시맨틱 웹에서 의미 검색을 위한 RDF 메타데이타 자동 생성

강상구*: 양재영 · 양승섭 · 최원종 · 최중민

Automatic Generation of RDF Metadata for Semantic Search in Semantic Web

Sanggu Kang*: Jaeyoung Yang · Seungsub Yang · Wonjong Choi · Joongmin Choi

요약

시맨틱 웹은 인간이 이해하는 것처럼 웹 문서의 의미를 컴퓨터가 처리할 수 있도록 하는데 있다. 그러나 인터넷 등 정보통신 기술의 발전으로 인해 정보량이 급증함으로써 이들 정보 자원을 효과적으로 검색하기에는 많은 어려움이 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 주석 에디터를 사용하여 논문에 대한 RDF 메타데이타의 자동 생성 방법을 제안한다. 사용자가 논문을 주석 처리할 때, 문서에 대한 특징을 추출하고 온토로지 인터페이스를 사용하여 문서를 분류 한다. 구현된 시스템을 통해 사용자는 추출된 메타데이타를 메타데이타 뷰를 통해 볼 수 있으며, HTML 뷰를 통해 메타데이타를 수동으로 수정이 가능하다. 이 메타데이타는 RDF Repository로 저장할 수 있으며, 주석 뷰를 통하여 RDF 메타데이타 생성을 확인할 수 있다. 이렇게 생성된 RDF 메타데이타는 웹 로봇이 내용의 의미 파악 및 카테고리 정보를 쉽게 알 수 있도록 해준다. 본 논문은 검색 엔진을 통하여 논문 검색 시 전체 내용보다 RDF 메타데이타 정보만으로 효율적인 검색을 할 수 있는 방법에 초점을 둔다.

Keywords : Semantic Web, Metadata, RDF, Ontology, Annotation, Classification

1. 서론

최근 들어 인터넷의 발전으로 엄청나게 늘어나고 있는 정보의 양은 사용자들에게 많은 지식과 다양한 서비스를 제공하고 있는 반면에 정보 과다(information overload)라는 새로운 문제점을 야기시켰다. 검색엔진이 개발되어 이러한 문제점을 해결하려고 시도하고 있지만 대부분의 검색엔진이 웹 문서의 내용보다는 단어나 구문 등 단편적인 방법으로 관련성을 검사하므로 사용자가 의미적으로 원하는 문서의 검색이 어려운 실정이다. 정보 자원(resource)을 효과적으로 검색하기 위한 방법으로 메타데이타(metadata)를 추가하여 효율적으로 웹 문서의 내용검색을 위한 모델에 대한 연구가 시도되고 있다. 이러한 검색 모델을 통해서 사용자는 필요한 정보 자원에 보다 쉽고 정확하게 접근할 수 있다. 이런 메타데이타는 정보 자원을 효율적으로 검색하고, 정보 자원에 태그를 부착함으로써 정보 자원을 재 포장하는 가치를 부여한다. 메타데이타는 텍스트 자료에 대한 정확한 검색뿐만 아니라 비텍스트적인 자료를 검색하고자 하는 경우에도 유용하며, 방대한 양의 각종 정보 자원에 대한 정보 관리 및 기록 관리 등의 다양한 목적을 위해 사용된다.

이런 관점에서 새롭게 제시된 RDF(Resource Description Framework)는 각기 다른 기술 구조를 가진 다양한 메타데이타를 상호운용성(interoperability)의 입장에서 통합하기 위한 메타데이타 구조이다. 그리고 상이한 메타데이타로 기술된 웹 자원에 효과적으로 접근하기 위한 더블린 코어(Dublin Core)나 워릭(Warwick) 구조의 개념을 구체적으로 실현하는 수단이다. 이 구조는 메타데이타를 교환하기 위한 하부구조로서, 데이타의 의미와 구문, 구조의 통일을 통해 메타데이타의 상호운용성을 확보하기 위한 것이며 기술언어로 XML(Extensible Markup Language)을 사용한다. 다양한 메타데이타를 인정하고 이들을 하나의 통합된 틀 안에서 운용하기 위한 시도로 메타데이타의 구조를 특정 형식으로 제한하지 않고 다양성을 인정한다. 그 배경은 기본적으로 하나의 메타데이타 형식이 독점적으로 사용될 수 없고, 또한 특정 형식이 무한정 확장되는 것도 불가능하다는 관점이다. 즉, 이용자의 수준과 응용분야마다 요구되는 데이타 요소와 수준이 다르기 때문에 어떤 단일 형식의 메타데이타도 모든 조건을 만족 시킬 수 없으므로 메타데이타의 다양성을 인정하고 이를 수용할 수 있는 포괄적인 구조가 RDF이다.

본 논문에서는 웹 문서에서 특징을 추출하고 이것

* 한양대학교 컴퓨터공학과

을 기반으로 웹 문서에 대한 RDF 메타데이타를 자동 생성할 수 있는 방법을 제안한다. 또한 주석(annotation) 에디터를 통해 잘못 분류된 문서를 수정하여 보다 정확하게 웹 문서에 메타데이타를 추가하는 방법을 소개한다. 여기에서 웹 문서의 정보를 더 정확하게 주기 위해 온토로지(ontology)를 기반으로 문서를 분류(classification) 한다. RDF 메타데이타에 타케고리(category) 정보를 추가하여 분류된 논문이 어느 분야에 속하는 지를 정확하게 알 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 설명하고 3장에서는 시맨틱 웹(semantic web)에 대한 개념과 핵심 기술인 RDF, 그리고 온토로지에 대해 설명한다. 4장에서는 주석 시스템의 전체적인 구성과 각 모듈에 대한 기능을 설명한다. 마지막으로 5장에서는 연구 내용을 요약하고 앞으로의 향후 과제와 함께 결론을 내린다.

2. 관련연구

2.1 **Annotea**

Annotea[1]는 공유된 주석들과 함께 W3C(World Wide Web Consortium) 협력 환경을 증진시키는 LEAD(Live Early Adoption and Demonstration) 프로젝트이다. 이 프로젝트에는 두 가지 중요한 목적이 있는데, 첫 번째는 공유된 웹 문서들을 유지하는 환경의 툴을 개발하기 위해 W3C와 협력을 유지하는 것이고, 두 번째는 주석 처리된 문서 내에서 주석의 위치를 알기 위한 XPointer, 메타데이타와 같은 주석을 표현하기 위한 RDF 기반의 주석 스키마(schema), 그리고 XLink와 HTTP등 존재하는 W3C 기술들을 재사용하는 것이다.

