전자 카탈로그를 위한 의미적 분류 모형
(A Semantic Classification Model for e-Catalogs)

김 동 규 [†] 이 상 구 [††] 전 종 훈 [†††] 최 동 훈 [††††]
(Dongkyu Kim) (Sang-goo Lee) (Jonghoon Chun) (Dong-Hoon Choi)

요 약 전자 카탈로그는 시장 참여자들이 제공하는 상품과 서비스에 대한 정보를 가지고 있으며 결과적으로 전자 상거래의 근간을 형성하고 있다. 카탈로그의 관리는 여러 가지 요소에 의해 복잡해지는데, 상품 분류는 이들의 핵심 요소이다. 분류 계층 구조는 지출 분석, 관세 규제의 실행, 상품의 식별 등을 위해 활용된다. 이와 같이 상품 분류 체계는 상품 데이타베이스의 설계에 토대가 되고, 상품 정보의 활용 및 관리의 거의 모든 면에서 중심적 역할을 한다. 그러나, 데이타 모형, 연산, 의미론 등의 측면에서 상품 분류에 대한 형식적인 연구는 거의 없었다. 분류에 관한 논리적 모형의 부재는 분류에 대해서뿐만 아니라 일반적인 상품 데이타베이스에 대해서도 비일관성 및 비융통성 등 많은 문제를 야기시킨다.

상품 정보의 효율적이고 편리한 활용을 위해 각 사용자의 관점에 따른 다양한 뷰를 제공할 필요가 있다. 새로운 상품이 출현하고 기존 상품이 사라짐에 따라 분류체계도 이에 따라 일관성을 유지하면서 변경 및 진화해야 한다. 또한 이질적인 다른 분류체계와 매핑되거나 병합될 필요가 있으며, 이 때 정보의 손실을 줄이는 것이 중요하다. 이들 요구사항에 대해, 분류체계는 제한된 시간 및 비용 내에서 수용할 수 있도록 충분히 동적이어야 한다. 그러나, UNSPSC 및 eCl@ss와 같이 현재 널리 사용중인 분류체계는 이러한 동적인 특성에 대한 요구사항을 만족시키지 못한다.

이 논문에서 우리는 상품 분류 체계의 의미를 이해하고, 기존의 분류 체계의 이면에 있는 의미를 포획하여 표현할 수 있는 방법으로 의미적 분류 모형을 제시하고자 한다. 상품 정보는 재료, 시간, 장소 등의 속성과 무결성 조건과 같은 많은 의미를 지니고 있다. 상품 데이타베이스의 동적 특성 및 이에 대한 기존 코드 기반 분류 체계의 한계점을 분석하고, 제안된 의미적 분류 모형이 상품 데이타베이스의 동적 특성에 관한 요구사항을 만족시킨다는 것을 설명한다. 이 모형은 상품 클래스를 명시적이고 형식적으로 정의할 수 있는 수단을 제공하며, 상품 클래스 간의 관계를 그래프로 구성한다.

이 모형은 분류 체계의 매핑을 용이하게 하며, 선행 연구에 의해 제기된 요구 사항 및 문제를 해결한다고 믿는다.

키워드 : 전자 카탈로그, 분류, 분류 체계, 의미적 분류 체계, 분류 모형, 의미적 분류 모형

Abstract Electronic catalogs (or e-catalogs) hold information about the goods and services offered or requested by the participants, and consequently, form the basis of an e-commerce transaction. Catalog management is complicated by a number of factors and product classification is at the core of these issues. Classification hierarchy is used for spend analysis, customs regulation, and product identification. Classification is the foundation on which product databases are designed, and plays a central role in almost all aspects of management and use of product information. However, product classification has received little formal treatment in terms of underlying model, operations, and semantics. We believe that the lack of a logical model for classification introduces a number of problems not only for the classification itself but also for the product database in general.

It needs to meet diverse user views to support efficient and convenient use of product information. It needs to be changed and evolved very often without breaking consistency in the cases of introduction of new products, extinction of existing products, class reorganization, and class specialization. It also needs to be merged and mapped with other classification schemes without

† 정 회 원 : (주)프람트 연구소장
dkkim@corelogix.co.kr
†† 종신회원 : 서울대학교 컴퓨터공학부 교수
sglee@snu.ac.kr
††† 종신회원 : 명지대학교 컴퓨터공학과 교수
jchun@mju.ac.kr
†††† 비 회 원 : 한국과학기술원 전산학과 교수
choid@paran.com
논문접수 : 2005년 4월 20일
심사완료 : 2005년 12월 1일

information loss when B2B transactions occur. For these requirements, a classification scheme should be so dynamic that it takes in them within right time and cost. The existing classification schemes widely used today such as UNSPSC and eCl@ss, however, have a lot of limitations to meet these requirements for dynamic features of classification.

In this paper, we try to understand what it means to classify products and present how best to represent classification schemes so as to capture the semantics behind the classifications and facilitate mappings between them. Product information implies a plenty of semantics such as class attributes like material, time, place, etc., and integrity constraints. In this paper, we analyze the dynamic features of product databases and the limitation of existing code based classification schemes. And describe the semantic classification model, which satisfies the requirements for dynamic features of product databases. It provides a means to explicitly and formally express more semantics for product classes and organizes class relationships into a graph. We believe the model proposed in this paper satisfies the requirements and challenges that have been raised by previous works.

Key words : electronic catalogs, classification, classification scheme, classification model, semantic classification scheme, semantic classification model

1. 서 론

오늘날 인터넷을 통한 비즈니스의 수행은 더 이상 학문적 개념이 아니라 일반적 현실로 받아들여지고 있다. 전자상거래 환경은 여러 다양한 시장 참여자들 간에 탐색, 주문, 배송, 지불, 송장, 중재와 같은 폭넓은 상호 작용 프로세스를 포함한다. 전형적인 전자상거래 트랜잭션은 상품과 서비스에 대한 정보 수집, 잠재적 시장 파트너의 탐색, 입찰 공고 및 참여, 상품의 운송, 지불, 거래 후처리 등으로 구성된다. 전자 카탈로그는 시장 참여자가 서로 요구하거나 제공하는 상품과 서비스에 관한 정보를 담고 있으며, 컴퓨터에 의한 전자 카탈로그의 관리는 이러한 전자 상거래 프로세스의 많은 부분을 자동화하고 간략화할 수 있는 기회를 제공한다. 결과적으로 전자 카탈로그는 전자 상거래의 근간을 형성하며, 전자 카탈로그 관리는 전자 조달, 이마켓플레이스, 공급망 관리, 전사적 자원 관리, 상품 정보 관리 등 거의 모든 전자상거래 응용에서 공유하고자 하는 기능이다.

전자 카탈로그 관리는 많은 요소로 인해 매우 복잡하다. 그 중의 하나가 상품 데이타베이스에서 테이블 구조의 다양성이다. 10만 개의 상품을 포함하는 전형적인 카탈로그는 수천 개의 테이블 정의를 포함하기도 한다[1]. 또 다른 하나는 사용자의 관점의 다양성이다. 생산자는 상품 제조에 관련된 원재료와 프로세스의 의미로 상품을 보는 경향이 있으나, 중개자는 상품의 크기, 무게 등에 더욱 관심을 갖는다. 관점의 차이는 현재 사용 중인 상품 분류 체계의 다양성에서 명확하게 드러난다.

동일한 특성의 상품을 클래스로 분류하고 이들 간의 관계를 정의하는 상품 분류는 이러한 문제의 중심에 있다. 즉, 사람들은 상품을 상품 클래스(또는 유형)의 의미로 생각하고 이에 근거하여 상품 데이타베이스에 질의를 던진다. 따라서, 분류체계는 상품 데이타베이스를 설계하는 데 토대가 되며, 상품 정보의 활용 및 관리의 거의 모든 면에서 중심적인 역할을 한다.

- 상품 데이타베이스의 설계: 분류 계층구조에 근거하여 상품 데이타베이스를 설계한다[9]. 상품 클래스의 상품 인스턴스는 공통적인 특성의 집합을 공유하는데, 이들 특성 집합은 데이타베이스에서 속성으로 모형화된다.
- 상품 식별 및 탐색: 상품 데이타베이스에 대한 대부분의 탐색 질의는 키워드 리스트의 형태이다. 이 리스트는 보통 하나 이상의 속성에 대한 제한 조건과 함께 상품 클래스를 포함한다.
- 지출 분석: 현업에서는 분류 체계를 이용하여 집계를 수행한다. 즉, 미리 정의되어 있는 클래스 집합과 이들 간의 계층구조는 분석적 처리의 드릴 다운(drill-down) 및 드릴 업(drill-up)을 위한 기초가 된다. 이와 같이 분류 체계를 통해 데이타를 다차원적으로 분석할 수 있다는 것은 의사결정 지원을 위해 매우 중요하다.
- 규제: HS 코드[2]는 국제 관세 규제를 위해 사용된다. 또한, 유통 통제, 유지 보수 및 관리를 위한 각국의 표준도 여러 가지가 있다. 이와 같은 활용을 위해 배타적이며 명확한 클래스의 정의는 매우 중요하다.

분류체계가 이와 같이 중요한 역할을 수행함에도 불구하고, 명시적이고 형식적 접근 방법을 통한 상품 분류체계의 의미적 표현 및 이에 대한 연산, 제한 조건 등에 관한 분류체계의 모형은 연구 주제로 주목을 받지 못했다. 이러한 사실은 오늘날 사용중인 분류체계에서 입증된다. HS, UNSPSC[3], eCl@ss[4]은 산업 전분야에서 사용되는 중요한 국제표준들이다. 또한 IEC61630(전자), UNCPC(운송), SITC(국제무역), CPV(조달계약) 등과 같은 특정 산업 분야를 위한 상품 분류표준도 매우 많

다. 이들 표준은 대상 상품을 분류하기 위한 그들 고유의 목적과 기준을 가지고 있다. 그러나, 이들 분류 체계는 트리 구조 및 분류 코드를 갖는 단순 코드 기반의 분류 모형을 적용하고 있다. 코드 기반의 분류 모형에서 수퍼 클래스의 코드(예를 들어, 4010)는 서브 클래스 코드(예를 들어, 401011, 401012 등)의 접두어가 된다. 상품 정보의 개념, 개념 간의 관계, 무결성 조건 등은 주로 명시적이 아닌 암시적인 방법으로 표현되거나 기껏해야 사람이 읽을 수 있는 가이드 라인에 명시되기도 한다. 코드 기반 분류체계를 위한 논리적 모형의 이와 같은 미흡한 점은 동일한 분류 체계 내에서 분류 기준 및 분류 수준의 혼재로 인한 비일관성 및 비융통성과 같은 여러 문제를 야기한다. 이러한 비융통성 및 비일관성은 상품 데이타베이스 전체에도 부정적인 영향을 미친다.

