메타데이터 연계성을 위한 RDF 응용스키마설계에 관한 연구

A Study on Designing Schemata with RDF for Interoperability of Metadata Formats

김이겸(Lee-Kyum Kim)*, 김태수(Tae-Soo Kim)**

목 차

초 록

상이한 메타데이터형식 상호간 어의적인 연계성을 확보하는 일은 메타데이터의 재사용성을 가능하게 할 뿐 아니라 메타데이타 처리에 일관성을 제공할 수 있다는 점에서 대단히 중요한 요인이라 할 수 있다. 이를 위해 RDF에 대한 제반 특성을 검토하고, 어의적 연계성을 위한 공통요소를 추출하였다. 이들 공통요소를 RDF스키마명세와 구문명세에 기반하여 공통스키마로 설계함으로써 어의적 연계성을 확보한 메타데이터스키마 설계에 기반으로 삼았으며, 공통스키마의 활용을 제시하기 위해 KORMARC를 표본으로 스키마를 설계하였다.

ABSTRACT

Keeping the semantic interoperability between different formats is very important for reuseing and consistently processing metadata. This article is to design common schema and KORMARC schema based on RDF schema. For this, common elements are made by comparison of Dublin Core elements and KORMARC fields. The schema of these elements is designed for construction of semantic interoperability, basing RDF schema specification and syntax specification. Eventually, the semantic interoperability between a number of metadata formats can be constructed by matching the common elements to each attribute of format. KORMARC schema is designed by the application of common schema.

키워드: RDF스키마, RDF구문, RDF모형, 메타데이터스키마, KORMARC 스키마

* 광주대학교 인문사회대학 문헌정보학과
** 연세대학교 문과대학 문헌정보학과
■ 논문 접수일 : 1999년 12월 20일

1 서 론

1.1 연구 목적

웹(Web)의 비약적인 발전은 기존의 도메인 중심 정보환경에서 웹 중심의 분산정보환경으로의 급속한 변화를 가져왔다. 이와 함께 웹상에 존재하는 무제한의 정보자료로부터 비롯된 정보기술(情報記述)환경의 변화는 전통적인 MARC형식의 비판과 더불어 웹에 기반한 새로운 메타데이터형식의 대량 출현을 가져오게 되었다. 그러나 이러한 상이한 형식의 난립은 메타데이터 상호간의 호환성 문제에 직면하게 되어 정보의 공유를 달성하는데 저해요인이 되고 있다.

한편 서로 다른 메타데이터형식간에는 기술요소의 범위나 종류, 어의, 요소명 등의 차이로 인하여 기술요소의 식별이나 정보의 공유를 위해서는 형식상호간 연계성 확보가 필수적이라 할 수 있다. 그러나 연계성 확보 수단으로 활용되고 있는 상이한 형식간 연결테이블(Mapping Table; Crosswalk)은 구문적 연계성을 포함하지 않으며, RDF(자원기술모형)는 각 형식이 정의하고 있는 기술요소를 그대로 사용할 수 있는 방법을 제공하면서 구문적인 통일성을 포함하지만, 기술요소간 어의적인 연계성을 제공하지 않는 문제가 있다. 이를 메타데이터의 공유라는 관점에서 보면 연결테이블은 특정 형식으로의 어의적 변환에, RDF는 구문적인 변환에만 활용될 수 있는 한계가 지적될 수 있다.

자원기술모형(RDF: Resource Description Framework)은 다양한 메타데이터 형식과의 연계를 통한 통합기술모형으로서(Lassila, and Swick 1999), 메타데이터 스키마 참조를 기초로 모든 형식의 메타데이터를 RDF구문으로 수용하기 위한 것이다. 이는 각 형식의 요소기준을 그대로 수용한다는 점에서 연결테이블이 갖는 단점을 극복할 수 있으나, 반대로 구문적인 통일성 확보를 근간으로 한 것이기 때문에 어의적인 연계성은 고려되지 않았다고 할 수 있다. 따라서 RDF구문으로 통합된 상이한 형식의 메타데이터가 형식상호간 어의적인 연계성을 확보할 수 있다면 메타데이터의 재사용성을 높일 수 있음은 물론 메타데이터의 처리에 유용성을 제공할 수 있을 것이다.

이러한 관점에서 본 연구는 RDF를 활용한 메타데이타의 기술(記述)이나 상이한 형식의 메타데이터간 어의적인 연계성을 제공하기 위한 방안을 연구하고자 한다.

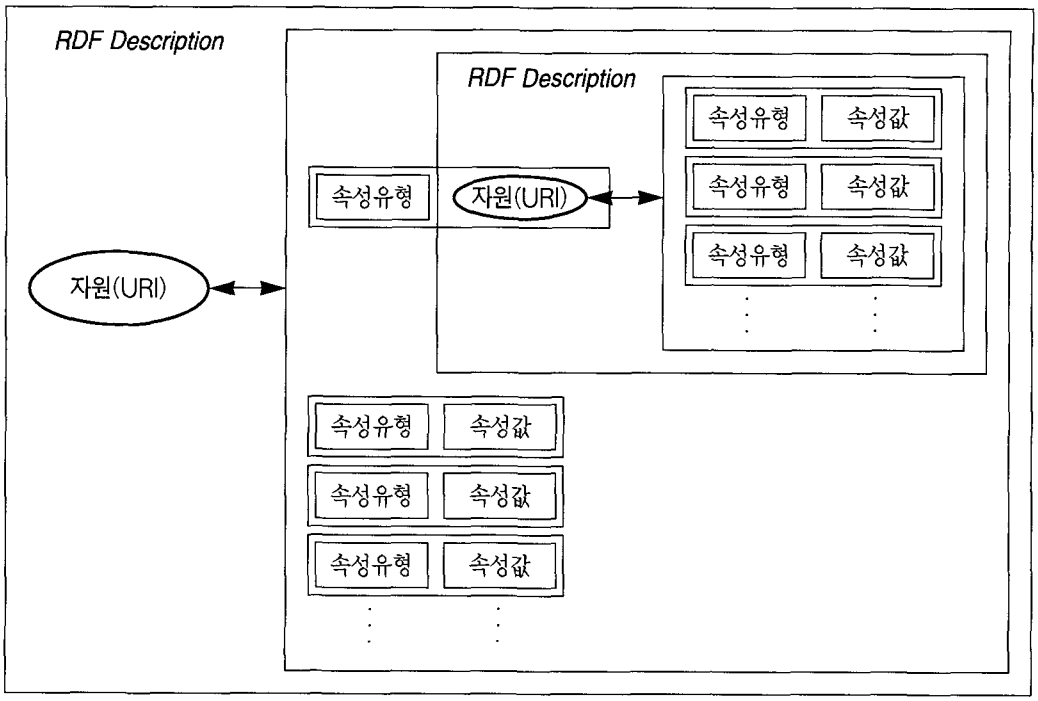
1.2 연구범위 및 방법

본 연구의 목적을 달성하기 위해서 RDF스키마 명세(Brickley and Guha 1999)와 RDF구문명세(Lassila and Swick 1999)에 근거하여 어의적 연계성확보를 위한 공통스키마를 설계하고, 이를 메타데이터 스키마 설계시 연계될 수 있도록 KORMARC을 표본으로 그 스키마를 설계하였다.

공통스키마에 포함된 속성은 더블린코어의 15개 요소 및 하위요소를 기초로 하고, 이들을 전통적인 기술구조인 KORMARC의 데이터 요소와 비교, 연구를 통하여 선택하였다.

1.3 연구의 제한점

웹자료를 기술하기 위한 메타데이터형식의 수(數)가 급속하게 증가되고 있으며, 그 상세성의 정도도 편차가 심하기 때문에 어의적인 연계성을 위한 공통요소를 도출하기 위해서는 다양한 형식의 기술요소에 대한 검토가 요구되지만, 본 논문에서는 KORMARC와 더블린코어형식만으로 제



〈그림 1〉 RDF 모형

한하였다.

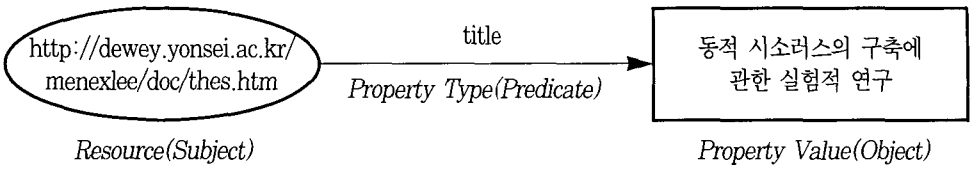
또한 어의적인 연계성 기준을 위하여 BIBLINK 프로젝트(Heery 1996, 44), Nordic(Hakala 1998) 및 DESIRE(1996) 등의 메타데이터 평가에서 간결성을 확보한 적절한 메타데이터 형식으로 추천된 더블린코어형식과 도서관이나 유관기관에서 전통적으로 목록레코드 생산을 위해 사용되어온 KORMARC형식만을 연구 대상으로 하였기 때문에 기타 형식에 대한 평가도 제외하였다.

공통요소를 통한 기술요소간 연계성의 문제와 RDF구문에서 메타데이터의 기술에 활용되는 메타데이터 스키마설계를 통한 어의적인 연계성을 확보하는 방안을 연구의 대상으로 하였기 때문에 RDF 메타데이터 생성을 위한 저작도구(template)나 검증도구 등의 설계 및 구현 등의 내용도 제외하였다.

2 자원기술모형(RDF)

2.1 RDF모형

RDF모형은 자원(Resources), 속성(Properties), RDF기본기술문(Statements)의 3가지 객체유형으로 구성된다(Lassila and Swick 1999). 자원은 식별자(URI와 ID)로 식별되는 모든 것 즉, 기술대상을 의미한다. 속성은 자원 기술을 위해 사용된 관계나 특징을 나타내며, 속성유형과 속성값으로 구성되지만(Lannella 1999), 속성은 보통 메타데이터 기술에서 기술속성(attribute)과 같은 의미로 사용된다. 속성값은 다른 자원이거나 리터럴(Literal)이 될 수 있다. 또한 RDF기본기술문은 단일 속성명과 그 속성값을 갖는 자원으로 정의할 수 있다.

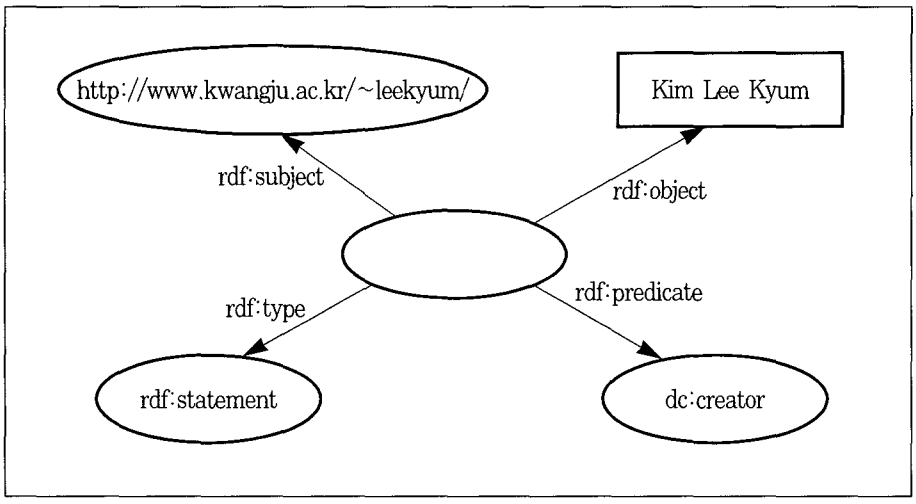


〈그림 2〉 RDF기본기술문(Statements)

따라서 RDF모형은 속성(속성유형+속성값)의 집합을 갖는 복수자원으로 구성되며, RDF구문명세에서 Description으로 구체화된다고 할 수 있다. 또한 RDF기본기술문이 단일 자원에 대한 속성유형과 속성값으로 구성되기 때문에 RDF모형은 RDF기본기술문의 집합으로 구성된다고 할 수 있다. 〈그림 1〉은 RDF모형과 구문명세(Lassila and Swick 1999)의 내용을 중심으로 RDF모형을 그림으로 나타낸 것이다.

