

# 메타데이터변환과 자원기술구조의 연구

## A Study on KORMARC Mapping to Dublin Core and Resource Description Framework for Encoding and Exchange of Metadata

김 태 수(Tae-Soo Kim)\*

### 목 차

- |                        |                       |
|------------------------|-----------------------|
| 1. 서 론                 | 3.2 메타데이터와 서지레코드간의 관계 |
| 2. 더블린 코어의 데이터요소       | 4. 메타데이터간의 연결구조       |
| 2.1 데이터 요소             | 4.1 자원과 메타데이터간의 연결    |
| 2.2 한정어                | 4.2 RDF 데이터모형         |
| 3. KORMARC 데이터의 DC 변환  | 4.3 RDF 구문            |
| 3.1 KORMARC과 DC의 대응 필드 | 5. 결 론                |

### 초 록

이 연구에서는 KORMARC 데이터와 더블린코어 메타데이터간의 데이터 변환에 따른 문제를 파악하고, 서지정보원으로서 메타데이터의 사용가능 여부를 평가하였다. 아울러 상이한 형식의 메타데이터를 처리하기 위한 자원기술구조의 모형을 분석하고, 메타데이터의 기술과 교환을 위한 공식 구문으로 XML의 가능성을 검토하였다.

### ABSTRACT

This study is to investigate the problems in mapping between MARC data and Dublin core metadata, and to evaluate metadata as a source of cataloging. In addition resource description framework and its XML syntax for processing metadata in different format were analyzed in the view from encoding and transporting metadata in the network environments.

키워드 : 메타데이터, 자원기술구조, 데이터변환, 기계가독목록, XML

\* 연세대학교 문헌정보학과 교수

■ 논문접수일 : 1998년 12월 2일

## 1. 서 론

네트워크자료를 포함한 각종 유형의 자료를 기술하고 접근하기 위한 다양한 시도가 일고 있다. 그 중의 하나는 모든 유형의 자료를 MARC형식으로 통합하여 기술하고, 이를 통해 접근하는 방식이다. 그것은 MARC이 모든 유형의 자료를 수용할 수 있고, 여기에 목록규칙이나 전거제어기법을 통해 기술의 일관성과 표준화가 가능하다는 점에 근거한 것이다(ALA. Committee on Cataloging 1988). 따라서 각종 형식의 메타데이터를 MARC형식으로 변환하기 위한 연구에 많은 관심을 기울이고 있다.

다른 하나는 MARC을 비롯한 다양한 유형의 메타데이터를 인정하는 관점이다. 그 주된 이유는 적용분야나 이용자의 수준에 따라 요구데이터나 정보가 다양하고, 어떤 단일형식의 메타데이터도 이러한 조건을 충족시킬 수 없기 때문이다. 예컨대 MARC은 그 구조의 경직성으로 인해 데이터구축에 많은 시간과 노력이 소요되고 따라서 간략기술을 위한 논의가 계속되어 왔다(Lambrecht 1992). 즉 MARC에서는 서지정보와 특정 도서관의 관리정보가 혼재되어 있을 뿐만 아니라 기술부와 접근부를 동시에 수용하고 있고, 변화되는 환경을 수용하기 어렵고, 특정 레코드 내에서 동일 정보가 중복 기술되어 시스템의 성능을 저하시키는 요인으로 지적되어 왔다. 따라서 현재로서는 이러한 상이한 관점에 대한 합의가 없고, 더욱이 메타데이터에서는 대상자원을 웹 자원으로 제한하고 있어, 실질적으로 수용 자료에 제한이 따른다. 따라서 이 방식에서는 데이터포맷의 통합보다는 자원에 대한 접근성을 확보하기 위한 포맷간의 연결구조의 개발에 많은 관심을 기울이고 있다.

그런데 최근에 와서 인쇄형태를 포함한 각종 유형의 자료를 웹 환경에서 한 지점에서 접근하기 위한 방안이 제기되고 있다(EULER 1997).

이 연구는 메타데이터 구축 환경의 변화과정을 분석하기 위한 것으로, 서지데이터와 메타데이터 간의 데이터 변환과 이에 따른 문제를 검토하여 서지데이터의 정보원으로서 메타데이터의 이용 가능성을 제시하고, 다양한 형식의 메타데이터로 표현된 자원에 접근하기 위한 자원기술구조와 공식 구문으로 제안된 XML의 가능성을 제시하였다.

## 2. 더블린 코어의 데이터요소

웹 자원을 기술하고 접근하기 위한 주된 메타데이터 형식으로 더블린코어(Dublin core elements set: DC)가 사용되고 있는데(DESIRE 1996; Nordic metadata project 1998; 첨단기술정보센터 1997), 그 이유는 레코드구조와 색인이 단순하고, 이용이 용이하며, 상호운용성을 확보할 수 있고, 자원의 접근성이 높다는 점을 들고 있다(Hansen 1998).

이 DC는 저자나 배포자가 일정 형식의 저작도구를 사용하여 웹 자원을 직접 기술할 수 있는 단순구조를 중요한 특징으로 하며, 수록 데이터요소에 대해서도 어느 정도 국제적인 합의가 이루어지고 있다. DC에서는 15개의 데이터요소를 규정하고, 필요에 따라 이들 데이터요소를 확장할 수 있어, 이론적으로 보면 목록규칙에서 규정한 요소를 메타데이터로 거의 완전하게 표현할 수 있다(김태수 1998).

## 2.1 데이터 요소

원칙적으로 모든 데이터요소는 선택사항이며, 반복 사용이 가능하다.