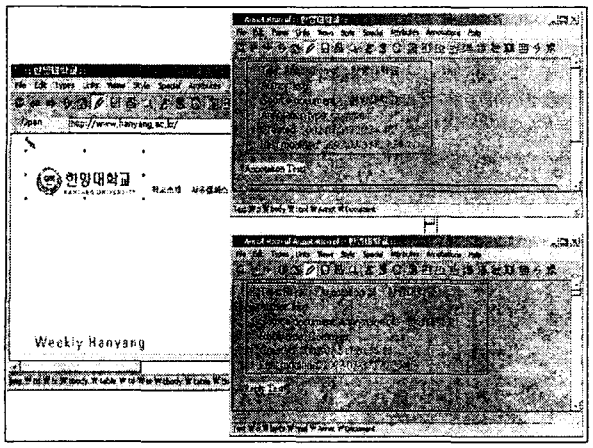
Annotea는 주석 서버와 클라이언트에서 실행하는 W3C의 Amaya editor/browser로 구성된다. Amaya는 로컬과 원격의 주석처리가 가능하며, [그림 1]과 같이 Amaya를 통하여 한양대학교 홈페이지내에서 한양대학교 마크에 대한 주석처리를 볼 수 있으며, 그리고 주석처리 된 문서에 대하여 다시 주석 처리한 것을 볼 수 있다.

Amaya의 기본적인 주석 클래스는 Annotation, Advice, Change, Comment, Example, Explanation, Question , SeeAlso로 구성 된다.

그리고 기본적인 주석 속성은 다음과 같다.

- rdf:type : 주석 처리할 때 제작자의 의도를 표시하며, 이 값은 Annotation 슈퍼클래스나 하위클래스가 사용됨.
- annotates : 주석을 적용하는 자원.
- body : 주석의 내용.
- context : 주석을 적용하는 자원에서 직접적으로 가리키는 문맥.
- author : 주석을 만드는 개인이나 조직 이름.
- created : 주석을 만든 날짜와 시간.
- modified : 주석을 수정한 날짜와 시간.
- related : 주석과 추가적인 자원들 사이의 연관성.

[그림 1] Amaya를 통한 주석 처리



2.2 **OntoMat-Annotizer**

웹 문서의 주석처리는 웹상에서 메타데이타를 만들기 위한 중요한 기술 중 하나이다. 그렇지만 지금까지의 주석 툴들은 풍부하게 연결된 그리고 정확히 기계가 처리할 수 있는 정보를 제공하기 위한 가능성이 제한되었다. 주석 툴들은 사용자가 더블린코어와 같은 템플릿 구조에 따라 텍스트에 주석을 다는 것을 제공한다.

여기에서 CREAM(Creating Relational, Annotation-based Metadata)[2]은 관계 메타데이타(Relational Metadata) 구조를 제공하는 주석 환경을 위한 프레임웍이다. 메타데이타는 클래스 인스턴스와 속성 인스턴스 그리고 관계 인스턴스로 구성된다. 이 인스턴스들은 더블린 코어 같은 고정된 구조 보다는 도메인 온토로지에 기반한다. 이 프레임웍을 실행한 것이 Ont-O-Mat Annotizer인데 이것은 컴포넌트 기반이며, 온토로지를 통한 주석 툴이다.

Ont-O-Mat Annotizer는 사용하기 쉬운 상호 교환적인 웹 페이지 주석 도구이며, 속성과 관계를 만들기 위해 온토로지를 기반한 DAML+OIL 마크업을 만들고 유지하는 작업과 함께 사용자를 지원한다. Ont-O-Mat Annotizer은 온토로지의 사용을 위해 온토로지 브라우져 그리고 인스턴스와 텍스트의 주석된 부분을 나타낼 수 있는 HTML 브라우져를 포함하고 있다.

Ont-O-Mat으로 메타데이타를 생성하는 방법에는 툴과 함께 상호작용 하면서 세 가지 유형에 의해 만들어 진다.

- 타이핑에 의해 주석을 만드는 것인데 대부분 ontology guidance/fact browser와 생성된 템플릿을 이용한다.
- 마크업에 의해 주석을 만드는 것인데 인스턴스 생성을 위해 document editor/viewer내에서 내용을 drag-and-drop을 사용한다.
- 저자(author)에 의해 웹 페이지와 메타데이타를 만드는 것인데 ontology guidance/fact browser에 있는 인스턴스 값을 drag-and-drop 해서 웹 문서 및 메타데이타를 생성

한다.

2.3 Shoe Knowledge Annotator

SHOE(Simple HTML Ontology Extensions)[3, 4]는 온토로지에 기반한 지식 표현 언어이며, 의미 정보를 웹 페이지에 삽입할 수 있도록 필요한 태그들을 추가한 HTML의 superset이다. SHOE 태그들은 두개의 카테고리로 나눌 수 있다. 첫 번째는 온토로지 생성을 위한 태그이다. SHOE에서의 온토로지는 객체를 정의하고 그것의 의미를 파악할 수 있는 규칙의 집합으로 정의할 수 있다. 두 번째는 웹 문서에 주석을 달기위해 사용되는 태그이다. 태그의 주된 역할은 웹 문서에 적합한 온토로지의 설정 및 데이타 선언 그리고 온토로지를 기반으로한 명제의 표현에 있다.

SHOE는 XML의 상호 운용성 문제를 해결하고 있다. XML은 문서 자체의 의미보다는 문서가 가지고 있는 정보를 표현하며 사용자의 임의대로 확장이 가능하다. 이런 XML의 확장성은 정보 처리 상호 운용성 문제를 가져오게 되는데, 예를 들어 만일 대학과 가구 가게가 태그 <Chair>를 양쪽 다 사용하면 XML에서는 같으나 다른 물건을 의미할 수도 있다. 또 다른 가구 가게가 태그 <Seat>를 사용하는 경우에 문제가 된다. XML DTD를 하나의 DTD로만 이용이 가능하다면 문서에 모든 것이 같은 태그를 사용하므로 정보 처리 상호 운용성을 해결할 수 있다. 그러나 모든 것을 기술하는 하나의 DTD를 만드는 것이 불가능 하므로 XML은 비즈니스간 자료 교환 언어와 전자상거래에서는 매우 유용할 수 있지만, 검색에 대해서는 불충분 하다. 이런 문제점을 해결하기 위해 SHOE에서는 온토로지를 기반한 지식 표현 언어를 사용한다.

