

교육용 자원 저장소를 위한 의미적 분류 모델

(A Semantic Classification Model for Educational Resource Repositories)

최 명 회 [†] 정 동 원 ^{**}
(Myounghoi Choi) (Dongwon Jeong)

요 약 이 논문에서는 교육용 저장소 자원의 체계적인 관리를 위한 분류 모델을 제안한다. 생성되는 자원들에 대한 체계적인 저장 및 관리, 정확한 검색, 그리고 활용성을 극대화하기 위해서는 정확한 분류 체계가 요구된다. 그러나 교육용 저장소 자원을 위한 분류 체계나 분류 모델에 대한 연구는 미비한 수준이다. 이는 교육 자원의 비효율적 관리, 부정확한 검색 및 낮은 활용성 등의 문제점을 초래한다. 상품 정보와 관련된 분야에서는 다양한 분류 체계에 대한 연구가 이루어져 왔다. 그러나 교육 자원 정보와 기존 연구 분야의 정보는 서로 다른 특성을 지닌다. 따라서 교육용 저장소 내 자원 관리를 위한 분류 체계 및 분류 모델에 대한 연구가 요구된다. 교육 자원들에 대한 효율적이고 편리한 활용을 위해서는 여러 관점을 반영하는 분류 체계에 따라 자원들을 일관성 있게 유지 관리하여야 한다. 이 논문에서는 교육 자원의 체계적인 관리 및 활용성 향상을 위한 분류 모델을 제안한다. 즉, 교육용 저장소의 자원들에 대한 분류 체계를 다양한 관점에 따라 동적으로 유지할 수 있는 분류 모델을 제안한다. 이러한 목적을 위해 먼저 관련된 과학기술분야 분류 체계들을 바탕으로 구현 자원들에 적합한 분류 체계를 정의한다. 특히 정의된 분류 체계를 동적으로 유지 관리할 수 있는 분류 모델을 정의한다. 제안된 분류 체계 및 분류 모델은 보다 정확하고 체계적인 구현 자원에 대한 관리를 가능하게 하며 또한 활용의 용이성을 향상시킨다.

키워드 : 교육용 저장소, 의미 분류, 분류 체계, 의미 분류 모델, 동적 의미 관리

Abstract This paper proposes a classification model for systematical management of resources in educational repositories. A classification scheme should be provided to systematically store and manage, precisely retrieve, and maximize the usability of the resources. However, there is little research result on the classification scheme and classification model for educational repository resources. It causes several issues such as inefficient management of educational resources, incorrect retrieval, and low usability. However, there are different characteristics between the educational resource information and information of the previous fields. Therefore, a novel research on the classification scheme and classification model for the resources in educational repositories is required. To achieve the goal for efficient and easy use of the educational resources, we should manage consistently the resources according to the classification scheme accepting several views. This paper proposes a classification model to systematically manage and increase the usability of the educational resources. In other words, the proposed classification model can manages dynamically the classification scheme for the resources in educational repositories according to various views. To achieve the objectives, we first define a proper classification scheme for the implementation resources based on the classification scheme in relevant scientific technology fields. Especially, we define a novel classification model to dynamically manage the defined classification scheme. The proposed classification scheme and classification model enable more precise and systematic management of implementation resources and also increase the ease of usability.

Key words : Educational Repository, Semantic Classification, Classification Scheme, Semantic Classification Model, Dynamic Semantic Management

· 이 논문은 2005년도 군산대학교 신임교수 연구비 지원에 의하여 연구되었음

[†] 학생회원 : 군산대학교 정보통계학과
cmh775@kunsan.ac.kr

^{**} 정 회원 : 군산대학교 정보통계학과 교수
djeong@kunsan.ac.kr

논문접수 : 2006년 8월 7일
심사완료 : 2006년 11월 24일

1. 서 론

교육용 저장소는 교육 과정에서 생성되는 다양한 자원들을 효율적이고 체계적으로 관리해줌으로써 학습 효과를 배가시키고 질적으로 향상된 교육 시스템을 개발

하는데 그 목적이 있다. 그러나 교육용 저장소 시스템에 관한 연구는 매우 미비한 상황이다. 특히 다양하고 방대한 교육 자원들을 관리하기 위해 필수적으로 요구되는 분류 체계에 대한 연구는 거의 이루어지지 않았다. 생성되는 자원들에 대한 체계적인 저장 관리, 정확한 검색 및 활용성의 극대화를 위해서는 정확한 분류 체계가 요구된다.

지금까지 다른 다양한 분야에서는 각 분야별 생성되고 이용되는 자원들에 대한 효율적이고 체계적인 관리 및 정확한 활용을 위해 분류 체계에 대한 연구가 진행되어 왔다[1-3]. 이와 같은 다양한 분류 체계가 정의되어 있으나 이러한 분류 체계는 대부분 상품과 서비스를 대상으로 하고 있기 때문에 교육용 저장소 자원 관리에 적용할 수 없다는 문제점을 지닌다. 또한 분류 체계를 수정하고 관리하는 분류 모델(분류 체계 관리 모델) 또한 정적 분류 모델을 기반으로 하고 있다는 문제점을 지닌다. 교육용 저장소에서 관리되는 자원은 매우 다양하고 또한 동적이라는 특징을 지닌다. 따라서 교육 자원의 분류 체계 관리 또한 동적으로 이루어져야 한다. 그러나 기존 관리 모델은 이를 지원하지 못한다. 이 논문에서 동적 관리란 분류 클래스에 의미를 부여하고 분류 클래스 간 의미 관계성을 정의할 수 있도록 함으로써 코드 기반 접근 방법의 정적 분류 체계 관리의 문제점을 보완하기 위한 개념이다.

교육 자원을 위한 분류 체계는 과학기술 분야에 대한 분류 체계 관리와 유사하다. 현재 국내의 과학기술 분류 체계 중에서 가장 많이 사용되는 분류 체계는 한국과학재단, 한국학술진흥재단에서 제안하고 있는 분류 체계이다[4-6]. 이러한 분류 체계의 문제점은 정적 분류 모델, 즉 정적 분류 관리 모델을 기반으로 한다는 점이다. 다시 말해서, 과거에 생성되었던 활동을 위주로 분류 체계를 정의하고 새로운 영역에 대한 추가나 변경, 혹은 기존 분류 클래스에 대한 분할 등이 정적으로 이루어진다. 또한 동일한 클래스가 다른 여러 상위 클래스와 중복되는 문제점을 지닌다. 무엇보다 분류 체계내의 분류 클래스들 간의 관계성이 표현되지 않아 의미적인 해석과 이해에 어려움이 있으며 분류 체계들 간의 연관성을 정의하여 서로 다른 자원들 간의 관계를 고려하여 활용할 수 없다. 무엇보다 이들 분류 체계만으로 교육용 저장소의 자원들을 보다 정확하게 분류하고 관리할 수 없다는 문제점을 지닌다. 이는 단순히 과학기술 관점에서 분류 클래스를 정의하였기 때문이다. 이에 반해, 교육용 저장소에서 관리하는 자원들은 단순한 과학기술 관점이 아닌 응용 관점 또한 분류 체계 정의에 반영되어야 한다[7].

