XML 라이브러리의 효율적 재사용을 위한 비즈니스 정보 개발 방안
(Research on a Development of Business Information for the Efficient Reuse of XML Libraries)

박찬권, 김형도

한양사이버대학교 경영학부
Tel: +82- 2290-{2813, 2812}, Fax: +82- 2-2290-2828
E-mail: { chankwon, hdkim }@hycu.ac.kr

Abstract

XML 기술의 발전에 따라 확장성 있는 ebXML(Electronic Business eXtensible Markup Language) 기반의 스키마 사용이 확산되면서 점차 기존 EDI 전자문서나 DTD(Document Type Definition) 구조로 되어 있는 전자문서를 XML 스키마 형태로 변환하거나 새롭게 개발하고 있는 추세이다. 한국전자거래진흥원에서는 XML 전자문서 개발 지침과 함께 XML 라이브러리를 개발하여 제시함으로써 표준 XML 전자문서의 개발을 지원하고 있다. XML 전자문서 개발 지침은 ebXML CCTS 기술규격에서 제시하고 있는 방법론을 준용하고, UN/CEFACT ATG 그룹에서 개발한 XML Naming & Design Rule 규격을 국내에 맞게 수정하여 적용하고 있다. 또한 XML 라이브러리는 전자문서 개발지침에 따라 사전에 정의하고 개발해 놓은 핵심 컴포넌트와 비즈니스 정보 개체들로 구성되어 있다. 하지만 전자문서 개발 지침상의 일부 규칙들이 의미상 명확하지 않은 상태에서 개발자의 임의적 선택의 폭을 허용하거나 개체의 명명에 대한 뚜렷한 기준을 제시하지 못함으로써 핵심 컴포넌트와 비즈니스 정보 개체의 검색 및 재사용을 제한하고 라이브러리 유지보수를 어렵게 하고 있는 실정이다. 본 연구에서는 XML 전자문서 개발지침 상의 일부 규칙에 대한 문제점을 분석, 규칙을 구체화함으로써 개발자 중립적인 핵심 컴포넌트와 비즈니스 정보 개체의 개발을 지원하고 라이브러리 관리를 효율적으로 수행할 수 있는 방안을 제시하고자 한다.

키워드:

XML전자문서, 핵심컴포넌트(Core Component), XML 라이브러리, 기업간전자거래

1. 서론

초기에 기업간 전자거래에 사용되던 전자문서는 주로 EDI를 기반으로 개발되었다. 하지만 최근 XML 기술의 발전에 따라 확장성 있는 ebXML(Electronic Business eXtensible Markup Language) 기반의 스키마 사용이 확산되면서 점차 기존 EDI 전자문서나 DTD(Document Type Definition) 구조로 되어 있는 전자문서를 XML 스키마 형태로 변환하거나 새롭게 개발하고 있는 추세이다[1, 2].

한편 업종별 B2B 네트워크 구축사업을 통해 산업별 전자문서 표준화 작업이 진행되면서, XML/EDI 전자문서, XML 전자문서 등이 개발되어 기업간 전자거래에 적용되었다. 하지만 업종들이 산업 표준 전자문서 개발을 하던 시기에는 전자문서 개발을 위한 표준 지침도 없었고 참고할 수 있는 라이브러리도 존재하지 않아 전자문서 개발에 많은 혼선이 있었다[2]. 이후 한국전자거래진흥원에서는 XML 전자문서 개발지침[3]과 함께 XML 라이브러리를 개발하여 제시함으로써 XML 전자문서 개발 지침을 준수하고 XML 라이브러리를 활용한 표준 XML 전자문서의 개발을 지원하고 있다. 이에 따라 무역, 해상운송, 전력, 조달 등 많은 업종에서 XML 표준 전자문서가 개발되고 있다[4].

XML 전자문서 개발 지침은 ebXML CCTS(Core Component Technical Specification) 기술규격[8]에서 제시하고 있는 방법론을 준용하고, UN/CEFACT ATG 그룹에서 개발한 XML Naming & Design Rule 규격[11]을 국내에 맞게 수정하여 적용하고 있다[4]. 또한 XML 라이브러리[4]는 XML 전자문서를 효율적으로 개발하거나 문서의 표준화를 도모하기 위하여 공통적으로 사용되는 항목들을 전자문서 개발지침에 따라 핵심 컴포넌트와 비즈니스 정보 개체로 개발하여 제공하고 있다. 하지만 전자문서 개발 지침상의 일부 규칙들이 의미상 명확하지 않은 상태에서 개발자의 임의적 선택의 폭을 허용하거나 개체의 명명에 대한 뚜렷한 기준을 제시하지 못함으로써 핵심 컴포넌트와 비즈니스 정보개체의 검색과 재사용을 제한하고 XML 라이브러리의 유지보수를 어렵게 하고 있는 실정이다[2].

본 연구에서는 XML 전자문서 개발지침 상의 일부 규칙에 대한 문제점을 분석하여 규칙을 보완, 구체화함으로써 개발자 중립적인 핵심 컴포넌트와 비즈니스 정보 개체의 개발을 지원하고 라이브러리

관리를 효율적으로 수행할 수 있는 방안을 제시하고자 한다.

이 논문은 다음과 같이 구성되어 있다. 2절에서는 UN/CEFACT의 핵심 컴포넌트 방법론[8]에서 사용되는 주요 개념과 전자문서 개발 절차에 대하여 정리한다. 3절에서는 XML 전자문서 개발지침[3]에서 제시하고 있는 XML 라이브러리의 검색, 핵심 컴포넌트 및 비즈니스 정보 개체 설계에 대한 규칙들을 살펴보고 이들 규칙이 가지고 있는 문제점들을 분석함으로써 개선방안을 제시한다. 4절에서는 개선 방안을 토대로 보다 체계적으로 XML 라이브러리를 검색하고 비즈니스 정보들을 개발할 수 있는 방법들을 구체적으로 제시하고자 한다. 마지막으로 5절에서는 본 연구의 결과를 정리하고, 추후 연구 연구방향에 대해 논의하고자 한다.