상품 정보 관리에 관한 많은 연구가 수행되어 다양한 문제에 대한 결과를 보여 주고 있다. Fensel과 Omelayenko[5]는 상품 정보의 통합 문제에 관한 연구 결과를 보여주고 있으며, 상품 정보의 구축, 유지, 통합에 대한 어려운 면을 나열하고 있다. Shulten[6] 등의 연구 결과에서, 두 분류 체계 간의 동기화의 어려움을 보여줌으로 써, 상품 분류에 특정된 문제를 언급하고 있다. 그들은 정보손실을 최소화하면서 서로 다른 분류체계 간의 매핑을 제공할 수 있는 모형에 대한 제안을 요청하기도 하였다. Leukel 등은[7] eCl@ss를 다루는 실용적인 설계 방안을 제시하고 있지만, 분류를 위한 데이타 모형을 제시하지는 못한다. Agrawal과 Srikant[8]는 기존의 분류체계가 주어졌을 때 상품의 집합을 자동 분류하는 방법을 제시하고 있다. 이들의 방법은 학습 및 분류와 같은 데이타 마이닝 측면에 초점을 맞추고 있으며, 분류 계층구조의 모형을 다루고 있지 않다. 이와 같은 기존의 연구는 상품 정보관리의 여러 다양한 문제와 해결책을 정연하게 설명하고 있으나, 상품분류를 위한 모형은 UNSPSC, eCl@ss, HS 등의 단순한 코드기반 계층모형을 가정하고 있다. 이 논문에서, 우리는 상품 분류가 무엇을 의미하는지 설명하고자 하며, 상품 분류 체계의 근거가 되는 의미를 충분히 포획하고 나아가 이질적 분류체계 간의 매핑을 수월하게 할 수 있는 최선의 새로운 분류체계 분류체계의 표현 방법을 제시하고자 한다. 이 모형은 [7]의 요구사항을 만족하며, [6]에서 제기된 문제에 대한 하나의 해결책이 된다.

Hepp[9]과 Lee[10]은 상품 정보 관리에서 속성의 중요성을 강조하고 있다. Hepp은 시맨틱 웹에서 기계가 읽을 수 있는 상품 서술을 위해 상품 분류 시스템, 속성 라이브러리, 기타 의미 표준 등에 포함된 정보를 활용하는 방법을 제안하고 있으며, 속성 리스트의 품질에 근거하여 상품 분류 표준의 품질을 평가하고 있다. [6]는 코드 기반 분류 계층 구조가 상품 집합에 대해 사용자가 요구하는 가능한 여러 뷰 중에서 단 하나만을 지원하고 있다는 것을 지적하면서, 이를 해결하기 위해서는 상품의 식별과 속성이 상품의 분류 방법에 독립적이어야 함을 주장하고 있다. 또한, 계측 구조보다는 속성에 초점을 맞춘 상품 데이타베이스 설계 문제와 가이드라인을 제시하고 있다. Guo와 Sun[11]은 상품 정보의 표현 및 교환을 위한 개념 중심의 공학적 접근 방법을 제안하였고, 상품 정보의 의미를 강조하였다. 이들 연구는 상품 데이타의 모형화에 중요한 의미와 특성을 강조하는 최신 연구 결과이다. 우리의 연구 결과는 상품 데이타 의미론의 필수적인 차원으로서 상품 분류를 위한 구체적인 모형을 제공할 뿐만 아니라, 이들의 연구 결과를 보완한다.

논문의 나머지 부분은 다음과 같이 구성된다. 2절에서 상품 집합과 상품 클래스를 다룰 때 발견되는 준거 사항을 소개하고, 분류 모형을 구성하는 여러 특성과 기존 분류모형의 한계점을 설명한다. 3절에서 이 논문의 중심 부분으로 의미론적 분류 모형을 제안하고, 이 분류모형에 따라 기존의 분류체계가 갖는 문제점을 해결하는 응용을 4절에서 설명한다. 5절에서 이 모형의 구현에 대해 간략히 소개하고, 6절에서 결론을 맺는다.

2. 분류 모형에 관한 사전 지식

UNSPSC와 eCl@ss는 클래스의 집합과 그들의 계층적 관계를 정의하는 카탈로그 분류의 각 인스턴스이다. 이러한 인스턴스를 분류체계라고 부른다. 분류 모형은 분류체계의 기초가 되는 데이타 모형으로, 클래스의 정의가 형식적인지 또는 비형식적인지, 클래스 간의 관계가 트리 또는 그래프를 형성하는지, 상품의 특성이 속성으로 정의되는지 아닌지, 상품이 다수의 말단 클래스에 속하는지 또는 단 하나의 말단 클래스에 속하는지 등과 같은 문제를 정의한다. 여기서 형식적인 서술이라 함은 기계가 이해할 수 있는 서술을 의미하며, 비형식적인 서술이란 기계는 이해할 수 없고 인간만이 이해할 수 있다는 것을 의미한다. 예를 들어, UNSPSC 및 eCl@ss는, 클래스 정의가 비형식적이고 클래스 관계가 트리 구조이며, 클래스의 속성을 정의하지 않고, 상품은 단 하나의 말단 클래스에만 속하는, 모형을 공유한다.

2.1 분류 모형의 준거 사항

먼저 의미론적 분류 모형의 토대가 되는 몇 가지 준거 사항을 나열한다. 이들 중 몇 가지는 아주 당연한 것일 수 있으나, 그럼에도 불구하고 나열한다. 왜냐하면 상업적 상품 데이타베이스에서 발생하는 많은 심각한 문제들이 이러한 사항을 망각하거나 또는 혼동한 결과

이기 때문이다.

준거 사항 1. 상품은 고유의 특성을 가진 개체이다.

준거 사항 2. 상품의 클래스는 상품의 집합이다.

상품 클래스는 개념과 집합, 두 가지로 모형화될 수 있다. "평면 TV" 클래스는 클래스의 구성원에서 나타나야 하는 특성에 의해 정의되거나, 특성을 참조하지 않고 구성원을 나열하여 정의할 수 있다.

준거 사항 3. 상품의 정체성(identity)은 상품이 어떻게 분류되느냐에 의존하지 않는다.

A4 크기의 흰색 용지는 eCl@ss에서 그것이 "type-writer paper" (24-26-03-01)로 분류되든, "fax paper" (24-26-04-01), 또는 "copy paper" (24-26-06-01)로 분류되든 상관없이 고유의 정체성을 그대로 보존한다. 당연한 준거 사항이지만 이것을 인식하지 못하면 여러 상품 데이타베이스에서 이미 널리 발생된 혼란을 가져온다.

준거 사항 4. 하나의 분류 체계는 상품 데이타 공간에 대한 단 하나의 뷰만을 표현한다.

단 하나의 분류체계만으로는 사용자들의 수많은 다양한 분류 목적에 적합한 뷰를 서비스할 수는 없다. 예를 들어, 생산자는 재료와 이에 관한 공정으로 상품을 보는 경향이 있으나, 중개자는 상품의 크기나 무게에 따른 분류에 더 관심이 있다. 또한, 회사의 지출 분석에 적합한 분류체계는 담당자의 뷰를 표현하고 있지만, 상품 데이타베이스 설계를 위한 가장 좋은 분류체계가 아닐 수 있다. 상품 데이타베이스의 설계에서는 공통 속성의 공유가 가장 중요한 분류 기준이어야 하는 반면, 지출 분석에서는 부서가 다르면 서로 다른 분류에 관심을 가질 수 있고 담당자의 관심도 시간에 따라 달라질 수 있기 때문이다. 이러한 모든 다양한 뷰를 융통성 있게 지원하는 단 하나의 분류체계를 구축하는 것은 거의 불가능한 일이다. 아직도 너무나 많은 회사가 몇 년에 한번씩 상품 데이타베이스를 재구축하기 위해 많은 자원을 쏟아 붓고 있는 실정이다.

준거 사항 5. 속성 값은 계층적이다.

'생산자' 속성의 '삼성'이라는 값은 '이동전화기 생산자' 클래스의 구성원이며, '이동전화기 생산자' 클래스는 'IT 회사'의 서브 클래스다. '재료 속성'의 값 '알루미늄'은 '특수 금속' 클래스에 속해 있으며, '특수 금속' 클래스는 '금속'의 서브 클래스다. 이 준거 사항은 상품 데이타베이스에서 지원해야 할 수많은 계층 구조의 뷰가 있다는 앞에 언급한 준거 사항 4를 재차 강조하는 것이다.

2.2 상품 분류 모형의 특징

현재 사용 중인 분류 체계의 문제점 중의 하나는 기초가 되는 모형이 암시적이라는 것이다. 예를 들어, 코드 기반의 분류체계에서 클래스의 코드(식별자)는 계층 구조 상에 있는 그의 조상들의 식별자를 연결해 놓은 것이기 때문에 하나의 클래스는 하나의 부모 클래스를 가지며 다중 상속(multiple inheritance)을 지원할 수 없다. 이러한 분류체계에서 다중 상속이 허용된다면, 클래스는 하나 이상의 식별자를 갖게 되는 모순된 결과를 초래한다. 따라서, 코드 기반 분류체계의 클래스 식별자는 암시적으로 트리 모형을 기초 모형으로 함축한다. 모형의 선택은 분류 체계에 대한 표현의 선택을 제한한다. 이 절에서 우리는 분류 모형을 정의하는 여러 가지 특성을 설명한다.