또 다른 측면에서, 기존 분류 체계는 트리 구조 혹은

분류 코드를 갖는 단순 코드 기반의 분류 모델을 기반으로 하고 있다는 문제점을 지닌다. 코드 기반의 분류 체계에서 수퍼 클래스의 코드(예, A10)는 서브 클래스 코드(예, A1011, A1012)의 접두어가 된다. 정보 즉, 분류 클래스의 개념, 개념 간의 관계 및 무결성 조건 등은 주로 명시적으로 표현되지 않고 묵시적으로 표현된다. 따라서 시스템적으로 처리할 수 없으며 단지 문서를 통해 가이드 라인을 작성하여 유지한다. 코드 기반 분류 체계를 위한 논리적 모델의 이러한 문제점은 동일한 분류 체계 내에서 분류 기준 및 분류 레벨의 혼재로 인한 일관성 및 융통성 결여 등의 여러 문제점을 초래한다[8].

이 논문의 목적은 교육용 저장소를 위한 분류 체계를 정의하고 정의된 분류 체계의 동적 관리를 위한 모델을 제시함에 있다. 분류 체계는 대표적인 한국과학재단과 한국학술진흥재단 등에서 제안한 과학기술 분류표를 기반으로 하여 두 가지 관점(과학기술 관점과 응용 관점)을 고려하여 정의한다. 동적 분류 모델 즉, 동적 분류 체계 관리 모델은 정의된 분류 체계의 분류 클래스의 추가, 병합, 분할 등이 가능하도록 하며 이를 위해 분류 클래스들 간의 관계성을 정의하고 관리할 수 있도록 한다. 특히 단순히 코드로 분류 클래스를 정의하는 것이 아니라 의미를 부여하고 이를 이용하여 분류 클래스 간의 관계성까지 정의함으로써 의미적으로 분류 체계를 정의하고 관리할 수 있도록 한다. 이는 앞서 언급한 코드 기반 분류 체계 관리에 비해 분류 클래스 의미를 정확하게 기술해 줌으로써 동적 관리 및 의미 관계성 관리를 용이하게 한다.

이 논문의 구성은 다음과 같다. 제2장에서는 교육용 저장소의 자원들을 위한 분류 모델 정의시 고려해야 할 사항과 몇 가지 제약 조건을 기술한다. 제3장에서는 이 논문에서 제안하는 분류 모델인 eCSM에 대하여 서술하고 제4장에서는 eCSM을 위한 구현 모델에 대하여 기술한다. 제5장에서는 eCSM의 구현 결과를 보이고 제6장에서는 이 논문에 대한 결론을 기술한다.

2. 고려 사항 및 제약 조건

이 장에서는 교육 저장소의 자원들의 특징에 대하여 기술한다. 또한 분류 모델에서 고려해야 할 고려 사항 및 제약 조건에 대하여 정의한다.

2.1 제약 조건 및 기본 용어 정의

교육용 저장소는 교육 전반에 걸쳐 생성되는 모든 자원을 관리한다. 자원은 교육에 이용되는 정보들로서 일차적으로는 교수와 같은 교육 지도자에 의해 생성되며 이차적으로는 학생과 같은 학습자들에 의해 생성된다. 학습자들에 의해 생성되는 자원은 일반적으로 과제에 대한 결과물로서 보고서, 단기 프로젝트 구현 소스, 사

용된 도구 및 이에 대한 정보 등을 포함한다. 이 논문에서는 이러한 자원들 중에서 전산학 관련 대학의 학과에서 학습자들에 의해 생성되는 프로젝트 결과물을 대상으로 한다. 자원의 범위를 프로젝트 결과물을 일차 대상으로 선정한 이유는 먼저 프로젝트를 통해 생성되는 구현자료 및 그에 따른 다양한 문서 등의 관리가 단순한 보고서와 같은 자료의 관리보다 어려움이 따르고 다양한 분야에 초점을 두어 생성될 수 있기 때문에 그 관리의 중요성이 더욱 요구되기 때문이다.

이 논문에서 사용하는 주요 용어에 대한 정의는 다음과 같다. 이러한 용어를 정의하는 이유는 이 논문에서는 사용하는 용어가 관련 분야에서 다양한 의미 해석될 수 있기 때문에 정확한 의미를 전달하고자 함에 있으며 또한 모호한 용어들에 대한 혼란을 방지하고자 함에 있다.

- 분류 체계: 분류 클래스와 그 클래스들 간의 관계를 정의한 결과를 의미한다.
- 분류 모델: 분류 체계 관리 모델의 단축 표현으로 분류 클래스들을 관리하기 위한 연산과 주요 연산 모듈의 집합을 분류 모델이라 한다.
- 분류 클래스: 분류 체계에서의 의미적인 노드를 의미하며 트리 구조로 표현할 경우 각 노드에 해당하며 실제 자원들에 대한 논리적인 그룹핑 단위가 된다.
- 단말 분류 클래스: 분류 체계는 트리나 그래프 형태로 표현되며 최하위 분류 클래스를 의미하며, 자원들을 직접적으로 포함하는 분류 클래스를 의미한다. 잎 노드(Leaf Nodes)라는 용어도 동일한 의미로 사용된다.
- 비단말 분류 클래스: 분류 체계에서 단말 클래스가 아닌 분류 클래스들의 집합으로 일반적으로 상위 분류 클래스 및 하위 분류 클래스를 지닌다.
- 루트 분류 클래스: 최상위 분류 클래스를 의미하며 이 논문에서는 크게 세 가지 관점에서 정의된 세 가지 형태의 분류 체계를 이용하므로 세 개의 루트 분류 클래스를 지닌다. 분류 체계를 위한 관점에 대한 상세한 내용은 제 2.2절에서 기술한다.
- 자원: 교육 과정에서 발생하는 모든 데이터로서, 이 논문에서는 앞선 제약 조건에 따라 대학 전산학 교육 과정에서 발생하는 프로젝트 결과물을 의미한다. 자원들은 각 분류 클래스의 인스턴스가 된다. 다시 말해서 분류 클래스에 의해 그룹핑되며 자원 객체와 동일한 의미를 지닌다.

2.2 교육용 저장소의 분류 모델을 위한 고려 사항

교육용 저장소의 자원들을 생성하는 주체에 따라 어떻게 분류할 것인가 하는 문제는 그다지 중요하지 않다. 분류 모델 정의를 위한 교육 저장소 자원의 중요한 특징은 자원들이 매우 다양한 분야를 대상으로 생성될 수 있다는 점이다. 또한 교육 과정에 따른 분류 또한 고려

되어야 한다. 이 논문에서 자원 범위의 특성상, 개발하는 언어에 따른 분류도 고려되어야 한다. 교육 과정을 먼저 예를 들어 보자. “데이터베이스”와 “알고리즘” 과정은 명백히 분류되어야 하며 각각 하나의 분류 클래스로 정의될 수 있다. 각 과정에서 생성된 자원들은 “데이터베이스” 혹은 “알고리즘”이라는 분류 클래스의 소속되는 자원 객체가 된다.

