

# DDI 메타데이터를 활용한 METS 설계에 관한 연구\*

## A Study on METS Design Using DDI Metadata

박진호 (Jin Ho Park)\*\*

### 초 록

이 연구는 데이터셋을 관리, 보존, 서비스하기 위해 DDI 메타데이터를 기반으로 METS를 활용하는 방안을 제시하였다. DDI는 통계 데이터 처리를 위한 표준으로 현재 DDI Codebook(DDI-C)과 DDI Lifecycle(DDI-L) 두 가지 버전이 존재한다. 본 연구에서는 DDI-C의 주요 요소를 주로 하였다. 이를 위해 우선 METS와 DDI-C의 구조와 요소를 분석하였다. 그리고 METS와 DDI-C의 주요 요소들에 대한 매핑작업을 수행하였다. 여기서 기준은 최종적으로 이를 표현할 형식인 METS로 삼았다. METS와 DDI-C가 완벽하게 1:1의 매핑을 보이지 않기 때문에 기준인 METS의 각 요소들에 가장 적합하게 부합하는 DDI-C 요소를 선택하였다. 그 결과 DDI-C 메타데이터요소를 활용한 새로운 데이터셋 관리전송 표준 METS를 설계하여 제시하였다.

### ABSTRACT

This study suggested a method of utilizing METS based on DDI metadata to manage, preserve, and service datasets. DDI is a standard for statistical data processing, and there are currently two versions of DDI Codebook (DDI-C) and DDI Lifecycle (DDI-L). In this study, the main elements of DDI-C were mainly used. First the structures and elements of METS and DDI-C were first analyzed. And the mapping of the major elements of METS and DDI-C. The standard was finally taken as METS, the format to express it. Since METS and DDI-C do not show a perfect 1:1 mapping, the DDI-C element that best matches each element of the standard METS was selected. As a result, a new dataset management transmission standard METS using DDI-C metadata elements was designed and presented.

키워드: 데이터셋, 데이터셋용 스키마, 표준 메타데이터, METS, DDI datasets, schema for datasets, standard metadata, METS, DDI

\* 본 연구는 한성대학교 학술연구비 지원과제임.

\*\* 한성대학교 도서관정보문화트랙 조교수(jhp@hansung.ac.kr)

■ 논문접수일자: 2021년 11월 20일 ■ 최종심사일자: 2021년 12월 7일 ■ 게재확정일자: 2021년 12월 18일  
■ 정보관리학회지, 38(4), 153-171, 2021. <http://dx.doi.org/10.3743/KOSIM.2021.38.4.153>

※ Copyright © 2021 Korean Society for Information Management

This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 (<https://creativecommons.org/licenses/by-nc-nd/4.0/>) which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.

## 1. 서론

구글(Google)은 2020년 1월부터 데이터세트 검색 베타 서비스를 시작했다. 현재 이 서비스는 'Dataset Search'라는 공식명칭을 사용하여 정식서비스 되고 있다. 구글은 이 데이터세트 검색 서비스의 공식 도메인으로 'https://datasetsearch.research.google.com/'을 사용하고 있다. 구글은 데이터세트 검색이 모든 이용자를 위한 것이고, 이 검색서비스는 웹 전체의 저장소에 호스팅된 데이터세트를 검색할 수 있다고 설명하고 있다. 그럼에도 도메인명에서 'research'를 사용하고 있는 점은 데이터세트 검색서비스에서 과학연구와 연구자를 얼마나 중요하게 생각하고 있는지를 짐작할 수 있게 한다. 즉 연구자와 연구데이터의 중요성을 인지하고 있다고 볼 수 있다.

연구분야에서 데이터세트는 이미 중요한 정보 자원의 하나로 인식되어왔다. 대표적인 사례가 1962년 설립된 미국 최대 사회과학데이터보관소인 ICPSR(The Inter-University Consortium for Political and Social Research)이다. ICPSR은 연구 데이터세트를 수집, 보존하고 있다. 또한 관련 데이터세트를 검색하고 이용자가 필요한 분석도구별 형식으로 다운로드를 제공하고 있다. 도구 활용이 어려운 이용자들을 위해서는 웹 기반의 온라인 분석도구서비스까지 제공하고 있다. ICPSR의 경우는 이 데이터세트에 대한 표준 메타데이터로 DDI(The Data Documentation Initiative) 메타데이터를 활용하고 있다. 연구 데이터세트에 대한 관심과 관리, 보존은 학술연구의 중심인 대학, 그 중에서도 대학도서관이 주체적으로 수행하고 있다. 이는 연구자금

제공기관의 요구와 연구투명성 확보를 위해 데이터관리계획(Data Management Plan) 작성이 연구자들에게 요구되면서 더욱 강화되었다고 볼 수 있다. 대표적으로 MIT 도서관은 도서관 홈페이지에 연구지원 메뉴(<https://libraries.mit.edu/research-support/>)를 별도로 두어, 데이터 공유, 저장, 관리 방법을 안내하고 있다. 여기서 데이터세트 관리에 필요한 메타데이터로 더블린코어(dublin core)를 권고하고 있다. 스탠포드 대학도서관도 연구데이터 서비스(<https://library.stanford.edu/research>)를 제공하고 있고 관련 메타데이터로 더블린코어, MODS(Metadata Object Description Schema), EAD(Encoded Archival Description), DDI를 권고하고 있다.

이렇듯 데이터세트는 연구자와 연구와 같이 국한된 이용자와 목적에 적합한 정보자원으로 볼 수 있다. 반면 오늘날 구글의 데이터세트 검색, 각 국 정부들이 운영하고 있는 열린 정부 프로젝트와 연계된 다양한 데이터세트 검색 서비스들은 데이터세트라는 정보자원의 이용 계층이 확대되었고 활용처 역시 다양하고 일반화되고 있음을 보여준다. 이런 영향은 도서관에도 상당한 변화를 요구한다고 볼 수 있다. 기존의 경우 대학도서관을 중심으로 연구지원서비스의 일환으로 데이터세트 관리를 위한 지원과 노력을 기울이고 있지만 향후에는 다양한 관중에 유사한 노력과 서비스가 필요할 것으로 본다.

도서관에서 정보자원에 대한 관리는 일반적으로 표준메타데이터를 활용하여 정리하고 보존과 서비스를 수행하는 것이 기본이라 할 수 있다. 연구 데이터세트를 중심으로 보면 대

학도서관이 중심이 되어 관련 연구를 수행한 것처럼 보이지만, 이미 디지털정보자원의 보존과 관리/전송을 위한 표준 제정과 활용을 위한 노력은 도서관 전반에 걸쳐 이루어져왔다고 볼 수 있다. METS(Metadata Encoding and Transmission Standard)는 이런 노력의 대표적인 결과물로 볼 수 있다. METS는 XML 형식으로 데이터의 인코딩과 전송에 필요한 메타데이터 정보를 담는 도구로 볼 수 있다. 이는 디지털 객체를 장기보존해야 하는 기관들이 관련 메타데이터를 유지해야 하는 문제와 이 대상 디지털 객체의 복잡도가 날로 복잡해짐에 따라 다양한 메타데이터가 필요함을 인지하고 이를 해결하기 위한 방안으로 만들어진 표준이다(Cantara, 2005). 반면 DDI는 1980년대 통계 데이터처리를 위한 메타데이터의 필요성이 증가함에 따라 만들어졌다. 현재는 주요 국가의 통계청이 주로 활용하고 있으며, 연구데이터를 관리, 서비스하는 기관에서의 활용이 높다고 볼 수 있다. METS와 마찬가지로 DDI 역시 XML 형식을 취한다.