SHOE는 Knowledge Annotator, Expose, Knowledge Base, SHOE Search로 구성된다. Knowledge Annotator는 자바 프로그램으로 되어 있으며 웹 문서에 주석을 달 수 있도록 만들어 졌다. Knowledge Annotator에 의해 주석 처리된 문서들은 웹 로봇인 Expose에 의해 수집되며 이렇게 수집된 문서들은 Knowledge Base에 저장된다. 그리고 SHOE Search를 통하여 검색을 하면 Knowledge Base에서 의미적으로 알맞은 문서를 가져온다.

3. 시맨틱 웹

3.1 시맨틱 웹 정의

W3C에서 시맨틱 웹[5, 6, 7]을 위한 표준 개발과 프로토타입 제공, 워크샵과 컨퍼런스를 통한 보급과 확대, 그리고 W3C의 XML과 관련 활동을 통합하는 작업등 다양한 방법을 통해 시맨틱 웹을 구체화해 나가고 있다. 시맨틱 웹은 광범위한 범위에서 기계들에 의해 의미를 쉽게 처리할 수 있는 서로 연결된 정보의 그물 망이다. 그리고 광범위하게 연결된 데이타베이스나 월드 와이드 웹 상에서 데이타를 표현하는 효율적인 방법이다. 즉, 인간이 이해하는 것처럼 웹에 존재하는 데이타들의 의미를 컴퓨터가 처리할 수 있음을 의미한다. 이러한 웹은 WWW, URI(Uniform Resource Identifier), HTTP, 그리고 HTML의 창안자인 팀 버너스리(Tim Berners-Lee)에 의해 고안 되었다.

현재의 시맨틱 웹은 독립적인 서비스보다도 더 큰 기능성과 상호 운용성을 제공하는 정보 매개자(information broker), 검색 에이전트(search agent), 그리고 정보 필터(information filter)와 같은 지능적인 서비스들을 개발하는데 초점을 두고 있다.

3.2 메타데이타

웹상의 자원은 인간이 사용하기 위해 작성된 것이며 웹에 있는 모든 것을 읽을 수 있긴 하지만 이 데이타를 기계가 이해할 수는 없다. 웹에서 모든 것을 자동화하는 것은 매우 어려우며 웹이 포함하고 있는 정보량이 많기 때문에 수작업으로 이것을 관리하는 것은 또한 불가능하다. 시맨틱 웹에서 제시하는 해결책은 웹에 수록된 정보의 기술을 위해 메타데이타를 사용하는 것이다. 메타데이타는 일반적으로 "데이타를 위한 데이타"라고 정의 한다.

<표 1> 더블린 코어 요소 집합

	요소	정의	설명
1	Title	제목	자원에 주어진 이름
2	Creator	제작자	자원의 내용에 주된 책임을 가진 개인이나 단체
3	Subject	주제, 키워드	자원의 내용에 대한 주제
4	Description	설명	자원의 내용에 대한 설명
5	Publisher	발행자	자원을 이용 가능하게 만든 개인이나 단체
6	Contributor	기타제작자	자원의 생성에 기여한 개인이나 단체
7	Date	날짜	자원을 생성한 날짜
8	Type	자료유형	자원의 범주 또는 유형
9	Format	표현형식	자원의 물리적 표현형식
10	Identifier	식별자	자원을 명백하게 식별하기 위한 고유한 문자나 숫자
11	Source	출처	자원의 출처가 된 원 자원에 대한 정보
12	Language	언어	자원의 내용을 기술하고 있는 언어
13	Relation	관계	자원에 관련된 다른 정보 자원과의 관계
14	Coverage	내용범위	자원의 시간적, 지리적 특성을 나타내는 정보
15	Rights	권한	자원이 가진 권리에 관한 정보

메타데이타의 중요한 특성은 간결성인데 이를 위해 현재 메타데이타의 표준으로 더블린코어(Dublin Core)가 제시되고 있다. 더블린 코어의 첫 워크샵은 1995년 미국의 더블린에서 여러 관계 학자들이 모여 네트웍 환경에서 이용되는 자원을 기술하기 위한 핵심 기술 요소를 제안했으며, 그리고 현재까지 계속해서 메타데이타의 표준 제정을 위해 수정 및 보완 작업을 해나가고 있다. 위의 <표 1>에서 알 수 있듯이, 더블린 코어의 메타데이타 요소 집합은 15개의 기술 요소에 대한 의미론적 정의들로 이루어져 있다

3.3 RDF

인터넷 특히 웹의 등장으로 엄청난 양의 자원이 생산 이용되고 있기 때문에 이러한 정보 바다에서 필요한 정보만을 선택할 수 있기 위해 자료에 대해 기술한 메타데이타의 역할은 매우 중요 하다. RDF[8 9]는 메타데이타의 기술과 교환을 위한 구조로 웹 상의 메타데이타를 지원하는데 필요한 구조를 정의

하기 위해 W3C에서 제안한 표준이다. 그리고 RDF는 인터넷 상에 존재하는 상이한 성격의 메타데이타 간의 상호 운용이 가능하도록 하는데 그 목적이 있다.

RDF의 가장 큰 유용성은 다음과 같다.

- 메타데이타간의 상호 운용성이 가능
- RDF를 기술하는 구문인 XML 자체가 확장성을 가지고 있으므로 RDF로 자원을 기술할 경우 내용을 자유롭게 확장 가능
- 문서의 전체 검색보다는 메타데이타 정보를 검색할 경우 기계가 의미적으로 원하는 정보를 쉽게 검색 가능

3.3.1 RDF 데이타 모델

RDF의 기초는 자원에 대해 지정된 속성과 그 값을 표현하기 위한 모델이다.

기본적인 데이타 모델은 아래와 같이 세 개의 객체 타입으로 구성된다.

- 자원 (Resources)
  RDF 형식으로 기술되는 모든 것을 자원이라고 한다. 자원은 웹 페이지 전체일 수도 있고, 문서 자원에 포함된 특정한 HTML 이나 XML 요소와 같이 웹 페이지 일부일 수도 있다. 따라서 자원은 URI로 식별 가능한 모든 객체를 의미한다.
- 속성 (Properties)
  속성은 자원을 기술하기 위해 사용된 특정한 관점, 특징, 속성, 관계이다. 각 속성은 특정한 의미를 가지며, 허용되는 값, 특성이 기술하는 자원의 유형, 다른 속성과의 관계를 정의한다.
- 문 (Statements)
  특정 자원과 지정된 속성, 그리고 그 속성의 값을 RDF 문이라고 한다. RDF 문은 세 부분으로 구성되는데 주어(subject)와 술어(predicate), 그리고 목적어(object)이다. RDF 문의 목적어는 다른 자원일 수도 있고, 리터럴(literal)일 수 있다.

