondition 등의 XBRL Formula에서 새로 정의한 자원들을 연결하여 이러한 추론 규칙을 표현할 수 있도록 지원하는 것이 필요하다. 이러한 방향에서는 시맨틱웹의 선도적인 결과들을 표준화에 반영하도록 하는 노력이 필요하다.

세 번째 문제점으로는 링크 자원들에 대한 메타 지식을 지원하기 어렵다는 점이다. 예를 들어 정의 링크를 이용하여 공식을 표현하는 경우와 같이 다양한 링크 자원들을 체계적으로 재활용하고, 이들에 대한 메타 지식을 부여할 수 있다면, XBRL 도구가 특정한 관계에만 한정되는 상황을 많이 개선할 수 있을 것이다. 이러한 지식의 제공은 두 번째 문제점과 같은 방향에서 접근하는 것이 필요하다.

이상과 같은 한계점이 있다고 해도, XBRL Formula를 사용하여 제약을 표현하고 활용하는 것은 비즈니스 보고 문서의 검증을 위해서는 매우 유용하다. 제출자와

검사자/활용자가 동의할 수 있는 제약사항을 XBRL Formula로 정의하여 공유한다면, 정확한 지식의 공유와 더불어서 중복된 프로그램 구현의 문제점을 제거할 수 있다. 표현된 제약을 이용하는 전용도구의 개발이 미비한 현재 상황에서는 Schematron[7], XCMML[14], XCSL[15], xlinkit[16]과 같은 XML문서에 대한 의미 제약 표현 언어를 이용하거나, XSLT와 같은 XML 변환 언어를 이용하는 것도 좋은 대안이다[18]. 규칙으로 변환하여 규칙엔진을 이용하는 것도 고려할 수 있다[19].

3장의 사례분석에서 살펴보았듯이, XML 형태로 XBRL Formula를 표현하고 이해하는 것이 많은 사용자들에게 쉽지 않기 때문에, 개념적인 수준에서 의미 제약 모델링을 지원하는 것도 매우 중요하다고 판단된다. [그림 7]과 유사하게 개념적인 수준에서 의미 제약을 표현하여 처리할 수 있다면 여러 가지 장점을 갖게 되는데, 무엇보다도 의미제약에 대한 정의와 이해가 쉽도록 지원함으로써 개발과 재사용을 촉진할 수 있다.

XBRL을 큰 틀에서 보면, 데이터를 모델링하는 쪽은 DW에 해당되며, 개념을 중심으로 메타데이터를 모델링하는 쪽은 시맨틱웹(Semantic Web)으로 볼 수 있다[20]. 이런 관점에서 XBRL을 시맨틱웹과 비교할 수 있는데, 시맨틱웹은 분야에 무관하게 일반적인 의미의 표현과 활용에 중점을 두고 있는 반면에, XBRL은 비즈니스보고에 특화되어 있다. 시맨틱웹이 특정 주제에서는 매우 앞선 부분이 있으나, 아직까지 표준(안)의 변경이 잦아서 안정된 사용에는 좀 더 시간이 필요하다[21]. 반면에 XBRL은 안정된 기존 기술만을 사용하고 있고, 재무보고와 같은 실무적인 분야에서 활용되고 있다는 것이 장점이다. 따라서 시맨틱웹이 가지고 있는 장점을 XBRL에 받아들이는 노력도 필요하다.

## V. 결론

이 논문에서는 XBRL Formula 기반의 제약 표현 능력을 검토하고, 해결해야 할 문제점에 대하여 분석하였다. XBRL Formula는 공식과 함께 값, 존재, 일관성에

관한 제약을 표현할 수 있어서 XBRL 사례문서를 관련된 주체들이 검증하는데 있어서 매우 유용하다. 그러나 다수의 사례문서들간의 제약 표현, 공식간의 연결을 통한 복합적인 공식 표현, 링크를 이용하는 제약의 표현 등에서 문제점이 있으며, 이러한 관점에서 보완이 필요하다.

이러한 문제점을 해결하기 위하여, 앞으로 추진하고자 하는 작업으로는, 문서간의 제약표현 방안을 제시하고, 실험적인 전용도구의 개발과 실제 프로젝트를 통해서 이를 검증하며, 표준화 작업에 이를 반영하는 것이다.

