행정정보 데이터세트의 장기보존을 위한 보존포맷 도입 방향*

An Direction of Introducing Preservation format for Long-term Preservation of Datasets for Administrative Information

양동민 (Dongmin Yang) (제1저자) I 전북대학교 기록관리학과 부교수, 문화융복합아카이빙연구소 연구원 I dmyang@jbnu.ac.kr
유남희 (Nam-Hee Yoo) (교신저자) I 전북대학교 기록관리학과 부교수, 문화융복합아카이빙연구소 연구원 I nhy1309@jbnu.ac.kr

목 차

초 록

최근 전자기록물의 유형이 다양해지고 대용량 파일이 증가하면서 이들을 유연하고 체계적으로 수용할 수 있는 보존정책과 보존전략이 필요해졌다. 국내에서는 실무와 밀접한 연관성을 가지고 있는 보존전략으로 마이그레이션(Migration)을 채택하고 있다. 그런데 행정 업무 수행하는데 활용되는 전자문서 유형의 전자기록물에 대해서는 마이그레이션 보존전략이 구체화되었지만, 전자문서 유형 이외에 대한 전자기록물에 대해서는 구체화되어 있지 않다. 특히, 행정정보시스템에서 생산되는 엄청난 규모의 행정정보 데이터세트에 대한 관리 및 보존 방안은 무엇보다 강하게 요구되어 왔으나 이에 대한 지침이 마련되고 있으나 아직까지는 제대로 제공되고 있지 않고 있다. 본 논문에서는 행정정보 데이터세트에 대한 장기보존을 위한 다양한 보존포맷들에 대해서 분석하고 시사점을 도출하여 행정정보 데이터세트 전자기록물에 대한 보존전략을 제시하고자 한다.

* **키워드 :** 전자기록물, 장기보존, 보존전략, 행정정보 데이터세트, SIARD

ABSTRACT

As recently electronic records have become more diverse and the number of large files increased, the need for preservation policies and preservation strategies to accommodate flexibly and systematically them has increased. Domestic preservation strategy related to practice has been migration. Although the preservation strategy has been specified for the electronic document type of records used for the administrative works, the preservation strategy for records other than the electronic document type is not. implications. In particular, management and preservation of enormous-scale datasets for administrative information produced by the administrative information system has been strongly demanded, but guidelines for this have been prepared, but it have been not provided properly. In this paper, we analyze various long-term preservation formats of datasets for administrative information, draw implications, and suggest preservation strategies for datasets for administrative information.

* **Keywords :** Electronic Records, Long-term Preservation, Preservation Strategy, Dataset for Administrative Information, SIARD

* 본 논문은 2019년 대한민국 교육부와 한국연구재단의 지원을 받아 수행된 연구임(NRF-2019S1A5B8099507).
• 논문접수일 : 2020년 8월 23일 • 최초심사일 : 2020년 8월 24일 • 게재확정일: 2020년 9월 28일

1. 서론

1.1. 연구배경 및 목적

1998년 전자결재 및 전자문서유통 활성화 기본계획 수립부터 본격적으로 추진된 전자정부 사업에 의해 대부분 기록이 전자적 형태로 생산되고 있다. 국가기록원은 전자기록 관리 프로세스를 마련하여 전자기록을 이관받아 관리하고 있다. 그러나 전자기록 관리 프로세스는 전자문서 중심으로 구체화되어 있기 때문에 행정정보 데이터세트, 시청각기록물, 웹기록물 등을 비롯한 다양한 유형의 전자기록물 관리체계는 미흡한 편이다. 특히, 국가 업무 행정 수행을 목적으로 구축한 16,400여 개(공공기관 기준)에서 생산되는 엄청난 규모의 행정정보 데이터세트를 법령상 기록관리대상이기 때문에 기록관리 실무자들로부터 행정정보 데이터세트 관리 및 보존 방안에 대한 요구가 강하게 이루어져 왔다. 그래서 최근 국가기록원은 2020년 3월 31일자로 개정된 『공공기록물 관리에 관한 법률 시행령』의 제25조 제 6항 및 제 34조의3을 신설하여 행정정보 데이터세트 관리를 위한 제도적 기반을 마련하였고, 실제 행정정보 데이터세트 관리를 위한 『행정정보 데이터세트 기록관리 기준 - 관리기준표의 작성 및 이관규격』의 표준 제정을 추진하고 있다. 현재까지는 행정정보 데이터세트를 기록관리의 대상으로 선언한 수준으로 이제 겨우 첫발걸음을 디디고 있는 상황이다. 앞으로 다양한 측면에서 깊이 있는 연구를 기반으로 체계적인 기록관리 업무절차를 마련하고 최신의 정보통신기술이 뒷받침되어야 한다. 무엇보다 실무자들이 원활하게 관리할 수 있도록 안정화 작업이 단계적으로 수행되어야 한다. 그리고 행정정보 데이터세트의 활용영역에서 뿐만 아니라 공공기관으로부터 행정정보 데이터세트가 영구기록물관리기관 등으로 이관된 이후에 보존 영역에 대한 보존전략이 필요하다.

