

계층적으로 구조화된 이러닝 시스템을 위한 질의 처리 기법

김연희*, 김지현**

A Query Processing Method for Hierarchical Structured e-Learning System

Youn-Hee Kim*, Jee-Hyun Kim**

요약

본 논문에서는 대표적인 학습 객체 메타데이터 형식들을 통합하여 상호운용성을 제공하면서 대학 강좌를 위한 학습 객체의 메타데이터를 기술할 때 사용되는 개념과 개념들 간의 의미적 관계를 정의하는 온톨로지를 설계한다. 그리고 서로 다른 형식을 이용해 학습 객체의 메타데이터를 기술하는 여러 지역 저장소에 대해 효율적인 학습 객체 검색이 가능하도록 계층적으로 구조화된 이러닝 시스템을 구성하고 추론에 기반한 질의 처리 기법을 제안한다. 본 논문에서 제안한 계층적으로 구조화된 이러닝 시스템에서 학습 객체 온톨로지를 이용한 질의 처리 기법을 적용하면 사용자 질의에 직접적으로 관련이 있는 학습 객체와 함께 의미적 연관성이 추론된 학습 객체도 검색되어 보다 정확하고 만족도 높은 검색 서비스를 제공할 수 있다.

▶ Keyword : 이러닝, 메타데이터, 온톨로지, 질의 처리 기법

Abstract

In this paper, we design an ontology which provides interoperability by integrating typical metadata specifications and defines concepts and semantic relations between concepts that are used to describe metadata for learning objects in university courses. And we organize a hierarchical structured e-Learning system for efficient retrieval of learning objects on many local storages that use different specifications to describe metadata and propose a query processing method based on inferences. The proposed e-Learning system can provide more accurate and satisfactory retrieval service by using the designed ontology because both learning objects that be directly connected to user queries and deduced learning objects that be semantically connected to them are retrieved.

▶ Keyword : E-Learning, Metadata, Ontology, Query Processing Method

• 제1저자 : 김연희

• 투고일 : 2011. 01. 28, 심사일 : 2011. 02. 08, 게재확정일 : 2011. 02. 12.

* 부천대학 e-비즈니스과(Dept. of e-Business, Bucheon University)

** 서울대학 컴퓨터소프트웨어과(Dept. of Computer Software, Seoul University)

※ 본 논문은 2009년도 서울대학 학술연구비에 의해 연구되었음.

I. 서론

컴퓨터와 정보통신 기술을 이용해 시간과 공간에 대한 제약 없이 원하는 학습 객체(Learning Object)에 접근하여 학습 및 교육하는 방식을 의미하는 이러닝(e-Learning)은 교육 분야뿐만 아니라 모든 산업 분야에 걸쳐 큰 영향을 끼치며 빠른 성장을 거듭해왔다[1, 2]. 학습 객체는 한 단위의 학습을 위해 자유롭게 결합될 수 있는 작은 학습 컴퍼넌트로 객체 지향 개념을 적용해 학습 자료를 이해한 것이다[2]. 계속적으로 진화하는 학습 객체들을 다수의 교수자와 학습자에게 즉각적으로 제공하기 위해서 많은 이러닝 시스템들이 웹을 기반으로 개발되었다. 초기의 이러닝 학습 모델은 교수자와 학습자의 역할이 명확하게 분리되어 교수자는 학습 객체를 개발 및 관리하고, 학습자는 텍스트 기반의 단순 키워드 검색 방법을 이용해 웹 상에 분산되어 있는 학습 객체에 접근하는 방식이었다. 그러나 이러닝 산업이 급속도로 발전하고 웹 기반 응용 서비스에 대해 다양한 사용자 요구가 발생하면서 교수자와 학습자에 대해 다양한 역할의 수행이 요구되고 있다. 즉, 교수는 기존 학습 객체들 중 적합한 것을 검색한 후 그 결과물들을 통합하여 새로운 학습 객체를 개발하고, 학습자는 학습 객체의 또 다른 제공자로서 지능화된 검색 방법을 이용해 개인의 특성에 맞는 학습 객체를 검색할 수 있어야 한다. 이러한 새로운 요구를 충족시키고 학습 객체의 공유 및 재사용을 지원하기 위해 학습 객체에 대한 메타데이터(Metadata)가 활발하게 연구되어 왔다.

메타데이터는 학습 객체의 특성과 다른 학습 객체와의 연관성을 기술하여 이러닝 시스템 내에서 학습 객체의 효율적인 검색 및 관리를 지원하는 역할을 담당한다. 학습 객체를 위한 메타데이터를 기술할 때 사용하는 개념과 관련해 발생할 수 있는 의미적 모호성이나 중첩성의 문제를 해결하고 새로운 지식의 추론이 가능하도록 하기 위해서는 메타데이터와 함께 온톨로지의 역할이 요구된다. 온톨로지는 메타데이터를 기술할 때 사용하는 개념과 개념들 간의 다양한 의미적 관계를 정의함으로써 정형화된 지식의 공유와 새로운 지식의 추론을 가능하게 한다. 그리고 각기 다른 메타데이터 형식을 이용하는 여러 학습 객체 저장소들에 대해 통합된 검색 서비스를 제공하기 위해 온톨로지를 활용할 수도 있다.

따라서 본 논문에서는 우리나라에서 사용되는 대표적인 학습 객체 메타데이터 형식들을 통합하여 상호운용성을 제공하면서 학습 객체의 메타데이터를 기술하는데 필요한 개념과 그들 간의 다양한 관계를 정의하는 온톨로지를 설계한다.

학습 객체를 위한 메타데이터와 온톨로지를 이러닝 시스템에서 사용하기 위해서는 형식적으로 기술할 필요가 있다. 본 논문에서는 개념들 간의 상속 관계뿐만 아니라 다른 언어로는 표현하기 어려운 역관계, 이행성 등을 기술할 수 있는 OWL(Web Ontology Language)을 이용한다[3].

학습 객체를 위해 기술된 메타데이터와 온톨로지도 학습 객체와 마찬가지로 이러닝 시스템 내에서 효과적으로 저장 및 관리되어야 한다. 따라서 본 논문에서는 여러 지역 저장소에 분산되어 있는 학습 객체들에 대해 통합적이고 효율적인 검색이 가능하도록 계층적으로 구조화된 이러닝 시스템을 구성하여 학습 객체를 위한 메타데이터와 온톨로지를 저장하고 추론에 기반한 질의 처리 기법을 제안한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 대표적인 학습 객체 메타데이터 형식들을 비교하고 기존 이러닝 시스템의 문제점을 제시한다. 3장에서는 본 논문에서 설계한 학습 객체 온톨로지를 설명한다. 4장에서는 계층적으로 구조화된 이러닝 시스템을 제안한다. 5장에서는 본 논문에서 제안한 질의 처리 기법을 이용한 실험 결과를 제시하고 6장에서 결론을 맺는다.

II. 관련 연구

1. 학습 객체 메타데이터

일반적으로 많이 사용되는 학습 객체 메타데이터 형식으로 DC(Dublin Core), LOM(Learning Object Metadata), IMS(Instructional Management Systems) 메타데이터 모델, SCORM(Sharable Content Object Reference Model) 등이 있다. 이 가운데 IMS와 SCORM은 LOM에 기반을 두고 있기 때문에 결과적으로 가장 대표적인 메타데이터 형식은 DC와 LOM이라고 할 수 있다[2]. 국가별로 지역화된 메타데이터 형식도 존재하지만 대부분이 DC와 LOM을 기반으로 확장하거나 변형된 형태이다. 우리나라에서는 2002년에 제안된 KEM(Korea Education Metadata) v1.0이 가지고 있던 여러 가지 문제점을 보완하여 2004년도에 KERIS가 LOM에 기반한 KEM v2.0을 개발하였다. 그러나 KEM v2.0은 초·중등 교육 관련 학습 객체를 기술하기 위한 목적으로 개발되었기 때문에 고등 교육 관련 학습 객체를 기술하는데 부족한 요소들이 많아 2005년 KEM v3.0이 새롭게 제안되었다[4]. 현재 우리나라에서는 교육기관을 중심으로 KEM이 주로 사용되고 있지만 연구 분야에서는 여전히 DC와 LOM도 중요하게 활용되고 있다[2].

