

상품 데이터베이스의 동적 특성을 지원하는 분류 모형

김 동 규[†] · 이 상 구^{**} · 최 동 훈^{***}

요 약

상품 분류체계는 상품 데이터베이스를 설계하는 토대이며 전자상거래에서 상품 정보의 관리 및 활용에 관한 거의 모든 면에서 중심적 역할을 한다. 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 eCI@ss와 같이 현재 널리 사용중인 분류체계는 이러한 동적인 특성에 대한 요구사항을 만족시키지 못한다. 상품 정보는 재료, 시간, 장소 등의 속성과 무결성 조건과 같은 많은 의미를 지니고 있다. 이 논문에서는 상품 데이터베이스의 동적 특성 및 이에 대한 기존 코드 기반 분류 체계의 한계점을 분석하고, [1]에서 제안된 의미적 분류 모형이 상품 데이터베이스의 동적 특성에 관한 요구사항을 만족시킨다는 것을 설명한다. 이 모형은 상품 클래스를 명시적이고 형식적으로 정의할 수 있는 수단을 제공하며, 상품 클래스 간의 관계를 그래프로 구성한다.

A Classification Model Supporting Dynamic Features of Product Databases

Dongkyu Kim[†] · Sang-goo Lee^{**} · Dong-Hoon Choi^{***}

ABSTRACT

A product classification scheme 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. 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 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 eCI@ss, however, have a lot of limitations to meet these requirements for dynamic features of classification. 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 proposed in [1], 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.

키워드 : 분류체계(Classification Scheme), 상품 데이터베이스(e-commerce), 의미적 분류 모형(Product Database), 의미적 분류 체계(Semantic Classification Model), 전자상거래(Semantic Classification Scheme)

1. 서 론

여러 사용자가 공유할 수 있는 상품 데이터베이스는 전자상거래에 필수적인 요소이다. 상품 정보 공급자는 상품 데이터베이스의 효율적 관리를 위해, 자신의 목적에 적합한 분류체계를 정의하고 이에 따라 상품을 분류하여 사용자에게 데이터베이스를 제공한다. 사용자는 분류체계를 상품 데이터베이스에 대한 탐색 경로로 이용하여 원하는 상품을 찾

는다. 분류체계는 이처럼 정보 공급자나 수요자에게 정보 관리와 활용의 중요한 근간이 된다.

상품 데이터베이스는 보통 다음과 같은 동적인 특성을 가지고 있다. 시장의 수요와 기술의 발달로 인해 신상품이 등장하고 기존 상품이 사라지는 것은 자주 발생하는 일이다. 이에 따른 상품데이터베이스에 대한 상품 정보의 추가, 삭제 및 갱신은 분류체계의 수정을 요구하며, 분류체계의 수정은 분류체계의 일관성의 유지를 요구한다.

상품 데이터베이스에 대해, 각 사용자는 자신의 관점으로 분류된 상품 정보를 원한다. 공급자가 제공하는 분류체계가 사용자의 관점과 다를 경우, 사용자는 원하는 상품을 탐색

[†] 정 회 원 : (주)코어로직스 연구소장
^{**} 종신회원 : 서울대학교 e-비즈니스 기술연구센터 센터장
^{***} 정 회 원 : 한국과학기술원 전산학과 초빙교수
 논문접수 : 2004년 10월 5일, 심사완료 : 2004년 12월 16일

하는데 불편을 느끼며 전자상거래 시스템에 비효율을 가져온다. 따라서 각 사용자에 따라 상품 데이터베이스의 분류 체계에 대한 다양한 뷰를 제공하는 것은 전자상거래의 효율과 함께 정보 활용에 매우 중요하다.

기업 간 상거래나 메타 쇼핑몰의 경우, 서로 다른 분류 체계에 의한 상품 정보의 제공은 서로 유사한 상품 간의 매핑을 필요로 한다. 또한, 서로 다른 분류 체계를 병합하여 통일된 분류체계를 생성해야 하는 경우도 있다. 분류 체계의 매핑 및 병합은 분류체계의 일관성을 유지하고 정보손실을 줄이는 것이 중요하다. 상품 데이터베이스가 이와 같은 동적 특성의 요구 사항을 신속하고 적은 비용으로 만족시키려면, 상품 정보 관리의 근간이 되는 분류 체계가 이들 요구 사항을 지원해야 한다.

분류 체계에서 이러한 동적 특성을 지원하기 위해 지금까지 많은 연구가 진행되어 왔다. B2B 에서 분류체계를 포함하여 상품 정보의 효율적인 통합 문제[2], 이질적인 분류 체계 간의 동시성을 해결할 때 나타나는 일관성 및 정보 손실 문제[3], 분류체계의 상호 호환을 위한 XML 데이터 모형 [4], 기존의 분류 체계에 상품을 자동으로 분류하는 문제 [5] 등은 이러한 문제를 해결하기 위한 노력을 보여준다. 이들 연구 결과는 상품 정보의 동적 특성을 지원하기 위한 해결책을 언급하고 있으나, 이들의 연구 대상은 Harmonized Commodity Coding System[6], UNSPSC[7], eCI@ss[8] 등의 단순한 형태의 코드 기반 분류 체계에 관한 것이었다. 따라서 이들 해결책은 코드 기반 분류 체계의 한계로 인해 상품 데이터베이스의 동적 특성을 지원하는데 여러 가지 한계점을 내포하고 있다. 이들 방법은 종종 많은 시간과 노력을 요구하는 상품 데이터베이스의 재구성을 필요로 하기 때문에 상품 정보의 일관성이나 전자상거래 시스템의 운용에 부정적인 영향을 미친다.

코드 기반 분류 체계는 클래스의 집합과 그들간의 관계 집합으로 구성되어 있다. 클래스는 클래스 코드와 클래스 설명으로 정의된다. 클래스 설명은 사람만이 이해할 수 있는 자연어 시술로 사람에 따라 다양한 해석이 가능하여 상당히 모호하다. 클래스 관계는 클래스 간의 부모-자식 관계를 나타내며, 클래스 계층구조는 트리를 형성한다. 클래스 코드는 부모의 클래스 코드를 접두사로 가지며, 상품 정보에 대한 아무런 의미도 함축하지 않는다.

본 논문에서는 상품 데이터베이스의 동적 특성 및 이에 대한 기존 코드 기반 분류 체계의 한계점을 분석하고, [1]에서 제안된 의미적 분류 모형이 상품 데이터베이스의 동적 특성에 관한 요구사항을 만족시킨다는 것을 설명한다. 의미적 분류 모형은 재료, 시간, 장소 등의 속성 및 무결성 조건 등과 같은 상품 정보에 고유한 의미에 기반한 모형으로, 전자 카탈로그 시스템에서 분류 체계를 관리 하기 위해 고안되었다. 의미적 분류 체계는 의미적 분류 모형의 인스턴스이고, 의미적 분류 모형은 의미적 분류 체계를 정의하기 위한 데이터 모형이다. 의미적 분류 모형은 상품 정보의 의미로부터 클래스의 속성 및 무결성 조건 등을 명시적 및 형식

적으로 정의하는 수단을 제공하며, 클래스 간의 관계도 의미를 통해 그래프 구조로 정의한다.

논문의 구성은 다음과 같다. 2절에서는 전자상거래 환경에서 상품 데이터베이스의 동적 특성에 대해 설명하고 기존의 코드 기반 분류 체계가 이들을 만족시키는 데 어떤 한계점이 있는지 분석한다. 3절에서는 의미적 분류 체계가 다루는 상품 정보의 의미와 특징을 설명하고, 이를 위한 의미적 분류 모형을 정의한다. 4절에서 의미적 분류 모형이 동적 특성의 요구사항을 어떻게 만족시키는지 제시한다. 5절에서 결론 및 향후 연구 방향을 설명한다.

2. 상품 데이터베이스의 동적 특성 및 기존 코드 기반 분류 체계의 한계점

분류 체계가 상품 데이터베이스의 동적 특성에 관한 요구사항에 대해 일관성을 유지하고 정보의 손실을 피하면서 신속하고 저렴하게 대응하는 것은 상품 정보의 관리에 매우 중요하다. 이 절에서는 동적 특성에 대한 요구사항을 설명하고, 이들 요구 사항을 만족시키지 못하는 eCI@ss와 UNSPSC와 같은 기존의 분류 체계의 한계점을 분석하기로 한다.

2.1 상품 데이터베이스의 동적 특성

2.1.1 상품 데이터베이스에 대한 다양한 뷰의 지원

상품 데이터베이스에 대한 사용자의 뷰는 상품 정보에 대한 사용자의 목적에 따라 매우 다양하다. 생산자는 상품에 대해 재료와 생산 관련 프로세스에 관심이 있으며, 판매자는 상품에 대해 크기와 무게 등에 관심이 있다. 이들 뷰를 사용자에게 제공하는 수단은 분류 체계이며, 사용자는 분류 체계를 통해 상품 정보를 활용한다. 뷰의 다양성은 요즘 사용되는 분류 체계가 다양하다는 것을 고려하면 명확하게 입증된다. 따라서, 하나의 상품 데이터베이스에 대해 각 사용자의 목적에 적합한 다양한 뷰를 분류 체계의 형태로 제공하는 것은 전자상거래의 효율과 편리성을 위해 매우 중요하다.