그러나 행정정보 데이터세트는 전자문서와 달리 보존전략을 수립하는데 에는 다음과 같은 어려움이 존재한다. 첫 번째, 행정정보 시스템에서 생산되는 행정정보 데이터세트는 전자파일로 보존되지 않고 DBMS라는 소프트웨어 도구에서 관리되고 있다. 그리고 소프트웨어 내부에서는 데이터세트를 파일로는 관리하고 있지만 DBMS 종류마다 다른 포맷 다른 방식으로 관리하고 있다. DBMS는 상용 제품으로 여러 기업에서 판매하고 있으며, 각 제품마다 다양한 버전이 존재하기 때문에 일관성 있는 보존전략을 수립하기 어렵다. 두 번째, DBMS는 데이터세트를 단순히 저장하는 것이 아니고, 다수의 사용자가 실시간으로 쉽고 빠르게 접근할 수 있는 질의-응답 기능을 수행하는 것이 필수적이다. 그러므로 전자기록물의 내용 및 외관뿐만 아니라 기능까지 보존할 수 있도록 전략을 수립해야 한다. 마지막으로, 대부분 DBMS는 공통적으로 SQL(Structured Query Language)라는 질의-응답을 위한 프로그래밍 언어를 지원하고 있다. SQL 언어는 ISO/IEC 9075과 ISO/IEC 13249로 표준화되어 있지만 상용 제품들은 표준의 기본적인 기능만 지원한다. 각 기업별로 추구하는 정책이나 철학에 따라 자신만의 기능으로 확장하거나 표준에 없는 기능을 제공하기도 한다. 그러므로 SQL 표준을 기준으로 보존

하려고 해도 행정정보 데이터세트가 생성되어 활용되고 있는 모든 기능을 보존하는 것은 불가능하다. 이러한 어려움을 극복하여 행정정보 데이터세트에 대한 보존전략을 수립하기 위해서는 데이터세트에 대한 기록관리학 그리고 컴퓨터 및 정보공학 관점에서의 연구가 집중적으로 이루어져야 한다. 이에 본 논문에서는 행정정보 데이터세트에 대한 보존전략의 핵심인 보존포맷에 초점을 맞춘다. 먼저 국내외에서 데이터세트의 보존포맷으로 언급되고 있는 다수의 보존포맷들을 조사한다. 이를 통해 각 보존포맷의 장단점을 분석하고 보존포맷 도입을 위한 시사점을 도출하여 가장 적합한 보존포맷을 제시하고자 한다.

1.2. 연구범위

보존전략에서 주전략으로 마이그레이션을 채택한 아카이브 기관에서는 전자기록물의 장기보존을 위해 오랜 기간이 지나도 원본의 모습 그대로 재현이 가능한 보존포맷들을 선정한다. 국가기록원에서는 문서 유형에 대한 보존포맷을 문서보존포맷이라 지칭하고 PDF/A-1 하나의 문서보존포맷을 규정하고 있다. 반면, 행정정보 데이터세트 유형에 대한 보존포맷은 정해져 있지 않다. 본 연구에서는 국내외에서 보존포맷으로 논의되고 있는 전자파일 포맷들을 분석한다. 그 대상은 <표 1>과 같다.

<표 1> 데이터세트 유형 보존포맷 검토 대상

구분	보존포맷	표준번호	홈페이지
텍스트 유형	JSON	RFC 7159, ECMA-404	http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf
	CSV	RFC 4180	https://tools.ietf.org/html/rfc4180
	XML	XML 1.1	https://www.w3.org/XML/
스프레드시트 유형	XLSX	ISO/IEC 29500:2008, ECMA-376	https://www.ecma-international.org/publications/standards/Ecma-376.htm
	ODS	ISO/IEC 26300:2006	https://www.iso.org/standard/43485.html
관계형 DB유형	SIARD	SIARD 2.0	https://dilcis.eu/content-types/siard