2. 핵심 컴포넌트 방법론

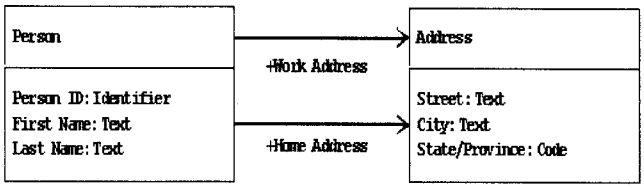
가. 핵심 컴포넌트와 비즈니스 정보 개체

비즈니스 데이터 교환에 있어서 보다 유연한 상호작용을 지원하기 위해서는 비즈니스 어의(Business Semantics)에 관한 표준화가 선행되어야 한다. 핵심 컴포넌트는 이러한 요구를 포착하고 재사용성을 최대화 하도록 사용되는 표준화된 데이터 요소이다. 즉, 전자문서를 조합하기 위한 의미(Semantic) 상의 빌딩블럭이라고 할 수 있다[8]. 이 같은 개념에서 출발한 UN/CEFACT의 핵심 컴포넌트(CC: Core Component) 방법론은 문서 구조에 대한 정의를 쉽게 할 수 있고, 과거의 정의를 여러 가지 방법으로 재활용할 수 있도록 핵심 컴포넌트(CC), 비즈니스 정보 개체 (BIE: Business Information Entity), 비즈니스 문맥 (Business Context) 등의 개념을 제공한다[8]. 이 가운데 핵심 컴포넌트는 집합 코어 컴포넌트(ACC: Aggregate CC)와 기본 코어 컴포넌트(BCC: Basic CC), 그리고 연관 코어 컴포넌트(ASCC: Association CC)로 구분하고 여기에 추가적으로 텍스트, 숫자, 날짜처럼 기본 핵심 컴포넌트(BCC)가 가질 수 있는 정보의 유형을 코어 컴포넌트 타입(CCT: CC Type)으로 정의할 수 있도록 하고 있다.

이러한 핵심 컴포넌트는 UML[6, 7]의 클래스 다이어그램에 기반을 두고 있기 때문에 사실상 UML 객체 클래스를 이용한 개념적인 모델링을 통해 전자문서의 개발을 진행할 수 있게 된다[8]. 일반적으로 하나의 클래스 다이어그램은 객체 클래스들과 특정 객체 클래스의 속성을 기술하는 프로퍼티들 그리고 객체 클래스들 간의 관계들을 표현한다. 핵심 컴포넌트는 이러한 UML의 객체 클래스 다이어그램 표현과 1:1로 직접 대응시킬 수 있다. 즉, 클래스 다이어그램으로부터 모두 네 개의 모델링 요소를 구분해낼 수 있는데, 그에 따라 핵심 컴포넌트도 다음과 같이 모두 네 가지 종류로 구분해서 대응시킬 수 있게 된다.

- 객체 클래스와 집합 핵심 컴포넌트(ACC)
- 객체 클래스의 프로퍼티와 기본 핵심 컴포넌트(BCC)
- 객체 클래스들 사이의 관계(Relationship)와 연관 핵심 컴포넌트(ASCC)
- 객체 클래스 프로퍼티의 데이터 타입과 핵심 컴포넌트 타입(CCT)

[그림1]과 같은 클래스 다이어그램을 예로 들면 'Person' 클래스와 'Address' 클래스는 모두 집합 핵심 컴포넌트(ACC)가 될 수 있으며, 'Person' 클래스의 'Person ID', 'First Name', 'Last Name' 프로퍼티와 'Address' 클래스의 Street', City', 'State/Province' 프로퍼티는 각각 기본 핵심 컴포넌트(BCC)들로 정의할 수 있다. 또한 두 객체 클래스 사이에 정의되는 'Work Address'와 'Home Address'라는 관계는 연관 핵심 컴포넌트(ASCC)로 정의된다.



[그림 1] 클래스 다이어그램의 예

각각의 핵심 컴포넌트에는 유일한 명칭과 함께 그 의미가 정의되며 라이브러리를 통해서 관리되는데 이 때 각각의 핵심 컴포넌트는 유일하게 구분될 수 있는 이름 즉, 사전기재이름을 갖게 된다. 집합 핵심 컴포넌트(ACC)는 객체 클래스 이름과 'Details'라는 예약어를 구두점과 공백으로 연결함으로써 사전기재이름을 정의하고, 기본 핵심 컴포넌트(BCC)는 객체 클래스 이름과 프로퍼티 이름, 그리고 프로퍼티가 가지는 값의 유형을 의미하는 데이터 타입(Data Type)을 각각 구두점과 공백으로 연결해서 사전기재이름을 작성한다. 또한 연관 핵심 컴포넌트(ASCC)는 객체 클래스 이름과 관계의 역할, 그리고 연관된 객체 클래스의 이름을 각각 구두점과 공백으로 연결시켜 정의하고 있다.

비즈니스 협업에서 실제로 교환되는 정보는 핵심 컴포넌트로 정의되는 것이 아니고, 비즈니스 문맥을 반영한 비즈니스 정보 개체(BIE)를 통해 정의된다. 즉, 적절한 핵심 컴포넌트(CC)를 찾아서 비즈니스 문맥에 맞게 제약을 가함으로써 비즈니스 정보 개체(BIE)를 만들고(혹은 비즈니스 문맥에 맞게 적절하게 제약된 비즈니스 정보 개체를 검색해서), 이들 비즈니스 정보 개체(BIE)들을

적절하게 조합함으로써 비즈니스 문서를 구성하게 되는 것이다[8].