본 논문에서는 대학 강좌와 관련한 학습 객체를 제공하는 이러닝 시스템 개발을 목표로 대표적인 학습 객체 메타데이터 형식인 DC, LOM, KEM v3.0을 통합하는 온톨로지를 정의한다.

표 1은 DC, LOM, KEM v3.0의 특징을 비교한 결과이다.

표 1. 학습 객체 메타데이터의 비교
Table 1. Comparison of Metadata for Learning Objects

	DC	LOM	KEM v3.0
목적	웹 상에 정보 자원들의 특성 기술	일반적인 학습 객체의 특성 기술	한국 초·중·고등 교육 분야의 학습 객체 특성 기술
구성	15개의 요소	계층적으로 9개 상위 요소와 77개 하위 요소로 구성	계층적으로 9개 상위 요소와 117개 하위 요소로 구성
특징	단순한 구조에 간단한 요소들로 구성되어 메타데이터 작성이 용이함	교육적인 요소는 물론 기술적인 요소까지 학습 객체의 다양한 특성을 기술할 수 있음	LOM의 기본 요소들을 그대로 반영하고 초·중등 교육 분야 관련 요소와 저작권 관련 요소 등을 추가
단점	학습 객체의 다양한 특성을 자세히 기술하기 어려움	구조가 복잡하고 요소가 많아 메타데이터 작성이 어려움	대학 교육 분야의 학습 객체에는 적합하지 않은 요소들도 존재함

2. 이러닝 시스템과 온톨로지

온톨로지는 시맨틱 웹의 핵심 기술로 보다 지능화된 정보 자원의 검색이 가능하게 한다. 이러한 온톨로지의 역할을 이러닝 시스템에 적용하여 학습 객체에 대한 검색 효율성을 높이고자 하는 연구들이 제안되었다.

참고문헌 [2]에서는 DC, LOM, KEM v1.0 형식을 통합할 목적으로 온톨로지를 설계하였다. 그러나 많은 변화가 수용된 KEM v3.0에 적합하지 않는 요소들이 존재한다. 그리고 DC, LOM, KEM v1.0 형식에 존재하는 모든 메타데이터 요소들을 온톨로지서 클래스(Class)로 정의하였다. 하지만 메타데이터 요소들은 일반적으로 학습 객체의 특성을 설명하는 역할을 담당하므로 클래스의 속성(Property)으로 정의하는 것이 적합하다. 따라서 본 논문에서는 KEM v3.0을 통합에 고려하고 추출된 메타데이터 요소들을 온톨로지서 학습 객체 클래스의 다양한 속성으로 정의한다.

참고문헌 [5]에서는 공업계고등학교의 전기·전자·통신 관련 학과에서 수업하는 전기회로 과목의 내용을 온톨로지서 정

의하고 이를 기반으로 질의를 처리하는 이러닝 시스템을 제안하였다. 그러나 참고문헌 [5]에서 정의한 온톨로지는 학습 객체를 위한 메타데이터 형식을 정형화한 것이 아니라 특정 과목의 내용을 위한 도메인 온톨로지이다. 따라서 특정 과목 내용과 관련한 질의 처리는 가능하지만 메타데이터 형식에 포함되어 있는 요소들의 특성을 이용한 추론 기반 질의 처리가 어려운 문제가 있다. 본 논문에서는 메타데이터 요소들의 특성과 그들 간의 의미적 관계를 온톨로지서 정의하여 질의 처리에 활용함으로써 추론에 기반한 지능화된 검색이 가능하도록 한다.

참고문헌 [6]에서는 KEM v3.0 형식을 OWL로 표현하기 위한 기본적인 원칙과 OWL로 표현한 최종 결과물을 제시하였다. 그러나 참고문헌 [6]에서는 초·중·고등 교육 분야에 적용하는 것을 목표로 KEM v3.0에 포함되어 있는 대부분의 메타데이터 요소들을 OWL로 표현한다. 본 논문에서는 대학 강좌를 위한 이러닝 시스템의 구성을 목표로 하기 때문에 KEM v3.0의 메타데이터 요소들 중 대학 강좌의 학습 객체에 적합한 요소만을 추출하여 온톨로지를 설계하는데 활용한다.

3. 분산 이러닝 시스템

네트워크 기술에 기반을 두고 있는 이러닝 분야에서는 다양한 지역 저장소에 분산되어 있는 학습 객체에 대해 통합적이고 효율적인 검색을 지원하는 질의 처리 방법이 필요하다. 특히, 학습 객체 메타데이터가 이러닝 시스템을 구축하는데 중요한 요소로 인식되면서 분산된 학습 객체를 빠르고 정확하게 검색하기 위해 메타데이터를 활용하는 방법도 제안되었다.

참고문헌 [7]은 각 피어(Peer)들이 학습 객체와 함께 학습 객체 메타데이터를 저장하고 있는 P2P 환경을 고려한다. 그리고 이질적인 저장 구조와 질의어를 사용하면서 서로 다른 학습 객체 메타데이터 형식을 사용하는 피어들 중에서 사용자의 질의에 가장 적합한 피어들만을 선택하여 질의를 전달하고 처리하도록 하는데 목표를 두고 있다. 그러나 여러 학습 객체 메타데이터 형식을 통합하고 학습 객체의 개념과 개념간의 관계를 정의하는 온톨로지의 역할을 고려하지 않기 때문에 추론에 기반한 지능적인 질의 처리가 어렵다. 그리고 모든 인덱스가 슈퍼 피어(Super-peer)에 존재하기 때문에 질의 처리 시 슈퍼 피어의 부담이 크다. 따라서 본 논문에서는 분산 환경의 이러닝 시스템에서 서로 다른 메타데이터 형식을 이용하는 학습 객체 저장소들을 통합하고 학습 객체의 다양한 의미적 관계에 기반한 추론 질의를 지원하도록 온톨로지를 구성하고 이를 이용한 질의 처리 기법을 제안한다.

III. 학습 객체 온톨로지

이번 장에서는 분산 환경의 이러닝 시스템에서 메타데이터 형식들 간의 상호운용성을 제공하면서 통합된 형태의 효율적인 학습 객체 검색이 가능하도록 본 논문에서 설계한 학습 객체 온톨로지에 대해 설명한다.

1. 대학 강좌를 위한 학습 객체의 특성

본 논문에서는 각 지역 저장소가 DC, LOM, KEM v3.0 중 하나의 형식을 이용해 학습 객체의 특성을 기술하는 이러닝 시스템을 위해 학습 객체 온톨로지를 설계하고 이것을 OWL로 표현한다. 특히, 본 논문에서는 대학의 원격 강의를 위한 학습 객체들을 대상으로 하기 때문에 대학 강좌의 특성을 고려하여 온톨로지를 설계한다.

일반적으로 학습 객체들은 텍스트 기반, 이미지 기반, 동영상 기반 등과 같이 물리적 형태에 따라 구분될 수 있다. 하지만 대학 강좌를 구성하는 학습 객체들은 단순히 물리적으로 구분하는 것 보다 논리적인 의미에 따라 구분해야 한다. 본 논문에서는 대학 강좌에서 활용되는 학습 객체의 특성을 파악하기 위해 MIT 대학에서 운영하는 OCW(OpenCourseWare) 웹 사이트를 통해 제공되는 학습 객체들을 분석하였다. OCW는 MIT 대학에서 운영되는 학부 및 대학원 강좌에 관련된 학습 자료를 무료로 제공하기 위해 2002년부터 운영되고 있는 웹 사이트이다[8]. OCW는 대학 강좌의 학습 자료를 웹 상에서 제공하는 대표적인 사이트이고 강좌의 운영과 구조는 대학들마다 유사하기 때문에 본 논문에서는 OCW를 분석의 대상으로 선택하였다.

대학 강좌에서 제공하는 학습 객체들은 계층적인 구조로 구성된다. 강좌는 여러 개의 강 또는 주 단위로 구성되어 있고 강좌의 각 강이나 각 주마다 관련된 강의록, 강의동영상, 과제, 평가, 교재 등이 존재한다. 따라서 본 논문에서는 학습 객체 타입(LRType)이라는 개념을 클래스로 정의하고 강좌(Course), 강(Lecture), 강의록(LectureNote), 강의동영상(VideoLecture), 과제(Assignment), 평가(Exam), 교재(Reading) 클래스를 하위 클래스로 포함하도록 온톨로지를 설계한다.