본 연구에서는 데이터셋을 보존하고 관리 및 서비스하기 위한 전송 표준으로 METS를 활용하고자 한다. METS는 구조를 정의하고 있는 표준으로 여기에 담아 사용해야 하는 표준 메타데이터들이 필요하다. 통상 해당 기관에서 사용하는 메타데이터가 그 대상이다. 본 연구에서는 그 대상을 DDI 로 하였다. DDI가 모든 데이터셋 관리를 대표하는 공인된 표준으로 보기는 어렵지만, DDI의 경우 ICPSR, 주요국의 통계청에서 이미 활용하고 있는 표준이다. 또 DDI의 경우는 데이터셋이 변수를 표현할 수 있는 요소를 함께 포함하고 있어 기존 메타데

이터보다 데이터셋에 적합하고 다양한 서비스가 가능하다 할 수 있다. 기존의 더블린 코어나 MODS로 데이터셋을 잘 설명할 수 있지만 해당 데이터셋을 응용한 서비스까지 고려한다면 또 다른 메타데이터의 도입도 고려할 필요가 있다. 도서관과 같은 정보자원관리 기관의 입장에서 데이터셋 관리, 서비스, 보존의 필요성이 증대되고 있는 시점에서 DDI와 METS 활용은 의미있는 시도가 될 수 있을 것으로 본다. DDI는 내부적으로 다양한 형식의 데이터셋 관련 정보를 담고 있다. 본 연구에서는 DDI 메타데이터를 활용하여 METS 형식으로 데이터셋을 관리, 전송, 서비스할 수 있는 형식 설계를 목적으로 한다.

## 2. 이론적 배경

### 2.1 METS 및 DDI 분석

METS는 메타데이터 인코딩과 전송 표준의 영문 약자이다. METS는 디지털도서관 내 객체에 대한 기술(descriptive), 관리(administrative), 구조(structural)적 메타데이터를 인코딩하기 위한 메타데이터 표준으로 XML로 표현한다. 이 표준은 디지털도서관연맹(DLF, Digital Library Federation) 주도로 개발되었으며, 미국의회도서관에서 지속적으로 유지관리를 담당하고 있다. 미국의회도서관은 이를 MARC 표준의 일부로써 다루고 있다(Library of Congress, 2021). METS에 대한 일반적인 내용은 안내서인 'METS Primer'에서 확인할 수 있는데 2010년 4월에 개정된 문서가 가장 최신 내용을 담

고 있다. 실제 METS를 구현하기 위한 설계도에 해당하는 XML Schema의 가장 최신 버전은 '1.12.1'로 2019년 10월에 발간되었다. 본 연구에서는 XML Schema 1.12.1에서 사용하는 요소를 기본으로 하였다. METS는 해당 정보 객체에 탐색과 관리에 필요한 모든 메타데이터 정보를 담고 있는 일종의 정보 패키지로 볼 수 있다. METS라는 용어 자체에도 표현되어 있지만 이런 정보 패키지가 시스템간 전송될 때 그 효과가 극대화될 수 있다고 볼 수 있다. 도서관을 포함한 정보기록물 관리 기관입장에서보면 ISO 14721 OAIS 모델에서 정의하고 있는 SIP(Submission Information Package), AIP(Archival Information Package), DIP(Dissemination Information Package)에 응용하여 활용 가능하다. 궁극적으로 METS는 텍스트와 다양한 형태의 멀티미디어 객체들에 대한 구조적 메타데이터를 구축하여 디지털도서관시스템 간에 객체 상호운용성과 장기보존을 지원하는데 있다(McDonough, 2006).

DDI는 통계 데이터용 메타데이터의 필요성이 증가하면서 2003년 설립된 조직이다. 주로 국가 통계기관, 정부, 기록관, 대학을 중심으로 DDI 메타데이터를 활용하고 있으며, 우리나라의 경우도 통계청에서 DDI를 활용하고 있다. DDI 메타데이터는 현재 DDI Codebook(DDI-C)과 DDI Lifecycle(DDI-L) 두 가지 버전이 존재한다. DDI Lifecycle의 경우는 복잡한 모듈화로 직접 활용이 어렵고 대부분 DDI-L 버전의 요소를 활용하고 있다. DDI의 경우 데이터관리계획(DMP, Data Management Plan) 작성 시 활용할 수 있는 메타데이터로 많이 활용된다. 특히 ICPSR의 경우도 데이터셋에 대한 표준 메

타데이터로 DDI를 표준으로 하고 있다.

## 2.2 선행연구

본 연구관련 선행연구는 METS 관련 연구, DDI 관련 연구, 연구데이터 관련 연구로 구분하여 살펴볼 수 있다. METS와 DDI는 메타데이터 관점의 연구로 볼 수 있는데, 관련 선행연구를 살펴보면 박옥남(2012)은 국립중앙도서관의 디지털정보자원을 영구보존하고 데이터 교환을 위해 METS와 PREMIS를 활용한 목록레코드 작성 방안을 제안한 바 있다. 구체적으로는 국립중앙도서관 담당자와의 면담과 PREMIS 분석을 통해 메타데이터를 도출하고 이를 METS에 적용하였다. 또 박옥남(2018)은 한국과학기술정보연구원의 NTIS 시스템, ICPSR, Harvard Dataverse, Dryad Digital Repository 등의 사례를 조사하여 연구데이터 관리를 위한 온톨로지 모델을 제시하는 연구를 수행한 바 있다. 이 과정에서 DDI 메타데이터를 상세히 분석하여 구성요소에 대한 설명을 함께 제시한 바 있다.

연구데이터 관련 연구는 메타데이터에 비해 다양하게 이루어져왔다. 이미화, 이은주, 노지현(2020)은 국립중앙도서관의 오픈액세스 리포지토리에서 연구데이터 확장방안을 제안하면서 중복연구방지, 데이터세트의 공유, 재활용을 기대효과로 제시한 바 있다. 이를 위해 문헌 연구, 사례조사, 관련자 면담을 수행하여 메타데이터 확장 방안을 설계한 것이 특징이다. 특정 지식분야 관련 연구로 김주섭 외(2020)는 임상분야 연구데이터셋을 관리하기 위한 표준메타데이터 선정을 위한 연구를 진행한 바 있다. 여기서는 메타데이터 항목을 도출하기 위하여

FGDC-CSDGM, ISO 19115, NEFIS, INSPIRE, ANZLIC, DataCite 4.3 그리고 TTA, KO-10,0976 등 7개 메타데이터를 대상으로 매핑작업을 진행하였고, 그 결과 필수 6개, 권고 13개 그리고 선택 요소 9개의 새로운 메타데이터셋을 도출하였다.

연구데이터셋의 서비스 측면을 다룬 연구를 보면 김지현(2014)은 미국대학도서관의 연구데이터관리 서비스 현황을 분석하여 연구데이터서비스에 있어서 필수적인 요소가 무엇인지를 파악하여 제안한 바 있다. 이 연구가 분석한 핵심 구성요소는 9가지로 DMP 작성지원, 데이터 파일 정리, 데이터 기술, 데이터 저장, 데이터 공유 및 접근, 데이터 보존, 데이터 인용, 데이터관리 교육, 데이터 지적재산권이다. 박진호, 고영만, 김현수(2019)는 연구데이터 서비스의 유용성을 평가할 수 있는 지표와 모델을 개발하여 이용자 검증을 수행한 바 있다. 여기서는 내재된 평가척도인 검색성, 접근성, 상호운용성, 재활용성 4개와 각각의 측정지표 총 20개를 도출하였고, 이용자 경험 기반의 유용성 측정지표 12개를 도출하였다. 도출된 모형은 이용자 설문조사를 거쳐 타당도와 신뢰도를 검증하였다. 이미 도서관에서 METS는 보존과 전송 표준으로 확고한 위치를 차지하고 있다고 볼 수 있다. 또한 DDI의 경우는 데이터셋을 관리하고 이용자에게 응용서비스를 제공하기 위해 필요한 검증된 표준요소로 볼 수 있다. DDI의 특징은 데이터셋의 변수를 표현가능하다는 것으로 이는 데이터셋 활용에 있어서 제일 중요한 측면으로 볼 수 있는 활용성을 높일 수 있는 특징이라 할 수 있다. 본 연구는 DDI 메타데이터 요소들을 활용하여 METS

기반의 데이터셋 보존 및 전송 표준을 설계해보는 것이다. 선행연구에서 검토한 바와 같이 데이터셋 관리와 서비스에 있어서 메타데이터의 중요성에 대한 언급과 제안을 있었지만, 기존 표준인 METS와 DDI를 결합한 연구는 없다는 점에서 중요한 접근이 될 수 있을 것이다. 또, METS 자체가 보존, 전성 표준이므로 향후 도서관을 포함한 다양한 기관에서 실제 활용할 것으로 기대하며, 특히 DDI 메타데이터를 활용함으로써 데이터셋의 활용성 증대에도 긍정적 영향을 줄 수 있을 것으로 본다. METS와 DDI 모두 XML을 기반으로 하며, XML Schema로 구조를 정의하고 있어, 두 표준을 결합하는 것은 구조 간의 연결 적합성을 확인하면 구현가능하다. 본 연구에서도 두 표준의 XML Schema를 기준으로 구조와 요소분석을 실시하였다.

