

메타데이터의 구축환경

연세대학교 문헌정보학과

김 태 수

서 론

디지털 기술과 네트워크 환경을 배경으로 한 새로운 정보환경의 출현으로, 종래와는 다른 새로운 네트워크 자원이 출현하게 되었다. 일반적으로 네트워크 자원은 복제와 수정이 용이하고, 이로 인해 정보의 신뢰성을 확보하기 어려우며(가변성), 자원의 소재위치가 자주 변경되며(이동성), 특정 자원의 구성요소가 여러 곳에 분산되어 존재하며(구성요소의 분산성), 자원의 구조와 소재 관련 내용, 발행기법이 다르다(포맷과 구조의 상이성). 여기에, 사용하는 시스템 및 네트워크 환경 등이 상이하여, 자원의 기술과 탐색에 어려움을 겪고 있다. 이런 점에서 네트워크자원은 내용과 형식(포맷), 접근방법, 이용 면에서 기존의 인쇄자원과 성질을 달리하고 있어, 이들 자원의 기술과 조직기법, 접근방안을 모색할 필요가 있다.

그런데 웹에서 전문을 대상으로 한 로봇색인은 자원의 탐색에 한계가 지적되고 있다. 이에 따라, 다양한 형식의 메타데이터가 개발되고 있는데, 더블린 코어, TEI (Text Encoding Initiative) 헤더(Header), 정부정보소재서비스(Government Information Locator Service: GILS), 지리정보(Federal Geographic Data Committee: FGDC), EAD (Encoded Archival Description) 등이 대표적인 예이다.

메타데이터란 말은 넓은 의미에서 데이터에 관한 데이터(Heery 1996, 345) 혹은, 전자자원의 기술(記述)에 사용된 데이터 요소로서, 최근에는 '웹 자원을 기술한 데이터'(Lassila 1998)란 의미로 주로 사용되고 있다. 이런 점에서 메타데이터는 데이터에 관한 구조화된 데이터(Miller 1998)로서, 자원과는 독립적으로 존재하면서 다양한 접근점과 네트워크 주소를

포함한 레코드라고 할 수 있다.

그 중에서 더블린 코어 메타데이터(Dublin Core metadata elements set: DC)는 자원의 기술에 필요한 일련의 데이터 요소를 규정하여 데이터의 호환성을 유지하고, 이들 자원의 신속한 검색을 목적으로 1995년 OCLC와 NCSA (National Centre for Supercomputer Applications)가 더블린(Dublin)에서 개최된 회의에서 합의된 메타데이터를 말한다. 개발의 주된 배경은 MARC와 같은 기존의 형식은 구조의 경직성으로 인해 네트워크 자원의 기술에 한계가 인정되고 있어, 이를 대체할 수 있는 단순구조의 형식을 필요로 하였기 때문이다. 이에 따라, 자원의 본질적인 특성을 기술요소로 하되(고유성), 규정된 필수 데이터 요소 이외에 부차적인 내용이나 특성을 기술요소로 사용하며(확장성), 응용분야나 표현기법을 규정하지 않으며(구문의 독립성), 각 요소의 수록여부를 강제하지 않으며(선택성), 모든 기술요소는 반복 사용할 수 있으며(반복성), 한정어를 사용하여 세부사항을 조정(수정 가능성)한다는 것이다(Weibel et al. 1997). 이에 따라, 15개의 데이터 요소를 기본요소(core elements set)로 규정하였다.

기본적으로 DC에서는 자원의 소재위치와 이 자원을 획득하는데 필요한 데이터만으로 요소를 구성하고, 안전이나 검증, 구독, 이용과 관련된 요소는 제외하였다. 이처럼 DC에서 네트워크자원의 식별에 필요한 데이터 요소만으로 제한한 기본의도는 데이터 요소의 단순화를 통해서 유용성과 이용의 용이성을 확보하고, 필요시 다양한 목적에 따라 확장하기 위한 것이다. 이 메타데이터의 기본요소에서 가장 중요한 점은 요소의 의미와 내용이 특정한 기술규칙이나 구문과는 독립적이라는 점이다.

더블린 코어의 데이터 요소

1) 기본 데이터 요소

기본데이터 요소와 그 의미는 다음과 같다.

요소명	정 의
1) Title	자원의 포제
2) Creator	자원의 내용에 주된 책임을 진 인물이나 단체
3) Subject	자원의 내용상의 주제(키워드나 구, 분류기호)
4) Description	자원의 내용에 관한 기술(초록이나 목차)
5) Publisher	자원의 발행처
6) Contributor	자원의 내용상에 기여한 인물이나 기관, 서비스
7) Date	자원의 제작 일자(ISO 8601)
8) Type	자원의 내용의 성질이나 장르
9) Format	자원의 물리적 디지털 형식
10) Identifier	자원의 고유한 식별기호(URL을 포함한 URI, DOI, ISBN)
11) Source	현 자원의 출처
12) Language	자원의 내용을 기술한 언어(RFC 1766, ISO 639, ISO 3166)
13) Relation	관련자원과의 관계
14) Coverage	자원의 범위(지명이나 좌표, 시대범위)
15) Rights	자원에 관한 권한 정보

2) 한정어(qualifier)

DC의 데이터 요소만으로는 완전한 기술이 불가능하다. 따라서 자원을 효과적으로 기술하기 위해 스킴(Scheme)과 유형(Type)을 한정어로 사용한다.

(1) 스킴(scheme)

LCSH나 KDC 등과 같은 주제명표나 분류표, 기타 외부표준을 제시하여, CONTENT에 수록된 값을 해석하는데 사용된다. 이를 통해 레코드의 일관성을 유지하고 일정 부분 표준화를 달성할 수 있다.