대학 강좌의 학습 객체에 대한 일반적인 질의 유형을 분석해보면 특정 주제, 즉 전공 분야에 관련된 학습 객체를 검색하는 경우가 많다. 대학 강좌의 주제는 다양하긴 하지만 관련성에 따라 계층적인 분류가 가능하다. 특히, 컴퓨터 과학 분야에서 대표적으로 많이 사용되는 주제의 분류는 ACM

CCS(Computing Classification System)이다[9]. ACM CCS는 주제를 4단계의 계층 구조로 분류한다. 본 논문에서는 ACM CCS에 따라 학습 객체의 주제를 분류하고 온톨로지 설계에 반영하기 위해 학습 주제(ACMTaxonomy)라는 개념을 클래스로 정의하고 ACM CCS의 분류에 따라 각각의 주제들을 하위 클래스로 정의한다. 표 2는 본 논문에서 ACM CCS의 주제 분류 중 “Software” 분야의 계층 구조 일부분을 온톨로지로 정의하여 OWL로 기술한 결과물이다.

표 2. 학습 주제 클래스 정의
Table 2. Definition of ACMTaxonomy Class

```
<owl:Class rdf:ID="ACMTaxonomy"/>
<owl:Class rdf:ID="Software">
  <rdfs:subClassOf rdf:resource="#ACMTaxonomy"/>
</owl:Class>
<owl:Class rdf:ID="ProgrammingLanguages">
  <rdfs:subClassOf rdf:resource="#Software"/>
</owl:Class>
<owl:Class rdf:ID="LanguageClassifications">
  <rdfs:subClassOf rdf:resource="#ProgrammingLanguages"/>
</owl:Class>
<owl:Class rdf:ID="ObjectOrientedLanguages">
  <rdfs:subClassOf rdf:resource="#LanguageClassifications"/>
</owl:Class>
```

2. 대학 강좌 학습 객체의 메타데이터 요소

본 논문에서는 LOM과 KEM v3.0에서 제공하는 메타데이터 요소 중 대학 강좌의 학습 객체 특성을 기술하는데 필요한 요소들을 추출하고 DC에서 제공하는 요소들과의 연관 관계를 파악하여 온톨로지를 정의한다. 예를 들어, KEM의 Pedagogy 관련 요소들은 초·중등 교육의 ICT 교수학습 활동에 관한 것이므로 본 논문에서는 고려하지 않는다. 그리고 학습 객체의 주요특성을 우선적으로 고려하기 위해서 LOM과 KEM에서 저작권 범주와 주석 범주에 속하는 메타데이터 요소들도 배제한다.

표 3. 온톨로지 정의를 위해 추출한 메타데이터 요소의 일부
Table 3. Subset of Metadata for Ontology Definition

LOM과 KEM에서 추출한 메타데이터 요소		DC에서 추출한 메타데이터 요소
general	title	dc:title
lifecycle	contribute	role
		entity
relation	hasPart	dc:source
	hasVersion	
	hasFormat	
	isBasedOn	
	replaces	

본 논문에서는 일반(general), 생명주기(lifecycle), 메타-메타데이터(meta-metadata), 기술(technical), 교육(educational), 관계(relation), 분류(classification)로 구성된 7개의 최상위 범주와 36개의 하위 요소로 학습 객체 온톨로지를 구성한다.

표 3은 대학 강좌와 관련한 학습 객체의 특성을 기술하기 위해 LOM과 KEM에서 추출한 메타데이터 요소 중 일반, 생명주기, 관계 범주에 속하는 일부 요소와 그것과 같은 의미의 DC 메타데이터 요소를 보여준다.

표 3에서 학습 객체의 일반적인 특성을 의미하는 일반 범주에서 <title> 요소만 언급했지만 실제로는 학습 객체의 사용 언어를 기술하는 <language>, 학습 객체의 주요 키워드를 기술하는 <keyword> 등 7개의 하위 요소가 존재한다.

LOM과 KEM에서 생명 주기 범주는 학습 객체의 버전 정보와 학습 객체의 제작에 기여한 사람이나 조직 등을 기술하는 요소를 포함하는데 본 논문에서는 그러한 요소들을 그대로 온톨로지 정의에 반영한다. 다만 LOM과 KEM에서는 학습 객체의 제작에 기여한 사람이나 조직이 누구인지를 <entity> 요소로 기술하고 그 역할의 종류는 <role> 요소를 이용해 기술하는 반면 DC에서는 역할의 종류를 별도로 언급하지 않고 <creator>, <publisher>, <contributor>라는 요소를 이용한다. 따라서 본 논문에서는 제약 사항은 있지만 <entity> 요소와 <creator>, <publisher>, <contributor> 요소가 같은 의미의 요소라는 것을 온톨로지에 정의한다.

LOM과 KEM에서 관계 범주에는 다른 학습 객체와의 연관 관계가 어떤 종류인지를 기술하는 <kind> 요소와 관계를 맺고 있는 다른 학습 객체가 무엇인지를 기술하는 <resource> 요소가 포함되어 있는데 <kind> 요소는 hasPart, isPartOf 등 10개의 값을 가질 수 있다. 그런데 hasPart, isPartOf 등과 같은 관계의 종류는 단순히 값으로 처리하는 것 보다 질의 처리 시 추론에 활용할 수 있도록 온톨로지를 이용해 정형화시킬 필요가 있다. 따라서 본 논문에서는 표 3에서 학습 객체의 의미적 관계를 나타내는 10개 요소를 관계 범주 아래에 하위 속성으로 정의하고 그들의 특성이나 그들 사이의 관계를 학습 객체 온톨로지에 정의한다.

LOM, KEM, DC에서 추출한 메타데이터 요소를 개념을 의미하는 클래스로 정의할 것인지 개념에 대한 특성을 의미하는 속성으로 정의할 것인지는 온톨로지를 설계하는데 중요한 선택 사항이 된다. 메타데이터 요소들이 일반적으로 학습 객체의 특성이나 학습 객체간의 의미적 연관성을 설명하는 역할을 담당하기 때문에 본 논문에서는 학습 객체의 개념을 클래스로 정의하고 LOM, KEM, DC에서 추출한 메타데이터 요

소들은 학습 객체 클래스의 속성으로 정의한다. 그리고 계층적으로 구조화되어 있는 학습 객체 메타데이터 요소들의 특성을 온톨로지에 반영하기 위해 subPropertyOf 관계를 이용해 상위 속성과 하위 속성을 정의한다. 그리고 속성이 가질 수 있는 값의 범위가 일반적인 데이터 값인 경우는 DatatypeProperty로 정의하고 값의 범위가 별도로 정의된 다른 클래스인 경우는 ObjectProperty로 정의한다. 표 4는 추출한 메타데이터 요소들의 특성과 그들 간의 의미적 관계를 온톨로지로 정의하고 OWL로 기술한 결과물의 일부분을 보여준다.

표 4. 학습 객체 온톨로지의 OWL 표현
Table 4. Representation of OWL for Learning Object Ontology

```
<owl:Class rdf:ID="LearningObject"/>
<owl:DatatypeProperty rdf:ID="title">
  <rdfs:subPropertyOf rdf:resource="#general"/>
  <rdfs:domain rdf:resource="#LearningObject" />
  <rdfs:range rdf:resource="#xsd:string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="dcTitle">
  <owl:equivalentProperty rdf:resource="#title"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="hasPart">
  <rdfs:subPropertyOf rdf:resource="#relation"/>
  <rdfs:type rdf:resource="#owl:TransitiveProperty" />
  <rdfs:domain rdf:resource="#LearningObject" />
  <rdfs:range rdf:resource="#LearningObject" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isPartOf">
  <owl:inverseOf rdf:resource="#hasPart" />
</owl:ObjectProperty>
```

표 4에서 학습 객체는 클래스로 정의하여 개념화하였다. 따라서 검색의 대상이 되는 모든 학습 객체는 학습 객체 클래스의 인스턴스가 된다. title 속성은 학습 객체의 제목을 기술하는 속성이기 때문에 학습 객체 클래스를 도메인 클래스로 지정하고 값의 범위는 문자열이기 때문에 DatatypeProperty로 정의하였다. 그리고 LOM과 KEM에서 추출한 title 속성은 DC에서도 제공되는 속성이므로 equivalentProperty를 이용해 동의어 관계임을 정의하였다. hasPart는 학습 객체가 다른 학습 객체들로 구성되어 있는 포함 관계를 나타내는 속성이기 때문에 학습 객체 클래스를 도메인 클래스로 지정하고 값의 범위도 학습 객체 클래스로 지정하였다. 그리고 hasPart는 관계 범주에 속하는 속성이기 때문에 subPropertyOf를 이용해 관계 속성의 하위 속성으로 정의하였다. 또한 hasPart는 이행적 특성을 가지고 있기 때문에 TransitiveProperty로 정의하였고 isPartOf와 역관계에 있음을 온톨로지에 정의하였다.

