발간등록번호
11-1312125-000014-01

연구결과보고서

# 행정기관 전자기록물 재현기술 연구 및 프로토타입 개발

# A study on the reproduction technology and the prototype for the electronic records of administrative agency

주관연구기관 : 명지대학교 산학협력단

국가기록원

# 용역연구사업 연구결과보고서

| | | | | |
|---|---|---|---|---|
| 관리 번호 | | | | |
| 사 업 명 | 2013년 기록보존기술 연구개발사업 | | | |
| 과 제 명 | 국문 | 행정기관 전자기록물 재현기술 연구 및 프로토타입 개발 | | |
| | 영문 | A study on the reproduction technology and the prototype for the electronic records of administrative ag | | |
| 주관연구기관 | 기 관 명 | | 소재지 | 대 표 |
| | 명지대학교 산학협력단 | | 경기도 용인시 처인구 명지로 116 | 이 종 명 |
| 주관연구 책임자 | 성 명 | | 소속 및 부서 | 전 공 |
| | 임 진 희 | | 명지대학교 기록정보과학전문대학원 | 컴퓨터공학 |
| 총연구기간 | 2013 년 07 월 12 일 - 2013 년 11 월 30 일( 4.59 개월) | | | |
| 총 연구비 | 일금 구천칠백이십칠만육천원 정 (₩97,276,000) | | | |
| 연구년차 | 연구기간 | | 연구비 | |
| 1차년도 | 2013. 07 . 12 - 2013. 11 . 30 | | 97,276천원 | |
| 총참여연구원 | 10명 (책임연구원: 1 명, 연구원: 7 명, 연구보조원: 2 명 보조원: 명) | | | |

2013년도 기록보존기술 연구개발사업에 의하여 수행중인 연구과제의 연구결과보고서를 붙임과 같이 제출합니다.

붙임 . 연구결과보고서 35부(별첨).

2013년 11월 20일

주관연구책임자 임 진 희 (인)

주관연구기관장 이 종 명 (직인)

국가기록원장 귀하

# 목 차


**Ⅰ. 연구개발결과 요약문**

(한글) ····· 1

(영문) ····· 2

**Ⅱ. 총괄연구개발과제 연구결과**

제1장 총괄연구개발과제의 최종 연구개발 목표 ····· 4

1.1 총괄연구개발과제의 배경 및 목표 ····· 4

가. 연구배경 ····· 4

나. 연구목표 ····· 9

1.2 총괄연구개발과제의 목표달성도 ····· 10

제2장 총괄연구개발과제의 최종 연구개발 내용 및 방법 ····· 13

2.1 연구내용 ····· 13

2.2 연구방법 ····· 14

2.2.1 T1 에뮬레이션 연구 사례분석 ····· 15

가. T1.1 에뮬레이션 연구 사례분석 ····· 15

2.2.2 T2 디지털 컴포넌트 유형별 보존 대상 연구 ····· 16

가. T2.1 행정기관의 전자기록물 포맷조사 ····· 17

나. T2.2 디지털 컴포넌트 보존 속성 연구 ····· 17

다. T2.3 전자문서 이외의 디지털 컴포넌트 유형 보존 속성 연구 ····· 17

2.2.3 T3 디지털 컴포넌트의 렌더링 객체 유형 연구 ····· 18

가. T3.1 보존포맷(PDF/A) 변환 시 소실되는 렌더링 객체 조사 ····· 18

나. T3.2 디지털 컴포넌트에 포함된 렌더링 객체 기능성 분석 ····· 21

다. T3.3 보존 대상 디지털 컴포넌트의 기능성 보존 방안 연구 ····· 21

2.2.4 T4 에뮬레이션 제안 및 프로토타입 구현 ····· 23

가. T4.1 에뮬레이션 추진 전략 및 개발 방안 ····· 23

나. T4.2 에뮬레이터 프로토타입 설계 및 구현 ····· 24

제3장 총괄연구개발과제의 최종 연구개발 결과 ····· 26

3.1 에뮬레이션 연구 사례분석 ····· 26

3.1.1 사례분석 개요 ····· 26

가. 용어정의 ····· 26

나. 장기보존 전략으로서 에뮬레이션의 특징 ····· 27

다. 사례분석의 목적과 필요성 ····· 32

3.1.2 CAMILEON 프로젝트 ····· 34

가. 개요 및 목적 ····· 34

나. CAMILEON의 보존전략 ····· 35
다. 지적 재산권 이슈 및 비용효과 ····· 40
라. 적용사례 ····· 40
3.1.3 Koninklike Bibliotheek(KB) 프로젝트 ····· 41
가. 개요 ····· 41
나. 에뮬레이션의 4가지 전략 ····· 42
다. 개념적 모델: 모듈 에뮬레이션(Modular Emulation) ····· 48
3.1.4 Planets 프로젝트 ····· 57
가. 개요 ····· 57
나. 주요 서비스 ····· 63
다. 적용 사례 ····· 72
라. 시사점 ····· 77
3.1.5 Keep 프로젝트 ····· 78
가. 개요 ····· 78
나. 주요 서비스 ····· 82
다. 메타데이터와 법적 문제 ····· 89
라. 시사점 ····· 93
3.1.6 사례 분석결과 및 한국형 에뮬레이션 전략 ····· 94
가. 사례 분석결과 종합 ····· 94
나. 에뮬레이션 전략의 법/비용 이슈 ····· 98
다. 한국형 에뮬레이션 전략 제안 ····· 113
3.2 행정기관의 디지털 컴포넌트 유형별 보존 대상 연구 ····· 119
3.2.1 방송통신위원회 디지털 컴포넌트 현황 조사 ····· 119
가. 조사 개요 ····· 119
나. 조사결과 분석 및 시사점 ····· 120
3.2.2 디지털 컴포넌트 보존 속성 연구 ····· 132
가. 에뮬레이션 대상 기능성 정의 ····· 133
나. 에뮬레이션을 위한 보존 평가 방법 ····· 136
3.2.3 전자문서 이외의 디지털 컴포넌트 유형 보존 속성연구 ····· 142
가. 배경 및 필요성 ····· 143
나. 주요 유형별 에뮬레이션 고려사항 ····· 147
3.3 디지털 컴포넌트의 렌더링 객체 유형 연구 ····· 149
3.3.1 장기보존 대상 디지털 컴포넌트의 포맷 변화 분석 ····· 149
가. 포맷변화 분석 변화 배경 및 필요성 ····· 149
나. 한/글 애플리케이션 변화 분석 ····· 150
다. 마이크로소프트의 엑셀 및 오피스 애플리케이션 변화 조사 ····· 159

라. 디지털 컴포넌트 실행을 위한 운영 체제 변화 조사 ····· 162
마. 포맷 변화 분석 결과 ····· 169
3.3.2 보존포맷(PDF/A) 변환시 소실되는 렌더링 객체 기능성성분석 ····· 173
가. 기능성 분석 배경 및 필요성 ····· 173
나. 한/글 포맷에서 소실되는 기능성 조사 ····· 173
다. 파워포인트 포맷에서 소실되는 기능성 조사 ····· 184
라. 엑셀 포맷에서 소실되는 기능성 조사 ····· 191
마. 소실되는 기능성 분석 결과 ····· 199
3.3.3 소실되는 기능성의 속성 분석 ····· 200
가. 기능성의 속성 분석 배경 및 필요성 ····· 200
나. 한/글 포맷에서 소실되는 기능성의 속성 구조 ····· 201
다. 파워포인트 포맷에서 소실되는 기능성의 속성 구조 ····· 209
라. 엑셀 포맷에서 소실되는 기능성의 속성 구조 ····· 218
마. 렌더링 기능성에 대한 비트스트림 분석 ····· 229
바. 기능성의 속성 분석 결과 및 시사점 ····· 234
3.3.4 디지털 컴포넌트의 기능성 보존 방안 연구 ····· 236
가. 기능성 보존 연구 개요 ····· 236
나. 국가기록원 기술정보은행(DFR) 구조 분석 ····· 238
다. 영국 국가기록원의 PRONOM & DROID 구조 분석 ····· 253
라. 기술정보은행(DFR)과 PRONOM의 포맷 기술 정보 비교 분석 ····· 278
마. 기능성 보존을 위한 XML Schema 개선안 ····· 280
3.4 에뮬레이션 제안 및 프로토타입 구현 ····· 283
3.4.1 에뮬레이션 방안 연구 ····· 283
가. 에뮬레이션 대상 연구 ····· 284
나. 에뮬레이션 방안 비교 ····· 288
다. 국가기록원의 한국형 에뮬레이션 대상 연구 ····· 296
라. 국가기록원의 한국형 에뮬레이션 방안 ····· 298
마. 한국형 에뮬레이션 추진 전략 제안 ····· 303
3.4.2 에뮬레이터 프로토타입 개발 ····· 309
가. 해외 에뮬레이터 구동 사례 ····· 309
나. 한국형 에뮬레이터 프로토타입 개발 ····· 315
제4장 총괄연구개발과제의 연구결과 고찰 및 결론 ····· 325
제5장 총괄연구개발과제의 연구 성과 ····· 328
제6장 기타 중요변경사항 ····· 330
제7장 참고문헌 ····· 331
제8장 첨부서류 ····· 334

# Ⅰ. 연구결과보고서 요약문

| 연구과제명 | 행정기관 전자기록물 재현기술 연구 및 프로토타입 개발 | | |
|---|---|---|---|
| 중심단어 | 전자기록 장기보존 전략, 에뮬레이션 프레임워크, 디지털 컴포넌트 포맷, DFR, 에뮬레이터 프로토타입 | | |
| 주관연구기관 | 명지대학교 산학협력단 | 주관연구책임자 | 임진희 |
| 연구기간 | 2013년 7월 12일 - 2013년 11월 30일 | | |

**❒ 최종연구목표**

○ 전자기록물 에뮬레이션 국내외 기술 동향 분석 제공

○ 디지털 컴포넌트 유형별 보존 대상 정의 및 중요 속성(property) 분석

○ 디지털 컴포넌트 유형별 렌더링 기술 기반 마련 및 객체 속성(property) 분석 방법론 개발

○ 에뮬레이터 프로토타입 개발

**❒ 연구내용 및 방법**

○ 전자기록물 에뮬레이션 해외 연구 동향 분석

- CAMiLEON, KB 에뮬레이션, Planet, KEEP 등의 사례 분석을 통한 연구 동향 분석 및 에뮬레이터 개발의 시사점 제공
- 에뮬레이터 구현 시 기술비용, 소유권 등 관련문제 해결방안 제시

○ 행정기관의 디지털 컴포넌트 유형별 보존 대상 연구

- 행정기관의 전자기록 디지털 컴포넌트별 버전 조사
- 디지털 컴포넌트 객체별 중요 속성(property) 정의 방법 및 절차 마련
- 전자문서 기반의 전자기록물 이외의 유형에 대한 에뮬레이션 전략 제안

○ 디지털 컴포넌트 렌더링 객체 유형 연구

- 보존문서(PDF/A) 로 변환시 소실되는 렌더링 객체 조사
- 디지털 컴포넌트에 포함된 렌더링 객체 속성 분석
- 보존 대상 디지털 컴포넌트의 렌더링 객체 보존 방안 연구

○ 에뮬레이터 프로토타입 설계

- 디지털 컴포넌트 유형별 에뮬레이터 프로토타입 설계 및 구현

**❒ 기대성과 및 활용방안**

○ 디지털 컴포넌트 유형별 렌더링 방법 마련 및 에뮬레이션 전략 적용 타당성 검증을 위한 에뮬레이터 프로토타입 개발

○ 본 디지털(born digital) 기록물을 원본 기반의 장기보존 및 활용성 고취

# Summary

<table>
<tr><td>Title of Project</td><td colspan="3">A study on the representation technology for digital records of administrative agencies and the prototype of emulator</td></tr>
<tr><td>Key Words</td><td colspan="3">Electronic records Long-term Preservation Strategies, Emulation Framework, Digital Component Format, DFR(Digital Format Registry), Emulator Prototype</td></tr>
<tr><td>Institute</td><td>Myungji Univ. Digital Archiving lab.</td><td>Project Leader</td><td>Yim, Jin-Hee</td></tr>
<tr><td>Project Period</td><td colspan="3">Jul, 12. 2013 - Nov, 30. 2013</td></tr>
<tr><td colspan="4">❒ Research Objects<br>○ Trend analysis of the emulation technology for digital records<br>○ Preservation target identification and important property analysis<br>○ Provide the technological base of rendering and develop the object property analysis methodology<br>○ Development of a prototype for the emulation</td></tr>
<tr><td colspan="4">❒ Research topics & Methodology<br>○ Research trend analysis of the emulation technology for digital records<br>• Case study : CAMiLEON, KB emulation, planet, KEEP<br>• Implications deduction of the development of emulator and the analysis of research trends<br>• Suggest a solution for technological cost and ownership of emulator<br>○ Preservation target identification of digital component<br>• Research version of digital components within electronic records<br>• Establish appraisal procedures for to define significant property of digital component<br>• Design of emulation strategy on electronic records not based on electronic documents<br>○ Analysis of rendering object type for digital component<br>• Look into rendering object become lost when converted into PDF/A<br>• Analysis of property for rendering object contained in the digital components<br>• Provide the methodology for preservation of target digital component<br>○ Design of a prototype for the emulation<br>• Design for a prototype of emulator according to the digital component type</td></tr>
</table>

❒ Expected Outcomes

- ○ Establish rendering type to according digital component and development of prototype for the emulation that validation of apply emulation strategy
- ○ Encourage usability and Long-term preservation of source-based records for born digital records

# Ⅱ. 총괄연구개발과제 연구결과

## 제1장 총괄연구개발과제의 최종 연구개발 목표

### 1.1 총괄연구개발과제의 배경 및 목표

#### 가. 연구배경

❒ 다양한 유형의 전자기록물에 대한 장기보존 전략 수립 필요
- ○ 전자기록물의 양적 증가와 다양화는 가속화되고 있으나, 전자기록물의 보존 전략 및 기술 연구는 미흡함
- ○ 전자기록물의 안정적이고 지속적인 보존을 위해 기록물 보존 전략을 수립하고 기술적 측면의 연구가 요구됨
- ○ 특히 국내의 전자기록관리 환경에 적합한 전자기록의 장기적 보존을 위한 기술적 전략에 관해 적극적으로 모색해야 함
- ○ 이를 위해 전자기록의 유형에 관한 분석과 보존을 위한 객체의 중요 속성(property) 표현에 관한 연구가 필요

❒ 문서 저장 포맷 및 속성의 변화
- ○ 기술의 발달과 함께 문서 저장 포맷 및 유형이 지속적으로 변화되고 있음
- ○ 문서를 생성하는 애플리케이션 중에서 더 이상 사용하지 않는 애플리케이션이 다수 존재함. 보석글, 훈민정음, 하나워드부터 CAD Application, MC, MD, Lotus123 등 스프레드시트 유형의 프로그램 들 그리고, 웹에디터 등 다양한 애플리케이션 프로그램들이 사용되다가 지금은 사용이 중단된 상태임
- ○ 현재 가장 많이 사용하는 한/글, 엑셀, 파워포인트 등의 애플리케이션도 다양한 버전이 존재하며, 버전 간 파일 포맷 및 속성의 호환에 문제가 발생되고 있음

❒ 컴퓨팅 환경의 변화로 O/S 및 애플리케이션의 변화
- ○ 전자기록물 원본 형식을 지원하는 애플리케이션을 보관하고 있지만, 운영환경이 변화하거나 컴퓨팅 환경이 변화할 경우 보관된 애플리케이션을 더 이상 수행할 수 없는 상황이 발생함
- ○ 현재까지 개발된 거의 대부분의 애플리케이션은 MS의 윈도우 운영환경에서 동

작하도록 개발되어 있으나, 운영 환경이 클라우드 서비스 환경으로 바뀔 경우 서버 환경에서 애플리케이션의 구동이 이루어져야 하지만 이미 사양화된 애플리케이션을 클라우드 환경에서 동작하도록 변환하는 기술적, 경제적 측면에서 어려운 경우에는 보관한 애플리케이션의 동작이 불가능하게 되어 원문을 인식하지 못하는 상태가 발생할 것으로 예측됨

❒ 애플리케이션의 보관과 함께 생성되는 파일 형식에 대한 보존이 필요

○ 현 시점에서 가장 중요한 것은 사라져 가는 애플리케이션의 파일 형식을 정확하게 파악하여 레지스트리를 구축하는 것이 필요

○ 이를 위해 국가기록원에서는 DFR(Digital Format Registry)를 구축하였으나, 현재의 DFR은 파일포맷 정보와 소프트웨어 정보만을 담고 있음

○ 스프레드시트나 워드프로세서, 파워포인트 등 각종 애플리케이션 파일에서 제공하는 기능성에 대한 표현 방법 연구와 이를 재현하기 위한 방법을 고려해야 함

❒ 문서보존포맷으로 변환된 디지털 컴포넌트의 재현 기술 한계

○ PDF는 시스템 간 이동성이 좋고 배포용으로 광범위하게 사용되고 있으며, 특히 PDF/A는 영구보존 문서용으로 채택되고 있음

- ISO 19005 (Electronic document file format for long-term preservation : PDF/A) 장기보존용 전자문서 파일포맷 PDF/A는 3가지 종류의 문서보존 포맷을 제시함
- 우리나라는 공공기록의 문서보존포맷과 전자문서류 컴포넌트 보존포맷으로 PDF/A-1을 채택하고 있음

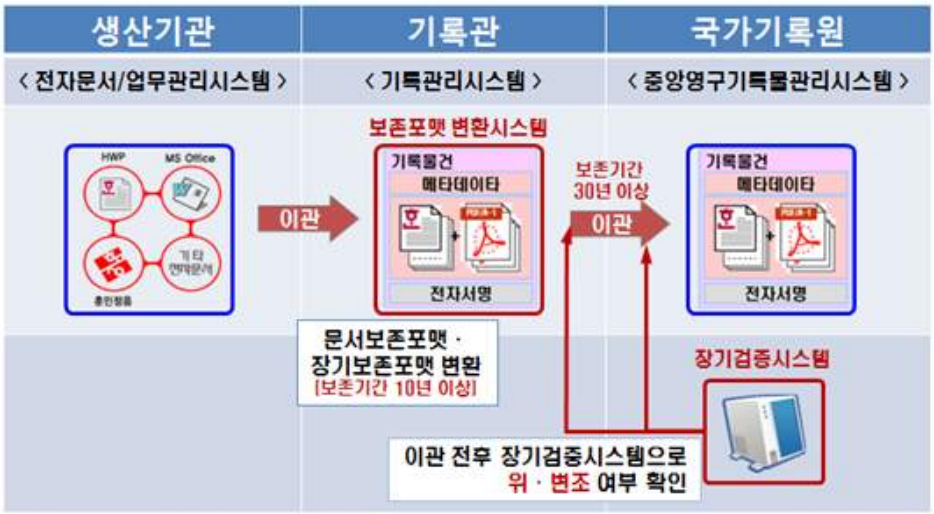
<표 1-1> ISO 19005의 구성

출처 : http://en.wikipedia.org/wiki/PDF/A

| Part | Name | Formal name | Release date | Standard | Based on PDF version |
|---|---|---|---|---|---|
| Part 1 | PDF/A-1 | Use of PDF 1.4 (PDF/A-1) | 2005 | ISO 19005-1 | PDF 1.4(Adobe Systems, PDF Reference third edition, 2001) |
| Part 2 | PDF/A-2 | Use of ISO 32000-1 (PDF/A-2) | 2011 | ISO 19005-2 | PDF 1.7(ISO 32000-1:2008) |
| Part 3 | PDF/A-3 | Use of 32000-1 with support for embedded files (PDF/A-3) | 2012 | ISO 19005-3 | PDF 1.7(ISO 32000-1:2008) |

○ 현재 국가기록원 장기보존 전략은 마이그레이션/인캡슐레이션 전략이라고 볼 수 있음(<그림 1-1> 참조)

- 문서보존포맷은 문서의 내용정보를 보존하기 위한 포맷으로 생성당시의 문서 편집기의 버전 및 소멸에 영향없이 내용보기가 가능한 국제표준규격을 준수한 포맷. 단, 정적인 형태의 인쇄 방식의 렌더링만 가능




<그림 1-1> 국가기록원의 장기보존 전략

- 한편, 장기보존포맷은 전자기록물을 장기적으로 보존 유지하기 위한 포맷으로 기록을 원문, PDF/A-1, 메타데이터 및 전자서명을 하나의 파일로 만든 포맷임(<그림 1-2> 참조)




<그림 1-2> 국가기록원의 장기보존 포맷의 생성과정

○ 그러나 PDF/A-1, PDF/A-2, PDF/A-3 모두 프린트 상태의 정적인 형태로 문서의 외형을 재현하는데 그침

<표 1-2> PDF/A 표현 가능 요소

| 보존포맷 | PDF/A-1 | PDF/A-2 | PDF/A-3 |
|---|---|---|---|
| 표현요소<br>(추가요소) | 텍스트<br>JPEG2000<br>Postscript<br>Unicode | 압축된 JPEG2000<br>투명 효과, 레이어<br>코멘트, 주석<br>고급 디지털 서명<br>여러 개의 PDF/A 문서<br>삽입 가능 | XML, CSV, CAD,<br>문서편집기 문서,<br>스프레드시트 문서 등<br>삽입 가능 |

○ PDF/A-2 이후 여러 개의 문서를 삽입하는 기능이 추가되었으나, 삽입된 문서의 경우 장기적 보존보다는 단기적, 임시적 활용을 위한 방편으로 인식되어야 한다는 의견이 제기되고 있음[1)]

○ PDF/A에서는 오디오, 비디오, 자바스크립트 등 동적인 요소들의 사용이 제한되며, 기능적 요소가 포함된 한/글, 엑셀, 파워포인트 등의 문서를 PDF/A로 변환할 경우 외형만 재현하고 본문에 첨부된 동영상이나 움직임 효과 등이 렌더링되지 않아 기록물 생산자의 의도나 내용의 맥락을 완벽하게 보존하지 못하는 한계가 발생함

❒ 디지털 컴포넌트 기반의 전자기록물에 대한 재현 기술 연구 필요

○ 장기보존 전자기록물에 포함된 디지털 컴포넌트에 대한 조사 연구를 통해 마이그레이션 방식과 에뮬레이션 방식에 대한 장단점을 분석한 후 장기 보존된 디지털 컴포넌트 기반의 전자기록물 재현을 위한 기술 개발 전략 수립 필요함

❒ 에뮬레이션 전략을 중심으로 한 전자기록 장기보존 방법론 모색

○ 기존의 마이그레이션 및 XML 패키지 변환 등의 방법으로 기능성 요소가 포함된 문서를 PDF/A로 변환할 경우 원본 생산자의 의도나 내용적 맥락이 재현되지 않는 문제점이 발생함

○ 현재 전자기록의 디지털 컴포넌트 대부분은 정적이며 고정된 형태의 화면 출력을 보여주는 것에 그치고 있어, 이러한 한계를 극복하기 위해 기능성들을 렌더링 하는 기술이 필요

○ 또한 기술적 문제로 마이그레이션이 실패할 경우 또한 대비할 필요성이 제기됨. 전자기록물을 XML 형태로 Base64 인코딩하여 패키징하는 경우에도 컴포넌트의 보존포맷은 PDF/A이므로 역시 정적인 형태의 재현에 머무르게 됨.

1) http://www.digitalpreservation.gov/formats/fdd/fdd000360.shtml

마이그레이션이 실패하거나 문서에 포함된 기능적 요소를 재현하지 못할 경우에 대비하여 에뮬레이션 전략을 병행하는 것이 필요함

- ○ PPT 슬라이드, 플래시, CAD/CAM 등의 기능적 요소가 포함된 컴포넌트가 생산당시의 H/W 및 S/W가 없는 상태에서 비트스트림만 보존되었을 때 해당 기록물이 생산된 본래 의도나 맥락을 재현하는 에뮬레이션 전략이 필요함
- ○ 전자기록의 유형별 컴포넌트의 중요 속성(property) 표현에 관한 기술 정보 평가 방법론을 정립하여 보존 대상 컴포넌트의 속성별 렌더링 방법에 대한 모색이 요구됨
- ○ 우리나라의 전자기록 환경에서 생산되는 보존 대상의 유형과 그에 적합한 에뮬레이터 프로토타입 개발이 필요함
- ○ 이러한 속성 정보를 DB 구축을 통해 DFR(Digital Format Registry)에 반영하여 정책개발 및 기술 기반으로 활용하도록 함

❒ 에뮬레이션 연구에 대한 관련 분야의 요구 증대

- ○ 대표적인 해외의 에뮬레이션 프로젝트인 CAMiLEON, KB 에뮬레이션, Planets, KEEP 등은 전자문서 류 뿐만 아니라 다양한 OS와 미디어, 응용 프로그램, 게임, 웹페이지 등을 재현 대상으로 선정하여 에뮬레이터를 개발하고 있음
- ○ EU의 KEEP 프로젝트는 디지털 객체의 장기보존에 대한 대표적 표준인 ISO 14721(OAIS)의 DIP단계에서 렌더링하여 사용자에게 에뮬레이션 결과를 보여주는 개념을 제시함
- ○ DIP를 에뮬레이션 하는 방식은 이미 패키징 된 원본 AIP를 교체하지 않아 디스크 보존 용량을 변화시키지 않으며, 다른 장기보존 솔루션에 영향을 주지 않는 장점이 있음
- ○ 국내 공공기관에서 생산 및 접수하고 있는 기록물의 형태 중 기능적 요소를 포함하고 있는 장기보존 대상을 평가하고 기술적 구현이 가능한 수준에서 에뮬레이터를 개발하는 것이 바람직함
- ○ 컴퓨팅 기술이 발전하면서 기능적 요소에 대한 기술 개발이 가속화하고 있음. 현재는 시각과 청각에 의존하는 기능적 요소에서 후각, 미각, 촉각에 대한 기능적 요소를 표현하는 기술 개발이 진행되고 있음
- ○ 디지털 컴포넌트를 생성하는 애플리케이션의 기능이 보다 다양하고 복잡한 기능적 렌더링 객체 요소를 포함하게 될 것으로 예상되므로, 이러한 기능적 요소를 포함한 전자문서의 영구보존을 위해 필요한 에뮬레이션에 대한 깊이 있는 연구가 필요한 시점임

## 나. 연구 목표

❒ 전자기록물 에뮬레이션 국내외 연구 동향 분석 제공

○ CAMiLEON, KB 에뮬레이션, Planets, KEEP 등의 사례 분석을 통한 연구 동향 분석 및 에뮬레이터 개발의 시사점 제공

❒ 행정기관의 전자기록물 유형별 보존 대상 연구

○ 행정기관의 전자기록 디지털 컴포넌트 포맷 조사

○ 디지털 컴포넌트 객체별 보존해야 할 기능성 정의 및 평가 방법론 개발

○ 딥 웹(deep web), 데이터세트(data set), SNS(Social Network Service)등 개체에 대한 기록화 방안 연구 및 에뮬레이션 전략 제안

❒ 디지털 컴포넌트 렌더링 객체 유형 연구

○ 보존문서(PDF/A) 로 변환시 소실되는 렌더링 객체 조사

○ 디지털 컴포넌트에 포함된 렌더링 객체 속성 분석

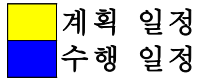
○ 보존 대상 디지털 컴포넌트(한/글, 엑셀, 파워포인트)의 렌더링 객체 보존 방안 연구

❒ 에뮬레이터 프로토타입 설계

○ OS에 종속된 응용소프트웨어 및 디지털 객체 재현을 위한 한국형 에뮬레이션 프로토타입 설계

○ 프로토타입 설계시 행정기관 전자기록의 주요 포맷 4종 (HWP, PPT, XLS, DOC) 고려하여 최종 프로토타입 구현 상세를 결정

## 1.2 총괄연구개발과제의 목표달성도




| 구분 | 세부 추진내용 | 진척률 | 세부 추진 일정 | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 7월 | | | | 8월 | | | | 9월 | | | | | 10월 | | | | 11월 | | | | |
| | | | 8일 | 15일 | 22일 | 29일 | 5일 | 12일 | 19일 | 26일 | 2일 | 9일 | 16일 | 23일 | 30일 | 7일 | 14일 | 21일 | 28일 | 4일 | 11일 | 18일 | 25일 |
| 사례 분석 | **T1.1 에뮬레이션 연구사례 분석** | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T1.1.1 사례분석 계획 및 방법론, 자료조사 | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T1.1.2 사례별 분석 1.Camileon | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T1.1.3 사례별 분석 2. KB, Planets | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T1.1.4 사례별 분석 3. Keep | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T1.1.5 시사점 및 종합토론 | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T1.1.6 집필 | 100 | | | | | | | | | | | | | | | | | | | | | | |
| 유형별 보존 | **T2.1 전자기록물 사례조사** | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T2.1.1 디지털 컴포넌트 포맷조사 내용 확정 | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T2.1.2 디지털 컴포넌트 포맷 현황자료 협조 요청 | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T2.1.3 디지털 컴포넌트 포맷 현황자료 입수 및 분석 | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T2.1.4 기능성 보존요구 대상에 대한 정의 | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T2.1.5 분석결과 종합 및 집필 | 100 | | | | | | | | | | | | | | | | | | | | | | |
| 대상 연구 | **T2.2 디지털 컴포넌트 보존 속성 연구** | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T2.2.1 에뮬레이션 대상 디지털 컴포넌트 정의 | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T2.2.2 디지털 컴포넌트 주요 사례별 보존 속성 정의 | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T2.2.3 생산자 단계의 기능성 보존평가 방법론 개발 | 100 | | | | | | | | | | | | | | | | | | | | | | |
| | T2.2.4 분석결과 종합 및 집필 | 100 | | | | | | | | | | | | | | | | | | | | | | |

| 구분 | 세부 추진내용 | 진척률 | 세부 추진 일정 | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 7월 | | | | 8월 | | | | 9월 | | | | | 10월 | | | | 11월 | | | |
| | | | 8일 | 15일 | 22일 | 29일 | 5일 | 12일 | 19일 | 26일 | 2일 | 9일 | 16일 | 23일 | 30일 | 7일 | 14일 | 21일 | 28일 | 4일 | 11일 | 18일 | 25일 |
| | **T2.3 전자문서 이외의 디지털 컴포넌트 보존대상 속성연구** | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T2.3.1 전자문서유형 이외의 보존대상 연구내용 확정 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T2.3.2 전자문서유형 이외의 보존대상 자료 조사 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T2.3.3 전자문서유형 이외의 보존대상 분석 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T2.3.4 분석결과 종합 및 집필 | 100 | | | | | | | | | | | | | | | | | | | | | |
| 렌더링 유형 연구 | **T3.1 보존문서 변환시 소실되는 렌더링 객체 속성 분석** | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T3.1.1 MS Windows 소프트웨어 이력 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T3.1.2 아래아한글의 소프트웨어 이력 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T3.1.3 MS 오피스 이력 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T3.1.4 HWP 동적 객체 속성 분석 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T3.1.5 PPT 동적 객체 속성 분석 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T3.1.6 XLS 동적 객체 속성 분석 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | **T3.2 디지털 컴포넌트에 포함된 렌더링 객체 속성 분석** | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T3.2.1 HWP 객체 속성 분석 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T3.2.2 PPT 객체 속성 분석 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T3.2.3 XLS 객체 속성 분석 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | **T3.3 보존대상 디지털컴포넌트의 렌더링 객체 보존 방안 연구** | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T3.3.1 DFR 현재 구조 분석 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T3.3.2 보존 방안 연구 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T3.3.3 DFR 구조 연구 | 100 | | | | | | | | | | | | | | | | | | | | | |
| 에뮬레이터 프로토타입 개발 | **T4.1 에뮬레이션 추진 전략 및 개발방안** | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T4.1.1 가상머신 조사 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T4.1.2 프로토타입 시나리오 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T4.1.3 시스템 구성 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T4.1.4개발 방안 수립 | 100 | | | | | | | | | | | | | | | | | | | | | |

| 구분 | 세부 추진내용 | 진척률 | 세부 추진 일정 | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 7월 | | | | 8월 | | | | 9월 | | | | | 10월 | | | | 11월 | | | |
| | | | 8일 | 15일 | 22일 | 29일 | 5일 | 12일 | 19일 | 26일 | 2일 | 9일 | 16일 | 23일 | 30일 | 7일 | 14일 | 21일 | 28일 | 4일 | 11일 | 18일 | 25일 |
| | T4.2 프로토타입 설계 및 구현 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T4.2.1 프로토타입 설계 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T4.2.2 시스템 구현 | 100 | | | | | | | | | | | | | | | | | | | | | |
| | T4.2.3 테스트 | 100 | | | | | | | | | | | | | | | | | | | | | |

# 제2장 총괄연구개발과제의 최종 연구개발 내용 및 방법

## 2.1 연구 내용

❒ 연구 과업내용 재구성

○ 제안요청서 상의 과업항목은 다음과 같은 과제로 구성됨

- 전자기록물 에뮬레이션 국내외 사례연구를 통한 한국형 에뮬레이터 제안
- 행정기관의 전자기록물 유형별 보존 대상 연구
- 전자기록물 렌더링 유형 연구
- 한국형 에뮬레이터 프로토타입 설계 및 구현

○ 제안요청서 상의 과업내용의 범위를 구체적이고 효율적으로 분석하기 위해 과업내용을 재설계할 필요가 있음. 에뮬레이터 제안 내용을 국내의 행정기관 디지털 컴포넌트 유형별 보존 대상 연구 및 렌더링 유형 연구 후, 연구 결과를 바탕으로 제안하여 프로토타입 설계 및 구현 연구로 진행함

- 에뮬레이션 연구 사례 분석
- 디지털 컴포넌트 유형별 보존 대상 연구
- 디지털 컴포넌트의 렌더링 객체 유형 연구
- 프로토타입 설계 및 오픈소스 기반 프로토타입 구현

❒ 단계별 과업 프로세스 및 연구내용

<표 2-1> 단계별 과업의 주요과제 및 연구내용

| 과업 | 주요 과제 | 세부 연구대상 및 범위 |
|---|---|---|
| T1. 에뮬레이션 연구 사례 분석 | T1.1. 에뮬레이션 연구 사례 분석 | - CAMiLEON<br>- KB(Koninklijke Bibliotheek) 에뮬레이션 연구 사례 분석<br>- Planets<br>- KEEP<br>- 에뮬레이터 구현시 기술비용, 소유권 등 해결방안<br>- 한국형 에뮬레이터 제안 |
| T2. 디지털 컴포넌트 유형별 보존 대상 연구 | T2.1. 행정기관의 전자기록물 사례 조사 | - 행정기관의 전자기록 디지털 컴포넌트 포맷조사<br>- 디지털 컴포넌트의 기능성 보존요구 조사 |

| 과업 | 주요 과제 | 세부 연구대상 및 범위 |
|---|---|---|
| | T2.2. 디지털 컴포넌트 보존 속성 연구 | - 디지털 컴포넌트 보존 속성 정의<br>- 디지털 컴포넌트별 보존 속성 평가 방법론 개발 |
| | T2.3. 전자문서유형 이외의 디지털 컴포넌트 보존대상 속성 연구 | - 딥웹(deep web), 데이터세트(data set), SNS(Social Network Service) 등에 대한 기록화 방안 연구 및 에뮬레이션 전략 제안 |
| T3. 디지털 컴포넌트의 렌더링 객체 유형 연구 | T3.1. 보존문서(PDF/A) 로 변환시 소실되는 렌더링 객체 조사 | - 한/글의 OLE 객체, 엑셀의 수식 및 그래프 객체 파워포인트의 애니메이션 및 동영상 객체에 대한 유형 조사 분석 |
| | T3.2. 디지털 컴포넌트에 포함된 렌더링 객체 속성 분석 | - 한/글, 엑셀, 파워포인트 문서에 포함된 동적 요소에 대한 정보 분석<br>- 한/글, 엑셀, 파워포인트 애플리케이션 버전별 포맷 및 속성 변화 분석 |
| | T3.3. 보존 대상 디지털 컴포넌트의 렌더링 객체 보존 방안 연구 | - 국가기록원의 DFR을 기반으로 에뮬레이션을 위한 렌더링 객체 정보 등록 방안 연구 |
| T4. 에뮬레이션 제안 및 프로토타입 구현 | T4.1. 에뮬레이션 추진 전략 및 개발 방안 | - 국내외 전자기록물 에뮬레이션 동향을 분석한 후, 가장 효과적인 에뮬레이션 추진 전략 및 개발 방안을 제시하고자 함<br>- 에뮬레이터 개발에 필요한 기술요소와 기술비용, 특허 및 소유권에 대한 조사를 수행하여 향후 에뮬레이터 개발에 따른 문제점 파악 |
| | T4.2. 에뮬레이터 프로토타입 설계 | - 에뮬레이션 추진 전략 및 개발 방안에 따라 프로토타입 설계<br>- 오픈소스 기반 프로토타입 구현 |

## 2.2 연구 방법

❒ 본 연구에서는 각 과업별로 다음과 같은 방법으로 연구를 진행하였음

❒ 아울러 본 연구팀에서는 연구 범위에 포함되는 특별한 이슈들 중에서 미국과 영국의 장기보존 전략, 에뮬레이션과 관련된 법률적 고려사항, 디지털 포렌직 관점에서 디지털데이터의 증거력에 관해 외부 자문을 진행하고 연구에 필요한 시사점

을 얻고자 하였음

○ 자문 결과는 각각 “3.1.6.가. 사례 분석결과 종합”, “3.1.6.나. 에뮬레이션 전략의 법/비용 이슈”, “3.1.6.다. 한국형 에뮬레이션 전략 제안” 부분을 구성하고 있음

### 2.2.1 T1 에뮬레이션 연구 사례분석

❒ 과업개요

○ CAMiLEON, KB 에뮬레이션, Planets, KEEP 등 전자기록물의 에뮬레이션 관련 해외 사례연구를 통해 시사점 도출

❒ 과업의 성공 요인

○ 사례분석 대상 선정의 적합성

○ 국내에 적용 가능한 에뮬레이션 기술 및 전략을 구분하여 시사점을 도출

❒ 과업 산출물

○ 해외 에뮬레이션 연구 사례에 대한 분석 보고서

○ 기능적 요소를 포함한 디지털 컴포넌트의 장기보존 전략 제안

#### 가. T1.1 에뮬레이션 연구 사례분석

❒ 해외 연구 사례 조사

○ CAMiLEON, KB 에뮬레이션, Planets, KEEP 등 전자기록물의 에뮬레이션 관련 해외 사례연구를 통해 시사점 도출

<표 2-2> 해외 연구사례 개요

| 프로젝트명 | 수행 주체 | 수행 연도 | 주요 연구분야 |
|---|---|---|---|
| CAMiLEON | 미시건대학 & 리드대학 | 1999~2002 | 소프트웨어의 장기 지속<br>에뮬레이션<br>마이그레이션 온 리퀘스트 |
| KB | 네덜란드 국립도서 | 2005~2007 | 장기보존 전략으로서의 다양한 에뮬레이션 전략<br>에뮬레이션 기술의 지속성/유연성 |

| | | | |
|---|---|---|---|
| | 관 & 네덜란드 국립 아카이브 | | |
| Planets | 브리티쉬 라이브러리 | 2006~2010 | 네트워크 서비스를 통한 장기보존 및 접근 방법<br>구형화 된 디지털 자산의 에뮬레이션<br>툴, 서비스 평가 및 테스트<br>보존 계획을 위한 테스트베드 제공 |
| KEEP | EU | 2009~2012 | 파일포맷, 플랫폼, OS의 에뮬레이션<br>에뮬레이션 서비스<br>법적 이슈 및 저작권 |

❒ 대상 선정 방법

- ○ 도서관이나 기록관 콜렉션의 장기보존을 위한 에뮬레이션 전략 및 기술에 관한 연구
- ○ 다양한 수준의 에뮬레이션 모델 검토 및 마이그레이션 전략과의 연계에 관한 연구
- ○ 에뮬레이터 개발, 테스트베드 제공, 에뮬레이션 서비스, 저작권, 기술 비용 등 다양한 관점의 연구

### 2.2.2 T2 디지털 컴포넌트 유형별 보존 대상 연구

❒ 과업개요

- ○ 현재 행정기관의 전자기록 유형 분석을 통하여 보존 대상 유형을 범주화하여 보존 속성 표현 대상 확인
- ○ 디지털 컴포넌트 객체별 중요 속성(property)을 정의하는 방법 및 절차에 대한 방법론을 설계
- ○ 전자문서 기반의 전자기록물 이외에 보존관리 대상으로서 점차 양적 증가를 보이고 있는 데이터세트, 딥웹, SNS 등의 객체 유형에 대한 기록화 방안 검토 및 에뮬레이션 전략 제안

❒ 과업의 성공요인

- ○ 행정기관의 샘플링을 통한 명확한 사례 분석
- ○ 디지털 컴포넌트와 중요 속성에 대한 명확한 이해를 바탕으로 보존 평가 방법론의 구체적 설계

❒ 과업 산출물

○ 샘플 기관의 기능적 요소를 포함한 디지털 컴포넌트 포맷 조사 보고서
○ 디지털 컴포넌트 기능적 요소의 보존 평가 매뉴얼

## 가. T2.1 행정기관의 전자기록물 사례 조사

❒ 행정기관의 전자기록 디지털 컴포넌트 포맷 조사
○ 국내 행정기관에서 에뮬레이션이 필요한 구체적인 대상을 파악하고 연구에 필요한 기초자료를 확보하기 위해서 장기보존 대상인 디지털 컴포넌트의 포맷 현황 파악 실시
○ 디지털 컴포넌트의 포맷 현황을 토대로 행정기관의 디지털 컴포넌트 유형에 대한 현황 분석에 기반한 에뮬레이션 전략을 수립

## 나. T2.2 디지털 컴포넌트 보존 속성연구

❒ 디지털 컴포넌트 보존 속성 평가 방법의 개발
○ 행정기관 전자기록 샘플링 조사에서 수집된 디지털 컴포넌트의 대표적인 사례(한/글, 엑셀, 파워포인트)를 대상으로 중요 보존 대상 속성을 정의
○ T2.1의 관련자 인터뷰 내용을 기반으로 컴포넌트별로 어느 시점에 보존 대상을 지정할 것인가, 기능성을 어느 수준까지 표현할 것인가 등의 보존 요구사항을 분석하여 보존 속성 평가 방법 및 절차 제시

❒ 디지털 컴포넌트별 보존 속성을 보존 전략 수립에 활용
○ 컴포넌트 보존 속성 평가 방법론의 개발을 통해, 향후 미리 정의된 절차에 따라 컴포넌트 등록(attach)시부터 보존하여 표현해야 할 기능적인 속성 정보들을 함께 관리할 수 있도록 제안
○ 이러한 정보는 메타데이터 요소 개발 및 기술 정보로 활용할 수 있도록 제시

## 다. 전자문서 이외의 디지털 컴포넌트 보존대상 속성 연구

❒ 딥 웹(deep web), 데이터세트(data set), SNS(Social Network Service) 등에 대한 기록화 방안 및 에뮬레이션 전략 제안
○ 전자문서 이외의 디지털 컴포넌트 유형을 확인하고, 전자문서와는 다른 이들 유형의 속성을 보존하기 위해서는 기록화 시점부터 보존 전략이 고려되어야 함
○ 대표적으로 딥 웹, 데이터세트, SNS 등의 유형에 대한 기록화 방안을 모색하기 위해 선행 연구에 대해 조사하고, 추가적인 과제를 제시함
○ 딥 웹, 데이터세트, SNS 등의 유형에 대한 에뮬레이션 전략 제안

### 2.2.3 T3 디지털 컴포넌트의 렌더링 객체 유형 연구

❒ 과업개요

- ○ 보존문서 형태(PDF/A)로 변환 시 소실되는 렌더링 객체 확인
- ○ 디지털 컴포넌트에 포함된 렌더링 객체 속성 분석
- ○ 디지털 컴포넌트의 렌더링 객체 보존 방안 연구

❒ 과업의 성공요인

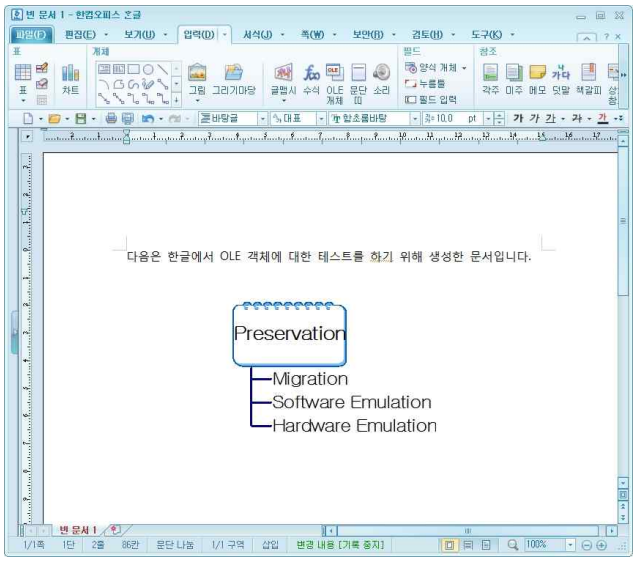
- ○ 한/글(한글과컴퓨터), 엑셀, 파워포인트(마이크로소프트) 등 애플리케이션 개발사로부터 포맷 관련 기술 문서의 확보
- ○ 디지털 컴포넌트 기반의 전자기록물 생성 시점부터 현재까지 애플리케이션 버전 변경에 따른 기능 및 포맷 변화에 대한 조사

❒ 과업 산출물

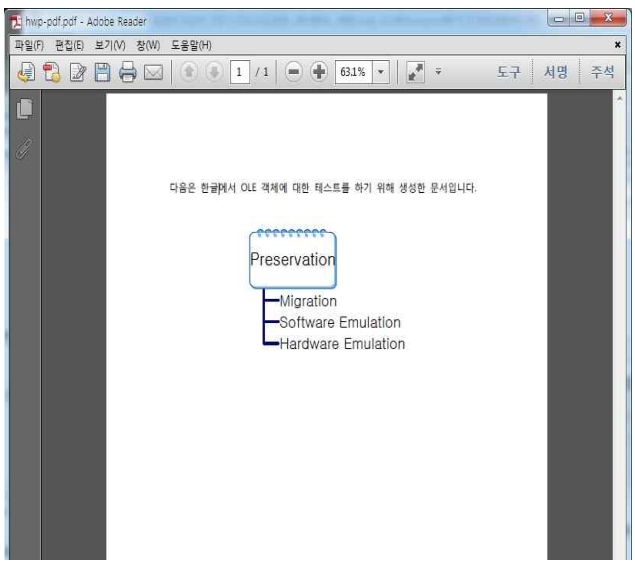
- ○ 버전별 디지털 컴포넌트 (한글, 엑셀, 파워포인트)에 포함된 렌더링 객체 속성 분석 보고서
- ○ 렌더링 객체를 추가한 DFR 구조 개선안

### 가. T3.1 보존문서(PDF/A)로 변환 시 소실되는 렌더링 객체 조사

❒ 한/글의 OLE 컴포넌트 객체에 대한 분석




<그림 2-1> 한/글 OLE 문서



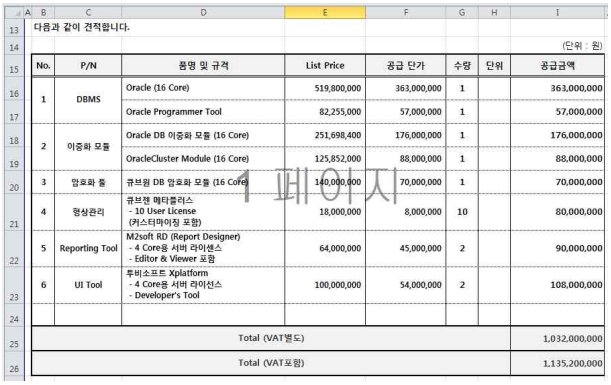

<그림 2-2 > 한/글 PDF/A 변환 문서

○ 한/글에는 OLE 컴포넌트 삽입 기능을 사용하여 외부 객체를 문서 안에 포함하여 한컴에서 표현하지 못하는 정보를 자세하게 표현함

○ 한/글 문서를 변환한 PFD/A 문서에서는 OLE 객체에 대한 정보를 전혀 알 수 없음. 한/글 문서가 갖고 있는 생성 정보가 소실되었음을 알 수 있음

○ 따라서, 본 연구에서는 OLE 객체를 표현하기 필요한 정보가 무엇이고, 어떻게 표현을 하는지 분석하고자 함

❒ 엑셀의 수식 및 그래프 컴포넌트에 대한 분석



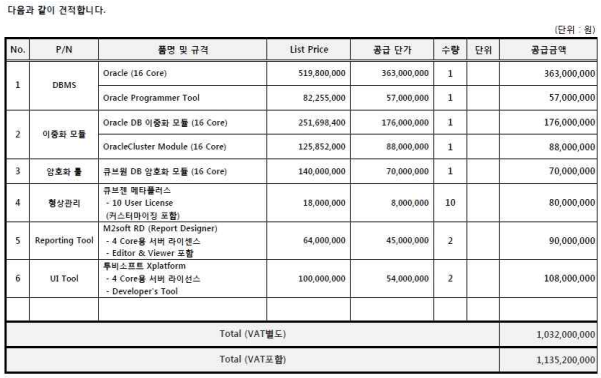
다음과 같이 견적합니다.

(단위 : 원)

| No. | P/N | 품명 및 규격 | List Price | 공급 단가 | 수량 | 단위 | 공급금액 |
|---|---|---|---|---|---|---|---|
| 1 | DBMS | Oracle (16 Core) | 519,800,000 | 363,000,000 | 1 | | 363,000,000 |
| | | Oracle Programmer Tool | 82,255,000 | 57,000,000 | 1 | | 57,000,000 |
| 2 | 이중화 모듈 | Oracle DB 이중화 모듈 (16 Core) | 251,698,400 | 176,000,000 | 1 | | 176,000,000 |
| | | OracleCluster Module (16 Core) | 125,852,000 | 88,000,000 | 1 | | 88,000,000 |
| 3 | 암호화 툴 | 큐브원 DB 암호화 모듈 (16 Core) | 140,000,000 | 70,000,000 | 1 | | 70,000,000 |
| 4 | 형상관리 | 큐브젠 메타플러스<br>- 10 User License<br>(커스터마이징 포함) | 18,000,000 | 8,000,000 | 10 | | 80,000,000 |
| 5 | Reporting Tool | M2soft RD (Report Designer)<br>- 4 Core용 서버 라이센스<br>- Editor & Viewer 포함 | 64,000,000 | 45,000,000 | 2 | | 90,000,000 |
| 6 | UI Tool | 투비소프트 Xplatform<br>- 4 Core용 서버 라이선스<br>- Developer's Tool | 100,000,000 | 54,000,000 | 2 | | 108,000,000 |
| | | | | | | | |
| Total (VAT별도) | | | | | | | 1,032,000,000 |
| Total (VAT포함) | | | | | | | 1,135,200,000 |

1 페이지

<그림 2-3> XLS 수식 문서



다음과 같이 견적합니다.

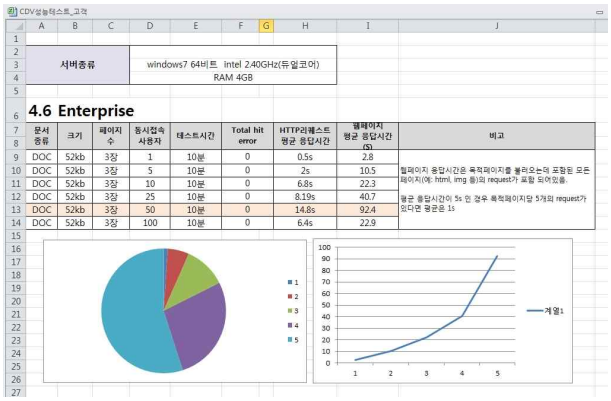
(단위 : 원)

| No. | P/N | 품명 및 규격 | List Price | 공급 단가 | 수량 | 단위 | 공급금액 |
|---|---|---|---|---|---|---|---|
| 1 | DBMS | Oracle (16 Core) | 519,800,000 | 363,000,000 | 1 | | 363,000,000 |
| | | Oracle Programmer Tool | 82,255,000 | 57,000,000 | 1 | | 57,000,000 |
| 2 | 이중화 모듈 | Oracle DB 이중화 모듈 (16 Core) | 251,698,400 | 176,000,000 | 1 | | 176,000,000 |
| | | OracleCluster Module (16 Core) | 125,852,000 | 88,000,000 | 1 | | 88,000,000 |
| 3 | 암호화 툴 | 큐브원 DB 암호화 모듈 (16 Core) | 140,000,000 | 70,000,000 | 1 | | 70,000,000 |
| 4 | 형상관리 | 큐브젠 메타플러스<br>- 10 User License<br>(커스터마이징 포함) | 18,000,000 | 8,000,000 | 10 | | 80,000,000 |
| 5 | Reporting Tool | M2soft RD (Report Designer)<br>- 4 Core용 서버 라이센스<br>- Editor & Viewer 포함 | 64,000,000 | 45,000,000 | 2 | | 90,000,000 |
| 6 | UI Tool | 투비소프트 Xplatform<br>- 4 Core용 서버 라이선스<br>- Developer's Tool | 100,000,000 | 54,000,000 | 2 | | 108,000,000 |
| | | | | | | | |
| Total (VAT별도) | | | | | | | 1,032,000,000 |
| Total (VAT포함) | | | | | | | 1,135,200,000 |

<그림 2-4> XLS 수식 PDF/A 변환 문서

○ 엑셀에서는 수식을 자동 계산을 많이 사용하며, 원문에서는 수식 계산에 의한 최종 결과 값과 수식을 모두 조회할 수 있으나, 엑셀 문서를 변환한 PDF/A 문서에서는 수식을 계산한 결과 값만 존재

○ 문서를 작성한 시점에서의 결과값도 중요하지만, 결과를 도출하기 위해 사용한 수식에 대한 보존도 필요하나, 이러한 수식 정보는 변환된 PDF문서에서는 찾을 수 없으므로 변환에 의해 소실되는 정보에 대해서 분석하고 원본의 정보를 표현하기 위해 필요한 것이 무엇인지 연구하고자 함

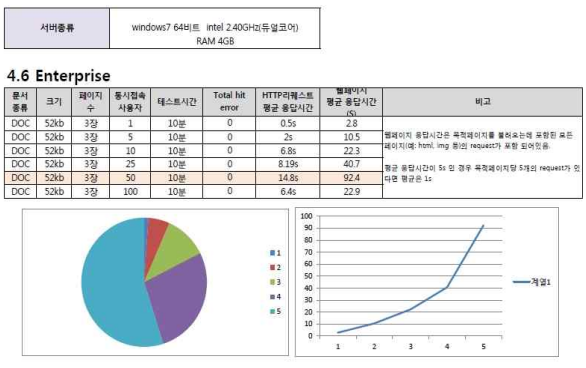


| 서버종류 | windows7 64비트 intel 2.40GHz(듀얼코어) RAM 4GB |
|---|---|

4.6 Enterprise

| 문서 종류 | 크기 | 페이지 수 | 동시접속 사용자 | 테스트시간 | Total hit error | HTTP리퀘스트 평균 응답시간 | 웹페이지 평균 응답시간 (S) | 비고 |
|---|---|---|---|---|---|---|---|---|
| DOC | 52kb | 3장 | 1 | 10분 | 0 | 0.5s | 2.8 | |
| DOC | 52kb | 3장 | 5 | 10분 | 0 | 2s | 10.5 | 웹페이지 응답시간은 목적페이지를 불러오는데 포함된 모든 페이지(예: html, img 등)의 request가 포함 되어있음. |
| DOC | 52kb | 3장 | 10 | 10분 | 0 | 6.8s | 22.3 | |
| DOC | 52kb | 3장 | 25 | 10분 | 0 | 8.19s | 40.7 | 평균 응답시간이 5s 인 경우 목적페이지당 5개의 request가 있다면 평균은 1s |
| DOC | 52kb | 3장 | 50 | 10분 | 0 | 14.8s | 92.4 | |
| DOC | 52kb | 3장 | 100 | 10분 | 0 | 6.4s | 22.9 | |

<그림 2-5> 엑셀 원본 문서



| 서버종류 | windows7 64비트 intel 2.40GHz(듀얼코어) RAM 4GB |
|---|---|

4.6 Enterprise

| 문서 종류 | 크기 | 페이지 수 | 동시접속 사용자 | 테스트시간 | Total hit error | HTTP리퀘스트 평균 응답시간 | 웹페이지 평균 응답시간 (S) | 비고 |
|---|---|---|---|---|---|---|---|---|
| DOC | 52kb | 3장 | 1 | 10분 | 0 | 0.5s | 2.8 | |
| DOC | 52kb | 3장 | 5 | 10분 | 0 | 2s | 10.5 | 웹페이지 응답시간은 목적페이지를 불러오는데 포함된 모든 페이지(예: html, img 등)의 request가 포함 되어있음. |
| DOC | 52kb | 3장 | 10 | 10분 | 0 | 6.8s | 22.3 | |
| DOC | 52kb | 3장 | 25 | 10분 | 0 | 8.19s | 40.7 | 평균 응답시간이 5s 인 경우 목적페이지당 5개의 request가 있다면 평균은 1s |
| DOC | 52kb | 3장 | 50 | 10분 | 0 | 14.8s | 92.4 | |
| DOC | 52kb | 3장 | 100 | 10분 | 0 | 6.4s | 22.9 | |

<그림 2-6> 엑셀 PDF/A 변환 문서

○ 엑셀 원본 문서와 PDF/A로 변환된 문서를 살펴 보면 그래프를 표현하는 수식(예: “=최신!$D$9:$D$13,최신!$I$9:$I$13”와 같은 정보들이 소실됨을 알 수 있음

○ 본 연구에서는 엑셀의 원본 문서가 포함하는 렌더링 객체에 대한 정보를 분석하고자 함

❒ 파워포인트의 애니메이션 및 동영상 컴포넌트에 대한 분석




<그림 2-7> PPT 초기 화면




<그림 2-8> PPT 애니메이션 2번화면




<그림 2-9> PPT 애니메이션 3번화면




<그림 2-10> PPT 애니메이션 4번화면

○ 파워포인트 애니메이션 정보는 보는 사람으로 하여금 이해도를 높이기 위해서 설정하는 기능으로, 예시와 같이 “Preservation”이라는 단어와 A, B, C의 내용이 좌측에서 나타나는 프리젠테이션 문서임

○ 위의 4가지 그림을 보면 파워포인트에서 애니메이션을 사용해서 만든 자료이고, PFD/A로 변환하는 순간 최종본인 <그림 2-10>의 형태만 존재하게 되므로, <그림 2-7>에 있는 “Preservation”이라는 단어는 PDF/A 문서에서는 A, B, C 항목에 의해 소실되므로 애니메이션 컴포넌트에 대한 정보를 분석하고자 함



<그림 2-11> 파워포인트 동영상 문서



<그림 2-12> 파워포인트를 PDF/A로 변환한 문서

○ 파워포인트에 포함된 동영상 정보는 PDF/A로 변환이 되는 순간 대부분의 정보가 손실되므로 동영상 정보를 보존하기 위해서 필요한 객체 요소의 표현 방식에 대하여 연구하고자 함

## 나. T3.2 디지털 컴포넌트에 포함된 렌더링 객체 속성 분석

❒ 한/글, 엑셀, 파워포인트 문서에 포함된 기능적 요소에 대한 정보 분석

○ 한/글의 기능성 객체 요소인 OLE 객체에 대한 정보 조사

○ 엑셀에서 자동 계산을 위한 수식 정보 표현 방법과, 데이터를 이용하여 표현하는 그래프에 대한 정보 조사

○ 파워포인트의 기능적 요소인 애니메이션을 표현하기 위해 사용되는 객체 및 동영상 정보를 표현하기 위해 사용되는 정보 조사

❒ 한/글, 엑셀, 파워포인트 애플리케이션 버전별 포맷 및 속성 변화 분석

○ 디지털 컴포넌트 기반의 전자기록물을 생성하기 시작한 시점부터 현재까지 애플리케이션 버전에 따른 기능 및 컴포넌트 구성 분석

## 다. T3.3 보존 대상 디지털 컴포넌트의 렌더링 객체 보존 방안 연구

❒ 국가기록원의 DFR(Digital Format Registry) 정보 분석

○ 국가기록원의 DFR 정보는 크게 2가지로 파일 포맷 정보와 소프트웨어 정보로 구성되어 있음

○ 파일포맷에 대한 정보는 파일 포맷을 확인할 수 있는 정도의 정보만 등록되어 있으며, 소프트웨어 관련 정보는 소프트웨어의 이름, 버전, 제작사, 연락처 등과 같은 정보만 등록되어 있음

<표 2-3> 파일포맷 구성요소

| Element | 설명 |
|---|---|
| General | 포맷의 일반적인 요소 |
| FormatID | 자동 생성되는 관리번호 |
| Identifier | 포맷 등록 시스템내에서 식별자 |
| FormatName | 포맷명 |
| FormatVersion | 포맷버젼 |
| IPR | 지적 재산권 |
| Developer | 개발자 정보 |
| Support | 포맷지원/유지기관 |
| Technical | 포맷의 기술적인 요소 |
| FormatDisclose | 포맷 공개 수준 |
| Orientation | Text기반/Binary기반 |
| RelatedFormat | 관련 포맷 |
| Specification | 포맷의 스펙 문서 리스트 |
| ExternalSignature | 외부 서명 항목 |
| InternalSignature | 내부 서명 항목 |
| TechnicalEnvironment | 기술적인 환경 |
| Well-formedNote | Well-formed에 관한 설명 |
| ValidityNote | Validity에 관한 설명 문서 위치 |
| Metadata | |
| Management | |
| Source | 정보제공자 |
| SourceDate | 포맷등록시스템 등록일 |
| LastUpdate | 최종 업데이트일 |
| Type | 포맷관리 상태 |
| Note | 정보주기 |

<표 2-4> 소프트웨어 구성요소

| Element | 설명 |
|---|---|
| General | 포맷의 일반적인 요소 |
| SoftwareID | 내부시스템 식별자 |
| Identifier | 외부실별자 |
| SoftwareName | 이름 |
| SoftwareVersion | 버젼 |
| SoftwareType | 타입 |
| Status | 상태 |
| Location | 소장여부 및 소장 위치 |
| ProcessFormat | 처리 가능한 포맷 |
| Language | 지원된 언더 |
| Feature | 일반적인 특징 |
| Image | 쎔네일 이미지 |
| IPR | 지적재산권 |
| Developer | 개발자 정보 |
| Corperation | 개발회사 정보 |
| Support | 지원/유지 기관 정보 |
| ReleaseDate | 발행일 |
| WithdrawDate | 지원 종료일 |
| Note | 정보주기 |
| Technical | 포맷의 기술적인 요소 |
| RelatedSoftware | 관련 소프트 웨어 |
| Document | 소프트웨어 스펙문서 |
| DocumentTitle | 문서의 제목 |
| DocuementType | 문서의 타입 |
| Author | 문서의 저자 리스트 |
| Publisher | 문서의 publisher 리스트 |
| DocumentEdition | 문서의 edition 정보 |
| DocumentDate | 문서 생성 일자 |
| DocumentEnum | serial volume and number |
| DocumentPage | 문서 전체 페이지 |
| Management | |
| Source | 정보제공자 |
| SourceDate | 포맷등록시스템 등록일 |
| LastUpdate | 최종 업데이트일 |
| Type | 포맷관리 상태 |

○ 파일포맷 정보는 <표 2-3>에서 보는 것과 같이 포맷을 나타내는 실제 정보가 아니라 포맷을 설명하기 위한 메타 정보만 등록되어 있음

○ 소프트웨어 정보에서는 이름, 버전, 처리 가능한 포맷, 관련 소프트웨어 등 소프트웨어의 기본적인 정보만 등록되어 있음

○ 따라서, 본 연구를 통해 렌더링 객체를 등록하기 위한 국가기록원의 DFR 개선 방향을 제시하고자 함

❒ 한/글의 OLE 객체, 엑셀의 수식 및 그래프 객체, 파워포인트의 애니메이션, 동영상 객체에 대하여 렌더링을 위한 정보를 기록하기 위한 데이터베이스 구조 연구

○ 렌더링에 필요한 파워포인트의 애니메이션, 동영상, 엑셀의 수식, 그래프 한글의 OLE객체를 위한 자료구조를 분석하고 그 결과를 이용하여 DFR 구조에 대한 개선

방향을 제시하고자 함

### 2.2.4 T4 에뮬레이션 제안 및 프로토타입 설계

❒ 과업개요
- ○ 에뮬레이션 추진 전략 및 개발 방안 연구
- ○ 디지털 컴포넌트를 재현할 수 있는 프로토타입 설계

❒ 과업의 성공요인
- ○ 애플리케이션과 동일하게 렌더링 객체가 포함하는 정보의 표현

❒ 과업 산출물
- ○ 프로토타입 설계서
- ○ 디지털 컴포넌트의 표현을 위한 오픈소스 기반 프로토타입 구현

#### 가. T4.1 에뮬레이션 추진 전략 및 개발 방안

❒ 국내외 전자기록물 에뮬레이션 동향을 분석한 후, 가장 효과적인 에뮬레이션 추진 전략 및 개발 방안을 제시하고자 함
- ○ 현재 전자기록물의 장기보존을 위해 첨부 파일을 PDF/A 형식으로 변환하여 보존함으로써 첨부 파일을 생성한 당시의 구동 애플리케이션이 없더라도 내용 조회는 가능함
- ○ 그러나, 장기보존형식(PDF/A)으로 변환할 경우 기능성 렌더링 객체 요소가 소실되어 재현이 불가능하므로 원문의 기능성 렌더링 객체를 재현할 수 있는 방안 제시가 필요함
- ○ 본 연구에서는 디지털 컴포넌트 기반의 전자기록물을 장기보존하면서 문서의 유효성을 살리기 위한 방안 중에서 에뮬레이션 방식과 마이그레이션 방식에 대한 장단점을 분석하고, 그 중 에뮬레이션 방식을 이용하여 장기보존기록물에 대한 기록을 재현하기 위해 필요한 사항을 조사 분석하여 에뮬레이션 추진 전략과 개발 방안을 제시하고자 함

❒ 에뮬레이터 개발에 필요한 기술요소, 특허 및 소유권에 대한 조사를 수행하여 향후 에뮬레이터 개발에 따른 문제점 파악
- ○ KEEP, planet, CAMiLEON, KB 등 해외 에뮬레이션에 대한 추진 방향과 기술 동향을 파악하여 에뮬레이터 개발에 필요한 기술 요소 정의하고자 함
- ○ 에뮬레이션을 위해서는 장기보존기록물에 포함된 첨부파일의 포맷 양식과 속성

을 정확한 파악이 필요가 있음. 현재 국내외  관련 연구 개발 결과를 살펴보면 파일 포맷에 포함된 정적 요소 객체에 대한 정의는 되어 있으나, 기능적 요소(렌더링 요소)에 대한 속성 정의는 포함되어 있지 않은 상태임

○ 국가기록원에서 2008년도 기록 관리 연구 개발(R&D) 사업 지원을 통해 구축한 "디지털 포맷 및 애플리케이션 기술정보은행(DFR)을 위한 시스템"은 디지털 기록물의 재현을 위한 기술 정보의 수집 및 관리 정보를 위한 요소들을 정의하고 기술정보를 이용하여 디지털 기록물의 장기보존을 위한 기능을 제공하며 또한 국내 실정에 맞는 디지털 기록물의 보존을 위한 포맷 레지스트리를 설계하였음

○ 그러나, 국가기록원의 기술정보은행(DFR)은 다음 표와 같이 포맷에 대한 기술정보와 애플리케이션 소프트웨어에 대한 정적 정보만을 포함하고 있으므로 기능적인 렌더링 객체 요소에 대한 속성을 포함하도록 DFR 구조의 개선에 대한 연구가 필요함

○ 본 연구에서는 국가기록원의 DFR에 추가할 기능성 렌더링 객체 요소에 대하여 파악한 후 에뮬레이터 개발 방안을 제시하고, 기술요소에 대한 특허 및 소유권에 대한 조사를 병행하여 향후 DFR 개선 시에 발생할 수 있는 문제점을 미리 파악하여 해결방안을 제시하고자 함

## 나. T4.2 에뮬레이터 프로토타입 설계 및 구현

❒ 에뮬레이션 추진 전략 및 개발 방안에 따라 OS에 종속된 응용소프트웨어 및 디지털 객체 재현을 위한 오픈소스 기반의 프로토타입 설계

○ 에뮬레이션은 '원래의 소프트웨어 기능이 현재의 컴퓨터 환경에서 재생산될 수 있도록 하는 것으로 전자기록물의 원래 운영 환경을 재생산하는 소프트웨어를 사용하게 하는 보존 방법'으로 정의

- 애플리케이션에 포함된 정적, 동적 기능성을 지닌 객체 요소에 대한 포맷 속성 정보를 DFR에 저장하여 새로운 O/S(운영체제)에서 DFR의 정보를 이용하여 해당 애플리케이션에 대응한 에뮬레이터를 개발하는 방식
- 문서의 유효성을 살리기 위한 마이그레이션 방식은 포맷 속성 정보를 기반으로 데이터를 변환하는 방식이고, 애플리케이션에 대한 에뮬레이터 방식은 포맷 속성 정보를 기반으로 애플리케이션을 변환하는 방식임

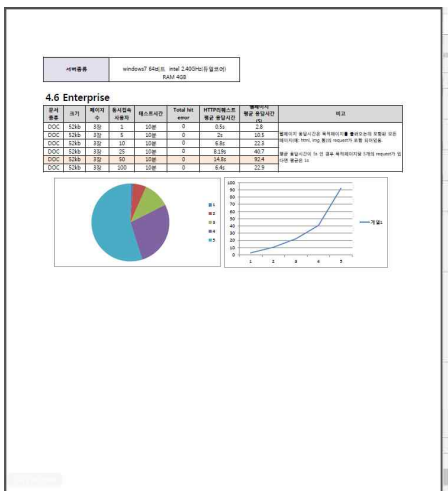
○ 본 연구에서는 조사 분석을 통해 수집된 디지털 객체에 대해 생성 당시의 환경을 재현하여 문서를 표현하는 프로토타입을 설계

○ 에뮬레이터에 대한 테스트를 위해 웹 기반의 문서 조회 화면을 개발하고자 함




<그림 2-13> 웹 기반의 문서 조회 화면

- 웹 기반의 문서 조회 화면은 <그림 2-13>과 같이 기록물 철-기록물 건에 포함된 첨부파일의 목록을 제공하고, 첨부파일 목록에서 'PDF'를 클릭하면 그림 2-14와 같이 PDF 뷰어를 통해 장기보존문서 형태로 변환된 PDF/A 문서를 조회하고, 'XLS'를 클릭하면 <그림 2-15>와 같이 에뮬레이터 상에서 Microsoft Excel을 통해 첨부 파일 원문을 조회하도록 구성




<그림 2-14> PDF 뷰어




<그림 2-15> Microsoft Excel

- 예시와 같이 장기보존형식(PDF/A)으로 변환된 파일에서는 최종 계산된 값만을 조회할 수 있으나, 원문에 대한 에뮬레이션을 통해서는 원문 그대로의 형식을 재현할 수 있음

# 제3장 총괄연구개발과제의 최종 연구개발 결과

## 3.1 에뮬레이션 연구 사례분석

### 3.1.1 사례분석 개요

#### 가. 용어정의

❒ 에뮬레이션(Emulation)

디지털 정보를 생산한 시점에서 사용된 하드웨어, 매체, 운영체제, 소프트웨어의 운용을 그대로 흉내 내어, 그 내용을 읽어내는 프로그램을 통해 재현하는 보존전략.

❒ 에뮬레이터(Emulator)

에뮬레이션을 구현하기 위해 고안된 소프트웨어 내지 응용 프로그램.

❒ 누적 에뮬레이션(Stacked Emulation)

최초에 개발된 에뮬레이터를 재사용하여 에뮬레이터의 중복적인 개발을 방지하기 위한 전략으로, 최초 에뮬레이터를 사용하기 위한 에뮬레이터를 새로운 환경에 맞추어 개발하는 방식.

❒ 마이그레이션된 에뮬레이션(Migrated Emulation)

에뮬레이션에 필요한 환경을 하드웨어 플랫폼이 변화함에 따라 컴파일러를 통해 에뮬레이터를 생성함으로써 누적 에뮬레이션의 단점을 보완하는 기술.

❒ 모듈 에뮬레이션(Modular Emulation)

네덜란드 국립도서관에 의해 개발된 에뮬레이션 방식 중 하나로, 개별 디지털 객체에 요구되는 하드웨어 환경을 모듈의 조합으로 가상 구현하여 원본이 사용될 수 있는 환경을 재현하는 기술.

❒ 가상머신(virtual Machine)

실제로 있는 컴퓨터 시스템에서 소프트웨어적으로 구성하여 만들어진 또 다른 가상의 컴퓨터. 하나의 컴퓨터를 여러 명의 사용자가 동시에 사용할 수 있도록 여러 대의 작은 컴퓨터로 분할 사용하거나, 운영체계나 하드웨어 등의 구성을 달리하여 운영하고자 할 때 주로 사용됨.

❒ 에뮬레이션 가상머신(Emulation Virtual Machine)

호스트 플랫폼과 에뮬레이터 사이에 추가적인 레이 어를 도입함으로써 시간 및 플랫폼으로부터 독립성을 확보하는 방식의 에뮬레이션 보존전략.

❒ UVC(Universal Virtual Computer)

UVC는 단순한 아키텍처를 지닌 가상의 컴퓨터로, 미래의 소프트웨어 개발자가 UVC를 위한 에뮬레이터를 작성할 수 있도록 하는 기본 도구를 지니게 됨. UVC는 에뮬레이션 전략의 변형 형식으로, 기본적으로는 에뮬레이션 전략에 기반을 두면서 마이그레이션 전략상의 몇 가지 요소 역시 함유함.

❒ 기능성

기능성(functionality)이란 일반적으로 “기능이 가지는 역할과 작용 효과의 정도”를 의미하며, IT 분야에서는 “소프트웨어 애플리케이션 혹은 컴퓨팅 장치와 같은 제품으로 사용자를 위해 무엇을 할 수 있는가의 어떤 측면이나 그 합”을 말함.

장기보존 활동에서도 기능성은 디지털 컴포넌트가 가지는 정적, 동적 렌더링을 포함한 다양한 표현 방식을 포괄하는 의미를 가질 것이나 본 연구에서는 특히, “에뮬레이션 대상이 되는” 기능성이라는 맥락에 주목하였으며 이때의 기능성은 디지털 컴포넌트의 보존할 속성이라는 전제를 가짐.

디지털 컴포넌트의 보존대상이 되고 있는 대표적인 기능성으로는 마이그레이션 방식을 통해 인쇄한 형태의 보여주기 기능성이 있음. 그러나 마이그레이션 방식은 인쇄한 형태의 보여주기 기능성만을 남기고 다 없애버림으로서 그 외의 다양한 기능성 소실이 발생함. 이들 중에는 보존 대상이 될 수 있는 OLE 기능, 애니메이션 기능 등의 기능성이 포함되어 있으며 인쇄 방식 이외의 렌더링이 필요한 기능성이 존재함.

따라서 마이그레이션으로는 보존되지 못하고, 인쇄 렌더링 방식으로는 재현 불가능하며 에뮬레이션이 필요한 기능성에 주목할 필요가 있으며 이러한 것들이 바로 본 보고서에서 주목하고자 하는 에뮬레이션 대상이 되는 기능성임.

## 나. 장기보존 전략으로서 에뮬레이션의 특징

### (1) 에뮬레이션의 개념

❒ 에뮬레이션의 개념

- ○ 에뮬레이션은 소프트웨어의 원래 기능이 현재의 컴퓨터상에서 재현될 수 있도록 하기 위해 디지털 정보가 생산·활용된 원래의 운영 환경을 재생산하는 것을 의미함
- ○ 보존 전략으로써 에뮬레이션에서는 하나의 컴퓨터에서 가동되면서 이를 통해

다른 컴퓨터를 실제적으로 재현할 수 있는 프로그램인 에뮬레이터란 소프트웨어를 필요로 하게 되는데, 에뮬레이션은 디지털 객체의 '기능'(functionality)이 중요한 경우 장기보존 전략으로서 큰 의미를 지니게 됨[2)]

❒ 장기보존 전략으로써 에뮬레이션의 채택 이유

○ 장기보존 전략으로써 에뮬레이션의 채택 이유는 디지털 정보자원을 원본 그대로의 정확한 형태로 보존할 수 있는 가능성을 제공해줌

○ 시간의 경과에 따른 시스템 내지 하드웨어·소프트웨어의 노후화에 따라 디지털 정보들을 수차례 변환시킬 필요가 없는 관계상, 매체 변환 및 특정 포맷에 대한 세부 사항들을 파악하는데 소요되는 시간과 비용을 절감할 수 있음

○ 이와 더불어 디지털 정보의 기타 보존 전략들에서는 디지털 정보의 포맷 및 유형, 기타 개별적 특성에 따라 분산화 된 복잡한 보존 전략들을 채택해야 함에 반해, 에뮬레이션은 일관된 보존 전략을 유지할 수 있으며, 디지털 기록을 원래의 포맷 그대로 보존시킬 수 있는 잠재력을 지니고 있음[3)]

### (2) 장기보존 전략으로서의 마이그레이션과 에뮬레이션 특징 분석

❒ 마이그레이션과 에뮬레이션

○ 전자기록의 장기보존 전략으로 가장 광범위하게 사용되는 것은 마이그레이션과 에뮬레이션 전략임

○ 마이그레이션은 하나의 기술로부터 다른 기술로 디지털 객체를 복제 내지 변환하는 방식으로, 디지털 객체의 환경이라기보다는 디지털 객체에 초점을 맞춘 장기보존 전략임

○ 에뮬레이션은 하드웨어 내지 소프트웨어를 흉내 내어 원래의 디지털 객체를 재현하는 방식으로, 원래의 컴퓨팅 환경을 재현하는 관계상 가장 정확한 형태로 장기보존 할 수 있는 전략임

○ 전자기록의 가장 일반적인 장기보존 전략인 이 두 가지 방식의 특징은 다음과 같음

❒ 마이그레이션의 특징[4)]

○ 장기보존 전략으로서 마이그레이션이 가지는 가장 큰 장점은 최신의 기술과 환경에서 기록물을 이용할 수 있다는 것임. 즉 현재의 환경에 변화를 주지 않고 기존의 기록물을 사용할 수 있기 때문에, 추가적인 교육이나 훈련이 필요하지

2) 국가기록원, 『전자기록물 장기보존 전략 연구 및 테스트베드 구축』, 국가기록원, 2008, p. 19.

3) Digital Preservation Testbed, Emulation : Context and Current Status, Digital Preservation Testbed White Paper, 2003, pp. 22~25.

4) 권도윤, 김희섭, 오삼균, 「전자기록물 장기보존을 위한 마이그레이션 전략에 관한 연구」, 『한국기록관리학회지』 9(2), 한국기록관리학회, 2009, pp. 24~25.

않음

○ 데이터를 재사용하거나 자유롭게 편집 활용할 수 있는 형태로 보존할 수 있음

○ 변환이 쉽고 다양한 도구가 제공된다는 점임. 해당 포맷의 공급자를 비롯한 다양한 관련자로부터 변환도구가 지속적으로 개발 공급되고 있기 때문에, 실행에 있어 개발 및 시행을 위한 비용이 낮음

○ 마이그레이션은 일반적으로 디지털 객체의 지적 내용을 보존하기 위한 신뢰성이 높은 방법으로서, 특히 페이지 기반의 기록물의 장기보존에 적합함

○ 마이그레이션은 원래의 응용프로그램을 보유할 필요가 없고, 마이그레이션을 위한 절차가 간결하게 잘 확립되어 있다는 장점이 있음

○ 가장 큰 단점은 데이터와 속성들의 잠재적인 손실 가능성으로, 전자기록 장기보존의 주요 이슈인 전자기록물의 무결성과 진본성을 손상시킬 수 있다는 점임. 마이그레이션은 원본의 논리적 구조 혹은 렌더링(rendering) 방법을 다르게 하여 새로 작성하는 방법으로, 빈번한 변환의 과정에서 언제든지 데이터 또는 속성의 일부가 변경될 가능성을 가지고 있음. 이를 감안할 때 마이그레이션된 디지털 객체는 최초의 원본(original)이란 개념이 아닌 진본(authentic copy)의 개념으로서 다루어 져야 하며, 진본성을 보장하기 위한 다양한 수단을 강구할 필요가 있음

○ 디지털 개체의 특성이 상실될 우려와 함께, 복잡한 레이아웃의 외형이 손상될 가능성도 있음

○ 지속적으로 실행해야만 한다는 것임. 디지털 환경이 변경되는 매 시점마다 마이그레이션 업무를 수행해야할 필요가 있음. 이는 장기적인 마이그레이션 전략의 수립을 어렵게 하는 한 요소이며, 마이그레이션 전략이 장기화될수록 비용이 점차 증가하게 됨을 의미함

❒ 에뮬레이션의 특징

○ 에뮬레이션의 장점

- 호주 국립기록청(NAA)에서는 디지털 정보의 장기보존 전략으로 에뮬레이션이 지닌 장점을 원래 데이터 포맷의 '외형과 느낌'(look and feel)을 재생산할 수 있다는 점을 들고 있음[5)]
- 네덜란드의 Digital Preservation Testbed에서는 에뮬레이션 전략이 지닌 강점으로 원본 형태 그대로 디지털 객체를 보존할 수 있으며, 저장되는 각 포맷 및 각각의 개별 객체에 대해서도 0에 가까운 증분비용(incremental cost)을 제공할 수 있다고 제시. 이와 더불어 에뮬레이션 전략은 보편적으로 적용할 수 있는 단일적이면서도 일관성 있는 보존 전략을 구현할 수 있게 해주며, 사양화된 포맷에 있는 기록물을 포기할 필요 없다는 점 역시 에뮬레이션 전략이

5) National Archives of Australia, Digital Recordkeeping : Guidelines for Creating, Managing and Preserving Digital Records[Exposure Draft], 2004.

