

핵심 컴포넌트 방법론 기반의 XBRL 문서 표준화 및 재활용 방안 연구

Research on the Standardization and Reuse of XBRL Documents Based on the Core Component Methodology

김형도 (HyoungDo Kim)*, 박찬권 (Chankwon Park)**

초 록

기업의 투명성과 사회적 책임이 강조되면서, 비즈니스 보고(Business Reporting)에 특화된 XML 언어인 XBRL(eXtensible Business Reporting Language)의 활용도가 크게 증가하고 있다. XBRL은 기업 보고 문서의 구성요소들(개념들과 이들의 구조체들)을 XSD(XML Schema Definition)로 정의한다는 점에서 기업간 전자문서 표준화와 기본적으로 동일하나, 그러나, 개념들의 유형이 특정하게 한정되어 있고, 이들 개념들을 연결하고 부가 정보를 제공하기 위한 다양한 관계 유형을 정의한다는 점에서 근본적인 차이가 있다. XBRL 응용이 활성화되면서 국가적인 수준에서의 표준화와 함께 체계적으로 구성요소들을 재활용해야 할 필요성이 증대되고 있으나, 이러한 근본적 차이로 인해서 표준화와 재활용에 어려움을 겪고 있다. 이 논문에서는 UN/CEFACT의 핵심 컴포넌트 방법론을 적용하여 XBRL 문서의 표준화 및 재활용이 가능하도록 하기 위하여, 문제점을 분석하고 대안을 제시한다.

ABSTRACT

In parallel with emphasizing more transparency and social responsibility of businesses, usage rate of XBRL, a XML-based language specific to business reporting, is rapidly increasing. XBRL takes the same approach with the standardization of B2B electronic documents in that it defines components (concepts and their structures) of business reporting documents using XSD(XML Schema Definition). However, it is fundamentally different from the standardization of B2B electronic documents because it restricts concepts to specific types and it defines diverse relationship types for connecting the concepts to themselves and other additional information. As XBRL applications are getting activated, one hot issue is the necessity for standardization of XBRL documents in the national level and reusing the components systematically. This paper proposes an alternative based on the problem analysis in order to make it possible to standardize and reuse XBRL documents systematically by applying the UN/CEFACT Core Component methodology.

키워드 : 확장성비즈니스보고언어, 핵심컴포넌트, 스키마, 확장성표기언어, 기업간 전자거래
XBRL, Core Component, Schema, XML, B2B

* 한양사이버대학교 경영학부 교수

** 한양사이버대학교 경영학부 교수, 교신저자

1. 서 론

기업의 투명성과 사회적 책임이 강조되면서, 비즈니스 보고(Business Reporting)에 특화된 XML[23] 언어인 XBRL(eXtensible Business Reporting Language)[6,7,8,24]의 활용도가 크게 증가하고 있다. 여기서 기업 보고란 일반적으로 재무제표, 재정 정보, 원장 처리, 연간 및 분기간 재무보고서와 같은 재무 정보를 기본으로 하고, 그 외의 비재무적 정보까지 포괄하는 기업의 대내외적인 모든 정보 보고를 의미한다. XBRL은 기업 보고 문서의 구성요소들(개념들과 이들의 구조체들)을 XSD(XML Schema Definition)[12,20]로 정의한다는 점에서 기존의 전자문서 표준화와 기본적으로 동일하다. 그러나, 개념들의 유형이 특정하게 한정되어 있고, 이들 개념들을 연결하고 부가 정보를 제공하기 위한 다양한 관계 유형을 XLink[21]와 XPointer[22]로 제공한다는 점에서 근본적인 차이가 있다. 달리 표현하면, XSD로 정의된 하나의 복잡한 수형 구조체로 전자문서를 규정하는 것이 기존의 접근 방법이었다고 한다면, XBRL에서는 개념들을 평면적으로 나열한 뒤 이들을 다차원적으로 연결하는 방법을 채택하고 있다. XBRL 응용이 활성화되면서 국가적인 수준에서의 표준화와 함께 체계적으로 구성요소들을 재활용해야 할 필요성이 증대되고 있으나, 이러한 근본적 차이로 인해서 표준화와 재활용에 어려움을 겪고 있다.

UN/CEFACT에서는 전자문서를 구성하는 데이터 항목을 컴포넌트로 만들어 등록/관리하고 이를 전자문서 개발 시 재활용할 수 있

는 핵심 컴포넌트 방법론과 라이브러리를 제공하고 있다[16]. 이 방법론은 국제적 EDI 표준인UN/EDIFACT개발 경험을 적절하게 반영한 것으로, 비즈니스와 관련된 의미를 가지는 요소와 그렇지 않은 요소로 구분해서 자원을 체계적으로 재활용할 수 있도록 지원한다. 여기서 후자는 전자문서를 정의하기 위한 온톨로지(Ontology) 역할을 하는 것으로 특정한 문맥(Context) 하에서 전자와 연결된다[9]. 또한 문법 중립적인 측면을 강조해서, 개념적이고 논리적인 수준에서 표준 문서에 대한 정의가 이루어지도록 하며, 이를 바탕으로 EDI나 XML 등의 다양한 문법을 적용할 수 있도록 지원한다. 이러한 방법론과 라이브러리를 적용하여 전자문서를 개발하고 활용하기 위해서는 각 산업별 또는 업종별로 진행되는 표준 전자문서 개발 과정에서부터 기존 자원에 대한 재활용 여부를 충분히 검토해야 한다. 이러한 기존 자원의 재활용은 중복된 자원의 개발을 예방하고, 표준 전자문서 개발 시간을 크게 단축시키게 된다.

핵심 컴포넌트 방법론은 현재 "XML 전자문서 개발 지침"[5]에도 반영되어 우리나라 전자문서 표준을 개발하고 심사하는 데 활용되고 있다. 이 논문에서는 이러한 UN/CEFACT의 핵심 컴포넌트 방법론을 적용하여 XBRL 문서의 표준화 및 재활용이 가능하도록 하는 데 있어서의 문제점을 분석하고 해결책을 제시하고자 한다. 이를 위하여 다음 제2장에서는 관련 연구로서 XBRL의 구성요소와 주요 개념을 소개하고, UN/CEFACT의 핵심 컴포넌트 방법론에서 사용되는 주요 개념과 표준화 절차에 대하여

정리한다. 그리고 제3장에서는 핵심 컴포넌트 방법론을 XBRL에 적용하기 위한 방안을 제시하고, 전자세금계산서를 사례로 하여 분석한다. 마지막으로 제4장에서는 결론과 향후 연구방향에 대해서 설명한다.

2. 관련 연구

2.1 XBRL

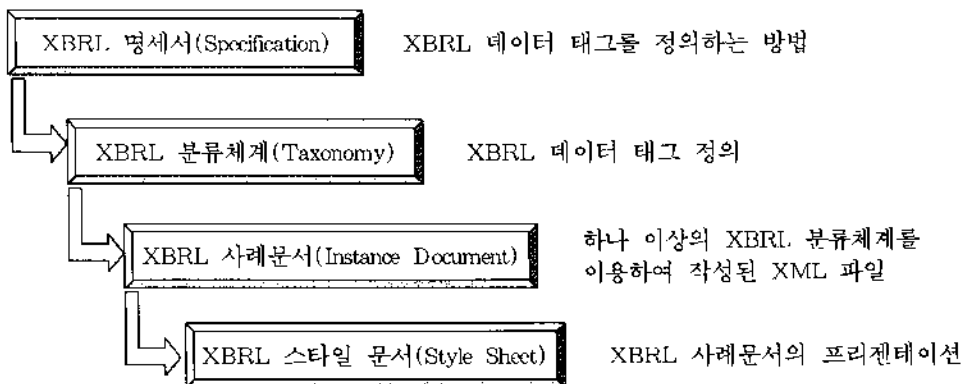
XBRL문서의 구조는 <그림 1>과 같이 4단계로 도식화할 수 있다. 이들 각각의 기술적 특징은 다음과 같다.