간단한 예로서 다음 문장을 생각해보자:

"강상구는 http://islab.hanyang.ac.kr/~sgkang/ 자원의 작성자이다."

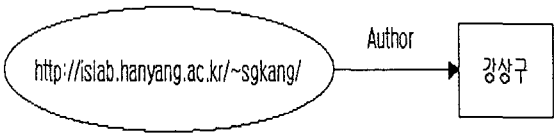
이 문장은 다음과 같이 RDF 문으로 변환할 수 있다.

주어 (자원)	http://islab.hanyang.ac.kr/~sgkang/
술어 (속성)	Author
목적어 (속성값)	"강상구"

방향그래프와 레이블을 사용하여 RDF 문을 다이어그램으로 나타낼 수 있다. 다이어그램에서 노드는 자원을 표현하는 것이고, 아크는 속성을 표현한 것이며 문자열 리터럴을 표현한 노드는 직사각형으로 표현되는 데 이것은 속성값이 된다.

위 문장을 간단한 다이어그램으로 나타내면 다음과 같다.

[그림 2] 노드-아크 다이어그램



3.3.2 RDF 구문

RDF 데이타 모델은 메타데이타를 사용하고 정의하기 위한 추상적이고 개념적인 구조를 제공하고 있는데, 이를 표현하는 구문 구조는 RDF를 통해 표현된 내용들을 기계가 읽을 수 있는 형태로 변환할 수 있어야 한다. RDF에서 사용하고 있는 표현 구문은 XML이며, RDF와 XML은 상호 보완적이다. RDF 구문[10]은 두 개의 XML구문으로 정의하고 있다.

첫 번째 연속 구문(serialization syntax)은 RDF의 표준구문으로서 일반적인 데이타 모형을 완전하게 기술하기 위한 구문이다.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ts="http://islab.hanyang.ac.kr/~sgkang/thesis-rdf-schema#">
  <rdf:Description rdf:about="http://islab.hanyang.ac.kr/~sgkang/">
    <ts:Author>강상구</ts:Author>
  </rdf:Description>
</rdf:RDF>
```

두 번째 단축형 구문(abbreviation syntax)은 데이타 모형을 간단한 형태로 표현하는 구문이다.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ts="http://islab.hanyang.ac.kr/~sgkang/thesis-rdf-schema#">
  <rdf:Description rdf:about="http://islab.hanyang.ac.kr/~sgkang/"
      ts:Author = "강상구" />
</rdf:RDF>
```

RDF 데이타 모델과 파서에 의한 해석은 두 구문이 동일하나 두 구문을 HTML 문서 헤드 태그에 삽입 시키면 두 개의 구문이 다르다는 것을 알 수 있다. 연속 구문은 속성값이 브라우져에 보여지고 그 반면에 축약 구문은 속성값이 보이지 않는다. 그래

서 RDF 구문을 웹 문서에 삽입할 때는 축약구문을 사용한다.

RDF 데이타 모델은 복수의 값을 가질 수가 있는데 Bag, Sequence, Alternative 이 세가지 유형의 컨테이너를 사용하여 RDF로 표현할 수 있다.

- Bag은 특성이 복수의 값을 가지고 있고, 그 값을 제시하는 순서는 중요하지 않을 때 사용한다. 값을 중복할 수 있다.
- Sequence는 특성이 복수의 값을 가지고 있고, 그 값의 순서가 매우 중요할 때 사용한다. 값을 중복할 수 있다.
- Alternative는 특성의 단일 값에 대한 또 다른 대안의 값을 표현하는 리터럴 리스트이다. 그리고 리스트에서 어떤 하나의 값을 선택할 수 있다.

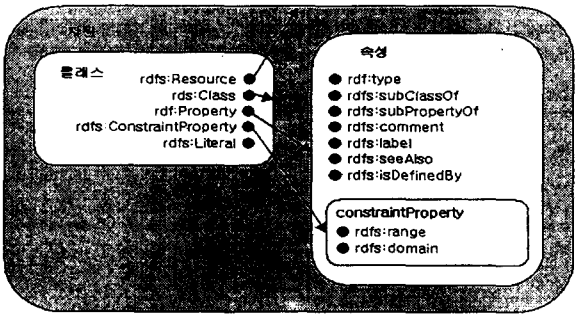
3.3.3 RDF 스키마

RDF 스키마[11, 12]는 자원들 사이의 속성 및 관계성을 표현하는 계층에 대한 정보의 집합으로, RDF 자원의 클래스에 대한 속성을 표현한다. RDF 스키마를 식별하기 위해 XML 이름 공간을 사용하며, URI를 이용하여 RDF 스키마를 인간과 기계가 동시에 처리할 수 있도록 한다.

RDF 스키마는 객체지향 프로그램 언어인 자바와 비슷하게 클래스와 속성의 타입을 사용하지만, 자바와는 다르게 속성 중심으로 접근함에 따라 새로운 속성을 추가하기가 쉽다.

RDF 스키마에 대한 클래스와 하위 클래스, 자원의 개념을 [그림 3]에서 표현하고 있다. 클래스는 둥근 사각형으로, 자원은 큰 점으로 표시되며, 화살표는 자원으로부터 화살표가 정의하는 클래스를 나타낸다. 그리고 하위클래스는 상위클래스에 포함 되어 있다.