2.2.1 트리 대 그래프

분류 체계는 트리나 그래프로 모형화될 수 있다. 트리 모형에서 클래스는 단 하나의 부모 클래스를 갖는 반면, 그래프 모형에서는 다수의 부모 클래스를 가질 수 있어서 다중 상속이 허용된다. 모든 코드 기반의 분류 체계는 클래스 코드가 길이가 동일한 두 개의 다른 접두어를 가질 수 없기 때문에, 모형이 트리일 수 밖에 없다. 그러나, 다중 상속의 특성을 갖는 그래프 모형은 더욱 융통성이 있고 자연스럽다. 예를 들어, 그림 1에 있는 코드 기반의 UNSPSC 분류 체계의 일부분을 보자. 왼쪽 그림에서 14-11-15-11 'Writing paper'는 14-11-15

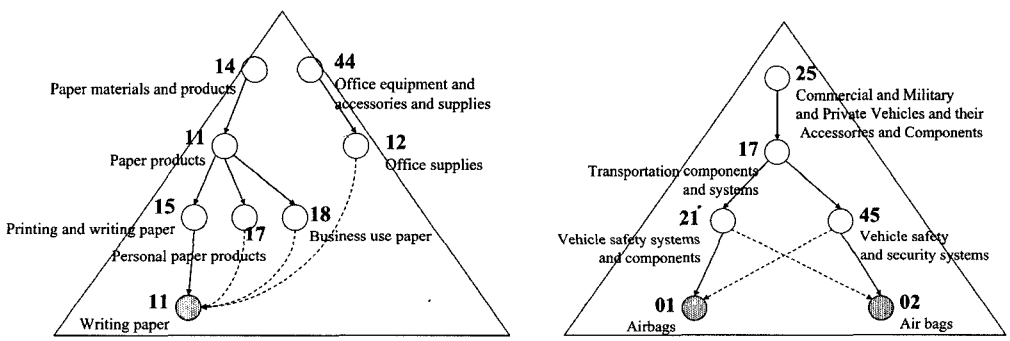

그림 1 UNSPSC의 일부분

'Printing and writing paper'의 아래에 있으나, 의미적으로 'Personal paper products' 또는 'Business use paper', 'Office supplies'의 아래에 위치할 수 있다. 실제로 eCl@ss에서는 'Writing paper'는 'Office supplies'의 아래에 있다. 다수의 부모가 허용되지 않기 때문에 이 분류체계는 부가적인 관계를 나타낼 수 없다.

2.2.2 형식적인 의미 대 비형식적인 의미

클래스의 서술은 클래스의 의미를 구성하며, 해석되어 그 클래스의 구성원을 결정하는 데 사용된다. 현재의 분류체계에서 사용되는 클래스 코드는 아무런 의미도 전달하지 못하고 단지 부모-자식 관계 만을 표현하는 수 단일 뿐이다. 이들 분류 체계에서 클래스의 의미는 클래스의 이름과 텍스트 서술의 형태로 표현되며, 이러한 표현은 의미가 모호하기 때문에 컴퓨터 프로그램에서 사용하기 어렵다.

형식적인 의미를 통해 모호성을 제거한 클래스는 프로그램에 의해 직접 사용될 수 있다. 예를 들어, 지출 분석을 위해 형식적인 의미를 집계 조건으로 사용할 수 있고, 모호성이 제거된 의미는 규제 실행을 위한 상품 탐색과 식별의 효과를 올릴 수 있다. 형식적인 의미가 제공하는 부가적인 정보는 자동 분류 및 데이타베이스 설계를 가능하게 한다.

2.2.3 속성의 필수성

속성의 정의는 클래스 정의에 필수적이다[4]. 1.2절의 준거 사항 1에서 언급되었듯이 같은 클래스의 상품 집합은 공통 특성을 공유하며 이들 특성은 상품의 속성으로 나열된다. 직간접적으로 속성을 언급하지 않고는 상품의 특성을 말할 수 없다. 따라서, 클래스를 위한 기초적인 속성의 집합을 정의하는 것은 매우 중요하며 필수적이다. 그러나, 현재의 분류 체계 대부분은 속성을 가지고 있지 않다. eCl@ss는 말단 클래스에 한해서 속성을 정의할 수 있고 UNSPSC는 속성을 정의할 수 있도록 작업을 진행 중이다.

현재 사용중인 대부분의 분류 체계는 사용의 편리함과 일관성을 위해 말단 클래스를 동일한 수준에 둔다. 그러나, 계층 구조의 특정 부분을 다른 부분보다 더욱 상세하게 정의해야 할 경우 이러한 제한은 부담이 된다. 설계자는 응용의 의미, 정책 등 여러 요인에 의존하여 이들 사항을 결정해야 한다. 이것은 분류 모형의 기본적인 특성은 아니고, 각 분류 체계에 개별적으로 이행 가능한 특정 조건으로 취급될 수 있다.

표 1은 지금까지 논의된 분류 모형의 특성을 요약한 것이다.

다음 절에서 제안될 의미론적 분류 모형은 이러한 특성을 두루 갖춘 일반적인 모형이다. 위에 언급된 선택 사항과 조건을 주의 깊고 명시적으로 선택함으로써, 설계자는 응용에 적절한 특성을 갖는 분류 체계를 체계적으로 정의할 수 있다.

2.3 동적 특성의 지원에 대한 기존 분류체계의 한계점

현재 사용되고 있는 분류체계에는 산업 전반에 걸쳐 사용되는 Harmonized Commodity Coding System, UNSPSC, eCl@ss를 비롯하여 특정 산업에서 사용되는 IEC61630 (전자), UNCPC (운송), SITC (무역), CPV (조달 계약) 등 여러 가지가 있다. 이들 표준은 대상 상품의 분류에 관한 고유의 목적과 기준이 있다. 이러한 분류체계는 단순히 코드 기반의 분류체계로 클래스의 정의는 비형식적이며 기계가 이해할 수 있는 형태가 아닌 사람이 이해할 수 있는 것이다. 각 상품은 하나의 클래스에만 속하며, 여러 클래스에 속할 수 없다. 클래스 간의 관계는 IS-A 관계(부모-자식 간의 관계)이며, 클래스 계층구조는 트리 구조를 형성한다.

코드 기반 분류체계의 각 클래스는 '클래스 코드'와 '클래스 설명'으로 구성되며, '클래스 설명'은 클래스 이름을 포함하는 자연어 서술이다. 수퍼 클래스의 클래스 코드는 서브 클래스 코드의 접두사가 된다. 즉, 클래스 코드 4010은 서브 클래스 401011 및 401012의 접두사이다. 이들 코드를 관찰하면 클래스 간의 부모-자식 관계를 알 수 있다.

이 절에서는 코드 기반 분류체계가 동적 특성의 요구사항을 만족시키는데 어떤 문제를 안고 있는지 분석하기로 한다.

2.3.1 코드 기반 분류체계에 근거한 상품 데이타베이스에서 다양한 뷰의 지원

코드 기반 분류체계에 근거한 상품 데이타베이스는 사용자에게 단 하나의 고정된 뷰만을 제공한다. 이와 다른 관점의 사용자는 제공된 뷰에 따라 상품을 탐색하는 데 불편함을 느끼며 전자상거래의 효율을 떨어뜨린다. 코드 기반 분류 체계는 두 개 이상의 클래스에 속한 상

표 1 분류 모형의 특성

특성	선택 사항	기본적 특성 여부	기타
클래스 계층구조	트리 또는 그래프	0	그래프를 선호. 트리를 사용할 수도 있겠으나 일관적인 트리를 유지하기에 어려움이 따름.
의미의 형식성	형식적 또는 비형식적	0	형식적인 명세가 필수적.
속성 정의	필수 또는 없음	0	속성의 형식적 명세가 필수적.
말단 클래스 수준	동일 수준 또는 제한없음	x	응용에 의존적.

품을 한꺼번에 탐색할 수 없기 때문에, 사용자가 탐색하고자 하는 상품이 두 개 이상의 클래스에 분류되어 있을 경우, 분류체계의 경로를 따라 올라 갔다가 다시 내려 와서 탐색해야 한다. 이것도 상품 정보의 활용 효율을 떨어뜨리는 원인이 된다.

그림 1의 오른쪽의 그림은 실제로는 동일하지만 25-17-21-01 'Airbag'와 25-17-45-02 'Air bags'의 다르게 정의된 두 클래스를 보여준다. 이것은 다수의 부모가 허용되지 않기 때문에 동일한 클래스가 중복되어 다르게 정의된 것이다. 코드 기반의 모형에서는 중복 클래스가 완전히 다른 클래스로 취급되며, 구매자가 어느 한 쪽의 에어백을 탐색하고 있을 경우 다른 쪽에 있는 동일한 클래스의 존재를 알 수 있는 체계적인 방법이 없다.

2.3.2 코드 기반 분류체계의 변화 및 진화

코드 기반 분류체계에서 클래스의 삽입 및 삭제는 연산 전과 후에 일관성을 유지하도록 주의해야 한다. 예를 들어 서로 배타적으로 정의된 클래스 '팩스'와 '프린터'가 이미 분류체계에 있는 상황에서, 팩스와 프린터의 속성을 공유하는 팩스 겸용 프린터가 새롭게 등장했다고 가정하자. 새로운 클래스 '팩스 겸용 프린터'를 분류체계에 단순히 추가하면 '팩스 겸용 프린터'는 '팩스'에도 속하고 '프린터'에도 속하기 때문에 클래스 간의 배타적 정의에 일관성이 없어진다. 배타적 정의를 유지하려면, 클래스 '팩스 겸용 프린터'를 추가할 때 클래스 '팩스'를 '프린터 기능이 없는 팩스'로 재정의해야 하며 '프린터'를 팩스 기능이 없는 프린터'로 재정의해야 한다. 이와 같이 분류체계를 일관적으로 변화 및 진화시키려면 삽입 또는 삭제되는 클래스에 영향을 받는 클래스를 식별하고 이들을 재정의하는 것이 필요하다.