XBRL 명세서(Specification)는 XBRL 프레임워크(Framework)와 문법(Syntax)을 기술하여 XBRL로 기업정보 문서를 작성하려는 사람들에게 기준을 제공한다. XBRL 명세서로는 2003년12월 최초로 배포되고 2005년11월에 오류가 수정된 권고안 버전 2.1[24]이 있다. 또한 이 명세서에 의해서 작성된 문서들을 응용에서 올바르게 처리함을 보장하기 위

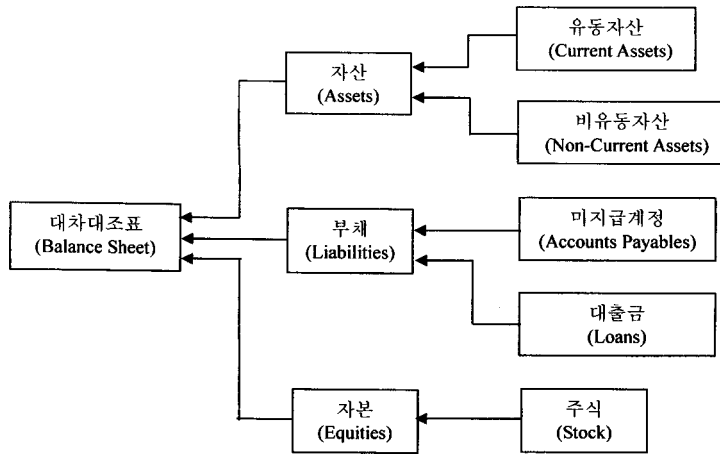
한 200개 이상의 테스트를 제공하는 일치성에 관한 후보 권고안 2.1[25]이 추가로 제공된다.

XBRL에서 기업보고는 XBRL 분류체계와 XBRL 사례문서라고 하는 두 종류의 문서들로 구성된다. XBRL 분류체계는 다시 '개념'과 '관계'라는 두 부분으로 구성된다. '개념'에 대한 정의만이 XBRL 분류체계에 포함되며, 이에 해당되는 구체적인 값은 XBRL 사례문서에 포함된다. '관계' 부분은 '개념'들 간의 관계와 함께 개념과 관련된 자원과의 관계를 포함한다. '개념' 부분은 XSD를 사용하여 정의되며, '관계' 부분은 W3C의 XLink[21]와 XPointer[22]를 사용하여 정의된다.

'개념'은 XSD의 element 요소를 사용해서 정의되는데, 아이템(item)과 튜플(tuple)의 두 종류가 있다. 아이템이라는 개념은 기업보고를 위한 구체적인 값을 포함할 수 있는 반면에, 튜플이라는 개념은 다른 개념들을 묶는 역할을 한다. <그림 2>에서 유동자산은 아이템에 해당하는 개념인 반면, 자산은 튜플에 속하는 개념이다. 개념에는 데이터 타입



<그림 1> XBRL 문서의 구조



〈그림 2〉 간단한 대차대조표의 주요 개념들[3]

(type) 이 설정되어, 포함되는 값에 대한 제약을 명확히 하게 된다. 튜플에 속하는 개념에는 tupleType이라는 단일한 데이터 타입만 정의되어 있는 반면, 아이템에 해당하는 개념에는 monetaryItemType, sharesItemType, decimalItemType, stringItemType, uriItemType, dateTimeItemType 등의 데이터 타입이 있다.

이렇게 정의된 개념들을 관계를 이용해서 상호 연결하거나 특정한 자원과 연결함으로써 분류체계를 완성하게 된다. XBRL 분류체계에서 사용되는 관계로는 정의(Definition), 계산(Calculation), 프리젠테이션(Presentation), 라벨(Label), 참조(Reference)의 5가지 종류가 있다. 앞의 3가지는 개념들을 연결하기 위한 것이고, 나머지 두 가지는 개념을 특정한 자원과 연결하기 위한 것이다.

정의(Definition) 관계는 개념들 간의 추상적인 연관 관계를 규정하기 위한 것으로, 표준으로 제공되는 관계 역할에는 일반화-구체화(general-special), 기본-대체(essence-alias),

유사-튜플(similar-tuples), 필요-엘리먼트(requires-element)의 4가지가 있다. 계산(Calculation) 관계는 개념들이 덧셈에 참여하는 방법을 정의하며, 이에 표준으로 제공되는 관계 역할로는 합산-아이템(summation-item)이 있다. 프리젠테이션 관계는 인덴테이션(Indentation)과 같이 어떻게 개념들이 화면에 제시되어야 하는지를 정의한다. 이러한 관계는 XBRL 소프트웨어에 의해서 해석되어 기업보고서를 제시하는 데 활용된다. 라벨은 개념에 사람이 읽고 이해할 수 있는 문장을 연결하는 것으로, 하나의 개념에는 다수의 라벨들이 연결될 수 있으며, 각각의 라벨은 다른 언어로 표현될 수 있다. 참조는 종이나 웹 등 다양한 자원을 개념에 연결하기 위한 것이다. 참조는 라벨과 유사한 방법으로 작동한다. 다만, 참조되는 자원에 대한 메타 데이터를 내용으로 포함하고 있다는 점이 다를 뿐이다.

이상과 같은 관계의 표현은 로케이터

(Locator)와 아크(Arc)를 이용하여 독립적으로 표현되며, 개념 정의 부분에서는 확인이 불가능하다. 여기서 로케이터는 특정한 개념을 URI로 지칭하며, 아크는 관계에 참여하는 로케이터를 지칭한다. 아크는 로케이터를 지칭하는데 출발점(from)과 종착점(to)을 설정하여 단일한 방향성을 갖는다.