[그림 3] 자원과 클래스 그리고 속성 집합

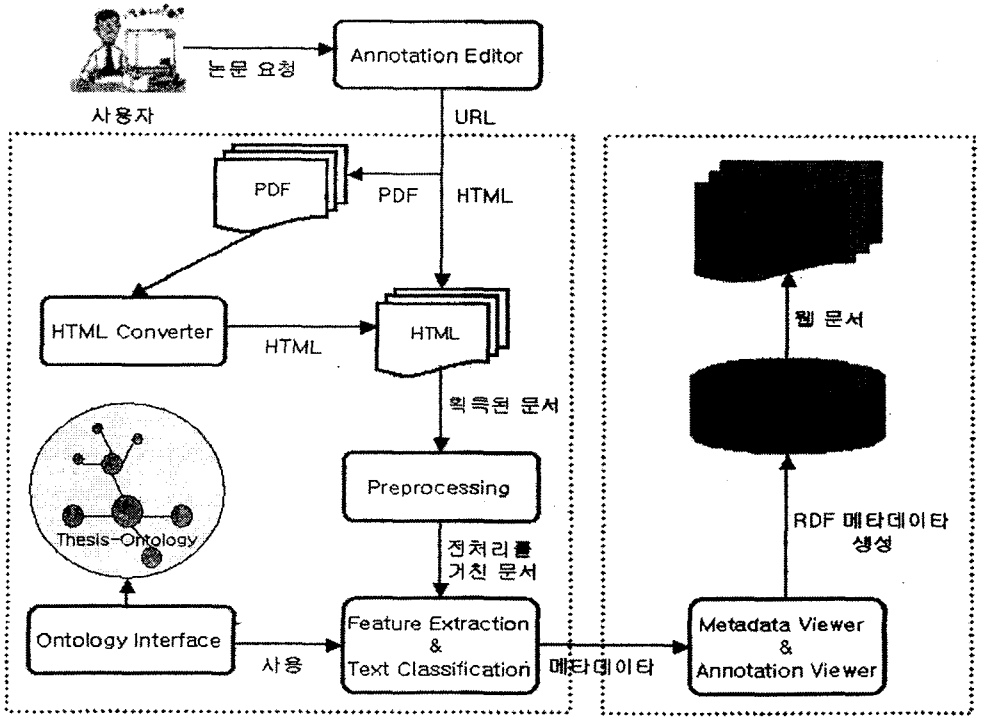


3.4 온토로지

온토로지[13, 14, 15, 16]는 과거에는 철학분야에만 국한되어 사용 되었으나 근래에는 컴퓨터공학 분야에 적용되어 널리 사용되고 있다. 특히 최근에는 지식공학, 지식표현, 데이타베이스 디자인, 정보 모델링, 정보 통합/관리/조직, 에이전트 기반 시스템 등 다양한 분야에 적용되고 있다.

인공지능(AI) 학문에 있어서 온토로지는 "개념화의 명세(specification of a conceptualization)"로 정의 된다. 이는 "engineering artifact"로써 어떤 사실을 기술하기 위해 필요한 객체(object)의 집합인 "vocabulary(universe of discourse)"와 이의 객체들간의 관계인 relation과 function들의 집합으로 이루어진다. 다른 말로 표현하면 온토로지는 객체의 집합과 객체들간의 관계의 정의에 어떤 사실이나 상태를 표현하

[그림 4] 전체 시스템 구조



고자 하는 지식 표현 기법이다. 시맨틱 웹이 발전하기 위해서는 RDF와 같은 기술로 모델링 되어야 하는데, 온토로지를 모델링하기에는 RDF 자체도 의미적인 면에서 한계성이 나타나고 있다. 그래서 RDF를 기반으로 확장된 언어들이 개발되고 있다.

온토로지 언어에는 대표적으로 OIL(Ontology Inference Layer), DAML(DARPA Agent Markup Language) 그리고 요즘 들어 DAML+OIL을 W3C에서 표준으로 규정하고 있다.

4. 시스템 구조

본 논문은 웹 상에 있는 논문에 메타데이타 정보를 추가하기 위한 시스템이다. 인공지능과 관련있는 논문을 대상으로 하고 있으며 본 논문의 시스템인 주석 에디터는 온토로지 인터페이스, 메타데이타 뷰, HTML 뷰, 주석 뷰로 구성된다.

- 온토로지 인터페이스는 문서를 분류하기 위해 사용된다.
- 메타데이타 뷰는 추출된 메타데이타를 수정 및 볼 수 있다.
- HTML 뷰는 문서의 내용을 볼 수 있다.
- 주석 뷰는 생성된 RDF 메타데이타를 볼 수 있다.

4.1 주석구조

본 논문에서 제안하는 전체 시스템 구조는 [그림 4]와 같다. 시스템에서 사용자가 논문을 요청하면 파일 형식이 PDF인 경우는 HTML로 변환하고 전처리 과정을 거친 문서에서 특징(feature)을 추출한다. 그런데 HTML로 변환하면서 태그의 정보들이 모두 사라지는 문제점이 발생하므로 실제적으로 태그의 정보를 이용해서 특징을 추출하는 방법은 어려움이 있다. 그래서 논문의 특징 추출은 세 부분으로 나누어서 하고 있다.

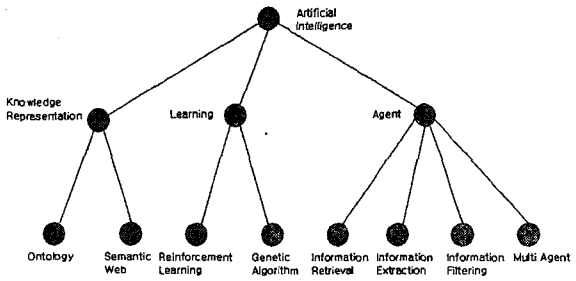
- 첫 번째는 논문의 시작부터 소개(introduction) 전까지 추출해서 그 내에서 제목, 저자, 메일, 학교, 요약, 키워드 정보를 추출한다. 제목은 위에서 몇 번째 줄에 나오는 것이 제목이다라고 지정을 하여 추출하고, 메일은 "@"를 기준으로 하여 추출한다. 학교는 "university" 앞이나 "university of" 다음에 학교가 나온다. 그리고 요약은 "abstract"에서 "introduction"까지 추출한다. 키워드는 논문에 "keywords" 단어를 기준으로 하여 추출하며 어떤 논문에서는 키워드가 없는 것이 있는데 이런 경우에는 논문의 시작부터 관련연구나 결론 전까지 부분을 추출해서 이내에서 단어가 자주 발생하는 것을 키워드로 세 개 추출한다. 저자는 추출이 어려워서 수동으로 추출하고 있다.
- 두 번째는 참고문헌(references)에서 논문의 끝까지 추출해서 그 내에서 참고문헌 부분을 추출하는데, 참고문헌 추출은 할 수 있으나 각 참고문헌의 제목을 뽑아내는 것이 어렵다. 본 논문에서는 세 개의 참고문헌만을 추출하는 것으로 하며, 수동으로 제목을 찾는 것으로 하고 있다.
- 마지막으로 문서를 분류하기 위한 특징은 논문의 시작부터 관련연구나 결론 전까지 부분을 추출한다. 그 이유는 관련성이 없는 부분은 제외하고 관련성이 있는 부분만 추출하여 그 내에서 단어가 많이 발생하는 특징을 사용하고 있다. 논문의 특성상 TF·IDF(Term Frequency/Inverse Document Frequency)보다는 TF를 사용하는 것이 문서의 분류를 더 정확하게 한다.