2. 이론적 배경

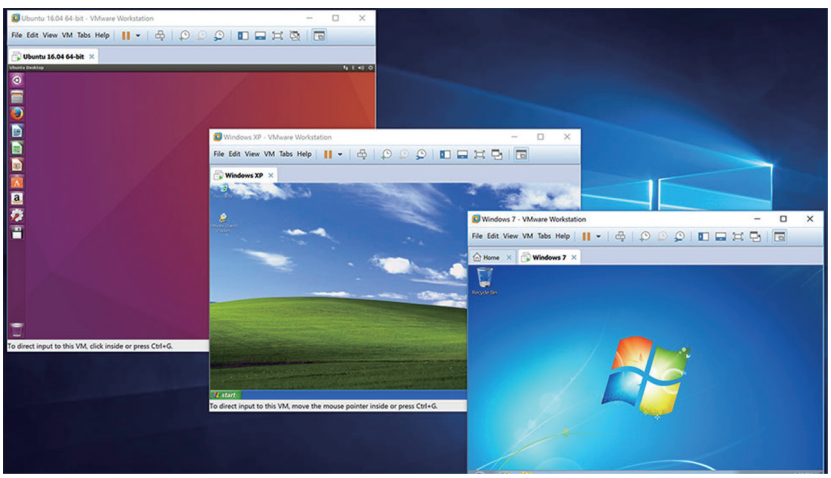
2.1. 보존전략

전자기록물의 보존전략으로 마이그레이션(Migration), 에뮬레이션(Emulation), 인캡슐레이션

(Encapsulation)이 대표적이다.

마이그레이션은 기술의 발전, 시스템 노후화로 인하여 운영체제 및 소프트웨어 업그레이드, 시스템 교체 및 업그레이드 등을 수행할 때 전자기록물이 온전하게 접근 및 재현이 가능하도록 데이터를 이관하는 전략을 뜻한다. 시스템 전체를 유지해야 하는 에뮬레이션 전략에 비하여 비용 및 기술적인 측면에서 효과적인 장점을 가지고 있지만, 안정적인 포맷을 지속적으로 모니터링해야 하며, 포맷 변환 시 비트스트림 손실이 발생할 수 있어 재현이 불가능하거나 원본의 룩앤필(Look and Feel)이 달라질 수 있다.

에뮬레이션은 하나의 컴퓨터 안에 전자기록물이 최초로 생산되었을 때와 동일한 형태의 하드웨어 및 소프트웨어로 구성된 가상의 다른 컴퓨터를 실행시키고 그 가상의 컴퓨터에서 전자기록물을 재현할 수 있도록 하는 보존전략이다. <그림 1>은 Windows 10에서 VMWare Workstation Pro를 실행시켜 (1) 리눅스(Ubuntu 16.04 64bit), (2) Windows XP, (3) Windows 7의 3개의 가상 컴퓨터를 실행시킨 화면을 보여 준다.



<그림 1> VMWare Workstation Pro 실행 화면

에뮬레이션은 위해서는 가상의 컴퓨터를 실행시킬 수 있는 에뮬레이터(Emulator)란 소프트웨어가 필요하다(김명훈 외 2013). VMWare, VirtualBox, Hyper-V, KVM 등이 대표적인 상용 에뮬레이터이다. 에뮬레이션 전략은 파일포맷의 버전 업그레이드가 필요 없고 원본 그대로의 모습을 재현할 수 있다는 큰 장점을 가지고 있다. 반면, 에뮬레이터와 함께 다양한 운영체제와 응용 프로그램을 담고 있는 대용량 시스템 이미지 파일을 보관하고 유지하기 위한 비용이 발생하며, 가상화 기술을 보유하고 있는 전문 인력도 요구된다. 그리고 시스템 노후화로 인하여 파일, 에뮬레이터, 시스템 이미지 파일의 저장매체 이관 작업은 여전히 필요하다.

대부분의 아카이브 기관에서는 마이그레이션과 에뮬레이션 중에서 하나를 주전략으로 채택하거나 하나는 주전략, 다른 하나는 부전략으로 사용하고 있다. 소정의 외(2018)에서 조사한 5개의 국립 아카이브 기관 중에서 4개 기관이 마이그레이션을 1개 기관에서는 에뮬레이션을 주전략으로 채택하고 있었다.

마지막으로 인캡슐레이션 전략이다. 인캡슐레이션은 관련 메타데이터와 무결성 메시지를 원본 문서와 함께 하나의 개체로 패키징(Packaging)하는 과정이다. 이는 향후 메타데이터를 통해 기록물을 이해하고 신뢰성과 진본성을 제공하는데 도움을 줄 수 있으며, 무결성 메시지를 통하여 기록의 무결성과 진본성을 검증할 수 있다. 인캡슐레이션 전략은 마이그레이션 또는 에뮬레이션 전략을 선택하는 것과 상관없이 사용할 수 있으며 전자기록물의 신뢰성, 진본성을 제공하기 위한 가장 일반적인 방법으로 대부분 아카이브 기관에서 도입하고 있다.

2.2. 문서보존포맷과 장기보존포맷

<그림 2>에서 보여 주듯이 문서보존포맷은 기록의 내용을 담고 있는 전자파일 포맷이고, 장기보존포맷은 문서보존포맷과 함께 메타데이터 및 무결성 메시지를 하나의 개체로 묶는 패키징 방법이다. 장기보존 관련 국제표준 ISO 14721의 OAIS 참조모델(Reference Model for an Open Archival Information System)에서 정보 모델(Information Model)의 정보 패키지(Information Package)는 장기보존포맷에, 정보패키지 안의 내용 정보(Content Information)는 문서보존포맷에 해당된다(CCSDS 2012).

