핵심 컴포넌트 저장을 위한 시스템 설계 및 구현

황 정 희[†] · 김 영 균[††] · 류 근 호[†††]

요 약

ebXML은 XML기반의 새로운 국제 표준 전자상거래 프레임워크이다. 서로 다른 산업분야 간의 상호운용성은 하부 데이터 구조인 핵심 컴포넌트의 재사용을 통하여 가능하다. 이러한 상호운용을 위하여 서로 다른 기업 간의 상호작용을 정의하는 비즈니스 프로세스에서의 핵심 컴포넌트의 역할과 적용 가능성에 대한 분석을 통한 핵심 컴포넌트의 명확한 개념 정립이 필요하다. 따라서 이 논문에서는 ebXML과 핵심 컴포넌트의 역할을 알아보고 등록/저장소에 등록하기 위한 핵심 컴포넌트를 생성하는 핵심 컴포넌트 저장 시스템을 설계 및 구현한다. 이 논문의 핵심 컴포넌트 저장을 위한 시스템은 등록/저장소에 등록하기 적합한 핵심 컴포넌트 구조를 생성하고, 비즈니스 프로세스의 모델링 단계에서 새로운 비즈니스 정보 개체의 생성을 위해서 이용되거나 재사용될 수 있다.

Design and Implementation of Generation System for Storing Core Components

Jeong Hee Hwang[†] · Young Gyun Kim[††] · Keun Ho Ryu[†††]

ABSTRACT

ebXML(Electronic Business using eXtensible Markup Language) is an XML based new international standard framework of the electronic business. Interoperability among mutually different industry fields can be realized by reusing core component, as a infrastructure of data. For this interoperability, it is needed to define the concrete concept of core component, through analyzing the role and applicability of the core component in business process which defines the interoperability among different companies. Therefore, in this paper, we inquire the role of the core component in the ebXML, and implement the core component storage system which generates core component to register in registry/repository. The designed system for storing the core component generates suitable core component structure to register in registry/repository, which can be used to generate new business information entity in a modeling step of business process.

키워드 : ebXML, 핵심 컴포넌트(Core Component), 비즈니스 프로세서(Business Process), 전자상거래(Electronic Commerce)

1. 서 론

기존의 가장 보편적인 기업간 전자상거래를 위한 프레임워크로 EDI(Electronic Data Interchange)가 사용되어 왔다. 그러나 거래 당사자간의 내부 시스템에서 사용되는 표준규약을 만들어야 하고 복잡한 인터페이스와 초기 투가 비용의 부담으로 EDI를 이용하는 것이 쉽지 않았다. 이러한 EDI는 인터넷과 웹(WEB) 기술을 이용한 WEB EDI와 인터넷 상의 자료 표현의 표준인 XML의 구조화된 데이터 표현 방식을 사용한 XML/EDI로 발전하였다. 이러한 EDI 외에도 또 다른 XML 기반 전자상거래 프레임워크들이 발표 및 사용되고 있다. 대표적인 예로는 전자부품 및 IT산업에 사용되는 RosettaNet, 여행업에 대한 OTA등을 들 수 있다. 이러한 대부분의 전자상거래 프레임워크들은 특정 산업분야를 염두에 두고 개발된 전자상거래 프레임워크 이거나 대부분 단순히 문서만을 표준화하는데 그쳤다[1].

전자상거래의 필요성은 갈수록 증가하고 있으며 이러한 다수의 전자상거래 프레임워크를 위한 표준화 노력이 필요하다. 이에 국제적인 표준으로 받아들여지고 있는 것이 ebXML (electronic business using eXtensible Markup Language)이다. ebXML은 모든 기업이 세계 어디에서 어느 누구와도 전자상거래를 할 수 있도록 하기 위한 국제기구인 UN/CEFACT와 OASIS가 공동 추진하고 있는 XML 및 인터넷 기반 개방형 전자상거래 표준 프레임워크이다. ebXML은 EDI나 xCBL 등 대부분의 기존 표준들이 단순히 문서만을 표준화하여 사용하는 것과는 대조적으로 비즈니스 프로세스를 모델링하여 시나리오를 작성함으로써, 기존의 B2B 전자상거래 시스템이 변화에 유연하게 대응하기 어려웠던 점을 극복할 수 있다. 그리고 특정 산업분야에 한정되었던 기존의 종적 전자상거래 프레임워크와 달리 여러 산업분야에 적용

※ 이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2003-002-D00280).
† 준 회 원 : 충북대학교 대학원 전자계산학과
†† 준 회 원 : 충북대학교 대학원 정보산업공학과
††† 종신회원 : 충북대학교 전기전자컴퓨터공학부 교수
논문접수 : 2003년 5월 30일, 심사완료 : 2003년 12월 9일

이 가능하다. 이렇게 서로 다른 산업분야 간에 상호운용을 가능하도록 하는 것이 ebXML의 두드러진 특징이며 장점이다. ebXML에서 그러한 상호운용을 위한 비즈니스 부분의 노력으로는 공통 비즈니스 프로세스와 핵심 컴포넌트의 정의와 사용이 있다. 이것은 단순히 문서를 재사용하는 단계를 넘어 시나리오 차원의 재사용과 하부 데이터 구조의 재사용이 가능하다[1]. ebXML은 이러한 재사용을 바탕으로 한 최상의 상호운용이 가능하다. 결국, ebXML을 업계에 적용할 경우 비용이 절감될 수 있으며, 효율성 증가 및 자동화, 상호운용성에 의한 전자상거래의 촉진 등의 효과가 예상된다.

그러나 아직까지는 ebXML의 실제 구현에 핵심 컴포넌트를 적용하기 위한 개념 정립이 미비한 실정이다. 현재까지 승인된 핵심 컴포넌트 기술 명세서가 존재하지 않는 것이 그 이유이기도 하다. 그렇지만, ebXML에서 하부 데이터 구조인 핵심 컴포넌트의 재사용을 통한 상호운용성은 매우 중요한 것이다. 상호운용을 위해서는 서로 다른 기업 간의 상호작용을 정의하는 비즈니스 프로세스에서의 핵심 컴포넌트의 역할과 적용 가능성에 대한 분석을 통한 핵심 컴포넌트의 명확한 개념 정립이 필요하다.

따라서 이 논문에서는 ebXML 핵심 컴포넌트의 정의와 이의 재사용이 상호운용에 미치는 영향에 대해 알아보고 핵심 컴포넌트가 저장소에 저장되기 위한 요구사항을 분석한다. 그리고 이를 바탕으로 핵심 컴포넌트를 등록/저장소에 등록하기 위해서 선행되어야 하는 핵심 컴포넌트 저장을 위한 시스템을 구현한다.

이 시스템은 새롭게 발견된 핵심 컴포넌트의 구조를 생성하며, 생성된 핵심 컴포넌트를 등록/저장소에 저장할 수 있다. 그리고 저장된 핵심 컴포넌트들은 이미 생성되거나 새롭게 생성될 비즈니스 메시지를 위해서 이용이 가능하다는 특징이 있다.