(2) 유형(Type)

기술에서 특정 요소의 하위요소를 기술할 때 사용한다. 예컨대, creator 요소가 “name”, “address” “telephone”과 같은 하위요소를 가지게 되면 이 요소를 type 값에 따라 반복 사용할 수 있다. 일반적인 형식은 다음과 같다.

```
<META NAME="DC.Element name. TYPE identifier" CONTENT="value">
```

따라서 저자의 이메일 주소는 다음과 같이 기술할 수 있다.

```
<META NAME="DC.Creator.Addressemail"
CONTENT="juha.hakala@helsinki.fi">
```

그런데 한정어를 사용하게 되면 입력자에 따라 기술수준이 상이하여, DC의 상호운용성에 제한을 줄 수 있다. 예컨대, date 요소에 스킴을 사용하게 되면, 날짜의 입력형식을 규정한 표준에 대한 지식을 필요로 한다.

3) 표현형식

(1) HTML 형식

일반적으로 웹에서 접근할 수 있는 자원은 대부분 HTML로 기술되어 있다. 다음은 HTML의 <META> 태그로 기본요소를 표현한 예이다.

```
<META NAME="DC.Title"
CONTENT="DC User Guidelines">
<META NAME="DC.Creator.PersonalName"
CONTENT="Hansen, Preben">
<META NAME="DC.Subject"
SCHEME="LCSH"
CONTENT="Cataloging of computer files">
<META NAME="DC.Description"
CONTENT="User guidelines for creation of
Dublin Core metadata using the Nordic DC
Metadata creation tool.">
```

```

<META NAME="DC.Publisher"
  CONTENT="Nordic Metadata Project">
<META NAME="DC.Contributor.CorporateName"
  CONTENT="SICS">
<META NAME="DC.Date.Created"
  SCHEME=ISO 8601
  CONTENT="1998-01-16">
<META NAME="DC.Resource Type"
  CONTENT="Text">
<META NAME="DC.Format"
  SCHEME=IMT
  CONTENT="text/html">
<META NAME="DC.Identifier"
  CONTENT="http://www.sics.se/%7Epreben/
  DC/DC_guide.html">
<META NAME="DC.Source"
  CONTENT="Levine and Baroudi: Internet se-
  crets">
<META NAME="DC.Language"
  SCHEME="ISO639-2"
  CONTENT="en">
<META NAME="DC.Relation"
  SCHEME="URL"
  CONTENT="http://www.lub.lu.se/cgi-bin/nm
  DC.pl">
<META NAME="DC.Coverage"
  CONTENT="Scandinavia">
<META NAME="DC.Rights"
  CONTENT="Public domain">

```

(2) XML Namespace에 의한 표현

RDF에서 XML의 구분으로 Dublin Core Namespace를 예시한 것이다. 먼저 RDF를 선언하고, 기술에서 각 데이터 요소를 제시한다(Miller, Miller, and Brickle. 1999)

```

<? xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/TR/REC-rdf-syntax#"
  xmlns:rdfs="http://www.w3.org/TR/WD-rdf-schema#"

```

```
xmlns:dc="">
```

```

<rdf:Description rdf:about="">
  <dc:title> The Dublin Core Element Set</dc:title>
  <dc:creator> The Dublin Core Metadata
  Initiative </dc:creator>
  <dc:description> The Dublin Core is a simple
  metadata element set intended to facilitate
  discovery of electronic resources.</dc:descri-
  ption>
  <dc:date> 1995-03-01 </dc:date>
</rdf:Description>

<rdf:Description ID="title">
  <rdf:type rdf:resource="http://www.w3.org/TR/REC-
  C-rdf-syntax#Property"/>
  <rdfs:label> Title</rdfs:label>
  <rdfs:comment>The name given to the resource,
  usually by the Creator or Publisher.
  </rdfs:comment>
  <rdfs:isDefinedBy=""/>
</rdf:Description>

<rdf:Description ID="creator">
  <rdf:type rdf:resource="http://www.w3.org/TR/REC-
  rdf-syntax#Property"/>
  <rdfs:label> Author/Creator</rdfs:label>
  <rdfs:comment>The person or organization pri-
  marily responsible for creating the intellectual
  content of the resource.</rdfs:comment>
  <rdfs:isDefinedBy=""/>
</rdf:Description>

<rdf:Description ID="subject">
  <rdf:type rdf:resource="http://www.w3.org/TR/REC-
  C-rdf-syntax#Property"/>
  <rdfs:label> Subject</rdfs:label>
  <rdfs:comment>The topic of the resource.</rd-
  fs:comment>
  <rdfs:isDefinedBy=""/>
</rdf:Description>

```

```

<rdf:Description ID="description">
  <rdf:type rdf:resource="http://www.w3.org/TR/REC-rdf-syntax#Property"/>
  <rdfs:label> Description </rdfs:label>
  <rdfs:comment> A textual description of the content of the resource, including abstracts in the case of document-like objects or content descriptions in the case of visual resources.
  </rdfs:comment>
  <rdfs:isDefinedBy=""/>
</rdf:Description>

```

```

<rdf:Description ID="publisher">
  <rdf:type rdf:resource="http://www.w3.org/TR/REC-rdf-syntax#Property"/>
  <rdfs:label> Publisher </rdfs:label>
  <rdfs:comment> The entity responsible for making the resource available in its present form, such as a publishing house, a university department, or a corporate entity. </rdfs:comment>
  <rdfs:isDefinedBy = ""/>
</rdf:Description>

```

```

<rdf:Description ID="contributor">
  <rdf:type rdf:resource="http://www.w3.org/TR/REC-rdf-syntax#Property"/>
  <rdfs:label> Other Contributors </rdfs:label>
  <rdfs:comment> A person or organization not specified in a Creator element who has made significant intellectual contributions to the resource but whose contribution is secondary to any person or organization specified in a Creator element (for example, editor, transcriber, and illustrator). </rdfs:comment>
  <rdfs:isDefinedBy = ""/>
</rdf:Description>