### 3. METS와 DDI 분석

#### 3.1 수행절차

본 연구에서는 특정 데이터셋을 하나의 객체로 보고 이를 METS기반으로 인코딩하고자 한다. 이 때 대상 데이터셋에 관한 기술, 관리 요소로 DDI 메타데이터를 사용한다. 구체적인 수행 절차는 <표 1>과 같다.

연구수행의 첫 번째는 절차는 METS 분석이다. 여기서는 METS Schema에 근거하여 METS 구조를 분석하고 각 구조에서 요구되는 메타데이터 요소나 필요 기술요소가 무엇인지를 구체적으로 확인한다. 두 번째는 DDI 분석으

〈표 1〉 연구수행 절차

절차	수행내용
METS 분석	METS XML Schema에 근거한 구조와 메타데이터 요소 분석
DDI 분석	DDI XML Schema에 근거한 구조와 메타데이터 요소 분석
DDI 기반 METS 인코딩	DDI 요소를 활용한 METS 인코딩

로 역시 DDI XML Schema에 근거하여 구조와 메타데이터를 분석한다. 마지막은 DDI의 각 구성요소와 메타데이터 요소가 METS의 주요 부분에 적합한 요소인지를 확인한다.

### 3.2 METS 구조와 메타데이터 요소분석

METS 구조는 METS XML Schema(<https://www.loc.gov/standards/mets/mets.xsd>)를 기준으로 분석하였다. METS는 'mets'를 단일요소(element)로 구성되어 있다. 여기에 14개의 complexType와 1개의 simpleType, 4개의 attributeGroups이 구성요건이 된다. 이를 정리하면 〈표 2〉와 같다. XML Schema에서 complexType은 자식요소나 속성을 포함하는 요소를 유형을 의미하며, simpleType은 그렇지 않은 경우이다.

〈표 2〉가 보여주는 METS의 특징은 구성요소가 단순하다는 점이다. 이는 느슨한 구조를

가지고 있다고 표현할 수 있다. 즉 METS는 특정 객체를 상세히 기술하기 위한 용도보다는 특정 객체를 기술하는데 필요한 다양한 정보자원을 부분으로 구성하고 해당하는 설명 정보(메타데이터)의 선택은 자유롭다.

METS의 핵심구조를 파악하기 위해 검토해야하는 것은 요소(elements)이다. 〈표 2〉에서 알 수 있듯이 METS의 요소는 'mets'이다. METS로 기술된 문서는 'mets'요소를 루트(root)요소로 7개의 보조섹션으로 구성된다. 이 7개의 보조섹션이 사실 METS 문서를 기술하는 핵심이지만 이는 METS 문서가 갖는 14개의 complexTypes 중 'metsType'을 구성하는 보조섹션이다. 이 7개 보조섹션은 〈표 3〉과 같이 정리된다.

〈표 3〉에서 확인할 수 있는 7가지 보조섹션은 'complex Type'으로 각각 정의되어 있으며, 〈metsType〉 요소에서 이를 직접 활용하도록 구성되어 있다.

〈표 2〉 METS Schema 구조 분석

구분	개수	요소
complexType	14	amdSecTyp, areaType, behaviorSecType, behaviorType, divType, fileGrpType, fileType, mdSecType, metsType, objectType, parType, seqType, structLinkType, structMapType
simpleType	1	URIs
elements	1	mets
attributeGroup	4	FILECORE, LOCATION, METADATA, ORDERLABELS

〈표 3〉 METS의 7개 보조섹션 상세설명

섹션이름	설명
metsHdr (METS 문서 헤더 섹션)	<ul style="list-style-type: none"> <li>• METS 문서가 인코딩하는 디지털 객체가 아니라 METS 문서 자체에 대한 메타데이터 정보 기술</li> </ul>
dmdSec (설명 메타데이터 섹션)	<ul style="list-style-type: none"> <li>• METS가 기술하는 객체 전체 혹은 일부분에 관련된 설명 메타데이터를 기록</li> <li>• &lt;dmdSec&gt;요소는 &lt;techMD&gt;, &lt;rightsMD&gt;, &lt;sourceMD&gt;, &lt;digiprovMD&gt;와 같은 수준의 일반 데이터 유형을 준수하고 동일한 하위요소와 속성을 활용함</li> <li>• 해당 설명 메타데이터 요소는 메타데이터를 래핑(mdWrap)하여 문서내 직접 표현하거나 외부 위치(mdRef)를 참조하여 표현할 수 있으며, 동시에 이 두가지 모두를 활용해도 무방함</li> <li>• METS는 여러개의 &lt;dmdSec&gt; 요소를 허용함</li> <li>• 설명 메타데이터는 DMDID 속성을 지원하는 어떤 METS 요소와도 연결할 수 있음</li> <li>• 설명 메타데이터는 MARC, MODS, Dublin Core, TEI Header, EAD, VRA, FGDC, DDI와 같은 표준은 물론 로컬에서 생성한 XML 스키마에 따라 표현할 수 있음</li> </ul>
amdSec (관리 섹션)	<ul style="list-style-type: none"> <li>• 디지털 객체, 디지털 객체의 구성요소와 디지털 객체가 파생된 원본과 관련된 관리적 메타데이터를 포함하여 기술함</li> <li>• &lt;amdSec&gt;은 4개의 하위 섹션으로 다시 구분되는데 이는 기술 메타데이터(techMD), 지적재산권(rightMD), 아날로그/디지털 소스 메타데이터(sourceMD), 디지털 출처 메타데이터(digiprovMD)임</li> <li>• 이 4개 하위 섹션은 메타데이터를 래핑(mdWrap)하여 문서내 직접 표현하거나 외부 위치(mdRef)를 참조하여 표현할 수 있으며, 동시에 이 두 가지 모두를 활용해도 무방함</li> <li>• METS 문서내에서 &lt;amdSec&gt; 요소의 여러 인스턴스가 발생할 수 있으며, 하위 섹션의 여러 인스턴스는 하나의 &lt;amdSec&gt;요소에서 발생할 수 있음</li> <li>• &lt;METS&gt;는 관리 메타데이터를 인코딩하기 위한 어휘나 구문을 정의하지 않음</li> </ul>
fileSec (파일 섹션)	<ul style="list-style-type: none"> <li>• METS 문서가 설명하는 디지털 객체를 구성하는 콘텐츠 파일의 목록과 위치 제공</li> <li>• 하위 섹션으로 fileGrp을 가지며, fileGrp으로 METS가 기술대상 객체가 갖는 다양한 이미지 그룹(썸네일, 고휘상도이미지, 원본 등)을 나열하여 표현함</li> </ul>
structMap (구조맵)	<ul style="list-style-type: none"> <li>• &lt;structMap&gt;은 METS 문서의 핵심임</li> <li>• METS 문서의 &lt;fileSec&gt;에 있는 &lt;file&gt; 요소가 나타내는 디지털 콘텐츠를 계층구조로 구성하는 방법을 제공</li> <li>• 이 계층 구조는 디지털 콘텐츠의 이해와 탐색을 용이하게 하기 위해 이용자에게 제공할 수 있음</li> <li>• 콘텐츠 파일 또는 콘텐츠 파일의 일부의 구조적 관계에 대한 이해 등 필요한 모든 목적에 모두 적용가능함</li> <li>• &lt;structMap&gt; 요소는 반복 가능함</li> <li>• &lt;structMap&gt;은 하위에 &lt;div&gt;요소를 가지면 이 요소안에 계층구조를 인코딩하여 표현함</li> </ul>
structLink (구조맵 연결)	<ul style="list-style-type: none"> <li>• 구조적 맵에 기술된 METS 구조의 서로 다른 구성요소 간의 하이퍼링크 허용</li> <li>• 구조 맵에서 두 노드 간의 하이퍼링크를 나타내는 반복 가능한 단일 요소인 &lt;structLink&gt;에 대한 컨테이너</li> <li>• METS 문서의 &lt;structLink&gt;섹션은 XML ID 속성을 사용하여 식별</li> </ul>
behaviorSec (동작 섹션)	<ul style="list-style-type: none"> <li>• 디지털객체를 활용하는데 필요한 소프트웨어의 작용을 기록</li> <li>• 특정 소프트웨어를 기술할 수도 있고, 렌더링에 필요한 특정 파라미터를 기술할 수도 있음</li> </ul>

### 3.3 DDI 구조와 메타데이터 요소분석

DDI XML Schema를 기준으로 구조를 살펴보면 〈표 4〉와 같다.