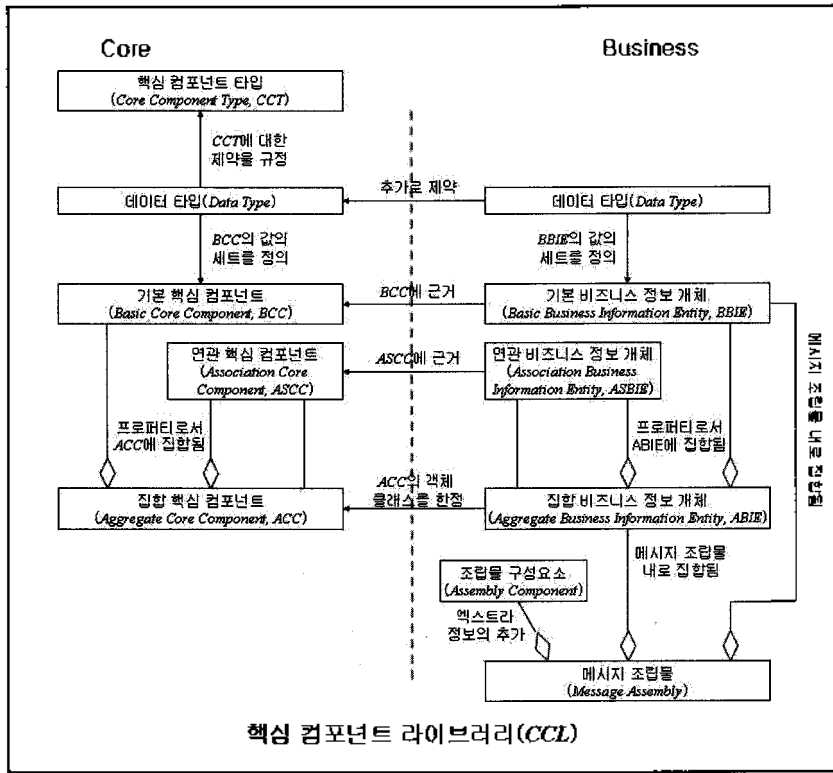
기업보고 데이터가 포함되어 있는 XBRL 문서를 XBRL 사례 문서라고 한다. 이 사례 문서는 하나 이상의 XBRL 분류체계에서 정의된 개념들과 관계들을 참조하여 사실들을 값으로 기술한다. 그리고 이러한 사실에 관한 문맥 정보도 추가적으로 제공된다. XBRL 사례 문서는 XML의 모든 규칙을 준수하는 XML 문서로서 XSD를 사용하여 검증될 수 있다. 따라서 XSD로 정의된 XBRL 분류체계에 의해서 검증될 수 있다. XBRL 사례 문서에서 핵심은 실제 사실에 대한 보고 내용인 아이템이다. 이것은 XBRL, 분류체계에서 아이템 개념으로 정의되었다. 각각의 아이템은 해당되는 개념의 이름을 사용하여 표현된다.

XBRL 스타일 문서는 XBRL 사례 문서를 사용자가 선호하는 양식으로 변환하기 위한 문서다. XML에서는 이러한 변환을 수행하기 위한 목적으로 W3C 국제표준인 XSL(eXtensible Stylesheet Language)이 제공된다. 이를 이용해서 XBRL 사례 문서를 HTML 등 다양한 형태로 변환하는 것이 가능하다.

2.2 ebXML 핵심 컴포넌트 방법론

2.2.1 핵심 컴포넌트와 비즈니스 정보 개체

비즈니스 데이터 교환에 있어서 보다 유연한 상호 작용을 지원하기 위해서는 비즈니스 어의(Business Semantics)에 관한 표준화가 선행되어야 한다. UN/CEFACT의 핵심 컴포넌트(CC : Core Component) 관련 기술은 현재 사용되는 일반적 형태의 비즈니스 데이터를 표현하는 공통의 어의 조립 블록들을 개발함으로써, 이를 신규 비즈니스 어휘 개발과 기존 비즈니스 어휘 재구조화에 활용할 수 있는 방법론을 제시한다. 이 방법론에서 핵심 컴포넌트는 UML[11,13]의 클래스 다이어그램에 기반을 두고 있기 때문에, 사실상 UML 객체 클래스에 대한 정보의 일반 표현에 해당된다. 하나의 클래스 다이어그램은 객체 클래스(Class)들, 특정 클래스를 수식하는 프로퍼티(Property)들, 그리고 클래스들간의 관계를 보여준다[16]. 클래스 다이어그램에는 이와 같이 네 가지 종류의 요소가 있기 때문에, 핵심 컴포넌트도 객체 클래스를 나타내는 집합 핵심 컴포넌트(ACC : Aggregate CC), 객체 클래스의 단순 프로퍼티를 나타내는 기본 핵심 컴포넌트(BCC : Basic CC), 하나의 객체 클래스가 또 다른 객체 클래스의 (복잡한) 프로퍼티가 되는 경우 두 객체 클래스 사이의 관계를 나타내는 연관 핵심 컴포넌트(ASCC : Association CC), 텍스트(Text), 숫자(Number), 날짜(Date)처럼 기본 핵심 컴포넌트가 가질 수 있는 정보의 유형을 정의하는 핵심 컴포넌트 타입(CCT : CC Type), 이



〈그림 3〉 핵심 컴포넌트와 비즈니스 정보 개체의 관계[16]

네 가지 종류로 분류될 수 있다. 각각의 핵심 컴포넌트에는 유일한 명칭과 함께 그 의미가 정의되며, UID(Universal Identifier)나 다수의 비즈니스 용어 혹은 동의어들도 규정된다.

비즈니스 협업에서 실제로 교환되는 정보는 핵심 컴포넌트로 정의되는 것이 아니고 비즈니스 문맥(Business Context)을 반영한 비즈니스 정보 개체(BIE : Business Information Entity)로 정의된다. 일단 비즈니스 문맥이 파악되면, 해당 비즈니스 문맥에서의 핵심 컴포넌트 사용에 필요한 한정과 정제를 고려하기 위하여 핵심 컴포넌트를 구별할 수 있다. 비즈니스 정보 개체는 핵심 컴포넌트를 특정 비

즈니스 문맥에서 사용한 결과이다. 따라서, 비즈니스 정보 개체는 각각의 핵심 컴포넌트 분류에 대해서 상응하는 방식으로 분류될 수 있다. 그 결과 집합 핵심 컴포넌트에 상응하는 집합 비즈니스 정보 개체(ABIE : Aggregate BIE), 연관 핵심 컴포넌트에 상응하는 연관 비즈니스 정보 개체(ASBIE : Association BIE), 그리고 기본 핵심 컴포넌트에 상응하는 기본 비즈니스 정보 개체(BBIE : Basic BIE)로 분류된다. 비즈니스 정보 개체는 비즈니스 프로세스 모델링을 하는 동안 파악될 수 있다. 하나의 비즈니스 정보 개체는 핵심 컴포넌트 이름에 '한정어(Qualifier)'를 추가

함으로써 상응하는 핵심 컴포넌트와 구별될 수 있다. 한정어는 객체 클래스 이름과 프로퍼티 용어에 추가될 수 있다. 이러한 핵심 컴포넌트와 비즈니스 정보 개체간의 관계의 특징은 <그림 3>과 같이 정리할 수 있다.

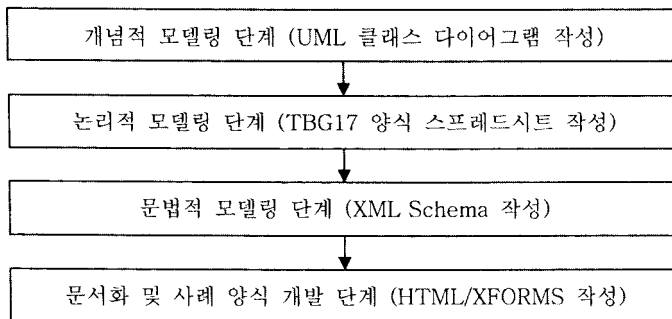
핵심 컴포넌트의 개념은 서로 다른 환경에서 교환되는 비즈니스 정보에는 많은 공통점이 존재한다는 가정에 근거를 두고 있다. 하지만 차이점이 여전히 존재하기 때문에, 비즈니스 정보의 정의와 구조상 공통점과 차이점을 관리하기 위한 목적으로 문맥이라는 개념이 도입되었다. 하나의 비즈니스 관계에 대한 문맥은 특정 비즈니스 정보 개체가 보다 일반적인 핵심 컴포넌트로부터 도출될 수 있도록 환경을 자세히 정의한다. 협업에 대한 비즈니스 문맥을 규정함으로써 비즈니스 문서(혹은 다른 데)에 있는 비즈니스 정보 개체를 통해서 교환되는 정보의 정확한 구조가 도출될 수 있다. 따라서 문맥은 핵심 컴포넌트를 비즈니스 정보 개체로 변환하는 일종의 필터가 된다.