RDF의 기본 단위는 RDF기본기술문(Statement)이며, 자원(Subject), 속성유형(Predicate), 속성값(Object)으로 구성된다. 〈그림 2〉는 RDF기본기술문을 방향성 그래프로 나타낸 것으로서 〔http://dewey.yonsei.ac.kr/memexlee/doc/thes.htm〕는 자원이며, 〔title〕은 속성유형, 〔동적 시소러스의 구축에 관한 실험적 연구〕는 속성값이라 할 수 있다.

RDF모형은 이와 같이 기본기술문을 단위로 하지만, 이를 더욱 구체화하면 4개의 기본기술문으로 모형화될 수 있다. 〈그림 3〉은 〈그림 2〉의 RDF기본기술문을 구체화하여 방향성 그래프를 사용하여 모형화한 것이다. 〈그림 3〉에서 보는 바와 같이 RDF기본기술문은 4개의 속성을 갖는 자원으로 모형화할 수 있으며, 모형화된 각각의 기본기술문을 "구체화된 기본기술문(Reified Statement)"라고 부른다. 〈그림 3〉은 중앙의 타원(자원)을 중심으로 4개의 속성과 속성값을 보여주고 있다.



〈그림 3〉 구체화된 기본기술문(Reified Statement)

이와 같이 모든 RDF기본기술문은 자원, 속성, 속성값으로 일관되게 표현되기 때문에 RDF모형은 triple(속성, 자원, 속성값)형태로 간결하게 나타낼 수 있다. 따라서 동일한 내용은 항상 동일한 RDF트리(Tree)를 갖게 된다. 이에 반하여 XML모형은 동일 내용의 다양한 표현이 가능하기 때문에 복수의 XML트리를 갖게 되고, 따라서 이를 논리적인 트리로 매핑하는 다양한 방법이 존재하게 됨으로써 XML트리에 대한 질의(Query)가 복잡하다고 할 수 있다(Berners-Lee 1998). 이는 RDF모형을 XML모형로 나타내면 복수의 XML모형으로 결과되기 때문에 기술구조를 파악하는 어려움이 있음을 지적한 것으로서 RDF모형의 유용성을 제시한 것이라 할 수 있다.

그러나 RDF를 적용하여 메타데이터를 기술(記述)하기 위해서는 속성유형으로 사용될 속성명을 해당 메타데이터 스키마에 정의된 것을 사용하기 때문에 메타데이터 스키마는 RDF기술(記述)의 핵심을 이룬다 할 수 있다. 따라서 메타데이터 스키마를 연계할 수 있는 경우 모든 형식을 RDF기술(記述)에서 혼용할 수 있으며, 동시에 이들 스키마에 기술속성간 어의적인 연계성을 부여할 수 있다면 메타데이터의 운용성을 확보할 수 있을 것이다.

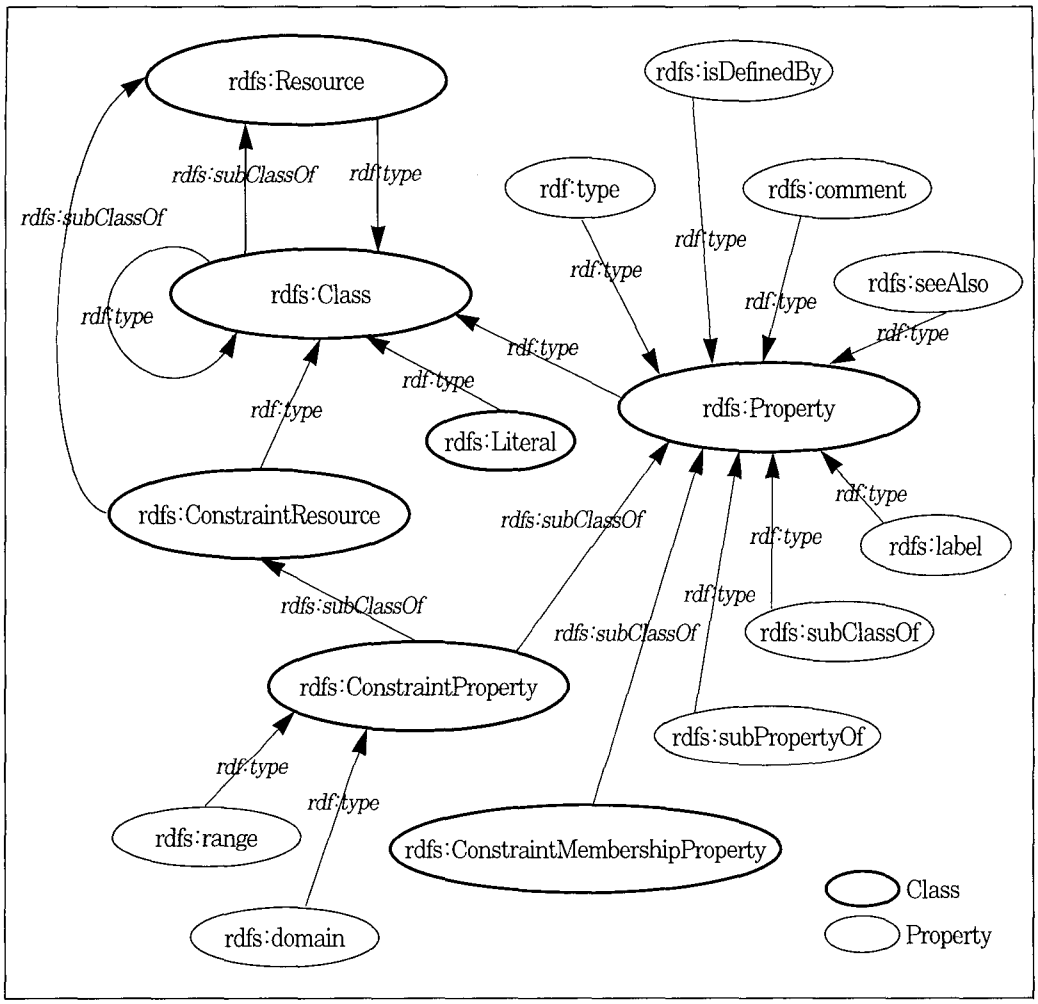
2.2 RDF스키마

RDF모형에서는 속성(Properties)을 선언하거나 속성과 자원과의 관계를 규정하고 있지 않으며, 이들을 RDF스키마명세(Brickley and Guha 1999)에 의해 구체화하였다. 그러나 이 명세는 서명이나 저자와 같은 기술요소를 직접 정의하거나 제공하는 것이 아니라, 그러한 기술요소나 클래스를 정의하는 수단을 제공하는 스키마정의 언어라 할 수 있다.

RDF스키마는 다른 RDF자원을 정의하는데 사용될 수 있는 자원들의 집합이며, 이들 자원(Resources)들은 클래스로부터 생성된 인스턴스(Instance)라고 할 수 있다. 즉, 메타데이터 스키마와 같은 응용스키마 정의에 사용될 수 있도록 Class, Resource, Property 등의 클래스와 이들을 인스턴스화하여 새로운 자원을 정의할 수 있도록 유형시스템(Type System)을 제공하고 있다.

〈그림 4〉는 RDF스키마명세에서 제공되는 자원들의 명세로서 rdfs:Class로부터 rdf:type속성을 사용하여 각각의 인스턴스(자원)가 생성되는 과정을 보여주고 있다. 즉, rdf:Property는 rdfs:Class의 인스턴스로, rdfs:comment는 rdf:Property의 인스턴스로 생성된 것이라 할 수 있다. 따라서 rdf:type 속성은 새로운 인스턴스를 생성하기 위한 수단이라 할 수 있으며, 이는 객체지향시스템에서 객체의 선언을 통하여 새로운 객체를 생성하는 것과 개념적으로 동일하다 할 수 있다. 결국, RDF스키마를 사용한 메타데이터 스키마는 RDF스키마 명세에 제공된 자원들을 활용하여 정의될 수 있으며, 구체적으로는 유형시스템(rdf:type)을 사용하여 클래스와 속성의 인스턴스(객체)를 생성하는 것이라 할 수 있다. 〈그림 4〉에 사용된 rdf나 rdfs는 구문명세와 스키마명세의 위치를 NameSpace에서 정의한 명칭이며, 콜론 뒤의 명칭은 자원명이라 할 수 있다.

한편, RDF스키마명세는 클래스와 속성의 구분을 위해 그 명명법(命名法)을 규정하고 있다. 즉, 클래스명(Class Name)은 첫 글자를 대문자로 하되, 2개 이상의 단어가 조합될 경우 단어마다 그 첫 글자를 대문자로 하고 나머지는 모두 소문자로 하며, 속성명(Property Name)은 모두 소문자로 하되, 2개 이상의 단어가 조합되는 경우 2



〈그림 4〉 RDF스키마의 유형과 계층

번째 이하의 단어들은 각각 그 첫 문자를 대문자로, 나머지 모두를 소문자로 표기한다.

2.2.1 RDF 핵심 클래스(Core Class)

객체지향개념에서 클래스는 구조와 메소드(Method)를 정의한 추상화된 틀이라 할 수 있다. 객체(Object)는 클래스를 구체화한 것으로서 클래스가 갖는 모든 속성 곧, 구조와 메소드를 상속받는다고 할 수 있으며, 클래스의 인스턴스를 객체라고도 부른다(구자철 1997, 61-63). 그러나 객체선언을 통하여 객체를 생성하는 일반적인 객체지향시스템과는 다르게 RDF스키마는 유형시스템을 통하여 해당 클래스의 객체(인스턴스)를 만드는 간접적인 수단을 제공하며, 따라서 메타데이터 스키마에서 각각의 속성들은 RDF스키마에 정의된 클래스의 인스턴스화 과정을 통하여 생성된 일종의 객체라고 할 수 있다. 다만, RDF스키마에서의 클래스는 메소드(Method)를 포함

하지 않으며, 공용(Public)모드, 보호(Protected)모드, 지역(private)모드 등의 구분없이, 모두 공용(public)모드로 정의되었다는 점이 다른 스키마 언어와 다르다(Chang 1998).

다음은 메타데이터 스키마 정의에 활용될 수 있는 RDF스키마명세(Brickley and Guha 1999)에 제시된 자원들의 의미와 내용을 정리한 것이다.

1) rdfs:Resource

RDF에 의해 기술되는 모든 대상을 자원(Resources)이라 하며, 이들 자원들은 클래스 rdfs:Resource의 인스턴스이다. RDF모형과 구문명세(Lassila and Swick 1999)의 RDF공식모형에 사용된 자원들을 나타낸다. 이것은 자바에서 객체(Object)개념과 유사하다. 다만, RDF/XML 연속구문에서 사용되는 구문속성(attribute)으로서의 'resource'와는 구별된다.