본 논문에서 제안한 학습 객체 온톨로지는 세 개의 대표적인 학습 객체 메타데이터 형식들을 통합하여 상호운용성을 제

공할 수 있다. 따라서 분산 환경의 이러닝 시스템에서 학습 객체 메타데이터 정보를 관리하고 검색하기 위한 통합 스키마로서의 역할을 담당한다. 그리고 정의된 온톨로지를 이용해 학습 객체의 중요한 특성은 물론 다른 학습 객체와의 의미적 연관성을 명확하게 정의함으로써 보다 정확한 검색을 지원할 수 있다.

3. 학습 객체 검색을 위한 질의 유형 분류

학습 객체를 검색하기 위해 교수자나 학습자가 제공하는 질의 조건은 다양하지만 대부분 검색하고자 하는 학습 객체의 특정 속성에 대한 값을 질의 조건으로 제시하는 것이 일반적이다. 본 논문에서는 이러한 질의 특성을 고려하여 학습 객체 검색을 위한 대표적인 질의 유형을 선별하고 각 질의 유형별로 효율적인 처리 기법을 제안한다.

OWL로 기술된 학습 객체 온톨로지와 메타데이터 정보는 (주어 서술어 목적어)로 구성된 트리플 구조의 문장 형태로 표현될 수 있다. 따라서 본 논문에서는 학습 객체 검색을 위한 질의를 (주어 서술어 목적어)의 트리플 형태로 간단히 표현한다. 질의의 트리플 표현에서 주어는 검색하고자 하는 학습 객체를 의미하기 위해 “?”로 나타낸다. 서술어는 질의 조건으로 주어진 속성의 이름을 의미하고 목적어는 질의 조건으로 주어진 속성의 값이 된다.

3.1 일반적인 속성-값 기반 질의

검색하고자 하는 학습 객체가 가지고 있는 속성의 이름과 그 값이 질의 조건으로 주어지는 유형이다. 이러한 질의 유형은 질의 조건으로 주어진 속성과 그것의 값이 메타데이터에 직접 기술되어 있는 학습 객체를 검색하면 된다. 제목이 “arrays”인 학습 객체를 검색하라는 질의 유형이 그 예가 될 수 있는데 이것은 (? title arrays)의 형태로 간단히 표현할 수 있다. 학습 객체에 대한 질의가 대부분 이 유형에 속하기 때문에 본 논문에서는 학습 객체가 가지고 있는 속성의 이름과 그 값에 대한 메타데이터 정보를 관계형 데이터베이스의 테이블에 저장하여 관리함으로써 이러한 유형의 질의 처리를 지원한다.

3.2 속성의 역관계 기반 질의

학습 객체 온톨로지에서 역관계가 성립한다고 정의된 속성들의 의미적 관계를 이용한 질의 유형이다. 이것은 학습 객체 온톨로지를 고려하지 않는 기존 이러닝 시스템에서는 처리가 어렵다. 본 논문에서는 이러한 질의 유형에 대해 질의 조건으로 주어진 속성과 그것의 값이 메타데이터에 직접 기술되어 있는 학습 객체를 검색하는 것은 물론 역관계가 성립하는 속

성에 대한 질의를 추가로 생성하여 함께 처리한다. 예를 들어 hasFormat 속성은 isFormatOf 속성과 역관계로 정의되어 있어 본 논문에서는 질의 처리 시 이 정보를 고려한다. 즉, 동영상 형태의 학습 객체 A와 동일한 내용을 포함하고 있지만 다른 형태로 존재하는 학습 객체를 검색하고자 한다면 (A hasFormat ?)로 표현된 질의와 함께 (? isFormatOf A)로 표현된 질의도 함께 처리해야 한다. 그러나 이러한 질의 확장을 질의 요청 시 실시간으로 수행하게 되면 질의 처리 시간이 길어질 수 있기 때문에 본 논문에서는 역관계로 정의된 속성에 대해서는 메타데이터 정보를 미리 확장하여 저장해둠으로써 질의 처리에 소요되는 시간을 줄이고자 한다.

3.3 속성의 이행성 기반 질의

학습 객체 온톨로지에서 이행적인 특징이 존재한다고 정의된 속성을 이용한 질의 유형이다. 이 유형도 학습 객체 온톨로지를 고려하지 않는 기존 이러닝 시스템에서는 처리가 어렵다. 본 논문에서는 이러한 질의 유형에 대해 질의 조건으로 주어진 속성과 그것의 값이 메타데이터에 직접 기술되어 있는 학습 객체를 검색한 후 그 속성의 이행성에 기반해 새로운 질의를 추가로 생성하여 함께 처리한다. 예를 들어, hasPart 속성은 이행적 특징을 가지고 있다. 만약, 학습 객체 A를 구성하고 있는 하위 학습 객체들을 검색하고자 한다면 (A hasPart ?)로 표현된 질의를 1차적으로 처리하고 나서 검색 결과로 주어진 학습 객체 B에 대해 (B hasPart ?)로 표현된 질의를 추가로 생성하여 처리해야 보다 정확한 검색 결과를 얻을 수 있다. 본 논문에서는 학습 객체에 대한 메타데이터 정보를 관계형 데이터베이스에 테이블 형태로 저장하기 때문에 이행성에 기반한 질의를 처리하기 위해서는 반복적인 자기 조인이 필요한데 이것은 질의 처리의 비효율성을 초래할 수 있다. 하지만 앞서 설명한 대로 대학 강좌를 위한 학습 객체들은 최대 3단계의 계층 구조를 가지고 있기 때문에 단 한번의 자기 조인만으로도 원하는 검색 결과를 얻을 수 있다. 본 논문에서 정의한 학습 객체 온톨로지에는 hasPart, hasVersion, hasFormat이 이행성을 가지고 있다.

3.4 학습 객체의 특성 기반 질의

대학 강좌를 위한 학습 객체들이 계층적인 구조를 가지고 있으므로 본 논문에서는 이 구조 내에 존재하는 개념적 상속 관계를 질의 처리에 고려한다. 특히, 본 논문에서 정의한 온톨로지서 학습 객체의 주제 또는 전공 분야를 의미하는 taxonEntry와 학습 객체의 주요 키워드를 의미하는 keyword 속성은 계층적인 구조를 따라 의미적으로 상속될 수 있다. 다만, taxonEntry는 상위 학습 객체에서 하위 학습

객체로 상속되고 keyword는 하위 학습 객체에서 상위 학습 객체로 상속되는 차이점이 존재한다. 예를 들어, “programming languages” 주제와 관련한 학습 객체를 검색하고자 한다면 (? taxonEntry ProgrammingLanguages)로 표현된 질의를 1차적으로 처리하고 나서 검색 결과로 주어진 학습 객체 A에 대해 (A hasPart ?) 라는 질의를 추가로 생성하여 처리해야 보다 정확한 검색 결과를 얻을 수 있다.

또한 본 논문에서는 학습 객체 온톨로지에 정의한 학습 객체 주제의 계층적인 관계를 질의 처리 시 고려한다. 예를 들어 “programming languages” 주제와 관련한 학습 객체를 검색하고자 한다면 표 2에서 제시한 계층 관계를 고려하여 (? taxonEntry ProgrammingLanguages)로 표현된 질의와 함께 하위 주제에 대한 (? taxonEntry LanguageClassifications), (? taxonEntry ObjectOrientedLanguages)의 질의 문장을 추가로 생성하여 처리하면 의미적으로 관련된 많은 학습 객체들을 검색 결과로 얻을 수 있다. 그러나 이러한 추론에 기반한 질의 확장을 한 곳에서 모두 처리한다면 부담이 커질 수밖에 없다. 따라서 본 논문에서는 추론 질의 처리의 부담을 효율적으로 분배하면서 학습 객체가 분산되어 존재하는 환경에 적합한 이러닝 시스템을 제안한다.