지닌 강점으로 꼽고 있음[6]

- 에뮬레이션에 관한 세계적 연구자인 Stewart Granger 역시 디지털 객체의 장기보존 전략으로 에뮬레이션이 다양한 이점을 지닌다고 보고 있는데, 우선 원본 디지털 객체를 가장 정확한 형태로 보존이 가능하다는 점과 더불어, 마이그레이션과 달리 시간의 흐름에 따라 지속적인 변환 프로세스를 거치지 않아도 된다는 점을 제시하고 있음. 그리고 마이그레이션 전략이 각 디지털 객체에 따라 서로 다른 형태를 띠게 되는 것에 비해, 에뮬레이션은 일관된 보존 전략을 수립할 수 있다는 점 역시 다양한 보존 전략들에 비해 에뮬레이션이 지닌 강점으로 제시하고 있음[7]

○ 에뮬레이션의 단점

- 그러나 에뮬레이션이 지닌 단점 역시 존재하는데, 호주 국립기록청(NAA)에서는 에뮬레이션의 직접적인 도구가 되는 에뮬레이터 소프트웨어의 생산이 고도의 기술력을 필요로 하고 많은 비용이 소요되며, 사유(proprietary) 소프트웨어의 에뮬레이션은 지적 자산과 저작권 문제를 발생시켜 장기보존 전략상의 효율성과 안정성을 손상시킨다는 점을 에뮬레이션 전략의 한계로 제시함[8]
- 이와 아울러 Stewart Granger는 에뮬레이션 전략이 고도의 기술 및 비용이 소요될 뿐만 아니라, 마이그레이션에 비해 기술적으로 고려할 사항이 많기 때문에 실패할 확률이 높다고 보고 있으며, 소유권 및 지적 재산권이 존재하는 경우 에뮬레이션 자체가 지적 재산권과 관련된 문제를 야기할 수 있어 보존 전략상의 효율성과 안전성을 침해할 위험성이 있다고 파악됨
- 그리고 에뮬레이션의 대상이 되는 시스템과 소프트웨어 환경 역시 다양하게 존재함으로 지속적인 에뮬레이터 개발이 필요한 점 역시 장기보존 전략으로써 에뮬레이션이 지닌 단점 중 하나로 분석함[9]

### (3) 에뮬레이션의 레벨[10]

❒ 에뮬레이션은 애플리케이션 소프트웨어 레벨, 시스템 소프트웨어 레벨 및 하드웨어 레벨 등 세 단계에서 수행 가능함

❒ 애플리케이션 및 시스템 소프트웨어를 에뮬레이트 하는 것은 설계 및 실행에 관

---

6) Digital Preservation Testbed, *Emulation : Context and Current Status*, Digital Preservation Testbed White Paper, 2003, pp. 46~48.
7) Stewart Granger, "Emulation as a Digital Preservation Strategy", *D-Lin Maganize* 6(10), 2000.
8) National Archives of Australia, *Digital Recordkeeping : Guidelines for Creating, Managing and Preserving Digital Records*[Exposure Draft], 2004.
9) Stewart Granger, "Emulation as a Digital Preservation Strategy", *D-Lin Maganize* 6(10), 2000.
10) 오삼균 외, 「전자기록물 장기보존을 위한 마이그레이션 전략에 관한 연구」, 『한국기록관리학회지』 9(2), 2009, pp. 16~17.

한 지식이 필요로 됨

❒ 하지만 소프트웨어, 하드웨어, 시스템 등을 에뮬레이션 하는 관계로 인해 기술 방식이 매우 복잡하며, 아울러 에뮬레이션에서 일반적으로 나타나는 지적재산권 문제가 발생함

❒ 아울러 애플리케이션 소프트웨어 레벨의 에뮬레이션은 각각의 애플리케이션이 전문 에뮬레이터를 필요로 한다는 문제가 발생함

❒ 하드웨어의 소프트웨어 에뮬레이션(Software Emulation of Hardware)는 소프트웨어를 통해 하드웨어 아키텍처를 모방함으로써 수행되는 방식임

❒ 이 방식에서 에뮬레이터는 하드웨어 상에서 직접 구동하는 대신, 호스트 OS의 상위에 위치하며, 애플리케이션 프로그래밍 인터페이스의 사용과 같이 호스트 OS에 의해 제공되는 서비스처럼 유연하게 이용할 수 있다는 장점이 있음

❒ 에뮬레이터는 호스트 플랫폼 상에서 가동되고, 에뮬레이션 하에서 목적 플랫폼의 하드웨어를 재상산하는 소프트웨어 프로그램임

### (4) 에뮬레이션의 향후 연구과제[11)]

❒ **기술적 개발 문제**

○ 에뮬레이터가 타겟 시스템의 속도, 디스플레이, 리프레시 율, 명암 및 색상, 사운드 등 모든 관련 기능을 완벽히 재생산할 수 있도록 하기 위해서는 현 기술력에 대한 보다 심도 있는 연구가 수행되어야 함

❒ **유효성(validation)의 확보**

○ 유효성은 현재 모든 디지털 보존전략상의 공통된 문제임

○ 에뮬레이션에 기반을 둔 보존전략은 유효성을 돕기 위한 자동화 방식의 형식테스트(formal test) 및 튜링 테스트(turing test)에 대한 연구 수행이 필요

❒ vernacular extraction의 수행 방식

○ 에뮬레이션이 지닌 강점 중의 하나는 원본으로부터 직접 대체 사본의 vernacular extraction이 용이하다는 점임. 따라서 vernacular extraction이 자동

11) Digital Preservation Testbed, *Emulation : Context and Current Status*, Digital Preservation Testbed White Paper, 2003, pp. 46~55.

적 내지 반자동적으로 수행되도록 하는 방안에 대한 연구가 필요함

❒ I/O device 및 storage device access의 에뮬레이팅 문제
- ○ 컴퓨터 플랫폼의 에뮬레이팅은 단지 CPU의 에뮬레이팅만으로는 부족함
- ○ 디지털 기록의 정확한 재현을 위해서는 I/O device나 storage device access와 같은 주변 장치 역시 에뮬레이팅을 해야 함

**❒ 다차원 기록의 에뮬레이션 문제**
- ○ 디지털 기록이 멀티미디어 문서로 구성되거나 여러 컴퓨터들 간의 기술적 연계 속에 생산되는 경우에는, 모든 관련 컴퓨터 및 네트워크 하드웨어들을 에뮬레이션 하는 방안을 연구해야 함

**❒ 메타데이터의 포착 및 관리방안**
- ○ 에뮬레이션에서는 특정 기술관련 메타데이터가 별도로 생산·저장될 필요가 없다고 그동안 제기되어 옴
- ○ 하지만 에뮬레이션을 통해 기록을 재생산하는데 필요한 절차를 기술하는 메타데이터들이 생산되어야 하며, 에뮬레이션에 기반을 둔 보존전략에서는 어떠한 종류의 메타데이터가 필요한지 향후 연구되어야 함

**❒ 지적 재산권 문제**
- ○ 에뮬레이션의 기반이 되는 원래의 재현 소프트웨어는 대부분 사유 포맷인 관계상, 에뮬레이션을 기초로 한 보존전략을 구현하기 위해서는 사유 포맷과 관련된 저작권 문제를 해결하는 방안을 강구해야 함

❒ 처리 및 비용 모듈의 개발
- ○ 디지털 기록의 장기보존은 무엇보다 경제적 요소가 성공의 기반임
- ○ 따라서 에뮬레이션 전략상의 적절한 처리 및 비용모듈이 개발되어야 함

## 다. 사례분석의 목적과 필요성

- ○ 최근 들어 전자기록물의 유형은 더욱 다양화되고, 행정기관에서 생성되는 전자기록물 유형 역시 과거와 같은 정적인 텍스트 기반이 아닌 기능적 구성요소들이 증가하면서, 다양한 유형의 전자기록물에 대한 장기보존 전략 수립이 필요함
- ○ 현재까지 개발된 거의 대부분의 애플리케이션은 MS의 윈도우 운영환경에서 동작하도록 개발되어 있으나, 운영 환경이 클라우드 서비스 환경으로 바뀔 경우 서버 환경에서 애플리케이션의 구동이 이루어져야 함. 그러나 이미 사양화된

애플리케이션을 클라우드 환경에서 동작하도록 변환하는 기술적, 경제적 측면에서 어려운 경우에는 보관한 애플리케이션의 동작이 불가능하게 되어 원문을 인식하지 못하는 상태가 발생할 것으로 예측됨

○ 이와 더불어 애플리케이션의 보관과 함께 생성되는 파일 형식에 대한 보존이 필요함. 현 시점에서 가장 중요한 것은 사라져 가는 애플리케이션의 파일 형식을 정확하게 파악하여 레지스트리를 구축하는 것이 필요. 이를 위해 국가기록원에서는 DFR(Digital Format Registry)를 구축하였으나, 현재의 DFR은 파일 포맷과 속성 정보만을 담고 있음. 스프레드시트나 워드프로세서, 파워포인트 등 각종 애플리케이션 파일에서 제공하는 기능에 대한 표현 방법 연구와 이를 재현하기 위한 방법을 고려해야 함

○ 이를 위해 마이그레이션과 함께 전자기록물 장기보존의 양대 전략인 에뮬레이션에 대한 본격적인 연구가 필요함

○ 장기보존 전략으로서 에뮬레이션이 지닌 강점은 다음과 같음

- 디지털 정보자원을 원본 그대로의 정확한 형태로 보존할 수 있는 가능성을 제공해줌
- 마이그레이션과 같이 디지털 정보들을 수차례 변환시킬 필요가 없는 관계상, 매체 변환 및 특정 포맷에 대한 세부 사항들을 파악하는데 소요되는 시간과 비용을 절감할 수 있음
- 디지털 정보의 포맷 및 유형에 관계없이 일관된 보존 전략을 유지할 수 있음

○ 이에 21세기 이후 디지털 객체의 장기보존 전략으로서 에뮬레이션을 중점적으로 연구한, 아래의 도표에 제시된 해외의 4가지 선진 프로젝트를 분석함으로써, 우리나라 공공기록물의 장기보존 전략을 위한 벤치마킹 요소를 도출할 필요가 있음

**<표 3-1> 해외 에뮬레이션 프로젝트별 벤치마킹 요소**

| 프로젝트명 | 수행 주체 | 수행 연도 | 주요 연구 분야 | 벤치마킹 요소 |
|---|---|---|---|---|
| CAMiLEON | 미시건 & 리드대학 | 1992~2002 | - 소프트웨어의 장기 지속에뮬레이션<br>- 마이그레이션 온 리퀘스트 | - 다양한 수준의 에뮬레이션 비교자료<br>- 도서관용 보존 툴<br>- 마이그레이션 및 에뮬레이션 전략 가이드라인<br>- 다양한 디지털 객체의 속성 정의 |

| | | | | |
|---|---|---|---|---|
| | | | | - 디지털 객체 보존 전략 |
| KB | 네덜란드국립도서관 & 네덜란드 국립 아카이브 | 2005~2007 | - 장기보존 전략으로서의 다양한 에뮬레이션 전략<br>- 에뮬레이션 기술의 지속성/유연성 | - 에뮬레이션 방법론 4종<br>- 모듈 에뮬레이터: Dioscuri<br>- 에뮬레이터 명세 문서 (ESD) |
| Planets | British Library | 2006~2010 | - 네트워크 서비스를 통한 장기보존 및 접근 방법<br>- 구형화 된 디지털 자산의 에뮬레이션 툴, 서비스 평가 및 테스트<br>- 보존 계획을 위한 테스트베드 제공 | 보존계획 도구 : Plato<br>에뮬레이션 프레임워크 : GRATE<br>Planet Testbed |
| KEEP | EU | 2009~2012 | - 파일포맷, 플랫폼, OS의 에뮬레이션<br>- 에뮬레이션 서비스<br>- 법적 이슈 및 저작권 | KEEP 전송 프레임워크<br>KEEP 에뮬레이션 프레임워크 |

### 3.1.2 CAMILEON 프로젝트

#### 가. CAMILEON 프로젝트 개요 및 목적

##### (1) CAMILEON 프로젝트 개요[12)]

❒ CAMILEON(Creating Creative Archiving at Michigan and LEEDS: Emulating the Old on the New)은 미국 미시건대학교와 영국 리드대학교과 공동으로, 디지털매체의 장기적 보존방법론을 에뮬레이션을 중심으로 하여 개발 연구 프로젝트임

❒ CAMiLEON 프로젝트는 디지털 자료들의 장기적 보존을 위한 기술적 전략들을 광범위하게 개발하고 평가하는 과정을 통해, 상황에 따라 어떠한 장기보존 전략을 적용할 수 있을지에 대해 연구

❒ 이용자 평가 연구(User Evaluation Studies)와 보존 비용분석은 이러한 전략들이

12) http://www2.si.umich.edu/CAMILEON/index.html

언제 어디에서 이용되어야 할지에 대한 방안을 제공

❒ CAMiLEON이 주요 보존전략으로 채택하고 있는 에뮬레이션은 구형화된 시스템이 미래의 알려지지 않은 시스템 상에서도 실행될 수 있도록 하는 디지털 보존 전략으로서 제안되어 왔으며, 이러한 전략은 디지털 문서들을 원래의 소프트웨어와 더불어 검색하고, 디스플레이하고, 이용할 수 있도록 함

### (2) CAMILEON의 주요 목표[13)]

❒ CAMiLEON은 디지털 객체의 원래의 기능성(functionality)과 '외형과 느낌(look and feel)'의 장기적 유지를 위한 선택사항들을 탐구함

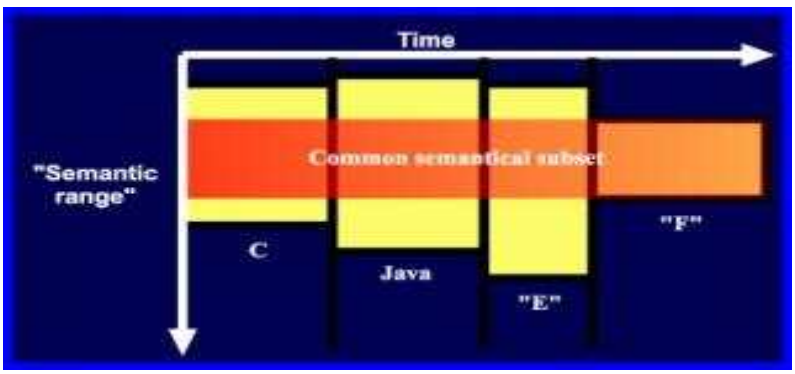
❒ 디지털 객체의 장기적 보존과 접근을 위한 장기적 전략으로서 기술 에뮬레이션(technology emulation)을 조사연구 수행함

❒ 이러한 에뮬레이션이 일련의 디지털 보존전략들과 어디서 어떻게 부합하는지(즉 어떠한 상황에서 에뮬레이션 전략이 적합한지)에 대한 검토를 주요한 목표로 함

## 나. CAMILEON의 보존전략

### (1) 소프트웨어 수명의 장기화(software longevity)[14)]

❒ CAMiLEON의 핵심 보존전략은 디지털 객체의 원래의 바이트스트림을 해독하고 표현하는 소프트웨어의 설계와 생산을 포함




<그림 3-1> 소프트웨어 수명의 장기화 전략 도래

출처: http://www2.si.umich.edu/CAMILEON/index.html

13) http://www2.si.umich.edu/CAMILEON/index.html
14) http://www2.si.umich.edu/CAMILEON/reports/camresearch/research.html#softwarelongevity

❒ 소프트웨어 도구는 물론 디지털 객체로서 시간의 흐름에 따라 보존될 필요가 있으며, 이를 위해 CAMiLEON은 소프트웨어 수명의 장기화에 대한 기술을 개발함

❒ 시간의 흐름에 따라 하나의 언어에서 다른 언어로의 간단한 소프트웨어 마이그레이션 통로를 생성함으로써, 소프트웨어 보존 도구는 경제적이고 정확하게 장기간 동안 유지될 수 있음

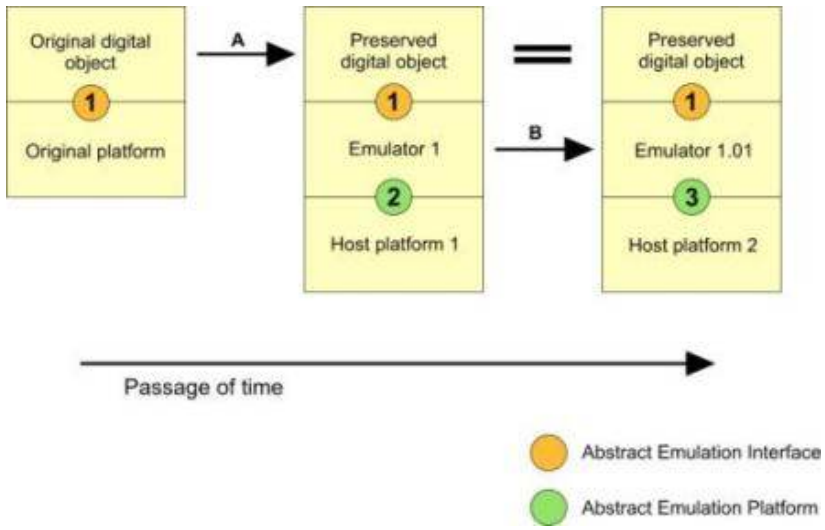
❒ 이는 소프트웨어 도구가 하나의 언어에서 다른 언어로 경제적인 방법으로 마이그레이션 될 수 있도록 함

### (2) 에뮬레이션 방식[15)]

❒ 에뮬레이션은 이전 시기에 생산된 디지털 객체를 열람하고, 이용하는데 요구되는 기술적 환경을 현재의 하드웨어 상에서 재생산하는 것을 의미함

❒ 최근까지도 에뮬레이션에 대한 실질적인 작업의 부족으로 인해 이러한 논쟁은 순전히 이론적인 수준으로만 남겨져 있으며, CAMiLEON 프로젝트는 이러한 이론과 실제의 불균형을 시정하기 위해 시도하였음

❒ 에뮬레이션 과정은 아래의 그림과 같음




**<그림 3-2> CAMiLEON 프로젝트의 에뮬레이션 과정**

출처: David Holdsworth, Paul Wheatley, "Emulation, Preservation, and Abstraction". <http://sw.ccs.bcs.org/CAMiLEON//dh/ep5.html>

15) David Holdsworth, Paul Wheatley, "Emulation, Preservation, and Abstraction", http://sw.ccs.bcs.org/CAMiLEON//dh/ep5.html

❒ 위의 그림은 먼저 원본 디지털 객체를 보존된 디지털 객체로 변형하고, 그것이 원본의 중요한 특성에의 적절한 접근을 제공하는 에뮬레이션 하에서 실행될 수 있음을 의미함

❒ 시간의 흐름은 화살표 B에 의해 나타나는 변형을 초래하며, 그 안에서 플랫폼1의 노후화는 다른 플랫폼 상에서 실행되는 에뮬레이터를 업데이트함으로써 조정. 여기서 등호는 보존된 디지털 객체가 항구적으로 같은 바이트 스트림으로 구성되어 있음을 의미함

❒ 논쟁의 핵심은 두개의 화살표에 의해 표시되는 변형이 경제적으로 수행될 수 있다는 것으로 이 가운데 가장 중요한 것은 디지털 구성요소들 사이에 있는 접합점 상의 동심원들에 의해 표시되는 추상적인 인터페이스의 설계임

❒ 1번 인터페이스는 원본 디지털 객체의 성공적인 운영에 필요한 API(Application Program Interface)이고, 에뮬레이터 1은 추상적인 에뮬레이션 세계 안에서 이러한 인터페이스를 재현하며, 이러한 인터페이스는 영원히 같은 상태로 남아있게 됨

❒ 에뮬레이터 자체는 하나의 디지털 객체이며 그것의 성공적인 운영을 위해 인터페이스 2에 의존. 핵심적인 전략적 목표는 거의 변경 없이 시간 테스트에서 생존할 가능성이 있는 유일한 특징을 포함하는 인터페이스 2를 선택함으로써 이루어지는 “미래의 입증(future proofing)”임. 따라서 인터페이스 3은 인터페이스 2와 매우 흡사하며, 호스트 2에 대한 에뮬레이터(에뮬레이터 1.01)는 에뮬레이터 1로부터 바로 도출될 것임

### (3) Migration on Request

❒ Migration on Request의 목적[16)]

- ○ 시간의 흐름에 따라 하나의 포맷에서 또 다른 포맷으로의 마이그레이션은 보존 작업을 수행하기 위해 널리 인정된 방법이지만, 이상적인 것과는 거리가 있음. 즉, 전통적인 마이그레이션은 변형하는 동안의 에러와 변형이 증가하며, 고비용이 소요되는 단점을 지니고 있음
- ○ 이에 따라 마이그레이션의 지속적인 수행에서 원형의 훼손을 최소화할 수 있는 마이그레이션 방법론을 개발함

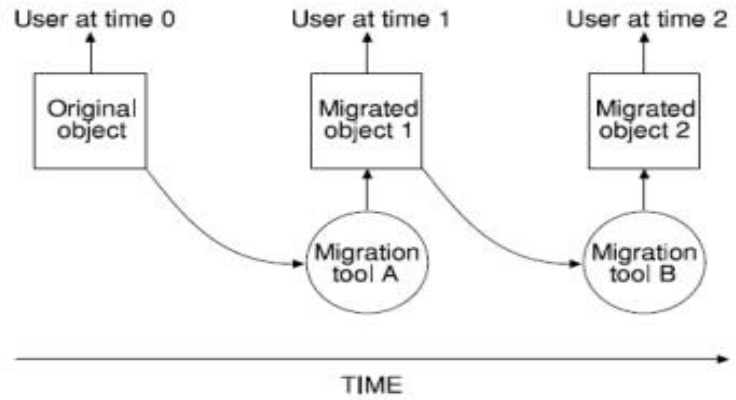
16) http://www2.si.umich.edu/CAMILEON/reports/mor/index.html

○ 아울러 벡터 그래픽(vector graphics) 자원을 대상으로 테스트를 수행함

❒ CAMiLEON의 마이그레이션 개선 방식[17]

○ CAMiLEON은 이에 대해 Migration on Request라는 대체적인 접근법을 개발했는데, 이는 이용의 시점에 객체를 마이그레이션 하는 도구와 함께 원래의 바이트스트림을 유지함으로써 보다 정확하고 비용 효율적이 되도록 함

○ 전통적인 마이그레이션의 경우 변환과정에서 파일에 원본과 차이점이 발생할 경우, 향후 지속적인 변환과정에서 이 오류가 누적됨. 이것은 원본의 정확성에 큰 문제점 발생. 아울러 마이그레이션이 수행될 때마다 새로운 파일 포맷이 생성되게 됨




<그림 3-3> Migration on Request 방식 도해

출처: http://www2.si.umich.edu/CAMiLEON/reports/mor/onrequest.html

○ Migration on Request는 이러한 단점을 극복하기 위해, 위의 그림과 같이 하나의 원본파일을 보존하며, 새로운 환경이 나타날 때마다 여기에 적합한 마이그레이션 도구를 개발하는 접근방식을 취함

○ 따라서 마이그레이션 과정에서 발생한 오류가 누적되는 현상을 피할 수 있으며, 새로운 환경이 등장할 때마다 새로운 도구를 개발할 필요 없이, 하나의 포맷을 대상으로 하는 변환도구를 개발할 수 있게 됨

○ Migration on Request는 소프트웨어 수명의 장기화(software longevity) 기술을 이용하여 모든 디지털 객체들을 재현하며 오랜 시간동안 유지될 수 있음. 그리고 디지털 객체는 단순히 디지털 객체의 원래 포맷으로 보관되어 항상 원래의 디지털 포맷으로 되돌아올 수 있으므로, 변환 과정동안 초래되는 오류가능성을 크게 감소시킴. 더불어 원래의 포맷에서 현재의 포맷으로 변환하기 위해 항시 단 한 번의 변환과정만 거치게 되며, 어떠한 경우에든 아카이브에 보유 중인 디지털 객체가 원본이 됨[18]

17) http://www2.si.umich.edu/CAMILEON/reports/mor/conventional.html

❒ Migration on Request의 강점[19]

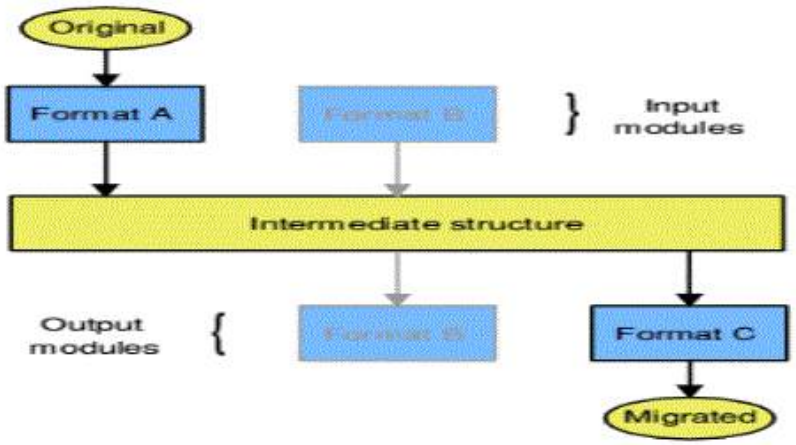
○ 특정 파일포맷을 읽고 해석한 코드는 단지 한번만 실행됨

○ 하나의 마이그레션 단계만을 이용함으로써, 마이그레이션의 정확성을 제고함

○ 디지털 객체가 원래 형태로 보존되기 때문에 진본성 문제들이 단순해짐

○ 마이그레이션 도구가 오직 마이그레이션 요구가 있을 경우에만 실행되기 때문에, 대용량 디지털 객체가 보존될 때 대용량의 저장기능 제공함

❒ 마이그레이션 Modular Design[20]




**<그림 3-4> CAMiLEON 프로젝트의 마이그레이션 모듈 디자인**

출처: Paul Wheatley, "Migration: A CAMiLEON Discussion Paper", Ariadne 29, 2001. <http://www.ariadne.ac.uk/issue29/camileon>

○ Migration on Request 도구는 단순한 Modular Design에 의해 수행됨

○ 위의 그림에서 제시한 바와 같이, 우선 투입 모듈에서는, 원본의 요소 및 파일 계층을 추출함과 더불어, 이들을 Intermediate Structure에 저장함

○ 산출 모듈에서는 Intermediate Structure에 저장된 원본의 요소 및 파일 계층을 불러들임과 더불어, 이들 요소들을 요구되는 포맷에 저장함

○ Intermediate Structure는 모든 지원 포맷으로부터의 특징을 적절히 재현할 수 있어야 함

○ Migration on Request는 네덜란드 KB의 마이그레이션 테스트에서 사용되어 장기보존을 위해 활용될 수 있음이 보고된 바 있으며, SOA(Service Oriented Architecture) 설계에 기반을 둔 문화유산기관 지원단체인 CRiB에서 진행하는 디지털 객체의 마이그레이션 서비스에서도 표준포맷 지정변환 방법과 함께 채

18) 오삼균 외, 「전자기록물 장기보존을 위한 마이그레이션 전략에 관한 연구」, 『한국기록관리학회지』 9(2), 2009, p. 35~36.
19) Paul Wheatley, "Migration: A CAMiLEON Discussion Paper", Ariadne 29, 2001. <http://www.ariadne.ac.uk/issue29/camileon>
20) <http://www2.si.umich.edu/CAMILEON/reports/mor/modular.html>

택되어 사용되고 있음[21]

## 다. 지적 재산권 이슈 및 비용효과

❒ 지적 재산권 이슈

- ○ CAMiLEON에서는 기술적 측면과 함께 보존 활동시 고려해야 할 지적 재산권 문제를 연구함
- ○ 지적재산권 검토 대상 콘텐츠로는 프로 사진가들이 촬영 한 사진, 대회 출품작 사진, 신문 및 잡지 등에 실린 텍스트, 1980년부터 1986년까지 영국에서 열린 뉴스와 스포츠 이벤트의 동영상, UK Ordnance Survey가 라이선스를 가진 지도 등이 있음
- ○ 미국 저작권법 42조 1항과 2항에 의하면, 도서관 및 아카이브즈에서 영구보존을 위해 기록물을 복사하는 경우, 소실이나 훼손되어 타 기관으로부터 복사해 오는 경우에 한해 복본 생성을 허용함
- ○ Domesday 디스크의 콘텐츠 경우, 원본 디스크는 제공자의 소유이며, 에뮬레이션 결과로 구동되는 콘텐츠는 새로운 버전이 아닌 복본으로 인정됨

❒ 에뮬레이션 비용 효과[22]

- ○ 전략으로서의 에뮬레이션 비용 효과는 개발된 에뮬레이터의 수명과 에뮬레이터를 업데이트하는 것이 얼마나 용이한가에 의존함
- ○ 에뮬레이터의 수명은 에뮬레이터에 대해 선택된 최초 플랫폼에 의해, 또한 에뮬레이터가 얼마나 많은 비표준 소스코드를 포함하고 있는가에 의해 결정됨
- ○ CAMiLEON에서는 에뮬레이터의 장기 수명을 보장하기 위한 최선의 해결책으로 C 프로그래밍 언어를 제안함

## 라. CAMiLEON 적용사례: BBC Domesday 프로젝트 보존[23]

❒ BBC Domesday 프로젝트는 11세기 영국의 인구조사 결과인 Domesday Book의 900주년을 기념하여, BBC와 Acorn Computer Ltd, Philips, Logica가 공동으로 조사한 영국 국민들의 삶에 대한 기록임

❒ 1984년에서 1986년 사이에 작성되어 1986에 발표됨

❒ Domesday는 영국 각지의 지도, 다양한 지역사회 사람들의 일상생활상이 지도,

21) 국가기록원, 『전자기록 관리와 보존을 위한 국제협력 아젠다 개발』, 국가기록원, 2007.
22) 국가기록원, 『전자기록물 장기보존 전략 연구 및 테스트베드 구축』, 국가기록원, 2008, p. 30.
23) 국가기록원, 『전자기록물 장기보존 전략 연구 및 테스트베드 구축』, 국가기록원, 2008, p. 21.

사진, 통계자료, 동영상을 수록함

❒ Domesday 프로젝트의 저장매체는 LV-ROM(Laser Vision Read Only Memory)임

❒ Domesday 프로젝트는 JPEG와 같은 포맷이 나오기 전에 진행되어 사양화에 대한 우려가 존재하는 대표적인 사례로 꼽힘

❒ 2002년 이후 Domesday 프로젝트의 보존 위기가 가시화 되었는데, LV-ROM을 사용할 수 있는 PC가 사라져 갔고, 동영상의 경우 최선 PC로는 정확한 재생이 불가능했기 때문임

❒ 이에 따라 CAMiLEON에서는 Domesday 프로젝트의 기록을 시간의 흐름 후에도 이용할 수 있도록 에뮬레이션 보존전략 연구를 수행함

❒ CAMiLEON 연구진들은 과거의 시스템을 구현할 수 있는 에뮬레이터를 개발하여 원본 기록과 함께 에뮬레이션 소프트웨어 내에 캡슐화 함

❒ BBC의 'BeebEm" 에뮬레이터를 사용하여 Domesday의 콘텐츠를 이용 가능하게 하였지만, 원본 디스크에 대한 다양한 참여자 사이의 저작권이 복잡하게 얽혀 있어 실제 서비스는 불가능하였음

### 3.1.3 Koninklike Bibliotheek(KB) 프로젝트

#### 가. KB 에뮬레이션 개요

❒ 개요

- ○ 에뮬레이션 기반의 새로운 보존전략에 관한 연구 프로젝트
- ○ 2004년 네덜란드국립도서관(Koninklike Bibliotheek 이하 KB)과 네덜란드 국립아카이브의 공동 프로젝트로 2005년부터 2007년까지 진행됨
- ○ KB는 디지털 객체의 장기보존 전략으로서 에뮬레이션의 다양한 가능성에 대해 연구함
- ○ 프로젝트 결과 누적 에뮬레이션과 마이그레이션된 에뮬레이션, 에뮬레이션 버추얼 머신, UVC 기반 에뮬레이션 등이 제시되었고 개념모형인 모듈 에뮬레이션을 제안함
- ○ 2007년 모듈화 에뮬레이터인 Dioscuri가 개발되었고, 이 에뮬레이터는 Planets

프로젝트의 일부분으로 흡수됨

❒ 핵심 연구 분야

○ 미래의 플랫폼에서 디지털 원본을 렌더링할 수 있는, 보존전략으로서의 에뮬레이션 가능성을 측정

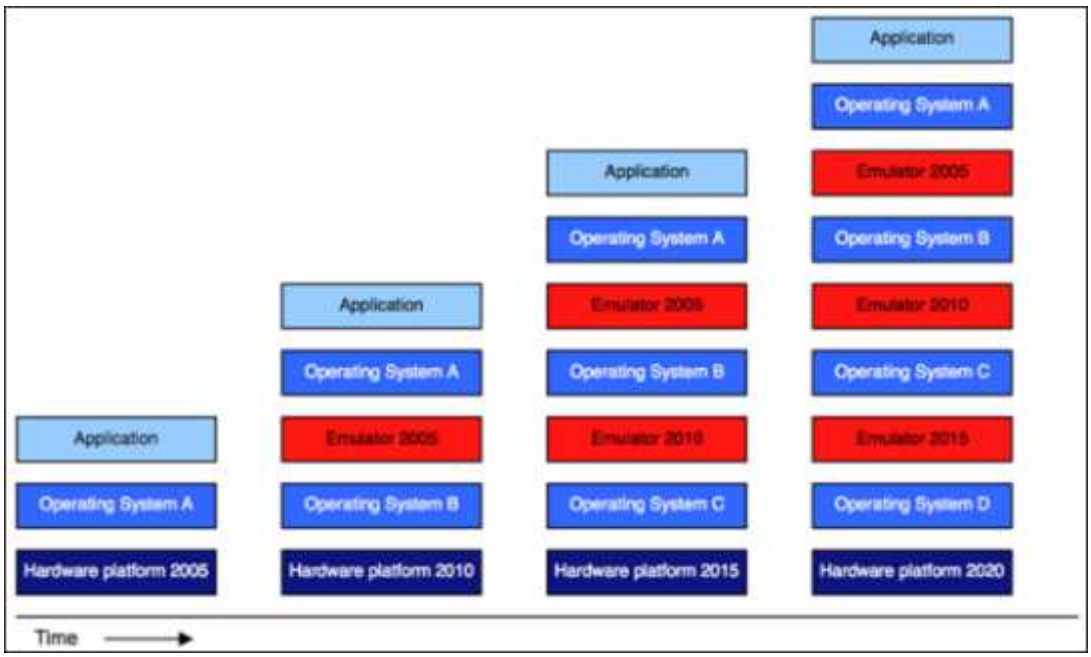
○ 단일 에뮬레이션 접근으로 다양한 디지털 객체를 보존할 수 있는지 평가

## 나. 에뮬레이션의 4가지 전략

### (1) 누적 에뮬레이션(Stacked Emulation)[24]

❒ 누적 에뮬레이션의 기본 모토는 에뮬레이터의 재사용으로, 최초에 개발된 에뮬레이터를 재사용하여 에뮬레이터의 중복적인 개발을 방지함

❒ 즉 과거의 플랫폼을 재구성하기 위해 각기 실행되는 복합적인 에뮬레이터를 필요로 하는 플랫폼 의존적인 에뮬레이션으로, 가장 일반적인 방식의 에뮬레이션 형태임




<그림 3-5> 누적에뮬레이션 방식

출처: Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), p. 17.

24) Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), pp. 16~17.

❒ 2005년도에 처음으로 개발된 에뮬레이터를 2010년 환경에 맞추어 개발하고, 다시 2010년 에뮬레이터를 2015년도 환경에, 그리고 2015년 에뮬레이터를 2020년 환경에 맞추어 개발함

❒ 이는 곧 최초 에뮬레이터를 사용하기 위한 에뮬레이터를 새로운 환경에 맞추어 개발하는 전략임

❒ 호스트 플랫폼의 한 부분에 대한 상세한 에뮬레이터에 의해 얻어지는 효율성을 지님

❒ 에뮬레이터가 플랫폼 환경의 일정 부분만을 상세화하기 때문에, 다른 플랫폼과의 호환성이 저하되며 영속성이 적음. 또한 각각의 에뮬레이터가 플랫폼 아래에서 매우 의존적이 되며, 에뮬레이터들 중 하나가 오류 내지 유실된다면 이 에뮬레이터에 의존하고 있는 체인은 불가피하게 손실될 것이고, 애플리케이션과 문서에 접근할 수 없는 결과가 초래됨[25)]

### (2) 마이그레이션된 에뮬레이션(Migrated Emulation)[26)]

❒ 에뮬레이션에 필요한 환경을 하드웨어 플랫폼이 변화함에 따라 컴파일러를 통해 에뮬레이터를 생성함으로써 누적 에뮬레이션의 단점을 보완하는 기술임

❒ 여기서 에뮬레이터의 논리적 구조는 변화되지 않고, 에뮬레이터와 변환된 플랫폼 사이의 매개체로써 컴파일러를 이용함

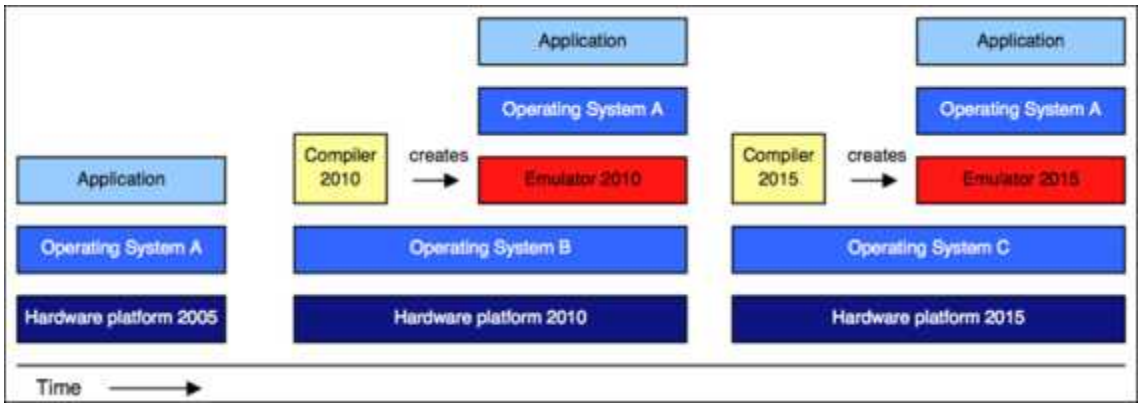
❒ 새로운 환경으로의 에뮬레이터 변환 과정에는 X언어로 쓰인 소스코드를 새로운 운영체제 플랫폼에서 실행될 수 있는 바이너리 코드로 전화해 줄 컴파일러가 필요함

❒ 이를 통해 최초의 에뮬레이터는 새로운 환경에 맞추어 지속적으로 변환될 수 있음

---

25) 국가기록원, 『전자기록물 장기보존을 위한 에뮬레이션 전략 연구』, 국가기록원, 2009, pp. 18~19.

26) Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), pp. 17~18.




<그림 3-6> 마이그레이션된 에뮬레이션

**출처:** Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), p. 17.

❒ 우선 에뮬레이터를 특정 환경에 맞추어 생성함

❒ 이 환경이 변화할 경우 에뮬레이터는 새로운 환경에 맞추어 변형함

❒ 이러한 변형은 새로운 플랫폼에서 실행될 수 있는 바이너리 형식의 언어로 소스코드를 변환하는 컴파일러를 통해 실행함

❒ 원본 에뮬레이터는 부분적으로 마이그레이션 되어 부가적인 환경을 생성함

❒ 소스코드는 미래에 생성되기 때문에 환경적 문제를 야기할 수 있으며, 컴파일러의 라이브러리는 원본과 양립하지 않을 수 있기 때문에 기능 고장이 있을 수 있음[27]

### (3) 에뮬레이션 가상머신(Emulation Virtual Machine)[28]

❒ 에뮬레이터는 시간 및 플랫폼으로부터 독립성을 확보하는 것이 가장 이상적임

❒ Jeff Rothenberg는 호스트 플랫폼과 에뮬레이터 사이에 추가적인 레이어를 도입함으로써 이러한 독립성을 확보하는, 소위 에뮬레이션 가상머신(Emulation Virtual Machine: 이하 EVM으로 약칭)이라 칭하는 방식을 개발함[29]

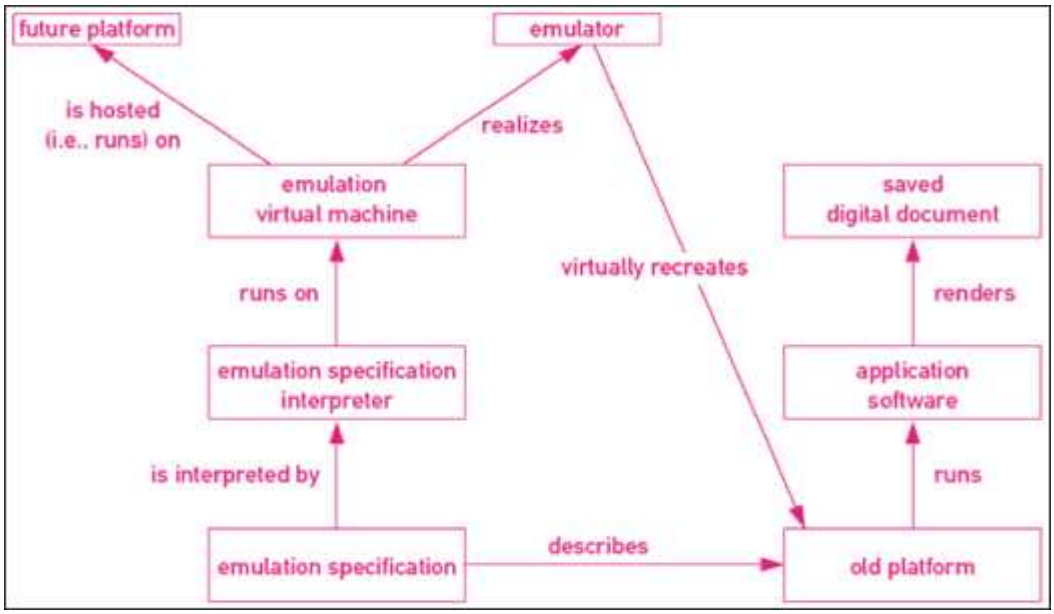
---

27) 국가기록원, 『전자기록 관리와 보존을 위한 국제협력 아젠다 개발』, 국가기록원, 2007, pp. 86~87

28) Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), pp. 18~19.

29) Jeff Rothenberg, "An Experiment in Using Emulation to Preserve Digital Publications", Koninklijke Bibliotheek, The Hague, 2000., http://www.studioautomata.com/itp/indestudy/emulationpreservationreport.pdf

❒ 아래의 그림과 같이 에뮬레이션 규정(Emulation Specification) 및 에뮬레이션 해석기(Emulation Interpreter)와 함께, 에뮬레이터는 EVM 상에서 가동될 수 있도록 생성될 수 있음




<그림 3-7> 에뮬레이션 가상머신 법칙

출처: Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), p. 18.

❒ 에뮬레이션 규정은 원래의 소프트웨어가 구 컴퓨터 플랫폼에서 가동되는지를 설명해 줌

❒ 그리고 이러한 에뮬레이션 규정은 구 플랫폼을 위한 에뮬레이터를 생성하는 에뮬레이션 해석기를 통해 해석됨

❒ 이를 통해 에뮬레이션 해석기와 생성된 에뮬레이터 양자는 EVM 상에서 가동되는데, 이는 곧 미래의 플랫폼에서 가동됨을 의미

❒ EVM은 시간의 흐름과 관계없이 안정적으로 운용되며, 아울러 다양한 호스트 플랫폼에서 가동될 수 있음

❒ 이러한 방식을 통해 에뮬레이터는 특정 플랫폼에 더 이상 의존적이지 않게 됨

❒ 이와 더불어 동일 EVM 상에서 다수의 에뮬레이터를 가동시킬 수 있게 됨

❒ 이러한 EVM 방식은 플랫폼 및 시간으로부터 독립성을 확보할 수 있지만, 다음과 같은 두 가지 단점을 지님

○ 먼저 구 플랫폼은 단일의 규정 내에서 전체적으로 정의될 수 있는지, 게다가 복합적인 해석기는 이러한 규정을 에뮬레이터로 변환하는데 필요한 기능을 충분히 구한하고 있는지의 문제가 아직 해결되지 못함

○ 또한 EVM은 시간의 흐름 속에서도 반드시 유지되어야 하는데, 이것 역시 매우 복잡한 문제를 지님

### (4) UVC-based emulation[30]

❒ UVC는 미래의 어떠한 시점에서도 디지털 객체의 원본 재현성을 항구적으로 제공해 줄 수 있는 보존방식임

❒ 이 방식은 에뮬레이션과 마이그레이션의 결합에 기초. 즉 마이그레이션은 원본 객체를 플랫폼 독립적인 포맷으로 변환할 때 수행되며, 에뮬레이션은 플랫폼에 의존하지 않게 하는 절차에 활용됨

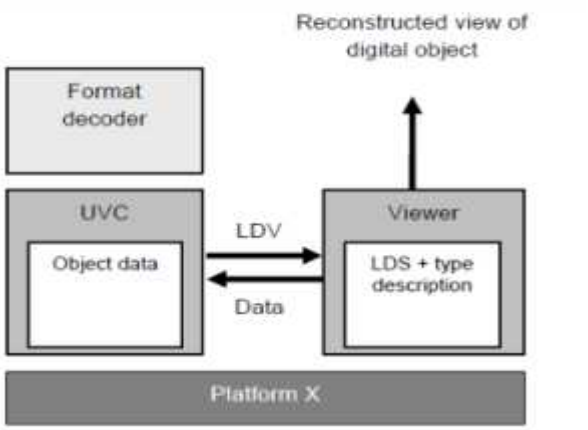
❒ UVC는 일반 목적의 컴퓨터를 재생산함과 아울러 현재 및 미래의 컴퓨터 플랫폼에서 쉽게 실행될 수 있는 소프트웨어 프로그램임

❒ 이러한 방식으로 UVC 언어로 작성된 프로그램은 플랫폼 독립적인 방식으로 UVC에서 실행됨

❒ UVC 프로그램(디코더)은 원본 객체의 포맷구조를 판독하고, 이러한 디코딩 과정은 Logical Data View(LDV)라고 불리는 XML과 유사한 구조를 제공. 아울러 Logical Data Schema(LDS)는 LDV에서 발생되는 요소들을 정의내리고 이 요소들이 의미하는 것에 대한 기타 정보들을 제공하며, 뷰어는 LDV 요소와 LDS 기술에 기초하여 원본 개체의 외형을 재구축함

❒ 이러한 절차는 아래의 그림과 같이 도식화할 수 있음

30) Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), pp. 19~20.




<그림 3-8> UVC 기반 에뮬레이션 방식

출처: Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), p. 19.

❒ 이러한 전략을 기반으로, 미래의 이용자들은 항상 원본 객체를 접근해 열람할 수 있지만, 현재 및 미래에 몇 가지 절차가 수행되어야 함

❒ 우선 현재 시점에서는, 어떻게 UVC가 구축되어야 하는지를 정의하는, UVC 규정(specification)이 보존되어야 함

❒ 또한 각각의 특정 파일포맷에 대한 디코더가 각기 개발되어야 함

❒ 또한 LDS는 디지털 객체 각각에 필요하며, 원본 객체 또한 보존되어야 함

❒ 검색 시점에서는, 미래의 프로그래머는 UVC 규정에 기반을 둔 UVC를 재개발해야 함

❒ 그리고 원본 객체의 재현으로 LDV를 번역해 줄, 하나 내지 그 이상의 뷰어가 생성되어야 함

❒ 검색 및 보존 시점에서 요구되는 행위들은 이 전략의 장단점을 드러내 줌. 즉 극단적으로 광범위한 파일 포맷 및 객체 유형을 감안할 때, 만일 UVC 전략이 디지털 객체의 보존전략으로 채택된다면 수많은 포맷 디코더 및 LDS 기술이 개발될 필요가 있음

❒ UVC의 성패는 어느 정도 범위까지 소프트웨어 및 컴퓨터 산업체들에 의해 수용되는지 여부와 함께, 포맷 디코더 및 LDS 기술 활용도에 달려 있음

❒ 하지만 일단 단일 LDS 및 디코더가 개발되면, UVC의 강점은 명백해짐. 즉 각각의 디지털 객체에 대해 마이그레이션을 수행하는 것에 비해, UVC 전략은 동일 포맷의 모든 객체들의 필요를 충족시킬 수 있음

❒ 아울러 에뮬레이션 전략은 각각의 하드웨어 세대에 대해 새로운 에뮬레이터를 개발해야 하지만, UVC는 가능한 한 단순하게 설계됨으로, 이러한 문제는 상대적으로 쉽게 해결될 수 있음

### 다. 개념적 모델: 모듈 에뮬레이션(Modular Emulation)

#### (1) 모듈 에뮬레이션 개요[31)]

❒ 현용 에뮬레이터의 평가 외, 보존맥락에서 전체 에뮬레이션(하드웨어 에뮬레이션)의 실현을 위한 새로운 접근 방법이 개발됨

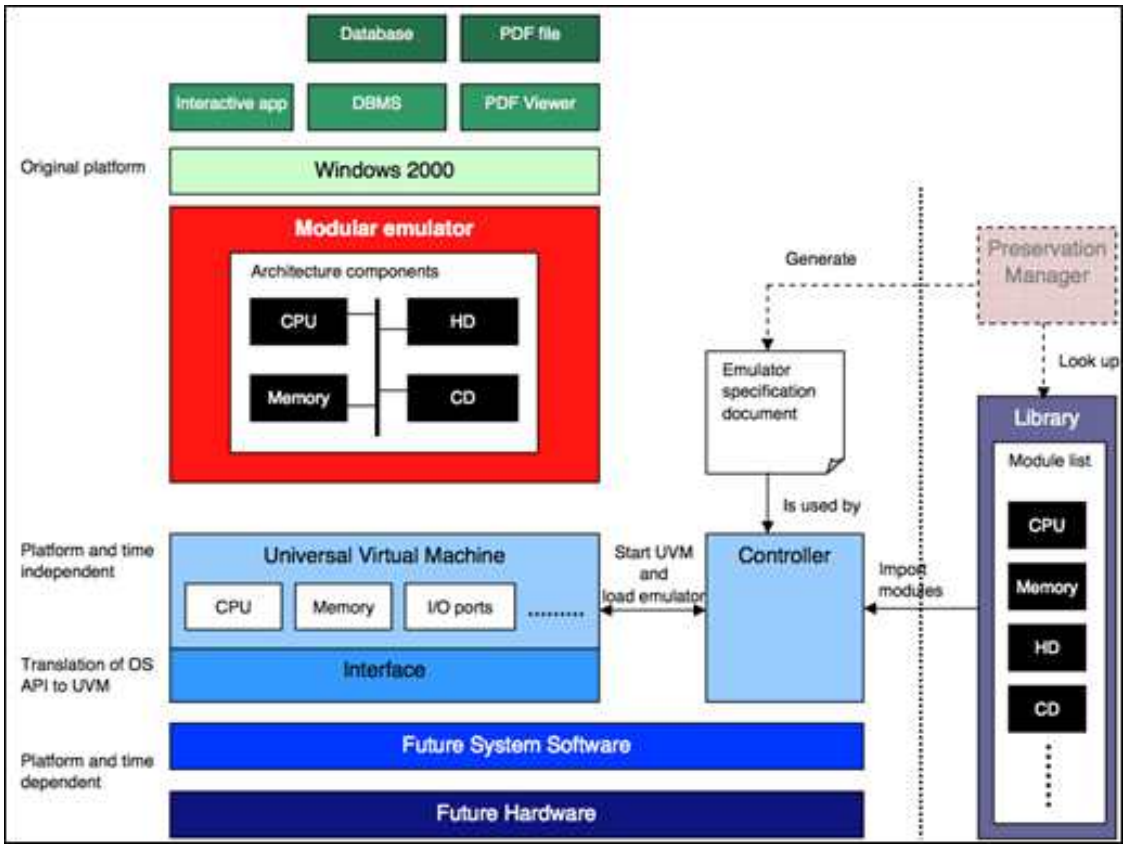
❒ 오랜 기간 동안 디지털 방식으로 모든 종류의 오브젝트에 대한 접근을 유지해야 하는 이 방법은 개념적 모델과 초기 의견결정 및 다른 방법에 대한 검증 역할을 하도록 구성됨

❒ 이 모델은 아래의 그림에 제시된 대로, 에뮬레이션 모듈식 접근 방식을 보여주는 관계상, 모듈 에뮬레이션 모델(modular emulation model)이라 칭함

❒ 모듈 에뮬레이션의 정의 : 개별 에뮬레이터와 에뮬레이션 전 과정 등을 형성하는 등의 하드웨어 아키텍처의 구성요소를 에뮬레이션한 전체 에뮬레이션임

❒ 각 에뮬레이터별로 서로 다른 모듈의 유연한 구성을 만들 수 있는 가능성을 제공하는 에뮬레이션 프로세스로 각기 별개의 모듈을 형성

31) Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), pp. 21~22.




<그림 3-9> 모듈에뮬레이션 도해

출처: Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), p. 2.

❒ 개념적 모델은 다음과 같은 부분으로 구성됨

- ○ 범용 가상 머신(Universal Virtual Machine)
- ○ 모듈 에뮬레이터(Modular emulator)
- ○ 구성요소 라이브러리(Component Library)
- ○ 에뮬레이터 규격문서 (Emulator specification document)
- ○ 제어장치(Controller)
- ○ 보존 관리자(옵션)[Preservation Manager (optional)]

### (2) 모듈 에뮬레이션의 구성요소[32]

❒ 범용 가상 머신(Universal Virtual Machine)

- ○ 범용 가상 머신(Universal Virtual Machine : UVM)은 모든 에뮬레이션의 기초임
- ○ UVM은 기본 미래 호스트 플랫폼의 상단의 독립적 계층으로 형성됨

32) Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), pp. 22~25.

○ 이는 UVM 프로세서와 메모리의 지원뿐만 아니라 UVC의 입력 호스트와 대상 플랫폼 사이의 주변장치와의 입출력(I/O)을 위한 추가 기능을 제공하는 고급버전임

❒ 모듈 에뮬레이터(Modular emulator)

○ UVM에 원본 소프트웨어 (OS 및 응용 프로그램)를 실행하려면 에뮬레이터가 필요함

○ 에뮬레이터는 모듈 에뮬레이터에서 모듈식 구조로 설계되어야 함. 원래 하드웨어 플랫폼의 구성요소는 이 구조의 모듈로 표시됨

○ 각 모듈은 UVM에 소프트웨어에 특정 하드웨어 구성 요소를 에뮬레이션 함

○ 대상 플랫폼의 재현은 다음 하드웨어 아키텍처에 따라 모든 필요한 모듈을 결합하여 실현될 수 있음

❒ 구성요소 라이브러리(Component Library)

○ 모듈 에뮬레이터를 구축하기 위해서는 각기 다른 모듈이 필요함

○ 라이브러리는 모든 에뮬레이터에서 사용할 수 있는 모듈을 포함하며 가능한 모듈을 추적하고 구성요소의 유형을 조직화해야함

○ 각 모듈은 최대한 실제 하드웨어 구성 요소의 기능 동작을 재현해야 함

○ 라이브러리는 적어도 하나 이상의 대상 플랫폼을 만들 모듈의 최소 집합을 제공해야 하지만 시간이 지남에 따라 더 많은 모듈을 확장 할 수 있음

❒ 제어 장치

○ 컨트롤러는 호스트 플랫폼에 UVM을 시작하고 호스트 플랫폼 및 UVM의 입출력(I/O)을 처리하는 가교의 역할을 수행함

○ 컨트롤러의 두 번째 역할은 모듈 에뮬레이터에서 에뮬레이션 규격 문서 및 모듈을 부여함

❒ 보존 관리자(옵션)[Preservation Manager (optional)]

○ 이론적으로, 에뮬레이션 규격 문서가 자동으로 보존 관리자(PM)에 의해 생성될 수 있음

### (3) 모듈 에뮬레이션의 실행 단계[33)]

❒ 실행에 앞서 개념적 모델의 몇 가지 단계가 필요하며, 가장 좋은 방법은 보존시점 및 검색시점에 수행할 단계를 구별하는 것임

33) Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), p. 26.

❒ 보존시간

- ○ 단계1 : 명확한 규격의 UVM 플랫폼을 정의함
- ○ 단계2 : 적어도 하나의 원래 플랫폼을 나타내며 모든 필요한 모듈로 구성된 구성요소 라이브러리를 구축함
- ○ 단계3 : 구성요소 라이브러리에서 만든 모듈을 사용하여 UVM에서 실행 될 모듈 에뮬레이터 개발함
- ○ 단계4 : 모듈 라이브러리 에뮬레이터의 규격 및 UVM의 규격과 함께 에뮬레이션에서 실행되는 기존의 디지털 객체를 보존함
- ○ 단계5 : 모듈 라이브러리와 협력하고 에뮬레이터 규격 문서의 생성을 자동화하는 보존 관리자에 의해 조정함

❒ 검색시간

- ○ 단계1 : UVM은 UVM을 준수하는 규격 문서를 개발을 통해 당시 UVM과 플랫폼 사이의 입출력(I/O) 통신을 위한 메커니즘을 구현함
- ○ 단계2 : UVM을 통제하고 UVM 사이에서 정보를 교환하는 방법을 제공하는 프로그램을 개발함
- ○ 단계3 : 대상 플랫폼을 설명하는 에뮬레이터 규격 문서를 사용하거나 제작함
- ○ 단계4 : 디지털 아카이브에서 필요한 모든 디지털 객체(또는 이미지 파일)를 검색하고 에뮬레이션 프로세스를 시작함

❒ 제어장치는 다음과 같은 작업을 수행 함 :

- ○ UVM 지원 설비(CPU, 메모리, 스토리지, 네트워크 등)의 인식
- ○ 에뮬레이션 규격 문서를 해석
- ○ 사용 가능하거나 상호지원을 통해 호환가능 한 모듈을 결정
- ○ 라이브러리에서 선택한 모듈을 검색
- ○ 모듈 에뮬레이터를 구축하기 위해 연결
- ○ 객체의 스트림(이미지 파일이나 전자 문서를 즉)을 로드
- ○ 재구성할 수 있는 디지털 객체의 모듈 에뮬레이터를 시작

### (4) 모듈 에뮬레이션의 적용

❒ KB 프로젝트의 첫 번째 목적은 에뮬레이터를 구축하는 것이지만, 에뮬레이터가 미래의 환경에서도 사용할 수 있다는 것을 보장하기 위한 방법을 고려하는 것임

❒ 환경은 원래 하드웨어 플랫폼뿐만 아니라 시스템 소프트웨어(드라이버 구성 등), 응용프로그램과 플러그인을 포함함

❒ 이 부분의 저장장치들은 다음 기준을 만족하는 방식으로 이루어져야 함
  ○ 생성 가능성, 수정(modify)과 환경을 에뮬레이트
  ○ ‘법률(de jure)’을 준수하고, ‘사실(facto)’에 기반
  ○ 디지털 객체를 재구성하는데 유용 : 대화형(interactive) 멀티미디어 응용프로그램, 데이터베이스 시스템 및 PDF 문서

❒ 에뮬레이터에 필요한 정보(시스템 소프트웨어, 응용프로그램 및 데이터)는 일반적으로 디스크 이미지로 제공함

❒ OS 및 응용 프로그램 또는 삽입된 광디스크 즉, 실제 CD 또는 DVD 드라이브를 모방한 것으로 CD-/DVD-ROM 디스크 이미지가 포함된 하드 디스크 이미지임

❒ 디스크 이미지는 실제 드라이브에 대용량 파일로 저장되며 디스크 이미지 볼륨 크기를 수동으로 정의할 수 있음

❒ 다양한 디스크 이미지 포맷은 raw 또는 압축 이미지(compressed images)로 서로 다른 접근방식을 기반으로 존재함
  ○ RAW 이미지 : 실제 드라이브의 내용과 레이아웃과 매우 유사하며, 또한 같은 크기(40 기가바이트 하드 디스크를 40기가바이트의 디스크 이미지로 생성)
  ○ 압축된 이미지(compressed images) : 훨씬 더 작은 디스크 이미지의 이점을 제공. 하지만 특정 압축 기술에 의존하고 형식이 더 복잡함

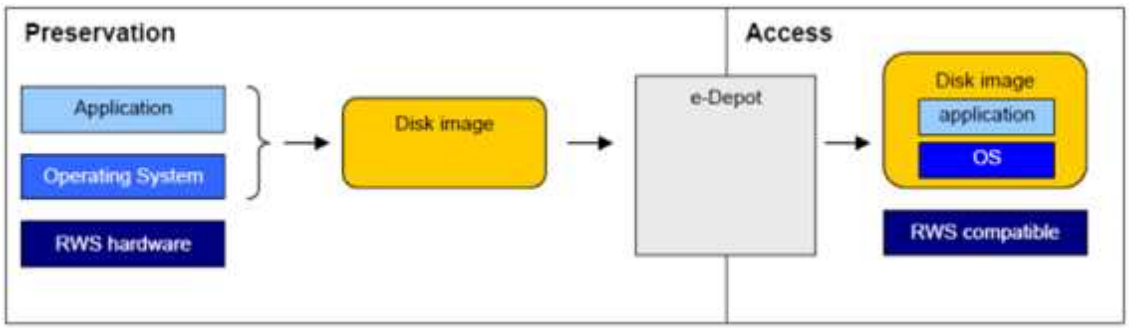
❒ 대화형 멀티미디어 응용 프로그램[34)]
  ○ 대화형 멀티미디어 응용 프로그램은 제작사가 CD-ROM 또는 DVD-ROM으로 제공함
  ○ 데이터베이스 시스템이 PDF 문서와 비교하여 갖는 차이점은 데이터가 응용 프로그램과 통합된다는 것임
  ○ KB의 각 대화형 멀티미디어 응용 프로그램은 RWS(Reference Work Station)[35)]에 설치되어 Microsoft Windows 2000 운영체제 (OS) 및 일부 추가 소프트웨어를 새로 설치·실행함
  ○ 대화형 멀티미디어 응용 프로그램, 드라이버 및 구성이 설치되는 하드 디스크의 내용이 압축 된 디스크 이미지를 만드는 프로그램은 Power Quest(현재 Symantec 소유하고 드라이브 이미지라고도 함)를 사용하여 수행함

---

34) Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), pp. 27~29.

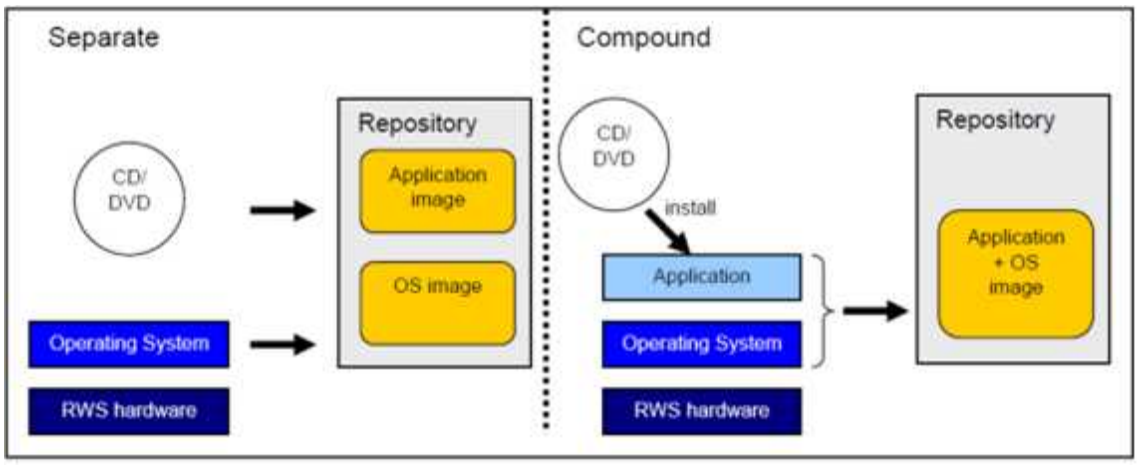
35) 표준화된 컴퓨터 플랫폼을 정의하는 것으로 KB에서는 RWS(Reference Work Station)로 표기한다.

○ 디스크 이미지는 e-Depot에 포함됨
○ 원래 응용 프로그램은 e-Depot(배포에 의해)에서 디스크 이미지를 가져오고 RWS 또는 호환 컴퓨터 시스템의 빈 하드 디스크에 로드하여 액세스함으로써 OS 및 응용 프로그램을 실행함




<그림 3-10> 멀티미디어 응용 프로그램에 대한 보존 및 접근 도해

○ 이 방법은 잘 작동 하지만 응용프로그램 및 시스템 소프트웨어에서는 응용프로그램을 분리 불가능한 디스크 이미지로 통합됨
○ 분리(Separate)는 OS가 오래된 경우에도 에뮬레이션에서 유용하며, 응용 프로그램은 다른 OS에서 다시 설치할 수 있음
○ 이는 OS와 응용 프로그램 사이에 빠른 구성으로 만들 수 있는 가능성을 제공하는데, 아래의 그림은 디스크 이미지 저장을 위한 두 가지 선택을 보여줌




<그림 3-11> 대화형 멀티미디어 응용 프로그램 보존 도해

○ 두 번째 장점은 디지털 객체를 저장하는데 필요한 저장 공간이 크게 감소.
○ 특정 OS를 저장하면 한 번에 여러 응용프로그램을 보존해야할 경우에는 특히 더 나은 효율을 제공

❒ 데이터베이스 시스템[36]

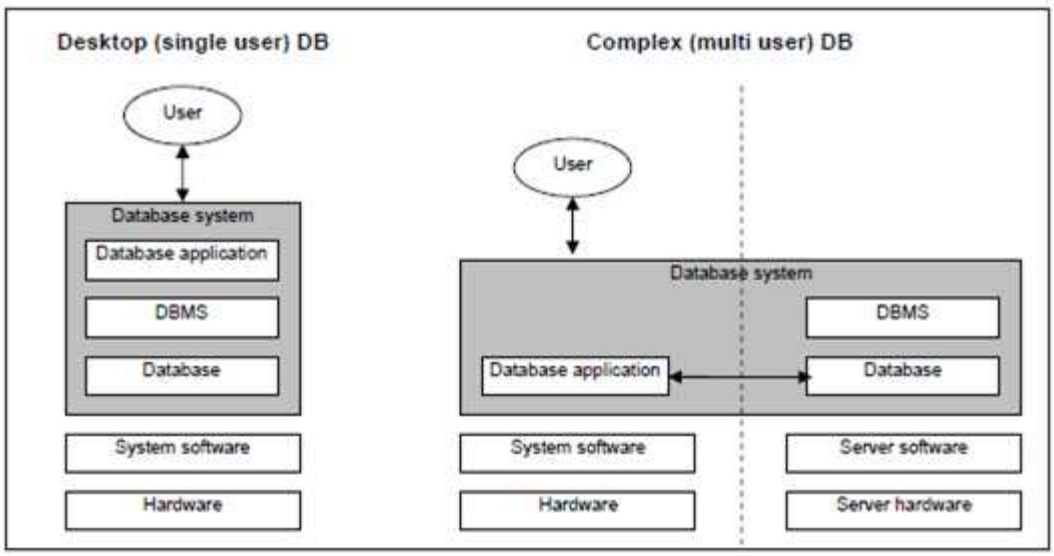
○ 데이터베이스 시스템 스토리지(DBS)는 대화형 멀티미디어 응용 프로그램과는 다른 접근이 필요함

○ OS와 DBS의 분리는 동일한 소프트웨어를 사용하여 다른 플랫폼을 결합하여 유연성을 제공하지만, DBS 설치 및 설정은 상당한 노력과 지식이 필요함

○ 데이터베이스 시스템은 주로 세 가지 구성 요소로 구성됨

- Database: 실제 콘텐츠 데이터베이스 테이블의 튜플 또는 테이블
- DBMS: 데이터베이스 관리 시스템은 데이터베이스를 관리하고 사용자가 요청한 데이터에 대한 작업을 수행하도록 설계된 컴퓨터 프로그램
- Database 응용프로그램: 특정 응용 프로그램과 문제 세트(Problem set)의 데이터를 관리하기 위해 작성된 컴퓨터 소프트웨어

○ 이러한 구성요소의 배치는 다양한 데이터베이스 시스템에서 다를 수 있으나 크게, 데스크톱(단일사용자)의 데이터베이스 시스템과 복합(다중 사용자)의 데이터베이스 시스템으로 구분됨



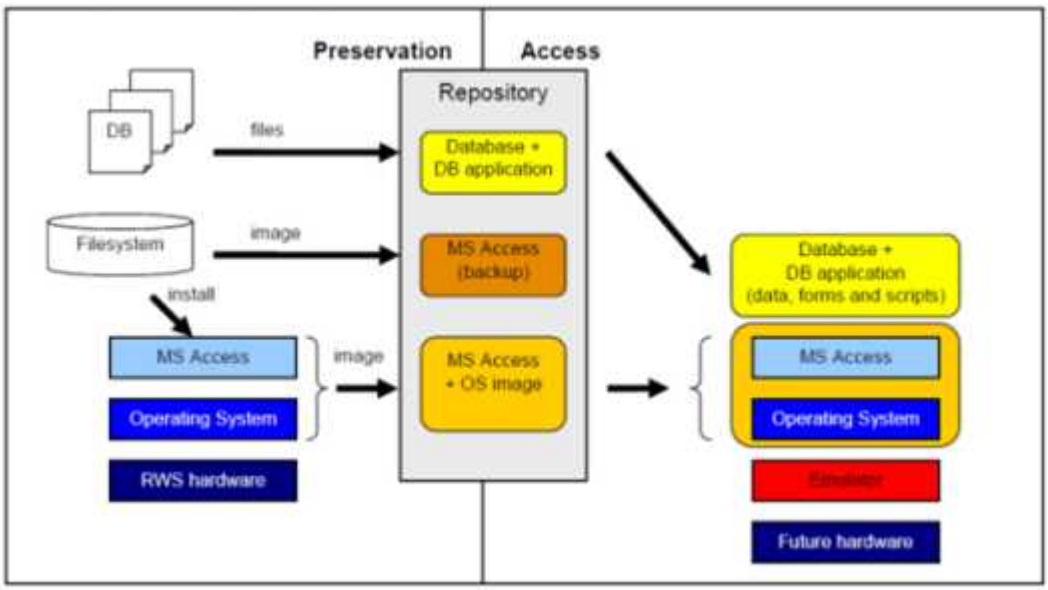

**<그림 3-12> 데스크톱 및 복합 데이터베이스의 데이터 아키텍처**

- 데스크톱(단일사용자) 데이터베이스 시스템은 종종 다음과 같은 특성을 보임
  - DBMS는 데이터베이스 자체와 응용 프로그램에 대한 완벽한 솔루션으로 사용 가능
  - 데이터베이스는 한 번에 하나의 사용자에 의해 사용
  - 복잡한 데이터베이스 시스템에 비해 사용하기 용이

36) Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), pp. 29~31.

- 사용자는 다른 DBMS의 형식 또는 동일한 DBMS의 새로운 버전으로 그들의 DBMS 마이그레이션 할 수 있음
- 데스크톱 DBMS의 예로 MS Access와 dBaseIV가 있음

• 복합(다중 사용자) 데이터베이스 시스템은 다음과 같은 특성을 보임
- 데이터베이스와 데이터베이스 응용프로그램이 분리
- 데이터베이스는 여러 사용자가 함께 사용
- 이용자가 설계, 시공 그리고 유지 및 보수의 전문기술이 필요
- 다른 데이터베이스 형식으로 데이터베이스의 내용을 변환할 수 있으며, 같은 DBMS의 새 버전으로 마이그레이션 할 수 있음
- 복합 데이터베이스 시스템의 예로는 Oracle 제품, Microsoft SQL Server와 DB2가 있음
- 데스크톱 데이터베이스 응용프로그램에서 만든 DBMS는 데이터베이스 내용뿐만 아니라, 스크립트와 양식을 포함하는 하나의 파일(MS는 mlb 포맷)으로 구성
- 이러한 파일은 모든 데이터베이스 범용 프로그램인 MS Access를 사용하였으며, 이는 데이터베이스 파일을 별도로 저장 할 수 있지만 MS Access는 OS에 미리 설치되어 있을 수 있음을 의미
- 아래의 그림에 제시된 대로, 모든 데이터베이스 파일을 재사용 할 수 있음
- 안전상의 이유로 데이터베이스 프로그램의 설치 파일을 포함하는 파일 시스템의 디스크 이미지는 보존되어야 함

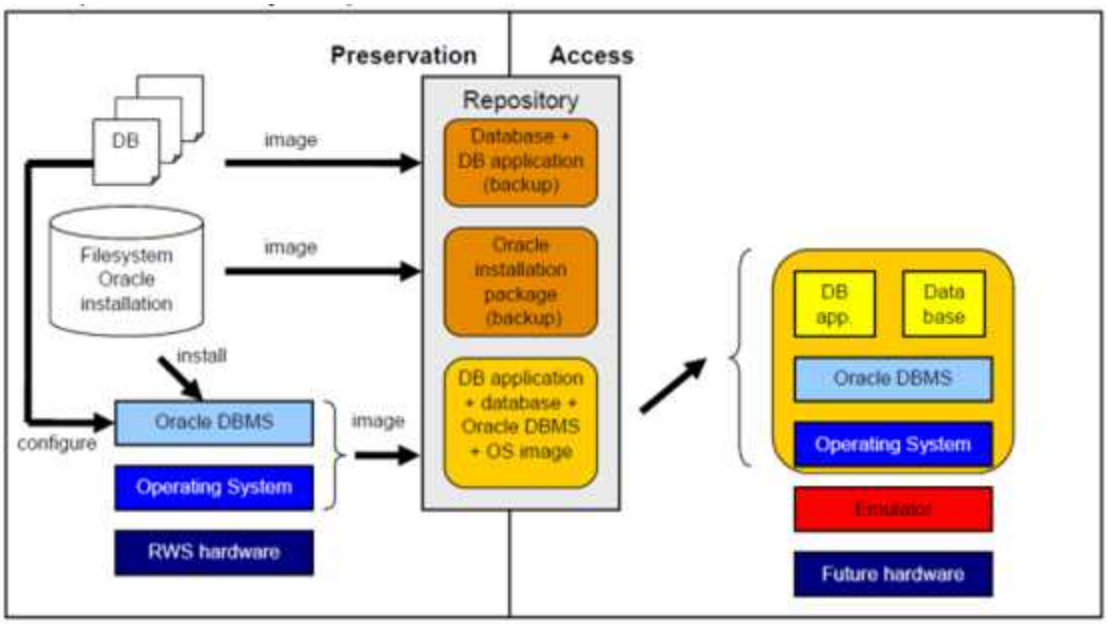



**<그림 3-13> 데스크톱 데이터베이스시스템의 보존 사례 : MS 엑세스**

- 복잡한 데이터베이스 시스템의 경우 DBMS와 데이터베이스에서 데이터베이

스 응용 프로그램의 분리가 가능하며, 데이터베이스 응용 프로그램을 동일하게 유지하면서 변경·변환하거나 기본 DBMS와 데이터베이스를 대체 할 수 있음

- 이는 시스템의 유연성을 제공하나, 각각의 복잡한 데이터베이스는 사용자 정의에 의해 만들어진 것임
- 하나의 데이터베이스에서 작동하도록 설계된 응용프로그램은 다른 데이터베이스에서 사용할 수 없음
- 따라서 오라클로 만든 복잡한 데이터베이스 스토리지는 전체적으로 수행해야 함
- 아래의 그림처럼, OS, DBMS, 데이터베이스 및 데이터베이스 응용프로그램을 하나의 디스크 이미지로 보존

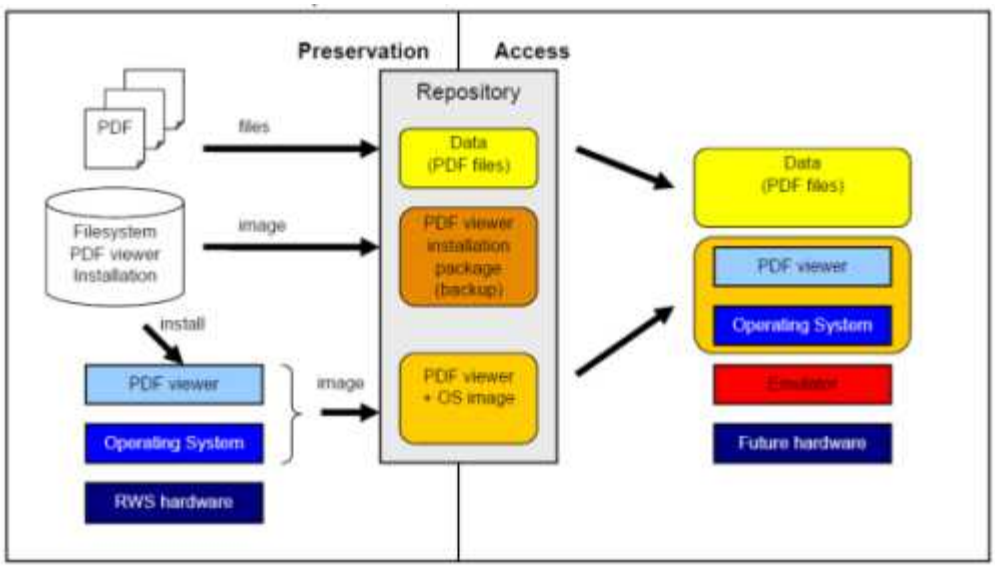



<그림 3-14> 복합 데이터베이스 시스템의 보존 사례 : DB

- 클라이언트 서버 아키텍처는 하나의 컴퓨터 환경(로컬에서 실행)에서 실현되어야 한다는 것을 지적해야 함

❒ PDF 문서[37]

37) Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline Slats, *Emulation: A Viable Preservation Strategy*, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005. 6. 20(Version 1.4), pp. 31~32.




<그림 3-15> PDF 문서 및 뷰어의 보존 도해

○ 데이터베이스와 마찬가지로 보기(view) 위해서는 PDF 문서 응용프로그램이 필요함
○ PDF 뷰어와 같은 응용프로그램은 OS에 사전에 미리 됨
○ OS에서 뷰어를 분리하면 큰 저장 공간을 차지하지 않는 장점을 지님
○ PDF 문서를 볼 때마다 뷰어를 설치해야하는 단점을 지님

### 3.1.4 Planets 프로젝트

#### 가. 개요

<표 3-2> Planets 프로젝트 개요

| 구분 | 내용 |
|---|---|
| 본 명칭 | Preservation and Long-term Access through NETworked Services |
| 진행기간 | 2006. 6. 1 - 2010. 5. 31 (48개월) |
| 주관기관 | The British Library (16개 기관) |
| ICT 세부 영역 | Access to and preservation of Cultural and Scientific resources. |
| 예산 | 13,966,318 EURO (206억1,433만) |
| 공식사이트 | http://www.Planets-project.eu/ |

### (1) 배경과 목표

❒ Planets(Preservation and Long-term Access through NETworked Services)은 유럽연합의 여섯 번째 프레임워크 프로그램(FP 6-ICT)의 지원을 받아 시작된 디지털 보존 프로젝트

❒ 2006년 영국 국립도서관(The British Library)을 주관으로 16개 기관[38]이 참여하고 소요된 예산 금액만도 1,300만 유로에 달하는 대규모의 프로젝트로서 48개월 후인 2010년에 종료되었음[39]

❒ 프로젝트의 목적은 디지털 형태의 문화 및 과학 자산들에 대한 장기적인 보존과 접근을 돕기 위한 실제적인 서비스와 도구를 만드는 것임. 아울러 디지털 객체를 관리하기 위한 일련의 도구들을 개발하고 시험하는데 사용할 수 있는 프레임워크와 인프라를 제공한다는 지향점을 공유함

❒ Planets은 문화 및 과학에 관한 지식들을 안전하게 보존하고 접근시켜야 하는 책무를 가진 국립 도서관과 아카이브즈들이 주도하였음. 그들은 근본적이고 장기적 전망에서 기관의 보존계획을 수립 운영해야 한다고 보았고 이를 위한 이론적, 실제적 측면에서의 연구가 필요하다는 것이 이 프로젝트의 주요한 배경이 되었음

❒ 특히, Planets은 종합적인 보존 계획(preservation planning)[40]의 수립 필요성에 주목, 보존 대상의 특성이나 상황에 맞게 현존하는 보존 도구들(마이그레이션 혹은 에뮬레이션 도구를 포함하여)을 포괄적으로 검토해 보존 전략을 수립, 운영해 나갈 필요성을 인식하였음

❒ Planets의 결과물들은 프로젝트 종료 이후 “Open Planets Foundation(OPF)”를 통해 유지운영 되고 있음. 이 조직은 영국에 거점을 둔 비영리기관임 (http://www.openPlanetsfoundation.org)

### (2) 세부 프로젝트 및 서비스

---

38) 유럽의 국립도서관과 아카이브, 대학 등 연구기관, 주요 IT 업체(IBM, Microsoft 등)가 참여하고 있다. The British Library, The National Library of the Netherlands, The Austrian National Library, The Royal Library of Denmark, The State & University Library, Denmark, The Dutch National Archives, The UK National Archives, The Swiss Federal Archives (BAR), University of Cologne, University of Freiburg, Humanities Advanced Technology and Information Institute (HATII), Vienna University of Technology, The Austrian Institute of Technology (AIT), IBM Netherlands, Microsoft Research Limited, Tessella plc http://www.planets-project.eu/about/

39) Planets의 결과물들은 프로젝트 종료 이후 영국에 거점을 둔 비영리기관인 “Open Planets Foundation(OPF)”를 통해 유지운영되고 있다 http://www.openPlanetsfoundation.org

40) 보존계획의 개념을 조직의 요구사항과 컬렉션의 특성에 맞춰서 적절한 보존 전략이 선별되는 과정으로 보고 있음

❒ 이 프로젝트는 여섯 개의 세부 프로젝트로 구분되어 진행되었고 이는 디지털 보존을 위한 통합 서비스(integrated services)를 제공하고자 하였던 목표에 근거한 것임

❒ 6개의 세부 프로젝트 각각은 보존계획(Preservation Planning), 보존 특성화(Preservation Characterization), 보존 행동(Preservation Action), 상호운용 프레임워크(Interoperability Framework), 테스트베드(Testbed), 보급(Dissemination & Takeup)로서 각각에 해당하는 서비스가 개발됨[41)]

○ 보존계획(Preservation Planning)은 조직이 보존 계획을 정의, 평가, 실행할 수 있게 계획하는 서비스를 개발하는 것을 목표로 함

- 조직의 디지털 보존 요구사항과 목적을 조직의 정책, 전략, 이용 매뉴얼 등을 활용하여 정의하게 하고 디지털 보존 목적에 적합한 보존 계획을 수립할 수 있게 해주는 것이 주요 내용임
- 이를 위해서 'Plato'라 명명하는 보존계획 서비스를 고안하였음
- Planets은 잠재적인 요구사항과 보존계획에 관한 지식(knowledge base)을 토대로 여러 기관들이 신뢰할 만한 보존 계획을 수립하는 것을 돕고자 하였음

○ 보존 특성화(Preservation Characterization)는 디지털 객체의 특성화를 위한 방법론과 도구와 서비스를 개발하는 것을 목표로 함

- 디지털 객체의 속성과 관련된 도구들을 학습시켜 주고 디지털 객체의 중요 속성을 추출하고 컬렉션의 특성을 분석할 수 있게 도와주는 것이 주요 내용임
- 특성화를 위해서 XCL언어를 활용
- XCL언어는 특성을 서술(eXtensible Characterization Description Language: XCDL)하고 추출(eXtensible Characterization Extraction Language: XCEL)하는 언어로 구분되어 있음

○ 보존 행위(Preservation Action)는 디지털 객체의 장기접근을 보장하는 보존 행위를 위한 혁신적인 솔루션 개발을 목표로 함

- 디지털 객체에 대한 컨버트 서비스를 제공해주는 마이그레이션 서비스와 초기의 운영시스템에서 소프트웨어를 실행시킬 수 있게 해주는 에뮬레이션 서비스로 구분될 수 있음
- 특히, 에뮬레이션을 위해서 GRATE라는 통합 에뮬레이션 프레임워크를 제공

○ 상호운용 프레임워크(Interoperability Framework)는 서비스 네트워크상에서 틈

41) 즉, 프로젝트의 여섯 개 세부 목표는 곧 해당되는 서비스의 주요내용이기도 함. 여기서는 기술의 편의상, 세부 프로젝트를 중심으로 핵심적인 내용을 소개함. Planets Tools and Services for Digital Preservation.