이 과정에서 핵심 컴포넌트(CC)는 비즈니스 정보 개체(BIE)의 기초 역할을 하게 되고, 비즈니스 정보 개체(BIE)는 핵심 컴포넌트(CC)를 특정 비즈니스 문맥에서 사용한 결과가 된다. 따라서 각각의 핵심 컴포넌트에 상응하는 비즈니스 정보 개체(BIE)들이 정의될 수 있다. 이에 따라 비즈니스 정보 개체는 집합 핵심 컴포넌트(ACC)에 상응하는 집합 비즈니스 정보개체(ABIE: Aggregate BIE), 연관 핵심 컴포넌트(ASCC)에 상응하는 연관 비즈니스 정보 개체(ASBIE: Association BIE), 그리고 기본 핵심 컴포넌트(BCC)에 상응하는 기본 비즈니스 정보 개체(BBIE: Basic BIE)로 분류된다. 하나의 비즈니스 정보 개체(BIE)는 핵심 컴포넌트(CC)의 객체 클래스 이름, 프로퍼티 이름, 그리고 데이터 타입이나 연관된 핵심 컴포넌트 이름에 각각 필요한 '한정어(Qualifier)'를 추가함으로써 상응하는 핵심 컴포넌트(CC)의 사전기재이름과 구별될 수 있다[8].

나. 전자문서 개발 절차

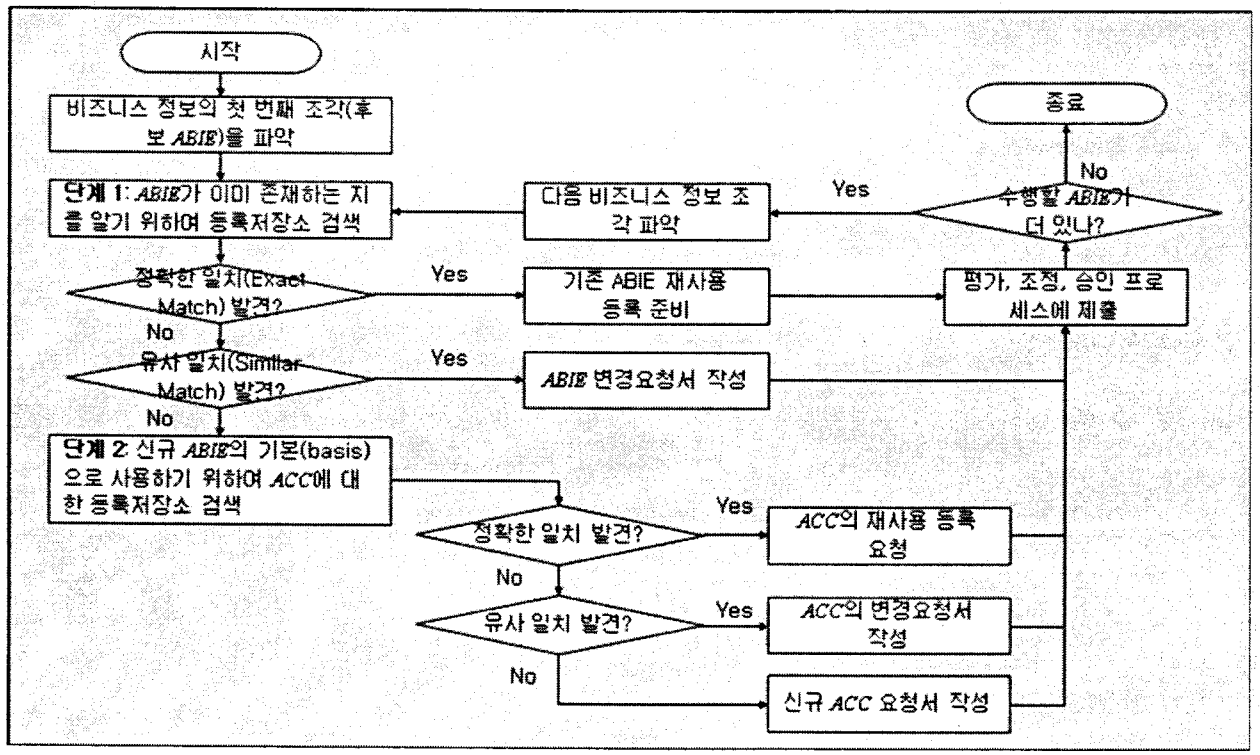
핵심 컴포넌트 방법론은 주어진 비즈니스 상황이나 환경 즉, 비즈니스 문맥과 독립적인 표준화된 데이터 요소를 핵심 컴포넌트로 정의하고, 이를 핵심 컴포넌트 라이브러리에 저장한 다음 특정 비즈니스 문맥에 맞게 재사용함으로써 XML 전자문서를 만들어낼 수 있도록 하는 방법이다[8]. 이를 위해서는 핵심 컴포넌트(CC)와 비즈니스 정보 개체(BIE)를 XML 라이브러리에 등록하고, 필요에 따라 이를 재사용함으로써 전자문서를 개발하는 체계적인 절차가 요구된다[8]. 이에 따라 XML 전자문서 개발지침[3]에서는 XML 라이브러리로부터 핵심 컴포넌트(CC)와 비즈니스 정보 개체(BIE)를 검색하는 절차를 [그림2]와 같이 제시하고 있으며, 한국전자거래진흥원에서는 REMKO[4]에 이들을 검색할 수 있는 등록저장소를 운영하고 있다.