본 논문에서는 온토로지를 "특정 주제에 대한 간단한 규칙들이나 의미적 연관관계와 단어들을 포함한 지식 용어들의 집합"으로 정의하기로 한다. 본 논문은 웹 문서의 분류를 용이하기 위해서 온토로지를 개념(concept)과 특징, 관계(relation) 그리고 제약조건(constraint)으로 구성된 노드(node)들로 표현한다.

- 특징은 개념을 표현할 수 있는 단어나 구의 집합으로 이루어진다.
- 관계는 노드(개념)간의 관계를 표현하며 "isA", "partOf", "hasPart"로 정의된다. "isA"는 개념간의 일반화(generalization)의 의미로 모든 링크는 "isA"관계를 가지며 "partOf"와 "hasPart"는 서로 상반되는 의미로써 노드간의 포함관계를 나타낸다.
- 제약조건은 "isRelatedTo", "follwedBy"로 표현되며 밀접한 관계를 가지는 특징들이나 요소들을 묶어 놓음으로써 단어기반의 분류가 가지는 모호성을 해결하고 분류의 정확성을 기하기 위해 중요한 의미를 지닌다.

[그림 5]와 같이 인공지능 관련 논문에 대해 XML 기반으로 온토로지를 수동 구축 하였으며, 구축된 온토로지는 약하게 구조화된 온토로지(weakly structured ontology)라 할 수 있다. 약하게 구조화된 온토로지에서는 개념 값(class value)이나 클래스 인스턴스(class instance), superclass-subclass, part-whole 등의 개념적인 관계가 명확히 구분되지 않는 특성을 갖고 있다.

[그림 5] 인공지능 관련에 대한 논문 온토로지



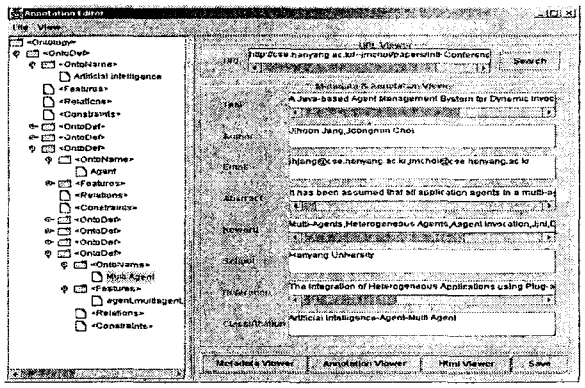
온토로지 노드의 표현은 [그림 6]과 같다. 이 온토로지는 "http://islab.hanyang.ac.kr/~sgkang/ontology.xml" URI에 인공지능 관련 논문에 대한 온토로지를 표현하고 있다.

[그림 6] 인공지능 관련 논문 노드의 표현

```
- <Ontology>
- <OntoDef>
  <OntoName>Agent</OntoName>
  <Features>agent,information,retrieval,extraction,filtering,knowledge,
            ie,ir,intelligent,collaborative
  </Features>
  <Relations />
  <Constraints />
+ <OntoDef>
  <OntoName>Information Retrieval</OntoName>
  <Features>ir,information,retrieval,query,document,relevant,searching,
            knowledge,trec,system
  </Features>
  <Relations />
  <Constraints />
  </OntoDef>
+ <OntoDef>
  <OntoName>Information Extraction</OntoName>
  <Features>extraction,wrapper,structure,ie,pattern,sentence,learning,
            text,natural,nlp
  </Features>
  <Relations />
  <Constraints />
  </OntoDef>
- <OntoDef>
  <OntoName>Multi Agent</OntoName>
  <Features>agent,multiagent,collaborative,learning,system,mas,intelligent,
            framework,software,design
  </Features>
  <Relations />
  <Constraints />
  </OntoDef>
  </OntoDef>
  </Ontology>
```

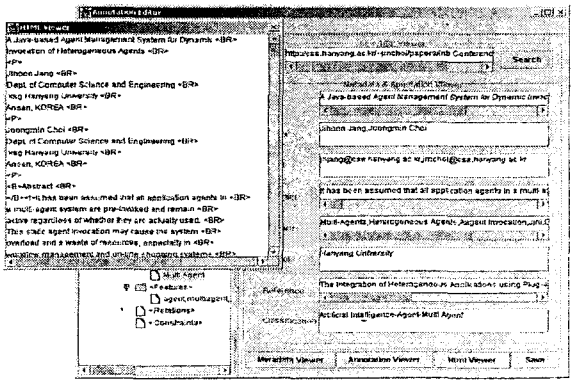
사용자의 요청에 의해 추출된 특징은 온토로지 인터페이스를 사용해 문서를 분류하고 생성된 메타데이타는 [그림 7과]같이 주석 에디터에 있는 메타데이타 뷰를 통해 상태를 확인할 수 있다. 여기에서 저자와 저자사이에는 콤마로 구분하며 메일이나 키워드 그리고 참고문헌도 콤마로 구분한다. 문서 분류는 "-"로 구분하고 있다.

[그림 7] 주석 에디터



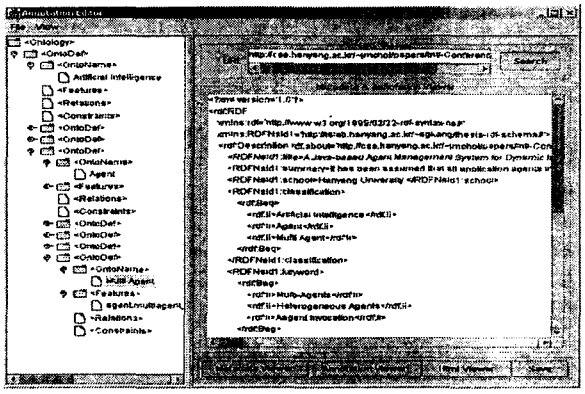
본 논문에서는 논문에 대한 정보가 없거나 아니면 추출을 하지 못할 경우에는 메타데이타가 생성 되지 않으므로, [그림 8]과 같이 HTML 뷰를 통하여 메타데이타를 정확하게 수정 및 추가할 수 있다.

[그림 8] HTML 뷰



이렇게 생성된 메타데이타가 정확하게 추출되고 분류 되었으면 저장한다. 그러면 RDF 메타데이타가 생성되고 [그림 9]와 같이 주석 뷰로 통하여 RDF 메타데이타를 확인할 수 있다.

[그림 9] 주석 뷰



[그림 10]은 주석 에디터를 통하여 논문에 대해 RDF 메타데이타를 생성한 인스턴스이다. 이렇게 논문에 주석을 부여 함으로써 웹 로봇이 문서 전체 내용보다는 주석 처리된 부분만을 사용하여 더 정확한 의미를 얻을 수 있다.

