

Metadata관련 사례 발표

-서울대학교 음악메타데이터¹⁾를 중심으로-

서울대학교 도서관

최유미

< 목 차 >

I. 서울대학교 음악메타데이터 개발의 필요성

II. 선행연구(IFLA FRBR 데이터 모델 및 Indiana 대학 DML 데이터 모델)

III. 서울대학교 음악메타데이터 개발 사례

IV. 향후 과제

I. 서울대학교 음악메타데이터 개발의 필요성

메타데이터는 데이터에 관한 데이터 (Data about data)로 정의할 수 있으며 일반적으로 다음의 세가지 종류로 구분할 수 있다.

(1) 기술 메타데이터(Descriptive Metadata)
자원의 발견(검색/브라우징) 및 확인을 위하여 자원의 특성을 기술하는 메타데이터

(2) 구조 메타데이터(Structural Metadata)
기술된 자원이 구조를 지닐 때 자원의 내부구조에 대한 정보를 기술하

여, 각 구성요소로의 자유로운 항해와 디스플레이를 지원

(3) 관리 메타데이터(Administrative Metadata)

파일 포맷, 인코딩 장비, 파일 생성일, 파일 제작자 등 디지털 자원의 생성시 발생하는 정보 및 저작권 관리에 필요한 정보로, 디지털화된 전자 자원의 관리에 필요한 항목에 대한 메타데이터

서울대학교 도서관 및 대다수의 도서관에서 정보자원관리를 위하여 국제적 표준으로 사용하고 있는 메타데이터인 MARC는 기계가 읽고 조작할 수 있는 방식으로서 정보자원에 관한 서지정보를 기술하는 국제적인 표준으로 자리매김해왔다.

MARC은 서지자료를 표현하는데는 매우 유용한 메타데이터이지만 구조적 경직성으로 인해 다양하고 상이한 기술적 특성을 지니는 정보자원들을 적절히 기술하는데는 제약이 따른다[6]. 예를 들어, 음

1) 서울대학교 음악메타데이터 연구는 “서울대학교 전자도서관 구축을 위한 메타데이터 연구”의 한 부분으로 오삼균 교수(성균관대학교 문헌정보학과)와 채진석 교수(인천대학교 컴퓨터공학과)와 서울대학교 중앙도서관과 공동으로 연구되었다.

<표 1> IFLA 모델 엔티티 설명

엔티티	개념	예	상태
Work	원 창작자에 의해 창작된 “작품자체”	개념, 사상, 아이디어	추상적. “착상/구상하다”
Expression	작품을 표현 혹은 실행하는 행위	행동	특정공간에서 일시적으로 존재함. “하다”
Manifestation	Expression이 구체화되어 물리적인 매체로 수록	물리적 최소 단위	실질적인 물리적 형태를 갖고 존재함. “만들다”
Item(copy)	Manifestation의 개개의 카피본		

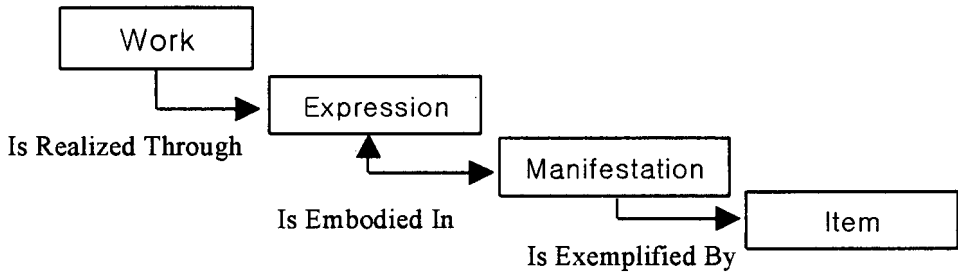
악분야의 정보자원의 특성을 표현하는데 있어서는 MARC과 같이 자원기술의 대상을 단일의 객체로 파악하는 단층 데이터 모델보다는 특정 음악작품에서 파생된 다양한 음악가의 버전간의 다층적인 연결관계의 표현 할 수 있는 데이터 모델이 더 적합할 것이다.

따라서, 서울대학교에서는 MARC을 적용할 수 있는 정보자원에는 MARC을 적용하고, MARC을 적용하였을 때, 정보자원의 특성이 제대로 표현되지 않는 정보자원을 대상으로 새로운 메타데이터를 개발한다는 정책을 수립하여, 새로운 메타데이터를 개발하였으며, 본 고에서는 그 일례로 서울대학교 음악메타데이터에 대하여 설명하고자 한다.