결론적으로, 이 논문에서는 세 가지의 관점을 고려하여 자원에 대한 분류 체계를 정의해야 함을 의미한다. 그림 1은 세 가지의 관점을 고려한 분류 체계에 대한 개념도를 보여준다. 이러한 세 가지 관점을 고려한다는 의미는 세 가지 형태의 분류 체계가 구성되어 하나의 전체 분류 체계를 완성한다는 것이다. 이는 하나의 자원이 이 세 가지의 관점에 따른 세 개의 다른 분류 체계 내의 특정 분류 클래스 내에 속하게 됨을 의미한다. 응용 관점은 달리 응용 사용자 관점이라 표현할 수 있다. 과학기술적인 분류 체계나 언어적인 분류 체계가 아닌 “개인 정보 관리 시스템” 혹은 “비디오 관리 시스템”과 같은 분류 체계에 더욱 익숙하거나 이러한 측면에서 검색하여 활용하고자 하는 사용자를 위한 관점이다. 이 논문에서 정의한 세 가지 분류 관점은 모든 응용 분야를 위한 일반적인 목적을 충족하고자 함이 아니라 이 논문의 목적인 다중 관점과 동적 분류 모델에 대한 사례 연구를 위해 정의한 것이다. 따라서 분류 관점 및 각 관점을 위한 분류 클래스는 대상 분야별로 정의되어야 한다. 그러나 이 논문에서 제안하는 분류 체계 관리 개념, 기본적인 구조 및 주요 연산 등은 모든 분야에 적용할 수 있다.

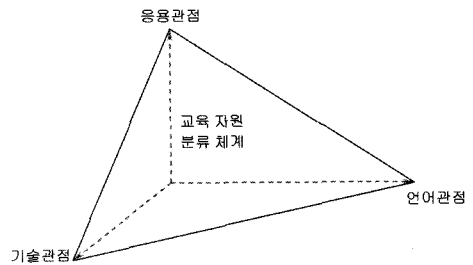


그림 1 분류 체계와 관점

3. eCSM 정의

이 논문의 목적은 교육용 저장소의 자원을 위한 분류 모델을 정의하는 것이며 이 논문에서 제안하는 분류 모델 즉, 분류 체계 관리 모델을 eCSM(Educational Classification Scheme Management Model)이라고 정의한다.

3.1 eCSM의 구조

이 논문에서는 자원들의 모든 집합을 R, 전체 분류

체계를 C라고 정의한다. 그리고 과학기술적 관점, 응용 사용자 관점, 언어 관점을 각각 S, A, L이라고 표기하며 각각에 의한 분류 체계를 C_S, C_A, C_L 로 정의한다. 루트 클래스의 집합을 C_R 이라 정의하며 $C_R = \{C_{RS}, C_{RA}, C_{RL}\}$ 이다. 즉, 세 가지 관점에 따른 분류 체계의 루트 클래스들의 집합이다. 클래스들의 관계성의 집합을 P이라 하고, 클래스의 변경을 위한 연산들의 집합을 OP라 할 때 eCSM은 다음과 같이 정의할 수 있다.

정의 1. 자원 R에 대한 분류 체계 eCSM은 5 튜플 $\langle R, C, C_R, P, OP \rangle$ 로 정의된다.

- R: 분류 대상이 되는 자원 집합으로서 3 튜플 $\langle R_{source}, R_{doc}, R_{u-guide} \rangle$ 로 구성된다.
- R_{source} : 실제 구현자원 즉, 구현소스
- R_{doc} : 구현자원에 대한 보고서 및 인터페이스
- $R_{u-guide}$: 구현 환경과 설치 등의 사용자 가이드
 - $C = C_S U C_A U C_L$
 - $C_R = \{C_{RS}, C_{RA}, C_{RL}\}$
 - P: 분류 클래스들 간의 관계성을 나타내며 제한하는 분류 체계는 트리가 아닌 그래프 구조를 지닌다.
 - OP: 클래스의 변경에 따른 연산들의 집합. 연산은 크게 클래스들에 대한 연산 OP_{class} 와 분류 클래스의 변경에 따른 자원들의 배치와 관련된 연산 $OP_{instance}$ 로 분류된다.

정의 1에서 자원 R은 분류 체계 C에 의해 그룹핑되며 C의 원소인 분류 클래스들 간의 관계성은 P에 의해 표현된다. 만일 새로운 분류 클래스가 추가될 경우 OP의 한 연산에 의해 C, R 및 P의 값이 수정된다. 그림 2는 eCSM의 구조에 대한 개념도를 보여준다.

그림 2에서, c_1 은 기술관점에서 정의된 분류 체계 중의 한 분류 클래스로서, 예를 들어, “데이터베이스” 혹은 “알고리즘”과 같은 과정명이 될 수 있다. c_1 을 “알고리즘”이라 가정할 때, 그 클래스에 속하는 자원들 즉, r_1, r_2 , 및 r_5 는 해당 과정 하에서 생성된 자원들의 집합이다. c_2 는 응용관점에서 정의된 분류 체계 내의 한 분류 클래스로서, “개인 정보 관리 시스템”이나 “비디오

샵 관리 시스템” 등이 될 수 있다. c_2 를 “개인 정보 관리 시스템”이라고 할 경우, 해당 자원 r_1 이나 r_3 는 개인 정보를 관리하는 구현 자원들이 된다. r_1 의 경우는 c_1 과 c_2 에 속하게 되므로 알고리즘 과정에서 개인 정보 관리 시스템을 주제로 한 단기 프로젝트의 결과임을 의미한다. c_3 은 언어관점에서 정의된 분류 체계의 한 분류 클래스로서 “Java”가 될 수 있다. 따라서 r_4 와 r_5 등은 자바 언어로 개발된 자원임을 의미한다.

3.2 eCSM에서의 관계성

보다 지능적이고 활용성 있는 분류 체계를 위해서는 분류 클래스 간 혹은 클래스와 자원과의 관계성이 정의되어야 한다. 관계성이란 두 개체들 간의 관계를 의미하며 이 논문에서는 크게 세 가지 형태의 관계성이 존재한다. 추가적으로, 이 논문에서의 관계성은 다른 유사 연구 분야에서는 술어(Predicate) 혹은 특성(Property) 등과 동일한 의미로 사용된다. 이 논문에서는 선택한 대상 분야에 사용 가능한 모든 의미 관계성을 정의하지 않는다. 향후 필요시 동적으로 관계성을 정의하여 활용할 수 있다. 이 논문에서의 관계성은 클래스 간의 관계성, 클래스와 자원과의 관계성 그리고 자원 간의 관계성 등의 세 가지 형태로 정의되며 eCSM에서의 관계성 P는 다음과 같이 정의할 수 있다.

