

전자기록 관리를 위한 기본 개념 이해

2019.6.26. 임진희

전자기록 관리에 대해 학습할 영역은

- (1) 전자기록을 구성하는 객체가 무엇인지 이해하는 것
 - (2) 관리 대상 객체 중 비트스트림의 실체와 포맷의 관계를 이해하는 것
 - (3) 관리 대상 객체에 대한 정보와 관리 과정을 담게 되는 메타데이터를 이해하는 것
 - (4) 진본성 확인 및 유지, 장기보존 가능성 유지라는 목표와 방법을 이해하는 것
 - (5) 기록관리시스템의 기능 요건과 최신 기술의 적용성을 이해하는 것
 - (6) 기록관리시스템의 설계, 구축, 유지하는 과정을 이해하는 것
- 등의 영역으로 나뉘볼 수 있다.

이 강의에서는 (1)과 (2)에 중점을 두어 설명하려고 한다.

이 강의는 공공 전자기록의 관리에 초점을 두고 있고,

전자기록의 유형은 다양하나 그 중에서도 전자결재문서 유형을 중심으로 설명하고 있다.

목차

I 부: 기본 개념 정리

1. 문서의 종류와 역할
2. 전자문서의 구조와 구성
3. 기록관리를 위한 용어 정리
4. 기록 메타데이터의 종류
5. 전자기록 관리의 핵심 이슈 : 진본성과 장기보존

II 부: 파일 포맷에 대한 이해

6. ASCII코드 이해하기
7. 문서파일 덤프해보기
8. 독자포맷과 개방포맷
9. DFR과 OAIS RI

[참고사이트]

<http://home.mju.ac.kr/yimjh> [4]전자기록관리론 수업자료
<https://github.com/yimjhkr68/Python-for-RecordsFields>

I 부: 기본 개념 정리

1. 문서의 종류와 역할

공공기관들은 ‘행정효율과 협업촉진에 관한 규정’과 ‘행정업무운영 편람’에 근거하여 업무를 진행한다. 업무를 수행하는 근거, 과정의 흔적, 수행 결과 등을 문서로 남기게 된다. 전자정부는 이러한 문서를 전자문서로 남기는 것을 기본으로 한다.

전자문서에 대한 정의는 전자정부법, 전자문서 및 전자거래기본법, 행정효율과 협업촉진에 관한 규정 등에 거의 유사하게 기술되어 있다. 공통적으로 ‘정보처리시스템’으로 “작성, 송수신, 저장”된 “정보 또는 문서”라고 설명된다.

전자정부법	7. "전자문서"란 컴퓨터 등 정보처리능력을 지닌 장치에 의하여 전자적인 형태로 작성되어 송수신되거나 저장되는 표준화된 정보를 말한다.
전자문서 및 전자거래 기본법	1. "전자문서"란 정보처리시스템에 의하여 전자적 형태로 작성, 송신·수신 또는 저장된 정보를 말한다.
행정효율과 협업촉진에 관한 규정	1. "공문서"란 행정기관에서 공무원 작성하거나 시행하는 문서(도면·사진·디스크·테이프·필름·슬라이드·전자문서 등의 특수매체기록을 포함한다. 이하 같다)와 행정기관이 접수한 모든 문서를 말한다. 2. "전자문서"란 컴퓨터 등 정보처리능력을 가진 장치에 의하여 전자적인 형태로 작성되거나 송신·수신 또는 저장된 문서를 말한다.

(현재의 정의를 놓고 보면 공문서의 기본은 종이문서인 것처럼 되어 있다. 규정에서 “~등의 특수매체기록을 포함한다”라고 하고 있기 때문이다. 이는 디지털시대에 제대로 조응하지 못한 정의라 생각한다. 전자정부가 기본인 것으로 하여 법령을 개선할 필요가 있다고 판단된다. 또한, 규정의 공문서와 공공기록법의 기록물에 관한 정의에서 콘텐츠의 종류와 매체의 종류를 섞어놓는 방식이어 혼란을 부추기므로 정돈이 필요하다.

규정의 전자문서와 공공기록법 시행령의 전자기록물 정의를 살펴보았을 때, 전자기록물이 전자문서를 포함하면서 확장하여 웹기록물과 행정정보 데이터세트 등을 더 포괄하는 것처럼 읽힌다. 하지만, 전자정부법과 전자문서 및 전자거래기본법에서 정의한 전자문서에는 데이터 세트류가 기본으로 포함되고 있어 이 또한 혼란을 불러올 수 있다.)

공공기록물관리에 관한 법률	2. "기록물"이란 공공기관이 업무와 관련하여 생산하거나 접수한 문서·도서·대장·카드·도면·시청각물·전자문서 등 모든 형태의 기록정보 자료와 행정박물관(行政博物館)을 말한다.
동 시행령	2. "전자기록물"이라 함은 정보처리능력을 가진 장치에 의하여 전자적인 형태로 작성하여 송신·수신 또는 저장되는 전자문서, 웹기록물 및 행정정보 데이터세트 등의 기록정보자료를 말한다.

문서의 종류는 다음과 같다.

- 법규문서, 지시문서, 공고문서, 비치문서, 민원문서, 일반문서
- 내부결재문서 vs (대내문서, 대외문서, 발신자와 수신자가 명의를 같은 문서)

편람에 따르면, 공공기관에서 문서의 역할은 다음과 같다.

- 1) 의사 기록 - 구체화
- 2) 의사의 전달
- 3) 의사의 보존
- 4) 자료 제공
- 5) 업무의 연결 - 조정

업무 추진에 사용하는 정보시스템에는 업무관리시스템, 전자문서시스템과 행정정보시스템이 있다. 하나의 기관에서는 업무관리시스템이나 전자문서시스템 중 하나를 결재시스템으로 채택하고 있다. 결재시스템 외에 업무처리를 위해 사용하는 모든 시스템을 행정정보시스템이라 부른다.

이 강의에서는 업무관리시스템이나 전자문서시스템에서 생산되는 일반문서 중 내부결재문서를 중심으로 설명하고자 한다.

결재란 조직의 의사결정과정을 의미하며, 결재를 마친 문서는 공문서로서의 효력을 띠게 되고, 그 즉시 기록으로 등록되어 관리되어야 한다.

2. 전자문서의 구조와 구성

시민들은 정보공개포털이나 정보소통광장의 원문공개를 통해 쉽게 공문서에 접할 수 있다. 다음은 정보소통광장(opengov.seoul.go.kr)에서 조회한 원문 사례이다. 결재문서본문과 첨부 1개로 구성되어 있다.

결재문서본문.hwp

투명하고 신뢰받는 청렴서울, 천만시민의 자랑입니다.

I·SEOUL·U
이·서울·유

서울역사박물관



수신 국가기록원장(공개서비스과장)

(경유)

제목 국가기록원 자료이용 허가 요청

1. 귀 기관의 무궁한 발전을 기원합니다.

2. 서울역사박물관에서 제작 중인 2018년 서울생활문화자료조사 반포본동 보고서 발간과 관련하여 국가기록원 소장 자료의 이용허가를 요청하오니 협조하여 주시기 바랍니다.

가. 대상자료 : 1건

관리번호	생산년도	생산기관	첨 제 목	기록물유형
C*****	1971년	국립영화제작소	대한뉴스 제843호	시청각기록물

나. 사용목적: 2018년 서울생활문화자료조사 반포본동 보고서 내지 수록
- 「대한주택공사 남서울아파트 건설 기공」 스틸컷 2컷(붙임 참조)

붙임 : 허가요청 자료 내역 1부. 끝.

서울역사박물관장

주유관
최보영
조사연구과장
03/18

박상빈

협조자

시행 조사연구과-353 () 접수 ()

우 03177 서울특별시 종로구 새문안로 55 (신문로2가) / www.museum.seoul.kr

전화 02-724-0139 / 전송 02-724-0206 / beyoung@seoul.go.kr / 대시민공개

허가요청자료내역(스틸컷).hwp

「대한주택공사 남서울아파트 건설 기공」 스틸컷 2컷



정보소통광장에서는 원문과 함께 다음과 같은 메타데이터를 제공한다.

* 본 문서는 공문서로서의 법적 효력은 없으며, 위조·변조·도용 등 불법적 활용으로 인하여 발생된 모든 책임은 불법적으로 활용한 자에게 있습니다

첨부된 문서

 결재문서본문 (0.03MB)	문서보기 원문
 허가요청 자료 내역(스틸컷) (2.78MB)	문서보기 원문

문서 설명

국가기록원 자료이용 허가 요청

문서 정보

공개구분	공개		
원본시스템	서울시		
분류	문화관광 > 문화재보존정책 > 박물관운영 > 향토사적조사연구 > 서울생활문화자료조사 같은 분류 문서 보기		
제공부서	조사연구과	작성자	최보영
전화번호	02-724-0139	등록일	2019-03-19
저작권		생산일	2019-03-18
보존기간	5년		
문서번호	조사연구과-353		
관리번호	D0000035801992		

다음은 서울시 업무관리시스템에서 확인한 문서관리카드의 정보이다.

문서관리카드조회-국가기록원 자료이용 허가 요청 - 행정포탈 [sportal72]

인기문서로 추천 재작성 시행문보기 이력보기 본문복사 회의정보 PDF변환 달기

결재정보 **본문** **공람**

1 문서정보 ☐ 열람시암호확인 ☐ 문서보안(DRM)

제목	국가기록원 자료이용 허가 요청	☐ 긴급
문서번호	조사연구과-353	
과제카드	단위 서울생활문화자료 조사 [5년] (서울생활문화자료조사)	
문서취지		
공개여부(시민)	☒ 대시민공개 ☐ 부분공개 ☐ 비공개 ☒ 행정안전부 문서제목 공개	
열람범위(내부)	☒ 기관 ☐ 부서	
열람제한(보안)	☒ 설정안함 ☐ 결재중 ☐ 제한종료일	

2 결재경로

순번	처리방법	직위(직급)	처리자	의견	처리상태	처리일시	본문버전
2	결재	조사연구과장	박상빈		완료	2019-08-18 16:08	
1	기안	주무관	최보영		완료	2019-08-18 15:06	1.0

3 시행정보

수신자	국가기록원장(공개 서비스과장)		
경유			
기관명	서울역사박물관	발신명	서울역사박물관장
시행종류	대외시행	발송구분	전자발송
☐ 자동발송 일괄기안, 첨부관인 사용불가			

4 관련정보

첨부

- 첨가
- PC저장
- PDF변환저장 (0 KB / 3 MB)

허가요청
자료 내역(...)

문서 사례에서 살펴보는 바와 같이 하나의 전자문서는 하나의 본문파일과 0개 이상의 첨부파일, 그리고 메타데이터로 구성되어 있다.

본문파일은 기안문 서식에 맞춰 작성되어 있으며, 결재과정에서 기입되거나 획득된 메타데이터와 콘텐츠(content)로 구성되어 있다. 기록의 일부 메타데이터는 본문파일에 콘텐츠의 일부로 포함되기도 한다.