<그림 2> 문서보존포맷과 장기보존포맷 개요

문서보존포맷으로 선정되려면 오랜 시간이 지나도 실행될 수 있어야 하며 문서의 내용과 모습을 그대로 재현이 가능해야 한다. 이를 위해서 공개용 표준, 편재성, 안정성, 상호운용성 등 다양한 측면에서의 체계적인 평가가 필요하다. 이러한 기준을 바탕으로 전자문서 관점에서 종합적으로 고려하여 PDF/A-1(ISO 19005-1:2005)를 문서보존포맷으로 채택하였다(국가기록원, 2008). 현재 문서보존포맷

으로 PDF/A-1 하나만 존재하기 때문에 다양한 유형의 파일로 작성된 전자기록물의 장기보존하는 데에 있어 한계가 드러나고 있다.

장기보존포맷은 문서보존포맷으로 변환된 전자기록물과 함께 기록관리 메타데이터 그리고 무결성 메시지를 하나의 개체로 묶는 방식이다. 국가기록원에서는 XML 형태로 메타데이터를 생성하고, 문서보존포맷으로 변환된 전자기록물을 Base64 인코딩하여 메타데이터의 특정 XML 요소값으로 설정한다. 이후 전자서명 및 시점확인정보의 무결성 메시지도 정해져 있는 XML의 요소값으로 설정하여 하나의 텍스트파일을 생성한다. 이를 NEO(NAK Encapsulated Object)라고 명명하였으며, 표준 규격은 2008년도에 제정되어 2017년 개정되었다. NEO는 Base64 인코딩으로 인한 패키징 속도 저하 및 저장 용량 증가에 대한 논의가 이루어지고 있다. 최근 『전자기록물 장기보존패키지 기술규격-제2부: 디렉토리로 구조화된 방식(NEO3)(v1.0)』라는 명칭으로 Base64 인코딩과 용량 증가 없는 방식으로 표준 제정을 추진하고 있다.

3. 행정정보 데이터세트의 장기보존을 위한 보존포맷 분석 및 시사점

3.1. 국외 데이터세트 유형 보존포맷 현황

<표 2>는 3개의 국외 국립 아카이브 기관에서 채택한 데이터세트 유형에 대한 보존포맷 현황을 보여 준다. CSV, ODS는 3개 기관 모두 채택하였고, JSON, XML, XLSX은 2개 기관이 채택하였고, SIARD는 1개 기관이 채택하였다.

<표 2> 국외 데이터세트 유형 보존포맷 현황

구분	보존포맷명	미국 NARA	캐나다 LAC	호주 NAA
텍스트 유형	JSON	√		√
	CSV	√	√	√
	XML	√		√
스프레트시트 유형	ODS	√	√	√
	XLSX	√		√
관계형 DB 유형	SIARD			√

3.2. 텍스트 유형 데이터세트 보존포맷(CSV, JSON, XML)

텍스트 유형 데이터세트형 보존포맷에서는 태그와 같은 일정한 표식(“<..>”, “,” “:” 등)으로 데이터와 데이터를 구분하고 기술하는 방식으로 ASCII, Unicode, EBCDIC와 같은 문자 인코딩 방식으로 표현된다.

먼저, CSV(2020)는 Comma-Separated Values의 약자로 필드를 쉼표(,)로 구분한 텍스트 파일이다. 확장자는 .csv이며, MIME(Multipurpose Internet Mail Extensions) 형식은 text/csv이다. CSV를 활용하여 <표 3>을 <그림 3>과 같이 표현할 수 있다.

<표 3> CSV에서 나타내고자 하는 예시 데이터 표
(출처: https://ko.wikipedia.org/wiki/JSON)

연도	제조사	모델	설명	가격
1997	Ford	E350	ac, abs, moon	3000.00
1999	Chevy	Venture "Extended Edition"		4900.00
1999	Chevy	Venture "Extended Edition, Very Large"		5000.00
1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	4799.00

<그림 3> CSV 예시(car.csv)

JSON(2020)은 JavaScript Object Notation의 약자로, 속성-값 또는 키-값의 쌍으로 이루어진 데이터 객체를 전달하기 위한 텍스트 기반 표준 포맷이다. 객체(object)는 중괄호({, })로 표현되며, 배열(array)는 대괄호([,])로 표현된다. 데이터 객체로 표현되기 때문에 CSV와 같은 단순히 나열하여 저장할 수 있을 뿐만 아니라 계층구조를 갖는 데이터도 나타낼 수 있다. <그림 4>는 JSON 포맷의 예시를 보여준다. "“unit”: 15 ’처럼 unit이라는 ‘속성’과 ‘15’라는 값이 “:’으로 짝지어져 있음을 알 수 있고, businessTime 속성이나 candles 속성의 값은 하나 이상의 속성-값들으로 구성되어 있는 객체(object)이다.