IV. 계층적으로 구조화된 이러닝 시스템

본 논문에서는 여러 개의 학습 객체 저장소가 분산되어 존재하는 이러닝 시스템에서 학습 객체 온톨로지와 메타데이터를 이용해 학습 객체 검색의 효율성을 높이고자 한다. 특히, 학습 객체 온톨로지에 기반한 추론 질의 처리를 통해 정확하고 풍부한 검색 결과를 반환하는 이러닝 시스템 개발에 목표를 둔다.

1. 시스템 구성

본 논문에서는 3-레이어의 계층적인 구조를 가지는 이러닝 시스템을 제안한다. 본 논문에서 제안한 이러닝 시스템은 그림 1과 같이 온톨로지 관리자(Ontology Manager), 그룹 관리자(Group Manager), 학습 객체 저장소(Learning Object Storage)로 구성된다. 온톨로지 관리자가 학습 객체 검색을 위한 질의를 요청받으면 계층적인 시스템 구조를 따라 질의에 가장 적합한 학습 객체 저장소를 검색하고 실제 질의 처리는 선택된 학습 객체 저장소에서 이루어진다.

온톨로지 관리자는 학습 객체 온톨로지 정보를 관리하고 사용자가 요청한 질의를 여러 학습 객체 메타데이터 형식에

맞게 1차적으로 확장하는 역할을 담당한다. 본 논문에서 제안한 이러닝 시스템은 여러 메타데이터 형식을 함께 사용하기 때문에 각 형식에 맞게 사용자 질의를 변환할 필요가 있다. 그리고 온톨로지 관리자는 학습 객체 주제의 계층적인 분류 정보도 별도로 저장하고 있기 때문에 사용자가 요청한 질의가 학습 객체 주제에 대한 조건을 제시하고 있다면 주제의 계층적 분류에 기반한 추론이 가능하도록 질의를 확장하는 역할도 담당한다. 온톨로지 관리자에 의해 확장된 질의는 처리가 가능한 그룹 관리자들에게 전달된다.

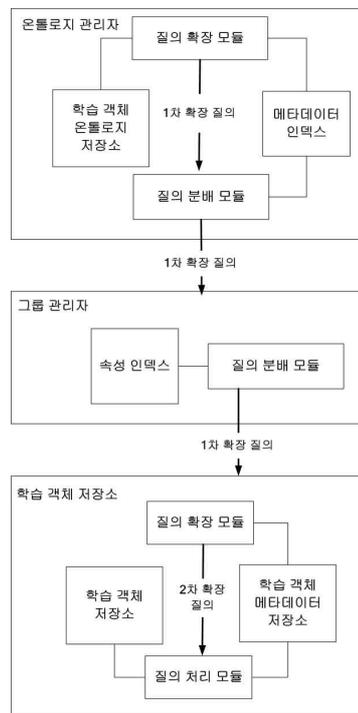


그림 1. 계층적으로 구조화된 이러닝 시스템의 구성
Fig 1. Organization of Hierarchical Structured e-Learning System

그룹 관리자는 그룹 내에 존재하는 여러 학습 객체 저장소들을 관리한다. 그리고 질의 처리에 적합한 학습 객체 저장소를 선별하여 온톨로지 관리자로부터 1차적으로 확장되어 전달된 질의를 분배한다. 학습 객체 저장소의 논리적 그룹을 구성하는 기준은 다양하지만 본 논문에서는 같은 메타데이터 형식을 사용하는 학습 객체 저장소들을 그룹으로 구성한다.

학습 객체 저장소는 학습 객체와 함께 학습 객체에 대한 메타데이터 정보를 저장 및 관리하고 사용자가 원하는 학습 객체를 검색하는 질의를 실제로 처리한다. 특히, 학습 객체

저장소는 속성의 이행성에 기반한 추론 질의와 학습 객체의 특성에 기반한 추론 질의가 가능하도록 그들 관리자로부터 전달받은 질의를 2차적으로 확장한 후 처리를 한다. 따라서 질의 조건으로 제시된 내용과 직접적으로 연관되어 있는 학습 객체뿐만 아니라 의미적 연관성이 추론된 학습 객체도 검색하여 사용자에게 반환할 수 있다.

본 논문에서 계층적인 구조로 이러닝 시스템을 구성하는 이유는 교수자나 학습자가 요청한 질의의 처리에 가장 적합한 학습 객체 저장소를 찾는 데 필요한 통신 비용을 최대한 줄이면서도 추론에 기반한 정확한 검색이 가능하도록 하기 위함이다.

2. 온톨로지 관리자의 구성

학습 객체 온톨로지 정보를 저장하고 사용자가 요청한 질의를 1차적으로 확장하는 것이 온톨로지 관리자의 가장 중요한 역할이다. 따라서 온톨로지 관리자는 학습 객체 온톨로지 정보를 효과적으로 저장할 수 있는 구조가 필요하다. 본 논문에서는 학습 객체 온톨로지 정보를 저장하기 위해 관계형 데이터베이스의 테이블을 이용한 저장 구조를 제안한다. 그림 2는 3장에서 정의한 학습 객체 온톨로지의 일부분을 본 논문에서 제안한 저장 구조에 저장한 예를 보여준다.

Class Table		Property Table				
ID	Name	ID	Name	Domain	Range	Flag
1	LearningObject	1	general	-	-	abstract
2	ACMTaxonomy	1.2	title	1	string	data
		9.2.1	taxonEntry	1	2	object

ACMTaxonomy Table		OntologyMapping Table	
ID	Name	ID	EquivalentProperty
D	Software	1.2	dcTitle
D.3	ProgrammingLanguages	9.2.1	dcSubject
D.3.2	LanguagesClassifications		
D.3.2.12	ObjectOrientedLanguages		

PropertyRelation Table		
SourceProperty	TargetProperty	Flag
hasPart	isPartOf	Y
hasFormat	isFormatOf	Y
hasVersion	isVersionOf	Y

그림 2. 학습 객체 온톨로지 정보의 저장 예
Fig 2. Example of Storing Learning Object Ontology

그림 2에서 Class 테이블은 온톨로지에 정의되어 있는 모든 개념, 즉 클래스에 대한 정보를 저장한다. Class 테이블에는 클래스를 구분하는 아이디와 이름을 저장하는 필드가 존재한다. 본 논문에서는 클래스를 구분하는 아이디를 Dewey 방식을 이용하여 생성한다. Dewey는 계층 구조를 쉽게 표현할 수 있는 장점을 가지고 있는 레이블링 기법으로 클래스들을 트리 형태의 계층 구조로 구성한 후 상위 클래스의 레이블과 동일 레벨에서 현재 클래스의 순서를 결합하여 레이블을 생성

한다[10].

Property 테이블은 클래스의 특성이거나 다른 클래스와의 관계를 표현하는 속성에 대한 정보를 저장한다. Property 테이블에서 ID 필드와 Name 필드는 속성을 구분하기 위한 아이디와 이름을 각각 저장하고 Domain 필드는 속성이 속한 클래스의 아이디를 저장한다. Range 필드는 속성이 가질 수 있는 값에 대한 정보를 저장하고 Flag 필드는 속성의 종류를 구분한다. 예를 들어, title 속성은 학습 객체 클래스의 특성이기 때문에 Domain 필드에는 학습 객체 클래스의 아이디인 1이 저장되고 문자열 타입의 값을 가지기 때문에 Range 필드에는 string 값이 저장된다. 그리고 Flag 필드에는 DatatypeProperty를 의미하는 data가 저장된다. taxonEntry는 학습 객체 클래스와 학습 객체 주제 클래스의 관계를 표현하는 속성이기 때문에 Range 필드에는 ACMTaxonomy 클래스의 아이디인 2가 저장되고 Flag 필드에는 ObjectProperty를 의미하는 object가 저장된다. general 속성은 개념적으로 존재하는 속성임을 표현하기 위해 Flag 필드에 abstract가 저장된다. 속성을 구분하는 아이디도 Dewey 방식을 이용하여 생성한다. 예를 들어 general 속성의 하위 속성인 title은 general 속성의 아이디 1과 하위 속성 중 자신의 순서 정보 2를 결합한 1.2를 아이디로 지정한다.