〈표 4〉는 DDI-C 2.5버전의 schema 구조에

대한 상세정보이다. DDC-C가 DDI-L에 비해 활용하기에 간단하다고 하지만 그럼에도 불구하고 많은 요소들로 구성되어있다. DDI-C 역시 핵심구조를 파악하기 위해 검토해야하는 것은 요소(elements)이다. 〈표 4〉에서 알 수 있

〈표 4〉 DDI Schema 구조 분석

구분	complexType	simpleType	elements	element groups	global attributes	attribute groups
개수	232	25	342(global) + 9(local)	73	7	70

듯이 DDI-C의 요소는 본 보고서에 모두 기술 하기에 어려울 정도 많다. 이 중 우리가 핵심요소로 검토해야하는 것은 'codeBook'이다. 사실 DDI-C의 핵심 기술요소들은 모두 여기에 포함 되어 있으며, 데이터셋트를 표현하고 보존, 서비스하는데 필요한 요소들을 모두 포괄하고 있다. codeBook의 schema의 구조는 〈표 5〉와 같다.

〈표 5〉에서 보듯이 DDI-C의 'codeBook'은 많은 양의 요소(elements)를 포함하고 있는 것을 확인할 수 있다. METS와 대비하여 설명하면

DDI-C는 특정 객체를 상세하게 설명할 수 있는 설명요소를 지정하여 포함하고 있음을 알 수 있다. 이 요소가 'codebook'이며, 데이터셋트를 객체로 보고 설명하기 위한 요소들은 모두 여기에 포함되어 있다. codeBook은 complexType이다. 실제 XML Schema에서는 'codeBookType'으로 정의되어 인코딩한다. 'codeBook' 요소는 총 5개의 하위 요소로 구성되어 있는데 이를 정리 하면 〈표 6〉과 같다.

각각의 요소들은 구체적인 속성(attribute) 정

〈표 5〉 DDI codeBook Schema 구조 분석

구분	complexType	simpleType	elements	element groups	attribute groups
개수	166	3	243(global) + 9(local)	2	1

〈표 6〉 DDI-C codeBook 요소 상세설명

요소명	설명	하위요소명	설명
docDscr	<ul style="list-style-type: none"> <li>• 문서설명(Document Description)</li> <li>• 문서설명은 DDI가 표현하는 문서자체에 대한 서지정보로 구성됨</li> </ul>	citation	<ul style="list-style-type: none"> <li>• 서지인용정보</li> <li>• 마크업(marked-up) 문서의 인용과 설명</li> <li>• 마크업(markup-up)문서의 출처에 대한 인용과 설명</li> <li>• 연구 자체에 대한 설명, 인용</li> <li>• 다른자료와 관련된 연구설명</li> <li>• 서지 정보에는 제목, 책임사항, 제작 및 배포 정보, 연속간행물의 버전정보, 서지 인용, 주석 정보들을 포함함</li> <li>• 인용에 대한 MARC 레코드에 연결하기 위해 MARCURI 속성 제공</li> </ul>
		guide	<ul style="list-style-type: none"> <li>• 코드북 가이드(Guide to Codebook)</li> <li>• 문서에 사용하고 있는 용어 정의목록</li> <li>• 이용자가 문서를 올바르게 사용할 수 있도록 지원</li> </ul>
		docStatus	<ul style="list-style-type: none"> <li>• 문서상태</li> <li>• 데이터와 문서의 배포 여부, 시기 등을 표현</li> </ul>



요소명	설명	하위요소명	설명
		docSrc	<ul style="list-style-type: none"> <li>문서의 소스</li> <li>원본 문서에 대한 인용정보를 표현</li> <li>제목, 책임사항, 제작 및 배포 정보, 연속간행물의 버전정보, 서지 인용, 주석 정보들을 포함함</li> <li>인용에 대한 MARC 레코드에 연결하기 위해 MARCURI 속성 제공</li> </ul>
		controlledVocabUsed	<ul style="list-style-type: none"> <li>사용 통제어휘</li> <li>코드값과 입수 목록에 대한 참조정보 제공</li> </ul>
		note	<ul style="list-style-type: none"> <li>주석</li> <li>상위요소에 대한 주석</li> <li>주석에도 아이디를 부여하여 반복사용과 식별이 가능하도록 함</li> </ul>
stdyDscr	<ul style="list-style-type: none"> <li>연구설명</li> <li>데이터 수집, 연구, 편집정보들을 구체적으로 기술</li> </ul>	citation	<ul style="list-style-type: none"> <li>서지인용정보</li> <li>마크업(marked-up) 문서의 인용과 설명</li> <li>마크업(markup-up) 문서의 출처에 대한 인용과 설명</li> <li>연구 자체에 대한 설명, 인용</li> <li>다른자료와 관련된 연구설명</li> <li>서지 정보에는 제목, 책임사항, 제작 및 배포 정보, 연속간행물의 버전정보, 서지 인용, 주석 정보들을 포함함</li> <li>인용에 대한 MARC 레코드에 연결하기 위해 MARCURI 속성 제공</li> </ul>
		studyAuthorization	<ul style="list-style-type: none"> <li>연구승인 정보</li> <li>연구 승인기관, 날짜, 설명 등을 구조화된 형태로 기술</li> </ul>
		stdyInfo	<ul style="list-style-type: none"> <li>연구범위 정보</li> <li>데이터 수집 범위를 상세히 기술</li> </ul>
		studyDevelopment	<ul style="list-style-type: none"> <li>연구개발 정보</li> <li>연구개발 과정 전체 활동을 단계별로 상세히 설명</li> <li>연구에 참여하는 참가자의 역할과 소속, 사용된 자원의 출처, 활동을 상세히 기술</li> </ul>
		method	<ul style="list-style-type: none"> <li>방법론 및 처리방법</li> <li>데이터 수집과 관련된 방법론과 처리방법을 기술</li> </ul>
		dataAccs	<ul style="list-style-type: none"> <li>데이터 접근</li> <li>데이터 수집 시 필요한 접근, 사용 조건을 설명</li> <li>파일 혹은 특정 변수 등 개별적인 접근방식을 구분하여 기술가능</li> </ul>
		othrStdyMat	<ul style="list-style-type: none"> <li>기타연구설명자료</li> <li>연구설명과 관련된 기타자료들로 부록, 샘플링, 가중치 기준, 연구관련 출판물 등 다양한 기타정보를 포함할 수 있음</li> </ul>
fileDscr	<ul style="list-style-type: none"> <li>데이터 파일 설명</li> <li>컬렉션을 구성하는 데이터 파일에 대한 정보로 여러 파일 정보를 반복해서 기술할 수 있음</li> </ul>	fileTxt	<ul style="list-style-type: none"> <li>파일별 설명</li> <li>데이터 파일에 대한 설명</li> <li>파일이름, 파일에 대한 서지인용정보</li> <li>데이터 파일의 물리적 속성 등을 기술</li> </ul>
		locMap	<ul style="list-style-type: none"> <li>위치 맵</li> <li>개별 데이터 항목에 대한 물리적 저장위치 정보</li> </ul>
		notes	<ul style="list-style-type: none"> <li>주석</li> <li>상위요소에 대한 주석</li> <li>주석에도 아이디를 부여하여 반복사용과 식별이 가능하도록 함</li> </ul>