<그림 4> JSON 예시

텍스트 유형 마지막으로 XML(2020)은 Extensible Markup Language의 약자이며, W3C에서 개발된, 다른 특수한 목적을 갖는 마크업 언어(Mark-up)를 만드는데 사용하도록 권장하는 다목적 마크업 언어(Mark-up)이다. XML도 JSON과 마찬가지로 데이터 객체(ojbect)로 구성되며, XML에서는 이를 요소(element)라고 불린다. 태그(tag)와 내용content)으로 구성되며, 각 태그는 하나 이상의 속성(attribute)-값(value)들도 가질 수 있다.

```
  <?xml version="1.0" encoding="UTF-8"?>
- <array-list>
  - <Goods xsi:type="Goods" goodsId="GOODS-00000000000003"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <name>도로표지판</name>
        <price>3916200</price>
        <maker>test 건설</maker>
    </Goods>
  - <Goods xsi:type="Goods" goodsId="GOODS-00000000000002"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <name>무선인식리더기</name>
        <price>1760000</price>
        <maker>디지털test</maker>
    </Goods>
  - <Goods xsi:type="Goods" goodsId="GOODS-00000000000001"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <name>증명발급기</name>
        <price>7800000</price>
        <maker>증명test</maker>
    </Goods>
  </array-list>
```

<그림 5> XML 예시

<그림 5>는 XML 예시를 보여 준다. <그림 5>처럼 시작태그 및 종료태그(<a>,)안에 내용을 구성하고, 태그 내에 해당 태그와 연관되어 있는 속성들을 포함할 수 있다. 또한, XML은 적법성(Well-Formedness) 검사뿐만 아니라 유효성(Validation) 기능까지 제공하여 문법적으로는 문제가 없는지 특정 양식으로 만들어졌는지를 검증할 수 있다.

3개의 전자파일 포맷들을 보존포맷 관점에서 비교분석했을 때, 가장 행정정보 데이터세트의 장기보존을 위한 보존포맷에 적합하다고 판단되는 전자파일 포맷은 XML이다. CSV, JSON, XML 모두 인간이 읽을 수 있고(Human-Readable), 운영체제에서 별도의 프로그램 설치 없이 기본적으로 확인할 수 있는 텍스트 파일로 생성되며 세 개의 전자파일 포맷들 모두 직관적으로 이해하기 쉬운 구조로 설계되었다. 그중에서 XML은 적법성(Well-Formedness)과 유효성(Validation) 검증을 통해 기록관리 관점에서 내용 구조 무결성을 확인할 수 있다. 그러므로 텍스트 유형에서는 3개의 전자파일 포맷 중 XML이 가장 보존포맷에 적합하다고 판단된다. 그러나 XML은 행정정보시스템에 포함되어 있는 데이터는 대부분 포함할 수 있지만, SQL 기능 등은 보존될 수 없다는 치명적인 단점을 가지고 있다.

3.3. 스프레드시트 유형 보존포맷(XLSX, ODS)

최근 XML로 포맷화되어 zip으로 압축된 개방형 오피스 계열 전자파일 포맷 사용이 증가하고 있다. 이러한 전자파일은 특정 기업의 소프트웨어로만 확인할 수 있었던 과거 이진 포맷(Binary Format)과는 달리 개방형으로 표준화되어 있다. 또한, 실제 내용도 zip 압축을 해제하고 텍스트(예: 메모장, 노트패드 등) 편집기, 브라우저 등 여러 범용 소프트웨어를 통해 확인할 수 있기 때문에 호환성 및 범용성 높다.

먼저, OOXML(Office Open XML)기반의 엑셀(ODS, 2020)은 마이크로 소프트 사에서 제공하고 있는 오피스 계열 제품군 중에서 2002년 가장 먼저 사용이 되었고, 그 이후 워드(DOCX), 파워포인트(PPTX) 포맷은 오피스 2003년부터 적용되었다. 표준은 2008년 ISO/IEC 29500 국제 표준으로 승인되었다.

OOXML과 함께 대표적인 XML기반 개방형 오피스 포맷으로 ODF(Open Document Format)가 있다(ODS, 2020). 2000년대 초반, 독자포맷(hwp, doc, xls, ppt 등)이 가진 문제점인 상호호환성, 구입비용 등의 문제가 대두되었고, 이러한 문제를 해소함과 동시에, 공공영역에서 널리 사용 중인 독자포맷에 대한 종속성과 비용 부담을 해소하고자 하는 움직임이 유럽에서 시작되었다. 마이크로 소프트 사의 오피스에 대한 종속성을 탈피하고자 썬 마이크로 시스템 사를 중심으로 개방형 문서 포맷(Open Document Format)을 개발되었다. ODF는 텍스트, 스프레드시트, 프리젠테이션, 데이터베이스, 그래픽 등을 표현하는 문서 규격을 제공하고 있으며, 표준은 2006년 ISO/IEC 26300 국제 표준으로 승인되었다. ODF 계열과 OOXML 계열의 장단점을 비교하면 <표 4>와 같이 요약될 수 있다.