2) rdf:Property

이는 RDF 자원의 하위세트인 Property를 나타내며, 메타데이터 스키마 정의시 기술속성들은 모두 이 클래스유형을 갖는다. 다음은 RDF연속구문을 사용한 스키마정의 예로서 rdf:Property을 인스턴스화하여 새로운 title속성을 정의한 것이다.

```
〈rdf:Description ID="title"〉
    〈rdfs:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns
      #Property"/〉
〈rdfs:comment〉...〈/rdfs:comment〉
    ........
〈/rdf:Description〉
```

3) rdfs:Class

이 클래스는 카테고리(category)라는 개념과 일치하며, 자바와 같은 객체지향 프로그래밍언어에서의 클래스(Class)와 유사하다. 스키마에서 새로운 자원을 클래스로 정의할 때 rdf:type의 속성값은 rdfs:Class가 된다. RDF의 클래스들은 웹페이지, 문서, 데이터베이스, 사람 등 거의 모든 것을 나타내기 위해 정의될 수 있다. 다음의 연속구문을 사용한 정의 예에서 자원 TitleClass는 rdfs:Class를 rdf:type 속성값으로 정의한 것이며, rdfs:Class의 모든 자원을 공유한다고 할 수 있다.

```
〈rdf:Description ID="TitleClass"〉
    〈rdf:type resource=
      "http://www.w3.org/TR/1999/PR-rdf-schema-
      19990303#Class"〉
    〈rdfs:subClassOf=
      "http://www.w3.org/TR/1999/PR-rdf-schema-
      19990303#Resource"/〉
    〈rdfs:label〉표제클래스〈/rdfs:label〉
〈/rdf:Class〉
```

2.2.2 RDF 핵심 속성(Core Property)

RDF의 모든 속성들은 rdf:Property의 인스턴스들이며, 클래스와 그들의 인스턴스 또는 클래스와 상위클래스간의 관계를 정의하는데 사용된다. 아래 예에서 보는 바와 같이 subClassOf는 rdf:type에 의해 Property 클래스의 인스턴스(객체)로 생성된 것이며, 상위클래스와 하위클래스간의 관계를 나타내는데 사용된다.

```
〈rdf:Description ID="subClassOf"〉
    〈rdf:type resource=
      "http://www.w3.org/TR/1999/PR-rdf-schema-
      19990303#Property"/〉
    〈rdfs:label〉subClassOf〈/rdfs:label〉
    〈rdfs:comment〉Indicates membership of a class
      〈/rdfs:comment〉
    〈rdfs:domain rdf:resource=
      ""http://www.w3.org/TR/1999/PR-rdf-schema-
      19990303#Class"/〉
```

```
〈rdfs:range rdf:resource=
  ""http://www.w3.org/TR/1999/PR-rdf-schema-
  19990303#Class"/〉
〈/rdf:Description〉
```

1) rdf:type

자원이 특정 클래스의 구성원임을 나타내기 위해 사용하며, 이 때 자원은 클래스가 갖는 모든 특징을 상속받는다. rdf:type 속성값은 rdfs:Class의 인스턴스이어야 하며, 생성되는 자원은 이 클래스의 인스턴스(객체)가 된다. 따라서 rdf:type 속성은 클래스의 인스턴스를 만드는 수단(객체생성수단)이며, 동시에 이를 통하여 응용 스키마에서 기술속성(인스턴스)을 정의할 수 있는 수단을 제공한다 할 수 있다.

2) rdf:subClassOf

이 속성은 클래스간의 계층관계를 표현하기 위한 것으로서 클래스 확장 수단을 제공한다 할 수 있다. rdf:Class의 인스턴스만이 속성 rdf:subClassOf를 갖을 수 있으며, 그 속성값은 항상 rdf:type rdf:Class가 된다. 특정 클래스는 복수 클래스의 하위클래스가 될 수 있으나 그 자체를 하위클래스로 정의할 수 없다.

3) rdfs:subPropertyOf

이 속성은 특정 속성의 세분 속성을 표현하기 위한 것으로 속성 확장 수단을 제공한다. 따라서 속성들은 하위속성이 없거나 subPropertyOf를 사용하여 복수의 하위속성을 정의할 수 있다. 다만, 특정 속성이 자신을 하위속성으로 선언될 수 없다.

4) seeAlso

이 속성은 주제정보를 나타내는 자원을 표현하기 위한 것으로 필요에 따라 subPropertyOf 속성을 사용하여 더 세분할 수 있다. 다만 클래스인 rdfs:Resource의 인스턴스인 경우에만 적용될 수 있다.

5) isDefinedBy

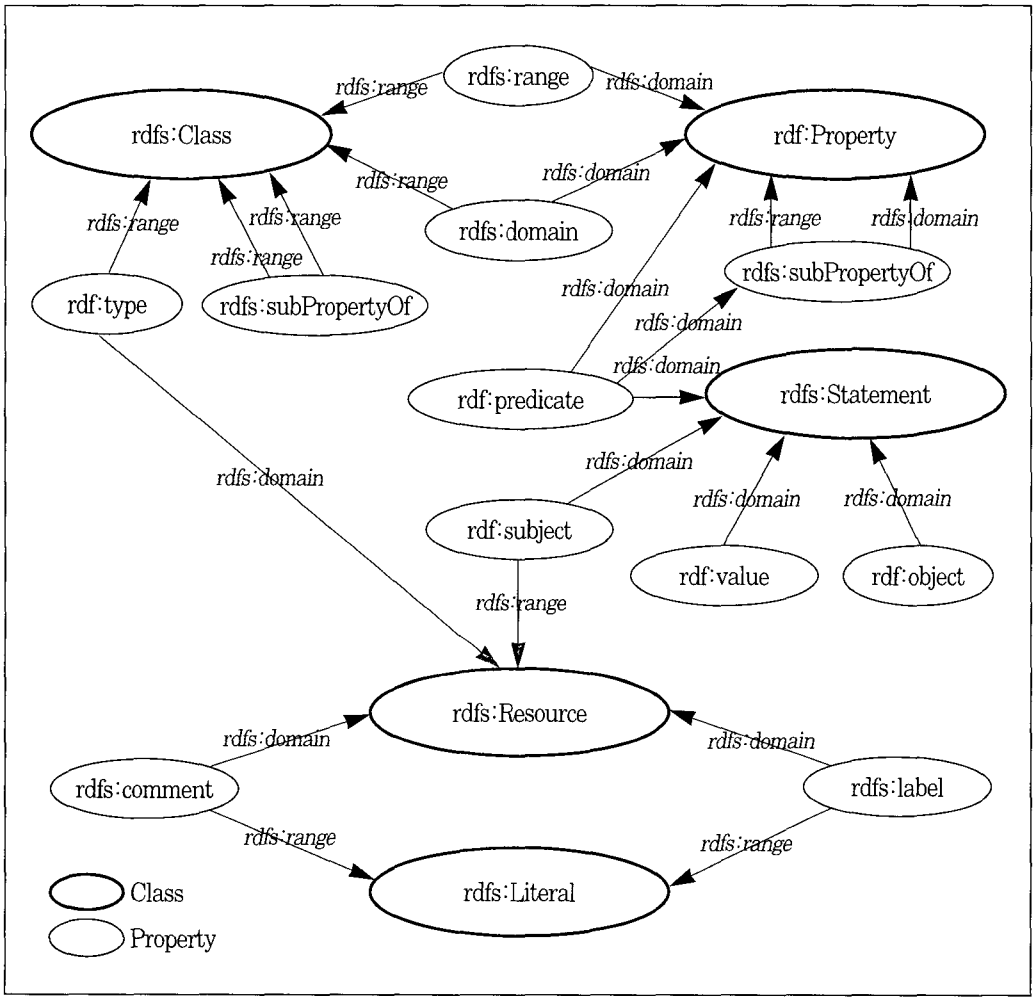
이 속성은 rdfs:seeAlso속성의 하위속성으로 주제자원을 정의하는 자원을 정의하기 위해 사용된다. seeAlso속성과 마찬가지로 rdfs:Resource의 인스턴스들에만 적용할 수 있는 속성이다.

2.2.3 RDF 제한 속성

RDF스키마는 속성을 정의할 때 그 속성과 관련한 제한을 정의하는 수단을 제공하고 있다. 이는 속성에 의해 클래스가 설계되는 다른 객체지향시스템과는 다르게, 클래스에 의해 속성이 정의되기 때문이라 할 수 있다. 예를 들면 전형적인 객체지향 시스템은 속성값이 문자열(Literal)인 author속성을 갖는 클래스 Book을 정의하는 클래스 중심 정의방식을 사용하지만, RDF스키마는 클래스가 Book에 속하고, 속성값이 문자열인 author속성을 정의하는 속성 중심 정의 방식을 사용한다.

따라서 RDF스키마는 속성과 별도로 필요한 클래스를 정의하고, 속성 정의시 이들 속성이 속하거나 제한 받을 필요가 있는 클래스를 선택하는 방식이라 할 수 있다. 이 클래스 선택을 위해 제한 속성들이 사용되며, 중요한 것으로는 rdfs:domain과 rdfs:range 속성을 들 수 있다. 이들 제한 속성은 속성값을 제한하는데 사용된다.

〈그림 5〉는 RDF스키마명세에 제시된 클래스와 각 속성간의 관계를 제한 속성을 사용하여 표현한 것이다. 이 그림에서 rdfs:label은 rdfs:Resource에 속하며, 그 속성값은 rdfs:Literal이어야 함을 의미한다 하겠다.



〈그림 5〉 클래스와 속성간의 관계

1) rdfs:ConstraintResource

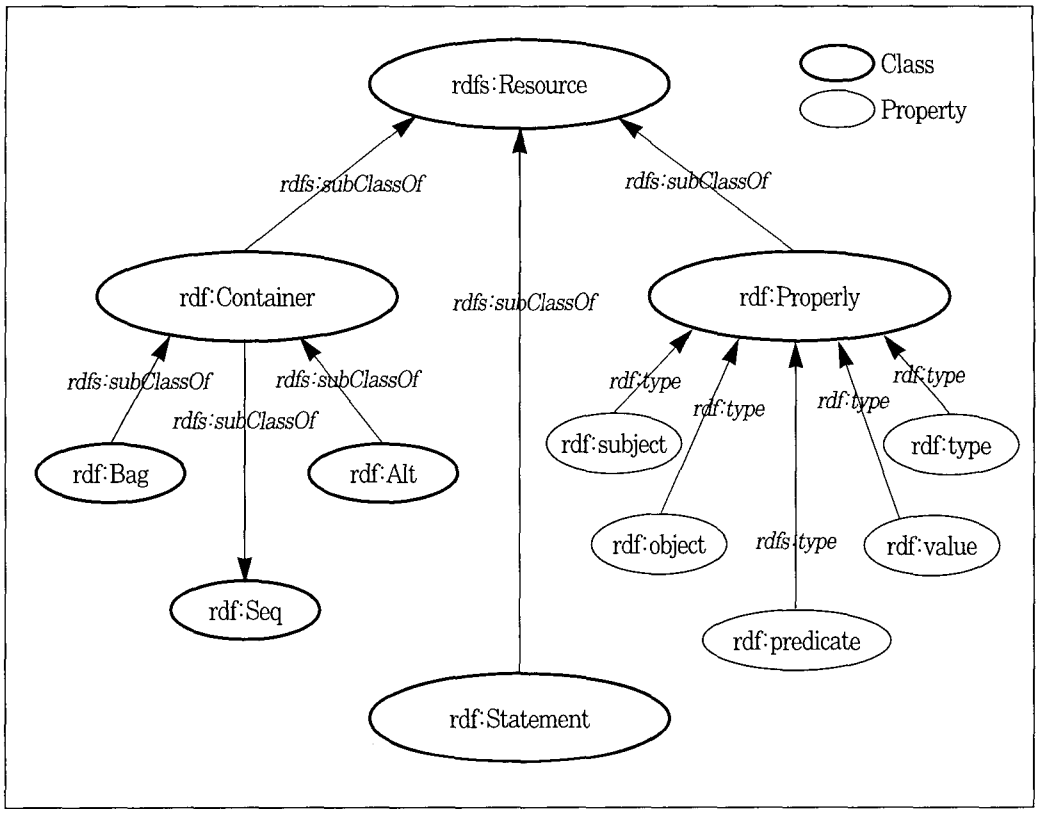
이 자원은 rdfs:Resource의 하위클래스이며, 이 클래스의 목적은 RDF 프로세서(Processor)가 RDF모델의 일관성을 검증할 수 있는 수단을 제공하기 위한 것이다.