새 없이 통합된 도구와 서비스를 위해 상호적으로 운영 가능한 프레임워크를 개발하는 것이 주요 목표임

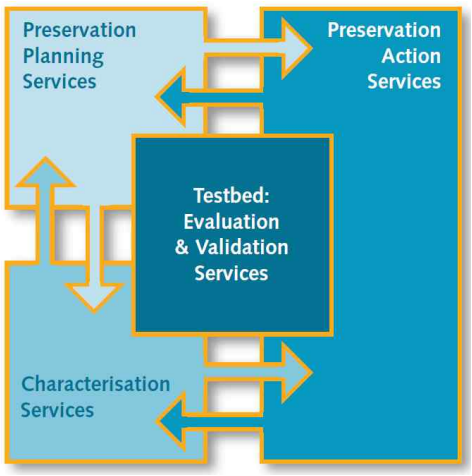
- 현재의 IT 시스템과 Planets이 통합할 수 있게 함으로써 기존에 구축된 디지털 저장소를 활용할 수 있게 해줌
- 서로 다른 보존 과업을 조합 가능하게 하는 역할을 수행하는 것임

○ 테스트베드(Testbed)는 디지털 보존을 위한 협력적인 연구 환경을 제공하며, 실험과 평가를 위한 통제된 환경을 제공하여 서로 다른 프로토콜, 도구, 서비스 그리고 보존 계획에 관한 평가를 제공하려는 목적을 가짐

- 이용자는 테스트베드를 통해서 특성화, 마이그레이션, 에뮬레이션 등을 통제된 환경에서 샘플 등의 내용물을 가지고 직접 실험

○ 보급(Dissemination & Takeup)은 효율적인 사용자 훈련을 위한 종합적 보급 프로그램을 개발하는 것을 목표로 함

- 사용자는 보존 감시 서비스를 통해 보존과 접근 목적을 위협하는 외부적인 변화를 식별할 수 있음.
- 이상의 여섯 가지 세부 과제를 토대로 개발된 Planets의 전체 서비스는 다음의 그림과 같음.




<그림3-16> Planets 서비스 개요

출처. Planet Brochure, 2009, p. 7.

### (3) Testbed[42]

❒ 디지털 보존을 위한 협력적인 연구 환경을 제공하며, 실험과 평가를 위한 통제된 환경을 제공함

42) http://testbed.Planets-project.eu/testbed/.

❒ 다양한 디지털 객체 타입에 관하여, 사전에 정의되고 잘 설명된 내용물을 갖고서, 보존 툴을 비교할 수 있음(체계적으로 실험을 실행할 수 있는 방법론, 부분적으로는 실험 결과를 자동화하여 평가 가능하고, 실험 그 자체에 대한 공유된 접근을 제공하고, 실험을 재생산가능하며, 구조화된 실험 문서화 능력이 있음, 실험 결과를 비교 가능함)

❒ 플랫폼 독립적임(자바 엔터프라이즈 에디션)

❒ 다양한 보존 도구와 서비스 중에서 원하는 실험을 실행할 수 있도록 함(웹 서비스 접근)

❒ Planets software suite(전체 프로젝트를 관통하는 공통적인 요소를 공유하고 있음) 의 일부임

❒ 세 개의 주요 범주로 나뉘며 특성분석 도구, 마이그레이션 도구, 에뮬레이션 도구가 그것임

❒ 실험 과정은 여섯 개의 단계로 구성되어 있음(실험 설계> 실험 실행> 실험 평가의 순으로 진행

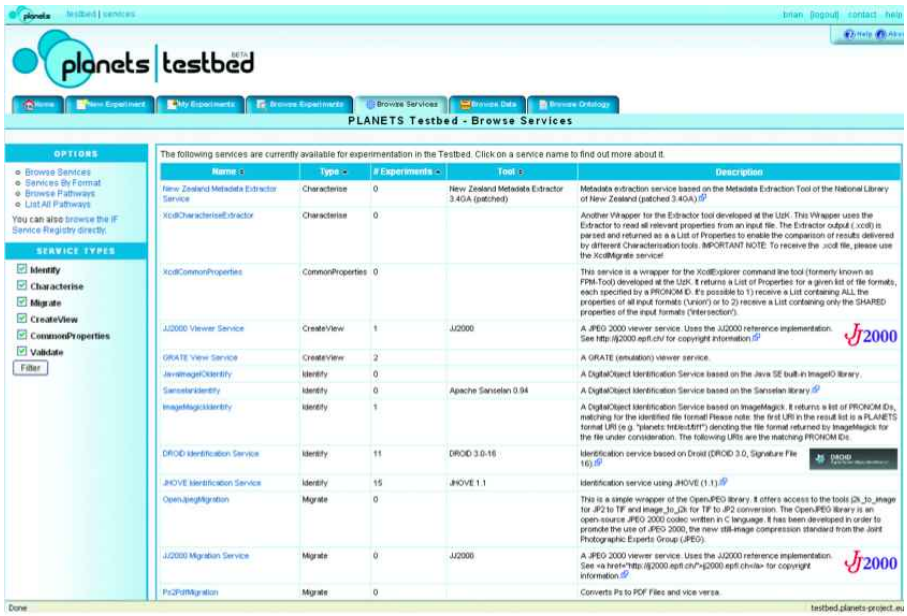


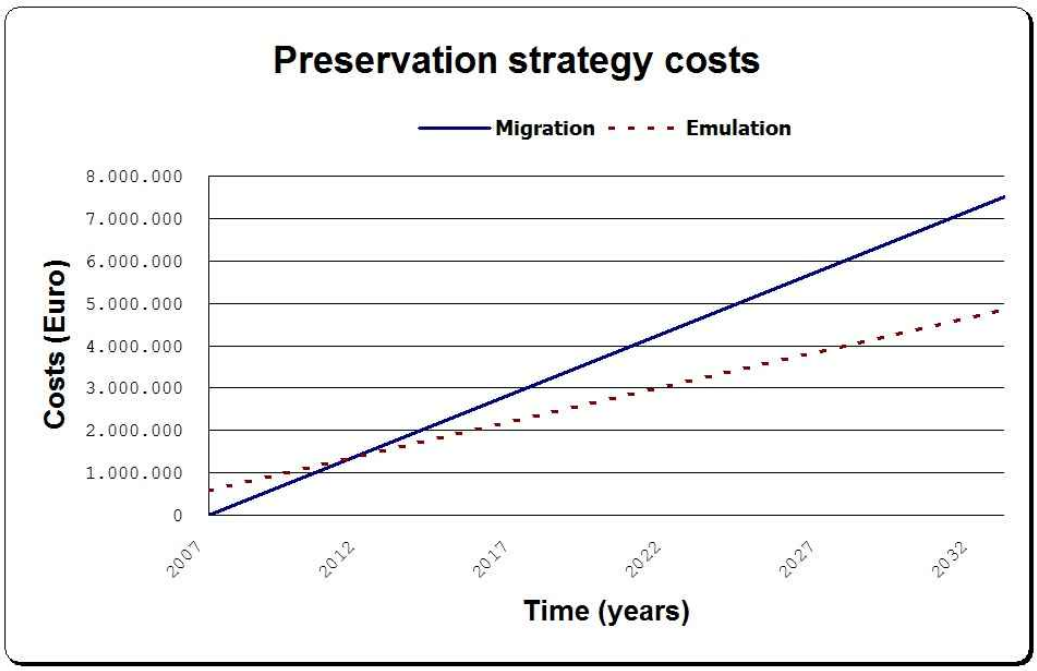

*Figure 1: The testbed interface, listing services.*

<그림 3-17> 테스트베드 인터페이스

출처. The website of the Planets Testbed tool, https://testbed.Planets-project.eu/testbed/

### (4) 비용

❒ Planets 프로젝트의 결과물을 유지 운영하고 있는 Open Planets Foundation에서 제공하고 있는 에뮬레이션에 관한 정보 서비스에서는 에뮬레이션 비용에 관하여 다음과 같은 의견을 제시하고 있음[43)]




<그림 3-18> 장기보존 전략별 비용 비교

❒ 보존 전략으로서 에뮬레이션을 적용하면 마이그레이션 보다 객체당 소요되는 비용이 높을 것이라 예상하겠지만 그것은 일반적인 오해일 수 있음

○ 에뮬레이터는 대개 한 번에 여러 형식을 지원 가능하며 앞으로도 그런 방향으로 진행될 것임. 또한 에뮬레이터의 개발은 목표로 하는 플랫폼별로 한번만 필요함

○ 반면에 마이그레이션은 최신 버전의 객체로 계속적인 업데이트가 필요하기 때문임

❒ 위 그림은 25년간 디지털 객체 천만 개(개당 1mb 크기)에 대한 접근을 유지하기 위해 필요한 마이그레이션 비용과 dioscuri의 실제 프로젝트 비용을 토대로 한 에뮬레이션 비용을 비교한 것임

○ 단일 디지털 객체는 KB의 e-Depot에 들어가 있는 것으로 유지비용은 약 0.015 (ERU/연간)가 소요되는 것으로 보았음.

---

43) Andrew Jackson, Cost of emulation, emulationSercivesFAQ, 2010. 11. 13, http://wiki.opf-labs.org/display/PT/emulationServicesFAQCosts

○ 유지비용은 스토리지와 서버 용량에 대한 비용을 포함하고 인건비, 교육, 하우징 등의 비용은 제외하였음

❒ 에뮬레이션과 마이그레이션에 소요된 비용을 비교하였을 때 설사 5년 후에 두 개의 보존전략이 중단된다고 해도 에뮬레이션이 보다 비용 효과적인 해결책이라고 볼 수 있음

○ 에뮬레이션에 소요된 비용 항목은 유지비용과 추가개발비용, 계정 유지비용을 포함하였으며 dioscuri를 계속 개발하고 유지하기 위한 비용으로 600K(ERU/연간), 계정 유지를 위한 비용을 연간 20K(ERU/연간)로 보았음

○ 마이그레이션을 위한 비용항목은 유지비용과 마이그레이션 비용인데 천만 개의 객체에 대한 25년간 마이그레이션 비용을 500만(ERU/연간)로 보았음

❒ 아울러 에뮬레이션에 필요한 비용요소는 다음과 같은 항목들을 고려해야 함

○ 에뮬레이터를 새로운 플랫폼에 맞게 포팅하거나 마이그레이팅에 필요한 비용: 새로운 플랫폼마다 한 번씩 진행. 복잡성이 존재할 수 있지만 가상화 등의 활용이 도움이 될 수 있음

○ 소프트웨어 아카이브의 운영에 필요한 비용: 몇몇 기관이 소프트웨어 아카이브를 관리하기 위한 비용을 공유할 수 있음

○ 외부 인터페이스 업데이트에 필요한 비용: Planet은 웹 서비스 인터페이스이지만 미래에는 네트워크 커뮤니테이션 표준이 다를 것이며 GRATE와 같은 플랫폼이 다른 시스템과 통신하는 방법이 업데이트 될 필요가 있음

○ 디지털 객체를 위한 메타데이터 관리에 필요한 비용: 메타데이터는 각 객체를 그것을 보여주는데 사용할 수 있는 특정 기술 환경에 링크하기 위해서 필요함

## 나. 주요 서비스

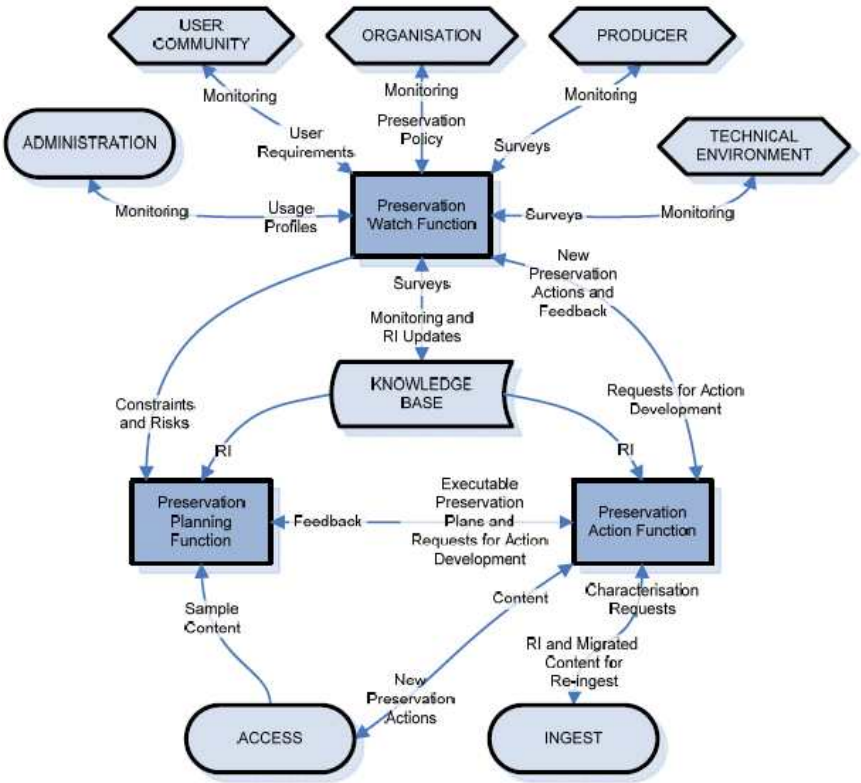
### (1) Planets의 기능모형(Functional Model)[44]

❒ Planets은 디지털 보존을 위한 통합 서비스(integrated services)를 제공하고자 하였으며 OAIS 참조모델을 준수하고 있음. 특히, Planets는 OAIS의 보존계획(Preservation Planning)기능을 실제로 구현한 사례로 볼 수 있음

❒ Planets이 수행하는 기능들의 개관을 살펴볼 수 있는 것이 아래 그림의 Planets 기능모델로서 상위 레벨의 기능 엔티티와 외부 엔티티(조직, 이용자, 커뮤니티, 생산자, 기술적인 환경), 데이터 및 메타데이터의 저장(Representation

44) PP7-D1. Report on Comparison of Planets with OAIS. 2007. 6.; PP7-D3-4. Report on The Planets Functional Model. 2009. 8.

Information 등), 저장소 기능 엔티티(입수, 접근, 관리)로 구성되어 있음)




<그림 3-19> Planets 기능모형

출처. PP7-D1. Report on Comparison of Planets with OAIS. 2007. 6.; PP7-D3-4. Report on The Planets Functional Model. 2009, p. 5

❒ Planets의 최상위 기능은 보존 동향 주지(preservation watch), 보존 계획(preservation planning), 보존 행동(preservation action)으로 구성됨

○ 보존동향 주지 기능은 광범위한 외부 엔티티를 모니터링 하고 제약이나 위험 요인을 보존 계획 기능에 반영함

○ 보존계획 기능은 디지털 객체의 보존 요건을 분석하여 가장 적합하고 실행 가능한 보존 전략을 선택함

○ 보존행동 기능은 디지털 객체에 대한 지속적 접근 가능성을 보장하기 위해 수행하는 조치로서 새로운 보존 실행 전략을 개발함

- 보존 행동은 디지털 객체 내용물(contents)을 명확하게 식별하는데 관련되어 필요한 하드웨어와 소프트웨어 환경을 갖추기 위한 것이라고 상정하고 있음
- 이러한 기능들을 수행하는 가운데 발생하거나 혹은 수행하는데 필요한 RI는 지식(Knowledge base)를 통해서 저장, 활용됨

❒ 에뮬레이션은 마이그레이션과 함께 보존행동의 주요 실행 전략임. Planets의 에

뮬레이션 서비스는 두 가지의 목적으로 수행될 수 있음

○ 첫 번째, 최종 이용자(end user setting)를 위한 것(최종 이용자를 위한 구현)

○ 두 번째, 마이그레이션의 과정이나 보존계획을 세우는데 있어서의 질적인 통제를 완성하려는 목적으로 수행. 이상의 설명된 Planets 기능을 중심으로 Planets의 서비스를 재구성하면 다음의 그림과 같음




<그림 3-20> Planets 기능과 서비스

### (2) Plato[45]

❒ Plato는 보존 계획을 위한 의사결정 지원 서비스의 명칭임

❒ 웹 기반 툴로서 네 개의 단계로 기관의 보존 계획을 정의하고 관리할 수 있게 해주는 체계임

❒ 보존 계획이란 것은 한번 쓰이면 변하지 않는 것이 아니라, 기술의 개선과 다른 외부 영향(예를 들어 예산 등)이 있으면 기존의 계획이 갱신되어야 할 것임. 따라서 Planets에서는 잠재적인 요구사항과 보존 계획에 관한 지식베이스를 토대로 여러 기관들이 신뢰할 만한 보존 계획을 수립하는 것을 돕기 위해 Plato를 개발한 것임

❒ The website of the Plato tool http://www.ifs.tuwien.ac.at/dp/plato/intro.html (Plato 4.2)(Plato 홈페이지에서 계정생성을 신청하면 Planning suite user account를 할당받고 사용할 수 있음)

45) The website of the Plato tool http://www.ifs.tuwien.ac.at/dp/plato/intro.html (Plato 4.2)

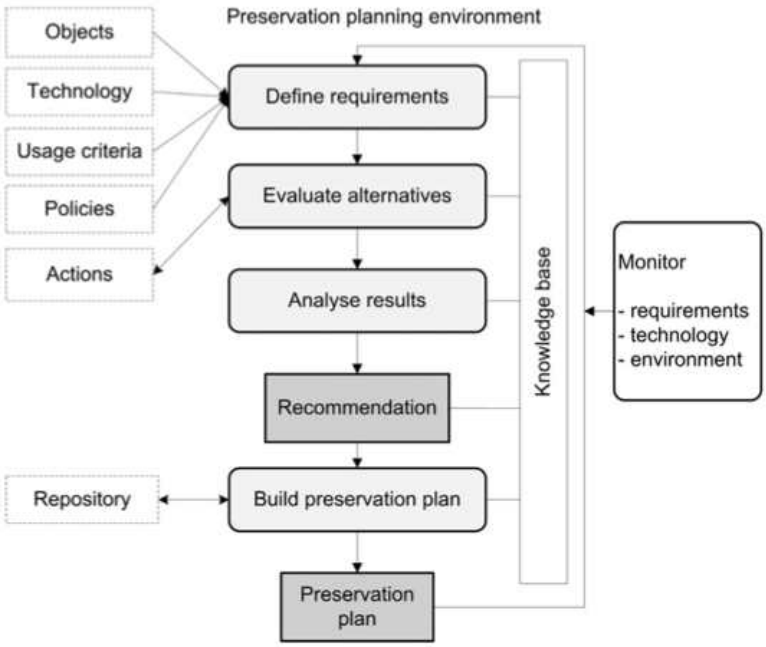
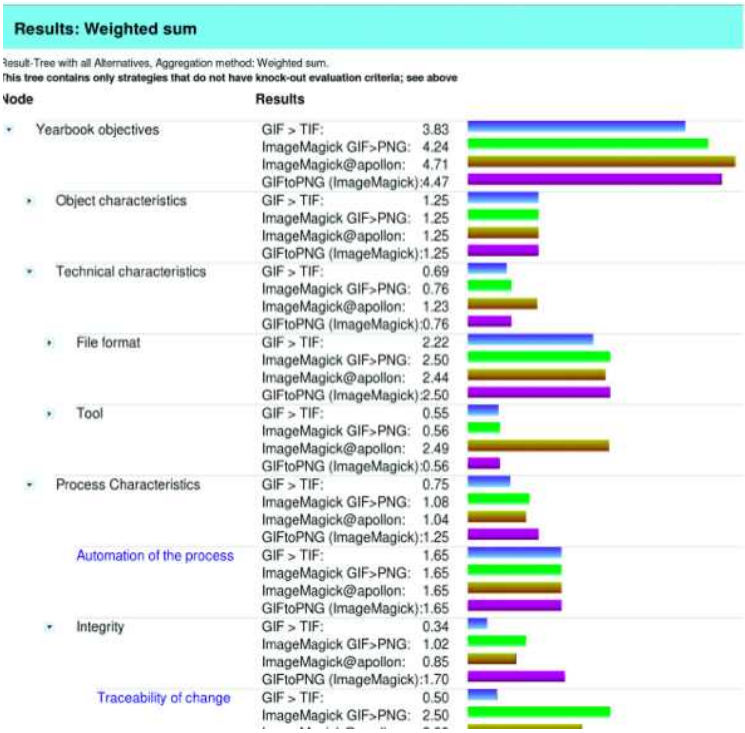



*Figure 1: Preservation planning environment.*

<그림 3-21> 보존 계획 환경

출처. Christoph Becker, Hannes Kulovits and Andreas Rauber, “Trustworthy Preservation Planning with Plato” ERCIM NEWS, 2010. 1, Number 80, p. 24.




<그림 3-22> 시각화된 결과물

출처. Christoph Becker, Hannes Kulovits and Andreas Rauber, “Trustworthy Preservation Planning with Plato” ERCIM NEWS, 2010. 1, Number 80, p. 25.

### (3) Characterization 서비스[46]

#### (가) 디지털 콘텐츠를 보존하기 위한 세 가지 기본적인 측면

❒ 내용 files

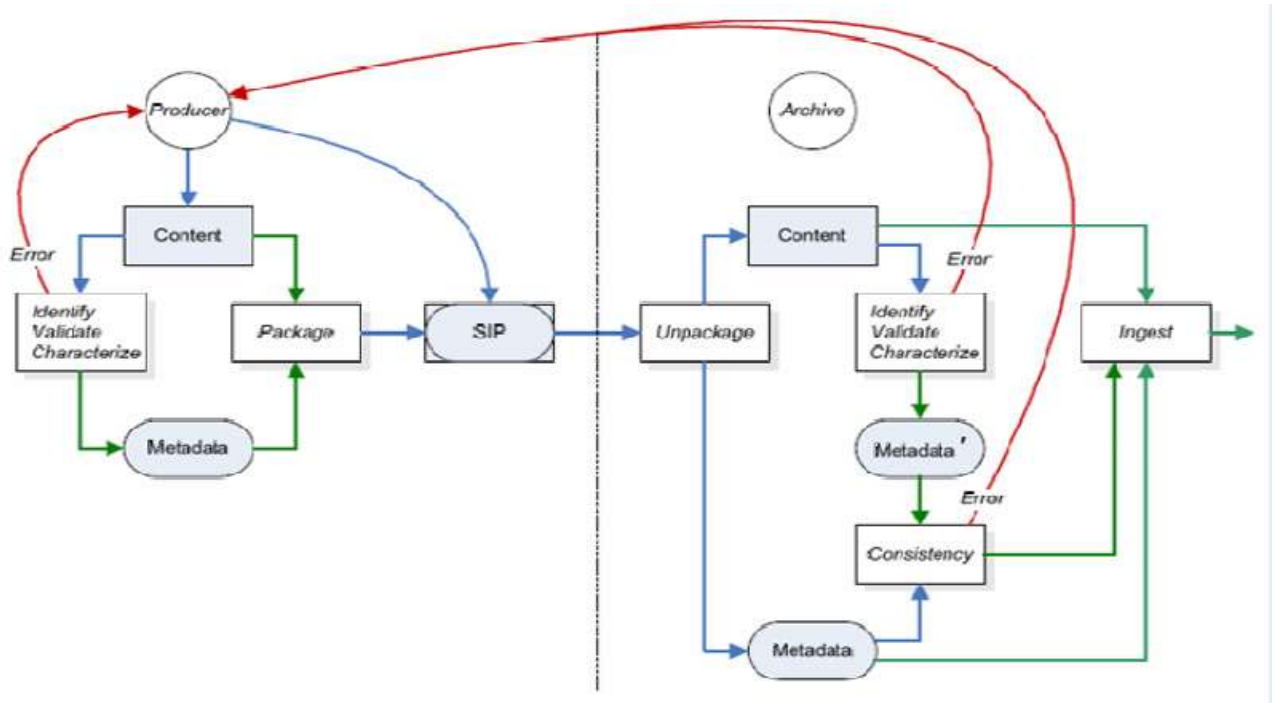
❒ 내용을 해석하기 위한 소프트웨어

❒ 내용을 디스플레이 하기 위한 하드웨어

#### (나) 서비스의 필요성

❒ 보존 계획을 위한 의사결정을 지원

❒ 보존 행동의 결과를 검증




<그림 3-23> 특성화 서비스

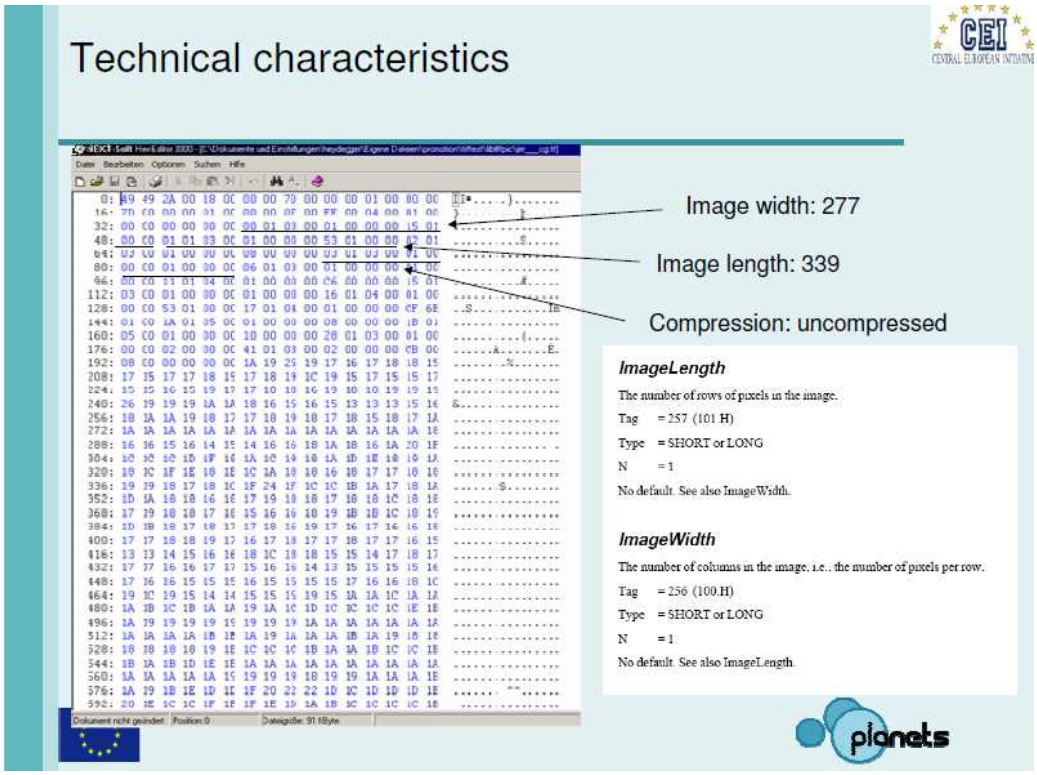
출처. Volker Heydegger and Jan Schnasse, "Characterisation of Digital Content" Digital Preservation - The Planets Way Sofia, 16 - 18 September 2009, p. 8

46) XCL에 관한 홈페이지 http://planetarium.hki.uni-koeln.de/planets_cms/ 관련 문서와 추출 소프트웨어 등이 제공되고 있다.

### (다) 서비스의 목적

❒ 파일 포맷[47]에 관한 기술적 메타데이터의 추출(이는 디지털 콜렉션을 관리하고 이해하는데 있어 중요한 속성임)

❒ 특성화는 전체 보존 프레임워크에서 중요한 부분, 파일포맷은 디지털 콘텐츠 재현에 있어서 중요한 개념, 하나의 포맷은 객체의 특성을 서술함. 많은 양의 포맷이 존재하지만 실제적으로 보존에 적합한 포맷은 그중 일부임




<그림 3-24> 기술적인 특성들

출처. Volker Heydegger and Jan Schnasse, "Characterisation of Digital Content" Digital Preservation - The Planets Way Sofia, 16 - 18 September 2009, p. 30

### (라) eXtensible Characterization Lanaguage(XCL)

❒ 보존계획 프레임워크를 지원하고, 파일 포맷 변환을 평가, 파일로부터 특성을 추출하기 위한 구체적인 모델, 효율성을 위한 도구임

❒ eXtensible Characterization Definition Lanaguage과 eXtensible Characterization Extraction Lanaguage로 구성됨

47) 포맷이란, 기본적 레벨에서 보면 바이너리 데이터일 뿐이지만 소프트웨어 레벨에서 의미 있는 일련의 바이트인 formatted data로서 저장됨. 그리고 가장 사람이 인식 가능한 형태로서 구현된 결과물로서 나타남.

### (마) 주요 내용

❒ 디지털 파일의 특별한 파일 포맷을 식별하기 위한 “식별 서비스”

❒ 디지털 파일의 내용이 식별된 파일 포맷의 스펙과 일치하는지의 여부를 검증하기 위한 ‘검증 서비스’

❒ 디지털 파일의 특별한 특성을 추출하기 위한 ‘추출 서비스’

❒ 추출 서비스에 기반하여 자동으로 서로 다른 디지털 파일의 특성을 비교하는 ‘비교 서비스

❒ Planets의 비교 서비스는 자동적으로 서로 다른 디지털 파일을 비교할 수 있도록 지원 됨

## (4) GRATE

❒ Planets이 제공하는 통합 에뮬레이션 프레임워크를 일컫는 명칭. GRATE란 Global Remote Access to Emulation Services의 약자로서 이용자는 웹 브라우저를 사용하여 원거리로 에뮬레이션 서비스를 제공받을 수 있음

❒ 아래의 그림은 에뮬레이션 프레임워크가 어떻게 구성되고 있는지를 가시적으로 보여줌. 이 모델의 핵심적인 요소는 에뮬레이터, 소프트웨어 아카이브, 파일 포맷 레지스트리임

- ○ GRATE 모델에 따르면 먼저, 이용자를 통해 구형의 디지털 객체가 GRATE 모델 안에 들어오게 되고 DROID/PRONOM[48]을 통해서 해당 디지털 객체의 정보가 식별됨
- ○ 다음으로 이 객체를 렌더링하기 위해서 어떤 에뮬레이터를 쓸 것인가가 결정됨. 즉, 식별된 구형의 디지털 객체를 보여주는데 필요한 애플리케이션 및 그것을 작동할 수 있게 해주는 에뮬레이터가 선택된다는 것임
- ○ 이러한 과정을 거치면서 구형의 디지털 객체는 이용자에게 제공될 수 있음

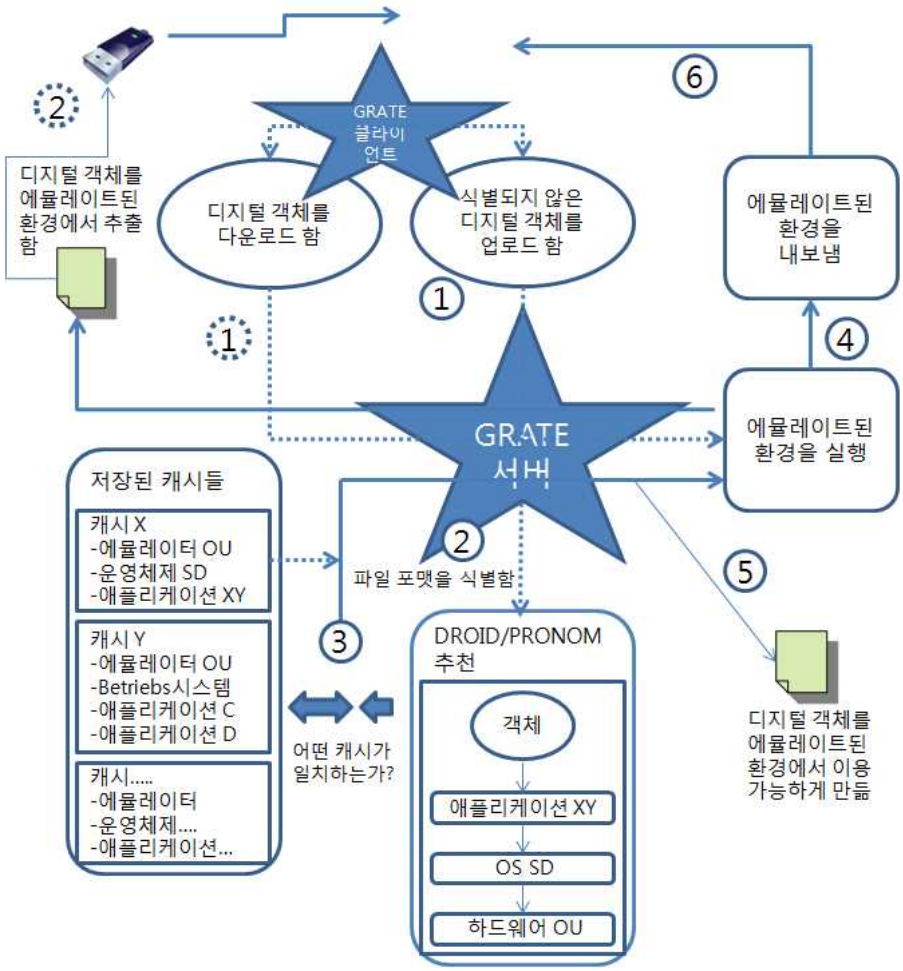
❒ GRATE에서는 Dioscuri, QEMU, UVC를 에뮬레이터로 사용하고 있음. Planets에서 사용되는 에뮬레이터들은 프로젝트를 통해서 직접 개발된 것이 아니지만 그

48) PRONOM은 TNA가 디지털 자원의 장기보존을 지원하고자 파일 포맷과 해당 파일을 지원하는 소프트웨어에 대한 정보를 정의하고 저장한 파일 포맷 저장소. PRONOM은 DROID(Digital Record Object Identification)과 함께 사용됨. 국가기록원, 『전자기록의 관리와 보존을 위한 국제협력 아젠다 개발』, 2007. p. 95.

중 Dioscuri는 KB 에뮬레이션 프로젝트를 통해 개발되어 Planets 및 KEEP 프로젝트를 통해 더 다양한 환경을 포괄할 수 있도록 후속개발이 이루어졌음

❒ 에뮬레이션 서비스는 객체 컨버전 계획(마이그레이션)속으로 통합될 수도 있음

### (가) GRATE의 기능




<그림 3-25> The GRATE framework

출처 : Planets, Frist Version of GRATE(PA/5-D7 v.1.0), 2006, p.17

❒ 서버 차원

○ 여러 개의 에뮬레이터 등의 소프트웨어를 임베딩하고

○ 로깅 세션들을 제어함

❒ 클라이언트 차원

○ HTML view를 통해 서버 속에 축적된 디지털 객체에 대한 메타 정보를 보여줌

○ 이용자는 별도의 인스톨 등 없이 원거리로 에뮬레이터를 사용함
○ 수정된 디지털 객체를 이용자의 시스템으로 다운로드 함
○ DROID/PRONOM 서비스를 통해서 이러한 객체를 자동으로 탐지할 수 있음
○ 탐지된 디지털 객체에 적합한 환경을 자동으로 실행함
○ 에뮬레이트 된 환경 내부에서 스크립트/소프트웨어를 실행함

❒ 에뮬레이터

GRATE에는 세 개의 에뮬레이터가 포함되어 있으며 각기 서로 다른 환경을 지원함

○ Dioscuri
  • Intel x86 하드웨어 기반
  • MS DOS 4.0, 5.0, 6.22, MS Windows 3.0, ELKS 16-bit Linux
○ QEMU
  • X86, ARM, SPARC, MIPS, MIPS64, N68k(Coldfire)
○ UVC(Universal Virtual Computer)
  • 플랫폼 독립 솔루션으로 텍스트, 스프레드시트, 이미지, 사운드와 같은 디지털 객체를 재현

### (나) Dioscuri

❒ Dioscuri는 2006-2008년 Tessella에 의해 개발이 주도되었고 2008년 이후로는 Planets, KEEP 프로젝트 내에서 네델란드 국립도서관이 개발에 주도적인 역할을 수행하였음[49)]

❒ GRATE에 포함되어 있는 대표적인 에뮬레이터로서 Open Source S/W이며, x86 컴퓨터 하드웨어 에뮬레이터로서 자바로 만들어졌음

❒ Planets, KEEP 프로젝트를 통해서 후속 개발이 이루어졌으며(더 다양한 환경에서 구동될 수 있는 방향으로) 최신 버전은 0.7.0 (2011. 1. 19)

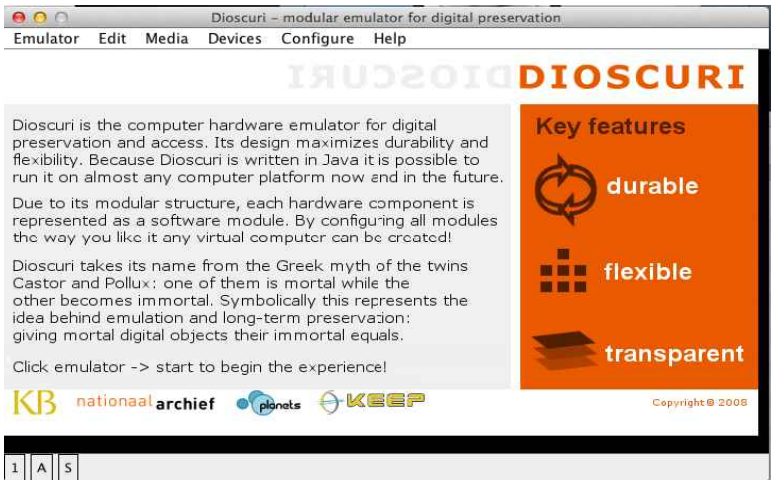
❒ 특징

○ 16/32-bit Intel 8086-based CPU (real-address mode only)
○ 1 MB RAM (expandable)
○ Storage devices: floppy, HDD

49) KB 프로젝트에서는 EVM(Emulation Virtual Machine), UVC(Universal Virtual Computer)의 개념에 기반을 두지만 모듈 방식과 내구성에 착안하여 "모듈러 에뮬레이션"이란 개념을 만들어냈고 이를 바탕으로 만든 에뮬레이터가 Dioscuri임. 모듈러 에뮬레이션 전략은 다섯 가지 주요 요소를 가짐 Universal Virtual Machine(UVM), Modular emulator, Component Library, Controller, Emulator Specification Document(ESD).

- Input devices: XT/AT/PS2 compatible keyboard
- Output devices: virtual screen
- VGA video graphics adapter
- DMA-support
- IRQ-handling based on a Intel 8259 PIC
- Timing mechanism based on an Intel 82C54 PIT and Crystal Clock
- Real-time clock with integrated CMOS
- System BIOS using Plex86/Bochs BIOS
- Video BIOS using VGA LGPl'ed BIOS

❒ Dioscuri 홈페이지는 http://dioscuri.sourceforge.net로, 에뮬레이터 최신버전과 소스코드 등을 직접 다운받아서 시험해 볼 수 있게 되어 있음




<그림 3-26> Dioscuri 에뮬레이터 실행 화면

## 다. 적용사례

### (1) 사례연구[50]

❒ Planets은 에뮬레이션 서비스의 방향을 결정하기 위해서 보존 방식으로서 에뮬레이션과 가상화에 관한 사례연구를 진행하였음

- 사례연구의 목적은 디지털 보존의 목적으로 에뮬레이터의 사용에 대한 이해와 에뮬레이터 사용과 관련하여 고려할 주요한 사항들을 파악하고자 한 것임
- 특히, 특정 에뮬레이션 도구들의 정확성을 테스트할 뿐 아니라 디지털 보존의 맥락에서 에뮬레이션과 가상화 도구의 사용과 관련된 다양한 측면을 파악하고

50) PA/5, 2008. 『Report describing results case studies』. 사례 연구 보고서를 토대로 작성하였음

자 하였음

- ○ 구체적으로 렌더링에 필요한 요소, 렌더링시 발생되는 문제, 추가적인 연구가 필요한 지점의 확인, 에뮬레이터를 구동하기 위해 필요한 지식 등을 연구함
- ○ 에뮬레이션과 가상화의 사례에서는 Planets 파트너가 제공하는 실제 컬렉션에서 제공된 디지털 객체들이 활용되었음
- ○ 에뮬레이션과 가상화  도구로는 Dioscuri, Virtual Box, Virtual PC, QEMU, VMware, MESS, UVC 가 사용됨

❒ 사례연구의 내용은 크게 다섯 가지로 구분되어 진행되었으며 오래된 웹사이트, 구형의 텍스트 문서, 오래된 지도 소프트웨어, 컴퓨터 게임, 상호작용의 보존에 관한 것임

- ○ Planets Dioscuri를 사용하여 웹 사이트 렌더링의 요구사항, 발생되는 문제점, 추가적인 연구 요구 등을 조사함
- ○ 1980년대 생산된 대표적인 워드 파일인 워드퍼펙트(Word Perfect 4.2)를 렌더링을 위해 필요한 사항과 단계, 문제점, 추가적인 연구 요구 등을 조사함
- ○ 영국 국립도서관의 오래된 지도용 소프트웨어에 접근하여 사용하기 위한 에뮬레이션 접근법을 조사함
- ○ 1980, 90년대의 인기 컴퓨터 게임을 대상으로 렌더링과 필요한 문서화와 관련된 사항을 조사함
- ○ IBM의 UVC를 사용하여 보존 객체가 외부세계와 갖는 상호작용을 보존하기 위한 사항들을 조사함

❒ 이 연구를 통해 Planets은 에뮬레이션과 가상화의 유의미성을 재확인하였음. 본 사례연구의 모든 사례에서 에뮬레이션 및 가상화 도구를 적용한 결과는 긍정적으로 평가되었음

- ○ 사례연구를 통한 교훈이 몇 가지로 정리되었음
  - 진본성: 진본성의 확보라는 측면에서 보았을 때 원래 환경이 여전히 활성화되어 있는 동안에 에뮬레이션 및 실제 환경에서의 비교 실험이 수행되어야 함.
  - 에뮬레이터의 준비: 에뮬레이터의 준비는 어렵고 복잡하며 따라서 이것을 자동화하기 위한 노력이 동반되어야 할 것임
  - 소프트웨어의 보존: 아직까지는 디지털 객체를 보존하는 것에 초점이 있지만 장차 운영체제, 응용프로그램, 드라이버, 플러그인, 폰트 등의 소프트웨어에 대한 보존 노력도 동반될 필요가 있음
  - 법적 시사점: 법적 관점에서 에뮬레이션은 노후된 용기(carriers)로부터 디스크 이미지(disk image)로 데이터를 복사함으로써 소프트웨어 복사 방지 메커니즘을 위반할 수 있음. 따라서 소프트웨어 라이선스 차원의 고려가 필요함.

- 컴퓨터 기술: 컴퓨터를 작동시키려면 하드웨어와 소프트웨어적인 지식이 필요함. 관련된 지식이 보존될 수 있도록 해야 함.

○ 추가적인 연구의 필요성은 다음과 같이 제안됨

- 디스크 이미지의 준비(Disk Image Preparation): 수많은 에뮬레이션과 가상화 도구는 가상의 미디어만을 지원하기 때문에 디스크 이미지가 준비되어야만 함
- 소프트웨어의 보존(Software preservation): 구식의 컴퓨터 환경을 재생산함에 있어서 가장 중요한 측면 가운데 하나가 소프트웨어로서 에뮬레이션을 위한 소프트웨어를 보존하는 것과 관련된 기술, 조직, 법적인 이슈들에 관한 연구가 필요함
- 원격 에뮬레이션(Remote emulation): 에뮬레이션 사용의 복잡성을 줄이고 사용자 편이성 및 상호 운용성 향상을 위해서는 원격 에뮬레이션이 필요함

### (2) Planet 서비스 적용사례[51]

❒ Planets은 개발된 서비스들에 대한 적용을 시도하였으며 그 중 핵심적으로 제시하고 있는 세 가지 적용사례는 다음과 같음

#### (가) Integrating Planets and Fedora Commons ; 2010. 8. 11. 포스트.

❒ 특성분석 서비스 적용사례임

❒ 덴마크의 주와 대학도서관이 보유하고 있었던 오래된 사운드 레코딩에 관한 것(180,000개에 달하는 물리적 아이템들로서 VHS tape, DAT tape, reel tape, wax cylinder, 신문들 등)

❒ 도서관에서는 새로운 디지털 객체관리시스템(DOMS)을 구축하였는데 Planets은 IF를 통해 외부 시스템과 상호 운용가능성을 확보할 수 있기 때문에, 특성분석 서비스를 통해 관리대상을 명확히 하고 PLANET Connector를 통해 외부의 저장시스템과 연결됨으로써 결과적으로 DOMS에서 디지털 객체가 관리될 수 있도록 하였음

51) Planets 사이트의 간행물 중 사례연구에 해당하는 것은 3건이 존재함, 해당 사례별 기사가 포스팅 되어 있음. Planets website>publication>case studies

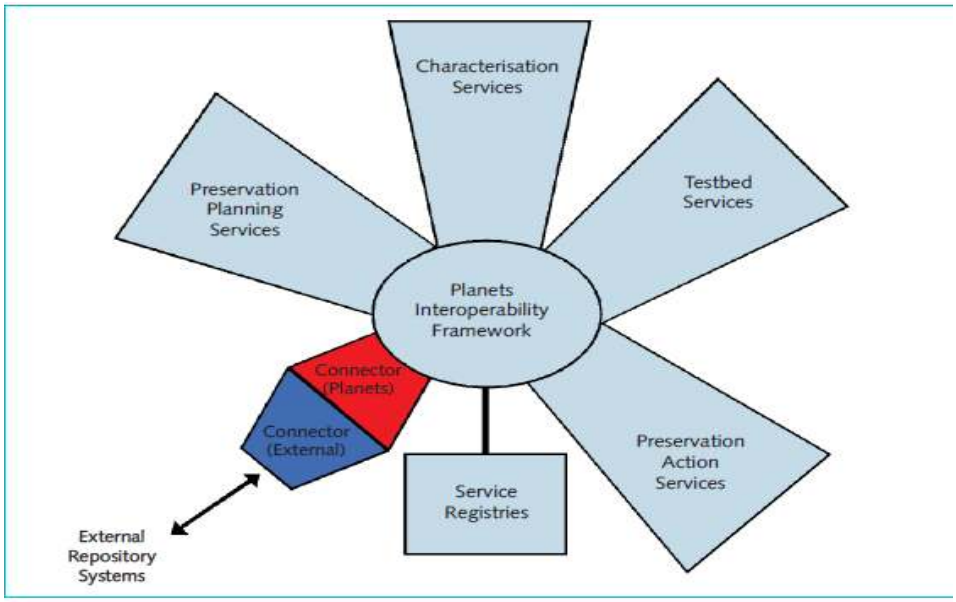



Figure 2 Overview of Planets Services and Interoperability Framework which unites them, showing the two parts of the adapter to external services.

<그림 3-27> 상호운용성 프레임웍을 활용한 Planets 서비스

### (나) 네덜란드 국립 아카이브와 에뮬레이션 : 2010. 7. 6. 포스트

❒ 이 사례는 마이그레이션으로는 적절하지 않은 스프레드 시트와 데이터베이스와 같은 동적인 레코드들의 보존에 있어서 에뮬레이션의 적정성을 살펴본 것임

❒ 이를 위해 렌더링과 마이그레이션 중간단계로서의 에뮬레이션의 두 가지 접근법을 검토하였음

❒ 보존 전략으로서 에뮬레이션의 필요성
- ○ Dutch Testbed는 기록관리의 입장에서 보았을 때 가장 중요한 것은 이메일이나 텍스트와 같은 디지털 객체의 내용을 보존하는 것이 가장 중요하다고 보고 마이그레이션이 가장 효과적인 전략이라고 결론지음
- ○ 또한, 에뮬레이션은 상호작용하는 요소가 있는 경우나 원래의 외형과 느낌을 보존할 필요가 있는 경우에 필요한데, 스프레드시트와 데이터베이스와 같이 포괄적인 보존을 위한 노력이 필요한 경우에 중요한 역할을 한다고 결론 맺음

❒ 에뮬레이션의 개관
- ○ 하드웨어 에뮬레이션이 일반적임, 이것은 호스트 시스템 상에서 작동되는 소프트웨어(에뮬레이터)를 만드는 것인데 이 소프트웨어가 타겟 시스템을 흉내 낸다고 설명, 타겟 시스템에서 작동되던 소프트웨어는 에뮬레이션 내부에서 작동한다고 설명

○ 에뮬레이터로서 Dioscuri, 이것은 Planets 프로젝트 내에서 더 개발 되었으며 보존 도구인 Planets suite 속에서 하나의 에뮬레이션 서비스를 형성

○ 하나의 에뮬레이터를 인스톨하고 구성하는 것이 일반 이용자에겐 어려운 일이며 최근까지 아카이브의 리딩 룸에서 에뮬레이터를 이용하곤 했음. Planets은 GRATE Framework를 사용해서 웹을 통해 원거리로 접근을 제공. 이용자는 원거리의 에뮬레이터에 그들의 집이나 사무실로부터 그들의 인터넷 브라우저를 사용하여 접근하며 이때는 어떠한 인스톨 등의 지식도 필요 없고 에뮬레이터를 실행시킬 필요도 없음

❒ 에뮬레이션 접근법

○ 렌더링(즉, 이용자를 위한 에뮬레이션)

- 데이터베이스와 같은 유형은 에뮬레이션 전략이 디지털 객체를 적절하게 구현하기 위한 최선의 방법이 됨
- 물론, 어떤 데이터베이스들은 표준 포맷으로 내용을 마이그레이션 하는 경우도 존재(SIARD S/W 관계형 데이터베이스)

○ 마이그레이션 과정 중에 행하는 중간단계로서의 에뮬레이션

- 마이그레이션은 텍스트 등에 있어서 가장 적합한 전략이지만 마이그레이션 진행 중에 에뮬레이션이 중요한 단계가 될 수 있음
- 만약 디지털 객체가 그것이 생산된 이래 오랫동안 관리되지 않은 채로 넘겨져 왔을 경우, 예를 들어 개인 컬렉션과 같이 그 안에 오래된 포맷으로 된 디지털 자료들을 포함하는 경우에 에뮬레이션이 파일 컨버전 이전에 먼저 시행될 필요가 있음. 또한 에뮬레이션은 Digital Depot과 같은 시스템에 입수시키기 전에 사전적인 단계로서 원래의 맥락을 재창조하기 위한 수단으로서 사용되기도 함

❒ 에뮬레이션이란 열린 선택을 유지하는 것

○ 대부분의 마이그레이션은 레코드의 특징들을 보존하며 그것은 마이그레이션 시점에 지정된 그 목적의 중요성에 따라 좌우됨. 그리고 그 외의 측면들은 유실되거나 변경됨

○ 원본을 유지한다면 우리는 미래의 보존과 접근 포인트를 극대화시킬 수 있음. 이것은 고고학 분야에서의 발전과 유사성이 존재함

○ 원래의 파일을 유지하고 에뮬레이션을 사용하는 것은 원래의 환경 속에 있는 객체에 대한 접근을 제공하고 그것에 대한 보존과 접근을 유지하는 열린 선택을 가능하게 하는 것이며, “디지털 고고학”에 대한 동시대적인 접근임

### (다) 디지털 민속자료의 보존 : 2010. 7. 6. 포스트

❒ 덴마크 민속 아카이브가 2008년에 덴마크 국립도서관에 병합되면서 두 기관이 보존 대상이 되는 디지털 객체를 프로파일 함에 있어서 Planets의 특성화와 계획 툴(Plato)을 어떻게 사용할 수 있는지를 시험해 보았음

## 라. 시사점

❒ 이 프로젝트는 디지털 객체에 관한 장기보존 서비스 및 도구를 전체적으로 제시해 줌으로써 보존 전략 중의 하나로서 에뮬레이션 활용에 대한 포괄적인 시각을 제공해 줌

○ Planets의 서비스들은 통합된 보존 계획 서비스라고도 이해할 수 있으며 장차 국내에서 보존관련 전체 서비스를 기획한다고 하였을 때 참고할만한 유용한 사례가 될 수 있음

○ 보존 계획은 한번 세워지면 변하지 않는 것이 아니라 기술의 개선과 다른 외부 영향 등에 의해서 계속적인 갱신의 필요성이 존재함

❒ 현존하는 보존 도구들의 취약성을 진단한 가운데 통합된 보존 계획 서비스의 사례를 제공해 줌

○ 현존하는 보존 도구들이 독립적인 애플리케이션으로 존재하여 보존 대상에 대한 포괄적인 접근이 어렵다는 점을 인식

○ 특정 요소(기능)가 임베디드 되어 있거나 다양한 포맷으로 구성된 복잡한 디지털 객체들에 적절한 보존 계획은 기존의 단일 전략 혹은 애플리케이션으로는 부족함[52]

❒ 보존 전략 중의 하나인 에뮬레이션이 어떻게 실제 보존 전략 체계 속에서 결합되어 구사되는 것인지를 살필 수 있게 해줌

○ 에뮬레이션 전략 자체를 목적으로 한 프로젝트는 아니지만 보다 근본적으로 보존계획 차원에서 에뮬레이션 전략을 구상하기 위한 의미 있는 시사점을 주고 있음

❒ 장기보존과 관련된 핵심적인 표준인 OAIS 보존계획이 어떻게 실제로 구현될 수 있는지에 관한 모델을 보여주고 있음

○ Planets 산출물 중에는 기능모델을 OAIS의 관점에서 비교 분석된 결과물을 포함하고 있으며 이는 장기보존 기능이 OAIS를 준수하면서 상호 운용되는 환경

52) Adam Farquhar & Helen Hockx-Yu, "Planets: Integrated Services for Digital Preservation", *The International Journal of Digital Curation*, Issue 2, Volume 2, 2007, p. 89.

으로 발전해 나가야 한다는 의지를 보여주고 있음

❒ 프로젝트가 종료되었지만 Open Planets Foundation을 통해서 그 결과물들이 유지 운영되고 있고 이를 통해 에뮬레이션을 포함하여 다양한 장기보존과 관련된 기법과 서비스들이 연구되고 공유되고 있어 계속적으로 주목 대상임

### 3.1.5 Keep 프로젝트

#### 가. 개요

#### (1) 배경과 목표

<표 3-3> Keep 프로젝트 개요

| 구분 | 내용 |
|---|---|
| 본 명칭 | Keeping emulation environments portable |
| 진행기간 | 2009. 2. 1. - 2012. 2. 29. (37개월) |
| 주관기관 | Bibliotheque Nationale de France |
| ICT 세부 영역 | Digital libraries and technology-enhanced learning |
| 예산 | 3,973,445 EURO (58억5,224만) |
| 공식사이트 | http://www.keep-project.eu/ |

❒ KEEP(Keeping emulation environments portable)은 2009년 유럽 연합의 일곱 번째 프레임워크 프로그램(FP 7-ICT)의 지원을 받아 시작된 프로젝트임

❒ 프랑스 국립도서관(BnF)이 주관하고 에뮬레이션에 관한 다양한 유럽의 이해관계자를 포함하는 9개 기관이 컨소시엄에 참여하였으며 2012년에 종료되었음[53]

❒ KEEP은 에뮬레이션에 기반한 영구적인 접근전략과 그것을 지원하는 몇 가지의 도구를 개발하고자 하였음

○ KEEP은 정적(텍스트, 사운드, 이미지, 비디오 파일 등), 동적(멀티미디어 문서, 소프트웨어, 웹사이트, 데이터베이스, 비디오게임 등) 특성을 포괄하는 디지털 객체를 정확하게 렌더링 할 수 있는 에뮬레이션 접근 플랫폼(emulation access

53) 프로젝트 기간은 2009. 2. 1. - 2012. 2. 29. 참여한 기관은 문화유산기관(도서관), 연구기관(대학), 상업적 파트너(법적 컨설턴트)와 게임회사들로서 Bibliothèque nationale de France, Joguin SAS, Koninklijke Bibliotheek, Computerspiele Museum, University of Portsmouth, Deutsche Nationalbibliothek, Cross Czech a.s, Tessella, European Games Developer Federation. 이다. http://www.keep-project.eu/

platform)을 목표로 하였음

❒ KEEP은 다양한 선행 연구들이 존재하고 있지만 실제적인 해결책이 제한적이라고 평가하고 있음.

○ 특히, 웹사이트, 게임, 소프트웨어 등의 복잡한 디지털 컬렉션들의 경우는 에뮬레이션 외에는 다른 대책은 현실적으로 의미가 없다고 판단함

○ 왜 이전 파일이나 프로그램에 더 이상 접근하지 못한다고 생각하는가에 대한 이용자의 답변[54]은 다음과 같음 :

- 본인의 새로운 컴퓨터가 그 작업을 더 이상 수행하지 못함(31%)
- 디스크나 테이프를 읽기 위한 드라이버를 갖고 있지 않음(30%)
- 저장용기가 손상(18%)
- 원본 소프트웨어를 더 이상 갖고 있지 않음(14%)
- 소수로 작동 법을 모른다 등의 답변이 있었음

○ 디지털 객체에 대한 보존(preservation), 접근(access), 구현(rendering)의 과제가 존재

- 디지털 오브젝트를 장기간 유지하고 쉽게 접근(정적/동적)
- 에뮬레이션 접근 플랫폼(emulation access platform)을 통해서 저장과 에뮬레이트를 허용하는 툴을 개발하고 원본 매체의 내용을 재현
- 에뮬레이터를 실행시키기 위한 '범용적인' 가상 머신을 생성(이 소프트웨어는 다양한 미래의 컴퓨터에 적용 가능함으로서 에뮬레이션 플랫폼의 유지가능성을 높여주는 기능을 함)

○ 에뮬레이션 대상과 관련하여 디지털 보존상의 해결될 이슈를 정리함

- 디지털 보존 도구(tools)와 환경(settings)의 관리
- 보존 과정에 대한 셋업 자동화
- 노후화된 저장용기[55]로부터 데이터의 검색

❒ KEEP이 개발하고자 하는 것은 에뮬레이션 보존 전략을 수행하기 위한 기반이 될 수 있는 서비스로서 프레임워크를 생성하는 것임

○ 이 프레임워크는 기존의 그리고 미래의 에뮬레이터 및 전송도구들을 포괄할 수 있는 것이며 자체로 에뮬레이터와 전송도구를 생성하는 것이 목적이 아님. 즉, 기존의 도구들을 통합적으로 활용할 수 있게 해주는 부가 서비스 체계로 이해할 수 있음

○ KEEP를 이러한 도구들을 프레임워크라고 표현하였는데 이것은 해당 기능을 제공해주는 여러 개의 다중의 전송도구, 에뮬레이터를 포함하고 통합적으로 활용할 수 있게 하는 서비스로서 이해할 수 있음

54) 2012년 1월 24-25일 Cardiff Workshop에서 발표된 KB, BnF, DNB의 조사결과 중.
55) 가지고 다닐 수 있는 저장장치를 의미하므로 저장용기, 용기로 표현하였음 예) 플로피디스크, SD 메모리 등

❒ KEEP의 주요 산출물은 다음과 같음 :

○ KEEP 에뮬레이션 프레임워크(KEEP Emulation Framework)

○ 에뮬레이션에 필요한 메타데이터의 측면을 밝히기 위한 보고서(기존 메타데이터 모델에 관한 연구, 기술 메타데이터 모델에 대한 설계, TOTEM(Trustworthy Online Technical Metadata Database)을 포함)

○ 전송과 관련하여 발생할 수 있는 법적 측면을 분석한 보고서(Layman's Guide)

○ KEEP 전송 도구 프레임워크(KEEP Transfer Tool Framework)

○ 가상 머신(Virtual Machine)

### (2) 세부 프로젝트별 서비스

❒ 이 프로젝트는 작업 계획과 관련된 활동들로 묶여진 일곱 개의 워크 패키지로 구성되어 있음

❒ 각각의 워크 패키지의 내용은 다음과 같음

○ WP 1 데이터 전송(transfer) 도구와 법적 연구

- 오래된 데이터 용기(carriers)상의 데이터를 안전하게 디지털 아카이브로 전송하는데 사용될 수 있는 TTF를 개발하고
- 이때, 미디어와 에뮬레이션 환경 간에 저작권이 있는 자료들의 전송과 관련된 법적인 문제를 함께 연구함

○ WP 2 핵심적인 에뮬레이션 프레임워크

- EF 중에서도 시스템 측면의 핵심적 EF를 다룸

○ WP 3 메타데이터 연구와 프론트엔드 프레임워크

- EF에서 접근 측면에 해당하는 프론트엔드 부분을 다룸
- 아울러 에뮬레이션에 필요한 메타데이터 구조도 함께 연구함

○ WP 4 에뮬레이터의 이동성(portability)

- 점차 노후화되는 에뮬레이터를 위해 다층 레이어드의 가상 머신 개발하여 에뮬레이터의 이동성을 확보하기 위한 연구를 진행

○ WP 5 통합 테스트, 검증, 기술적 유지가능성

- 일련의 통합 솔루션인 기술적 작업 패키지의 산출물을 시험하고 검증
- 프로젝트의 산출물을 유지하기 위해 장기보존 전략을 생성하고 실행

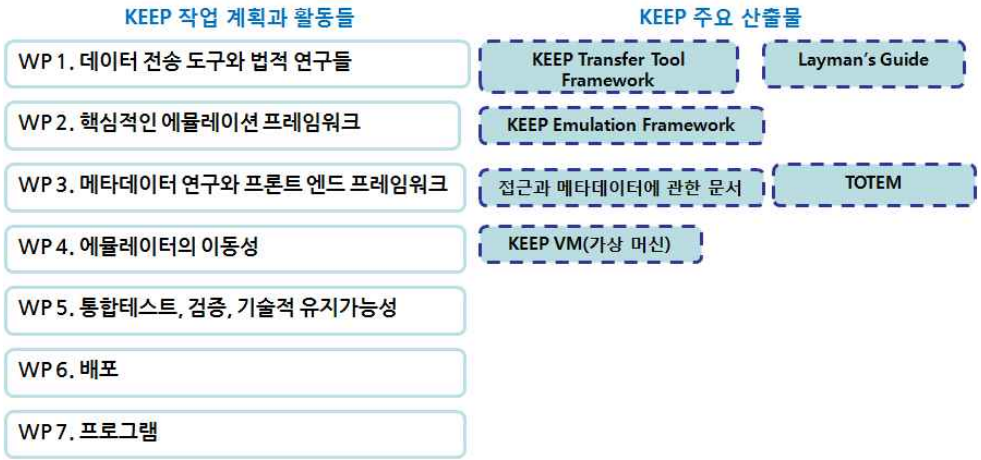
○ WP 6 배포

- 유럽 및 전 세계의 과학, 산업, 학술, 게임 개발 커뮤니티에 결과물을 배포하여 프로젝트의 목표를 실현

○ WP 7 프로그램 관리와 조정

- KEEP 프로젝트 전반에 걸친 관리 및 조정
- EU와의 관계 유지

❒ KEEP의 워크 패키지들과 그에 따른 산출물을 정리한 것이 아래의 그림으로 에뮬레이션을 위한 핵심적인 두 개의 프레임워크를 중심으로 이것을 실질적으로 지원하기 위한 활동들로 구성되어 있음




<그림 3- 28> KEEP의 주요 활동과 산출물

(3) 용어

KEEP에서 사용된 용어정의는 다음과 같음

❒ Emulation Framework(EF)

에뮬레이션 프레임워크란 사용자 친화적인 방법으로 컴퓨터 환경을 자동으로 재생산할 수 있게 하는 소프트웨어임. 원래 사용되던 컴퓨터 환경 속에 있는 데이터를 렌더링할 것임. 파일 컨버전은 진본성 보장을 위해 요구되지 않음

❒ Transfer Tool Framework(TTF)

다양한 디지털 매체 캐리어들이 담고 있는 원래의 비트 스트림을 안전하고 정확하게 전송할 수 있게 하는 프레임워크를 말함. 오래된 데이터 캐리어 상에 데이터를 여전히 놔두고 있는 조직들을 도울 것

❒ Analysis Tools

미디어 저장용기 이미지를 분석하기 위한 소프트웨어

❒ Technical metadata

디지털 매체 저장용기(carrier) 그 자체에 관한 메타데이터

❒ Technical registry, External technical registry

외부 레지스트리는 예를 들어 파일 포맷 등에 관한 기술적 데이터(technical data)를 제공해 주는 것으로서 웹 서비스를 통해 검색될 수 있음

❒ Transfer tool

매체 저장용기의 이미지를 생성하기 위한 소프트웨어

❒ Transfer Workflow

디지털 매체 용기 이미지를 통합된 전송/분석 툴을 사용하여 전송하고 확인하기 위해 요구되는 단계를 포함하는 커스터마이즈 가능한 업무 흐름. 내부 데이터베이스로부터 매체 전송에 관한 지식이 유지됨

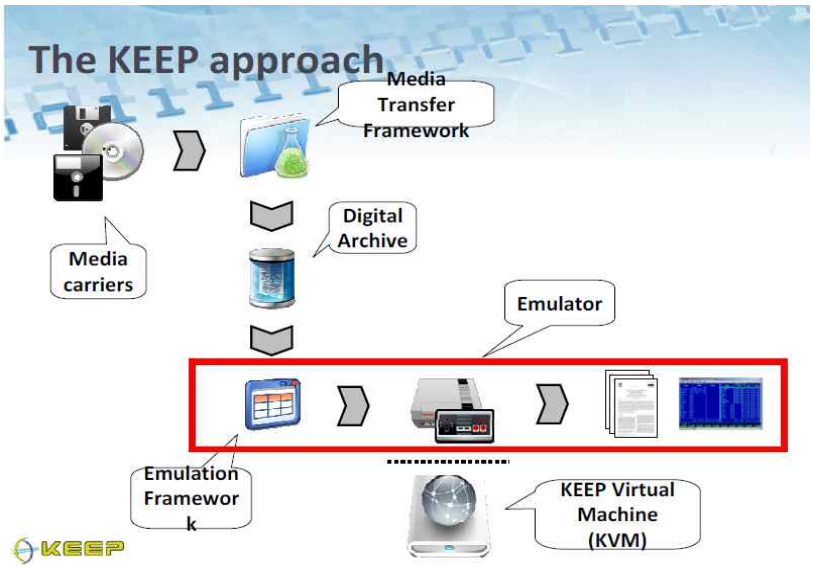
## 나. 주요 서비스

### (1) 서비스의 전체 개념

❒ KEEP은 에뮬레이션에서 핵심적으로 해결되어야 할 두 가지 이슈를 위한 서비스 프레임워크로 이해될 수 있음. 이를 위해 두 가지의 도구(프레임워크로 표현됨)가 통합적으로 제공되도록 설계되어 있음

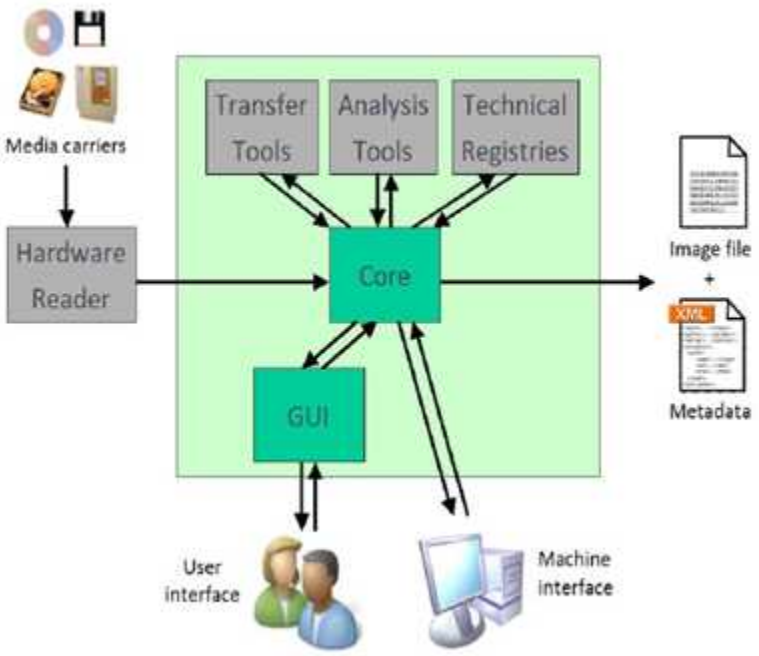
❒ KEEP이 제공하고 있는 서비스는 크게 두 가지의 차원에서 활용(접근)됨

- ○ 첫 번째는 아카이빙 할 때로서 오래된 매체에 있는 데이터를 TTF를 통해서 추출하고 추출 결과를 저장 매체(disk image)에 메타데이터와 함께 담음. 이것은 디지털 아카이브에 저장될 것임
- ○ 두 번째는 이용자의 에뮬레이션 요구가 있을 때로서 오래된 컴퓨터 환경을 구현할 필요가 발생하면 EF를 사용해서 자동적으로 가장 적합한 에뮬레이터를 가동시켜서 구현함. 이때 에뮬레이터 자체도 노후화되기 때문에 KEEP VM 을 통해서 실행. 이러한 에뮬레이션 활동과 관련된 KEEP의 접근법을 전체적으로 보여주고 있는 것이 아래의 그림임




<그림 3-29> 에뮬레이션을 위한 KEEP의 접근법

(2) 전송도구 프레임워크 (Transfer Tool Framework)




<그림 3-30> TTF의 개요

출처. WP1, "Requirement specification for the Transfer Tools Framework ", D1.2a, 2009, p.16

❒ KEEP의 전송도구 프레임워크(이하, TTF)는 점차 노후화되고 있는 저장용기[56] 로부터 보존 대상이 되는 디지털 객체 원래의 비트스트림을 안전하게 포착하기 위한 도구

56) floppy disk, Compact Disc, SD memory 등과 같은 휴대가 가능한 저장매체를 의미한다.

○ TTF란 다양한 디지털 매체 용기들이 담고 있는 원래의 비트 스트림을 안전하고 정확하게 전송할 수 있게 하는 프레임워크를 말하는데 오래된 데이터 용기상에 데이터를 여전히 놔두고 있는 조직들에게 유용할 것임

❒ TTF의 주요 요소는 Core와 GUI이고 커다란 네모가 관련 범위임. 주요 요소들을 통해 디지털 미디어 용기들로부터 Image file과 Metadata가 생성됨

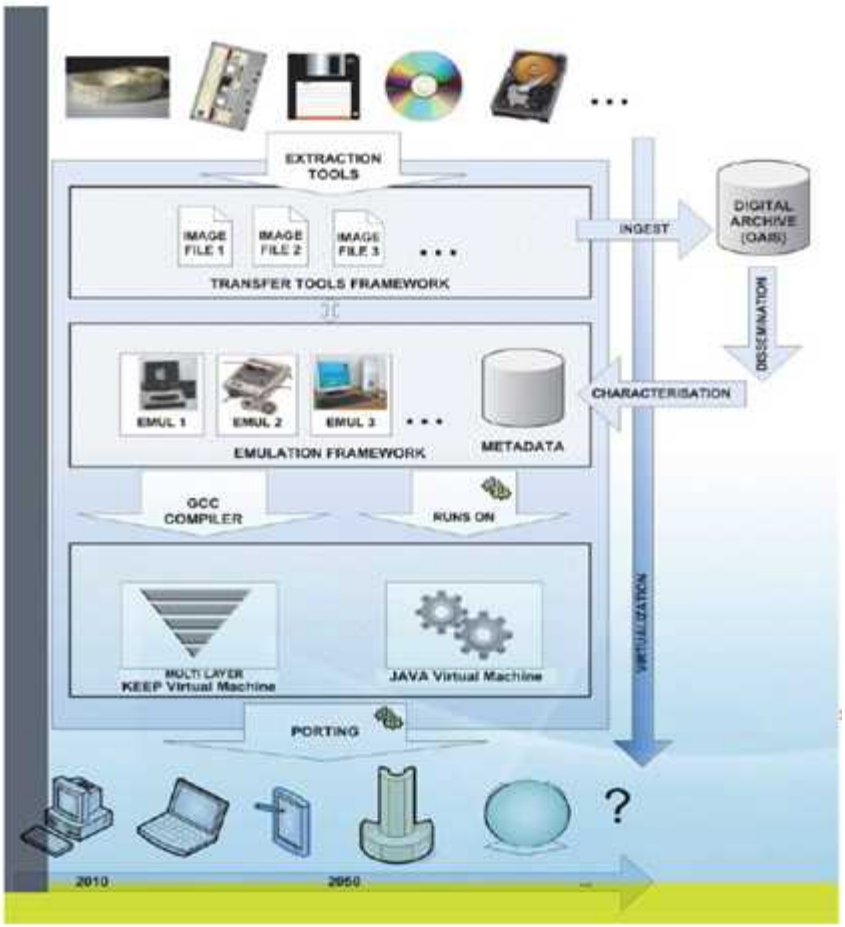
❒ Metadata는 서술적(descriptive) 메타데이터와 기술적(technical) 메타데이터로 구분됨, 이때 외부의 레지스트리로부터 데이터가 입수될 수 있음

❒ 글로벌 파일 포맷 레지스트리로서 PRONOM[57], PCR, UDFR, Mediapedia 등의 technical registry는 전송 도구와 이미지 포맷에 관한 정보에 접근할 수 있는 유용한 정보원이 됨

❒ 한편, 이때 사용할 수 있는 Characterization tool로는 DROID, JHOVE 등이 있음

❒ 시나리오
- ○ 단일 매체 용기의 전송
- ○ 관리자가 새로운 전송/분석 도구를 더하거나 기존의 전송/분석 도구를 편집 혹은 제거
- ○ 관리자가 새로운 전송 워크플로우를 셋업하거나 기존의 전송 워크플로우를 편집 혹은 제거
- ○ 이용자가 완성된 전송 워크플로우에 피드백
- ○ 일련의 매체 용기들의 전송
- ○ 이용자가 생성된 메타데이터를 보거나 편집 등

57) TNA이 디지털 자원의 장기보존을 지원하고자 파일 포맷과 해당 파일을 지원하는 소프트웨어에 대한 정보를 정의하고 저장한 파일 포맷 저장소. PRONOM은 DROID(Digital Record Object Identification)과 함께 사용되는데 이것은 디지털 객체의 바이너리 구조를 분석하여 자동적으로 포맷을 식별할 수 있는 도구임. 국가기록원,『전자기록의 관리와 보존을 위한 국제협력 아젠다 개발』, 2007. p. 95.




<그림 3-31> 에뮬레이션 접근 플랫폼에서의 TTF

출처. WP1, "Requirement specification for the Transfer Tools Framework ", D1.2a, 2009, p.13

### (3) 에뮬레이션 프레임워크(Emulation Framework)

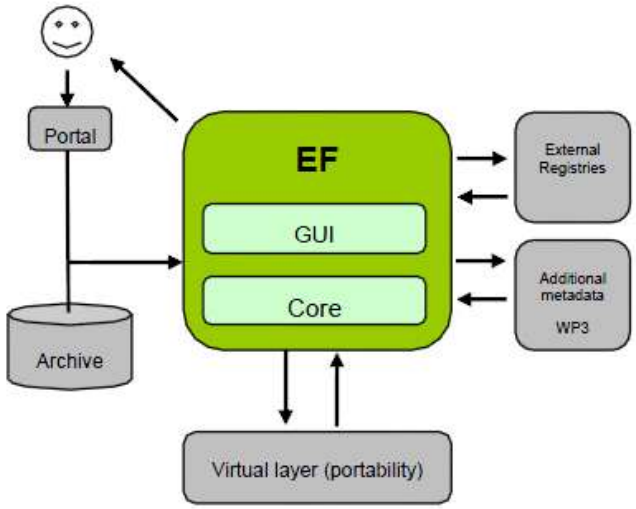
❒ KEEP의 에뮬레이션 프레임워크(이하, EF)는 에뮬레이션을 위한 자동화된 작업 체계를 제공해 주는 것으로 주요 기능은 세 가지로 기존의 아카이빙 솔루션과 상호작용하며, 디지털 오브젝트와 메타데이터를 EF로 보내야 하고 렌더링을 해야 함

❒ EF의 핵심적인 요소는 에뮬레이션 코어 엔진, 소프트웨어 아카이브, 에뮬레이터 아카이브, 파일 포맷 레지스트리임.

- ○ 코어 엔진은 시스템의 기술적 심장으로 자동으로 파일 포맷의 특성화를 하고 요구되는 소프트웨어를 선택하고 에뮬레이션 환경을 구성함
- ○ 소프트웨어와 에뮬레이터의 선택을 위해서 코어 엔진은 파일 포맷 분류체계 등을 갖고 있는 기술 레지스트리 등과 상호작용함
- ○ 소프트웨어 아카이브는 사전에 정의된 디스크 이미지안에 포착되어 있는 소프트웨어를 갖고 있고 에뮬레이터 아카이브는 EF에서 이용 가능한 에뮬레이터를

갖고 있음

❒ EF가 구현해야 할 기능들의 요건은 GUI, 외부 인터페이스, 에뮬레이션 프로세스, 에뮬레이션 환경, 사용성의 다섯 영역으로 구분되어 있으며 전체 45개 항목으로 구성되어 있음. 각 항목은 필수 요구사항(M), 바람직한 요구사항(D), 선택적 요구사항(O), 향후 개선사항(E)으로 나뉘어져 이 중 선택적 요구사항부터는 프로젝트 관리자의 재량에 따라 기능 구현이 최종 결정됨.

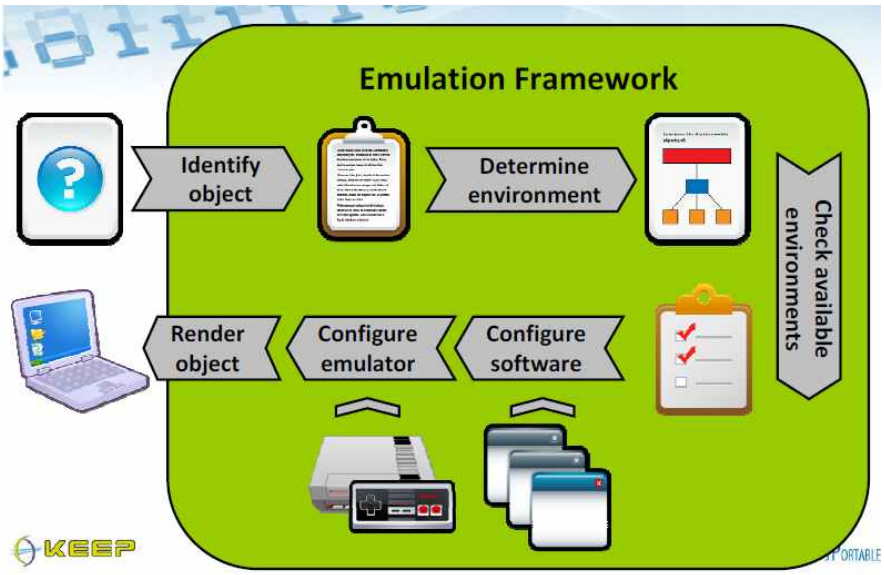



<그림 3-32> EF의 기본 개념

출처, WP2, "Requirements and design documents for services and architecture of emulation framework Part A: Requirements document for services and architecture of emulation framework", D2.2a, 2009, p.4

❒ EF의 작업은 다음의 순서로 진행됨

- ○ 작업 대상인 객체를 식별함
- ○ 객체와 관련된 환경을 결정함, 이때 EF 내에서 이용가능한 환경을 체크함
- ○ 이러한 정보를 토대로 소프트웨어와 에뮬레이터를 각각 구성함
- ○ 결과적으로 이용자에게 객체를 렌더링해 줌. 이상의 내용은 아래 그림과 같이 정리될 수 있음




<그림 3-33> Emulation Framework Workflow

❒ EF의 시스템을 에뮬레이션 환경의 측면에서 필요한 구성요소별로 확인할 수 있는 것이 다음의 그림임

- ○ 이용자의 요구로 오래된 컴퓨터 환경을 구현할 필요가 발생하게 되면 코어 엔진이 보존대상이 되는 파일 포맷에 대한 특성화 작업을 진행함
- ○ 이때, DROID/PRONOM을 포함하는 외부의 파일 포맷 레지스트리가 해당 디지털 객체의 정보를 식별하는데 활용됨
- ○ 이러한 결과를 토대로 에뮬레이터 아카이브 및 소프트웨어 아카이브로부터 자동적으로 해당 파일 포맷에 가장 적합한 에뮬레이터를 가동 시켜서 구현이 이루어지게 됨
- ○ 이를 위해서 EF는 7개 에뮬레이터와 6개 플랫폼을 포함하고 있음[58)]

❒ EF은 KEEP 홈페이지를 통해서 테스트 가능함

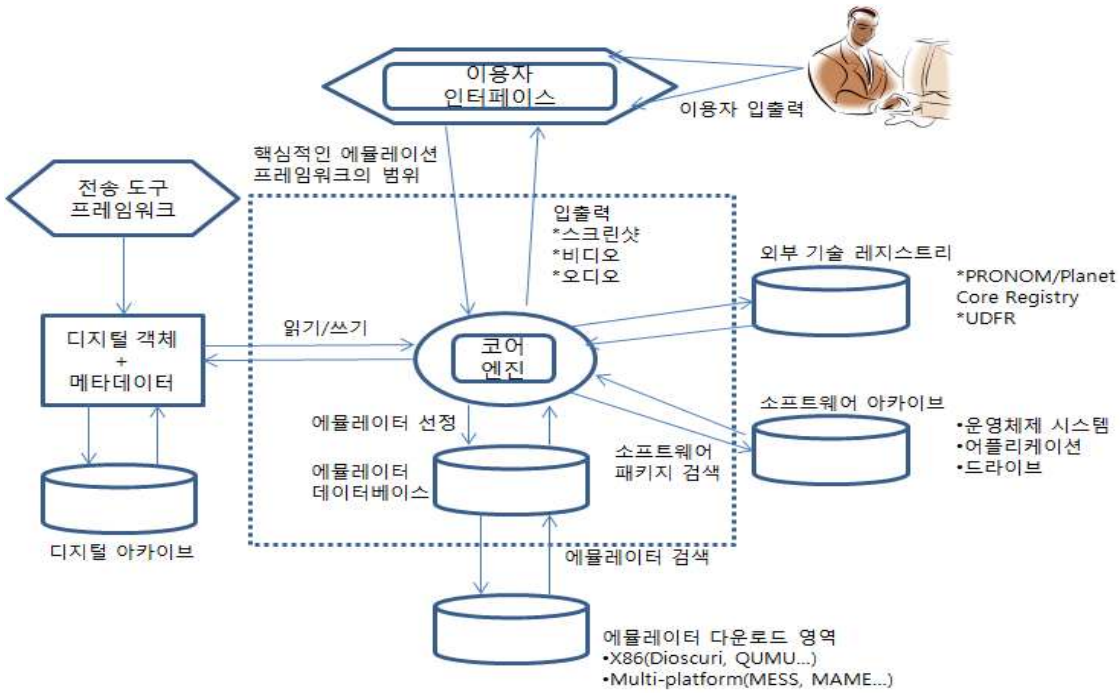
- ○ 홈페이지(http://emuframework.sourceforge.net/)에서 패키지를 다운하고 테스트할 수 있음
- ○ 최신 (2.1.0) 버전 (오픈소스 소프트웨어만 포함하고 있음. MS 윈도우나 워드퍼펙 같은 것은 없음. 소프트웨어 라이선스를 관리하는 것은 본 프로젝트의 범위를 넘어서는 것임)
- ○ 사양은 7개 에뮬레이터와 6개 플랫폼을 포함: Qemu(x86), Dioscuri(x86), WinUAE(Amiga), Vice(c64), JavaCPC(Amstrad/Schneider), BeebEm(BBCmicro), Thomson(Thomson T07)
- ○ 25개 파일들을 포함: PDF, TXT, XML, TIFF, PNG, BMP, Quark, ARJ, EXE,

58) Qemu(x86), Dioscuri(x86), WinUAE(Amiga), Vice(c64), JavaCPC(Amstrad/Schneider), BeebEm(BBCmicro), Thomson(Thomson T07)). 또한 25개 파일들(PDF, TXT, XML, TIFF, PNG, BMP, Quark, ARJ, EXE, disk/tape images 등

disk/tape images 등

○ EF는 다운 가능한 패키지의 형태로 이용가능하기 때문에 얼마의 IT 기술력이 있다면 테스트를 위한 셋업이 가능함

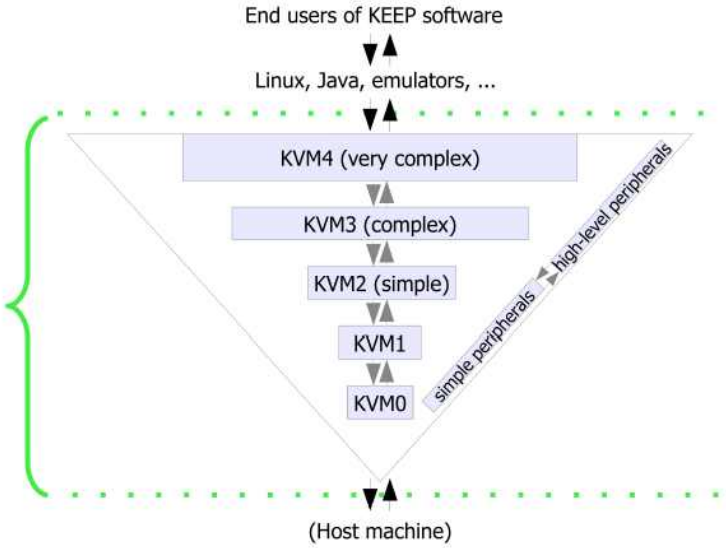
- 그러나 에뮬레이션 프레임워크는 하나의 프로토타입이기 때문에 서비스 지원 같은 것이 되지 않으며 공식 제품은 아님




<그림 3-34> EF system의 환경 개요

출처 : KEEP, “Requirements and design documents for services and architecture of emulation framework: Part B: Architectural Design for the Core Emulation Framework(WP2-D2.2b)“, 2009, p.43.