2.2.2 전자문서 개발 절차

XML 전자문서 개발 지침[5]에서 채택하

고 있는 OASIS UBL[10] 기술위원회의 전자문서 개발 절차에는 특정 전자문서에 포함될 비즈니스 정보 개체들에 대한 UML 클래스 다이어그램을 먼저 작성하고, 이를 기반으로 UN/CEFACT TBG17의 제출 절차 및 포맷 [17,19]을 준용한 스프레드시트를 작성하여, XML 명명 및 설계규칙[18]을 준수하는 XML Schema로 자동 변환함으로써 전자문서를 문법적으로 정의하게 된다. 이후에 새로 작성된 XML Schema의 내용을 구체적인 UML 클래스 다이어그램으로 문서화하여 사용자들이 XML Schema의 내용을 쉽게 이해할 수 있도록 참고자료로 제공한다. 이상과 같은 4단계는 <그림 4>와 같이 개념적 모델링 단계, 논리적 모델링 단계, 문법적 모델링 단계, 문서화 및 사례 양식 개발 단계로 체계화할 수 있다.

개념적 모델링 단계에서는 업무 분석 단계에서 얻어진 내용을 토대로 우선 비즈니스 개체를 추출하고, 각 비즈니스 개체 내에 있는 속성들을 구성하며, 이를 클래스 다이어그램으로 정의한다[2,10]. 논리적 모델링 단계는 전 단계의 개념적 모델링 결과를 바탕으로



<그림 4> 전자문서 개발 절차

UN/CEFACT TBG17에서 정의한 스프레드 시트 양식을 이용해서 비즈니스 정보를 구체적으로 기술하게 된다. 스프레드 시트 작성 과정에서 비즈니스 정보의 발견 및 명명 (개념적 모델링 단계에서 추출한 비즈니스 용어를 기존에 개발되어 관리되고 있는 기준 항목과 매칭시키는 작업), 비즈니스 정보의 관계 차수 할당 (개념적 모델링 단계에서 추출한 비즈니스 정보들이 문서나 컴포넌트 내에서 몇 번 반복하는지에 대한 관계 차수를 할당하는 작업), 비즈니스 정보의 정의 할당 (개념적 모델링 단계에서 추출한 비즈니스 정보들에 대한 정확한 의미를 할당하는 작업) 등을 통해서 비즈니스 정보는 정규화되게 된다. 문법적 모델링 단계에서는 사용하고자 하는 구체적인 문법을 선택하여 스프레드 시트의 내용을 실제 전자거래에서 사용 가능한 스키마로 작성하는 것이다. 마지막으로 문서화 및 사례 양식 개발 단계는 문법적으로 규정된 스키마를 사용자들이 쉽게 이해하고 활용할 수 있도록 구체적인 UML 모델로 표현하며, 해당 스키마를 준수하는 사례 문서를 작성하기 위한 양식을 개발하는 단계이다.

3. 핵심 컴포넌트 방법론 기반의 XBRL 응용 문서 개발

3.1 문법적 모델링 단계

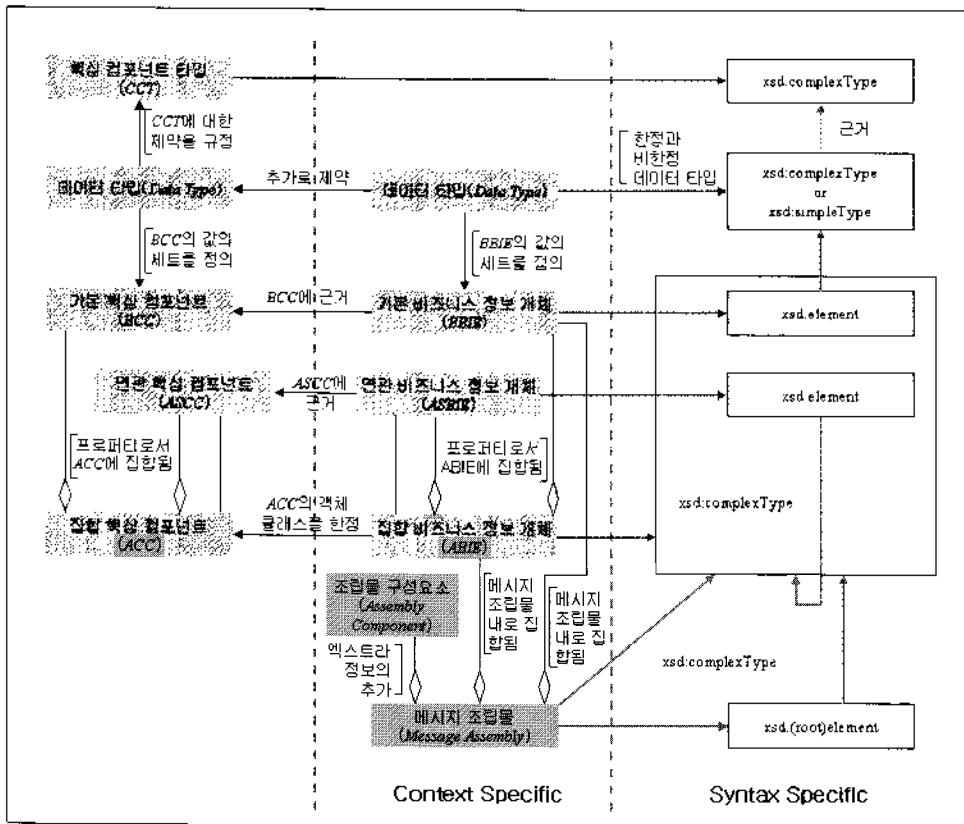
핵심 컴포넌트 방법론을 적용하기 위해서는, 먼저 문법적 모델링 단계에서 작성되는 내용을 XBRL로 작성되는 내용과 비교하는

작업이 필요하다. 왜냐하면 XBRL은 XSD(XML Schema Description)를 사용한 문법적 모델링 방법만을 제공하므로, 이 두 가지를 쉽게 비교할 수 있기 때문이다. 이 단계에서 밝혀야 할 중요한 사항은 XBRL의 어떤 구성요소들이 CC, BIE 등에 대응될 수 있는지를 확인하는 것이다.

핵심 컴포넌트 방법론에서 XSD는 CC를 기반으로 하는 BIE로부터 개발되는데, <그림 5>는 CC, BIE, XSD 구성요소 간의 관계를 보여준다.

이 그림에 나타난 관계는 다음과 같은 기본 원칙을 따르고 있다.

- 메시지는 XML 스키마의 루트 엘리먼트로 지정되는 `xsd:complexType`으로 표현된다.
- ABIE는 `xsd:complexType`으로 정의된다.
- ASBIE는 연관된 ABIE를 표현하는 `xsd:complexType` 내에서 지역 엘리먼트로 선언된다.
- BBIE는 상위 ABIE를 표현하는 `xsd:complexType` 내에서 지역 엘리먼트로 선언되며, DT를 기반으로 한다.
- DT는 `xsd:complexType`이나 `xsd:simpleType`으로 선언된다. XSD 기본 데이터 타입은 CCT의 보조 컴포넌트에 할당된 타입과 DT에 설정한 Facet이 동일한 경우에는 언제든지 사용될 수 있다.
- CCT는 `xsd:complexType`으로 정의되어야 하며, 보조 컴포넌트는 CCT에 할당된 속성으로 선언된다.