한편 XML 전자문서 개발 지침[3]에서 채택하고 있는 OASIS UBL[5] 기술위원회의 전자문서 개발 절차에서는 특정 전자문서에 포함될 비즈니스 정보 개체들에 대한 UML 클래스 다이어그램을 먼저 작성하고, 이를 기반으로 UN/CEFACT TBG17의 제출 절차 및 포맷[9, 10]을 준용한 스프레드시트를 작성하여, XML 명명 및 설계규칙[11]을 준수하는 XML Schema로 자동 변환함으로써 전자문서를 문법적으로 정의하도록 하고 있다. 이후에 새로 작성된 XML Schema의 내용을 구체적인 UML 클래스 다이어그램으로 문서화하여 사용자들이 XML Schema의 내용을 쉽게 이해할 수 있도록 참고자료로 제공한다.

3. XML 전자문서 개발 지침 분석

가. 주요 개발 규칙과 문제점

XML 전자문서 개발지침[3]은 전자문서 개발을 위한 기본 원칙과 비즈니스 정보 개발 원칙, 그리고



[그림 2] 비즈니스 정보의 발견 절차

비즈니스 정보의 명명에 대해 모두 57개의 규칙을 제시하고 있다. 이때 비즈니스 정보란 핵심 컴포넌트(CC)와 비즈니스 정보 개체(BIE) 모두를 가리키는 표현이다[3].

이들 규칙 가운데 XML 라이브러리 즉, 등록저장소의 검색과 비즈니스 정보의 설계에 직접적으로 영향을 미치는 규칙은 약 30여 가지에 이르지만 본 연구에서는 5개 규칙을 대상으로 하고 있다. 이들 규칙을 정리하면 다음 [표1]과 같다. 여기서 'Must'와 'Should'는 지침어휘로서 'Must'는 반드시 따라야 할 필수 요소, 'Should'는 가급적 지침의 규칙을 따를 것을 권고하는 선택 요소를 각각 의미한다.

[표 1] 주요 개발 규칙

구분	규칙	내용
전자문서 개발원칙	8	비즈니스 정보 개체와 비즈니스 용어는 명확히 구분하여야 한다. (Must)
비즈니스 정보 개발	25	비즈니스 정보 간에 동의어가 존재하는 경우, 동의어 처리를 한다. (Should)
비즈니스 정보 검색	27	비즈니스 정보의 발견 절차는 [그림2]와 같은 절차를 반복하여야 한다. (Should)
비즈니스 정보 명명	30	비즈니스 정보의 이름은 한글을 기준으로 해야 하며, 한글과 대응하는 영문명을 병기하여야 한다. (Must)
	56	DT의 사전기재이름에서, 표현용어 이름은 CCTS 규격에서 제시하고 있는 표현용어 목록에 규정된 1차나 2차 표현용어 중 하나이어야 한다. (Must)

비즈니스 정보의 검색과 설계 과정에서 이들 규칙들이 가지는 문제점은 다음과 같다.

- [규칙8]: 비즈니스 용어를 대표할 수 있는 비즈니스 정보 개체(BIE)의 정의 방법이 제시되지 않고 있어서 비즈니스 용어를 비즈니스 정보 개체(BIE)의 사전기재이름으로 무분별하게 혼용해서 사용하게 됨
- [규칙25]: 비즈니스 정보 즉, 핵심 컴포넌트와 비즈니스 정보 개체 사이의 동의어 처리 원칙만 제시하고 핵심 컴포넌트와 비즈니스 정보 개체를 구성하는 객체 클래스 용어, 프로퍼티 용어, 표현용어, 그리고 한정어 사이의 동의어 사용 기준과 방법이 제시되지 못함으로써 실제적인 동의어 관리가 이루어지지 않고 있다. 이에 따라 비즈니스 정보의 재사용에 많은 제약이 따르고 있다.
- [규칙27]: [그림2]에서 보는 것처럼 핵심 컴포넌트와 비즈니스 정보의 재사용여부를 결정짓는 중요한 기준은 요구사항에 부합되는 핵심 컴포넌트나 비즈니스 정보개체를 찾아냈는지에 달려있다. 이 같은 기준은 현재 '완전 일치(exact match)'와 '유사 일치(similar match)'로 구분하고 있으나 완전 일치와 유사 일치를 판정하는 기준 자체가 정의되어 있지 않은 관계로 검색된 개체를 재사용할 지 혹은 신규 개체로 등록할 지를 판단하기 어렵다.
- [규칙30]: 사전적으로 한글에 상응하는 여러 영문 동의어가 존재함으로써 개발자별로 서로 다른 영문명을 사용하게 되고, 이로 인하여 핵심 컴포넌트와 비즈니스 정보 개체의 수가 늘어나지만 그에 따라 재사용은 어려워지게 됨
- [규칙56]: 데이터 타입에 대한 1차 표현용어와 2차 표현용어는 모두 20개가 주어진다. 이 가운데 데이터 값의 명료성으로 인하여 어떤 데이터 타입을 적용해야 하는지가 분명한 표현용어가 있는 반면, 일부 표현용어들은 동일한 원시타입이 적용됨으로써 어떤 조건에서 각각의 표현용어를 사용할 지가 분명하게 정의되지 않고 있다. 이로 인하여 개발자별로 상이한 데이터 타입을 적용하는 혼란이 초래되고 있다.

이 밖에 [규칙40]과 [규칙41], 그리고 [규칙46]은 기본 핵심 컴포넌트(BCC)와 연관 핵심 컴포넌트(ASCC), 그리고 집합 핵심 컴포넌트(ACC)에 대한 사전기재이름을 정의하는 방법을 제시하고 있다. 이들 규칙은 전자문서 개발절차의 2단계 논리적 모델링을 진행하는 과정에서 [그림1]의 예와 같은 접근방법에 의해 사전기재이름을 정의하는 것을 전제로 제시된 규칙들이다. 하지만 문제는 비교적 간단한 클래스 다이어그램이라고 하더라도 클래스들 사이에는 다양한 형태의 관계(relationship)들이 존재할 수 있다는 점이다. 따라서 이러한 다양한 형태의 관계를 논리적으로 설계하는 명확한 방법이 제시되어 있지 않아서 연관 핵심 컴포넌트(ASCC)와 집합 핵심 컴포넌트(ACC)를 설계하는 데 어려움이 따르게 된다.