### (4) VM(Virtual Machine)



The KEEP Virtual Machine components and their interactions with application software at one end and host computer platform at the other

<그림 3-35> KEEP의 VM

KEEP, “Specification document, assembler software and user manual for all layers of general-purpose virtual processors(WP4-D4.1)”, 2010, p. 3

## 다. 메타데이터와 법적 문제

### (1) 에뮬레이션을 위한 메타데이터

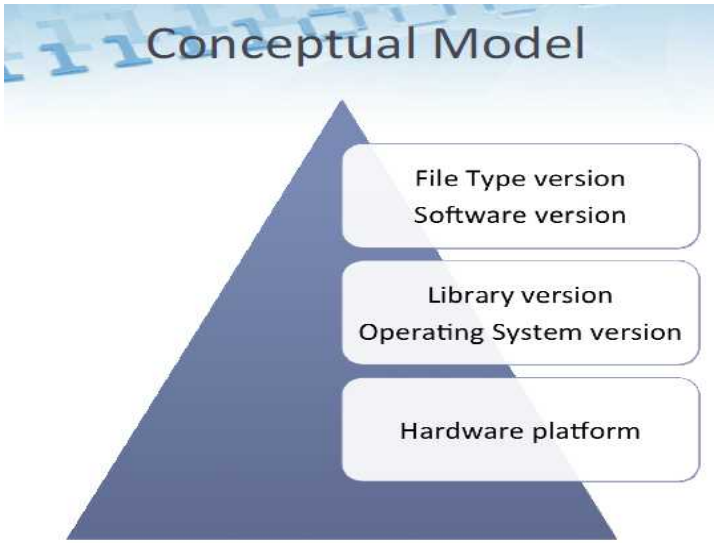
❒ KEEP에서는 현재 메타데이터의 상태와 에뮬레이션 환경을 지원하기 위한 적합성 등을 조사하는 연구를 진행하였음[59]

- ○ 이 연구를 통해 기술 메타데이터(technical metadata) 부분에서의 취약성을 발견하고 표준적인 디지털 도서관 내용(예. pdf 형식의 ‘과학 논문’)과 복잡한 디지털 오브젝트(예. 컴퓨터 게임) 양자를 포괄하는 메타데이터를 지향함[60]
- ○ 결과적으로 에뮬레이트하려고 하는 디지털 오브젝트가 있을 때 어떤 에뮬레이터를 선택의 문제를 해결하기 위해서는 ‘기술 또는 환경’에 관한 메타데이터가 필요하다고 결론지음(생성한 애플리케이션, 운영 시스템 등)
- ○ 따라서, 기술 메타데이터의 개념적 모델을 제시함(그림 참조)
- ○ 기술 환경 메타데이터란 디지털 오브젝트가 무엇이고 그래서 어떤 에뮬레이터

59) D3.1, 『Preliminary document analyzing and summarizing metadata standards and issues across Europe』; D3.2 b. 『Documents describing meta-data for the specified range of digital objects, as well as requirements and design for the browsing system and user interface of the Emulation Framework』; D3.3, 『Final document analyzing and summarizing metadata standards and issues across Europe』

60) PREMIS는 보존 목적을 위한 상위 레벨의 메타데이터 스킴을 정의하였으나, 실제적으로 에뮬레이션 보다는 마이그레이션을 지원하고 있다는 점을 발견하였음.

가 필요한지를 소프트웨어 버전, 라이브러리 버전, 운영시스템 버전, 하드웨어 버전 등의 맥락에서 서술해 주는 것을 말함



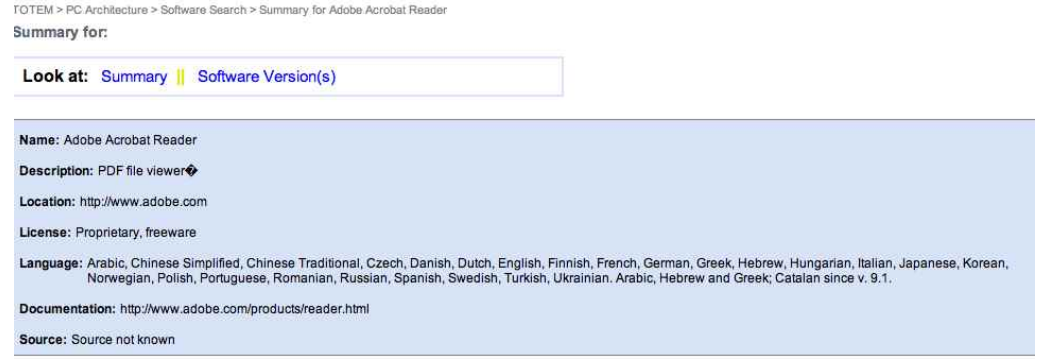

**<그림 3-36> 기술 메타데이터의 개념적 모델**

출처. Janet Delve, Leo Konstantelos, Antonio Ciuffreda, "(KEEP Training Meterial) Metadata", Cardiff Workshop, 2012. 1. 24-25, University of Portsmouth, p. 12

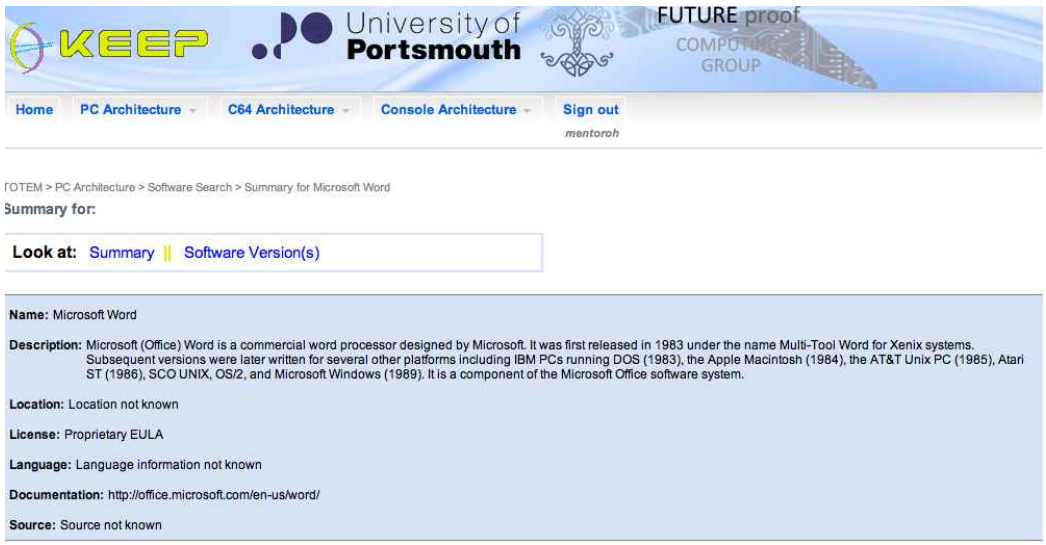
❒ 이와 관련 KEEP에서는 TOTEM이라는 이름으로 웹에서 기술 메타데이터를 담은 데이터베이스를 시범적으로 구축함

- ○ TOTEM이란 Technical Environment Metadata database의 약칭으로 기술 메타데이터에 대한 검색과 브라우징 기능을 제공
- ○ 이 데이터베이스는 현재 온라인으로 일부의 사례가 구축되어 있기 때문에 네트워크를 통해 접근 가능하며 서로 다른 이용자 역할에 따른 요구사항을 만족시킬 수 있도록 개인화되어 있음
- ○ 현재 PC x386 architecture, Commodore 64 Architecture, Cosole gaming platforms에 관한 메타데이터를 테스트 가능함. http://keep-totem.co.uk[61]

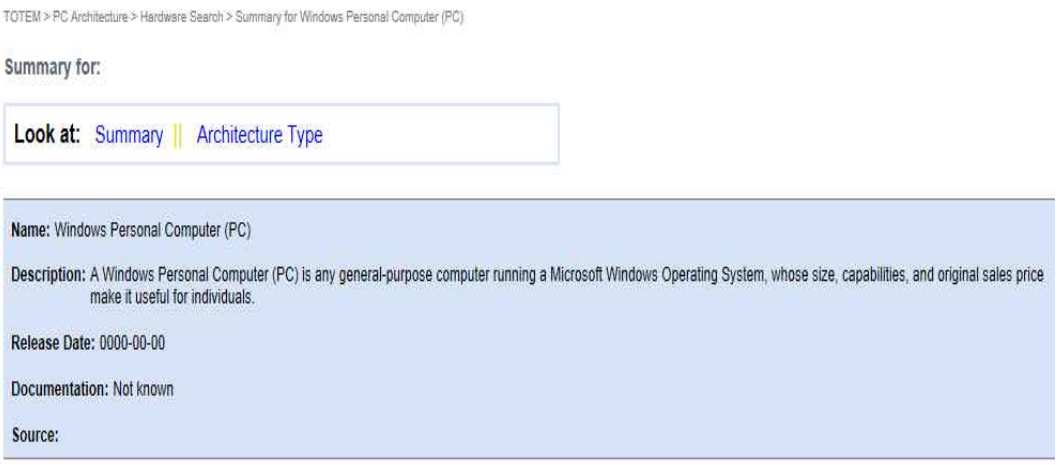
61) 아래에 TOTEM 기술정보는 웹사이트에서 임의로 검색한 결과를 담고 있음



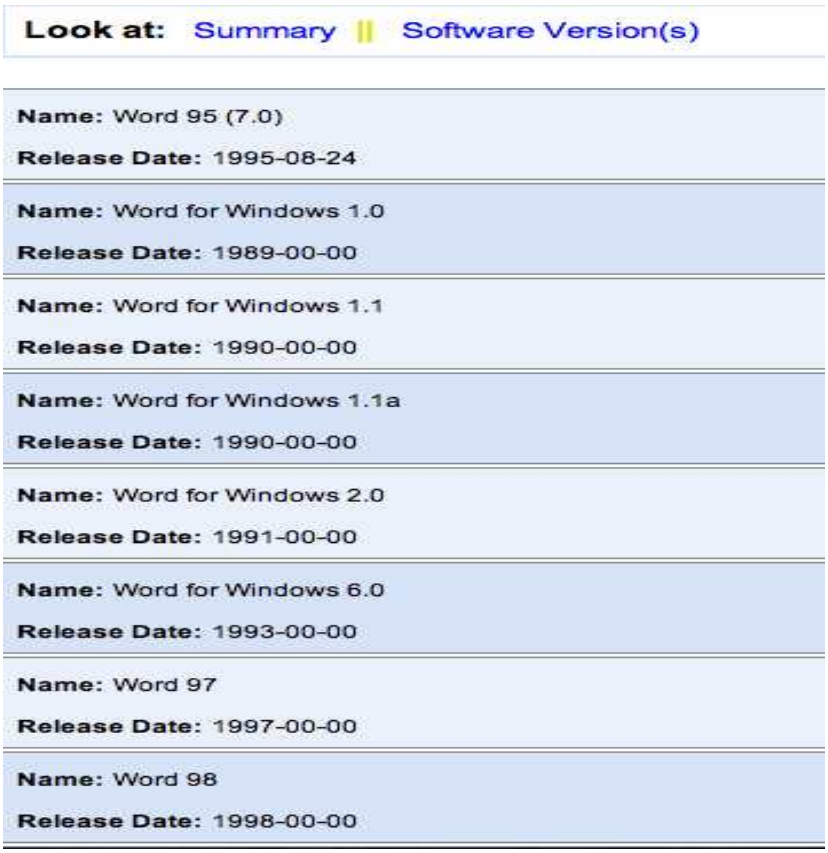

<그림 3-37> TOTEM의 PDF에 대한 요약 기술정보



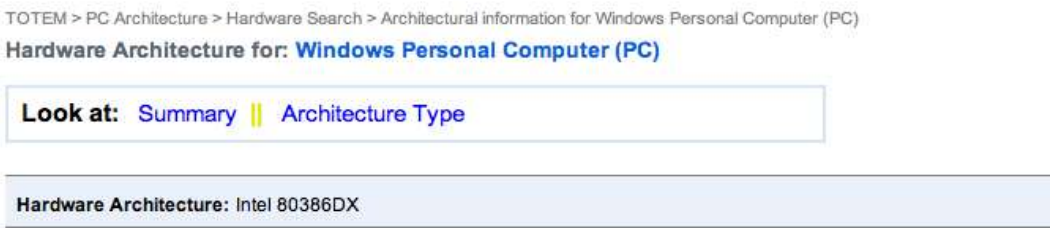

<그림 3-38> TOTEM의 MS WORD에 대한 요약 기술정보




<그림 3-39> TOTEM의 Windows Personal Computer에 대한 요약 기술정보




<그림 3-40> TOTEM의 MS WORD에 대한 버전 이력




<그림 3-41> TOTEM의 MS WORD에 대한 Architecture 기술정보

### (2) 법적 문제

❒ KEEP에서는 저작권, 프라이버시 분야의 전문가들이 참여하여 문화와 과학 정보에 대한 장기 접근 관련 법률을 검토하였음(2009-2010.3)

○ 특히 에뮬레이션과 관련하여 오래된 디지털 데이터를 담고 있는 미디어와 에뮬레이션 환경 간에 발생할 수 있는 법적인 문제를 연구

○ 에뮬레이션이 다양한 법적인 시사점을 가질 수 있지만 주요한 것으로 미디어 전송, 소프트웨어와 하드웨어의 에뮬레이션을 주요 토픽으로 다룸

- 미디어 전송: 디지털 객체 자체는 그것을 담고 있는 캐리어보다 오래가기 때문에 재생산이 필요하다는 관점
- 소프트웨어와 하드웨어의 에뮬레이션: Reverse engineering의 법적 가능 여부 등을 조사

○ 국가마다의 저작권법의 관점이 다르기 때문에 특히, 프랑스, 독일, 네덜란드의 관점에서 진행

○ 그 결과를 정리한 것이 Layman's Guide이며[62] 법적인 이슈는 KEEP 전략의 토대가 됨

❒ Layman's Guide의 주요 결론

○ 권고사항
- 관련된 법적 문제의 복잡성을 감안할 때 보존 작업 착수 전에 저작권 동의자의 동의를 확보하는 것이 바람직
- 저작권 예외 사항에 관한 권리자와 이용자 간의 지침과 라이센스 모델을 커뮤니티 단계에서 확보하는 것이 바람직(포맷 변화, 사본 가능한 수량, 교육 및 연구와 관련된 구체적인 사항 명시)

○ KEEP의 원래 고려 사항
- 통합된 구형 미디어 이관 솔루션의 개발
- 기술적 보호조치의 통과
- 에뮬레이션과 관련된 온라인 서비스 제공

○ 법적 연구에 따른 결정
- 새로운 미디어 전송 도구를 개발하지 않도록 함
- 제 3자 미디어 전송 도구의 수정 및 축소
- 에뮬레이션과 관련된 온라인 서비스 불가능

### 라. 시사점

❒ KEEP의 가장 큰 성과는 에뮬레이션에 기반을 둔 보존 전략을 세우고 그것을 지원해주는 실제적인 도구를 개발하였다는 점임

○ 이는 Planets이 장기보존의 전체적인 계획 차원에서 진행된 프로젝트였다면, KEEP 프로젝트는 에뮬레이션 자체를 핵심전략으로 구사하는 것과 관련된 사례라는 특징을 가짐

○ 특히, KEEP은 특정 유형의 디지털 객체에 있어서 에뮬레이션과 가상화라는 장기보존 전략의 유의미성을 인정한 가운데 본격적인 에뮬레이션 적용에 초점을 둔 사례임

62) D 2.6, 『layman's guide to the KEEP legal studies』

❒ KEEP에서는 에뮬레이션에 기반 한 보존전략과 그것을 지원해주는 도구로서 EF(Emulation Framework)와 TTF(Transfer Tool Framework)을 개발함

○ 오삼균 등(2009)에서는 에뮬레이션의 성공을 위해서는 다양한 문제들이 남아있다고 하면서 특히, “구형의 데이터 매체로부터 디지털 콘텐츠를 안전하게 캡처할 수 있는 방법이 필요하다. 그리고 현재 사용하고 있는 디지털 아카이빙 시스템들에 에뮬레이션을 통합할 수 있는 지식과 경험이 더 축적되어야 한다.”고 지적한바 있음.

○ 위와 관련하여 KEEP의 TTF는 바로 전자를 구현한 것이 되겠음

○ 또한, Planets과 KEEP에서는 IF(상호운용성 프레임워크)를 통하여 현재 사용하고 있는 디지털 아카이빙 시스템과의 통합을 지향하고 있기도 하다는 점에도 주목할 만함

❒ KEEP은 에뮬레이션의 시각에서 필요한 메타데이터가 무엇인지에 대한 연구개발 결과를 포함하고 있음

○ 기술 메타데이터의 개념적 모형은 장차 에뮬레이션에 필요한 기술 항목들로 DFR(Digital Format Registry)에 추가되어야 할 요소를 추출하는데 참고가 될 수 있을 것임

❒ KEEP에서 제시하고 있는 에뮬레이션 전략의 필수적인 구성요소들에 대해서는 개별 요소별로 현재화된 해석과 적용이 요구됨

○ 에뮬레이션을 성공적으로 진행하기 위해서 필요한 도구와 환경, 자동화, 소프트웨어 아카이브, 법적관점, 메타데이터 등의 이슈에 대한 판단과 적용사항, 결과물 등

### 3.1.6 사례 분석결과 및 한국형 에뮬레이션 전략

#### 가. 사례 분석결과 종합

❒ 21세기 들어 수행된 디지털 객체의 장기보존 전략으로서 에뮬레이션을 중점적으로 연구한 해외의 선진 4개 프로젝트를 분석한 결과를 종합하면 다음과 같음

❒ CAMiLEON

○ CAMiLEON 프로젝트는 디지털 객체의 장기적 보존을 위한 기술적 전략들을 광범위하게 개발하고 평가하는 과정을 통해, 상황에 따라 어떠한 장기보존 전략을 적용할 수 있을지에 대해 연구한 프로젝트로, 디지털 객체의 원래의 기능성

(functionality)과 '외형과 느낌(look and feel)'의 장기적 유지를 위한 각종 사항들을 탐구함

○ CAMiLEON 프로젝트는 디지털 객체의 장기보존 전략을 하나의 단일 전략만을 적용하는 것이 아닌, 디지털 기록의 유형 및 아카이브의 상황에 맞추어 적용할 수 있는 전략을 소프트웨어 수명의 장기화 전략, 에뮬레이션, Migration on Request란 세 가지로 구분하여 고찰함

○ 특히 CAMiLEON 프로젝트에서 제시한 Migration on Request 전략은 이용의 시점에 객체를 마이그레이션 하는 도구와 더불어 원래의 바이트스트림을 유지함으로써 보존전략이 보다 정확하고 비용 효율적이 되도록 하고 있음. 이와 더불어 에뮬레이션 전략의 경우 소프트웨어 및 하드웨어, 시스템에 대한 지적 재산권 문제의 해결이 난해함을 증명해 줌

❒ KB

○ 네덜란드의 KB 프로젝트는 디지털 객체의 장기보존 전략으로서 에뮬레이션의 다양한 가능성에 대해 연구함을 통해, 누적 에뮬레이션과 마이그레이션된 에뮬레이션, 에뮬레이션 버추얼 머신, UVC 기반 에뮬레이션 등의 방법론을 제시하였고, 개념모형인 모듈 에뮬레이션을 제안함

○ 누적 에뮬레이션의 경우 에뮬레이터가 플랫폼 환경의 일정 부분만을 상세화하기 때문에 다른 플랫폼과의 호환성이 저하되며 영속성이 적으며, 마이그레이션된 에뮬레이션은 소스코드가 미래에 생성되기 때문에 환경적 문제를 야기할 수 있으며, 컴파일러의 라이브러리는 원본과 양립하지 않을 수 있기 때문에 기능 고장이 있을 수 있는 한계를 지님. 그리고 EVM 방식은 구 플랫폼이 단일의 규정 내에서 전체적으로 정의될 수 있는지, 게다가 복합적인 해석기는 이러한 규정을 에뮬레이터로 변환하는데 필요로 되는지의 문제가 아직 해결되지 못하였고, UVC 전략의 궁극적인 성공은 부분적으로는 소프트웨어 및 컴퓨터 산업 영역에 의해 채택되는 정도에 의존한다는 점을 들고 있음

○ KB 프로젝트에서는 에뮬레이션상의 이와 같은 다양한 문제점을 해결하기 위한 방안으로, 개별 에뮬레이터와 에뮬레이션 전 과정 등을 형성하는 등의 하드웨어 아키텍처의 구성요소를 에뮬레이션한 전체 에뮬레이션인 모듈 에뮬레이션을 2007년 Dioscuri란 명칭으로 개발하였는데, 이 에뮬레이터는 Planets 프로젝트에 의해 계속 연구를 수행 중에 있음

○ 그리고 KB 프로제트에서는 텍스트, 데이터베이스, 스프레드시트의 세 유형에 대한 에뮬레이션, 마이그레이션, XML 보존전략에 대한 비교 분석을 통해, 텍스트 문서의 경우 PDF와 XML의 결합방식은 가장 효과적으로 텍스트의 장기보존을 위한 전략이며, XML은 데이터베이스의 맥락정보나 내용, 구조를 아주 잘

표현해줄 수 있으므로 현재 상태에서 데이터베이스 장기보존을 위해서는 XML이 가장 효과적인 전략으로 판단함

❒ Planets

○ Planets 프로젝트는 디지털 객체에 관한 장기보존 서비스 및 도구를 전체적으로 보여주고 있음. 보존대상을 어떻게 정하고 대상에 적절한 보존 전략을 세우기 위한 분석을 실시하고 구체적인 보존 행위들을 진행할 수 있게 관련된 도구들로 구성된 테스트베드를 제공해 주고 있음

○ 이 프로젝트를 통해 디지털 객체의 장기보존 전략 중의 하나인 에뮬레이션이 어떻게 실제 보존 전략 체계 속에서 결합되어 구현되는 것인지를 살필 수 있음. 실제로 장기보존 전략은 단일로 구현될 수 없고 해당 환경, 대상 객체에 맞게 혼합형으로 적용되어야 할 것임

○ 아울러 Planets 프로젝트는 장기보존과 관련된 핵심적인 표준인 OAIS가 어떻게 실제적으로 활용될 수 있는지를 보여주고 있음. Planets 기능모델이 OAIS의 관점에서 비교 분석된 결과물을 포함하고 있음

○ 이와 더불어 프로젝트 자체가 규모가 크고(연구비 206억원) 다양한 기관(16개)이 연계되어 있어서 이것이 단일 기관만의 문제일 수 없음을 극명하게 보여주고 있으며, 프로젝트가 종료되었지만 Open Planets Foundation을 통해서 그 결과물들이 유지 운영되고 있음으로 계속적으로 주목해볼 필요가 있음. 그리고 디지털 객체의 장기보존과 관련된 전체 서비스를 기획한다고 하였을 때 어떻게 갈 수 있을 것인지에 대한 사례를 보여주는 점 역시 벤치마킹해야 할 요소임

❒ KEEP

○ Planets 프로젝트가 장기보존의 전체적인 계획 차원에서 진행된 프로젝트였다면, KEEP 프로젝트는 에뮬레이션 자체를 핵심 전략으로 구현하는 것과 관련된 사례임. 에뮬레이션에 기반을 둔 영구적인 접근 전략을 세우는 것을 목표로 하고 있음

○ 이미 많은 연구 개발이 진행되었지만 여전히 실제적인 해결책은 제한적임. 특히 복잡한 디지털 컬렉션 예를 들어 웹사이트, 게임 그리고 다른 소프트웨어에 관한 것들은 오래된 컴퓨터 시스템을 현행적으로 이용가능하게 얼라이브 시키는 것 말고는 다른 전략은 의미 없음

○ KEEP 프로젝트에서는 에뮬레이션에 기반을 둔 보존전략과 그것을 지원해주는 도구를 개발하였는데, EF(Emulation Framework)와 TTF(Transfer Tool Framework)이며, 또한 에뮬레이션의 시각에서 필요한 메타데이터가 무엇인지에 대한 연구개발 결과를 포함하고 있음. 에뮬레이션을 위한 메타데이터는 표준적인 디지털 도서관 내용(가령 PDF 형식의 '과학 논문')과 복잡한 디지털 오

브젝트(가령 컴퓨터 게임) 양자를 포괄하는 메타데이터를 지향함. 이 결과의 일부는 웹을 통하여 제공하고 있음

❒ 한편, 지금까지 살펴본 해외사례는 에뮬레이션과 관련하여 개념과 서비스를 포괄하는 일련의 사례로서 중요한 의미를 갖고 있으나 다소 유럽연합의 사례에 국한되어 있음. 이에 본 연구에서는 국내 기록관리 영역에서 계속적으로 주목하고 있었던 미국, 영국에 대한 에뮬레이션과 관련된 장기보존 전략의 최신 동향을 포함하기로 하고(2013. 11. 6. 정기회의) 해외 기록 관리 부문에 정통한 전문가인 이상민 박사에게 해당 내용에 대한 자문을 의뢰하고 그 결과(2013. 11. 19)를 첨부하였음(8장 첨부서류. “8. 4. 추가 해외 사례 자문” 참조)

○ 미국, 영국의 경우 에뮬레이션 관련 전략이 아직까지 드러나고 있지 않은 것으로 보이며 이는 이들 국가가 장기간 장기보존 전략에 관한 연구를 진행하고 있음에도 장기보전 전략으로서 에뮬레이션 전략이 해결할 난제들이 존재하고 있음을 시사함

❒ 이상의 해외사례 분석 내용을 종합하여 다음과 같은 시사점을 도출하였음

❒ 전자기록의 장기보존 전략으로서 에뮬레이션은 비트스트림의 변화 없이 전자기록의 원형을 그대로 재현할 수 있다는 점에서, 아울러 마이그레이션처럼 전자기록 유형별로 개별적인 전략을 적용할 필요 없이 일괄적인 보존전략을 구현할 수 있다는 점에서 상당한 강점을 지닌 보존전략임

❒ 특히 현재 우리나라 전자기록의 장기보존 포맷에서는 구현해 주지 못하는 전자기록의 기능성을 재현해 줄 수 있다는 점에서, 현재 공공기관에 보유 중인 전자기록 유형 조사를 기반으로 그 실현 방안을 연구할 필요가 있음

❒ 에뮬레이션 전략에 관한 해외 4개 선진사례의 분석결과를 기반으로 볼 때, CAMiLEON 프로젝트에서 제시하는 것처럼, 디지털 객체의 장기보존 전략을 하나의 단일 전략만을 적용하는 것이 아닌, 디지털 기록의 유형 및 아카이브의 상황에 맞추어 적용할 수 있는 전략을 고려할 필요가 있음

❒ 물론 에뮬레이션 전략을 주축의 장기보존 전략으로 채택하더라도, 다양한 유형의 전자기록에 맞게 아울러 전자기록의 4대 속성을 체계적으로 구현할 수 있는 보존전략을 에뮬레이션 전략을 뒷받침하는 부수적인 보존전략으로 도입할 필요가 있음

❒ 네덜란드의 KB 프로젝트에서는 텍스트, 데이터베이스, 스프레드시트의 세 유형에

대한 에뮬레이션, 마이그레이션, XML 보존전략에 대한 비교 분석은 다양한 유형의 전자기록 보존전략 선택 시 고려해야 할 다양한 요소들을 벤치마킹하는데 유용한 연구결과를 제공해 줌

❒ 아울러 Planets 프로젝트에서 제시해주는 바대로 에뮬레이션 전략 및 방법론 개발 시, 전체 디지털 아카이빙 아키텍처 내에서 기타 프로세스들과 잘 융합되도록 개발해야 함

❒ 한편, 호주 국립기록청(NAA)에서 제시한 바와 같이, 에뮬레이션의 직접적인 도구가 되는 에뮬레이터 소프트웨어의 생산이 고도의 기술력을 필요로 하고 많은 비용이 소요되며, 사유(proprietary) 소프트웨어의 에뮬레이션은 지적 자산과 저작권 문제를 발생시켜 장기보존 전략상의 효율성과 안정성을 손상시킨다는 점은, 추후 우리나라 국가기록원에서 에뮬레이터 개발 시 충분히 고려해야 하며, 아울러 현행 우리나라 소프트웨어의 저작권법과 관련하여 그 대책 마련이 필요함

## 나. 에뮬레이션 전략의 법/비용 이슈

❒ 디지털 보존 측면에서 에뮬레이션 전략을 채택하기 위해서는 기술, 운영, 재정, 법적인 측면에서 다양한 이슈들을 고려해야 할 것인데 그 중에서 에뮬레이션을 위한 비용과 법적인 측면들은 전략의 채택과 운영에 막대한 영향을 미치고 있음

### (1) 법적 고려사항

❒ 법적 검토의 필요성

○ 해외사례 검토를 통해 에뮬레이션 전략 수립에 있어 에뮬레이션과 관련된 법적 문제들이 발생할 수 있음을 확인하였으며 이러한 문제들은 각국의 법률 환경 아래 검토될 필요성이 존재함.

○ CAMiLEON 프로젝트의 BBC Domesday 사례의 경우 원저작자와의 컨텐츠 저작권과 관련된 문제로 에뮬레이션 서비스가 불가능한 사례를 발견하였으며 KEEP 프로젝트는 소프트웨어 저작권의 문제를 유럽 법의 테두리에서 검토하고 그 결과를 “Layman’s 가이드”로 간행하였음.

○ 본 연구의 범위는 행정기관의 전자기록물의 에뮬레이션 전략을 다루고 있고 이 경우 컨텐츠의 소유권이 국가 귀속이므로 본 절에서는 에뮬레이션 서비스 과정에서 소프트웨어의 사용과 관련된 문제들에 한정하여 법적 고려사항을 검토하고자 함

❒ 법적 검토대상 내용

○ 행정기관의 전자기록물에 대한 에뮬레이션 서비스를 위해 공공기관이 소프트웨어의 장기간 보유 및 보유하고 있는 소프트웨어를 에뮬레이션 서비스에 사용하는 것과 관련된 법적 문제 및 해결방안

○ 에뮬레이션 서비스와 관련되어 현 법률 제, 개정과 관련된 문제 및 해결방안

❒ 법률 자문 추진 개요

○ '한국소프트웨어 저작권 협회'에 법률 자문을 진행하였음

- "전자기록물 재현을 위한 컴퓨터프로그램저작권 관련 문의" (2013. 10. 29. 접수, 2013. 11. 8. 법률 의견서 회신)
- 문의 내용은 저작권법 제, 개정과 관련한 질문과 에뮬레이션을 위한 서비스 내용을 포함한 프로젝트팀 시나리오 관련 질문으로 구분하여 진행하였음

  - 세부 내용은 <8. 1. 나. 한국소프트웨어 저작권 협회 의견서> 참조

○ '한국저작권위원회' 에 법률 상담을 진행하였음

- 이 기관의 경우 무료 법률 상담을 제공하고 있었음(변호사 자격증을 가진 전문 직원), 상담 결과이기 때문에 증빙성 및 공증성을 갖기는 어려우나 참고할 만한 것으로 판단하여 직접 방문하여 상담을 진행하였음(2013. 11. 5)
- 문의 내용은 동일함

  - 세부 내용은 <8. 1. 다. 한국저작권위원회 상담결과> 참조

❒ 법률 자문 검토 결과

○ 에뮬레이션 서비스 내용

<표 3-4> 에뮬레이션 서비스 내용관련 사항

| 서비스 내용 | 법률 의견 |
|---|---|
| 소프트웨어의 장기간 보유 | -기간제한 없는 라이선스 계약을 체결한 경우 전혀 문제 없음<br>-구입한 소프트웨어는 '국유재산"에 해당되므로 관련법에 우선될 수 있음 |
| 에뮬레이션 서비스를 위해 구형 소프트웨어의 재설치, 사용 | -소프트웨어 라이선스란 라이선스 계약에 의해 소프트웨어를 사용할 권리를 허락한 경우이므로 계약이 종료되면 권리도 소멸되는 것임<br>-라이선스 계약 종료 후 저작사의 허락 없이 재설치·사용하는 것은 저작권법에 저촉됨(법 제 14조 이용허락 조항 위반, 법 제 136조 제 1항 복제권 침해)<br>-패키지 구입 시 영구적인 사용권한은 일반적으로 구입한 버전에 한정되는 것이며 구 버전에 대한 사용권은 상세 해석 필요(해당 소프트웨어 제작사와 협의 필요)<br>-소프트웨어의 반출은 저작자의 대여권을 침해(법 제 136조 제 1항 제 1호) |
| 대국민 대상의 에뮬레이션 서비스 제공 | -공적인 사용의 경우 법이 규정하는 저작재산권 제한 사유에 해당되어야 가능함(대민서비스의 목적과 범위가 광범위하기 때문에 합당한 사유가 필요함)<br>-공적인 경우라도 임의로 구 버전의 소프트웨어를 재설치 사용하거나 반출한 경우 당사자 사이의 계약 위반으로 민사상 손해배상 책임을 질 수 있음(법 제 125조) |

○ 저작권법 제, 개정 관련

<표 3-5> 저작권 법 제·개정 관련 사항

| 에뮬레이션 관련 조항 | 법률 의견 |
|---|---|
| 지적 재산권 제한 사유로서 '기록물관리기관'이 도서관 등등과 같이 복제를 허용하는 시설로서 포함될 수는 없는가 | -도서관 등의 저작재산권 제한사유는 가장 강력한 저작재산권 제한 사유 중 하나로서 기록관리법상 그 적용대상이 "공공기관" 한정되는 기록물을 저작권법 제31조의 "도서관 등"에 포함시키기는 어려울 것으로 사료됨 |
| 기록 열람을 위해 필요한 프로그램으로서 기록을 디스플 | -저작권법상 도서를 열람하는 과정에서 필연적으로 발생하는 프로그램의 설치 및 재생의 경우 그 기능을 하 |

| | |
|---|---|
| 레이해주는 프로그램에 대해 저작권 제한 조항이 추가될 수는 없는가 | 는 프로그램과 기록 등을 디스플레이해 줄 수 있는 프로그램의 성격은 전혀 다름. |
| 절판된 구형 프로그램에 한하여 디지털 납본 제도 등을 도입할 수 있는 방법은 무엇인가 | -도서관법에 의한 납본 제도에서도 디지털 납본은 아직까지 의무화되어 있지 않음. 또한 기증은 사적 자치에 속하여 정부차원에서 강제하기 어려움. 따라서 우선적으로 공공 필요를 입증하는 것이 선행되어야 할 것임<br>-이와 관련하여 1) 전자기록물 재현을 위한 목표 달성에 적합한 수단에 해당하고 2) 이와 같은 목적 달성을 위해 디지털 납본 제도를 도입하는 것이 가장 적은 침해를 가져오는 수단이고 3) 동 제도를 도입함으로 인하여 얻어지는 공익과 이로 인하여 침해되는 저작사의 사익(재산권 침해)을 비교 형량하여 공익이 더 크다는 것을 입증해야 할 것임<br>-현재 판매하지 않고 있어 구입이 불가능한 프로그램의 경우에 한하여 문제되고 있는 당해 기록을 재현할 목적으로만 한정하는 것이 바람직함 |

○ 시사점

- 행정기관의 에뮬레이션 서비스와 관련된 소프트웨어 저작권과 관련된 문제 해결을 위해서는 우선적으로 에뮬레이션 서비스의 목적과 대상이 되는 디지털 컴포넌트 및 해당되는 소프트웨어를 명확히 하여 제한된 상황을 기반으로 법적 세부사항을 검토할 필요가 있음
- 지적 재산권의 제한사항이 저작권법에 존재하고 있지만 기본적으로 공공목적으로 인해 최소한의 경우에 대한 제한적 허용 방침임. 따라서 에뮬레이션 서비스의 대상이 되는 소프트웨어의 범위(버전, 종류 등)를 명확히 할 필요성이 존재함
- 이를 바탕으로 에뮬레이션 서비스 전략을 수립 시에 소프트웨어 저작사와 사전적으로 법적 문제를 협의하고 구체화 해야 할 것임
- 특히, 협의 대상이 되는 소프트웨어 저작사마다 소프트웨어 라이선스 정책이 상이하기 때문에 에뮬레이션 전략 수립 시에 사용권한, 소프트웨어의 설치 및 사용, 대중 서비스, 필요시 Reverse Engineering 등에 관한 사항을 구체적으로 협의해야 할 것임
- 아울러 문화유산 기관으로서 기록물관리기관이 기록을 에뮬레이션 서비스함에 있어 필요한 정당성을 확보하기 위한 제반 노력이 요구되어야 할 것임

- 도서관 등의 저작재산권 제한사유는 가장 강력한 저작재산권 제한 사유 중 하나임. 기록물 역시 개인이나 조직이 활동이나 업무과정에서 생성, 접수한 것으로 인류의 문화유산의 일부로서 도서 등과 견주어 뒤지지 않을 것임
- 그러나 현재 기록 관리법의 적용대상이 공공기관에 한정되고 이는 저작권법 제 2조 1호에서 규정되는 저작물과 그 의미와 범위에 있어서 상당한 차이가 존재하고 있는 것으로 해석되고 있음
- 따라서 당장 저작권법 개정을 통해 기록을 포함시키는 등 지적 재산권 제한 조항 추가의 가능성은 극히 낮음

### (2) 비용 고려사항

❒ 에뮬레이션을 위한 비용 분석 개요

○ 에뮬레이션은 비트스트림의 변화 없이 원본적 속성을 있는 그대로 재현해준다는 점에서 강점을 지닌 장기보존 전략임

○ 하지만 고도의 기술력 및 개발비용이 필요하다는 점에서, 에뮬레이션을 장기보존 전략으로 채택할 경우 발생하게 되는 비용에 대한 분석이 반드시 수행되어야 함

○ 다만 실제 에뮬레이션 수행에 소요되는 비용은 해당 전자기록의 유형 및 수량, 기관의 사정, 컴퓨터 기술력 수준 등 수많은 변수가 존재하게 됨

○ 이에 여기서는 그동안 에뮬레이션 전략 수행에 소요되는 비용을 추산한 해외 사례를 고찰한 다음, 우리나라의 적용시 소요되는 비용을 하나의 가안으로 제시하고자 함

❒ **마이그레이션 vs. 에뮬레이션 비용 분석**[63]

○ 에뮬레이션 비용 산출

- 에뮬레이션에 소요되는 비용은 다음과 같음

• one-time cost: 에뮬레이션 장치의 개발 비용
• recurring costs: 원래 하드웨어 플랫폼의 요소를 위한 에뮬레이터 개발
• access time: 에뮬레이터와 적절한 소프트웨어 환경 운영

- 에뮬레이션 장치 개발 비용은 UVC의 데모버전 생산과 관련됨. 이는 2004년 IBM에 의해 수행되었으며, 연구와 디자인을 포함해 주당 40시간씩 32주가 소요되었음
- 시간당 $120이라고 가정할 경우 one-time cost 비용은 약 $150,000이 되며,

63) Erik Oltmans, Nanda Kol, "A Comparison between Migration and Emulation in Terms of Costs", RLG DigiNews 9(2), 2005. <http://worldcat.org/arcviewer/2/OCC/2009/08/11/H1250012115408/viewer/file2.html>

실질적 비용은 160명이 참가하여 시간당 약 $20,000이러 할 수 있음
- 에뮬레이션 장치개발 이후 관리가 요구되며 관리 비용은 상대적으로 개발비용보다 적은데, 연간 $2,00-3,000이 소요됨
- 하지만 일부에서는 에뮬레이션 장치 개발비용이 디지털보존 저장소와의 협력을 통해 절약할 수 있다고 봄. 즉 동일 플랫폼을 사용하는 경우 동일한 에뮬레이터를 사용할 수 있기 때문임

○ 마이그레이션 비용 산출
- 마이그레이션은 많은 컨버전 도구가 이용가능하기 때문에 상대적으로 비용이 저렴함
- 그러나 마이그레이션은 아카이브즈에 소장된 전체 컬렉션에 적용되어야 하며, 이는 아카이브가 커질수록 비용이 증가하게 됨

○ 에뮬레이션과 마이그레이션의 비용모델 공식 적용
- 양 보존전략의 비용을 비교하기 위해 Shenton이 제안한 비용모델 공식을 이용

• 비전자단행본 보존비용

K(t)=s + a + c+ p1 +h1 + p(t) + h(t)
K(t) : t year동안 자료 소장의 전체 비용
s : 선정
a : 접근
c : 목록
p1 : 초기 보존
h1 : 초기 핸들링
p(t) : 장기보존
h(t) : 저장

• 전자출판물 보존비용

K(t,a)=s(a) + I(a) + h(t,a)
K(t,a) : t year 동안 개체 소장의 총 비용
s : 선정[64]
I : 인제스트[65]
h : 저장[66]

- 위의 공식은 아래의 두 가지 공식을 위한 기초이며, 아래의 공식에는 선정과 인제스트 비용을 포함하지 않음. 왜냐하면 선정과 인제스트는 에뮬레이션과 마이그레이션 모두에 동일해서 비용 비교에 영향을 주지 않기 때문이며, 이러한 비용은 출판사와의 아카이빙 협약에 의존하고, 소프트웨어의 형태에 의존하는 관계상 비용 산출이 실질적으로 어렵기 때문임

- 마이그레이션 비용산출 K(t,a)=h(t,a) + m(t,a)

| |
|---|
| K(t,a)[67] : t years 동안 개체 소장의 총 비용<br>h : 저장비용<br>m : 마이그레이션 비용 |

- 에뮬레이션 비용산출 K(t,a)=h(t,a) + E + e(t)[68]

| |
|---|
| K(t,a) : t years 동안 개체 소장의 총 비용<br>h : 저장 비용<br>E : emulation virtual machine 세팅 비용<br>e(t) : 시간에 따른 에뮬레이션 비용 |

○ 마이그레이션과 에뮬레이션에 관한 비용은 문헌상으로는 이용할 수 없어, 몇 가지 가정을 기초로 두 가지 비용을 비교함

- 개체 수는 매년 증가
- 연간 개체 당 $0.05로 저장비용이 요구
- 두 전략의 저장 공간은 동일
- 마이그레이션의 평균비용은 연간 개체 당 $0.10임
- 에뮬레이션 장치 개발비용은 $200,000임
- 새 개체에 대한 에뮬레이션 장치의 관리와 에뮬레이션 도구 개발 평균 비용은 연간 $30.000임
- 에뮬레이터 운영비용은 제외

64) 개체 획득, 추가 처리를 위한 준비
65) 자동처리가 포함되며, 이용가능한 포맷으로의 메타데이터 변환, 저장시스템에 디지털개체의 저장 포함
66) 저장매체의 구입, 매체 리프래쉬, 데이터베이스 관리시스템 유지 등이 포함
67) 디지털 객체의 마이그레이션 비용은 시간 t와 개체 수 a에 의존. 즉 객체의 보존시간이 길수록 컨버젼이 자주 발생하며, 객체가 많을수록 많은 컨버젼이 수행되어야 함
68) 에뮬레이션 장치 개발비는 E, 시간에 따른 에뮬레이터 관리비용은 e임. 관리 비용과 에뮬레이션도구 개발비용은 개체의 수와는 독립적이지만, 에뮬레이션 도구는 시간에 따라 관리되어야 하기 때문에 관리비용은 시간에 의존하게 됨

○ 결과적으로 50년간 1,000,000개 객체를 보존하는 비용을 계산한 결과 마이그레이션 비용이 더 높았음
○ 장서의 크기가 마이그레이션 비용을 상승시키기 때문에 아카이브에 개체 수가 커질수록 마이그레이션 비용은 상승하게 됨

### ❐ Digital Preservation Testbed의 에뮬레이션 비용 분석[69]

○ 비용 분석을 위한 가정
- 다음과 같은 6개 부류의 직원이 직원당 연간 1620시간의 일을 수행
  - 관리지원(1): 시간당 EUR14
  - 아키비스트/레코드매니저(2): 시간당 EUR25
  - 슈퍼바이저(3): 시간당 EUR32
  - 데이터관리지원(4): 시간당 EUR18
  - 프로그래머(5): 시간당 EUR25
  - 상급 IT 지원(6): EUR32

- 아카이브와 보존시스템은 4가지 공간을 요구함.
  - 디지털아카이브 시스템 공간: 1FTE; EUR1,897,400(1년), 이후 EUR632,403 ; 저장공간 100Tb
  - 사무공간: 1.4FTE ; EUR7,400(1년) 이후 EUR2,466
  - 개발과 테스트 공간: 3FTE; EUR50,200(1년) 이후 EUR16,731 이 공간은 부가적 소프트웨어와 하드웨어로 구성됨. 이 시설은 디지털 보존도구의 개발을 위한 시설임
  - 디지털보존시스템을 위한 공간: 3FTE ; EUR279,600(1년) 이후 93,190; 수령, 변환, 테스팅을 위한 공간. 장비는 시간당 10Tb 레코드를 저장할 수 있음. 필요시 디지털아카이브를 위한 공간과 결합될 수 있음
  - 시설은 전체 100Tb 레코드를 관리할 수 있어야 함. 이는 1000Tb (1Pb)이상으로 확장될 수 있음
  - 이 시설은 연간 전자메일 40 배치-batch(각 배치당 2000개메일) 20 텍스트문서의 배치, 스프레드시트 20 배치, 20 데이터베이스의 수집과 변형을 다룰 수 있음
  - 위의 배치처리는 총 5Gb의 저장공간을 요구함(50kb/전자메일당, 100kb/텍스트문서당, 250kb/스프레드시트당, 2Mb/데이터베이스당 계산시)

○ 문서의 생성
- 생산된 지 0년 2년 10년된 레코드의 repair의 비용은 오래될수록 repair 비용

69) Digital Preservation Testbed, *From Digital Volatility to Digital Permanence: Preserving Databases*(version 1.0), Dutch National Archives, The Hague, December 2003. <부록 3>

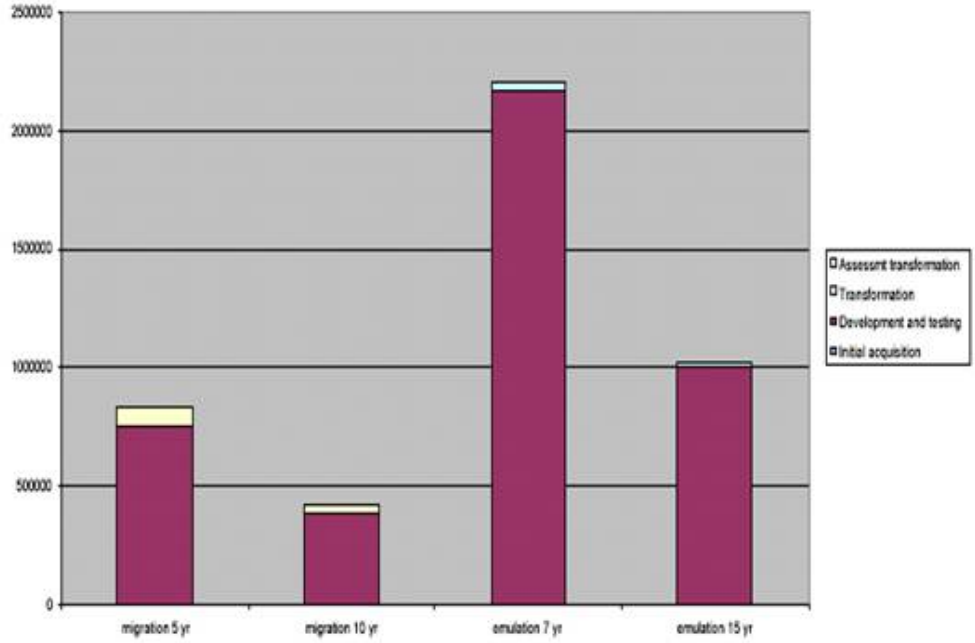
이 증가됨을 알 수 있음

- 0년 : EUR0.33 / 2년 : EUR0.67 / 10년 : EUR10
- XML 전자메일이 메타데이터 생성에 비용절감을 가져옴(XML이 구조화되었기 때문). 일반메일 : EUR1.41, XML 메일 : EUR0.06

○ 비용 분석 결과

- 비용 산출에 소요되는 전제는 다음과 같음
  - 마이그레이션은 매 5년이나 10년마다 필요
  - 에뮬레이터는 7년이나 15년마다 필요
  - 진본성요구가 규명되면, 요구사항은 재평가될 필요가 없음
  - 추가된다면 메타데이터는 더 수정이 필요하지 않음
  - 에뮬레이터 개발은 3명이 1.75년을 노력해야 함

- 전자메일, 텍스트, 스프레드시트, 데이터베이스를 포함한 레코드 수집의 유지를 위한 비용을 비교하면, 아래의 그림은 100년 동안의 전자메일, 텍스트, 스프레드시트, 데이터베이스의 레코드 집합의 관리비용을 비교함
- 결과적으로 에뮬레이션에 소요되는 비용이 마이그레이션에 비해 상대적으로 고비용으로 나타남
- 이에 Digital Preservation Testbed에서는 텍스트 문서의 경우 PDF와 XML의 결합방식이 가장 효과적으로 텍스트의 장기보존을 위한 전략이며, XML은 데이터베이스의 맥락정보나 내용, 구조를 아주 잘 표현해줄 수 있으므로 현재 상태에서 데이터베이스 장기보존을 위해서는 XML이 가장 효과적인 전략으로 판단하고 있음




<그림 3-42> 디지털 장기보존 전략별 보존 비용비교

❒ 네덜란드 국립기록청의 에뮬레이션 비용 산출[70]

○ 네덜란드 국립기록청에서는 전자메일, 텍스트, 스프레드시트, 데이터베이스란 4유형의 전자기록을 대상으로, 다양한 보존전략에서 소요되는 비용을 다음과 같이 산출함

70) Nationaal Archief, "Digital Preservation Cost Model", 2005.<http://www.nationaalarchief.nl/kennisbank/digital-preservation-cost-model-2005>

## <표 3-6> 디지털 장기보존 비용 모델

| Digital Preservation Costmodel | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Version 1.2 | | | | | | | | | | | | | |
| 30-Aug-05 | | | | | | | | | | | | | |
| 시트3 비용계산. 이 모델은 IT와 기록 보존의 문제에 주의하여 디지털 보존에 익숙한 숙련된 팀과 전용 보관 및 저장시설이 있다는 것을 전제로 하고있다. 인원의 유형은 다음과 같다. 1. 관리보조원(admin asst); 2. 기록관리자(record keeper) ; 3. 슈퍼바이저(supervisor); 4 데이터 입력 운영자(data entry operator); 5. 소프트웨어 엔지니어(software engineer); 6. 정보통신기술 수석. 이페이지는 행위당 비용, 일괄처리등에 초점을 맞추고 있다. | | | | | | | | | | | | |
| 기록유형별 보존방법 | | | 가존의 전자메일을 XML으로 | 새로운 전자메일을 XML으로 | 기존의 텍스트(마이그레이션) | 기존의 텍스트를 XML으로 | 새로운 텍스트를 XML으로 | 기존의 스프레드시트(마이그레이션) | 기존의 스프레드시트를 XML으로 | 기존의 스프레드시트(UVC) | 새로운 스프레드시트를 XML으로 | 기존의 데이터베이스(마이그레이션) | 기존의 데이터베이스를 XML으로 | 새로운 데이터베이스를 XML으로 |
| 일괄 처리 사이즈 | | | 2000 | 2000 | 200 | 200 | 200 | 20 | 20 | 20 | 20 | 1 | 1 | 1 |
| 일괄처리 횟수 / 년 | | | 20 | 20 | 10 | 10 | 10 | 5 | 5 | 5 | 5 | 2 | 2 | 2 |
| 기록관리활동 | From Cost Basis sheet | | | | | | | | | | | | | |
| 선별 및 평가 ( PT 2) | 31.25 | costs/batch | 125.00 | 125.00 | 125.00 | 125.00 | 125.00 | 125.00 | 125.00 | 125.00 | 125.00 | 125.00 | 125.00 | 125.00 |
| 메타데이터 입력 (PT 2 and 4) | 26.88 | costs/item | 0.90 | 0.00 | 0.90 | 0.90 | 0.00 | 2.24 | 2.24 | 2.24 | 0.00 | 26.88 | 26.88 | 0.00 |
| 기록복원 (PT 4 and 5) | 26.88 | costs/record to be repaired | 13.44 | 0.00 | 26.88 | 26.88 | 0.00 | 107.50 | 107.50 | 107.50 | 0.00 | 215.00 | 215.00 | 0.00 |
| *일괄처리시 기록관리의 유로화* | | | 4,604.17 | 125.00 | 841.67 | 841.67 | 125.00 | 384.79 | 384.79 | 384.79 | 125.00 | 173.38 | 173.38 | 125.00 |
| *건당 기록유지시 유로화 비용* | | | 2.30 | 0.06250 | 4.21 | 4.21 | 0.63 | 19.24 | 19.24 | 19.24 | 6.25 | 173.38 | 173.38 | 125.00 |
| 보존방법의 개발 | | | | | | | | | | | | | | |
| 요구사항의 수집 (PT 2와 6) | 35.63 | costs per record type | 285.00 | 0.00 | 285.00 | 285.00 | 0.00 | 285.00 | 285.00 | 285.00 | 0.00 | 570.00 | 570.00 | 0.00 |
| 방법의 개발 (PT5와 6) | 35.63 | costs per record type | 4275.00 | 0.00 | 2850.00 | 5700.00 | 0.00 | 2850.00 | 4275.00 | 71250.00 | 0.00 | 7125.00 | 11400.00 | 0.00 |
| Test approach (PT 2 and 4) | 26.88 | costs per record type | 322.50 | 0.00 | 215.00 | 430.00 | 0.00 | 215.00 | 322.50 | 5375.00 | 0.00 | 537.50 | 860.00 | 0.00 |
| *기록유형별 연간 비용* | | | 4,882.50 | 0.00 | 3,350.00 | 6,415.00 | 0.00 | 3,350.00 | 4,882.50 | 76,910.00 | 0.00 | 8,232.50 | 12,830.00 | 0.00 |
| *건당 비용* | | | 0.1221 | 0.00 | 1.68 | 3.21 | 0.00 | 33.50 | 48.83 | 769.10 | 0.00 | 4,116.25 | 6,415.00 | 0.00 |
| 디지털 보존 활동 | 시간당 유로화 비용 | | | | | | | | | | | | | |
| 보존활동 (PT | 22.50 | 비용/일괄처리 | 45.00 | 0.00 | 45.00 | 45.00 | 0.00 | 45.00 | 45.00 | 45.00 | 0.00 | 180.00 | 180.00 | 0.00 |
| 평가 (PT2) | 31.25 | 비용/ 일괄처리 | 125.00 | 0.00 | 250.00 | 250.00 | 0.00 | 250.00 | 250.00 | 250.00 | 0.00 | 500.00 | 500.00 | 0.00 |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *일괄처리시 유로화 보존비용* | | | 170.00 | 0.00 | 295.00 | 295.00 | 0.00 | 295.00 | 295.00 | 295.00 | 0.00 | 680.00 | 680.00 | 0.00 |
| *건당 보존시 유로화 비용* | | | 0.0850 | 0.00 | 1.48 | 1.48 | 0.00 | 14.75 | 14.75 | 14.75 | 0.00 | 680.00 | 680.00 | 0.00 |
| | | | | | | | | | | | | | | |
| 일괄처리시 PT별 총 시간 | | | | | | | | | | | | | | |
| PT 2 시간 | | | 1307.29 | 125.00 | 504.17 | 516.67 | 125.00 | 451.04 | 463.54 | 1051.04 | 125.00 | 921.88 | 1015.63 | 125.00 |
| PT 4 시간 | | | 1926.75 | 0.00 | 354.00 | 363.00 | 0.00 | 171.75 | 180.75 | 603.75 | 0.00 | 312.75 | 360.25 | 0.00 |
| PT 5 시간 | | | 1656.25 | 0.00 | 437.50 | 562.50 | 0.00 | 375.00 | 500.00 | 5375.00 | 0.00 | 1575.00 | 2512.50 | 0.00 |
| PT 6 시간 | | | 128.00 | 0.00 | 176.00 | 336.00 | 0.00 | 352.00 | 512.00 | 8032.00 | 0.00 | 2160.00 | 3360.00 | 0.00 |
| *일괄처리시 유로화 보존비용* | | | 5018.29 | 125.00 | 1471.67 | 1778.17 | 125.00 | 1349.79 | 1656.29 | 16061.79 | 125.00 | 4969.63 | 7268.38 | 125.00 |
| *건당 보존시 유로화 비용* | | | 2.5091 | 0.062500 | 7.36 | 8.89 | 0.63 | 67.49 | 82.81 | 803.09 | 6.25 | 4969.63 | 7268.38 | 125.00 |
| | | | | | | | | | | | | | | |
| PT별 연간 총 시간 | | | | | | | | | | | | | | |
| PT 2 총 시간 | | | 26,145.83 | 2,500.00 | 5,041.67 | 5,166.67 | 1,250.00 | 2,255.21 | 2,317.71 | 5,255.21 | 625.00 | 1,843.75 | 2,031.25 | 250.00 |
| PT 4 총시간 | | | 38,535.00 | 0.00 | 3,540.00 | 3,630.00 | 0.00 | 858.75 | 903.75 | 3,018.75 | 0.00 | 625.50 | 760.50 | 0.00 |
| PT 5 총 시간 | | | 33,125.00 | 0.00 | 4,375.00 | 5,625.00 | 0.00 | 1,875.00 | 2,500.00 | 31,875.00 | 0.00 | 3,150.00 | 5,025.00 | 0.00 |
| PT 6 총 시간 | | | 2,560.00 | 0.00 | 1,760.00 | 3,360.00 | 0.00 | 1,760.00 | 2,560.00 | 40,160.00 | 0.00 | 4,320.00 | 6,720.00 | 0.00 |
| Total costs for PT 1 | | | 12,856.67 | 280.00 | 1,759.33 | 2,067.33 | 140.00 | 750.17 | 904.17 | 8,142.17 | 70.00 | 1,034.60 | 1,496.60 | 28.00 |
| Total costs for PT 3 | | | 29,386.67 | 640.00 | 4,021.33 | 4,725.33 | 320.00 | 1,714.67 | 2,066.67 | 18,610.67 | 160.00 | 2,364.80 | 3,420.80 | 64.00 |
| *Total costs per year* | | | 142,609.17 | 3,420.00 | 20,497.33 | 24,574.33 | 1,710.00 | 9,213.79 | 11,252.29 | 107,061.79 | 855.00 | 13,338.65 | 19,454.15 | 342.00 |
| | | | | | | | | | | | | | | |
| N년의 기간동안 PT별 총 시간 | 20 | M년후 반복 | 10 | 10 | 3 | 10 | 10 | 3 | 10 | 20 | 10 | 3 | 10 | 10 |
| PT 2 총 시간 | | | 784,375.00 | 311,458.33 | 388,208.33 | 155,000.00 | 76,666.67 | 173,651.04 | 69,531.25 | 105,104.17 | 35,677.08 | 141,968.75 | 60,937.50 | 25,312.50 |
| PT 4 총시간 | | | 1,156,050.00 | 385,350.00 | 272,580.00 | 108,900.00 | 36,300.00 | 66,123.75 | 27,112.50 | 60,375.00 | 9,037.50 | 48,163.50 | 22,815.00 | 7,605.00 |
| PT 5 총 시간 | | | 993,750.00 | 331,250.00 | 336,875.00 | 168,750.00 | 56,250.00 | 144,375.00 | 75,000.00 | 637,500.00 | 25,000.00 | 242,550.00 | 150,750.00 | 50,250.00 |
| PT 6 총 시간 | | | 76,800.00 | 25,600.00 | 135,520.00 | 100,800.00 | 33,600.00 | 135,520.00 | 76,800.00 | 803,200.00 | 25,600.00 | 332,640.00 | 201,600.00 | 67,200.00 |
| Total costs for PT 1 | | | 385,700.00 | 134,166.67 | 135,466.67 | 62,020.00 | 23,473.33 | 57,762.83 | 27,125.00 | 162,843.33 | 10,441.67 | 79,664.20 | 44,898.00 | 15,526.00 |
| Total costs for PT 3 | | | 881,600.00 | 306,666.67 | 309,642.67 | 141,760.00 | 53,653.33 | 132,029.33 | 62,000.00 | 372,213.33 | 23,866.67 | 182,089.60 | 102,624.00 | 35,486.00 |
| *Total costs per N year* | | | 4,278,275.00 | 1,494,491.67 | 1,576,294.67 | 737,230.00 | 279,943.33 | 709,461.96 | 337,568.75 | 2,141,235.83 | 129,622.92 | 1,027,076.05 | 583,624.50 | 201,381.50 |

○ 비용기준

- 기록의 보관을 위한 직원과 시설 및 보존시설의 자본을 계산
- 인건비, 사무실공간비용, 하드웨어 및 소프트웨어 비용을 유로화 기준으로 전체적인 비용을 계산
- 인건비의 경우 다음과 같이 6개의 직원 유형으로 구분
  - 관리보조원(PT 1)
  - 기록관리자(PT 2)
  - 슈퍼바이저(PT 3)
  - 데이터입력 운영자(PT 4)
  - 소프트웨어 엔지니어 (PT 5)
  - 수석 IT직원(PT 6)

- 네덜란드의 경우 일일근무시간 8시간, 주간 근무시간 40시간, 연간근무시간을 1620시간으로 설정하고 만근시 25%의 추가 급여를 지급함
- 사무실 공간 비용의 경우 직원 유형에 따라 각기 다른 3가지 유형으로 구분
  - 개발 및 테스트 공간
  - 보존사무실
  - 아카이브 사무실

- 하드웨어와 소프트웨어 비용의 경우 크게 10가지 유형으로 구분
  - 데스크탑
  - 노트북
  - 오피스 소프트웨어
  - 프로그래밍 소프트웨어
  - 서버
  - 데이터베이스 소프트웨어
  - 아카이브 시스템
  - 스토리지 시스템
  - 스토리지
  - 주크박스

- 하드웨어의 경우 구입하는 첫해 이후 년간 33%의 감가상각비용이 추가함으로써 2년차부터 하드웨어와 소프트웨어의 비용이 크게 감소

○ 시간 계산

- 시간의 계산 시트에서는 기록의 총 4개의 기록유형별 보존방법 및 수행하는

기록관리 행위의 시간을 계산함

- 기록의 유형별 보존방법은 다음과 같음
  - 전자메일: 기존의 전자메일을 XML으로/새로운 전자메일을 XML으로
  - 텍스트: 기존의 텍스트(마이그레이션)/기존의 텍스트를 XML으로/새로운 텍스트를 XML으로
  - 스프레드시트: 기존의 스프레드시트(마이그레이션)/기존의 스프레드시트를 XML으로/기존의 스프레드시트(UVC)/새로운 스프레드시트를 XML으로
  - 데이터베이스: 기존의 데이터베이스(마이그레이션)/기존의 데이터베이스를 XML으로/새로운 데이터베이스를 XML으로

- 보존행위와 일괄처리에 요구되는 시간에 중점을 두고 있으며 보존행위는 다음과 같음
  - 기록관리 활동: 선별 및 평가, 메타데이터 입력, 기록복원, PT2 일괄처리 시간, PT4 일괄처리 시간, PT5 일괄처리 시간
  - 보존방법의 개발: 요구사항의 수집, 방법의 개발, 테스트 방법, PT2 일괄처리 시간, PT4 일괄처리 시간, PT5 일괄처리 시간, PT6 일괄처리 시간
  - 디지털 보존활동: 보존활동, 평가, PT 2 일괄처리 시간, PT 4 일괄처리 시간

○ 전자메일, 텍스트, 스프레드시트, 데이터베이스 중 에뮬레이션 전략을 사용하는 대상은 스프레드 시트임(IBM사의 UVC를 이용한 에뮬레이션)

- 스프레드시트는 기록관리 활동의 경우 다른 기록유형들과 약소한 차이를 시간계산을 보여주고 있으나, 보존방법의 개발의 경우 여타 다른 기록유형에 비해 PT 5. 소프트웨어 엔지니어(1000시간), PT6. 정보통신 기술수석(1004시간)이 많은 시간을 개발에 투자하는 것을 알 수 있음
- 하지만 한번 개발 시 컴퓨팅 환경의 변화로 인한 에뮬레이터의 업데이트가 되지 않는 이상 추가적인 시간의 계산이 필요 없는 에뮬레이션 전략의 특성으로 인하여 20년에 한번 씩 에뮬레이터를 개발하였을 개발 시간에 비해 낮은 시간이 소요되는 것을 확인 할 수 있으나, 마이그레이션 및 XML 생산 및 마이그레이션 전략에 비해 많은 시간을 필요로 하는 것을 알 수 있음

○ 비용 계산

- 에뮬레이션 전략의 경우 개발에 많은 시간이 소요되기 때문에 장기보존에 있어 많은 비용을 요구함. 특히 소프트웨어 엔지니어(PT.5)와 정보통신기술수석(PT. 6)의 연봉이 다른 직원의 유형에 비해 많이 높은 것을 감안 할 때 에뮬레이션 외의 전략과는 많은 차이를 보여주고 있음
  - 보존방법의 개발의 경우 XML스프레드시트(0), 기존의 스프레드 시트 마이그

레이션(3.350.00), 기존의 스프레드 시트를 XML으로(4.882.50), 기존의 스프레드 시트 UVC 에뮬레이션(76.910.00)순으로 에뮬레이션이 여타 보존방법에 비해 약 10배 이상 차이가 남

- 하지만 한번 개발 시 컴퓨팅 환경의 변화로 인한 에뮬레이터의 업데이트가 되지 않는 이상 추가적인 비용의 계산이 필요 없는 에뮬레이션 전략의 특성으로 인하여 20년에 한번 씩 에뮬레이터를 개발하였을 개발 비용이 비해 낮은 개발비용을 필요로 함을 확인 할 수 있으나, 마이그레이션 및 XML 생산 및 마이그레이션 전략에 비해 많은 비용을 필요로 하는 것을 알 수 있음

❒ 우리나라 공공기록에 대한 에뮬레이션 비용 산출

○ 이상의 에뮬레이션 수행 비용과 관련된 해외의 사례를 고찰할 때, 에뮬레이션 비용은 사례마다 편차가 있기는 하지만 일반적으로 마이그레이션에 비해 고비용이 소요됨을 알 수 있음

○ 단 국가마다 또한 보유한 전자기록의 유형마다 편차가 있는 관계상, 해외 사례를 통한 명확한 비용 산출은 어려움이 있음

○ 본 연구프로젝트에서 제안한 기능성 속성을 지닌 디지털 구성요소를 대상으로 에뮬레이션을 실시하는 방안을 가정할 때, 그 비용 산출은 현실적으로 어려움

○ 그 이유는 본 연구에서 샘플 대상으로 한 기관의 경우, 최근에 신설된 기관을 대상으로 한 것인 관계상, 에뮬레이션의 실제 대상을 구체적으로 도출하기 어려우며, 아울러 아직 구체적인 에뮬레이션 방안 및 대상, 범위 등에 관해 정책으로 결정된 것이 없기 때문임

○ 따라서 우리나라 공공기록을 대상으로 한 에뮬레이션의 구체적인 비용을 산정하기 위해서는, 제2년차 사업을 통한 보다 구체적인 에뮬레이션의 실제 기술 및 구체적인 대상이 확정되어야 할 것으로 사료됨

## 다. 한국형 에뮬레이션 전략 제안

### (1) 배경

❒ 한국형 에뮬레이션 전략의 필요성

○ 한국형 에뮬레이션 전략은 현재 우리나라 전자기록의 장기보존 포맷에서는 구현해 주지 못하는 디지털 컴포넌트의 기능성을 재현해 줄 수 있다는 점에서, 현재 공공기관에 보유 중인 디지털 컴포넌트 장기보존 전략으로서 의미 있음

○ 에뮬레이션 전략에 관한 이상의 선진사례의 분석결과를 기반으로 볼 때 디지털 컴포넌트의 장기보존 전략은 하나의 단일 전략만을 적용하는 것이 아닌, 디지털 기록의 유형 및 아카이브의 상황에 맞추어 하이 브리드하게 적용할 수 있는 전략을 고려할 필요가 있음

○ 에뮬레이션 전략은 기술의 복잡성이 높고 구현 비용의 추산이 어려워 효과적인 에뮬레이션이 현실 적용 가능한지 판단이 어려움. 일찍이 해외에서도 다양한 에뮬레이션 전략연구들이 진행된 바 있지만 결과적으로 뚜렷한 해답을 주기 보다는 다양한 견해와 사례가 존재하고 있음.

○ 디지털 보존 시장은 아직 초기 단계이며 모든 부문에 영향을 미칠 수 있어 성장 잠재력이 충분함. 특히, 에뮬레이션은 마이그레이션이 포괄할 수 없는 고유의 역할을 가지고 있기 때문에 앞으로 마이그레이션과 함께 미래의 디지털 자료에 대한 접근을 가능하게 해주는 능동적인 접근법으로서 의미 있는 역할을 수행할 수 있을 것임[71]

○ 우리나라의 경우 전자기록관리가 전면적으로 시행되고 있고 당장 2015년부터 국가기록원에 대량의 전자기록물이 이관될 예정으로 있어 관리대상인 디지털 컴포넌트의 특성에 입각한 포괄적인 보존전략과 그 일환으로서의 에뮬레이션 전략의 수립은 매우 필요한 사안임

❒ 한국 공공기관의 특징

○ 본 연구를 통해 방송통신위원회에서 생산된 디지털 컴포넌트 포맷 현황을 조사한 결과에 따르면 우리나라 공공기록물은 생산, 이용되는 플랫폼이 매우 한정적이고 특정 포맷의 디지털 컴포넌트가 대다수를 차지하는 것으로 나타남

○ 우리나라 행정기관의 경우 한/글, 엑셀 포맷이 93.41%로 가장 많으며 이를 포함한 8종의 포맷이 99.04%에 해당함. 아울러 이를 생산한 애플리케이션의 버전의 경우 한/글 97과 한/글 2007, 오피스 2003과 오피스 2007을 에뮬레이션 대상 포맷 버전으로 선정할 수 있었으며 이들 애플리케이션 윈도우 7이 모두 지원하고 있음(행정기관의 파일 포맷, 환경에 관한 상세한 내용은 3.3.1 참조)

---

71) Planets, “An Emerging Market”, Planets White Paper, 2010, p. 3.

○ 이러한 상황을 토대로 보았을 때 에뮬레이션과 관련하여 한국 공공기관은 제한된 생산 환경과 포맷으로 에뮬레이션의 약점으로 간주되는 복잡성과 비용을 상당부문 통제할 수 있을 것이란 점을 확인할 수 있음

○ 결과적으로 기본적으로 최대한 마이그레이션 전략으로 장기 보존 하되 디지털 컴포넌트별 보존대상이 되는 기능성이 마이그레이션 과정에서 손실될 수 있다고 판단되는 대상에 대해 에뮬레이션 전략을 병행하는 것이 필요하다고 판단됨

### (2) 주요 내용

❒ 한국형 에뮬레이션 전략의 목적과 대상

○ 목적

- 국가기록원이 보유하고 있는 디지털 컴포넌트 중에서 인쇄 렌더링으로는 재현할 수 없는 기능성을 보유하고 있는 것들을 원래의 생산 환경과 동일하게 재현하기 위한 체제 마련
- 국내 행정기관에서 생산되고 있는 디지털 컴포넌트의 현황에 대한 분석을 토대로 에뮬레이션 서비스의 대상물을 구체화, 세분화하여 실행 가능하며 체계적인 사전 준비 체계를 마련

○ 서비스 대상물

- 2015년부터 국가기록원에 대량으로 이관되어 오는 것을 우선적인 대상으로 할 필요가 있음
  - 본 연구에서 행정기관에서 사용한 운영체제, 애플리케이션 및 파일 포맷 변화를 분석한 결과 에뮬레이션 대상 포맷 버전을 선정할 수 있었음(3.3.1.라).
  - 이는 에뮬레이션을 위한 법적/비용적 고려사항, 에뮬레이터 개발을 위한 구체적인 환경 고려사항을 확인할 수 있게 하여 에뮬레이션 범위와 접근 전략을 수립하기 위해 핵심적인 근거가 될 수 있음

- 또한 현재 생산되고 이후 국가기록원으로 이관될 예정인 디지털 컴포넌트들 역시 잠재적인 에뮬레이션 대상이므로 전체 에뮬레이션 전략에 포함되어야 할 것임
- 그 외에 국가기록원이 현행 보유하고 있는 오래된 전자 기록물에 대해서는 에뮬레이션의 필요성 여부에 대한 검토를 통해 에뮬레이션이 필요한 기능성의 존재에 대한 전략적인 판단이 필요하다고 봄

<표 3-7> 한국형 에뮬레이션 전략의 서비스 대상

| 구분 | 설명 | 에뮬레이션 전략의 적용 |
|---|---|---|
| 2015년부터 이관 예정 | 2015년부터 이관되어 CAMS 상에서 관리예정인 기록물 | 1차적인 에뮬레이션 대상 |
| 현재 생산 중 | 잠재적인 에뮬레이션 대상 | “에뮬레이션을 위한 보존 평가 방법”을 통해 우선적으로 에뮬레이션 대상으로 선별된 이후 본격적인 에뮬레이션 방식을 고려 |
| 기 이관되어 보유 중 | 현재 CAMS 등 국가기록원 보유 오래된 디지털 컴포넌트 | 에뮬레이션 필요성 여부 등에 대한 별도검토 |

○ 에뮬레이션 관련 활동의 기본 흐름

- 기록물 생산 즉시, 관련 현황이 파악되어야 함, 비트스트림이 남아있을 때 필요한 기술 정보를 포착해야 함
- 현황 및 여건 등을 고려하여 어떤 식의 조치를 취할 것인지를 결정
- 소프트웨어 유지, 환경 VM, 뷰어 개발 등 필요한 프레임워크 구상 및 도구 구상
- 재현 도구 구동 등 처리

❒ 한국형 에뮬레이션 전략의 방향

○ 전체 에뮬레이션 방식

- 환경 전체에 대한 에뮬레이션으로 비트스트림을 남기고 해당 소프트웨어를 비롯한 환경을 남겨서 에뮬레이션 하는 방식임(Device와 O/S에 대한 에뮬레이션을 의미함)
- 국가기록원이 Planets의 GRATE나 KEEP의 TTF/EF와 같이 에뮬레이션을 위한 범용적인 서비스를 제공하는 방식임
- 이러한 서비스를 제공하기 위해서는 에뮬레이터 코어엔진, 소프트웨어 아카이브, 에뮬레이터 아카이브, 파일 포맷 레지스트리 등을 개발하는 것이 핵심적인 관건임

○ 응용 프로그램 에뮬레이션 방식

- 일종의 진보한 뷰어(advanced viewer) 방식으로 특정 파일 포맷에 대한 보여주기를 제공하는 방식의 에뮬레이션임
- 이것은 공공기록물에 특화한 서비스로만 가능할 것이며 제한된 생산 환경과 포맷에 대한 사전적인 파악을 통해서 어떤 파일 포맷에 대한 뷰어를 만들 것

인지에 대한 상세가 준비되어야 할 것임

- 본 연구를 통해 수행된 행정기관 파일 포맷 조사 분석 결과에 따르면 우선적으로 한글, 엑셀, 파워포인트에 대한 뷰어를 제공하는 방식이 될 수 있음

○ 특성화 서비스(Characterization) 개발의 필요성

- 기록관리 기관이 보유하는 디지털 콜렉션(디지털 컴포넌트의 의미 있는 묶음들)은 다양한 포맷으로 구성되어 있기 때문에 기관 차원에서 장기보존 전략을 수립함에 있어서 우선적으로 특성화 서비스를 통해서 보존하고자 하는 대상의 특성에 기반한 구체적인 서비스 전략을 수립해야 할 것임
- Planets의 사례에서는 Plato를 통해 장기보존계획을 세우도록 하는 과정 중에 특성화(Characterization)의 단계를 거쳐서 구체적인 보존행동(마이그레이션/에뮬레이션)을 취하도록 하고 있음. 또한 이때 보존대상 디지털 컴포넌트들로부터 자동적으로 특성 정보를 추출/기술하기 위한 언어(XCL)가 개발되어 활용되고 있었음
- 특성화 서비스는 특히 자동화된 형태로 제공되어야 할 필요가 있으며 이와 관련하여 본 연구보고서에서는 객체 속성 정보를 기반으로 디지털 컴포넌트의 비트스트림에서 렌더링 객체의 직접 추출 가능성을 검토하여 문서에 렌더링 객체가 포함되었는지를 자동으로 판별하기 위한 검토를 진행하고 특성화를 위한 근거를 확인한 바 있음(3.3.4. 마. 기능성 보존을 위한 XML Schema 개선안 참조)
- 이러한 자동화된 특성화 서비스를 통해서 생산된 디지털 컴포넌트가 기능성을 보유하고 있는 경우 에뮬레이션 대상으로서 자동적으로 확인되고 그들의 특성에 맞는 적절한 에뮬레이션 방식이 설계될 수 있을 것이기 때문에 보다 체계적인 장기 보존 활동이 가능하게 될 것임
- 특성화 서비스를 위해서는 해당 기능성에 관한 기술 메타데이터가 파일 포맷 레지스트리를 통해서 관리되어야 한다는 이슈가 존재함. 레지스트리는 렌더링 해야 하는 모든 포맷에 대한 표현 정보를 포함하고 있어야 함. 이상의 내용을 정리한 것이 아래의 <표 3-8>임

<표 3-8> 한국형 에뮬레이션 전략의 방향

| 구분 | 전체 에뮬레이션 방식 | 응용 프로그램 에뮬레이션 방식 |
|---|---|---|
| 특징 | -환경 전체에 대한 에뮬레이션으로 비트스트림과 해당 소프트웨어 및 환경을 남겨서 재현하는 것 | -특정 파일 포맷에 대한 보여주기를 제공하는 방식으로 재현하는 것 |
| 주요 관건 | -범용적 에뮬레이션을 위한 서비스를 어떻게 구성하여 제공할 것인가 | -어떤 파일 포맷에 대한 보여주기를 제공할 것인가 |
| 관련 해외사례 | -Planets의 GRATE<br>-KEEP의 EF/TTF | -특화된 다양한 뷰어들 |
| 공통사항 | -특성화 서비스를 통해 보존대상의 특성 및 기능성을 파악<br>-관련된 충분한 기술정보를 확보/축적 위한 파일 포맷 레지스트리 구축 및 운영<br>-OAIS 등 디지털 보존 관련 표준 준수 및 기존 아카이빙 시스템과의 상호 운용성 확보 | |

❒ 한국형 에뮬레이션 전략에 근거한 “에뮬레이션 서비스 활용 시나리오”

○ (국가기록원 이용자) 국가기록원 소장 기록 중 에뮬레이션이 필요한 경우 국가기록원의 플랫폼에서 에뮬레이션 서비스를 이용함

○ 이용자는 기록물을 검색 후 특정 디지털 컴포넌트를 오픈하게 되면 에뮬레이션이 필요한 경우 “국가기록원 에뮬레이션 서비스”로 환경이 변경되고 DFR에 등록된 정보에 따라 적절한 애플리케이션이 구동되어 렌더링이 이루어짐

### (3) 기타 사항

❒ 한편 본 연구에서는 디지털포렌직 관점에서 디지털데이터의 증거력에 관해 자문을 받았으며 그 결과로서 핵심적인 시사점만을 요약하면 다음과 같음

❒ 전자기록의 장기보존 전략은 한편으로는 전자기록이 증거력을 유지하기에 용이한 방식으로 결정될 필요가 있음. 여기서 증거력이란 증거가치, 증명력 등과 유사하게 사용되는 법정 용어로, "증거로서 믿을 만한 실질적인 가치. 형사 소송에서는 증거가 주요 사실을 인정하는 자료로 사용될 수 있는 법률적인 능력인 증거 능력을 전제로 하여 그 증거가 가지는 신빙성의 정도"를 의미함(네이트사전 alldic.nate.com 참조) 기록관리기관에 보관 중인 전자기록이 법정에서 증거력을 가질 수 있어야 하는 것은 기본적인 요구임. 전자기록의 증거력은 디지털컴포넌트의 진본성과 신뢰성, 무결성, 이용가능성 등 전자기록이 유지해야하는 기본 품

질요건 모두가 충족되어야 확보가 가능하며 영구기록물관리기관이 전자기록을 장기간 보존하는 과정에서도 이러한 증거력은 안정적으로 유지되어야 할 것임. 따라서, 마이그레이션 전략과 에뮬레이션 전략을 상호 비교 검토할 때도 증거력의 유지라는 측면에서도 살펴볼 필요가 있을 것임

❒ 디지털컴포넌트의 포맷특성과 저장매체의 특성을 잘 알고 있는 디지털포렌직 전문가로부터 자문을 받은 결과, 전자기록의 장기보존 과정에서 디지털 컴포넌트는 특정 종류의 매체에 저장되어 있다가 매체의 노후화 주기에 맞춰 복제, 미러링 등의 방식으로 매체가 재생산되어야 할 것임. 또한, 마이그레이션 전략을 수용한 경우에는 디지털 컴포넌트 혹은 포맷 구식화 시점에 맞춰 포맷을 변환하여 새롭게 생성한 디지털컴포넌트를 새로운 매체에 저장해야 함. 디지털포렌직 전문가의 견해에서 볼 때, 디지털데이터가 증거력을 유지하는 가장 중요한 요건은 보관의 연속성을 보여주는 것이므로 전자기록의 장기보존 과정에서 매체의 변화, 디지털 컴포넌트의 변화 과정이 모두 상세히 기록화되어 있어야 하며 기록 자체와 보관의 연속성을 보여주는 메타데이터가 긴밀히 연관되어 보관되어야 할 것임

❒ 에뮬레이션 전략의 경우 원래의 포맷 그대로 비트스트림을 보존하는 것이므로 마이그레이션 전략에 비해 법적 증거력의 유지에 유리하다고 보여짐. 특히, 디지털 포렌직 과정에서 비트스트림이 표현하고자 하는 주요 메시지 뿐 만 아니라 그 안에 숨겨진 메타데이터를 중시하고 이를 근거로 중요한 사실을 포착해내기도 한다는 점을 고려한다면, 전자기록이 생산될 당시의 비트스트림을 그대로 유지하는 에뮬레이션 전략이 법적 증거력 보전이라는 목표에는 보다 합목적적이라 할 수 있음. 마이그레이션 전략을 사용하는 경우 포맷이 변화되는 과정에서 표현하고자 하는 것의 일부만을 선택적으로 보존하게 되고, 또한 메타데이터의 일부가 상실되기도 한다는 점을 고려한다면 디지털포렌직 관점에서 마이그레이션 전략은 전자기록의 법적 증거력을 보전하기에 불충분한 솔루션으로 여겨질 수 있음

❒ 그러나, 한편으로는 디지털포렌직 전문가가 보여준 재판 사례를 통해 얻을 수 있는 다른 측면의 시사점으로는 마이그레이션 전략이 기관이 전자기록 관리를 통해 명확한 증거력을 확보하고 유지하기에는 훨씬 용이하다는 점을 제시할 수 있음. 즉, 기관이 보유한 각종 저장매체의 정보들 중에 증거가치를 지닌 기록을 명시적으로 선언함으로써 증거력 있는 정보를 선별하는 절차를 거치고, 다음으로는 기록의 디지털컴포넌트 내에 묻어있는 혼란스런 메타데이터정보들을 마이그레이션 과정을 통해 삭제함으로써 기관에 의미 있는 증거정보만 남길 수 있도록 한다면 증거로서의 기록관리가 갖는 효과성이 배가될 수 있을 것임

## 3.2 행정기관의 디지털 컴포넌트 유형별 보존대상 연구

### 3.2.1 방송통신위원회 디지털 컴포넌트 현황 조사

#### 가. 조사 개요

❒ 조사의 필요성

○ 국내 행정기관에서 에뮬레이션이 필요한 구체적인 대상을 파악하고 연구에 필요한 기초 자료를 확보하기 위해서는 장기보존 대상인 디지털 컴포넌트의 포맷 현황 파악이 필요함

○ 당장 2015년부터 국가기록원으로 대량의 전자기록물이 이관되어올 예정이기 때문에 이 디지털 컴포넌트의 포맷 현황을 파악하고 행정기관의 디지털 컴포넌트 유형에 대한 현황 분석에 기반 한 에뮬레이션 전략이 수립되어야 할 것임

○ 특히, 이러한 본 연구를 진행하기 위해서는 적어도 한 기관의 모든 포맷 현황을 파악할 수 있도록 전수조사를 실시할 필요가 있음

❒ 조사기관: 방송통신위원회

❒ 조사내용

○ 방송통신위원회에서 생산된 디지털 컴포넌트를 대상으로 조사를 수행함.

