

메타데이터 연결 및 통합을 위한 체계



인터넷상에 존재하는 네트워크 자원이 급증하면서 이들 자원을 식별·기술하는 메타데이터도 여러 가지 다른 형식으로 개발 및 이용되고 있다. 따라서 독립적으로 운용되는 메타데이터간의 상호호환성을 높이기 위한 메타데이터의 연결이나 통합 구조가 요구된다. 이러한 통합 구조의 논의는 DC의 제2차 워크숍에서 제안된 WF(Warwick Framework)에서 시작되어 RDF(Resource Description Framework)로 이어지고 있다.

■ 이재진 / 한국데이터베이스진흥센터 정책연구과

연재순서

- 1 메타데이터의 개요
- 2 DC(Dublin Core)
- 3 GILS(Government Information Locator Service)
- 4 IAFA Templates
- 5 MARC
- 6 PICS(Platform for Internet Content Selection)
- 7 RFC 1807
- 8 SOIF
- 9 TEI header
- 10 URx
- 11 Warwick Framework RDF(Resource Description Framework) - 이번호
- 12 메타데이터 향후 방향

개요

네트워크 자원의 식별·기술 수단인 메타데이터는 대상 자원의 유형이나 특징에 따라 서로 다른 메타데이터가 적용될 수 있으며 따라서 다양한 메타데이터가 개발되어 이용되고 있다. 이러한 다양한 메타데이터의 이용은 자원의 효과적인 기술 수단으로는 유효하지만 메타데이터간의 상호호환이나 특정 메타데이터에서 수용하지 못하는 점을 수용하기 위한 데이터 공유 등에 있어서 불합리한 문제점이 발생된다. 따라서 DC(Dublin Core) 메타데이터 워크숍에서는 이에 대한 문제제기와 해결방안을 위한 논의가 진행되었고 그 결과 1996년, 영국 워릭(Warwick)에서 열린 2차 메타데이터 워크숍에서 개별적인 메타데이터의 패키지를 논리적, 물리적으로 모으기 위한 메커니즘으로서 WF(Warwick Framework)를 제안하였다. WF는 DC 메타데이터 요소 집합의 확장 메커니즘으로 제안된 것으로 이후 지속적으로 연구·개발되고 있는 메타데이터의 통합 및 연결을 위한 구조의 기본 개념이 되었다.

WF의 시도를 통해 이를 실질적으로 구현하여 이용할 수

있는 발전적인 구조로서 제안된 것이 RDF(Resource Description Framework)이다. RDF는 W3C에서 개발한 것으로서 구조화된 메타데이터의 인코딩, 교환, 재사용을 가능하게 해주는 기반 구조이다.

RDF는 다양한 메타데이터의 구문(syntax)과 스키마 정의 언어의 호환성을 높이기 위해 구문 명세(Syntax specification)와 스키마 명세(Schema specification)를 제공한다.

WF(Warwick Framework)

WF는 다양한 메타데이터 집합을 모으기 위한 아키텍처로서 크게 컨테이너(container)와 패키지(package)의 2가지 요소로 구성된다. 컨테이너는 유형별 메타데이터 집합을 모으는 단위로서 이들 메타데이터 집합이 패키지가 된다.

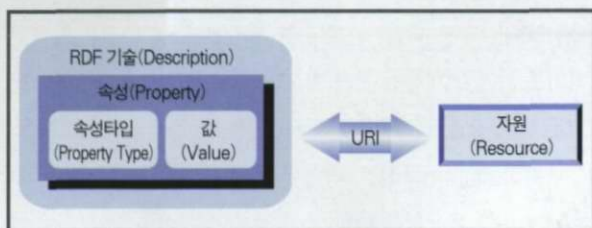
WF의 패키지는 세가지 유형으로 정의된다.

첫번째는 '메타데이터 집합(metadata set)'으로서 이는 실제 메타데이터를 포함한다. MARC나 Dublin Core 레코드, 개정 내역, 디지털 서명, 접근을 위한 용어나 조건 등이 그 예이다. 두번째는 '간접 참조(indirect)'로서 컨테이너에 포함되지 않고 외부의 다른 객체로 연결되는 패키지로써 흔히 URL을 통해 참조된다. 세번째는 '컨테이너'로서 컨테이너 자체가 패키지가 되는 경우이다. <그림 1>은 이러한 WF의 간단한 컨테이너 구조를 나타낸 것으로 메타데이터인 DC와 MARC 레코드는 패키지로써 컨테이너 안에 물리적으로 존재한다. 다른 메타데이터 패키지는 접근 조건이나 규정에 관한 메타데이터로서 컨테이너 외부에 존재하며 URI에 의해 간접 참조된다.