나. 개선 방안

앞서 분석된 문제점을 개선하기 위한 방안은 크게 두 가지 관점에서 접근할 수 있다. 하나는 핵심 컴포넌트와 비즈니스 정보 개체에 대한 검색 기준과 절차의 개선이며 다른 하나는 신규 핵심 컴포넌트의 설계 기준을 개선하는 것이다.

검색 기준과 절차의 개선을 위해서는 무엇보다도 '완전 일치'와 '유사 일치'를 판정할 수 있는 세부 기준이 제시되어야 하고, 비즈니스 용어에 대한 동의어 검색 기준과 방법이 개선되어야 한다. 이 가운데 비즈니스 용어에 대한 동의어 검색 기준과 방법은 신규 핵심 컴포넌트 설계시 동의어를 어떻게 반영하고 관리할 것인 지와 밀접하게

관련된다.

신규 핵심 컴포넌트의 설계 기준을 개선하는 문제는 클래스 다이어그램으로부터 다양한 관계의 유형을 고려해서 이들 관계들을 핵심 컴포넌트의 사전기재이름으로 변환하는 방법과 데이터 타입의 적용 기준을 상세화 하는 방법으로 개선할 수 있다. 이와 함께 사전기재이름에 대한 비즈니스 용어 관리 기준을 제시하고, 객체 클래스 용어 및 프로퍼티 용어에 대한 동의어 관리 기준과 방법을 구체화해야 한다.

본 연구에서는 이들 개선 방안 가운데, '완전 일치'와 '유사 일치'에 대한 판정 기준의 개발, 클래스 다이어그램으로부터 핵심 컴포넌트로의 맵핑 기준 개발, 그리고 데이터 타입의 적용 기준 제시를 중심으로 XML 라이브러리 상의 비즈니스 정보의 재사용성을 높일 수 있는 방법을 제시하였다.

4. 핵심 컴포넌트 검색 기준 및 설계 기준

가. '완전 일치'와 '유사 일치' 판정 기준

등록저장소 즉, XML 라이브러리로부터 집합 핵심 컴포넌트(ACC)를 검색할 때 검색된 집합 핵심 컴포넌트가 사용자 요구와 완전 일치하기 위해서는 핵심 컴포넌트의 정의와 비즈니스 용어를 포함하여 동일한 클래스 명을 사용하여야 한다. 정의와 비즈니스 용어를 포함한다는 의미는 클래스 명이 부분적으로 다르다고 하더라도 정의와 비즈니스 용어 등에서 본질적으로 '완전 일치'하는 핵심 컴포넌트로 판정할 수 있다는 의미이다. 아울러 속성의 개수 즉, ACC를 구성하고 있는 BCC와 ASCC의 수가 요구하는 속성 개수와 같거나 많아야 하고, 이들 BCC와 ASCC가 모두 요구하는 속성과 완전 일치해야 한다.

동일한 클래스 명이더라도 요구하는 속성의 수보다 적은 수의 BCC 혹은 ASCC를 가지는 경우는 기존 ACC의 확장이 필요한 '유사 일치'로 판정할 수 있다. 또한 요구하는 속성의 개수 기준을 만족시키더라도 개별 BCC나 ASCC가 '유사 일치'인 경우는 ACC가 '유사 일치'하는 것으로 판정할 수 있다.

한편, BCC의 경우는 기본적으로 프로퍼티 명과 사용 데이터 타입이 동일하고 반복 수(cardinality) 또한 일치해야 '완전 일치'로 판정할 수 있으며, '유사 일치'의 경우라도 데이터 타입은 반드시 동일해야 하며 반복 수는 달라도 된다. ASCC의 경우는 관계를 표현하는 프로퍼티 명이 동일하고 연관되는 ACC가 '완전 일치' ACC일 때 '완전 일치'로 판정할 수 있다. 이 때 반복 수 또한 동일해야 한다. 만약 연관되는 ACC가 '유사 일치'이거나 요구하는 반복 수와 다를 경우는 '유사 일치' ASCC로 간주될 수 있다. 이상의 기준을 정리하면 [표2]와 같이 요약될 수 있다.

나. 클래스 다이어그램과 핵심 컴포넌트의 맵핑

일반적인 클래스 다이어그램은 객체 클래스와 객체 클래스 사이의 관계(relationship)를 표현한다. 이 가운데 관계는 기본적으로 '연관'에 의해서 표현되는데, 연관의 특수한 형태로 집합연관(aggregation)과 복합연관(composition)이 사용된다. 또한 객체의 클래스화와 관련해서는 일반화 요소가 많이 사용될 수 있다. 일반화의 경우는 어의 중립적인 핵심 컴포넌트의 정의와 재사용 메커니즘에 가장 부합되는 모형화 요소로서 이러한 일반화 관계를 적절히 활용할 때 핵심 컴포넌트의 재사용성이 극대화 될 수 있다. 여기서는 집합연관과 복합연관, 그리고 일반화(generalization)에 대한 표현을 핵심 컴포넌트로 변환할 수 있는