서울대학교 음악메타데이터 연구는 지식자원의 특성을 잘 표현할 수 있는 단일한 국제적 메타데이터 표준이 존재하지 않기 때문에, 선진사례조사를 통하여 다양한 자원을 기술하기 위한 메타데이터 표준들을 참조하여 진행되었다.

음악자료는 다른 자료와는 달리, 원작품

(Work)에 대한 정보뿐만 아니라, 이 작품이 후세에 연주되었을 때, 해당 연주에 관한 정보 및 이러한 연주곡이 수록된 물리적인 매체에 대한 정보 모두 중요하다는 특징이 있다. 따라서, 각 버전간의 입체적인 연결관계가 존재하는 음악자료를 기존의 MARC레코드에서 물리적인 매체위주로 단순화하여 표현하는 것은 음악자료의 특성을 제대로 표현할 수 없다는 한계점을 지닌다고 할 수 있다. 따라서, MARC 데이터 모델의 한계점을 극복함과 동시에 음악이라는 주제분야의 특성을 표현하기 위하여, INDECS(Interoperability of Data in E-Commerce System)와 같은 전자상거래 분야에서 활발하게 사용되고 있는 IFLA의 FRBR(Functional Requirements for Bibliographic Records)과 Indiana 대학의 디지털 음악 도서관(DML : Digital Music Library)모델을 적용하였고, 데이터의 호환성을 감안하여 더블링크어(Dublin Core)의 요소를 기반으로 하여 음악분야의 다양한 자원간의 연계관계를 효율적으로 표현할 수 있도록 하였다.



<그림 1 >

다음 절에서는 음악메타데이터 연구에 참조한 IFLA FRBR모델과 Indiana 대학의 DML 모델에 대하여 간단히 설명하도록 하겠다.

II. 선행연구(IFLA FRBR 데이터 모델 및 Indiana 대학 DML 데이터 모델)

IFLA FRBR 모델에서는 정보자원을 다음의 4가지 엔티티 중의 하나로 규정하고 있다[2][4].

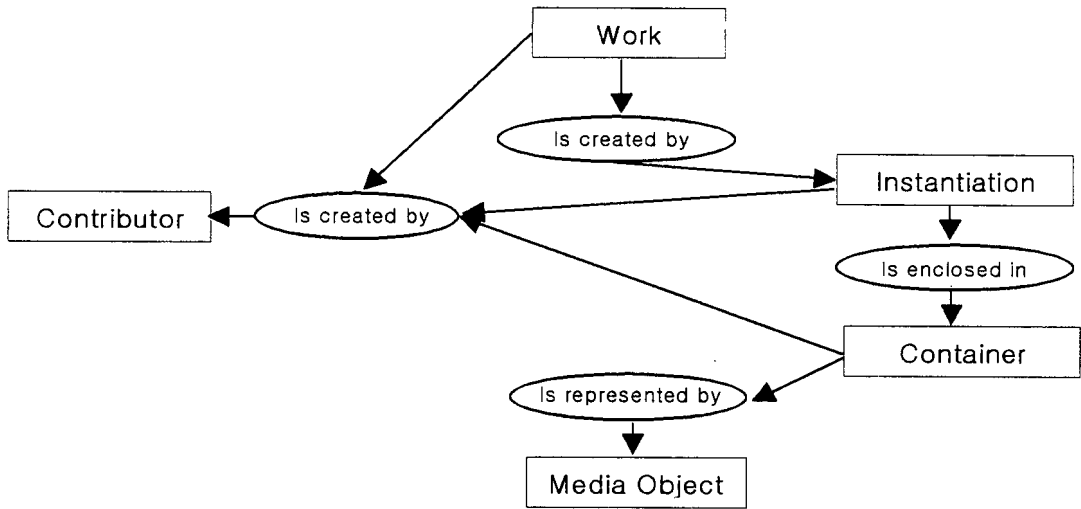
각 엔티티간의 관계는 <그림 1>과 같이 같이 표현할 수 있다(카디널리티(Cardinality)는 화살표로 표현됨).

이 데이터모델을 베토벤의 5번 운명 교향곡에 적용하면, 베토벤이 어떤 아이디어에 의하여 작곡한 5번 교향곡 자체는 Work(작품)에 해당하며, 이 베토벤의 작품은 다양한 음악가별로 연주라는 행위를 통하여 Express(표현)되어 특정 공간과 일정한 시간에서만 존재한다. 각 연주를 녹화한다거나, 녹음하여 비디오나, CD등

의 매체에 수록하는 작업을 통하여 물리적인 형태로 Manifestation(실현)이 되며, 음반회사에서 CD라는 매체로 만들어진 제품을 대량으로 생산하였을 때, 개개의 CD를 Item(항목)이라 한다. 궁극적으로 도서관에서는 이 Item을 대상으로 정리/대출/반납 등의 업무를 수행하게 된다.