○ 방송통신위원회 기록관리시스템의 /emc/storage1/orign 폴더 하위에 있는 파일에 대한 전수조사를 실시하였음

○ 디지털 컴포넌트의 기간범위

- 2008년에서 2013년에 걸쳐 RMS로 이관된 것들임
- 방송통신위원회의 RMS 서버에서 관리되고 있는 디지털 컴포넌트들은 이관연도별로 폴더를 구성하고 있음

❒ 조사일시

○ 총 2차에 걸쳐 수행하였으며 각 회차 별 일시와 수량은 다음과 같음 :

- 1차 : 2013 9. 2 10:00 - 9/3 18:00(약 60만건)
- 2차 : 2013. 9. 23 10:00 - 9/24 18:00(약 200만건)

❒ 조사 방법

○ 방송통신위원회의 RMS에 있는 파일(약 270만건) 조사를 위해서 파일들을 PC로 내리는 작업[72]을 우선적으로 수행하였음

72) 방송통신 위원회와의 협조를 통해 방통위 RMS 서버 /emc/storage1/orign 폴더 하위의 디지털 컴포넌트를 FTP를 이용하여

○ 한꺼번에 파일을 내릴 수가 없어서 폴더이름을 기준으로 TAR를 이용하여 묶어서 전송 준비를 수행하였음

○ 작업순서는 다음과 같음 :

• ① RMS 서버에 접속하여 폴더명을 기준으로 파일을 묶음

- 폴더의 이름이 이관년도(4)+월(2)+일(2)+시간(2) 이렇게 10자리로 구성이 되어 있어서 끝자리를 기준으로 TAR 파일 생성
- end0.tar, end1.tar……end9.tar

• ② FTP로 RMS 서버에 접속하여 파일을 다운로드

- 전체 용량 : 400G, 서버에서 다운로드 속도 평균 : 5Mbps

• ③ PC에서 다운로드 된 TAR 파일을 다시 품

• ④ DROID6, Extract Format를 이용하여 필요한 정보를 추출한다.

• ⑤ 결과 정리

• ⑥ 원본문서 삭제

## 나. 조사결과 분석 및 시사점

### (1) 방송통신위원회 디지털 컴포넌트 포맷 보유 현황

❒ 방송통신위원회의 디지털 컴포넌트 포맷 보유 현황은 크게 두 가지로 구분할 수 있음

○ 주요 디지털 컴포넌트 포맷 분포현황

○ 기타 디지털 컴포넌트 포맷 분포현황

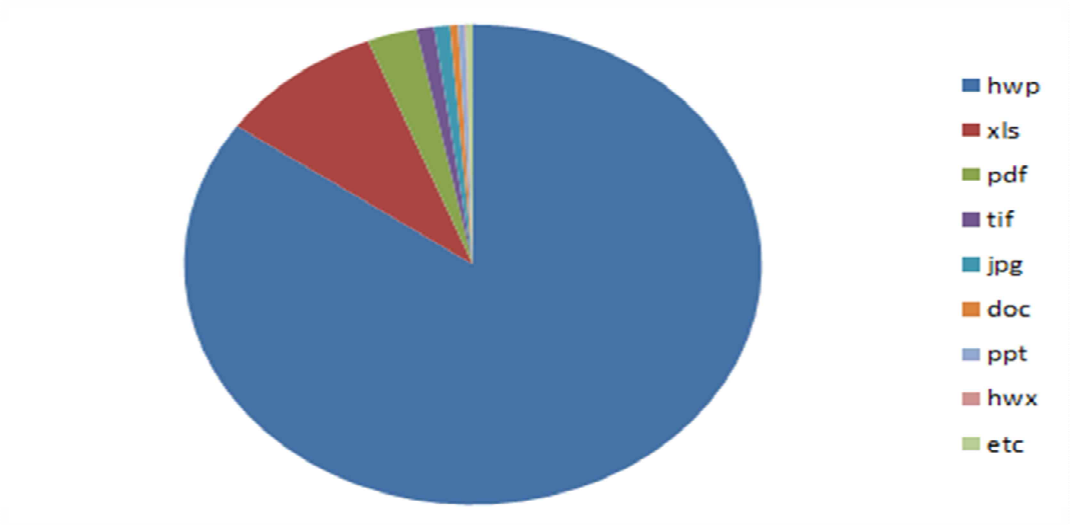
❒ 주요 디지털 컴포넌트 포맷 분포 현황

○ 방송통신위원회의 주요 디지털 컴포넌트의 포맷별 분포현황은 다음 표와 그림과 같음

지정된 PC에 다운로드를 수행하는 작업

<표 3-9> 디지털 컴포넌트 포맷 분포현황

| 포맷 구분 | 수량(건) | 비율(%) |
|---|---|---|
| hwp | 2,143,085 | 84.78 |
| xls | 234,874 | 9.29 |
| pdf | 70,378 | 2.78 |
| tif | 25,131 | 0.99 |
| jpg | 22,196 | 0.88 |
| doc | 11,535 | 0.46 |
| ppt | 9,574 | 0.38 |
| hwx[73] | 1,744 | 0.07 |
| | 2,518,517 | 99.63 |




<그림 3-43> 파일포맷 현황 분포도

○ 방송통신위원회의 경우 HWP, XLS문서가 94% 정도를 차지하고 있고, 기타, PDF, TIF, JPG의 문서가 각각 약 1% 이상의 비중을 차지하고 있음
○ 앞서 언급한 8개의 문서가 전체의 99.63%를 차지하고 있음

❒ 기타 디지털 컴포넌트 포맷 현황
○ 방송통신위원회의 기타 디지털 컴포넌트의 포맷유형 현황은 다음 표와 같음

73) 핸디소프트 전자결제 문서

<표 3-10> 디지털 컴포넌트 기타 포맷 유형

| 포맷 확장자 | 포맷 유형 | 포맷 확장자 | 포맷 유형 |
|---|---|---|---|
| 7z | 7Zip format | k3g | 휴대폰 동영상 파이 |
| ai | Acrobat- Portable Document Format | mht | MHTML의 파일 |
| alz | 압출파일의 한 종류 | mp3 | MPEG 1/2 Audio Layer 3 |
| bat | Batch file (executable) | nxl | Microsoft Excel 97 Workbook (xls) |
| bmp | Windows Bitmap | opt | |
| body | | ozd | oz에서 생산된 문서 |
| csv | Comma Separated Values | png | Portable Network Graphics |
| dwg | AutoCAD Drawing | psd | Adobe Photoshop |
| egg | 압출파일의 한 형태 | rqr | Extensible Markup Language |
| eml | Hypertext Markup Language | rst | VGA Planets Game File (Tim Wisseman) 와 주로 관련된 텍스트 파일 |
| enc | Encore에서 만들어진 음악악보 파일 | rtf | Rich Text Format |
| er1 | GameGuard Log File (INCA Internet Co Ltd.) 와 주로 관련된 기타 파일 | sql | Structured Query Language Data |
| exe | Windows Portable Executable | swf | Adobe Flash |
| flw | Kivio Flowchart (theKompany.com) 와 주로 관련된 기타 파일 | tmp | OLE2 Compound Document Format |
| gif | Graphics Interchange Format | txt | Plain Text File |
| gux | 훈민정음 백업 파일 | wav | Waveform Audio(PCMWAVEFORMAT) |
| hml | Extensible Markup Language | wma | Windows Media Audio |
| hox | Extensible Markup Language | wmv | Windows Media Video |
| hpt | | xlk | Microsoft Excel 97 Workbook (xls) |
| htm | Hypertext Markup Language | xlsm | Microsoft Excel Macro-Enabled |
| html | Hypertext Markup Language | xlsm | Microsoft Excel Macro-Enabled |
| hwt | OLE2 Compound Document Format | xlt | Microsoft Excel 97 Workbook (xls) |
| hwx | 핸디소프트 전자결제 파일 | xml | Extensible Markup Language |
| jp001 | | | |

○ 포맷의 body와 같은 확장자는 일반적으로 사용하지 않는 확장자임
○ ‘gux’와 같은 파일로서 가치가 없는 파일도 있으며 이는 생산자가 문서를 잘못 첨부한 것으로 보임
  • 파일시스템에서 특정한 폴더 아래에 있는 파일을 조사를 하였기 때문에 실제 RMS시스템과는 불일치가 발생할 수도 있으며, Garbage 파일들이 존재 할 수 있음

❒ 수집된 자료를 토대로 본 연구팀 내에서 수행중인 애플리케이션 히스토리 조사결과와 매핑 하여 에뮬레이션 대상 버전을 파악하는 등 분석을 진행하고 이러한 분석결과를 에뮬레이션 세부 계획 수립에 반영할 것임

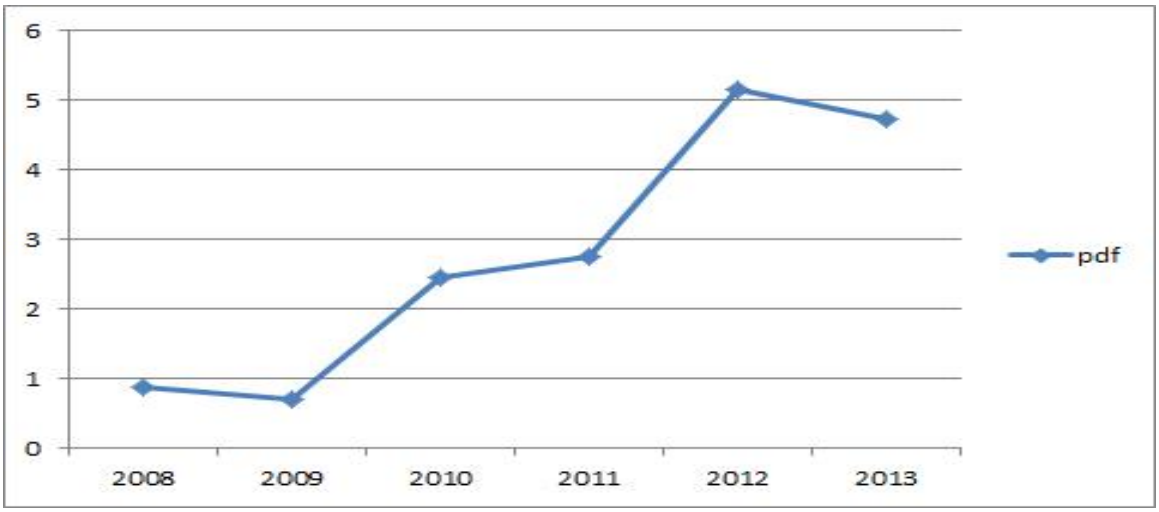
### (2) 방송통신위원회 디지털 컴포넌트 포맷별 문서 생산추이

❒ 다음과 같이 방송통신위원회의 최근 5년간 생산된 다섯 가지 유형의 디지털컴포넌트 포맷의 생산 추이를 분석하였음
○ PDF
○ XLS와 XLSX
○ PPT와 PPTX
○ DOC와 DOCX
○ TIF와 JPG

❒ 연도별 PDF 포맷 문서 생산 추이
○ 최근 5년간 PDF 포맷의 문서 생산 추이는 다음 표와 그림과 같음

<표 3-11> 최근 5년간 PDF 포맷 문서 생산 현황

| 구분(년도) | 생산량(건) | 비율(%) |
|---|---|---|
| 2008 | 1,454 | 0.886261 |
| 2009 | 5,058 | 0.714346 |
| 2010 | 3,408 | 2.466866 |
| 2011 | 18,338 | 2.763903 |
| 2012 | 20,170 | 5.167581 |
| 2013 | 21,949 | 4.731571 |




<그림 3-44> PDF 포맷 문서 생산 증가량

○ 위 표와 그림을 통해 PDF포맷의 문서가 시간이 지남에 따라 점차 생산량이 증가함을 알 수 있음

○ 2008년 이관된 문서는 0.88%에서 2012년에 이관된 문서의 경우는 5.16% 정도까지 증가함

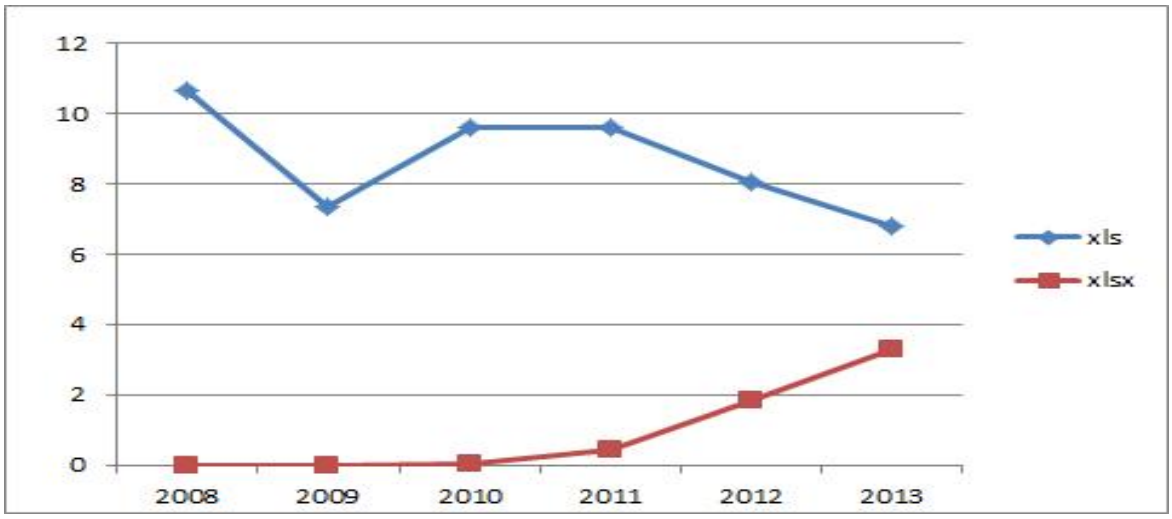
❒ XLS와 XLSX[74]포맷 문서 생산 추이

○ 최근 5년간 XLS와 XLSX 포맷의 문서 생산 추이는 다음 표와 그림과 같음

<표 3-12> XLS와 XLSX 포맷 문서 생산 현황

| 구분 (년도) | XLS | | XLSX | | XLS + XLSX |
|---|---|---|---|---|---|
| | 생산량(건) | 비율(%) | 생산량(건) | 비율(%) | 연간 비율 합계(%) |
| 2008 | 17,451 | 10.63696 | 0 | 0 | 10.63696 |
| 2009 | 51,979 | 7.341045 | 0 | 0 | 7.341045 |
| 2010 | 13,255 | 9.594574 | 40 | 0.028954 | 9.623528 |
| 2011 | 63,820 | 9.61895 | 2,922 | 0.440404 | 10.05935 |
| 2012 | 31,462 | 8.060607 | 7,097 | 1.818261 | 9.878868 |
| 2013 | 31,497 | 6.789844 | 15,349 | 3.308801 | 10.09865 |

74) MS Office 2007 이후에 생산된 문서, XLSX에서 X는 XML을 나타낸다.




<그림 3-45> EXCEL(XLS, XLSX)포맷 문서 생산 증가량

○ XLSX 포맷의 경우와 같이 확장자에 “X”가 붙어 있는 경우는 MS Office 2007 부터 생산된 문서임

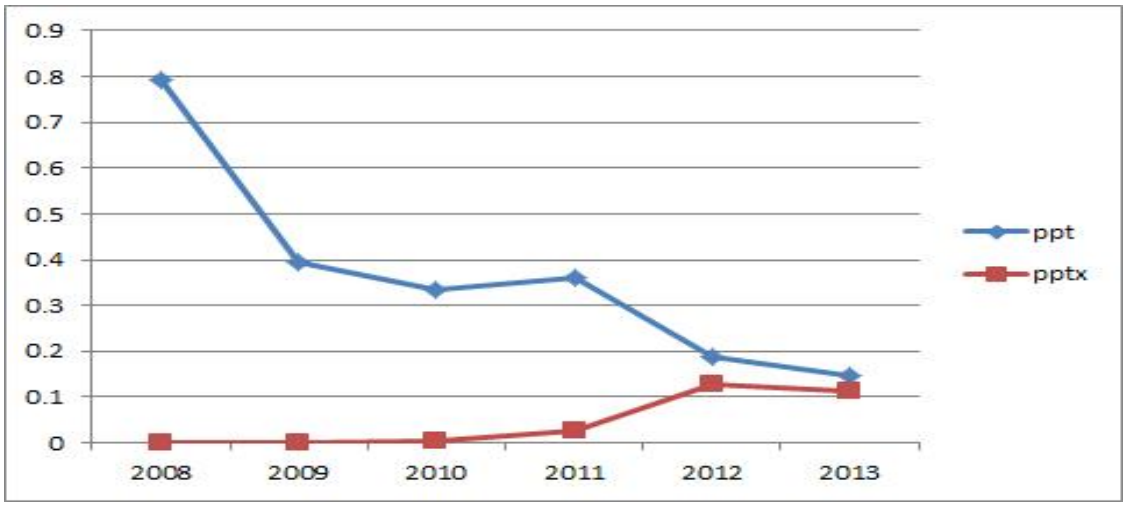
○ 스프레드시트인 엑셀포맷(XLS + XLSX)문서의 경우 2008년부터 2013년까지 포맷명과 유형은 달라져도 총합은 약 10%가량 유지하고 있음

❒ PPT와 PPTX 포맷 문서 생산 추이

○ 최근 5년간 PPT와 PPTX 포맷의 문서 생산 추이는 다음 표와 그림과 같음

<표 3-13> PPT와 PPTX 포맷 문서 생산 현황

| 구분 (년도) | PPT | | PPTX | | PPT + PPTX |
|---|---|---|---|---|---|
| | 생산량(건) | 비율(%) | 생산량(건) | 비율(%) | 연간 비율 합계(%) |
| 2008 | 1,300 | 0.792393 | 0 | 0 | 0.792393 |
| 2009 | 2,797 | 0.395023 | 0 | 0 | 0.395023 |
| 2010 | 461 | 0.333693 | 5 | 0.003619 | 0.337312 |
| 2011 | 2,399 | 0.361577 | 175 | 0.026376 | 0.387953 |
| 2012 | 733 | 0.187796 | 496 | 0.127076 | 0.314872 |
| 2013 | 681 | 0.146804 | 527 | 0.113606 | 0.26041 |




<그림 3-46>Power point(PPT, PPTX)포맷 문서 생산 증가량

○ PPTX 포맷 경우와 같이 확장자에 “X”가 붙어 있는 경우는 MS Office 2007부터 생산된 문서임

○ PPTX 포맷 문서의 경우 양이 적어 추세를 분석하기에는 무리가 있음

○ PPTX 포맷 문서는 2010년부터 생산된 문서이기 때문에, 좀 더 시간을 두어야 추세를 분석할 수 있음

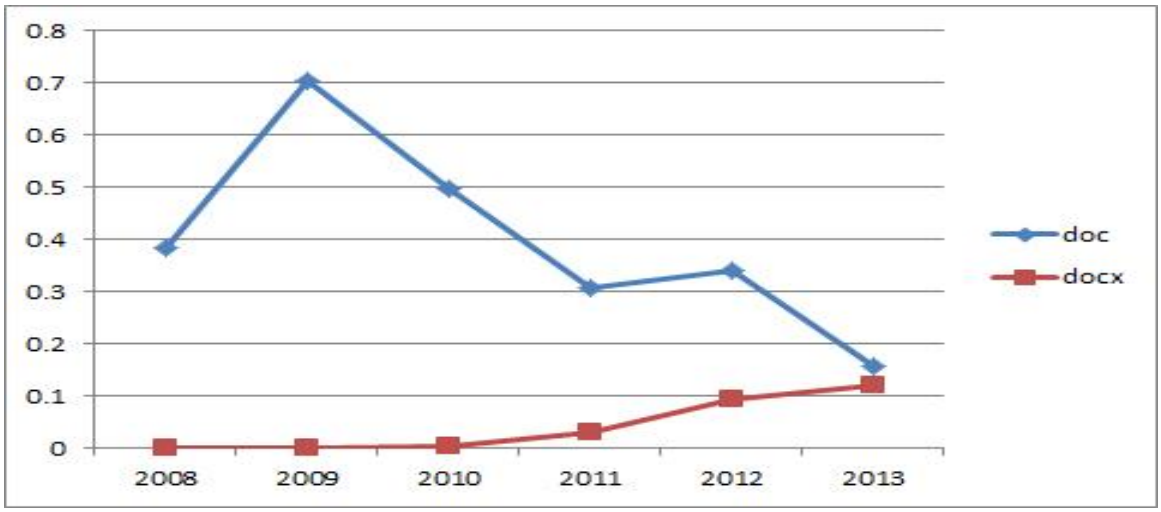
○ Powerpoint(PPT, PPTX)포맷 문서의 경우 전체적으로 감소추세에 있음

❒ DOC와 DOCX 포맷 문서 생산 추이

○ 최근 5년간 DOC와 DOCX 포맷의 문서 생산 추이는 다음 표와 그림과 같음

<표 3-14> DOX와 DOCX 포맷 문서 생산 현황

| 구분 (년도) | DOC | | DOCX | | DOC + DOCX |
|---|---|---|---|---|---|
| | 생산량(건) | 비율(%) | 생산량(건) | 비율(%) | 연간 비율 합계(%) |
| 2008 | 632 | 0.385225 | 0 | 0 | 0.385225 |
| 2009 | 4,979 | 0.703189 | 0 | 0 | 0.703189 |
| 2010 | 686 | 0.496558 | 5 | 0.003619 | 0.500177 |
| 2011 | 2,045 | 0.308222 | 202 | 0.030445 | 0.338667 |
| 2012 | 1,336 | 0.342285 | 369 | 0.094538 | 0.436823 |
| 2013 | 727 | 0.15672 | 553 | 0.119211 | 0.275931 |




<그림 3-47> Ms word(DOC, DOCX)포맷 문서 생산 증가량

○ DOCX 포맷의 경우와 같이 확장자에 “X”가 붙어 있는 경우는 MS Office 2007 부터 생산된 문서임
○ MS워드 문서는 경우는 0.2 ~ 0.3%정도의 비중을 차지하고 있음
○ DOCX문서의 경우는 점점 증가하는 추세로 보이지만, 증가율이 미약하여 증가 추세로 보기에는 무리임
○ 다양한 사례의 데이터 분석이 필요함

❒ TIF와 JPG 포맷 생산 추이

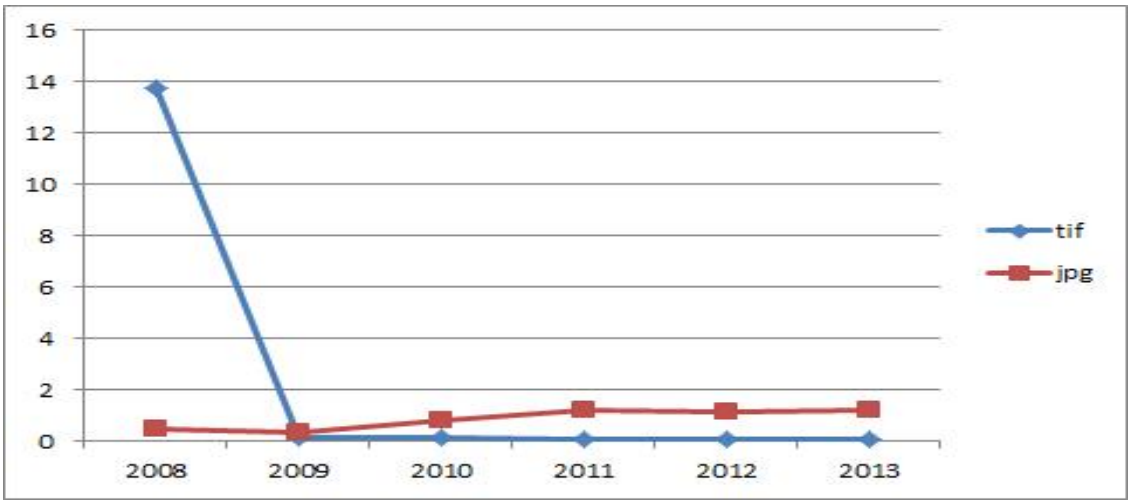
○ 최근 5년간 TIF와 JPG 포맷의 생산 추이는 다음 표와 그림과 같음

<표 3-15> TIF 포맷 현황

| 구분(년도) | 생산량(건) | 비율(%) |
|---|---|---|
| 2008 | 22,538 | 13.73766 |
| 2009 | 1,052 | 0.148575 |
| 2010 | 222 | 0.160694 |
| 2011 | 678 | 0.102188 |
| 2012 | 292 | 0.074811 |
| 2013 | 349 | 0.075234 |

<표 3-16> JPG 포맷 현황

| 구분(년도) | 생산량(건) | 비율(%) |
|---|---|---|
| 2008 | 778 | 0.474217 |
| 2009 | 2,194 | 0.309861 |
| 2010 | 1,138 | 0.823736 |
| 2011 | 8,085 | 1.218571 |
| 2012 | 4,350 | 1.114476 |
| 2013 | 5,651 | 1.218192 |




<그림 3-48> TIF, JPG 포맷 생산 증가량

○ TIF 포맷의 경우 2008년 이전에 생산된 문서가 주를 차지하고 있어 특수 용도로 사용을 하였거나 비 전자문서를 DB화 하면서 생산된 문서로 추정됨
○ JPG의 문서는 약소하지만 증가 추세에 있음

### (3) 방송통신위원회 주요 디지털 컴포넌트별 생산 포맷 정보

❒ 방송통신위원회에서 생산한 세 가지 유형의 주요 디지털컴포넌트의 포맷정보를 분석하였음
○ XLS 포맷정보
○ PPT 포맷정보
○ HWP 포맷정보

❒ XLS 문서 생산 포맷 정보

<표 3-17> EXCEL문서 포맷 현황

| 확장자 | 파일 포맷 | 버전 | 건수 |
|---|---|---|---|
| xlsx | Microsoft Excel for Windows | 2007 onwards | 9180 |
| xlsx | Microsoft Office Encrypted Document | 2007 Onwards | 51 |
| xlsm | Microsoft Excel Macro-Enabled | 2007 | 117 |
| xlsb | Microsoft Office Open XML | 2007 onwards | 3 |
| xls | OLE2 Compound Document Format | unknown | 926 |
| xls | Microsoft Excel 97 Workbook (xls) | 8 | 58512 |
| xls | Microsoft Excel for Macintosh | 3 | 41 |

| | | | |
|---|---|---|---|
| xls | Microsoft Excel for Macintosh | 4 | 41 |
| xls | Microsoft Excel for Macintosh | 98 | 41 |
| xls | Microsoft Excel for Macintosh | 2001 | 41 |
| xls | Microsoft Excel for Macintosh | 2002 | 41 |
| xls | Microsoft Excel for Macintosh | 2004 | 41 |
| xls | Microsoft Excel for Macintosh | X | 41 |
| xls | Microsoft Excel 95 Workbook (xls) | 7 | 41 |
| xls | JPEG File Interchange Format | 1.01 | 2 |
| xls | Exchangeable Image File Format (Compressed) | 2.2 | 3 |
| xls | Graphics Interchange Format | 89a | 1 |
| xls | Acrobat PDF 1.4 - Portable Document Format | 1.4 | 3 |
| xls | Acrobat PDF 1.1 - Portable Document Format | 1.1 | 2 |
| xls | Microsoft Powerpoint Presentation | 97-2002 | 4 |
| xls | Acrobat PDF 1.5 - Portable Document Format | 1.5 | 1 |
| xls | Windows Bitmap | 3 | 1 |
| xls | Acrobat PDF 1.3 - Portable Document Format | 1.3 | 1 |
| xls | Acrobat PDF 1.6 - Portable Document Format | 1.6 | 1 |
| xls | Hypertext Markup Language | unknown | 200 |
| xls | Hypertext Markup Language | 4.01 | 1 |
| xls | Extensible Hypertext Markup Language | 1 | 1 |
| xls | Hypertext Markup Language | 4 | 1 |

○ 확장자가 모두 XLS(X,M,B)로 끝나는 문서이나, 실제 내용을 보면 이미지, PDF 문서, HTML문서들이 존재함

○ 확장자의 부여가 잘못된 경우일 수 있음

- 웹에서 첨부물을 다운로드하여 첨부하는 경우 잘못된 문서(HTML)를 첨부하는 경우가 있음

❒ PPT 문서 생산 포맷 정보

<표 3-18> Powerpoint 문서 포맷현황

| 확장자 | 파일 포맷 | 버전 | 건수 |
|---|---|---|---|
| pptx | Microsoft Powerpoint for Windows | 2007 onwards | 441 |
| pptx | OLE2 Compound Document Format | 97 | 2 |
| ppt | OLE2 Compound Document Format | 97 | 64 |
| ppt | Microsoft PowerPoint for Macintosh | 4 | 9 |
| ppt | Microsoft PowerPoint for Macintosh | 98 | 9 |
| ppt | Microsoft PowerPoint for Macintosh | 2001 | 9 |
| ppt | Microsoft PowerPoint for Macintosh | X | 9 |
| ppt | Microsoft Word Document | 97-2003 | 1 |
| ppt | Microsoft Powerpoint Presentation | 97-2002 | 2106 |
| ppt | Microsoft Excel 97 Workbook (xls) | 8 | 5 |

○ 2종류 6개의 문서는 확장자는 PPT이지만, 실제 내용은 엑셀문서와 워드문서로 되어 있음

- 이와 같은 경우는 확장자가 잘못 부여된 경우로 보임. 즉, 헤더를 분석하여 나온 파일 포맷이 실제 내용이라고 하였을 때 확장자 값이 잘못 부여 있다는 사실을 본 조사를 통해 확인할 수 있었음

❒ HWP 문서 생산 포맷 정보

<표 3-19>방송통신위원회 보유 한/글 포맷 현황

| 확장자 | 포맷 | 버전 | 건수 |
|---|---|---|---|
| hwp | HWP Document File | 5.0 | 1505880 |
| hwp | HWP Document File | 3.0 | 6204 |

○ 위 표에서와 같이 한/글 문서는 크게 2개 버전의 포맷으로 생산되었음

- 5.0 포맷은 2000 워디안 제품 이후에 생산된 문서임
  - 한/글 2002, 한/글 2004, 한/글 2005, 한/글 2006, 한/글 2007, 한/글 2008, 한/글 2010, 한/글 2013으로 생산된 문서
- 3.0 포맷은 한/글 97 이전
  - 한/글 3.0, 한/글 프로 96, 한/글 97, 한/글 8.15

○ 한/글 97이 가장 많이 사용되었고, 지금까지도 사용되고 있음

<표 3-20> 한/글 문서 포맷현황

| 확장자 | 포맷 | 버전 | 건수 |
|---|---|---|---|
| hwp | HWP Document File | 5 | 1505880 |
| hwp | PK | 0 | 52 |
| hwp | PDF | 0 | 322 |
| hwp | FasooSecureContai | 0 | 822 |
| hwp | JFIF | 0 | 242 |
| hwp | GIF87aN | 0 | 1 |
| hwp | HWP Document File | 3 | 6204 |
| hwp | xml | 0 | 3896 |
| hwp | SESSION | 0 | 8 |
| hwp | FILE | 0 | 3 |
| hwp | BM | 0 | 11 |
| hwp | BMf | 0 | 1 |
| hwp | Handysoft | 0 | 333 |
| hwp | GIF89ao | 0 | 1 |
| hwp | rtf1 | 0 | 3 |
| hwp | GIF89aL | 0 | 1 |
| hwp | RIFF | 0 | 2 |
| hwp | NCEF | 0 | 3 |
| hwp | GIF89a | 0 | 2 |
| hwp | xls | 0 | 2 |
| hwp | GIF89aR | 0 | 1 |
| hwp | Exif | 0 | 20 |
| hwp | Photoshop | 0 | 3 |
| hwp | ALZ | 0 | 6 |

○ 위 표는 HWP 문서에서 Header 부분을 추출한 결과임

○ 확장자는 모두 “hwp”인데 Header 부분을 보면 HWP문서로 볼 수 없는 문서들이 존재함

- 확장자가 잘못 부여된 문서로 판단됨

### (4) 기타 사항

❒ 포맷 조사를 진행하는 과정에서 발견한 몇 가지 시사점이 존재함

❒ 기록관리시스템(RMS)의 데이터가 일반 스토리지 외에 아카이빙 스토리지에 들어가 있는 경우가 다수 존재함. 이 경우 데이터에 접근하는 방식에 대한 별도의 고려가 요구됨

○ 이와 관련하여 당초 조사대상을 안전행정부로 설정하고 진행하였으나 조사 과정에서 안전행정부의 데이터가 아카이빙 스토리지에 들어가 있어 본 조사방법을 적용하기 어려움 등의 문제가 발생하여서 조사대상이 변경되었음

❒ 기록관이 보유하고 있는 디지털 컴포넌트에 대한 기본적인 통계 데이터가 파악되고 있지 않으며 그에 대한 대책이 요구됨

○ 보존계획의 입안, 비용 산정 등 기록관이 보유하고 있는 디지털 컴포넌트의 현황에 대한 기초 데이터는 다양하게 활용될 수 있기 때문에 기록관이 보유하고 있는 디지털 컴포넌트의 현황을 파악하기 위한 명확한 주체, 도구가 시급히 준비되어야 할 것임

❒ 당초 현황 분석 결과를 토대로 기능성 요소 등 기능성 보존 요구사항과 관련된 샘플 자료를 추가로 확보하여 기능성 보존이 필요한 행정기관의 구체적인 사례로서 확보하고자 하였으나 일차적인 현황 조사 결과, 이후의 샘플 추출의 효용성을 재고하였고 추가 작업으로 진행하지 않았음

○ 본 현황 조사는 방송통신위원회에서 생산된 디지털 컴포넌트의 전수조사로서 RMS 서버상에 존재하는 전체 디지털 컴포넌트에서 파일 헤더 정보를 추출하여 그 데이터를 활용한 통계현황임

○ 샘플 조사를 진행할 경우에 헤더 정보 수준의 접근 권한을 넘어 디지털 컴포넌트의 내용에 대한 접근권한을 가질 수 있어야 할 것임

○ 또한 260만 건의 디지털 컴포넌트들은 특정 애플리케이션이 제공하는 기능성을 사용할 수 있는 잠재적인 대상이지 그 중에서 어떤 디지털 컴포넌트가 기능성을 직접적으로 사용하고 있는지를 판단하기 위해서는 별도의 작업이 필요함

○ 이는 애초 샘플 추출을 통해서 행정기관에서 실제로 구현된 기능성을 확인해보고자 하는 목적을 넘어서는 작업이 요구되는 것임. 아울러 본 연구에서는 행정기관에서 생산되는 디지털 컴포넌트의 구체적인 사례를 분석하는 별도의 과업이 병행되고 있기 때문에(3. 3절에서 보존 포맷으로 변환시 소실되는 기능성에 대한 포괄적인 분석 및 사례 고찰함) 별도의 효용성이 없다고 판단하였음

### 3.2.2 디지털 컴포넌트 보존 속성 연구

❒ 에뮬레이션 대상이 될 디지털 컴포넌트별 보존 속성을 정의하여 공공 기록관리기관의 생산자 단계에서 활용될 수 있는 보존 평가 방법론을 개발하는 연구를 진행

## 가. 에뮬레이션 대상 기능성 정의

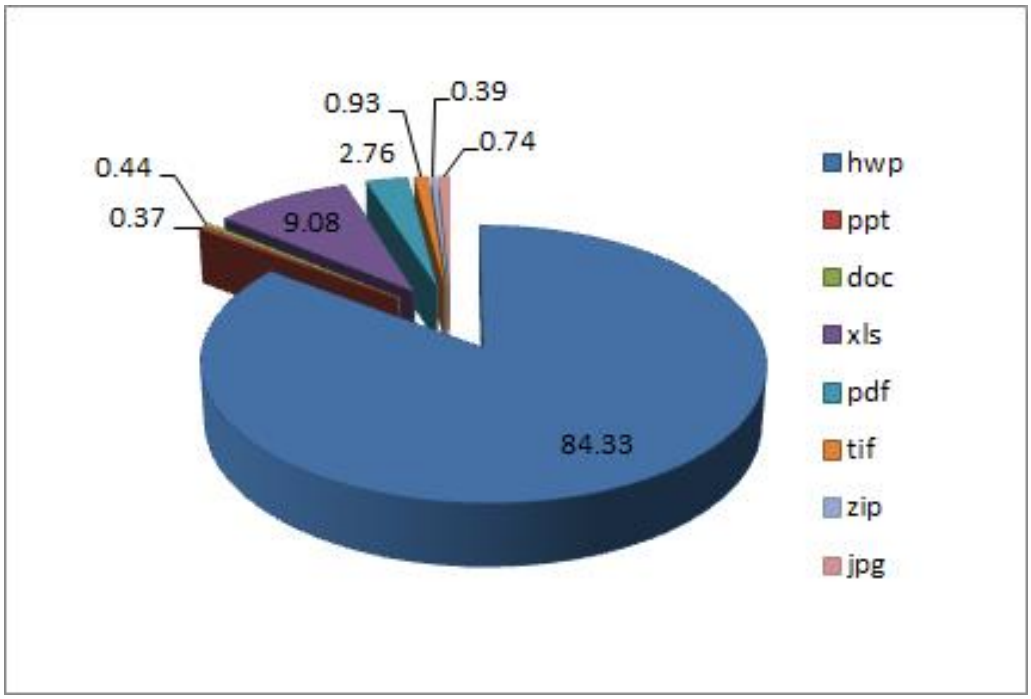
❒ 에뮬레이션 대상이 되는 기능성이란, 디지털 컴포넌트별로 세부 보존할 기능성이 다르기 때문에 본 연구에서는 우선 우리나라 공공기관에서 에뮬레이션 대상 디지털 컴포넌트를 정의하고 개별 디지털 컴포넌트별로 기능성 정의가 필요함을 제안함

❒ 에뮬레이션 대상 디지털 컴포넌트 정의 필요성

- ○ 현재 우리나라 공공기록을 대상으로 에뮬레이션 보존전략을 도입할 때 가장 역점을 두어야 하는 분야는 공공기관에서 사용 중인 디지털 컴포넌트의 분석 및 에뮬레이션 전략의 적용이 필요한 기능성의 파악임
- ○ 현재 우리나라 공공기록에 대한 장기보존 전략은 PDF/A로 마이그레이션하는 전략을 취하고 있지만, PDF/A는 오디오, 비디오, 자바스크립트 등 동적인 요소들의 사용이 제한되어 동적 요소가 포함된 한/글, 엑셀, 파워포인트 등의 문서를 PDF/A로 변환할 경우 외형만 재현하고 본문에 첨부된 동영상이나 움직임 효과 등이 렌더링 되지 않아 기록물 생산자의 의도나 내용의 맥락을 완벽하게 보존하지 못하는 한계가 발생
- ○ 에뮬레이션 전략은 고도의 기술력 및 에뮬레이터 개발 비용이 소요되며, 아울러 소프트웨어의 급진전 및 다양화에 따라 향후 에뮬레이터의 개발이 수시로 이루어져야 함을 염두에 둘 때, 현행 PDF/A를 기반으로 한 마이그레이션 전략을 기반으로 하되 PDF/A에서 구현이 어려운 기능성을 파악하여 에뮬레이션 전략을 적용하는 것이 최선의 방안임
- ○ 이에 따라 전자기록에 포함된 디지털 컴포넌트의 포맷을 분석하여 시간이 경과한 후에도 재현이 가능하도록 PDF/A 형식의 문서보존포맷으로 변환 시 소실되는 기능성을 파악해, 이들 기능성을 지닌 디지털 컴포넌트에 초점을 맞추어 에뮬레이션 전략을 적용하는 것이 활용 가능한 인적 물적 자원으로 최대한의 장기보존 효과를 가져 올 수 있는 방안이라 할 수 있음
- ○ 우리나라 공공기관에서 사용하고 있는 대부분의 응용프로그램은 마이크로소프트사의 윈도우 환경에서 구동되고 있으나 향후에는 지금과 전혀 다른 운영체제 및 컴퓨팅 환경이 지원될 기능성도 있음으로, 기존의 응용프로그램을 더 이상 지원하지 않는 경우가 발생할 수 있음
- ○ 현재 버전의 응용프로그램을 보존하더라도 새로운 시스템 환경에서 동작하지 않을 것으로 예측되고 있음

○ 따라서 현 시점에서 가장 중요한 것은 장기 보존이 필요한 디지털 컴포넌트의 포맷 형식을 정확하게 파악하여 레지스트리를 구축하는 것이 필요함
○ PDF, JPG, MPEG 등 포맷 정보가 표준화되어 공개된 디지털 컴포넌트는 다양한 응용프로그램에서 조회가 가능하므로 변화된 컴퓨팅 환경에서 조회가 가능하지만, HWP, PPT, XLS 등 특정 사기업에 의존한 포맷을 사용하는 디지털 컴포넌트는 기업에서 지속적으로 응용프로그램을 제공해야만 조회가 가능함
○ 이에 따라 우리나라 공공기관에서 가장 일반적으로 사용되는 디지털 컴포넌트의 기능성을 분석한 다음, 이러한 기능성을 지닌 전자기록을 대상으로 에뮬레이션 전략을 적용하는 방안을 모색해볼 필요가 있음

❒ 에뮬레이션 대상 디지털 컴포넌트 정의
○ 현행 우리나라에서는 공공기록이 생성·활용되는 플랫폼이 한정적이며 특정 포맷의 디지털 컴포넌트가 대다수를 차지하는 양상을 보이고 있음
○ 본 연구 사업에서 실시한 방송통신위원회를 대상으로 기록관리시스템에 등록된 디지털 컴포넌트의 유형에 대한 조사에 따르면, 한/글(hwp) 파일 포맷이 84.33%를 차지하여 거의 대다수 파일이 아래아한글로 작성됨을 알 수 있고, 다음으로 마이크로소프트사의 엑셀(xls) 파일 포맷이 9.08%를 차지하여 전체 디지털 컴포넌트 중 2가지 포맷이 차지하는 비중(93.41%)이 거의 대부분임이 파악됨
○ 그리고 파워포인트(ppt) 파일 포맷으로 작성된 디지털 컴포넌트의 비중은 0.37%로 다른 포맷 유형보다 적은 반면 pdf 파일 포맷은 2.76%를 유지하여 비교적 많이 사용됨이 파악됨
○ 아래의 그림에서와 같이 아래아한글(hwp)과 마이크로소프트의 엑셀(xls) 포맷이 93.41%로 가장 많으며, 8종(hwp, xls, ppt, doc, pdf, jpg, tif, zip) 포맷이 99.04%로 거의 대부분을 차지함에 따라 향후 에뮬레이션 전략을 수립함에 있어 가장 먼저 아래아한글 포맷과 마이크로소프트의 오피스 포맷에 대한 연구가 시급함을 파악할 수 있음
○ 아울러 pdf 및 이미지 파일 포맷(tif, jpg)은 포맷 정보가 표준으로 공개되어 있을 뿐만 아니라 정적 이미지 정보로 구성되어 있으므로 에뮬레이션 전략 수립 대상에서 제외할 수 있다고 판단됨




<그림 3-49> 우리나라 공공기록물의 디지털 컴포넌트 유형 분포

○ 이에 에뮬레이션 대상 디지털 컴포넌트 유형으로 hwp, xls, ppt를 선정한 다음 각각의 디지털 컴포넌트가 지닌 기능성을 파악할 필요가 있음

❒ 에뮬레이션 대상 디지털 컴포넌트별 기능성

○ 에뮬레이션 대상이 되는 디지털 컴포넌트별 기능성은 디지털 컴포넌트를 생산하는 애플리케이션 버전 및 파일 포맷 버전별로 다름. 본 연구에서는 3.3.2 보존포맷 변환시 소실되는 렌더링 객체 속성분석을 통하여 한/글, 엑셀, 파워포인트에서 PDF/A 변환시 소실되는 기능성들을 분석하였음.

○ 상기의 분석결과를 정리한 것이 아래의 <표 3-21>인데 이것은 우리나라 행정기관에서 대부분을 점하는 포맷들이 인쇄 렌더링 과정에서 소실한 기능성임. 이것은 일차적으로 에뮬레이션 대상 디지털 컴포넌트별 기능성의 대상이 될 것임[75]

○ 그러나, 이러한 결과만으로는 에뮬레이션 대상 기능성 자체라고 할 수 없음. 예를들어 한/글의 교정부호, 글맵시, 책갈피, 스크립트 등은 공공기관에 있어 굳이 보존할 기능성이라 판단하게 어려움.

○ 최종적으로 디지털 컴포넌트별 기능성 중에서 보존대상 기능성으로서 에뮬레이션이 필요한 기능성을 선별하기 위한 추가적인 활동이 필요할 것임.

75) 각 디지털 구성요소별 동적 속성 요소들에 대한 상세한 기능 설명에 대해서는 3.3.1 보존포맷 변환시 소실되는 렌더링 객체 속성 분석 을 참조.

<표 3-21> 디지털 컴포넌트별 인쇄 렌더링시 소실되는 기능성

| 구분 | 기능성 명칭 | 설명 |
|---|---|---|
| 한글 | OLE 개체 | 문서 내에 Windows에 설치된 프로그램들이 자신의 기능을 서로 사용하는 기능 |
| | 메모장 | 단어나 문장을 선택하고 부가적인 설명을 추가하는 기능 |
| | 교정부호 | 단어나 문장이 틀렸을 때 수정할 수 있게 하는 기능 |
| | 양식 개체 | 명령단추, 선택상자, 라디오단추, 입력상자, 목록상자 등을 사용하는 기능 |
| | 글맵시 | 문서 내에서 글씨를 아름답게 표현하는 기능 |
| | 책갈피 | 문서를 편집하는 도중에 표시를 해 두었다가 바로 이동할 수 있게 하는 기능 |
| | 스크립트 | 양식 개체와 연결하여 여러 가지 선택을 할 수 있게 하는 기능 |
| 파워포인트 | 삽입된 비디오 파일 | 문서 내에 비디오 파일을 플레이 할 수 있게 하는 기능 |
| | 삽입된 오디오 파일 | 문서에서 오디오 파일을 삽입할 수 있고, 오디오 링크를 클릭하여 오디오를 플레이 하는 기능 |
| | OLE 개체 | Windows에서 제공하는 외부 응용 프로그램의 결과물을 삽입하는 기능 |
| | 애니메이션 | 프리젠테이션을 하기 위해 화면에 나타내기, 날아오기 등의 기능을 사용하여 부가적인 설명을 하는 기능 |
| 엑셀 | 스파크라인 | 단일 워크시트 셀 안의 작은 차트로서 데이터의 추세를 시각적으로 표현하는 기능 |
| | 메모장 | 셀 단위로 부가적인 설명을 추가하는 기능 |
| | 차트 | 셀 단위의 정보를 이용하여 시각적으로 표현하는 기능 |
| | 수식정보 | 셀 단위의 정보를 이용하여 셀에 합계, 평균과 같은 수식을 이용하여 계산하는 수학적 기능 |
| | 접힌 셀 | 셀이 접혀있는 경우, PDF/A로 변환할 때 접힌 상태로 변화되기 때문에 문제 발생 |

## 나. 에뮬레이션을 위한 보존 평가 방법

### (1) 목적과 범위

❒ 에뮬레이션을 위한 보존 평가 방법은 공공기관에서 생산되는 디지털 컴포넌트 중에서 보존할 속성으로서 기능성을 가진 것들을 선별하기 위한 기준 및 절차를 마련하려는 목적을 가짐

❒ 기능성이란 에뮬레이션의 대상이 되는 기능성으로서 마이그레이션과 같이 인쇄 방식의 렌더링을 취했을 때 사라지기 때문에 별도의 조치를 통해 생산단계에서부터 관리해주어야만 하는 것으로 “인쇄 렌더링으로는 표현할 수 없는 기능성”을 의미하며 에뮬레이션의 방식을 통해서 보존이 필요한 대상임

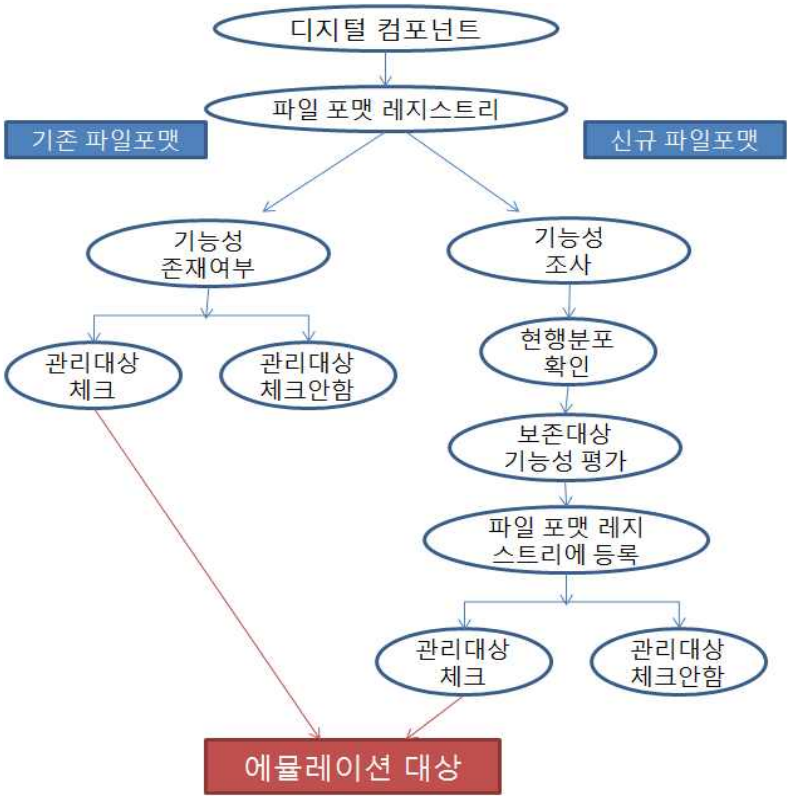
❒ 에뮬레이션을 위한 보존 평가 방법을 통해 디지털 컴포넌트별로 에뮬레이션이 필요한 기능성을 정의하고 기록 생산시점에 디지털 컴포넌트가 보유한 기능성을 식별하여 기록물관리기관에서 해당 디지털 컴포넌트에 적절한 렌더링을 제공하기 위한 조치와 관련된 활동을 지원할 수 있음

❒ 기록물관리기관은 에뮬레이션을 위한 보존 평가 방법을 통해 에뮬레이션이 필요한 대상을 식별하기 위한 기준과 절차를 마련함으로써 공공기관에서 필요한 에뮬레이션 도구, 환경을 준비하는데 도움을 얻을 수 있음

❒ 보존 평가 방법은 에뮬레이션 대상을 현행적으로 파악함으로써 현실에 적합한 장기보존 전략으로서 에뮬레이션의 필요성과 범위를 구체화하기 위한 방안임
  ○ 본 연구에서 제안한 에뮬레이션 전략 1안, 2안의 방식을 결정하기 위한 데이터로서 사용되어야 할 것임.
  ○ 특히, 한정한 환경과 포맷에 기반 하는 애플리케이션 에뮬레이션의 경우 구체적인 에뮬레이션 대상을 파악하는데 활용될 수 있을 것임

### (2) 에뮬레이션을 위한 보존 평가 방법 개발

❒ 보존 평가 절차의 개요
  ○ 전자기록의 생산 이전단계에서 디지털 컴포넌트별 에뮬레이션이 필요한 기능성에 대한 기준 마련
  ○ 디지털 컴포넌트 포맷별 인쇄 렌더링시 소실되는 기능성 조사, 소실되는 기능성 중에서 보존할 기능성에 대한 평가 및 선별
  ○ 이 방법론을 통해 생산 단계에서 에뮬레이션 대상을 파악할 수 있게 되면 장기적이고 체계적인 에뮬레이션 전략을 고안할 수 있을 것임
  ○ 생산 이전단계에서 에뮬레이션을 위한 보존 평가 절차의 세부 프로세스는 다음의 <그림 3-50>과 같음




**<그림 3-50> 에뮬레이션을 위한 보존 평가 절차**

❒ 파일포맷 레지스트리에 등록 여부 조사

- ○ 파일포맷 레지스트리에 등록되어 있는지 여부를 조사
- ○ 앞으로 기능성 요소가 포함된 다양한 포맷들이 계속적으로 등장할 것이기 때문에 이러한 등록 여부의 조사 필요
- ○ 파일포맷 레지스트리에 이미 등록된 경우, 기존 디지털 컴포넌트에 대한 기능성 요소의 존재 여부를 평가

❒ 기능성 존재 여부 판단

- ○ 디지털 레지스트리 가운데 기능성 요소가 포함된 경우에는 생산시스템의 컴포넌트 창에 에뮬레이션 관리대상임을 체크
- ○ 기능성 요소가 포함되어 있지 않을 경우에는 관리대상 여부 체크 안함

❒ 기능성 요소 조사

- ○ 신규 파일포맷의 경우에는 디지털 레지스트리 가운데 인쇄 방식의 렌더링으로 표현할 수 없는 기능성 요소를 조사

❒ 현행 분포 확인

- ○ 신규 파일포맷의 경우 현재 각 기관에서 해당 파일이 어느 정도 생산되고 있는지 여부를 확인

❒ 보존 대상 기능성 판단

○ 디지털 컴포넌트의 기능성 중에서 보존 대상 기능성 여부를 판단

○ 보존 대상 평가기준은 디지털 구성요소 가운데 기능성이 포함되었는지 여부를 업무담당자가 체크하는 것으로, 그 기준은 다음과 같음

<표 3-22> 기능성 평가 기준(안)

| 디지털 구성요소 | 보존속성 평가기준 | 평가기준 설명 |
|---|---|---|
| 아래한글 | OLE 개체 | 프로그램들이 각자 기능을 상호 사용 |
| | 메모장 | 부가적인 설명 추가 |
| | 교정부호 | 교정부호를 사용해 수정 |
| | 양식 개체 | 명령단추, 선택상자, 라디오단추, 입력상자, 목록상자 |
| | 글맵시 | 글씨의 폰트 변형 |
| | 책갈피 | 문서 편집시 바로 이동 가능 |
| | 스크립트 | 양식 개체와 연결해 복수 선택 기능 |
| MS 파워포인트 | 삽입 비디오 파일 | 비디오 파일 삽입, 플레이 |
| | 삽입 오디오 파일 | 오디오 파일 삽입, 플레이 |
| | OLE 개체 | 외부 애플리케이션 결과물 삽입 |
| | 애니메이션 | 프리젠테이션 기능 |
| MS 엑셀 | 스파크라인 | 작은 차트로 데이터 추세 표현 |
| | 메모장 | 셀 단위로 부가적인 설명 추가 |
| | 차트 | 그래픽 차트 재현 |
| | 수식정보 | 합계, 평균 등 연산기능 |
| | 접힌 셀 | 셀 감추기 기능 |

○ 보존 대상 기능성 판단 시기는 전자기록이 생성시스템에서 만들어진 직후 업무담당제에 의해 수행되는 것으로, 이 시기 이후에는 전자기록이 지닌 양적 시간적 문제로 평가하기가 실질적으로 곤란함

○ 평가 대상은 신전자문서시스템 및 업무관리시스템에서 생성된 전자기록 가운데, 동적 속성을 포함하는 한/글, 파워포인트, 엑셀을 대상으로 하며, 보존기간이 30년 이상으로 책정된 영구기록물관리기관으로 이관되는 전자기록을 개상으로 함

❒ 파일포맷 레지스트리 등록

○ 신규 파일포맷의 제반 정보 및 기능성 요소를 파일포맷 레지스트리에 등록

❒ 관리대상 체크

○ 기능성 요소가 포함된 경우 생산시스템의 컴포넌트 창에 에뮬레이션 관리대상임을 체크

○ 기능성 요소가 포함되어 있지 않을 경우에는 관리대상 여부 체크 안함

❒ 에뮬레이션 대상 파악

○ 기존 파일포맷 및 신규 파일포맷 중 기능성 요소가 존재하는 것으로 체크된 디지털 컴포넌트를 에뮬레이션 대상으로 지정

○ 에뮬레이션이 필요한 디지털 컴포넌트에 대한 생산 당시로부터의 식별을 통해 장기적이고 체계적인 에뮬레이션 전략을 수립할 수 있음

### (3)기록관리 단계별 역할 구분

보존 대상 기능성 평가와 관련하여 업무담당자, 기록관, 영구기록물관리기관 단계별 역할을 요약하면 <그림 3-51>과 같음




**<그림 3-51> 기록관리 단계별 역할 개요**

❒ 생산자(업무담당자)

○ 보통 첨부물의 형태인 디지털 컴포넌트 안에 인쇄 렌더링으로 재현될 수 없는 기능성의 존재여부를 가장 정확하고 빠르게 판단 가능한 주체는 생산자임

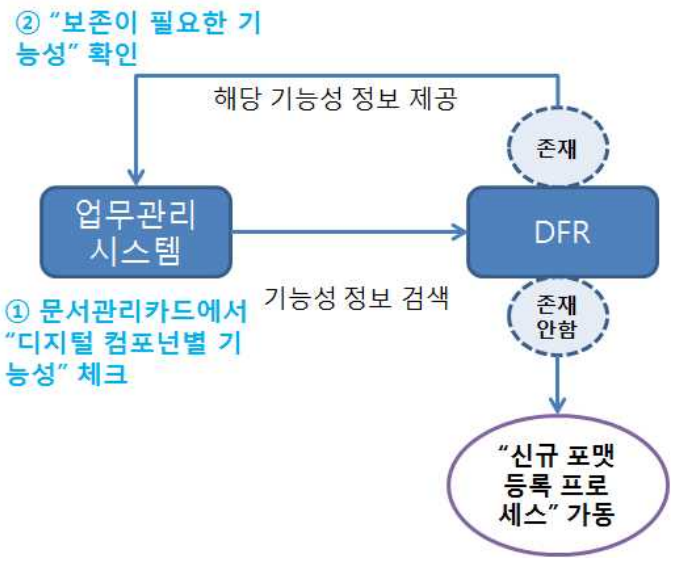
○ 기능성 평가절차 상의 업무 담당자 역할은 다음과 같음 :

- 파일포맷은행 등록 여부 조사
- 기능성 존재여부 평가하며 업무관리시스템 『문서관리카드』의 첨부파일 레벨에서 실시




<그림 3-52> 디지털 컴포넌트별 보존할 기능성을 체크하는 UI (안)

- 위의 <그림 3-52>에서 업무담당자는 각 첨부파일 포맷별 기능성 중에서 특정 기능성을 보존할 필요가 있는가의 여부를 체크해 주면 됨
- 한편, 업무담당자가 체크하는 기능성 요소는 DFR 등 에뮬레이션이 필요한 파일 포맷의 기능성 정보를 관리해주는 레지스트리와 연계되어야 할 것이며 아래의 <그림 3-53>은 그것을 표현하고 있음




<그림 3-53> 기능성 평가와 DFR 관계

- 이 과정에서 에뮬레이션을 위한 기초 자료가 축적될 수 있으며 현황에 근거한 체계적인 장기보존 전략을 수립할 수 있게 될 것임

❒ 기록관

○ 생산 당시 보존할 기능성 여부가 판단되어 기록관에 이관된 전자기록을 대상으로 에뮬레이션을 위해 필요한 조치를 수행

○ 기록관 단계에서 수행하게 되는 역할은 다음과 같음 :

- 기록관에 이관되어 온 전자기록을 대상으로 기능성 대상 확인 및 관련 통계정보 작성, 유지
- 기능성을 지닌 디지털 컴포넌트에 관한 DFR 관련 국가기록원과 협조 체제 유지
- 업무담당자 대상의 교육 프로그램 제공
- 기능성 보유 디지털 컴포넌트의 일괄 리스트화 및 영구기록물관리기관 보고
- 신규 파일포맷상의 기능성 존재여부 조사 및 영구기록물관리기관에 통보

❒ 영구기록물관리기관(국가기록원)

○ 기능성을 지닌 디지털 구성요소에 대한 에뮬레이션 구현 및 절차 운영의 총괄적 책임을 담당

○ 영구기록물관리기관(국가기록원)이 수행하게 되는 역할은 다음과 같음

- 기능성 요소를 지닌 디지털 컴포넌트를 위한 DFR 구축 및 유지 관리
- 기능성을 지닌 디지털 컴포넌트의 장기보존을 위한 에뮬레이션 기술적 동향 관리
- 기록생산시스템에 디지털 컴포넌트별 기능성 체크 기능 구현
- 자동화된 기능성 필터링 기능 구현
- 기능성을 지닌 디지털 컴포넌트를 재현할 소프트웨어 확인 및 보유
- 실제 에뮬레이션 실시
- 에뮬레이션 방식에 대한 총괄적 책임 및 역할 수행

### 3.2.3 전자문서 이외의 디지털 컴포넌트 유형 보존 속성 연구

❒ 본 연구과제는 주로 행정기관에서 생산된 전자문서 기반의 전자기록물의 기능성을 재현함으로써 '외형과 느낌'을 보존하기 위한 에뮬레이션 전략을 다루고 있음. 이러한 디지털 보존을 위한 에뮬레이션 전략은 인쇄 렌더링으로 재현할 수 없는 기능성을 보여줄 수 있다는 잠재력으로 그 가능성을 인정받고 있으며 공공 기록물관리 영역에서 더 다양한 디지털 컴포넌트 유형에 대한 장기보존 전략으로서

적용 가능성을 타진해 봐야 할 것임

❒ 에뮬레이션 전략 적용의 필요성 차원에서 특히 주목해볼 만한 것은 특히, 딥웹(Deep web), 데이터 셋트(Data set), SNS(Social Network)임. 이들은 현재 행정기관에서 활발하게 생산, 활용되고 있고 이들의 장기보존 필요성에 대한 인식이 어느 정도 형성되어 있음

❒ 이에 본 절에서는 장차 행정기관에서 에뮬레이션 전략이 적용될 주요한 대상으로서 판단되는 딥웹(Deep web), 데이터 셋트(Data set), SNS(Social Network)에 관한 국내외 연구 및 기술현황을 토대로 에뮬레이션 전략의 적용 필요성 및 방향에 대한 제언을 포함하고자 함

### 가. 배경 및 필요성

에뮬레이션 전략의 적용 필요성을 찾기 위해서는 현재까지 각 디지털 컴포넌트 유형별로 진행되어온 기록화 현황과 그것이 가지는 문제점을 고찰할 필요가 있음.

#### (1) 딥웹(Deep web)

❒ 딥웹(Deep web) 정의
- ○ 웹의 경우 일반적인 검색엔진으로 접근이 용이한  표면웹(surface web)과 일반적인 검색도구로 접근이 불가능한 딥웹(deep web)으로 구분됨
- ○ 심층 웹 기록이란 표면 웹 기록물 아카이빙에서 수집 불가한 관계형 데이터베이스에 저장되어 동적으로 페이지를 생산하는 데이터[76]를 의미함
- ○ 일반적으로 심층 웹 기록은 사용자들의 접근이 불가능한 데이터베이스를 통해 구성되며 사용자의 요구가 변경될 때마다 내용이 갱신되는 특징을 가짐[77]

❒ 국내 공공 딥웹 기록물 장기보존의 필요성
- ○ 국내 공공 딥웹 기록물의 경우 2009년 국가기록원에서 딥웹 기록물의 장기보존과 관련한 선행연구가 수행됨[78]
- ○ 국내 공공기관의 웹사이트가 포함하고 있는 웹 기록은 다음과 같이 유형화하고 정의할 수 있음
  - 웹문서

76) 이규철 외, 「공공기관 심층 웹 기록물 아카이빙을 위한 머타데이터 설계」, 한국전자거래학회지 제 14권 제4호, 2009
77) 이규철 외, 국가 심층 웹기록물 보존 포맷과 OAIS 참조모델 간의 데이터 맵핑, 2010 한국 컴퓨터 종합학술대회 논문집, 2010
78) 국가기록원, 웹기록물 아카이빙 기반기술 연구 및 적용시험, 2009

- 크롤러를 이용하여 수집할 수 있는 텍스트나 이미지로 구성된 웹페이지

• 전자문서

- 워드, 엑셀, PDF 등과 같은 문석 파일을 제공하는 웹 페이지

• 데이터베이스

- 데이터베이스의 데이터들로 동적으로 구성되는 웹페이지

• 멀티미디어

- 사운드, 이미지를 제공하는 웹페이지

○ 웹문서, 전자문서, 멀티미디어의 경우 웹 아카이빙을 통해 보존이 가능하지만 데이터베이스 경우 데이터의 동적인 구조의 특성으로 기존의 표면 웹 기록물 보존방식으로 보존할 수 없음

○ 즉, 국내 공공기관에서 생산하는 딥웹 기록의 경우 앞서 유형화 한 표면웹에서 수집 불가능한 데이터베이스에 저장되어 동적으로 페이지를 생성하는 데이터이며 이러한 특성을 장기보존 전략에 반영해야 함.

❒ 관계형 데이터베이스에 저장된 동적인 데이터인 딥웹 기록물의 장기보존 포맷을 개발하였으며, 'KoDeWeb'(Korea Deep Web records)로 명명하였음

○ KoDeWeb은 XML로 구성되며 소프트웨어나 하드웨어 독립적이고 하나의 포맷으로 구성되어 문서보존 포맷으로 전송 및 보존 과정의 손실이 적음

○ 국내 공공기관 딥웹 기록물 장기보존 포맷인 KoDeWeb은 KoDeWeb 메타데이터와 KoDeWeb 컨텐츠로 구분됨

• KoDeWeb 메타데이터

- KoDeWeb의 메타데이터는 기록물건 메타데이터[79], 심층웹 메타데이터[80], 기술메타데이터[81]로 구성되어 있음

• KoDeWeb의 실질적인 보존 대상이 되는 컨텐츠는 데이터베이스의 정보의 포맷과 마찬가지고 데이터베이스의 내용에 해당하는 테이블들을 XML파일로 저장하고 이를 ZIP으로 압축하여 저장함

### (2) 데이터 세트

❒ 데이터 세트 정의

○ 컴퓨터가 처리하거나 분석할 수 있는 형태로 존재하는 관련 정보의 집합체로 데이터 파일이나 데이터베이스와 동의어로 사용되기도 함[82]

○ 컴퓨터가 분석할 수 있도록 형식을 갖춘 연관성 있는 정보들의 집합

79) 식별자, 기술정보, 기록물유형, 기록물형식, 키워드, 관련법규, 생산자, 생산일시, 출처정보, 보존일시, 수집자, 공개범위, 이용정보, 관련정보, 수용범위, 언어, 권한등급, 사용자, 제목, 소스

80) 기여자, 스키마, 언어, 역할, 이용자, 용량, 권한정보, 범위

81) 저장포맷 버전, 아카이빙 도구, 데이터베이스, 연결정보, 호스트정보

82) 한국기록학회, 기록학 용어사전, 역사비평사, 2008

○ 행정정보 데이터세트는 행정정보시스템을 통해 생산되고 축적되는 데이터세트 즉, 전자문서시스템, 업무관리시스템을 제외한 모든 행정정보시스템에서 생산되는 데이터 세트임[83]

❒ 행정정보시스템 데이터세트 장기보존 연구 필요성

○ 행정정보시스템에서 축적되는 기록은 업무활동의 증거이자 대체할 수 없는 성격을 갖는 정보자원으로 기록화 및 관리가 필요함

○ 현재 행정정보시스템의 기록관리관점에서 현황 파악이 안되어 기록화 어려움

○ 행정정보시스템의 현황 파악 및 유형기준 마련과 함께 기록화 대상에 대한 연구가 필요

### (3) SNS

❒ 행정기관의 SNS 정의

○ Social Network Service는 개인들 간의 관계를 통해 형성된 연결을 공유하고 그 연결을 바탕으로 일어나는 개인들 간의 상호작용을 지원하는 웹 기반 서비스로 웹 2.0의 정신을 구현하는 핵심적인 도구[84]

❒ 공공 SNS 기록의 장기보존 필요성

○ 현재 우리나라 페이스북, 트위터와 같은 SNS 이용자는 꾸준히 증가하는 추세에 있음

○ 개인을 비롯한 민간영역에서 부터 중앙행정기관 및 지방자치단체에 이르기까지 공공영역에서도 SNS를 꾸준히 운용하고 있으며 공공기관의 SNS수요 역시 증가하는 추세임

○ 공공기관에서 SNS 활용의 목적 및 기대효과는 다음과 같음[85]

- 국민들과의 소통 및 관계개선
- 공공기관의 이미지 제고
- 미디어 채널의 구축
- 정책인지도의 강화
- 대국민 의견 수렴
- 이벤트의 활성화
- 이슈 및 위기관리

○ 웹 또는 모바일 기반의 공공영역의 SNS는 국민과의 소통데이터의 분석을 통해

83) 조은희, 임진희, 「행정정보 데이터세트 기록의 기준 및 절차연구」, 『기록학연구』 제 19호, 2009
84) 송주형, 「기록관리 대상으로서 SNS연구」, 한국외국어대학교 석사학위 논문, 2012
85) KPR 소셜 커뮤니케션 연구소에서 2013년 국내 기업 및 공공기관 소셜 미디어 운영 현황 및 전망조사를 수행하였음

공공영역의 정책 마케팅 및 이미지 관리에 활용함으로써 다양한 시너지 효과를 기대할 수 있어 기록화와 장기보존이 필수적임

❒ SNS 기록물 장기보존을 위한 국내외 동향

○ SNS 기록의 장기보존을 위한 전략으로 국내외 다양한 시도가 있었으며 해외 SNS장기보존 연구 및 서비스 현황은 다음 <표3-22>와 같음

**<표 3-23> 해외 SNS 장기보존 연구 및 서비스 현황**

| 구분 | 개요 | 시행 서비스 |
|---|---|---|
| Archive-iT | NASA에서 우주항공기관의 소셜미디어 행위를 저장하기 위한 목적을 가진 프로젝트 웹 아카이빙 방식으로 소셜네트워크서비스 기록물을 보존 | 페이스북 이외에도 유투브, 플리커, 유스트림, 마이스페이스등을 저장 |
| Facebook Archives | 페이스북에서 제공하는 공식기능으로써 개인 계정의 sns기록물을 보존 | 개인 계정의 뉴스피드의 HTML 문서와 사진, 동영상 파일을 다운로드 |
| Archive Facebook | 웹 브라우저인 파이어폭스(FireFox)의 페이스북 sns 기록 보존 | 개인 계정의 뉴스피드의 HTML 문서와 사진, 동영상 파일을 다운로드 |
| Arkovi | BMRW & Associates, INC에서 제공하는 상업용 도구로서 페이스북을 비롯해 트위터, 웹 아카이빙, RSS 기록 아카이빙을 지원 | 기록물을 직접 저장하는 것이 아니고, 데이터만 가져와서 저장. 사잔의 경우 자체적으로 저장하지 않고 URL정보만 가져와 링크를 시켜줌 |
| Social Safe | Arkovi와 마찬가지로 데이터만 가져와서 저장해주는 형태 | API기반의 기록물 백업방법을 사용하여 기록물들을 개인의 컴퓨터에 저장 및 관리 |

○ 국내에서는 행정기관의 SNS 기록물의 수집, 보존, 재현에 대한 연구가 선행적으로 수행된 바가 있음[86)]

- 행정기관에서 운영하는 SNS 기록물의 수집방안 및 도구와 장기보존을 위한 보존 전략을 개발함
- 대표적인 SNS인 페이스북/트위터의 재현 도구 및 브라우저를 개발하여 보건복지부(트위터), 청와대(페이스북)을 재현함

86) 국가기록원, 행정기관의 SNS 기록물 수집·보존·재현 도구 연구개발, 2011

## 나. 주요 유형별 에뮬레이션 고려사항

❒ 디지털 컴포넌트별 기록화와 관련된 현황을 보았을 때 공통적으로 주목해야 할 것은 사전적인 평가 부재임. 즉, 각 디지털 컴포넌트를 기술적으로 재현하고자 하는 목적에 충실한 나머지 기록 관리적으로 의미 있는 맥락이 함께 고려되고 있지 못함

❒ 따라서 기록화 대상에 따른 선별 및 획득 대상에 대한 고려 없이 기록화 된 결과물이 이후 관리 및 활용에 대한 제약사항으로 이어짐. 한번 기록화 된 결과물이 이후로의 서비스 형태를 좌우함. 따라서 어떤 방식으로 기록화 하느냐에 따라 재현 가능한 기능성에 제약을 가질 수 있다는 고려가 반드시 필요함.

### (1) 딥웹(Deep web)

❒ 딥웹 에뮬레이션 방안

○ 데이터베이스에서 저장된 동적인 데이터를 공공기관의 딥웹으로 정의한 선행연구의 결과를 바탕으로 함

○ 우선적으로 보존할 기능성을 평가한 후에 기록화 방식을 결정하는 순서로 진행되어야 할 것임

○ 딥웹 에뮬레이션의 경우 동적 데이터가 데이터베이스상에 저장된다는 성격을 가지므로 이하에 기술될 데이터세트 에뮬레이션 방안과 동일한 맥락에서 방안을 모색할 수 있음

### (2) 데이터 세트(data set)

❒ 데이터 세트 에뮬레이션 방안

○ 데이터세트 기록화 이슈

- 기능성 재현이 필요한 대상이 존재하는지 여부 포함하여 사전적인 선별 및 획득대상을 식별해야 하며, 이를 위해 식별단위의 기준을 수립해야 함
- 행정정보시스템 설계 당시부터 기록관리 요건을 반영해야 함
- 데이터베이스의 테이블, 필드 등에 대한 기술이 필요함
- 데이터세트 유형에 맞춘 장기보존 전략이 필요함
- 데이터세트의 서비스 및 활용 방안에 대한 연구와 동반되어야 할 것임

○ 데이터세트 에뮬레이션의 이슈

- 현재의 데이터 세트 보존은 데이터 세트 자체를 보존하고 평범한 형태의 공통 UI를 통해 활용 시 내용을 재현하는 방식이 주를 이룸

- 네델란드 데이터세트 아카이브의 사례를 보면, 다양한 데이터세트가 UI를 통해 그 내용을 검색활용

• 하지만 데이터 세트는 다양한 유형을 가지며 데이터세트만 보존해도 되는 경우도 있지만 생산될 당시의 독특한 외형과 느낌(look & feel)을 함께 보존하는 것이 일정한 의미를 가지는 경우가 존재할 수 있음

- 기상관측 데이터의 경우는 기초 데이터(raw data)를 메타데이터와 함께 보존하는 방식으로도 무방함
- 하지만 인사기록카드, 학생생활기록부 등의 카드류 등은 해당 카드 형식(Form)이 중요하며 외형과 느낌을 살려 보여줄 필요가 존재함

• 이러한 특성을 고려하였을 때 데이터 세트의 에뮬레이션 문제는 해당 데이터의 외형과 느낌을 보여줄 수 있게 해주는 것이 의미 있는 데이터 세트 경우를 선별해 내고 그러한 경우에 필요한 애플리케이션 및 데이터베이스 스키마, 화면 설계서 등의 기술문서 등을 함께 보존하는 것이 핵심적인 이슈가 될 것임

• 데이터와 함께 보존될 응용프로그램의 역할은 데이터를 원래의 외형과 느낌에 해당되는 화면의 모습 그대로 보여주는(viewing) 기능을 수행

○ 에뮬레이션 방식에 따른 세부 방안

• 환경에 대한 전체ll) 에뮬레이션의 경우에는 해당 데이터세트를 생산당시의 응용프로그램의 환경에서 활용할 수 있기 때문에 “외형에 대한 보여주기가 필요한 데이터 세트”를 기본적으로 재현해줄 수 있음

• 하지만 그렇지 않은 경우는 에뮬레이션을 위해서 다음과 같은 요건들을 갖춰야 할 것임

- “외형에 대한 보여주기가 필요한 데이터 세트”가 무엇인지 선별
- 해당 데이터 세트의 내용을 보여주는 응용프로그램과 함께 영구보존
- 이때, 필요시 해당 응용프로그램을 개발해야 하며 개발될 소프트웨어의 기능은 보여주기 위한 화면과 관련된 기능들로서 한정될 수 있을 것임

### (3) SNS(Social Network Service)

❒ 공공영역의 SNS에뮬레이션 이슈

○ 현재 다양한 SNS 채널이 존재하고 있으며 우선적으로 해당하는 SNS의 특성에 적합하게 재현하고자 하는 기능성을 정의하고 에뮬레이션 전략의 수립

• 해당 SNS기록의 플랫폼을 정의하고 해당 플랫에 부합하는 에뮬레이션 전략의 성립이 필요함

- 기존의 페이스북, 트위터, 블로그와 같은 SNS기록물의 경우 웹기반 한 플랫폼의 SNS 기록임

- 수요가 급성장 하고 있는 모바일(mobile)플랫폼에 기반한 '카카오 스토리'[87]의 경우 모바일 환경에서의 외형과 느낌을 보존할 수 있는 에뮬레이션 전략이 필요함

• 생산된 SNS 기록에 포함되어 있는 다양한 디지털 컴포넌트의 재현을 위한 에뮬레이션 전략의 성립

- SNS기록에 포함되어 있는 이미지, 비디오, 음성 등 다양한 디지털 컴포넌트의 기능성을 보존할 수 있는 에뮬레이션 전략의 수립이 필요함
- 페이스북, 유투브, 모바일 어플리케션, 트위터 등 다양한 SNS간의 공유를 통한 상호 운용성을 반영할 수 있는 에뮬레이션 전략의 수립이 필요함

• SNS 채널이 다양해지는 것은 그에 대한 사용자의 인식을 동반하고 있으며 기록화 방식 역시 보다 이것을 반영하여 다양한 활용 방식에 대한 고민에 근거할 필요가 있음

- 최근의 카카오톡(KAKAO TALK), 밴드(BAND) 등의 서비스에서는 다양한 이모티콘[88]이 사용되는데 단순히 감정이나 느낌을 전달하는 것에서 나아가 이모티콘을 통한 대화 자체도 가능할 정도로 하나의 추세로 나타나고 있음. 이러한 변화하는 소통 방식에 대한 인식이 세대별로 차이가 있을 수 있겠지만 시대와 문화를 기록화 함에 있어서 이러한 변화상은 영역을 구분하지 않고 발생되며 의미를 가질 수 있다는 점을 고려할 필요가 있음

## 3.3 디지털 컴포넌트의 렌더링 기능성 유형 연구

### 3.3.1 장기보존 대상 디지털 컴포넌트의 포맷 변화 분석

#### 가. 포맷 변화 분석 배경 및 필요성

❒ 디지털 컴포넌트 포맷 변화 분석 배경

○ 기록물철/건에 포함된 디지털 컴포넌트의 포맷 유형을 조사하여 분석한 결과 한/글(hwp)과 마이크로소프트의 엑셀(xls) 포맷이 93.41%로 가장 많으며, 8종(hwp, xls, ppt, doc, pdf, jpg, tif, zip) 포맷이 99.04%로 거의 대부분을 차지함에 따라 향후 에뮬레이션 전략을 수립함에 있어 가장 먼저 한/글 포맷과 마이크로소프트의 오피스 포맷에 대한 연구가 시급하다고 판단

○ 또한, 보존문서 표준 포맷(pdf), 이미지 디지털 컴포넌트 포맷(tif, jpg)은 포맷 정보가 표준으로 공개되어 있을뿐만 아니라 정적 이미지 정보로 구성되어 있으

87) 카카오 스토리의 경우 KPR 소셜 커뮤니케션 연구소에서 수행한 2013년 국내 기업 및 공공기관 소셜 미디어 운영 현황 및 전망조사에서 페이스북, 유투브에 이어 세 번째로 새롭게 주목해야할 소셜미디어 채널으로 선정됨

88) Emotion + Icon의 합성어로 컴퓨터나 휴대 전화의 문자와 기호, 숫자 등을 조합하여 만든 그림 문자. 감정이나 느낌을 전달할 때 사용한다, 출처: 대중문화사전

므로 에뮬레이션 전략 수립 대상에서 제외할 수 있다고 판단

○ 따라서, 본 연구에서는 향후 동적 요소가 포함되어 있는 디지털 컴포넌트를 재현하는 대상으로 한/글과 마이크로소프트사의 엑셀, 파워포인트 (3종) 포맷을 대상으로 선정하여 에뮬레이션을 수행할 때 필요한 고려 사항들을 검토함으로써 공공기관에 필요한 에뮬레이션 전략을 수립하고자 함.