〈그림 5〉 CCTS와 XSD 구성요소간의 관계

반면에 XBRL 전자문서는 〈그림 6〉과 같이 xbrl이라고 하는 최상위 엘리먼트 내에 item, tuple, context, unit, footnoteLink가 다수 포함되는 구조를 가지고 있다. 아이템과 튜플의 경우에는 전자문서를 정의할 때 substitutionGroup을 이용하여 사용자 정의 타입으로 대체하게 되며, 이들 사용자 정의 타입 간에는 관계를 추가적으로 정의할 수 있다. 나머지 3개의 엘리먼트는 사례문서에서 직접 사용되게 된다.

3.1.1 아이템 타입과 튜플 타입

XBRL의 모든 아이템 타입은 〈표 1〉과 같이 사전에 설정된 것들과 이들 중 하나로부터 파생된 것이어야 한다. 대부분이 XSD의 기본 타입을 이용한 것이지만, monetaryItemType, sharesItemType, pureItemType, fractionItemType 등 4가지는 XBRL 자체적으로 정의하고 있다.

예컨대, XBRL에서는 monetary라고 하는 단순 타입을 확장한 monetaryItemType을 이용해서 화폐와 관련된 개념들을 표현할 수 있

```

1.      <schema targetNamespace = "http://www.xbrl.org/2003/instance"
2.          xmlns = "http://www.w3.org/2001/XMLSchema"
3.          xmlns:xbrli = "http://www.xbrl.org/2003/instance"
4.          xmlns:link = "http://www.xbrl.org/2003/linkbase"
5.          elementFormDefault = "qualified">
6.      <element name = "xbrl">
7.          <annotation>
8.              <documentation>XBRL instance root element.</documentation>
9.          </annotation>
10.         <complexType>
11.             <sequence>
12.                 <element ref = "link:schemaRef" minOccurs = "1" maxOccurs = "unbounded" />
13.                 <element ref = "link:linkbaseRef" minOccurs = "0" maxOccurs = "unbounded" />
14.                 <element ref = "link:roleRef" minOccurs = "0" maxOccurs = "unbounded" />
15.                 <element ref = "link:arcroleRef" minOccurs = "0" maxOccurs = "unbounded" />
16.                 <choice minOccurs = "0" maxOccurs = "unbounded">
17.                     <element ref = "xbrli:item" />
18.                     <element ref = "xbrli:tuple" />
19.                     <element ref = "xbrli:context" />
20.                     <element ref = "xbrli:unit" />
21.                     <element ref = "link:footnoteLink" />
22.                 </choice>
23.             </sequence>
24.             <attribute name = "id" type = "ID" use = "optional" />
25.             <anyAttribute namespace = "http://www.w3.org/XML/1998/namespace"
26.                 processContents = "lax" />
27.         </complexType>
28.     </element>
29. </schema>
    
```

<그림 6> XBRL 전자문서의 구조

<표 1> XBRL에서 정의된 아이템 타입

XBRL Item Type	XSD Base Type	XBRL Item Type	XSD Base Type
decimalItemType	decimal	stringItemType	string
floatItemType	float	booleanItemType	boolean
doubleItemType	double	hexBinaryItemType	hexBinary
integerItemType	integer	base64BinaryItemType	base64Binary
nonPositiveIntegerItemType	nonPositiveInteger	anyURIItemType	anyURI
negativeIntegerItemType	negativeInteger	QNameItemType	QName
longItemType	long	durationItemType	duration
intItemType	int	dateTimeItemType	dateTime
shortItemType	short	timeItemType	time
byteItemType	byte	dateItemType	date

〈표 1〉 XBRL에서 정의된 아이템 타입

XBRL Item Type	XSD Base Type	XBRL Item Type	XSD Base Type
nonNegativeIntegerItemType	nonNegativeInteger	gYearMonthItemType	gYearMonth
unsignedLongItemType	unsignedLong	gYearItemType	gYear
unsignedIntItemType	unsignedInt	gMonthDayItemType	gMonthDay
unsignedShortItemType	unsignedShort	gDayItemType	gDay
unsignedByteItemType	unsignedByte	gMonthItemType	gMonth
positiveIntegerItemType	positiveInteger	normalizedStringItemType	normalizedString
monetaryItemType	xbri:monetary	tokenItemType	token
sharesItemType	xbri:shares	languageItemType	language
pureItemType	xbri:pure	NameItemType	Name
fractionItemType	정수인 numerator 와 0이 아닌 정수 인 denominator의 복합체	NCNameItemType	NCName

도록 하고 있는데, 이것은 핵심 컴포넌트 방법론에서의 데이터타입 역할을 하게 된다. 다만 유의할 점은 핵심 컴포넌트 방법론에서는 엘리먼트나 타입을 명명하는 데 대문자 카멜 케이스(UCC : Upper-camel-case)를 사용하여야 한다는 점이다. 이러한 요구사항은 “국내 XML 전자문서 개발 지침”[5]에 규정된 것으로, 정확하게 이를 준수하여 국내 표준문서로 등록하고자 하는 경우에는 엘리먼트나 타입의 이름을 부분적으로 수정해야 함을 의미한다. 그러나 이러한 이름의 수정은 기존의 XBRL 표준을 위반하게 되므로 적절한 중재 방안이 마련되어야 한다. 따라서 XBRL의 모든 아이템 타입은 monetaryItemType과 유사한 방법을 이용하여 데이터타입으로 처리할 수 있다.

튜플의 경우에는 사전에 정의된 타입이 존

재하지 않으며, 따라서 각 문서를 정의할 때 필요한 튜플은 ACC 및 ABIE를 이용해서 정의될 수 있다.

3.1.2 XBRL 사례문서의 엘리먼트

아이템과 튜플 이외에도 추가적으로 고려되어야 할 사항으로는 모든 XBRL 사례문서에 존재할 수 있는 특별한 종류의 엘리먼트에 대해 어떻게 처리할 것인가 하는 문제이다. 즉, Taxonomy에는 포함되지 않지만 모든 XBRL 사례문서에 존재할 수 있는 context, unit, footnoteLink와 같은 엘리먼트에 대한 처리가 필요하다. 이는 ABIE와 ASBIE를 통해서 정의할 수 있다.