2) rdfs:ConstraintProperty

이 자원은 rdf:Property의 하위 클래스이며, 이 클래스의 인스턴스들은 제한사항을 기술하는데 사용되는 속성들이다. 또한 이 자원은 rdfs:ConstraintResource의 하위 클래스이다. 아래의 rdfs:range와 rdfs:domain은 이 클래스의 인스턴스이다.

3) rdfs:range

이 속성은 클래스 ConstraintProperty의 인스턴스로서 속성값을 제한하기 위하여 사용된다. range속성의 속성값은 항상 클래스가 되며, 한



〈그림 6〉 RDF구문을 위한 스키마 계층

속성은 복수의 rdfs:range속성을 포함할 수 없다. 다만, range속성을 사용하지 않을 수는 있다. 따라서 rdfs:range가 클래스 A인 속성값은 클래스 A의 인스턴스에 제한을 받게 된다.

메타데이터 스키마의 속성정의에서 rdfs:range는 속성값의 데이터 유형(Type)을 한정하는데 사용될 수 있으나 현재 제공된 데이터유형은 문자열 클래스(rdf:Literal)만을 지원하므로 메타데이터 기술과 관련하여 Numeric이나 Date 등을 위한 클래스를 제공할 필요가 있다.

4) rdfs:domain

이 속성은 클래스 ConstraintProperty의 인스턴스로서 특정 속성이 사용되는 클래스를 한정하기 위해 사용된다. 특정 속성은 복수의 rdfs:domain속성을 포함할 수 있다.

2.3 RDF 구문

MARC형식이 목록데이터의 상호교환을 위한 표준형식인 것처럼 메타데이터를 교환할 수 있는 공동의 구문을 제공하기 위해 웹 컨소시엄(W3C)에서는 RDF의 기술(encoding)구문으로 XML을 채택하였다.

XML은 SGML과 HTML의 단점을 보완한 것으로서 웹페이지내에서 사용자 DTD를 선언하여

기존의 DTD와 함께 사용할 수 있는 등 유연성이 우수하며(채규혁 1998, 42-55), 메타데이터 스키마에 정의된 기술속성과의 연계 수단으로 Name Space기능을 제공함으로써 RDF에 적합한 기술 환경을 제공한다 할 수 있다. W3C는 RDF모형과 구문명세(Lassila and Swick 1999)에서 RDF의 공식구문을 제시하고, RDF구문기술을 위한 클래스와 속성을 RDF스키마(Brickley and Guha 1999)에 정의하였다. 따라서 RDF구문은 이들 자원과 메타데이터스키마에 정의된 기술요소(RDF스키마 자원의 인스탄스)를 기반으로 기술된다.

〈그림 6〉은 이들의 관계를 계층적으로 나타낸 것으로 RDF스키마를 토대로 재구성한 것이다(제한사항은 〈그림 5〉참조). RDF스키마도 RDF구문으로 기술되기 때문에 이들 내용이 RDF스키마 명세에 포함되지만, RDF구문명세에 제시된 클래스와 속성을 일별하기 위해 제시된 것이다. 〈그림 6〉에서 보는 바와 같이 RDF구문명세는 크게 동일 기술속성을 갖는 복수의 속성값을 기술하기 위한 3가지의 컨테이너 클래스와 기본 기술속성들을 제공하며, 이들 클래스와 속성들은 메타데이터 스키마와 함께 메타데이터의 기술에 활용된다.

RDF구문은 복수의 기본기술문(Statements)들로 구성되는 Description을 기반으로 하고 있다. 즉, Description은 그 안에 기술되는 복수의 속성과 속성값의 공동 자원이 되기 때문에 단일 웹자원을 중심으로 복수의 기본기술문들이 기술되는 형식을 갖게 된다. 이때 자원은 Description의 세부속성인 about나 ID의 속성값으로 주어 지며, about속성값은 URI로, ID속성값은 ID명으로 기술된다. ID속성은 내부자원(인-라인 자원) 참조를 지원하기 위한 것으로서 이 속성을 포함하는 Description은 새로운 자원을 생성한다는 의미이며(Lassila and Swick 1999), 다른 Decsription에서 이 Description을 참조할 때 ID 속성값을 사용하게 된다.

RDF구문은 인터넷 자료을 기술하기 위한 구문이기 때문에 기본적으로 URI로 접근할 수 없는 비전자자료의 경우에는 메타데이터를 비전자자원의 대체물로 생성하고, 이를 새로운 Description에서 ID속성값으로 연결하기 때문에 ID속성값으로 사용될 정보를 규정할 필요가 있다. 그러나 본 논문에서는 메타데이터간의 연계성을 확보할 수 있는 방안을 연구대상으로 하였기 때문에 스키마설계에 적용되는 구문만으로 한정하고자 한다.

2.3.1 Name Space기능

RDF모형은 다양한 자원과의 연계를 통하여 자원을 기술하는 연계모형이기 때문에 최소한의 기술속성과 연계방안 및 기술모형만을 제공한다 할 수 있다. 즉, RDF는 메타데이터의 기술속성을 포함하지 않으므로 메타데이터 스키마를 연계함으로써 기술속성을 사용한다 할 수 있다.

XML의 Name space기능은 다른 자원과의 연계 수단이며, 메타데이터기술에 사용된 기술속성의 의미를 정확하게 표현하기 위해 메타데이터 스키마를 연계(참조)하는 수단이라 할 수 있다. 따라서 이 기능(Bray 1999)을 사용하여 사용할 메타데이터 스키마를 접두어와 함께 선언하고, 선언된 이름(접두어)을 스키마가 정의하고 있는 속성(Element)들과 결합하여 사용할 수 있는 방법을 제공한다 할 수 있다.

아래의 예는 "http://www.w3c.org/1999/02/22-rdf-syntax-ns#"과 "http://purl.org/metadata/dublin_core#"에 위치한 스키마를 연계하여 사용할 것과 이들 URL을 각각 rdf, dc로

대신할 것을 Name Space로 선언한 것이다. 선언에서 URI의 마지막에 부여된 #는 완전한 참조를 생성하기 위하여 Name Space명과 각각의 속성명을 결합시키기 위해 사용한다. 예를 들면 xmlns:dc="xmlns:dc="http://purl.org/metadata/dublin_core#" 라는 선언으로 〈dc:creator〉와 같이 사용할 수 있으며, 실제는 URL과 creator가 결합되어 완전한 참조를 생성한다고 할 수 있다. 즉 〈http://purl.org/metadata/dublin_core#creator〉와 같다. 따라서 Name space기능을 통한 접두어의 사용으로 좀 더 간결하고 정확한 기술이 가능하게 되었다.

Name Space의 선언은 XML구문에서 일괄 선언 방식과 기술요소에 포함시키는 방법이 사용될 수 있으며, 본 논문에서는 스키마설계에 일괄선언 방식을 적용하였다. 다음은 일괄 선언방식을 사용한 기술의 예이다.

```
〈rdf:RDF
    xmlns rdf = "http://www.w3c.org/1999/02/22-rdf-
        syntax-ns#"
    xmlns dc = "http://purl.org/metadata/dublin_core#"〉
    〈rdf:Description about= "http://purl.org/metadata
        /dublin_core_elements"〉
      〈dc:title〉Dublin Core Metadata Element Set:
        Reference Description〈/dc:title〉
      〈dc:creator〉
       . . .
    〈/rdf:Description〉
〈/rdf:RDF〉
```

Name Space 선언은 위의 예에서와 같이 처음에 일괄선언하고 이를 속성(기술요소)의 접두어로 사용하는 방법을 제공함은 물론 다음 2가지 예와 같이 개개의 기술요소에서 Name Space기능을 사용할 수 있다.

```
〈RDF xmlns= "http://www.w3.org/1999/02/22-rdf-
        syntax-ns#"〉
    〈description about = "http://www.w3.org/Home
        /Lassila"〉
      〈Creator xmlns= "http://description.org/ schema/"〉
        Ora Lassila〈/Creator〉
    〈/description〉
〈/RDF〉

〈RDF xmlns= "http://www.w3.org/1999/02/22-rdf-
        syntax-ns#"〉
    〈description about = "http://www.w3.org/Home
        /Lassila"〉
      〈Creator xmlns:dc= "http://description.org/
        schema/"〉 Ora Lassila〈/dc:creator〉
    〈/description〉
〈/RDF〉
```

결과적으로 Name space기능은 스키마정의를 참조(연결)하기 위한 기법으로서 특정 메타데이터에서 서로 다른 형식을 연동할 수 있는 보다 유연한 방법을 제공하고 있다고 할 수 있다. RDF구문은 스키마와 연계되지 않고는 정확한 기술요소의 사용 여부를 검증할 방법이 없게 된다고 할 수 있다. 그러나 Name Space기능으로 기술(記述)에 사용된 메타데이타 형식을 식별해 낼 수 있는 수단은 제공하지만 이 서로 다른 형식의 기술요소간의 연계성은 제공해 주지 못한다 할 수 있다.

2.3.2 연속구문과 축약구문

RDF는 Description을 중심으로 복수의 기본 기술문(Statements)들을 기술하는 모형으로서 이를 엔코딩(encoding)하기 위한 XML구문은 연속구문(Serialization Syntax)방식과 축약구문(Abbreviated Syntax)방식을 모두 지원한다. 연속구문방식은 상세구문방식이며 축약구문방식은 특정 기술요소 안에 모든 속성과 속성값을 기술하는 간략기술방식이라 할 수 있다.

간략기술방식은 3가지의 경우에 적용할 수 있

다. 첫째는 단일 Description내에 동일 속성을 갖는 속성값이 반복되지 않으면서 속성값이 문자열인 경우, 둘째로 RDF기본기술문(Statement)의 대상이 또 다른 자원이고 그 자원의 인라인 속성값들이 문자열인 경우, 셋째로 Description요소가 자원의 type속성을 포함하고 있을 경우 적용될 수 있다. 다음 예는 이 두가지 방식을 적용한 것이다.

연속구문

```
<rdf:Description ID="Resource">
    <rdf:type resource="#Class"/>
    <rdfs:label>Resource</rdfs:label>
    <rdfs:comment>The most general class</rdfs:
        comment>
</rdf:Description>
<rdf:Description about="http://www.w3.org/1999/02
        /22-rdf-syntax-ns#type">
    <rdf:type resource="http://www.w3.org/1999 /02
        /22-rdf-syntax-ns#Property"/>
    <rdfs:label>type</rdf:label>
<rdfs:comment>Indicates membership of a class</rdfs
        :comment>
    <rdfs:range rdf:resource="#Class"/>
</rdf:Description>
```

축약구문

```
<rdf:Class ID="Resource"
    <rdfs:label="Resource"
    <rdfs:comment="The most general class"/>
<rdf:Property about="http://www.w3.org/1999/02
        /22-rdf-syntax-ns#type" rdfs:label="type"
    rdfs:comment="Indicates membership of a class">
    <rdfs:range rdf:resource="#Class"/>
</rdf:Property>
```

위 예에서 보는 바와 같이 RDF스키마를 활용한 응용스키마의 설계시에는 해당 자원의 인스턴스를 생성하는 것이기 때문에 유형시스템(type속성)이 필수적으로 사용됨을 볼 수 있으며, 축약구문방식을 적용할 수 있다. 따라서 표현의 간결성을 위하여 본 논문에서는 스키마 설계시 type속성을 사용한 부분을 축약구문방식을 적용하였으며 연속구문방식을 보조적으로 사용하였다.