2.1.2 분류 체계의 진화

상품은 시장의 요구와 기술의 발전에 따라 지속적으로 변화하기 때문에 상품 분류 체계 역시 변화되고 진화되어야 하는 것이 필요하다. 분류 체계에 새로운 클래스가 삽입되거나 기존 클래스가 삭제될 때, 분류 체계의 일관성이 유지되어야 한다. 분류 체계의 일관성이란, 각 클래스의 정의에 일관되도록 모든 상품이 분류되어 있음을 의미한다. 변화하기 전의 분류 체계에서 서로 배타적으로 정의된 클래스들은 변화 후에도 서로 배타성을 유지해야 하며, 비배타적인 클래스들은 비배타적인 상태에 머물러야 한다. 여기서, 배타성이란 하나의 상품이 여러 클래스에 속할 수 없다는 것을 의미하며, 비배타적인 클래스 간에는 공통적인 상품을 가지고 있다. 따라서, 클래스의 삽입이나

삭제 시, 단순히 그 클래스만을 삽입하거나 삭제하는 것은 이러한 일관성을 해칠 수가 있으며, 삽입 및 삭제에 의해 영향을 받는 클래스를 식별하여 이들을 제정되어야 한다.

2.1.3 이질적 분류체계 간의 매핑

B2B 전자상거래나 한꺼번에 여러 B2C 사이트를 탐색해야 하는 메타 쇼핑몰의 경우, 이질적인 분류체계간의 상호 운용성은 매우 중요하다. 매핑은 어느 한 분류체계의 각 클래스에 대해 다른 분류체계에서 대응되는 클래스가 무엇인지를 유사성에 근거하여 정의하는 것을 말하며, 전자상거래에서는 이러한 목적으로 크로스워 테이블과 같은 매핑 테이블을 널리 사용한다[13, 14]. 이러한 매핑 테이블은 정보 손실을 최소화하는 것이 중요하다. 즉, 두 분류체계 사이에 의미적으로 합축된 모든 매핑 관계를 명시적으로 정의해야 한다.

2.1.4 이질적 분류체계의 병합

병합은 이질적인 두 분류체계를 하나로 통합하는 것이다. 이를 위해 어느 한쪽의 분류체계에 속한 각 클래스가 다른 분류체계의 어느 클래스에 대응되는지 결정해야 하며, 새로 생성된 분류체계는 병합된 두 분류체계에 대해 일관성을 유지해야 한다.

2.2 동적 특성의 지원에 대한 기존 분류체계의 한계점

현재 사용되고 있는 분류체계에는 산업 전반에 걸쳐 사용되는 Harmonized Commodity Coding System, UNSPSC, eCI@ss를 비롯하여 특정 산업에서 사용되는 IEC61630(전자), UNCPC(운송), SITC(무역), CPV(조달 계약) 등 여러 가지가 있다. 이들 표준은 대상 상품의 분류에 관한 고유 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(부모자식 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. 슈퍼 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.

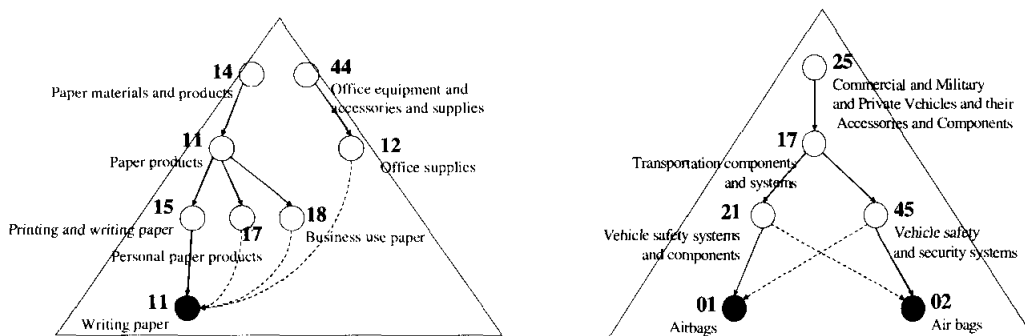
이 절에서는 코드 기반 분류체계가 동적 특성의 요구 사항을 만족시키는데 어떤 문제를 안고 있는지 분석하기로 한다.

2.2.1 코드 기반 분류체계에 근거한 상품 데이터베이스에 서 다양한 뷰의 지원

코드 기반 분류체계에 근거한 상품 데이터베이스는 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 상품을 한꺼번에 탐색할 수 없기 때문에, 사용자가 탐색하고자 하는 상품이 두 개 이상의 클래스에 분류되어 있을 경우, 분류 체계의 경로를 따라 올라 갔다가 다시 내려 와서 탐색해야 한다. 이것도 상품 정보의 활용 효율을 떨어뜨리는 원인이 된다.

(그림 1)은 UNSPSC의 일부분을 보여준다. 왼쪽 그림에서, 14-11-15-11 ‘Writing paper’는 14-11-15 ‘Printing and writing paper’의 서브 클래스로 명시되어(실선 표현 부분) 있으나, 의미적으로는(점선 표현 부분) ‘Personal paper products’, ‘Business use paper’ 또는 ‘Office supplies’의 서브 클래스가 될 수 있다(실제로, eCI@ss에서, ‘Writing paper’는 ‘Office supplies’의 서브 클래스로 되어 있다). 그러나, 코드 기반 분류 체계는 하나 이상의 부모 클래스를 허용하지 않기 때문에, 이러한 여러 가지 뷰를 지원하지 못한다.

오른쪽 그림에서 25-17-21-01 ‘Airbags’과 25-17-45-02 ‘Air bags’는 의미적으로(점선) 동일한 특성을 나타내는 상품 클래스이지만, 부모 클래스가 다르기 때문에 전혀 다른 것처럼 명시적으로(실선) 중복 정의되어 있다. 이러한 중복은 클래스 정의의 일관성을 저해하는 요인이 된다.



(그림 1) UNSPSC의 일부분

2.2.2 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 분류 체계 전문가만이 할 수 있다. 이것은 시장의 변화에 대해 신속한 적응을 요구하는 전자상거래 환경에 부적합하다.

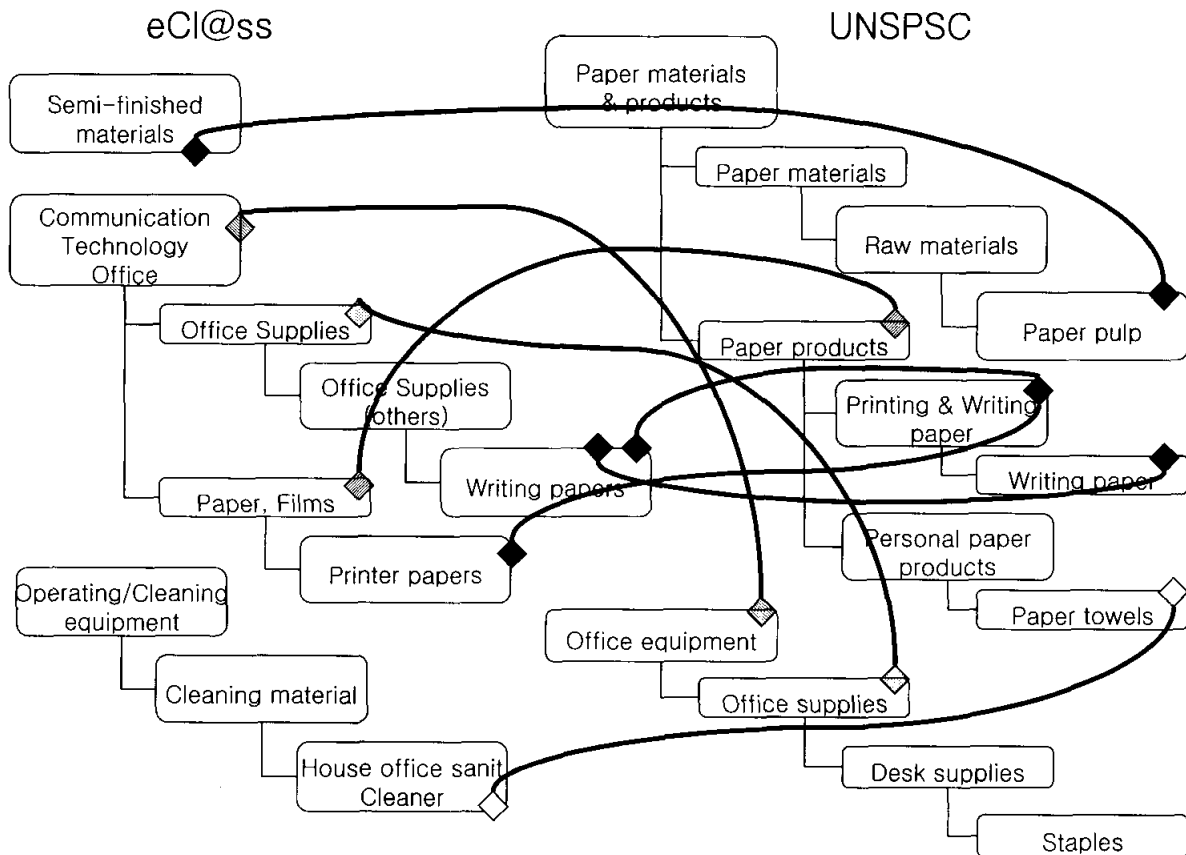
2.2.3 이질적 코드 기반 분류 체계 간의 매핑

전자상거래의 상호운용성을 위해, 국제표준인 UNSPSC와 eCI@ss 간에 정의된 매핑 테이블이 이미 널리 사용되고 있다[14]. 그러나, (그림 2)에서 볼 수 있듯이 이들 코드 기반 분류체계 간의 매핑은 정보 손실을 내포하고 있다[2]. 즉, 의미적으로는 대응되지만 매핑 테이블에 명시적으로 정의되지 않은 대응 관계가 존재한다.