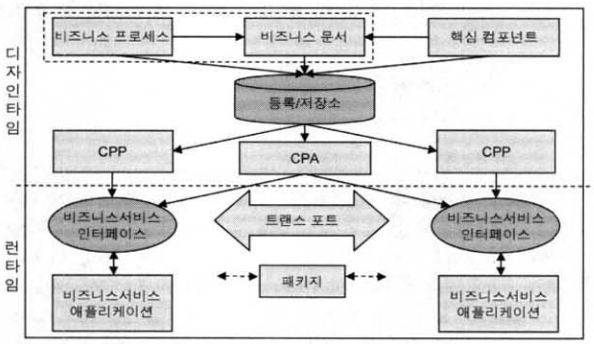
이 논문의 구성은 다음과 같다. 2장에서는 ebXML의 구성요소 및 주요 컴포넌트인 비즈니스 프로세스와 핵심 컴포넌트의 관계 및 특성을 기술한다. 3장에서는 핵심 컴포넌트의 정의 및 구성요소의 역할과 관계를 기술한다. 그리고 핵심 컴포넌트를 저장소에 저장하기 위한 요구사항을 설명한다. 4장에서는 3장의 핵심 컴포넌트 저장을 위한 요구사항을 바탕으로 핵심 컴포넌트 저장을 위한 시스템을 설계하고 5장에서는 시스템의 구현 및 수행 결과에 대하여 평가를 한다. 마지막으로 6장에서는 결론을 맺는다.

2. 관련 연구

이 장에서는 ebXML의 구성요소 및 주요 컴포넌트인 비즈니스 프로세스와 핵심 컴포넌트의 관계 및 특성과 역할에 대하여 기술한다.

2.1 ebXML의 구성요소

(그림 1)은 ebXML의 구성요소를 나타내며, 크게 디자인타임과 런타임으로 구분한다. 이것은 실제적인 비즈니스 발생의 이전과 이후를 구분하는 것이다. 디자인타임의 상단에 점선으로 표현된 부분이 비즈니스 프로세스 정의와 비즈니스 문서를 생성하는 비즈니스 프로세스와 정보 모델링 단계이다. 이때 비즈니스 문서를 표현하기 위한 하부의 데이터 구조로 핵심 컴포넌트가 이용된다. 이러한 핵심 컴포넌트는 상호운용과 재사용을 위해 비즈니스 프로세스와 비즈니스 문서와 함께 등록/저장소에 저장되어 진다. 그리고, 비즈니스 거래를 수행하기 위한 기업들은 실제 거래가 발생되기 이전에 등록/저장소에 비즈니스 프로파일인 협업 프로토콜 프로파일(Collaboration Protocol Profile, CPP)을 등록하여야 하고 이를 토대로 실제 거래를 수행하기 위한 협업 프로토콜 약정(Collaboration Protocol Agreement, CPA)을 도출한다. 이러한 디자인타임 부분의 수행을 통해 실제 거래를 위한 준비 단계가 완료된다.



(그림 1) ebXML 구성요소

2.2 비즈니스 프로세스와 핵심 컴포넌트

ebXML의 주요 컴포넌트 중에서 비즈니스 부분에 해당하는 것은 비즈니스 프로세스와 핵심 컴포넌트이다. 비즈니스 프로세스는 요구사항들을 수집하고 이를 분석하여 거래절차와 비즈니스 문서에 대하여 규정하게 된다. 그리고 이러한 비즈니스 거래에서 주고받는 비즈니스 문서를 구성하는 하부 데이터 구조로는 핵심 컴포넌트가 사용된다. 이와 같이 핵심 컴포넌트는 비즈니스 프로세스와 밀접한 연관성을 갖는다.

2.2.1 비즈니스 프로세스

ebXML 비즈니스 프로세스는 어떤 기업이 다른 거래 기업과의 상호작용을 편리하게 하기 위하여 공유된 역할(role), 관계(relationship), 의무 사항(responsibility)을 어떻게 수행할 것인가를 상세히 정의한 것이다. 각 비즈니스 거래는 전자 비즈니스 문서(Business Document)의 교환으로 표현된다. 비즈니스 문서의 교환 순서는 비즈니스 프로세스에 의하여 결정된다. 비즈니스 거래를 구현하기 위해 표준 패턴

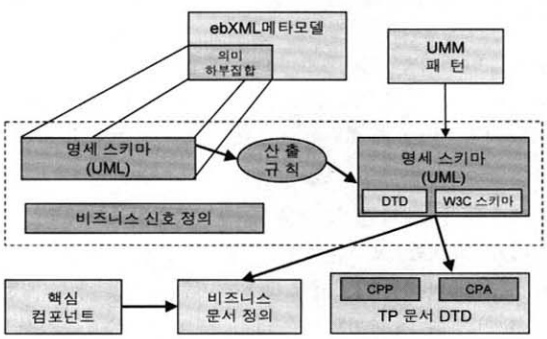
을 사용한다. 이러한 패턴은 거래 기업 간의 메시지와 비즈니스 신호(Business Signal)들을 교환하는 형태를 정의할 수 있다.

2.2.2 비즈니스 프로세스와 핵심 컴포넌트

비즈니스 거래란 두 기업 간의 거래에서 가장 작은 단위의 비즈니스를 지칭한다. 하나의 비즈니스 거래는 하나의 비즈니스 요청 행위(Requesting Business Activity)와 하나의 비즈니스 응답 행위(Responding Business Activity)로 구성된다.

비즈니스 프로세스 명세 스키마는 비즈니스 프로세스를 정의하는 명세를 작성할 수 있도록 사용 가능한 모델링 요소의 속성과 요소간의 관계에 관한 메타정보를 제공하기 위한 것이다. 하나의 프로세스 정의에는 비즈니스 문서(Business Document), 비즈니스 거래(Business Transaction), 비즈니스 협업(Business Collaboration) 등이 포함된다.

(그림 2)는 명세 스키마와 거래 파트너(TP) 및 핵심 컴포넌트 간의 관계를 보여준다. ebXML 명세 스키마는 UML 명세 스키마, DTD 명세 스키마, 생성 규칙, 비즈니스 신호 정의, UMM 패턴 등 5가지의 기능적인 요소로 구성되어 있음을 알 수 있다. 점선 박스로 표시된 영역이 비즈니스 프로세스 명세에 관한 부분이다[2]. 이들 중에서 DTD 명세 스키마와 UMM 패턴 그리고 비즈니스 신호 정의의 세 가지는 동반되는 비즈니스 문서들과 함께 거래 실행을 위한 CPP, CPA 문서에 첨부된다. 이러한 내용과 함께 기술적 변수들이 설정된 CPP, CPA는 각각의 거래 기업에서 실행되는 소프트웨어에 대한 완전한 설정을 가능하게 한다.



(그림 2) 비즈니스 프로세스 명세 스키마

비즈니스 프로세스 명세는 ebXML 비즈니스 서비스 인터페이스(Business Service Interface)를 위해 필요한 기계 해독 가능한 런타임 비즈니스 프로세스 명세이다. 그러므로 비즈니스 프로세스 명세는 ebXML 거래 파트너 협업 프로토콜 프로파일(CPP) 및 협업 프로토콜 약정(CPA)에 의해 참조되거나 연동된다.

또한 의미적으로 중립적인 데이터 요소인 핵심 컴포넌트가 비즈니스 문서 정의와 같은 비즈니스 메시지의 작성에 이용되기 위해서는 비즈니스 프로세스에 의해 제공되는 컨텍스트와 결합되어야 한다. 이러한 핵심 컴포넌트는 새로운 비즈니스 프로세스의 모델링 단계에서 새롭게 생성하거나 재 사용될 수 있다. 서로 다른 산업 분야 간에도 전자상거래가 가능한 상호운용성을 최대화 할 수 있는 것은 재사용을 통해서이다. 이러한 재사용성이 UMM, 비즈니스 프로세스와 핵심 컴포넌트의 중요한 아이디어이기도 하다. 따라서 재사용성을 위하여 일반적인 정보와 특정 컨텍스트를 분리하는 것이다.