예컨대, context의 경우는 context는 〈그림 7〉과 같이 정의되어 있어서, entity, period, scenario의 세 가지로 구성된 ABIE로 간주할

```

1.      <element name="context">
2.          <complexType>
3.              <sequence>
4.                  <element name="entity" type="xbri:contextEntityType" />
5.                  <element name="period" type="xbri:contextPeriodType" />
6.                  <element name="scenario" type="xbri:contextScenarioType" minOccurs="0" />
7.              </sequence>
8.              <attribute name="id" type="ID" use="required" />
9.          </complexType>
10.     </element>
11.     <complexType name="contextScenarioType">
12.         <sequence>
13.             <any namespace="# #other" processContents="lax"
14.                 minOccurs="1" maxOccurs="unbounded" />
15.         </sequence>
16.     </complexType>
17.     <simpleType name="dateUnion"><union memberTypes="date dateTime" /></simpleType>
18.     <complexType name="contextPeriodType">
19.         <choice>
20.             <sequence>
21.                 <element name="startDate" type="xbri:dateUnion" />
22.                 <element name="endDate" type="xbri:dateUnion" />
23.             </sequence>
24.             <element name="instant" type="xbri:dateUnion" />
25.             <element name="forever"><complexType /></element>
26.         </choice>
27.     </complexType>
28.     <complexType name="contextEntityType">
29.         <sequence>
30.             <element name="identifier">
31.                 <complexType>
32.                     <simpleContent>
33.                         <extension base="token">
34.                             <attribute name="scheme" use="required">
35.                                 <simpleType>
36.                                     <restriction base="anyURI">
37.                                         <minLength value="1" />
38.                                     </restriction>
39.                                 </simpleType>
40.                             </attribute>
41.                         </extension>
42.                     </simpleContent>
43.                 </complexType>
44.             </element>
45.             <element ref="xbri:segment" minOccurs="0" />
46.         </sequence>
47.     </complexType>
48.     <element name="segment">
49.         <complexType>
50.             <sequence>
51.                 <any namespace="# #other" processContents="lax"
52.                     minOccurs="1" maxOccurs="unbounded" />
53.             </sequence>
54.         </complexType>
55.     </element>

```

〈그림 7〉 context 엘리먼트의 정의

수 있다. entity는 identifier와 segment로 구성된 또 다른 ABIE로 볼 수 있으며, context와는 ASBIE로 연결될 수 있다.

3.1.3 Taxonomy 사이의 링크

마지막으로 고려되어야 할 것이 Taxonomy 요소들간의 링크 관계다. 이러한 링크 관계는 전자문서의 구조에 관한 직접적인 내용이라기 보다는 엘리먼트 간의 제약사항이나 보조적인 정보를 제공하기 위한 것이며, 현재의 전자문서 표준화 대상에서 대부분이 제외되어 있는 내용이다. 이들에 대한 모델링은 개념적이고 논리적인 수준에서 이루어지고 있고, 따라서 이들은 XLink와 XPointer로 변환하는 수준에서 이루어지는 것이 바람직하다.

3.2 전자문서 사례

현재 전자문서 표준으로 채택되어 사용하고 있는 전자세금계산서를 대상으로 XBRL을 적용함으로써 비즈니스 보고 분야에서의 CC 방법론의 유용성을 구체적으로 설명하고자 한다. 표준 전자세금계산서의 개략적인 구조는 <표 2>와 같다. 이 표에서는 DataArea에 포함되는 Summary의 요소들이 모두 제시되어 있는데, 여기서는 이들을 XBRL로 표현하고 재활용하는 방법을 설명한다. 기존의 XBRL을 적용하여 이 부분을 표현하게 되면 <그림 8>과 같은 형태로 표현할 수 있고, XBRL의 관계는 <그림 9>와 같이 표현할 수 있다. <그림 9>를 보면 ChargeAmount와 TotalTaxAmount를 더한 것이 TotalAmount이라는 것을 계산 관계로서 정의되고 있다는

<표 2> 표준 전자세금계산서의 개략적인 구조

ApplicationArea		Creation. DateTime	
		...	
DataArea	Header	Document. DateTime	
		...	
	Line	Line. Number	
		...	
	Summary	Charge. Amount	
		TotalTax. Amount	
		Total. Amount	
		PaymentMethod	
			Payment.Method. Code
			DomesticCurrency. Amount
	ForeignCurrency. Amount		
	TotalForeignCharge. Amount		

```

<element name="ChargeAmount" type="xbri:monetaryItemType" substitutionGroup="xbri:item"
  nillable="true" id="id_ChargeAmount" xbrli:periodType="instant"/>
<element name="TotalTaxAmount" type="xbri:monetaryItemType" substitutionGroup="xbri:item"
  nillable="true" id="id_TotalTaxAmount" xbrli:periodType="instant"/>
<element name="TotalQuantity" type="xbri:nonNegativeIntegerItemType"
  substitutionGroup="xbri:item" nillable="true" id="id_TotalAmount" xbrli:periodType="instant"/>
<element name="TotalAmount" type="xbri:monetaryItemType" substitutionGroup="xbri:item"
  nillable="true" id="id_TotalAmount" xbrli:periodType="instant"/>
<element name="PaymentMethodCode" type="xbri:monetaryItemType" s
  substitutionGroup="xbri:item" nillable="true" id="id_PaymentMethodCode"
  xbrli:periodType="instant"/>
<element name="DomesticCurrencyAmount" type="xbri:monetaryItemType"
  substitutionGroup="xbri:item" nillable="true" id="id_DomesticCurrencyAmount"
  xbrli:periodType="instant"/>
<element name="ForeignCurrencyAmount" type="xbri:monetaryItemType"
  substitutionGroup="xbri:item" nillable="true" id="id_ForeignCurrencyAmount"
  xbrli:periodType="instant"/>
<element name="TotalForeignChargeAmount" type="xbri:monetaryItemType"
  substitutionGroup="xbri:item" nillable="true" id="id_TotalForeignChargeAmount"
  xbrli:periodType="instant"/>
<element name="PaymentMethod" abstract="false" substitutionGroup="xbri:tuple"
  id="id_PaymentMethod">
  <complexType>
    <sequence>
      <element ref="prefix0:id_PaymentMethodCode"/>
      <element ref="prefix0:id_DomesticCurrencyAmount"/>
      <element ref="prefix0:id_ForeignCurrencyAmount"/>
    </sequence>
  </complexType>
</element>
<element name="Summary" abstract="false" substitutionGroup="xbri:tuple"
  id="id_Summary">
  <complexType>
    <sequence>
      <element ref="prefix0:id_ChargeAmount"/>
      <element ref="prefix0:id_TotalTaxAmount"/>
      <element ref="prefix0:it_TotalQuantity"/>
      <element ref="prefix0:it_TotalAmount"/>
      <element ref="prefix0:id_PaymentMethod"/>
      <element ref="prefix0:id_TotalForeignChargeAmount"/>
    </sequence>
  </complexType>
</element>

```

〈그림 8〉 XBRL에 의한 전자세금세산서의 개념 표현

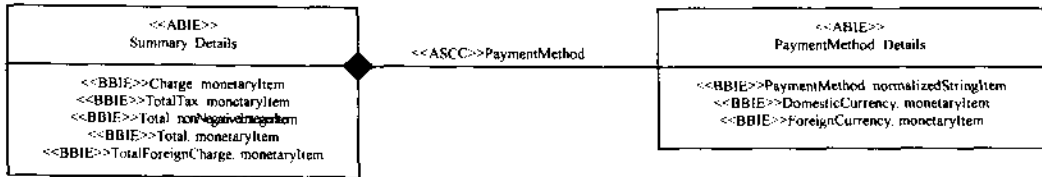
것을 파악할 수 있다. 크게 보면 XBRL에서 는 평면적으로 아이템들을 먼저 정의하고, 구조화하며, 이들간의 관계를 XLink와 XPointer 로 표현하게 된다는 점이 기존의 스키마 개발 방법과 차이가 있음을 알 수 있다.