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

2.2.4 이질적인 분류체계의 병합

코드 기반의 두 분류체계를 병합하면 2.2.2와 2.2.3절에서 이미 설명한 바와 같이 정보 손실이 발생하며 일관성 유지에도 문제가 있다. 이러한 문제의 해결은 분류 전문가의 도



(그림 2) UNSPSC와 eCI@ss 간의 매핑(일부분)

움을 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 데이터베이스의 동적 특성을 위한 요구사항을 만족시키기 위해, 상품 정보에 함축된 의미에 근거하여 분류체계를 명시적이고 형식적으로 정의하는 데이터 모형을 제안한다. 우리는 이것을 *의미적 분류모형*이라고 부른다. 앞에서 언급한 코드 기반의 분류체계는 이러한 의미를 명시적으로 정의하는 데이터 모형이 존재하지 않는다.

의미적 분류모형에 의해 정의된 의미적 분류 체계는 상품 정보를 추상화하여 추출된 클래스의 집합과 클래스 간의 관계로 구성된다. 클래스는 동일한 특성을 가진 상품의 집합이며, 상품 고유의 의미를 나타내는 속성 및 무결성 조건 등에 의해 명시적으로 정의된다. 클래스의 명시적인 정의는 클래스에 대한 의미의 모호성을 제거하는 동시에, 분류체계의 의미를 기계가 이해할 수 있도록 한다. 클래스 간의 관계는 IS-A 관계를 나타내며, 분류 계층구조는 그래프 구조를 형성한다.

3.1 의미적 분류 모형에서 의미의 표현

3.1.1 속성 및 무결성 조건

의미적 분류 모형은 분류체계 설계자에게 상품이 공유하는 정보의 의미를 더욱 많이 명시적으로 표현하기 위해 그 상품을 속성과 무결성 조건으로 정의할 수 있도록 한다. 그 중 어떤 속성은 속성값이 취할 수 있는 범위를 명시하는 제한조건을 정의하는데 쓰이기도 한다.

UNSPSC, eCI@ss와 같은 기존의 분류체계에서도 데이터 모형을 정의하지는 않지만 속성을 정의하기 위해 노력하고 있다. eCI@ss는 계층구조의 말단 클래스에만 제한적으로 정의된 속성을 제공하고 있고, UNSPSC는 속성을 제공하기 위해 작업 중에 있다. 이들의 속성 정의는 표준기구에 의해 제공되기는 하지만, 의미적 분류 모형만큼 자유롭지는 않다.

3.1.2 클래스 관계 및 계층 구조

부모-자식 클래스 간의 관계의 의미는 코드 기반 분류체계와 마찬가지로 IS-A 관계를 나타낸다. 이들의 계층구조는 자식 클래스가 여러 부모 클래스를 가질 수 있고 다중 상속을 지원하는 그래프 구조를 형성한다. 다중 상속은 사용자에게 분류체계에 대한 다양한 뷰를 지원할 수 있는 융통성을 제공하며, 질의에 주어진 특성을 만족하는 모든 클래스의 상품을 한번에 찾을 수 있는 효율성을 제공한다.

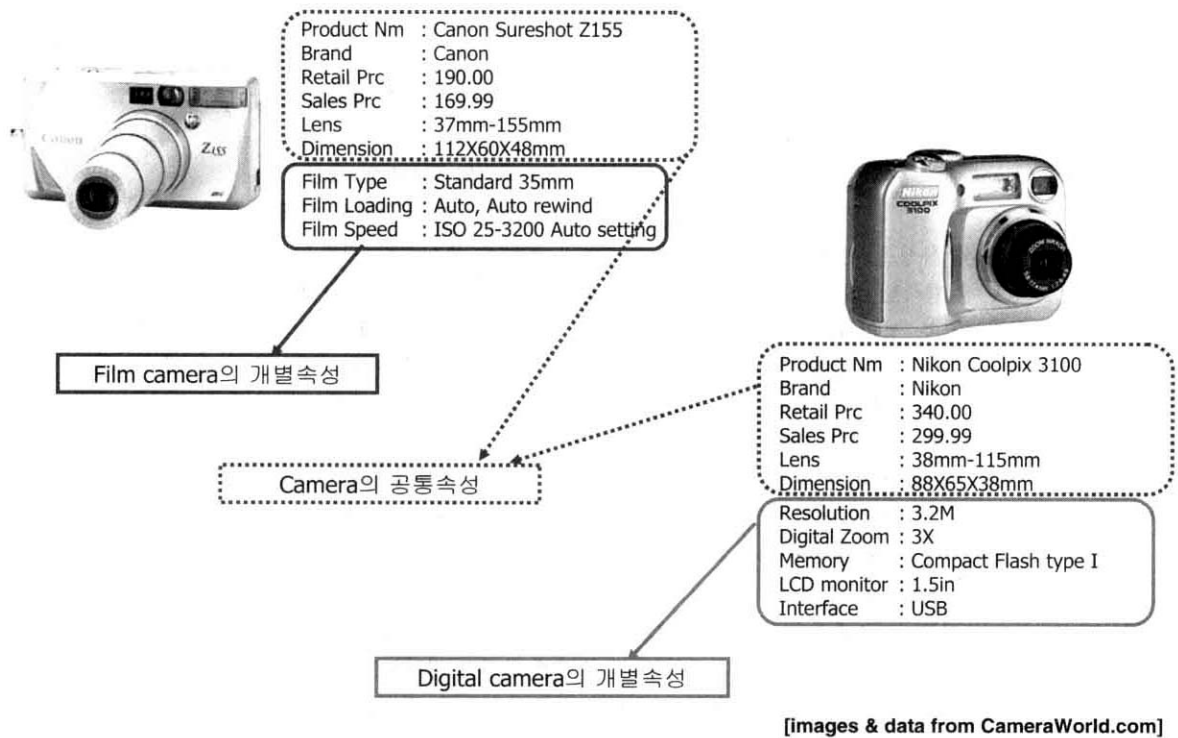
기존의 코드 기반 분류체계에서 클래스 계층 구조는 클래스 코드가 서로 다른 두 부모 클래스를 허용하지 않기 때문에 트리 구조에 제한될 수밖에 없다.

3.2 의미의 명시적인 표현

의미적 분류 모형에서 클래스는 속성 및 무결성 조건 등의 형태로 명시적인 의미로 정의되며 모호성이 없어서 기계가 이해할 수 있다. 이러한 명시적인 의미를 통한 클래스의 정의는 속성 및 무결성 조건을 통해 상품의 식별을 가능하게 하고, 동질성과 이질성을 구별 가능하게 하여, 기계가 각 상품이 어느 클래스에 속해야 할지 결정하는 데 사용될 수 있어 상품의 자동분류를 가능하게 한다(반면에, 코드 기반의 분류체계에서 사용되는 클래스 코드는 의미를 전달하지 못하며 단순히 부모-자식 간의 관계를 나타낼 뿐이다. 의미라고 할 수 있는 클래스 이름과 클래스 설명 등은 자연어로 모호하게 서술되어 있고 사람만이 이해할 수 있는 형태로 표현되어 있어서 컴퓨터 프로그램에서 활용하기엔 문제가 많다).

명시적으로 정의된 상품 정보의 공통적인 의미로부터 상품의 동질성을 확보하여 클래스를 정의하고, 의미의 차이로부터 추출되는 상품 간의 이질성에 근거하여 클래스 간의 관계를 정의한다. (그림 3)은 Cannon Sureshot Z155와 Nikon Coolpix 3100의 속성을 보여준다. Product Nm, Brand, Retail Prc, Sames Prc, Lens, Dimension 등의 속성은 클래스 'Camera'에 공통적이며 이들 상품의 동질성을 나타낸다. Cannon Sureshot Z155의 속성 Film Type, Film Loading, Film Speed는 'Film camera'에 특정된 것이고, Nikon Coolpix 3100의 속성 Resolution, Digital Zoom, Memory, LCD Monitor, Interface는 'Digital camera'에 특정된 것이다. 이러한 속성의 차이는 'Film Camera'와 'Digital camera' 간의 이질성을 나타낸다. 이와 같이 속성의 동질성과 이질성으로부터 각 클래스와 클래스 간의 부모-자식 간의 관계를 찾아 낼 수 있다. 즉, 공통적인 속성으로부터 클래스 'Camera'를 정의할 수 있고, 각 상품에 특정된 속성으로부터 'Camera'를 세분화한 클래스 'Film camera' 'Digital camera'를 정의할 수 있다. 'Film camera'와 'Digital camera'는 'Camera'를 세분화한 것으로 서브 클래스가 된다. 이렇게 하여 속성으로부터 클래스를 정의하고 이들 간의 관계를 정의한다.