[표 2] ACC의 완전일치(Exact match)와 유사일치(Similar match) 판단 기준

		ACC	
		완전 일치	유사 일치
클래스 명		• 동일 클래스 명(정의 및 비즈니스 용어 포함)	• 동일 클래스 명(정의 및 비즈니스 용어 포함)
속성의 개수		• 요구하는 속성 개수(즉, BCC와 ASCC의 수)보다 많거나 동일	• BCC 혹은 ASCC의 개수가 요구하는 속성의 수보다 적은 경우
BCC와 ASCC		• 아래 기준에 따라 구성 BCC와 ASCC들이 모두 완전일치일 것	• 아래 기준에 따라 구성 BCC와 ASCC들이 완전일치 혹은 유사일치일 것
BCC	프로퍼티 명과 데이터 타입	• 요구하는 프로퍼티(BCC)와 동일한 프로퍼티 명(정의 및 비즈니스 용어 포함) • 동일한 데이터 타입	• 요구하는 프로퍼티(BCC)와 유사한 프로퍼티 명 • 동일한 데이터 타입
	반복 수	• 동일한 반복 수	• 요구하는 반복 수와 다른 경우
ASCC	프로퍼티 명과 연관된 ACC	• 요구하는 프로퍼티(연관)와 동일한 프로퍼티 명(연관된 ACC가 동일한 ACC일 경우)	• 요구하는 프로퍼티(연관)와 동일 또는 유사한 프로퍼티 명(연관된 ACC가 요구하는 ACC와 유사한 경우)
	반복 수	• 동일한 반복 수	• 요구하는 반복 수와 다른 경우

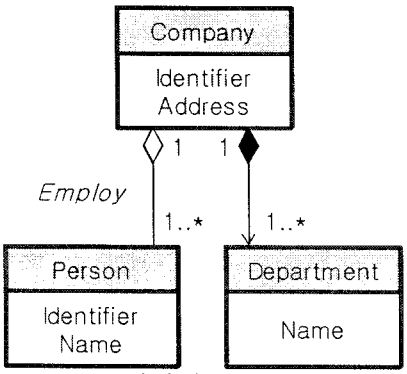
방법을 제시한다.

1) 집합연관과 복합연관

집합연관은 기본적으로 'has_a'의 관계로 [그림3]에서 보는 것처럼 '전체'(빈 다이어몬드 기호)와 '부분'의 관계를 명확하게 표현한다. 복합연관은 집합연관의 하나로 '전체'(검은색 다이어몬드 기호)가 '부분'의 생성과 삭제를 담당하며, '전체'가 삭제되면 '부분'도 연쇄 삭제됨으로써 동일한 수명주기를 가지는 관계를 묘사한다.

이와 같은 집합연관과 복합연관에 대한 핵심 컴포넌트 표현은 큰 차이가 없으며 동일한 방법에 의해서 사전기재이름을 정의할 수 있다. 즉, 집합연관과 복합연관 모두 '부분' 클래스를 소유하는 '전체' 클래스의 ASCC로 표현할 수 있다. 따라서 연관 핵심 컴포넌트(ASCC)의 사전기재이름은 다음과 같은 방법으로 정의할 수 있다.

- 객체 클래스 용어는 '전체'에 해당하는 클래스(다이어몬드 기호가 있는 클래스)의 이름을 사용
- 프로퍼티 용어는 집합(복합) 연관의 연관이름을 사용하되, 연관이름이 명시되지 않을 경우 'Has'를 프로퍼티 용어로 사용
- 연관된 ACC 이름으로는 '부분'에 해당하는 클래스(다이어몬드 기호가 없는 쪽 클래스)의 이름을 사용



[그림3] 집합연관과 복합연관

이에 따라 [그림3]의 클래스 다이어그램에 대해서는 다음의 [표3]과 같은 핵심 컴포넌트들이 정의될 수 있다.

[표3] 집합연관과 복합연관에 대한 ASCC 표현

CC 유형	사전기재이름
ACC	Company. Details
BCC	Company. Identifier
BCC	Company. Address. Text
ACC	Person. Details
BCC	Person. Identifier
BCC	Person. Name. Text
ACC	Department. Details
BCC	Department. Name. Text
ASCC	Company. Employee. Person
ASCC	Company. Has. Department

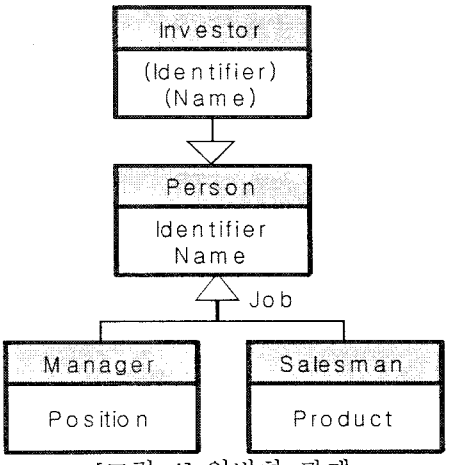
2) 일반화(generalization)

일반화는 [그림4]의 예에서 보는 것처럼 일반화된 사물(수퍼 클래스 혹은 부모 클래스, 빈 화살표 머리가 표시된 쪽)과 좀 더 특수화된 사물(서브 클래스 혹은 자식 클래스) 사이의 관계를 표현한다. 이 때의 관계는 보통 'is_a'의 의미를 가지게 된다. 일반화 관계에서 자식 클래스는 부모 클래스의 프로퍼티를 상속받으며, 경우에 따라서는 추가적인 프로퍼티를 가질 수 있다.

일반화 관계는 자식 클래스에 추가적인 프로퍼티가 있는 경우와 그렇지 않은 경우로 나누어서 접근할 수 있다. 자식 클래스에 추가 프로퍼티가 있는 경우는 자식 클래스와 부모 클래스 각각에 해당하는 ACC를 정의하고, 부모 클래스를 자식 클래스의 ASCC로 표현하되 ASCC의 반복수는 1로 설정해야 한다. 이는 자식 클래스의 추가 프로퍼티가 부모로부터 상속받은 프로퍼티의 확장에 해당되기 때문이다. 이때 ASCC의 사전기재이름은 다음과 같은 방법으로 정의할 수 있다.