정의 2. 클래스 간, 클래스와 자원 간, 그리고 자원들 간의 관계성 집합을 각각을 P_{CC}, P_{CR}, P_{RR} 이라고 할 때, eCSM의 관계성 P는 3 튜플 $\langle P_{CC}, P_{CR}, P_{RR} \rangle$ 로 정의된다.

먼저, 클래스 간의 관계성은 분류 클래스 간의 관계성을 의미하며 일반적으로, $parentOf(superClassOf)$ 나 $childOf(subClassOf)$ 와 같은 관계성을 지닌다. 클래스와 자원 간의 관계성은 단순한 is_a 관계이다. 단순히 자원들이 해당 단말 분류 클래스에 소속되어 그룹핑됨을 의미한다. 이 논문에서 클래스 간 혹은 클래스와 자원 간의 관계성과는 매우 일반적인 개념에 따른 관계성들이 존재하는 반면, 자원 간의 관계성은 교육용 저장소를 위한 자원이라는 특성으로 인해 특별한 관계성들을 지닌다. 다시 말해서, 이 논문에서의 대상은 대학의 전산학 분야에서 생성되는 단기 프로젝트 결과물을 주 대상으로 하고 있다. 그러므로 생성되는 자원이 개발 결과(구현 자료, 프로그램 소스), 결과 보고서, 개발 환경 등과 같은 사용자 가이드 등이다. 따라서 자원 간에는 $sourceOf, documentOf, userGuideOf$ 와 같은 관계성을 지닌다. 추가적으로, 제한하는 분류 모델은 언급한 관계성의 새로운 관계성을 정의할 수 있다.

그림 3은 앞에서 언급한 관계성을 부여한 한 예를 보여준다. 그림에서, c_R 은 c_1 의 부모 클래스임을 의미하며 방향성이 반대일 경우에는 그 의미가 역이 된다. 즉, c_4

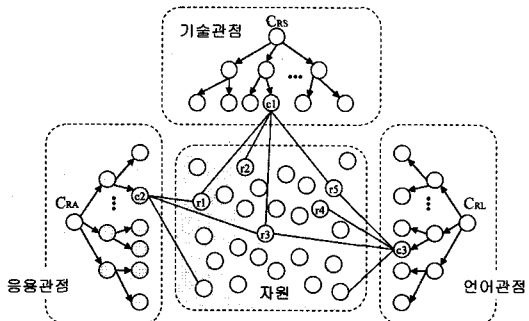


그림 2 eCSM의 개념 모델

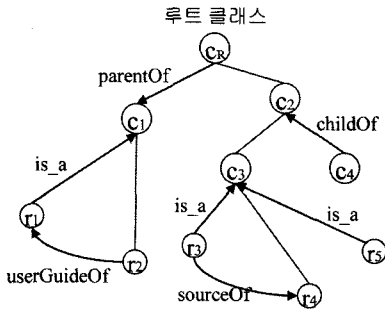


그림 3 eCSM에서의 관계성의 표현 예

가 c_2 의 자식 클래스라는 관계성을 지니는 것과 같다. 클래스와 자원 간에는 is_a 관계성만이 존재하는 것을 알 수 있다. 자원 간의 관계성에서, r_1 은 r_2 의 사용자 지침서라는 의미이며 따라서 이는 두 자원이 동일한 하나의 단기 프로젝트를 통해 나온 결과물임을 알 수 있다.

또한 이 논문에서 제시한 eCSM에서 모델의 일관성이나 무결성 등을 고려하기 위한 클래스간 관계성 표현의 제약사항은 다음과 같다. 이 제약사항은 eCSM에서 제시할 클래스 연산에서 연산과정이 맞는지를 판별하는 기준이 된다.

- 자신과의 관계성: C_1 은 자기 자신인 C_1 과 관계성을 갖지 않는다. 즉 자신을 참조하여 관계성을 생성하는 경우로서 일반적인 상황에서는 발생하지 않는다. 그러나 관리자의 관계 설정상의 실수로 인해 발생될 수 있기 때문에 이를 검사하여 올바른 분류 체계를 유지하도록 한다.
- 부모클래스의 부모클래스와 관계성: C_4 는 C_R 과 관계성을 갖지 않는다. 또한 직계 부모클래스와도 관계성을 갖지 않는다.
- 부모클래스와 형제(sibling) 관계에 있는 클래스와 관계성: C_4 는 C_1 과 관계성을 가진다.
- 자신과 sibling관계에 있는 클래스와 관계성: C_3 는 C_4 와 관계성을 갖지 않는다.
- 부모클래스와 sibling관계 클래스의 부모클래스와 관계성: 두 클래스간 관계성을 갖지 않는다. 이는 세 번째 언급한 제약사항과 관련이 있다. 자신의 부모클래스와 sibling관계에 있는 클래스의 부모클래스는 하위 클래스로 좀더 세분화된 클래스들이 존재하기 때문에 논리적인 분류로 봤을 때 적합하지 않기 때문이다.

3.3 동적 분류 체계 관리를 위한 주요 연산

OP는 분류 클래스들의 변경에 따른 연산들의 집합으로 크게 분류 클래스들에 대한 연산과 분류 클래스 변경에 따른 자원의 재배치에 따른 연산으로 구분된다. 다음은 분류 클래스에 대한 주요 연산들이다.

- 생성: 새로운 분류 클래스의 생성. 이 연산은 추가, 일

반화, 병합, 분할 등의 연산과 연관성을 지닌다. 생성 연산은 기존 분류 클래스에 대한 변경 없이 발생하는 생성 연산(추가, 일반화 연산과 직접적인 연관성을 지님)과 기존 분류 클래스의 변화를 가져오는 생성 연산(병합이나 분할 연산과 관련됨)으로 분류된다.

- 노드 추가: 생성 연산 이후에 단말 혹은 단말이 아닌 노드로서 분류 클래스가 추가되는 연산. 추가 연산은 새로운 자원을 위해 필요한 분류 클래스를 생성하여 단말 노드(단말 분류 클래스)로 추가하는 연산이다.
- 관계 추가: 관계 추가 클래스들 간의 관계성을 추가해주는 연산이다.
- 병합: 생성되거나 기존 클래스 속성에 따른 클래스 병합. 비슷한 속성을 가진 클래스를 하나의 클래스로 통합하고 단말 클래스들의 병합일 경우에는 자원들도 새로운 클래스 내에 소속된다.
- 일반화: 이 연산은 병합 연산과는 달리 하위의 여러 분류 클래스들을 하나의 상위 분류 클래스로 정의하는 연산.
- 분할: 클래스 속성에 따른 세부 분할. 기존 클래스 안에서 다른 속성을 일 때, 새로운 두 클래스를 생성하고 자원들을 클래스에 맞게 분할된다. 단말 노드의 분할이 아닐 경우에는 자원들의 분할에는 영향을 주지 않는다.
- 소멸: 병합, 분할하지 못하는 클래스 소멸. 분류 클래스간의 병합이나 분할 등의 연산이나 적절하지 못한 클래스 속성을 지닐 때 클래스를 제거한다.