❒ 디지털 컴포넌트 포맷 변환 분석 범위 선정

○ 한/글 워드프로세서에 대한 에뮬레이션 전략을 수립하기 위해 1989년에 출시된 한/글 1.0부터 2010년에 출시된 한/글2010 까지 버전의 특징 및 주요 기능과 디지털 컴포넌트 포맷에 대한 분석이 필요하므로 한/글 워드프로세서에 대한 버전 조사 및 포맷 특성에 대한 분석 수행

○ 또한, 마이크로소프트사가 1985년에 처음 출시한 엑셀 1.0부터 2013년에 출시한 오피스2013까지 엑셀 및 오피스 프로그램의 버전별 기능 및 포맷 버전의 변화에 대한 분석 수행

○ 에뮬레이션을 위해서는 한/글 워드프로세서 및 마이크로소프트 오피스 프로그램이 실행되는 운영체제에 대한 정보가 중요하므로 국내 공공기관에서 가장 많이 사용되는 마이크로소프트사의 윈도우 운영체제 버전과 실행 가능 디지털 컴포넌트에 대한 정보 분석 수행

○ 마지막으로 한/글 워드프로세서의 애플리케이션 버전과 디지털 컴포넌트 포맷 버전과의 관계 및 마이크로소프트 오피스의 애플리케이션 버전과 디지털 컴포넌트 포맷 버전에 대한 조사 분석을 통해 현재까지 등록된 전자기록물에 대한 에뮬레이션 대상을 선정

## 나. 한/글 애플리케이션 변화 분석

### (1) 국내 워드프로세서 관련 히스토리 조사

❒ 한/글 2.0이 발표된 1992년은 멀티미디어와 윈도우 운영 체제의 보편화가 시작된 해이기도 함. 마이크로소프트에서 1992년 4월 발표한 영문 윈도우 3.1을 다음해인 1993년 6월 한글 윈도우 3.1로 출시하면서 멀티미디어 PC 또는 홈 PC 라는 개념이 본격적으로 등장하기 시작하였는데, 한글 윈도우 3.1의 장점을 살린 여러 가지 한글 워드프로세서가 등장하였는데 한메소프트의 '파피루스', 금성소프트의 '원워드', 마이크로소프트의 '워드 5.0', 삼성전자의 '훈민정음', 핸디소프트의 '아리랑', 휴먼컴퓨터의 '글사랑' 등이 등장하면서 한글 워드프로세서 춘추전국시대가 시작

❒ 1990년대 초반부터 등장한 워드프로세서는 각자의 포맷 형식을 정의하여 사용하

기 시작하였는데, 포맷 형식이 다르더라도 큰 불편을 느끼지 못한 이유는 워드프로세서를 사용하여 출력한 종이가 의사 전달의 수단이었기 때문임. 그러나, 1990년대 중반으로 갈수록 워드프로세서로 작성한 디지털 컴포넌트를 공유하기 시작하면서 디지털 컴포넌트 포맷을 표준화할 필요성이 강하게 대두되었으며 XML을 비롯한 다양한 표준 포맷에 대한 의견이 제시되었으나 국내에서 가장 많은 사용자를 확보한 한/글 워드프로세서의 디지털 컴포넌트 포맷이 국내 표준으로 자리잡기 시작함.

❒ 한글과컴퓨터에서는 윈도우 기반의 한/글 3.0판을 1995년 3월 출시하였는데 윈도우 한/글의 첫 출시는 마이크로소프트사의 윈도우95 출시와 함께 그 해 애플리케이션 시장의 가장 큰 이슈가 되었다. 윈도우 한/글은 지금까지 등장했던 다른 윈도우용 한글 워드프로세서와는 달리 운영체제에 의존하지 않는 독립 인터페이스, 독립 입출력 방식을 채택하였으며 완성형 한글 코드만을 지원하는 윈도우 3.1의 입력 체계를 사용하지 않고 도스용 한/글의 입력 체계인 조합형 한글 코드를 변형한 한컴 2바이트 코드를 사용함으로써 11,172자 현대 한글은 물론 옛한글까지 표현함

❒ 1996년 7월에는 한글과컴퓨터에서 다양한 애플리케이션와의 통합패키지로 한/글 프로96과 한/글 오피스 96을 출시하였는데 한/글 프로96은 사용자 인터페이스를 비롯해 인터넷 문서 편집, 문서모음집인 문서마당 등 다양한 기능이 업그레이드된 워드프로세서인 한/글 96을 기반으로 주소록, 타자연습, 폼프로세서인 틀마름이, 팩스 프로그램, 메일 프로그램 등 다양한 유틸리티를 추가한 종합 패키지로 변환함. 그러나 대중적으로 어느 정도 성공을 거둔 한/글 프로96과 달리 기업체 판매를 목적으로 출시된 한/글 오피스96은 포함된 로터스 1-2-3 5.0과 프리랜스 그래픽스 프로그램의 기능 미비로 마이크로소프트의 오피스에 비해 별다른 반향을 얻지 못함

❒ 1996년 11월에는 한/글 96을 기반으로 중국어, 일본어 인터페이스 및 다국어 입력기를 채용한 한/글 국제판과 일본 시장을 목표로 한 한/글 일본판이 출시되었고 1997년에는 한/글 97을 탑재한 한컴홈 97과 한컴 오피스 97이 출시되었는데, 한/글 97은 '실행 취소' 기능이 매우 취약했지만 역대 윈도우용 한/글 시리즈 중에서 가장 완성도가 높았으며 가장 높은 보급률을 보여준 제품이며 1990년대 중반부터 시작된 공공기관 업무 프로세스 시스템 구축 사업과 함께 공공기관에 도입되면서 가장 오랜기간 사용된 한/글 워드프로세서 버전으로 기억됨

❒ 그러나 1998년 6월에는 한글과컴퓨터가 마이크로소프트의 투자를 받는 조건으로

한/글 소스코드를 마이크로소프트사에 넘기고 더 이상의 한/글 개발을 중단하기로 하는 양해 각서를 마이크로소프트사와 체결해 큰 사회적 파장을 일으킴.

❒ 국내 사용자의 반대 여론에 부딪친 한글과컴퓨터는 이 계약을 취소하는 조건으로 '한글지키기운동본부'로부터 100억원 투자 유치를 받아 한/글815 버전을 출시하였고 2000년에는 한/글 워디안을 출시

### (2) 한/글 애플리케이션 개요

❒ 한/글 애플리케이션는 한글과컴퓨터에서 개발된 워드 프로세서로서, 이름은 현대 한국어에서 쓰이지 않는 ㆍ를 쓴 한/글로 표기하고, 'HWP(Hangul Word Processor의 준말)' · '아래아 한글' · '하안글' · '한/글' 등으로 대체하여 표기했으나, 국가 표준인 완성형 한글 코드에선 이 이름을 표현할 수 없기 때문에 2002년부터는 한/글로도 표기하기로 결정

○ 1998년 한글과컴퓨터에서는 현대 한국어에서 사용하지 않는 'ㆍ'를 쓴 한/글(아래아한글) 워드프로세서 최초 상용 버전 1.0판을 1989년 4월 24일 출시.

○ 한/글 워드프로세서는 이전까지 사용되던 보석글보다 우수한 위지위그 편집 기능, 8가지 한글/로마자 비트맵 글꼴 지원, 180 DPI 도트 매트릭스 프린터 지원, 특수 문자 입력 및 선 그리기 기능 등을 지원

○ 이 후 지속적인 기능 개선을 통해 1992년 일반용과 전문가용의 2.0판을 출시하였는데 그 기능을 간단히 살펴보면 다음과 같음

- 1포인트부터 127포인트까지의 글꼴 가변 크기 조정 기능
- 자간, 장평, 가변 조정 기능 및 문단 모양 편집 지원
- 글꼴과 문단 모양을 저장하여 사용하는 스타일 지정/적용 기능
- 한/글 2.0 전문용에서 비트맵 글꼴과 윤곽선 글꼴을 모두 지원
- 전문용에서 추가 확장팩을 통해 2수준 확장한자(4888자+1만여 자) 지원
- 수식 편집 기능
- 표를 작성하고 편집하는 기능,
- 다단 편집 기능, 글상자, 그림 등의 틀 자유 배치 기능
- 한글 맞춤법 검사 기능
- 하이퍼텍스트 방식의 도움말 개선
- 메모리 관리 방식의 변화로 최대 편집 쪽수의 증가
- 마우스 지원

❒ 2001년에 한/글 워디안을 기반으로 이전 버전 디지털 컴포넌트와의 호환성 문제 및 안정성 문제를 해결한 한/글2002를 출시하였으며 다양한 기능 개선을 추가한

한/글2004, 한/글2005, 한/글2006, 한/글2007, 한/글2008 (Linux 용)을 지속적으로 출시

❒ 한/글 2007의 차기작으로 리본 메뉴 인터페이스를 도입한 한/글2010을 2010년 3월 출시한 후 2010년 6월 HWP문서의 디지털 컴포넌트 형식을 공개하였는데 시스템 내용 구성이 이전의 2000 시리즈와는 완전히 다른 새로운 형태의 시스템으로 구성하였으며 한/글 2013을 출시 예정

### (3) 버전별 주요 기능 및 특징

❒ 1989년에 출시된 한/글 1.0부터 2010년에 출시된 한/글2010 까지 버전의 특징 및 주요 기능에 대한 지원 사항 분석

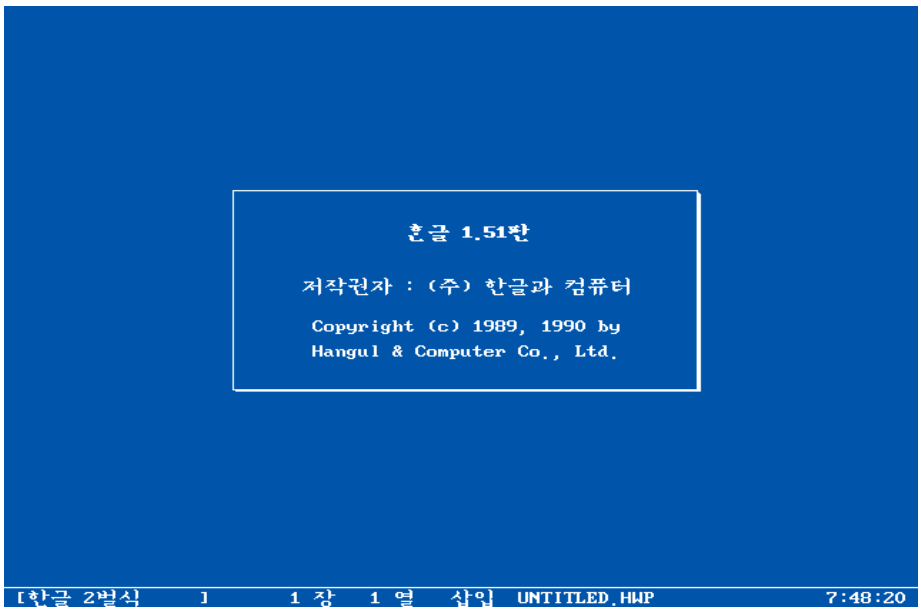
○ 1989년 한/글 1.0, 1.1, 1.2, 1.3

- 베타 버전인 0.9판을 1989년 3월 출시
- 최초의 상용버전인 1.0판이 1989년 4월 24일 출시
- 한/글 1.0 판의 기능
  - 위지위그 편집 기능
  - 8가지 한글/로마자 비트맵 글꼴 지원
  - 180 DPI 도트 매트릭스 프린터 지원
  - 특수 문자 입력 및 선 그리기 기능 등
- 한/글 1.2판에서 추가된 기능
  - 문서 압축 저장 기능
  - ASCII 텍스트 디지털 컴포넌트 저장 기능 강화
  - 한/영 전환키가 shift+space 키로 변경
  - 선 그리기로 만든 표 보호 기능 등

○ 1990년 한/글1.5

- 1990년 12월 20일에 아래 그림과 같은 1.51판이 도트판과 레이저판의 두 종류로 나뉘어 출시
- 한/글은 옛한글, 외국어를 포함해 완성형 코드로 표현 불가능한 1172자의 한글을 조합형을 이용하여 모두 표현함
- 한글 1.5에 추가된 기능
  - 도트판에서도 그림 인쇄 가능
  - 로마자 가변폭 글꼴 추가
  - 한글 필기체 추가

- 자동 쪽 번호 매김(페이징), 구역(컬럼 블록, 단축키 F4) 설정 등




<그림 3-54> 한/글 1.51 판

○ 1992년 한/글 2.0

- 1992년 7월 27일 한/글 2.0 전문용이, 9월 21일에 일반용이 출시됨
- 워드 프로세서를 이용한 전자 출판의 가능성을 염두에 두고 다음과 같은 기능들이 추가 됨
  - 1포인트부터 127포인트까지의 글꼴 가변 크기 조정 기능이 추가됨
  - 자간, 장평, 가변 조정 기능 및 문단 모양 편집을 지원함
  - 글꼴과 문단 모양을 저장하여 사용하는 스타일 지정/적용 기능
  - 한/글 2.0 전문용에서 비트맵 글꼴과 윤곽선 글꼴을 모두 지원
  - 전문용에서 추가 확장팩을 통해 2수준 확장한자(4888자+1만여 자) 지원
  - 수식 편집 기능
  - 표를 작성하고 편집하는 기능, 다단 편집 기능, 글상자, 그림 등의 틀 자유 배치 기능
  - 한글 맞춤법 검사 기능
  - 하이퍼텍스트 방식의 도움말 개선
  - 메모리 관리 방식의 변화로 최대 편집 쪽수의 증가
  - 마우스 지원

○ 1993년 한/글 2.1

- 1993년 9월 한/글 2.1 버전이 출시됨
- 전문가용, 일반용, 수검용[10], 꼬마용 등으로 구분

- 한/글 2.1 판에서 추가된 기능
  - 80386 프로세서용 32비트 코드로 제작 (전문가용)
  - 사용자 인터페이스의 직관화
  - 글꼴을 그리는 알고리즘을 개선, 글꼴의 추가 및 삭제 관리 지원
  - 가상메모리를 사용, 편집 매수 제한이 100여 쪽에서 500쪽으로 늘어남 (전문가용)
  - 전문용에서 컬러 인쇄 및 1200 DPI 인화지 출력을 지원함으로써 워드 프로세서를 이용한 전자출판의 가능성을 제시
  - 출력에 필수적인 포스트스크립트(PS) 호환용 로마자 글꼴, 샘체, 안상수체, 휴먼고딕 등 휴먼컴퓨터 사 글꼴 기본 제공 (전문가용)

○ 1994년 한/글 2.5
- 1994년 SEK전시회에서 한/글 2.5가 출시됨
- 한/글 2.5 판에서 추가된 기능
  - 덧실행(add-on) 프로그램 지원
  - 신명시스템, 태시스템 등의 추가 글꼴 및 프라임 영한사전을 제공하는 확장팩 출시
  - 인쇄 속도 개선 및 모아 찍기, 나눠 찍기 등의 변형 인쇄 기능 지원
  - 글꼴 미리보기 기능 지원
  - 수식 및 확장 한자 윤곽선 글꼴 지원
  - 불법 복제 방지용 락을 제거
  - 신명조를 기본글꼴로 사용함

○ 1995년 한/글 3.0 발표
- 한/글 3.0판 윈도용이 1995년 3월 출시됨
- 윈도용 한/글의 첫 출시는 윈도 95 출시와 함께 그 해 애플리케이션 시장에서 가장 큰 이슈가 되었음
- 윈도용 한/글은 지금까지 등장했던 다른 윈도용 한글 워드프로세서와는 달리, 운영체제에 의존하지 않는 독립 인터페이스, 독립 입출력 방식을 채택
- 또한 완성형 한글 코드만을 지원하는 윈도 3.1의 입력 체계를 사용하지 않고, 도스용 한/글의 입력 체계, 즉 조합형 한글 코드를 변형한 한컴 2바이트 코드를 사용으로써, 11,172자 현대 한글은 물론 옛한글까지 표현함
- 도스용 한/글의 마지막 버전인 한/글 3.0이 1995년 6월 출시됨
- 한/글 3.0 도스용에서 추가된 기능
  - 윈도용 한/글 3.0에서 저장한 디지털 컴포넌트 열기 지원(단, 글맵시 등은 불러오지 못함)

- 인터페이스 개선 및 도구 상자 추가
- 한/영 자동 변환 기능 및 빠른 교정 기능
- 문서 내 하이퍼 링크(하이퍼텍스트) 기능
- 덧실행 기능이 더 강화되어 PC통신 에뮬레이터, 공학용 계산기, CD 플레이어 등을 한/글 내부에서 실행할 수 있도록 함
- 수채화라는 그래픽 편집/드로잉 프로그램을 도스용 한/그림이라는 이름으로 번들로 제공
- 타자 연습 프로그램 제공

○ 한/글 프로 96

- 1996년 7월에는 다양한 애플리케이션와의 통합패키지로 선보인 한/글 프로 96와 한/글 오피스 96이 출시
- 한/글 프로 96은 기본적으로 사용자 인터페이스를 비롯해 인터넷문서 편집, 서모음집인 문서마당 등 다양한 기능이 업그레이드된 워드프로세서인 한/글 96을 기반으로, 주소록, 타자연습, 폼프로세서 틀마름이, 팩스 프로그램, 메일 프로그램 등 다양한 유틸리티를 추가한 종합 패키지로 변환됨
- 대중적으로 어느 정도 성공을 거둔 한/글 프로 96과는 달리 기업체 판매를 목적으로 출시된 한/글 오피스 96은 포함된 로터스 1-2-3 5.0과 프리랜스 그래픽스 프로그램의 기능 미비로 인해, MS 오피스에 비해 별다른 반향을 얻지 못함

○ 한/글 97

- 1996년 11월에는 한/글 96을 기반으로 중국어, 일본어 인터페이스 및 다국어 입력기를 채용한 한/글 국제판과 일본 시장을 목표로 한 한/글 일본판이 출시됨
- 1997년에는 한/글 97을 탑재한 한컴홈 97과 한컴 오피스 97이 출시됨

○ 한/글 8.15

- 1998년 8월 출시됨
- 한/글 97은 '실행 취소' 기능이 매우 취약했지만 역대 Windows용 한/글 시리즈 중 가장 완성도가 높았으며, 가장 높은 보급률을 보여준 제품임
- 매킨토시용으로도 개발되어 맥사용자의 대부분을 차지하는 전자 출판 분야에 많이 보급됨

○ 한/글 워디안

- 2000년 출시됨

- 소스 코드를 새로 작성함으로써 이전 버전과는 다른 모습을 보여줌
- 한/글 워디안에 추가된 기능
  - 256단계의 다단계 되돌리기/반복(실행 취소) 기능을 지원
  - 변형 다단 편집
  - 표 서식 기능 강화
  - 엣센스 영한사전과 표준 국어 대사전을 삽입하는 등 커다란 개선 사항
  - 워디안에서 작성한 문서를 이전 버전에서 읽을 수 없는 문제
  - 이전 버전의 문서를 워디안에서 본래대로 읽을 수 없는 문제
  - 프로그램의 안정성 문제가 지적

○ 한/글 2002

- 2001년에 출시됨
- 한/글 워디안을 기반으로 이전 버전 디지털 컴포넌트와의 호환성 문제 및 안정성 문제를 해결한 제품
- 이후 프로그램의 안정성을 향상시킨 한/글 2002 SE로 무상 업그레이드

○ 한/글 2004

- 2004년 출시됨
- XML 문서 원과 공개 키 기반 암호화, 스킨, 작업창 기능이 추가된 제품임
- 한컴오피스 2004는 국산 스프레드시트인 넥셀과 자체 개발한 프레젠테이션 프로그램인 한컴 슬라이드 2004를 포함함으로써 토종 오피스 스위트로의 제 모습을 갖추게 됨

○ 한/글 2005

- 2005년 출시
- 협동 작업을 위한 버전 관리 기능, 메모 기능의 추가
- 한/글 2005에 추가된 기능
  - XML로 작성된 한/글 서식과 ODBC의 연동
  - 표의 가로 세로 대칭 뒤집기
  - 아랍어 등의 기능이 추가됨

○ 한/글 2006

- 애플컴퓨터코리아의 기술 지원으로 Mac OS X용이 2005년 출시됨
- 기존의 한/글이 가진 기능을 Mac OS X 환경에서 구현함
- 한/글 2006에 추가된 기능
  - 한/글 제품과 마이크로소프트 워드, 훈민정음, 이치타로(一太郎) 문서 등을 불

러올 수 있음
- 트루타입과 한/글 전용 글꼴을 지원

○ 한/글 2007
• 2007년 출시됨
• 개체 투명도 적용 등 고급 기술을 적용하여 더 화려한 문서 제작이 가능하지만, 그만큼 프로그램이 무거워지고 고 사양을 요구함
• 한/글 2007에 추가된 기능
- 선과 도형에 앤티에일리어스가 적용
- 반투명한 선을 구현
- 마이크로소프트 오피스와의 호환성이 향상
- 워터마크, 암호화 기능, 세로쓰기, 공인 인증서 암호화 기능, 개체 연결선, 개체 그림자 삽입 등 보안성이 향상
- 추가로 ValuePack를 설치하면 다양한 기능과 더불어 PDF 문서를 쉽게 생성할 수 있는 무료 Haansoft PDF 문서 변환 드라이버를 설치할 수 있어 편리한 문서 변환 기능을 제공

○ 한/글 2008
• 한/글 2007을 기반으로 리눅스 사용자들을 위해 만들어 짐

○ 한/글 2010
• 2010년 3월 3일 출시
• 한/글 2010에 추가된 기능
- 한/글 2007의 차기작으로 리본 메뉴 인터페이스를 도입
- ODF문서의 일기/저장 기능
- MetaBlog API를 한글자체에서 적용하여 한글로 작성한 문서를 블로그로 업로드 가능
- 개인정보 유출 방지를 위한 기능
- 차트에다가 복잡한 선을 그리는 이전 버전 하위호환 기능이 삭제

❒ 한/글 2.1부터 한/글97까지는 커널 기반에서 커다란 변화가 없어서 디지털 컴포넌트의 호환성을 유지할 수 있었지만 2000년에 출시된 한/글 워디안은 모든 소스코드를 새로 작성함으로써 이전 버전과는 다른 모습을 보여줄 뿐만 아니라, 워디안에서 작성한 문서를 이전 버전에서 읽을 수 없는 문제와 이전 버전의 문서를 워디안에서 제대로 읽을 수 없는 버전간 포맷 호환성 문제가 발생

## 다. 마이크로소프트의 엑셀 및 오피스 애플리케이션 변화 조사

### (1) 국내 스프레드시트 애플리케이션 관련 히스토리 조사

❒ 1982년 멀티플랜이라는 스프레드시트 프로그램을 출시했던 더그 클런더(Doug Klunder)의 말에 따른 '1-2-3로 할 수 있는 것은 모두 할 수 있고 더 잘 할 수 있도록' 하기 위해서 엑셀이라고 명명된 새로운 스프레드시트 프로그램의 개발

❒ 그러나 초기의 엑셀 프로그램은 비지칼크(Visicalc)를 제치고 스프레드시트 분야의 1인자였던 로터스 1-2-3에 밀려 고전

❒ 1987년 11월 최초의 윈도우용 버전인 엑셀 2.0은 윈도우 기반으로 전환하지 못한 로터스의 1-2-3를 제치고 스프레드시트 분야의 최강자가 되었으며, 1993년부터 엑셀은 비주얼 베이직에 기반해 엑셀에서의 작업 자동화와 워크시트에서의 사용자 지정 함수 사용을 가능하게 한 VBA(Visual Basic for Application) 언어를 포함하여 제공

❒ 1990년에 마이크로소프트 오피스 3.0을 윈도우용으로 처음 출시한 후 워드6.0, 엑셀5.0, 파워포인트4.0을 포함한 마이크로오피스 4.0을 1994년도에 출시하였으며 이후로는 오피스4.3(v6.0), 오피스95(v7.0), 오피스97(v8.0), 오피스2000(v9.0), 오피스XP(v10.0), 오피스2007(v12.0), 오피스2010(v14.0), 오피스2013(v15.0)을 지속적으로 발표

❒ 이처럼 다양한 버전 중에서 'MS 오피스97(v8.0)'은 이전 버전과 엄청나게 많은 변화와 향상이 있었던 버전으로 가장 오랜기간 사용했던 제품이며 거의 모든 스프레드시트 계통의 문서를 장악. 또 2007년에 출시된 'MS 오피스 2007(v12.0)'은 이전 오피스와는 완전히 새로운 인터페이스인 리본 UI를 사용하였고 Open Office XML File Format을 지원

### (2) 엑셀 및 오피스 프로그램의 버전별 주요 기능 및 특징

○ 엑셀 1.0

- 1985년 매킨토시용으로 엑셀 출시
- 차트 기능이 포함됨 스프레드시트 프로그램

○ 엑셀 2.0

- 1987년 11월 출시
- 1988년 로터스사의 1-2-3의 판매 부진으로 엑셀이 판매량이 역전함

○ 마이크로소프트 오피스 3.0

- 1990년에 Windows용으로 처음 출시된 마이크로소프트 오피스임

○ 마이크로소프트 오피스 4.0

- 1994년에 출시되었으며 워드 6.0, 엑셀 5.0, 파워포인트 4.0을 포함함
- 워드는 전 버전이 2.0임에도 불구하고 맥용 오피스와 버전 번호를 같이 하기 위하여 2.0 다음 버전을 6.0으로 맞춤
- 1993년부터 엑셀은 비주얼 베이직에 기반해 엑셀에서의 작업 자동화와 워크시트에서의 사용자 지정 함수(UDF, User defined function) 사용을 가능하게 한 VBA(Visual Basic for Application) 언어를 포함

○ 마이크로소프트 오피스 4.3 (버전 6.0)

- 마지막 16비트 지원 오피스
- 이 버전은 Windows 3.1x과 Windows NT 3.1 그리고 Windows NT 3.5를 마지막으로 지원하는 버전.
- Windows NT 3.51은 오피스 97까지 지원

○ 마이크로소프트 오피스 95 (버전 7.0)

- 버전 7.0, 5.0과 6.0은 워드,엑셀, 파워포인트의 버전을 맞추려고 넘어감
- Windows 95와 호환되기 위하여 모두 32비트로 구성됨
- 16bit암호화를 지원함
- 오피스 95 스탠다드와 오피스 95 프로페셔널의 두가지 에디션임
- 스탠다드 버전은 워드 7.0, 엑셀 7.0, 파워포인트 7.0, 스케줄+ 7.0을 포함
- 프로페셔널 에디션은 스탠다드 에디션에 액세스 7.0을 추가한 버전임

○ 마이크로소프트 오피스 97 (버전 8.0)

- 엄청나게 많은 변화와 향상이 있었던 버전임
- VBA가 공식 매크로언어로 채택됨, 기존 Excel Basic 삭제됨
- VBA를 이용하여 개발자를 만들고, UserForms을 사용하고, DB기능 등 아주 새로운 인터페이스를 제공함
- 40bit암호화를 지원함
- 거의 모든 스프레드시트 계통의 문서를 장악함

- 일본과 한국을 대상으로 오피스 98(버전 8.5)도 출시됨

○ 마이크로소프트 오피스 2000 (버전 9.0)
- 매크로 바이러스를 방지하기 위한 디지털 서명 등 새로운 보안 기능을 추가함
- 오피스 2000은 Windows 95를 지원하는 마지막 버전임
- 제품 인증을 포함하지 않는 버전임

○ 마이크로소프트 오피스 XP (버전 10.0)
- 2003년 출시됨
- 오피스 2002라고도 불림
- 안전 모드라는 기능을 추가했는데. 만약에 레지스트리 수정 등의 이유로 한 프로그램의 로딩이 실패할 시 안전 모드가 문제를 감지하고 수정이 가능함
- 제품 인증이라는 기능도 추가됨
- 오피스 XP는 Windows 98, Windows 98 SE, Windows Me, Windows NT 4.0을 지원하는 마지막 버전임
- Windows 비스타에서 호환된다고 보고된(정식으로 호환되지는 않음) 첫 번째 버전임
- 이스터 에그(부활절 달걀)(애플리케이션의 숨겨진 기능)이 없어짐

○ 마이크로소프트 오피스 2003 (버전 11.0)
- 마이크로소프트 인포패스와 마이크로소프트 원노트 프로그램이 추가됨
- Windows XP 스타일의 아이콘과 디자인을 사용한 첫 버전으로, 아웃룩 2003은 정크 메일 필터를 강화됨
- 오피스 2003은 Windows 2000을 지원하는 마지막 버전임

○ 마이크로소프트 오피스 2007 (버전 12.0)
- 2007년에 출시됨
- 완전히 새로운 인터페이스인 리본 UI를 사용함
- 오피스 2003은 Windows XP 서비스 팩 2~3, Windows 서버 2003 서비스팩 1 이상, 또는 Windows 비스타에서 사용 가능함
- 오피스 2007 서비스팩 2부터 오피스 2007은 리버오피스/오픈오피스에서 쓰이는 ODF 포맷을 지원함
- Open XML 채택. 16384열, 1백만행 지원, 128비트 암호화 지원함
- xlsx를 OfficeExcel2007의 기본 확장자임
- VBA매크로 코드를 저장할 수 없음(저장하기 위해서는 xlsm으로 저장해야

함)

○ 마이크로소프트 오피스 2010 (버전 14.0)
- 2010년 9월 출시됨
- 버전 13.0 출시되지 않음
- 64bit코드 지원함

○ 마이크로소프트 오피스 2013 (버전 15.0)
- 2012년 7월 출시됨
- Windows XP와 Windows 비스타는 지원하지 않음
- MS클라우드 서비스(SKYDRIVE)와 연동, 계정 통합됨
- 스마트폰이나 태블릿 PC에서 입력을 하기 때문에 터치나 펜으로 입력 가능함
- 방사형태의 다양한 편집 기능을 제공함
- MS-Word에서 HWP(97이하 버전), PDF 문서를 읽기/수정 가능함

## 라. 디지털 컴포넌트 실행을 위한 운영 체제 변화 조사

### (1) PC 플랫폼에 적용된 운영제체 조사

❒ 기능적 요소를 포함한 디지털 컴포넌트 중 공공기관에서 가장 많이 사용되는 한/글과 마이크로소프트사의 엑셀에 대한 조사를 진행하기 위해 한/글과 엑셀 컴포넌트를 실행하는 플랫폼 운영체제 (O/S)에 대한 버전 히스토리를 우선 조사하였고 아래 <표 3-23>과 같음

○ 1980년대 이후 PC 플랫폼의 최강자로 군림했던 마이크로소프트사의 DOS 및 윈도우 운영체제

○ 1981년 최초의 IBM PC에 탑재된 PC-DOS 첫 버전을 출시하고 1982년 MS-DOS라는 이름으로 IBM PC 호환기종의 운영 체제를 발표

○ 1983년에 발표된 2.0 버전에서는 서브디렉토리, 핸들 기반 디지털 컴포넌트 운용, 명령어 입출력 리디렉션, 파이프 등을 지원하고 하드디스크와 308KB 플로피 디스크 드라이브를 지원하였으며 경로명 구분 문자를 기존의 슬래시'/' 대신 대부분의 운영체제에서 사용하는 역슬래시'\'로 변경

○ 1983년에는 영어 이외의 언어와 날짜 형식을 지원함으로써 1985년 2.25버전을 통해 한글, 일본어, 한자 지원을 확대

○ 1984년 출시된 3.0버전에서는 PC AT를 지원하며 1.2MB 플로피 디스크 및 32메가바이트의 하드 디스크 파티션을 지원하였고 3.1에서는 마이크로소프트 네트워킹 3.2에서는 3.5인치 720KB 플로피 디스크 드라이브를 지원하였으며 3.3에

서는 여러 개의 논리 드라이브를 지원

- 1988년 4.0에서는 도스 셸을 제공하고 199년도에 출시된 5.0에서는 메모리 관리, 전체 화면 편집기, QBasic 프로그래밍 언어, 온라인 도움말, 작업 전환 기능이 추가된 도스 셸 등을 제공
- 1990년에 처음으로 출시된 윈도우즈3.0은 많은 기능 오류로 인해 사용자로부터 외면 받았으나, 1992년에 윈도우즈3.1을 발표한 후 1994년에 발표한 DOS 6.22를 마지막으로 더 이상 단일 제품을 출시하지 않음으로 사용자는 윈도우 운영체제로 이전
- 이후 1995년에 윈도 95를 발표하면서 DOS의 기능은 윈도우에 내장되었는데, MOSOS.SYS의 기능이 IO.SYS에 통합되었으며 LBA(논리 블록 어드레싱) 지원 DOSLFN 등 별도의 드라이버를 이용하면 긴 디지털 컴포넌트 이름 사용이 가능

### (2) MS 운영 체제의 버전별 적용 시기 조사

❒ 1970년대에 PC(Personal Computer) 시장을 선도하던 Apple에 대항하여 마이크로소프트사의 MS-DOS가 IBM PC 계열의 주 운영체제(O/S)로 성장할 수 있었던 이유는 개방 정책 때문 이었는데, 1995년 윈도우 계열로 운영체제가 바뀌면서 마이크로소프트도 운영체제에 대한 폐쇄 정책으로 방향을 선회하여, 운영체제에 대한 정보 공개가 급속히 줄어들면서 이후 2000년 후반까지 윈도우에 종속된 프로그램 개발이 보편화됨

<표 3-24> PC 플랫폼 운영체제 변화

| 년도 | 영문 MS-DOS / MS-Windows | 한글 MS-DOS / MS-Windows |
|---|---|---|
| 1981.08 | 영문 MS-DOS 1.0 | |
| 1982.08 | 영문 MS-DOS 1.25 | |
| 1983.03 | 영문 MS-DOS 2.0 | |
| 1984 | 영문 MS-DOS 3.0/3.1 | |
| 1985 | | 한글 MS-DOS 2.11 |
| 1985.11 | 영문 윈도우즈 1.0 | |
| 1986.04 | 영문 MS-DOS 3.2 | |
| 1987.04 | 영문 MS-DOS 3.3, 영문 윈도우즈 2.0 | |
| 1988.07 | 영문 MS-DOS 4.0 | |
| 1988.11 | 영문 MS-DOS 4.01 | |
| 1990.05 | 영문 윈도우즈 3.0 | 한글 MS-DOS 4.0, 한글 윈도우즈 |

| | | |
|---|---|---|
| | | 3.0 |
| 1990.07 | | 한글 윈도우즈 2.1 |
| 1990.10 | | 한글 윈도우즈 3.0 |
| 1991.06 | 영문 MS-DOS 5.0 | |
| 1991.07 | | 한글 MS-DOS 5.0 |
| 1992.04 | 영문 윈도우즈 3.1 | |
| 1992.10 | 영문 윈도우즈 3.1 for Workgroup | |
| 1993.05 | | 한글 윈도우즈 3.1 |
| 1993.08 | 영문 MS-DOS 6.0, 윈도우즈 NT 3.1 | 한글 MS-DOS 6.0 |
| 1993.11 | 영문 윈도우즈 3.11 for Workgroup | |
| 1994.02 | 영문 MS-DOS 6.21 | |
| 1994.03 | 영문 MS-DOS 6.21 | |
| 1994.04 | 영문 MS-DOS 6.22 | |
| 1994.09 | 영문 윈도우즈 NT 3.5 | 한글 MS-DOS 6.2 |
| 1995.06 | 영문 윈도우즈 NT 3.51 | 한글 윈도우즈 NT 3.5, MS-DOS 7.0 |
| 1995.08 | 영문 윈도우즈 95 | |
| 1995.11 | | 한글 윈도우즈 95 |
| 1996.05 | | 한글 윈도우즈 NT 3.51 |
| 1996.08 | 영문 윈도우즈 NT 4.0 | |
| 1996.10 | | 한글 윈도우즈 NT 4.0, MS-DOS 7.1 |
| 1998.06 | 영문 윈도우즈 98 | |
| 1998.08 | | 한글 윈도우즈 98 |
| 1999.05 | 영문 윈도우즈 98SE | |
| 1999.07 | | 한글 윈도우즈 98SE |
| 2000.02 | 영문 윈도우즈 2000 | MS-DOS 8.0 |
| 2000.03 | | 한글 윈도우즈 2000 |
| 2000.07 | 영문 윈도우즈 ME | |
| 2001.10 | 윈도우즈 XP | 한글 윈도우즈 XP |
| 2003.04 | 윈도 서버 2003(윈도 .net) | 윈도 서버 2003(윈도 .net) |
| 2007.01 | 윈도우 비스타, 코드네임 롱혼 | 윈도우 비스타 |
| 2008.02 | 윈도우 서버 2008 | 윈도우 서버 2008 |

| | | |
|---|---|---|
| 2009.10 | 윈도우7(코드네임 블랙콤) | 윈도우 7 |
| 2012.10 | 윈도우 8 | 윈도우 8 |

❒ 마이크로소프트의 운영체제 독과점에 대항하여 여러 가지 운영 체제 프로그램이 발표되었지만, 애플리케이션 및 데이터 디지털 컴포넌트의 호환성 문제에 부딪치면서 다수 사용자의 사랑을 받지는 못함

❒ 여기서 주목할 점은 국내 공공기관의 업무 시스템 구축이 본격적으로 시작된 시점이 90년대 후반이기 때문에 대부분의 공공기관에 도입된 PC에 설치된 운영 체제는 윈도우95부터 이후 마이크로소프트사의 윈도우 계열로 유지됨

### (3) MS 운영 체제의 버전별 주요 기능 및 특징

❒ 1981년에 출시된 MS-DOS 1.0부터 영문 2000년에 출시된 MS-DOS 8.0 까지 버전별 특징 및 주요 기능에 대한 지원 사항 분석

○ MS-DOS 1.0
- 1981년에 출시됨
- 최초의 IBM PC에 탑재된 첫 버전임(PC-DOS)

○ MS-DOS 1.25
- 1982년 출시됨
- MS-DOS라는 이름으로 출시된 IBM PC 호환기종용 첫 버전임

○ MS-DOS 2.0
- 1983년 출시됨
- 서브디렉토리, 핸들 기반 디지털 컴포넌트 운용, 명령어 입출력 리디렉션, 파이프 등을 지원
- 마이크로소프트는 경로명 구분 문자를 대부분의 도스나 CP/M 프로그램에서 명령 스위치로 사용하고 있는 슬래시(/)대신 역슬래시(\)로 사용하기로 함
- 하드 디스크와 360KB 플로피 디스크 드라이브 지원함
- 영어 이외의 언어와 날짜 형식 지원함
- MS-DOS 2.25 1985년 출시
- 한글과 일본어 한자 지원 확대됨
- 동아시아 국가에서만 출시함

○ MS-DOS 3.0
- 1984년 출시됨
- PC AT를 지원함
- 1.2MB 플로피 디스크 및 최대 32 메가바이트의 하드 디스크 파티션 지원함 (한 개의 기본 파티션과 한 개의 논리 드라이브 지원)
- 마이크로소프트 네트워킹 지원함
- 3.5 인치 720KB 플로피 디스크 드라이브 지원함
- 여러 개의 논리 드라이브 지원함

○ MS-DOS 4.0
- 1988년 출시됨
- 도스 셸 제공함

○ MS-DOS 5.0
- 1991년 출시됨
- 메모리 관리, 전체 화면 편집기, QBasic 프로그래밍 언어, 온라인 도움말, 작업 전환 기능이 추가된 도스 셸 등 제공함

○ MS-DOS 6.00
- 1993년 출시됨
- 더블스페이스, 영문 사용자 메시지 지원(한글판), 조합형 한글 코드 페이지(1361) 지원함(한글판)
- 스택 일렉트로닉스와의 소송으로 인해 더블스페이스 기능이 제외됨(v6.21)

○ MS-DOS 6.22
- 1994년 출시됨
- 단일 제품으로는 마지막 버전임
- 더블스페이스 대신 동일한 기능의 드라이브스페이스 제공함

○ MS-DOS 7.0
- 1995년 출시됨
- 윈도 95에 내장됨
- MSDOS.SYS의 기능이 IO.SYS에 통합되어, MSDOS.SYS에는 참고용 텍스트만이 들어있음
- LBA(논리 블록 어드레싱) 지원함
- DOSLFN 등의 별도 드라이버를 이용하면 긴 디지털 컴포넌트 이름을 사용 가능함

- 윈도 95 OSR2와 그 이후 버전 및 윈도 98,윈도 98 SE에 내장됨
- FAT32 디지털 컴포넌트 시스템 지원함

○ 영문 MS-DOS 8.0
- 2000년 출시됨
- 윈도 Me에 내장됨
- 윈도 XP, 윈도 비스타, 윈도 7에서 플로피 디스크를 포맷할 때 'MS-DOS 시동 디스크 만들기'를 선택하면 이 버전의 최소한의 디지털 컴포넌트를 포함한 MS-DOS가 디스크에 설치함
- 이 버전의 MS-DOS는 7.1버전에 비해 향상된 점도 있으나 몇 가지 중요한 제한점을 지니고 있음

○ 윈도우즈 3.1
- 1992년 4월 6일 첫 출시. 개발 코드네임은 Janus
- 윈도우즈는 OS가 아니라 도스 프로그램 이였지만, 도스에 쓰기 편하고 쉬운 GUI를 씌워주고, 자체적인 API를 통해 윈도우 전용 프로그램 구현 가능
- 전작인 윈도우 3.0에서 본격적인 가상 메모리의 사용으로 멀티태스킹 능력이 대폭 강화
- 그래픽 카드의 성능 향상으로 인해 좀더 미려하고 화려한 인터페이스를 제공
- 각종 사운드라든지 멀티미디어 팩을 '멀티미디어 익스텐션' 이라는 이름으로 따로 제공

○ 윈도우즈 95
- 1995년 8월 24일 첫 출시. 국내는 1995년 11월 28일 출시.코드명 "시카고(Chicago)“
- 최소화/최대화/닫기 버튼, 시작 메뉴, 작업 표시줄, 바탕 화면이라는 윈도우 UI의 기본을 정립시킨 버전.
- DOS와의 호환성을 가정용 OS에서 포기하기는 싫었으므로 여전히 DOS기반
- 32비트 처리가 가능하고, 거기다 기존 16비트 커널까지 둘 다 갖춘데다 32비트 한정이긴 하지만 선점형 멀티태스킹을 달고 나와서 3.1보다 높은 안정성
- 16비트 호환성에 문제 발생.
- 플러그 앤 플레이가 가능해졌고, 램을 512MB까지 지원
- 시작 메뉴, 작업 표시줄, 바탕 화면, 프로그램 메뉴 등, 8까지 내려오고 있는 윈도우 UI(User Interface)의 기본도 윈도우 95로부터 시작.

○ 윈도우즈 98

- 1998년 6월 25일에 발매되었고, 개발코드명 "멤피스(Memphis)“
- Windows 95 후기버전에서야 추가된 USB 지원을 본격적으로 넣었으며 FAT32 디지털 컴포넌트시스템을 좀더 제대로 지원
- USB 및 IEEE 1394의 지원 강화 및 각종 드라이버의 지원이 확대
- 인터넷 익스플로러 5 탑재 , 아웃룩 익스프레스 5, 넷미팅 3를 탑재함으로써 인터넷 지원이 개선
- Windows Media Player 6.1 및 DirectX 6.1 탑재로 멀티미디어 지원 향상
- DOS에서 돌아가는 마지막 윈도우즈

○ 윈도우즈 ME
- 개발코드명 "밀레니엄(Millennium)" 2000년 9월 14일 출시.
- Microsoft Windows 역사상 가장 큰 혹평을 받은 제품.
- 메모리 누수 문제 심각
- UPnP 지원, USB 메모리 기본지원, 시스템 복원 기능
- MS-DOS 단종후 첫 윈도우즈

○ 윈도우즈 XP
- 2001년 10월 25일 출시. 개발 코드네임은 "휘슬러(Whistler)“
- MS는 90년대 후반 9x 커널에 한계점을 느끼고, 이때까지 가정용의 9x와 업무용의 NT로 나뉜 윈도우 계열을 NT로 통합
- x64를 처음으로 지원한 윈도우
- MS는 DOS/9x 커널을 단념하고 윈도우 XP를 발매하여 가정용 OS에도 NT 커널을 도입하고 안정성을 추구
- 32비트에 최적화 되어있으며, 싱글코어의 리소스를 최대한 활용할 수 있는 방향으로 설계
- 윈도우즈 XP SP3의 연장지원은 오는 2014년 4월 8일에 종료.

○ 윈도우즈 VISTA
- 2007년 1월 30일 첫 출시. 개발 코드명 "롱혼(Longhorn)“
- 인터페이스도 완전히 새로 디자인되었고, UI에 3D그래픽을 사용하여 기존의 윈도우와는 이질감이 존재
- 윈도우 시리즈 중 최초로 위젯을 내장
- 슈퍼펫치, 레디부스트[89], 터보 메모리[90] 등 여러 신기술이 적용

89) Microsoft Windows에서 USB 메모리 및 플래시 메모리 카드에 있는 저장공간을 사용하여 시스템의 속도를 향상시킬 수 있는 기능. 윈도우 비스타 이상에서 기본적으로 지원하고 있으며, 특정 프로그램을 이용하면 XP에서도 사용 가능하다.

90) Robson Module 이라고 불리기도 한다. 인텔에서 만든 플래시 메모리 규격이다. 주로 mini PCI-E 인터페이스로 출시되어있다.

- 기본적으로 주요 폴더를 색인화(Indexing)하며, 이에 따라 검색 시간이 획기적 향상
- x64를 공식 지원
- 윈도우 OS 시리즈 중 가장 안정적이고 보안성이 높게 설계

○ 윈도우즈 7

- 2009년 10월 22일 정식 출시. 개발 초기의 코드명은 "윈도 블랙콤(Windows Blackcomb)"이었으나, 2006년 1월 "비엔나(Vienna)"로 변경
- WDDM(Windows Display Driver Model)이 1.0에서 1.1로 업그래이드 되었고 이로 인해 DX9이하 세대의 VGA는 구조상 WDDM 1.1이 지원 불가능
- 2009년 4월 24일 윈도우 7 프로페셔널(비즈니스), 얼티밋 버전 한정으로 윈도우 XP 가상모드를 지원
- 가상 하드 드라이브(Virtual Hard Drive)인 VHD디지털 컴포넌트안에 윈도를 설치하고, 윈도가 설치된 VHD디지털 컴포넌트가 직접 부팅 가능
- SSD를 지원하고, 읽고 쓰는 방식 자체를 플래시 메모리에 최적화

○ 윈도우즈 8

- 2012년 10월 26일 발매한 OS.
- 윈도우는 항상 x86, x86-64 위에서만 돌아갔는데, 윈도우 8부터는 ARM을 정식으로 지원
- 윈도우폰 7에서 선보인 바 있는 Windows Style UI (舊 메트로 UI, 이하 혼용) 지원
- Windows Vista부터 기본으로 들어갔던 Windows Media Center와 DVD 소프트웨어 디코더 제외됨.
- 윈도우의 디지털 컴포넌트용량이 비스타나 7보다 가벼워졌고 오히려 구형 컴퓨터에서도 잘 돌아갈 만큼 사양이 낮아짐.
- 윈도95부터 17년간 이어져 내려온 '시작'버튼이 사라짐.(8.1 버전에서 부활)

## 마. 포맷 변화 분석 결과

### (1) 운영 체제와 실행 가능한 애플리케이션 관계 분석

❒ 아래 표와 같이 에뮬레이션 대상 포맷을 선정하기 위해 윈도우 운영체제에서 사용되었던 한글 애플리케이션 및 마이크로소프트 오피스 버전을 조사

○ 한/글 애플리케이션과 MS 오피스 프로그램이 어떠한 윈도우 운영체제에서 실행되었는지를 조사한 결과 아래 표와 같이 하나의 버전이 여러 버전에 걸쳐 사

용되었음을 확인할 수 있었음

○ 한/글 애플리케이션 버전에서 가장 큰 특징을 갖는 한/글 97 버전은 윈도우 98부터 윈도우7까지 장기간 사용됨을 확인할 수 있음.

○ MS 오피스 프로그램에 대하여 조사한 결과 운영 체제인 윈도우 버전에 맞추어 업그레이드된 신규 버전을 출시한 후 이전 버전에 대한 지원을 중단하는 정책으로 진행하였음을 확인할 수 있음

<표 3-25> 한글 애플리케이션 및 MS 오피스 사용 이력

| 연도 | 운영체제 | 한글 애플리케이션 | MS 오피스 |
|---|---|---|---|
| 1990.05 | 한글 윈도우즈 3.0 | 한글 2000<br>삼보 보석글<br>금성 하나워드<br>한/글 1.5<br>삼성 훈민정음 | MS Office 3.0 |
| 1993.05 | 한글 윈도우즈 3.1 | 한/글 2.0<br>하나 워드프로세서 | MS Office 3.0<br>MS Office 4.0 |
| 1995.11 | 한글 윈도우즈 95 | 한/글 2.5<br>한/글 3.0<br>한/글 96 | MS Office 95<br>MS Office 2000 |
| 1998.08 | **한글 윈도우즈 98** | 한/글 97<br>한/글 815 특별판 | MS Office 97<br>MS Office 2000<br>MS Office XP |
| 2000.03 | 한글 윈도우즈 2000 | 한/글 97<br>한/글 워디안 | MS Office 2000<br>MS Office XP<br>MS Office 2003 |
| 2001.10 | 한글 윈도우즈 XP | 한/글 97<br>한/글 2002~2007 | MS Office XP<br>MS Office 2003 |
| 2007.01 | 윈도우 비스타 | 한/글 97<br>한/글 2002~2007 | MS Office XP<br>MS Office 2003<br>MS Office 2007 |
| 2009.10 | **윈도우 7** | **한/글 97**<br>**한/글 2002~2007** | **MS Office 2003**<br>**MS Office 2007**<br>MS Office 2010<br>MS Office 2013 |
| 2012.10 | 윈도우 8 | 한/글 2010<br>한/글 2013 | MS Office 2007<br>MS Office 2010<br>MS Office 2013 |

### (2) 애플리케이션과 디지털 컴포넌트 포맷 버전 비교

❒ 한/글 애플리케이션 버전과 디지털 컴포넌트 포맷 버전 비교

○ 한/글 애플리케이션 버전에 대한 디지털 컴포넌트 포맷 버전을 비교 검토한 결과 한/글 1.0부터 한/글 2.5까지는 포맷에 대한 안정화 기간으로 애플리케이션 버전이 업그레이드 될 때마다 포맷 버전도 변화가 이루어짐

○ 한/글 3.0부터 정착된 HWP 3.0 Format은 한/글 97까지 애플리케이션 기능은 많이 향상되었지만 포맷은 3.0 버전에서 소규모 버전 업그레이드를 통해 안정화 됨

○ 한/글 8.15와 이를 보완한 한/글 워디안 버전에서는 새로운 포맷 버전인 HWP 5.0 Format을 선보였으나 이전의 HWP 3.0 Format과의 호환성 문제가 제기되면서 한/글 2002를 통해 이전의 3.0 Format과 5.0 Format을 동시에 사용할 수 있도록 제공

○ 이후의 애플리케이션 버전에서는 HWP 3.0 Format과 HWP 5.0 Format을 동시에 지원하지만 서로 호환성을 갖고 있지 않으므로 한/글 포맷에 대한 분석은 HWP 3.0과 5.0에 대하여 개별 수행할 필요가 있음

<표 3-26> 한/글 애플리케이션과 포맷 버전 비교

| 애플리케이션 버전 | 포맷 버전 | | |
|---|---|---|---|
| | 이전 계열 | 3.0 계열 | 5.0 계열 |
| 한/글 1.0 | HWP 1.0 Format | | |
| 한/글 1.5 | HWP 1.5 Format | | |
| 한/글 2.0 | HWP 2.0 Format | | |
| 한/글 2.1 | HWP 2.1 Format | | |
| 한/글 2.5 | HWP 2.5 Format | | |
| 한/글 3.0 | | HWP 3.0 Format | |
| 한/글 프로 96 | | HWP 3.0 Format | |
| 한/글 97 | | HWP 3.0 Format | |
| 한/글 8.15 | | △ | HWP 5.0 Format |
| 한/글 워디안 | | △ | HWP 5.0 Format |
| 한/글 2002 | | HWP 3.0 Format | HWP 5.0 Format |
| 한/글 2004 | | HWP 3.0 Format | HWP 5.0 Format |
| 한/글 2005 | | HWP 3.0 Format | HWP 5.0 Format |
| 한/글 2006 | | HWP 3.0 Format | HWP 5.0 Format |
| 한/글 2007 | | HWP 3.0 Format | HWP 5.0 Format |
| 한/글 2008 | | HWP 3.0 Format | HWP 5.0 Format |

| | | | |
|---|---|---|---|
| 한/글 2010 | | HWP 3.0 Format | HWP 5.0 Format |
| 한/글 2013 | | HWP 3.0 Format | HWP 5.0 Format |

❒ 엑셀 애플리케이션 버전과 디지털 컴포넌트 포맷 버전 비교

○ 엑셀 애플리케이션 버전에 대한 디지털 컴포넌트 포맷 버전을 비교 검토한 결과 엑셀 2.0부터 엑셀 95까지는 애플리케이션 버전 업그레이드에 맞춰 포맷 버전을 업그레이드함

○ 그러나, 엑셀 97부터 적용된 BIFF 8.0 포맷이 엑셀의 기본 포맷으로 적용되면서 향후 새로운 버전의 애플리케이션에서도 동일한 포맷 버전을 유지하다가 엑셀 2007부터는 기존의 97~2003(XLS) 포맷과 XLSX 포맷을 동시에 지원하는 형태로 변화함

○ 따라서, 엑셀 애플리케이션에 대한 분석은 97~2003(XLS)의 BIFF 8.x 포맷과 XLSX 포맷을 대상으로 함

<표 3-27> 엑셀 애플리케이션과 포맷 버전 비교

| 애플리케이션 버전 | 포맷 버전 | | |
|---|---|---|---|
| | 이전 계열 | 97-2003(XLS) 계열 | XLSX 계열 |
| 엑셀 2.0 | BIFF 2.0 | | |
| 엑셀 3.0 | BIFF 3.0 | | |
| 엑셀 4.0 | BIFF 4.0 | | |
| 엑셀 5.0 | BIFF 5.0 | | |
| 엑셀 95 | BIFF 7.0 | | |
| 엑셀 97 | | BIFF 8.0 | |
| 엑셀 2000 | | BIFF 8.x | |
| 엑셀 2002 | | BIFF 8.x | |
| 엑셀 2003 | | BIFF 8.x | |
| 엑셀 2007 | | BIFF 8.x | xlsx |
| 엑셀 2010 | | BIFF 8.x | xlsx |
| 엑셀 2013 | | BIFF 8.x | xlsx |

### (3) 에뮬레이션 대상 선정

❒ 윈도우 운영체제와 한/글 애플리케이션 및 MS 오피스 프로그램에 대한 이력 조

사를 통해 한/글 애플리케이션은 1998년 이전 버전인 한/글97과 이후 버전인 한/글 워디안의 포맷 호환성이 결여되었음을 확인할 수 있었고, MS 오피스 프로그램도 2007년 이전 버전인 오피스2003과 이후 버전인 오피스2007간 포맷의 변화가 있음을 확인

❒ 따라서 에뮬레이션 대상으로 한/글 애플리케이션에 대해서는 주요 버전인 한/글97과 한/글2007을 선정하였고, MS 오피스 프로그램에 대해서는 오피스2003과 오피스 2007을 주요 분석 대상으로 선정

❒ 또한 에뮬레이션 대상 운영체제는 한/글97, 한/글2007, 오피스2003 과 오피스2007을 모두 지원하는 윈도우7 운영체제를 에뮬레이션 대상 운영체제로 선정

### 3.3.2 문서보존포맷(PDF/A) 변환 시 소실되는 기능성 분석

#### 가. 기능성 분석 배경 및 필요성

❒ 3.3.1절에서 분석한 바와 같이 한/글과 파워포인트, 엑셀 애플리케이션은 업그레이드를 통해 다양한 버전과 포맷 정보를 갖고 있으나 본 절에서는 각 애플리케이션의 가장 중요한 포맷 버전에 대하여 문서보존포맷(PDF/A)으로 변환할 때 발생하는 렌더링 기능성 소실 가능성에 대하여 분석을 수행함

❒ 에뮬레이션 연구를 위해 가장 중요한 것은 애플리케이션 포맷을 인쇄 렌더링 하였을 때 소실되는 속성을 가진 렌더링 기능성이 있는가를 조사하는 것이 중요하므로 본 절에서는 주요 애플리케이션에 대하여 문서보존포맷(PDF/A)으로 변환할 경우 인쇄 렌더링에 의해 속성의 변화를 갖는 렌더링 기능성이 무엇인지를 조사하고자 함

#### 나. 한/글 포맷에서 소실되는 기능성 조사

❒ 한/글 포맷에서 인쇄 렌더링으로 소실되는 렌더링 기능성 조사를 위해 한/글97의 데이터 포맷을 정의한 한/글 문서 포맷 구조 (HWP Document File Formats revision 1.0:20100630) 문서를 참조하여 분석함

❒ 한/글 포맷에서 인쇄 렌더링으로 속성 정보가 소실되는 기능성에 대하여 조사한 결과 아래 표와 같이 OLE개체, 메모장, 교정부호, 양식 개체, 글맵시, 책갈피, 스

크립트 등과 같이 7개의 기능성이 조사됨

<표 3-28> 한/글에 포함된 렌더링 기능성

| 기능성 | 설명 |
|---|---|
| OLE 개체 | 문서내에 Windwos에 설치된 프로그램들이 자신의 기능을 서로 사용하는 기능 |
| 메모장 | 단어나 문장을 선택하고, 부가적인 설명을 추가하는 기능 |
| 교정부호 | 단어나 문장이 틀렸을 때, 교정부호를 사용하여 수정할 수 있게 하는 기능 |
| 양식 개체 | 명령단추, 선택상자, 라디오단추, 입력상자, 목록상자 |
| 글맵시 | 문서내에서 글씨를 아름답게 표현하는 기능 |
| 책갈피 | 문서를 편집하는 도중에 표시를 해 두었다가, 바로 이동할 수 있게 하는 기능 |
| 스크립트 | 양식 개체와 연결하여 여러 가지 선택을 할 수 있게 하는 기능 |

❒ 본 절에서는 각 기능성이 원문에서 표현되는 방식과 인쇄 렌더링 후에 표현되는 방식에 대하여 비교 검토하여 소실되는 속성이 무엇인지를 파악하고자 함.

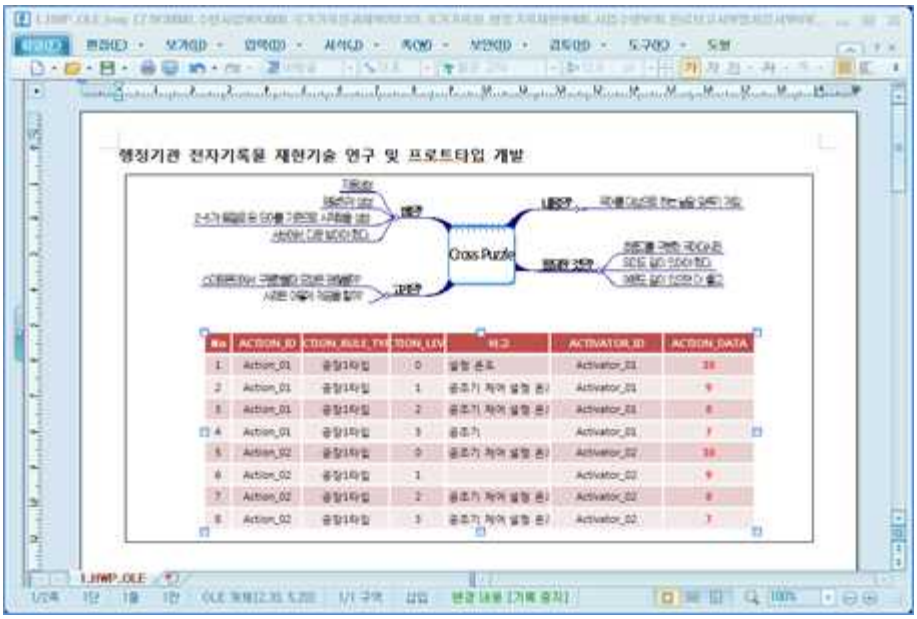
### (1) OLE 개체

❒ OLE 개체 개요

- OLE(Object Linking and Embedding)는 Windows에 설치된 프로그램들이 서로 자신의 기능을 연결해 빌려 쓸 수 있도록 한 것
- 개체(Object)란 윈도우용 애플리케이션이 만들어낸 결과물.
- 한글과 같이 문서를 작성하는 프로그램에서는 문서가 개체가 되고, 음악을 만드는 프로그램에서는 음악 파일이 개체가 된다. 이와 같이 개체는 글자, 그림(정지영상, 동영상), 소리, 차트, 수식, 표 등 종류가 매우 다양.
- 개체 연결과 포함(OLE:Object Linking & Embedding)은 윈도우에서 각각의 애플리케이션들이 생산한 각양각색의 자료들을 독립적으로 사용하는 것이 아니라 각각의 자료들을 서로 유기적로 엮어 놓은 것

❒ 한/글 워드프로세서에서 재현한 한/글 OLE 개체

- OLE 개체(마인드맵, 엑셀) 가 포함된 한/글 포맷을 한/글 프로그램으로 조회

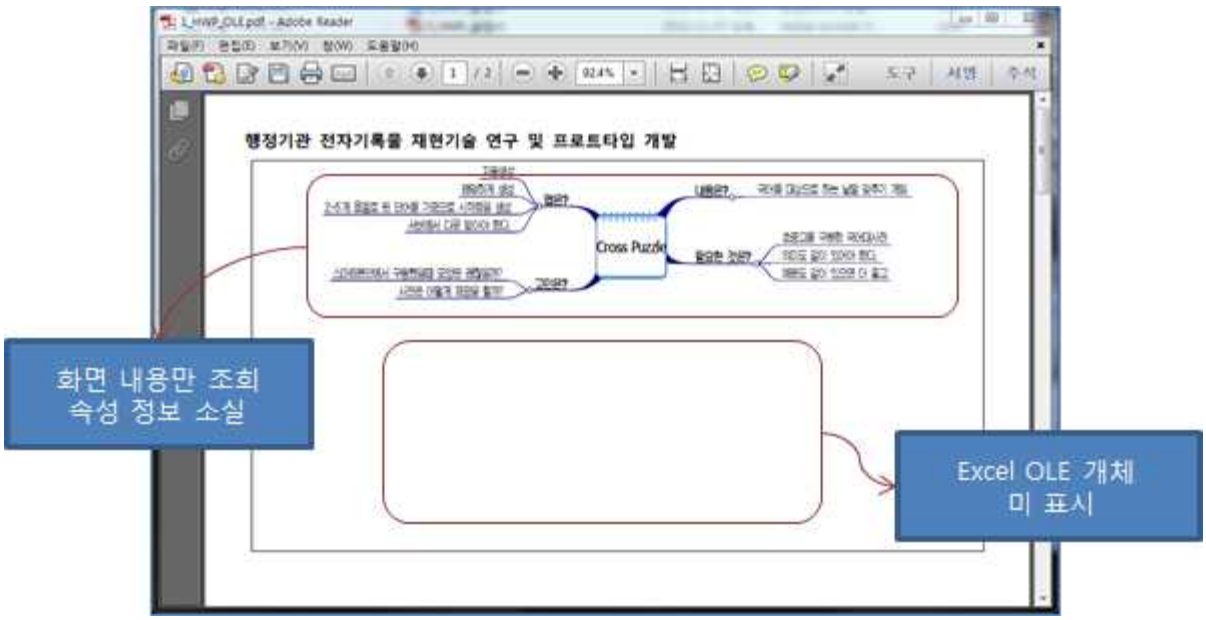



<그림 3-55> 한/글에서 재현한 OLE 개체

○ OLE 개체(마인드맵, 엑셀) 가 포함된 한/글 포맷을 한/글 프로그램으로 조회할 경우 화면 조회가 가능하며 OLE 개체를 더블 클릭하면 해당 애플리케이션 기능을 통해 속성 정보 조회가 가능함

❒ 문서보존포맷(PDF/A)으로 변환된 한/글 OLE 개체

○ OLE 개체 (엑셀, 마인드맵) 가 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-56> 문서보존포맷(PDF/A)으로 변환된 OLE 개체

○ 동영상과 같은 동적 OLE 요소는 문서보존포맷(PDF/A)으로 변환 시 모두 소실되고 정적 화면만 존재함

○ 위의 그림과 같이 마인드맵 OLE 개체를 삽입한 경우 원본에 대한 속성은 모두 소실되고 정적인 그림 형태만 존재하고, 엑셀 OLE 개체를 삽입하지 않고 개체 연결을 한 경우에는 표 자체가 표시되지 않음

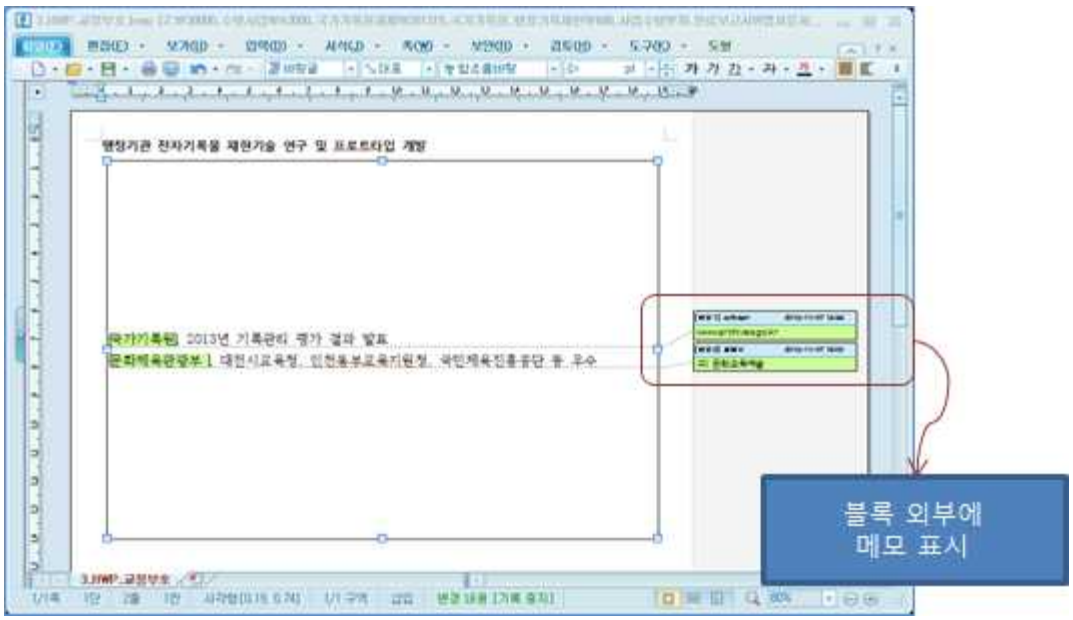
### (2) 메모장

❒ 메모장 개요

○ 메모 기능을 이용하면 편집 중인 문서에서 특정 단어나 블록으로 설정한 문자열에 기억해야 할 내용을 메모로 넣을 수 있음.

❒ 한/글 워드프로세서에서 재현한 한/글 메모장

○ 메모장 기능성이 포함된 한/글 포맷을 한/글 프로그램으로 조회

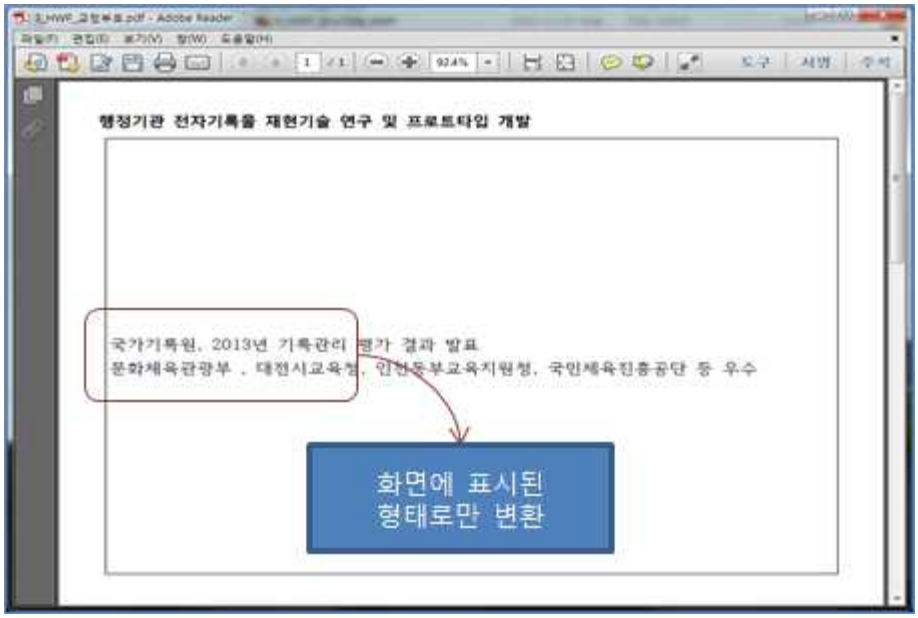



<그림 3-57> 한/글에서 재현한 메모장

○ 한/글에서는 메모장을 등록하면 편집 용지(블록)외부에 메모를 표시

❒ 문서보존포맷(PDF/A)으로 변환된 한/글 메모장 기능성

○ 메모장 기능성이 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-58> 문서보존포맷(PDF/A)으로 변환된 메모장

○ 메모장이 포함된 한/글 문서를 문서보존포맷(PDF/A)으로 변환 시 메모장이 소실되며 메모  존재 자체를 알 수 없음.

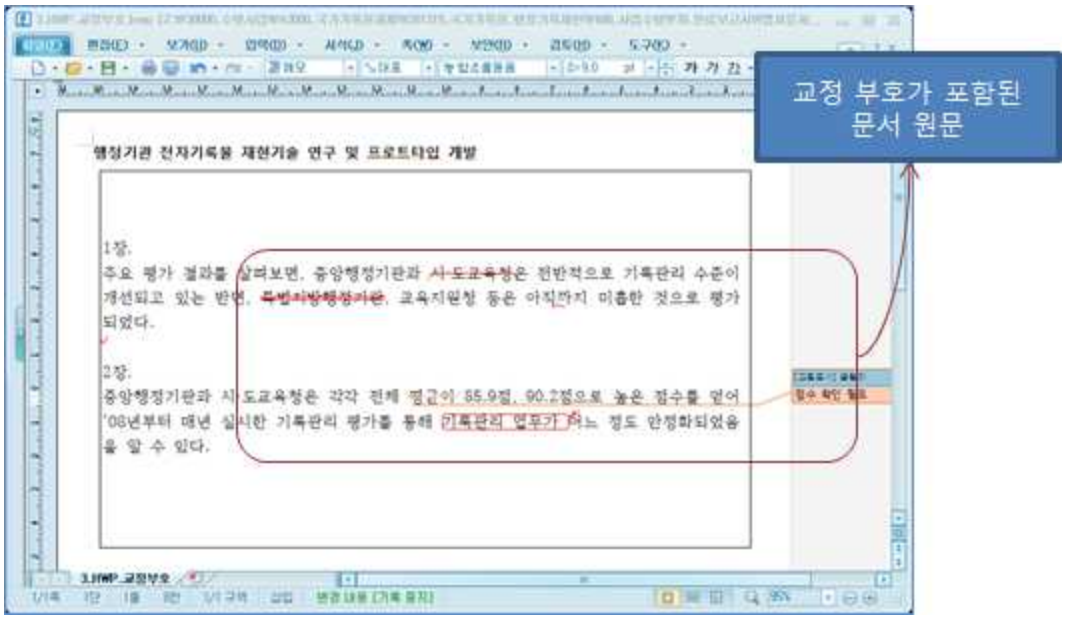
### (3) 교정부호

❒ 교정부호 개요

○ 교정 부호를 이용하여 문서에서 잘못 사용된 맞춤법, 띄어쓰기, 활자 크기, 문장부호, 줄 바꿈, 오자, 탈자 또는 어색한 표현 등을 바로 잡기 위한 다양한 교정부호를 삽입 가능

❒ 한/글 워드프로세서에서 재현한 한/글 교정부호

○ 교정부호 기능성이 포함된 한/글 포맷을 한/글 프로그램으로 조회

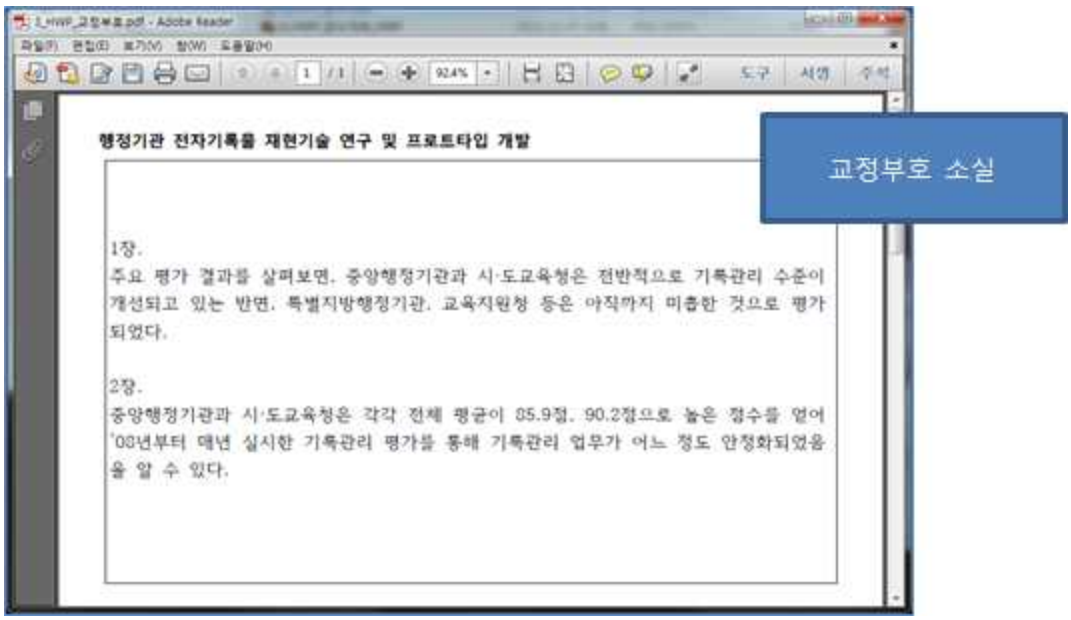



<그림 3-59> 한/글에서 재현한 교정부호

○ 교정부호를 등록하면 한글 편집기에서는 위의 그림과 같이 원문에 교정부호가 포함되어 조회됨

❒ 문서보존포맷(PDF/A)으로 변환된 한/글 교정부호

○ 교정부호 기능성이 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-60> 문서보존포맷(PDF/A)으로 변환된 교정부호

○ 위의 그림과 같이 교정부호가 포함된 한/글 포맷 문서를 보존 포맷으로 변환할 경우 교정부호가 소실되어 사라짐

### (4) 양식 개체

❒ 양식 개체 개요

○ 명령단추, 선택상자, 목록상자, 라디오단추, 입력상자 등 사용자의 명령/선택을 받을 수 있는 기능.

○ Script와 연결하여 여러 가지 사용자에게 선택의 다양성을 줄 수 있음.

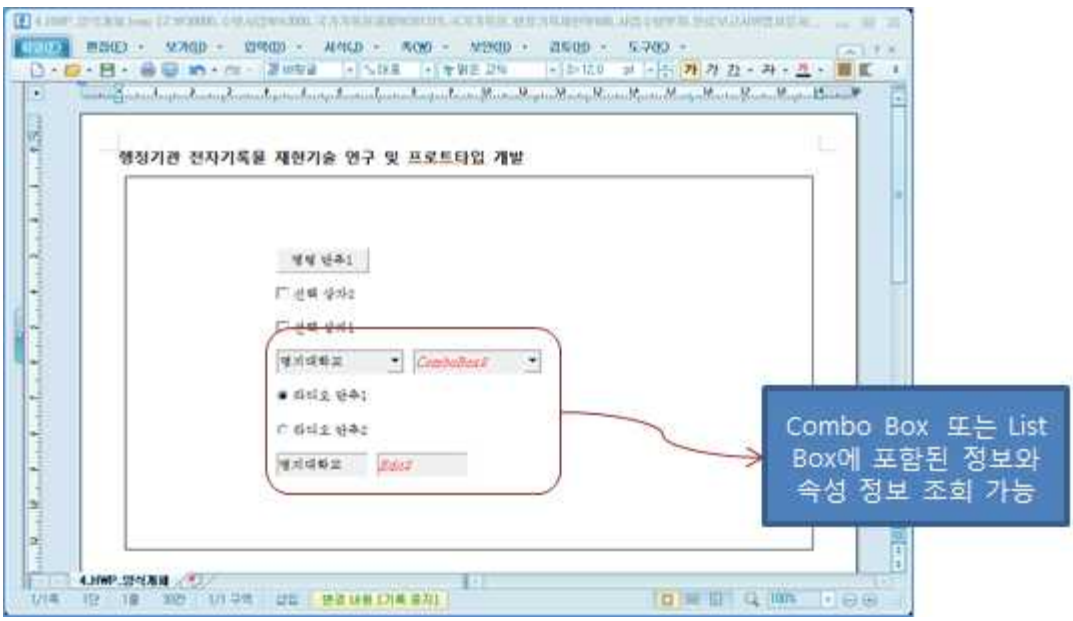
○ Scripts 스토리지에는 Script 코드를 저장.

○ JScriptVersion 스트림에는 Script Version이 저장.

○ DefaultJScript 스트림에는 Script 헤더, 소스, Pre 소스, Post 소스가 저장.

❒ 한/글 워드프로세서에서 재현한 한/글 양식 개체

○ 양식 개체가 포함된 한/글 포맷을 한/글 프로그램으로 조회

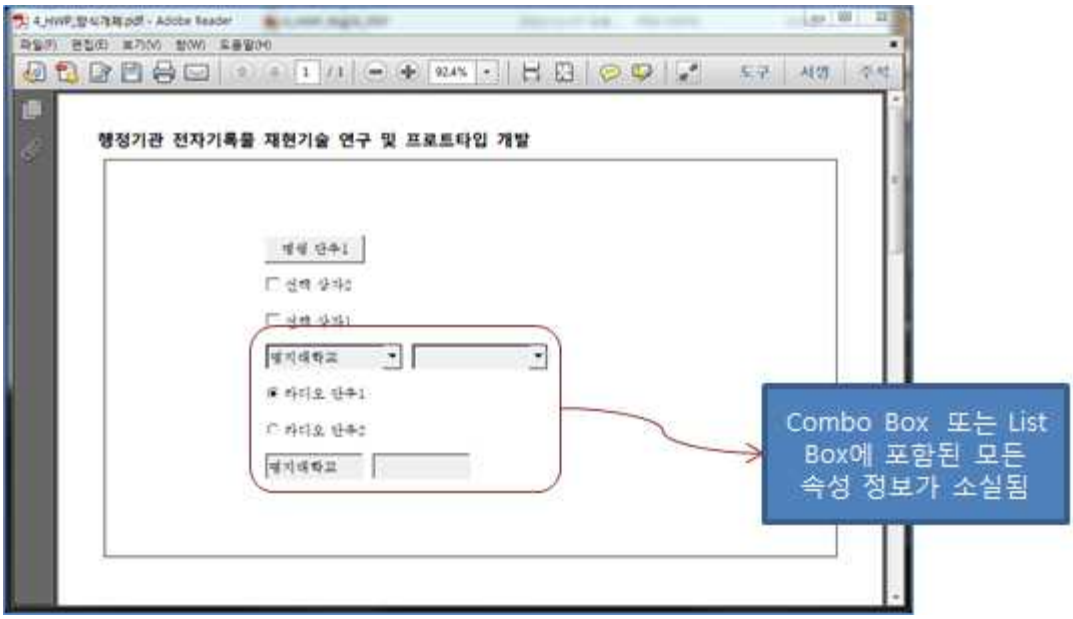



<그림 3-61> 한/글에서 재현한 양식 개체

○ 한글 편집기에서는 위의 그림과 같이 양식 개체 중 명령상자, 목록상자, 입력상자에 등록된 내용 및 속성 정보 조회가 가능함

❒ 문서보존포맷(PDF/A)으로 변환된 한/글 양식 개체

○ 양식 개체가 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-62> 문서보존포맷(PDF/A)으로 변환된 양식 개체

○ 보존 포맷으로 변환할 경우 명령상자, 목록상자, 입력상자에 등록된 내용을 비롯하여 모든 속성 정보가 소실되어 사라짐

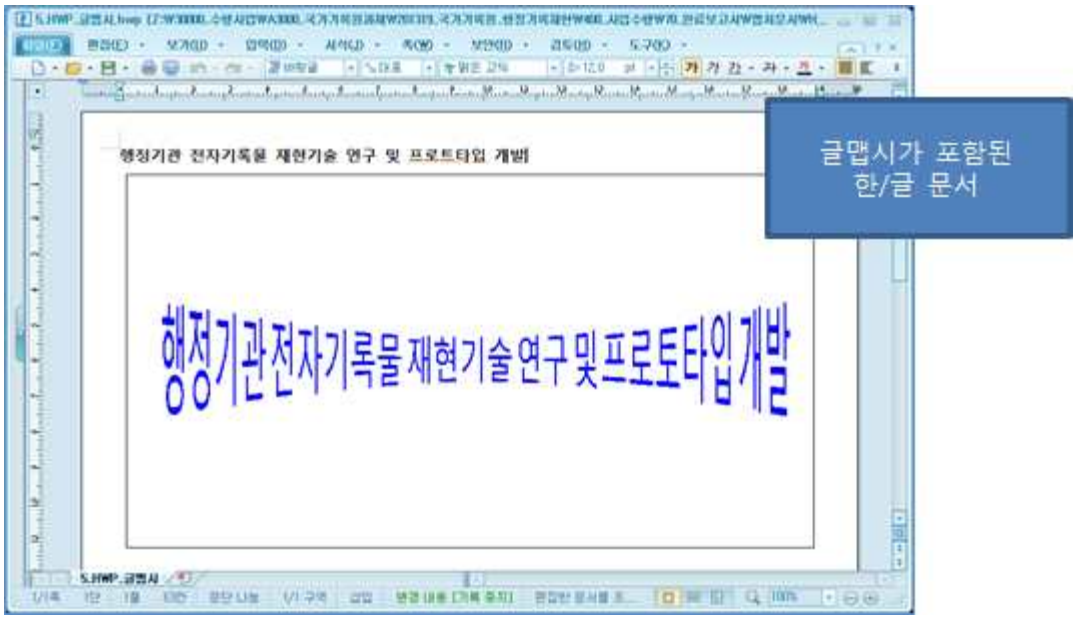
### (5) 글맵시

❒ 글맵시 개요

○ 글맵시는 글자를 구부리거나 외곽선, 채움효과, 그림자 회전 등의 효과를 사용하여 글자를 꾸미는 기능. 글맵시는 한글의 그리기 기능으로 구현되어 있으며, 문서에 포함된 글맵시는 자유롭게 편질할 수 있음.