(기안문 서식은 과거 종이문서 시절부터 형성되어 온 것이다. 전자문서로 바뀌면서 문서관리 카드에 메타데이터를 분리하고 순수한 콘텐츠만 본문파일에 넣는 방식으로 변경되었어야 하는 게 아닌가 생각해본다. 혹시, 본문파일에 몇 몇 중요 메타데이터를 확보해 두는 것이 문서의 진본성 유지나 확인에 필요하기 때문에 서식을 포기할 수 없다고 한다면 그것은 다른 대안이 있을 수 있다. 서식이 있어서 업무나 유통에 유리한 점이 무엇인지 확인해보아야 한다.

콘텐츠, 라는 단어는 ‘콘텐츠’라는 단어가 많은 혼란을 불러온다는 지적에 따라 영어발음 그대로 표기하는 방식으로 정한 것이다. 기록의 실제 내용을 담고 있는 객체로서 주로 본문파일과 첨부파일 자체를 지칭한다.)

3. 기록관리를 위한 용어 정리

전자문서를 기록관리 대상으로 인식할 때, 구성요소의 종류별로 구분하여 이름을 지을 필요가 있다. 구성요소별 용도와 특성이 다르기 때문이다.

이 강의에서는 전자기록 구성요소별 명칭에 대해 다음과 같이 정하고자 한다.

- (1) 기록 건(Record Item) : 전자결재를 마친 공문서 하나를 전체적으로 지칭. 시스템 상으로는 기록 건 메타데이터가 데이터베이스에 입력되고 조회 시 목록으로 표현되며, 컴포넌트 정보가 건에 포함되는 하위 정보로 함께 보여짐
- (2) 컴포넌트(Component) : 결재문서를 구성하는 본문파일, 첨부파일들이 각각 하나씩의 컴포넌트가 됨. 즉, 기록 건 : 컴포넌트 = 1: 다. 시스템 상으로는 각 컴포넌트별 메타데이터가 데이터베이스에 입력되고 건 하위에 목록으로 표현됨
- (3) 콘텐츠(Content)는 컴퓨터 파일로 스토리지 등 저장매체에 저장되고, 시스템에는 해당 콘텐츠의 이름과 위치 정보가 관리되어야 함

결국, 기록으로서 하나의 공문서에 대해 관리할 대상 객체는 3 종류로 나뉘게 된다.

- (1) 기록 건에 대한 메타데이터
- (2) 컴포넌트에 대한 메타데이터
- (3) 콘텐츠(컴퓨터 파일)

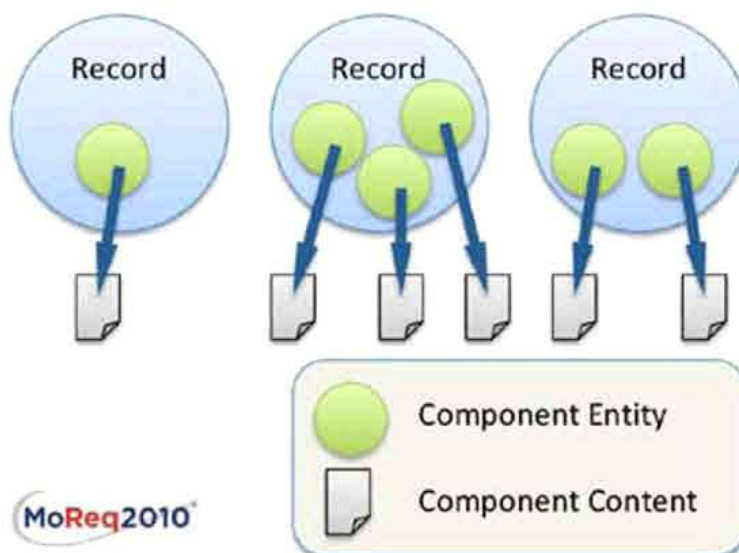
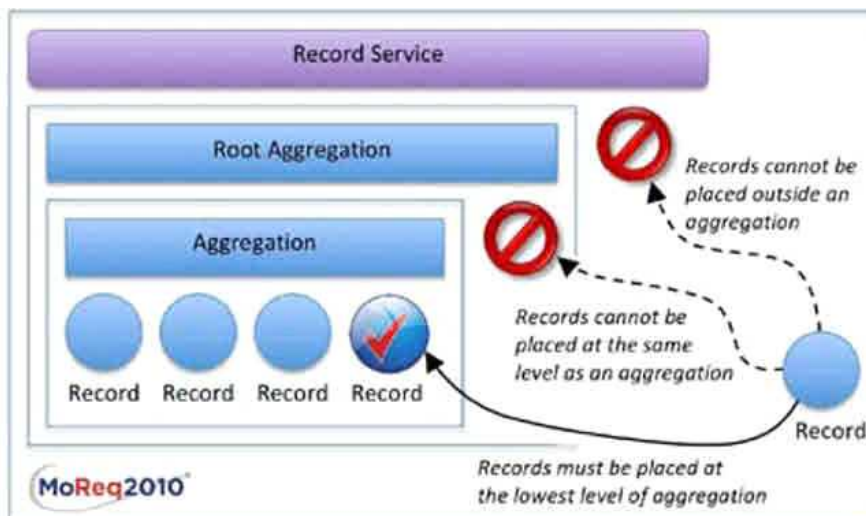
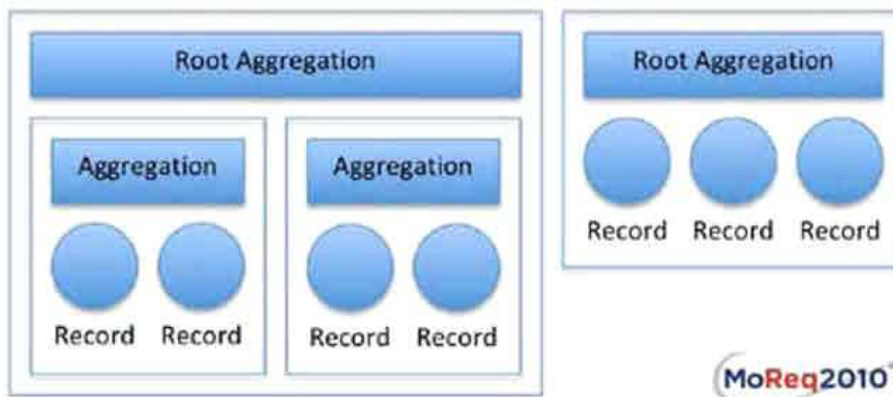
기록 건이 여러 개 모여서 집합체(Aggregation)를 이루게 된다.

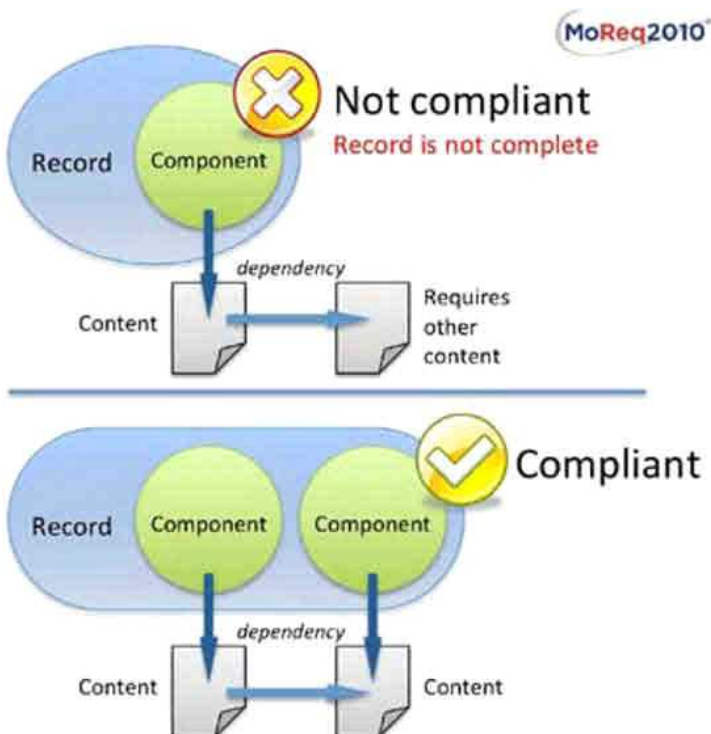
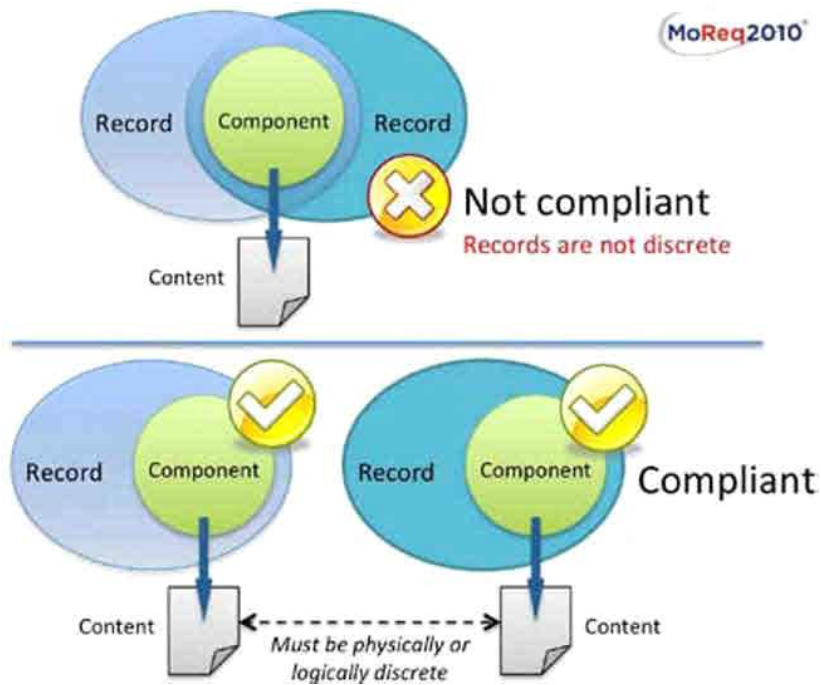
하나의 단위과제카드에 모인 문서 덩어리를 기록물철로 관리하고 있으며, 해마다 동일한 단위과제에서 만들어진 기록물철들은 시리즈(Series)로 묶어 관리할 때 기록물철, 시리즈 등이 집합체에 해당한다. 집합체는 계층구조를 갖게 되고 메타데이터와 레코드스케줄을 상속받을 수 있다.