```

```

<rdf:Description ID="date">
  <rdf:type rdf:resource="http://www.w3.org/TR/REC-rdf-syntax#Property"/>

```

```

  <rdfs:label> Date </rdfs:label>
  <rdfs:comment> A date associated with the creation or availability of the resource.
  </rdfs:comment>
  <rdfs:isDefinedBy=""/>
</rdf:Description>

```

```

<rdf:Description ID="type">
  <rdf:type rdf:resource="http://www.w3.org/TR/REC-rdf-syntax#Property"/>
  <rdfs:label> Type </rdfs:label>
  <rdfs:comment> The category of the resource, such as home page, novel, poem, working paper, technical report, essay, dictionary.
  </rdfs:comment>
  <rdfs:isDefinedBy = ""/>
</rdf:Description>

```

```

<rdf:Description ID="format">
  <rdf:type rdf:resource="http://www.w3.org/TR/REC-rdf-syntax#Property"/>
  <rdfs:label> Format </rdfs:label>
  <rdfs:comment> The data format of the resource, used to identify the software and possibly hardware that might be needed to display or operate the resource. </rdfs:comment>
  <rdfs:isDefinedBy = ""/>
</rdf:Description>

```

```

<rdf:Description ID="identifier">
  <rdf:type rdf:resource="http://www.w3.org/TR/REC-rdf-syntax#Property"/>
  <rdfs:label> Identifier </rdfs:label>
  <rdfs:comment> A string or number used to uniquely identify the resource. </rdfs:comment>
  <rdfs:isDefinedBy = ""/>
</rdf:Description>

```

```

<rdf:Description ID="source">
  <rdf:type rdf:resource="http://www.w3.org/TR/REC-rdf-syntax#Property"/>

```

```

<rdfs:label> Source</rdfs:label>
<rdfs:comment>Information about a second re-
source from which the present resource is
derived.</rdfs:comment>
<rdfs:isDefinedBy=""/>
</rdf:Description>

```

```

<rdf:Description ID="language">
<rdf:type rdf:resource="http://www.w3.org/TR/REC-
rdf-syntax#Property"/>
<rdfs:label> Language</rdfs:label>
<rdfs:comment>The language of the intellectual
content of the resource.</rdfs:comment>
<rdfs:isDefinedBy=""/>
</rdf:Description>

```

```

<rdf:Description ID="relation">
<rdf:type rdf:resource="http://www.w3.org/TR/REC-
rdf-syntax#Property"/>
<rdfs:label> Relation</rdfs:label>
<rdfs:comment> An identifier of a second reso-
urce and its relationship to the present resour-
ce.</rdfs:comment>
<rdfs:isDefinedBy=""/>
</rdf:Description>

```

```

<rdf:Description ID="coverage">
<rdf:type rdf:resource="http://www.w3.org/TR/REC-
rdf-syntax#Property"/>
<rdfs:label> Coverage</rdfs:label>
<rdfs:comment>The spatial or temporal characte-
ristics of the intellectual content of the
resource.</rdfs:comment>
<rdfs:isDefinedBy=""/>
</rdf:Description>

```

```

<rdf:Description ID="rights">
<rdf:type rdf:resource="http://www.w3.org/TR/REC-
rdf-syntax#Property"/>
<rdfs:label> Rights</rdfs:label>
<rdfs:comment> A rights management statement,

```

an identifier that links to a rights management statement, or an identifier that links to a service providing information about rights management for the resource.</rdfs:comment>

```

<rdfs:isDefinedBy=""/>
</rdf:Description>

```

```
</rdf:RDF>
```

4) 자원과 메타데이터간의 연결

일반적으로 웹에서는 하이퍼링크를 통해 메타데이 터나 자원을 탐색하고 색인한다. 이것은 웹이 다음 과 같은 3가지 기본 개념에 기초하고 있기 때문이 며, 이 중에서 HTML이 하이퍼텍스트문서 포맷에서 중요한 역할을 한다(Raggett, et al. 1997).

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

(1) HTML 구문

HTML에서는 메타태그와 연결태그를 사용하여 자 원과 메타데이터를 연결한다. 메타태그는 메타데이 터를 HTML문헌의 HEAD부분으로 전환하기 위한 것이고, 링크태그는 일련의 메타데이터와 간접적으 로 연결하기 위한 것이다. 웹 환경에서 자원과 메타

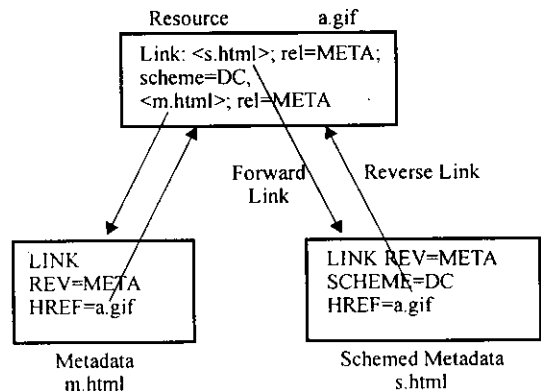


그림 1. 자원과 메타데이터간의 연결

데이터는 다음과 같은 구조로 연결된다.