### 2.1.1 요소명: Title

#### 1) 한정어(하위요소)

DC.Title

DC.Title.Alternative

### 2.1.2 요소명: Creator

#### 1) 한정어(하위요소)

DC.Creator.PersonalName

DC.Creator.CorporateName

DC.Creator.PersonalName.Address

DC.Creator.CorporateName.Address

#### 2) scheme

LCNAF (Library of Congress Name Authority File)

### 2.1.3 요소명: Subject

#### 1) scheme

LCSH

KSH (한글주제명표)

MeSH

LCNAF (Library of Congress Name Authority File)

DDC (Dewey Decimal Classification)

KDC (Korean Decimal Classification)

LCC (Library of Congress Classification)

UDC (Universal Decimal Classification)

### 2.1.4 요소명: Description

#### 1) scheme

## URI

### 2.1.5 요소명: Publisher

#### 1) 한정어(하위요소)

DC.Publisher.PersonalName

DC.Publisher.CorporateName

DC.Publisher.PersonalName.Address

DC.Publisher.CorporateName.Address

### 2.1.6 요소명: Contributor

#### 1) 한정어(하위요소)

DC.Contributor.PersonalName

DC.Contributor.CorporateName

DC.Contributor.PersonalName.Address

DC.Contributor.CorporateName.Address

#### 2) scheme

LCNAF (Library of Congress Name Authority File)

### 2.1.7 요소명: Date

#### 1) 한정어(하위요소)

DC.Date.Created

DC.Date.Issued

DC.Date.Available

DC.Date.Valid

DC.Date.Acquired

DC.Date.Accepted

DC.Date.DataGathered

DC.Date.Modified

#### 2) scheme

ISO 8601

### 2.1.8 요소명: Type

#### 1) 한정어(하위요소)

DC.Type.Item

DC.Type.Collection

### 2.1.9 요소명: Format

#### 1) scheme

IMT: Internet Media Types(MIME)

Physical: 구체적인 표현매체

### 2.1.10 요소명: Identifier

#### 1) scheme

URI

ISBN

ISSN

### 2.1.11 요소명: Source

#### 1) 한정어(하위요소)

DC.Source.Creator

DC.Source.Publisher

DC.Source.Contributor

DC.Source.Rights

DC.Source.Title

DC.Source.Subject

DC.Source.Description

DC.Source.Language

DC.Source.Relation

DC.Source.Coverage

DC.Source.Date

DC.Source.Type

DC.Source.Format

DC.Source.Identifier

#### 2) scheme

URI

ISBN

ISSN

### 2.1.12 요소명: Language

#### 1) scheme

ISO639-2

### 2.1.13 요소명: Relation

#### 1) 한정어(하위요소)

DC.Relation.IsVersionOf

DC.Relation.HasVersion

DC.Relation.IsBasedOn

DC.Relation.IsBasisFor

DC.Relation.IsPartOf

DC.Relation.HasPart

DC.Relation.IsFormatOf

DC.Relation.HasFormat

DC.Relation.References

DC.Relation.IsReferencedBy

DC.Relation.Requires

DC.Relation.IsRequiredBy

#### 2) scheme

URI

ISBN

ISSN

### 2.1.14 요소명: Coverage

#### 1) 한정어(하위요소)

DC.Coverage.PlaceName

DC.Coverage.PeriodName

#### 2) scheme

ISO 8601

LCSH

### 2.1.15 요소명: Rights

#### 1) scheme

URI

2.2 한정어

DC에서는 자원의 접근에 필요한 기본요소 이외에 데이터요소의 값을 해석하거나 의미를 구체적으로 제한하기 위해 한정어(qualifier)를 사용한다. 그런데 한정어를 사용하게 되면 기술수준이 상이하어, 상호운용성에 제한을 줄 수 있다. 현재 사용되는 한정어는 다음과 같다.

2.2.1 언어

이 한정어는 자원의 본문언어가 아니라 데이터요소인 주제(Subject)나 표제(Title), 기술(Description)에 사용된 언어를 제시하는 기능을 한다. 이를 통해 상이한 언어로 표현된 자원의 소재를 확인하는 웹 색인의 탐색능력을 확장할 수 있다.

2.2.2 스킴

스킴(scheme)은 LCSH나 KDC 등과 같은 주제명표나 분류표, 각종 표준(예: ISO-8601 등)을 제시한다. 이를 통해 CONTENT에 수록된 값을 해석할 수 있으며, 레코드의 일관성을 유지하고 일정 부분 표준화를 달성할 수 있다.

2.2.3 하위요소

특정 데이터요소가 하위요소를 지닐 때 이 하위요소(type)를 제시하기 위한 것이다. 예컨대 creator 요소가 "name", "address" "telephone" 과 같은 하위요소를 지닐 때, 이를 구분하기 위한 것이다.

3. KORMARC 데이터의 DC 변환

메타데이터는 기술수준에서 보면 MARC데이터에 비해 간략한데 그 주된 이유는 메타데이터가 지닌 단순구조 때문이다. 앞으로의 기술발전을 고려할 때, MARC환경보다는 네트워크 환경에서 운용이 가능한 메타데이터로 통합되는 것이 효과적이라는 관점에서, MARC데이터를 DC데이터로 변환하는 과정을 제시하였다. 다음은 MARC 데이터와 이에 대응되는 DC 데이터를 제시한 것이다(Library of Congress 1997; Miller 1997; Thorborg 1997; Attig 1998).

3.1 KORMARC과 DC의 대응 필드

1) Title

- KORMARC : 245\$a (본표제)  
DC.Title
- : 245\$b (부표제)  
DC.Title.alternative
- : 245\$x (대등표제)  
DC.Title.alternative
- : 240\$a (통일표제)  
DC.Title.alternative
- : 440\$a (총서표제)  
DC.Title.alternative
- : 500\$j (대등표제)  
DC.Title.alternative
- : 500\$i (번역표제)  
DC.Title.alternative
- : 740\$a (본표제와 다른 표제)  
DC.Title.alternative

현재 DC에서는 MARC의 본표제 이외의 표제

는 DC.Title.alternative로 사용하며, KORMARC의 245\$b(부표제), 245\$x(대등표제), 240\$a(통일표제), 440\$a(총서표제)와 직접 대응되는 DC 필드가 없다. 따라서 이들 데이터를 변환하고자 하는 경우 DC.Title.Subtitle과 DC.Title.Parallel, DC.Title.Series와 같은 필드를 고려할 수 있다.

### 2) Creator

MARC 레코드의 1XX필드의 \$a와 250필드의 \$b, 700필드(개인명)와 710필드(단체명)의 \$a를 대상으로 한다. 다만 7XX필드에서 저자 이외의 역할을 한 인물이나 단체, 그리고 245필드의 \$d와 \$e, 주기필드(5XX)에 수록된 관련저자는 변환대상에서 제외한다.