핵심 컴포넌트를 이용한 비즈니스 문서의 정의는 표준화가 늦어져 실제 적용에 어려움이 있다[3]. 현재는 기업 간 합의에 의해 미리 정의되어 있는 xCBL이나 RosettaNet 등의 문서를 이용하고 있고, 최근에 ATG(Applied Technologies Group)에서 핵심 컴포넌트에서 정의하고 있지 않았던 비즈니스 메시지 규칙을 새롭게 시작하고 있다. ATG의 비즈니스 메시지 작업은 OASIS UBL 메커니즘이 ebXML의 핵심 컴포넌트에 영향을 미칠 것으로 예상된다[4].

3. 핵심 컴포넌트

이 장에서는 핵심 컴포넌트의 정의 및 구성요소의 역할과 관계를 기술한다. 그리고 비즈니스 프로세스 모델링 단계에서 핵심 컴포넌트의 활용 단계에 대해 기술하고, 발견된 핵심 컴포넌트를 저장소에 저장하기 위한 구성요소의 관계에 대하여 설명한다.

3.1 핵심 컴포넌트 정의

ebXML은 서로 다른 산업분야 간에도 상호운용을 가능하게 하기 위해서 비즈니스 부분에서는 시나리오 차원의 재활용과 핵심 컴포넌트의 재활용을 통해 상호운용성을 보장한다. 하부 데이터 구조인 핵심 컴포넌트의 비즈니스 프로세스 모델링에서 비즈니스 메시지를 구성하기 위해서 재활용되거나 새로운 핵심 컴포넌트로의 변환을 위해 이용될 수 있다. 즉 핵심컴포넌트는 의미적으로 정확하고 의미 있는 정보 교환 단위의 생성을 위한 빌딩 블록이므로 특정한 개념을 설명하기 위해 필요한 정보만을 포함한다. 이러한 핵심 컴포넌트의 역할은 서로 다른 산업분야 간의 전자상거래에서도 계속적으로 자신들의 언어를 사용하는 것을 가능하게 한다[5].

3.1.1 핵심 컴포넌트

비즈니스 프로세스와 정보 모델링은 각 단계에서 교환되는 각 비즈니스 정보를 구분한다. 그러나, 상호운용성을 위해서는 같은 비즈니스 의미가 다른 비즈니스 프로세스에서 사용될지라도 같은 정보 구조를 이끌 수 있어야만 한다. ebXML에서 이러한 상호운용성에 대한 문제 해결은 재사용 가능한 빌딩 블록인 핵심 컴포넌트(Core Component)와

그것이 사용되고 있는 환경에 대한 설명인 컨텍스트(Context) 그리고 핵심 컴포넌트와 컨텍스트의 결합에 의한 비즈니스 정보 개체(Business Information Entity)와 같은 핵심 컴포넌트 구조에 의해 이루어진다.

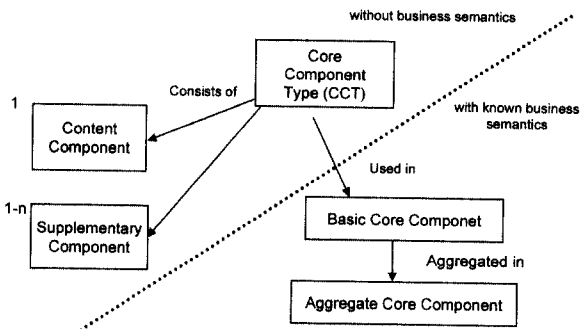
핵심 컴포넌트는 다양한 비즈니스 상황과 영역에서 나타난다는 사실에 의해 특성화된 것으로써 여러 가지 다른 비즈니스 상황과 영역에서 나타나는 공통적인 또는 일반적인 빌딩 블록이므로 기본적으로 여러 비즈니스 분야에 걸쳐 사용될 수 있다. 따라서 핵심 컴포넌트는 컨텍스트에 구애받지 않는 자유로운 빌딩 블록이다. 또한 자연스럽게 조화되는 핵심 컴포넌트는 집합 핵심 컴포넌트로 그룹화 될 수도 있다. 마찬가지로 이러한 집합 핵심 컴포넌트는 여러 비즈니스 분야에서 사용될 수 있다. 비즈니스 프로세스에서 사용되는 핵심 컴포넌트는 컨텍스트에 맞게 변화되어야 한다.

핵심 컴포넌트 기술 명세서에서 중요한 개념 세 가지는 핵심 컴포넌트(Core Component), 비즈니스 컨텍스트(Business Context), 비즈니스 정보 개체(Business Information Entity)이다. 핵심 컴포넌트는 모든 e-비즈니스 메시지를 구성하기 위해 사용될 수 있는 의미적 빌딩 블록이다. 이러한 핵심 컴포넌트는 세 가지, 즉 기본 핵심 컴포넌트(Basic Core Component), 집합 핵심 컴포넌트(Aggregate Core Component), 핵심 컴포넌트 유형(Core Component Type)으로 세분화되어 구성된다[6]. 그 정의를 살펴보면 다음과 같다.

- 기본 핵심 컴포넌트(BCC) : 핵심 컴포넌트는 유일한 비즈니스 의미적 정의를 갖는 단일 비즈니스 개념을 나타내는 것이다. 기본 핵심 컴포넌트는 핵심 컴포넌트 유형을 사용함으로써 구성되며, 집합 핵심 컴포넌트를 개발하기 위해 사용된다.
- 집합 핵심 컴포넌트(ACC) : 별개의 비즈니스 의미를 나타내기 위한 핵심 컴포넌트들의 모임이다. 집합 핵심 컴포넌트는 두 개 이상의 기본 핵심 컴포넌트로 구성되거나, 또는 적어도 한 개 이상의 기본 핵심 컴포넌트에 한 개 이상의 집합 핵심 컴포넌트들로 구성된다.
- 핵심 컴포넌트 유형(CCT) : 핵심 컴포넌트는 단 하나의 내용 핵심 컴포넌트(Content Component)와 본질적인 추가 정의를 갖는 하나 이상의 보충 컴포넌트(Supplement Component)로 구성된다. 핵심 컴포넌트 유형(CCT)은 비즈니스 의미를 갖지 않는 핵심 컴포넌트로써 기본 핵심 컴포넌트를 구성하기 위해 필요하다.

(그림 3)은 핵심 컴포넌트의 세 가지 분류인 기본 핵심 컴포넌트와 집합 핵심 컴포넌트 그리고 핵심 컴포넌트들 간의 관계를 나타내고 있다. 점선을 기준으로 좌측 상단은 핵심 컴포넌트 유형(CCT)를 보여주고 있다. 우측 하단의 기본 핵심 컴포넌트를 구성하기 위해서 핵심 컴포넌트 유형이 사용된다. 이러한 기본 핵심 컴포넌트는 집합 핵심 컴포넌트를 구성하는데 사용될 수 있다. 우측 하단의 기본 핵심 컴포넌트와 집합 핵심 컴포넌트는 핵심 컴포넌트 유형과는 달리 알려진 비즈니스 의미를 가지고 있다.

핵심 컴포넌트가 실제 비즈니스 상황에서 사용될 때 그것은 비즈니스 정보 개체로 정의되며 이것은 핵심 컴포넌트가 특정 비즈니스 환경(business context)에서 사용된 결과라 할 수 있다. 비즈니스 정보 개체는 기본 비즈니스 정보개체와 집합 비즈니스 정보 개체로 나누어진다. 기본 비즈니스 정보 개체는 기본 핵심 컴포넌트에 특정한 비즈니스 컨텍스트가 적용된 것이고, 집합 비즈니스 정보 개체는 기본 비즈니스 정보 개체들로 구성되며 집합 핵심 컴포넌트에 비즈니스 컨텍스트가 적용된 것이다.