<표 4> ODF, OOXML 비교

구분	ODF	OOXML
목적	특정 SW에 종속되지 않는 개방형 XML기반의 문서포맷 개발	MS의 바이너리 문서포맷의 호환성과 기능성을 지원하기 위한 문서포맷의 개발
특성	Mixed Content Model 사용으로 서술적 정보표현이 용이하고 XHTML 등과 상호변환이 용이함	Non-Mixed Content Model로 구조적인 데이터를 표현하고 활용하는데 유리함
	기존 국제표준을 많이 활용함으로써 700페이지 규모의 간결한 문서표준 사양	기존의 국제표준을 활용하기보단 자체완결적인 표준을 지향하여 6,000 페이지가 넘는 방대한 표준
장·단점	OOXML에 비해 부족한 기능	기존의 바이너리 문서포맷에 대한 높은 호환성과 다양한 기능 제공
	표준이 구체적이지 않아 문서 간 상호호환성의 수준이 낮음	맞춤형 XML 스키마의 경우 상호운용성이 저해될 우려가 존재
	다양한 개발사에 의한 SW 존재	오피스 SW의 종류가 제한적임

(출처: 정제호 외, 2008)

2개 전자파일 포맷을 행정정보 데이터세트의 장기보존을 위한 보존포맷 관점에서 비교분석했을 때, 2개 모두 비슷한 수준으로 보존포맷 적합성을 지니고 있다고 판단된다. zip으로 압축되어 있지만 압축을 해제하면 텍스트 파일인 XML로 구성되어 있으므로 모두 인간이 읽을 수 있다(Human-Readable). 텍스트 유형의 CSV, JSON, XML처럼 운영체제에서 기본적으로 제공하는 소프트웨어로 생산 당시의 모습을 그대로 재현하지는 못하지만 텍스트 편집기, 그림판 등으로 전자파일의 구성요소들을 각각 확인할 수 있다. 또한 전자파일 포맷이 개방형 표준으로 공개되어 있기 때문에 언제든지 소프트웨어를 개발하여 생산 당시의 전자기록물을 재현하는 것이 가능하다. 그러나 텍스트 유형의 전자파일 포맷과 마찬가지로은 행정정보시스템에 포함된 데이터는 대부분 변환하여 보존할 수 있지만, SQL 기능 등 데이터 이외에 부분은 포함될 수 없다는 치명적인 단점을 가지고 있다.

3.4. SIARD(Software Independent Archiving of Relational Databases)

SIARD(2020)는 관계형 데이터베이스에 저장되어 있는 데이터세트를 소프트웨어와 독립적으로 하나의 파일로 보관할 수 있도록 개발된 표준이다. SIARD의 첫 번째 버전 SIARD 1.0은 2007년 SFA(Swiss Federal Archive: 스위스 연방 아카이브스)에서 개발되어 2013년에 eCH-0165라는 표준으로 되었고, SIARD 2.0은 E-ARK 프로젝트의 후원으로 SIARD 1.0기반으로 SFA가 개발하고 있으며, 2018년 기준으로 SIARD 2.1까지 완료되었다.

SIARD는 Unicode, XML, SQL:2008, URI(Uniform Resource Identifier), ZIP 등의 표준을 기반으로 개발되었다. SIARD 2.0이 개발되면서 SIARD 1.0에 비해 새로운 기능이 추가되었다. 첫 번째로 SQL:1999에서 SQL:2008로 업그레이드되면서 SQL:2008의 모든 데이터 타입을 지원하며 사용자 정의 데이터 타입(UDT: User-Defined Data Type)도 사용할 수 있다. 두 번째로 정규표현식(Regular Expression)을 사용하여 데이터 타입에 대한 규칙 준수 여부를 검증할 수 있다. 세 번째는 SIARD파일이 데이터베이스에 속하지만 외부에 저장되어 있는 대용량의 객체를 "file:" URI를 이용하여 참조할 수 있다. 마지막으로 압축 방법으로 deflate 방식을 지원한다.

표준과 더불어 SiardSuite과 같은 오픈소스로 제공하고 있다(SFA SIARD 2020). SIARD 파일 생성 과정을 설명하기 위해 <그림 6>과 같이 MySQL DBMS(Database Management System)에 3개의 테이블(도서, 고객, 주문)로 구성되어 있는 서점 스키마를 가정한다.