〈그림 8〉과 같은 스키마를 개발하는 과정 에서 핵심 컴포넌트 개발 방법론을 적용하기 위해서는 먼저 개념적 모델링 단계에서 UML 클래스 다이어그램을 작성하는 것이 필요하 다. 〈그림 10〉은 전자세금세산서의 Summary

```

<calculationLink xlink:type="extended"
  xlink:role="http://www.xbrl.org/2003/role/ValidationCheck">
  <loc xlink:type="locator" xlink:href="DigitalTaxInvoice.xsd#id_ChargeAmount"
    xlink:label="loc_426305264" xlink:title="TotalAmount"/>
  <loc xlink:type="locator" xlink:href="DigitalTaxInvoice.xsd#id_TotalAmount"
    xlink:label="loc_31540560" xlink:title="ValueofSupply"/>
  <calculationArc xlink:type="arc"
    xlink:arcrole="http://www.xbrl.org/2003/arcrole/summation-item"
    xlink:from="loc_426305264" xlink:to="loc_31540560"
    weight="1" order="1.0" xlink:use="optional"/>
  <loc xlink:type="locator" xlink:href="DigitalTaxInvoice.xsd#id_TotalTaxAmount"
    xlink:label="loc_483257080" xlink:title="TotalTaxAmount"/>
  <calculationArc xlink:type="arc"
    xlink:arcrole="http://www.xbrl.org/2003/arcrole/summation-item"
    xlink:from="loc_426305264" xlink:to="loc_483257080"
    weight="1" order="2.0" xlink:use="optional" />
</calculationLink>
    
```

〈그림 9〉 전자세금계산서 XBRL 관계 표현



〈그림 10〉 전자세금계산서의 Summary에 대한 개념적 모델링

〈표 3〉 전자세금계산서의 Summary에 대한 논리적 모델링 (TBG17 품)

Dictionary Entry Name (English)	Indicator	Referenced CC		...
		Source	Entry Name	
Summary. Details	ABIE			...
Charge.monetaryItem	BBIE			...
TotalTax.monetaryItem	BBIE			...
Total.nonNegativeIntegerItem	BBIE			...
PaymentMethod	ASBIE			
PaymentMethod. Details	ABIE			
PaymentMethod.normalizedStringItem	BBIE			
DomesticCurrency.monetaryItem	BBIE			
ForeignCurrency.monetaryItem	BBIE			

```

<xsd:complexType name="SummaryType">
  <xsd:sequence>
    <xsd:element name="ChargeMonetaryItem" id="id_ChargeMonetaryItem"
      type="udt:monetaryItemType"/>
    <xsd:element name="TotalTaxMonetaryItem" id="id_TotalTaxMonetaryItem"
      type="udt:monetaryItemType"/>
    <xsd:element name="TotalNonNegativeIntegerItem" id="id_TotalNonNegativeIntegerItem"
      type="udt:nonNegativeIntegerItemType"/>
    <xsd:element name="TotalMonetaryItem" id="id_TotalMonetaryItem"
      type="udt:monetaryItemType"/>
    <xsd:element name="PaymentMethod" id="id_PaymentMethod"
      type="ram:PaymentMethodType"/>
    <xsd:element name="TotalForeignChargeMonetaryItem"
      id="id_TotalForeignChargeMonetaryItem" type="udt:monetaryItemType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="PaymentMethodType">
  <xsd:sequence>
    <xsd:element name="PaymentMethodNormalizedStringItem"
      id="id_PaymentMethodNormalizedStringItem"
      type="udt:normalizedStringItemType"/>
    <xsd:element name="DomesticCurrencyMonetaryItem"
      id="id_DomesticCurrencyMonetaryItem"
      type="udt:monetaryItemType"/>
    <xsd:element name="ForeignCurrencyMonetaryItem"
      id="id_ForeignCurrencyMonetaryItem"
      type="udt:monetaryItemType"/>
  </xsd:sequence>
</xsd:complexType>

```

(a) BBIE 포함 방식

```

<xsd:element name="ChargeMonetaryItem" id="id_ChargeMonetaryItem" type="udt:monetaryItemType"/>
<xsd:element name="TotalTaxMonetaryItem" id="id_TotalTaxMonetaryItem" type="udt:monetaryItemType"/>
<xsd:element name="TotalNonNegativeIntegerItem" id="id_TotalNonNegativeIntegerItem"
  type="udt:nonNegativeIntegerItemType"/>
<xsd:element name="TotalMonetaryItem" id="id_TotalMonetaryItem" type="udt:monetaryItemType"/>
<xsd:element name="PaymentMethod" id="id_PaymentMethod" type="ram:PaymentMethodType"/>
<xsd:element name="TotalForeignChargeMonetaryItem" id="id_TotalForeignChargeMonetaryItem"
  type="udt:monetaryItemType"/>
<xsd:element name="PaymentMethodNormalizedStringItem" id="id_PaymentMethodNormalizedStringItem"
  type="udt:normalizedStringItemType"/>
<xsd:element name="DomesticCurrencyMonetaryItem" id="id_DomesticCurrencyMonetaryItem"
  type="udt:monetaryItemType"/>
<xsd:element name="ForeignCurrencyMonetaryItem" id="id_ForeignCurrencyMonetaryItem"
  type="udt:monetaryItemType"/>
<xsd:complexType name="SummaryType">
  <xsd:sequence>
    <xsd:element ref="ram:id_ChargeMonetaryItem"/>
    <xsd:element ref="ram:id_TotalTaxMonetaryItem"/>
    <xsd:element ref="ram:id_TotalNonNegativeIntegerItem"/>
    <xsd:element ref="ram:id_TotalMonetaryItem"/>
    <xsd:element ref="ram:id_PaymentMethod"/>
    <xsd:element ref="ram:id_TotalForeignChargeMonetaryItem"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="PaymentMethodType">
  <xsd:sequence>
    <xsd:element name="ram:id_PaymentMethodCode"/>
    <xsd:element name="ram:id_DomesticCurrencyAmount"/>
    <xsd:element name="ram:id_ForeignCurrencyAmount"/>
  </xsd:sequence>
</xsd:complexType>

```

(b) BBIE 참조 방식

〈그림 11〉 핵심 컴포넌트 방법론 기반의 XBRL 전자세금계산서 참조 표현

부분에 대한 스키마를 UML 클래스 다이어그램으로 표현한 것이다. 이 다이어그램에서 <<ABIE>>, <<BBIE>>, <<ASBIE>>는 스테레오 타입을 표현한다. 참고로 이 다이어그램에서 monetaryItem과 같이 첫 번째 문자가 소문자로 표현되고 있는데 이는 명명규칙에 위배되는 것이라는 점을 유의해야 한다.

이렇게 작성된 클래스 다이어그램을 바탕으로 <표 3>과 같은 형식의 논리적 모델링이 진행된다. 여기서는 참조한 CC 등에 대한 보다 상세한 정보가 추가적으로 기술되게 된다.

이러한 논리적 모델링을 바탕으로 <그림 11>과 같은 논리적 모델링이 진행되는데, 이러한 작업은 자동화가 가능하다. <그림 11>의 (a)에서는 문서의 구조적인 특성을 그대로 반영하여 BBIE 엘리먼트들을 ABIE 타입에 포함시키고 있는 반면에, (b)에서는 BBIE 엘리먼트들을 먼저 열거하고 이를 참조하여 ABIE 타입을 구성하는 형식을 취함으로써 XBRL에 보다 충실한 형태가 되도록 하고 있다. 이러한 문법적 결과물에 대하여 XBRL의 속성과 값을 추가하고, 이름공간을 일부 조정하며, 모든 사례 문서에 공통적으로 포함되는 엘리먼트들을 추가함으로써 XBRL의 결과물을 최종적으로 완성하게 된다. 이 과정에서 우리는 BIE, CC, DT 등에 대한 체계적인 정리와 재활용을 할 수 있음을 확인할 수 있다.