Indiana 대학은 이러한 IFLA 데이터 모델을 수용하여 다음과 같은 서비스를 제공하고자, 구조 메타데이터를 적용하여 DML 모델을 개발하였다[5].

- 1) 레코드의 트랙이나 악보의 섹션(Section)간의 이동과 같이 특정 위치에 저장된 사운드 파일 혹은 악보 이미지 파일 등에 계층적 구조로 저장된 각 항목간의 항해(navigation) 서비스
- 2) 동일한 작품에 대한 사운드 파일, 악보 이미지, 해제 파일 등과 같은 다양한 표현(Expression 혹은 Instantiation)과의 관계에 대한 서비스 제공. 즉, 이용자가 특정 작품의 한 연주버전을 검색하였을 경우, 동일 작품에 대한 다른 연주버전과의 연결서비스를 제공.



<그림 2>

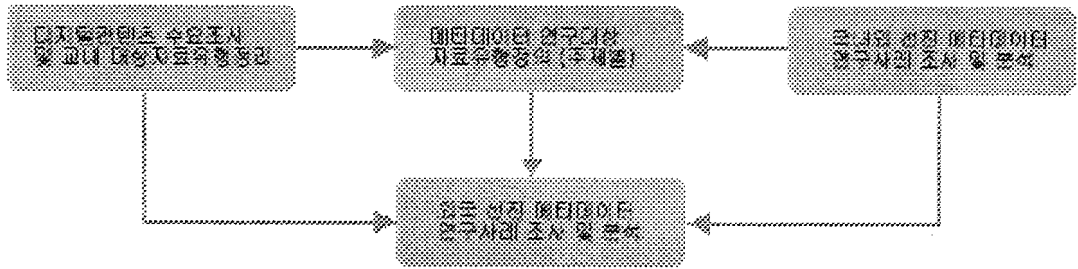
- 3) 한 작품의 특정 섹션과 이 작품을 표현한 여러 표현물(Instantiation)의 동일 섹션과의 연결 서비스 (예: 모차르트 40번 교향곡의 2악장과 다양한 음악가에 의해 표현된 모차르트 40번 교향곡 2악장과 연결 서비스).
- 4) 한 작품과 이 작품을 표현한 여러 표현물과 연결 서비스 (예 : 모차르트 40번 교향곡과 이 작품에 대한 다양한 연주나, 악보 등과의 연결 서비스)

Indiana 대학의 DML 데이터 모델은 IFLA FRBR 모델과 유사하지만 <그림 2>에서와 같이 Contribution이라는 메타데이터 셋을 통하여, 도서관의 전거통제 역할 및 작곡/작사가 등의 정보를 제공할 수 있도록 하였고, IFLA Model의 Expression과 Manifestation은 이 모델에서는 각각 Instantiation과 Container에 해당한다. Indiana 대학의 DML의 궁극적 목적은 디지털

탈화된 음악관련 파일을 이용자에게 직접 서비스하는 것이므로, 디지털 자원에 대한 내용을 기술해주는 Media Object라는 엔티티가 존재한다.

인디애나 대학의 구조 메타데이터는 작품구조(Work Structure)와 일반 도서의 목차정보와 유사한 매체구조 (Container Structure)를 기술하도록 하고 있으며, 이러한 작품이나 수록매체의 구조의 각 부분에 해당하는 디지털파일(Media Object)을 바로 연결할 수 있도록 하여, 이용자가 작품전체나 수록매체 전체를 처음부터 끝까지 이용하는 것이 아닌, 원하는 부분만 선택하여 이용이 가능한 서비스 제공을 염두에 두고 있다.

Ⅲ. 서울대학교 음악메타데이터 개발 사례



<그림 3>

서울대학교 음악메타데이터 개발은 <그림 3>과 같이 학내 콘텐츠 구축 수요조사를 통하여 서울대학교에서 보유하고 있는 자원의 성격을 분석한 후, 새로운 메타데이터의 개발이 요구되는 주제분야의 정보 자원에 대하여, 국내의 선행연구 조사분석 및 서울대학교 음악대학의 음악전문가의 의견을 참조하여 진행되었다.