<그림 7>은 <그림 6>의 MySQL 서점 스키마를 SIARD 파일로 변환하는 과정을 보여 준다. 서점 스키마를 SIARD 파일로 변환하기 위해서는 가장 먼저 MySQL DBMS에 접속하고, SQL 데이터베이스 질의어 표준인 SQL:2008를 이용하여 데이터베이스의 데이터를 가지고 온다. 이 데이터는 content 폴더와 header 폴더에 일정한 규칙에 의해 XSD(XML Schema Definition)과 XML 형태의 텍스트 파

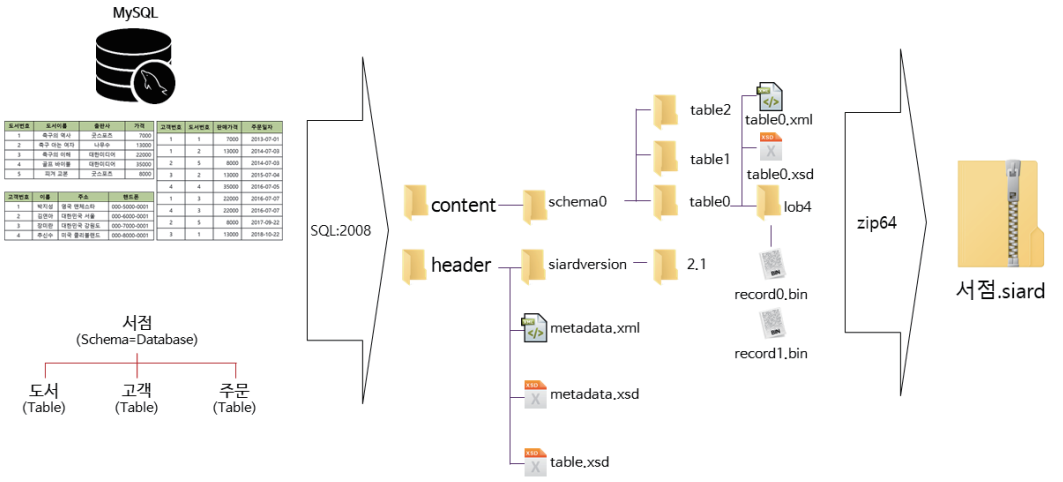
도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구 아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000

고객번호	이름	주소	핸드폰
1	박지성	영국 맨체스타	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001

고객번호	도서번호	판매가격	주문일자
1	1	7000	2013-07-01
1	2	13000	2014-07-03
2	5	8000	2014-07-03
3	2	13000	2015-07-04
4	4	35000	2016-07-05
1	3	22000	2016-07-07
4	3	22000	2016-07-07
2	5	8000	2017-09-22
3	1	13000	2018-10-22

<그림 6> MySQL의 서점 스키마

일로 저장된다. header 폴더에는 SIARD의 버전 번호와 서점 스키마의 구조가 저장되고, content는 실제 데이터가 저장된다. 예를 들어, header 폴더 내의 metadata.xml을 열어보면, 도서 테이블은 도서번호, 도서이름, 출판사, 가격의 열로 구성되어 있다는 테이블 구조, 각 열의 데이터 타입(정수, 실수, 문자열 등) 정보, DBMS 이름 및 접속 방법 등을 담고 있다. 그리고 content 폴더 내의 table0.xml에는 도서 테이블의 행들의 실제 값(1, 축구의 역사, 굿스포츠, 7000 등)들이 저장되어 있다. 또한, 데이터베이스 내에 파일이 저장되어 있을 경우에는 데이터의 값을 보관하는 요소값 란에 “file:”URI와 파일 크기를 기입한다. 그리고 그 파일이 속해 있는 테이블의 값들이 저장되는 table 폴더에 LOB(Large Object)으로 시작하는 폴더를 생성하여 record0.bin record2.txt 등으로 이름과 확장자를 변경하여 저장한다. 이후 content 및 header 폴더는 zip32 또는 zip64으로 압축 저장하고 파일 확



<그림 7> SIARD 파일 생성과정

장자 명을 'siard'로 설정한다.

SIARD 파일로 만들어진 데이터베이스의 내용을 보고 싶다면, MySQL DBMS에 원상복구를 시켜서 볼 수도 있고, siard 파일을 zip 응용프로그램을 사용하여 압축을 해제시켜 XSD 및 XML 파일을 열어서 볼 수 있다.

3.5. 행정정보 데이터세트의 보존포맷 분석 기반 시사점 도출

지금까지 행정정보 데이터세트의 장기보존을 위한 세가지 유형(텍스트 유형, 스프레드시트 유형, 관계형 DB 유형)의 보존포맷을 분석해 보았으며 이를 통해 다음과 같은 시사점을 도출하였다.

