

메타데이터간의 상호호환성을 위한 구조 연구 : Resource Description Framework를 중심으로

A Study on Architecture for Interoperability Between Metadata Standards : Using RDF

김세정, 한승희, 연세대학교 대학원 문헌정보학과

Kim Sei-Jung, Han Seung-Hee

Department of Library and Information Science, Yonsei University.

웹상에 존재하는 상이한 메타데이터들간의 상호호환성을 위해서는 각각의 메타데이터 형식에서 지정하고 있는 다양한 어의, 구조, 구문을 모두 수용할 수 있는 통합된 구조가 필요하다. 본고에서는 이를 위한 방안으로써 개념적 체계를 제시한 워릭구조와 이의 영향을 받아 W3C에서 제안한 RDF를 중심으로 고찰하며, 더블린코어를 RDF로 구현하고자 하는 방안을 모색하고자 한다.

1. 서론

일반적으로 '데이터에 대한 데이터'라는 의미를 가지고 있는 메타데이터는 네트워크 상의 디지털 형태의 정보 자원의 소재나 내용을 기술하고 네트워크 환경에서 이를 탐색하고 이용하기 위한 데이터 요소들의 집합이다.

인터넷이라는 현재의 정보환경 하에서 정보 자원의 통제 및 탐색 기능을 지원해 주는 메타데이터의 중요성이 커지고 있다. 이러한 메타데이터의 중요성의 인식과 더불어 다양한 형식의 메타데이터가 제작되고 있으며, 그 형식도 단순한 형식으로부터 아주 정교한 기술 수준을 가지는 복잡한 형식에 이르기까지 그 수와 종

류가 매우 다양하다.

특정 정보 자원을 기술하는데 있어서 어느 한 가지 형식의 메타데이터로 기술하기 보다 다양한 유형의 메타데이터들을 이용함으로써 정보 자원의 내용 및 특성을 보다 정확하게 기술할 수 있으며, 디지털 정보 자원을 효율적으로 이용하고 통제하기 위해서 기술적인 메타데이터 뿐만 아니라, 구조적 메타데이터와 관리 메타데이터와의 연계가 필요하다.

이러한 요구들을 만족시키기 위한 방안으로 제2차 더블린 코어(Dublin Core) 워크숍에서는 분산된 메타데이터 모형들간의 상호교환과 통합을 위한 방안으로 워릭 구조(Warwick Framework)를 제시하였고, 제 5차 워크숍에서

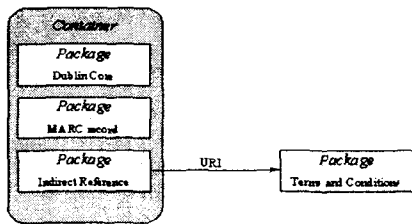
는 워릭 구조를 실질적으로 구현하고자 RDF(Resource Description Framework)의 개발을 제안했다. 워릭 구조가 개념적 수준에서 제시하고 있는 메타데이터의 구조(Framework)라면, RDF는 이를 실제적으로 구현한 것이다.

본고에서는 메타데이터간의 상호호환성을 위한 방안으로써 워릭 구조의 개념과 구체적인 방안으로써 RDF를 고찰하고자 한다.

2. 워릭 구조(Warwick Framework)

워릭 구조는 크게 패키지와 컨테이너 개념으로, 논리적으로 혹은 물리적으로 서로 분산된 메타데이터 패키지들간의 교환과 호환을 위해 컨테이너라는 구조속으로 통합시킨 형태이다.

워릭 구조의 특성은 상이한 메타데이터 형식들을 각각의 패키지로의 모듈화, 새로운 메타데이터 형식으로의 확장성, 외부의 메타데이터 객체로부터 참조되거나 참조할 수 있는 분산성, 패키지가 컨테이너로써 또 다른 패키지들을 수용함으로써 하나의 정보자원이 이와 연관된 또 다른 메타데이터 객체를 가질 수 있는 반복성으로 요약할 수 있다.



<그림 1> 패키지-컨테이너 구조

워릭 구조는 다양한 유형의 메타데이터를 사용하고 정의할 수 있는 폭넓은 구조를 제공하고, 탐색 도구와 에이전트로 하여금 각각의 패키지에 선택적으로 접근하여 처리할 수 있도록 함으로써 확장성과 상호호환성을 증진시킬 수 있는 방안을 제시한 점에서 그 의의를 찾을 수 있다. 하지만 컨테이너속에 각각의 메타데이터 유형을 구분함으로써 의미상 서로 중복되는 부분, 여러 가지 유형의 전송구문 및 패키지들간

의 관계 표현에 있어서 문제점등이 남아 있다.

따라서 이러한 문제점들을 부분적으로 보완할 수 있는 방안으로 RDF를 제시할 수 있다.

RDF에서는 서로 다른 응용분야에서 특정 정보 자원의 기술을 위한 기술요소들을 선택적으로 정의할 수 있고, XML이라는 통일된 구문의 사용하여, 패키지들간의 관계표현을 위해 노드와 아크로 모형화함으로써 컨테이너 안과 밖에서 관련된 다양한 정보 자원들간의 관계를 명시할 수 있다.