(그림 3) 핵심 컴포넌트들의 관계

3.2 핵심 컴포넌트 활용

이 절에서는 비즈니스 프로세스 모델링 단계에서 핵심 컴포넌트의 이용에 대하여 기술한다. 특히, 핵심 컴포넌트의 발견 단계와 재사용에 관한 것이 중심이 된다.

3.2.1 비즈니스 프로세스 정의 단계

비즈니스 프로세스의 분석을 통해 요구사항의 서술, 비즈니스 협업을 정의할 수 있게 된다. 즉, 각 프로세스 단계의 타이밍과 목적의 서술 및 정의한다. 이 레벨에서 비즈니스 프로세스들의 자세한 조사를 통해서 사용되어지는 개개의 비즈니스 정보와 그것들을 주고받는 단계를 밝힌다. 그 결과의 하나로 비즈니스 프로세스를 위한 비즈니스 정보 개체를 확인할 수 있으며, 비즈니스 정보 개체는 특정한 비즈니스 컨텍스트에서 사용되는 핵심 컴포넌트이고 유일한 이름이 주어진다. 기본 핵심 컴포넌트는 특정한 비즈니스 컨텍스트에서 사용될 수 있고, 그 구조는 바뀌지 않는다. 이를 토대로 UN/EDIFACT MIG, XML schema(XML Schema, DTD) 또는 유사한 구문으로 표현된 표준 메시지 구조를 정의하게 된다. 핵심 컴포넌트 개념에서, 핵심 컴포넌트의 정의와 저장 그리고 연관된 컨텍스트 메커니즘은 MIG나 XML schema를 생성하기 전에 발생한다. 이 모든 것이 비즈니스 프로세스 모델 정의 단계 동안 자연스럽게 발생된다[6].

3.2.2 핵심 컴포넌트 발견

핵심 컴포넌트 발견에서의 과정은 준비와 검색이다. 실제 핵심 컴포넌트 라이브러리를 정의하기 위해서는 규정된 준비와 검색 단계를 따라야 한다. 여기서는 많은 비즈니스 정보 개체들이 발견되었다고 가정하고 컴포넌트의 발견 단계와 관련된 검색 단계를 설명한다. 이 과정을 통해 비즈니스 정보 개체 그리고 핵심 컴포넌트와 같은 핵심 컴포넌트 라이브러리들이 재사용 및 변경될 수 있다.

단계 1 : 같은 정의를 갖는 집합 비즈니스 정보 개체가 존재할 수 있으므로 집합 비즈니스 정보 개체의 카탈로그를 검색한다.

- 비즈니스 요구에 적합한 정의를 갖는 집합 비즈니스 정보 개체가 있다면, 재사용을 등록한다.
- 비즈니스 요구에 적합하게 수정될 가능성이 있는 집합 비즈니스 정보 개체가 있다면, 집합 비즈니스 정보 개체를 적합하게 변경한다.
- 적합한 정의를 갖는 비즈니스 정보 개체가 없다면, 단계 2로 간다.

단계 2 : 새롭게 요구되는 집합 비즈니스 정보 개체를 형성하기에 적합한 일반적인 정의와 구조를 갖는 이미 존재하는 집합 핵심 컴포넌트를 위해 핵심 컴포넌트 카탈로그를 검색한다[6].

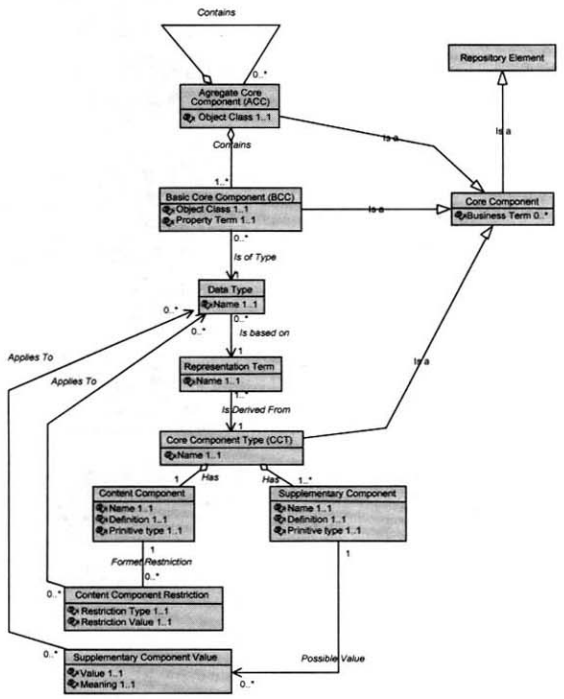
- 비즈니스 요구에 적합한 정의와 구조를 갖는 집합 핵심 컴포넌트가 존재한다면, 집합 비즈니스 정보 개체를 위해서 집합 핵심 컴포넌트의 재사용을 등록한다.
- 집합 핵심 컴포넌트가 비즈니스 요구에 적합하게 수정될 가능성이 있는 정의와 구조를 갖는다면, 집합 핵심 컴포넌트를 적합하게 변경한다.
- 적합한 정의와 구조를 갖는 집합 핵심 컴포넌트가 존재하지 않는다면, 새로운 집합 핵심 컴포넌트 요청을 준비한다.

3.3 핵심 컴포넌트 저장

저장소에 핵심 컴포넌트를 저장하기 위해서는 개발자와 사용자 모두에 의해 반드시 다루어져야 하는 요구사항들을 포함하고 있어야 한다. 즉, 개발자와 사용자가 사용하고 있는 등록저장소와 프로세스 모델링, 그리고 거래의 상호교환적 결합을 위해 핵심 컴포넌트를 사용하는 프레임워크가 만족할 수 있는 요구사항을 포함해야 한다.

(그림 4)는 [6]의 핵심 컴포넌트 기술 명세 V1.8에 기반한 핵심 컴포넌트의 유형과 저장을 위한 요구사항의 관계를 설명하고 있는 UML 모델이며 각 클래스에 대한 설명은 다음과 같다.

저장되는 핵심 컴포넌트는 항상 기본 핵심 컴포넌트, 집합 핵심 컴포넌트, 핵심 컴포넌트 유형 중에 하나로써 정의되어야 한다. 그리고, 저장되는 핵심 컴포넌트는 Business Term(optional, repetitive) 속성을 포함해야 한다. 이는 비즈니스에서 공통적으로 알려지고 사용되어지는 용어이다.



(그림 4) 핵심 컴포넌트의 유형과 저장을 위한 UML 모델

기본 핵심 컴포넌트는 항상 객체를 표현하는 Object class, 비즈니스 항목의 특징을 나타내는 Property Term, Data Type에 기반 한다. 핵심 컴포넌트 유형은 기본형을 정의하는 내용 컴포넌트와 내용 컴포넌트에 의미를 주는 하나 이상의 보충 컴포넌트를 포함한다. 그리고, 의미 있는 유형 이름인 Name 속성을 포함한다.

집합 핵심 컴포넌트는 두 개 이상의 기본 핵심 컴포넌트나, 적어도 하나의 기본 핵심 컴포넌트와 하나 이상의 집합 핵심 컴포넌트로 구성된다. 저장되는 집합 핵심 컴포넌트는 Object Class 속성을 포함한다. 데이터 유형은 특정 기본 핵심 컴포넌트를 위해 사용될 수 있는 유효 값의 완전한 범위를 정의하며 Name 속성을 포함한다. 데이터 유형은 핵심 컴포넌트 유형에서 파생된 표시어에 기반한다.