4. 결 론

본 연구에서는 문법적 수준에서 개발이 이루어지고 있는 XBRL 응용에 대하여, 개념적

이고 논리적인 수준에서 모델링이 진행됨으로써 자원의 재활용이 용이하고 국가적인 수준에서의 문서 표준화도 가능하도록 지원하기 위하여, 핵심 컴포넌트 개발 방법론을 적용하는 방안을 제시하였다. 구체적으로 전자세금계산서에 이러한 방안을 적용하여 그 가능성을 검토하였다. 개념적이고 논리적인 수준에서의 모델링은 문서 구성요소에 대한 인식을 쉽게 할 수 있도록 지원함으로써, 중복된 자원의 개발을 예방하고, XBRL 전자문서 개발 시간을 크게 단축시킬 수 있다.

핵심 컴포넌트를 기반으로 하는 전자문서 개발 방법론과 XBRL 전자문서의 개발 방법을 상호 참조하거나 연동할 수 있는 국제 표준이나 국내 표준화 활동은 아직 이루어지지 않고 있다. 그럼에도 본 논문에서 제시한 방안을 토대로 보다 활발한 표준화 논의의 진행을 위해서 향후 추진하고자 하는 연구 방향은 다음과 같다. 먼저, 다양한 비즈니스 보고에 적용하여 CC/BIE 라이브러리를 정립하고 확장하는 작업이 필요하다. 현재 국가적으로 운용되고 있는 CC/BIE 라이브러리에는 비즈니스 보고 관련 요소가 빠져있어서 이러한 작업에 큰 의미가 있다. 무엇보다도 금융, 환경 등의 분야에서 비즈니스 보고를 위한 수요가 급속히 증가함에 따라[1,3,4,14,15,26] 이러한 라이브러리가 조속히 정립되어야 필요성이 있다. 둘째로, 이 논문에서 제시한 방안이 보다 효율적으로 적용되기 위해서는 XBRL과 CC 방법론 모두에 반영되면 좋은 사항들도 있다. 이를 위해서는 표준화 과정에 참여하여 적극 반영하는 작업이 필요하다. 셋째로 관계에 관한 모델링이 개념적이고 논리적인 수준

에서 이루어지고, 이것을 문법적인 수준에서 생성하는 과정을 지원하는 것이 필요하다. 마지막으로 이 논문에서 제시한 방안에 따라서 단계별로 진행되는 과정을 자동화된 도구로 지원함으로써, 세부적인 사항에 매몰되지 않고 사람의 역할이 정말로 필요한 부분에 집중할 수 있을 것이다.

참 고 문 헌

- [1] 구정욱과 김강정, "XBRL과 회계정보투명성," *회계정보연구*, 제22권 제2호, 2004년6월, pp. 179-202.
- [2] 김형도, "비즈니스 프로세스 및 정보모델 기반의 전자문서 구조 개발지침 (ECIF 104 : 2004)," *2004 ECIF 정기총회 및 기념심포지엄 발표 논문집*, 2004년3월.
- [3] 김형도, "중소기업 생산성 향상을 위한 XBRL기반의 지속가능성 보고 방안 연구," *생산성논집*, 제19권, 제4호, 2005년 12월, pp. 147-169.
- [4] 노미현, "인터넷 재무보고와 XBRL에 관한 연구," *인터넷비즈니스연구*, 2005, pp. 139-159.
- [5] 한국전자거래진흥원, *XML 전자문서 개발 지침 v3.0*, 2005년5월.
- [6] 한국전자거래진흥원, *2006 e-Biz 표준화 백서*, 2006년8월.
- [7] Cohen, E. E. and N. Hannon (2000), "How XBRL Will Change Your Practice," *The CPA journal*, Vol. 70, No. 11, 2000, pp. 36-41.
- [8] Debreceeny, R. and G.L. Gray (2001), "The Production and Use of Semantically Rich Accounting Reports on the Internet : XML and XBRL," *Int'l Journal of Accounting Information Systems*, Vol. 2, 2001, pp. 47-74.
- [9] Hofreiter, B. and C. Huemer, "B2B Integration - Aligning ebXML and Ontology Approaches," *Proceedings of the First EurAsian Conf. on Advances in Information and Communication Technology*, Teheran, Iran, October 2002.
- [10] OASIS, *Universal Business Language 1.0*, September 2004.
- [11] OMG, *Unified Modeling Language Specification (Version 1.3)*, <http://www.omg.org/>, 1999.
- [12] Roy, J. and A. Ramanujan, "XML Schema Language : Taking XML to the Next Level," *IT Professional*, Vol. 3, No. 2, 2001, pp. 37-40.
- [13] Rumbaugh, J., I. Jacobson, G. Booch, *The Unified Modeling Language Reference Manual*, Wesley, 1999.
- [14] Seetharman A. and Subramaniam R. and S. Y. Shyong, "Internet Financial Reporting : Problems and Prospects," *Corporate Finance Review*, Sep/Oct 2005, pp. 29-35.
- [15] Seetharman A. and Subramaniam R. and S. Y. Shyong, "Internet Financial

- Reporting : Problems and Prospects (Part II).” *Corporate Finance Review*, Jul/Aug, pp. 23-34.
- [16] UN/CEFACT, *Core Component Technical Specification v2.01*, November 2003.
- [17] UN/CEFACT, *Submission Guidelines and Procedures v1.0*, September 2004.
- [18] UN/CEFACT, *XML Naming and Design Rules v2.0*, February 2006.
- [19] UN/CEFACT, *TBG17 Library Administration Procedures v1.0*, March 2006.
- [20] W3C, *XML Schema*. <http://www.w3.org/XML/Schema>. 2006.
- [21] W3C, *XML Linking Language (XLink) Version 1.0*. <http://www.w3.org/TR/xlink/>, June 2001.
- [22] W3C, *XML Pointer Language (XPointer)*, <http://www.w3.org/TR/xptr/>, August 2002.
- [23] W3C, *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, <http://www.w3.org/TR/xml>, August 2006.
- [24] XBRL International (2005a), *XBRL 2.1*, <http://www.xbrl.org/SpecRecommendations/>, Nov. 2005.
- [25] XBRL International(2005b), *XBRL 2.1 Conformance Suite*. <http://www.xbrl.org/SpecRecommendations/>, 2005.
- [26] XBRL Korea (2004), *Korean GAAP Taxonomy*. <http://www.xbrl.org/Taxonomy/korean/fr/gaap/ci/pfs/2004-01-19/korean-gaap-ci-pfs-2004-01-19.doc>, 2004.

저 자 소 개



- 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, 이키텍처, 객체지향모델링, 데이터 마이닝 등



박찬권

(E-mail : chankwon@hycu.ac.kr)

1987.

서울대학교 산업공학과 (학사)

1989.

서울대학교 대학원 산업공학과 (석사)

1996.

서울대학교 대학원 산업공학과 (박사)

1989~2002.

LG전자(주)

1996~1997.

서울대학교 자동화시스템연구소 특별연구원

1997~2003.

영산대학교 정보경영학부 교수

2003~현재

한양사이버대학교 경영학부 교수

2006~현재

한국전자거래진흥원 전자문서워킹그룹위원장

관심분야

XML, 전자문서, ERP/SCM, 정보시스템 개발 방법론