KORMARC : 100\$a  
 DC.Creator.PersonalName  
 : 110\$a  
 DC.Creator.CorporateName  
 : 111\$a  
 DC.Creator.CorporateName  
 : 250\$b  
 DC.Creator.PersonalName  
 : 700\$a  
 DC.Creator.PersonalName  
 : 710\$a  
 DC.Creator.CorporateName

원칙적으로 KORMARC의 1XX는 Creator와 대응되고, 7XX필드는 Contributor와 대응된다.

### 3) Subject

주제는 분류기호와 주제명(또는 키워드)으로 표현되며, 다음 필드와 대응된다.

KORMARC : 056\$a  
 DC.Subject.scheme=KDC  
 : 080\$a  
 DC.Subject.scheme=UDC  
 : 082\$a  
 DC.Subject.scheme=DDC  
 : 611\$a  
 DC.Subject  
 : 630\$a  
 DC.Subject.  
 : 650\$a  
 DC.Subject.scheme=KSH  
 : 653\$a (비통제주제명)  
 DC.Subject

그런데 KORMARC의 600필드(주제명—인명)와 610필드(주제명—단체명)에 대응되는 DC 요소는 없다. 아울러 MARC의 6XX필드에서는 \$a로부터 \$2까지 모든 하위필드를 고려해야 하는 문제가 있다. 따라서 600과 610필드는 변환에서 제외한다.

### 4) Description

자원의 내용이나 목적을 문장형식으로 간략하게 기술한 요소로서, 다음의 필드를 대상으로 한다.

KORMARC: 250\$a (판표시)  
 DC.Description.Edition  
 : 520\$a (요약)  
 DC.Description

### 5) Publisher

KORMARC: 260\$b DC.Publisher

6) Contributor  
 KORMARC : 700\$a  
     DC.Contributor.PersonalName  
         : 710\$a  
     DC.Contributor.CorporateName  
 이들 필드의 \$a에 포함된 인물이나 단체를 변환의 대상으로 한다. 이 경우 245\$b는 변환의 대상에서 제외한다.

7) Date  
 KORMARC의 008/07-10에 연도만이 수록되고, 이에 대응되는 DC.Date는 디폴트 값으로 'YYYY-MM-DD' ("1998-09-23")를 가지므로, 완전한 변환이 어렵다.  
 KORMARC: 008/07-10  
     DC.Date

8) Type  
 KORMARC: 502\$a (학위논문주기)  
     DC.Type  
 KORMARC: 516\$a (컴퓨터파일주기)  
     DC.Type

9) Format  
 KORMARC:538\$a (시스템사항에 관한 주기)  
     DC.Format.scheme=IMT  
         300 (형태사항에 관한 주기)  
     DC.Format.scheme=Physical

10) Identifier  
 KORMARC: 020\$a (ISBN)  
     DC.Identifier.scheme=ISBN  
         022\$a (ISSN)  
     DC.Identifier.scheme=ISSN

024\$a (기타 표준부호)  
 DC.Identifier

가) KORMARC의 다음 필드와 대응되는 DC 필드는 필요에 따라 다음과 같이 설정할 수 있다.

015\$a (국가부호)와 \$b (서지번호)  
     DC.Identifier.scheme=NBN  
 024\$a (기타 표준부호)  
     DC.Identifier.scheme=ISRC  
 027\$a (표준기술보고서번호)  
     DC.Identifier.scheme=STRN

나) NBN, ISBN, ISRC, ISMN, STRN 데이터가 메타데이터에서 반복되는 경우, 첫 번째 이후에 출현한 모든 번호는 무시한다.

다) USMARC의 데이터를 그대로 사용하는 경우, USMARC의 856필드는 다음과 같이 대응된다.

856\$g (URN)  
     DC.Identifier.scheme=URN  
 856\$u (URL)  
     DC.Identifier.scheme=URL  
 856\$b (접근번호)  
     DC.Identifier.IP address

11) Source  
 KORMARC: 507\$a (원저작자)  
     DC.Source.Creator  
         507\$t (원본의 표제)  
     DC.Source.Title  
         507\$z (원본의 ISBN)  
     DC.Source.scheme=ISBN  
         507\$x (원본의 ISSN)

DC.Source.scheme=ISSN

현재 KORMARC에서는 786필드가 없으나 USMARC의 786\$a (데이터정보원)를 적용한다. 대응되는 DC 필드는 다음과 같다.

786\$a (데이터정보원)

DC.Source.scheme=url

### 12) Language

자원의 내용을 표현한 언어로서, 컴퓨터 언어 (C++, Java, Pascal 등)는 대상에서 제외한다.

KORMARC: 041\$a (언어부호)

DC.Language

008/35-37

DC.Language.scheme=ISO639-2

041필드를 사용한 경우, 008/35-37은 변환하지 않으며, 041\$a에서 3자씩 끊어 변환한다.

### 13) Relation

KORMARC: 760\$t (상위총서)

DC.Relation.HasPart

762\$t (하위총서)

DC.Relation.IsPartOf

765\$t (원저저록)

DC.Relation.HasVersion

767\$t (번역저록)

DC.Relation.IsVersionOf

773\$t (기본자료저록)

DC.Relation.IsPartOf

775\$t (이판저록)

DC.Relation.HasVersion

776\$t (기타형태저록)

DC.Relation.HasFormat

아울러 KORMARC의 다음 필드와 대응되는 DC필드를 설정한다.

770\$a (보유저록)

DC.Relation.IsChildOf

772\$a (모체저록)

DC.Relation.HasChild

### 14) Coverage

KORMARC: 651\$a (지명)

DC.Coverage.PlaceName

045\$a (연대부호)

DC.Coverage.PeriodName

연대부호를 해당 연도로 바꾸어준 다음 변환한다.

043\$a (지역부호)

DC.Coverage.PlaceName

지역부호를 해당 지명으로 바꾸어준 다음 변환한다.

### 15) Rights

이 필드와 대응되는 KORMARC 필드는 현재 설정되어 있지 않고 있으나, USMARC에서는 856\$u에서 규정되고 있다.