① **메타태그**; HTML에서는 개개의 메타태그를 이름(name)과 내용(content)이라는 두 개의 속성 값으로 기술한다. HTML문헌의 HEAD에는 여러 형태의 메타요소를 포함할 수 있다.

```
<META NAME="title" CONTENT="Gone with the Wind">
```

② **링크태그**; <schema_name>과 같은 유형의 메타데이터를 포함한 자원이 특정 <URL>을 지니고 있는 경우, 링크태그는 다음과 같은 구문을 취한다.

```
<LINK REL=META.<schema_name> HREF="<URL">">
```

메타데이터 구축 환경

메타데이터 레코드의 제작을 용이하게 하고 자료를 수집, 색인하기 위해서는 각종 도구의 개발이 필요하다. 현재, 메타데이터를 제작하기 위한 도구에는 두 가지 유형이 있다. 하나는 일반적으로 편집기(editor)로서, 데이터 요소를 입력할 수 있는 템플릿을 통해 메타데이터를 작성하는 방식이다. 각종 지원 프로그램을 통해 해당 데이터를 HTML <META>태그로 표현한다. 다른 하나는 저작도구(generators)

로서, html로 작성된 기존의 자원에서 메타데이터를 추출하여 메타데이터를 작성하는 도구이다.

1) Dublin Core Metadata template

이 템플릿에서는 메타데이터의 모든 속성과 스킴을 제시하며, 각종 도움말을 제공하여 데이터변환을 지원한다. 사용포맷은 HTML v3.2와 v4.0으로서, 주된 특징은 최소수준의 메타데이터를 제작할 수 있다. 다음은 템플릿의 유형을 표본으로 제시한 것이다.

2) Reggie

이 편집기는 한정어를 데이터 요소에 사용할 수 있으며, 각종 도움말을 제공하여 메타데이터를 작성할 수 있다. 사용 포맷은 HTML v3.2, v4.0, RDF/XML이며, 주된 특성은 RDF를 지원한다는 점이다. Netscape v4.0 혹은 Internet Explorer v4.0 환경을 필요로 한다.

- 개발자: the Distributed Systems Technology Centre
- URL: <http://metadata.net/dstc>
- DC수준: 한정어 사용가능
- 포맷: HTML v3.2, v4.0, RDF/XML
- 특성: RDF지원
- 한계: JAVA사용시 속도가 느림
Netscape v4.0 혹은 Internet Explorer v4.0

3) DC-dot 저작도구

이 저작도구는 기술대상자원의 url을 입력하면 해당 웹 페이지를 검색하여 이를 DC의 메타태그로 자동 제작하는 도구로서(UKOLN 1998), DC의 기본 데이터 요소만을 사용하고, 한정어를 사용하지 않는다. 사용포맷은 HTML로서, PERL v5.0과 JAVA v1.1에서 사용이 가능하다.

- 개발자: UKOLN
- URL: <http://www.ukoln.ac.uk/metadata/dcdot/i>
- DC: 수준한정어 사용불가
- 포맷: HTML (version unspecified)
- 특성: 다른 사이트에서 템플릿의 기초로 사용됨
PERL v5.0과 JAVA v1.1에서 사용가능
- 한계: 작성된 메타데이터가 주제에서 일관성 상실

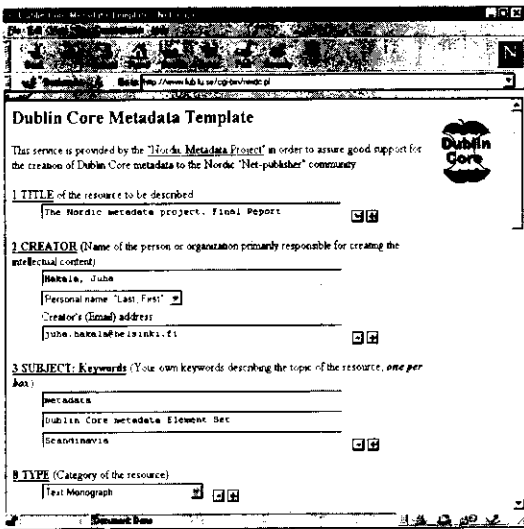


그림 2. 노르딕 메타데이터 템플릿

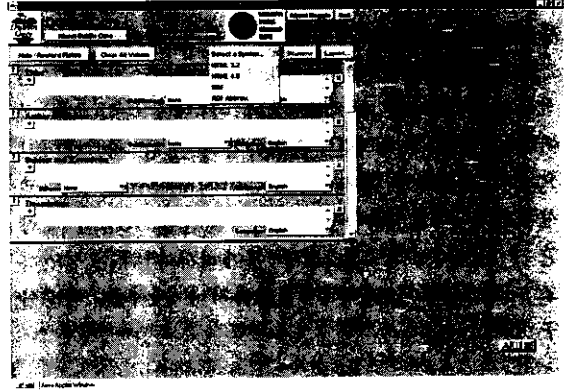
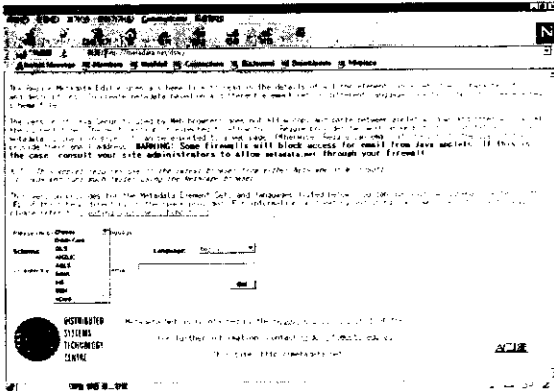