WF는 다양한 유형의 메타데이터를 함께 사용하여 네트워크 자원을 탐색해야 하는 탐색도구가 가장 적합한 메타데이터를 선택하여 이용함으로써 확장성이나 메타데이터간의 상호 호환성을 높일 수 있다. 그렇지만 다양한 메타데이터 집합간의 의미적인 충돌이나 중첩이 발생될 수 있다는 점, 패키지에 사용되는 메타데이터 집합에 대한 정보가 미리 등록되어 있어야 하는 점, 데이터 인코딩을 위한 일관된 구문 사용이 불가능하다는 점, WF의 강력한 회기성이나 분산성의 특징이 오히려 실제 시스템 구현시 효율성을 방해하게 된다는 점, 그리고 컨테이너나 패키지의 교환과 검색이 가능하도록 하는 프로토콜이 개발되어야 한다는 점 등에서 제약을 받아왔다. 따라서 WF의 기본 개념을 이어받아 이를 실제로 시스템 상에서 구현



<그림 1> Simple Warwick Framework Container



<그림 2> RDF Data Model

하기 위한 RDF가 새롭게 제안되었다.

RDF(Resource Description Framework)

RDF는 메타데이터를 처리하기 위한 기반으로써 네트워크 상에서 정보자원을 교환하는 응용간의 상호 운용성을 제공한다. RDF 메타데이터는 자원 탐색, 특정 웹사이트나 페이지에서 이용가능한 자원과 자원간의 관계 기술, 내용 등급 체계 등의 여러 가지 분야에서 활용할 수 있다. RDF는 명확한 의미 표현을 위해 XML(Extensible Markup Language)을 이용하며, 추가적으로 분산된 정보 환경에서 메타데이터 의미의 확장과 재사용이 가능하도록 다양한 어휘를 사용할 수 있는 수단도 제공한다. XML을 이용하는 RDF의 구조적인 제약은 일관된 코딩을 지원하고 표준화된 메타데이터를 교환으로 서로 다른 자원 기술 환경에서 정의된 독립된 메타데이터 패키지의 상호호환, 교환을 가능하게 한다. 인터넷상의 지식 표현 메커니즘인 메타데이터 프레임워크로서 RDF는 최초의 메타데이터 연구 과제였던 PICS(Platform for Internet Content Selection)와 DC, WF 등의 영향을 받아 설계, 개발되었다.

(1) RDF 데이터 모델

RDF는 자원을 기술하기 위한 데이터 모델을 제공한다. 기본적인 개념은 자원(resource)이 RDF 기술(description)이라

불리는 속성(property) 집합을 통해 기술되며, 이들 각 속성은 속성유형(property type)과 값(value)을 갖는다. URI로 식별가능한 모든 자원은 RDF로 기술될 수 있다. 이러한 RDF 데이터 모델은 <그림 2>와 같이 나타낼 수 있다.

이와 같은 RDF 데이터 모델의 3가지 구성요소를 구체적으로 살펴보면 다음과 같다.

- 자원(resources) : RDF 표현으로 기술되는 대상. 자원은 웹페이지 전체, 웹페이지의 일부, 웹사이트, 혹은 웹을 통해 직접 접근이 불가능한 것일 수도 있다.

- 속성(properties) : 특성은 자원을 기술하는데 사용되는 특정 측면, 특징, 속성, 관계 등을 가리킨다. 각 특성은 특정한 의미를 가지며 제한된 값과 기술할 수 있는 자원의 타입, 그리고 다른 특성과의 관계를 정의한다.

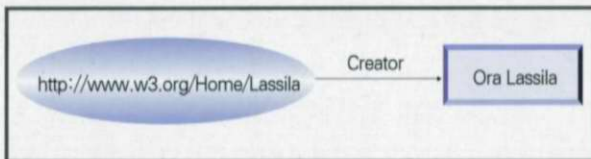
- 문장(statements) : 명명된 특성을 가진 특정 자원과 그 자원에 대한 특성값을 더한 것이 RDF 문장이다. 문장은 주제(subject), 속성(predicate), 객체(object)로 구성된다.