856\$u

DC.Rights

## 3.2 메타데이터와 서지레코드간의 관계

서지레코드는 목록에 수록된 자료와 관련된 데이터의 집합으로서, 자료의 기술과 검색을 위한 수준 높은 도구이다. 즉 제어기법을 적용하여 탐색기능을 지원하며, 자료에 기재된 정보를 이용하여 자료를 식별하며, 판사항을 통해 자원의 선정을 지원한다. 서지레코드에 포함되는 데이터 요소는 다음과 같다(IFLA. Study Group on the



Functional Requirements for Bibliographic Records 1998, 7).

- 1) 국제표준서지기술에서 규정한 기술데이터
- 2) 색인도구인 저자나 단체, 서명, 주제명
- 3) 분류기호나 초록, 요약과 같은 레코드파일의 조직에 관련된 데이터
- 4) 청구기호나 접근기호와 같은 장서 중의 특정 자료의 선정에 관한 데이터

그런데 기본적으로 서지데이터베이스의 정확성과 일관성은 수록된 모든 레코드에 동일한 표준이 적용되느냐에 따라 결정된다. 이 기준에 미달하게 되면 수록된 데이터를 신뢰하기 어렵다. 이에 따라 미국도서관협회(ALA)에서는 목록규칙을 적용하지 않은 메타데이터를 서지데이터베이스에 포함하는 것은 서지데이터베이스의 질에 손상을 줄 수 있고, 따라서 특정 메타데이터형식에 따라 작성된 레코드는 별도의 데이터베이스로 유지하는 것이 바람직하다고 결론짓고 있다. 미국도서관협회에서 서지데이터베이스와 메타데이터의 관계를 규정한 내용은 다음과 같다(ALA, Committee on Cataloging 1988).

- 1) 메타데이터는 목록의 대체수단으로 개발된 것은 아니지만 목록작성자에게 유용한 정보원이 될 수 있다.
- 2) 메타데이터에서는 자원간의 관계나 동일 자원의 상이한 판(version)을 식별하는데 필요한 충분한 정보를 수용하지 않고 있다. 메타데이터는 서지레코드의 대체수단으로 개발된 것이 아니며, 특히 DC는 자원의 기술이 목적이 아니라 접근을 목적으로 개발된 것이다. 따라서 목록과 같이 일정한 정보원을 전제로 하지 않고, 자원 자체와 직접 연결되며, 식별요소를 데이터로 포함하지 않는다.

3) 메타데이터에서는 전거파일이나 전거제어기법을 적용하지 않기 때문에 메타데이터를 도서관 목록으로 사용하는 것은 부적절하다.

4) 메타데이터 제작자들은 데이터의 의미나 내용을 규정하는 대신, 변환구문을 통해 상이한 메타데이터간의 데이터를 변환하는데 관심을 두고 있다. 이에 따라 DC 데이터를 MARC으로 변환할 수 있지만 일관성이나 표준화라는 점에서, 목록규칙에서 규정하고 있는 조건에는 부적절하다.

5) DC와 같은 메타데이터요소는 데이터의 등록과 유지관리를 위해 책임 있는 기관의 지원을 필요로 한다.

비록 DC는 웹 자원을 탐색하고 입수하기 위한 것이지만 서지데이터의 기능을 부분적으로 수행하고 있다. 특히 목록데이터는 메타데이터와 달리, 제어어휘의 사용과 전거제어를 통해 탐색기능을 확장하고 있으며, 정보원에 포함된 데이터를 그대로 사용하며, 판사항을 통해 자료의 식별과 선정기능을 지원하고 있다. 결국 DC에서는 특정한 기술규칙을 적용하지 않기 때문에 MARC의 서지데이터와 비교해 보면 데이터가 불완전하고 신뢰성이 낮다는 한계를 지니고 있다.

#### 4. 메타데이터간의 연결구조

MARC이 서지데이터의 교환용 형식이듯이 메타데이터를 교환할 수 있는 공동의 구문이 요구된다. 이를 위해 웹 컨소시엄(W3C)에서는 자원기술구조(Resource Description Framework: RDF)를 제안하고 있다(W3C RDF Schema Working Group 1998). 이 구조는 메타데이터를 교환하기 위한 하부구조로서, 데이터의 의미와

구문, 구조의 통일을 통해 메타데이터의 상호운용성(interoperability)을 확보하기 위한 것이다. 상호운용성이란 말은 상이한 메타데이터를 식별할 수 있는 시스템의 능력을 의미하는 말이다 (ALA. Committee on Cataloging: Description and Access. 1988).

RDF는 특정 분야에서 개개의 자원을 기술하는데 적용되는 의미를 규정한 것이 아니라, 적용 분야의 필요에 따라 메타데이터 요소를 정의하기 위한 것이다. 특정 메타데이터에 의해서 정의된 속성유형의 집합, 즉 데이터요소의 표준화를 통해 여러 분야에서 동일한 의미로 사용하거나 확장할 수 있다. RDF는 분산된 속성을 통합할 수 있는 구조로서, 해당 분야에서 어휘를 선언하고 이를 필요에 따라 재정의하거나 재사용, 확장할 수 있다.

#### 4.1 자원과 메타데이터간의 연결

웹에서는 하이퍼링크를 통해 메타데이터나 자원을 탐색하고 색인한다. 이것은 웹이 다음과 같

은 3가지 기본 개념에 기초하고 있기 때문이며, 이 중에서 HTML이 하이퍼텍스트문서 포맷에서 중요한 역할을 한다(Raggett 1997).