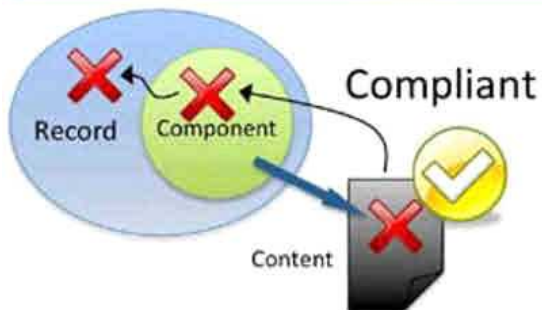
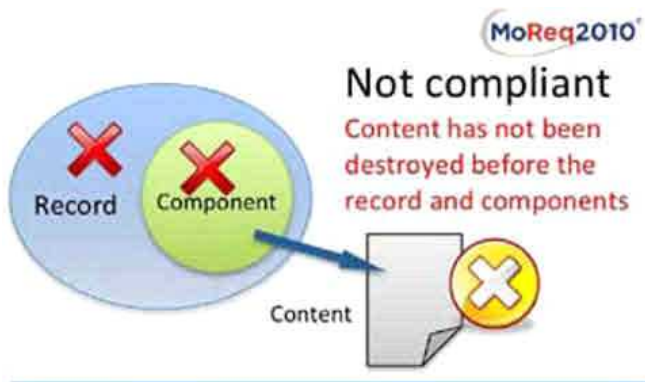
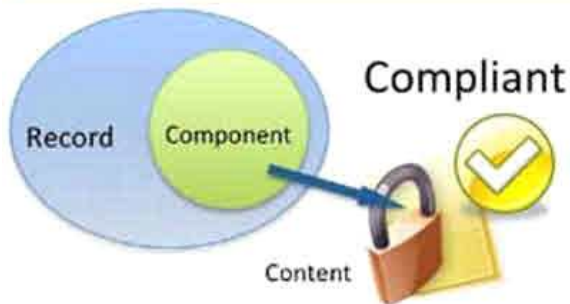
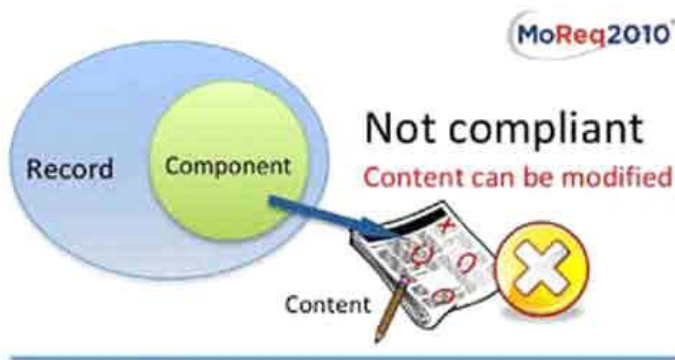
기록관리 메타데이터 표준(NAK 8:2016(v2.1))에서는 기록계층(Aggregation Level)이라고 명명하면서 기록물을 기술하거나 통제하는 계층으로 기록물철, 기록물건, 컴포넌트를 지정하고 있다.

영구기록물 기술규칙NAK 13:2011(v2.0)에서는 기술계층(Level of description)이라고 명명하면서 일반적인 영구기록물의 분류체계 상 기록물군이나 컬렉션, 기록물계열, 기록물철, 기록물건 등을 지정하고 있다.

[참고] MoReq2010의 용어 (https://moreq.info/files/moreq2010_vol1_v1_1_en.pdf)







4. 기록 메타데이터의 종류

전자기록 관리의 대상이 되는 객체는 온갖 종류의 메타데이터와 기록의 내용을 담고 있는 콘텐츠 두 가지로 나뉜다.

메타데이터는 데이터에 대한 데이터라 정의할 수 있다.

기록의 메타데이터란 기록의 내용 그 자체와는 대별되는 정보이다. 즉, 기록에 관한 여러 측면에서 필요한 추가적인 정보를 의미한다. 기록관리 메타데이터 표준(NAK 8:2016(v2.1))에서는 기록관리 메타데이터(Metadata for managing records)란 시간과 공간을 초월하여 기록의 생산, 관리와 이용이 가능하도록 하는 구조화된 혹은 반구조화 된 정보라고 정의하고 있다.

메타데이터는 역할에 따라 두 종류로 나뉜다.

설명적인 정보로서의 메타데이터와 통제의 기준이 되는 메타데이터이다.

내용메타데이터나 맥락메타데이터들은 기록에 관한 설명적인 정보를 갖고 있게 된다. 예를 들어, 누가 언제 이 기록을 만들었고 제목과 주제가 무엇이다 라는 정보이다.

반면 관리메타데이터 중에는 통제의 기준이 되는 정보가 상당수 있다. 처분일시와 처분행위 같은 메타데이터의 경우, 처분일시가 되면 시스템에서 기록관리자에게 해당 기록의 목록을 띄워주고 정해진 처분행위를 할 때가 되었음을 알려주도록 하거나, 혹은 미리 설정된 경우 자동으로 실행하도록 할 수 있다.

기록관리자는 통제의 기준이 되는 메타데이터를 잘 설정하여 자동화 효율화된 기록관리를 할 수 있도록 시스템을 구성할 필요가 있다.

메타데이터 표준의 경우 다음과 같이 개선될 필요가 있다.

첫째, 다중 엔티티모형으로 변경되어야 한다는 것이다. 예를 들어, 현재는 6.6 주제(Subject) (정의: 기록물에 포함된 중요한 내용을 주제어를 통해 기술, 목적:기록물의 내용에 대한 주제어를 통해 기록물을 검색할 수 있는 접근점 제공)에 대해 적용기록 계층이 기록물절, 기록물건이라고 되어 있고, 6.1 주제 유형(Subject Type), 6.2 주제명(Subject Words)의 하위요소가 정의되어 있다. 기록에 대해 주제분류를 하고자 한다면, 주제분류값이 별도의 엔티티로 정의되어야 하고, 해당 주제에 속하는 기록물 건이나 기록물절에 대한 관계정보를 관리해야 한다. 기록물만을 중심으로 보아서는 기관 차원에서 기록의 주제분류를 관리해야 한다는 것이 소홀해질 수 있고, 데이터구조 상으로도 맞지 않게 된다.

메타데이터 표준의 개선 방향 두 번째는, 기록 유형별 확장적인 메타데이터 구조가 가능해야 한다는 점이다. 기록유형 자체를 코드화하고, 기록유형별로 필수메타데이터와 선택메타데이터를 정할 수 있도록 해야 한다. 기록유형은 미래시점에 새로운 것이 추가될 수 있다는 점을 고려하여 시스템을 설계해야 한다. 오라클 같은 RDBMS를 사용하는 경우 메타데이터 항목이 추

가되는 융통성을 구현하기 어렵다. NoSQL 데이터베이스를 채택하여 메타데이터를 관리하는 것을 시도해볼 필요가 있다.

다음으로 메타데이터와 기술(Description)의 차이에 관해 생각해 보고자 한다.

앞에서 살펴본 바와 같이 국가기록원이 제정한 **메타데이터 표준**에서는 메타데이터를 기술하는 대상 계층을 기록물철, 기록물건, 컴포넌트 3계층으로 지정하고 있다. **영구기록물 기술규칙**에서는 기술 단위를 기록물군이나 컬렉션, 기록물 계열, 기록물철, 기록물건 등으로 정하고 있다.

전자기록 시대에 메타데이터항목과 기술항목은 다른 것인가? 메타데이터 항목에 값을 입력하는 것과 기술하는 행위는 다른 것인가?

종이기록 중심일 때의 메타데이터는 어떻게 기술되고 활용되었을까? 전자기록 중심일 때, 혹은 전자기록관리시스템을 이용하여 기록을 관리하고자 할 때의 메타데이터는 어떻게 기술되고 활용되어야 할까?

종이기록에서는 육안으로 식별되는 기록의 실체를 중심으로 메타데이터를 기술하는 것이 자연스러웠을 것이다. 철별로 그 안에 편철된 기록의 제목과 작성자, 작성일 등을 기술해 놓고, 보존박스 별로 그 안에 보관된 철의 목록을 기술해 놓고, 서고의 서가 별로 서가에 보관된 보존박스의 목록과 위치정보를 기술해 놓았을 것이다. 여기까지는 메타데이터 항목으로 설명된다.

기록관리기관이 영구기록물관리기관이라면 기록물철들을 시리즈, 레코드그룹으로 묶는 등 집합체구성을 한 후 시리즈 단위에 대한 설명과 레코드그룹 단위에 대한 설명 정보를 추가해 둘 것이다. 건 상위의 집합체에 관한 설명은 주로 기술항목으로 설명되고 있다.

ISO23081과 ISAD(G)의 항목들을 비교해보면 메타데이터와 기술항목 간에 겹치는 부분이 많다. 예를 들어, 메타데이터 표준의 6.5 기술(Description) 항목의 경우, 기록물의 내용이나 목적에 대한 자유로운 설명이라고 정의되며, 기록물의 내용에 대한 정보를 제공함으로써 제목으로 표현하지 못하는 기록물 내용에 대한 검색을 지원하며, 기록물에 대한 이해도를 제고하기 위한 목적의 항목으로, 적용되는 기록 계층은 기록물철과 기록물건이라고 소개되어 있다. 기술항목은 하위 요소 기술 유형(Description Type)과 기술 내용(Description Text)로 구성된다.

기술하는 행위는 종이기록 시절에 먼저 행해지던 아키비스트의 관리행위였을 것이다. 기술 내용이 시스템화되면서 세부적인 메타데이터의 항목으로 나뉘어 정의되고, 그 밖의 설명정보를 기술 항목에 입력하는 방식으로 발전해 온 것이다. 물론 기존에 관리하지 않았지만 디지털 환경에서 필요해진 여러 정보 항목들을 메타데이터에 추가하는 방식으로 발전해왔을 것이다.

결국 기록관리시스템을 이용하여 기록을 관리하는 체계에서는 메타데이터의 입력이나 기술하는 행위는 결국 동일한 활동이라고 보여진다. 발생적 차이, 관점의 차이, 아키비스트 입장에서의 용도 차이만 있을 뿐 근본적으로는 동일하다고 보여진다.

5. 전자기록 관리의 핵심 이슈 : 진본성과 장기보존

우리나라 공공영역에서 전자기록 관리의 두 가지 핵심 목표는 (1) 진본성 확인 및 유지, (2) 장기보존 가능성 유지라고 할 수 있다.

전자기록은 콘텐츠의 복사와 가공이 쉽다보니 원래의 상태를 유지하고 있는지 확인할 수 있는지 여부가 매우 중요하다. 진본성에 대한 정의, 진본성 추정의 기준, 진본 유지 방법, 진본사본의 재생산 가능성 유지 등에 관해 전세계적으로 오랫동안 논의해 왔다.

InterPARES(<http://interpares.org/>) 프로젝트의 결과를 바탕으로 하여 국내 전자기록 진본성, 장기보존 관련 논의도 풍성해졌다. 해외 사례를 기반으로 참여정부 시절 진본성과 장기보존에 관한 프로세스 개선 및 시스템 혁신을 추진하게 되었다.