다음과 같은 문장을 예로 들어보면,

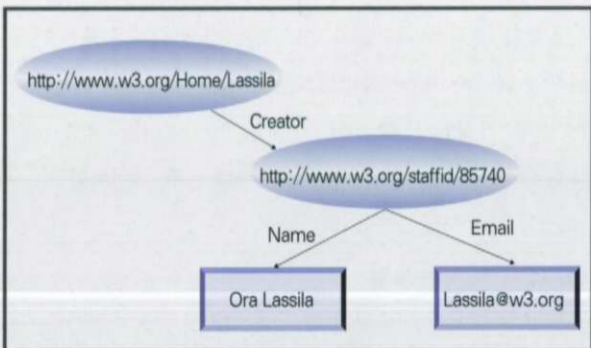
Ora Lassila is the creator of the resource
<http://www.w3.org/Home/Lassila>

이 문장은 다음과 같이 분석할 수 있다.

주제(Resource)	http://www.w3.org/Home/Lassila
속성(Property)	Creator
객체(Literal)	"Ora Lassila"



<그림 3> Node & Arc Diagram



<그림 4> Structured value with identifier

RDF 문장은 노드와 아크로 표현하는 다이어그램으로 그릴 수 있으며 노드는 자원을, 아크는 명명된 속성을 표현하고, 문자열로 나타나는 객체 노드는 직사각형으로 나타낸다. 위의 예시 문장을 간단한 노드와 아크 다이어그램으로 표기하면 <그림 3>과 같다.

이 때, 속성 유형인 Creator 속성은 Name이나 Email과 같은 다른 속성과 연결될 수 있는데 이 경우 속성은 구조화된 값을 갖게 된다. 흔히 다른 속성으로의 연결은 이들 속성을 포함하고 있는 다른 자원을 통해 연결된다. 구조화된 값을 갖는 예를 다이어그램으로 표현해 보면 <그림 4>와 같다.

(2) RDF 구문

RDF의 자원 기술 모델을 표현하는 구문(syntax)은 데이터 모델을 통해 추출된 추상적이고 개념적인 구조를 실제 기계가 독 파일로 변환하여 저장하고 여러 응용에서 상호 교환할 수 있도록 작성하기 위한 수단으로서, XML을 채택하고 있다. 또한 RDF는 XML의 명명체계(namespace) 기능을 사용하여 속성을 정의하는 메타데이터 스키마를 URI로 선언하여 각 속성간의 관계를 정확하게 연결하도록 하고 있다.

보고서를 기술한 RDF 구문의 예는 다음과 같다.

```
<? xml version="1.0" ?>
(RDF xmlns = "http://w3.org/TR/1999/PR-rdf-syntax-19990105#"
  xmlns:DC = "http://purl.org/DC#"
  xmlns:AGLS = "http://naa.gov.au/AGLS#" )

<Description about = "http://dstc.com.au/report.html" >
  <DC:Title> The Future of Metadata </DC:Title>
  <DC:Creator> Jacky Crystal </DC:Creator>
  <DC:Date> 1998-01-01 </DC:Date>
  <DC:Subject> Metadata, RDF, Dublin Core </DC:Subject>
  <AGLS:Function> Information Management - Internet
</AGLS:Function>
  </Description>
</RDF>
```

RDF 구문은 크게 나열형(Serialization)과 축소형(Abbreviated)으로 구분할 수 있는데 나열형 구문은 하나의 자원에 대한 여러 가지 속성을 한꺼번에 모아서 Description 엘리먼트로 기술하는 것으로 RDF 모델의 구조를 가장 분명하게 나타낼 수 있다. 반면 축소형 구문은 XML DTD를 따르는 문서가 직접 RDF 모델로 변환할 수 있도록 한 것으로 XML 태그안에 모든 속성을 기술한다. 앞서 제시한 예는 나

열형 구문으로 표기한다.

(3) RDF 스키마

RDF는 자원의 속성에 대해 선언하거나 이들 속성과 다른 자원간의 관계를 정의하기 위한 메커니즘으로 RDF 스키마를 사용한다. RDF 스키마는 속성 집합의 의미, 특징, 관계를 정의하는 것으로서 값에 대한 제한이나 다른 스키마로부터의 속성 계승이 가능하다. 이러한 스키마는 특정 RDF 어휘(vocabulary)로 RDF 자원을 기술하는데 사용되며 핵심이 되는 스키마 어휘는 'rdfs'로 표기한다.