❒ 한/글 워드프로세서에서 재현한 한/글 글맵시

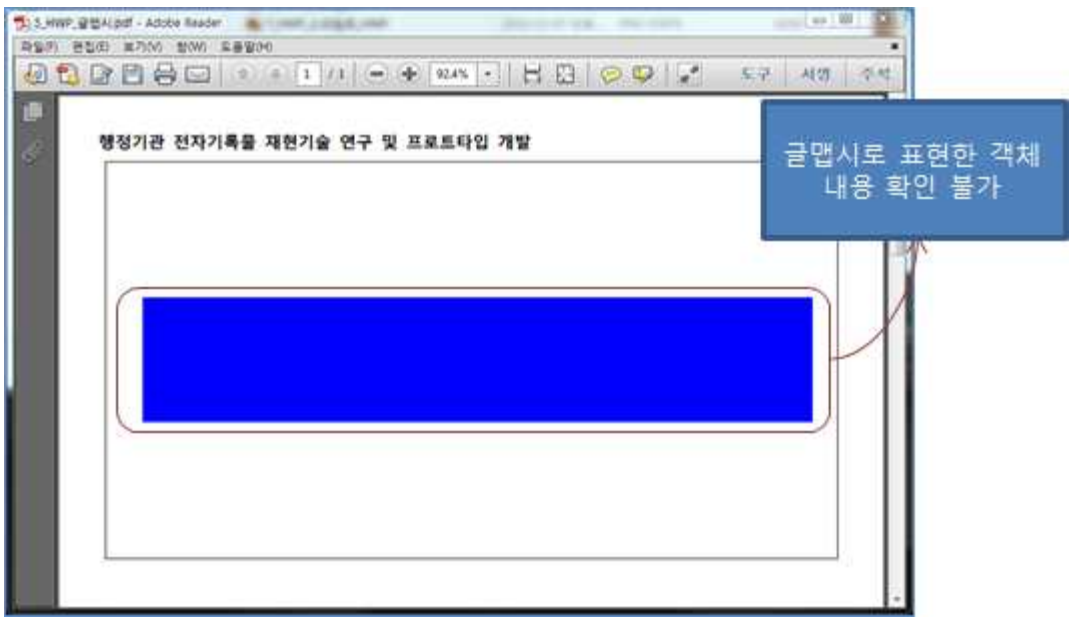
○ 글맵시 기능성이 포함된 한/글 포맷을 한/글 프로그램으로 조회




<그림 3-63> 한/글에서 재현한 글맵시

❒ 문서보존포맷(PDF/A)으로 변환된 한/글 글맵시

○ 글맵시 기능성이 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-64> 문서보존포맷(PDF/A)으로 변환된 글맵시

○ 글맵시 기능성이 포함된 한/글 포맷 문서를 문서보존포맷(PDF/A)으로 변환할 때 사용되는 변환기에 따라 글맵시가 보이기도 하지만 인쇄 렌더링 기반에서 PDF/A 포맷으로 변환하면 글맵시는 사라지게 됨.

### (6) 책갈피

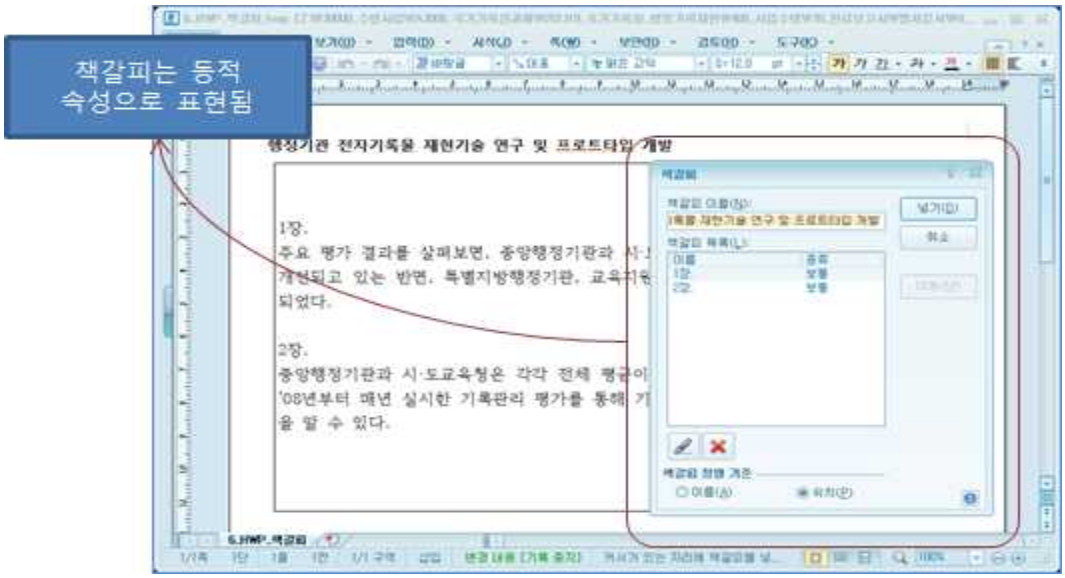
❒ 책갈피 개요

○ 책갈피 기능을 이용하면 문서를 편집하는 도중 본문의 여러 곳에 표시를 해 두었다가 현재 위치에 상관없이 표시해 둔 곳으로 바로 커서를 이동할 수 있음

○ 대용량 문서를 작성할 때, 페이지 이동을 자유롭게 이동하여 편리하게 이용하는 기능

❒ 한/글 워드프로세서에서 재현한 한/글 책갈피

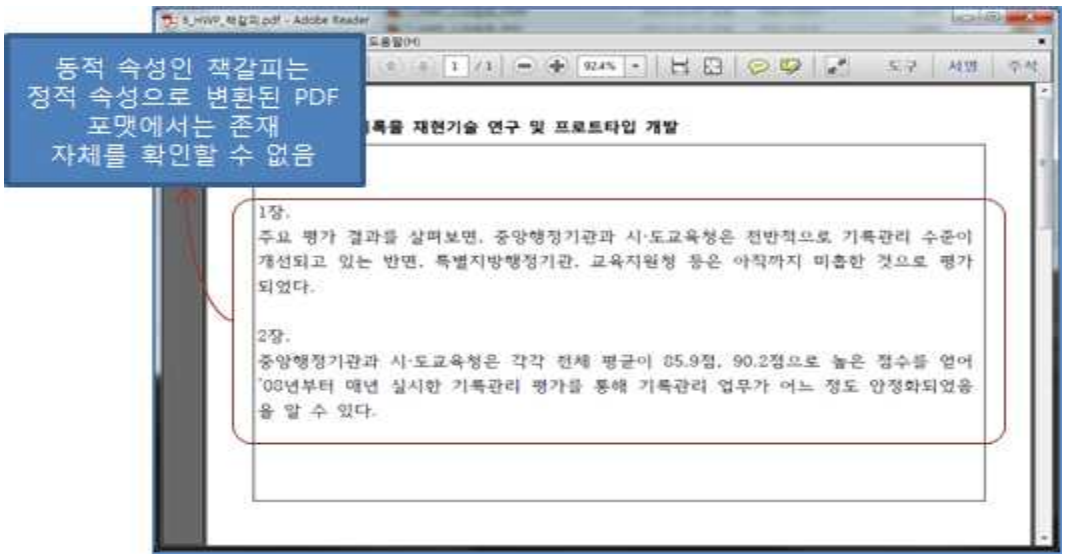
○ 책갈피 기능성이 포함된 한/글 포맷을 한/글 프로그램으로 조회




<그림 3-65> 한/글에서 재현한 책갈피

❒ 문서보존포맷(PDF/A)으로 변환된 한/글 책갈피

○ 책갈피 기능성이 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-66> 문서보존포맷(PDF/A)으로 변환된 책갈피

○ 문서를 저작하는 경우에 편리하게 이용할 수 있는 책갈피 기능은 기능적 속성으로 정적인 PDF/A로 변환되는 순간 모두 소실되어 사라짐.

### (7) 스크립트

❒ 스크립트 개요

○ 한글에서 사용하는 스크립트는 인터넷에서 사용하는 스크립트와 같은 일반적인 스크립트를 사용할 수 있을 뿐만 아니라, 한글의 기능을 제어하기 위한 목적으로 스크립트를 사용할 수 있음.

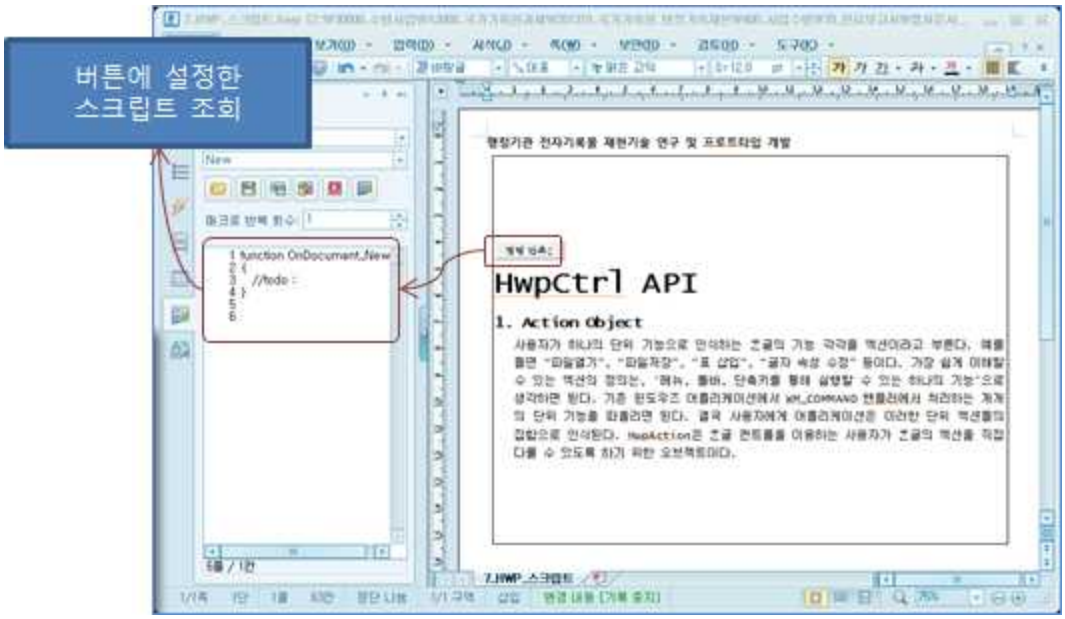
○ 한글의 ActiveID와 함께 스크립트를 사용하면 한글 문서 내에서 한글의 기능을 제어 가능.

○ 수식 개체와 수식 개체 속성으로 구성되어 있고, 전체의 길이는 스크립트의 길이에 따라 가변적으로 구성.

○ 스크립트 기능은 화면에 보이기 보다는 양식 개체와 어울려 화면에 여러 가지 조정하는 기능을 가짐.

❒ 한/글 워드프로세서에서 재현한 한/글 스크립트

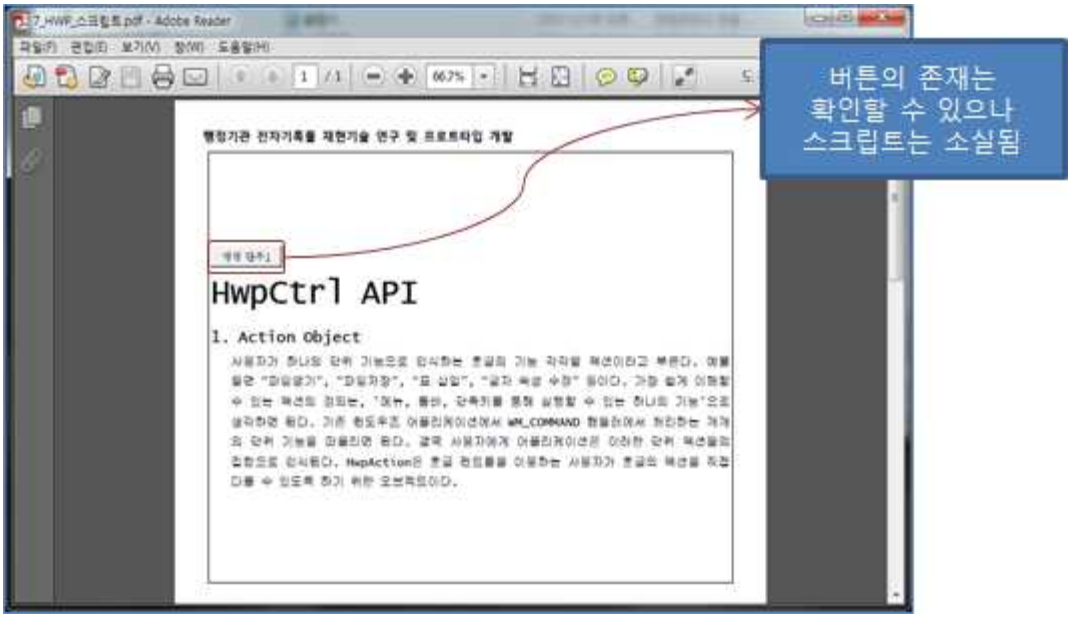
○ 스크립트 기능성이 포함된 한/글 포맷을 한/글 프로그램으로 조회




<그림 3-67> 한/글에서 재현한 스크립트

❒ 문서보존포맷(PDF/A)으로 변환된 한/글 스크립트

○ 스크립트 기능성이 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-68> 문서보존포맷(PDF/A)으로 변환된 스크립트

### 다. 파워포인트 포맷에서 소실되는 기능성 조사

❒ 파워포인트 포맷에서 인쇄 렌더링으로 속성 정보가 소실되는 렌더링 기능성에 대하여 조사한 결과 아래 표와 같이 삽입된 비디오파일, 삽입된 오디오 파일, OLE 개체, 애니메이션 등과 같이 4개의 기능성이 조사됨

<표 3-29> 파워포인트에 포함된 렌더링 기능성

| 기능성 | 설명 |
|---|---|
| 삽입된 비디오 파일 | 문서내에 비디오 파일을 삽입하여 플레이 할 수 있게 하는 기능 |
| 삽입된 오디오 파일 | 문서에서 오디오 파일을 삽입할 수 있고, 오디오 링크를 클릭하여 오디오를 플레이 하는 기능 |
| OLE 개체 | Windwos에서 제공하는 외부 애플리케이션의 결과물을 삽입하는 기능 |
| 애니메이션 | 프리젠테이션을 하기 위해 화면에 나타내기, 날아오기 등의 기능을 사용하여 부가적인 설명을 하는 기능 |

❒ 본 절에서는 각 기능성이 원문에서 표현되는 방식과 인쇄 렌더링 후에 표현되는 방식에 대하여 비교 검토함으로써 소실되는 속성이 무엇인지를 파악하고자 함.

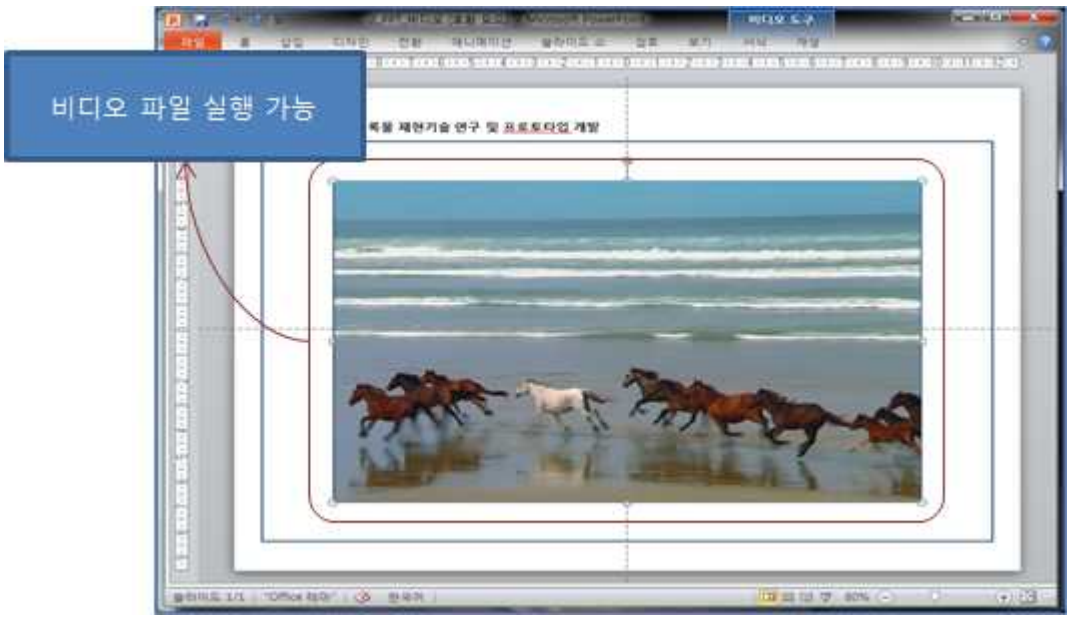
#### (1) 삽입된 비디오 파일

❒ 삽입된 비디오 파일 개요

○ 파워포인트에서 동영상 파일을 재현할 수 있도록 비디오 파일 삽입 기능이 제공됨

❒ 파워포인트에서 재현한 비디오 파일

○ 비디오 파일이 삽입된 파워포인트 포맷 문서를 파워포인트로 조회

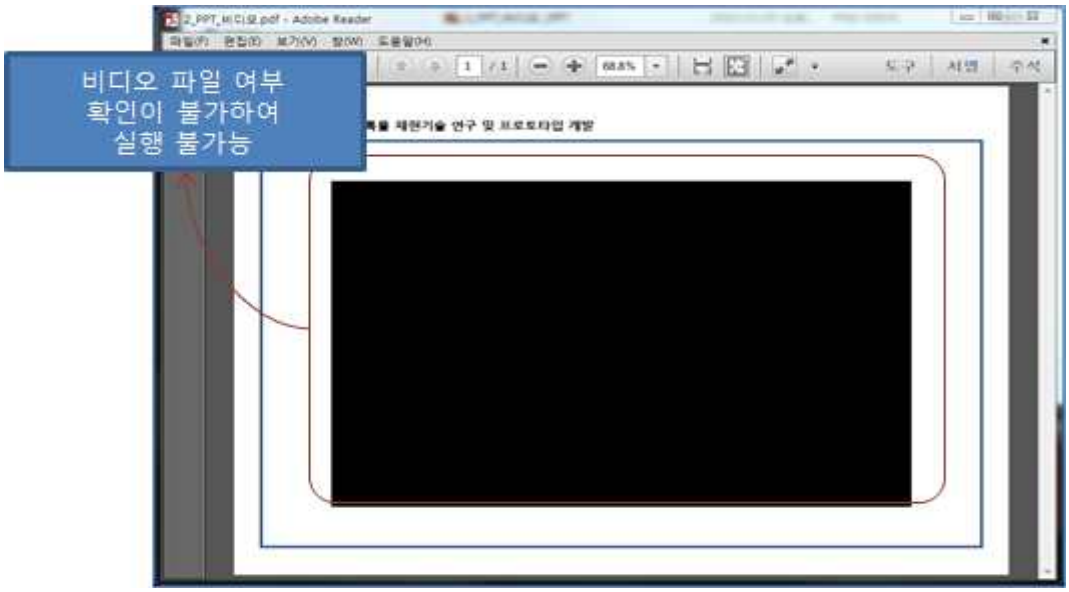



<그림 3-69> 파워포인트에서 재현한 비디오 파일

○ 파워포인트에서는 삽입된 비디오 기능성을 더블클릭하였을 때 동영상 뷰어를 통해 재현이 가능함.

❒ 문서보존포맷(PDF/A)으로 변환된 비디오 파일

○ 비디오 파일이 포함된 파워포인트 포맷 문서가 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-70> 문서보존포맷(PDF/A)으로 변환된 비디오 파일

○ 비디오 파일이 포함된 파워포인트 포맷 문서를 PDF/A 포맷으로 변환하는 경우 검은색 화면으로 변환되어 비디오파일에 대한 실제 정보가 소실됨

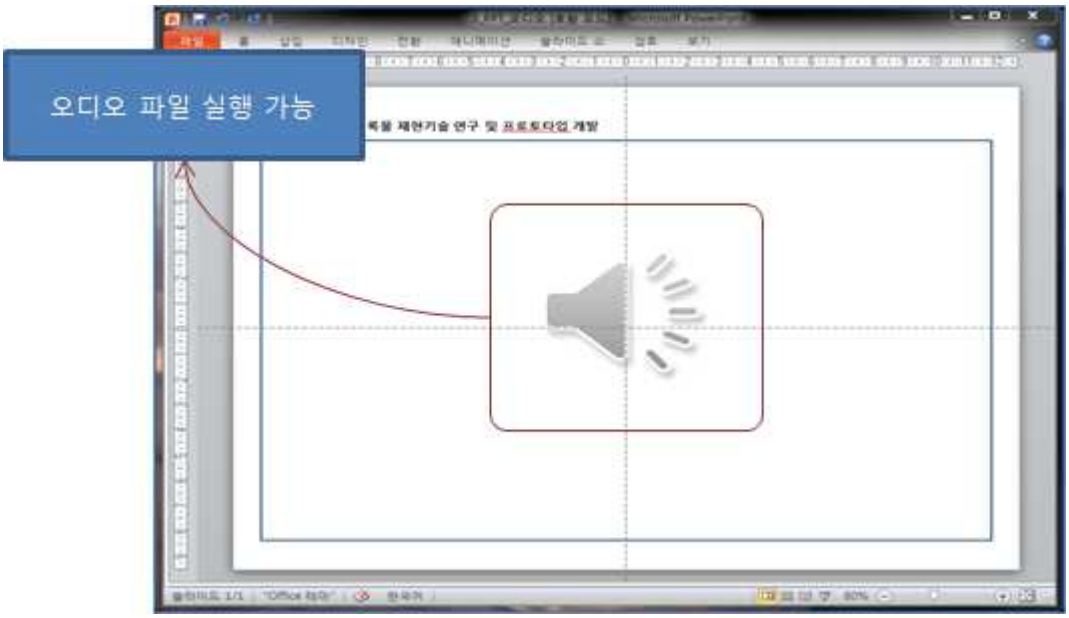
### (2) 삽입된 오디오 파일

❒ 삽입된 오디오파일 개요

○ 파워포인트 문서에는 오디오 파일을 삽입 가능하고, 오디오 링크를 클릭하면 화면 안에서 오디오 플레이 가능

❒ 파워포인트에서 재현한 오디오 파일

○ 오디오 파일이 삽입된 파워포인트 포맷 문서를 파워포인트로 조회

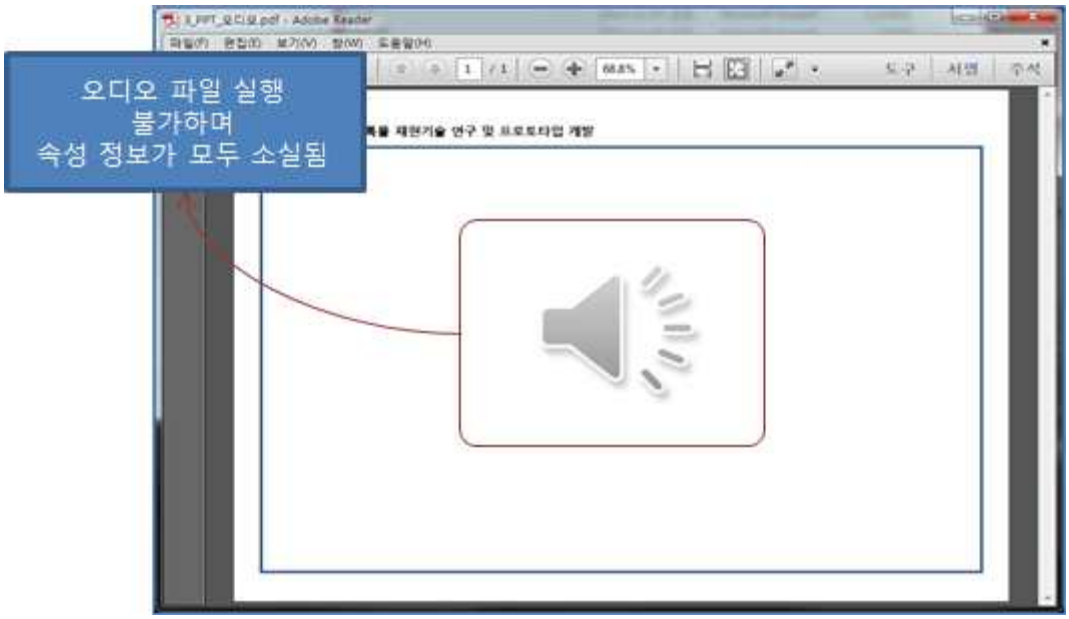



<그림 3-71> 파워포인트에서 재현한 오디오 파일

○ 위의 그림과 같이 파워포인트 문서에서 가운데에 있는 오디오 버튼을 클릭하면 오디오가 플레이되어 음성 정보 확인 가능

❒ 문서보존포맷(PDF/A)으로 변환된 오디오 파일

○ 오디오 파일이 포함된 파워포인트 포맷 문서가 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-72> 문서보존포맷(PDF/A)으로 변환된 오디오 파일

○ 오디오 파일이 포함된 파워포인트 포맷 문서를 PDF/A 문서로 변환하는 경우

오디오 이미지만 존재하고 실제 음성 정보는 확인할 수 없음.

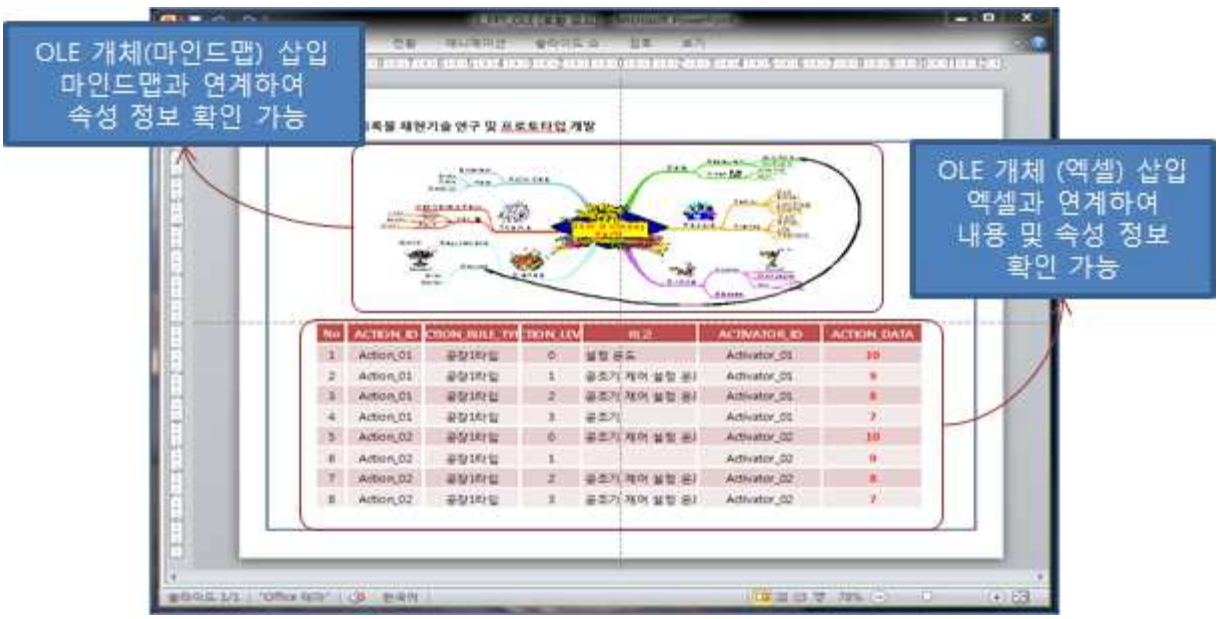
### (3) OLE 개체

❒ OLE 개체 개요

- ○ OLE(Object Linking and Embedding)는 Windows에 설치된 프로그램들이 서로 자신의 기능을 연결해 빌려 쓸 수 있도록 한 것
- ○ 개체(Object)란 윈도우용 애플리케이션이 만들어낸 결과물.
- ○ 한글과 같이 문서를 작성하는 프로그램에서는 문서가 개체가 되고, 음악을 만드는 프로그램에서는 음악파일이 개체가 된다. 이와 같이 개체는 글자, 그림(정지영상, 동영상), 소리, 차트, 수식, 표 등 종류가 매우 다양.
- ○ 개체 연결과 포함(OLE:Object Linking & Embedding)은 윈도우에서 각각의 애플리케이션들이 생산한 각양각색의 자료들을 독립적으로 사용하는 것이 아니라 각각의 자료들을 서로 유기적로 엮어 놓은 것

❒ 파워포인트에서 재현한 OLE 개체

- ○ OLE 개체가 포함된 파워포인트 포맷 문서를 파워포인트로 조회

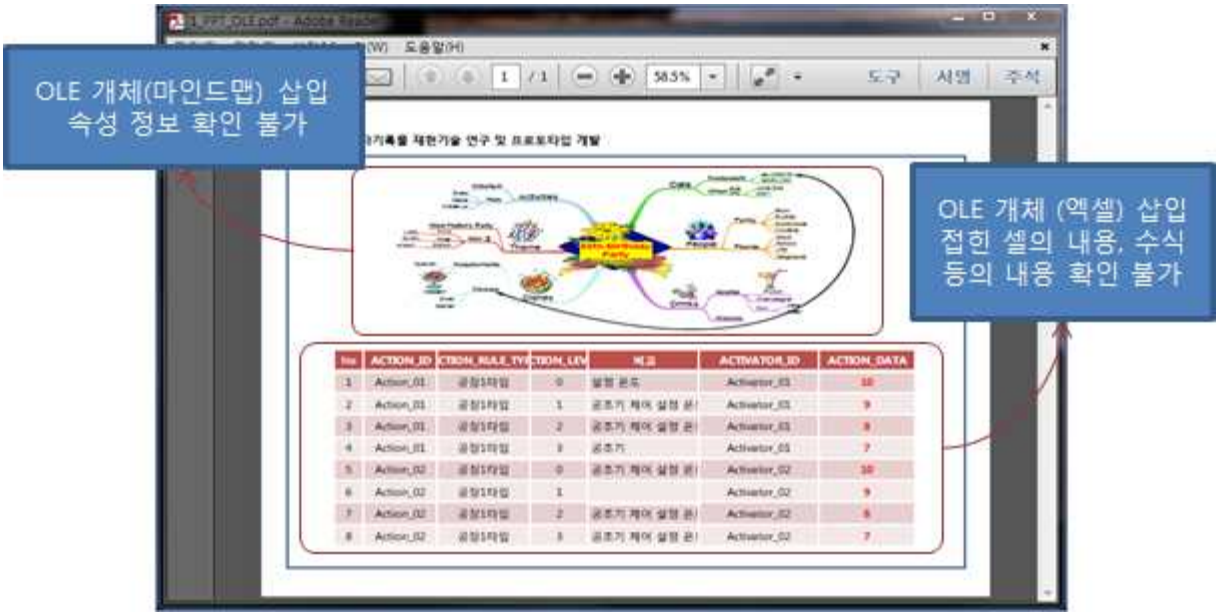



<그림 3-73> 파워포인트에서 재현한 OLE 개체

- ○ OLE 개체가 삽입되어 있는 경우는 내용에 대한 조회가 가능하나 속성 정보 조회가 불가능하며, OLE 개체가 연결되어 있는 경우는 OLE 개체에 대한 내용 및 속성 정보 자체가 표현되지 않음

❒ 문서보존포맷(PDF/A)으로 변환된 OLE 개체

○ OLE 개체가 포함된 파워포인트 포맷 문서가 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-74> 문서보존포맷(PDF/A)으로 변환된 OLE 개체

○ OLE 개체가 포함된 파워포인트 포맷 문서를 PDF/A로 변환한 문서에서는 개체의 속성에 대한 정보를 확인할 수 없으며 엑셀 OLE 개체의 경우 셀이 접힌 경우의 소실된 내용과 수식 등 엑셀에서 소실되는 속성이 동일하게 적용됨

### (4) 애니메이션

❒ 애니메이션 개요

○ 파워포인트 문서를 프리젠테이션할 때 시각적 효과를 극대화하기 위해 기능성에 대한 애니메이션 기능을 활용

○ 애니메이션 기능은 기능성이 화면에 나타나는 방법 및 시간 등 하나의 화면에서 다양한 기능성을 순차적으로 보이도록 하여 프리젠테이션 효과를 향상시키는 동적 기능

❒ 파워포인트에서 재현한 애니메이션

○ 애니메이션 기능이 포함된 파워포인트 포맷 문서를 파워포인트로 조회




<그림 3-75> 파워포인트에서 재현한 애니메이션

○ 다수의 개체가 하나의 화면에 표현되어 있을 경우, 기능성의 순서 조절을 통해 뒤 기능성의 내용을 확인할 수 있음

❒ 문서보존포맷(PDF/A)으로 변환된 애니메이션

○ 애니메이션 기능이 포함된 파워포인트 포맷 문서가 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-76> 문서보존포맷(PDF/A)으로 변환된 애니메이션

○ 애니메이션 기능이 포함되어 기능성이 겹쳐 있는 파워포인트 문서를 PDF/A 포

맷으로 변환하면 정적 문서로 변환되어 객체가 겹쳐지고 뒤 기능성의 내용은 소실되어 사라짐.

○ 위의 그림은 2개의 기능성에 애니메이션 기능이 적용되어 있지만 PDF/A로 변환된 문서에서는 뒤 기능성의 내용을 전혀 파악할 수 없음.

### 라. 엑셀 포맷에서 소실되는 기능성 조사

❒ 엑셀 포맷에서 인쇄 렌더링으로 속성 정보가 소실되는 기능성에 대하여 조사한 결과 아래 표와 같이 스파크라인, 메모장, 차트, 수식 정보, 접힌 셀 등 5개의 기능성이 조사됨

<표 3-30> 엑셀에 포함된 렌더링 기능성

| 기능성 | 설명 |
|---|---|
| 스파크라인 | 단일 워크시트 셀 안의 작은 차트로서 데이터의 추세를 시각적으로 표현 |
| 메모장 | 셀 단위로 부가적인 설명을 추가하는 기능 |
| 차트 | 셀 단위의 정보르 이용하여 시각적으로 표현하는 기능 |
| 수식정보 | 셀 단위의 정보를 이용하여 셀에 합계, 평균과 같은 수식을 이용하여 계산하는 수학적 기능 |
| 접힌 셀 | 셀이 접혀 있는 경우, PDF/A로 변환할 때 접힌 상태로 변화되기 때문에 문제 발생 |

❒ 본 절에서는 각 기능성이 원문에서 표현되는 방식과 인쇄 렌더링 후에 표현되는 방식에 대하여 비교 검토함으로써 소실되는 속성이 무엇인지를 파악하고자 함.

#### (1) 스파크라인(SparkLine)

❒ 스파크라인(SparkLine) 개요

○ 단일 워크시트 셀 안의 작은 차트인 스파크라인을 사용하여 데이터의 추세를 시각적으로 표현.

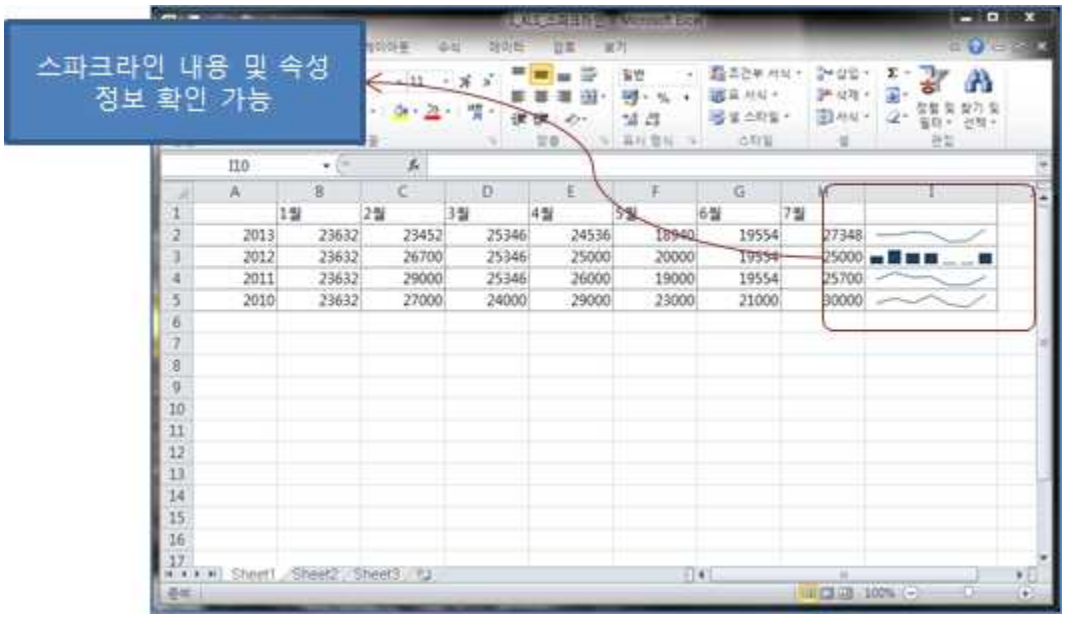
○ 스파크라인은 계절별 증감이나 경기 순환과 같은 내용을 눈에 띄게 표시하고 최대값과 최소값을 다른 색으로 강조 표시할 수 있음.




<그림 3-77> 스파크라인

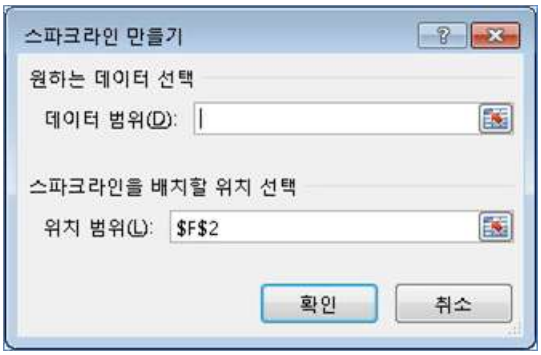
❒ 엑셀에서 재현한 스파크라인

○ 스파크라인 기능성이 포함된 엑셀 포맷 문서를 엑셀로 조회




<그림 3-78> 엑셀에서 재현한 스파크라인

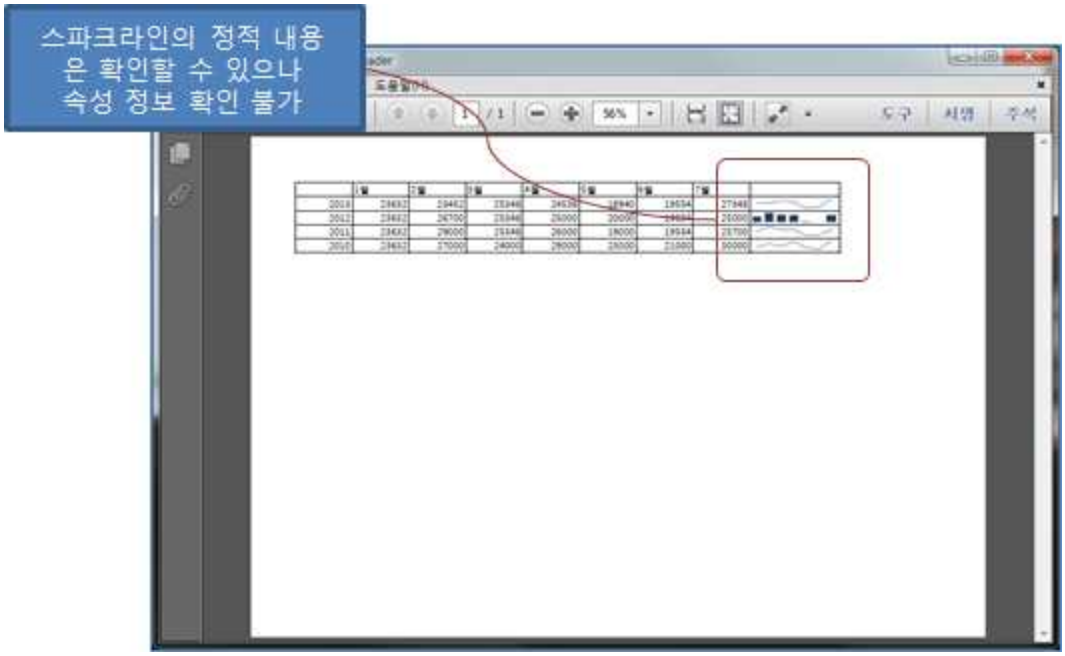
○ 스파크라인이 등록된 경우 스파크라인 셀을 더블클릭하면 아래 그림과 같이 스파크라인의 데이터 범위 확인이 가능하여 스파크라인의 의미 파악이 가능




<그림 3-79> 스파크라인 속성

❒ 문서보존포맷(PDF/A)으로 변환된 스파크라인

○ 스파크라인 기능성이 포함된 엑셀 포맷 문서가 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-80> 문서보존포맷(PDF/A)으로 변환된 스파크라인

○ 스파크라인이 포함된 엑셀 포맷 문서를 PDF/A로 변환하면 스파크라인은 표현되지만 스파크라인의 실제 데이터 범위 확인이 불가하여 스파크라인 차트에 대한 의미 파악이 어려움

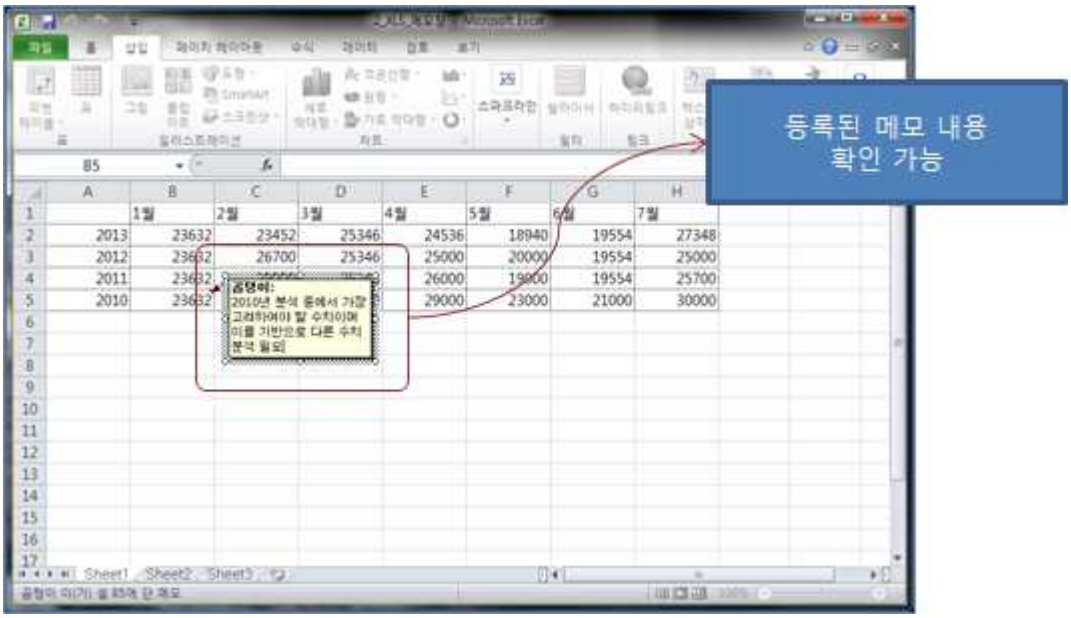
### (2) 메모장

❒ 메모장 개요

○ 메모 기능을 이용하면 편집 중인 문서에서 셀 단위로 설명 문구를 삽입하여 기억해야 할 내용을 메모로 넣을 수 있음.

❒ 엑셀에서 재현한 메모장

○ 메모장 기능성이 포함된 엑셀 포맷 문서를 엑셀로 조회

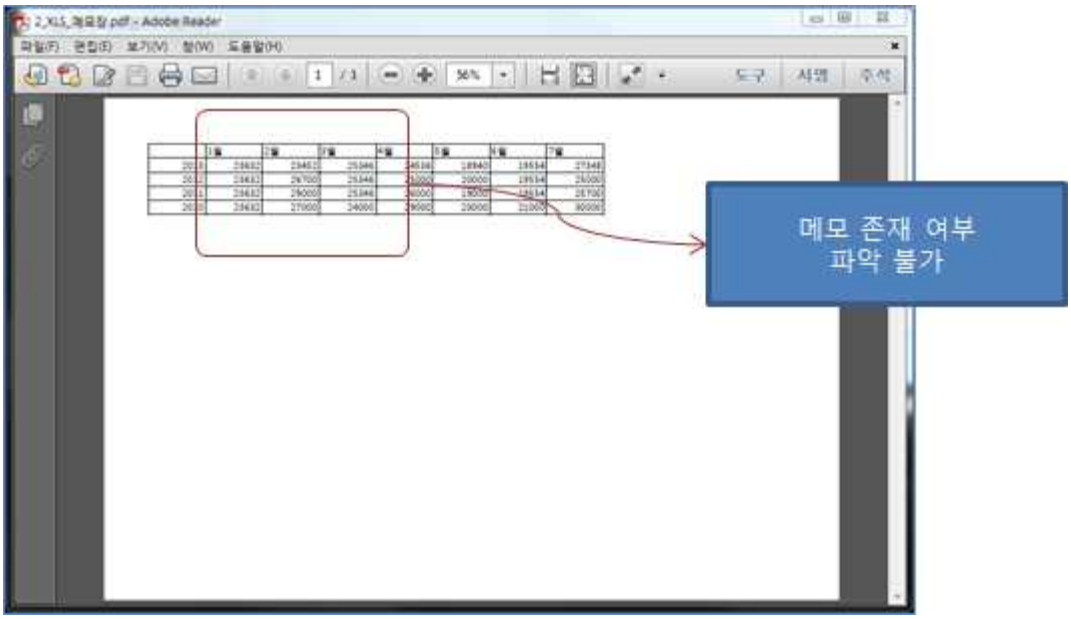



<그림 3-81> 엑셀에서 재현한 메모장

○ 메모가 등록된 셀은 우측 상단에 빨간 삼각기호가 표시되며 삼각기호를 클릭하면 등록된 메모 내용 확인이 가능

❒ 문서보존포맷(PDF/A)으로 변환된 메모장

○ 메모장 기능성이 포함된 엑셀 포맷 문서가 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-82> 문서보존포맷(PDF/A)으로 변환된 메모장

○ PDF/A로 변환한 문서에서는 메모 존재 여부를 확인할 수 없음

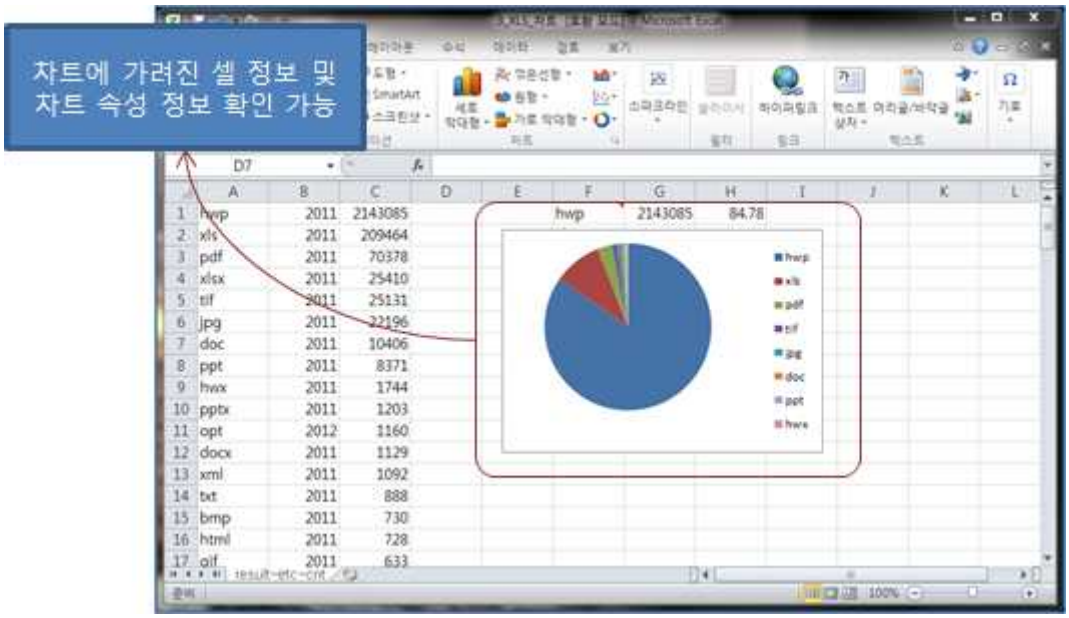
### (3) 차트

❒ 차트 개요

○ 셀의 정보를 이용하여 사용자에게 시각적으로 표현하여 정보를 제공

❒ 엑셀에서 재현한 차트

○ 차트 기능성이 포함된 엑셀 포맷 문서를 엑셀로 조회




<그림 3-83> 엑셀에서 재현한 차트

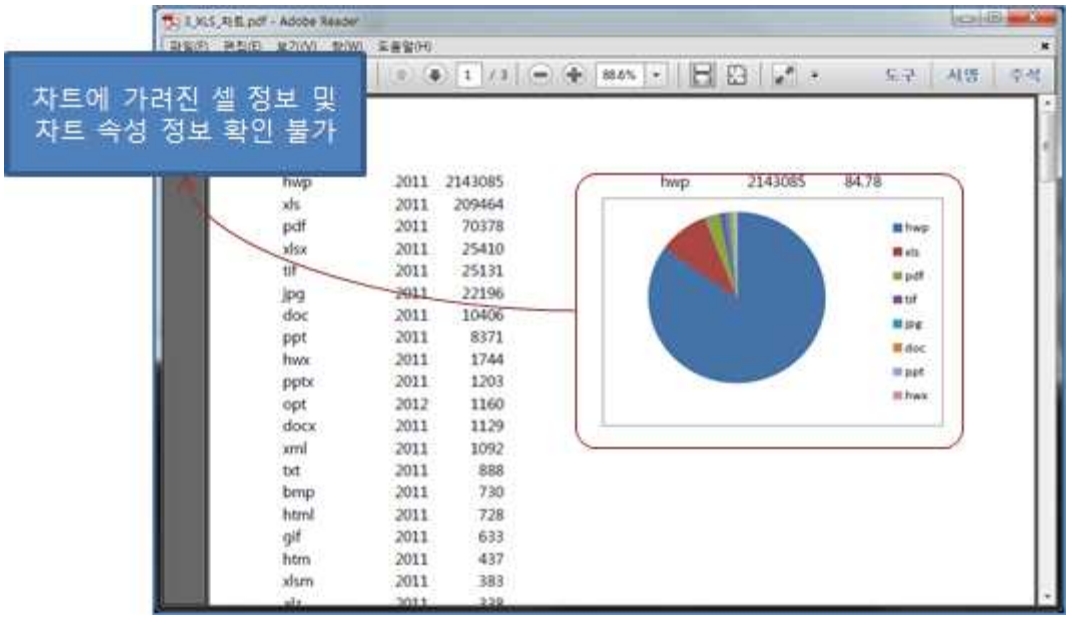
○ 차트를 이동하면 차트에 의해 가려진 데이터를 확인할 수 있고 또한 아래 그림의 차트 데이터 범위 및 속성 정보 조회가 가능하여 차트에 대한 명확한 이해가 가능




<그림 3-84> 차트의 데이터 범위 속성

❒ 문서보존포맷(PDF/A)으로 변환된 차트

○ 차트 기능성이 포함된 엑셀 포맷 문서가 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-85> 문서보존포맷(PDF/A)으로 변환된 차트

○ 차트가 포함된 엑셀 문서를 PDF/A로 변환하면 차트에 의해 가려진 셀 정보는는 한 문서에서는 CELL의 범위를 표시하는 정보가 사라지며, 차트의 속성 정보인 데이터 범위 정보가 소실되어 차트에 대한 정확한 이해가 어려움

### (4) 수식

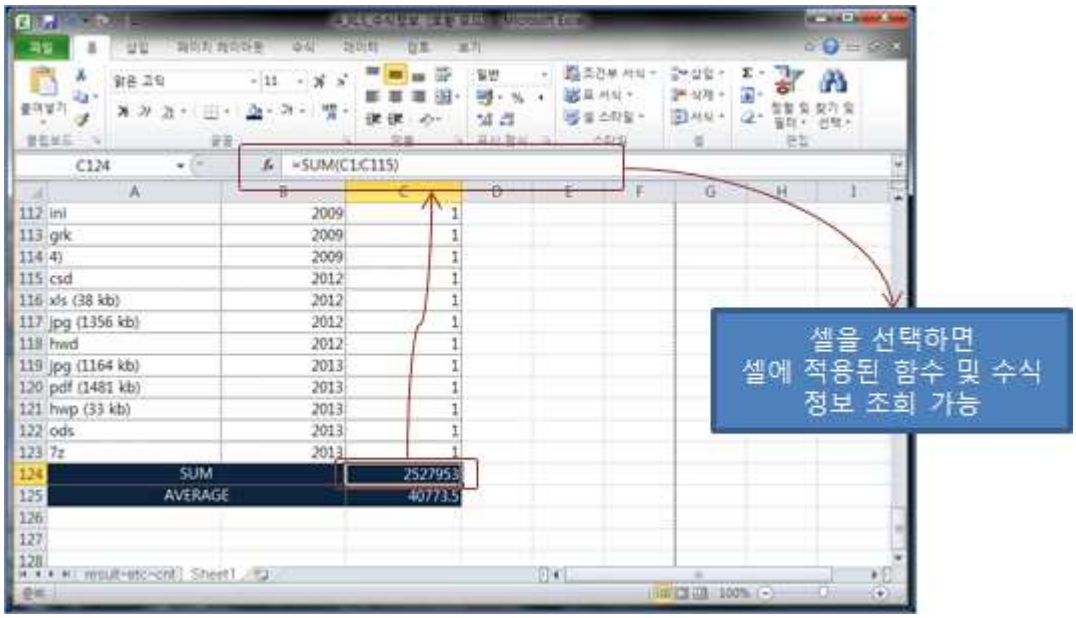
❒ 수식 개요

○ 수식은 엑셀에서 셀에 등록된 데이터의 합산 및 평균을 계산하는 함수와 셀 간의 산술식 계산에 사용

○ 수식 정보는 엑셀의 해당하는 셀을 클릭했을 때, 화면의 상단에 나타남.

❒ 엑셀에서 재현한 수식

○ 셀에 수식이 적용된 엑셀 포맷 문서를 엑셀로 조회

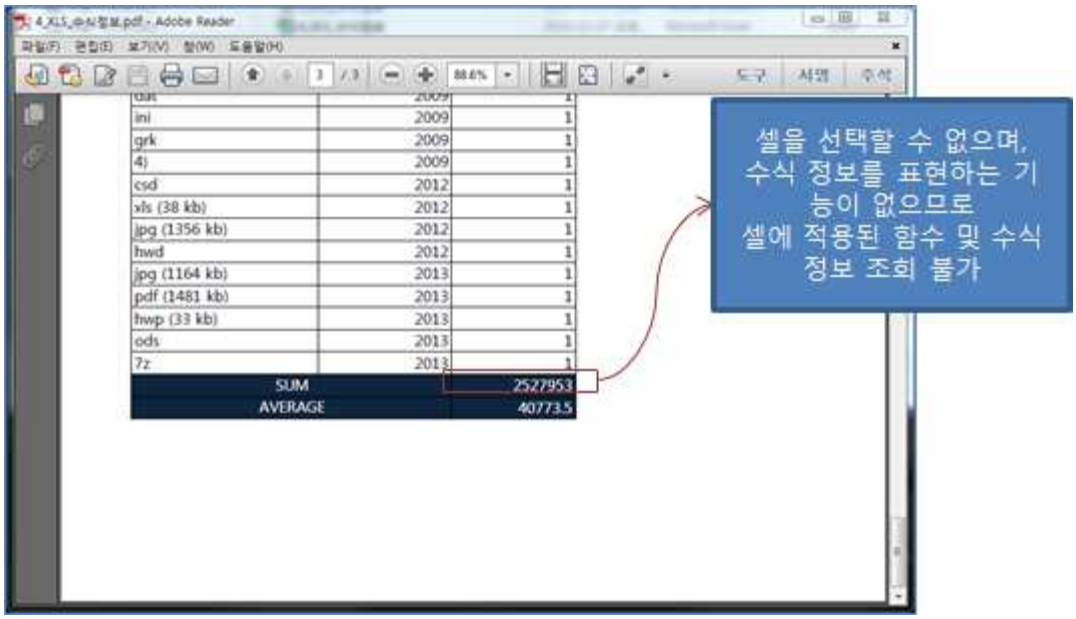



<그림 3-86> 엑셀에서 재현한 수식

○ 각각의 셀을 선택하면 셀에 적용된 수식이 적용된 엑셀 포맷 문서를 엑셀로 조회

❒ 문서보존포맷(PDF/A)으로 변환된 수식

○ 셀에 수식이 적용된 엑셀 포맷 문서가 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-87> 문서보존포맷(PDF/A)으로 변환된 수식

○ 수식이 적용된 엑셀 포맷 문서를 문서보존포맷(PDF/A)으로 변환하면 엑셀에서 현재 화면으로 보여주는 정적인 화면으로 셀을 선택할 수 없으며 셀에 적용된 수식을 확인할 수 없음

○ 즉, 셀에 적용된 함수 및 수식은 모두 소실되고 계산된 결과값만 확인이 가능하므로 다양한 수식이나 함수가 적용된 엑셀 시트는 많은 정보가 소실되었다고 판단할 수 있음

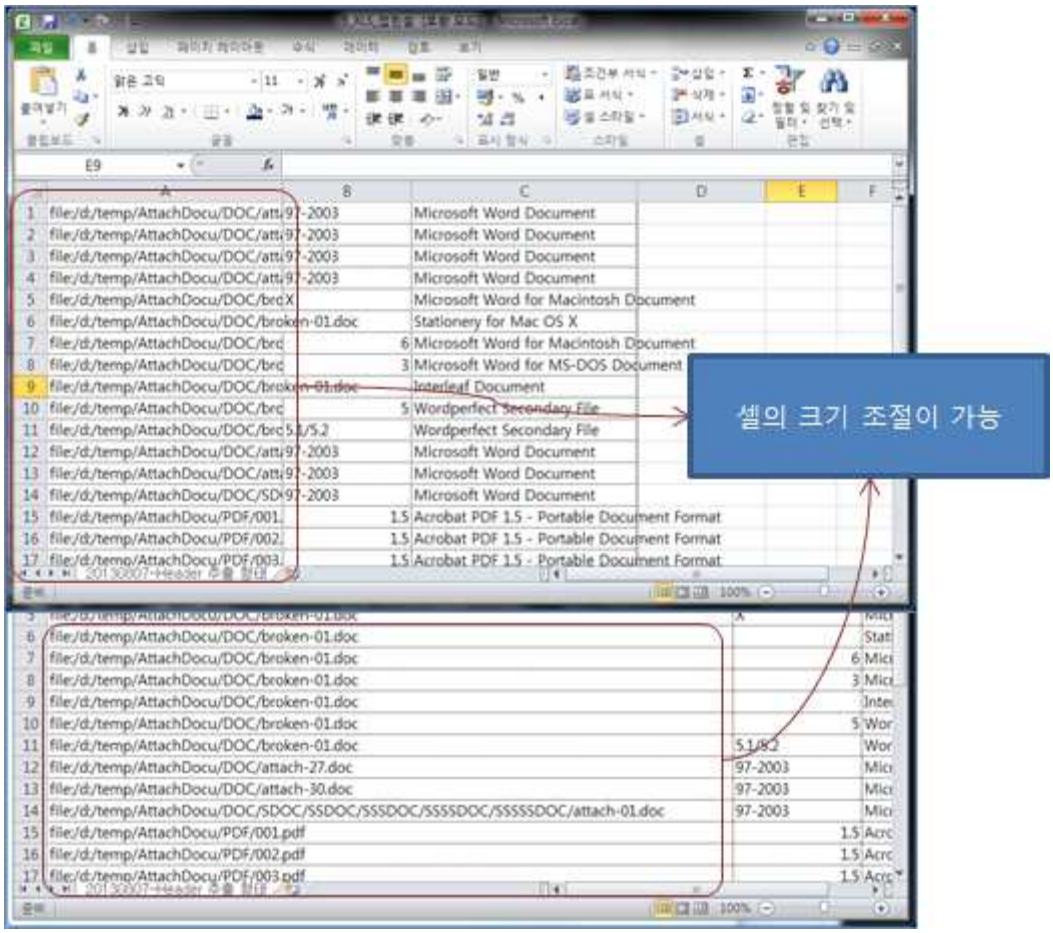
### (5) 접힌 셀

❒ 접힌 셀 개요

○ 스프레드시트인 엑셀은 셀의 크기 조절 등 다양한 기능을 제공하는데, 데이터가 등록된 셀의 크기를 줄여 많은 셀을 하나의 화면에서 작업할 수 있도록 지원

❒ 엑셀에서 재현한 접힌 셀

○ 셀이 접혀 있는 경우가 포함된 엑셀 포맷 문서를 엑셀로 조회

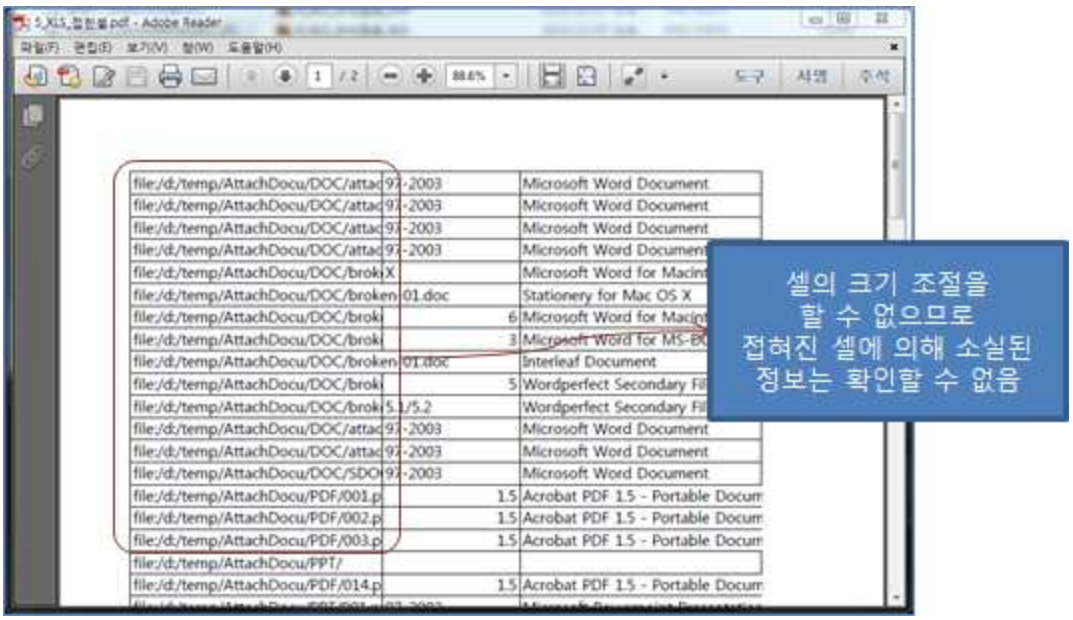



<그림 3-88> 엑셀에서 재현한 접힌 셀

○ 엑셀 포맷 문서에서는 셀의 크기 조절이 가능하여 셀에 등록된 데이터 확인이 언제든 가능

❒ 문서보존포맷(PDF/A)으로 변환된 접힌 셀

○ 셀이 접힌 경우가 포함된 엑셀 포맷 문서가 문서보존포맷(PDF/A)으로 변환되어 PDF 뷰어로 조회한 화면




<그림 3-89> 문서보존포맷(PDF/A)으로 변환된 접힌 셀

○ 셀이 접혀 있는 경우 PDF/A로 변환한 문서에서는 셀이 접힌 상태로 보이기 때문에 화면에 나타나지 않은 데이터는 소실됨

## 마. 소실되는 기능성 분석 결과

❒ 한/글 포맷에서 인쇄 렌더링으로 소실되는 렌더링 기능성 조사 결과

○ 보존문서(PDF/A) 형식으로 변환하는 과정에서 소실되는 렌더링 기능성 조사를 위해 한/글97의 데이터 포맷을 정의한 “한/글 문서 포맷 구조 (HWP Document File Formats revision 1.0:20100630)” 문서를 참고하고 다양한 버전의 애플리케이션 실행을 통해 기능성 조사

○ 한/글 포맷에서 인쇄 렌더링으로 속성 정보가 소실되는 기능성은 OLE 개체, 메모장, 교정부호, 양식 개체, 글맵시, 책갈피, 스크립트 등 7개의 기능성 확인

○ 향후 소프트웨어 에뮬레이션 방식으로 에뮬레이터를 개발하기 위해서는 포맷에 대한 정보와 함께 포맷에 포함된 렌더링 기능성 유무를 확인하는 것이 필요하

므로 렌더링 기능성이 포함된 XML 구조에 대해서는 3.3.4절에서 XML Schema를 제안하고자 함

❒ 파워포인트 포맷에서 인쇄 렌더링으로 소실되는 렌더링 기능성 조사 결과
- ○ 보존문서(PDF/A) 형식으로 변환하는 과정에서 소실되는 렌더링 기능성 조사를 위해 파워포인트의 데이터 포맷을 정의한 “MICROSOFT OFFICE POWERPOINT 97-2007 BINARY FILE FORMAT SPECIFICATION [*.ppt]” 문서를 참고하고 다양한 버전의 애플리케이션 실행을 통해 기능성 조사
- ○ 파워포인트 포맷에서 인쇄 렌더링으로 속성 정보가 소실되는 기능성은 삽입된 비디오 파일, 삽입된 오디오 파일, OLE 개체, 애니메이션 적용에 의해 겹쳐진 기능성 등 4개의 기능성 확인
- ○ 소프트웨어 에뮬레이션 방식으로 에뮬레이터를 개발하기 위해서는 포맷에 대한 정보와 함께 포맷에 포함된 렌더링 기능성 유무를 확인하는 것이 필요하므로 3.3.4절의 DFR 구조 개선에서 렌더링 기능성이 포함된 XML Schema를 제안하고자 함

❒ 엑셀 포맷에서 인쇄 렌더링으로 소실되는 렌더링 기능성 조사 결과
- ○ 보존문서(PDF/A) 형식으로 변환하는 과정에서 소실되는 렌더링 기능성 조사를 위해 엑셀의 데이터 포맷을 정의한 “MICROSOFT OFFICE EXCEL 97-2007 BINARY FILE FORMAT SPECIFICATION [*xls (97-2003) format]” 문서를 참고하고 다양한 버전의 애플리케이션 실행을 통해 기능성 조사
- ○ 엑셀 포맷에서 인쇄 렌더링으로 속성 정보가 소실되는 기능성은 스파크라인, 메모장, 차트, 수식정보, 접힌 셀 등 5개의 기능성 확인
- ○ 파워포인트와 함께 소프트웨어 에뮬레이션 방식으로 에뮬레이터를 개발하기 위해 필요한 XML Schema를 제안하고자 함

### 3.3.3 소실되는 기능성의 속성 분석

#### 가. 기능성의 속성 분석 배경 및 필요성

❒ 애플리케이션 에뮬레이션 방식으로 에뮬레이터를 개발하기 위해서는 디지털 컴포넌트 포맷에 대한 분석과 함께 3.3.2절에서 조사된 렌더링 기능성의 속성을 보존할 수 있도록 렌더링 기능성의 속성 정보 분석이 필요

❒ 본 절에서는 한/글 포맷의 렌더링 기능성에 대해서 “한/글 문서 포맷 구조 (HWP Document File Formats revision 1.0:20100630)” 문서에 정의된 테이블 정

의서 및 속성 정의서를 분석하고, 파워포인트는 “MICROSOFT OFFICE POWERPOINT 97-2007 BINARY FILE FORMAT SPECIFICATION [*.ppt]” 문서, 엑셀은 “MICROSOFT OFFICE EXCEL 97-2007 BINARY FILE FORMAT SPECIFICATION [*xls (97-2003) format]” 문서에 등록된 렌더링 기능성의 컨테이너 정의서 및 속성 정의서를 분석하여 렌더링 기능성 보존 방안을 연구

❒ 포맷 구조 및 기능성 분석을 통해 국가기록원 기술정보은행(DFR)에 렌더링 기능성에 대한 정보를 보완할 수 있도록 XML Schema 개선안을 제시하고자 함.

❒ 또한, 기능성 정보를 기반으로 디지털 컴포넌트의 비트스트림에서 렌더링 기능성의 직접 추출 가능성을 검토하여 문서에 렌더링 기능성이 포함되었는지 자동으로 판별하는 Characterization 연구의 기초를 수립함

### 나. 한/글 포맷에서 소실되는 기능성의 속성 구조[91)]

#### (1) OLE 개체 (Tag ID : HWPTAG_BIN_DATA)

❒ OLE 개체 정의

- ○ OLE 개체에는 크기와 테두리 두께와 속성과 BinData ID를 가지고 있고, BinData ID를 이용하여 바이너리 데이터 영역에 저장되어 있는 바이너리 스트림 정보와 연결
- ○ OLE 기능에 대해서는 HWPTAG_BIN_DATA, 개체 공통 속성을 포함하는 컨트롤, OLE 개체로 나누어져 표현
- ○ HWP_BIN_DATA는 데이터의 저장에 관한 내용을 담고 있음
- ○ 개체 공통 속성을 포함하는 컨트롤은 위치, 가로/세로 크기, 개체에 대한 설명 등과 같은 오브젝트 속성에 대한 정보를 담고 있음
- ○ OLE 개체 부분에서는 OLE 개체 본체에 대한 정보를 담고 있고, 개체 공통속성, OLE 개체 속성에 대한 정보의 상위 레코드임
- ○ 개체 공통 속성에서 크기(오브젝트의 폭, 오브젝트의 높이)와 OLE 개체 속성의 크기(오브젝트 자체의 extent x크기, 오브젝트 자체의 extnt y크기)와 같은 경우는 중복되는 경우가 발생

❒ OLE 개체 관련 속성 정의서

91) 한/글 문서 포맷 구조 Hwp Document File Formats revision 1.0:20100630

○ OLE 개체는 아래 그림과 같이 바이너리 데이터 테이블로 정의

| 자료형 | 길이(바이트) | 설명 | |
|---|---|---|---|
| 그림, OLE 등의 바이너리 데이터 아이템에 대한 정보 | | | |
| UINT16 | 2 | 속성(표 13 참조) | |
| WORD | 2 | Type이 "LINK"일 때, 연결 파일의 절대 경로 길이 (len1) | |
| WCHAR array[len1] | 2×len1 | Type이 "LINK"일 때, 연결 파일의 절대 경로 | |
| WORD | 2 | Type이 "LINK"일 때, 연결 파일의 상대 경로 길이 (len2) | |
| WCHAR array[len2] | 2×len2 | Type이 "LINK"일 때, 연결 파일의 상대 경로 | |
| UINT16 | 2 | Type이 "EMBEDDING"이거나 "STORAGE"일 때, BINDATASTORAGE에 저장된 바이너리 데이터의 아이디 | |
| WORD | 2 | Type이 "EMBEDDING"일 때, 바이너리 데이터의 형식 이름의 길이 (len3) | |
| WCHAR array[len3] | 2×len3 | Type이 "EMBEDDING"일 때 extension("." 제외) | |
| | | 그림의 경우 | jpg |
| | | | bmp |
| | | | gif |
| | | OLE의 경우 | ole |
| 전체 길이 | 가변 | 10 + (2×len1) + (2×len2) + (2×len3) 바이트 | |

<그림 3-90> 바이너리 테이블 정의

○ 바이너리 테이블의 Bit값에 따른 바이너리 데이터 속성 값은 다음과 같음

| 범위 | 구분 | 값 | 설명 |
|---|---|---|---|
| bit 0 - 3 | Type | 0x00000000 | LINK. 그림 외부 파일 참조 |
| | | 0x00000001 | EMBEDDING. 그림 파일 포함 |
| | | 0x00000002 | STORAGE. OLE 포함 |
| bit 4 - 5 | 압축 | 0x00000000 | 스트리지의 디폴트 모드 따라감 |
| | | 0x00000010 | 무조건 압축 |
| | | 0x00000020 | 무조건 압축하지 않음 |
| bit 16 - 17 | 상태 | 0x00000000 | 아직 access 된 적이 없는 상태 |
| | | 0x00010000 | access에 성공하여 파일을 찾은 상태 |
| | | 0x00020000 | access가 실패한 에러 상태 |
| | | 0x00030000 | 링크 access가 실패했으나 무시된 상태 |

<그림 3-91> 바이너리 데이터 속성

○ OLE 개체의 공통 속성은 아래 그림과 같이 정의됨

| 자료형 | 길이(바이트) | 설명 |
|---|---|---|
| UINT32 | 4 | ctrl ID |
| UINT32 | 4 | 속성(표 65 참조) |
| HWPUNIT | 4 | 세로 오프셋 값 |
| HWPUNIT | 4 | 가로 오프셋 값 |
| HWPUNIT | 4 | width 오브젝트의 폭 |
| HWPUNIT | 4 | height 오브젝트의 높이 |
| INT16 | 2 | z-order |
| HWPUNIT16 array[4] | 2×4 | 오브젝트의 바깥 4방향 여백 |
| UINT32 | 4 | 문서 내 각 개체에 대한 고유 아이디(instance ID) |
| WORD | 2 | 개체 설명문 글자 길이(len) |
| WCHAR array[len] | 2×len | 개체 설명문 글자 |
| 전체 길이 | 가변 | 36 + (2×len) 바이트 |

<그림 3-92> 개체 공통 속성

○ OLE 개체는 다음과 같이 OLE 개체와 OLE 개체 속성, OLE 개체 속성의 속성

그리고 테두리 속성으로 정의됨

| 자료형 | 길이(바이트) | 설명 |
|---|---|---|
| BYTE stream | n | 개체 공통 속성(표 63 참조) |
| BYTE stream | 24 | OLE 개체 속성(표 112 참조) |
| 전체 길이 | 가변 | 24 + n 바이트 |

<그림 3-93> OLE 개체

| 자료형 | 길이(바이트) | 설명 |
|---|---|---|
| UINT16 | 2 | 속성(표 113 참조) |
| INT32 | 4 | 오브젝트 자체의 extent x크기 |
| INT32 | 4 | 오브젝트 자체의 extent y크기 |
| UINT16 | 2 | 오브젝트가 사용하는 스토리지의 BinData ID |
| COLORREF | 4 | 테두리 색 |
| INT32 | 4 | 테두리 두께 |
| UINT32 | 4 | 테두리 속성(표 82 참조) |
| 전체 길이 | 24 | |

<그림 3-94> OLE 개체 속성

| 범위 | 구분 | 값 | 설명 |
|---|---|---|---|
| bit 0 - 7 | | | DVASPECT_CONTENT = 1,<br>DVASPECT_THUMBNAIL = 2,<br>DVASPECT_ICON = 4,<br>DVASPECT_DOCPRINT = 8 |
| bit 8 | | | TRUE if moniker is assigned |
| bit 9 - 15 | | | 베이스라인. 0은 디폴트(85%)를 뜻하고, 1 - 101이 0 - 100%를 나타낸다. 현재는 수식만이 베이스라인을 별도로 가진다. |
| bit 16 - 21 | 개체 종류 | 0 | Unknown |
| | | 1 | Embedded |
| | | 2 | Link |
| | | 3 | Static |
| | | 4 | Equation |

<그림 3-95> OLE 개체 속성의 속성

| 범위 | 구분 | 값 | 설명 |
|---|---|---|---|
| bit 0 - 5 | 선 종류 | | |
| bit 6 - 9 | 선 끝 모양 | 0 | round 그림일 때는 "Round", 그리기 개체들일 때는 "Flat"이 디폴트 |
| | | 1 | flat |
| bit 10 - 15 | 화살표 시작 모양 | | |
| bit 16 - 21 | 화살표 끝 모양 | 0 | 모양 없음 |
| | | 1 | 화살 모양 |
| | | 2 | 오목한 화살모양 |
| | | 3 | 속이 빈 다이아몬드 모양 |
| | | 4 | 속이 빈 원 모양 |
| | | 5 | 속이 빈 사각 모양 |
| | | 6 | 속이 채워진 다이아몬드 모양 |
| | | 7 | 속이 채워진 원 모양 |
| | | 8 | 속이 채워진 사각 모양 |
| bit 22 - 25 | 화살표 시작 크기 | | |
| bit 26 - 29 | 화살표 끝 크기 | 0 | 작은-작은 |
| | | 1 | 작은-중간 |
| | | 2 | 작은-큰 |
| | | 3 | 중간-작은 |
| | | 4 | 중간-중간 |
| | | 5 | 중간-큰 |
| | | 6 | 큰-작은 |
| | | 7 | 큰-중간 |
| | | 8 | 큰-중간 |
| bit 30 | 시작부분 화살표 채움 여부 | | |
| bit 31 | 끝부분 화살표 채움 여부 | | |

<그림 3-96> 테두리 속성

### (2) 메모장 (TagID : HWPTAG_MEMO_SHAPE, HWPTAG_MEMO_LIST)

❒ 메모장 정의

○ 메모에 관한 TagID는 2개이며, 컨트롤 아이디는 MAKE_4CHID('%','%','m','e'

○ 개체 표현을 위해서 개체 공통 속성을 사용.

○ 화면 구성을 어떻게 하여야 하는지에 대한 정보가 부족하고 렌더링을 위해서는 메모 데이터를 화면에 표현하는 방식에 대한 설명이 보강되어야 함.

❒ 메모장 관련 속성 정의서

○ 메모장 개체의 공통 속성은 다음과 같이 정의됨

| 자료형 | 길이(바이트) | 설명 |
|---|---|---|
| UINT32 | 4 | ctrl ID |
| UINT32 | 4 | 속성(표 65 참조) |
| HWPUNIT | 4 | 세로 오프셋 값 |
| HWPUNIT | 4 | 가로 오프셋 값 |
| HWPUNIT | 4 | width 오브젝트의 폭 |
| HWPUNIT | 4 | height 오브젝트의 높이 |
| INT16 | 2 | z-order |
| HWPUNIT16 array[4] | 2×4 | 오브젝트의 바깥 4방향 여백 |
| UINT32 | 4 | 문서 내 각 개체에 대한 고유 아이디(instance ID) |
| WORD | 2 | 개체 설명문 글자 길이(len) |
| WCHAR array[len] | 2×len | 개체 설명문 글자 |
| 전체 길이 | 가변 | 36 + (2×len) 바이트 |

<그림 3-97> 개체 공통 속성

○ 메모장 개체 공통 속성의 속성은 다음과 같이 정의됨

| 범위 | 구분 | 값 | 설명 |
|---|---|---|---|
| bit 0 | 글자처럼 취급 여부 | | |
| bit 1 | 예약 | | |
| bit 2 | 줄 간격에 영향을 줄지 여부 | | |
| bit 3-4 | 세로 위치의 기준 (VertRelTo) | 0 | paper |
| | | 1 | page |
| | | 2 | para |
| bit 5-7 | 세로 위치의 기준에 대한 상대적인 배열 방식 | 0 | VerRelTo이 'paper'나 'page' 이면 top, 그렇지 않으면 left |
| | | 1 | VerRelTo이 'paper'나 'page' 이면 center |
| | | 2 | VerRelTo이 'paper'나 'page' 이면 bottom, 그렇지 않으면 right |
| | | 3 | VerRelTo이 'paper'나 'page' 이면 inside |
| | | 4 | VerRelTo이 'paper'나 'page' 이면 outside |
| bit 8-9 | 가로 위치의 기준 (HorzRelTo) | 0 | page |
| | | 1 | page |
| | | 2 | column |
| | | 3 | para |
| bit 10-12 | HorzRelTo에 대한 상대적인 배열 방식 | | bit 5-7 참조 |
| bit 13 | VertRelTo이 'para'일 때 오브젝트의 세로 위치를 본문 영역으로 제한할지 여부 | 0 | off |
| | | 1 | on |
| bit 14 | 다른 오브젝트와 겹치는 것을 허용할지 여부 | | 오브젝트의 위치가 본문 영역으로 제한되면 언제나 false로 간주한다. |
| bit 15-17 | 오브젝트 폭의 기준 | 0 | paper |
| | | 1 | page |
| | | 2 | coloum |
| | | 3 | para |
| | | 4 | absolute |
| bit 18-19 | 오브젝트 높이의 기준 | 0 | paper |
| | | 1 | page |
| | | 2 | absolute |
| bit 20 | VertRelTo이 para일 때 크기 보호 여부 | 0 | off |
| | | 1 | on |
| bit 21-23 | 오브젝트 주위를 텍스트가 어떻게 흘러갈지 지정하는 옵션 | | |
| | Square | 0 | bound rect를 따라 |
| | Tight | 1 | 오브젝트의 outline을 따라 |
| | Through | 2 | 오브젝트 내부의 빈 공간까지 |
| | TopAndBottom | 3 | 좌, 우에는 텍스트를 배치하지 않음 |
| | BehindText | 4 | 글과 겹치게 하여 글 뒤로 |
| | InFrontOfText | 5 | 글과 겹치게 하여 글 앞으로 |
| bit 24-25 | 오브젝트의 좌/우 어느 쪽에 글을 배치할지 지정하는 옵션 | 0 | BothSides |
| | | 1 | LeftOnly |
| | | 2 | RightOnly |
| | | 3 | LargestOnly |
| bit 26-28 | 이 개체가 속하는 번호 범주 | 0 | none |
| | | 1 | figure |
| | | 2 | table |
| | | 3 | equation |

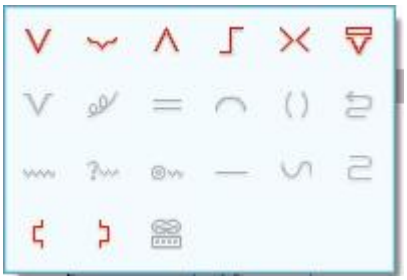
<그림 3-98> 개체 공통 속성의 속성

### (3) 교정부호 (Tag ID : HWPTAG_PARA_RANGE_TAG)

❒ 교정부호

○ 교정 부호에 관련된 태그는 영역 태그로서 “형광펜 기능”등과 동일하게 사용되고 관련 정보는 다음과 같음

○ 교정부호도 기능(삭제, 수정, 줄바꿈, 붙여쓰기, 자리 옮김표 등)에 따라 여러 가지 모양으로 화면에 나타나는데, 이러한 부분에 대한 기술 내용이 없음



<그림 3-99> 교정부호

○ 기술 문서를 기반으로 뷰어를 개발하기 위해서는 기술정보은행(DFR)에 교정부호 렌더링을 위한 표현 정보도 포함되어야 함

❒ 교정부호 관련 속성 정의서

○ 교정부호를 저장하기 위한 문단의 영역 태그는 다음과 같이 정의됨

| 자료형 | 길이(바이트) | 설명 |
|---|---|---|
| UINT32 | 4 | 영역 시작 |
| UINT32 | 4 | 영역 끝 |
| UINT32 | 4 | 태그(종류 + 데이터)<br>: 상위 8비트가 종류를 하위 24비트가 종류별로 다른 설명을 부여할 수 있는 임의의 데이터를 나타낸다. |
| 전체 길이 | 가변 | 12×n |

<그림 3-100> 문단의 영역 태그

○ 영역 태그는 12Byte로 구성되며 영역 시작(4Byte), 영역 끝(4Byte) 그리고 나머지 4Byte중 8Bit는 종류를 나타내며 24Bit는 종류별로 다른 설명을 부여할 수 있는 임의의 데이터를 나타냄

### (4) 양식 개체 (Tag ID : HWPTAG_FORM_OBJECT)

❒ 양식 개체 정의

○ 양식 개체에 대한 HWPTAG_FORM_OBJECT를 사용하고 있지만, 문서에서는 존재하지 않음

○ 기술 문서를 기반으로 뷰어를 개발하기 위해서는 기술정보은행(DFR)에 교정부호 렌더링을 위한 표현 정보도 포함되어야 함

### (5) 글맵시 (Tag ID : HWPTAG_SHAPE_COMPONENT_TEXTART)

❒ 글맵시 정의

○ 한/글 2010에 기능상으로는 존재하나 기술문서 “HWP Document File 5.0” 문서

에서는 존재하지 않음.