메타데이터 셋의 저장 및 교환을 위한 방안으로, 국제적으로 데이터 전송의 표준 포맷으로 인정받은 XML을 활용하며, 메타데이터 교환시 데이터의 무결성을 보장하기 위하여, XML Schema 기반으로 구현하기로 하였다. XML Schema는 문서중심(Document Centric)의 W3C의 표준을 따르지 않는 XML DTD와는 달리, 데이터 중심(Data Centric)이고, Schema 자체가 W3C의 표준을 따르고, DTD에서는 제공하지 않는 namespace지원 및 다양한 데이터타입을 지원한다는 특징을 지니고 있어[1], 본 연구에 채택되었다.

서울대학교 음악 메타데이터는 이용자 서비스와 관리자의 관리 측면을 고려하여 개발되었다. 우선 이용자 서비스 측면에서는 다음의 조건을 고려하여 이를 충족시킬 수 있는 메타데이터가 설계되었다.

1) 서양음악 및 국악 작품(Work)의 구조 즉, 섹션(Section)정보를 표현할 수 있어야 한다.

작품 및 작품의 구조정보는 고유한(Unique)것으로, 작품구조는 이 작품이 다양한 연주 및 악보 등으로 표현되는 부분(part)에 대한 정보이다. 즉, 베토벤 운명교향곡의 1악장 Allegro는 어느 누가 어떠한 기법을 사용하여 연주하더라도, 1악장을 연주했다는 사실은 변치 않는 사실이 될 것이다.

예) 베토벤의 운명 교향곡은 4악장으로 구성되어 있으므로, 이 곡은 각각 악장이라는 섹션으로 구성되며, 이러한 구성정보가 작품의 구조정보가 된다.

2) 전자도서관에서의 통합검색 시, 검색된 작품과 관련한 다른 작품 및 작품에서 파생된 다양한 버전(Instantiation)과 수록매체(Container)와의 모든 관계정보를 표현할 수 있어야 한다.

예) 이용자가 “베를린 필하모니

오케스트라가 연주한 베토벤 운명교향곡”을 검색했을 때, 검색의 상세서지에서는 연주에 대한 정보가 제공되며 또한, 다음과 같이 관련된 정보로 자유롭게 네비게이션 할 수 있다.

- ① 베토벤 운명 교향곡의 작품설명(Work)의 상세서지 레코드로 이동.
- ② 다른 연주자나 음악가가 연주 혹은 편집한 베토벤 운명 교향곡의 연주 혹은 악보의 상세서지 레코드로의 이동.
- ③ 베토벤 운명 교향곡의 각 악장 중 재생을 원하는 악장(Section)을 선택했을 때, 원하는 악장에 대한 음악가별 연주나 악보정보가 디스플레이 됨.
- ④ 이 베를린 필하모니 오케스트라의 연주곡을 수록하고 있는 매체(Container)의 상세서지 레코드로의 이동. 실황녹음테이프, CD, 및 연주회 비디오 등과 같이 수록매체의 종류별로 이용자가 선택하여 이동.

3) 특정 작품의 버전(Instantiation)을 디지털화한 디지털 자원은 검색의 Instantiation 상세서지에서 바로 연결 가능 하여야 한다.

예) 2)의 ③에서 해당 악장의 특정 연주자 버전(Instantiation)을 선택하면, 오디오 파일, 동영상 파일, PDF(악보의 경우) 등과 같이 이용을 원하는 디지털 파일로 이동.

4) 매체(Container) 검색 시, 매체의 상

서서지에서 매체의 구조정보(Container Structure)제공 및 매체에 속한 각 항목별로 멀티미디어 파일의 재생이 가능해야함.

예) “클래식 명반”이라는 음악테잎에 대한 상세서지 레코드에서는 매체의 물리적 특징 및 음반사 등에 대한 정보가 제공되며, 다음과 같은 서비스를 제공한다.

- ① “클래식 명반”의 구조정보(Container Structure)를 제공한다. 즉, Side A라는 항목(item) 아래 수록된 곡(div)의 리스트와 Side B라는 항목아래 수록된 곡의 리스트를 제공.
- ② “클래식 명반”이라는 매체에 수록된 곡 중 원하는 곡을 선택한 후, 이 곡에 대한 멀티미디어 파일 중 원하는 형태의 파일을 선택하여 이동.
- ③ 각 곡별 작품설명(Work) 상세서지 레코드로의 이동.
- ④ 각 곡을 다양한 음악가가 연주나 악보로 표현하였을 경우, 이러한 정보가 디스플레이 되어, 이용자가 이 곡에 대한 다른 음악가의 연주정보(Instantiation) 상세서지 레코드로 이동할 수 있도록 서비스 제공.