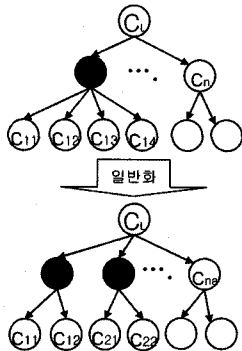
표 1은 앞서 언급한 연산들에 대한 연산자와 발생 가능한 경우들에 대한 설명을 기술한다.

표 1에서 알 수 있듯이, 각 연산자들은 분류 클래스가 단말인 경우와 비단말 클래스인 경우에 따라, 그리고 연산 결과에 대해 영향을 받는 대상이 클래스인지에 자원인지에 따라 세부 연산이 여러 형태를 지닌다. 일반화 연산의 경우 그림 3에서와 같이 c_1 의 하위 클래스가 두 가지의 분류 형태를 정의될 때 일반화 연산을 수행하게 된다. 생성 연산을 통해 새로운 클래스를 생성하고 생성된 클래스인 c_{1a} , c_{2a} 에 기존의 c_1 에 속해있는 하위 클래스들을 각각의 클래스에 분배한다. 다음으로 클래스 간의 노드(addClass)와 관계성(addProperty)을 추가한다. 마지막으로 기존 클래스인 c_1 를 제거하면 일반화 연산이 완료된다.

병합연산(merge())의 경우 그림 5에서와 같이 두 개의 분류 클래스를 병합하게 되면 새로운 분류 클래스인 c_{1a} 를 생성(new())하고, 생성된 c_{1a} 와 C_1 간의 노드를 생성(addClass())하여 두개의 클래스 간의 관계성(addProperty())을 추가한다. 또한 기존의 c_1 과 c_2 의 하위 클래스인 $c_{11}, c_{12}, c_{21}, c_{22}$ 와 새로 생성된 분류 클래스 c_{1a} 와의

표 1 분류 클래스에 대한 연산자

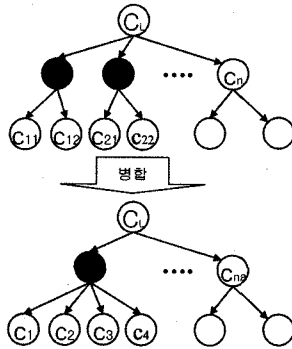
연산	연산자	설명
생성	$new(c_i)$	새로운 분류 클래스 c_i 를 생성
노드추가	$addClass(C, c_i, c_j)$ /** $addClass(C_S, C_i, c_j)$ OR $addClass(C_A, C_i, c_j)$ 등	새로운 분류 클래스 c_j 를 기존 분류 클래스 집합 C 의 분류 클래스 c_i 의 하위 클래스로 추가되며 정의 1에서 C 는 $\{C_S, C_A, C_L\}$ 중의 하나
관계추가	$addProperty(C_i, c_j, C_m, c_n)$	두 분류 클래스 c_j 와 c_n 간의 관계성(relationship, property)을 추가하는 연산으로 C_i 와 C_m 는 세 가지 분류 클래스 집합 $\{C_S, C_A, C_L\}$ 중의 하나가 됨
병합	$merge(C, c_i, c_j, c_p)$	클래스 집합 C 의 두 분류 클래스 c_i 와 c_j 를 합병하며 합병된 결과는 c_p 라는 분류 클래스 명으로 대체된다. c_p 가 주어지지 않을 경우에는 c_j 로서 대체
일반화	$generalize(C, c_i, \dots, c_m, c_p)$	분류 클래스 c_i, c_j, \dots, c_m 을 c_p 라는 분류 클래스를 생성하여 일반화 하며, 단말 분류 클래스가 아닌 비단말 분류 클래스가 생성되어 추가되는 연산
분할	$split(C, c_i, C_m, c_n)$	분류 클래스 집합 C 의 c_i 를 c_m 과 c_n 으로 분리
소멸	$delete(C, c_i)$	분류 클래스 집합 C 의 분류 클래스 c_i 를 삭제



```

new("c1a")
new("c2a")
generalize(CL,c11,c12,c1a) & generalize(CL,c11,c12,c1a)
{
  addClass(CL,cl,c1) & addClass(CL,cl,c2){
    addProperty(CL,CL,parantOf,CL,c2)
    addProperty(CL,CL,parantOf,CL,c1)
    for(i=1;i<=n;i++){
      addProperty(CL,c1,parantOf,CL,c1i)
    }
  }
}
delete(CL,c1)
    
```

그림 4 일반화 연산



```

merge(CL,C1,C2,C1a)
addClass(CL,cl,c1a){
  addProperty(CL,cl,parantOf,CL,c1a)
  for(i=1;i<=n;i++){
    addProperty(CL,c1,parantOf,CL,c1i)
  }
  for(j=1;j<=m;j++){
    addProperty(CL,c2,parantOf,CL,c2j)
  }
}
delete(CL,c1)
delete(CL,c2)
    
```

그림 5 병합연산

관계성(addProperty())를 추가한. 최종적으로 각각의 클래스간의 관계성이 만들어지면, 병합전의 분류 클래스인 c_1 과 c_2 를 분류 클래스에서 삭제하게 된다.

여러 연산들을 통해 클래스가 생성되면 상위 클래스의 속성을 상속받는다. 생성 연산(new)이나 소멸 연산(delete)의 경우 연산자의 사용이 쉽고, 일반화 연산과 병합 연산에서 사용 방법을 설명했으므로 추가적인 설명은 생략한다.

3.2절에서 언급한 바와 같이 클래스 간 발생하는 여러 가지 연산과정에서 분류클래스 간 제약사항에 위배되지

않을 경우 정상적으로 연산이 종료된다. 그러나 임의의 연산과정에서 앞서 언급한 제약조건에 위배될 경우 연산과정은 완료되지 않는다.

4. eCSM 구현 모델

이 장에서는 eCSM을 위한 구현 모델에 대하여 기술한다. 그림 6은 eCSM의 구현을 위한 메타모델을 보여준다.