4. 핵심 컴포넌트 저장 시스템 설계

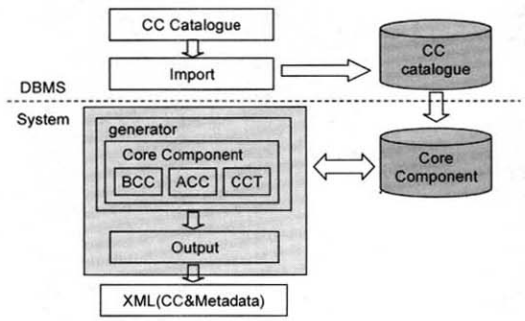
상호운영이 가능한 비즈니스 메시지의 구성을 가능하게 하는 핵심 컴포넌트는 등록/저장소에 등록 및 저장되어야 한다. 이는 상호운영의 최상의 방법인 재사용을 통한 상호운영을 위해서도 반드시 필요한 과정이다. 이 장에서는 비즈니스 프로세스 모델링 단계에서 새롭게 발견되는 핵심 컴

포넌트를 등록/저장소에 등록하기 위한 시스템을 설계한다.

4.1 시스템 구조

시스템의 구현은 Windows 환경에서 Java 1.3을 이용하여 구현하였다. 시스템에서 데이터베이스와의 연결을 위해서 JDBC Driver Type 1에 해당하는 JDBC-ODBC Bridge를 사용하였고 Graphic User Interface를 위해서는 Java Swing을 사용하였다. 그리고 작업 데이터베이스를 위한 DBMS로는 MS-Access를 이용하였다.

(그림 5)의 핵심 컴포넌트 저장을 위한 시스템의 구성도를 살펴보면 크게 두 부분으로 나누어진다. 그것은 상단의 DBMS 차원의 임포트(import) 단계와 임포트를 통해 생성된 데이터베이스를 처리하는 시스템의 수행 단계이다.



(그림 5) 핵심 컴포넌트 저장을 위한 시스템 구성도

먼저, DBMS 차원의 임포트 단계는 시스템에서 처리하기 위한 데이터베이스 생성을 위해서 선행되어야 하는 단계이다. DBMS 차원에서 임포트를 위해 사용하는 입력은 핵심 컴포넌트 카탈로그이다. 이것은 표준화 그룹에 의해 발견 및 승인된 핵심 컴포넌트를 자세히 표현하기 위해서 핵심 컴포넌트 카탈로그 형태로 작성될 것이기에 이러한 방식을 선택하였다. 이러한 핵심 컴포넌트 카탈로그로는 ebXML 1차 개발 단계의 결과물로써 핵심 컴포넌트의 구조를 보여주는 (그림 9)의 핵심 컴포넌트 구조 v1.04가 사용되었다. 이렇게 핵심 컴포넌트 카탈로그의 임포트를 통해 시스템 수행을 위한 데이터베이스 생성이 이루어진다.

시스템은 임포트 단계를 통해 생성된 데이터베이스를 입력으로 하여 등록/저장소에 저장하기 위해 필요한 핵심 컴포넌트의 정보를 데이터베이스에 저장한다. 이렇게 저장된 핵심 컴포넌트는 실질적으로 등록/저장소에 저장하기 위한 XML 형태의 문서를 생성하는 것을 가능하게 한다. 앞의 3.3절에서 설명된 핵심 컴포넌트의 저장 요구사항과 세분화된 컴포넌트들의 관계 분석을 토대로 핵심 컴포넌트 저장을 위해 필요한 정보를 결정하였고 이를 저장하기 위한 XML 문서의 스키마를 결정하였다. 저장되는 핵심 컴포넌트를 세분화하면 집합 핵심 컴포넌트, 기본 핵심 컴포넌트, 핵심 컴포넌트 유형으로 나뉜다. 또한 이 시스템은 입력으로 사용된 핵심 컴포넌트 카탈로그 외에도 새로운 핵심 컴포넌트의 저장과 XML 문서의 생성이 가능하고 저장된 정보들의 수정이 가능하다.

4.2 저장을 위한 스키마

등록/저장소에 저장할 핵심 컴포넌트에 대한 XML 구조를 결정하기 위해서 핵심 컴포넌트의 세분화된 컴포넌트들의 관계 분석에 기반하여 다음과 같은 XML 스키마를 생성하였다.

```
<!DOCTYPE CoreComponents [
<!ELEMENT CoreComponents (RepositoryElement, CoreComponent)>
<!ELEMENTRepositoryElement (UID, Version, DictionaryEntryName,
    Definition, UsageRule?)>
<!ELEMENT UID (#PCDATA)>
<!ELEMENT Version (#PCDATA)>
<!ELEMENT DictionaryEntryName (#PCDATA)>
<!ELEMENT Definition (#PCDATA)>
<!ELEMENT UsageRule (#PCDATA)>
<!ELEMENT CoreComponent
    (BusinessTerm?, AggregateCoreComponent)>
<!ELEMENT BusinessTerm (#PCDATA)>
<!ELEMENT AggregateCoreComponent
    (ObjectClass, CoreComponentChildren)>
<!ELEMENT ObjectClass (#PCDATA)>
<!ELEMENT CoreComponentChildren (Child)+>
<!ELEMENT Child (#PCDATA)>
]>
```

(그림 6) 집합 핵심 컴포넌트를 위한 스키마

```
<!DOCTYPE CoreComponents [
<!ELEMENT CoreComponents
    (RepositoryElement,CoreComponent)>
<!ELEMENTRepositoryElement
    (UID, Version, DictionaryEntryName, Definition, UsageRule?)>
<!ELEMENT UID (#PCDATA)>
<!ELEMENT Version (#PCDATA)>
<!ELEMENT DictionaryEntryName (#PCDATA)>
<!ELEMENT Definition (#PCDATA)>
<!ELEMENT UsageRule (#PCDATA)>
<!ELEMENT CoreComponent
    (BusinessTerm?, BasicCoreComponent)>
<!ELEMENT BusinessTerm (#PCDATA)>
<!ELEMENT BasicCoreComponent
    (ObjectClass, PropertyTerm, DataType, Representa tionTerm,
     CoreComponentType)>
<!ELEMENT ObjectClass (#PCDATA)>
<!ELEMENT PropertyTerm (#PCDATA)>
<!ELEMENT DataType (Name)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT RepresentationTerm (Name)>
<!ELEMENT CoreComponentType (Name)>
]>
```

(그림 7) 기본 핵심 컴포넌트를 위한 스키마

(그림 6)은 집합 핵심 컴포넌트의 XML 문서 생성을 위한 스키마를 DTD(Document Type Definition)로 작성한 것이다. RepositoryElement 엘리먼트가 수용하는 UID, Version, DictionnaryEntryName, Definition 엘리먼트는 필수적이고

UsageRule 엘리먼트는 선택적이다. 또한 모든 핵심 컴포넌트는 BusinessTerm 엘리먼트를 수용한다. 집합 핵심 컴포넌트는 기본 핵심 컴포넌트와 마찬가지로 ObjectClass 엘리먼트를 수용하고, 두 개 이상의 핵심 컴포넌트(기본/집합 핵심 컴포넌트)를 자식으로 갖는 CoreComponentChildren 엘리먼트를 수용한다.

(그림 7)은 기본 핵심 컴포넌트의 XML 문서 생성을 위한 스키마로 집합 핵심 컴포넌트와 동일한 RepositoryElement 엘리먼트와 ObjectClass 엘리먼트를 수용한다. 그리고 PropertyTerm 엘리먼트와 DataType, Representation, CoreComponentType의 Name 엘리먼트를 수용한다.