클래스의 의미는 속성에 제한되지 않고 무결성 조건에 의해서도 정의된다. 위의 예에서 새로운 클래스를 조건을 추가하여 정의할 수 있다. 'High end camera'는 Retail Prc > 1000.00이라는 조건을 추가하여 정의할 수 있다. 이들 조건은 클래스의 속성, 비교연산자(<, >, = 등), 속성 값의 형태를 취한다.



[images & data from CameraWorld.com]

(그림 3) 카메라의 명시적인 속성

3.3 의미적 분류 모형의 정의

의미적 분류 모형[1]은 의미적 분류체계, 이에 대한 연산, 무결성 조건 등으로 구성된다. 이 절에서는 의미적 모형의 정의 중에서 의미적 분류 체계, 무결성 조건을 소개하고, 기존의 코드 기반 분류 체계가 의미적 모형에 의해 어떻게 표현될 수 있는지, 의미의 표현에 어떤 차이가 있는지 예를 들어 보여준다. 또한 의미적 분류 모형의 특징으로, 클래스 간의 명시적 관계 이외에 의미적 관계를 의미에 근거하여 찾을 수 있다는 것을 설명한다.

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 상품이 들어오고 나가고 할 때 클래스의 정의를 지속적으로 수정해야 하기 때문에 분류체계의 정의 및 관리를 위해 좋은 방법이 아니다. 대신에 클래스를 정의하기 위해 클래스의 구성원의 특성을 명세화하여 내포(intension)를 정의하는 방법을 사용한다. 예를 들어, 클래스는 클래스 이름, 클래스에 대한 간략한 설명, 논리적 술어 공식 등에 의해 정의된다.

의미적 분류 체계는 상품 정보의 의미를 모형화한 것으로, 상품을 클래스로 분류하고 클래스 간의 관계를 계층구조로 표현한다. 클래스는 공통 특성을 공유하는 상품의 집합을 의미하며, 각 상품에 대한 특성을 나타내는 속성, 이들 속성에 대한 조건인 구성원 자격함수, 무결성 조건 등으로 정의된다. 클래스는 조건이 더욱 한정된 조건 및 속성

을 갖는 서브 클래스를 하나 이상 가질 수 있다. 클래스 간의 관계 집합은 방향성의 비순환 그래프(Directed Acyclic Graph)를 형성한다. 이 분류 체계에서 클래스 코드는 단순히 기계적으로 할당된 것이 아니라 상품 정보의 의미를 반영한 것이다.

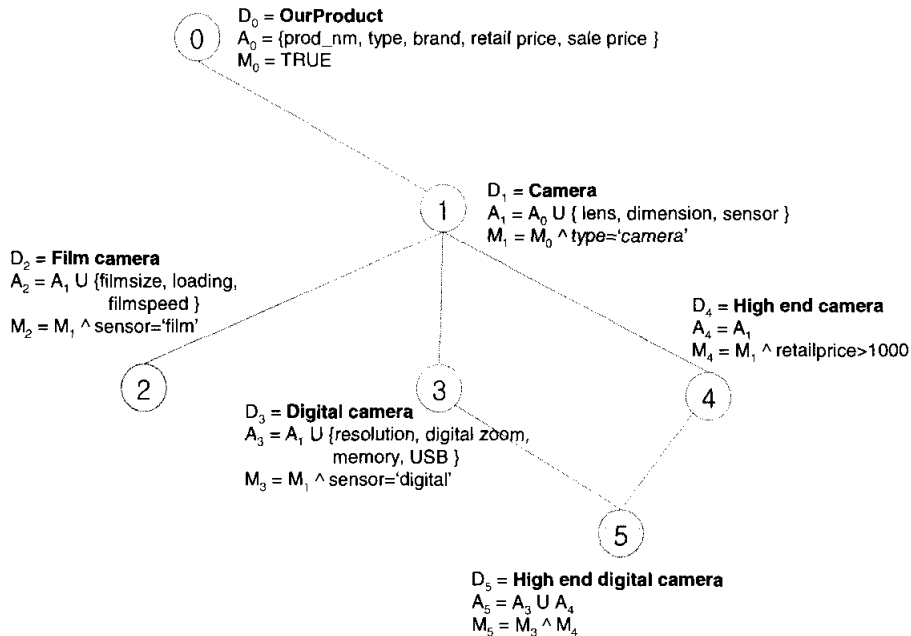
[정의 3.1] 상품 집합 S에 대한 의미적 분류 체계는 5-튜플 $\langle S, C, C_0, E, IC \rangle$ 이며, 여기서 S는 분류 대상인 모든 상품들의 집합이고, $C = \{C_0, C_1, \dots, C_n\}$ 는 클래스의 집합으로, 각 C_i 는 3-튜플 $\langle D_i, A_i, M_i \rangle$ 이다. D_i 는 클래스 이름과 간략한 문장의 설명이고, A_i 는 C_i 의 모든 구성원에 공통적인 속성들의 집합이고, M_i 는 A_i 에 대해 논리적으로 정의된 구성원 자격함수이다. $C_0 \in C$ 는 S의 모든 구성원을 포함하는 뿌리 클래스이다. $E \subseteq C \times C$ 는 C의 클래스 간의 부모-자식 관계를 표현하는 간선의 집합이다. IC 는 의미적 분류 체계에 정의된 무결성 조건의 집합이다. ■

[정의 3.2] 무결성 조건의 집합 IC는 분류 체계에 따라 서로 다를 수 있다. 어떤 체계는 다중 상속을 허용하는 반면, 다른 분류 체계는 그것을 허용하지 않는다. 그

러나 분류체계가 실용적이기 위해서는, 다음과 같은 기본적인 무결성 조건을 만족해야 한다.

- 1) $M_0 = \text{TRUE}$ (모든 상품에 의해 만족되는 것으로, C_0 의 정의에 따르면 만족 여부를 알 수 있다.)

- 2) E (부모-자식 관계)는 순환을 형성하지 않는다.
- 3) 만일 $\langle C_i, C_j \rangle \in E$ 즉, C_i 가 C_j 의 부모라면, $A_i \subseteq A_j$ 이다.
- 4) 만일 $\langle C_i, C_j \rangle \in E$ 라면, M_j 는 M_i 를 함축한다. 즉, $C_j \subseteq C_i$ 이다.



(그림 4) 의미적 분류 체계

[예 3.1] (그림 4)는 의미적 분류 체계의 예를 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 retailprice의 값에 따라 분류된다. 의미적 정의를 통해 'High end camera'가 다른 두 형제 클래스의 분류 기준과 서로 다르다는 것을 명확하게 알 수 있다. 'High end digital camera'는 'Digital camera'와 'High end camera'의 서브 클래스이다. 양쪽 부모 클래스의 속성은 자식 클래스에게 상속되며 구성된 자격 함수는 양쪽 부모의 함수를 논리곱(\wedge)으로 연결한 것이다. ■

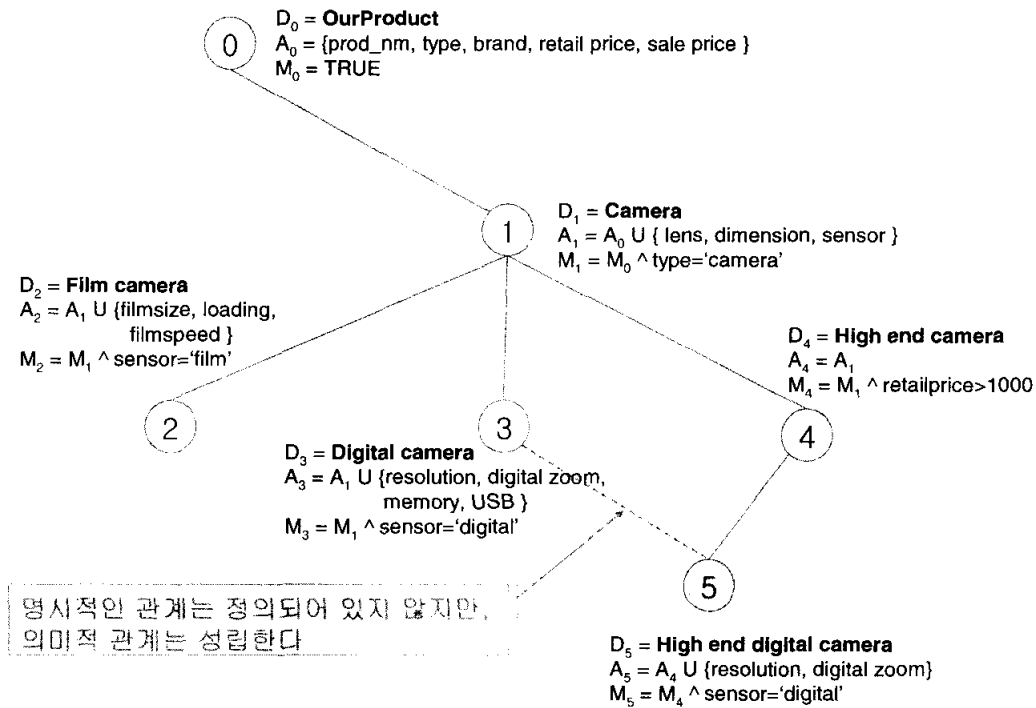
다음의 예는 의미적 분류모형이 현재의 코드 기반 분류 체계를 어떻게 표현할 수 있는지 보여준다. 또한, 코드 분류 체계가 의미적 분류체계보다 의미가 부족함을 명확하게 보여준다.