요소명	설명	하위요소명	설명
dataDscr	• 변수 설명	varGrp	• 변수집합 • 공통주제 혹은 단일 질문의 해석에 사용할 수 있고 다른 요인에 의해 연결가능한 변수 그룹
		nCubeGrp	• nCube 집합 • 공통주제 혹은 단일 질문의 해석에 사용할 수 있고 다른 요인에 의해 연결가능한 nCube 그룹
		var	• 변수 • 사회 과학 데이터 파일에 있는 단일 변수의 모든 기능을 설명 • 다국어사용을 허용함
		nCube	• nCube • n차원 배열의 논리적 구조를 설명
		notes	• 주석 • 상위요소에 대한 주석 • 주석에도 아이디를 부여하여 반복사용과 식별이 가능하도록 함
otherMat	• 부록, 샘플링 정보, 가중치 세 부정보, 방법론, 연구내용에 기반한 출판물, 관련 연구와 같이 주로 연구 내용이나 용도를 설명하는 자료를 기술 • 주로 연구에 이용했거나 연구 분석에 유용한 자료를 포함하거나 연결하기 위한 용도	labl	• 레이블 • 상위 요소에 대한 간단한 설명
		txt	• 설명 텍스트 • 상위요소에 대한 더 긴 수준의 설명
		notes	• 주석 • 상위요소에 대한 주석 • 주석에도 아이디를 부여하여 반복사용과 식별이 가능하도록 함
		table	• 표정보
		citation	• 서지인용정보 • 마크업(marked-up) 문서의 인용과 설명 • 마크업(markup-up) 문서의 출처에 대한 인용과 설명 • 연구 자체에 대한 설명, 인용 • 다른자료와 관련된 연구설명 • 서지 정보에는 제목, 책임사항, 제작 및 배포 정보, 연속간행물의 버전정보, 서지 인용, 주석 정보들을 포함함 • 인용에 대한 MARC 레코드에 연결하기 위해 MARCURI 속성 제공
		otherMat	• 기타연구관련 자료 • 연구 관련 기타 자료

보를 포함하고 있지만 이를 모두 기수하지 않았다. 이는 DDI-C의 요소를 METS에 기술하면 해당 요소는 해당 속성을 상속받아 그대로 구현 가능하기 때문이다.

### 3.4 METS와 DDI 구조 매핑

본 연구는 DDI요소를 기준으로 METS에 데이터셋 정보를 인코딩하는 것이다. 4장에서

분석한 내용을 기반으로 METS 기본 구조에 매핑 가능한 DDI 요소를 분석해보면 <표 7>과 같이 정리할 수 있다.

<표 7>에서 보는 것처럼 METS와 DDI는 1:1로 완벽하게 일치하지는 않는다. METS는 앞서 언급한 것처럼 데이터 인코딩 전송 표준으로 기존의 정보자원을 보존하고 관련 정보를 활용할 수 있는 유연한 구조를 갖고 있다. 반면 DDI는 데이터셋, 특히 연구와 통계처리를 목

〈표 7〉 METS 구조와 DDI 요소 매핑

METS 구조	DDI 요소
metsHdr(METS 문서 헤더 섹션)	
dmdSec(설명 메타데이터 섹션)	<ul style="list-style-type: none"> <li>• fileDscr(데이터 파일 설명)</li> <li>• dataDscr(변수 설명)</li> </ul>
amdSec(관리 섹션)	<ul style="list-style-type: none"> <li>• docDscr(문서설명)</li> <li>• stdyDscr(연구설명)</li> <li>• otherMat(기타자원)</li> </ul>
fileSec(파일 섹션)	<ul style="list-style-type: none"> <li>• fileDscr(데이터 파일 설명)</li> </ul>
structMap(구조맵)	
structLink(구조맵 연결)	
behaviorSec(동작 섹션)	<ul style="list-style-type: none"> <li>• dataDscr(변수 설명)</li> </ul>

적으로하는 데이터세트를 관리하고 서비스하기 위한 목적이 크다. METS에 비해 목적 객체가 분명하기 때문에 관련 요소들의 양도 풍부하고 구조는 DDI에 비해 엄격하다 할 수 있다.

〈표 7〉의 매핑표를 보면 우선, METS의 'metsHdr'는 헤더 영역으로 METS 문서 자체에 대한 메타데이터 정보를 기술하기 때문에 DDI와 매핑되는 요소는 없다. 하지만 실제로 데이터세트를 METS로 인코딩한다면 해당 데이터세트를 기술하고있는 문서 정보를 기술하기 때문에 실제 독립적으로 입력 가능한 요소로 볼 수 있다.

두 번째는 METS의 'dmdSec'이다. 'dmdSec'는 METS가 대상으로하는 객체의 일부 혹은 전체를 설명하는 메타데이터를 기록한다. 사실 〈표 3〉에서 설명한 것처럼 METS의 경우 'dmdSec'에 MARC, MODS, Dublin Core, TEI Header, EAD, VRA, FGDC, DDI와 같은 표준 메타데이터 전체를 기술할 수 있도록 허용하고 있다. 그럼에도 본 연구에서는 DDI가 단순히 대상 객체를 설명하는데 그치지 않고 활용할 수 있는 요소들을 가지고 있어 향후 장기

보존과 서비스를 고려하여 세분화여 매핑하고 활용할 수 있는 방법을 제안하였다. DDI의 경우 여기에 부합하는 요소는 두 가지로 볼 수 있다. 데이터세트의 파일을 설명하는 'fileDscr'와 해당 데이터세트의 변수를 설명하는 'dataDscr'가 여기에 해당한다고 볼 수 있다. 'fileDscr'은 해당 데이터세트의 파일명과 서지사항, 위치, 주석을 모두 포괄하고 있다. 'dataDscr'은 DDI의 가장 큰 특징인 변수 정보를 모두 포함하고 있어 이용자가 데이터세트를 검색하는데 있어서 중요한 탐색 접근점으로 활용할 수 있는 여지가 크다. METS의 'dmdSec'가 기술용 메타데이터 정보를 포함하고 있고 대부분 기술용 정보들은 이용자가 해당 객체를 탐색하는데 있어서 중요한 정보를 포함해야한다는 점에서 이 두 가지 요소를 여기에 매핑하였다.

세 번째 METS 요소는 'amdSec'이다. 이 섹션은 관리적 메타데이터 요소들을 포함하는데, 기술 메타데이터(techMD), 지적재산권(rightMD), 아날로그/디지털 소스 메타데이터(sourceMD), 디지털 출처 메타데이터(digiprovMD) 요소를 활용한다. DDI 요소 중 여기에 해당할 수 있는

요소는 docDscr(문서설명), stdyDscr(연구설명), otherMat(기타자원)이다. 이 요소들은 기술대상 데이터세트를 직접적으로 언급하지는 않지만, 해당 데이터세트와 직접 혹은 간접적으로 연관된 다양한 정보자원과의 관계를 보여준다. METS의 '기술메타데이터(techMD)'의 경우 DDI의 'docDscr'에서 관련 정보들을 기술할 수 있고, 지적재산권(rightMD)의 경우 DDI 'docDscr'의 경우 'docSrc'에서 책임사항과 주석정보를 활용하여 표현가능하며, 'stdyDscr'의 'citation' 요소로도 표현가능하다. 또한 출처 메타데이터 경우 원본 연구나 저작물과의 연결이 필요하고 이 부분은 'docDscr', 'stdyDscr'에 해당 내용이 모두 포함되어 있다.

네 번째 METS 요소는 'fileSec' 섹션이다. 여기에 부합할 수 있는 DDI 요소는 'fileDscr'(데이터 파일 설명)이다. METS의 'fileSec'은 해당 디지털 객체를 구성하는 콘텐츠 파일 목록과 위치 정보를 포함하고 있다. 즉 실제 디지털 객체의 인스턴스 정보를 담을 수 있는 섹션이다. 유사하게 'fileDscr'은 데이터세트에 대한 설명을 담고 있다. 그런데 'fileDscr'의 내용은 METS의 'dmdSec(설명 메타데이터 섹션)'에도 부합하는 요소이다. 이런 관점에서보면 중복으로도 볼 수 있지만 XML Schema의 구조에서는 필요 시 반복적인 사용을 허용한다. 때문에, 본 보고에서는 'fileDscr'을 METS의 'dmdSec(설명 메타데이터 섹션)'과 'fileSec(파일 섹션)'에 반복적으로 사용하였다.