- 1) 자원을 포괄적으로 지칭하는 체계인 URL
- 2) 명명된 자원에 접근하기 위한 프로토콜인 HTTP
- 3) 다른 자원과의 연결기능을 하는 HTML (hypertext markup language)

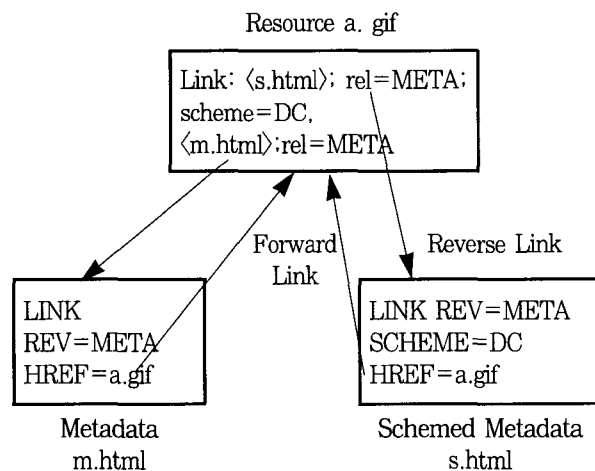
##### 4.1.1 연결구조

일반적으로 웹 환경에서 자원과 메타데이터는 다음과 같은 구조로 연결된다.

HTML에서는 특정 자원에서 대상자원으로 연결되는 구조로서, 이 대상자원(종착점)을 하이퍼텍스트에서 참조(anchor)라고 하고, 이 요소의 태그(tag)로 'a'를 사용한다.

```
<a href="http://www.w3.org/W3C/">
```

연결에서는 소스참조와 목표참조를 가지는데, HREF속성에서 URL을 사용하여 목표참조를 표현한다. 시작태그와 종료태그 사이의 값은 연결



<그림 1> 자원과 메타데이터간의 연결

의 label로 기능한다.

`<a href=중착점> link label </a>`

#### 4.1.2 자원과 메타데이터의 연결방법

HTML 자원에서 연결방법은 다음과 같다.

1) 자원에서 메타데이터로 연결

가) 자원의 헤드에서 관련된 메타데이터의 URI를 지시한다.

`<LINK REL=META HREF="/ml/ch4.1.2.H.html">`

`<LINK SCHEME=FGDC REL=META HREF="http://vancouver-webpages.com/ml/ch4.1.2.sgml">`

나) HTML 본문의 자원으로부터 메타데이터로 연결한다.

`<"A" *("SCHEME" "=" scheme) "REL" "=" "META" "HREF" "=" URI "> label </A">`

2) 메타데이터에서 자원으로 연결

가) 자원의 헤드에서 관련된 자원의 URI를 지시한다.

`<LINK REV=META SCHEME=DC HREF="ch4.1.2.gif">`

나) HTML 본문에서 메타데이터로부터 자원으로 연결

`<A REV=META HREF="http://vancouver-webpages.com/ml/ch4.1.2.gif"> Barkley Sound</A>`

#### 4.2 자원기술구조(RDF) 데이터모형

RDF는 자원을 기술하고 상호교환하기 위한

구조로서 자원과 속성, 속성값으로 구성되며, 다음과 같은 의미로 사용된다(Brickley, Guha, and Layman 1998).

1) URI(Uniform Resource Identifier)로 고유하게 식별되는 객체를 자원으로 규정한다.

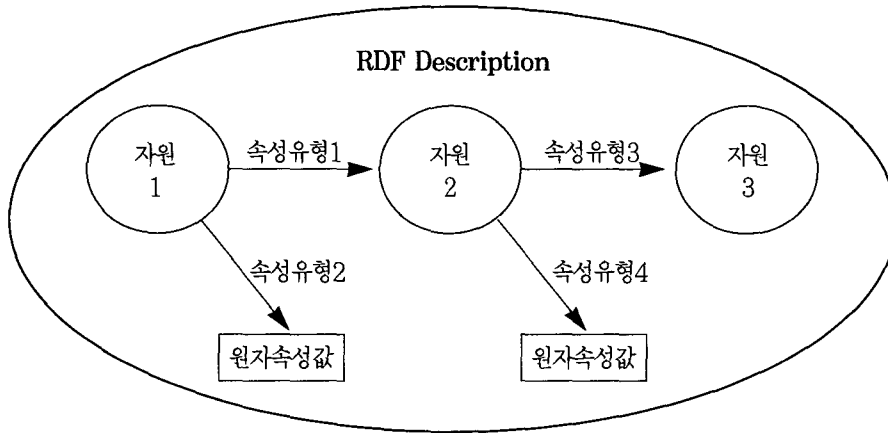
2) 자원과 관련된 속성을 속성유형(property-types)으로 규정하고, 속성유형도 하나의 자원으로 인정한다. 이 속성유형은 RDF에서 독자적으로 정의하여 사용되는 것이 아니라 각 메타데이터에서 정의한 스키마를 선언하고, 선언된 스키마에 정의된 속성유형(요소)을 RDF구문에서 사용하기 때문에 특정 메타데이터에 의존한 구문구조가 아니며, 따라서 상이한 메타데이터간의 교환이 가능하다. 속성유형에는 속성의 의미, 속성값, 객체의 유형, 속성간의 관계 등이 포함된다.

3) 속성유형은 값(values: 문자열이나 숫자)을 가지며 이 값은 다른 자원일 수 있고, 독자적인 속성일 수 있다.

4) 특정 자원의 속성집합을 기술(description)이라고 하고, RDF에서는 자원을 표현하는데 있어 구문 독립적이다(Iannella 1998).

다음은 RDF 데이터 모형의 구성요소를 도시화 한 것으로 전체 RDF의 기술은 자원과 속성유형, 속성값으로 표현되며 속성값이 또 다른 자원이 될 수 있다.

<그림 2>에서 [자원1], 속성유형2, "원자속성값"은 자원1의 속성집합을 이루는 하나의 기술이다. 이처럼 RDF에서는 자원과 속성유형, 값으로 구성된 기술을 통해 모든 자원을 표현하고 이와 관련된 자원도 표현할 수 있다. RDF 모형에서 자원은 노드(타원)로, 속성유형은 화살표(arc)로, 최종 속성 값은 사각형으로 표현된다.