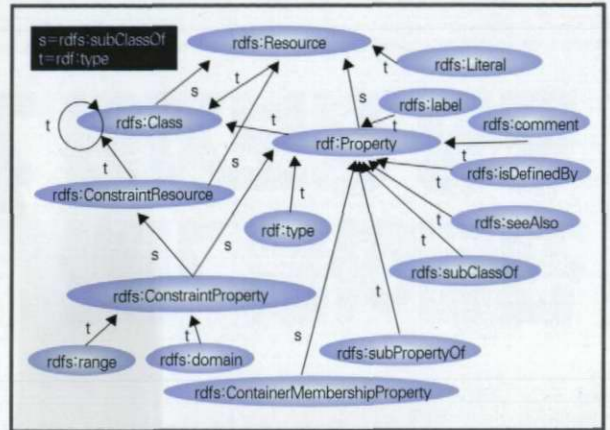
1999년 3월에 제안된 RDF 스키마 명세에서는 자원을 하나 이상의 클래스의 인스턴스로 보고 핵심 클래스로서 RDF 표현으로 기술되는 모든 자원을 포함하는 'rdfs:Resource', RDF 자원의 하부집합으로 표현되는 속성을 나타내는 'rdfs:Property', 그리고 유형이나 범주의 개념으로서 새로운 클래스가 정의되었을 때 갖게 되는 기본값으로서의 'rdfs:Class'를 지정하고 있다. 또한 핵심 속성으로는 특정 클래스의 인스턴스로 식별할 수 있는 'rdf:type'과 클래스간의 상하위 관계 속성을 나타내는 'rdfs:subClassOf'를 설명하고 있다. (그림 5)는 이러한 RDF 스키마의 클래스 계층을 그림으로 나타낸 것이다.

RDF의 응용 사례

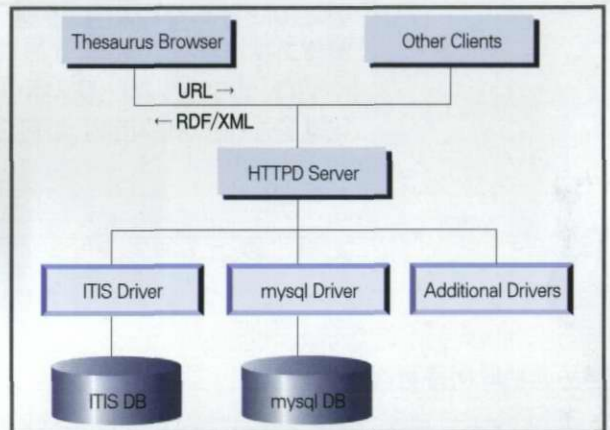
다양한 메타데이터의 연결과 유연한 데이터 교환, 그리고 통합된 구조를 제공하는 RDF는 최근들어 구문이나 스키마 명세가 발표되는 등 지속적인 연구가 수행되고 있는 새로운 시도로서 각종 프로젝트에서 시험적으로 적용되고 있다.

이러한 프로젝트 가운데 미국 캘리포니아 자원국(California Resources Agency)에서 개발하고 있는 캘리포니아 환경자원 평가시스템(California Environmental Resources Evaluation System : CERES)과 국가 생물정보 기반구조(National Biological Information Infrastructure : NBII)의 생물자원부(Biological Resources Division : BRD)는 공동으로 통합 환경 시소러스와 메타데이터 개발과 키워드 탐색을 위한 도구를 개발하고 있다.(<http://ceres.ca.gov/thesaurus/>)

이 시소러스는 분산된 클라이언트와 서버 환경에서 운용되며 HTTPD와 RDF/XML에 기반한 프로토콜을 사용하고 있



(그림 5) Class Hierarchy for the RDF Schema



(그림 6) CERES/NBII Thesaurus Project

다. 즉, HTTPD 서버를 통해 질의가 전달되며 서버와 클라이언트간에 전달되는 질의 결과는 RDF/XML 문서 포맷을 사용한다. (그림 6)은 RDF 포맷을 사용한 CERES/NBII 시소러스 프로젝트의 운영 체계를 나타낸 것이다.

결론

최근의 메타데이터에 관한 논의는 공통된 구문과 구조적 프레임워크 내에서 다양한 메타데이터를 지원할 수 있는 체계의 개발로 귀결되고 있다. 이러한 추세를 반영하여 WF에서 시작된 메타데이터의 통합 논의는 RDF로 이어져 활발하게 연구되고 있다.

RDF의 발전가능성은 메타데이터의 생성과 관리의 통합기능을 제공하는 종합적인 도구의 개발에서 살펴볼 수 있으며 더욱 다양한 메타데이터를 포괄하는 방향으로 진행될 것이다. 그러나 다양한 메타데이터 집합에 깔려 있는 개별 데이터 모델의 공통된 면을 식별하고 이들을 조화시키는 것은 여전히 RDF가 풀어나가야 할 문제이다.