3 공통요소선별

메타데이터형식은 인터넷자료에 대한 정보를 기반으로 하며, 주제범주나 목적에 따라 기술요소와 구문이 서로 다르다. 이러한 다양한 형식의 난립은 인터넷이라는 개방환경에서의 형식 상호간 연계성에 저해요인이 되고 있으며, 이로 인해 모든 자료유형에 공통으로 적용될 수 있는 통합 메타데이터형식의 필요성이 증대되고 있다.

그러나 자료의 유형이나 수록 매체마다 기술되는 정보의 종류가 다르고, 형식에 따라 기술요소의 어의나 종류 및 그 사용범위 또한 각기 다르기 때문에 상이한 모든 형식을 수용할 수 있는 통합 메타데이터 형식을 규정하는 일이 가능하지 않을 뿐 아니라 현실성이 없다고 할 수 있다. 따라서 이에 대한 대안으로 각 형식을 그대로 인정하고, 형식 상호간 연계성을 확보하려는 시도가 진행되어 왔다.

이러한 형식간의 연계성을 확보할 수 있는 수단으로 기술형식간 연결테이블을 통하거나 모든 형식에 공통으로 적용될 수 있는 기술의 모형화와 그 구문을 통한 방법이 제시되었다. 그러나 첫 번째 방법은 기술요소간 1 : 1 대응의 문제로 인하여 연계성에 한계가 있으며, 두 번째 방법은 개개 메타데이터형식의 요소규정을 그대로 따른 통일적인 기술모형의 확보를 목적으로 하기 때문에 기술요소간 어의적인 연계성을 확보할 수 없는 문제가 있다.

따라서 상이한 형식 상호간 연계성을 확보하기 위한 방안으로 다양한 형식과 연계될 수 있는 중

개(仲介)적인 표준형식의 필요성이 제기되었다(Pierre and LaPlant 1998). 즉, 개개 형식을 중개형식과 연결테이블을 작성함으로써 결과적으로 모든 형식이 어의적으로 연계성을 갖도록 하는 방안이라 할 수 있다.

BIBLINK 프로젝트(Heery 1996, 44)에서는 더블린코어형식이 국제적인 합의의 산물임을 강조하고, 이 형식을 핵심기술을 위한 대안으로 인정하였다. 또한 Nordic(1998)에서도 인터넷자료 기술을 위한 효과적인 기술형식으로 평가한 바 있다.

Weiber와 Erway(1997)는 더블린코어형식이 MARC에 대한 탐색요소집단으로, 다른 형식의 메타데이터에 대한 창으로 사용될 수 있음을 제시하였고, 메타데이터 형식이 복잡하면 형식상호간 운용성은 급속히 저하되기 때문에 단순한 형식이 되어야 한다는 최소주의자(Minimalists)의 입장을 선택한 더블린코어형식(Weibel, and Hakala 1998)은 어의적인 연계성 확보를 위한 중개형식으로 사용될 수 있음을 의미한다 하겠다.

MARC는 상세한 기술요소를 포함한 형식이라 할 수 있다. 그러나 MARC의 모든 기술요소가 자료를 식별하는데 필수적으로 사용되어야 하는 것은 아니며, 이들 요소들을 적절하게 선별할 필요가 있다. 이러한 측면은 AACR2(ALA 1988) 규칙에서도 필요에 따라 기술수준을 선택할 수 있도록 하고 있으며, IFLA(1998)는 서지레코드의 데이터 모형화를 통하여 기본수준의 국가서지레코드를 구성하기 위하여 데이터요소를 선별하였다. 한편, LC(1996d)는 핵심기술요소의 선별의 필요성을 인식하고 자료 유형별 기술요소 표준을 제정하여, 이를 협동목록시스템인 PCC(Program for Cooperative Cataloging)과 CONSER(Cooperative ONline SERials)에 적용하였다(LC 1999b). Hyslop(1997)은 핵심수준목록에 관한 LC(1997f)의 조사를 기초로 상세수준과 큰 차이가 없는 것으로 평가하였으며, UCLA(1996)의 조사에서도 만족할 만한 것으로 평가하였다. 이러한 상세목록의 대체수단으로서의 핵심수준목록의 적합성에 대한 긍정적인 평가는 LC에서 이를 채택하는 결과로 이어졌다(LC 1997e). 이러한 결과는 자료의 식별성에 근거한 핵심데이터요소를 파악하는데 중요한 근거로 삼을 수 있으며, 동시에 공통요소를 선별할 수 있는 기준을 제공한다 할 수 있다.

결과적으로 더블린코어는 중개형식으로 사용될 수 있는 핵심기술요소로 구성된 단순 형식이기 때문에 더블린코어의 15개 기술요소 및 그 하위요소를 기초로 하고, 상세형식인 KORMARC의 데이터요소를 비교하여 공통요소를 선별하는 것은 의미있는 일이며, 대부분의 메타데이터 형식을 포괄할 수 있을 것이다. 〈표 1〉은 더블린코어요소와 KORMARC를 비교하여 얻어진 결과이다.

〈표 1〉에서 보는 바와 같이 내용적으로 볼 때 KORMARC의 대부분의 요소가 더블린코어 요소에 포함됨을 볼 수 있으며, 기타의 요소들만 차이가 있다. 따라서 기타 요소들을 포함한다면 거의 대부분의 정보를 망라한다고 볼 수 있다.

본 논문에서는 〈표 1〉에 제시된 정보를 기준으로 기술규칙에서 규정한 핵심기술수준 및 LC에서 제시한 핵심기술요소, 더블린코어 요소 등을 분석하여 공통요소로 사용될 요소를 집단화하였다. 〈표 2〉는 최종적으로 얻어진 결과이다.

〈표 2〉에 제시된 공통요소는 더블린코어요소를 기준으로 하였기 때문에 연결테이블 작성을 위한 중계형식으로서의 의미가 그대로 수용되었다고 할 수 있다. 따라서 이들 공통요소를 기준으로한 기술요소의 집단화 방식은 특정 형식과의 어의적인 연계성을 위한 기준요소들로 활용될 수 있을

〈표 1〉 더블린코어요소와 KORMARC필드 비교

DC요소	KORMARC	DC요소	KORMARC
Title	245$a(본표제) 130(통일표목) 210(축약표제) 222(등록표제) 240(통일표제) 245$x(대등표제) 245$b(부표제) 500$e(부서명) 500$f(판권기서명) 500$g(표지서명) 500$h(책등서명) 500$i(번역서명) 730(부출통일표제) 740(부출표제)	Relation Source	247(이전/변경표제) 440, 490(총서사항) 501(합철주기) 505(내용주기) 507(원서주기) 510(인용주기) 530(이용가능한 다른형태주기) 533(복제주기) 534(원본주기) 581(참조정보원주기) 760(상위총서저록) 762(하위총서저록) 765(원저저록) 767(번역저록) 775(이판저록) 776(기타형태저록) 780(선행저록) 785(후속저록)
Creator Contributor	100(기본표목-개인명) 110(기본표목-단체명) 111(기본표목-회의명) 245$d(저자) 245$e(두번째이하저자) 250$b(해당판저자) 500$d(저작자주기)-(연) 508(제작진주기)-(비) 511(연주자/배역진)-(비) 700(부출목표-개인명) 710(부출표목-단체명) 711(부출표목-회의명)	Subject	050$a(LCC) 056$a(KDC) 080(UDC) 082(DDC) 085(기타분류기호) 600(주제명부출-개인명) 610(주제명부출-단체명) 611(주제명부출-회의명) 630(주제명부출-통일표제) 650(주제명부출-일반주제) 651(주제명부출-지명) 653(비통제주제명)
Format	300(형태기술사항) 516$a(컴퓨터화일과데이터유형주기) 538$a(시스템 주기)	Type	Leader/06(레코드형태) 007/00(자료범주표시) 256(컴퓨터화일특성)
Publisher	260$b(발행자)	Date	260$c(발행년) 008/07-14
Coverage	043(지역부호) 045(연대부호) 651$y(시대세목) 651$a(지명)	Language	008/35-37(언어부호) 041$ (언어부호) 546$ (언어주기)
Description	520(요약, 초록, 해제주기) 500$a(일반주기)	Rights	020$
Identifier	010(LCCN) 015(NBN=국가서지번호) 020(ISBN0 017(저작권등록번호) 022(ISSN) 024 $a(ISRC) 024 $a(UPC) 024$a(ISMN) 024$a(EAN) 024 $a(SICI) 027 $a(STRN) 028(발행자번호) 030(코덴부호) 074(GPO 번호) 856(전자접근위치)-USMARC		
기타	검증기관부호(042) 주기정보(500$a, 502, 510, 533, 534, 546) 지도수치정보(034, 255) 권호표시정보(362) 판사항(250) 최근간행빈도(310)		

〈표 2〉 공통요소

상위요소	하위요소	상위요소	하위요소
표제		저자	
주제	분류기호 키워드	내용범위	취급된 년대범위 취급된 장소범위
발행일		식별요소	표준번호 자료접근주소
지도수치데이터		관련정보	HasPartClass, IsPartOfClass IsVersionOfClass, HasVersionClass IsFormatOfClass, HasFromatClass ReferencesClass, IsReferencedByClass IsBasedOnClass, IsBasisForClass RequiresClass, IsrequiredByClass IsSuccessionOfClass, IsSucceededByClass
유형		주기정보	
언어		이용조건	
개요/초록		발행자	
간행빈도		주소정보	

것이다.

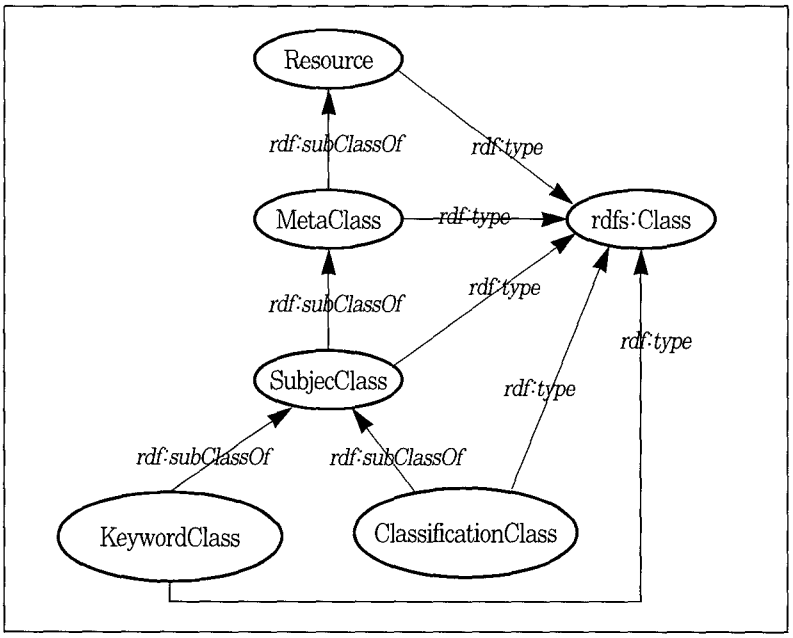
RDF구문은 기술요소를 메타데이터 스키마에 기초하기 때문에 어의적인 연계성 또한 스키마 연계를 통해 달성된다고 할 수 있다. 따라서 메타데이타 스키마는 어의적인 연계성의 기준을 포함해야 하며, 이를 위해 공통요소들은 공통스키마 설계에 반영하였다.