- 클래스 용어는 자식 클래스의 클래스 이름
- 프로퍼티 용어는 일반화 관계를 명확히 하기 위해 'Is A'를 사용
- 연관된 ACC 이름은 부모 클래스 이름을 사용

이에 따라 [그림4]의 예에서는 <Manager. Is A. Person>과 <Salesman. Is A. Person>이라는 2개의 ASCC를 정의할 수 있게 된다.



[그림 4] 일반화 관계

한편, 자식 클래스에 추가 프로퍼티가 없는 경우는 자식 클래스가 부모 클래스의 제한

(restriction)에 해당되기 때문에 이 때는 별도의 자식 클래스를 위한 별도의 ACC를 정의하지 않고, 부모 클래스를 한정해서 집합 비즈니스 정보개체(ABIE)로 정의하도록 한다. 따라서 자식 클래스에 대한 ABIE의 사전기재이름은 다음과 같이 정의될 수 있다.

- 클래스 용어는 부모 클래스의 클래스 이름
- 클래스 한정어는 자식 클래스의 클래스 이름

이에 따라 [그림4]의 예에서 <Investor_ Person. Details>라는 ABIE가 정의될 수 있으며, [그림4]를 통해서 정의되는 핵심 컴포넌트와 비즈니스 정보개체는 [표4]와 같이 정리할 수 있다.

[표4] 일반화 관계에 대한 CC와 BIE

CC 유형	사전기재이름
ACC	Person. Details
BCC	Person. Identifier
BCC	Person. Name. Text
ACC	Manager. Details
BCC	Manager. Position. Text
ACC	Salesman. Details
BCC	Department. Product. Code
ASCC	Manager. Is A. Person
ASCC	Salesman. Is A. Person
ABIE	Investor_ Person. Details

다. 데이터 타입의 상세 적용 기준

XML 전자문서 개발지침의 [규칙56]에서는 모두 20개의 1차 혹은 2차 표현용어 중에 적절한 표현용어를 통해 데이터 타입을 적용하도록 하고 있다. 이들 표현용어들은 모두 decimal과 integer, string, binary를 원시타입으로 사용하고 있는데, 이 가운데 개발자들이 가장 혼동하는 부분은 decimal 원시타입을 사용하는 데이터 타입과 string 원시타입을 사용하는 일부 데이터 타입이다. 본 논문에서는 UN/CEFACT CCTS v2.01[8]을 토대로 이들 데이터 타입의 적용 기준을 구체적으로 제시하고자 한다.

1) decimal 원시 타입을 사용하는 데이터 타입

Decimal 원시 타입을 사용하는 표현용어로는 금액(Amount) 데이터 타입, 치수(Measure) 데이터 타입 그리고 수량(Quantity) 타입이 있다. 수치(Numeric)의 경우는 decimal인 integer를 선택해서 사용할 수 있다. 이들 표현용어는 모두 1차 표현용어들로 이들의 적용이 어려운 이유는 이들이 단독으로 사용되기 보다는 해당 값을 보충적으로 표현하는 '단위'와 함께 사용되기 때문이다. 따라서 이들 표현용어의 적용기준은 어떤 단위를 적용할 지, 그리고 해당 단위가 어떤 특성을 가졌는지를 파악하는 데서부터 비롯된다고 할 수 있다. 이들 표현용어의 적용기준을 정리하면 [표5]와 같이 나타낼 수 있다.

[표 5] 금액, 치수, 수량, 수치 데이터 타입의 구분

1차 표현용어	정의	사용
금액 (Amount)	통화의 단위가 명시적 혹은 묵시적으로 주어질 때 해당 통화로 지정되는 화폐단위의 수	• 비용, 보수, 요금과 같이 일반적인 화폐 가액을 표현할 때 사용 • ISO 4217의 통화 단위를 기본으로 사용 • 원시타입은 decimal로 표현
치수 (Measure)	측정의 단위가 명시적 혹은 묵시적으로 주어질 때 어떤 객체를 측정함으로써 결정되는 숫자 값	• 숫자 값은 객체를 측정해서 결정되고, 측정은 측정 단위로 기술됨 • 측정 단위는 UN/ECE Rec. 20의 측정단위를 디폴트로 사용 • 이 표현용어는 measured coefficients (예, m/s)에도 사용됨 • 시각과 시각 사이의 간격으로 측정되는 시간도 시간 단위와 함께 사용됨 • 페이지, 박스, 개 등은 측정단위가 아님(Quantity가 적절) • 원시타입은 decimal로 표현
수량 (Quantity)	분수(fractions)를 포함해서 비금전적 단위로 계수된(counted) 숫자	• 이 표현용어는 계수된 수량 단위에 사용됨 • 측정단위를 필요로 하는 "Measure "와 혼동해서는 안 됨(Quantity는 여러 객체 인스턴스의 수를 세는 개념으로, Measure는 측정 단위를 통해 어떤 객체의 속성을 표현하는 개념으로 구분할 수 있음) • 원시타입은 decimal로 표현
수치 (Numeric)	수량(Quantity), 치수(Measure), 혹은 금액(Amount)에 대한 단위를 필요로 하지 않는 숫자 정보	• 순서매김(sequencing), 계산, 계수(count)로 할당되거나 결정되는 숫자 정보를 표현하는 데 사용 • 수량단위나 계량단위는 요구하지 않음. • 이 표현용어는 Ratios(즉, 2개의 단위가 포함되지 않거나 그들이 동일한 경우의 비율)와 퍼센트 등에도 사용됨 • 숫자 값은 숫자로 표현된 차수 없는 수리적인 값 • 원시타입은 decimal 혹은 integer로 표현

2) string 원시 타입을 사용하는 데이터 타입

String 원시 타입을 사용하는 표현용어로는 코드(Code), 식별자(Identifier), 지시자(Indicator), 그리고 날짜시각(DateTime)이 있다. 이 가운데 어떤 타입을 적용할 지 가장 판단이 어려운 것은 코드(Code), 식별자(Identifier), 지시자(Indicator)이다. 이들은 모두 1차 표현용어로서 이들 표현용어를 적용하기 위한 기준을 구분하면 [표6]과 같이 정리할 수 있다.