코드 기반 분류체계에서 삽입 및 삭제되는 클래스에 영향을 받는 클래스를 식별하고 재정의하여 상품 정보의 일관성을 유지하는 것은 자동화 대상이 아니며 분류체계 전문가만이 할 수 있다. 이것은 시장의 변화에 대해 신속한 적응을 요구하는 전자상거래 환경에 부적합하다.

2.3.3 이질적 코드 기반 분류 체계 간의 매핑

전자상거래의 상호운용성을 위해, 국제표준인 UNSPSC와 eCl@ss 간에 정의된 매핑 테이블이 이미 널리 사용되고 있다[3]. 그러나, 그림 2에서 볼 수 있듯이 이들 코드 기반 분류체계 간의 매핑은 정보 손실을 내포하고 있다[4]. 즉, 의미적으로는 대응되지만 매핑 테이블에 명시적으로 정의되지 않은 대응 관계가 존재한다.

UNSPSC에서 'Printing & Writing paper'는 eCl@ss의 'Writing Papers'와 'Printer Papers'의 결합으로 매핑되어 있다. 반면, UNSPSC에서 'Printing & Writing paper'의 수퍼 클래스인 'Paper Products'는 eCl@ss의 'Paper, Films'에 매핑되어 있지만 'Writing Papers'에 매핑되어 있지 않다. 수퍼 클래스의 대응 관계는 의미상 서브 클래스의 대응 관계를 포함해야 하는데도 불구하고, 매핑 테이블에 포함되지 않았다. 이것은 크로스웍 매핑 테이블이 정보 손실을 내포하고 있다는 것을 말해준다.

2.3.4 이질적인 분류체계의 병합

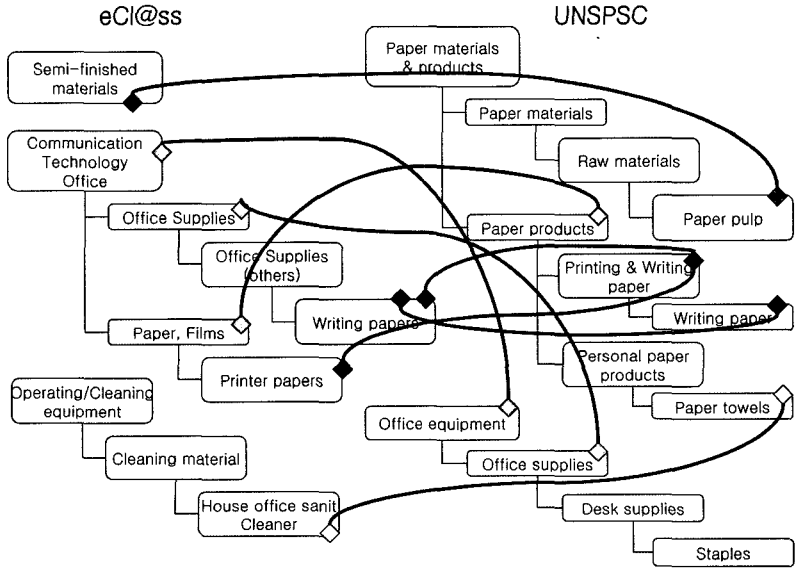

그림 2 UNSPSC와 eCl@ss 간의 매핑(일부분)

코드 기반의 두 분류체계를 병합하면 2.3.2와 2.3.3절에서 이미 설명한 바와 같이 정보 손실이 발생하며 일관성 유지에도 문제가 있다. 이러한 문제의 해결은 분류 전문가의 도움을 반드시 필요로 하고 상당한 시간과 노력을 요구하기 때문에, 코드 기반 분류체계의 사용은 전자상거래 환경의 변화에 대한 능동적인 대처에 부정적인 영향을 미친다.

3. 의미적 분류 모형

상품 클래스는 동일한 특성을 갖는 상품의 집합이다. S를 분류 대상인 모든 상품의 집합이라 할 때, 클래스는 S의 부분 집합이다. 클래스는 소속 상품을 나열하여 정의할 수도 있으나, 이 방법은 상품의 인스턴스가 새로 삽입되고 삭제될 때마다 클래스를 지속적으로 수정해야 하기 때문에 좋은 방법이 아니다. 대신에 클래스를 정의하기 위해 클래스의 구성원의 특성을 명세화하여 내포를 정의하는 방법을 사용한다. 예를 들어, 클래스는 클래스 이름, 클래스에 대한 간략한 설명, 논리적 술어 공식 등 데이타베이스 질의에서 클래스 구성원을 검색할 때 필요한 특성에 의해 정의된다.

클래스 정의가 모호하지 않아야 하고 기계가 이해할 수 있는 형태여야 한다는 것은 매우 중요하다. 모호하지 않다는 의미는 그 정의를 누가 해석하든 관계없이 각 클래스에 대해 동일한 상품의 집합이 식별되어야 함을 뜻한다. 기계가 이해할 수 있는 형태의 클래스 정의는 일관성 검토를 용이하게 하며 분류체계의 활용성 및 효율을 개선한다. UNSPSC와 같은 현재의 분류 체계는 이들 조건을 만족시키지 못한다. 클래스 코드는 계층 구조 내에서 클래스의 위치를 식별할 뿐이고 클래스의 의미에 관한 정보를 제공하지 못한다. 클래스 이름 및 텍스트 서술은 인간만이 이해할 수 있는 형태이며 종종 모호하기까지 하다.

우리가 제안하는 의미적 분류모형(Semantic Classification Model, SCM)은 클래스 계층구조를 각 클래스의 형식적인 의미로 확장시킨다. SCM은 상품의 집합, 클래스, 클래스 간의 관계를 모형화하기 위한 확장 분류 모형이다. SCM에서는 각 상품에 대해 상품 번호, 이름, 생산자, 설명, 가격 등의 속성 집합이 존재하며, 이들 속성은 각 상품을 설명하기 위해 사용된다. 클래스는 공통 특성을 공유하는 상품의 집합으로 정의되며, 이들 특성은 속성과 제한 조건으로 표현된다. 클래스는 조건이 더욱 한정된 제한 조건 및 속성을 갖는 서브 클래스를 하나 이상 가질 수 있다. SCM에서 클래스의 집합은 방향성의 비순환 그래프(Directed Acyclic Graph)를 형성한다. 이 모형에 근거한 분류 체계는 클래스 코드를 기계적으로 할당한 것이 아니라 상품 집합 S의 의미적 표현이다.

3.1 구조

SCM에서 분류체계의 구조는 기본적으로 DAG이다. 노드는 상품 클래스를 나타내고 간선은 클래스 간의 부모-자식 관계를 나타낸다.

정의 3.1 상품 집합 S에 대한 의미적 분류 체계는 5-튜플 $< \boldsymbol{S}, \boldsymbol{C}, C_0, \boldsymbol{E}, \boldsymbol{IC} >$이며, 여기서

$\boldsymbol{S}$는 분류 대상인 모든 상품들의 집합이고,

$\boldsymbol{C} = \{C_0, C_1, \cdots, C_n\}$는 클래스의 집합으로,

각 C_i는 3-튜플 $<D_i, A_i, M_i>$이다. D_i는 클래스 이름과 간략한 문장 서술로 이루어진 텍스트 설명이고, A_i는 C_i의 모든 구성원에 공통적인 속성들의 집합이고, M_i는 A_i에 대해 논리식으로 정의된 구성원 자격 함수(membership function)이다.

$C_0 \in \boldsymbol{C}$는 $\boldsymbol{S}$의 모든 구성원을 포함하는 루트 클래스이다.

$\boldsymbol{E} \subseteq \boldsymbol{C} \times \boldsymbol{C}$는 C의 클래스 간의 부모-자식 관계를 표현하는 간선의 집합이다.

$\boldsymbol{IC}$는 의미적 분류 체계에 정의된 무결성 조건의 집합이다. ■

속성 집합 A_i는 클래스C_i를 정의하고 구성원 자격함수 M_i는 그 의미를 정의한다. 의미적 정의는 클래스의 상품 집합을 식별한다. 각 클래스에 대해 상품 인스턴스에 대한 포인터를 유지하든, 상품 인스턴스가 말단 노드에 대한 포인터를 유지하든, 이것은 구현 상의 문제이다. SCM은 이러한 상세 사항을 모형으로부터 분리하여 분류체계 설계자가 분류의 의미적 내용에 초점을 맞출 수 있도록 한다.

예 3.1) 그림 3은 의미적 분류 체계를 보여준다. 클래스 'Camera'는 'sensor' 속성의 값에 따라 'Film camera'와 'Digital camera'로 세분화된다. 각 서브클래스는 부모 클래스로부터 상속된 속성 이외에도 고유의 속성을 가진다. 'High end camera' 클래스는 'camera'의 서브클래스로 'retail price'의 값에 따라 분류된다. 명시적인 의미의 정의를 통해 'High end camera'가 다른 두 형제 클래스의 분류 기준과 서로 다르다는 것을 명확하게 알 수 있다. 이것은 사용자에게 분류 기준의 차이에 의한 다양한 관점을 제공할 수 있는 근거가 된다.