전자기록의 진본성을 확인한다는 것은 기록 자체가 원래 목적되었던 바로 그것이었는지, 콘텐츠가 무결한지를 확인하는 것이다. 보유 중인 기록이 원래의 진본이 맞느냐를 확인한다기 보다는 여러 근거에 기반하여 진본이라 추정할 수 있느냐를 따지는 것이다. 진본 여부를 판단할 때는 기록의 메타데이터와 관리이력, 내용 모두를 점검하게 된다.

기록보관소에서는 보유한 기록이 진본임을 증빙하기 위한 여러 조치를 취해야 한다. 입수 기록이 진본인지 확인하고, 이후 진본인 상태로 유지를 해주어야 한다. 진본성 유지를 위한 관리행위로는 적절한 메타데이터를 생성하고 유지하는 것, 기록 객체를 관리하는 절차와 프로세스를 잘 정의하여 진본성 확인 시 의심의 여지가 없게 만들어야 하는 것, 접근통제와 보안기술을 접목한 시스템을 구축하여 안전하게 보관하고 있음을 증빙하는 것 등이 포함된다. 기록에 행해진 여러 관리 행위에 대해 이력정보를 잘 남기고 이를 기록과 함께 제시할 수 있어야 한다.

전자기록의 장기보존 과제는 컴퓨팅 환경이 빠르게 변화한다는 점 때문에 점점 어려운 과제가 되고 있다. 전자기록을 100년 후에도 지금처럼 열어볼 수 있기 위해 어떤 조치를 취해야 할 것인가를 고민하는 것이다.

예를 들어, 불과 30년 전 보석글, 훈민정음 문서편집기로 만든 문서파일을 지금 열어볼 수 있는가? 지금 만드는 hwp문서파일을 100년 뒤에도 열어볼 수 있을까? 라는 질문에 답하려는 것이다.

현재의 컴퓨팅 환경을 구성하는 소프트웨어/하드웨어가 더 이상 존재하지 않게 되었을 때에도 기록의 내용을 확인할 수 있도록 지속적으로 기술의 변화를 관찰하고 마이그레이션, 에뮬레이션, 인캡슐레이션 등의 전략을 잘 선택하여 실행해 가야 한다.

이 강의에서는 진본성과 장기보존에 관한 고민의 첫 출발점으로 기록 객체의 특성을 파악하는데 중점을 두고자 한다. 특히, 비트스트림의 실체에 관해 들여다 보기로 한다.

II 부: 파일 포맷에 대한 이해

6. ASCII코드 이해하기

6.1 십진수

우리가 일상생활에서 사용하는 숫자는 십진수(十進數)이다.
다음은 십진수이다. 읽어보자.

- 340
- 5,130
- 3,456,234
- 23,787,654,345

십진수(Decimal Number) 0,1,2,3,4,5,6,7,8,9 등 열 종류의 수를 이용하여 표현한다.
한 자리마다 1의 자리, 10의 자리, 100의 자리로 값이 매겨진다.
즉, 10의 0제곱, 10의 1제곱, 10의 2제곱등으로 계산된다.
참고로, 모든 수의 0제곱을 한 값은 1이다.

- $340 = 0 \times 1 + 4 \times 10 + 3 \times 100$
- $5,137 = 7 \times 1 + 3 \times 10 + 1 \times 100 + 5 \times 1000$

6.2 이진수

이진수(Binary Number)란 이진법(二進法)으로 표시된 수를 의미하며 0,1 두 종류의 수를 이용하여 표현한다.
한 자리마다 1의 자리, 2의 자리, 4의 자리가 된다.
즉, 2의 0제곱, 2의 1제곱, 2의 2제곱등으로 계산된다.

아래 이진수의 값을 십진수로 계산해보면 다음과 같다.

- $1011 = 1 \times 1 + 1 \times 2 + 0 \times 4 + 1 \times 8 = 11$
- $1110 = 0 \times 1 + 1 \times 2 + 1 \times 4 + 1 \times 8 = 14$
- $101010 = 0 \times 1 + 1 \times 2 + 0 \times 4 + 1 \times 8 + 0 \times 16 + 1 \times 32 = 42$

동일한 값은 여러 진법으로 변환할 수 있다.

아래 십진수의 값을 이진수로 변환하면 결과는 다음과 같다.

십진수 값을 2로 나눌 수 있을 때까지 나누면서 나머지값들을 나열하면 된다.

- $2 \rightarrow 10$

- 5 -> 1001
- 15 -> 1111
- 40 -> 101000

6.3 비트스트림과 십육진수

컴퓨터는 연산을 할 때도 기억을 할 때도 모든 정보를 비트단위로 처리한다.

비트(Bit)란 Binary Digit의 약자이다.

어떤 정보를 표현하기 위해 비트들이 주욱 나열되어 있는 것을 비트스트림(Bitstream)이라고 부른다.

컴퓨터는 우리가 이해하는 문자 자체가 아니라 비트스트림만을 취급하며, 엔지니어들은 이진수의 긴 비트스트림을 축약하여 십육진수로 변환하여 보는 경향이 있다.

이진수 4개가 모이면 하나의 십육진수로 바로 변환이 가능하므로 직관적이다.

십육진수에는 0123456789ABCDEF 등 16개 문자가 사용된다.

다음의 비트스트림을 십육진수로 변환하면 다음과 같다.

- 01110010 => 0111 0010 => 7 2
- 10100111 => 1010 0011 => A 3
- 10111111 => 1011 1111 => B F

다음의 십육진수를 이진수, 십진수로 변환해 보면 다음과 같다.

- 4A0 => 0100 1010 0000, $0 \times 1 + 10 \times 16 + 4 \times 16 \times 16 = 416$
- 23EF => 0010 0011 1110 1111, $15 \times 1 + 14 \times 16 + 3 \times 16 \times 16 + 2 \times 16 \times 16 \times 16 = 489,693,184$

6.4 정보를 비트로 표현하기

Yes/No 혹은 True/False를 구분하여 표기하기 위해서는 몇 개의 비트가 필요할까?

1비트면 가능하다. 예를들어, 1은 “Yes”, 0은 “No”라고 **약속**하면 된다.

- 1 : Yes, 0 : No
- 1 : True, 0 : False

만약 사과, 귤, 바나나, 딸기 등 네 종류의 과일이 있고, 이들을 구분하여 컴퓨터에 표기하려면 몇 개의 비트가 필요할까?

2개의 비트가 필요하다. 2개 비트는 4개의 비트 패턴을 만들 수 있고, 각각에 대해 과일을 지정하여 **약속**하면 된다.

- 00 : 사과, 01 : 귤, 10 : 바나나, 11 : 딸기

그렇다면, 영어 문자들을 표현하기 위해 몇 개의 비트가 필요할까?

알파벳 개수는 26개이다. 그런데, 대문자와 소문자를 고려하면 52개의 서로 다른 문자가 존재한다.

1비트 - 2종류
 2비트 - 4종류
 3비트 - 8종류
 4비트 - 16종류
 5비트 - 32종류
 6비트 - 64종류
 7비트 - 128종류
 8비트 - 256종류

따라서, 6비트면 알파벳 대소문자를 모두 표현할 수 있다.

타자기로 편지를 쓰던 것이 컴퓨터로 대체되면서 타자기 자판에 있는 문자들을 모두 비트로 표현할 필요가 생겼다. 알파벳의 대문자, 소문자 외에도 ,!%#^\$&+-_/? 등 여러 문자들을 포함해야 했고, 최종적으로 7비트가 필요했다.

이러저러한 이유로 위의 기본 문자들은 1바이트(Byte), 즉 8비트로 표현하고 있다.

퀴즈) 성별 구분을 위해서는 몇 개의 비트가 필요할까?

6.5 국제표준코드

문자와 비트패턴 간의 매핑은 국제적 약속, 즉 표준으로 정하고 있다.

<http://www.asciitable.com/>

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	##32; Space	64	40	100	##64; @	96	60	140	##96; `			
1	1	001	SOH (start of heading)	33	21	041	##33; !	65	41	101	##65; A	97	61	141	##97; a			
2	2	002	STX (start of text)	34	22	042	##34; "	66	42	102	##66; B	98	62	142	##98; b			
3	3	003	ETX (end of text)	35	23	043	##35; #	67	43	103	##67; C	99	63	143	##99; c			
4	4	004	EOT (end of transmission)	36	24	044	##36; \$	68	44	104	##68; D	100	64	144	##100; d			
5	5	005	ENQ (enquiry)	37	25	045	##37; %	69	45	105	##69; E	101	65	145	##101; e			
6	6	006	ACK (acknowledge)	38	26	046	##38; &	70	46	106	##70; F	102	66	146	##102; f			
7	7	007	BEL (bell)	39	27	047	##39; '	71	47	107	##71; G	103	67	147	##103; g			
8	8	010	BS (backspace)	40	28	050	##40; (	72	48	110	##72; H	104	68	150	##104; h			
9	9	011	TAB (horizontal tab)	41	29	051	##41;)	73	49	111	##73; I	105	69	151	##105; i			
10	A	012	LF (NL line feed, new line)	42	2A	052	##42; *	74	4A	112	##74; J	106	6A	152	##106; j			
11	B	013	VT (vertical tab)	43	2B	053	##43; +	75	4B	113	##75; K	107	6B	153	##107; k			
12	C	014	FF (NP form feed, new page)	44	2C	054	##44; ,	76	4C	114	##76; L	108	6C	154	##108; l			
13	D	015	CR (carriage return)	45	2D	055	##45; -	77	4D	115	##77; M	109	6D	155	##109; m			
14	E	016	SO (shift out)	46	2E	056	##46; .	78	4E	116	##78; N	110	6E	156	##110; n			
15	F	017	SI (shift in)	47	2F	057	##47; /	79	4F	117	##79; O	111	6F	157	##111; o			
16	10	020	DLE (data link escape)	48	30	060	##48; 0	80	50	120	##80; P	112	70	160	##112; p			
17	11	021	DC1 (device control 1)	49	31	061	##49; 1	81	51	121	##81; Q	113	71	161	##113; q			
18	12	022	DC2 (device control 2)	50	32	062	##50; 2	82	52	122	##82; R	114	72	162	##114; r			
19	13	023	DC3 (device control 3)	51	33	063	##51; 3	83	53	123	##83; S	115	73	163	##115; s			
20	14	024	DC4 (device control 4)	52	34	064	##52; 4	84	54	124	##84; T	116	74	164	##116; t			
21	15	025	NAK (negative acknowledge)	53	35	065	##53; 5	85	55	125	##85; U	117	75	165	##117; u			
22	16	026	SYN (synchronous idle)	54	36	066	##54; 6	86	56	126	##86; V	118	76	166	##118; v			
23	17	027	ETB (end of trans. block)	55	37	067	##55; 7	87	57	127	##87; W	119	77	167	##119; w			
24	18	030	CAN (cancel)	56	38	070	##56; 8	88	58	130	##88; X	120	78	170	##120; x			
25	19	031	EM (end of medium)	57	39	071	##57; 9	89	59	131	##89; Y	121	79	171	##121; y			
26	1A	032	SUB (substitute)	58	3A	072	##58; :	90	5A	132	##90; Z	122	7A	172	##122; z			
27	1B	033	ESC (escape)	59	3B	073	##59; ;	91	5B	133	##91; [	123	7B	173	##123; {			
28	1C	034	FS (file separator)	60	3C	074	##60; <	92	5C	134	##92; \	124	7C	174	##124;			
29	1D	035	GS (group separator)	61	3D	075	##61; =	93	5D	135	##93;]	125	7D	175	##125; }			
30	1E	036	RS (record separator)	62	3E	076	##62; >	94	5E	136	##94; ^	126	7E	176	##126; ~			
31	1F	037	US (unit separator)	63	3F	077	##63; ?	95	5F	137	##95; _	127	7F	177	##127; DEL			

Source: www.LookupTables.com

Source: www.LookupTables.com

컴퓨터 초창기에 “ASCII(American Standard Code for Information Interchange) 코드”표를 만들어서 모든 컴퓨터에서 이 약속을 지키도록 했다.