(그림 8)은 핵심 컴포넌트 유형을 위한 스키마로 RepositoryElement는 앞선 핵심 컴포넌트들과 동일하다. 핵심 컴포넌트 유형의 Name 엘리먼트와 핵심 컴포넌트 유형에 대한 하나의 내용 컴포넌트와 하나 이상의 보충 컴포넌트에 관한 Name, Definition, PrimitiveType 엘리먼트를 수용한다.

```
<!DOCTYPE CoreComponents [
<!ELEMENT CoreComponents
      (RepositoryElement,CoreComponent)>
<!ELEMENT RepositoryElement
      (UID,Version,DictionaryEntryName,Definition,UsageRule?)>
<!ELEMENT UID (#PCDATA)>
<!ELEMENT Version (#PCDATA)>
<!ELEMENT DictionaryEntryName (#PCDATA)>
<!ELEMENT Definition (#PCDATA)>
<!ELEMENT UsageRule (#PCDATA)>
<!ELEMENT CoreComponent
      (BusinessTerm?,BasicCoreComponent)>
<!ELEMENT BusinessTerm (#PCDATA)>
<!ELEMENT CoreComponentType (Name, Components)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Components
      (ContentComponent,SupplementaryComponent)>
<!ELEMENT ContentComponent
      (Name,Definition,PrimitiveType)>
<!ELEMENT PrimitiveType (#PCDATA)>
<!ELEMENT SupplementaryComponent
      (Name,Definition,PrimitiveType)>
]>
```

(그림 8) 핵심 컴포넌트 유형을 위한 스키마

5. 구현 및 평가

5.1 시스템 구현을 위한 단계

시스템의 구현은 생성된 스키마를 바탕으로 등록/저장소에 저장하기 적합한 구조의 핵심 컴포넌트를 생성하기 위한 작업이다. 먼저, 입력으로 사용하기 위한 핵심 컴포넌트는 핵심 컴포넌트 구조 v1.04를 임포트 하여 이용하였다. 이렇게 입력된 핵심 컴포넌트에 대하여 등록/저장소에 저장하기 적합한 XML 구조를 생성하기 위한 테이블 구조를 정의하고, 이러한 핵심 컴포넌트와 테이블 구조를 바탕으로 Java Swing을 사용하여 응용프로그램을 구현 및 수행하였다.

5.1.1 임포트

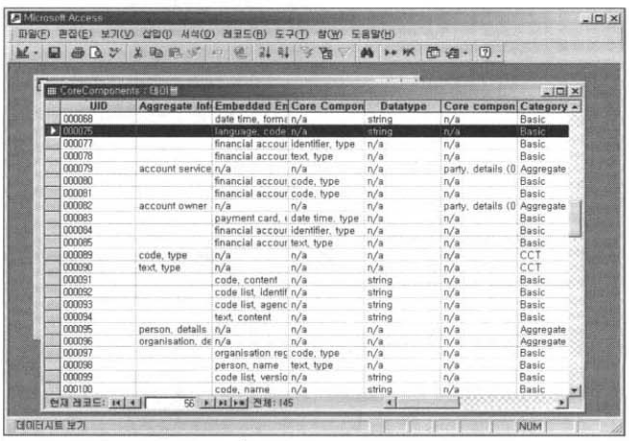
임포트 하는 과정은 입력으로 사용된 핵심 컴포넌트 구조를 이용하여 DBMS에서 핵심 컴포넌트를 데이터베이스로 임포트 하는 것이다. 이렇게 임포트된 핵심 컴포넌트들은 시스템에서 저장 스키마에 적합한 XML 구조를 생성하기 위해 사용되어진다.

(그림 9)는 엑셀 파일로 작성되어 있는 핵심 컴포넌트 구조를 나타내고 있다. 이러한 핵심 컴포넌트 구조를 MS-Access의 스프레드시트 가져오기 마법사를 이용하여 임포트 하게 된다.

이러한 임포트 과정의 결과는 (그림 10)과 같이 나타난다. 데이터베이스의 테이블 구조는 핵심 컴포넌트 구조의 정보와 동일하다. 데이터베이스의 내용은 시스템에서 등록/저장소에 저장을 위한 구조로 변환되기 위해서 처리된다.

UID	Aggregate Information Entity	Embedded Entity	Core Component Type (CCT) used	Datatype	Core component re-used	Category Type (CCT, Basic, Aggregate)	Required (R)	Definition	Remarks
SECTION 1: Core Component Types									
000066	date time. type	n/a	n/a	n/a	n/a	CCT		A particular point in the progression of time together with relevant supplementary information.	Can be used for a date and/or time
000067		date time. content	n/a	string	n/a	Basic	R	The particular point in the progression of time.	
000068		date time. format. text	n/a	string	n/a	Basic		The format of the date/time content.	Reference ISO 8601

(그림 9) 핵심 컴포넌트 구조 v1.04



(그림 10) 임포트된 데이터베이스 테이블의 구조

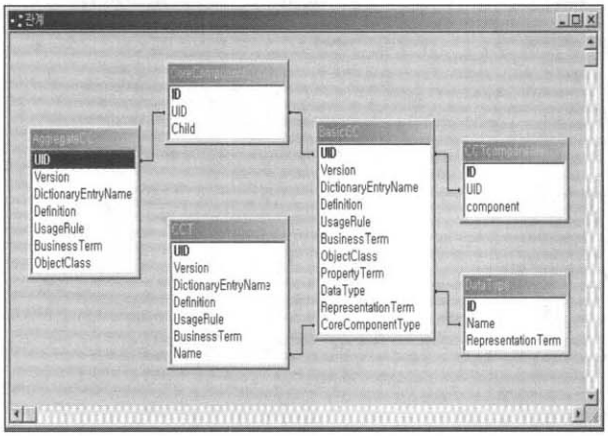
이렇게 임포트된 데이터베이스에 대한 JDBC-ODBC Bridge Driver를 통한 시스템으로의 연결을 위해서 ODBC 데이터 원본 관리자의 시스템 DSN(Data Source Name)에 Project라는 이름으로 임포트된 데이터베이스를 추가하였다.

5.1.2 핵심 컴포넌트 저장 및 XML 문서 생성

임포트된 각각의 핵심 컴포넌트에 대한 XML 문서 생성을 위해서 필요한 새로운 저장 구조는 다음과 같다.

(그림 11)은 데이터베이스에서 각각의 테이블간의 관계를

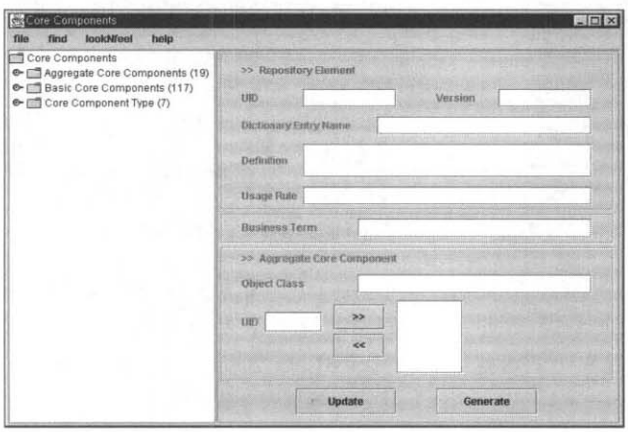
보여주고 있다. 이러한 형태로 저장된 핵심 컴포넌트 정보들은 시스템의 수행 시에 수정 및 저장이 가능하고 이를 토대로 스키마 형식에 맞게 XML 문서를 생성하게 된다.