그림 3. Reggie

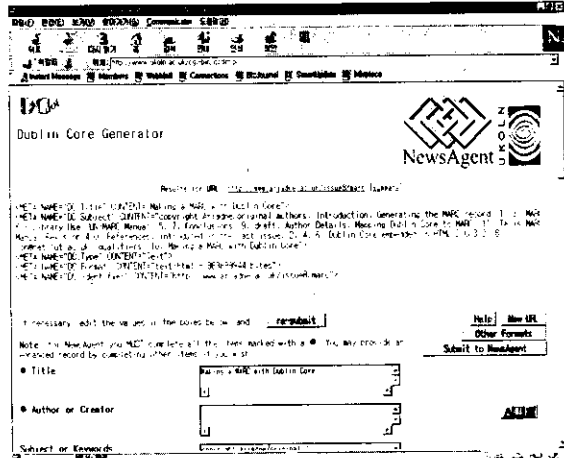
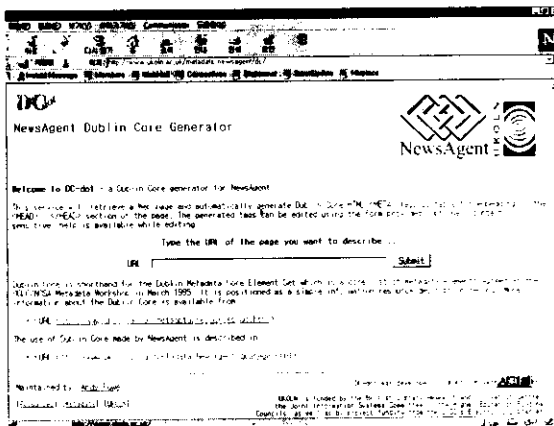


그림 4. DC-dot 제작 도구

4) 기타 도구

메타데이터를 작성하는 주된 이유는 인터넷상의 자원을 검색하기 위한 것이기 때문에 주제의 기술과 검색은 중요한 의미를 지닌다. 따라서, 템플릿이나 주제지연도구를 통하여 Subject 요소를 작성하고, 시소러스나 분류표, 전거파일과 연결되어야 한다.

아울러, 메타데이터로 표현된 자원은 웹 색인에 의해 구체적으로 색인되고 탐색된다. 따라서, 템플릿은 robot으로 작성되는 웹 색인과 긴밀하게 연결되

고, 최종적으로는 종합목록과 연결하는 도구의 개발이 필요하다.

자원기술구조

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

데이터의 의미와 구문을 통해 메타데이터의 상호운용성(interoperability)을 확보하기 위한 것이다.

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

1) RDF의 데이터모형

RDF는 메타데이터를 웹 상에서 효과적으로 작성하고 교환, 이용하기 위한 것으로, 구조를 정의하는 대신, 의미에 관한 몇 가지 전제를 두고 있으며 전송구문으로 XML을 권고하고 있다. 따라서, RDF에서 제시된 구조와 XML 구문을 결합하여 DC의 데이터 요소를 의미적으로 보완하게 된다.

RDF에서는 속성(크기, 이름, 등)을 통해 자원을 식별하고 확인하며, 이 속성은 관련된 값(value)을 가진다. 특수한 경우, 속성의 진술(statement) 자체가 또 하나의 자원이 되며, 자원의 연결정보에 의해 이 모형을 무한히 확장할 수 있다. RDF에서는 자원과 속성을 다음과 같은 의미로 사용하고 있다(Brickley, Guha, and Layman 1998; Iannella 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 사용되는 것이 아니라 각 메타데이터에서 정의된 스키마의 속성유형(요소)을 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.

RDF 기술에서 자원과 속성, 값의 관계를 가장 기본적인 단정(assertion)으로 도시하면 그림 5와 같다.

그림 5에서 http://doc는 자원이며, author는 속성, Joe Smith는 속성의 값이다. 다시 말해, Joe Smith는 http://doc의 저자이다. 이 모형에서 자원은 항상 타원형으로, 속성은 화살표로(진술의 주체로부터 대상으로 연결) 표현된다. 진술이 단순한 텍스트 열("Joe Smith")인 경우, 사각형으로 제시된다. 진술 그 자체가 자원인 경우, 진술은 또 다른 속성이나 진술, 자원을 지시하는 타원형으로 표현된다.

만약, "Joe Smith"가 자원(http://doc)에서 삽화가로서의 역할을 한 경우는 그림 6과 같이 표현할 수 있다.

2) XML의 사용

(1) Name Space기능

RDF는 다양한 자원과의 연계를 통하여 자원을 기술하는 모형이기 때문에 최소한의 기술속성과 연계방안 및 기술모형만을 제공한다. XML의 Name space 기능은 다른 자원과의 연계 수단이며, 메타데이터기술에 사용된 기술속성의 의미를 정확하게 표현하기 위해 메타데이터 스키마를 연계(참조)하는 수단이다. 따라서, 이 기능을 이용하여 사용할 메타데이터 스키마를 접두어와 함께 선언하고, 선언된 이름(접두어)을 스키마에서 정의된 속성과 결합하는

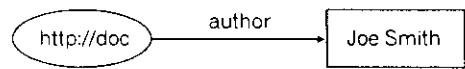


그림 5. 단순한 RDF 단정(assertion)

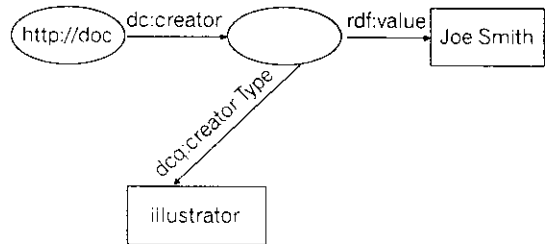


그림 6. 저자와 삽화가로서의 인물 표현

방법이다.

컨텐츠를 제공하는 각 분야에서 자원을 기술할 때 사용한 표현형식을 XML과 RDF를 통해 선언한다. 이것은 각 분야에서 자원의 모든 관점을 기술하는 포괄적인 시스템을 가지기보다는 하나의 기술을 통해 상호운용성을 확보하기 위한 것이다. 예컨대, Creator라는 용어는 여러 분야에서 다양한 의미를 가질 수 있다. 만약, 이 용어가 DC에서 사용되는 요소라면 이 요소의 의미와 내용은 훨씬 분명해 질 것이다. 모든 분야에서 'Creator'라는 개념을 공유하여 동일한 요소를 이 개념으로 기술할 수 있는 것이다. 이와 같은 상호운용성을 실현하는 도구를 XML Namespace라고 한다.