'High end digital camera'는 'Digital camera'와 'High end camera'의 서브 클래스이다. 양쪽 부모 클래스의 속성은 자식 클래스에게 상속되며 구성원 자격 함수는 양쪽 부모의 함수를 논리곱(∧)으로 연결한 것이다. ■

3.2 제한 조건

무결성 조건의 집합 IC는 분류체계에 따라 서로 다를 수 있다. 어떤 체계는 다중 상속을 허용하는 반면, 다른

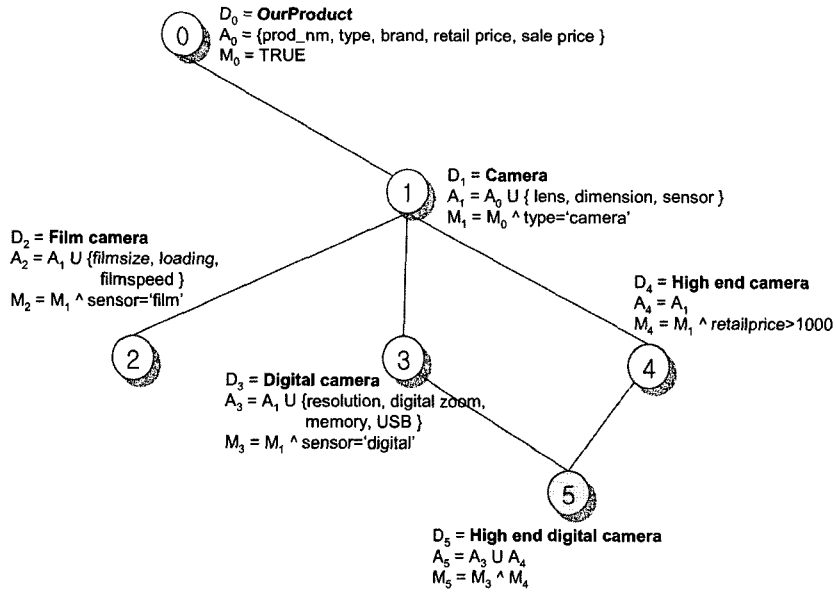

그림 3 의미적 분류 체계

분류 체계는 그것을 허용하지 않는다. 그러나 분류체계가 실용적이기 위해서는, 다음과 같은 기본적인 무결성 조건을 만족해야 한다.

정의 3.2 기본적 무결성 조건

C1. M_0=TRUE (모든 상품에 의해 만족되는 것으로, C_0의 정의에 따르면 만족됨을 알 수 있다.)

C2. $\boldsymbol{E}$ (부모-자식 관계)는 순환을 형성하지 않는다.

C3. 만일 $< C_i, C_j > \in \boldsymbol{E}$ 즉, C_i가 C_j의 부모라면, $A_i \subseteq A_j$,이다. (속성상속 제한조건)

C4. 만일 $< C_i, C_j > \in \boldsymbol{E}$ 라면, M_j는 M_i를 함축한다. 즉, $C_j \subseteq C_i$ 이다. (건전성 제한조건)

의미적 분류체계는 클래스($\boldsymbol{C}$)를 노드로 나타내고 부모-자식 관계($\boldsymbol{E}$)를 간선으로 나타내는 그래프로 정의된다. 우리는 의미적 분류체계에서 두 가지의 형식적 관계를 정의한다.

정의 3.3 의미적 분류체계 $CL = < \boldsymbol{S}, \{C_0, C_1, \cdots, C_n\}, C_0, \boldsymbol{E}, \boldsymbol{IC} >$에 대해,

명시적 관계(Topological Relationship) $\geq_T$를 다음과 같이 정의한다.

만일 CL에서 C_i가 C_j의 조상이면, $C_i \geq_T C_j$이다.

의미적 관계(Semantic Relationship) $\geq_S$를 다음과 같이 정의한다.

만일 M_j가 M_i를 함축하면(imply), $C_i \geq_S C_j$ 이다. ■

명시적 관계란 분류체계 그래프에서 클래스 간의 경로가 명시적으로 표현되어 있는 관계를 말한다. 이들 관계는 코드 기반의 분류체계의 클래스 코드처럼 설계자의 의도를 명시적으로 나타낸다. 반면에 의미적 관계란 설계자의 의도와 무관하게 클래스 간에 만족하는 포함(subsumption) 관계이다. 분류체계를 무결성 조건 C4가 만족되도록 올바르게 정의했다면, 명시적 관계에 있는 각 클래스 쌍은 의미적 관계를 갖는다. 이것을 형식적으로 정의하면 다음과 같다.

정의 3.4 건전성(Soundness)과 완전성(Completeness)

만일 각 명시적 관계가 의미적 관계를 함축하면, 그 분류체계는 의미적으로 건전하다.

만일 각 의미적 관계가 명시적 관계를 함축하면, 그 분류체계는 의미적으로 완전하다. ■

예 3.2) 예 3.1의 분류체계에서 설계자가 클래스 5('High end digital camera')를 정의할 때 클래스 3('Digital camera')를 인식하지 못했다고 가정하자. 이 경우, 클래스 5는 클래스4의 서브 클래스로 정의되고 속성 'sensor'에 대한 조건식 sensor='digital'이 구성원 자격함수에 논리곱(∧)으로 추가된다(그림 4 참조). $M_5 = M_1 \wedge$ type = 'camera' $\wedge$ retailprice > 1000 $\wedge$ sensor = 'digital'이고 $M_3 = M_1 \wedge$ sensor = 'digital'이므로, M_5가 M_3를 함축한다. 따라서, $C_3 \geq_S C_5$이다. 반면, 이들 간의 명시적 관계가 존재하지 않으므로(즉, $\neg C_3 \geq_T C_5$이기 때문에), 이 분류체계는 의미적으로 불완전하다. 그러나, 모든 명시적 관계가 의미적 관계를 함축하므로, 이 분류체계는 건전하다. ■

위의 예에서 속성 'memory'와 'USB'는 A_3에는 있으나 A_5에는 없다. SCM에 근거한 시스템은 일관성 검토를 실행하여 설계자에게 이와 같은 상황을 알린다. 설계자는 분류체계에 실행하고자 하는 무결성 조건에 의존하여 이러한 상황을 선택적으로 수정한다.

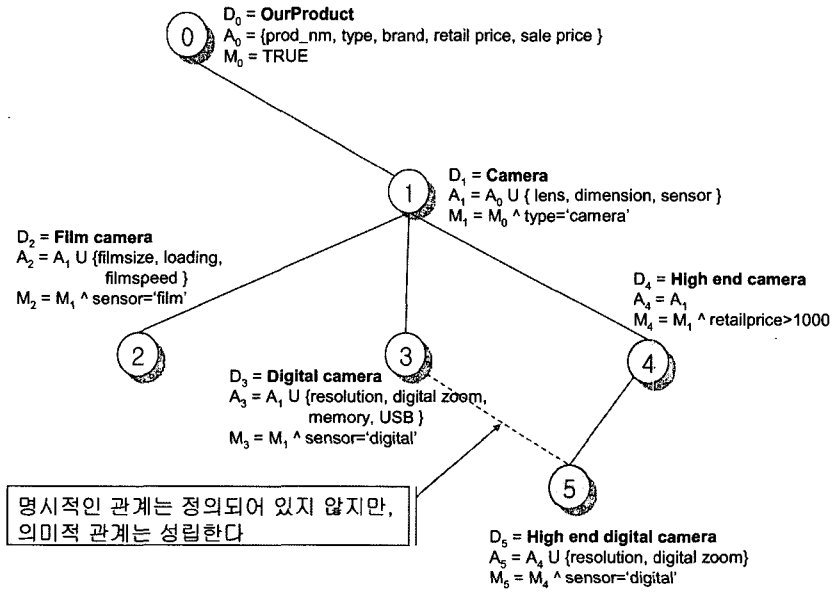

그림 4 의미적으로 건전하지만 불완전한 분류체계

분류 체계의 개별적 특성을 정의하기 위해 부가적인 제한 조건이 명세화될 수 있다. 예를 들어, 다음과 같은 제한 조건 C5를 정의하여 각 노드의 부모를 기껏해야 하나로 제한할 수 있다. 이렇게 하여 다중 상속을 억제하고 분류체계의 구조를 트리로 제한한다.

C5. 각 노드는 기껏해야 하나의 부모를 갖는다. (트리 제한 조건)

만일 $< C_i, C_j > \in E$ (C_i는 C_j의 부모)이고 $< C_k, C_j > \in E$ 이면, 모든 $C_i, C_j, C_k \in C$에 대해 $C_i = C_k$이다.

SCM은 개별적인 분류 체계를 정의하는 프레임웍을 제공한다. 설계자는 이 프레임웍을 이용하여 분류 체계를 위한 제한 조건을 명세화함으로써 분류 체계의 특성을 정의할 수 있다. 이러한 접근 방법의 장점은 제한 조건이 명시적이라는 것이다. 반면 현재 사용중인 대부분의 분류체계 표준에서는 제한 조건이 가정이나 지침에 암시되어 있어서 활용하기 어렵다.

3.3 연산

분류체계의 처리에 어떤 연산이 필요한지 알아보기 위해 UNSPSC의 경우를 예로 들어 설명한다.

예 3.3) UNSPSC는 새로운 버전을 발표할 때마다 분류체계의 수정에 관한 감사적을 포함하는 감사 파일을 발간한다. 그림 5는 상품 'Alcohols'에 대한 감사적의 예를 보여준다. 첫째 레코드는 이 상품이 UNSPSC 1.0에 'Alcohol'로 추가되었음('add')을 나타낸다. 마지막 레코드는 UNSPSC 2.09에서 'Alcohol'은 12101500에서 12211500으로 이동('move')하여 'Alcohols'로 바뀌었음을 보여준다. 다른 두 레코드는 이러한 이동이 일어나면서 두 가지 상품 'Unsaturated monohydric alcohol' 및 'Methylated spirit'이 삭제('delete')되면서 클래스 'Alcohols'로 대체되었음을 보여준다. ■

SCM에서 연산은 구조 연산과 내용 연산의 두 가지 유형으로 나누어 진다. 구조 연산은 그래프의 구조를 조작하는 연산으로 명시적 관계를 변경하며, 내용 연산은 속성의 삽입 및 삭제, 구성원 자격함수의 변경, 서술의 변경 등 클래스의 내용을 수정한다. 이와 관련하여, 연산 전에 제한 조건이 만족되었다면 연산 후에도 제한

eff_ ver	chg_ type	chg_ ver	chngd_ code	chngd_ id	changed_title	eff_code	eff_ id	eff_title
1.00	add	0.00	0	0	-	12101500	417	Alcohol
2.09	delete	2.08	12101503	420	Unsaturated monohydric alcohol	12211500	417	Alcohols
2.09	delete	2.08	12101507	424	Methylated spirit	12211500	417	Alcohols
2.09	move	2.08	12101500	417	Alcohol	12211500	417	Alcohols

그림 5 UNSPSC의 버전 통제의 예

조건이 만족되어야 한다. 내용 연산은 편집 기능과 유사하므로, 구조 연산을 설명하기로 한다. 그림 6은 이들 연산을 설명하기 위한 것이다.