그림 6에서, View는 관점에 대한 정보를 정의하며 Classification_Class는 분류 체계 내의 각 분류 클래스

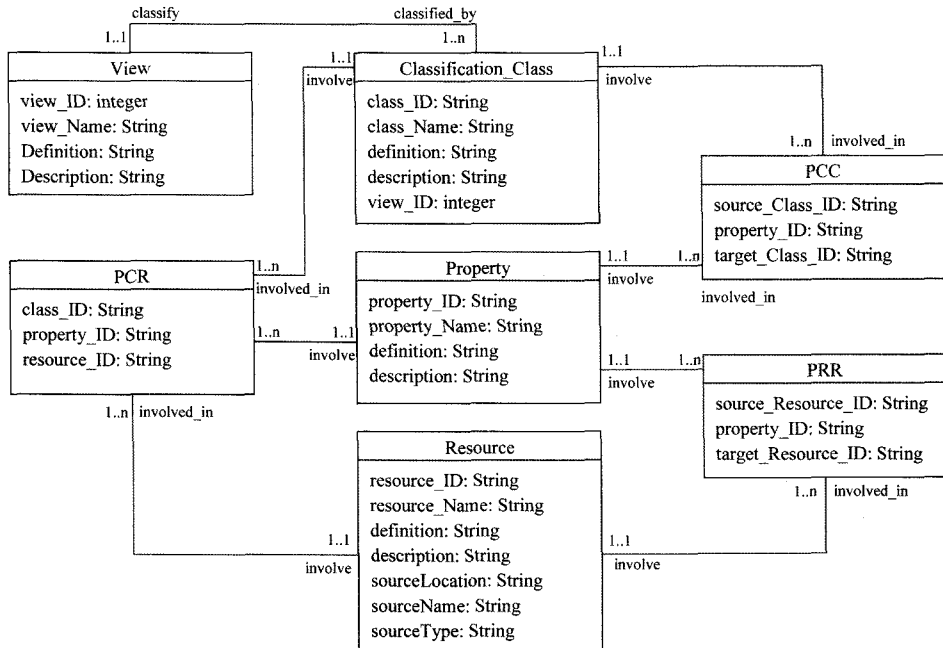


그림 6 eCSM 구현을 위한 메타모델

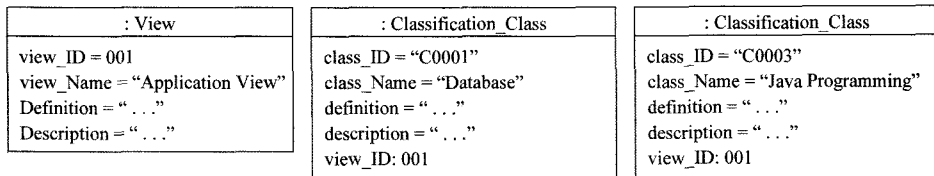


그림 7 View와 Classification_Class의 예

를 정의하는데 이용한다. 특정 View 포함되는 모든 Classification_Class 집합이 해당 관점에 의한 분류 체계를 이루게 된다. 따라서 이 논문에서는 실질적으로 다음과 같이 정의될 수 있다.

그림 7에서, 분류 클래스 “Database”와 “Java Programming”은 각각 view_ID=001을 지닌다. 따라서 “Application View” 즉, 응용관점에서의 분류 체계는 이 두 분류 클래스를 포함한다. 그림 6에서, Property는 관계성을 정의하기 위한 요소이며 Resource는 자원들을 정의한다. PCR, PCC, PRR은 각각 클래스와 자원과의 연관성, 클래스 간의 연관성, 자원 간의 연관성을 정의하기 위한 요소들이다. 이는 관계형 데이터베이스 모델을 고려하여 메타모델이 정의되었기 때문이며, 실질적으로 다양한 물리적 모델로 구체화되어 구현될 수 있다.

그림 8은 클래스와 자원 간의 관계성 및 자원 간의 관계성을 실질적으로 표현한 예를 보여준다. 그림에서, resource_ID가 “R0101”인 자원은 분류 클래스 “C0001”에 속하게 되고 관계성은 “is_a”이다. 이러한 관계를

PCR에서 표현하고 있다. 자원 간의 관계성은 PRR에서 표현하고 있으며, 자원 “R0101”이 다른 자원 “R1100”에 대한 구현 소스가 된다는 관계성을 나타낸다.

클래스와 인스턴스 그리고 관계성을 물리적으로 정의하는 방법은 매우 다양하다. 일반적으로 프로그래밍 언어를 이용하여 정의하는 방법과 XML, RDF, DAML 등과 같은 기술 언어를 이용하는 방법이 있다[9-11]. 그러나 최근에는 OWL(Web Ontology Language)을 이용한 접근 방법이 관심을 끌고 있다[12,13]. OWL은 XML에 보다 나은 지능성을 부여한 언어로서 RDF나 DAML을 통합한 차세대 기술 언어로 인정받고 있다.

5. eCSM 구현

이 장에서는 eCSM을 구현하기 위한 시스템 아키텍처를 소개하고 그 구현 결과를 보인다.

5.1 eCSM을 위한 시스템 아키텍처

그림 9는 eCSM을 이용한 교육용 저장소 시스템을 개발하기 위한 시스템 아키텍처를 보여 준다.

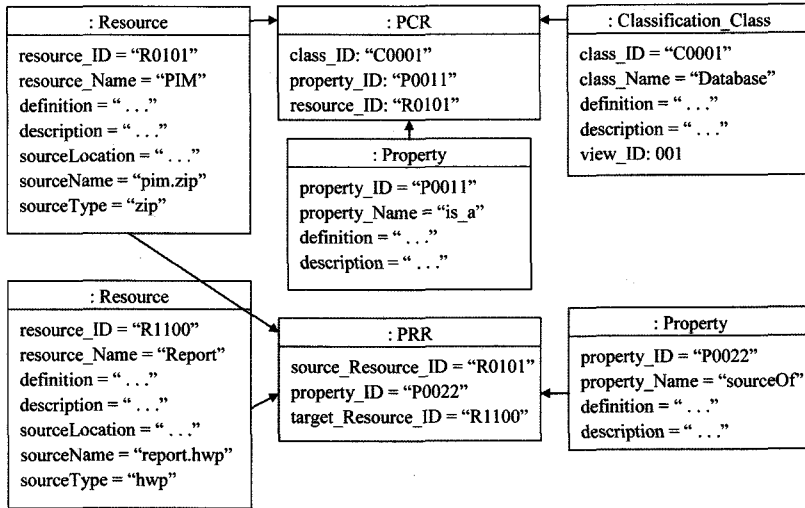


그림 8 관계성 표현의 예

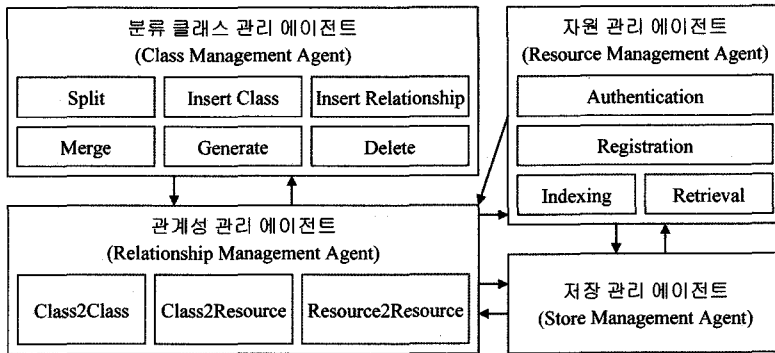


그림 9 eCSM 기반의 교육용 저장소 시스템을 위한 시스템 아키텍처

그림에서 자원 관리 에이전트는 자원에 대한 등록 및 검색을 포함하여 사용자 인증과 사용자 인터페이스를 담당한다. 사용자가 등록한 자원은 관계성 관리 에이전트에 의해 관계성을 생성하여 추가하게 된다. 등록 이벤트가 발생하면 관계성 관리 에이전트는 먼저 자원들 간의 관계성을 생성하게 되고 사용자가 선택한 분류 정보에 따라 자원 관리 에이전트에 있는 인덱스 테이블을 통하여 자동적으로 클래스와 자원 간의 관계성이 생성된다. 추가적인 관계성 즉, 자동적으로 생성될 수 없는 관계성 정보는 관리자에 의해 수동으로 생성된다. 이 논문에서 “자동적으로 생성”한다는 의미는 분류 클래스를 시스템이 자동적으로 생성하여 관리하는 것이 아니라 시스템의 자원과 함께 사용자에게 의해 정의된 분류 정보를 이용하여 자원과 클래스(분류 클래스) 간의 관계성이 자동적으로 생성, 관리됨을 의미한다.