## 참고 문헌

- [1] 김형도, 최선, 김기철, “중소기업 생산성 향상을 위한 XBRL 지속가능성 보고 방안 연구”, 생산성 논집, 제19권, 제4호, pp.147-169, 2005.
- [2] 김형도, 박찬권, “핵심 컴포넌트 방법론 기반의 XBRL 문서 표준화 및 재활용 방안 연구”, 한국전자거래학회지, 제11권, 제4호, pp.67-86, 2006.
- [3] 김형도, 박찬권, 염지환, 이성훈, “XBRL 기반의 의사결정 모형 표현과 공유”, JITAM, 제14권, 제2호, pp.117-127, 2007.
- [4] <http://www.xbrl.org/Specification/XBRL-RECOMMENDATION-2003-12-31+Corrected-Errata-2006-12-18.htm>
- [5] D. Lee and W. W. Chu, “Comparative Analysis of Six XML Schema Languages,” ACM SIGMOD Record, Vol.29, No.3, pp.76-87, 2000.
- [6] <http://www.w3.org/TR/xslt2/>
- [7] <http://www.ascc.net/xml/resource/schematron/schematron.html>, 2002.
- [8] <http://www.xbrl.org/Specification/XDT-REC-2006-09-18.htm>
- [9] <http://www.xbrl.org/Specification/formula/CR-2008-03-28/formula-CR-2008-03-28.html>
- [10] XBRL Korea Domain Working Group, “Korean

GAAP Taxonomy," Public Working Draft, 2004.

[11] <http://xbrl.us/pages/us-gaap.aspx>

[12] [http://xbrl.iasb.org/int/fr/ifrs/gp/2005-05-15/summary\\_page.htm](http://xbrl.iasb.org/int/fr/ifrs/gp/2005-05-15/summary_page.htm)

[13] <http://www.globalreporting.org/ReportingFramework/G3Guidelines/XBRL/>

[14] J. Hu and L. Tao, "An Extensible Constraint Markup Language: Specification, Modeling, and Processing," Proceedings of the XML 2004 Conference, Washington D.C., 2004.

[15] M. H. Jacinto, G. R. Librelotto, J. C. Ramalho, and P. R. Henriques, "Constraint Specification Languages: Comparing XCSL, Schematron, and XML-Schemas," Proceedings of the XML Europe 2002, Barcelona, 2002.

[16] C. Nentwich, L. Capra, W. Emmerich, and A. Finkelstein, "xlinkit: A Consistency Checking and Smart Link Generation Service," ACM Transactions on Internet Technology, Vol.2, No.2, pp.151-185, 2002.

[17] M. H. Jacinto, G. R. Librelotto, J. C. Ramalho, and P. R. Henriques, "Constraint Specification Languages: Comparing XCSL, Schematron, and XML-Schemas," Proceedings of the XML Europe 2002, Barcelona, 2002.

[18] 김형도, 김종우, "의미제약 기반의 ebXML BPSS 사례검증", 한국전자거래학회지, 제10권, 제4호, pp.1-18, 2005.

[19] J. W. Kim and H. D. Kim, "Semantic Constraint Specification and Verification of ebXML Business Process Specifications," Expert Systems with Applications, Vol.27, pp.571-584, 2004.

[20] P. Chameni, "XBRL and Business Intelligence," New Dimensions of Business Reporting and XBRL, pp.178-189, 2007.

[21] R. Lala, I. Cantador, and P. Castells, "XBRL

Taxonomies and OWL Ontologies for Investment Funds," Proceedings of the ER Workshops 2006, LNCS 4231, pp.271-280, 2006.

저자 소개

김형도(HyoungDo Kim)

정회원



- 1985년 2월 : 서울대학교 산업공학과(학사)
  - 1987년 2월 : 한국과학기술원 경영학과(석사)
  - 1992년 8월 : 한국과학기술원 경영학과(박사)
  - 1993년 ~ 1999년 : ㈜데이콤 EC인터넷 기술 팀장
  - 2000년 ~ 2002년 : 아주대학교 정보통신전문대학원 교수
  - 2001년 ~ 현재 : 전자상거래표준화통합포럼(ECIF) 전자문서기술위원회 부위원장
  - 2003년 ~ 현재 : 한양사이버대학교 경영학부 교수
  - 2004년 ~ 2006년 : ebXML 전문위원회 위원장
  - 2007년 ~ 현재 : 한양사이버대학교 경영학부장
- <관심분야> : 전자상거래, XML, 비즈니스 프로세스, 디지털 워터마킹, 데이터 마이닝, e-러닝