***Insert_Class*(C_p, C_{new})**: 새로운 클래스C_{new}가 C_p의 서브 클래스로 삽입된다. C_p의 구성원 자격함수는 C_{new}의 삽입에 따라서 변경되어야 한다.

***Insert_Edge*(C_p, C_c)**: C_p로부터 C_c로 새로운 간선이 삽입된다. 즉, 명시적 관계 $C_p \geq_T C_c$가 생성된다. 이들 간에 의미적 관계 $C_p \geq_S C_c$가 반드시 만족되어야 한다.

***Merge_Classes*(C_i, C_j)**: 이 연산은 기존의 두 클래스 C_i, C_j를 하나의 새로운 클래스로 병합하는 것이다. 구성원 자격함수는 논리합(OR)로 연결되고, 속성은 합집합 한다. 의미적 관계가 새로운 클래스에 전이되므로, 서브 클래스의 구성원 자격함수는 변하지 않는다.

***Generalize_Classes*(C_1, ···, C_m)**: 이 연산은 C_1, ···, C_m를 제거하지 않고 새로운 클래스 C_{new}를 이들 클래스의 부모로 생성한다. C_{new}는 C_1, ···, C_m의 가장 가까운 공통 조상의 서브 클래스로 삽입된다. C_{new}의 구성원 자격함수는 이들 클래스의 자격함수에 논리합(OR)을 적용하여 생성하고, C_{new}의 속성은 이들 클래스의 공통 부분과 부모 노드로부터 상속된 속성을 합하여 정의한다.

***Split_Class*(C_i, P_s)**: C_i는 P_s에 근거하여 두 클래스로 분리되어 C_i를 대체한다. 만일 C_i가 자식 클래스를 가지고 있으면, *Split_Class*는 P_s에 근거하여 자식 클래스를 반복적으로 분리한다.

***Delete_Subgraph*(C_{del})**: C_{del}와 그 자손들을 모두 삭제한다.

***Delete_Node*(C_{del})**: C_{del}를 그래프로부터 삭제하고 그 자손들은 삭제하지 않는다. 결과적으로 C_{del}의 부모로부터 C_{del}의 자식까지 간선으로 연결한다. 의미적 관계가 변하지 않기 때문에 구성원 자격함수를 변경할 필요는 없다.

UNSPSC에서는 분류체계에 대한 네 가지 유형의 연산이 있다. 분류체계를 수정하는 *add, delete, move*와 클래스명을 수정하는 *edit*이 그것이다. *add, delete, move*는 클래스 코드를 수정하여 클래스 간의 관계를 변경한다. *move*는 *delete*와 *add*로 구현 가능하다. *edit*을 제외한 UNSPSC의 나머지 연산은 SCM의 구조 연산으로 표현 가능하고 edit은 SCM의 내용 연산으로 수행 가능하기 때문에 SCM의 연산은 UNSPSC의 연산을 지원하기에 충분하다. 또한, SCM의 *split, merge* 연산은 drill-down, drill-up과 같은 분석적 처리에 유용하다.

4. 의미적 분류 모형(SCM)의 응용

SCM에서는 코드 기반 모형과 달리 상품의 분류와 무관하게 상품 식별이 가능하다. 클래스의 구성원 자격함수는 그 클래스의 구성원이 지녀야 할 특성을 명시한다. 상품의 특성이 시간에 따라 변하면 그 상품이 속한 클래스의 구성원 자격함수도 시간에 따라 달라진다. 이렇게 하여 상품의 식별은 분류와 무관하게 유지된다.

SCM은 2.1절에서 설명한 준거 사항 4,5가 요구하는 다른 뷰를 예 3.1과 같이 분류 기준을 명시적으로 정의하여 표현할 수 있다. 클래스 정의는, 코드를 재할당하거나 데이타베이스를 재구성하지 않고, 응용 또는 규제의 변화하는 관점을 반영하여 지속적으로 수정될 수 있다. 이것은 클래스 정의로부터 구현에 관한 상세 사항을 추상화하고 상품 식별 문제를 분류 문제와 분리함으로써 가능하다. 이러한 SCM은 기존 코드 기반 분류 체계

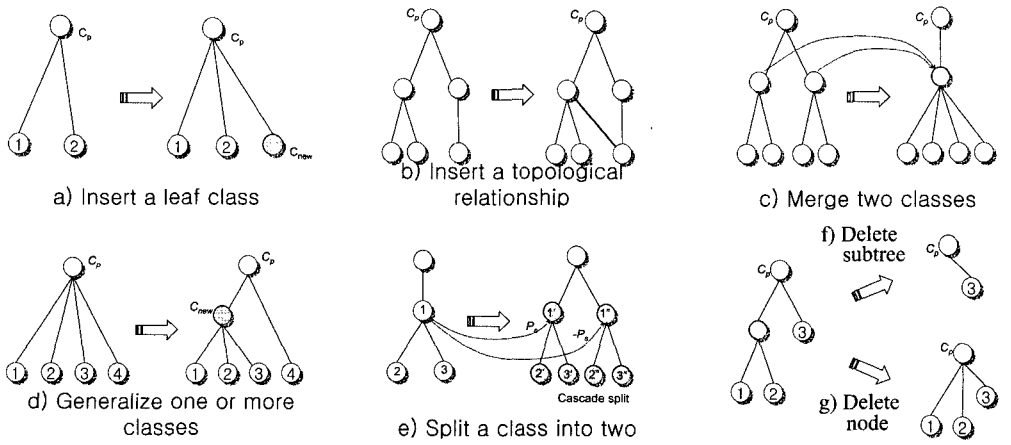

그림 6 분류체계에 대한 연산의 예

의 의미적 정의, 두 분류 체계 간의 매핑 및 병합 등에 응용될 수 있다.

4.1 의미적 분류 모형(SCM)을 이용한 코드 기반 분류 체계의 정의

SCM은 의미적 분류 체계뿐만 아니라 기존의 분류 표준을 정의하는 토대가 될 수 있다. UNSPSC와 같은 표준 분류 체계의 정의와 유지에 형식적인 모형을 사용하면, 그 표준은 더욱 정확하게 만들어 질 수 있다. SCM은 이러한 표준 분류 체계에 대한 의미적 정의를 용이하게 하며 결과적으로는 그 표준을 의미적으로 풍부하게 개선한다. 또한, 적절한 무결성 조건을 명세화하고 강제적으로 실행하여 분류 체계의 구조에 관한 제한 조건 이나 가정 사항을 더욱 명시적으로 만든다. 다음의 예는 SCM이 현재 널리 사용 중인 코드 기반 분류 체계를 표현할 수 있음을 보여준다. 또한, 이들 분류 체계가 의미적 분류 체계에 비해 의미가 부족함을 명확하게 보여준다.

SCM을 이용하여 그림 1에 있는 UNSPSC 분류 체계의 클래스 'Office supplies'와 'Writing paper'를 정의하면 다음과 같다.

$C_{Off\ supp}$ = < $D_{Off\ supp}$ = *"Office supplies"*,
$A_{Off\ supp} = A_0$,
$M_{Off\ supp}$ = (*class_code* = *4412**) >

$C_{wr\ paper}$ = < $D_{wr\ paper}$ = *"Writing paper"*,
$A_{wr\ paper} = A_0$
$M_{wr\ paper}$ = (*class_code* = *14111511*) >

여기서 A_0 = {*class_code*}이고, 뿌리 클래스 A_0를 위해 정의된 속성은 모든 클래스에 상속된다.

UNSPSC의 현재 버전에는 속성이나 구성원 자격 함수 등을 명시적으로 정의하지 않는다. 암시적으로 모든 상품이 자신이 속하고 있는 클래스의 식별자를 갖기 위해 *'class_code'*라는 속성을 가지고 있다는 것을 가정하고 있다. SCM에서는 뿌리 클래스 A_0에 속성을 할당하여 이러한 암시적 속성을 표현하고 모든 클래스로 상속시킨다. 그리고, 이 속성에 의존하여 클래스를 위한 구성원 자격함수를 구성한다. 이와 같이, SCM은 기존의 코드 분류 체계를 표현한다. 물론, SCM이 코드 분류 체계의 근본적인 한계를 극복해 주지는 못한다. 즉, *'class_code'*를 통해 'Writing paper'(14111511) 클래스와 'Office supplies'(4412) 클래스 간에 부분집합관계가 있다는 것을 알려 주지 못한다(이들 간에 의미적 관계가 있음에도 불구하고). 그 이유는 코드 기반 분류 체계가 명시적 관계 이외에 의미적 관계를 찾아 낼 수 있는 각 클래스의 의미라고는 모호해서 기계가 이해할 수 없는 클래스 이름 밖엔 없기 때문이다.

UNSPSC에 대한 제한 조건은 기본적 무결성 조건과 트리 제한 조건을 포함한다. 또한 말단 노드는 트리 구조에서 동일한 수준에 있어야 하고, 상품은 단 하나의 말단 노드에 속한다.

4.2 매핑

SCM은 이질적 전자 카탈로그의 통합에 관한 문제를 해결하는데 적절하다. 명시적인 속성과 무결성 조건, 구성원 자격함수 등의 의미로부터 이질적 분류체계 간에 클래스의 유사성을 추출하기 때문에, 정보의 손실은 상당히 감소한다. 마찬가지로 두 분류체계의 병합에 있어서도, 전문 지식을 가진 사용자의 도움없이 병합된 분류체계의 일관성이 유지된다.