〈그림 2〉 RDF 데이터모형

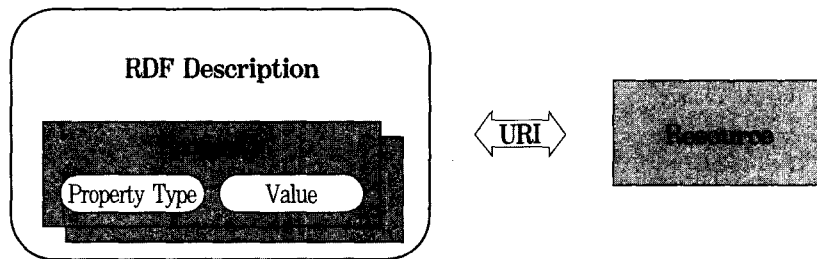
다음은 RDF에서 기술과 자원과의 관계를 도시한 것이다(Iannella 1998).

### 4.3 RDF 구문

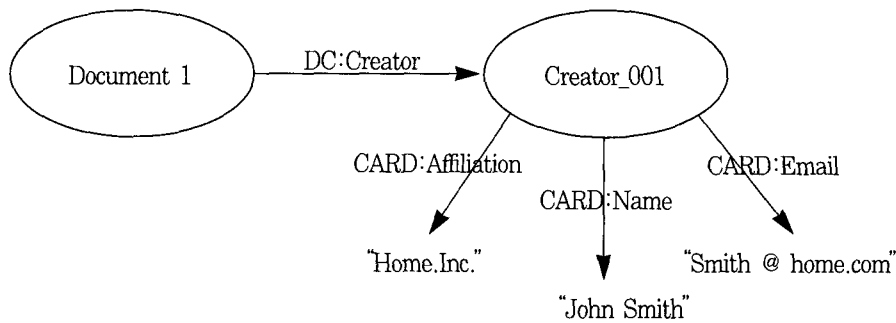
RDF 데이터 모형은 메타데이터를 정의하고 사용하기 위한 추상적이고 개념적인 구조로서, 실제로 메타데이터를 교환하고 작성하기 위해서는 구체적인 구문이 필요하다. 그런데 HTML은 웹 페이지의 구조를 표현하는데는 우수하지만 내용표현에 한계가 있으며, SGML은 내용을 상세하게 표현할 수 있으나 구조가 복잡하다. 이에 따라 RDF에서는 의미의 일관된 표현을 위해

XML(eXtensible Markup Language) 구문을 사용하고 있다. 이 XML은 확장할 수 있고, 복잡한 구조를 표현할 수 있다는 점에서 그 적용가능성이 인정되고 있다. 더욱이 최근 Explorer와 Netscape가 XML형식을 지원함에 따라 그 사용이 일반화될 것으로 보인다. 따라서 DC를 사용한 메타데이터에서는 RDF모형을 적용한 XML구문이 일반적으로 사용될 전망이다(Nordic metadata project 1998).

RDF에서는 메타데이터마다 데이터의 의미를 정의하도록 허용하고 있다. 그런데 상이한 분야에서 동일한 속성유형을 서로 다른 의미로 사용하는 것은 분명 문제가 된다. 이를 방지하기 위해



〈그림 3〉 자원과 RDF 기술과의 관계



〈그림 4〉 하위 속성유형의 표현

RDF에서는 XML의 'namespace'를 사용하여 속성유형의 의미와 적용규정을 고유한 방식으로 선언한다. 이 namespace를 통해 동일 의미를 정의한 특정한 DC 자원으로 연결하여, 메타데이터를 일관되게 입력하고 교환할 수 있다(Bray, Hollander, and Layman 1998; Miller 1998).

예컨대 DC에서 특정 문헌(document 1)의 저자(Creator)인 'John Smith'를 XML namespace를 사용하여 표현하면 다음과 같다.

```

<?xml:namespace ns = "http://www.w3.org/RDF/RDF/" prefix = "RDF" ?>
<?xml:namespace ns = "http://purl.oclc.org/DC/" prefix = "DC" ?>
<RDF:RDF>
<RDF:Description RDF:HREF = "http://uri-of-Documnet-1">
<DC:Creator>John Smith</DC:Creator>
</RDF:Description>
</RDF:RDF>
    
```

아울러 DC의 CREATOR 속성유형이 부차적인 속성(예: "name", "email", "affiliation")을 가진 경우, 이를 표현할 수 있다. 즉 문헌1의 값

인 "John Smith"에 대해 주소나 소속기관 등을 추가로 기술할 필요가 있을 때에는 "John Smith"를 하나의 독립된 자원으로 기술한다. 따라서 특정 자원은 그 자원의 고유한 속성유형과 값을 가지는 또 다른 기술(Description)로 표현된다. 그림 4에서 "John Smith"는 name과 email, affiliation과 같은 속성유형을 지닌 Creator\_001이라는 고유한 자원으로 대치된다. John Smith라는 값은 특정 자원의 저자일 수 있고, 아울러 특정 회사일 수도 있기 때문에 상이한 속성유형의 값을 가질 수 있고, 이 자원을 고유하게 식별하기 위한 기호로 연결한다.

그림 4에서 RDF스키마와 특정 메타데이터의 스키마가 각각 "RDF"와 "DC", "CARD"라는 형식으로 선언되고, namespace를 통해 특정한 URI를 가진 RDF와 DC, CARD스키마와 연결된다. DC:Creator 속성유형과 관련된 값이 다시 자원이 되고, 내부식별기호나 외부 URI를 사용하여 자원과 연결된다. 이와 더불어 DC의 CREATOR요소의 의미가 CARD 스키마에 따라 특정 의미로 정의된다(Guha, and Bray 1997). 이 구조는 워터구조와 유사하여, 자원의 기술에 사용된 패키지를 독립적으로 유지하거나 상호 교환할 수 있다. RDF를 통해 메타데이터의 일관된

입력과 교환이 가능하고 이를 통해 상이한 분야에서 정의된 독립된 메타데이터 패키지를 상호 교환할 수 있다.

XML의 namespace는 속성유형(요소)의 의미를 정확히 규정하기 위해 해당 속성유형을 정의한 스키마를 식별하기 위한 것으로, 기술되는 정보에 적용된 전거(HTML에서 스킴)를 표현하기 위한 수단이기도 하다. 따라서 특정 메타데이터 레코드 내에서 다양한 형식을 표현할 수 있다.