Appendix A는 [그림 10]의 데이타 모델을 RDF 구문으로 표현한 것이다. 본 논문에서는 RDF 메타데이타와 논문의 URI를 RDF Repository에 저장한다. 현재의 웹 문서는 저자만이 수정할 수 있어서 다른 사용자가 문서에 대해 주석 처리하더라도 여러 사용자들이 같이 공유할 수가 없다. 이러한 문제점을 해결하고자 본 논문에서는 앞에서 온토로지에 의해 분류된 것을 기반해서 만든 웹 문서에 논문의 URI와 RDF 메타데이타를 같이 삽입한다. 삽입된 RDF 메타데이타를 통해서 내용에 대한 의미 파악 및 카테고리 정보를 알 수 있다.

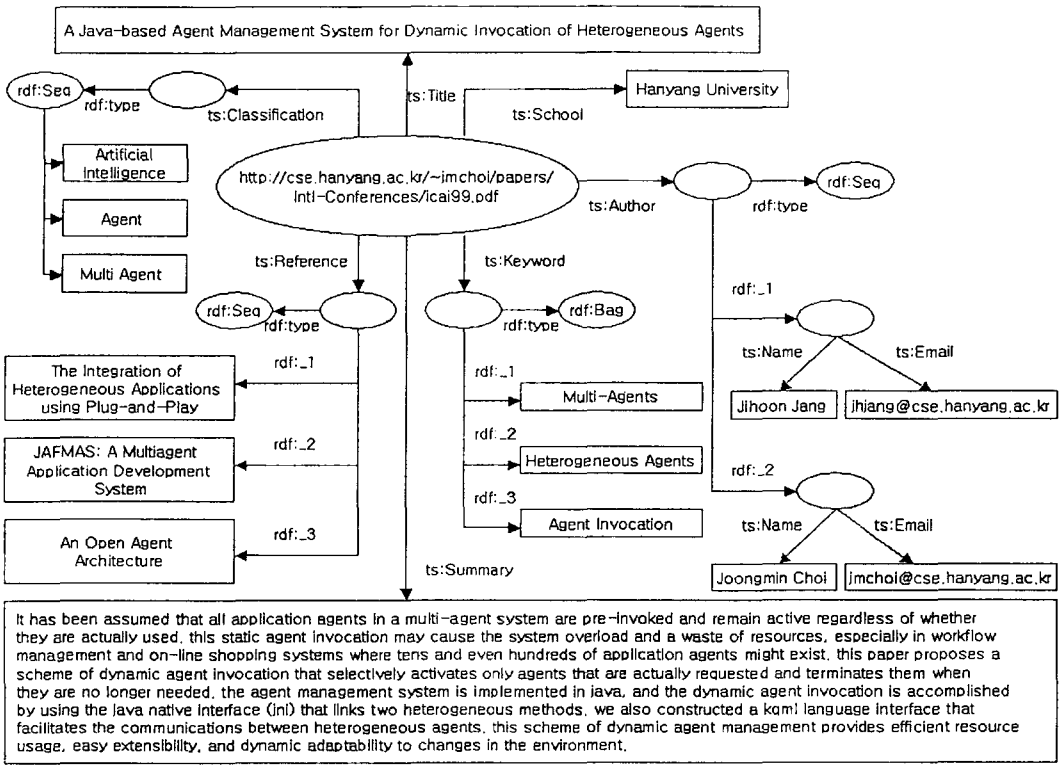
본 논문에서는 Appendix B와 같이 논문에 대한 RDF 스키마를 자체적으로 만들어서 사용하고 있다. "http://islab.hanyang.ac.kr/~sgkang/thesis-rdf-schema#" URI에 논문에 대한 스키마가 정의 되어 있다.

5. 결론 및 향후 연구과제

본 논문에서는 주석 에디터를 사용하여 논문을 요청하고, 전처리 과정을 통해 획득된 문서에서 특징을 추출 한다. 그리고 온토로지 인터페이스를 사용해서 논문을 분류하며, 추출된 메타데이타는 메타데이타 뷰를 통해 볼 수 있으며, HTML 뷰를 통해 메타데이타를 수동으로 수정이 가능하다. 이렇게 생성된 메타데이타는 RDF Repository로 저장할 수 있도록 구현하였다. RDF 메타데이타로 주석 처리된 웹 문서는 웹 로봇이 검색 시 의미적으로 원하는 정보를 쉽게 검색할 수 있도록 성능을 높여 준다.

앞으로의 향후과제는 인공지능 관련 논문뿐만 아니라 다른 웹 문서에도 RDF 메타데이타를 자동 생성할 수 있도록 하는 것이다.

[그림 10] RDF 모델의 인스턴스



참고문헌

[1] José Kahan, Marja-Riitta Koivunen, Eric Prud'Hommeaux, Ralph R. Swick, Annotea: An Open RDF Infrastructure for Shared Web Annotations, In *Proceedings of the WWW10 International Conference*, Hong Kong, May 2001.

[2] Siegfried Handschuh, Steffen Staab, Authoring and Annotation of Web Pages in CREAM, In *Proceedings of the 11th International World Wide Web Conference, WWW 2002*, Honolulu, Hawaii, May 7-11, 2002.

[3] Jeff Heflin, James Hendler, Semantic Interoperability on the Web, In *Proceedings of Extreme Markup Languages 2000*, pp. 111-120, 2000.

[4] Jeff Heflin, James Hendler, Searching the Web with SHOE, In *Artificial Intelligence for Web Search. Papers from the AAAI Workshop.* WS-00-01. AAAI Press, Menlo Park, CA, 2000. pp. 35-40.

[5] Stefan Decker, Sergey Melnik, Frank Van Harmelen, Dieter Fensel, Michel Klein, Jeen Broekstra, Michael Erdmann, Ian Horrocks, The Semantic Web: The Roles of XML and RDF, *IEEE Internet Computing*, Vol. 4(5), pp. 63-74, Sept/Oct 2000.

[6] Jeen Broekstra, Michel Klein, Stefan Decker, Dieter Fensel, Ian Horrocks, Adding formal semantics to the Web building on top of RDF Schema, In *Proceedings of ECDL 2000 Workshop on the Semantic Web*, 21 September 2000.

[7] Steffen Staab, Michael Erdmann, Alexander Maedche, Stefan Decker, An Extensible Approach for Modeling Ontologies in RDF(S), In *Proceedings of ECDL 2000 Workshop on the Semantic Web*, 21 September 2000.