이후 알파벳 외 여러 다른 문자들을 포괄하는 코드표가 만들어지면서 2바이트 이상으로 코드값이 길어졌으나, 이 ASCII코드값 만큼은 호환되도록 하고 있다.

6.6 육십사진수

십육진수보다 더 축약해서 보고 싶을 때는 육십사진수를 쓴다.

XML 문서에 비트스트림을 담고자 할 때 64Base Encoding을 하게 되는데, 비트를 6개씩 쪼개어 하나의 64진수를 만드는 방식이다.

육십사진수를 구성하는 문자 64개는 다음과 같다.

0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz/

퀴즈) 775라는 십진수를 64진수로 변환하고, 이를 다시 이진수로 변환하면?

7. 문서파일 덤프해보기

7.1 비트스트림 인코딩

우리가 작성하는 문서 파일들은 컴퓨터에 어떤 비트스트림으로 저장되는지 확인해 보고자 한다.

먼저, 메모장을 열어 다음의 내용을 입력하고 찾기 쉬운 위치에 sample.txt 로 저장한 후 메모장을 닫는다.

Hello! I'm 24 years old~

메모장에 위의 내용을 입력하고 저장하는 순간, 메모장프로그램이 비트스트림을 만들어 윈도우 운영체계에 전달하면 하드디스크에 비트스트림을 새기게 된다.

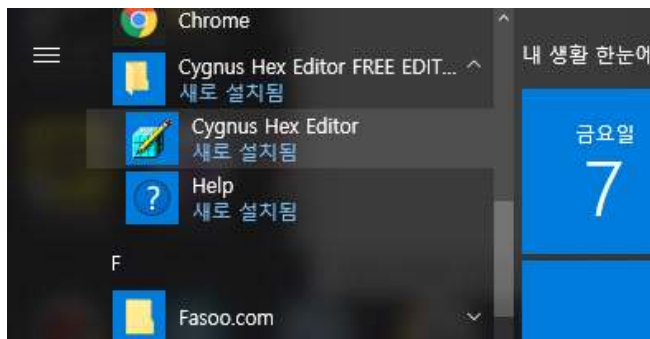
이 때, 메모장이 비트스트림의 패턴을 만들어 내는 과정을 “인코딩(Encoding)”이라 하며, 동일한 내용에 대해서도 문서편집기의 종류마다 인코딩하는 방식이 다를 수 있고, 결과적으로 만들어지는 비트스트림의 패턴이 달라질 수 있다.

sample.txt라는 메모장으로 열어서 볼 때는 비트스트림을 디코딩(Decoding)하여 우리가 알아볼 수 있는 문자열로 변환한 후 결과를 화면에 뿌려주게 된다.

7.2 비트스트림 덤프 소프트웨어 설치

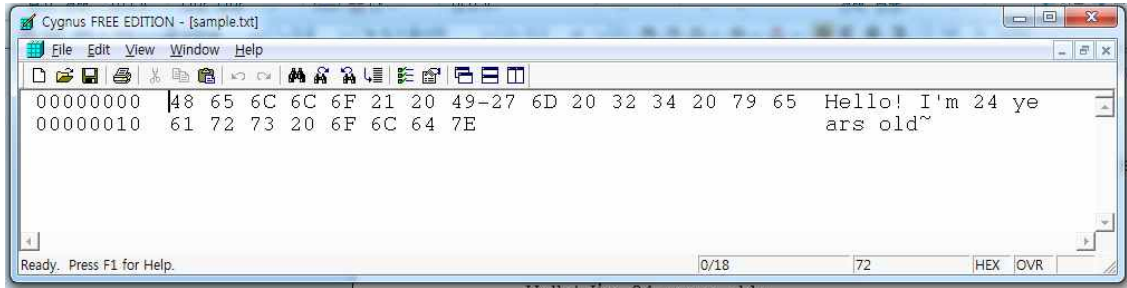
하드디스크에 저장되어 있는 비트스트림을 꺼내 보기 위해 소프트웨어 설치가 필요하다. 아래 페이지 하단의 링크를 눌러 HexEditor 프로그램을 설치해 보자.

<http://www.softcircuits.com/Products/CygnusFE>



7.3 txt 파일 덤프 결과

Cygnus Hex Editor 프로그램을 구동하여 sample.txt 파일을 열어 결과를 확인해 보자.



맨 왼쪽은 줄 번호, 가운데는 비트스트림을 십육진수로 8바이트씩 끊어서 보여주고 있고, 맨 오른쪽은 비트스트림을 ASCII코드값으로 해석한 결과를 보여준다.

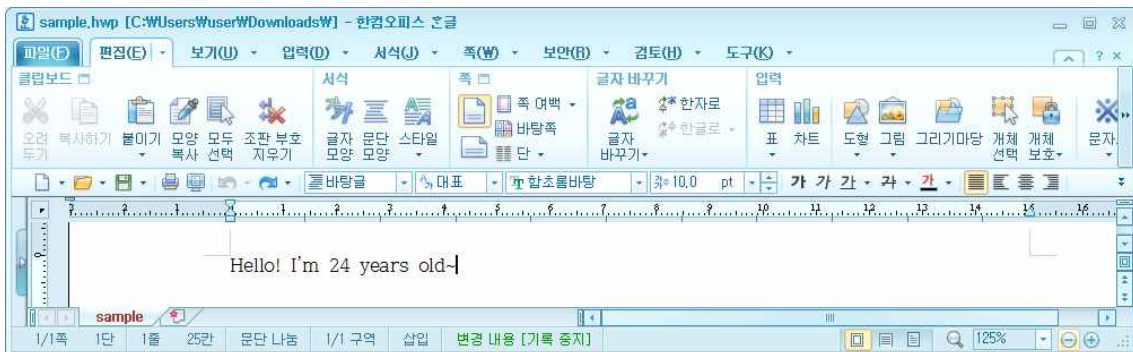
십육진수를 이진수로 변환해보면 다음과 같다.

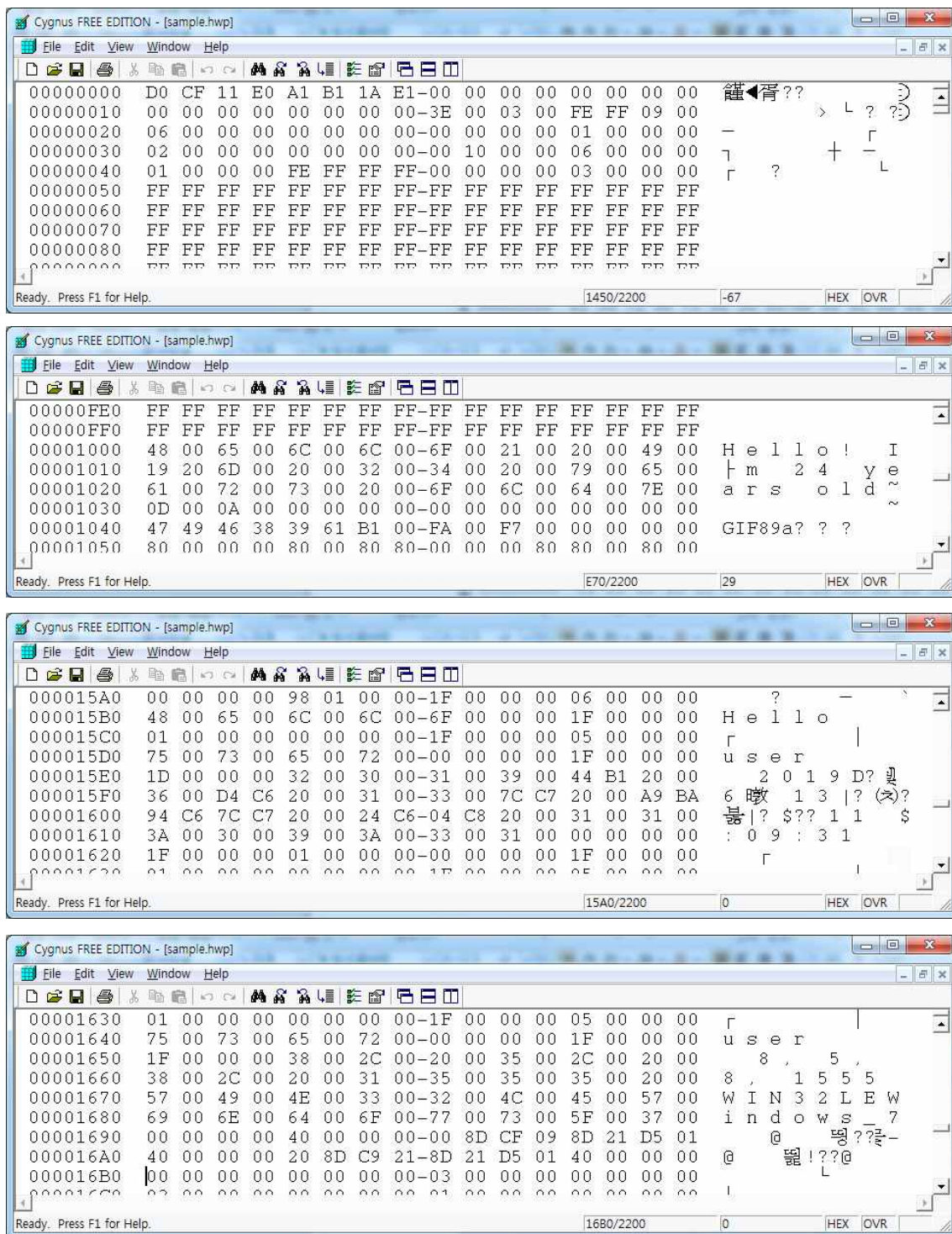
H : 48 => 0100 1000
 e : 65 => 0110 0101
 l : 6C => 0110 1100
 l : 6C => 0110 1100
 o : 6F => 0110 1111
 ! : 21 => 0010 0001
 (space) : 20 => 0010 0000

메모장에 입력된 문자들은 앞에서 살펴본 ASCII코드값 그대로 인코딩되고 있음을 확인할 수 있다.