XML구문은 연속구문(serialization syntax)방식과 축약구문(abbreviated syntax)방식을 모두 지원하며, 동일 자료의 복수의 저자나 표제, 상이한 주제 등을 표현하기 위한 컨테이너모형을 제공하는 등 정보를 유연하고 간결하게 표현할 수

있는 장점이 있다. 특히 컨테이너 모형의 Sequence는 동일한 속성의 최종 속성값이 다수일 경우 표현을 간결하게 할뿐만 아니라 최종 속성값의 순서를 부여할 수 있어 다른 속성을 지닌 다수의 원자값과 1 : 1 대응이 가능하며, 서지정보에서 내용주기(505필드)의 TITLE과 AUTHOR의 표현에 적합하다. 아울러 alternative는 표제와 대응되는 상이한 형식의 표제(동일 문헌의 다른 표제)를 기술하는 데에, bag은 동일 속성을 지닌 정보가 다수인 경우 이를 표현하는 데 적합한 구문방식으로 평가되고 있다.

다음은 RDF모형의 xml구문을 표현한 것이다(Lassila 1998).

```
[1] RDF          ::= '<rdf:RDF>' obj* '</rdf:RDF>'
[2] obj          ::= description | container
[3] description  ::= '<rdf:Description' idAboutAttr? bagIdAttr? propAttr* '/>'
                  | '<rdf:Description' idAboutAttr? bagIdAttr? propAttr* '>'
                  | property* '</rdf:Description>'
                  | typedNode
[4] container    ::= sequence | bag | alternative
[5] idAboutAttr  ::= idAttr | aboutAttr | aboutEachAttr
[6] idAttr       ::= 'ID="' IDsymbol '"'
[7] aboutAttr    ::= 'about="' URI-reference '"'
[8] aboutEachAttr ::= 'aboutEach="' URI-reference '"'
[9] bagIdAttr    ::= 'bagID="' IDsymbol '"'
[10] propAttr    ::= propName '=' string '"'
                  (with embedded quotes escaped)
[11] property    ::= '<' propName idAttr? '>' value '</' propName '>'
                  | '<' propName idRefAttr? bagIdAttr? propAttr* '/>'
[12] typedNode   ::= '<' typeName idAboutAttr? bagIdAttr? propAttr* '/>'
                  | '<' typeName idAboutAttr? bagIdAttr? propAttr* '>'
                  | property* '</' typeName '>'
[13] propName    ::= QName
[14] typeName    ::= QName
[15] idRefAttr   ::= idAttr | resourceAttr
[16] value       ::= obj | string
[17] resourceAttr ::= 'resource="' URI-reference '"'
```

- [18] QName ::= [ NSName ':' ] name
- [19] URI-reference ::= (see RFC1738, RFC1808, [URI])
- [20] IDsymbol ::= (any legal XML name symbol)
- [21] name ::= (any legal XML name symbol)
- [22] NSName ::= (any legal XML namespace prefix)
- [23] string ::= (any XML text, with "<", ">", and "&" escaped)
- [24] sequence ::= '<rdf:Seq' idAttr? '>' member\* '</rdf:Seq'>
- [25] bag ::= '<rdf:Bag' idAttr? '>' member\* '</rdf:Bag'>
- [26] alternative ::= '<rdf:Alt' idAttr? '>' member+ '</rdf:Alt'>
- [27] member ::= referencedItem | inlineItem
- [28] referencedItem ::= '<rdf:li' resourceAttr '/'>
- [29] inlineItem ::= '<rdf:li>' value '</rdf:li'>

다음은 위의 RDF구조에 따라 XML을 사용하여 연속구문으로 DC 메타데이터를 기술한 예이다.

```

<RDF: RDF
  xmlns RDF="http://www.w3c.org/RDF/"
  xmlns DC="http://purl.org/RDF/DC/"
  <RDF:Description about="http://purl.org/metadata/dublin_core_elements">
    <DC:Title>Dublin Core Metadata Element Set: Reference Description</DC:Title>
    <DC:Creator>Stuart Weibel</DC:Creator>
    <DC:Creator>Eric Miller</DC:Creator>
    <DC:Subject>Metadata, Dublin Core element, resource description </DC:Subject>
    <DC:Description>This document is the reference description of the Dublin Core
      Metadata Element Set designed... discovery.</DC:Description>
    <DC:Publisher>OCLC Online Computer Library Center, Inc.</DC:Publisher>
    <DC:Format>text/html</DC:Format>
    <DC:Type>Technical Report</DC:Type>
    <DC:Language>en</DC:Language>
    <DC:Date>1997-11-02</DC:Date>
  </RDF:Description>
</RDF: RDF>

```

## 5. 결 론

메타데이터 구축 환경에서 종래는 MARC데이터와 메타데이터간의 데이터변환 과정에 많은 관심을 기울였다. 그러나 이 변환과정에서는 완전한 데이터변환이 불가능함을 알 수 있다.

1) KORMARC의 245\$b(부표제), 245\$x(대등표제), 240\$a(통일표제), 440\$a(총서표제)와 직접 대응되는 DC 필드가 없다. 따라서 이들 데

이터를 변환하고자 하는 경우 DC.Title.Subtitle과 DC.Title.Parallel, DC.Title.Series와 같은 필드를 고려할 수 있다.

2) 저자 변환과정에서 MARC 레코드의 1XX 필드의 \$a와 250필드의 \$b, 700필드(개인명)와 710필드(단체명)의 \$a를 대상으로 한다. 다만 7XX필드에서 저자 이외의 역할을 한 인물이나 단체, 그리고 245필드의 \$d와 \$e, 주기필드(5XX)에 수록된 관련저자는 변환대상에서 제외

한다.

3) Subject에서 KORMARC의 600필드(주제명-인명)와 610필드(주제명-단체명)에 대응되는 DC 요소는 없다. 아울러 MARC의 6XX필드에서는 \$a로부터 \$2까지 모든 하위필드를 고려해야 하는 문제가 있다.

4) KORMARC의 008/07-10에는 연도만이 수록되고, 이에 대응되는 DC.Date는 디폴트 값으로 'YYYY-MM-DD' ("1998-09-23")를 가지므로 완전한 변환이 어렵다.