의미적 분류체계 CL_1과 CL_2 간에 매핑을 정의할 때, 분류체계 CL_1의 클래스 C_{1i}과 분류체계 CL_2의 클래스 C_{2j} 간의 대응 관계를 속성 및 제한 조건으로 구성된 구성원 자격함수를 이용하여 정확하게 정의할 수 있다. 두 분류 체계 간의 매핑은 구성원 자격 함수 간의 공통적인 표현을 찾아 내는 과정이다. 그림 7은 두 분류체계 A와 B 간의 매핑을 SCM에 기반하여 어떻게 해결하는지 보여준다. 일대일 대응을 정의하는 기존의 크로스웍 테이블은 기껏해야 A의 'Film camera'는 B의 'High end camera'와 'Economy camera'에 대응될 수 밖에 없다. 그러나, 의미적 분류체계에서는 클래스에 정의된 구성원자격함수를 활용하여, A의 'Film camera' 중에서 제한 조건 'retailprice>1000'을 만족하는 상품 집합은 B의 'High end camera'로 매핑되고, 'Film camera'의 나머지 상품 집합은 B의 'Economy camera'로 매핑된다. 이렇게 하여 의미적 분류체계는 매핑에 관한 한 정보의 손실을 줄일 수 있다.

4.3 병합

그림 7의 두 분류 체계 A와 B를 병합한다고 가정하자. A의 'Camera'의 구성원 자격함수가 B의 'Camera'와 동일하기 때문에 B의 'Camera'에 대응되어 하나로 합병하면 된다. 그러나, 'Camera'의 서브 클래스 'Film camera'와 'Digital camera'는 'High end camera' 및 'Economy camera'와 구성원 자격함수에서 서로 공통되는 표현이 전혀 없다. 이것은 'Camera'의 서브 클래스를 나눌 때, 분류 체계 A의 분류 기준이 B의 기준과 전혀 다르다는 의미하며, 이들 클래스 간에 서로 배타적일 필요가 없다는 것을 의미한다. 이와 같이 비배타적인 클래스를 합병하는 경우, 이들 클래스와 수퍼 클래스 간의 관계를 병합된 분류체계에서 그대로 유지한다. 따라서, 병합된 분류체계에서 'Camera'의 서브 클래스는 'Film camera', 'Digital camera', 'High end camera', 'Economy camera'가 된다. 반면, 'Film camera'와 'Digital camera' 간에는 서로 배타적이며, 'High end camera'와 'Economy camera' 간에도 서로 배타적이다.

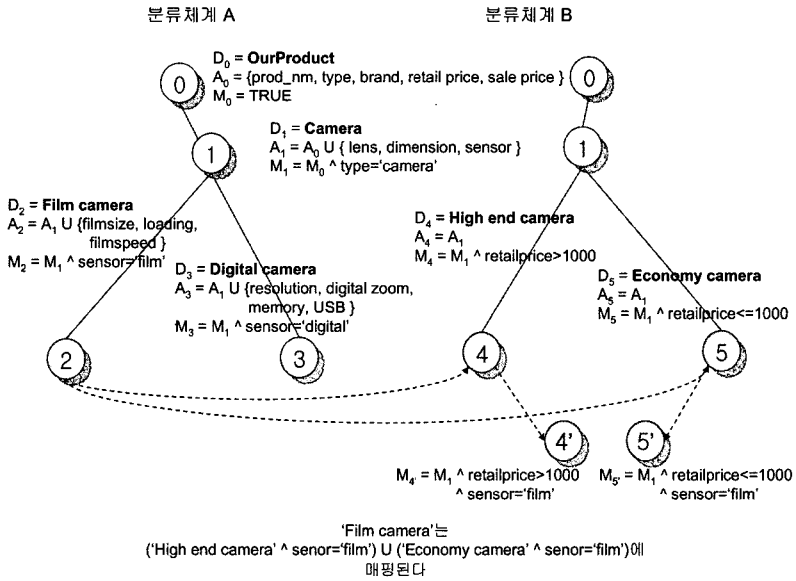

그림 7 의미적 분류체계의 매핑

이 정보 또한 병합된 분류체계에서 유지가 되어야 한다. 이를 위해 의미적 분류체계에서는 이들 클래스를 분리하여 서브 클래스를 그림 8과 같이 정의한다. 이렇게 하여 클래스 간의 배타성 및 비배타성을 유지하여 일관성을 훼손하지 않는다.

두 분류 체계의 올바른 병합에 관한 일반적인 합의를 찾기는 어려운 일이다. 이것은 응용에 따라 병합된 분류체계가 어떻게 활용될지에 따라 다양하다. 그러나, 우리는 올바른 병합을 최소한의 요구사항으로 다음과 같이 정의할 수 있다.

정의 4.1 CL_1과 CL_2을 병합될 분류 체계라 하고, CL을 두 분류 체계를 병합하여 새롭게 생성된 분류 체계라고 하자. 다음과 같은 조건을 만족한다면, CL은CL_1과 CL_2을 올바르게 병합한 것이다.

1. CL_1과 CL_2의 모든 클래스를 보존한다. 즉, 모든 클래스 C_i에 대해
   만일 $C_i \in CL_1$이거나 $C_i \in CL_2$ 이면, $C_i \in CL$이다.
2. CL_1과 CL_2의 모든 명시적 관계를 보존한다. 즉, 모

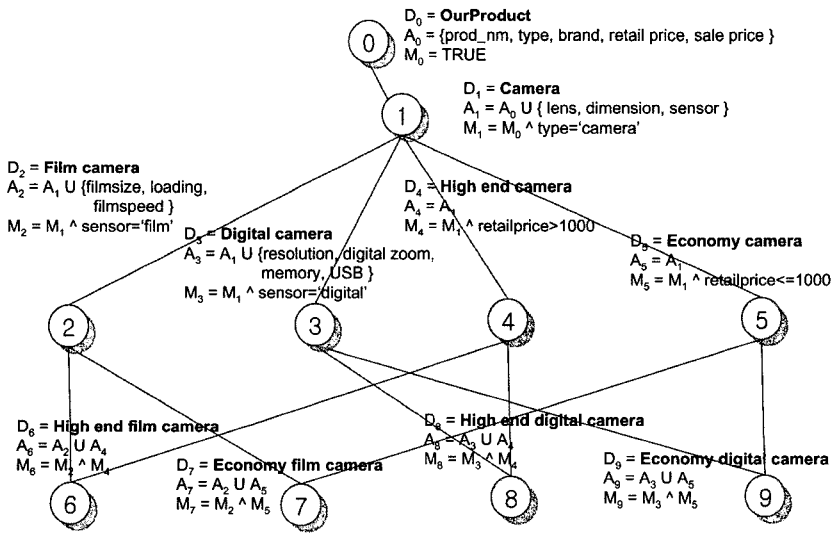

그림 8 의미적 분류체계의 병합

든 클래스C_i, $C_j \in CL_1$ 에 대해,
만일 $C_i \geq_T C_j$ 이 CL_1에서 성립하면, $C_i \geq_T C_j$은 CL에서도 성립한다.
CL_2에 있는 모든 관계에 대해서도 성립해야 한다.

3. 건전성을 보존한다. 즉, 모든 클래스 $C_i, C_j \in CL$에 대해 만일 $C_i \geq_T C_j$가 CL에서 성립하면 $C_i \geq_S C_j$ 도 항상 성립해야 한다. ■

5. 의미적 분류 모형(SCM)의 구현

우리는 SCM에 근거하여 카탈로그 관리 시스템[12]을 구현하였으며, 시스템의 핵심 요소는 의미적 분류 체계를 관리하는 Class Code Engine이다. 시스템의 나머지 구성요소가 자신의 기능을 수행하기 위해 분류체계를 사용하기 때문에, CCE는 이들에게 서비스를 공급하는 필수적인 구성요소다. 예를 들어, 새로운 상품 클래스를 정의할 때, CCE는 분류체계를 탐색하여 적절한 부모 상품 클래스를 선택하는 역할을 한다. CCE는 여러 분류 체계를 동시에 지원하여, 사용자의 관점에 적절한 분류체계를 제공한다.

그림 9는 CCE의 개념적 자료구조를 보여준다. ProductClass는 의미적 분류체계의 각 상품 클래스를 정의하기 위한 클래스로서, 클래스코드(classCode),. 클래스명(className), 분류체계식별자(classScheme) 등의 속성으로 이루어지며, ParentOf관계에 의해 상품 클래스 간의 부모 자식 관계를 표현한다. ProductAttribute는 상품의 속성의 집합으로, PropertyOf라는 관계를 통해 각 상품 클래스에 대한 속성의 할당이 이루어 진다. 이들 ProductClass와 ProductAttribute를 바탕으로 구성원 자격함수를 위한 MembershipPreidcates를 정의한다. 이렇게 하여 ProductClass는 ProductAttributes로부터 속성을 참조(PropertyOf)하고 MembershipPredicates으로부터 해당 분류의 소속 상품을 정의하는 구성원 자격함수를 참조(DefinedAs)한다. 이 때 MembershipPredicates은 ProductAttributes의 상품속성을 구성원 자격함수의 연산항(ruleTerm)으로 사용한다.

이 외에 ClassificationScheme은 하나 이상의 분류체계에 대한 정보를 가지고 있으며 ProductClass의 인스턴스들이 어떤 분류체계에 해당되는지를 알 수 있게 한다. UnitOfMeasure 클래스는 ProductAttributes의 속성 인스턴스들이 각각 다르게 사용할 수 있는 하나 이상의 단위형식을 참조하도록 사용된다. 뿐만 아니라 각 단위형식들 간의 환산이 가능하도록 같은 유형의 단위들끼리 서로 관계를 맺을 수 있다.

CCE에서는 분류체계 조작을 위해 ***Insert_Class, Delete_Class Delete, Split_Class, Merge_Class*** 의 네 가지 오퍼레이션을 제공하고 있다. 일반적인 그래프 오퍼레이션에서는 각 구조체의 노드와 관계 설정으로 이들 오퍼레이션이 동작하지만 SCM 기반의 CCE에서는 분류체계의 안전성(soundness)을 보장하기 위해 각 분류 간의 구성원 자격함수의 포함관계를 유지하도록 추가적인 동작이 필요하다. 따라서, 분류의 삽입, 삭제, 분할, 병합은, 다음과 같은 공통적인 알고리듬을 수행한다.