7.4 hwp 파일 덤프 결과

다음으로는 동일한 내용을 한컴오피스 한글2010으로 작성하여 sample.hwp를 만들고, Cygnus Hex Editor 프로그램으로 열어보자.





무엇인지 모를 비트스트림이 대량으로 인코딩되어 있음을 확인할 수 있다.

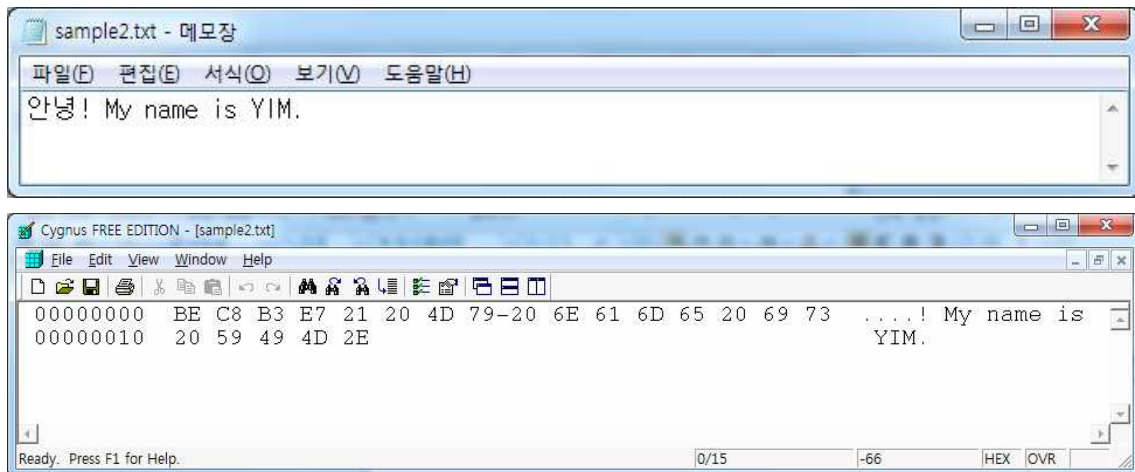
Header 영역을 두어 뭔가 정보를 넣는 영역이 있는 것 같고, 비트스트림이 저장된 시각정보와 컴퓨터정보가 보이기도 하고, 윈도우 운영체제의 정보도 보인다.

직접 입력했던 정보도 중간에 보이는데, '48 00 65 00...'과 같은 방식으로 ASCII 코드값 한

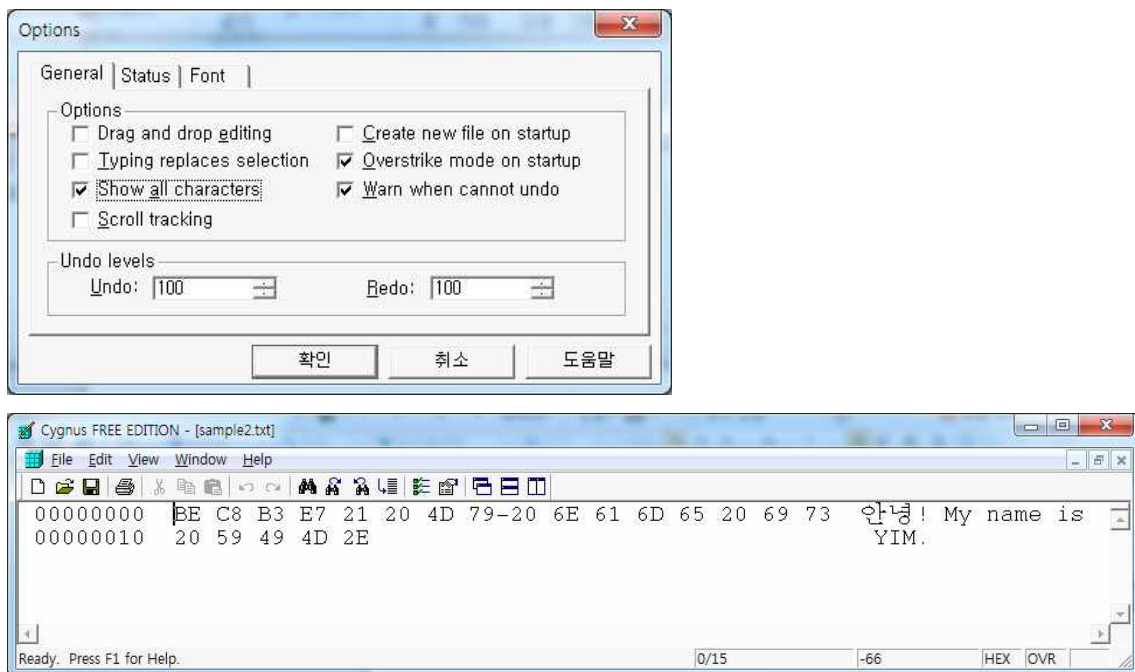
바이트에 00이라는 한 바이트를 덧붙여서 인코딩되었음을 볼 수 있다. ASCII 코드값을 활용하여 영문을 인코딩하되 2바이트로 늘려서 만들고 있음을 확인할 수 있다.

7.5 한글의 인코딩 결과

이번에는 한글이 섞인 내용을 입력하여 비트스트림 인코딩 결과를 확인해 보자.



Cygnus Hex Editor 프로그램의 View -> Options에서 Show all characters를 클릭하면 오른쪽에 표준코드값으로 해석했을 때의 문자들이 표현된다.



이 결과를 보면, 한글인 '안녕'이 'BE C8 B3 E7'로 인코딩되었음을 알 수 있다.

여기서 잠깐, 그렇다면 우리 한글은 컴퓨터에서 어떻게 표현되고 있을까에 대해 잠시 생각해 보자.

컴퓨터 초창기에는 ASCII 코드로 충분했지만, 이후 컴퓨터가 여러 국가에 보급되면서 일본어, 중국어 등 서로 다른 문자들을 포괄하여 표현하는게 필요해졌다. 이에 다국어 문자집합(character set)을 포괄하는 표준적인 코드체계가 만들어지기 시작했다.

컴퓨터가 우리나라에 도입된 초기에 한글의 인코딩 방식을 두고 “조합형 vs 완성형”의 대결국면이 있었다.

조합형이란 한글의 제자 원리에 기반하여 초성, 중성, 종성에 각각 코드를 할당하여 글자대로 코드를 조합하는 방식이다.

완성형이란 '가', '각', '간'과 같은 완성된 문자에 서로 다른 코드값을 할당하는 방식이다.

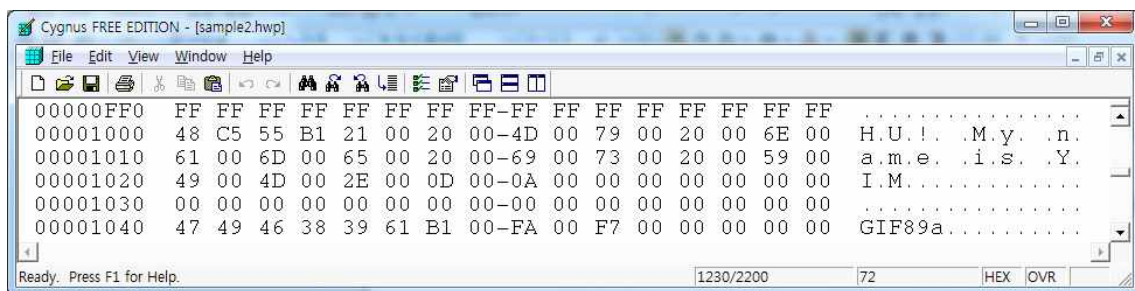
긴 논쟁 끝에 결국 완성형이 한글의 표준 코드체계로 채택되었다. 한 편에서는 완성형이 한글의 다양한 조합을 표현하지 못한다고 비판이 일었고, 한글과 컴퓨터는 독자적인 방식으로 한글을 표현하는 방향을 선택하기도 했다.

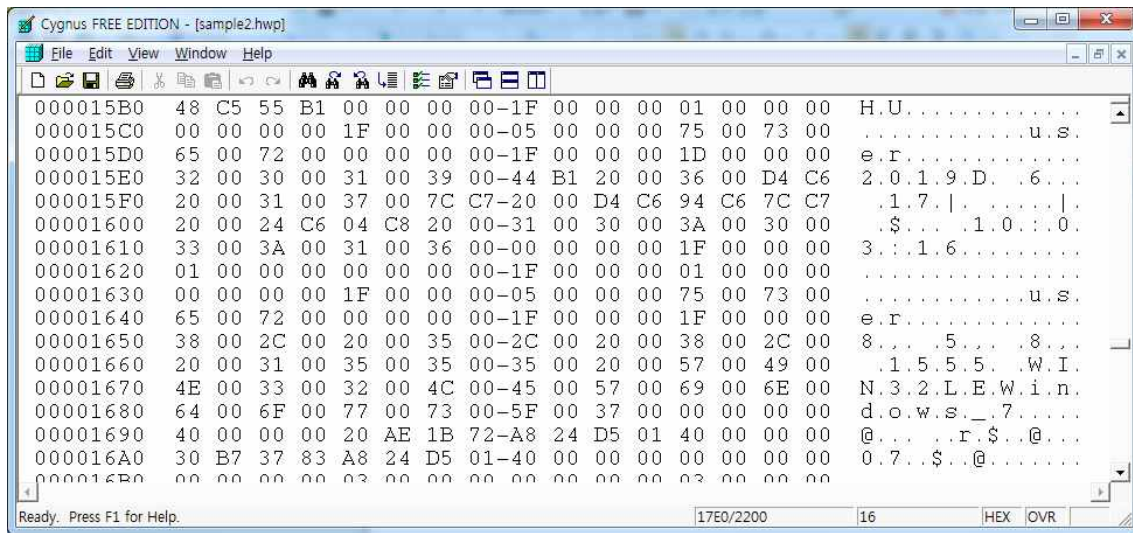
(자세한 역사는 <http://d2.naver.com/helloworld/19187> 참조)

한글 뿐만 아니라, 중국어, 일본어, 유럽의 여러 문자들을 컴퓨터에서 한꺼번에 표현할 수 있도록 하기 위해 만들어 진 것이 유니코드이다. 유니코드는 공식적으로 31비트 문자집합이다.

유니코드 중에서도 UTF-8 인코딩 방식을 가장 많이 사용되고 있다.

동일한 입력 내용으로 hwp파일을 만들어 열어보자.





위의 결과를 보면, 한글인 ‘안녕’이 ‘48 C5 55 B1’로 인코딩된 것으로 유추된다. 메모장에서
의 ‘안녕’과는 다른 코드값이다. 또한, 영문자 하나가 1바이트가 아니라 2바이트로 처리되어
있다.

즉, 자기만의 방식으로 인코딩하고 있음을 알 수 있다.

이런 비트스트림을 읽어서 원래의 메시지를 보여주기 위해서는 작성 시의 편집기나 적합한 뷰
어 프로그램이 필요하다.