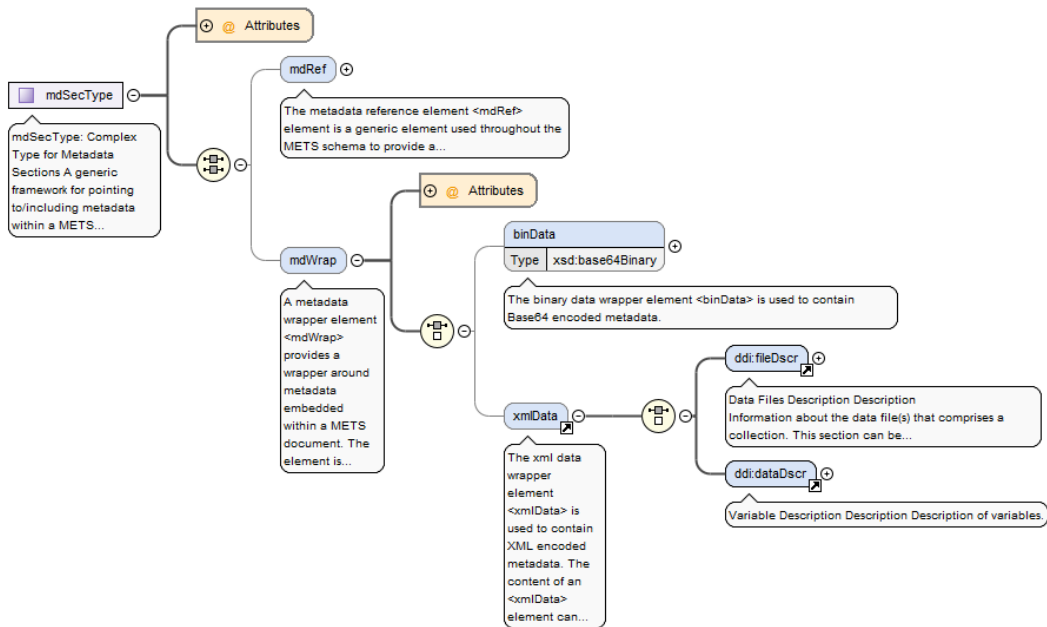
다섯 번째 METS 요소는 'behaviorSec'이다. 여기에 부합하는 DDI 요소는 'dataDscr(변수 설명)'이다. 이 요소 역시 'dmdSec(설명 메타데이터 섹션)'에서 기술했던 요소이다. 하지만

변수의 경우는 해당 데이터세트를 동작시키는데 필요한 파라미터의 일종으로 'behaviorSec'에 적합한 요소로 볼 수 있다.

매핑되지 않은 남은 METS의 요소는 structMap(구조맵), structLink(구조맵 연결)이다. METS에서 필수요소는 'metsHdr(METS 문서 헤더 섹션)'이며, 다른 요소들은 선택항목이다. 즉, 현재 구조로 METS로 인코딩하고 발행해도 표준에 어긋나지 않는다. 또한 'structMap'의 경우는 'fileSec'에서 기술한 대상 객체가 계층 구조를 가질 경우 표기 가능한 부분으로 대상 객체에 따라 활용 가능한 요소이다. 마찬가지로 'structLink'의 경우도 'structMap'에 기술된 대상 객체의 연결관계를 표현하는 요소이므로 활용 가능한 요소에 속한다.

지금까지 작업을 반영하면 새로운 METS-DDI schema 작성이 가능하다. 본 연구에서는 이 작업을 위해 'mets.xsd' 파일과 'codebook.xsd' 파일을 저장하여 활용하였다. 기본 문서는 'mets.xsd'를 활용하였으며, 이 문서 내에 'codebook.xsd' 파일을 입수(import)하여 활용하였다.

먼저 설명 메타데이터 섹션에 해당하는 'dmdSec' 부분에 DDI-C의 fileDscr(데이터 파일 설명), dataDscr(변수 설명) 요소를 추가하였다. 이를 위해서는 METS Schema 문서의 'mdSecType'을 수정해야한다. 'mdSecType'에 이를 표현하기 위한 방법은 외부 링크를 활용하는 방법과 직접 XML 문서를 표현하는 방법이 존재한다. 본 연구에서는 직접 XML 기술을 표현하는 방법을 선택했으며, 이를 위해서는 'mdWrap' 요소에 'xmlData'의 하위요소로 구성해야한다. 이를 인코딩하여 결과를 도식화하면 <그림 1>과 같다.

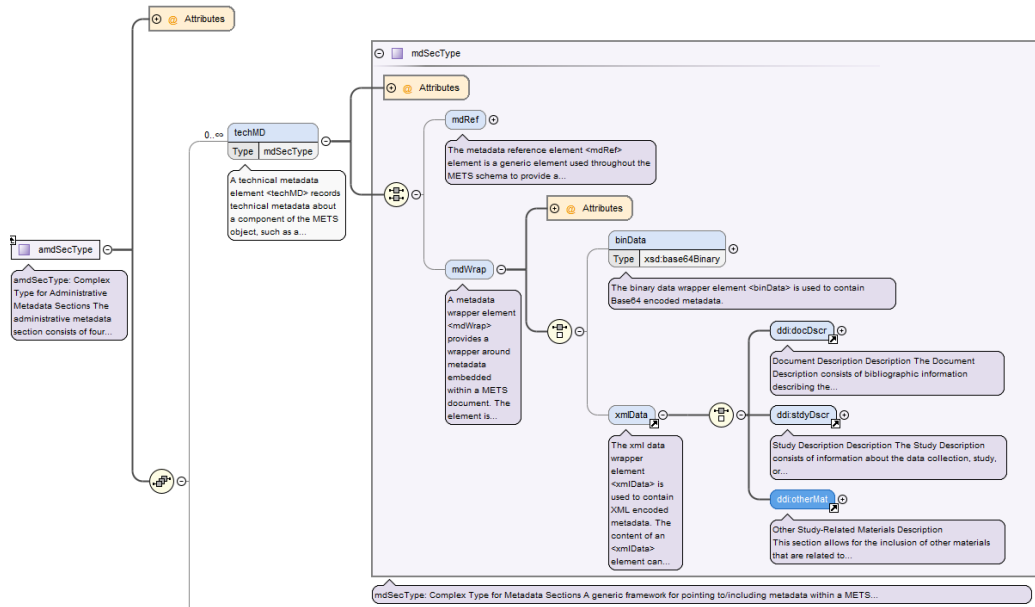


〈그림 1〉 METS Schema에 DDI fileDscr, dataDscr 인코딩한 결과

두 번째로 관리섹션에 해당하는 'amdSec' 부분에 DDI-C의 docDscr(문서설명), stdyDscr(연구설명), otherMat(기타자원) 요소를 추가하였다. 이를 위해서는 METS Schema 문서의 'amdSecType'을 수정해야한다. 'amdSecType'에 이를 표현하기 위해서는 하위요소인 기술 메타데이터(techMD), 지적재산권(rightMD), 아날로그/디지털 소스 메타데이터(sourceMD), 디지털 출처 메타데이터(digiprovMD)에 구분하여 저장해야한다. 여기서 문제는 METS에서 세분화한 하위요소별로 DDI-C의 요소를 세분화하기 어렵다는 점이다. 또 개별 요소를 모두 해체할 경우 본래의 schema 구조에 문제가 발생할 수 있다. 때문에 본 연구에서는 기술 메타데이터(techMD), 지적재산권(rightMD), 아날로그/디지털 소스 메타데이터(sourceMD), 디지털 출처 메타데이터(digiprovMD) 4개 하위요

소에 각각 docDscr(문서설명), stdyDscr(연구설명), otherMat(기타자원)요소를 모두 배치하였다. 'dmdSec'와 같이 직접 XML 기술 문서를 표현하는 방법을 선택했으며, 'mdWrap' 요소에 'xmlData'의 하위요소로 구성하였다. 이를 인코딩하여 결과를 도식화하면 〈그림 2〉와 같다. 단, 〈그림 2〉에서는 전체 그림 크기를 고려하여 'techMD'하나의 결과만을 표현하였다.