1. ProductClass의 인스턴스(분류체계의 상품 클래스)의

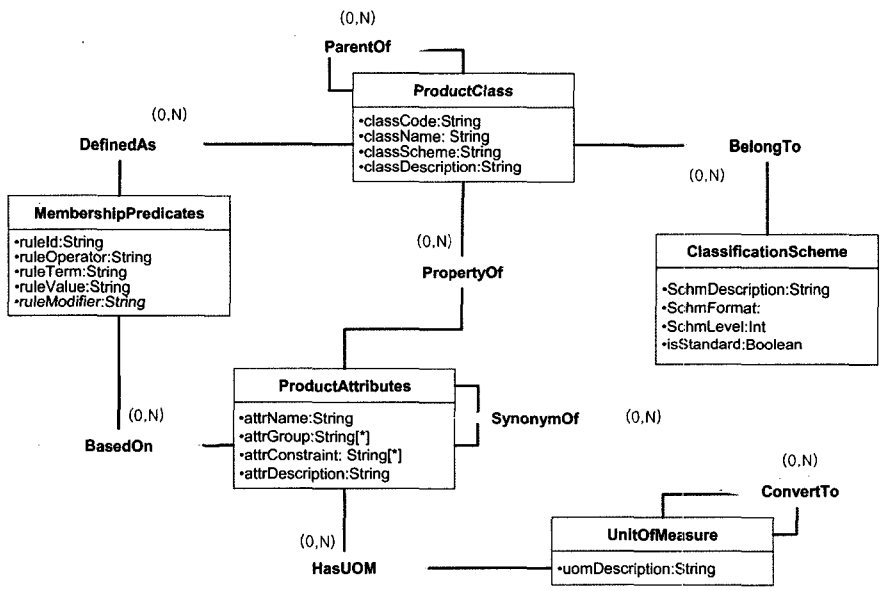

그림 9 CCE의 개념적 자료 구조

변경시, ProductClass의 인스턴스 간의 ParentOf 관계 변경

2. 1에 의해 적용 받는 ProductClass의 인스턴스(분류체계의 상품 클래스)에 정의되어 있는 MembershipPredicates의 변경과 ProductAttributes와의 PropertyOf 관계의 변경
3. 2의 변경 내용을 ProductClass의 인스턴스(분류체계의 상품 클래스)의 모든 하위 인스턴스(상품 클래스)에 상속

이와 같이 분류 간에 존재하는 계층관계의 변경 결과에 따라 구성원 자격함수의 내용이 변경되어야 하는데 ProductClass의 분류들 간의 계층관계(ParentOf) 변경과 더불어 이 변경내용이 그 하위의 모든 분류들로 상속되어야 한다. 이러한 구성원 자격함수와 속성집합의 상속을 통해 분류체계의 변경 이후에도 분류 간에 명시적으로 존재하는 계층관계(ParentOf)와 상품집합에 대한 선택술어로서 연산될 수 있는 구성원 자격함수 및 개별속성집합이 모순 없이 유지되어 안전성(soundness)을 보장한다.

이 시스템의 다른 구성 요소인 Modeler는 CCE에 의해 정의된 분류체계에 근거하여 상품 데이타베이스 스키마를 생성한다. 따라서, 사용자는 별도로 상품 데이타베이스를 설계하여 분류체계에 연관시켜야 하는 것을 염려하지 않아도 된다. 상품 데이타베이스 스키마에 대한 상세한 설명은 [13]을 참조하면 된다.

6. 결 론

SCM의 연구에 대한 동기로서 현재의 분류체계의 한계와 여러 측면을 분석하여 분류 모형을 위한 준거 사항을 식별하였으며, 이에 근거한 SCM을 제안하였다.

우리가 제안한 SCM은, 의미적 분류 체계뿐만 아니라, 기존 분류 체계의 토대로서 사용될 수 있다. 이 모형은 클래스에 대해 다중 상속, 형식적인 의미 표현, 속성 정의 등을 가능하게 한다. 이 모형으로 모든 분류체계가 만족시켜야 할 특성을 기본적인 무결성 조건으로 정의하고, 개별적인 분류체계의 특성에 맞도록 특정 제한 사항 및 조건을 부가적인 무결성 조건으로 정의할 수 있다. 이 모형은 코드나 구현에 따른 특성에 의존하지 않기 때문에 대부분의 환경에서 응용 가능하다. 분류체계 조작 연산을 모형의 의미에 대해 정의하였다. 분류기준을 형식적 명시적으로 표현하기 때문에 다중 뷰를 쉽게 지원한다.

우리는 SCM이 표준화 노력에 매우 중요한 역할을 한다고 믿는다. 상품 클래스를 정확하게 의미적으로 정의하여 표준에 대한 개인의 해석에 의한 편차를 줄일 수 있기 때문이다. 또한, 이 모형으로 정의된 분류체계는 분류체계 간의 의미적 매핑을 가능하게 하여 더욱 정확한 통합에도 기여한다. 이들 두 가지 응용은 기업 간의 통합과 상호운용성 측면에서 매우 중요하다.

SCM에 근거한 의미적 분류체계의 생성은 다소 복잡하고 많은 시간과 노력을 필요로 한다. 그러나 즉흥적인 모형에 의한 손실은 이러한 생성 비용을 능가할 수 있다. 이 모형을 통해 상품 분류 기구에서 사용하는 여러 가지 지침과 규칙을 일관성 있게 구성할 수 있다.

SCM은 상업용 카탈로그 관리 시스템[14]의 핵심 구성요소로 구현되었다. 그리고, 이 모형은 정부가 추진 중인 B2B 시범 사업에서 상품정보의 산업 표준을 위한 지침으로 활용되고 있다. 이 사업은 각 산업 분야를 대표하는 40개의 컨소시엄으로 구성되었으며 지난 4년간 지속되어 왔으며, 이 모형의 수정 버전은 상품 정보를 위한 표준 모형으로 채택되었다. 이 모형에 온톨로지 개념을 합쳐서 확장하는 것은 매우 중요하고 흥미있는 일이다. 조달청은 이 모형을 표준 상품 온톨로지를 위한 토대로 채택하였다. 우리는 현재 온톨로지의 설계 및 구형 중에 있으며, 온톨로지 관리 및 참조 시스템을 개발하고 있다.

참 고 문 헌

[1] Anant Jhingran, "Moving up the food chain: Supporting E-Commerce Applications on Databases," *SIGMOD Record* 29(4), 2000.

[2] Harmonized system committee, "Harmonized System Convention," available at http://www.wcoomd.org/ie/En/Topics_Issues/HarmonizedSystem/harmonizedsystem.htm, 2004.

[3] UNDP, "United Nations Standard Products and Service Code, White papr," available at http://www.unspsc.org/, 2001.

[4] Cologne Institute for Business Research, "eCl@ss - New Standardized Material and Service Classification," available at http://www.eclass-online.com/, 2004.

[5] Fensel, Omelayenko, Ding, Schulten, Botquin, Brown, Flett, "A Product Data Integration in B2B Electronic Commerce," *IEEE Intelligence System*, 16(3), 2001.

[6] Ellen Schulten, Hans Akkermans, Guy Botquin, Martin Dörr, Nicola Guarino, Nelson Lopes, Norman Sadeh, "The E-Commerce Product Classification Challenge," *IEEE Intelligence System*, 16(4), 2001.

[7] Jörg Leukel, Volker Schmitz, Frank-Dieter Dorloff, "A Modeling Approach for Product Classification Systems," *proc. of DEXA Workshops*, 2002.

[8] Rakesh Agrawal, Ramakrishnan Srikant, "On integrating catalogs," *proc. of WWW*, 2001.

[9] Hepp, M., "Measuring the Quality of Descriptive

Languages for Products and Services," in Dorloff., et al, eds. *E-Business - Standardisierung und Integration, Tagungsband zur Multikonferenz Wirtschafisinformatik 2004, Cuvillier, Gottingen, 2004*, 157-168.

[10] Lee, S.-g., "Design & Implementation of an e-Catalog Management System," Tutorial at DASFAA 2004, (Jeju Island, Korea, 2004).

[11] Guo, J., and Sun, C., "Context Representation, Transformation and Comparison for Ad Hoc Product Data Exchange," Proc. of ACM Symposium on Document Engineering, Grenoble, France, 2003, 121-130.

[12] Dongkyu Kim, Sang-goo Lee, Jonghoon Chun, Sangwook Park, Jaeyoung Oh, "Catalog Management in E-Commerce Systems," *proc. of Computer Science & Technology*, 2003.

[13] Kiryoong Kim, Dongkyu Kim, Jeuk Kim, Sangwook Park, Ighoon Lee, Sang-goo Lee, Jong-hoon Chun, "An Evaluation of Dynamic Electronic Catalog Models in Relational Database Systems," *Managing E-Commerce and Mobile Computing Technologies*, IRM press, 73-90, 2003.

[14] CoreLogix, "eCliX product overview," available via http://www.corelogix.co.kr , 2001.

김 동 규

서울대학교 계산통계학과 졸업(학사). 서울대학교 전산과학과 졸업(석사). 서울대학교 컴퓨터공학과 졸업(박사). 2001년 Georgetown University ISIS 연구원 2002년~현재 (주)프람트 연구소장. 관심분야는 데이타베이스, 전자상거래, 전자카탈로그, 온톨로지, 정보검색

이 상 구

서울대학교 계산통계학과(학사). Northwestern University 전산과학과(석사, 박사). 현재 서울대학교 컴퓨터공학부 교수. 2002년~현재 서울대 e-비즈니스 기술연구센터 센터장. 관심분야는 데이타베이스, 전자상거래, 전자카탈로그, CRM, 디지털 라이브러리

전 종 훈

University of Denver 전산과학과(학사) Northwestern University 전산과학과(석사, 박사). 현재 명지대학교 컴퓨터공학과 교수. 2001년~현재 (주)프람트 대표이사 사장. 관심분야는 전자상거래, 의료정보, CRM, 디지털 라이브러리

최 동 훈

서울대학교 계산통계학과 졸업(학사). 한국과학기술원 전산학과 졸업(석사). Northwestern University 전산학과 졸업(박사). 1983년~1986년 한국증권전산(주) 1989년~1992년 한국국방연구원 선임연구원. 1992년~1999년 동덕여자대학교 전산학과 부교수. 2004년~현재 한국과학기술원 전산학과 초빙교수. 관심분야는 데이타베이스, 시맨틱 웹