3. RDF(Resource Description Framework)

RDF는 상이한 메타데이터간의 어의, 구문 및 구조에 대한 공통적인 규칙을 지원하는 기법을 통해 웹상에 존재하는 기계 해독형(machine-understandable) 정보를 교환하기 위하여 W3C에서 제안한 것으로, 메타데이터간의 효율적인 교환 및 상호호환을 목적으로 한다. 이를 위해서 명확하고 구조화된 의미표현을 제공해 주는 공통의 기술언어로 XML(eXtensible Markup Language)을 사용한다.

RDF는 데이터 모형, 데이터의 상호교환을 위한 구문, 스키마 모형, 기계 해독형 스키마를 위한 구문, 질문과 프로파일 프로토콜과 같은 요소로 구성된다.

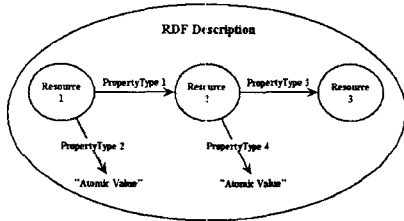
3.1 RDF 데이터 모형(Data Model)

RDF 데이터 모형은 메타데이터의 일관적인 인코딩과 교환 및 작성을 지원하기 위한 표현 모형으로 여러 가지 유형의 메타데이터가 가지고 있는 어의(semantics)를 정의할 수 있도록 함으로써 구조적인 상호호환성을 제공한다.

RDF 데이터 모형은 정보 자원(Resource), 속성 유형(PropertyType), 속성값(Value)으로 구성된다.

RDF데이터 모형에서 기술되는 정보 자원은 그 형태에 관계없이 URI로 식별가능한 모든 객체를 의미하며, 하나의 정보 자원은 여러 개의 속성 유형과 속성값을 가질 수 있다. 속성

유형은 '저자', '서명' 등과 같이 자원의 속성을 적절한 이름으로 표현한 것이며, 속성값은 속성유형에 상응하는 값으로, 문자열이나 숫자 등과 같은 자연어로 상세하게 기술될 수도 있으며, 또다른 정보 자원이 되어 고유의 속성(Property)을 가질 수 있다. 속성이란 정보 자원과 속성 유형, 속성값을 모두 포함한 것으로, 속성 그 자체가 다른 속성의 값이 되기도 하며, 또는 그 자신이 또다른 속성을 가질 수 있다. 이렇게 동일한 정보 자원을 참조하고 있는 속성들의 집합을 기술(Description)이라고 한다. <그림 2>는 RDF 기술(RDF Description)을 RDF 데이터 모형 명세에서 제시하고 있는 노트와 아크로 도식화한 것이다.



<그림 2> RDF 데이터 모형

3.2 RDF 구문(Syntax)

RDF 데이터 모형은 메타데이터를 정의하고 사용하기 위한 추상적이고 개념적인 구조를 제공하는 것이며, 실제로 메타데이터를 교환하고 작성하기 위해서는 구체적인 구문이 필요하다. 이를 위해서 RDF에서는 상호호환성과 확장성, 검증, 인간 및 기계가독성, 웹 상에서의 운용가능성, 그리고 복잡한 문서의 논리적 구조를 표현해 줄 수 있는 장점을 가지는 XML을 사용한다. 다만 RDF는 XML 문서유형정의부(DTD)를 정의하지 않는 유연한 구조를 취하고 있다.

RDF에서는 동일 어의를 가진 여러 표현간의 불명확성을 고려하기 위해 XML의 Namespace 기법을 사용하여 RDF기술 내에서 사용되는 어휘들에 대한 정의를 제공하는 메타데이터 스키

마를 URI로 선언해 줌으로써 새롭게 작성된 메타데이터 어휘를 기계 해독이 가능하도록 처리해준다. Namespace정의는 다음과 같은 형식으로 선언된다.

```
<?xml:namespace name="URI" as="some-abbreviation"?>
```

RDF 구문은 나열형구문(Serialization Syntax)과 축소형구문(Abbreviated Syntax)의 두 가지 형식으로 표현될 수 있는데, 축소형구문은 HTML 문서 내에 RDF 메타데이터를 포함시킬 경우에 사용된다.

3.3 RDF 스키마 (Schema)

RDF 스키마는 특정 메타데이터에서 정의하고 있는 어휘들을 선언하기 위해서 사용된다. 어휘(Vocabularies)란 속성집합으로 자원을 기술하기 위해 각 메타데이터 형식들에서 정의하고 있는 메타데이터 요소집합을 말한다.

인간이 읽을 수 있고(human-readable) 기계 처리가 가능한(machine-processable) 어휘들을 정형화하는 것은 상이한 메타데이터 형식들간의 어휘 확장과 재사용, 상호교환을 가능하게 해주는 것이며, 이러한 정형화를 위한 것이 바로 RDF 스키마이다.