4 메타데이터 연계성을 위한 스키마 설계

스키마는 공통스키마와 KORMARC스키마를 각각 설계하였다. 공통스키마는 어의적으로 동일한 기술요소들을 집단화하기 위해 사용되는 것으로서 메타데이터 스키마 설계시 공통스키마의 해당 클래스를 rdfs:domain속성값으로 정의하였다. 스키마 설계는 RDF스키마명세(Brickley and Guha 1999), 구문명세(Lassila and Swick 1999)에 기반하였다. 다만, 스키마설계에 사용된 Name Space의 URI들은 공식적으로 주어지지 않았기 때문에 임의로 사용하였다.

4.1 공통스키마 설계

RDF모형과 구문은 각기 다른 구조와 구문을 갖는 메타데이터형식을 통일된 구문으로 기술하기 위한 구문적 운용성을 기반으로 하기 때문에 형식간 기술요소의 어의적 연계성을 제공하지 못하는 문제가 있다. 어의적 연계성은 색인정보 생성 등 메타데이터의 처리에 일관성을 확보할 수



〈그림 7〉 공통스키마의 SubjectClass 계층

있을 뿐만 아니라 특정 형식으로의 메타데이터 변환에 기초가 된다는 점에서 대단히 중요한 요인이라 할 수 있다. 따라서 메타데이터 스키마는 기술(記述)을 위한 스키마 참조뿐만 아니라 상이한 형식간 어의적 연계성을 제공해야 하며, 메타데이터를 RDF구문으로 변환하는데 변환대상요소기준으로 활용될 수 있어야 한다.

메타데이터 스키마의 기술속성들은 rdfs:domain속성을 사용하여 기술속성들이 속한 클래스를 제한할 수 있기 때문에 공통스키마에 정의된 클래스를 rdfs:domain속성값으로 갖는 기술요소들을 파악할 수 있다. 따라서 공통스키마와 연계하여 메타데이터 스키마를 설계할 경우 각 기술속성이 속한 클래스를 rdfs:domain속성값으로 갖게 되므로 기술속성은 공통 스키마의 클래스와 어의적 연계성을 갖는다고 할 수 있다.

RDF스키마에 기반한 스키마 설계는 모든 클래스가 rdf:type속성을 사용하여 인스턴스를 생성하기 때문에(rdf:type속성이 사용되면 축약구문방식이 가능함) XML의 연속구문방식과 축약구문방식이 가능하며, 본 논문에서는 축약구문방식과 연속구문방식을 혼용하였다. 〈그림 7〉은 본 연구에서 설계한 공통스키마의 SubjectClass클래스를 중심으로 다른 자원과의 관계 및 유형정의를 나타낸 것이다.

〈그림 7〉에서 보는 바와 같이 Resource의 하위 클래스로 MetaClass를, MataClass의 하위 클래스인 SubjectClass를, SubjectClass의 하위 클래스인 KeywordClass와 ClassificationClass를 각각 rdf:Class의 인스턴스로 생성하여 스키마를 정의한다. 다음은 〈그림 7〉의 내용을 기초로 설계한 스키마 예이다.

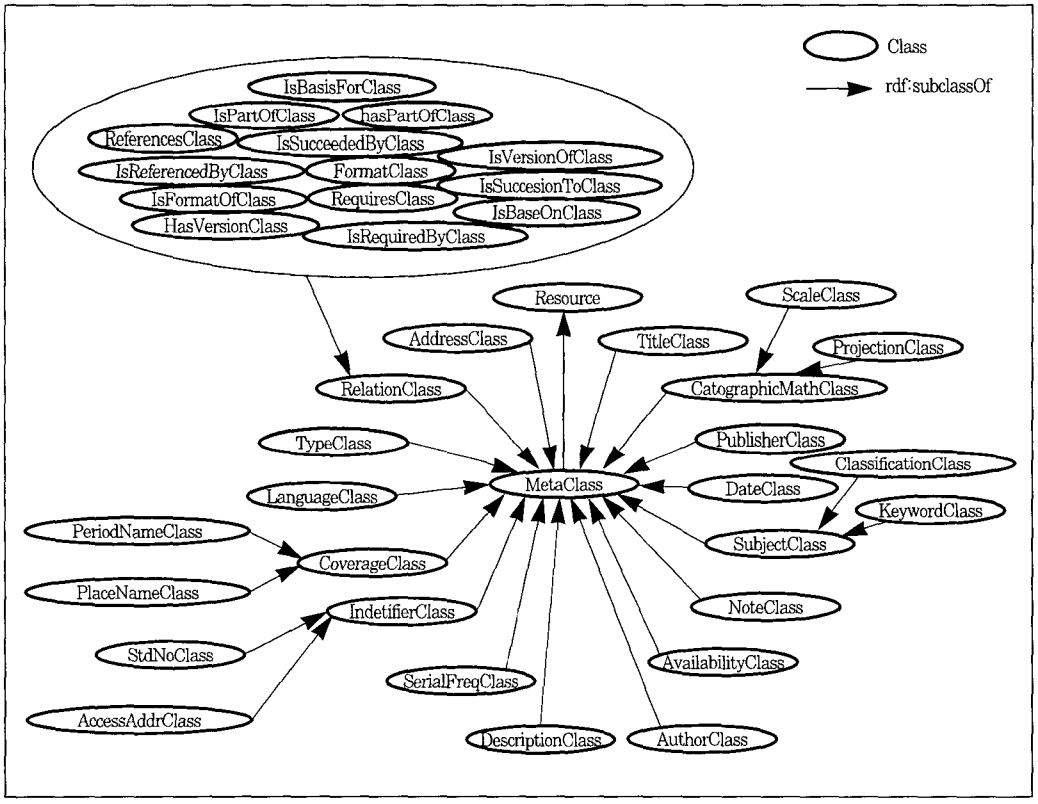
```
〈?xml version="1.0" encoding="KSC5601"〉
〈rdf:RDF xml:lang="ko"
    xmlns="http://www.w3.org/1999/02/22-rdf-
        syntax-ns#"
```

```
    xmlns="http://www.w3.org/TR/1999/PR-rdf-
        schema-19990303#">
    <rdfs:Class rdf:ID="MetaClass"
      rdfs:label="공통클래스"
      rdfs:subClassOf="http://www.w3.org/TR
        /1999/PR-rdf-schema-19990303#Resource"/>
     <rdfs:Class rdf:ID="SubjectClass"
      rdfs:label="주제클래스"
      rdfs:subClassOf="#MataClass"/>
    <rdfs:Class rdf:ID="ClassificationClass"
      rdfs:label="주제-분류번호 클래스"
      rdfs:subClassOf="#SubjectClass"/>
    <rdfs:Class rdf:ID="KeywordClass"
      rdfs:label="주제-키워드 클래스"
      rdfs:subClassOf="#SubjectClass"/>
</rdf:RDF>
```

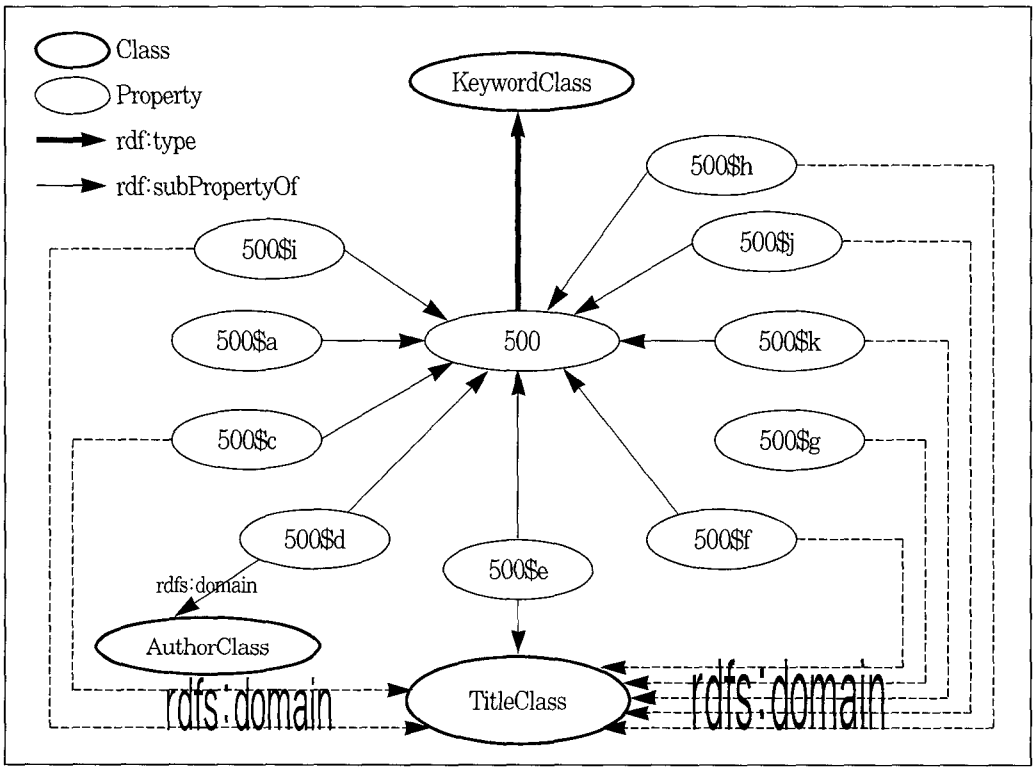
〈그림 8〉은 4장의 연구결과인 〈표 2〉에 제시된 공통요소를 기준으로 설계한 공통스키마의 클래스를 계층으로 나타낸 것으로서 모든 클래스들은 MetaClass의 하위 클래스로 설계한 내용을 보여주고 있으며, MataClass에 속한 각 클래스들은 어의적 연계성 확보를 목적으로 사용되는 클래스들이라 할 수 있다.

4.2 KORMARC스키마 설계

메타데이터 스키마 설계는 메타데이터형식에 정의된 모든 기술요소를 속성(Property)의 인스턴스로 생성해야 하며, 메타데이터 형식이 갖는 모든 특성과 구조가 그대로 반영되어야 한다. 동



〈그림 8〉 공통스키마의 클래스 계층



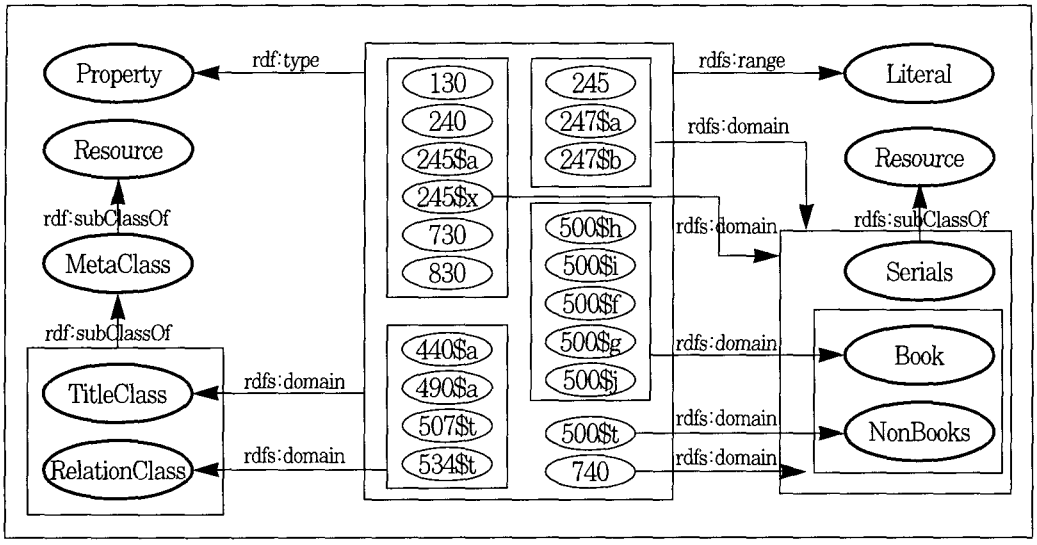
〈그림 9〉 KORMARC스키마의 속성 계층

시에 해당 형식과 스키마간의 어의적 연계성을 위하여 형식에 정의된 기술요소명이 스키마설계시 변경되지 않아야 한다.