기본적으로 XML Namespace는 어떤 자원기술요소의 의미를 분석하기 위한 수단이다. 이 개념을 분명하게 식별하기 위해 <http://purl.org/dc/elements/1.0/creator>와 같은 고유한 단정(assertion)을 사용함으로써, XML/ RDF는 메타데이터에 있는 용어를 기술의 대상으로 처리하게 된다. 나아가 용어와 개념간의 상호관계를 표현할 수 있는 기반을 제공하게 된다.

상이한 'namespaces'에 '사용된 용어간을 구분하기 위해, XML의 핵심요소가 아닌 모든 요소 앞에 그 용어가 사용된 namespaces를 간략한 식별기호로 표기한다. 예를 들어, Creator가 DC의 요소임을 제시하기 위해, 'dc:'를 관기한다. 'dc'와 Dublin core간의 관계를 연결하기 위해, XML namespace로 구성된 RDF 기술에서는 사용되는 namespace와 위치를 선언하면 관련 소프트웨어가 해당 기술(description)을 분석하게 된다. 예컨대, 'dc:'와 Dublin Core를 연결하기 위해, XML로 표현된 RDF기술은 다음과 같다.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.0/">
```

소프트웨어는 dc:creator를 <URL:<http://purl.org/dc/elements/1.0/creator>>로 변환하기 위해 이 진술을 분해하며, 후속되는 링크에 의해 이 요소의 의미를 해석하게 된다. 따라서 Name space기능을 통한 접두어의 사용으로 간결하고 정확한 기술이 가능하다.

(2) RDF에서 Dublin Core의 구분

Dublin Core 데이터 요소는 주로 자원의 기술을 위한 의미와 관련되어 있다. 구분은 일반적으로 개발자의 재량이지만, RDF에서는 데이터 요소의 의미를 유지하고 이를 표현하기 위해 XML을 구분으로 사용한다.

① Dublin Core의 Namespace; RDF에서 사용되는 XML의 Namespaces는 각 분야에서 자원의 탐색수단으로 독자적인 개발이 가능하며, 필요에 따라 특수한 구조를 선정할 수 있다.

Dublin Core의 15개 요소가 하나의 Namespace를 구성한다(2.3.2 Dublin Core 요소 세트 Namespace 참조). Namespace를 식별하기 위해 기술에서는 어떤 문자열이라도 사용이 가능하지만, 일반적으로 영문자 소문자('dc:')의 사용이 권고되고 있다.

② RDF에서 Dublin Core의 XML 입력방법; XML로 인코딩된 Dublin Core의 기본요소

ㄱ) XML 버전 선언: 현재 사용되는 버전(version)은 1.0이다. 따라서, 문서는 다음과 같은 내용을 첫 행에 포함해야 한다.

```
<?xml version="1.0"?>
```

ㄴ) RDF 선언: RDF를 사용하였음을 선언하는 과정이 필요하다. 이 선언을 통해, XML 문서에 일정한 기능과 제한을 하게된다. 다음과 같이, XML 버전 선언 다음 행의 RDF 기술의 첫 행에 이를 선언하고, </rdf:RDF>로 종료된다.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
```