세 번째로 파일 섹션에 해당하는 'fileSec' 부분에 DDI-C의 fileDscr(데이터 파일 설명) 요소를 추가하였다. 이를 위해서는 METS Schema 문서의 'fileSecType'을 수정해야한다. 'fileSecType'에 이를 표현하기 위해서는 하위요소인 'file'을 활용해야한다. 'file'은 하위에 형식별로 'FLocat', 'FContent', 'stream', 'transformFile', 'file'이 존재한다. 데이터세트의 형식은 연구에 따라서 서로 다르고 같은 데이터세트가 서로 다른 형



〈그림 2〉 METS Schema에 DDI docDscr, stdyDscr, otherMat를 인코딩한 결과 중 일부

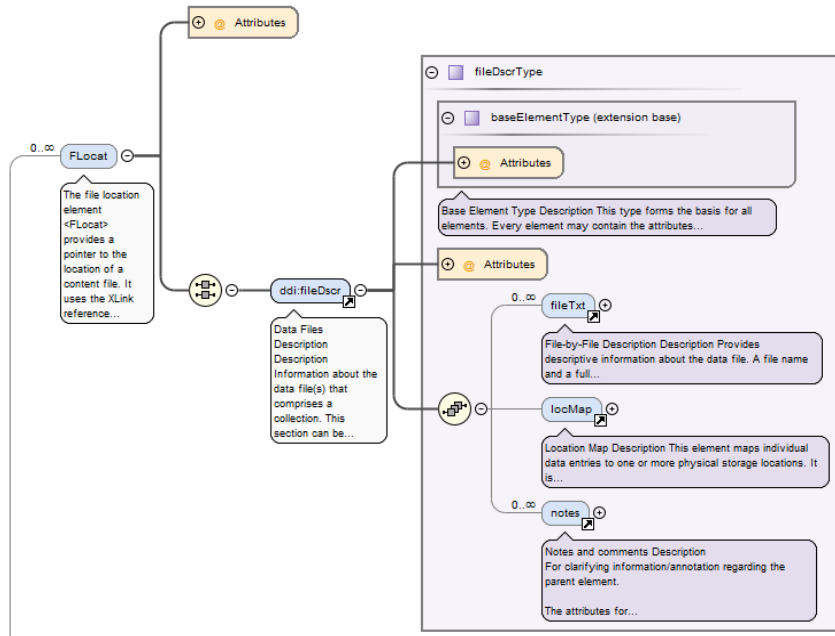
식으로 존재할 수도 있으므로 모든 하위 요소에 'fileDscr'을 배치하였다. 이 경우 FLocat, Fcontent는 별도의 요소 'type'이 정해져 있지 않아 하위요소에 기술하였다. 'stream', 'transformFile', 'file'은 'type'에 'xsd:anyType'이 설정되어 있어 이를 'fileDscrType'으로 대체하였다. 이를 인코딩하여 결과를 도식화하면 〈그림 3〉, 〈그림 4〉와 같다. 〈그림 3〉은 하위요소를 둔 경우이며, 〈그림 4〉는 'type'에 'fileDscrType'을 지정한 경우이다.

마지막은 동작 섹션에 해당하는 'behaviorSec' 부분에 DDI-C의 dataDscr(변수 설명) 요소를 추가하였다. 이를 위해서는 METS Schema 문서의 'behaviorSecType'을 수정해야 한다. 'behaviorSecType'에 이를 표현하기 위해서는 하위요소인 'behaviorSec'과 'behavior'를 활용해야 한다. 이 경우도 위와 마찬가지로 데이터

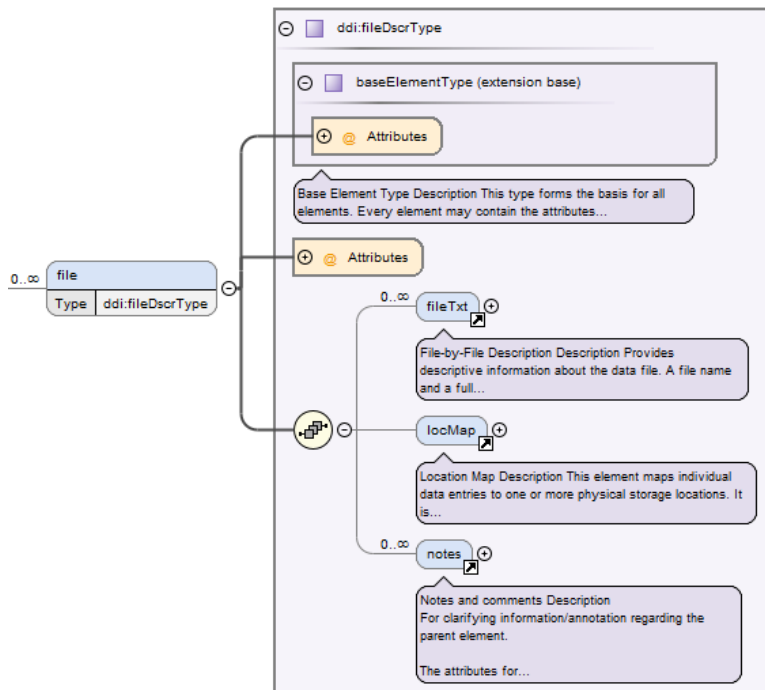
세트별로 선택할 수 있는 요소가 다를 수 있으므로 이 두 가지 요소에 모두 'fileDscr'을 배치하였다. 하위요소가 존재하지 않으므로 'type'에 'fileDscr'을 선택하여 인코딩 하였다. 〈그림 5〉는 이를 fileDscr도식화한 것이다.

#### 4. 결론

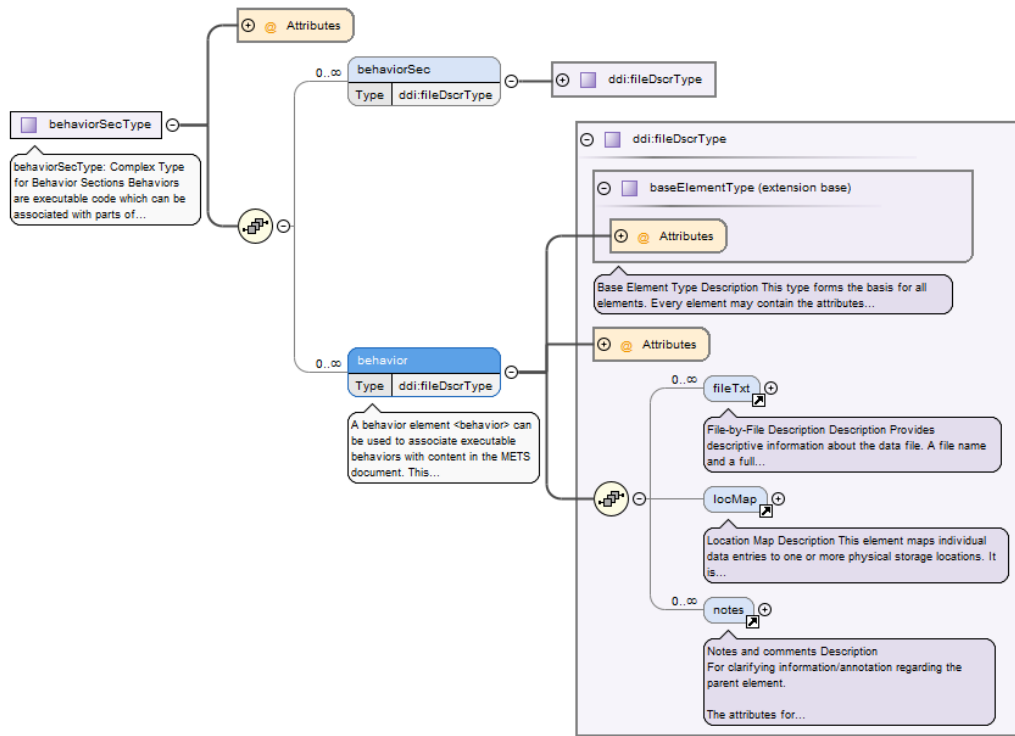
본 연구에서는 데이터 보존과 전송을 위한 표준인 METS를 활용하여 최근 새로운 정보 자원으로 활용도가 높아지고 있는 데이터세트를 표현하고자 하였다. METS는 느슨한 구조의 다양한 대상 객체를 표현하기 위한 메타데이터를 담을 수 있는 구조로 설계되었다. 데이터세트를 표현하는 가장 대표적인 표준 메타데이터는 DDI이다. 그 중에서도 DDI-C는 가장 많