따라서, 비트스트림 만으로는 완전한 기록이라 볼 수 없다!!!

한글편집기의 메뉴에서 “파일” -> “문서정보”를 클릭해보면 컴퓨터 이름, 파일 저장 시각, 문
서 제목과 작성자 등 여러 메타데이터들이 보인다. 자동으로 입력되는 값도 있지만, 이 메뉴
를 이용하여 키워드를 입력해둘 수도 있다.

기록관리자는 문서정보와 같이 비트스트림에 내재되는(embedded) 메타데이터의 존재를 알고
있어야 한다.

컴퓨터 이름이 user라서 기본값으로 작성자가 user로 비트스트림에 적히게 되는데, 실제 이
문서의 작성자는 임진희이다. 임진희에 관한 정보가 데이터베이스에 입력되어 있다해도 100년
뒤 만약 메타데이터 없이 비트스트림만 발견된다면 작성자가 user라고 믿게될 것이다.

문서의 저장시각을 컴퓨터 시계가 알려주는 값으로 비트스트림에 새겨넣게 되는데, 이것은 인
코딩된 시점의 정보이고 결재문서의 작성일은 최종 결재일시가 된다. 만약 작성일 메타데이터
가 분실된 상태에서 이 비트스트림만 남겨지게 된다면 작성일시에 대해 혼동할 수 있을 것이
다.

기본으로 컴퓨팅 환경에서 값을 찾아 저장하게 되는 내재된 메타데이터를 비트스트림에서 제
거할 것인지 여부를 판단해야 한다.

8. 독자포맷과 개방포맷

메모장과 한글편집기의 차이는 메모장은 입력된 문자열을 그대로 비트스트림으로 인코딩하는 반면, 한글편집기는 다양한 메타데이터 정보를 함께 비트스트림으로 인코딩한다는 것이다. 결과적으로 비트스트림의 길이가 길어지고, 다양한 정보를 구조화한 영역에 나름의 코드값 대로 저장하므로 한글편집기나 뷰어가 아니면 비트스트림을 열어 그 의미를 확인할 수가 없다. hwp파일을 외국 친구에게 보내면 열어볼 수 있을까?

이처럼, 소프트웨어마다 입력값을 인코딩하는 방식이 다를 수 있으며, 이에 따라 포맷이 결정된다.

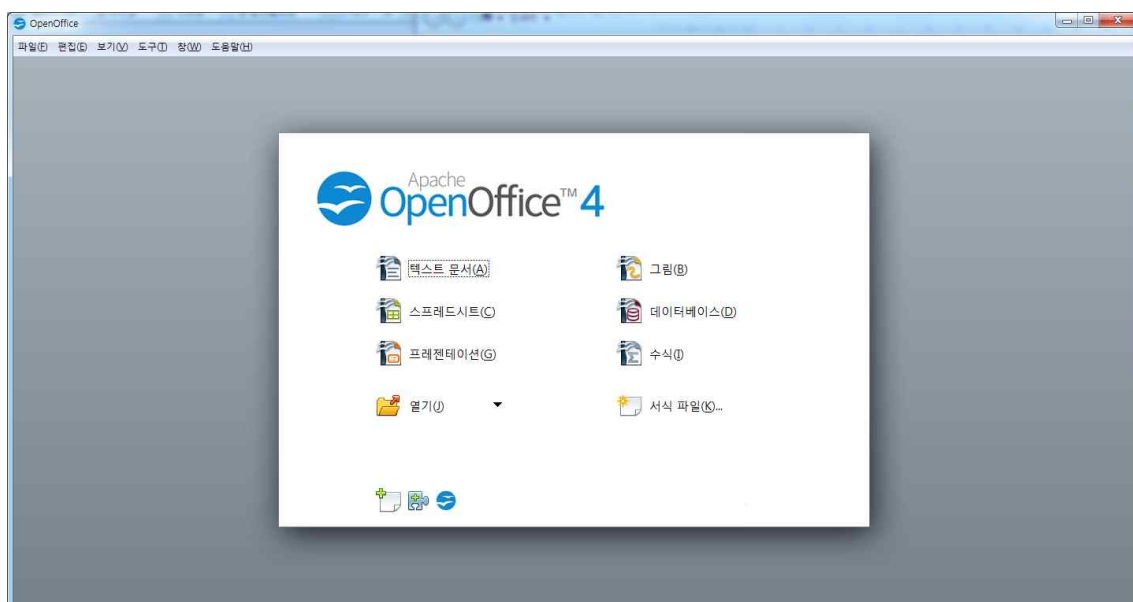
한글 .hwp처럼 자기만의 방식으로 비트스트림을 구조화하고 문자의 비트패턴까지도 자기만의 방식을 채택하는 경우, 이를 디코딩하여 보여주기 어렵다.

이런 포맷을 **독자포맷(Proprietary Format)**이라고 한다.

반면에 ODF(Open Document Format)처럼 인코딩 체계를 공개함으로써 누구든 뷰어를 만들어 비트스트림을 읽고 표현할 수 있도록 하는 포맷을 **개방포맷(Open Format)**이라고 한다. 개방포맷의 경우 대개 이를 편집하고 읽기 위한 소프트웨어도 함께 오픈소스로 제공되고, 쉽게 다운로드 받아 설치하여 사용할 수 있다.

정부가 공문서의 포맷을 독자포맷으로 하게 되면 사용자인 시민들도 해당 포맷을 사용하기 위해 특정 어플리케이션을 구매해야만 한다.

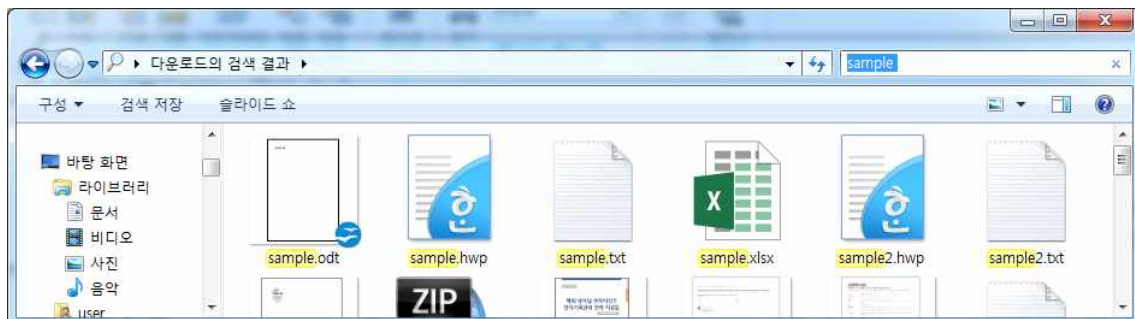
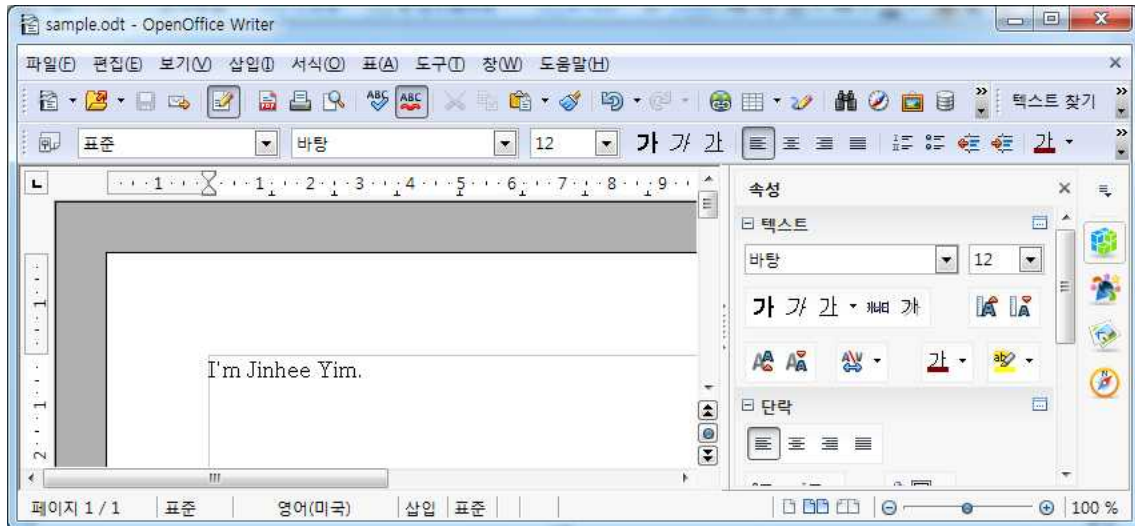
ODF의 경우는 공개소프트웨어가 제공된다. (<http://openoffice.com/>)