분류 클래스 관리 에이전트는 가장 중요한 모듈 중의 하나로서, 분류 체계를 생성하고 관리하는 역할을 담당

한다. 이미 제 3장에서 언급한 다양한 연산(생성, 추가, 분할, 병합, 일반화 및 삭제)이 이루어진다. 일단 분류 클래스에 대한 연산에 대한 이벤트가 발생하면 관계성 관리 에이전트를 통해 클래스 간 관계성이 생성되거나 수정된다. 또한 자원과의 관계성이 변경되어야 할 경우 이를 위한 작업이 수행된다. 이러한 작업에 대해서 일부 자원과의 관계에 영향을 주지 않는 연산들은 자동으로 처리가 가능하지만 영향을 주는 연산의 경우 자동화가 어렵기 때문에 수동으로 조작하거나 혹은 효율적인 인터페이스를 통해 반자동화할 수 있다.

마지막으로, 저장 관리 에이전트 데이터베이스와 연동을 통해 정보들을 저장하고 필요시 적절한 정보를 각 에이전트에게 전달해 주는 역할을 담당한다. 데이터베이스와의 연동은 ODBC나 JDBC와 같은 데이터베이스 연결자(Database Connector)를 이용하여 쉽게 구현할 수 있으며 이는 이미 일반적으로 널리 알려진 기술이다.

5.2 구현 및 적용 결과

이 절에서는 제안하는 eCSM을 기반으로 한 개발한 교육용 저장소 프로토타입을 보인다. 적용 대상은 이미 언급하였듯이 대학 전산학 분야의 단기 프로젝트를 대상으로 한다. 이 논문에서의 목적인 교육용 저장소의 자원에 대한 효율적인 관리를 위한 분류 체계를 동적으로 관리하고 다양한 관점을 고려하여 정의할 수 있는 분류 모델을 개발하는데 있다. 따라서 이 논문에서는 교육용 저장소를 위한 완벽한 분류 체계를 정의하여 보이지 않는다. 그러나 프로토타입 개발을 위해 일부 필요한 분류 체계를 정의하여 이용한다.

그림 10은 사용자가 자원을 등록하기 위해 정의된 분류 체계에 따라 적절한 분류 클래스를 선택하고 관련 자원을 제공하는 과정을 보여 준다.

그림 10에서, “관점별 분류 선택” 부분은 정의된 각 관점별 분류 체계에 따라 적절한 분류 클래스를 선택하기 위한 인터페이스로서, 사용자는 상황에 따라 분류 클래스(단말 노드)를 다중 선택할 수 있다. 예를 들어, 개발에 이용된 언어가 Java, JSP 및 OWL을 이용했다면 언어 관점에서의 분류 체계에서 해당 분류 클래스들을 선택할 수 있다.

“자원정보” 부분은 실제 자원들에 대한 정의, 설명 및 제공할 결과물에 대한 정보를 제공하는 인터페이스이다. 각 자원들의 입력이 완료되면 이들 자원 간의 관계성이 자동으로 설정된다. 예를 들어, 그림 10에서, 입력 완료 이벤트가 발생할 때 “Implementation.zip”이라는 자원과 “User_guide.doc” 자원 간에는 (Implementation.zip, sour-

ceOf, User_guide.doc)이라는 관계성이 설정된다. 이는 그 역관계성인 (User_guide.doc, sourceOf, Implementation.zip)이라는 관계성과 의미적으로 동일하다.

일단 제출 과정을 거쳐 등록된 정보는 관리자의 검토 과정을 거쳐 제공된 정보의 완전성과 정확성을 검사하게 된다. 제출된 정보가 부족하거나 잘못된 정보가 있을 경우 다시 추가 하도록 요청하게 되고 이러한 반복 과정을 거쳐 최종적으로 등록을 완료하게 된다. 이 논문에서는 이러한 등록 과정을 국제 표준인 ISO/IEC 11179를 따른다[14]. ISO/IEC 11179의 목적은 데이터베이스 간의 데이터 상호운용성을 극대화하기 위한 표준이다. 즉, 데이터 요소를 등록하고 이를 활용하여 데이터의 공유 및 교환을 가능하게 한다. 이 과정에서 데이터 요소를 등록하는 절차가 요구되므로 이 표준에는 이러한 과정을 체계적으로 정의하여 표준으로 제안하였다. 이 논문에서의 등록 대상과 이 표준에서의 등록 대상이 동일하지 않으나 제출, 확인 및 등록 과정 및 이에 관여하는 사용자 등의 개념이나 절차가 동일하다. 따라서 이 논문에서 국제 표준인 ISO/IEC 11179에서 권고하고 있는 표준 등록 절차에 따라 자원을 등록하도록 한다. ISO/IEC11179에서의 표준 등록 절차를 기반으로 한 교육 자원의 등록 프로세스는 그림 11과 같다.

일단 등록이 완료되면 사용자들은 일반적으로 사용되는 검색 기능을 통해 등록된 자원들을 활용할 수 있다. 부가적으로, 검색 기능은 가장 널리 이용되는 키워드 검색을 통해 자원을 검색할 수 있으며 또한 분류 체계에