ACMTaxonomy 테이블은 ACM CCS에 따라 분류된 학습 객체 주제 정보를 저장한다. ACMTaxonomy 테이블에는 주제를 구분하는 아이디와 이름을 저장하는 필드가 존재한다.

OntologyMapping 테이블은 equivalentProperty를 이용해 동의어로 정의된 속성들에 대한 정보를 저장한다. 특히, DC는 KEM이나 LOM과는 다른 구성의 메타데이터 형식이기 때문에 OntologyMapping 테이블은 KEM이나 LOM에 속해 있는 속성과 같은 의미를 가지는 DC의 속성을 검색할 수 있도록 지원한다. 예를 들어, 아이디가 1.2인 title 속성은 dcTitle 속성과 같은 의미이다. OntologyMapping 테이블에 저장된 정보는 서로 다른 메타데이터 형식에 존재하는 동의어 속성을 검색하여 온톨로지 관리자가 1차적으로 질의를 확장하는데 활용한다.

PropertyRelation 테이블은 속성의 역관계와 이행성에 대한 정보를 저장한다. 예를 들어, hasPart는 isPartOf와 역관계에 있고 hasPart는 이행성을 가지고 있음을 표현하기 위해 SourceProperty 필드에는 hasPart, TargetProperty 필드에는 isPartOf가 저장된다. 그리고 Flag 필드에는 Y가 저장된다.

그림 2의 저장 구조를 이용한 온톨로지 관리자의 1차 질의 확장 규칙은 표 5와 같다.

표 5. 온톨로지 관리자의 1차 질의 확장 규칙
Table 5. 1st Query Expansion Rules for Ontology Manager

규칙 1	메타데이터 형식에 기반한 질의 확장 1.1 Property 테이블에서 질의 조건으로 제시된 속성의 아이디를 검색한다. 1.2 Ontology/Mapping 테이블에서 검색된 속성과 동의어인 속성을 검색한다. 1.3 동의어 속성을 이용해 새로운 질의 문장을 생성한다.
규칙 2	학습 객체의 주제 분류에 기반한 질의 확장 2.1 ACMTaxonomy 테이블에서 질의 조건으로 주어진 학습 객체 주제의 아이디를 검색한다. 2.2 아이디를 이용해 ACMTaxonomy 테이블에서 하위 주제를 검색한다. 2.3 검색된 하위 주제를 이용하여 새로운 질의 문장을 생성한다.

그림 2에서 제시한 저장 구조와 함께 표 5의 질의 확장 규칙을 이용해 새로운 질의 문장이 생성되면 온톨로지 관리자는 질의와 관련이 있는 그룹 관리자들을 선택하여 질의 문장을 전달한다. 따라서 온톨로지 관리자에는 사용자가 요청한 질의와 관련된 그룹 관리자를 빠르게 선택할 수 있는 인덱스 구조가 필요하다. 본 논문에서는 이를 위해 그림 3과 같은 인덱스 구조를 제안한다.

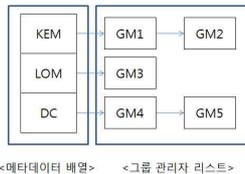


그림 3. 메타데이터 인덱스 구성 예
Fig 3. Example of Metadata Index

그림 3은 온톨로지 관리자에서 유지하는 메타데이터 인덱스의 구성 예를 보여준다. 메타데이터 인덱스는 특정 메타데이터 형식을 이용하는 학습 객체 저장소들을 관리하는 그룹 관리자에 대한 정보를 저장하는데 메타데이터 배열과 그룹 관리자 리스트로 구성된다. 예를 들어, KEM 메타데이터 형식을 이용하는 학습 객체 저장소가 소속된 그룹 관리자는 GM1과 GM2임을 인덱스를 이용해 쉽게 검색할 수 있다.

3. 그룹 관리자의 구성

그룹 관리자는 같은 메타데이터 형식을 이용하는 학습 객체 저장소들의 그룹을 관리한다. 특히 온톨로지 관리자로부터 전달받은 질의를 분석한 후 질의 처리에 가장 적합한 학습 객체 저장소를 선별하여 질의를 재분배하는 것이 그룹 관리자의

가장 중요한 역할이다. 따라서 그룹 관리자는 특정 속성과 관련한 메타데이터 정보를 유지하고 있는 그룹 내 학습 객체 저장소를 빠르게 검색할 수 있도록 지원하는 인덱스 구조가 필요하다. 본 논문에서는 이를 위해 그림 4와 같은 인덱스 구조를 제안한다.

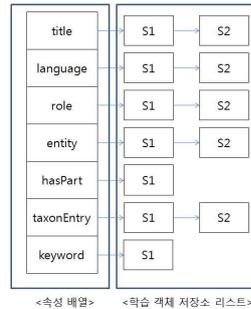


그림 4. 속성 인덱스 구성 예
Fig 4. Example of Property Index

그림 4는 KEM 메타데이터 형식을 이용하는 학습 객체 저장소들로 구성된 그룹의 관리자를 위한 속성 인덱스 구성 예를 보여준다. 속성 인덱스는 특정 속성에 관련한 메타데이터 정보를 유지하고 있는 학습 객체 저장소에 대한 정보를 저장하기 위해 속성 배열과 학습 객체 저장소 리스트로 구성된다. 예를 들어 온톨로지 관리자로부터 전달된 질의가 title 속성과 keyword 속성에 대한 질의 조건이 제시된 것이라면 두 개의 속성 모두와 관련되어 있는 S1 학습 객체 저장소로 질의를 전달한다.

4. 학습 객체 저장소의 구성

학습 객체와 학습 객체에 대한 메타데이터 정보를 관리하면서 실제로 질의를 처리하는 것이 학습 객체 저장소의 가장 중요한 역할이다. 따라서 학습 객체 저장소는 학습 객체의 메타데이터 정보를 효과적으로 저장하면서 2차적인 질의 확장에 필요한 정보를 유지할 수 있는 저장 구조가 필요하다. 본 논문에서는 이를 위해 그림 5와 같이 관계형 데이터베이스의 테이블을 이용한 저장 구조를 제안한다. 그림 5는 KEM 형식을 이용해서 기술된 학습 객체의 메타데이터 일부분을 본 논문에서 제안한 저장 구조에 저장한 예를 보여준다.

그림 5에서 Triple 테이블은 학습 객체에 대해 기술된 메타데이터 정보를 저장한다. 메타데이터 정보는 트리플 구조로 표현할 수 있기 때문에 Triple 테이블은 Subject, Property, Object 필드로 구성한다. Subject 필드는 메타데이터 정보를 가지고 있는 주체가 되는 학습 객체를 저장한다. Property

필드는 학습 객체가 가지고 있는 속성의 이름을 저장하고 Object 필드는 속성의 값을 저장한다. Triple 테이블의 첫 번째 튜플은 LN1 학습 객체의 title 속성 값이 “arrays” 라는 의미이다.

Triple Table		
Subject	Property	Object
LN1	title	arrays
VL1	language	en
L1	keyword	C++
C1	taxonEntry	ObjectOrientedLanguages

hasFormat Table		hasPart Table	
Subject	Object	Subject	Object
LN1	VL1	L1	LN1
		L1	VL1
		C1	L1

hasVersion Table	
Subject	Object
-	-

그림 5. 학습 객체 메타데이터 정보의 저장 예
Fig 5. Example of Storing Learning Object Metadata

그림 5에서 hasPart 테이블은 학습 객체를 구성하고 있는 하위 학습 객체들과의 관계를 저장한다. 그림 5에서 hasPart 테이블은 C1 강좌가 L1 강의로 구성되어 있고 L1 강은 강의 룩 LN1과 강의동영상 VL1으로 구성되어 있다는 정보를 저장하고 있다. hasPart 속성은 isPartOf 속성과 역관계가 성립하기 때문에 질의 처리 시 이 관계를 고려해야 정확한 검색 결과를 반환할 수 있다. 따라서 본 논문에서는 isPartOf 속성으로 기술되어 있는 메타데이터 정보를 hasPart 속성의 형태로 변형하여 hasPart 테이블에 함께 저장한다. 예를 들어, (B isPartOf A)의 형태로 기술되어 있는 메타데이터 정보는 (A hasPart B)의 형태로 바꾸어 hasPart 테이블에 미리 저장해두고 별도의 질의 확장 과정 없이 바로 처리함으로써 질의 처리 시간을 줄이고자 한다.