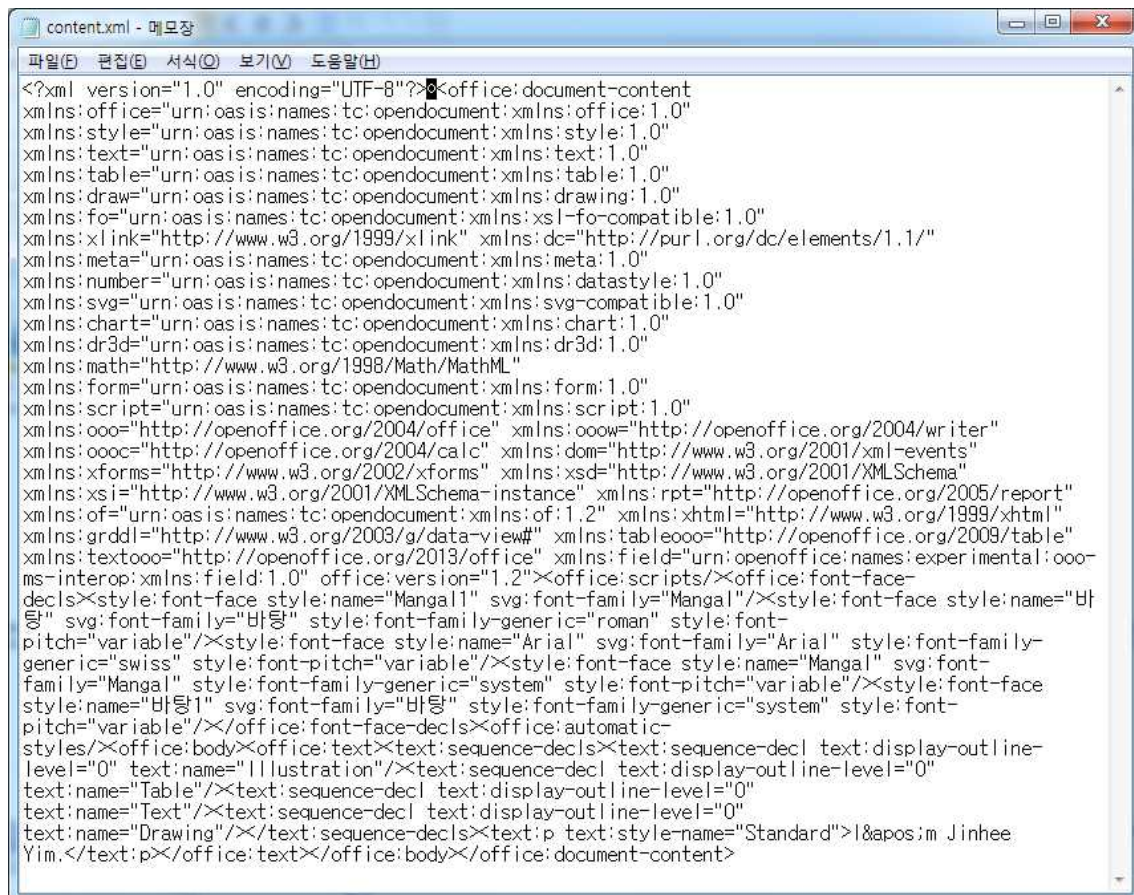
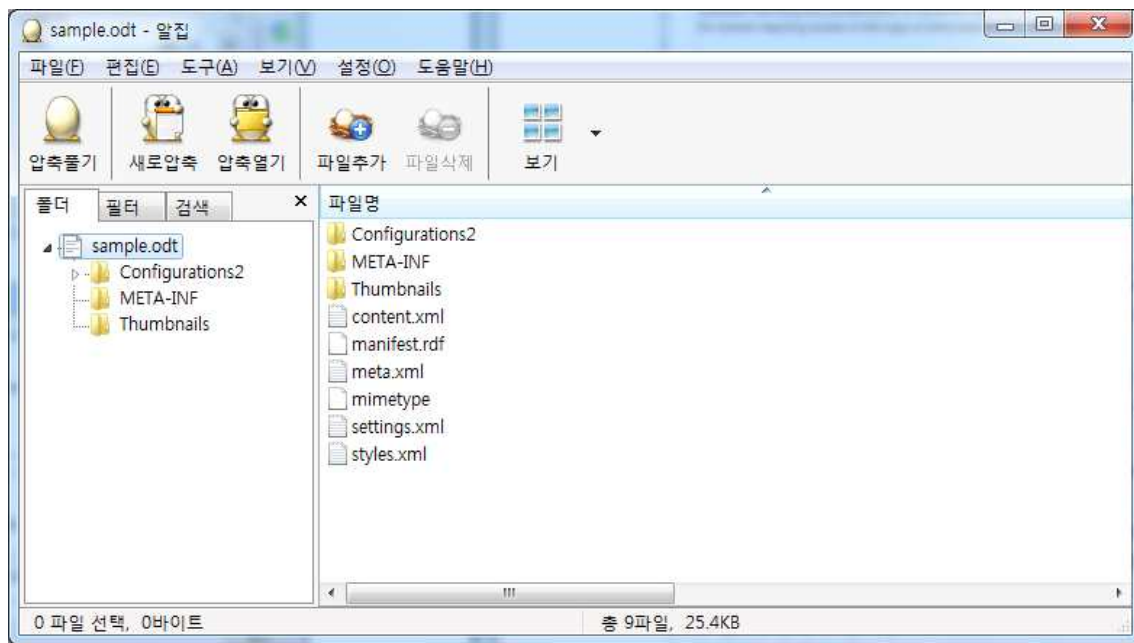


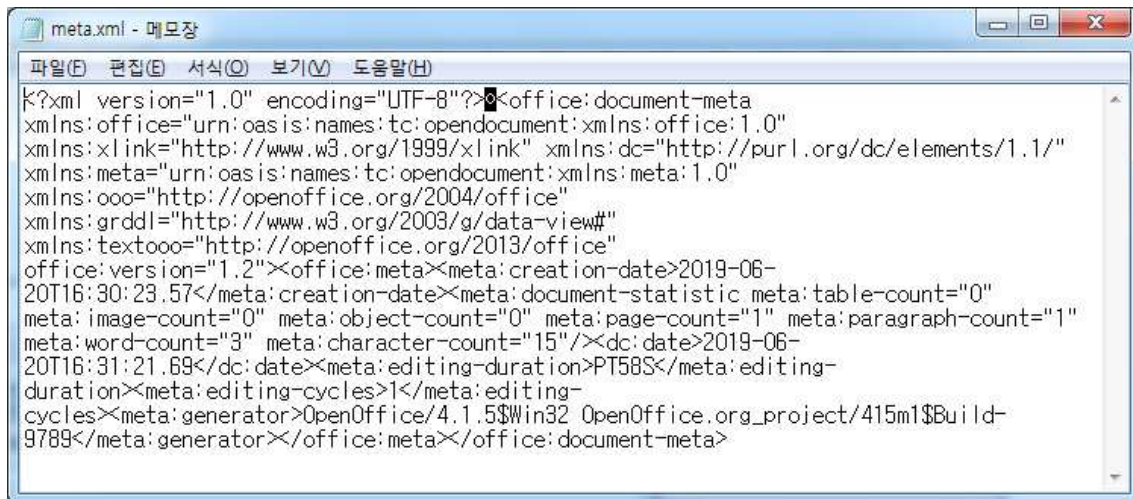
최근 정부가 민원서류를 ODF로 만들기 시작했고, 중앙행정기관의 온나라 문서2.0에서 결재문서의 본문을 ODT로 생산하도록 하고 있다.

정보공개 측면에서도 기록관리 측면에서도 개방포맷은 장점을 갖는다.

다음은 openoffice를 이용하여 작성한 odt 파일의 압축을 풀어 내용을 살펴본 것이다.



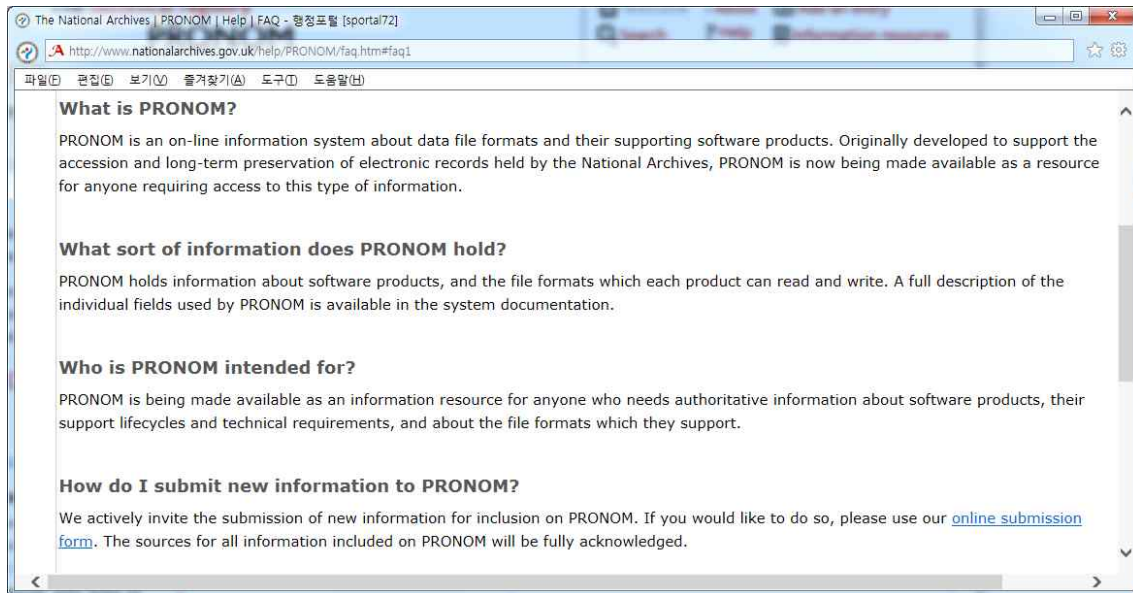




9. DFR과 OAIS의 RI

국가기록원은 DFR(Digital Format Registry)를 구축하는 중이다. 이는 영국 TNA가 운영하는 PRONOM과 유사한 서비스를 지향한다.

(<http://www.nationalarchives.gov.uk/PRONOM/Default.aspx#>)



새로운 파일포맷이 발견되면 DFR에 등록하고, 정체를 알 수 없는 파일이 생겼을 때 DFR 서비스를 이용해서 파일포맷이 무엇인지 알아낼 수 있어야 한다.

전자기록을 장기보존하기 위해 파일 포맷에 관한 정보, 이 파일을 디코딩하기 위한 소프트웨어에 대한 정보를 함께 보관해야 한다.

아키비스트는 컴퓨팅 환경을 변화가 있을 때 DFR 정보를 참조하여 위험에 처하는 파일포맷을 찾아내고, 어떻게 마이그레이션해야 할지 전략을 수립할 수 있어야 한다.

OAIS 참조모형에서 RI(Representation Information)의 중요성을 강조하고 있다. 비트스트림을 해석하여 원래의 메시지로 표현해주고, 정보의 구조와 의미를 알려주기 위해 필요한 각종 정보들을 RI로 범주화하고 있다.

(<https://public.ccsds.org/Pubs/650x0m2.pdf>)

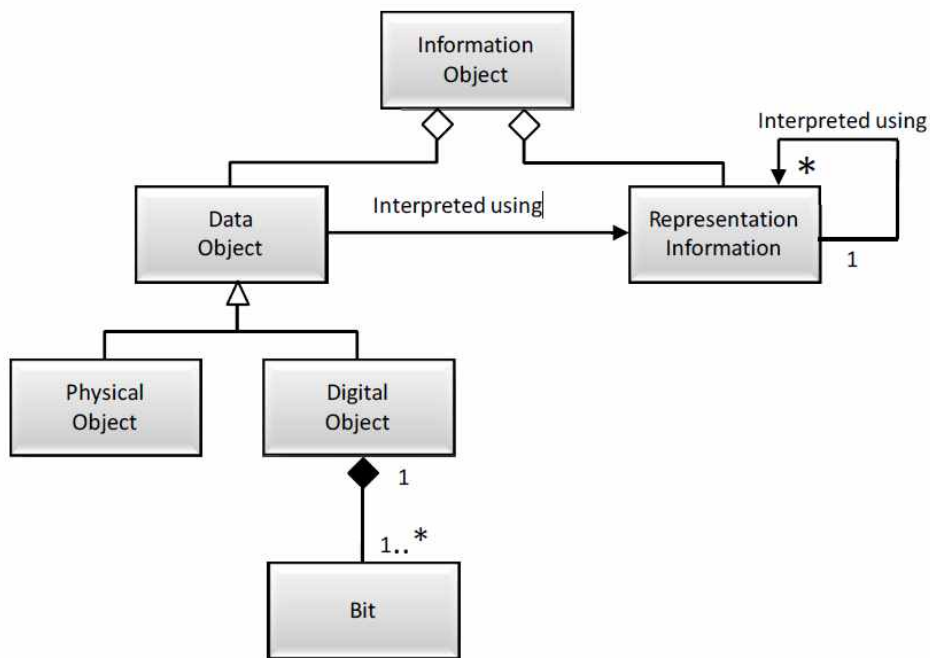


Figure 4-10: Information Object