5) 음악분야 학술 연구자가 간편하고 신속하게 연구 자료를 수집할 수 있어야 한다.

지금까지 상술한 기능을 위주로 설계된

서울대학교 전자도서관 음악메타데이터 모델의 ER-Diagram 및 이용자 서비스 흐름은 별첨 <그림 4>, <그림 5>, <그림 6>과 같다[6].

이용자 서비스 측면이외에, 전자도서관의 다양한 멀티미디어 파일의 관리를 위하여, 관리메타데이터의 개발 역시 간과할 수 없는 부분이다.

기술메타데이터와 이 기술메타데이터의 구조를 표현하는 구조메타데이터와 더불어, 디지털이징한 멀티미디어 파일에 대한 관리정보 역시 전자도서관에서 중요한 요소라 하겠다. 즉, 작품 구조 및 매체구조에 따라 부분 및 전체 재생이 가능하려면, 부분재생에 대하여 타임코드별로 파일이 관리가 되거나 각 부분을 파일 그룹단위로 관리해야 할 경우가 발생하며, 각 파일에 대한 접근권한 등에 대한 저작권관리 정보 역시 필요하다. 또한, 디지털 자원 파일에 문제가 발생하였을 때, 이에 대해 관리자가 원활하게 대처하기 위해서는 해상도, 밝기, 타임코드, 파일 작성자, 최종 수정일 등 디지털화된 파일의 물리적인 정보 및 파일 연계 정보 등이 별도로 관리되어야 한다.

서울대학교 전자도서관의 음악 메타데이터 및 멀티미디어 관리 메타데이터는 <부록 1>, <부록 2>에 수록하였다.

IV. 향후 과제

지금까지 살펴본 서울대학교 전자도서관의 음악메타데이터는 현재까지, 메타데이터의 기본 요소 및 하위 요소와 이에

대한 일부 인코딩 스킴(encoding scheme)에 대하여 정의하고, 또한 XML Schema로 구현하였다. 그러나, 서울대학교 음악대학의 음악전문가의 협조를 받아 현재 조사중에 있는 국악 및 서양음악 관련 섹션타입(Section Type), 악기종류, 음악작품형식 등에 대한 인코딩 스킴이 확정되어, 검색제한자로서 서비스되어야 한다.

또한, 서울대학교 메타데이터를 네임 스페이스로 관리하기로 하였으나, 현재까지 서울대학교에서 자체 정의한 요소에 대한 네임스페이스 정의 및 이에 대한 XML Schema 구현이 아직까지는 완료되지 않아, 이 역시 향후 진행되어야 할 과제로 남아 있다.

마지막으로, XML Schema로 메타데이터를 구현한 기본 목적인 상호운영성, 즉, 외부기관과의 자유로운 자료반입/반출을 고려하여 서울대학교 메타데이터를 더블린 코어의 15개 요소를 기반으로 개발하였다. 그러나, 서울대학교가 보유한 자료의 특성을 반영하기 위해서, 음악분야에서 요구하는 요소의 추가는 필수적인 것이라 할 수 있다. 그러나, 이러한 추가요소들은 표준안이 아니므로, 외부 데이터 반출/반입시에는 서울대학교가 자료의 특성을 기술하기 위하여 추가한 요소는 무시되고, 더블린 코어의 15개 요소에 매핑되어 이루어 질 수 밖에 없다는 문제가 발생하며, 그 만큼 자원에 대한 기술이 일반화된다는 문제가 있다. 이러한 문제가 시사하는 바는 각 주제분야별 특성을 표현할 수 있는 메타데이터 요소가 개발될 때에는 동일한 주제분야의 자료를 보유하고 있는 여러 기관이 다각적인 분석을 통하여, 메