그림 5에서 hasFormat 테이블은 학습 객체들 간의 hasFormat 관계에 대한 정보를 저장하고 hasVersion 테이블은 학습 객체들 간의 hasVersion 관계에 대한 정보를 저장한다.

학습 객체 저장소는 속성의 이행성과 학습 객체의 특성에 기반한 추론 질의가 가능하도록 그룹 관리자로부터 전달받은 질의를 2차적으로 확장하여 처리한다. 그림 5의 저장 구조를 이용하는 2차 질의 확장 규칙은 표 6과 같다.

학습 객체 저장소는 표 6의 질의 확장 규칙을 통해 새로운 질의문을 생성함으로써 질의 조건과 직접적으로 연관되어 있는 학습 객체는 물론 의미적으로 연관되어 있는 학습 객체도 검색하여 사용자의 만족도를 높인다.

표 6. 학습 객체 저장소의 2차 질의 확장 규칙
Table 6. 2nd Query Expansion Rules for Learning Object Storage

규칙 1	속성의 이행성에 기반한 질의 확장 1.1 질의 조건으로 제시된 속성에 맞게 테이블을 선택한다. 1.2 선택한 테이블에서 질의 조건으로 제시된 학습 객체와 관계를 맺고 있는 학습 객체를 검색한다. 1.3 검색된 학습 객체와 관계를 맺고 있는 또 다른 학습 객체를 검색하기 위해 새로운 질의 문장을 생성한다.
규칙 2	학습 객체의 특성에 기반한 질의 확장 2.1 keyword에 대한 질의는 Triple 테이블에서 조건에 맞는 속성 값을 가지고 있는 학습 객체를 검색한 후 검색된 학습 객체의 상위 학습 객체를 검색할 수 있도록 새로운 질의 문장을 생성한다. 2.2 taxonEntry에 대한 질의는 Triple 테이블에서 조건에 맞는 속성 값을 가지고 있는 학습 객체를 검색한 후 검색된 학습 객체의 하위 학습 객체를 검색할 수 있도록 새로운 질의 문장을 생성한다.

5. 질의 처리 기법

계층적으로 구조화된 이러닝 시스템에서 사용자의 요구에 부합하는 학습 객체를 검색하는 질의 처리의 세부적인 과정을 살펴본다.

하나의 온톨로지 관리자와 5개의 그룹 관리자가 존재하며 각 그룹별로 2개의 학습 객체 저장소가 존재하는 이러닝 시스템을 가정한다. 온톨로지 관리자는 그림 2의 저장 구조와 그림 3의 메타데이터 인덱스 구조를 유지하고 있고 그룹 관리자는 GML은 그림 4의 속성 인덱스 구조를 유지하고 있다고 가정한다. 그리고 GML이 관리하는 학습 객체 저장소 중 S1 학습 객체 저장소는 그림 5의 저장 구조를 유지하고 있다고 가정한다.

만약 학습자가 “ProgrammingLanguages”의 주제와 관련이 있는 학습 객체의 검색을 요구하면 다음과 같은 질의 처리 과정을 거쳐 최종 결과를 반환한다.

- 1단계 : 질의문을 트리플 형태로 변환한다.
 학습자가 제시한 질의문을 다음과 같이 표현한다.
 1:(? taxonEntry ProgrammingLanguages)

- 2단계 : 온톨로지 관리자는 1차적인 질의 확장을 수행하고 질의와 관련되어 있는 그룹 관리자로 질의를 전달한다.
 DC 형식을 사용하는 학습 객체 저장소도 질의 처리에 참여할 수 있도록 표 5의 질의 확장 규칙 1에 따라 다음과 같이 한 개의 질의문을 추가한다.

2:(? dcSubject ProgrammingLanguages)

그리고 학습 객체의 주제와 관련된 내용이기 때문에 질의 조건으로 제시된 주제의 하위 주제와 관련이 있는 학습 객체

도 검색할 수 있도록 표 5의 질의 확장 규칙 2에 따라 다음과 같이 네 개의 질의문을 추가한다.

- 3:(? taxonEntry LanguageClassifications)
- 4:(? taxonEntry ObjectOrientedLanguages)
- 5:(? dcSubject LanguageClassifications)
- 6:(? dcSubject ObjectOrientedLanguages)

다음으로 그림 3의 메타데이터 인덱스를 검색하여 KEM이나 LOM 형식을 이용하는 그룹 관리자에는 1, 3, 4번 질의를 전달하고 DC 형식을 이용하는 그룹 관리자에는 2, 5, 6번 질의를 전달한다.

- 3단계 : 질의를 전달받은 그룹 관리자는 질의와 관련된 학습 객체 저장소를 선택하여 질의를 전달한다.
 그룹 관리자 GM1은 그림 4에서 제시한 속성 인덱스를 이용해 질의와 관련된 taxonEntry 속성 정보를 유지하고 있는 학습 객체 저장소 S1과 S2로 질의를 전달한다.
- 4단계 : 질의를 전달받은 학습 객체 저장소는 2차적인 질의 확장을 수행한 후 각 질의를 처리하여 최종적으로 결과 학습 객체를 반환한다.

그룹 관리자 GM1로부터 질의를 전달받은 S1 학습 객체 저장소는 그림 5에서 제시한 Triple 테이블에서 4번 질의 처리의 결과로 C1 학습 객체를 검색한다. 학습 객체의 주제는 상위 학습 객체에서 하위 학습 객체로 상속되는 특징을 가지고 있으므로 표 6의 질의 확장 규칙 2에 따라 hasPart 테이블에서 C1 학습 객체의 하위 객체인 L1 학습 객체를 검색한다. hasPart는 이행성을 가지고 있으므로 표 6의 질의 확장 규칙 1에 따라 hasPart 테이블에서 L1 학습 객체의 하위 객체인 LNI, VL1 객체도 결과로 반환한다.

학습자가 요청한 질의 조건을 직접적으로 만족시키는 학습 객체는 존재하지 않는다. 그러나 본 논문에서 제안한 질의 처리 기법을 이용하면 추론 질의를 통해 의미적 연관성이 높은 C1, L1, LNI, VL1 학습 객체의 검색이 가능하기 때문에 사용자의 검색 만족도를 높일 수 있다.

V. 실험 평가

본 논문에서 제안한 계층적으로 구조화된 이러닝 시스템에서 학습 객체 온톨로지를 이용하는 질의 처리 기법은 질의 조건과 직접적으로 관련이 있는 학습 객체는 물론 추론에 기반한 질의 확장을 통해 의미적 연관성이 높은 학습 객체도 함께

검색하는데 목표를 두고 있다. 이러한 질의 처리 기법의 성능 개선 효과를 평가하기 위해 한 개의 온톨로지 관리자, 한 개의 그룹 관리자, 한 개의 학습 객체 저장소로 이러닝 시스템을 구성하고 학습 객체 온톨로지와 메타데이터 정보를 이용한 질의 확장을 적용한 경우와 적용하지 않은 경우를 비교 대상으로 선택하여 질의 처리 결과로 반환되는 학습 객체의 개수와 질의 처리에 소요되는 시간을 비교하였다.

질의 처리를 담당하는 학습 객체 저장소는 펜티엄4 3.4GHz의 CPU와 1GB 메모리를 가지고 Windows XP와 오라클 10g가 설치되어 있는 컴퓨터에 C언어로 구현하였다. 실험 데이터를 구성하기 위해 3장에서 설명한 MIT 대학의 OCW 웹 사이트를 통해 제공되는 학부와 대학원 강좌 중 "Electrical Engineering and Computer Science"와 관련된 247개 강좌에서 20개 강좌를 임의로 선택하였다. 그리고 각 강좌에서 제공하는 학습 객체들의 메타데이터 정보를 본 논문에서 제안한 학습 객체 온톨로지에 기반하여 OWL로 직접 기술해 실험 데이터로 활용하였다. 실험 데이터는 강좌의 수, 강좌에서 제공하는 학습 객체의 수, 트리플 구조로 변환했을 때 문장의 개수에 따라 표 7과 같이 세 가지로 구성된다.