[8] Johan Hjelm, *Creating the Semantic Web with RDF : Professional Developer's Guide*, WILEY Press 2001.

[9] Kasim Selcuk Candan, Hauan Liu, Reshma Suvarna, Resource Description Framework: Metadata and Its Applications, *SIGKDD Explorations*, Volume 3, Number 1, pp. 6-19, July 2001.

[10] Ora Lassila, Ralph R. Swick, Resource Description Framework (RDF) Model and Syntax Specification, Technical Report, W3C Recommendation. http://www.w3.org/TR/REC-rdf-syntax, 1999.

[11] Dan Brickley, R.V. Guha, Resource Description Framework (RDF) Schema Specification, Technical Report, W3C Candidate Recommendation. http://www.w3.org/TR/2000/CR-rdf-schema-20000327, 2000.

[12] Charlotte Jenkins, Mike Jackson, Peter Burden, Jon Wallis, Automatic RDF Metadata Generation for Resource Discovery, In *Proceedings of the 8th International WWW Conference*, 1999.

[13] 정현섭, 개인화 된 웹 네비게이션을 위한 온톨로지 기반 추천 에이전트, 한양대학교 컴퓨터공학과 석사 학위 논문, 2002.

[14] James Hendler, Agents and the Semantic Web, *IEEE Intelligent Systems*, 16(2), pp. 30-37, 2001.

[15] Nicola Guarino, Formal Ontology in Information Systems, In *Proceedings of FOIS'98*, IOS Press, pp. 3-15, 1998.

[16] Brian McBride, Jena: Implementing the RDF Model and Syntax Specification, In *Proceedings of the Second International Workshop on the Semantic Web - SemWeb'2001*, Hong Kong, China, May 1, 2001.

Appendix A

```
<?xml version='1.0'?>
<rdf:RDF
        xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
        xmlns:RDFNsId1='http://islab.hanyang.ac.kr/~sgkang/thesis-rdf-schema#'>
        <rdf:Description rdf:about='http://cse.hanyang.ac.kr/~jmchoi/papers/
                        Intl-Conferences/icai99.pdf'>
            <RDFNsId1:title>A Java-based Agent Management System for Dynamic Invocation of Heterogeneous
                        Agents</RDFNsId1:title>
            <RDFNsId1:summary>It has been assumed that all application agents in a multi-agent system are pre-
                                invoked and remain active regardless of whether they are actually used. this static
                                agent invocation may cause the system overload and a waste of resources,
                                especially in workflow management and on-line shopping systems where tens
                                and even hundreds of application agents might exist. this paper proposes a
                                scheme of dynamic agent invocation that selectively activates only agents that are
                                actually requested and terminates them when they are no longer needed. the agent
                                management system is implemented in java, and the dynamic agent invocation is
                                accomplished by using the java native interface (jni) that links two heterogeneous
                                methods. we also constructed a kqml language interface that facilitates the
                                communications between heterogeneous agents. this scheme of dynamic agent
                                management provides efficient resource usage, easy extensibility, and dynamic
                                adaptability to changes in the environment.
            </RDFNsId1:summary>
            <RDFNsId1:school>Hanyang University</RDFNsId1:school>
            <RDFNsId1:classification>
                <rdf:Seq>
                    <rdf:li>Artificial Intelligence</rdf:li>
                    <rdf:li>Agent</rdf:li>
                    <rdf:li>Multi Agent</rdf:li>
                </rdf:Seq>
            </RDFNsId1:classification>
            <RDFNsId1:keyword>
                <rdf:Bag>
                    <rdf:li>Multi-Agents</rdf:li>
                    <rdf:li>Heterogeneous Agents</rdf:li>
                    <rdf:li>Agent Invocation</rdf:li>
                </rdf:Bag>
            </RDFNsId1:keyword>
            <RDFNsId1:author>
                <rdf:Seq>
                    <rdf:li rdf:parseType='Resource'>
                        <RDFNsId1:name>Jihoon Jang</RDFNsId1:name>
                        <RDFNsId1:mail>jhjang@cse.hanyang.ac.kr</RDFNsId1:mail>
                    </rdf:li>
                    <rdf:li rdf:parseType='Resource'>
                        <RDFNsId1:name>Joongmin Choi</RDFNsId1:name>
                        <RDFNsId1:mail>jmchoi@cse.hanyang.ac.kr</RDFNsId1:mail>
                    </rdf:li>
                </rdf:Seq>
            </RDFNsId1:author>
            <RDFNsId1:reference>
                <rdf:Seq>
                    <rdf:li>The Integration of Heterogeneous Applications using Plug-and-Play</rdf:li>
                    <rdf:li>JAFMAS: A Multiagent Application Development System</rdf:li>
                    <rdf:li>An Open Agent Architecture</rdf:li>
                </rdf:Seq>
            </RDFNsId1:reference>
        </rdf:Description>
</rdf:RDF>
```

Appendix B

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

    <rdf:Description ID="Author">
      <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
      <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Creator"/>
      <rdfs:label>Author</rdfs:label>
      <rdfs:comment>An entity primarily responsible for making the content of the resource.</rdfs:comment>
    </rdf:Description>

    <rdf:Description ID="Title">
      <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
      <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Title"/>
      <rdfs:label>Title</rdfs:label>
      <rdfs:comment>The title of the resource taken from TITLE element of HTML.
      </rdfs:comment>
    </rdf:Description>

    <rdf:Description ID="Summary">
      <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
      <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Description"/>
      <rdfs:label>Summary</rdfs:label>
      <rdfs:comment>An account of the content of the resource.
      </rdfs:comment>
    </rdf:Description>

    <rdf:Description ID="Keyword">
      <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
      <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Subject"/>
      <rdfs:label>Keyword</rdfs:label>
      <rdfs:comment>The topic of the content of the resource.</rdfs:comment>
    </rdf:Description>

    <rdf:Description ID="Reference">
      <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
      <rdfs:label>Reference</rdfs:label>
    </rdf:Description>

    <rdf:Description ID="School">
      <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
      <rdfs:label>School</rdfs:label>
    </rdf:Description>

    <rdf:Description ID="Name">
      <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
      <rdfs:label>Name</rdfs:label>
    </rdf:Description>

    <rdf:Description ID="Email">
      <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
      <rdfs:label>Email</rdfs:label>
    </rdf:Description>

    <rdf:Description ID="Classification">
      <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
      <rdfs:label>Classification</rdfs:label>
      <rdfs:comment>This is an Onotlogy-based classification that has been assigned to the document as a result of the
      automatic classification process.</rdfs:comment>
    </rdf:Description>
</rdf:RDF>
```