5) KORMARC의 각종 표준부호(ISBN, ISSN 등)는 DC로 변환이 가능하다. 기타 부호는 필요에 따라 DC에서 대응되는 필드를 설정할 필요가 있다. 다만 ISBN이나 ISRC, ISMN, STRN 등이 메타데이터에서 반복되는 경우, 첫 번째 이후에 출현한 모든 번호는 무시한다.

6) KORMARC의 언어부호(041\$a)는 변환과정이 복잡하다.

7) 자료간의 관계유형을 표현하기 위한 Relation의 관계설정이 요구된다.

8) KORMARC의 내용연대는 연대부호(045\$a)를 해당 연도로 바꾸어준 다음 변환해야 한다.

9) KORMARC의 지역부호(043\$a)는 지명으로 전환한 후 변환해야 한다.

이와 같이 데이터변환에 의한 포맷의 통합을 통한 자료의 접근과는 달리 새로운 접근방안으로 RDF를 개발하게 되었다. 일반적으로 웹 자원은 분산되어 있고 동일 자원이 다양한 형식(포맷)으로 표현될 수 있다는 점을 특성으로 한다. 그런데 네트워크자원을 확인하고 접근하기 위해 지금까지 적용분야마다 독자적인 메타데이터형식이 사용되어 왔으며, 이런 이유로 자원의 접근에 제약을 받아 왔다. 따라서 상이한 메타데이터간의 연결기법이 중요한 과제가 되어 왔다.

DC에서는 워릭구조를 통해 이 문제를 해소하고자 하였다. 워릭구조는 컨테이너와 패키지를 통해 상이한 메타데이터를 연결하는 구조인데 비해서, RDF는 데이터와 메타데이터간을 구분하지 않고 모든 자원을 독립적으로 취급하고, 특정 자원과 이와 관련된 자원을 연결하는 모형이라고 할 수 있다. RDF의 유용성은 자원의 성격을 표현하는 속성유형과 값을 하나의 독자적인 자원으로 인정하는 것이다. 그런데 분산된 정보를 탐색하고 접근하기 위해서는 의미와 구조, 구문에서 공통의 규칙을 필요로 한다. 다양한 분야에서 메타데이터를 교환하고 이용하기 위해서는 기계가 처리할 수 있는 데이터요소의 체계적 구문이 요구되고 있다. XML은 메타데이터의 의미를 일관되게 표현하고 교환하기 위한 공식적인 구문으로 발전될 전망이다.



## 참 고 문 헌

- 김태수. 1998. "더블린 코어". <http://dewey.yonsei.ac.kr/metadata/DC/htm>
- 첨단학술정보센터. 1997. 디지털 정보표현을 위한 메타데이터 표준개발에 관한 연구. [서울]: 동 센터.
- ALA. Committee on Cataloging: Description and Access. 1988. "Final Report". <http://www.ala.org/alcts.organization/cs/ccda/tf-tei7.html>.
- Attig, John. 1998. "Metadata and cataloging: Supporting common user tasks". <http://www.ala.org/alcts.organization/cs/ccda/tf-tei3.html>
- Bray, Tim, Dave Hollander, and Andrew Layman. 1998. "Name spaces in XML". W3C Note. <http://www.w3.org/TR/1998/NOTE-xml-names-0119>.
- Brickley, Dan, R.V. Guha, Andrew Layman. 1998. "Resource Description Framework(RDF) Schemas". W3C Working Draft, 9 April 1998. WD-rdf-schema-19980409. <http://www.w3.org/TR/WD-rdf-schema>
- DESIRE. 1996. "A review of metadata: A survey of current resource description formats". <<http://www.ukoln.ac.uk/metadata/desire/overview/>>.
- EULER. 1997. "Project description: Objectives of the project". <http://www.emis.de/projects/EULER/objectives.html>.
- Guha, R.V., and Tim Bray. 1997. "Meta content framework using XML". <http://www.w3.org/TR/NOTE-MCF-XML/>.
- Hansen, Preben. 1998. "User guidelines for Dublin core creation". [http://www.sics.se/~preben/DC/DC\\_guide.html](http://www.sics.se/~preben/DC/DC_guide.html).
- Iannella, Renato. 1998. "A Idiot's Guide to the Resource Description Framework". <http://www.dstc.edu.au/RDU/reports/RDF-Idiot/>.
- IFLA. Study Group on the Functional Requirements for Bibliographic Records. 1998. "Digital libraries: metadata resources". <http://www.nlc-bnc.ca/ifla/II/metadata.htm#tools>.
- Lambrecht, Jay H. 1992. Minimal Level Cataloging by National Bibliographic Agencies. UBCIM Publications, New Series vol. 8. München : K.G.Saur.
- Lassila, Ora. 1997. "Introduction to RDF metadata". W3C NOTE. <http://www.w3.org/TR/NOTE-rdf-simple-intro>.
- Library of Congress. 1997. "Metadata, Dublin core and USMARC: a review of current efforts". [gopher://marvel.loc.gov:70/00/listarch/usmarc/dp99.doc](http://gopher://marvel.loc.gov:70/00/listarch/usmarc/dp99.doc)
- Miller, Eric, 1997. "Monticello electronic library: Dublin core element set crosswalk". <http://222.oclc.org:5046/~emiller/DC/crosswalk.html>.
- Miller, Eric. 1998. "An introduction to the resource description framework". <http://www.dlib.org/dlib/may98/miller/>

- 05miller.html.
- Nordic Metadata Project. 1998. "Final report".  
[http://www.ub.lu.se/metadata/DC\\_creator.html](http://www.ub.lu.se/metadata/DC_creator.html))
- Raggett, Dave, Dan Connolly, Tim Berners-Lee, Murray Maloney, Liam Quin. 1997. <http://www.w3.org/TR/WD-htmllink-970328>.
- "Resource description framework(RDF) model and syntax". <http://www.w3.org/RDF/Group/WD-rdf-syntax/>.
- Thorborg, Susanne. 1997. "Dublin core/dan MARC2/GILS crosswalk". <http://linnea.helsinki.fi/meta/dcdancr.html>.
- W3C RDF Schema Working Group. 1998. <http://www.w3.org/TR/WE-RDF-Schema/>.