(그림 11) 테이블 관계도

5.2 시스템 수행 결과

구현된 시스템의 수행결과는 다음과 같다. 먼저 (그림 12)는 시스템을 수행했을 시의 초기 화면이다. 좌측의 트리 구조에 표현된 것은 임포트된 데이터베이스의 핵심 컴포넌트들의 그 종류인 집합 핵심 컴포넌트, 기본 핵심 컴포넌트 그리고 핵심 컴포넌트 유형으로 분류하여 UID에 따라 정렬된 트리 구조 형태로 보여주고 있다. 우측의 패널은 집합 핵심 컴포넌트, 기본 핵심 컴포넌트 그리고 핵심 컴포넌트 유형 각각에 대한 편집과 저장이 가능하다. 그리고 편집 및 저장된 핵심 컴포넌트 정보를 그 스키마에 맞게 XML 문서로의 변환이 가능하다. 초기화면에서는 새로운 집합 핵심 컴포넌트의 입력이 가능하게 되어있다.

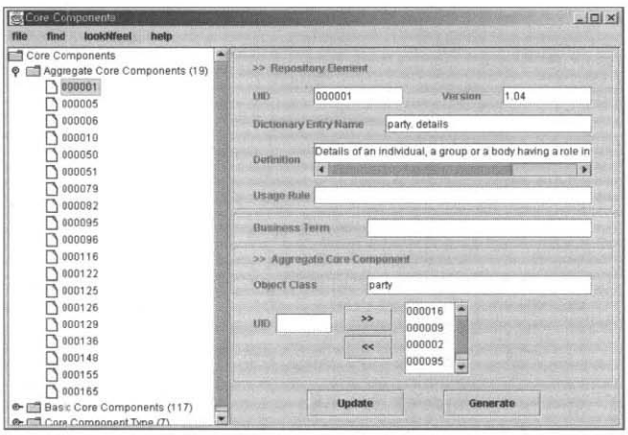


(그림 12) 초기화면

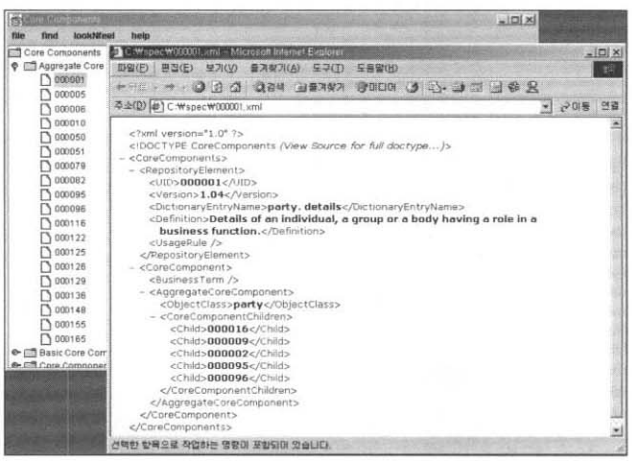
(그림 13)은 좌측의 트리 구조에서 집합 핵심 컴포넌트에 속하는 핵심 컴포넌트의 UID를 선택하면 우측에 그에 대한 정보들이 나타나는 것을 보여준다. 이와 동일한 표현방식으로 기본 핵심 컴포넌트에 속하는 핵심 컴포넌트의 UID에 대한 정보와 핵심 컴포넌트 유형에 대한 정보를 보여준다.

(그림 14)는 선택된 집합 핵심 컴포넌트를 이용하여 등록/저장소에 저장하기 위한 XML 문서를 생성하기 위해서 generate 버튼을 선택한 결과를 보여주고 있다. 버튼이 선택되면 해당 집합 핵심 컴포넌트의 정보를 이용하여 스키마에 적합한 XML 문서를 생성하고 해당 XML 문서를 Internet Explorer로 나타낸다. 기본 핵심 컴포넌트와 핵심 컴포넌트 유형의 경우에도 이와 동일한 작업을 수행할 수 있다.

이렇게 생성된 XML 구조의 핵심 컴포넌트들은 문서를 구성하기 위해 사용되어지고 또한 새로운 비즈니스 프로세스 모델링 과정에서 재사용을 위해서 이용될 수 있다.



(그림 13) 집합 핵심 컴포넌트 화면



(그림 14) 집합 핵심 컴포넌트에 대한 XML 문서 생성 화면

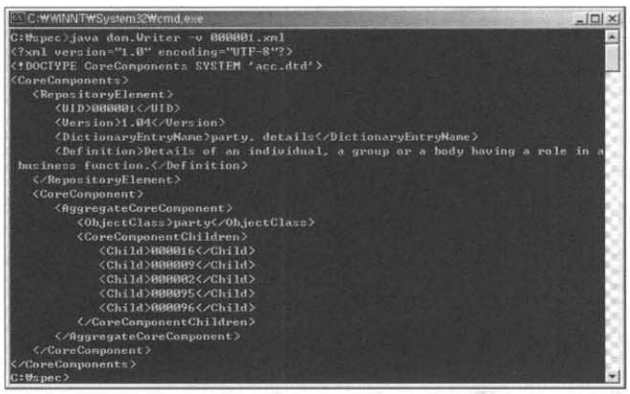
5.3 평 가

구현 결과의 평가는 시스템 수행 결과로 생성된 XML 문서가 유효(valid)한가를 확인한다. 만약 생성된 핵심 컴포넌트에 관한 XML 문서가 스키마인 문서형식정의(DTD)에 부합되고 동의된다면, 생성된 XML 문서는 유효하게 되는

것이다. 이러한 유효성 검사는 xml4j-4_0_5을 이용하였다. xml4j-4_0_5은 IBM에서 개발한 Java를 위한 XML 파서이다. 파서에 포함된 샘플 프로그램을 이용하여 유효성 검사를 수행하게 된다. 샘플 프로그램인 dom.Writer은 파싱된 문서를 출력한다. 이때 -v 옵션으로 유효성을 검사하여 출력한다[8].

수행 결과로 생성되는 집합 핵심 컴포넌트, 기본 핵심 컴포넌트, 핵심 컴포넌트 유형의 XML 문서에 대해서 유효성을 검사하게 된다. 이에 대한 각각의 파일명은 000001.xml, 000002.xml, 000066.xml 이고, 파일명은 해당 UID로 지정하였다.

(그림 15)는 집합 핵심 컴포넌트인 000001.xml 문서에 대한 유효성 검사의 결과이며 오류가 없이 파싱된 문서를 보여주고 있다. 이와 동일한 방식으로 000002.xml, 000066.xml 문서에 대한 기본 핵심 컴포넌트와 핵심 컴포넌트 유형에 대한 유효성 검사를 수행하였고 이러한 유효성 검사를 통해 앞의 4.2절에서 정의한 스키마인 문서형식정의(DTD)에 부합되는 유효한 XML 문서를 생성하는 것을 확인할 수 있었다. 이렇게 유효한 핵심 컴포넌트들은 등록/저장소에 저장된다. 저장된 핵심 컴포넌트들은 비즈니스 프로세스의 모델링 단계에서 새로운 비즈니스 정보 개체의 생성을 위해서 이용되거나 재사용 되어 ebXML에서의 데이터 구조의 재활용을 통한 상호운용을 가능하다.



(그림 15) 집합 핵심 컴포넌트(000001.xml) 유효성 검사

6. 결 론

ebXML은 단순히 문서를 재사용하는 단계를 넘어 시나리오 차원의 재사용과 하부 데이터 구조의 재사용이 가능하기 때문에 주목받고 있다. ebXML에서 재사용을 통한 서로 다른 산업분야 간의 상호운용성을 가능하게 하는 하부 구조에는 핵심 컴포넌트가 존재한다. 이러한 핵심 컴포넌트들의 발견을 통해 핵심 라이브러리를 구성하게 되고 이 핵심 라이브러리는 상호운용을 위한 하부 구조에서 이용된다. 핵심 라이브러리를 구성하기 위해서 비즈니스 프로세스 모델링 단계에서 핵심 컴포넌트가 발견되어 요구사항에 맞게 등록/저장소에 등록되어야 한다.