[예 3.2] (그림 1)에 있는 UNSPSC 분류 체계에서 두 개의 클래스 'Office supplies'와 'Writing paper'는 다음과 같이 정의된다.

$C_{\text{off_supp}} = \langle D_{\text{off_supp}} = \text{"Office supplies"} \rangle$
 $A_{\text{off_supp}} = \emptyset$
 $M_{\text{off_supp}} = (\text{class_code} = 4412^*) \rangle$
 $C_{\text{wr_paper}} = \langle D_{\text{wr_paper}} = \text{"Writing paper"} \rangle$
 $A_{\text{wr_paper}} = \emptyset$
 $M_{\text{wr_paper}} = (\text{class_code} = 14111511) \rangle$

UNSPSC의 현재 버전에는 속성이나 구성원 자격 함수 등을 명시적으로 정의하지 않는다. 암시적으로 각 상품이 자신이 속하고 있는 클래스의 식별자를 갖기 위해 class_code라는 속성을 가지고 있다는 것을 가정하고 있다. 따라서 이 예에서 보여준 것처럼 클래스를 위한 구성된 자격 함수를 만들 때 이러한 암시적인 속성에 의존해야 한다. 이들 코드는 'Writing paper'(14111511)와 'Office supplies'(4412) 간에 부분집합 관계가 있다는 것을 시스템에 알려주지 못한다. 각 클래스의 의미를 이해 할 수 있는 거라고는 모호하지만 클래스 이름밖엔 없다. ■

의미적 분류체계는 그래프로 표현되며, 노드는 클래스(C)를 나타내고 간선은 부모-자식 관계(E)를 나타낸다. 의미적 분류체계에서 부모-자식 관계는 두 가지의 형식적 관계로 정의한다.



(그림 5) 의미적 관계의 예

[정의 3.2] 의미적 분류체계 $CL = \langle S, \{C_0, C_1, \dots, C_n\}, C_0, E, IC \rangle$ 에 대해,

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

만일 CL 에서 C_i 가 C_j 의 조상이면, $C_i \geq_T C_j$ 이다.

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

만일 M_j 가 M_i 를 함축하면(imply), $C_i \geq_S C_j$ 이다. ■

명시적 관계란 분류체계 그래프에서 클래스 간의 경로가 명시적으로 표현되어 있는 것을 의미한다. 이들 관계는 코드 기반의 분류체계의 클래스 코드처럼 설계자의 명시적인 의도를 나타낸다. 반면에 의미적 관계란 설계자의 의도와 무관하게 클래스 간에 포함(subsumption) 관계를 만족하는 것을 의미한다.

[예 3.3] 예 3.1의 분류체계에서 설계자가 클래스 5('High end digital camera')를 정의할 때 클래스 3('Digital camera')를 인식하지 못했다고 가정하자. 이 경우, 클래스5는 클래스4의 서브 클래스로 정의되고 속성 sensor에 대한 조건식 $\text{sensor}='digital'$ 이 구성원 자격함수에 추가(\wedge)된다 ((그림 5) 참조). $M_5 = M_1 \wedge \text{type}='camera' \wedge \text{retailprice} > 1000 \wedge \text{sensor}='digital'$ 이고 $M_3 =$

$M_1 \wedge \text{sensor}='digital'$ 이므로, M_5 가 M_3 를 함축한다. 따라서, $C_5 \geq_S C_3$ 이다. ■

4. 의미적 분류 모형의 동적 특성의 지원

의미적 분류 모형은 2절에서 제시한 동적 특성을 위한 요구사항을 만족시킨다. 이 절에서는 이들 각 요구사항을 의미적 분류 모형이 어떻게 수용하는지 설명한다.

4.1 다양한 사용자 뷰의 제공

기존의 분류체계에서 상품의 식별은 분류체계의 클래스 코드와 밀접한 관계가 있으며, 상품의 분류가 바뀌면 클래스 코드가 바뀌어 상품의 식별자도 바뀐다. 반면, 의미적 분류체계는 클래스의 구성원 자격함수를 통해 상품의 분류와 무관하게 상품을 식별할 수 있다. 클래스의 구성원 자격함수는 그 클래스의 구성원이 만족해야 할 조건을 명시한다. 시간이 지남에 따라 상품의 분류가 변하게 마련인데, 이 때 구성원 자격함수를 갱신하여 분류의 변화를 반영한다. 이렇게 하여 상품의 식별을 분류와 무관하게 유지한다. 또한, 구성원 자격함수를 통해 명시적 관계 이외에 의미적 관계도 도출할 수 있다. 상품 분류와 식별의 무관성 및 의미적 관계의 도출 등은 사용자의 관점에 따른 다양한 뷰를 제공하는 토대가 된다. 즉, 상품에 대한 사용자의 관점에 따라 다양한 뷰를 정의할 때, 상품의 식별자는 그대로 두고 클래스의 구성원 자격함수를 각기 다르게 정의함으로써 동일한 상품 집합에 대해 사용자의 관점에 따라 클래스와 클

래스 간의 관계를 정의하여 다양한 뷰를 제공한다.

4.2 분류체계의 변경 및 진화

의미적 분류체계는 클래스 코드를 재할당하거나 상품 데이터베이스를 재구성하지 않고서도 클래스 정의를 지속적으로 수정하면서 사용자의 변화하는 관점을 수용한다. 분류체계에 새로운 클래스를 삽입하기 전, 분류체계의 일관성 유지를 위해 사용자는 새로운 클래스에 의해 영향을 받는 클래스를 식별하여 의미를 재정의해야 한다. 기존의 분류체계에서는 이러한 클래스의 식별은 분류 설계자에 의해서만 가능하였으나, 의미적 분류체계에서는 속성과 무결성 조건과 같은 클래스의 의미를 토대로 하여 기계에 의해 자동으로 식별된다. 분류체계 설계자는 식별된 클래스의 의미를 재정의하기만 하면 된다.

새로운 클래스의 삽입에 의해 영향을 받는 클래스의 식별은 구성원 자격함수의 포함관계에 의존한다. 우선 새로운 클래스의 부모 클래스를 결정해야 하는데, 삽입된 클래스의 구성원 자격함수 M_{new} 와 분류체계의 모든 클래스의 구성원 자격함수 M_i ($i = 1, \dots, n$)를 비교하여 M_{new} 를 완전히 포함하면서 가장 최소한의 요소(factor)로 구성된 M_p 를 갖는 클래스 C_p 를 부모로 한다. C_p 의 자식 클래스 중에서 M_{new} 와 공통 부분이 있는 클래스는 새로운 클래스와 배타적이어야 하므로 모두 제외해야 할 후보 클래스가 된다. 이 중에서 어느 클래스를 재정의할 것인가 결정하는 것은 분류체계 설계자의 몫이다. 이렇게 하여 일관성을 유지하면서 변경 및 진화를 가능하게 한다.

4.3 이질적인 의미적 분류 체계의 통합

의미적 분류모형은 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 도움 없이 병합된 분류체계의 일관성이 유지된다.