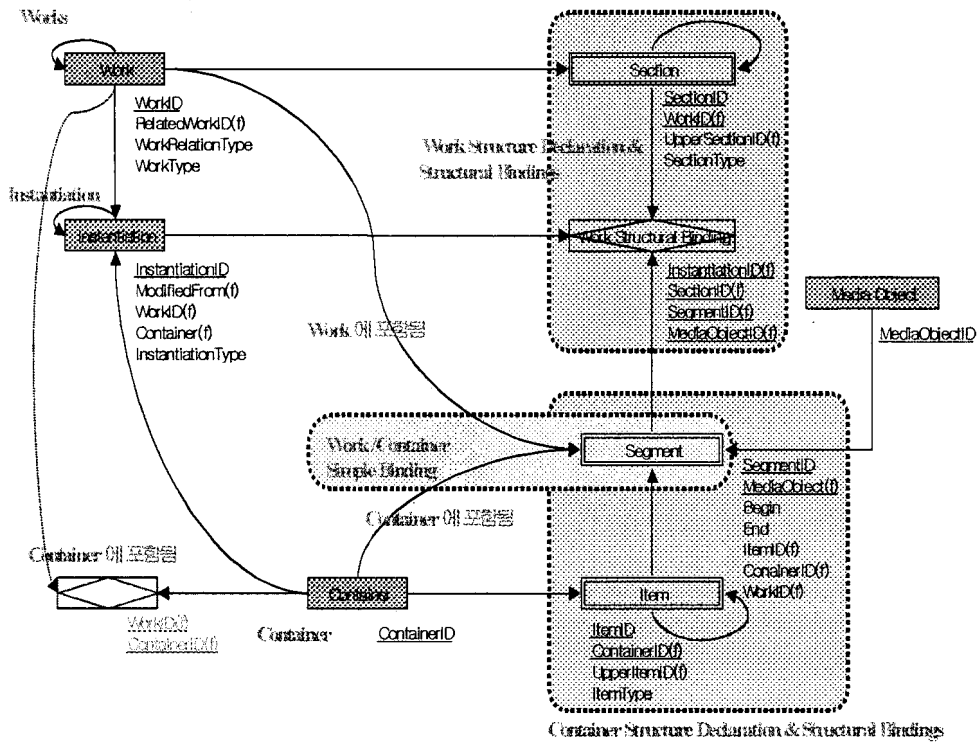
타데이터의 요소를 정하고 또 이를 해당 주제분야의 표준으로 자리매김하는 과정이 필요하다는 것이다.

서울대학교의 메타데이터 연구는 향후, 국내대학 전자도서관 개발의 메타데이터 연구의 기초자료로 활용될 수 있으며, 또한 이를 계기로 모든 기관이 수용할 수 있는 각 주제분야별 메타데이터의 국내표준에 대한 필요성이 대두되어, 새로운 메타데이터 표준이 정립하는데 일조할 수 있다면, 이것만으로도 큰 의의를 지닌다고 할 수 있을 것이라 사료된다.

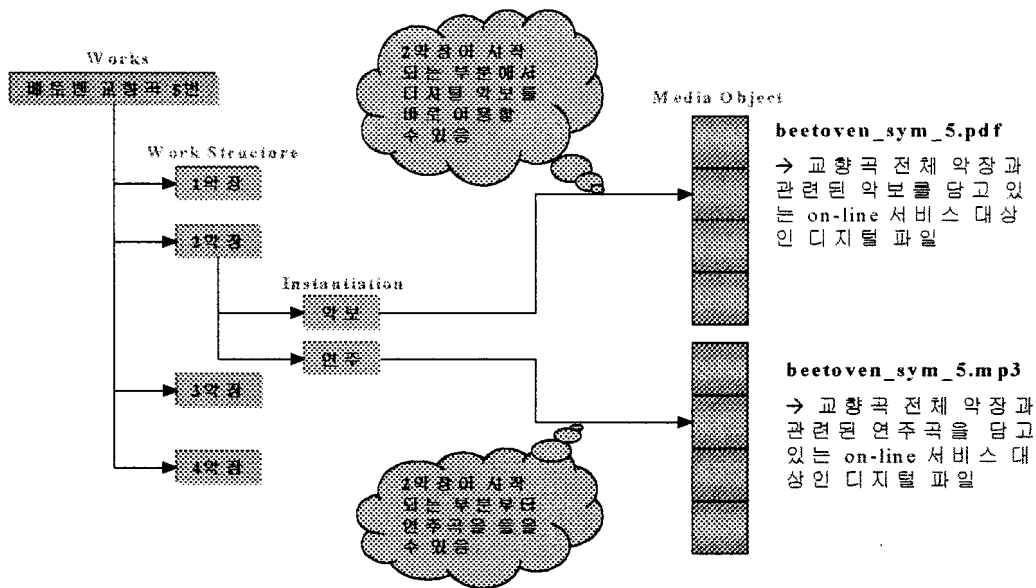
한 XML 스키마 메타데이터 구현에 관한 연구. 서울:서울대학교 중앙도서관., 1p., 2002.