〈그림 3〉 METS Schema에 DDI fileDscr를 인코딩한 결과 중 하위요소를 활용한 경우



〈그림 4〉 METS Schema에 DDI fileDscr를 인코딩한 결과 중 type을 활용한 경우



〈그림 5〉 METS Schema에 DDI fileDscr를 인코딩한 결과

이 활용되고 있는 데이터세트 메타데이터 집합으로 볼 수 있다. 그래서 본 연구에서는 DDI-C 메타데이터를 활용하여 METS에 인코딩하였다. 이를 위해 우선 METS의 구조를 분석하였다. 이는 어떤 메타데이터 정보를 입력하고 구성해야하는지를 파악하기 위한 것이다. 다음으로 DDI-C의 구조와 요소를 분석하였다. 이는 데이터세트를 관리, 보존, 서비스 및 활용하는데 필요한 요소가 무엇인지를 파악하기 위한 것이었다. 그 다음으로 METS와 DDI-C의 주요 요소들에 대한 매핑작업을 수행하였다. 여기서 기준은 최종적으로 이를 표현할 형식인 METS로 삼았다. 기준인 METS의 하위 섹션 7개(metsHdr, dmdSec, amdSec, fileSec,

structMap, structLink, behaviorSec)에 부합하는 DDI-C의 5개 요소(docDscr, stdyDscr, fileDscr, dataDscr, otherMat)를 매핑시켰다. METS와 DDI-C가 완벽하게 1:1의 매핑을 보이지 않기 때문에 기준인 METS의 각 요소들에 가장 적합하게 부합하는 DDI-C 요소를 선택하였다. 이 과정에서 fileDscr은 METS의 dmdSec, fileSec에 dataDscr은 dmdSec와 behaviorSec에 동시에 배치하였다. 이는 DDI-C의 각 요소들이 포함하고 있는 정보들이 복합적이기 때문으로 METS의 각 섹션에 나누어 설명이 가능하다. 그러나 여기서 각 요소들을 모두 분해하여 배치하면 DDI-C 표준 구조를 잃게 된다. 때문에 본 연구에서는 이 요소집합 전체를



포함하여 기술할 수 있도록 새로운 schema 구조를 설계하였다. 전체 schema 코딩 문서는 분량을 고려하여 일부만 부록에 첨부하였으며 본문에는 이를 그래프로 표현하여 그림으로 삽입하였다.

데이터세트에 대한 중요성은 연구자, 연구기관뿐만 아니라 일반 대중에게 까지 확대되고 있다. 이는 현재 정보화사회에서 데이터의 중요성이 날로 증가하고 있고 데이터를 활용할 수 있는 능력이 일반적으로 요구되는 기본 능력 중에 하나로 인식되고 있음을 반증한다고 볼 수 있다. 공급자 입장에서 본다면 이제 도서관은 데이터세트를 하나의 정보자원으로 보관, 보존, 관리, 서비스해야하며, 이를 위한 준비가 필요한 시점이라 할 수 있다. 본 연구는 이 방법

중 하나로 METS와 DDI-C를 활용한 새로운 보존, 전송 XML Schema를 설계하였다. 물론 본 연구에서 제시한 모델링 방법만이 유일한 방법은 아니다. 각 기관의 상황에 따라 혹은 설계자에 따라 다양한 방식으로 설계할 수 있다. 또 DDI 외에 다양한 메타데이터를 활용할 수 있는 방법도 존재하며, 이 역시 기관의 상황에 따라 다르게 적용가능하다. 이는 본 연구가 갖는 한계점 중의 하나로 볼 수 있다. 앞으로도 데이터세트 관리, 보존, 서비스 문제는 지속적으로 야기될 것이며, 해결방안도 다양한 관점에서 제시될 것이다. 본 연구의 결과는 데이터세트 구조 설계에 있어 참고할 수 있는 방안 중 하나로 고려될 수 있을 것이다.

## 참 고 문 헌

- 김주섭, 한연중, 유원재, 김선태 (2020). 임산공학 분야 연구데이터 관리를 위한 메타데이터 설계에 관한 연구. 한국문헌정보학회지, 54(4), 169-194. <http://doi.org/10.4275/KSLIS.2020.54.4.169>
- 김지현 (2014). 대학도서관의 연구데이터관리서비스에 관한 연구: 미국 연구중심대학도서관을 중심으로. 한국비블리아학회지, 25(3), 165-189. <http://doi.org/10.14699/kbiblia.2014.25.3.165>
- 박옥남 (2012). PREMIS 기반 보존 메타데이터 요소 개발에 관한 연구: 국립중앙도서관 디지털 자료를 중심으로. 한국문헌정보학회지, 46(2), 83-113. <http://doi.org/10.4275/KSLIS.2012.46.2.083>
- 박옥남 (2018). 연구데이터 관리를 위한 온톨로지 설계에 대한 연구. 한국기록관리학회지, 18(1), 101-127. <http://doi.org/10.14404/JKSARM.2018.18.1.101>
- 박진호, 고영만, 김현수 (2019). 연구데이터 서비스의 유용성 평가 모형 연구. 정보관리학회지, 36(4), 129-159. <http://doi.org/10.3743/kosim.pub.36.4.129001>
- 이미화, 이은주, 노지현 (2020). 연구데이터 관리를 위한 OAK 메타데이터 확장 방안 연구. 한국도서관·정보학회지, 51(3), 27-51. <http://doi.org/10.16981/kliss.51.3.202009.27>

- Cantara, L. (2005). METS: The metadata encoding and transmission standard. *Cataloging & classification quarterly*, 40(3-4), 237-253. [https://doi.org/10.1300/j104v40n03\\_11](https://doi.org/10.1300/j104v40n03_11)
- Library of Congress [n.d.]. Metadata Encoding & Transmission Standard. Available: <https://www.loc.gov/standards/mets/mets-home.html>
- McDonough, J. P. (2006). METS: standardized encoding for digital library objects. *International Journal on Digital Libraries*, 6(2), 148-158. <https://doi.org/10.1007/s00799-005-0132-1>

• 국문 참고문헌에 대한 영문 표기  
(English translation of references written in Korean)

- Kim, J. H. (2014). A study on research data management services of research university libraries in the U.S. *Journal of the Korean Biblia Society for Library and Information Science*, 25(3), 165-189. <http://doi.org/10.14699/kbiblia.2014.25.3.165>
- Kim, J. S., Han, Y. J., Youe, W. J., & Kim, S. T. (2020). A study on the design of metadata for research data management in forestry engineering. *Journal of the Korean Society for Library and Information Science*, 54(4), 169-194. <http://doi.org/10.4275/KSLIS.2020.54.4.169>
- Lee, M. H., Lee, E. J., & Rho, J. H. (2020). A preliminary study on extending OAK metadata for research data. *Journal of Korean Library and Information Science Society*, 25(3), 165-189. <http://doi.org/10.16981/kliss.51.3.202009.27>
- Park, J. H., Ko, Y. M., & Kim, H. S. (2019). A study on evaluation model for usability of research data service. *Journal of the Korean Society for Information Management*, 36(4), 129-159. <http://doi.org/10.3743/kosim.pub.36.4.129001>
- Park, O. N. (2012). A study on developing preservation metadata based on PREMIS focusing on digital data in national library of Korea. *Journal of the Korean Society for Library and Information Science*, 46(2), 83-113. <http://doi.org/10.4275/KSLIS.2012.46.2.083>
- Park, O. N. (2018). A study on ontology design for research data management. *Journal of Korean Society of Archives and Records Management*, 18(1), 101-127. <http://doi.org/10.14404/JKSARM.2018.18.1.101>

## [ 부록 ] METS와 DDI-C 모델링 코드 중 일부

### 1. mdSecType에서 ddi:fileDscr 코딩 부분

```

<xsd:element name="mdRef" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation xml:lang="en"></xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:choice>
      <xsd:element minOccurs="1" ref="ddi:fileDscr"/>
      <xsd:element minOccurs="1" ref="ddi:dataDscr"/>
    </xsd:choice>
    <xsd:attribute name="ID" type="xsd:ID" use="optional">
      <xsd:annotation>
        <xsd:documentation xml:lang="en"></xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attributeGroup ref="LOCATION"/>
    <xsd:attributeGroup ref="xlink:simpleLink"/>
    <xsd:attributeGroup ref="METADATA"/>
    <xsd:attributeGroup ref="FILECORE"/>
    <xsd:attribute name="LABEL" type="xsd:string" use="optional">
      <xsd:annotation>
        <xsd:documentation xml:lang="en"></xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="XPTR" type="xsd:string" use="optional">
      <xsd:annotation>
        <xsd:documentation xml:lang="en"></xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>

```

### 2. fileSec에서 fileGrp요소의 하위요소 file내 ddi:fileDsrType 코딩 부분

```

<xsd:element name="file" minOccurs="0" maxOccurs="unbounded" type="ddi:fileDscrType">
</xsd:element>

```

### 3. behaviorSec에서 behavior요소의 ddi:fileDsrType 코딩 부분

```

<xsd:element name="behavior" type="ddi:fileDscrType" minOccurs="0" maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>

```