표 7. 실험 데이터
Table 7. Experimental Data

	강좌 수	학습 객체 수	트리플 문장 개수
데이터1	4개	126개	1,380개
데이터2	10개	433개	4,760개
데이터3	20개	914개	10,050개

표 8은 실험에 사용한 세 가지 유형의 질의를 보여준다. 첫 번째 질의는 검색하고자 하는 학습 객체의 속성과 값이 주어진 일반적인 유형이다. 두 번째 질의는 이행성을 가지고 있는 hasPart 속성을 이용하는 유형이다. 세 번째 질의는 학습 객체의 주제가 질의 조건으로 제시된 유형이다.

표 8. 실험 질의
Table 8. Experimental Queries

	질의 내용
질의1	제목에 "introduction"이 포함된 학습 객체를 검색하라.
질의2	"Samuel Madden"이 제작한 학습 객체를 구성하는 하위 학습 객체들을 검색하라.
질의3	"programming languages" 주제와 관련된 학습 객체를 검색하라.

표 9는 학습 객체 온톨로지를 이용하는 질의 확장을 적용한 경우와 적용하지 않은 경우를 대상으로 검색된 학습 객체의 수와 검색에 소요된 초 단위의 처리 시간을 비교한 결과이다.

표 9. 질의별 검색된 학습 객체 수
Table 9. The number of learning objects

		질의 처리 기법 제외		질의 처리 기법 적용	
		검색된 학습 객체 수	질의 처리 시간	검색된 학습 객체 수	질의 처리 시간
질의1	데이터1	14	0.21ms	14	0.29ms
	데이터2	32	0.28ms	32	0.34ms
	데이터3	71	0.36ms	71	0.43ms
질의2	데이터1	24	0.54ms	63	38.89ms
	데이터2	50	1.37ms	140	53.14ms
	데이터3	73	2.29ms	189	64.33ms
질의3	데이터1	3	0.11ms	101	89.09ms
	데이터2	5	0.15ms	262	157.24ms
	데이터3	8	0.18ms	427	210.69ms

본 논문에서 제안한 질의 처리 기법을 적용하면 제시된 질의 조건과 직접적으로 관련이 없지만 의미적 연관성이 추론된 학습 객체들까지 검색된다. 따라서 속성의 이행성에 기반한 두 번째 질의와 학습 객체의 주제가 가지고 있는 특성을 고려하는 세 번째 질의는 질의 처리 기법을 적용한 경우 더 많은 학습 객체가 결과로 반환된 것을 표 9에서 확인할 수 있다. 그에 비해 첫 번째 질의는 이행성이나 역관계가 존재하지 않는 일반적인 속성이 질의 조건으로 주어졌기 때문에 검색된 학습 객체의 수는 같지만 질의 처리 기법을 적용할 경우 더 많은 처리 시간이 요구된다. 이것은 질의 처리 기법을 적용하는 경우 온톨로지 관리자에서 1차 질의 확장을 하는 시간과 학습 객체 저장소에서 2차 질의 확장 여부를 판단하는 시간이 추가적으로 필요하기 때문이다. 그러나 질의 처리 기법을 적용하면 다른 메타데이터 형식을 이용하는 학습 객체 저장소가 존재하는 경우에도 검색이 가능하기 때문에 결과적으로 더 많은 개수의 학습 객체가 검색될 수 있다. 따라서 질의 처리 기법을 적용하면 질의 처리 시간이 더 요구되지만 사용자의 불만을 초래할 만큼 긴 시간은 아니므로 결과로 반환되는 학습 객체의 수를 고려할 때 본 논문에서 제안한 질의 처리 기법을 적용하는 계층적으로 구조화된 이러닝 시스템은 활용 가치가 있다고 판단된다.

VI. 결론

이러닝 시스템에서 교수자와 학습자의 새로운 요구를 충족시키고 학습 객체에 대한 검색 및 관리를 지원하기 위해 시맨틱 웹의 핵심 기술인 메타데이터와 온톨로지를 적용하는 것은 필연적인 선택이다. 본 논문에서는 대표적인 학습 객체 메타데이터 형식을 통합하여 상호운용성을 제공하면서 대학 강좌에서 제공하는 학습 객체의 메타데이터를 기술할 때 사용하는 개념과 개념들 간의 다양한 의미적 관계를 정의한 학습 객체 온톨로지를 설계하였다. 그리고 여러 지역 저장소에 분산된 학습 객체들에 대해 통합적이고 효율적인 검색이 가능하도록 계층적으로 구조화된 이러닝 시스템을 구성하고 시스템 내에서 학습 객체를 위한 메타데이터와 온톨로지를 저장하는 기법과 추론에 기반한 질의 처리 기법을 제안하였다.

본 논문에서 제안한 계층적으로 구조화된 이러닝 시스템에서 학습 객체 온톨로지를 이용하는 질의 처리 기법을 적용하면 질의 조건과 직접적으로 관련이 있는 학습 객체는 물론 추론을 통해 의미적 연관성이 높은 학습 객체도 함께 검색되어 보다 정확하고 만족도 높은 검색 서비스를 제공할 수 있다.

본 논문에서는 같은 메타데이터 형식을 사용하는 학습 객체 저장소들로 그룹을 구성하였으나 향후에는 질의 처리 성능을 향상시키기 위해 학습 객체 저장소의 논리적 그룹을 구성하는 다양한 방법들을 고려하고 그것에 맞게 질의 처리 기법을 개선하고자 한다.

참고문헌

- [1] twkyung, and khkim, "The Trend and Development Directions of e-Learning Industry," Korea Contents Association, Vol. 7, No. 3, pp. 24-27, Sept. 2009.
- [2] hjchoe, and tykim, "Intelligent e-Learning Systems Using Ontology," Communications of the Korean Institute of Information Scientists and Engineers, Vol. 22, No. 8, pp. 41-49, Aug. 2004.
- [3] OWL Web Ontology Language Guide
<http://www.w3.org/TR/2004/REC-owl-guide-20040210/>
- [4] ykchoi, and ykchung, "A Study on Metadata Elements for Digital Course Resources in

Universities," Journal of Information Management, Vol. 39, No. 3, pp. 23-48, Sept. 2008.

[5] chshin, and jhpark, "A Study on Application for e-Learning Based on the Semantic Web Ontology," Proceedings of the Korean Institute of Maritime Information and Communication Sciences Conference, Vol. 13, No. 2, pp. 993-996, Oct. 2009.

[6] bochang, jhcha, and dhham, "The KEM OWL Binding," Journal of Digital Contents Society, Vol. 7, No. 2, pp. 125-131, June 2006.

[7] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch, "EDUTELLA: A P2P Networking Infrastructure Based on RDF," In Proc. of 11th International World Wide Web Conference, pp. 604-615, May 2002.

[8] MIT OCW, <http://ocw.mit.edu/index.htm>

[9] ACM Computing Classification System, <http://www.acm.org/about/class/1998>

[10] V. Christophides, D. Plexousakis, M. Scholl, and S. Tourounis, "On Labeling Schemes for the Semantic Web," In Proc. of 12th International World Wide Web Conference, pp. 544-555, May 2003.

저 자 소 개



김 연 희

2000 : 홍익대학교
컴퓨터공학과 공학사
2002 : 홍익대학교
컴퓨터공학과 공학석사
2006 : 홍익대학교
컴퓨터공학과 공학박사
현 재 : 부천대학
e-비즈니스과 강의전담교수
관심분야 : 시맨틱 웹, 이리닝 시스템
Email : yhkim@bc.ac.kr



김 지 현

1978 : 이화여자대학교
수학과 이학사
1994 : 단국대학교
전자정보전공 경영학석사
2004 : 단국대학교
전산통계학과 이학박사
1997 : 정보관리 기술사
현 재 : 서일대학
컴퓨터소프트웨어과 부교수
관심분야 : 웹공학, 데이터베이스,
품질 관리
Email : jhkim@seoil.ac.kr