그러나 KORMARC는 웹기반 기술형식으로 적합하지 않은 부분이 다수 포함되어 있기 때문에 모든 내용을 스키마에 반영하는 것은 바람직하지 않다고 할 수 있다. 따라서 부적합한 부분을 제외해야 할 필요가 있으며, 스키마설계시 RDF구문으로 변환되어야 할 정보를 선별하여 스키마설계에 반영해야 할 것이다. 이는 RDF구문이 메타데이터 스키마 연계구문이므로 기술요소의 선택은 스키마참조가 전제된다. 또한 메타데이터의 통합에 이를 활용한다면 메타데이터 스키마는 RDF구문으로 변환될 변환대상 기술요소리스트로 활용되어야 하기 때문이다.

KORMARC은 설계원칙에서 제시한 필드의 집단화방식이 각 필드에 직접적으로 반영되지 않았으므로 각각의 필드마다 독립적인 성격을 갖는다고 할 수 있다. 즉, 1xx의 하위필드로 100, 110, 111필드들이 존재하는 것이 아니며, 따라서 1xx 등의 내용을 상위속성으로 정의하는 것은 바람직하지 않다고 할 수 있다.

RDF구문에서 기술속성은 단일 속성이므로 하위필드의 집합으로 구성되는 KORMARC의 각 필드들을 각각 독립적인 기술요소로 정의하되, 특정 필드에 속한 하위필드들을 집단화하기 위해 각 필드들은 상위요소로 설계되어야 한다. 다만, 하위요소명이 $a, $b 등 기술속성을 정의할 경우



〈그림 10〉 KORMARC스키마의 제한관계

에는 그 고유성이 없기 때문에 필드명(표시기호)과 하위필드명(식별기호)을 결합하여 기술요소명을 부여할 필요가 있다. 〈그림 9〉는 본 논문에서 설계한 KORMARC스키마의 속성 계층 및 공통스키마의 클래스와의 제한관계를 500필드를 대상으로 나타낸 것이다.

기술요소(속성)들은 그들이 사용될 수 있는 클래스를 제한하기 위해 rdfs:domain속성을 사용하며, rdfs:range의 속성값으로 rdfs:Literal을 갖는다. 한편, KORMARC형식은 단행본용, 연속간행물용, 비도서용으로 별개로 구성되었으나 이를 통합하기 위하여 Books, NonBooks, Serials 3개의 클래스를 설계하였다. 따라서 모든 속성은 이들 중 하나 이상의 클래스와 공통스키마의 해당 클래스를 rdfs:domain속성값으로 갖게 된다.

〈그림 10〉은 본 논문에서 설계한 공통스키마의 TitleClass를 rdfs:domain속성값으로 갖는 KORMARC스키마의 기술속성 및 다른 자원과의 전체적인 관계를 나타낸 것이다. 〈표 3〉은 〈그림 10〉의 내용을 기반으로 스키마 설계 예를 제시한 것이다. 다만, 공통 스키마를 meta_schema로 하고, 그 URI는 가상적으로 사용하였다.

〈표 3〉의 스키마 설계 예에서 보는 바와 같이 모든 속성들은 rdfs:domain속성값으로 공통스키마의 TitleClass를 갖도록 설계하였다. 따라서 KORMARC스키마를 사용한 메타데이터에서 표제를 식별하기 위해서는 KORMARC스키마에서 TitleClass에 속하는 기술요소를 집단화함으로써 달성될 수 있다. 즉, TitleClass에 속하는 KORMARC의 모든 기술속성들은 표제를 의미하는 기술속성이라 할 수 있다.

이와 같은 관점에서 모든 형식의 스키마를 설계할 때 본 논문에서 설계한 공통 스키마의 해당 클래스를 rdfs:domain속성값으로 정의한다면 기술요소의 차이에서 오는 어의적인 문제를 극복할 수 있으며, 메타데이터의 변환이나 색인생성을 위한 기술요소의 식별에 일관성을 제공할 수 있다 하겠다.

〈표 3〉 스키마 설계 예

```
〈?xml version="1.0" encoding="KSC5601"〉
〈rdf:RDF xml:lang="ko"
        xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"〉

〈!—#################################245필드 ################################—〉

〈rdf:Property ID="245"〉
〈rdfs:label〉표제저자사항〈/rdf:label〉
〈rdfs:domain rdf:resource="Books"/〉
〈rdfs:domain rdf:resource="NonBooks"/〉
〈rdfs:domain rdf:resource="Serials"/〉
〈rdfs:range rdfs:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal"/〉
〈/rdf:Property〉

〈rdf:Property ID="245$a"/〉
〈rdfs:label〉본표제〈/rdf:label〉
〈rdfs:subPropertyOf rdf:resource="#245"/〉
〈rdfs:domain rdf:resource="http://hosim.kwangju.ac.kr/~leekyum/meta_schema#TitleClass"/〉
〈/rdf:Property〉

〈rdf:Property ID="245$b"/〉
〈rdfs:label〉부표제〈/rdf:label〉
〈rdfs:subPropertyOf rdf:resource="#245"/〉
〈rdfs:domain rdf:resource="http://hosim.kwangju.ac.kr/~leekyum/meta_schema#TitleClass"/〉
〈/rdf:Property〉

〈rdf:Property ID="245$c"/〉
〈rdfs:label〉잡제〈/rdf:label〉
〈rdfs:subPropertyOf rdf:resource="#245"/〉
〈rdfs:domain rdf:resource="http://hosim.kwangju.ac.kr/~leekyum/meta_schema#TitleClass"/〉
〈/rdf:Property〉

〈rdf:Property ID="245$x"〉
〈rdfs:label〉대등표제〈/rdf:label〉
〈rdfs:subPropertyOf rdf:resource="#245"/〉
〈rdfs:domain rdf:resource="http://hosim.kwangju.ac.kr/~leekyum/meta_schema#TitleClass/〉
〈/rdf:Property〉

〈!— ################################### 500필드 ################################### —〉
〈rdf:Property ID="500"〉
〈rdfs:label〉일반주기〈/rdf:label〉
〈rdfs:range rdfs:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal"/〉
```

〈표 3〉 스키마 설계 예(계속)

```
〈/rdf:Property〉

〈rdf:Property ID="500$c"〉
〈rdfs:label〉표제에 관한 주기〈/rdf:label〉
〈rdfs:subPropertyOf rdf:resource="#500"/〉
〈rdfs:domain rdf:resource="NonBooks"/〉
〈rdfs:domain rdf:resource="Serials"/〉
〈rdfs:domain rdf:resource="http://hosim.kwangju.ac.kr/~leekyum/meta_schema#TitleClass"/〉
〈/rdf:Property〉

〈rdf:Property ID="500$e"〉
〈rdfs:label〉일반주기-부표제〈/rdf:label〉
〈rdfs:subPropertyOf rdf:resource="#500"/〉
〈rdfs:domain rdf:resource="Books"/〉
〈rdfs:domain rdf:resource="http://hosim.kwangju.ac.kr/~leekyum/meta_schema#TitleClass"/〉
〈/rdf:Property〉

〈rdf:Property ID="500$f"〉
〈rdfs:label〉일반주기-판권기 표제〈/rdf:label〉
〈rdfs:subPropertyOf rdf:resource="#500"/〉
〈rdfs:domain rdf:resource="Books"/〉
〈rdfs:domain rdf:resource="http://hosim.kwangju.ac.kr/~leekyum/meta_schema#TitleClass"/〉
〈/rdf:Property〉

〈rdf:Property ID="500$g"〉
〈rdfs:label〉일반주기-표지표제〈/rdf:label〉
〈rdfs:subPropertyOf rdf:resource="#500"/〉
〈rdfs:domain rdf:resource="Books"/〉
〈rdfs:domain rdf:resource="http://hosim.kwangju.ac.kr/~leekyum/meta_schema#TitleClass"/〉
〈/rdf:Property〉

〈rdf:Property ID="500$h"〉
〈rdfs:label〉일반주기-책등표제〈/rdf:label〉
〈rdfs:subPropertyOf rdf:resource="#500"/〉
〈rdfs:domain rdf:resource="Books"/〉
〈rdfs:domain rdf:resource="http://hosim.kwangju.ac.kr/~leekyum/meta_schema#TitleClass"/〉
〈/rdf:Property〉

〈rdf:Property ID="500$i"〉
〈rdfs:label〉일반주기-번역표제〈/rdf:label〉
〈rdfs:subPropertyOf rdf:resource="#500"/〉
〈rdfs:domain rdf:resource="Books"/〉
〈rdfs:domain rdf:resource="http://hosim.kwangju.ac.kr/~leekyum/meta_schema#TitleClass"/〉
〈/rdf:Property〉
```

〈표 3〉 스키마 설계 예(계속)

```
〈rdf:Property ID="500$j"〉
〈rdfs:label〉일반주기-대등표제〈/rdf:label〉
〈rdfs:subPropertyOf rdf:resource="#500"/〉
〈rdfs:domain rdf:resource="Books"/〉
〈rdfs:domain rdf:resource="http://hosim.kwangju.ac.kr/~leekyum/meta_schema#TitleClass"/〉
〈/rdf:Property〉

〈rdf:Property ID="500$k"〉
〈rdfs:label〉일반주기-잡제〈/rdf:label〉
〈rdfs:subPropertyOf rdf:resource="#500"/〉
〈rdfs:domain rdf:resource="Books"/〉
〈rdfs:domain rdf:resource="http://hosim.kwangju.ac.kr/~leekyum/meta_schema#TitleClass"/〉
〈/rdf:Property〉
```

5 결 론

서로 다른 기술형식 상호간 어의적인 연계성을 확보하는 일은 메타데이터의 재사용성을 가능하게 할 뿐만 아니라 접근점 정보에 관한 일관성있는 식별에 대단히 중요한 요인이라고 할 수 있다. 그러나 연결테이블(Mapping Table ; Crosswalk)은 상이한 형식의 기술요소간 어의적인 연계성을 제공하는 수단이라 할 수 있으나 각 형식이 정의하고 있는 기술요소의 종류나 그 적용범위가 일치하지 않기 때문에 이를 메타데이터 변환에 활용하는 것은 한계가 있다고 할 수 있다.

RDF는 모든 형식이 갖는 기술요소를 그대로 사용하여 메타데이터를 기술할 수 있는 구문의 통일성을 목적으로 한 것이라 할 수 있다. 따라서 복수의 형식을 사용할 경우 기술요소의 어의식별은 개별적인 형식에 의존해야 하는 문제가 제기될 수 있으므로 기술요소간의 어의적인 연계성이 요구된다 하겠다.