RDF에서 Dublin Core를 기술하기 위해서는 Dublin Core Namespace를 사용해야 하기 때문에, 이 내용은 대개 다음과 같이 확장된다.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.0/">
```

이 확장된 진술을 통해 RDF가 사용되고 있음을 이용자와 applications에게 알리게 되며, 접두사 'rdf.'에 의해 문서를 인식하게 된다. 이밖에, 'dc.'에 의해 Dublin Core가 사용되고 있음을 알 수 있다.

ㄷ) 요소명에서 대소문자의 유지: XML에서는 요소명으로 사용된 대소문자는 대단히 중요하다. 따라서, 'Creator', 'creator', 'CREATOR', 'cREATOR'는 완전히 다른 4개의 요소로 해석된다. 현재는 요소명을 모두 영문자 소문자로 표현하도록 규정하고 있다.

ㄹ) XML 태그는 종료되어야 한다: XML 문서에서는 시작 태그 <TAG>와 종료 태그 </TAG>로 표시된 경우에만 정당한 문서로 취급된다.

3) Dublin Core의 확장

Dublin Core에서는 한정어와 하위구조를 통해 기술의 내용을 확장해 왔다. 이를 위해, 'scheme', 'type' 등이 사용되어 왔다.

(1) 한정어의 구성요소

Dublin Core를 한정하기 위한 기본구문을 도시하면 그림 7과 같다.

① 자원(Resource); 식별되는 자원으로, 전자문서나 이미지, 서비스("today's weather report for Los Angeles"), 다른 자원의 집합일 수 있다. RDF에서 자원은 항상 URI로 해 식별된다.

② 요소(Element); 요소는 자원의 속성으로서, RDF 데이터 모형에서 자원은 속성과 유사하다. Dublin core로 정의된 각 요소(혹은 속성)는 특수한 의미를 가지며 일정한 값으로 정의되며, 자원의 유형, 다른

요소와의 관계일 수 있다.

③ 리터럴(Literal); 리터럴은 단순한 텍스트 열로서, 구조화된 내용을 가질 수 있다.

④ 요소 한정어(Element qualifiers); 특정 자원의 관점을 15개의 핵심요소보다 더 정확하게 제공하기 위한 것으로, 요소 한정어는 날짜의 범위를 제작일자, 수정일자와 같이 더 전개할 수 있다. 요소 한정어는 'types'로 표현된다.

⑤ 값 한정어(Value qualifiers); 값 한정어는 값을 명세하기 위한 것으로, 일반적으로 제어어휘집을 지시한다. 예컨대, 1999-04-12는 ISO 8601의 규정에 따른 것으로 이를 통해 12월 4일이 아니라 4월 12일을 의미하게 된다. 이 값 한정어는 Dublin core의 'schemes' 개념과 유사하다.

⑥ 매개노드(Intermediate nodes); 그림 7의 중앙의 타원형을 'Intermediate Node'라고 하며, 일련의 관련된 진술을 한자리에 모으는 수단이다. 타원형 내부의 레이블 텍스트 형태와 같이 이름으로 명명된 자원과 달리, 이 매개노드는 감지하거나 식별할 수 있는 자원을 표현한 것이 아니다.

⑦ 값의 구성요소(Value components); 값의 구성요소는 Dublin Core에서 중요한 요소로서, 자원기술의 일부가 한자리에 모이게 되고 기술되는 자원과는 완전히 독립적인 요소의 관점을 나타낸다. 값의 구성요소는 값의 의미특성(이름, 머리색깔, 눈의 색깔, 보험증번호 등)으로서, 이 특성은 기존의 어휘집이나 특정 분야로부터 확보할 수 있다.

(2) 반복(Repetition)

실제 적용과정에서 요소나 값을 반복하거나 집산화할 필요가 있다. RDF는 이러한 집산화를 위해 유용한 구조를 제공한다. 여기에는 다음과 같은 사항이 포함된다.

① Bag; 체계화되지 않은 자원이나 리터럴이다. Bags은 하나의 속성이 다수의 값을 가지고 있고 값의 제시순서에는 아무런 의미도 없음을 선언하기 위해 사용된다.

② Sequence; 자원이나 리터럴의 순서로서, 속성이 다수의 값을 가지고 있고 그 값의 순서가 의미를 가지고 있음을 선언하기 위해 사용된다. 예컨대, 값

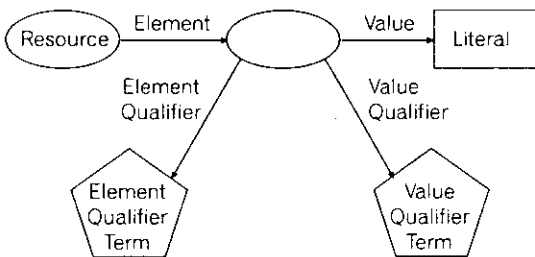


그림 7. Dublin Core의 한정어 모형

의 자모순 체계를 유지할 필요가 있을 때 Sequence의 사용이 가능하다. 복수의 값을 인정한다.

③ **Alternative**; 속성이 지닌 특정 값에 대한 대체 값을 표현한 자원이나 리터럴의 리스트이다. 대체 값은 저작의 표제에 대한 번역표제나, 자원의 인터넷 미리 사이트를 제공하기 위해 사용된다.

이 구조는 요소나 값, 요소한정어, 값 한정어를 반복할 때 사용된다.

(3) XML의 적용예

Dublin Core를 RDF의 XML로 표현하기 위해서는 먼저, Namespace 선언을 통해 Dublin core의 사용을 진술해야 하고, 데이터 요소를 Dublin Core 요소로 대치해야 한다. 그 예는 다음과 같다.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-
    rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.0/" >
<rdf:Description
  rdf:about="http://www.ukoln.ac.uk/metadata/
    resources/dc/datamodel/WD-dc-rdf/" >
<dc:title> Guidance on expressing the Dublin Core
  within the Resource Description Framework (RDF)
</dc:title >
<dc:creator> Eric Miller </dc:creator >
<dc:creator> Paul Miller </dc:creator >
<dc:creator> Dan Brickley </dc:creator >
<dc:subject> Dublin Core; Resource Description Fr-
  amework; RDF; eXtensible Markup Language;
  XML </dc:subject >
<dc:publisher> Dublin Core Metadata Initiative
</dc:publisher >
<dc:contributor> Dublin Core Data Model Working
  Group </dc:contributor >
<dc:date> 1999-07-01 </dc:date >
<dc:format> text/html </dc:format >
<dc:language> en </dc:language >
</rdf:Description >
```

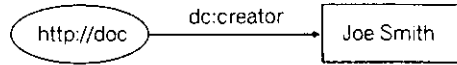


그림 8. 저자의 RDF 표현

</rdf:RDF >

① 그림 8은 "Joe Smith"가 자원 http://doc의 저자임을 제시한 것이다.

이를 RDF 구문으로 표현하면 다음과 같다.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-
  rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.0/" >
<rdf:Description rdf:about="http://doc" >
  <dc:creator> Joe Smith </dc:creator >
</rdf:Description >
</rdf:RDF >
```

② 그림 6을 RDF 구문으로 표현하면 다음과 같다.

```
<?xml version='1.0'?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-
  rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.0/"
  xmlns:dcq="http://purl.org/dc/qualifiers/1.0/" >
<rdf:Description rdf:about = "http://doc" >
  <dc:creator >
  <rdf:Description >
  <rdf:value > Joe Smith </rdf:value >
  <dcq:creatorType > Illustrator </dcq:creator
  Type >
  </rdf:Description >
  </dc:creator >
  </rdf:Description >
</rdf:RDF >
```

③ 그림 9는 "Joe Smith"가 자원(http://doc. Illustrator)의 삽화가이며 Creator이지만, 이 용어는 제어된 namespace로부터 얻은 것이다.

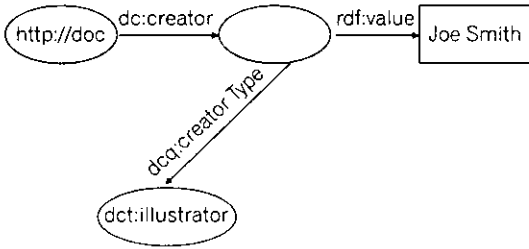


그림 9. 그림에 의한 RDF 표현

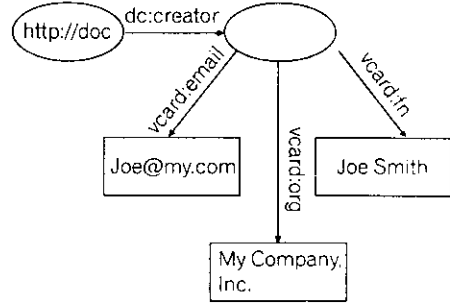


그림 10. RDF 표현

이를 RDF 구문으로 표현하면 다음과 같다.