<참고문헌>

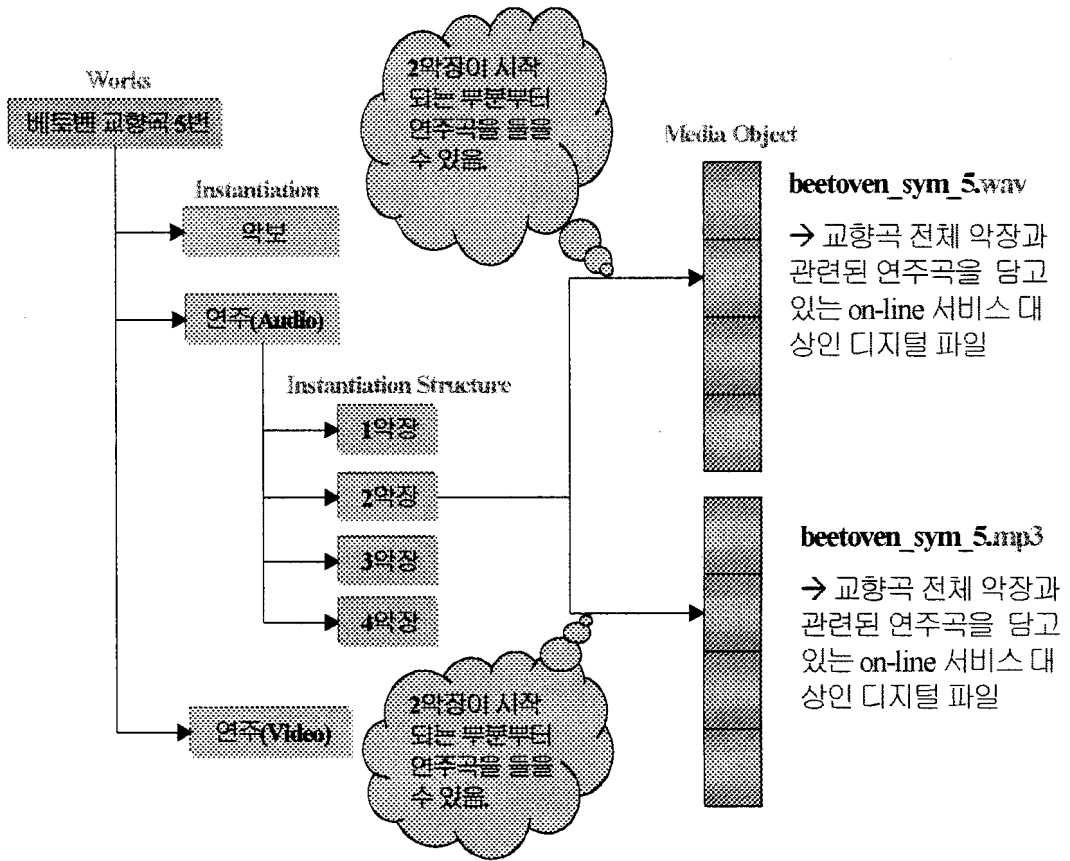
- [1] Ahmed, Kal et al. Professional XML Meta Data. Birmingham : Wrox Press., 25-60p., 2001.
- [2] Berman, David et al. A common Model for Support Interoperable Metadata http://www.oclc.org/research/publication/arr/1998/miller_weibel/model.htm
- [3] Digital Library Metadata Model <http://dml.indiana.edu/pdf/DML-meta-chart.pdf>
- [4] IFLA Functional Requirements for Bibliographic Records
<http://www.ifla.org/VII/s13/frbr/frbr.pdf>
- [5] IU Digital Music Library Data Model Specification V1
<http://dml.indiana.edu/pdf/DML-DataModel-V1.pdf>
- [6] 채진석 등저. 디지털도서관 구축을 위