RDF는 메타데이터의 기술에서 기술요소는 해당 형식의 스키마 참조를 기초로 하기 때문에 메타데이터 스키마는 기술에 있어 기술요소의 기반이 된다. 그러므로 RDF스키마에 기초한 응용스키마를 설계함으로써 형식상호간 어의적인 연계성을 확보할 수 있다.

본 논문은 더블린코어형식과 KORMARC형식을 분석하여 어의적인 연계성 요소로 사용될 수 있는 공통요소를 집단화하고, 이를 기반으로 공통스키마를 설계하였고, 이 공통스키마를 연동하여 KORMARC스키마 설계하였다. 이 연구에서 얻어진 결론은 다음과 같다.

첫째, RDF구문은 스키마참조를 기초한 구문이기 때문에 스키마는 기술(記述)에서 기술요소의 기반이 되지만, RDF스키마를 인스탄스화하여 응용스키마를 제공하면 상이한 형식간 어의적인 연계성을 확보할 수 있다.

둘째, 어의적인 연계성 요소를 집단화는 더블린코어요소를 기준으로 KORMARC필드를 비교연구하였다. 그 결과 16개 집단 33개 요소를 얻었다.

셋째, RDF스키마명세는 어의적 연계성을 위한 공통정보클래스를 제공하지 않기 때문에 이를 별

도로 정의해야할 필요가 있다. 따라서 RDF스키마명세와 구문명세를 활용하여 16개 집단 33개 요소를 클래스로 정의하여 공통스키마를 설계하였다. 또한 이들 공통스키마의 클래스들을 범주화하기 위해 최상위 클래스를 MetaClass로 명명하여 설계하였으므로 MetaClass에 속한 모든 클래스는 어의적인 연계성 확보를 위한 클래스집단이라 할 수 있다. 이 공통스키마는 각 형식의 스키마와 연계됨으로써 메타데이터 상호간 연계성을 확보할 수 있는 수단을 제공한다.

넷째, 메타데이타 스키마 설계에서 공통스키마의 활용을 보이기 위해 KORMARC를 표본으로 선택하여 그 스키마를 설계하였다. RDF스키마명세에서 rdfs:domain속성은 기술요소(속성)이 속한 클래스를 한정하기 위해 사용되기 때문에 공통스키마의 해당 클래스들을 rdfs:domain속성값으로 기술요소를 정의하였다. 따라서 공통스키마를 각 메타데이터스키마설계에 적용한다면 결과적으로 속성들이 속하는 공통클래스로 집단화할 수 있으므로 어의적인 연계성을 갖게 되어 기술요소의 어의식별에 일관성을 부여할 수 있다.

본 논문에서는 메타데이터형식 상호간 어의적 연계성을 위한 방안으로 더블린코어와 KORMARC형식만을 표본으로 분석 연구하여 실험적인 수준에서 공통요소를 얻은 것이다. 따라서 이에 대한 연구의 폭을 더욱 확대할 필요가 있어야 할 것이며, RDF스키마에 이를 반영하여 그 활용성을 높여야 할 필요가 있다.

참 고 문 헌

구자철. 1997. 『신나는 자바 프로그래밍』. 서울: 높이 깊이.

국립중앙도서관 편. 1993. 『한국문헌자동화목록형식: 단행본용』. 서울: 同도서관. KS C 5867

국립중앙도서관 편. 1995. 『한국문헌자동화목록형식: 연속간행물용』. 서울: 同도서관. KS C 5795.

국립중앙도서관 편. 1996. 『한국문헌자동화목록형식: 비도서자료용』. 서울: 同도서관. KS C 5969

ALA. 1988. *Anglo-American cataloguing rules.* 2nd ed. Chicago: American Library Association.

Berners-Lee, Tim. 1998. Why RDF model is different from the XML model. [cited 1999.4.6]. <http://www.w3.org/DesignIssues/RDF-XML.html>.

Bray, T., D. Hollander and A. Layman. 1999. Namespaces in XML. [cited 1999. 2.16] <http://www.w3.org/TR/ REC-xml-names>.

Brickley, D. and R. V. Guha. 1999. Resource description framework(RDF) Schema Specification. Proposed Recommendation (1999-03-03). [cited 2000.3.5]. <http://www.w3.org/TR/1999/PR-rdf-schema-19990303>.

Chang, W. W. 1998. A discussion of the relationship between RDF-Schema and UML. [cited 1999.4.19]. <http://www.w3.org/TR/1998/NOTE-drf-uml-19980804>.

Clayphan, R. 1997. BIBLINK - LB 4034 D3.1

Minimum Data Set. [cited 1999.3.29]. <http://hosted.ukoln.ac.uk/biblink/wp3/d3.1/>.

Dempsey, L. and Rachel Heery. 1998. "METADATA: a current view of practice and issues." *Journal of documentation*, 54(2): 145-172.

DESIRE. 1996. Resource description : initial recommendations for metadata formats. [cited 1999.3.11]. <http://www.ukoln.ac.uk/metadata/desire/recommendations/doc0001.htm>.

Dublin Core Metadata Initiative. 1997b. Coverage Element working draft 1997-09-30. [cited 1999.2.16]. <http://purl.org/DC/documents/working_drafts/wd-coverage-current.htm>.

Dublin Core Metadata Initiative. 1998b. Subelement Working Draft 1998-02-11. [cited 1999.2.16]. <http://purl.org/DC/documents/working_drafts/wd-subelement-current.htm> .

Guenther, Rebecca. 1997. Dublin Core Qualifiers/Substructure. [cited 1999.3.20]. <http://www.loc.gov/marc/dcqualif.html>.

Hakala, J. 1998. The Nordic metadata project. Finial report. [cited 1999.7.25]. <http://linnea.helsinki.fi/meta/nmfinal.htm>.

Heery, R. 1996. BIBLINK-LB 4034 D1.1 Metadata formats. [cited 1999.3.29]. <http://hosted.ukoln.ac.uk/biblink/wp1/d1.1/>.

Hyslop, C. F. 1997. "Highlights of the program for cooperative cataloging : the core record and consolidation of CONSER and PCC." *ALCTS Newsletter*, 8(4) supplement. [cited 1999.3.1]. <http://lcweb.loc.gov/catdir/pcc/hyslop2.html>.

Iannella, Renato. 1998. A Idiot's Guide to the Resource Description Framework. [cited 1999.3.15]. <http://www.dstc.edu.au/RDU/reports/RDF-Idiot/>.

IFLA. 1998. *Functional Requirements for Bibliographic Records*. Final report. Munchen : Saur. (UBCIM publications. New Series ; v. 19)

Knight, J. and Martin Hamilton. 1997. Dublin Core Qualifiers. [cited 1999.3.20]. <http://www.roads.lut.ac.uk/Metadata/DC-SubElements.html>.

Lambrecht, J. H. *Minimal Level Cataloging by National Bibliographic Agency*. Munchen ; London ; NY ; Paris : Saur, 1992. (UBCIM publications. New Series ; v. 8).

Lassila, Ora and Ralph R. Swick. 1999. Resource description framework(RDF) model and syntax specification. W3C Recommendationt. [cited 2000.3.5]. <http://www.w3.org/TR/1999/PR-rdf-syntax-19990222/>.

LC. 1996a. Core bibliographic record for books. [cited 1998.7.23]. <http://lcweb.loc.gov/catdir/pcc/corebook.html>.

LC. 1996b. Core bibliographic record for music and non-music sound recordings. [cited

1998.7.23]. <http://lcweb.loc.gov/catdir/pcc/coremusic.html>.

LC. 1996c. Core bibliographic record for printed and manuscript music. [cited 1998.7.23]. <http://lcweb.loc.gov/catdir/pcc/coremusmss.html>.

LC. 1996d. CONSER record requirement for Full, core, and minimal level records. [cited 1998.8.22]. <http://lcweb.loc.gov/acq/conser/recordreq.html>.

LC. 1997a. Core bibliographic record for graphic materials(PCC CBR-GR/Final). [cited 1999.3.3]. <http://lcweb.loc.gov/catdir/pcc/coregm.html>.

LC. 1997b. Core bibliographic record for moving Image Materials(PCC CBR-MI/Final). [cited 1999.3.3]. <http://lcweb.loc.gov/catdir/pcc/coremim.html>.

LC. 1997c. Dublin core/MARC/GILS crosswalk. [cited 1998.7.29]. <http://lcweb.loc.gov/marc/dccross.html>.

LC. 1997d. Program for cooperative cataloging(PCC) core bibliographic record for monographic computer files[1]. [cited 1999.3.1]. <http://lcweb.loc.gov/catdir/pcc/corecf.html>.

LC. 1997e. LC cataloging newsline : Online newsletter of cataloging directorate, Library of Congress. 5(7). [cited 1998.7.24]. <http://lcweb.loc.gov/catdir/lccn/lccn0507.html>.

LC. 1997f. LC cataloging newsline : Online newsletter of cataloging directorate, Library of Congress. 5(2). [cited 1998.7.23]. <http://lcweb.loc.gov/catdir/lccn/lccn0502.html>.

LC. 1999a. Core bibliographic record for rare books(DCRB core). [cited 1999.3.2] <http://lcweb.loc.gov/catdir/pcc/dcrbcore.html>.

LC. 1999b. The program for cooperative Cataloging. [cited 1999.3.1]. <http://lcweb.loc.gov/catdir/pcc/pccinfo.html>

Miller, Eric. 1997. Monticello Electronic Library: Dublin Core Element Set Crosswalk. [cited 1998.7.29]. <http://www.oclc.org:5046/~emiller/DC/crosswalk.html>.

Olson, Nancy B. 1997. Cataloging internet resources : a manual and practical guide. 2nd ed. OH : OCLC Online Computer Library Center. [cited 1999.5.25]. <http://www.oclc.org/oclc/man/9256cat/toc.htm>.

Pierre M. S. and W. P. LaPlant. 1998. Issues in crosswalking content metadata standards. [cited 1999.6.9]. <http://www.niso.org/crsswalk.html>.

Research Libraries Group. 1997. Guidelines for extending the use of Dublin Core elements to create a generic application integrating all kinds of information resources. [cited 1998.8.6]. <http://www.rlg.org/metawg.html>.

Richard, L. 1998. 『XML: 차세대웹의 혁명』. 채규혁 역. 서울 : 민음사.

UCLA/OCLC. 1996. "UCLA/OCLC core record pilot project : preliminary report".

Library Resources & Technical Services, 40(3): 251-260.

W3C. 1999. RDF Schema. <http://xml.com/xml/pub/r/RDF_Schema> [cited 1999.4.5].

Weibel, S. 1995. "Metadata: The foundations of Resource Description". *D-Lib magazine*(1995. 7). [cited 1999.3.13]. <http://www.dlib.org/dlib/July95/07weibel.html>.

Weibel, S., and Juha Hakala. 1998. "DC-5 : The Helsinki metadata workshop - A report on the workshop and subsequent developments". *D-Lib magazine*(1998.2). [cited 1998.2.27]. <http://www.dlib.org/dlib/february98/02weibel.html>.

Weibel, S. and Ricky Erway. 1997. "The news talks with metadata experts Weibel and Erway". *RLG News issue*, 44(1997. Fall). [cited 1999.2.24]. <http://www.rlg.org/rlgnews/news44.html>.

Weibel, S., Renato Lannella, and Warwick Cathro. 1997. "The 4th Dublin Core metadata workshop". *D-Lib magazine* (1997. 3). [cited 1998.7.27]. <http://www.dlib.org/dlib/june97/metadata/06weibel.html>.