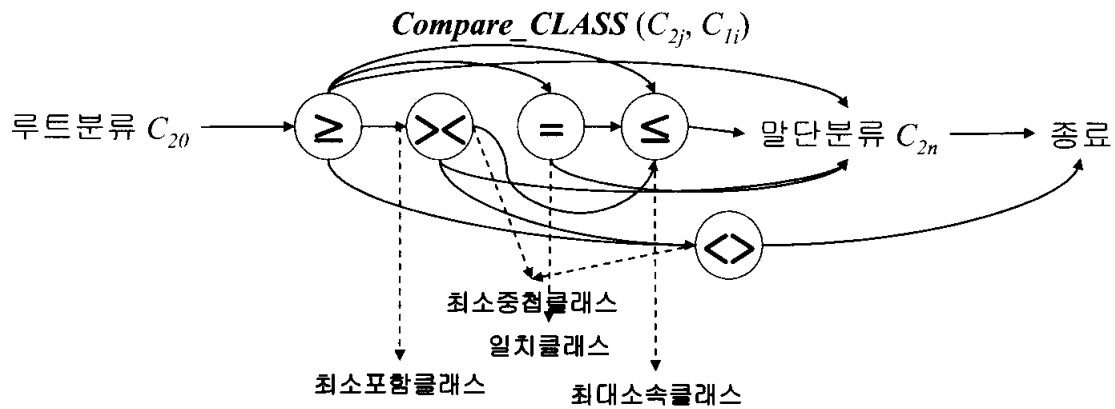
4.3.1 매핑

의미적 분류체계 CL_1 과 CL_2 간에 매핑을 정의할 때, 분류체계 CL_1 의 클래스 C_{1i} 와 분류체계 CL_2 의 클래스 C_{2j} 간의 대응 관계를 속성, 제한 조건, 구성원 자격함수를 이용하여 정확하게 정의할 수 있다. 이러한 클래스 간 대응 관계는 각 클래스의 구성원 자격함수 비교를 통하여 수행되는데, 이러한 역할을 수행하는 함수로 **Compare_CLASS** (C_{2j}, C_{1i})를 정의한다. Compare_CLASS는 두 클래스 간의 구성원 자격함수를 비교하여 다음과 같은 5가지 비교결과를 출력한다.

- 포함 ($C_{2j} > C_{1i}$) : 클래스 C_{2j} 에 해당되는 상품집합이 클래스 C_{1i} 의 모든 상품을 포함하는 경우
- 중첩 ($C_{2j} >< C_{1i}$) : 클래스 C_{2j} 에 해당되는 상품집합과 클래스 C_{1i} 의 상품집합이 서로 중첩되는 경우
- 일치 ($C_{2j} = C_{1i}$) : 클래스 C_{2j} 의 상품집합과 클래스 C_{1i} 의 상품집합이 정확히 일치하는 경우
- 소속 ($C_{2j} < C_{1i}$) : 클래스 C_{2j} 의 상품집합이 클래스 C_{1i} 의 상품집합에 소속되는 경우
- 배타 ($C_{2j} <> C_{1i}$) : 클래스 C_{2j} 의 상품집합과 클래스 C_{1i} 의 상품집합에 공통상품이 없는 경우

위와 같은 비교결과는 구성원 자격함수의 논리비교만으로 수행이 가능하다.

따라서, 의미적 분류 모형을 활용한 매핑은 다음과 같은 과정을 거친다. (1) Compare_CLASS을 이용해 매핑하려고 하는 클래스 C_{1i} 를 매핑의 대상이 되는 분류체계 CL_2 에 존재하는 모든 클래스경로에 대해 수행하여 (2) 일치하는 클래스를 찾거나, 일치하는 클래스가 없다면 포함하는 클래스, 소속하는 클래스, 또는 중첩하는 클래스를 찾아 (3) 클래스 C_{1i} 의 상품집합이 정확하게 선택될 수 있도록 각각 경우에 따라 설정하게 된다. 이 때, 하나의 분류경로 상에 포함, 소속, 중첩클래스가 둘 이상이 발견될 수 있는데, 이 경우에는 클레



(그림 6) 클래스 간의 포함관계 탐색

스 C_i 와 가장 유사한 클래스를 찾는 것이 목적이므로 (4) 포함하는 클래스와 중첩하는 클래스 중에서는 최소상품집합을 갖는 클래스를 선택하고, 소속되는 클래스 중에서는 최대상품집합을 갖는 클래스를 선택하면 된다. 이러한 과정을 모든 클래스에 대해 반복함으로써 전체 분류체계의 매핑을 수행한다.

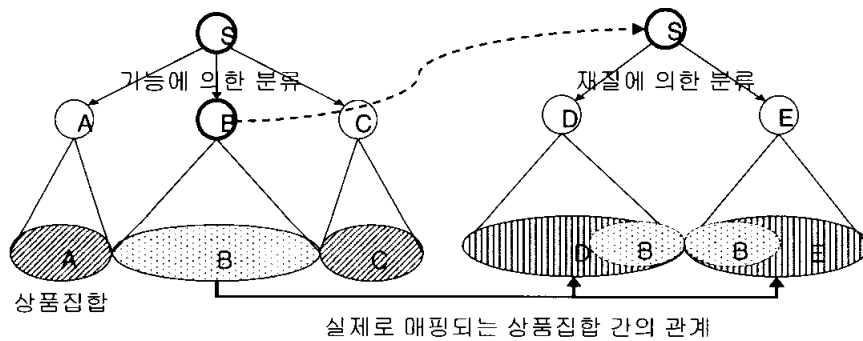
(그림 6)은 C_{2j} 가 분류경로 C_{20} 에서 C_{2n} 사이에 있을 때 **Compare_CLASS**(C_{2j} , C_{li})를 통해 발생할 수 있는 모든 경우의 수를 그래프로 표현한 것이다. 결국 의미적 분류체계의 매핑은 분류경로의 탐색에서 발견되는 모든 경우의 수에서 **Compare_CLASS**를 수행한 후 최소포함클래스, 최소중첩클래스, 일치클래스, 최대소속클래스를 찾아 이들의 조합으로 매핑결과를 결정한다.

의미적 분류체계에 의한 매핑은 크로스워 테이블 매핑을 두 가지 측면에서 개선한다.

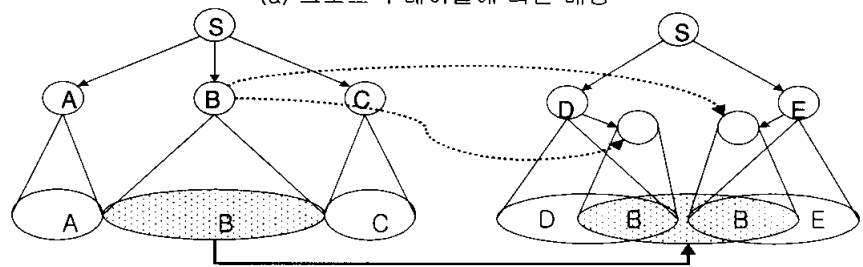
첫째, 매핑하려는 원천분류체계의 어떤 클래스 C_s 에 대해 매핑되는 대상분류체계의 클래스들을 탐색하여 클래스 C_s 와 의미적으로 일치하거나 최소로 포함하는 클래스 C_T 를 찾을 수 있다. 이는 의미적으로 포함되는 관계의 탐색이 가능함을 뜻한다. 이 측면만으로도 기존의 분류체계 매핑에 비해 정확한 클래스 매핑이 가능하며 적어도 안전한 분류체계 매핑을 보장한다. (그림 7)의 (a)는 이러한 클래스의 의미적 포함관계 탐색을 이용하여 안전한 매핑

을 할 수 있음을 보여준다. 하지만, 이것만으로는 정확한 매핑을 보장할 수는 없다. 그림에서 보여지듯이 클래스 A, B, C, D, E 는 각각 상품집합 A, B, C, D, E 를 나타내고 있는데, 이렇게 동일한 상품집합에 대한 서로 다른 두 분류체계가 있다고 할 때, 한 쪽은 상품의 기능에 의해 A, B, C 세 가지 클래스로 나누어져 있고, 다른 한 쪽은 재질에 의해 D, E 두 가지 클래스로 나누어져 있을 때, 클래스 B 와 정확히 매핑될 수 있는 클래스는 존재하지 않는다. 따라서 의미적으로 클래스 B 를 포함하는 대상클래스를 탐색하게 되면 클래스 S 만이 조건을 만족함을 알 수 있다. 결국 클래스 B 가 클래스 S 에 매핑됨으로써 안전한 매핑은 보장되지만 실제로 매핑되는 상품집합은 원하는 상품집합 B 를 포함하는 $D \cup E$ 가 되어 정확한 매핑을 하지 못하는 결과를 가져온다. 따라서 보다 정확한 매핑을 위해 추가적인 작업이 필요하다.