따라서 이 논문에서는 핵심 라이브러리의 재사용을 통한 상호운용을 위해서 반드시 선행되어야 하는 새로운 핵심 컴포넌트의 발견과 등록/저장소에 등록하기 위한 요구사항과 핵심 컴포넌트의 역할을 검토하였고, 등록/저장소에 등록하기 위하여 핵심 컴포넌트를 생성하는 핵심 컴포넌트 저장 시스템을 설계 및 구현하였다.

그리고 시스템 구현을 위해 핵심 컴포넌트의 세부 컴포넌트들의 관계 분석을 기반으로 등록/저장소에 저장하기 적합한 스키마를 생성하였다. 또한 이러한 저장 스키마를 이용하여 핵심 컴포넌트의 저장 시스템을 설계하였고 핵심 컴포넌트의 요구사항에 맞게 저장 및 수정하고 XML 구조로 변환할 수 있는 시스템을 구현하였다. 마지막으로 스키마에 대한 유효성 검사를 통해 시스템의 수행 결과를 평가하였다.

이 논문에서 구현한 시스템은 핵심 컴포넌트 구조를 생성하고 이것은 등록/저장소에 저장이 가능하므로 이미 생성되거나 새롭게 생성될 비즈니스 메시지를 위해 이용될 수 있다.

향후 연구로는 핵심 라이브러리를 구성하고 이를 바탕으로 비즈니스 문서의 편집과 자동화된 문서의 구성을 위한 시스템 개발을 목표로 하고 있다.

참 고 문 헌

[1] 한국전자거래진흥원, "2002년 ebXML 백서", 산업자원부 한국전자거래진흥원, 2002.

[2] ebXML Business Process Project Team, Business Process Specification Schema v1.01, http://www.ebxml.org/specs/ebBPSS.pdf, 2001.

[3] 임영철, "ebXML 방식의 전자문서 개발", 2002 ebXML Conference, 2002.

[4] 김형도, "ebXML Business Process & Core Component 최신 동향", 2002 ebXML Conference, 2002.

[5] Alan Kotok, David R. R. Webber, "ebXML : The New Global Standard for Doing Business on the Internet," New Riders Publishing, 2001.

[6] ebXML Core Components Specification Project Team, Core Components Technical Specification V1.8, http://www.ebxml.org/projects/documentation/core/CoreComponentsTS1.80.pdf, 2002.

[7] IBM, XML Parser for Java, http://www.alphaworks.ibm.com/tech/xml4j, 2002.

[8] James Cole, Zoran Milosevic, "Extending support for contracts in ebXML," Information Technology for Virtual Enterprises, 2001. ITVE 2001. Proceedings. Workshop on,

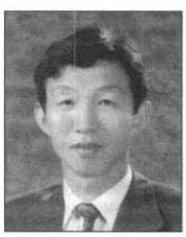
pp.119-127, 2001.
[9] ebXML Business Process Team, Business Process and Business Information Analysis Overview v1.0, http://www.ebxml.org/specs/bpOVER.pdf, 2001.
[10] ebXML Core Components Team, Core Component Discovery and Analysis v1.04, http://www.ebxml.org/specs/ebCCDA.PDF, 2001.
[11] ebXML Core Components Team, Document Assembly and Context Rules v1.04, http://www.ebxml.org/specs/ebCCDOC.pdf, 2001.
[12] ebXML Core Components Team, Naming Convention for Core Components v1.04, http://www.ebxml.org/specs/ebCCNAM.pdf, 2001.
[13] ebXML Core Components Team, Core Component Overview v1.05, http://www.ebxml.org/specs/ccOVER. pdf, 2001.
[14] ebXML Core Components Team, Context and Re-Usability of Core Components v1.04, http://www.ebxml.org/specs/ebCNTXT.pdf, 2001.
[15] ebXML Core Components Team, Guide to the Core Components Dictionary v1.04, http://www.ebxml.org/specs/ccCTLG.pdf, 2001.
[16] ebXML Core Components Team, Core Component Dictionary v1.04, http://www.ebxml.org/specs/ccDICT.pdf, 2001.
[17] OASIS/ebXML Registry Technical Committee, Registry Services Specification v2.0, http://www.ebxml.org/specs/ebrs2.pdf, 2001.
[18] ebXML Technical Architecture Project Team, ebXML Technical Architecture Specification v1.04, http://www.ebxml.org/specs/ebTA.pdf, 2001.
[19] ebXML Requirements Team, ebXML Requirements Specification v1.06, http://www.ebxml.org/specs/ebREQ. pdf, 2001.
[20] OASIS/ebXML Registry Technical Committee, Registry Information Model v2.0, http://www.ebxml.org/specs/ebrim2.pdf, 2001.
[21] ebXML Core Components Team, Core Component Structure v1.04, http://www.ebxml.org/specs/ccSTRUCT.xls, 2001.
[22] R. J. Glushko, J. M. Tenenbaum, B. Meltzer, "An XML Framework for Agent-based E-commerce," Communications of the ACM, Vol.42, No.3, pp.106-114, 1999.
[23] Birgit Hofreiter, Christian Huemer, Wolfgang Klas, "ebXML : status, research issues, and obstacles," Research Issues in Data Engineering : Engineering E-Commerce/E-Business Systems, 2002. RIDE-2EC 2002. Proceedings. Twelfth International Workshop on, pp.7-16, 2002.
[24] Open Applications Group, OAGI White Paper, http://www.openapplications.org/downloads/whitepapers/whitepaperdocs/whitepaper.htm, 2000.
[25] Thomas Y. T. Lee, "ebXML Technology Development in Hong Kong," 2002 ebXML Conference, 2002.
[26] 박병용, "e-Business 기업간 비즈니스 프로세스 분석", 2002 ebXML Conference, 2002.
[27] 이규철, "OASIS ebXML Registry V3.0 Building on a solid foundation", 2002 ebXML Conference, 2002.
[28] 채진석, "전자문서 개발을 위한 Best XML Schema 설계방법", 2002 ebXML Conference, 2002.

황 정 희

e-mail : jhhwang@dblab.chungbuk.ac.kr
1991년 충북대학교 전산통계학과(이학사)
2001년 충북대학교 대학원 전자계산학과 (이학석사)
2001년~현재 충북대학교 대학원 전자계산학과 박사과정
관심분야 : XML, 데이터마이닝, 능동 데이터베이스, 시공간 데이터베이스

김 영 균

e-mail : ygkim@dblab.chungbuk.ac.kr
2000년 서원대학교 전자계산학과(이학사)
2003년 충북대학교 대학원 정보산업공학과 (공학석사)
관심분야 : XML, 전자상거래, 데이터베이스



류 근 호

e-mail : khryu@dblab.chungbuk.ac.kr
1976년 숭실대학교 전산학과(이학사)
1980년 연세대학교 공업대학원 전산전공 (공학석사)
1988년 연세대학교 대학원 전산전공 (공학박사)
1976년~1986년 육군군수 지원사 전산실(ROTC 장교), 한국전자통신연구원(연구원), 한국방송통신대 전산학과(조교수) 근무
1989년~1991년 Univ. of Arizona Research Staff(TempIS 연구원, Temporal DB)
1986년~현재 충북대학교 전기전자컴퓨터공학부 교수
관심분야 : 시간 데이터베이스, 시공간 데이터베이스, Temporal GIS 및 지식기반 정보검색 시스템, 데이터 마이닝 및 데이터베이스 보안, 바이오 인포메틱스