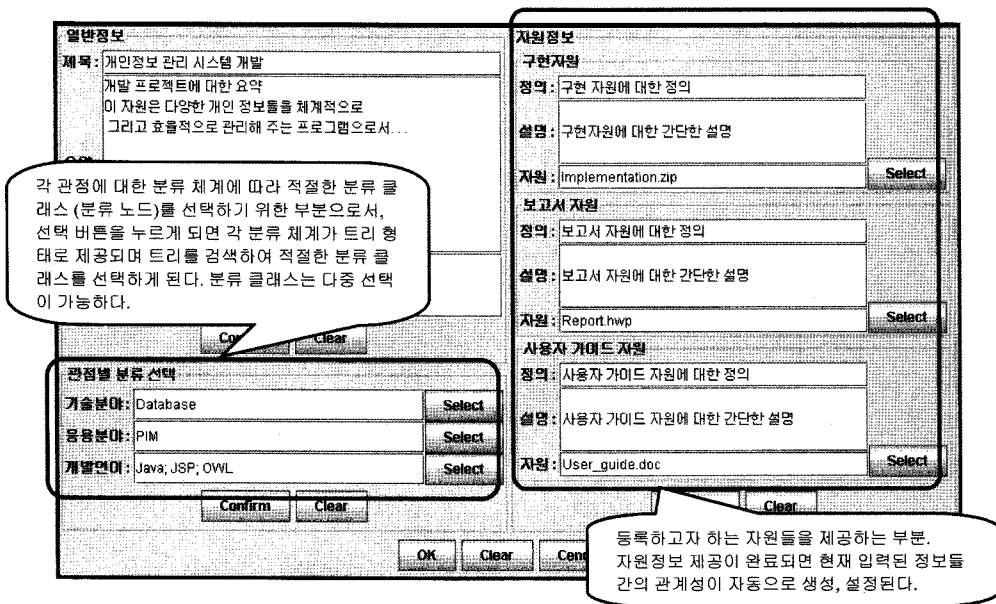


그림 10 자원 등록 화면

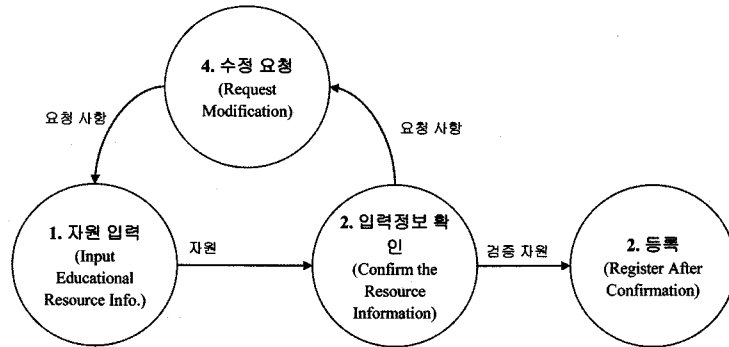


그림 11 eCSM에서의 자원 등록 프로세스

따라 검색하는 디렉토리 검색 기능을 이용하여 자원을 검색하여 활용할 수 있다.

6. 결론

이 논문에서는 교육용 저장소 시스템의 자원에 대한 일관성 있는 관리 및 활용성 향상을 위한 분류 모델인 eCSM을 제안하였다. 또한 교육용 저장소의 자원에 적합한 분류 체계를 위해 국내 주요 기관에서 이용한 과학기술분류체계를 분석하고 분류 클래스를 추가로 정의하였다. 특히 제안한 eCSM은 단순한 기술관점에 따른 분류뿐 아니라 사용자 관점(응용관점)과 언어관점을 고려함으로써 사용자들에게 활용의 용이성을 제공한다. 또한 분류 체계의 동적 관리를 위한 방법 및 그에 따른 주요 연산들을 정의하였으며, 분류 체계를 이용한 개발에 필요한 구현 모델을 정의하고 구현된 결과를 보였다.

제안한 eCSM은 먼저 교육용 저장소 관리 시스템 개발을 위한 효율적인 자원 관리 모델을 제시한다. 자원을 효율적으로 관리하고 활용하기 위해서는 효율적인 분류 체계와 그에 대한 관리가 필수적으로 요구된다. eCSM은 이러한 요구 사항에 대한 해결책을 제시해 준다. 또한 연구가 미비한 교육 자원 관리를 위한 분류 체계 마련의 기초 자료로서의 역할과 함께 보다 나은 교육용 저장소 관리 시스템 개발을 용이하게 한다.

향후 연구로는, 보다 완벽한 분류 체계를 정의하는 연구가 필요하며 이는 관련 분야의 많은 연구자들과의 많은 논의와 협력을 필요로 한다. 또한 대상 범위를 확대하고 이를 통해 보다 실용적인 시스템을 개발함으로써 제안하는 eCSM의 문제점을 상세하게 분석하는 연구가 요구된다.

참 고 문 헌

[1] UNDP(United Nations Development Programme), "United Nations Standard Products and Service

Code," White Paper, <http://www.unspsc.org>, 2001.

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

[3] UNDP(United Nations Development Programme), "Better Supply Management with UNSPSC," White Paper, <http://www.unspsc.org>, 2006.

[4] 한국학술진흥재단, "과학기술분류표," <http://www.krf.re.kr/>, 2006.

[5] 한국과학재단, "기술분류표," <http://www.kosef.re.kr/main/main.html>, 2006.

[6] 서태실, "과학기술정보 분류 현황분석," 한국과학기술정보연구원, 2006.

[7] 대한화학회, 화학정보(한국과학재단 학문분류표), <http://newkcsnet.kcsnet.or.kr/>, 2006.

[8] 김동규, 이상구, 전중훈, 최동훈, "전자 카탈로그를 위한 의미적 분류 모형," 정보과학회논문지: 데이터베이스, 제 33권 제 1호, pp. 102~116, 2006.

[9] François Yergeau, John Cowan, Tim Bray, Jean Paoli, C. M. Sperberg McQueen, Eve Maler, "Extensible Markup Language (XML) 1.1, W3C Recommendation," W3C, Feb. 2004.

[10] Dave Beckett, "RDF/XML Syntax Specification (Revised), W3C Recommendation," W3C, 2004.

[11] The DARPA Agent Markup Language, "About DAML," <http://www.daml.org/>, 2003.

[12] Deborah L. McGuinness, Frank van Harmelen, "OWL Web Ontology Language Overview, W3C Recommendation," W3C, 2004.

[13] Michael K. Smith, Chris Welty, Deborah L. McGuinness, "OWL Web Ontology Language Guide, W3C Recommendation," W3C, 2004.

[14] ISO/IEC JTC 1/SC 32, "ISO/IEC 11179: Information technology - Metadata Registries (MDR) - Part 1~6," ISO/IEC, 2005.



최 명 회

국립군산대학교 석사과정. 2006년 군산대학교 정보통계학과(이학사). 2006년~현재 군산대학교 정보통계학과(석사과정)
관심분야는 데이터베이스, 시맨틱 웹, 정보 접근 제어



정 동 원

국립군산대학교 교수. 1997년 군산대학교 컴퓨터과학과(이학사). 1999년 충북대학교 전산과(이학석사). 2004년 고려대학교 컴퓨터학과(이학박사). 1999년~2000년 ICU 부설 한국정보통신교육원(전임강사). 2000년~2001년 지구넷 부설 연구소(선임연구원). 2002년~2005년 라임미디어 테크놀로지(연구원) 2004년 고려대학교 정보통신기술연구소(연구교수). 2005년 Pennsylvania State University(PostDoc.) 2002년~현재 TTA 표준화위원회 메타데이터표준화 프로젝트 그룹 PG406(특별위원). 2005년~현재 군산대학교 정보통계학과(교수). 2006년~현재 ISO/IEC JTC1/SC32 국내위원회(전문위원). 관심분야는 데이터베이스, 시맨틱 웹, 정보 접근 제어, 유비쿼터스 컴퓨팅