둘째, 의미적으로 포함하는 클래스에 매핑하는 동시에 매핑되는 클래스에 제약조건을 주어 의미적으로 일치하게 할 수 있다. (그림 7)의 (b)를 보면 클래스 B 의 매핑이 단순히 1:1로 클래스 B 를 의미적으로 포함하는 클래스로 매핑된 것은 아님을 알 수 있다. 클래스 B 와 의미적으로 중첩관계에 있는 클래스 D 와 클래스 E 를 탐색한 후 의미적으로 중첩된 부분만을 선택하기 위해 각각에 대해 구성된 자격함수를 적용한 후 양 클래스를 합집합하였다. 결국 안전한 매핑과 더불어 정확한 매핑을 하기 위



(a) 크로스워 테이블에 의한 매핑



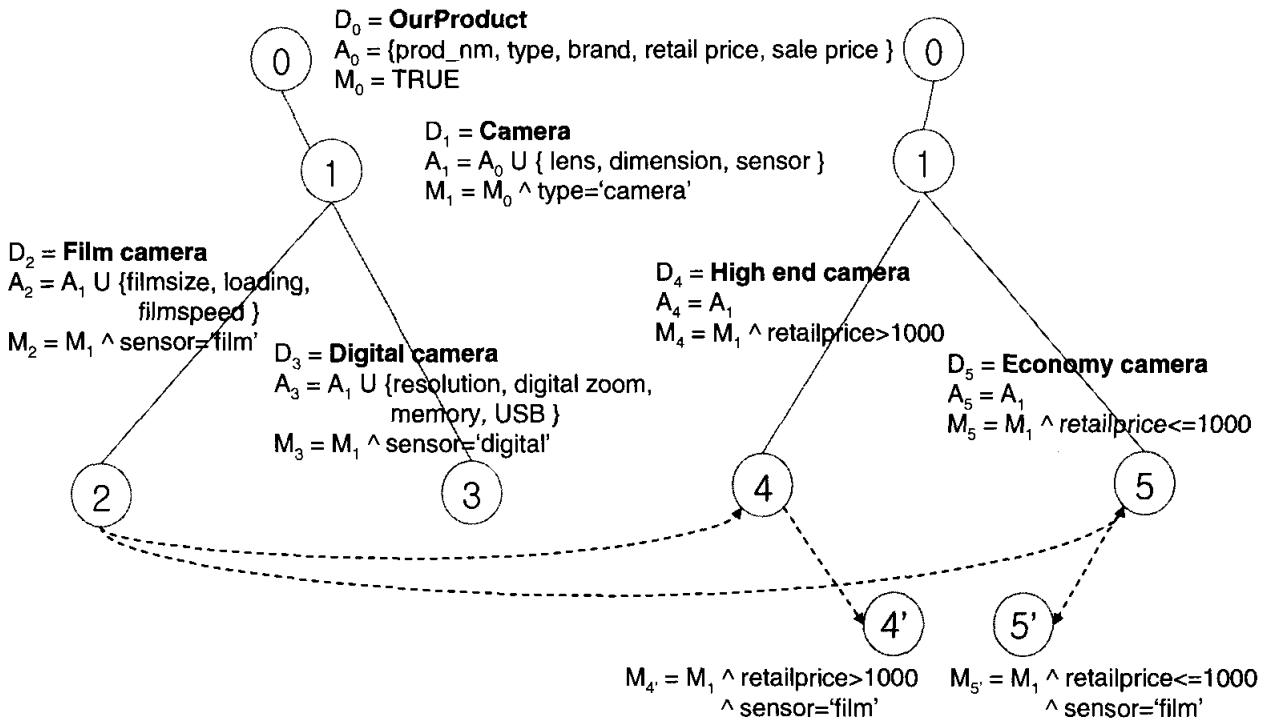
$$(D \text{의 구성원자격함수} \wedge B \text{의 구성원자격함수}) \cup (E \text{의 구성원자격함수} \wedge B \text{의 구성원자격함수})$$

(b) 의미적 클래스 매핑

(그림 7) 의미적 분류 모형을 이용한 매핑

분류체계 A

분류체계 B



'Film camera'는 ('High end camera' ^ sensor='film') U ('Economy camera' ^ sensor='film')에 매핑된다

(그림 8) 의미적 분류체계의 매핑

위해 원천클래스와 대상클래스 사이의 의미적 포함관계를 좀더 상세히 하여 의미적으로 일치하는 부분만을 취합하는 형태이다.

(그림 8)은 두 분류체계 A와 B 간의 매핑을 의미에 기반하여 어떻게 해결하는지 보여준다. 일대일 대응을 정의하는 기존의 크로스워크 테이블은 기껏해야 A의 'Film camera'는 B의 'Camera'에 대응될 수밖에 없다. 그러나, 의미적 분류체계에서는 클래스에 정의된 구성원자격함수를 활용하여, B의 모든 클래스경로를 탐색하게 되면 B의 각 클래스경로에서 'High end camera'와 'Economy camera'가 A의 'Film camera'의 최소중첩클래스임을 알 수 있다. 이러한 Compare_CLASS에 의한 탐색결과를 바탕으로 A의 'Film camera'의 구성원자격함수와 공통되는 부분인 sensor='film'을 매핑정보로 추가함으로써 최종적인 결과는 'High end camera'이면서 sensor가 film인 클래스와 'Economy camera'이면서 sensor가 film인 클래스로 매핑된다. 이렇게 하여 의미적 분류체계는 정보의 손실을 줄일 수 있다.

4.3.2 병합

분류체계 병합은 분류체계 매핑과 기본적으로 동일한 절차를 밟는다. 두 분류체계 CL_1 과 CL_2 가 있을 때, CL_1 에

서 CL_2 로의 분류체계 매핑에서는 매핑하려는 CL_1 의 어떤 클래스 C 에 대해 매핑 대상이 되는 분류체계 CL_2 에서 의미적으로 같거나 가장 유사한 클래스를 찾는 것이 첫 번째 단계였듯이 CL_1 과 CL_2 를 병합하여 CL 라는 새로운 분류체계를 만드는 분류체계 병합에서도 일단 CL_1 의 모든 클래스에 대해 CL_2 의 삽입 가능한 가장 적합한 클래스를 찾는 것이 첫 번째 단계이다. 따라서 분류체계 병합에서도 분류체계 매핑에서와 마찬가지로 Compare_CLASS를 이용한 의미적인 포함관계의 탐색이 필요하다. 하지만, 분류체계 매핑이 CL_1 의 클래스 C 에 대해 CL_2 에서 의미적으로 중첩된 클래스들을 찾아내는 작업을 하는 대신 분류체계 병합에서는 의미적으로 소속하는 클래스를 찾아내는 것이 그 다음 단계이다. 왜냐하면, 분류체계 매핑은 CL_2 의 클래스 자체를 분할하고 조합하여 클래스 C 와 최대한 의미적으로 동일해 지게끔 하는 것이 목적이지만 분류체계 병합에서는 CL_1 의 원본 클래스를 그대로 유지하면서 CL_2 의 가장 적합한 위치에 삽입하는 것이 목적이기 때문이다. 또한 분류체계 병합에서는 CL_1 의 클래스들 간 계층관계들도 유지하면서 CL_2 와의 병합을 통해 새로운 분류체계를 만드는 것이므로 결과적으로 CL_1 의 클래스들을 CL_2 에 삽입해야 하는 등 분류체계 매핑과는 달리 분류체계 구조의 조

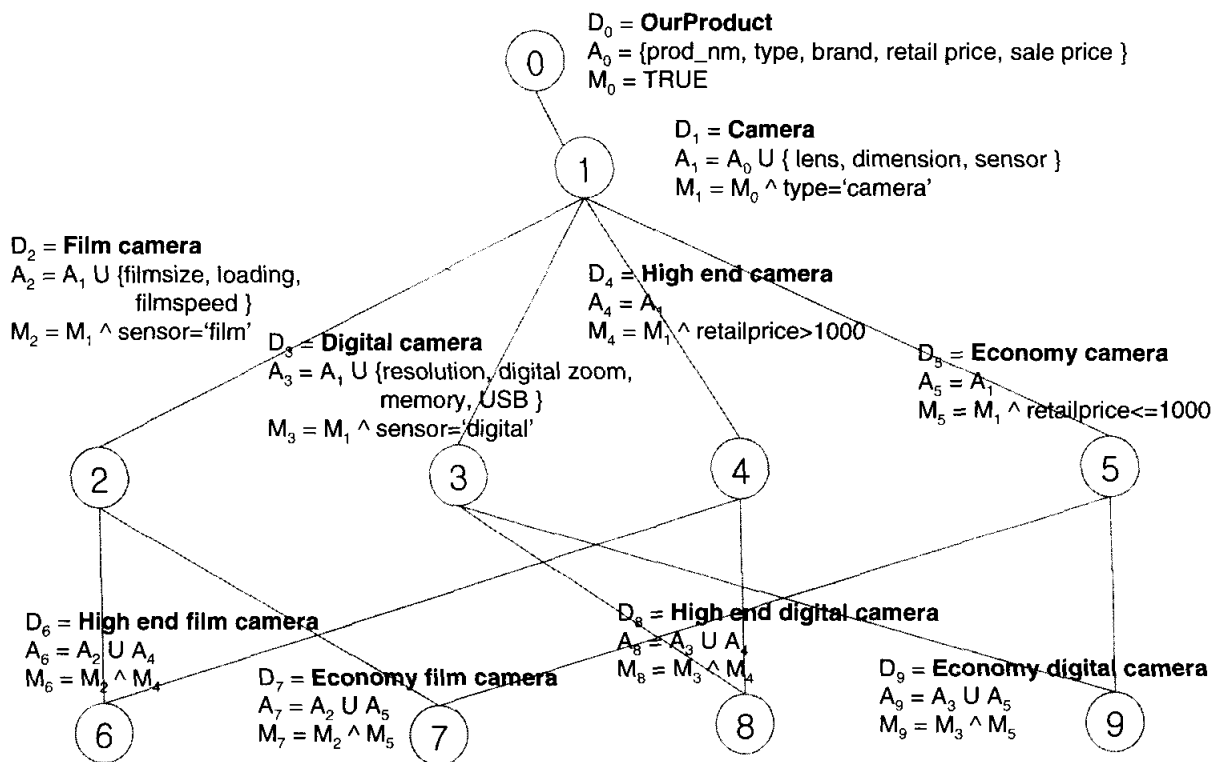
작이 필요하다.