첫 번째, 텍스트 유형과 스프레트시트 유형은 행정정보 데이터세트가 관리 및 활용되고 있는 관계형 데이터베이스를 고려하지 않고 설계된 전혀 다른 형태의 전자파일 포맷이다. 즉, 행정정보 데이터세트의 데이터는 보존이 가능하지만 행정정보 데이터세트와 연결되어 있는 SQL 등의 기능은 보존될 수 없다. 그래서 텍스트 유형과 스프레드시트 유형의 전자파일 포맷은 전자기록물을 생산할 때부터 사용하거나 행정정보 데이터세트의 데이터만 보존해도 되는 경우에는 보존포맷으로 사용하기에 적합하다. 그러나 관계형 데이터베이스에 존재하는 행정정보 데이터세트는 대부분 데이터와 기능 모두 보존해야 하는 경우가 대부분이므로 적합하지 않다고 판단된다. 두 번째, SIARD 포맷은 처음부터 관계형 데이터베이스를 고려하여 개발된 표준으로 관계형 데이터베이스에 담겨있는 행정정보 데이터세트의 데이터뿐만 아니라 기능까지 모두 보존할 수 있다는 장점이 있다. 그러나 지원되는 관계형 DBMS의 종류가 한정되어 있고, SQL:2009 표준을 벗어나는 기능은 보존될 수 없다는 단점이 있다.

4. 결론

본 연구에서는 관리 및 보존 방안에 대한 필요성이 높아지고 있는 행정정보 데이터세트의 장기보존을 위해, 다양한 보존포맷들에 대해서 분석하고 시사점을 도출하였다. 현재 행정정보 시스템에 있는 엄청난 규모의 행정정보 데이터세트가 다양한 기업의 DBMS 소프트웨어에 관계형 데이터베이스 형태로 저장·관리·활용되고 있다. 관계형 DBMS의 행정정보 데이터세트를 마이그레이션 방식 기반으로 보존전략을 세우기 위해서는 다양한 기업의 DBMS 소프트웨어 종류, 소프트웨어마다 존재하는 여러 버전, 소프트웨어나 버전마다 제공하는 데이터 및 기능 등을 고려해야 한다. 여러 보존포맷을 분석한 내용을 바탕으로 다음과 같은 행정정보 데이터세트 보존포맷 방안을 제시하고자 한다. 우선, 데이터만 저장해도 되는 경우와 데이터와 기능을 모두 저장해야 하는 행정정보 데이터세트를 분류하고, 데이터만 저장하는 경우에는 XML 포맷을, 데이터와 기능을 모두 저장해야 하는 경우에는 SIARD 포맷을 사용하는 것이 적합하다. 단, 보존하고자 하는 행정정보 데이터세트의 DBMS가 SIARD 오픈소

스가 지원하는 DBMS에 포함되지 않을 경우에는, 해당 DBMS를 지원하는 과정을 거쳐 보존포맷으로 변환해야 한다.

참고문헌

김명훈, 오명진, 이재홍, 임진희 (2013). 전자기록 장기보존 전략으로서의 에뮬레이션 사례 분석. 기록학연구, (38), 265-309.

국가기록원 (2008). 기록관리업무 표준 NAK 30:2008(v1.0):전자기록물 문서보존포맷 기술규격. 대전: 국가기록원.

소정의, 한희정, 양동민 (2018). 국외 전자기록물의 장기보존 정책 비교 분석. 한국기록관리학회지, 18(4), 125-148.

정제호, 손원성, 임순범 (2008). ODF와 OOXML을 중심으로 한 사무용 전자문서 국제표준화 동향. 정보과학회지, 26(6), 20-28.

CCSDS (2012). Reference Model for an Open Archival Information System(OAIS), Recommended Practice. Issue 2. 650.0-M-2.

CSV (2020). RFC 4180. Retrieved September 12, 2020, from https://tools.ietf.org/html/rfc4180

JSON (2020) ECMA-404. Retrieved September 12. 2020, from http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf

ODS (2020). ISO/IEC 26300:2006. Retrieved September 12, 2020, from https://www.iso.org/standard/43485.html

SFA SIARD (2020). SIARD Suite. Retrieved September 12. 2020, from https://www.bar.admin.ch/bar/en/home/archiving/tools/siard-suite.html

SIARD (2020). SIARD 2.0. Retrieved September 12, 2020, from https://dilcis.eu/content-types/siard

XLSX (2020). ECMA-376. Retrieved September 12, 2020, from https://www.ecma-international.org/publications/standards/Ecma-376.htm

XML (2020). XML 1.1. Retrieved September 12, 2020, fromhttps://www.w3.org/XML/

국한문 참고문헌의 영문 표기

(English translation / Romanization of reference originally written in Korean)

Cheong, Je-Ho, Sohn, Won-Sung & Lim, Soon-Bum (2008). Study on International Standards for Two XML-Based Document Formats : ODF and OOXML. Communications of the Korean Institute of Information Scientists and Engineers, 26(6), 20-28.

Kim, Myung-Hun, Oh, Myung-Jin, Lee, Jae-Hong & Yim Jin-Hee (2013). An Analysis of Cases of Emulation for Long Term Electronic Records Preservation Strategy. The Korean Journal of Archival Studies, (38), 265-309.

So, Jeong-Eui, Han, Hui-Jeong & Yang, Dong-min (2018). A Comparative Analysis of Long. JRMASK, 18(4), 125-148.