```
<?xml version='1.0'?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.0/"
xmlns:dcq="http://purl.org/dc/qualifiers/1.0/">
<rdf:Description rdf:about="http://doc">
<dc:creator>
<rdf:Description>
<rdf:value> Joe Smith </rdf:value>
<dcq:creatorType rdf:resource = "http://purl.org/
dc/terms/1.0/creator/type/illustrator"/>
</rdf:Description>
</dc:creator>
</rdf:Description>
</rdf:RDF>
```

④ <그림 10>은 “Joe Smith”가 자원(http://doc)의 저자이고, email 주소(“joe@my.com”), 소속기관(“My Company, Inc.”)이라는 특성을 포함하고 있다. Dublin Core 데이터 모형에서는 Joe Smith가 지닌 이들 특성을 값의 구성요소로 취급한다.

이를 RDF 구문으로 표현하면 다음과 같다.

```
<?xml version='1.0'?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.0/"
xmlns:vcard="http://www.imc.org/vcard/3.0/">
<rdf:Description rdf:about="http://doc">
```

```
<dc:creator>
<rdf:Description>
<vcard:fn> Joe Smith </vcard:fn>
<vcard:email> joe@my.com </vcard:email>
<vcard:org> My Company, Inc. </vcard:org>
</rdf:Description>
</dc:creator>
</rdf:Description>
</rdf:RDF>
```

결론

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 메타데이터 형식이 사용되어 왔으며, 이런 이유로 자원의 접근에 제약을 받아 왔다. 따라서 상이한 메타데이터 간의 연결기법이 중요한 과제가 되어 왔다.

다양한 형식의 전자정보를 이용하기 위해서는 이들 자원의 효과적인 기술수단이 요청된다. XML 구문과 RDF 구조를 통해 Dublin Core 메타데이터는 이들 자원의 처리에 강력한 도구로 기능하게 되었다. Dublin Core는 학제적인 차원에서 다양한 자원을 탐색할 수 있는 수단이 되고 있으며, XML/RDF는 Dublin Core 정보의 분명한 표현구조가 되고 있다. 이 XML은 메타데이터의 의미를 일관되게 표현하고 메타데이터를 교환할 수 있는 공식구문으로 발전될 전망이다.

참 고 문 헌

- 1) 김이경. RDF를 이용한 메타데이터 통합에 관한 연구. 박사학위논문. 연세대학교 대학원 문헌정보학과. 1998.
- 2) 김태수. "디블린코어 메타데이터". 메타데이터의 형식과 구조. 문헌정보처리연구회. 1998.
- 3) 첨단학술정보센터. 디지털 정보표현을 위한 메타데이터 표준개발에 관한 연구. 동센터. 1997.
- 4) 첨단학술정보센터. 메타데이터를 이용한 도서종합목록 구축에 관한 연구. 동센터. 1998.
- 5) Attig, John. "Metadata and cataloging: Supporting common user tasks". 1998. <http://www.ala.org/alcts/organization/ccs/ccda/tf-tei3.html >.
- 6) Bray, Tim, Dave Hollander, Andrew Layman. "Name spaces in XML". W3C Note. 1998. <http://www.w3.org/TR/1998/NOTE-xml-names-0119 >.
- 7) Brickley, Dan, RV. Guha, Andrew Layman. "Resource Description Framework (RDF) Schemas". W3C Working Draft, 9 April 1998. WD-rdf-schema-19980409. <http://www.w3.org/TR/WD-rdf-schema >.
- 8) Guha RV, Tim Bray. "Meta content framework using XML". 1997. <http://www.w3.org/TR/NOTE-MCF-XML/>.
- 9) Hansen, Preben. "User guidelines for Dublin core creation". 1998. <http://www.sics.se/~preben/DC/DC_guide.html >
- 10) Heery, Rachel. "Review of metadata format" *Program*, 1996. 30(4): 345-373.
- 11) Iannella, Renato. "A Idiot's Guide to the Resource Description Framework". 1998. <http://www.dstc.edu.au/RDU/reports/RDF-Idiot/>
- 12) Lassila, Ora. "Introduction to RDF metadata". W3C NOTE. 1997. <http://www.w3.org/TR/NOTE-rdf-simple-intro >
- 13) Miller, Eric. "An introduction to the resource description framework". 1998. <http://www.dlib.org/dlib/may98/miller/05miller.html >
- 14) Miller, Eric, Paul Miller, Dan Brickle. Guidance on expressing the Dublin Core within the Resource Description Framework (RDF). 1999. <http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/WD-dc-rdf-19990701.html >
- 11) Nordic Metadata Project. "Final report". 1998. <http://www.ub.lu.se/metadata/DC_creator.html >
- 12) Raggett, Dave, Dan Connolly, Tim Berners-Lee, Murray Maloney, Liam Quin. 1997. <http://www.w3.org/TR/WD-htmllink-970328 >
- 13) "Resource description framework (RDF) model and syntax". <http://www.w3.org/RDF/Group/WD-rdf-syntax/>
- 14) Weibel, Stuart, Jean Godby, Eric Miller, Ron Daniel. "Dublin core metadata element set: Reference description". 1997. <http://purl.org/metadata/dublin_core_elements >
- 15) W3C RDF Schema Working Group. 1998. <http://www.w3.org/TR/WE-RDF-Schema/>