따라서, 분류체계 병합은, (1) 일단 분류체계 CL_1, CL_2 각각의 루트클래스를 비교하여 병합할 범위를 정하고, (2) 분류체계 매핑과 같이 모든 클래스경로에 대해 Compare_CLASS를 이용하여 각 클래스의 삽입될 최적 위치를 결정하는 과정이 있는 후, (3) 실제 클래스삽입 과정을 처리하고, (4) 마지막으로 이전에 삽입된 원본 클래스 간의 관계를 그대로 유지하여 재설정한다.

이 분류체계 병합 과정에서 반드시 지켜야 하는 사항이 바로 정보의 손실이 없는 분류체계의 병합이다. 이러한 목적을 달성하기 위해서는 분류체계로서의 일관성을 보장해야 함은 물론이고 대상분류체계와 원천분류체계의 계층 구조 즉 원본의 클래스들과 관계들이 병합 이후에도 그대로 유지되어야 한다. 병합 이후 일관성의 보장은, 병합된 분류체계의 모든 명시적 포함관계 $C_i \geq_r C_j$ 인 모든 C_i, C_j 에 대해 의미적 포함관계 $C_i \geq_s C_j$ 가 존재함을 보

여주면 된다. 앞서 설명한 병합 과정에서 볼 수 있듯이 탐색 후 삽입 과정은 Compare_CLASS에 의해 의미적으로 소속하는 클래스를 찾아 이루어지므로, 모든 삽입으로 생성되는 명시적 관계는 의미적으로도 포함관계에 있음을 알 수 있다. 따라서 병합 과정에서 분류체계의 일관성은 보장된다.

이와 더불어 병합 후 원본 정보의 유지 문제는, CL_1 의 모든 클래스 C_i 와 CL_2 의 모든 클래스 C_j 가 병합된 분류체계에 존재하도록 하고, CL_1 에 있는 모든 명시적 포함관계 $C_i \geq_r C_j$ 와 CL_2 에 있는 모든 명시적 포함관계 $C_m \geq_r C_n$ 가 병합된 분류체계에 존재함을 보이면 된다. 그런데, 본 병합과정에서는 원본 분류체계의 모든 클래스를 대상 분류체계와 비교하여 삽입하게 되므로 모든 클래스들이 그대로 유지되고 있으며 그 관계 또한 삽입 이후 재설정함으로써 그대로 유지한다. 따라서, 원본 정보의 유지 또한 보장된다.



(그림 9) 의미적 분류체계의 병합

(그림 8)의 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 의미이며, 이들 클래스 간에 서로 배타적인 필요가 없다는 것을 의미한다. 이와 같이 비배타적인 클래스를 합병하는 경우, 이들 클래스와 슈퍼 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. 이 정보 또한 병합된 분류체계에서 유지가 되어야 한다. 이를 위해 의미적 분류체계에서는 이들 클래스를 분리하여 서브 클래스를 (그림 9)와 같이 정의한다. 이렇게 하여 클래스 간의 배타성 및 비배타성을 유지하여 일관성을 훼손하지 않는다.

5. 결론 및 향후 연구과제

전자상거래 환경에서 요구되는 상품 데이터베이스의 동적 특성과 이에 대한 기존 코드 기반 분류체계의 한계점을 분석하였으며, 의미적 분류 모형이 이러한 동적 특성의 지원에 적합하다는 것을 설명하였다. 의미적 분류 모형은 속성 및 부결성 조건과 같은 의미에 근거하여 클래스를 명시적이고 형식적으로 정의할 수 있는 수단을 제공하며, 클래스 간의 관계를 트리가 아닌 그래프로 구성한다.

의미적 분류 모형은 상품 정보에 대해 서로 다른 관점을 가진 사용자에게 다양한 뷰를 제공하며, 클래스의 삽입 및 삭제에 대해 일관성을 유지할 수 있는 방법 및 수단을 제공한다. 이질적인 두 분류 체계 간의 매핑에서도 정보의 손실을 감소시키며, 병합 시에도 일관성 유지 및 정보 손실 문제를 감소시킨다. 이렇게 하여 의미적 분류 모형은 상품 데이터베이스의 동적 특성을 쉽고 저렴하게 지원할 수 있다.

동적 특성의 효율적인 지원 이외에도, 의미적 분류 모형은 분류 체계의 명시적인 정의를 통해 사용자 간의 분류체계에 대한 오해를 줄일 수 있어서 분류체계의 표준화 노력에도 기여할 수 있다. 의미적 분류 모형은 eCliX 시스템에 구현되어[9, 10], 조달청에서 정부 조달 물품의 표준 분류를 위한 지침으로 활용되고 있다.

분류체계의 진화 시에 일관성 유지에 관한 문제는 정보의 인스턴스에 매우 미묘하기 때문에 형식적인 접근에 의한 연구가 더욱 필요한 부분이며, 매핑 및 병합을 위한 과정(알고리즘)이 일관성 및 정보 무손실의 보장에 관해 수학적 모형을 통한 증명도 향후 연구되어야 한다. 더욱 효율적이고 정교한 매핑 및 병합을 위해 상품 정보 및 분류체계에 관한 의미적 지식을 온톨로지에 표현[11, 12]하여 의미적 분류 모형을 활용하는 것도 연구 주제로 가치 있는 방법이다.

참 고 문 헌

[1] Dongkyu Kim, Sang-goo Lee, Jonghoon Chun, "A Semantic Classification Model for e-Catalogs," IEEE 6th Conference on E Commerce Technology(CEC 2004),

2004.
 [2] Fensel, Omclayenko, Ding, Schulten, Botquin, Brown, Flett, "A Product Data Integration in B2B Electronic Commerce," *IEEE Intelligence System*, 16(3), 2001.
 [3] 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.
 [4] Jörg Leukel, Volker Schmitz, Frank-Dieter Dorloff, "A Modeling Approach for Product Classification Systems," *proc. of DEXA Workshops*, 2002.
 [5] Rakesh Agrawal, Ramakrishnan Srikant, "On integrating catalogs," *proc. of WWW*, 2001.
 [6] Harmonized system committee, "Harmonized System Convention," available at http://www.wcoomd.org/ie/En/Topics_Issues/HarmonizedSystem/harmonizedsystem.htm, 2004.
 [7] UNDP, "United Nations Standard Products and Service Code, White paper," available at <http://www.unspsc.org/>, 2001.
 [8] Cologne Institute for Business Research, "eCl@ss New Standardized Material and Service Classification," available at <http://www.eclass-online.com/>, 2004.
 [9] Dongkyu Kim, Sang-goo Lee, Jonghoon Chun, Sangwook Park, Jaeyoung Oh, "Catalog Management in E-Commerce Systems," *proc. of Computer Science & Technology*, 2003.
 [10] Corelogix, "eCliX product overview," available via <http://www.corelogix.co.kr>, 2001.
 [11] Y. Ding, M. Korotkiy, B. Omelayenko, V. Kartseva, V. Zykov, M. Klein, E. Schulten, and D. Fensel, "GoldenBullet in a Nutshell," The 15th International FLAIRS Conference, 2002.
 [12] D. Fensel, "Ontologies : Silver Bullet for Knowledge Management and Electronic Commerce," Springer-Verlag, 2001.
 [13] Erhard Rahm and Philip A. Bernstein, "A survey of approaches to automatic schema matching," *VLDB Journal* 10(4), 2001.
 [14] ECCMA, "Conversion crosswalk table : CPV to UNSPSC," available via <http://www.unspsc.org>, 2000.

김 동 규



e-mail : ddkim@corelogix.co.kr
 서울대학교 계산통계학과(학사)
 서울대학교 전산과학과(석사)
 서울대학교 컴퓨터공학과(박사)
 2001년 Georgetown University ISIS 연구원
 2002년~현재 ㈜코어로직스 연구소장

관심분야 : 데이터베이스, 전자상거래, 전자카탈로그, 온톨로지, 정보검색

이 상 구



e-mail : sglee@snu.ac.kr
서울대학교 계산통계학과(학사)
Northwestern University 전산학과
(석사, 박사)
현재 : 서울대학교 컴퓨터공학부 교수
2002년~현재 서울대학교 e-비즈니스 기술
연구센터 센터장

관심분야 : 데이터베이스, 전자상거래, 전자카탈로그, CRM
디지털 라이브러리

최 동 훈



e-mail : choid@paran.com
서울대학교 계산통계학과(학사)
한국과학기술원 전산학과(석사)
Northwestern University 전산학과(박사)
1983년~1986년 한국증권전산(주)
1989년~1992년 한국국방연구원 선임연구원

1992년~1999년 동덕여자대학교 전산학과 부교수
2004년~현재 한국과학기술원 전산학과 초빙교수
관심분야 : 데이터베이스, 시맨틱 웹