<그림 4>



<그림 5>



<그림 6>

<부록 1>

	값을 취하지 않고 하부 요소만을 취하는 요소			
	실질적인 값을 취하는 요소			
3.5. Container				
Mapping		Scheme	Label	Definition
요소명	제1하위요소			
dc:title(필)			Title	
	dc:mainTitle(D)		Main Title	Container의 대표적인 제목
	dc:seriesTitle(D)	"ARN, BIB, NG, 혹은 AACR2 규칙을 따르는 목록규칙"	Series Title	Container의 시리즈 명
dc:creator(D)		Authority Control	Contributor	Container 제작에 참가한 모든 사람들
	vCard:FN			
	vCard:N			
	Card:role		Role	작품의 작곡/작사가의 작품에서의 역할
dc:description			Description	Container에 관한 설명
dc:publisher(D)			Publisher	Container를 출판한 사람
dc:date (검색제한자)			Date	Container와 관련된 일시정보
	dc:issued(D)	DCMI Period.W3C-DTF	Issued	Container의 출판일
	dc:available	DCMI Period.W3C-DTF	Available	record 사용 기간
dc:format		MARC21 007 도입고려	Format	Container의 특징에 대한 정보
	dc:extent		Extent	"Container의 시간, 양"
	dc:medium		Medium	container의 물질적인 매체
dc:identifier		URI	Container Id	자동적으로 생성되는 Container의 식별번호
dc:language (검색제한자)		"ISO 639-2, RFC 1766"	Language	Container에 사용된 언어
dc:relation			Relation	Instantiation와 관련된 자원 기술
	dc:containsWork	URI	Contained Work	관련된 작품(Work) ID(s)
	dc:containsInstantiation	URI	Contained Instantiation	"관계된 Instantiation, container에 담긴 모든 Instantiation ID"
dc:rights			Rights	Container에 대한 저작권정보를 기술한다.
dc:version		"AACR2를 따르는 목록규칙(ARN, BIB, 서울대)"	Edtion	Container의 판차정보
dc:condition			Condition	"Container 상태 정도, 디지털화 될 때 물리적인 것의 상태"
dc:jacket		URI	Jacket	Container의 대표가 되는 image 소스
dc:hasStructure		Boolean	Has Structure	내부적인 하부 구조를 가지는지의 유무
dc:simpleBinding			Simple Binding	Container가 내부적인 구조를 가지지 않는 경우의 Binding 기술
	dc:mediaObjectID	URI	Media Object ID	Container가 내부적인 구조를 가지지 않는 경우에 관련된 Media Object의 ID(s)
	dc:begin		Begin	Media Object의 시작부분
	dc:end		End	Media Object의 종료부분

<부록 2> 서울대학교 멀티미디어 콘텐츠 관리 메타데이터

구분	요 소	정 의	Audio	Visual	Image
right	access_category	Access Management Sys에서 이용자 ID와 함께 사용될 접근 유형 코드			
	access_expiration_date	접근유형이 변경되는 날짜			
	access_information	접근관리에 필요한 추가적인 정보			
	access_rights	저작권자에 관한 내용을 수록한 레코드의 위치정보 제공. Pointer임			
File Info	file_identifier	파일 ID (URI)			
	file_name	파일명			
	capture_device_identification	디지털 캡처에 사용된 장비를 구분할 수 있는 장비번호			
	capture_entity_corporate	file/bitstream 제작사			
	capture_entity_individual	파일을 제작자(개인)			공통
	creation_date_time	파일 생성일 및 시간			
	updated_date_time	최종 업데이트일자 및 시간			
	deposit_date_time	제작된 파일이 DB에 upload된 일시정보			
	duration	오디오나 비디오의 시간			
	feature_label	이용자 화면에 뿌려줄 레이블명			
	file_extension	파일의 확장자명			
	size	파일 크기			
	audio_channel_configuration	"디지털파일에 대한 오디오 채널 정보 (예:스테레오, 모노, 서라운드, 음성다중 등)"	○		
	audio_sampling_frequency	"초당 샘플수, (이미지의 해상도와, 비디오의 data_rate과 비슷한 개념)"	○		
	frame_rate	1초에 보여지는 프레임들(frames per second)에서 표현되는 비디오재생 비율		○	
	image_bit_depth	정지/동화상 파일에서 픽셀당 비트 수		○	○
	image_color_space	"컬러이미지에 대한 컬러 모드 (RGB, CMYK 등). "		○	○
	image_pixels_horizontal	이미지나 비디오의 가로 픽셀수(horizontal dimension)		○	○
	image_pixels_vertical	이미지나 비디오의 세로 픽셀수 (vertical dimension)		○	○
	image_spatial_resolution	이미지의 해상도		○	○
video_data_rate	초당 메가피드수 (megabits per second ; mbps)		○		
File Group	file_contents	"파일의 구조 (서론, 본론, 결론 등)"	○	○	
	file_offset	파일의 구조에 따른 Offset 정보 (파일의 각 부분의 Time Code)	○	○	
	related_file_identifier	유관자료 파일 ID	○	○	
	related_file_offset	유관자료 파일의 해당 Time Code에서의 링크정보	○	○	