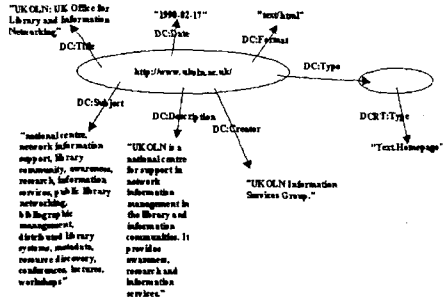
RDF 스키마는 RDF의 데이터 모형과 구문 명세에 의해서 표현된다. 현재 RDF스키마 명세는 개발단계에 있으며, 더 많은 연구가 필요하다. 기본적인 RDF스키마 유형으로는 property, propertyType, instanceOf, subclassOf, allowedPropertyType, Range 가 있다.

3.4 더블린 코어에서의 RDF 적용

더블린 코어 제 5차 워크숍에서 다양한 형식의 메타데이터를 지원하기 위한 구조로 RDF를 채택하였으며, 특히 더블린코어 메타데이터를 RDF로 구현하였다.

RDF를 이용하여 더블린 코어를 기술하기 위해 우선 더블린 코어의 15가지 기술 요소를 더블린코어 스키마로 정의하고, Namespace 기법을 이용하여 더블린 코어를 선언한 후에 실제

정보자원의 기술을 위한 더블린 코어 레코드를 작성한다.



```
<?xml:namespace href="http://www.w3.org/RDF/" as="RDF"?>
<?xml:namespace href="http://purl.oclc.org/RDF/DC/" as="DC"?>
<?xml:namespace href="http://sunsite.berkeley.edu/Metadata/structuralist.html" as="DCRT"?>
<RDF:RDF>
  <RDF:Description RDF:RDF="http://www.ukoln.ac.uk">
    <DC:Title>
      UKOLN: UK Office for Library and Information Networking
    </DC:Title>
    <DC:Creator>
      UKOLN Information Services Group
    </DC:Creator>
    <DC:Subject>
      national centre, network information support, library community, awareness, research,
      information services, public library networking, bibliographic management, distributed
      library systems, metadata, resource discovery, conferences, lectures, workshops
    </DC:Subject>
    <DC:Description>
      UKOLN is a national centre for support in network information management in the
      library and information communities. It provides awareness, research and information
      services
    </DC:Description>
    <DC:Date>
      1998-02-17
    </DC:Date>
    <DC:Format>
      text/html
    </DC:Format>
    <DC:Type>
      <RDF:Description>
        <DCRT:Type>Text:Homepage</DCRT:Type>
      </RDF:Description>
    </DC:Type>
  </RDF:Description>
</RDF:RDF>
```

<그림 3> 더블린 코어에서의 RDF 적용 예
- UKOLN 홈페이지

4. 결론 및 제안

RDF는 상이한 메타데이터 체계를 인정하고 이를 하나의 통합된 구조 속으로 모두 수용할 수 있고, 인터넷상의 메타데이터를 다루는 응용프로그램들간의 상호호환성을 보장한다는 점에서 메타데이터간의 상호호환성을 위한 구조

로써 상당한 발전가능성을 가지고 있다. 하지만, RDF는 아직 제안 수준에 머물러 있으므로 명세의 불완전성, 프로토콜 문제, 더블린 코어 스키마에 대한 RDF 스키마의 효과적인 지원 여부, MARC와 같은 기존의 기술 구조와의 호환성 문제 등이 남아 있다. 또한 RDF 그 자체에는 메타데이터 작성을 위한 특정 어휘들을 정의하고 있지 않으므로 RDF를 위한 표준 어휘를 확정함으로써 이를 바탕으로 여러 메타데이터들과의 상호호환성을 제공할 수 있는 방안을 모색하여야 한다. 이를 위해서는 여러 메타데이터 형식의 표준으로 가장 유력한 더블린코어를 중심으로 더블린코어 스키마 혹은 더블린코어 확장 스키마를 개발하여 공식화하여야 하며, 또한 RDF를 인식할 수 있는 탐색도구나 브라우저의 개발이 함께 이루어져야 한다.

참고 문헌

Brickley, Dan, R. V. Guha and Andrew Layman, Resource Description Framework Schemas, W3C working Draft, Apr 1998.
 <http://www.w3.org/TR/1998/WD-rdf-schema-19980409/>
 Lassila, Ora and Ralph R. Swick, Resource Description Framework Model and Syntax, W3C Working Draft, Feb 1998.
 <http://www.w3.org/TR/WD-rdf-syntax/>
 Miller, Eric, An Introduction to the Resource Description Framework,
 <http://www.dlib.org/dlib/may98/miller/05miller.html>
 Dublin Core in RDF.
 <http://www.ukoln.ac.uk/metadata/resources/rdf/examples/1/>
 Lagoze, Carl, Clifford A. Lynch and Ron Daniel Jr., The Warwick Framework: A Container Architecture for Aggregating Sets of Metadata.
 <http://cs-tr.cs.cornell.edu:80/Dienst/.2.0/Body/>