[표 6] 코드, 식별자, 지시자 데이터 타입의 구분

1차 표현용어	정의	사용
코드 (Code)	간결성 혹은 언어 독립성의 목적으로 관련 보조 정보와 함께 하나의 속성에 대한 일정한 값이나 텍스트를 대표하거나 대체하는 데 사용될 수 있는 문자 스트링 (문자, 숫자, 혹은 기호들)	• 간결성은 어떤 속성에 대해 텍스트나 값이 복잡하거나 반복적일 때 이를 간결하게 대표 혹은 대체하는 것을 의미 • 이때 속성은 각 객체 인스턴스를 식별하는 속성이 아니므로, 해당 문자 스트링을 기준으로 복수의 객체 인스턴스들을 개념적으로 그룹핑할 수 있음(예: 성별. 코드) • 문자스트링이 어떤 객체클래스의 인스턴스나 실세계의 객체를 유일하게 식별하는 경우에는 식별자 (Identifier) 표현용어를 사용해야 함 • 원시타입은 string으로 표현
식별자 (Identifier)	관련되는 보충 정보와 함께 특정 식별 체계에 있는 하나의 객체 인스턴스를 동일한 식별 체계 하의 다른 객체들로부터 유일하게 식별하고 구별해낼 수 있도록 하는 문자 스트링	• 주민등록번호, 차량번호 등과 같은 식별 체계 안에서 한 객체의 한 인스턴스를 다른 모든 객체에 대해 유일하게 식별하고 구별하기 위해 사용(식별자는 정보를 다루는 모든 체계에서 내부적으로 사용되는데, 정보를 처리하기 위해서는 그 정보를 가리킬 방법이 있어야 하기 때문임) • 코드(code) 타입과 구분이 필요함 • 원시타입은 string으로 표현
지시자 (Indicator)	하나의 프로퍼티가 유일하게 가질 수 있는 상태를 표현하는 상호 배타적인 2개의 값의 목록	• 전형적으로 yes/no나 on/off 혹은 true/false 등과 같이 한 개의 Boolean 값을 선택할 때 사용 • 3개 이상의 값의 목록이 필요할 경우는 Code. Type을 사용해야 함 • 원시타입은 string으로 표현

5. 결론

본 연구에서는 XML 전자문서 개발지침 상의 일부 규칙에 대한 문제점을 분석하여 규칙을 보완, 구체화함으로써 개발자 중립적인 핵심 컴포넌트와 비즈니스 정보 개체의 개발을 지원하고 라이브러리 관리를 효율적으로 수행할 수 있는 방안을 제시하였다. 이를 위해서 핵심 컴포넌트의 검색 과정에서 컴포넌트의 재사용성을 높이기 위해 '완전 일치'와 '유사 일치'를 명확히 판단할 수 있는 기준을 제시하였고, 클래스 다이어그램으로부터 핵심 컴포넌트를 정의하기 위해 집합연관과 일반화 관계의 변환 방법도 개발하였다. 그리고 개발자들이 가장 많은 혼돈을 일으키는 데이터 타입의 구체적 적용 기준도 제시하였다.

본 연구에 덧붙여 추가적으로 보완해야 할 사항은 앞서 제안했던 동의어 관리 기준이다. 핵심 컴포넌트 기반의 전자문서 개발 방법론의 유용성은 전자문서를 개발하고 사용하는 기관들 사이에서 이들 핵심 컴포넌트와 비즈니스 정보 개체들을 제출받아 등록하고 관리하는 XML 라이브러리의 품질을 유지하는 데 있다고 하겠다. 동의어를 중심으로 한 사전기재이름에 대한 연구가 추가될 경우 본 연구에서 제시된 방법과 함께 보다 높은 품질의 라이브러리를 유지하는 데 기여함으로써 재사용성을 크게 제고할 수 있을 것으로 기대된다.

References

[1] 안경림, 김동희, 박찬권, 박정천, "철도 물류 정보 표준화 방안 및 정보시스템 개선에 대한 연구," 「한국전자거래학회논문지」 제 13 권, 제 3 호, 2008. 8, pp. 121-135.

[2] 산업자원부, 한국전자거래진흥원, 「e-비즈니스 표준화 백서」 2006. 8.

[3] 한국전자거래진흥원, 「XML 전자문서 개발 지침 v3.0」 2005. 5.

[4] 한국전자거래진흥원, XML 등록저장소, www.remko.or.kr

[5] OASIS, Universal Business Language 1.0, September 2004.

[6] OMG, Unified Modeling Language Specification (Version 1.3), http://www.omg.org/, 1999.

[7] Rumbaugh, J., I. Jacobson, G. Booch, The Unified Modeling Language Reference Manual, Wesley, 1999.

[8] UN/CEFACT, Core Component Technical Specification v2.01, November 2003.

[9] UN/CEFACT, Submission Guidelines and Procedures v1.0, September 2004.

[10] UN/CEFACT, TBG17 Library Administration Procedures v1.0, March 2006a.

[11] UN/CEFACT, XML Naming and Design Rules v2.0, February 2006b.