○ 기술 문서를 기반으로 뷰어를 개발하기 위해서는 기술정보은행(DFR)에 교정부호 렌더링을 위한 표현 정보도 포함되어야 함

❒ 글맵시 속성 정의서

○ TEXTART 엘리먼트, TEXTARTSHAPE 엘리먼트, OUTLINEDATA 엘리먼트에 대한 속성은 다음과 같이 정의됨

| TEXTART | | | | |
|---|---|---|---|---|
| 설명 | | 글맵시 | | |
| 부모 엘리먼트 | | TEXT | | |
| 자식 엘리먼트 | | TEXTARTSHAPE, OUTLINEDATA | | |
| 속성 | Text | 내용 | | |
| | X0 | (X0, Y0) (X1, Y1) (X2, Y2) (X3, Y3) : master 좌표 | | |
| | Y0 | | | |
| | X1 | | | |
| | Y1 | | | |
| | X2 | | | |
| | Y2 | | | |
| | X3 | | | |
| | Y3 | | | |

<그림 3-101> TEXTART 엘리먼트(HWPML)

| TEXTARTSHAPE | | | | |
|---|---|---|---|---|
| 설명 | | TEXTARTSHAPE | | |
| 부모 엘리먼트 | | TEXTART | | |
| 자식 엘리먼트 | | SHADOW | | |
| 속성 | FontName | 폰트 이름 | | |
| | FontStyle | 폰트 스타일 | | Regular |
| | FontType | HPDT_TTF or HPDT_HFT | ttf \| htf | ttf |
| | TextShape | 0(shape 1) ~ 39(shape 40) | 0 ~ 39 | 0 |
| | LineSpacing | | 50 ~ 500 | 120 |
| | CharSpacing | | 50 ~ 500 | 100 |
| | Align | | Left \| Right \| Center \| Full \| Table | Left |

<그림 3-102> TEXTARTSHAPE 엘리먼트(HWPML)

| OUTLINEDATA | | | | |
|---|---|---|---|---|
| 설명 | | 외각선 | | |
| 부모 엘리먼트 | | TEXTART | | |
| 자식 엘리먼트 | | POINT | | |
| 속성 | Count | 외각선 포인트 개수 | 0 이상의 정수 | |

<그림 3-103> OUTLINEDATA 엘리먼트(HWPML)

### (6) 책갈피 (Tag ID : HWPTAG_CTRL_DATA)

❒ 책갈피 정의

○ 책갈피 컨트롤은 책갈피로서 갖는 정보로서 “책갈피 이름” 밖에 없으며, 컨트롤

의 임의의 데이터인 HWP_CTRL_DATA로 레코드 됨.

○ 기술 문서를 기반으로 뷰어를 개발하기 위해서는 기술정보은행(DFR)에 교정부호 렌더링을 위한 표현 정보도 포함되어야 함

❒ 책갈피 관련 속성 정의서

○ 책갈피 기능성의 파라미터 셋, 파라미터 아이템 및 아이템 종류에 대한 속성 정의는 다음과 같음

| 자료형 | 길이(바이트) | 설명 |
|---|---|---|
| Parameter Set | 가변 | 파라미터 셋(표 45 참조) |
| 전체 길이 | 가변 | |

<그림 3-104> 파라미터 셋

| 자료형 | 길이(바이트) | 설명 |
|---|---|---|
| WORD | 2 | 파라미터 아이템 ID |
| WORD | 2 | 파라미터 아이템 종류(표 47 참조) |
| Parameter Item Type | 가변 | 파라미터 아이템 데이터 |
| 전체 길이 | 가변 | 4 + 가변 바이트 |

<그림 3-105> 파라미터 아이템

| 값 | 구분 | 자료형 | 설명 |
|---|---|---|---|
| 0 | PIT_NULL | UINT | NULL |
| 1 | PIT_BSTR | WORD | 문자열 길이(slen) |
| | | WCHAR array[len] | 문자열 |
| 2 | PIT_I1 | UINT | INT8 |
| 3 | PIT_I2 | UINT | INT16 |
| 4 | PIT_I4 | UINT | INT32 |
| 5 | PIT_I | UINT | INT |
| 6 | PIT_UI1 | UINT | UINT8 |
| 7 | PIT_UI2 | UINT | UINT16 |
| 8 | PIT_UI4 | UINT | UINT32 |
| 9 | PIT_UI | UINT | UINT |
| 0x8000 | PIT_SET | Parameter Set | 파라미터 셋 |
| 0x8001 | PIT_ARRAY | INT16 | 파라미터 셋 개수 |
| | | ParameterArray | 파라미터 셋 배열 |
| 0x8002 | PIT_BINDATA | UINT16 | 바이너리 데이터 ID |

<그림 3-106> 파라미터 아이템 종류

### (7) 스크립트 (Tag_ID : HWPTAG_EQEDIT)

❒ 스크립트 속성 정의

○ 스크립트 개체 속성 정의는 다음과 같음

| 자료형 | 길이(바이트) | 설명 |
|---|---|---|
| UINT32 | 4 | 속성<br>스크립트가 차지하는 범위.<br>첫 비트가 켜져 있으면 줄 단위, 꺼져 있으면 글자 단위. |
| WORD | 2 | 스크립트 길이(len) |
| WCHAR array[len] | 2×len | 스크립트 |
| HWPUNIT | 4 | 수식 글자 크기 |
| COLORREF | 4 | 글자 색상 |
| INT16 | 2 | base line |
| 전체 길이 | 가변 | 16 + (2×len) 바이트 |

<그림 3-107> 스크립트 개체 속성

## 다. 파워포인트 포맷에서 소실되는 기능성의 속성 구조[92)]

### (1) 삽입된 비디오 파일

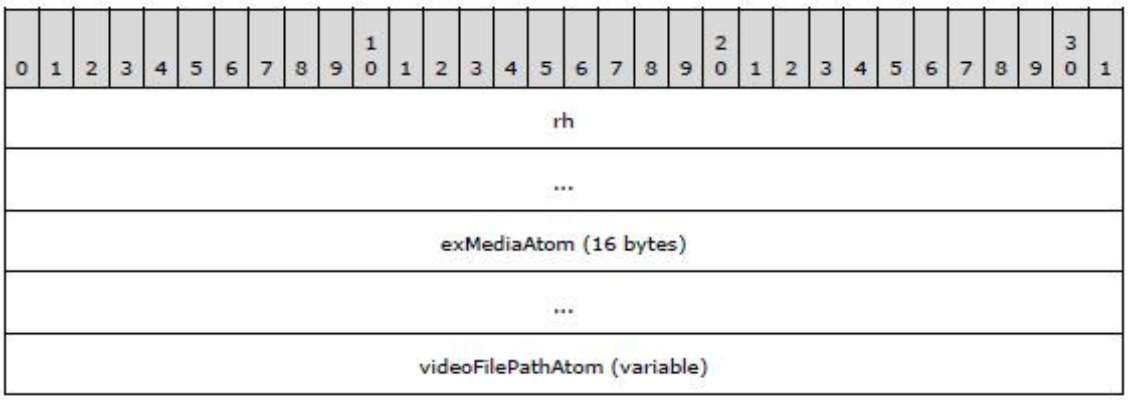
❒ 삽입된 비디오 파일 정의

○ 상위 레코드인 DocumentContainer, DocumentAtom 컨테이너에서 화면의 위치, 크기와 같은 정보를 가지고 있음.

○ 삽입된 비디오 파일의 정보를 담고 있는 정보는 다음과 같음

❒ 비디오 파일 관련 컨테이너 정의서

○ 비디오 파일을 저장하는 컨테이너는 ExVideoContainer, ExMediaAtom, VideoFilePathAtom가 존재하며 각각의 컨테이너 정의는 다음과 같음

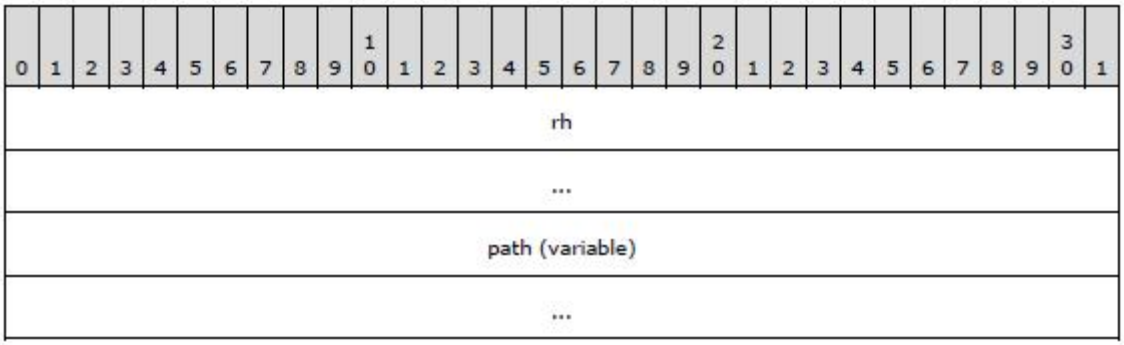



<그림 3-108> ExVideoContainer

92) *[MS-PPT] — v20130726 PowerPoint (.ppt) Binary File Format Copyright © 2013 Microsoft Corporation Release: July 30, 2013*




<그림 3-109> ExMediaAtom




<그림 3-110> VideoFilePathAtom

❒ 비디오파일 관련 속성 정의

○ ExVideoContainer의 속성 정보

- rh (8 bytes): A RecordHeader structure that specifies the header for this record. Sub-fields are further specified in the following table

<표 3-31> Record Header

| Field | Meaning |
|---|---|
| rh.recVer | MUST be 0xF. |
| rh.recInstance | MUST be 0x000. |
| rh.recType | MUST be an RT_ExternalVideo. |

- exMediaAtom (16 bytes): An ExMediaAtom record (section 2.10.6) that specifies information about the external video
- videoFilePathAtom (variable): An optional UncOrLocalPathAtom record that specifies the UNC or local path to a video file. The length, in bytes, of the field is specified by the following formula:
  rh.recLen - 16

○ ExMediaAtom의 속성 정보

- rh (8 bytes): A RecordHeader structure that specifies the header for this record. Sub-fields are further specified in the following table

<표 3-32> Record Header

| Field | Meaning |
|---|---|
| rh.recVer | MUST be 0x0. |
| rh.recInstance | MUST be 0x000. |
| rh.recType | MUST be an RT_ExternalMediaAtom. |
| rh.recLen | MUST be 0x00000008. |

- exObjId (4 bytes): An ExObjId that specifies the identifier for the audio or video data
- A - fLoop (1 bit): A bit that specifies whether the audio or video data is repeated continuously during playback
- B - fRewind (1 bit): A bit that specifies whether the audio or video data is rewound after playing
- C - fNarration (1 bit): A bit that specifies whether the audio data is recorded narration for the slide show. It MUST be FALSE if this ExMediaAtom record is contained by an ExVideoContainer record
- reserved (13 bits): MUST be zero and MUST be ignored
- unused (2 bytes): Undefined and MUST be ignored

○ VideoFilePathAtom의 속성 정보

- rh (8 bytes): A RecordHeader structure (section 2.3.1) that specifies the header for this record. Sub-fields are further specified in the following

<표 3-33> Record Header

| Field | Meaning |
|---|---|
| rh.recVer | MUST be 0x0. |
| rh.recInstance | MUST be 0x000. |
| rh.recType | MUST be an RT_CString. |
| rh.recLen | MUST be an even numbr. |

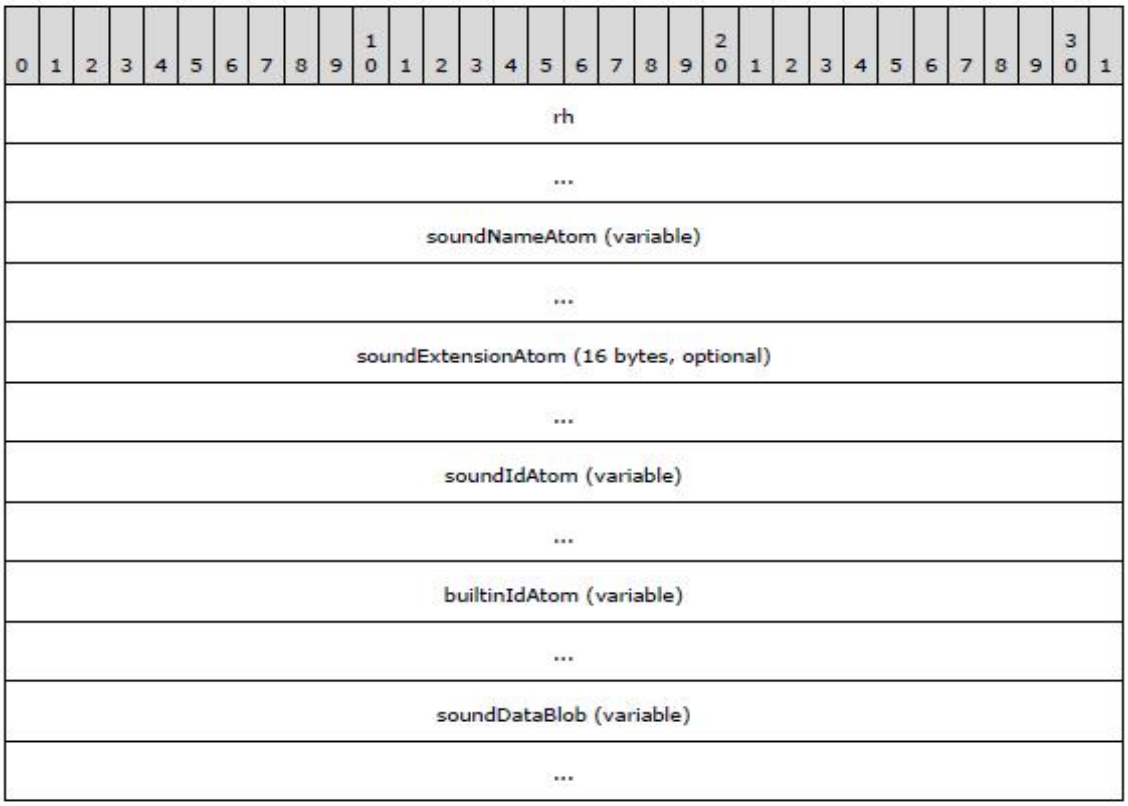
- table

- path (variable): A UncOrLocalPath that specifies the UNC or local path to a file. The length, in bytes, of the field is specified by rh.recLen

### (2) 삽입된 오디오파일

❒ 삽입된 오디오 파일 정의
- ○ 오디오의 이름, 포맷 형태, 컨테이너 ID, 실제적인 데이터가 들어가 있는 BLOB에 대한 정보를 가지고 있음
- ○ 사용자에게 보여지기 위한 위치, 크기와 같은 정보는 DocumentContainer에 저장되어 있음

❒ 오디오 파일 관련 컨테이너 정의서
- ○ 오디오에 대한 정보를 담고 있는 SoundContainer는 다음과 같이 정의됨




<그림 3-111> SoundContainer

❒ 오디오파일 속성 정의
- ○ Sound와 관련된 속성 정보

- rh (8 bytes): A RecordHeader structure that specifies the header for this record. Sub-fields are further specified in the following table

<표 3-34> Record Header

| Field | Meaning |
|---|---|
| rh.recVer | MUST be 0xF. |
| rh.recInstance | MUST be 0x000. |
| rh.recType | MUST be RT_Sound. |

- soundNameAtom (variable): A SoundNameAtom record that specifies the name of the sound.
- soundExtensionAtom (16 bytes): An optional SoundExtensionAtom record that specifies the format of the audio data.
- soundIdAtom (variable): A SoundIdAtom record that specifies the sound identifier for the sound.
- builtinIdAtom (variable): An optional SoundBuiltinIdAtom record that specifies an identifier that describes the sound.
- soundDataBlob (variable): A SoundDataBlob record that specifies the audio data for the sound.

### (3) OLE 개체

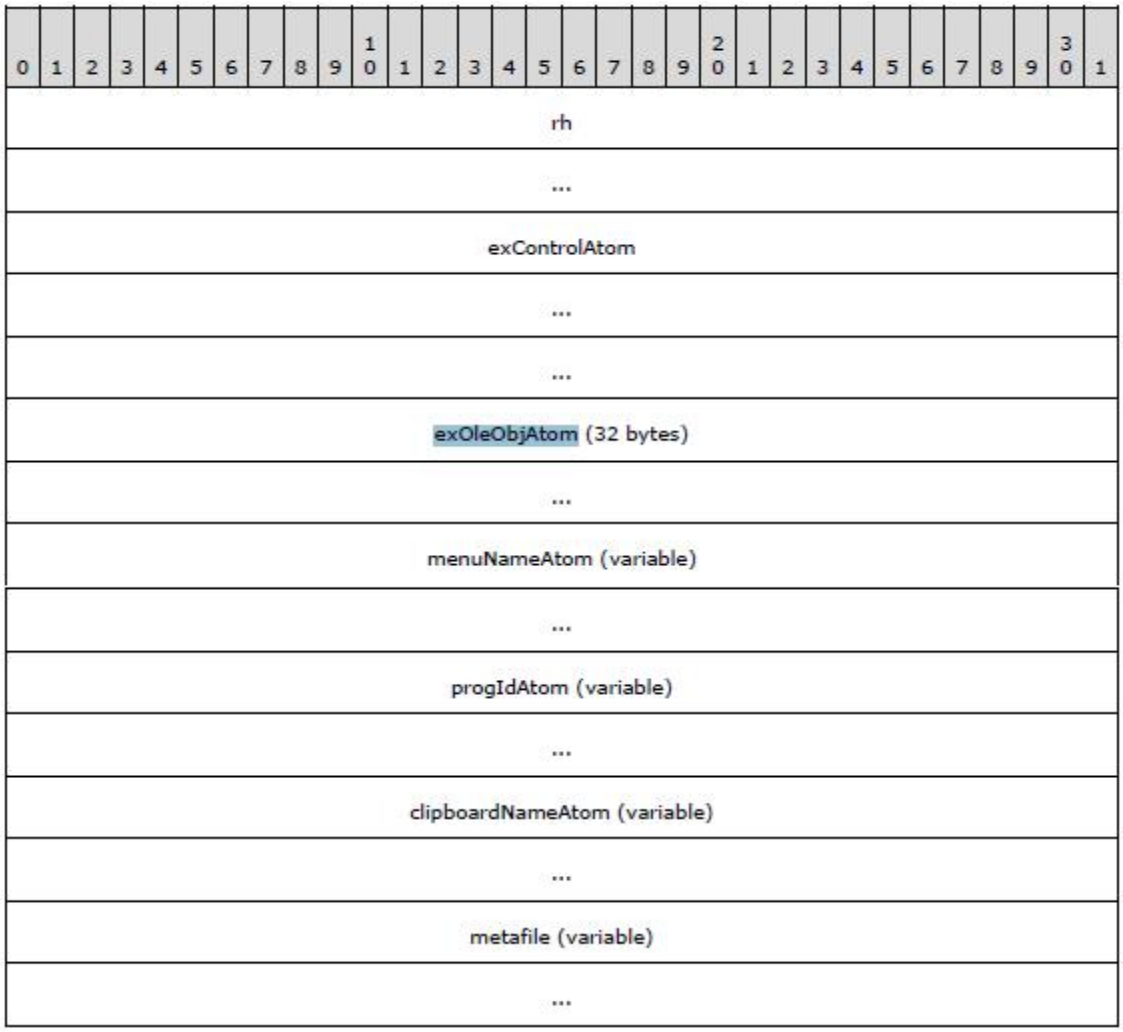
❒ OLE 개체 정의

○ OLE 개체와 관련된 외부 데이터는 ExObjListContainr, ExObjListSubContainer를 통해 데이터를 저장함

○ OLE 개체를 화면에 표현하기 위한 정보와 정보가 들어 있는 위치를 가리키는 ID 정보는 ExOleObjAtom에 저장됨

❒ OLE 개체 관련 컨테이너 정의서

○ OLE 정보를 담고 있는 ExControlContainer는 다음과 같이 정의됨




<그림 3-112> ExControlContainer

❒ OLE 개체 속성 정의

○ OLE 개체의 exControl에 대한 속성 정의는 다음과 같음

- rh (8 bytes): A RecordHeader structure that specifies the header for this record. Sub-fields are further specified in the following table

<표 3-35> Record Header

| Field | Meaning |
|---|---|
| rh.recVer | MUST be 0xF. |
| rh.recInstance | MUST be 0x000. |
| rh.recType | MUST be an RT_ExternalOleControl. |

- exControlAtom (12 bytes): An ExControlAtom record that specifies the

identifying information for this ActiveX control.

- exOleObjAtom (32 bytes): An ExOleObjAtom record that specifies information about this ActiveX control as an OLE object.
- menuNameAtom (variable): An optional MenuNameAtom record that specifies the name to use in the user interface.
- progIdAtom (variable): An optional ProgIDAtom record that specifies the ProgID (described in [MSDN-COM]) of the ActiveX control.
- clipboardNameAtom (variable): An optional ClipboardNameAtom record that specifies the name used by the user interface during copy and paste operations.
- metafile (variable): An optional MetafileBlob record that specifies the icon for the ActiveX control

### (4) 애니메이션

❒ 애니메이션 정의

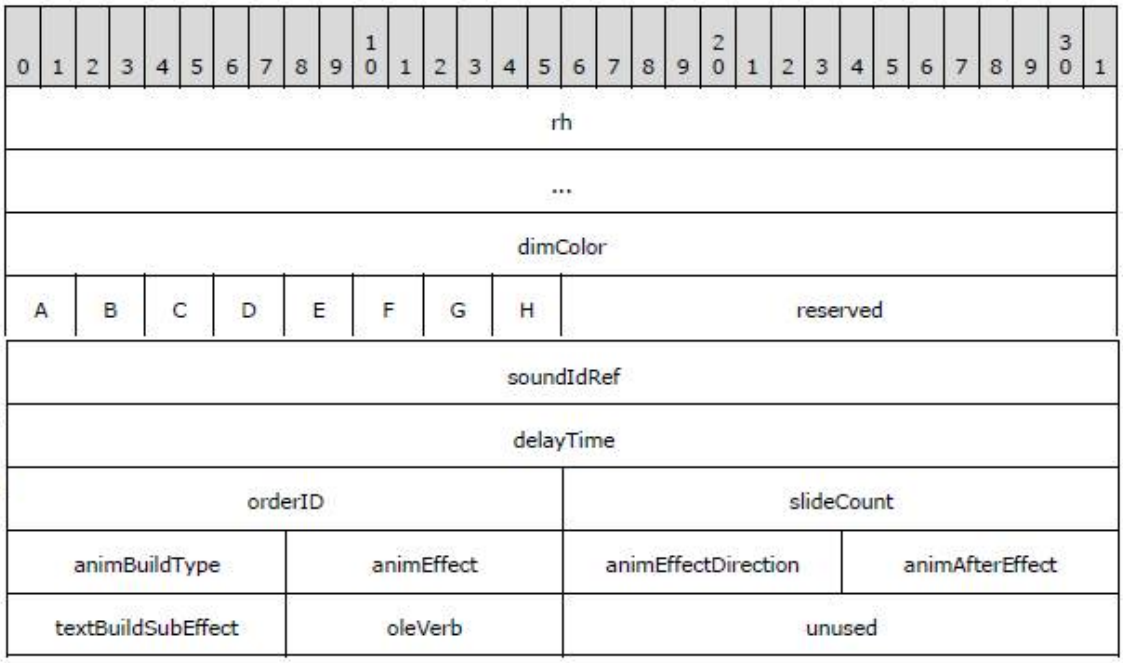
○ 애니메이션 정보는 AnimationInfoContainer에 저장하고 있고, 화면에 보여주기 위한 정보는 AnimationInfoAtom에 저장되어 있음

○ 사운드에 대한 정보, 딜레이 타임, 자동 넘김 기능 화면형태 그림형태로 잘 구성되어 있음

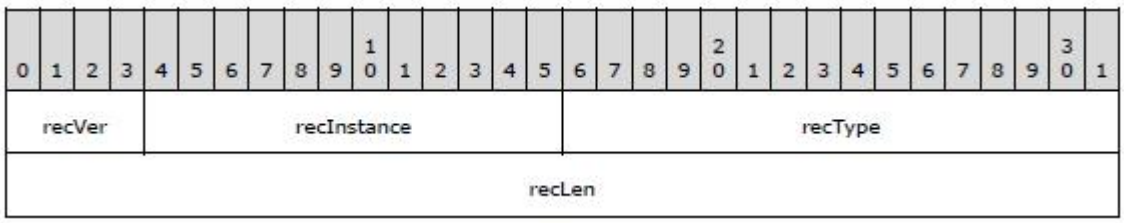
❒ 애니메이션 관련 컨테이너 정의서

○ 애니메이션을 위한 컨테이너는 AnimationInfoAtom 컨테이너와 레코드 헤더에 대한 정의는 다음과 같음

○ RecordHeader 구조에서 Record Type이 RT_AnimationInfoAtom = 0x0FF1(애니메이션) 임




<그림 3-113> AnimationInfoAtom




<그림 3-114> Record Header(rh)

❒ 애니메이션 속성 정의

○ 애니메이션에 대한 속성 정의는 다음과 같음

- rh (8 bytes): A RecordHeader structure (section 2.3.1) that specifies the header for this record. Sub-fields are further specified in the following table

<표 3-36> Record Header

| Field | Meaning |
|---|---|
| rh.recVer | MUST be 0x1. |
| rh.recInstance | MUST be 0x000. |
| rh.recType | MUST be an RT_AnimationInfoAtom. |
| rh.recLen | MUST be 0x0000001C |

- dimColor (4 bytes): A ColorIndexStruct structure that specifies a color for the dim effect after the animation is complete.

○ 애니메이션 타입에 대한 속성 정의는 다음과 같음[93)]

- 애니메이션 타입에 대한 자세한 속성 정의는 “PowerPoint (.ppt) Binary File Format v20130726 page227-235” 참조
- A - fReverse (2 bits): An unsigned integer that specifies whether the animation plays in the reverse direction. It MUST be a value from the following table
- B - fAutomatic (2 bits): An unsigned integer that specifies whether the animation starts automatically. It MUST be a value from the following table
- C - fSound (2 bits): An unsigned integer that specifies whether the animation has an associated sound. It MUST be a value from the following table
- D - fStopSound (2 bits): An unsigned integer that specifies whether all playing sounds are stopped when this animation begins. It MUST be a value from the following table
- E - fPlay (2 bits): An unsigned integer that specifies whether an associated sound, media or action verb is activated when the shape is clicked. It MUST be a value from the following table
- F - fSynchronous (2 bits): An unsigned integer that specifies that the animation, while playing, stops other slide show actions. If the shape is a media or OLE object, this field is valid; otherwise, it MUST be ignored. It MUST be a value from the following table
- G - fHide (2 bits): An unsigned integer that specifies whether the shape is hidden while the animation is not playing. If the shape is a media or OLE object, this field is valid; otherwise, it MUST be ignored. It MUST be a value from the following table
- H - fAnimateBg (2 bits): An unsigned integer that specifies whether the background of the shape is animated. It MUST be a value from the following table
- reserved (16 bits): MUST be zero, and MUST be ignored
- soundIdRef (4 bytes): A SoundIdRef that specifies the value to refer to in the SoundCollectionContainer record (section 2.4.16.1) to locate the

93) PowerPoint (.ppt) Binary File Format v20130726 page227-235 참조

embedded audio

- delayTime (4 bytes): A signed integer that specifies the delay time, in milliseconds, before the animation starts to play. If fAutomatic is 0x1, this value MUST be greater than or equal to 0; otherwise, this field MUST be ignored
- orderID (2 bytes): A signed integer that specifies the order of the animation in the slide. It MUST be greater than or equal to -2. The value -2 specifies that this animation follows the order of the corresponding placeholder shape on the main master slide or title master slide. The value -1 SHOULD NOT<104> be used
- slideCount (2 bytes): An unsigned integer that specifies the number of slides that this animation continues playing. This field is utilized only in conjunction with media. The value 0xFFFFFFFF specifies that the animation plays for one slide
- animBuildType (1 byte): An AnimBuildTypeEnum enumeration that specifies the animation build type for the animation effect
- animEffect (1 byte): An unsigned integer that specifies the animation effect type for the shape. The following diagrams are for example purposes only. Exact rendering of any animation effect is determined by the rendering application. As such, the same animation effect can have many variations depending on the implementation

## 라. 엑셀 포맷에서 소실되는 기능성의 속성 구조[94)]

### (1) 스파크라인

❒ 스파크라인 정의

- ○ 스파크라인은 XLSX 문서에서만 존재하는 기능
- ○ CT_SparklineGroups, CT_SparklineGroup, CT_Sparklines, CT_Sparkline 엘리먼트로 구성되어 있음
- ○ 최대값, 최소값, Color와 같은 것은 비교적 잘 나타나 있음
- ○ 그래프의 형태로는 “꺽은선형, 열, 승패”로 구성되어 있지만, 문서에서는 형태를 정하는 속성에 대해서는 표현되지 않음

94) [MS-XLS] – v20130726 Excel Binary File Format (.xls) Structure Copyright © 2013 Microsoft Corporation. Release: July 30, 2013

❒ 스파크라인 관련 컨테이너 정의서

○ 스파크라인은 XLSX 포맷만을 지원하여 컨테이너정의서 대신 XML Schema로 정의됨

```
<xsd:complexType name="CT_SparklineGroup">
  <xsd:sequence>
    <xsd:element name="colorSeries" minOccurs="0" maxOccurs="1" type="x:CT_Color"/>
    <xsd:element name="colorNegative" minOccurs="0" maxOccurs="1" type="x:CT_Color"/>
    <xsd:element name="colorAxis" minOccurs="0" maxOccurs="1" type="x:CT_Color"/>
    <xsd:element name="colorMarkers" minOccurs="0" maxOccurs="1" type="x:CT_Color"/>
    <xsd:element name="colorFirst" minOccurs="0" maxOccurs="1" type="x:CT_Color"/>
    <xsd:element name="colorLast" minOccurs="0" maxOccurs="1" type="x:CT_Color"/>
    <xsd:element name="colorHigh" minOccurs="0" maxOccurs="1" type="x:CT_Color"/>
    <xsd:element name="colorLow" minOccurs="0" maxOccurs="1" type="x:CT_Color"/>
    <xsd:element ref="xm:f" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="sparklines" type="CT_Sparklines" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="manualMax" type="xsd:double" use="optional"/>
  <xsd:attribute name="manualMin" type="xsd:double" use="optional"/>
  <xsd:attribute name="lineWeight" type="xsd:double" use="optional" default="0.75"/>
  <xsd:attribute name="type" type="ST_SparklineType" use="optional" default="line"/>
  <xsd:attribute name="dateAxis" type="xsd:boolean" use="optional" default="false"/>
  <xsd:attribute name="displayEmptyCellsAs" type="ST_DispBlanksAs" use="optional"
default="zero"/>
  <xsd:attribute name="markers" type="xsd:boolean" use="optional" default="false"/>
  <xsd:attribute name="high" type="xsd:boolean" use="optional" default="false"/>
  <xsd:attribute name="low" type="xsd:boolean" use="optional" default="false"/>
  <xsd:attribute name="first" type="xsd:boolean" use="optional" default="false"/>
  <xsd:attribute name="last" type="xsd:boolean" use="optional" default="false"/>
  <xsd:attribute name="negative" type="xsd:boolean" use="optional" default="false"/>
  <xsd:attribute name="displayXAxis" type="xsd:boolean" use="optional" default="false"/>
  <xsd:attribute name="displayHidden" type="xsd:boolean" use="optional" default="false"/>
  <xsd:attribute name="minAxisType" type="ST_SparklineAxisMinMax" use="optional"
default="individual"/>
  <xsd:attribute name="maxAxisType" type="ST_SparklineAxisMinMax" use="optional"
default="individual"/>
  <xsd:attribute name="rightToLeft" type="xsd:boolean" use="optional" default="false"/>
</xsd:complexType>
```

<그림 3-115> CT_SparklineGroup

❒ 스파크라인 속성 정의[95]

○ 스파크라인과 관련된 엘리먼트 정의는 다음과 같음

- 스파크라인에 대한 자세한 엘리먼트 정의는 “Excel (.xlsx) Extensions to the Office Open XML SpreadsheetML File Format v20130726 apge 103-106” 참조
- colorSeries: A CT_Color element that specifies the color for each sparkline in this sparkline group. The auto attribute of the CT_Color element MUST NOT exist
- colorNegative: A CT_Color element that specifies the color of the negative data points (2) for each sparkline in this sparkline group. The auto attribute of the CT_Color element MUST NOT exist
- colorAxis: A CT_Color element that specifies the color of the horizontal

95) Excel (.xlsx) Extensions to the Office Open XML SpreadsheetML File Format *v20130726 apge 103-106 참조*

axis for each sparkline in this sparkline group. The auto attribute of the CT_Color element MUST NOT exist

- colorMarkers: A CT_Color element that specifies the color of the data markers for each sparkline in this sparkline group. The auto attribute of the CT_Color element MUST NOT exist
- colorFirst: A CT_Color element that specifies the color of the first data point for each sparkline in this sparkline group. The auto attribute of the CT_Color element MUST NOT exist
- colorLast: A CT_Color element that specifies the color of the last data point for each sparkline in this sparkline group. The auto attribute of the CT_Color element MUST NOT exist
- colorHigh: A CT_Color element that specifies the color of the highest data point for each sparkline in this sparkline group. The auto attribute of the CT_Color element MUST NOT exist
- colorLow: A CT_Color element that specifies the color of the lowest data point for each sparkline in this sparkline group. The auto attribute of the CT_Color element MUST NOT exist
- xm:f: An f element that specifies the date range for the sparkline group. The syntax, as specified in section 2.2.2, for this f element is the following:
- sparklinegroup-formula = single-sheet-area / [single-sheet-prefix / book-prefix] name
- Additionally, if a single-sheet-area is specified, that single-sheet-area MUST contain cells from either a single row (2) or a single column (2).
- sparklines: A CT_Sparklines element that specifies properties for individual sparklines
- Attributes:
- manualMax: A double attribute that specifies the maximum for the vertical axis that is shared across all sparklines in this sparkline group. This attribute MUST NOT exist if maxAxisType does not equal "custom"
- manualMin: A double attribute that specifies the minimum for the vertical axis that is shared across all sparklines in this sparkline group. This attribute MUST NOT exist if minAxisType does not equal "custom"
- lineWeight: A double attribute that specifies the line weight for each sparkline in the sparkline group, where the line weight is measured in points. MUST be greater than or equal to zero, and MUST be less than

or equal to 1584

- type: An ST_SparklineType attribute that specifies the type of the sparkline group
- dateAxis: A Boolean attribute that specifies whether this sparkline group uses a date axis

### (2) 메모장

❒ 메모장 정의

○ NoteSH 레코드로 구성되어 있음

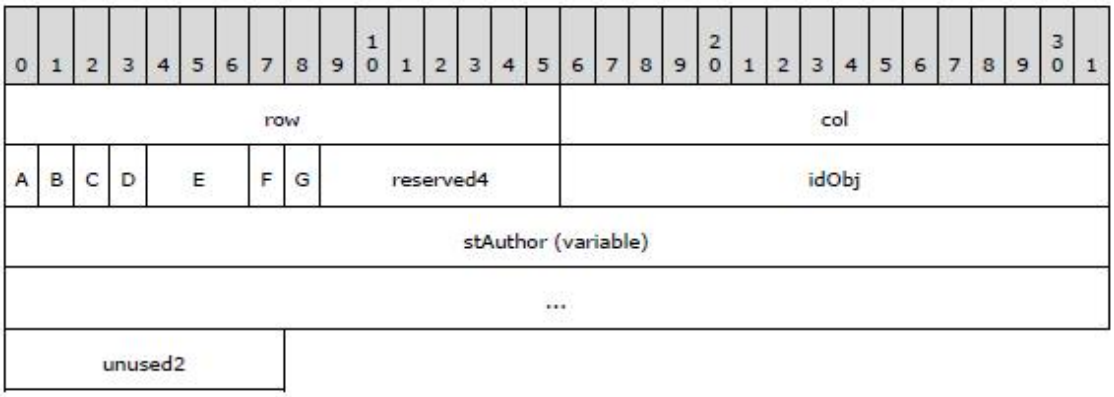
○ 셀의 위치, 코멘트의 실제 내용이 저장된 오브젝트 ID, 메모의 길이, 저자 등에 대한 정보들이 저장되어 있음

○ 저장된 내용을 위한 정보는 레코드 형태도 저장되어 있지만 화면 렌더링에 대한 요소는 정의되어 있지 않음

○ 스펙문서를 기준으로 Viewer을 만들기 위해서는 DFR 정보에 메모장 렌더링을 위한 정보도 포함하고 있어야 함

❒ 메모장 관련 컨테이너 정의서

○ 메모장에 대한 NoteSh 컨테이너는 다음과 같이 정의됨




<그림 3-116> NoteSh

❒ 메모장 속성 정의

○ 메모장과 관련된 속성 정의는 다음과 같음

- row (2 bytes): A RW that specifies the row of the cell to which this comment is associated
- col (2 bytes): A Col that specifies the column of the cell to which this comment is associated
- A - reserved1 (1 bit): MUST be zero and MUST be ignored
- B - fShow (1 bit): A bit that specifies whether the comment is shown at

all times

- C - reserved2 (1 bit): MUST be zero and MUST be ignored
- D - unused1 (1 bit): Undefined and MUST be ignored
- E - reserved3 (3 bits): MUST be zero and MUST be ignored
- F - fRwHidden (1 bit): A bit that specifies whether the row specified by row is hidden
- G - fColHidden (1 bit): A bit that specifies whether the column specified by col is hidden
- reserved4 (7 bits): MUST be zero and MUST be ignored
- idObj (2 bytes): An ObjId that specifies the Obj record that specifies the comment text
- stAuthor (variable): An XLUnicodeString that specifies the name of the comment author. String length MUST be greater than or equal to 1 and less than or equal to 54
- unused2 (1 byte): Undefined and MUST be ignored

### (3) 차트

❒ 차트정의

○ 차트는 Visual한 폼으로 데이터의 SET들 사이의 관계와 데이터를 화면에 보여주기는 그래픽 요소

○ 차트는 데이터의 SET(Char Data Cache)과 데이터의 디스플레이를 위한 Chart Sheet로 구성되어 있음

○ 차트에 대한 정보는 Chart Sheet Substream에 정보가 저장되어 있음

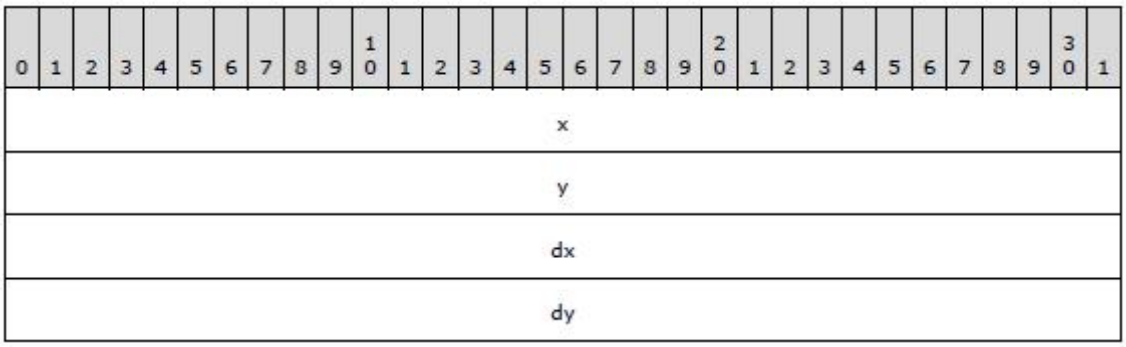
○ Font, 선행된 데이터와 연관을 나타내는 Continue, 라인 형태, 영역 형태처럼 화면에 나타내기 위한 정보를 저장하고 있음

○ 3D 차트인 경우는 Chart3D에 따로 데이터에 대한 정의를 하고 있음

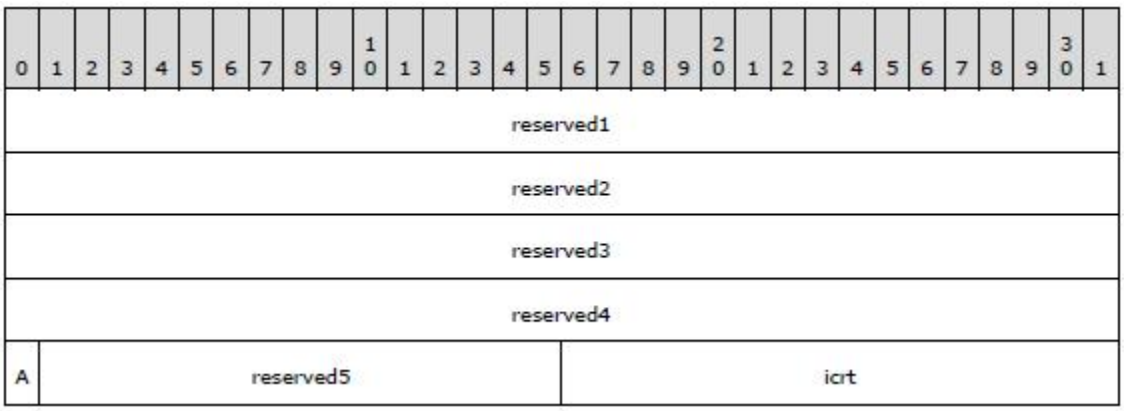
○ 셀에 있는 정보의 복사본을 사용하여 화면에 표현하기 위한 Chart 정보에 사용

❒ 차트 관련 컨테이너 정의서

○ 차트 기능성을 저장하기 위한 Chart 컨테이너와 ChartFormat 컨테이너에 대한 정의는 다음과 같음




<그림 3-117> Chart




<그림 3-118> ChartFormat

❒ 차트 속성 정의

○ 차트 컨테이너에 저장되는 x, y, dx, dy 속성은 차트 기능성을 화면에 표현하기 위해서 위치를 나타내는 x, y와 그래프의 높이와 넓이를 나타내는 dx, dy로 구성되며 다음과 같음

- x (4 bytes): A FixedPoint as specified in [MS-OSHARED] section 2.2.1.6 that specifies the horizontal position of the upper-left corner of the chart in points. SHOULD<45> be greater than or equal to zero
- y (4 bytes): A FixedPoint as specified in [MS-OSHARED] section 2.2.1.6 that specifies the vertical position of the upper-left corner of the chart in points. SHOULD<46> be greater than or equal to zero
- dx (4 bytes): A FixedPoint as specified in [MS-OSHARED] section 2.2.1.6 that specifies the width in points. SHOULD<47> be greater than or equal to zero
- dy (4 bytes): A FixedPoint as specified in [MS-OSHARED] section 2.2.1.6 that specifies the height in points. SHOULD<48> be greater than or equal to zero

○ ChartFormat에 대한 속성은 다음과 같음

- reserved1 (4 bytes): MUST be zero, and MUST be ignored
- reserved2 (4 bytes): MUST be zero, and MUST be ignored
- reserved3 (4 bytes): MUST be zero, and MUST be ignored
- reserved4 (4 bytes): MUST be zero, and MUST be ignored
- A - fVaried (1 bit): A bit that specifies whether the color for each data point and the color and type for each data marker varies. If the chart group has multiple series, or the chart group has one series and the type is either a surface, stock, or area chart group, then this field MUST be ignored, and the data points do not vary. For all other chart group types, if the chart group has one series, then a value of 0x1 specifies that the data points vary. MUST be a value from the following table:

<표 3-37> A - fVaried

| Field | Meaning |
|---|---|
| 0x0 | The color for each data point and the color and type for each data marker does not vary. |
| 0x1 | The color for data points or the color or type for data markers varies. |

- reserved5 (15 bits): MUST be zero, and MUST be ignored
- icrt (2 bytes): An unsigned integer that specifies the drawing order of the chart group relative to the other chart groups, where 0x0000 is the bottom of the z-order. This value MUST be unique for each instance of this record and MUST be less than or equal to 0x0009

### (4) 수식정보

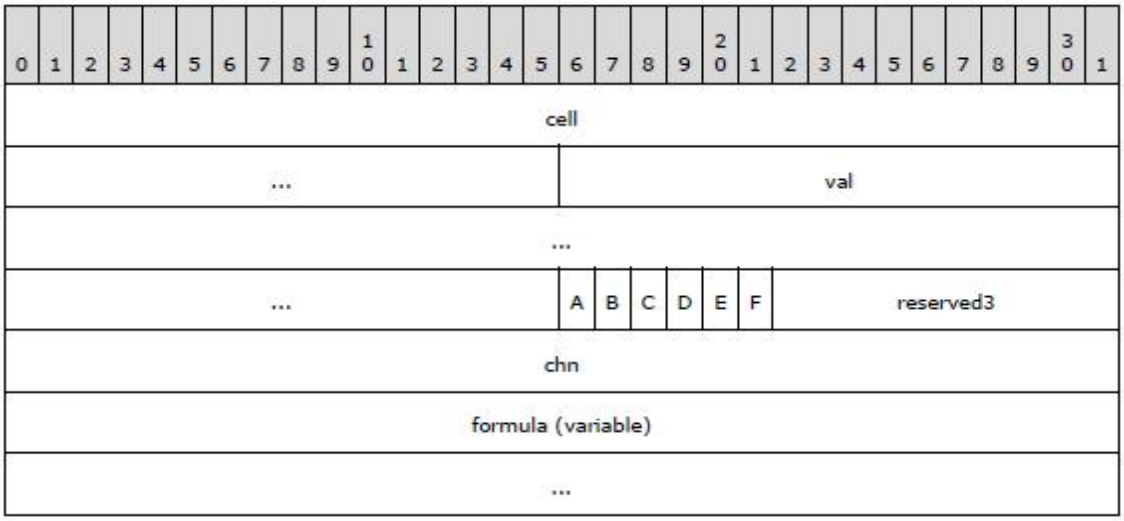
❒ 수식정보 정의

○ 셀의 값, 셀의 이름, 함수와 연산자로 셀의 새로운 값을 만들어 내는 함수로서 화면에 렌더링 하는 요소는 없지만 셀 정보에 수식이 저장되어 있고 어떠한 셀들을 이용하여 계산하는지에 대한 정보가 저장되어 있음

○ 수식 정보는 화면 렌더링을 위한 정보가 없으므로 문서보존포맷(PDF/A)으로 변환할 때 정보 자체가 사라짐

❒ 수식정보 관련 컨테이너 정의서

○ 수식정보를 저장하기 위한 Formula 컨테이너에 대한 정의는 다음과 같음




<그림 3-119> Formula

❒ 수식정보 속성 정의

○ 수식정보 저장에 사용되는 속성 정보는 다음과 같음

- cell (6 bytes): A Cell structure that specifies a cell on the sheet
- val (8 bytes): A FormulaValue structure that specifies the value of the formula
- A - fAlwaysCalc (1 bit): A bit that specifies whether the formula needs to be calculated during the next recalculation
- B - reserved1 (1 bit): MUST be zero, and MUST be ignored
- C - fFill (1 bit): A bit that specifies whether the cell has a fill alignment or a center-across-selection alignment

<표 3-38> C - fFill

| Field | Meaning |
|---|---|
| 0 | Cell does not have a fill alignment or a center-across-selection alignment. |
| 1 | Cell has either a fill alignment or a center-across-selection alignment. |

- D - fShrFmla (1 bit): A bit that specifies whether the formula is part of a shared formula as defined in ShrFmla. If this formula is part of a shared formula, formula.rgce MUST begin with a PtgExp structure
- E - reserved2 (1 bit): MUST be zero, and MUST be ignored
- F - fClearErrors (1 bit): A bit that specifies whether the formula is

excluded from formula error checking

- reserved3 (10 bits): MUST be zero, and MUST be ignored
- chn (4 bytes): A field that specifies an application-specific cache of information. This cache exists for performance reasons only, and can be rebuilt based on information stored elsewhere in the file without affecting calculation results
- formula (variable): A CellParsedFormula structure that specifies the formula

### (5) 접힌 셀

❒ 접힌 셀 정의

○ 셀은 텍스트, 숫자 데이터, 수식 등의 정보를 저장하는 기본 단위

○ 셀이 접혀 있는 경우 문서보존포맷(PDF/A)으로 저장할 때 접힌 상태로 변환이 되기 때문에 정보의 손실 위험이 존재

○ 문서보존포맷(PDF/A)으로 변환할 때 접힌 셀의 경우 셀에 들어가 있는 정보가 보여지도록 셀을 펼친 상태에서 변환 필요

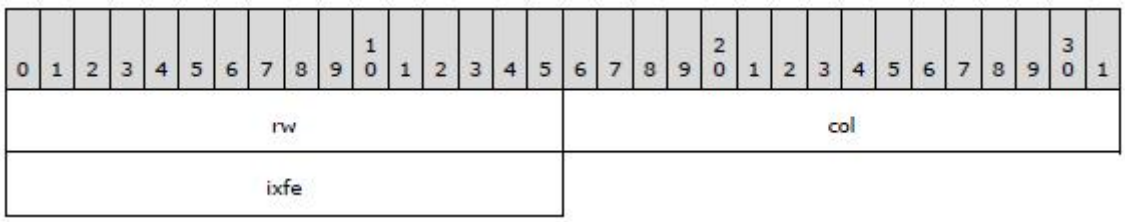
○ 특정 솔루션을 사용하여 문서를 변환할 때, 셀을 펼친 상태에서 변환이 가능한지에 대한 기능 점검 필요

○ 셀 구조에는 위치(Row, Column) 정보가 저장됨

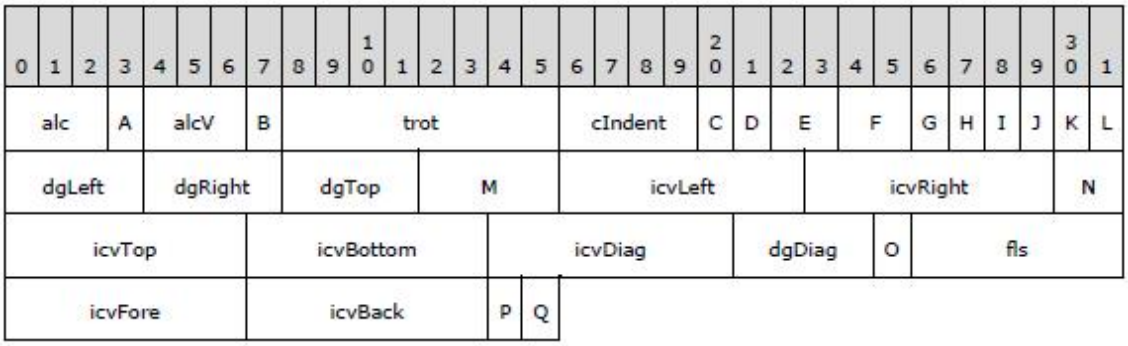
○ CellXF에 화면에 표현을 위한 정보(정렬, 경계선(상/하/좌/우), 색깔 등)가 들어가 있음

❒ 셀 관련 컨테이너 정의서

○ 셀 구조에 대한 Cell 컨테이너와 CellXF 컨테이너에 대한 정의는 다음과 같음




<그림 3-120> Cell




<그림 3-121> CellXF

❒ 셀 속성 정의

○ Cell에 대한 속성은 다음과 같음

- rw (2 bytes): An Rw that specifies the row
- col (2 bytes): A Col that specifies the column
- ixfe (2 bytes): An IXFCell that specifies the XF record

○ CellXF에 대한 속성은 다음과 같음[96]

- CellFX에 대한 자세한 속성 정의는 “Excel (.xlsx) Extensions to the Office Open XML SpreadsheetML File Format v20130726 apge 622-625” 참조
- alc (3 bits): A HorizAlign that specifies the horizontal alignment
- A - fWrap (1 bit): A bit that specifies whether the cell text is wrapped
- alcV (3 bits): A VertAlign that specifies the vertical alignment
- B - fJustLast (1 bit): A bit that specifies whether the justified or distributed alignment of the cell is used on the last line of text (setting this to 1 is typical for East Asian text but not typical in other contexts). If this field equals 1, then alc MUST equal 7
- trot (1 byte): An XFPropTextRotation that specifies the text rotation
- cIndent (4 bits): An unsigned integer that specifies the text indentation level. MUST be less than or equal to 15
- C - fShrinkToFit (1 bit): A bit that specifies whether the cell is shrink to fit
- D - reserved1 (1 bit): MUST be 0, and MUST be ignored
- E - iReadOrder (2 bits): A ReadingOrder that specifies the reading order
- F - reserved2 (2 bits): MUST be 0, and MUST be ignored
- G - fAtrNum (1 bit): A bit that specifies that if the ifmt field of the XF record specified by the ixfParent field of the containing XF record is

96) *Excel Binary File Format (.xls) Structure v20130726 page 622-625 참조*

updated, the corresponding field of the containing XF record will not be set to the same value

- H – fAtrFnt (1 bit): A bit that specifies that if the ifnt field of the XF record specified by the ixfParent field of the containing XF record is updated, the corresponding field of the containing XF record will not be set to the same value
- I – fAtrAlc (1 bit): A bit that specifies that if the alc field, or the fWrap field, or the alcV field, or the fJustLast field, or the trot field, or the cIndent field, or the fShrinkToFit field or the iReadOrder field of the XF record specified by the ixfParent field of the containing XF record is updated, the corresponding fields of this structure will not be set to the same values
- J – fAtrBdr (1 bit): A bit that specifies that if the dgLeft field, or the dgRight field, or the dgTop field, or the dgBottom field, or the dgDiag field, or the icvLeft field, or the icvRight field, or the grbitDiag field, or the icvTop field, or the icvBottom field, or the icvDiag field of the XF record specified by the ixfParent field of the containing XF record is updated, the corresponding fields of this structure will not be set to the same values
- K – fAtrPat (1 bit): A bit that specifies that if the fls field, the icvFore field, or the icvBack field of the XF record specified by the ixfParent field of the containing XF record is updated, the corresponding fields of this structure will not be set to the same values
- L – fAtrProt (1 bit): A bit that specifies that if the fLocked field or the fHidden field of the XF record specified by the ixfParent field of the containing XF record is updated, the corresponding fields of the containing XF record will not be set to the same values
- dgLeft (4 bits): A BorderStyle that specifies the logical left border formatting
- dgRight (4 bits): A BorderStyle that specifies the logical right border formatting
- dgTop (4 bits): A BorderStyle that specifies the top border formatting
- M – dgBottom (4 bits): A BorderStyle that specifies the bottom border formatting
- icvLeft (7 bits): An unsigned integer that specifies the color of the logical left border. The value MUST be one of the values specified in IcvXF or

0. A value of 0 means the logical left border color has not been specified. If this value is 0, then dgLeft MUST also be 0

- icvRight (7 bits): An unsigned integer that specifies the color of the logical right border. The value MUST be one of the values specified in IcvXF or 0. A value of 0 means the logical right border color has not been specified. If this value is 0, then dgRight MUST also be 0
- N - grbitDiag (2 bits): An unsigned integer that specifies which diagonal borders are present (if any)
- icvTop (7 bits): An unsigned integer that specifies the color of the top border. The value MUST be one of the values specified in IcvXF or 0. A value of 0 means the top border color has not been specified. If this value is 0, then dgTop MUST also be 0
- icvBottom (7 bits): An unsigned integer that specifies the color of the bottom border. The value MUST be one of the values specified in IcvXF or 0. A value of 0 means the bottom border color has not been specified. If this value is 0 then dgBottom MUST also be 0
- icvDiag (7 bits): An unsigned integer that specifies the color of the diagonal border. The value MUST be one of the values specified in IcvXF or 0. A value of 0 means the diagonal border color has not been specified. If this value is 0 then dgDiag MUST also be 0
- dgDiag (4 bits): A BorderStyle that specifies the diagonal border formatting
- O - fHasXFExt (1 bit): A bit that specifies whether an XFExt will extend the information in this XF
- fls (6 bits): A FillPattern that specifies the fill pattern. If this value is 1, which specifies a solid fill pattern, then only icvFore is rendered
- icvFore (7 bits): An IcvXF that specifies the foreground color of the fill pattern
- icvBack (7 bits): An unsigned integer that specifies the background color of the fill pattern. The value SHOULD<145> be an IcvXF value
- P - fsxButton (1 bit): A bit that specifies whether the XF record is attached to a pivot field drop-down button
- Q - reserved3 (1 bit): MUST be 0 and MUST be ignored

## 마. 렌더링 기능성에 대한 비트스트림 분석

### (1) 비트스트림 분석 개요

❒ 디지털 컴포넌트의 인쇄 렌더링에 의해 소실되는 기능성의 존재 유무를 파악하기 위해 기술 문서에서 제시한 기능성의 태그 정보를 디지털 컴포넌트의 비트스트림에서 직접 추출하여 테이블 정의서 또는 컨테이너 정의서의 속성 정보와 비교 분석하고자 함

❒ 디지털 컴포넌트에 포함된 기능성의 존재 유무를 확인하기 위해 비트스트림에서 직접 기능성을 추출/분석하는 연구를 통해 향후 디지털 컴포넌트에 대한 Characterization 연구 방안을 제시하고자 함

### (2) 비트스트림 분석 결과

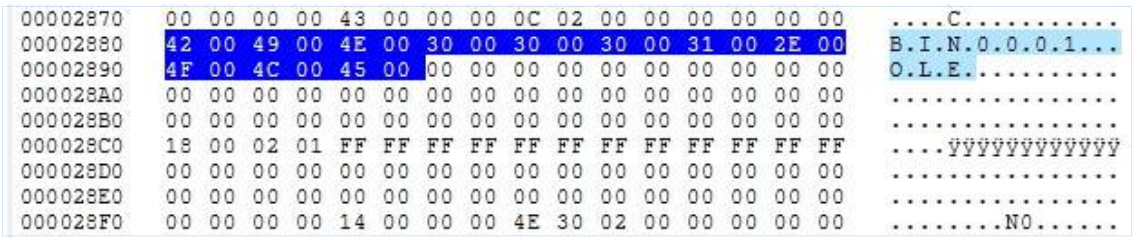
❒ 한/글 디지털 컴포넌트의 OLE 개체 추출

○ OLE 개체를 표현한 바이너리 데이터가 저장되는 바이너리 테이블 구조는 다음과 같음

| 자료형 | 길이(바이트) | 설명 | |
|---|---|---|---|
| 그림, OLE 등의 바이너리 데이터 아이템에 대한 정보 | | | |
| UINT16 | 2 | 속성(표 13 참조) | |
| WORD | 2 | Type이 "LINK"일 때, 연결 파일의 절대 경로 길이 (len1) | |
| WCHAR array[len1] | 2×len1 | Type이 "LINK"일 때, 연결 파일의 절대 경로 | |
| WORD | 2 | Type이 "LINK"일 때, 연결 파일의 상대 경로 길이 (len2) | |
| WCHAR array[len2] | 2×len2 | Type이 "LINK"일 때, 연결 파일의 상대 경로 | |
| UINT16 | 2 | Type이 "EMBEDDING"이거나 "STORAGE"일 때, BINDATASTORAGE에 저장된 바이너리 데이터의 아이디 | |
| WORD | 2 | Type이 "EMBEDDING"일 때, 바이너리 데이터의 형식 이름의 길이 (len3) | |
| WCHAR array[len3] | 2×len3 | Type이 "EMBEDDING"일 때 extension("." 제외) | |
| | | 그림의 경우 | jpg |
| | | | bmp |
| | | | gif |
| | | OLE의 경우 | ole |
| 전체 길이 | 가변 | 10 + (2×len1) + (2×len2) + (2×len3) 바이트 | |

<그림 3-122> 바이너리 데이터 테이블

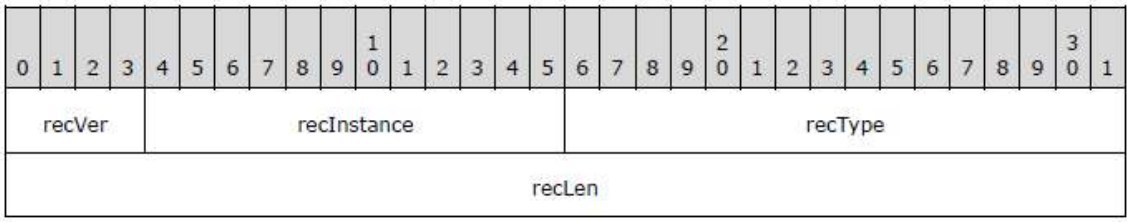
○ 아래 그림은 OLE 개체를 포함한 비트스트림 데이터이며 Type은 “OLE”이고 6바이트인 “0x4F 0x00 0x4C 0x00 0x45 0x00”로 저장




<그림 3-123> OLE 개체 실제 데이타

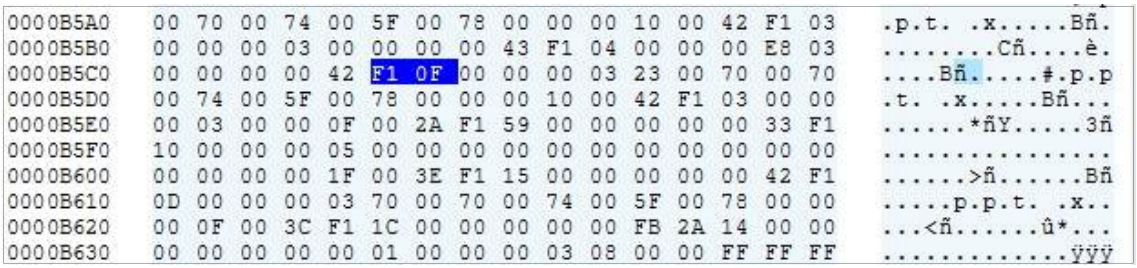
❒ 파워포인트 디지털 컴포넌트의 애니메이션 기능성 추출

○ PPT의 Record Header 컨테이너는 다음과 같이 정의됨




<그림 3-124> RecordHeader 구조

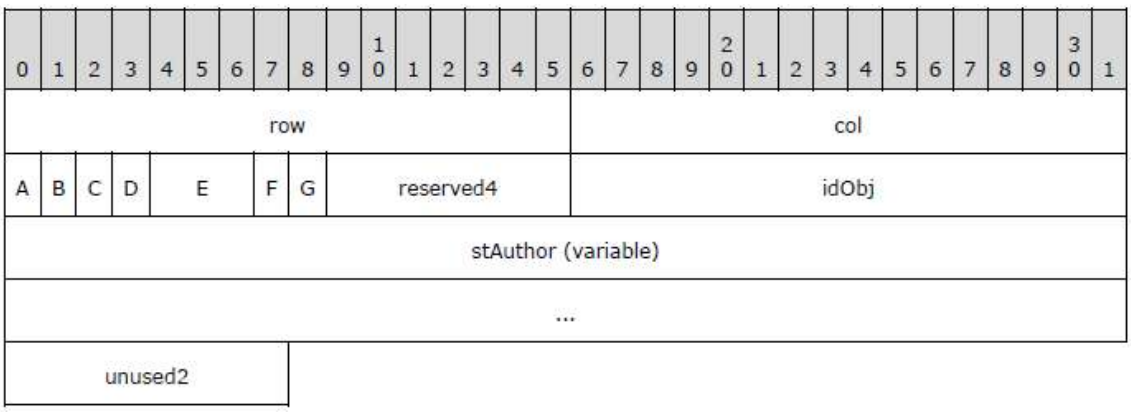
○ 애니메이션 기능성(AnimationInfoAtom)는 Record Header 컨테이너의 recType이 0x0FF1로 정의되며 아래 그임은 비트스트림에서 애니메이션 기능성을 추출한 결과이며 리틀인디언 방식[97]으로 저장되어 있음




<그림 3-125> 애니메이션 기능성 Record Header 실제 정보

❒ 엑셀 디지털 컴포넌트의 메모장 기능성 추출

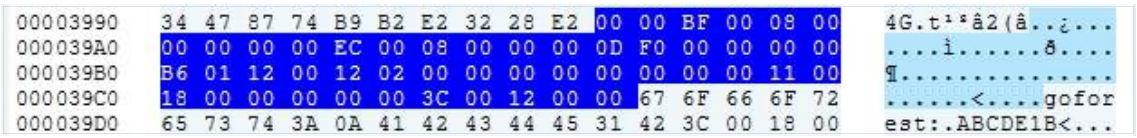
○ XLS에서 메모를 저장하는 컨테이너는 다음과 같이 정의됨




<그림 3-126> NoteSH 컨테이너

97) 데이터가 가장 낮은 바이트부터 저장하는 방식이고 윈도우시스템에서 많이 사용하는 방식이다.

○ 다음 그림은 비트스트림에서 ‘goforest’라는 저작자가 만든 메모 추출 예시




<그림 3-127> NoteSH 실제 데이타

### (3) Characterization 연구 필요성

❒ 디지털 컴포넌트가 렌더링 기능성을 포함하고 있는지 여부를 파악하기 위해 디지털 컴포넌트의 비트스트림에서 렌더링 기능성에 대한 속성을 직접 추출

○ 아래 표는 Characterization을 위한 구성요소를 제시하는데, 향후 디지털 컴포넌트에 포함된 기능성 추출이 가능하도록 기능성 속성 정보를 기술정보은행(DFR)에 등록할 수 있도록 Element 정의

○ 아래 표는 Characterization을 위한 구성요소를 제시하였는데, 향후 기능성 자동 추출 도구 개발 및 활용을 위해서는 현재 국가기록원 기술정보은행(DFR)에 Characterization 기술 정보를 추가하여 확장하여야 함

<표 3-39> Characterization을 위한 구성요소

| Element | 설명 |
|---|---|
| General | 포맷의 일반적인 요소 |
| Character | 내부시스템 식별자 |
| ID | 고유한 식별자 |
| name | 파일포맷 이름 |
| version | 파일포맷 버젼 |
| containername | 컨테이너 이름 |
| InternalSignature | 포맷의 기술적인 요소 |
| SignatureID | 시그너처 식별자 |
| name | 시그러처 이름 |
| note | 설명 |
| bytesequence | 시그너처 정보가 들어 있는 저장소 |
| positionType | 위치 정보(BOF, EOF, COF) |
| offset | 오프셋 |
| sequence | 바이트 시퀀스 값 |
| shift | 바이트 시퀀스 값을 분리하여 저장하는 곳 |
| byte | 바이트 시퀀스 값의 위치 |
| value | 바이트 시퀀스 값 |

○ 향후 개발이 필요한 Characterization 추출 도구는 컨테이너 정보 및 시퀀스 값을 기반으로 디지털 컴포넌트에 포함된 렌더링 기능성을 비트스트림과 비교하여 직접 추출하여야 함.

○ 아래에 Characterization 구성 요소 형식에 따라 한/글, 파워포인트, 엑셀 애플리케이션에 포함된 렌더링 기능성 추출을 위한 속성 정보를 등록하도록 XML Schema를 정의하였는데 다음 연구를 통해 XML Schema에 기능성에 대한 정확한 속성 정보 등록 및 관리가 필요

<표 3-40> Characterization을 위한 XML Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http//www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:element name="FileCharacter">
        <xs:annotation>
            <xs:documentation>Root Element</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
            <!-- 포맷에 관한 ID, 버젼과 같은 일반 적인 정보 저장 -->
                <xs:element ref="General"/>
                <!-- 바이크 시퀀스 값을 저장하는 곳 -->
                <xs:element ref="InternalSignature"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>    <!-- <xs:element name="FileCharacter"> -->
    <!-- 렌더링을 위한 일반 정보 -->
    <xs:element name="General">
        <xs:annotation>
            <xs:documentation>Characterization을 위한 일반 정보</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Character">
                    <!-- 고유한 값 -->
                    <xs:attribute name="ID"/>
                    <!-- 파일포맷의 이름 -->
                    <xs:attribute name="name"/>
                    <!-- 파일 포맷에 대한 버젼 정보 -->
                    <xs:attribute name="version"/>
                    <!-- 컨테이너 이름 -->
                    <xs:attribute name="containername"/>
                </xs:element>    <!-- <xs:element name="Character"> -->
            </xs:sequence>
        </xs:complexType>
    </xs:element>    <!-- <xs:element name="General"> -->
    <!-- 렌더링을 위한 일반 정보 -->
    <xs:element name="InteralSignature">
        <xs:annotation>
            <xs:documentation>signature을 위한 기술적인 정보</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="signatureID">
                    <!-- 컨테이너의 이름 -->
                    <xs:attribute name="name"/>
                    <xs:attribute name="note"/>
                    <xs:complexType>
                        <!-- 판별하기 위한 byte 정보를 저장하는 곳 -->
```

```
                        <xs:element name="bytesequence">
                            <xs:complexType>
                                <!-- BOF|EOF|COF -->
                                <xs:element name="positionType"/>
                                <!-- 오프셋 -->
                                <xs:element name="offset"/>
                                <!-- 바이트 시퀀스 값 -->
                                <xs:element name="sequence"/>
                                <!-- 바이트 시퀀스 값을 분리하여 값과 위치 저장 -->
                                <xs:element name="shift">
                                    <xs:attribute name="byte"/>
                                    <xs:attribute name="value"/>
                                </xs:element>
                            </xs:complexType>
                        </xs:element>
                    </xs:complexType>
                </xs:element>    <!-- <xs:element name="signatureID"> -->
            </xs:sequence>
        </xs:complexType>
    </xs:element>    <!-- <xs:element name="InteralSignature"> -->
</xs:schema> <!-- 끝 -->
```

❒ 국가기록원의 전자기록물에 대한 에뮬레이션 전략을 수립하기 위해서는 전자기록물에 포함된 디지털 컴포넌트 특성 파악이 필요하며, 이를 위해 다음과 같은 기능을 갖는 디지털 컴포넌트에 대한 Characterization 추출 도구 개발이 필요

○ Characterization 추출 도구는 디지털 컴포넌트에 있는 비트스트림을 분석하여 에뮬레이션이 필요한 요소가 들어가 있는지 추출함으로써 에뮬레이션 필요성 유무를 자동으로 파악할 수 있는 도구를 말함

○ 이러한 Characterization 추출 도구는 기록물 생산단계에서 자동으로 에뮬레이션의 대상이 되는 요소가 있는지 자동으로 파악할 수 있도록 하거나, 이미 생산된 기록물에 대해서는 CAMS시스템과 연동하여 기록물을 제공되는 시점에서 에뮬레이션 요소를 자동으로 파악할 수 있도록 개발되어야 함

### 바. 기능성의 속성 분석 결과 및 시사점

❒ 디지털 컴포넌트 개발사에서 제공하는 기술 문서를 기반으로 인쇄 렌더링으로 소실되는 렌더링 기능성의 디지털 컴포넌트 구조 속성을 분석한 결과 한/글 기술문서에서는 다음과 같은 시사점을 도출함

○ HWP 기술 문서에 대한 저작권

- 기술 문서는 공개하였지만 지적재산권의 자유로운 사용을 허용한 것은 아니기 때문에 특허, 상표권 등의 지적재산권을 포함하고 있음
- 기술 문서를 기반으로 특정 소프트웨어를 구현하였을 때 개발사에 피해를 초

래할 경우에는 사용권 제한이 가능함. (주)한글과컴퓨터에 사전에 공지가 필요함

○ 계층구조 및 개체 공통 속성에 대한 정보가 불명확

- “모든 레코드는 계층구조로 표현할 수 있는데.”라고 되어 있지만 전체 구조를 파악할 수 있는 계층구조에 대한 설명이 없음
- “개체 공통 속성”과 같은 경우는 여러 곳에서 공통으로 사용하고 있는데, 그것을 참조하고 있는 레코드들이 불명확하게 표시되어 있음

○ 한/글 버전별 필드 변경 사항 누락.

- 문단모양 (HWPTAG_PARA_SHAPE)의 속성2, 속성3, 줄간격 필드의 경우, 모든 5.0대 디지털 컴포넌트 포맷에 존재하는 것이 아니고, 마이너 버전이 업그레이드 되면서 특정 버전 이상의 디지털 컴포넌트 포맷에만 존재하는듯하지만 이에 대한 정보가 부족함
- 표 개체(HWPTAG_TABLE)의 경우에도 Valid Zone Info Size, 영역 속성 필드 역시 특정 버전 이상에만 존재하지만 정보가 부족함

○ 구성요소의 속성 정보인지 하위 레코드인지 불분명

- Format 5.0 디지털 컴포넌트를 구성하는 레코드들은 레벨에 따라 트리를 구성하게 되며 각 레코드들은 각자의 필드 외에 하위 레코드를 두는데 기술 문서에는 필드와 하위 레코드를 구분하지 않고 표기

○ 태그 등 정보에 대한 설명 누락

- 문단의 영역 태그의 경우 12Byte로 구성되어 있지만, 하위 4Byte에 대해서는 종류를 나타내는 1Byte와 임의의 데이터를 나타내는 3Byte로 구성되어있는데 종류와 임의의 데이터에 대한 내용이 없음

○ 누락되거나 잘못된 필드 및 레코드 존재

- 문단 헤더 (HWPTAG_PARA_HEADER)”의 경우, “control mask” 필드에 대한 설명이 누락됨
- 문단 헤더의 경우 “text(=chars)” 필드의 경우 최상위 비트에 대한 설명이 누락됨
- 글자 모양 속성에서 “Kerning 여부”라고 되어 있고, 화면에 대한 설명이 전혀 없음

○ 나열된 필드에 대한 설명 부족

- HWPTAG_LIST_HEADER의 경우 문맥에 따라 표 캡션 리스트나 표 셀 리스트 등으로 해석되기도 하는데 이러한 차이에 대한 구체적인 설명 없이 단순히 필드만 나열하고 있어 정확한 내용을 파악하기 곤란
- HWPTAG_BORDER_FILL의 경우 글자 모양 속성에서 취소선 형태로 사용될 때는 전혀 다른 의미로 사용되고 있는데 의미 파악의 혼동을 줄 수 있음
- 탭 정의에서 “채움 종류(표20참조)”라고 되어 있는데, 표20은 테두리선 종류에 대한 정보를 나타내고 있어서 표20에 대한 부가적인 설명 필요

○ 필드 정보에 대한 설명 누락
- 문단 속성 모양에서 필드 전체에 대한 값과 설명 정보가 누락되어 있어서 의미를 파악 불가
- OLE 개체 속성의 속성과 같은 경우는 프로그램 내부적으로 사용되는 코드를 설명 자료로 사용하여 의미 전달이 불가능
- “HWP Docuement File 5.0”의 덧말의 경우 Fsizeration, Option, Styl number에 대한 부가적인 설명 자료가 없어서 의미 파악이 불가능

○ 외부 참조 필드에 대한 설명 부족
- OLE 개체 속성의 속성과 같은 경우는 외부 문서를 참조로 되어 있음
- Summary Information에 대한 자세한 설명은 MSDN 참조로 되어 있음

❒ 기술 문서를 기반으로 애플리케이션과 동일한 기능을 구현하기 위해서는 기능성을 저장하기 위한 Format 정보(테이블 정의서, 컨테이너 정의서, 속성 정의서)와 더불어 기능성을 화면에 표시하는 View 정보가 필요한데 기술 문서에는 화면 표시와 관련된 View정보는 존재하지 않음

### 3.3.4 디지털 컴포넌트의 기능성 보존 방안 연구

#### 가. 기능성 보존 연구 개요

#### (1) 연구 배경 및 필요성

❒ 본 절에서는 앞 절에서 분석한 렌더링 기능성 정보를 보존하기 위해 국가기록원의 기술정보은행(DFR)에 렌더링 기능성 정보를 추가하는 방안에 대하여 연구하고자 함

❒ 먼저 국가기록원의 기술정보은행(DFR)의 레지스트리 구성과 레지스트리에 등록

된 XML 데이터에 대하여 살펴본 후 비교 분석을 위해 국내외에서 가장 많이 사용되는 포맷 식별 도구인 PRONOM & DROID의 레지스트리 구성과 XML 데이터의 상태를 비교 분석하고자 함

❒ 마지막으로 인쇄 렌더링으로 소실되는 렌더링 기능성을 기술정보은행(DFR)의 레지스트리에 저장할 수 있도록 개선된 XML Schema 구조를 제안하고자 함

### (2) 국내외 포맷 레지스트리 연구 현황

❒ 국내외 모든 포맷 식별 도구들은 파일 확장자 또는 포맷 시그니처를 이용하여 파일의 포맷을 식별하며 식별 도구마다 자체적으로 포맷 시그니처를 표현할 수 있으나 국내외 동향에서 살펴본 결과에 의하면 주요 오픈 소스 식별 도구들은 DROID의 식별 방법을 채택하고 있으며 PRONOM에서 얻은 시그니처 파일을 이용하고 있음. 따라서 식별 기술 연구에 있어 DROID의 식별 방법과 포맷 시그니처 표현 규칙을 참조할 필요가 있다고 판단

❒ PRONOM 포맷 특성 정보

- ○ DROID에서는 포맷 특성 정보가 비트스트림 외부에 존재하느냐 내부에 존재하느냐에 따라 외장 시그니처와 내장 시그니처로 구분됨
- ○ 파일 외부에 존재하는 외장 시그니처란 디지털 객체의 비트스트림 바깥에 존재하는 포맷 특성 데이터로서 Windows의 경우 파일 확장자, Mac OS의 경우 데이터 폭스(data forks)가 외장 시그니처에 해당됨
- ○ 반면 내장 포맷 시그니처란 디지털 객체의 비트스트림 속의 특정 위치에 존재하는 포맷에 대한 특성 값임
- ○ 포맷과 시그니처와의 관계를 살펴보면, 각 포맷은 다수의 내장 시그니처 또는 파일 확장자를 가지며 각 내장 시그니처는 1개 이상의 바이트 시퀀스로 구성됨. 포맷에는 식별자가 부여되고 포맷과 포맷 사이에는 수많은 관계를 가질 수 있음

❒ 국내 포맷 레지스트리 연구 현황

- ○ 해외에서는 2000년대 초에 이미 전자 기록의 보존에서 포맷의 영향을 인식하고 포맷 레지스트리와 식별 방법에 대한 연구를 시작하여 PRONOM과 GDFR이 이미 구축되어 포맷 정보를 관리하고 디지털 객체의 포맷을 식별하는 방법을 연구하여 DROID 등의 주요 소프트웨어를 개발하여 활용하고 있으나 국내에서는 전자 기록 보존에 대한 연구 역사가 짧은 만큼 포맷 레지스트리나 식별, 검증 등에 대한 연구가 전무

○ 국내외 관련 연구는 2008년 국가기록원의 기록 관리 연구 개발 (R&D) 사업을 통해 수행된 “디지털 포맷 및 애플리케이션 기술정보은행(DFR)을 위한 시스템 설계” 사업이 유일하며, 이 연구 성과로서 기술정보은행의 프로토타입이 개발되었으나 국가기록원은 물론 공공기관의 기록보존소에서 기술정보은행의 기능을 활용하고 있지 않은 실정임

❒ 국내 포맷 레지스트리 구축 현황

○ 국가기록원의 2008년도 기록 관리 연구 개발 (R&D) 사업 지원에 의해 수행된 “디지털 포맷 및 애플리케이션 기술정보은행(DFR)을 위한 시스템 설계” 사업 결과로 디지털 기록물의 재현을 위한 기술정보의 수집 및 관리 정보를 위한 요소들을 정의하고 기술정보를 이용하여 디지털 기록물의 장기보존을 위한 기능을 제공하고 있다. 또한 국내 실정에 맞는 디지털 기록물의 보존을 위한 포맷 레지스트리를 설계하고 이를 기술정보은행 (DFR:Digital Format Registry)이라고 명명

○ 1차 연구 이후 보다 다양한 문서 형식을 지원하기 위한 고도화 사업을 수행하여 업무 환경에 가장 빈번히 사용되는 문서 형식으로 2006년, 2008년에 각각 ISO 26300과 ISO 29500 이라는 국제 표준 승인된 ODF(Open Document Format) - (추가 15종의 문서 형식)과 OOXML(Office Open XML) - (추가 3종의 문서 형식) 개방형 오피스 문서 형식을 추가 지원하게 되었으며, ISO 28500 WARC라는 웹 기록물 저장 국제 표준 형식에 대한 유효성 검증 기능을 개발하여 지원하고 있다. 또한 국내 실정을 반영하여 MS오피스, 한컴 오피스, 하나워드, 아리랑, 훈민정음과 같은 레거시 포맷과 국가기록원의 장기보존포맷인 NEO 패키지 파일에 대한 유효성 검증 기능까지 추가

## 나. 국가기록원 기술정보은행(DFR) 구조 분석

### (1) 국가기록원 기술정보은행(DFR) 구성 정의

❒ 국가기록원 기술정보은행 기능 구성 요소

○ 국가기록원 기술정보은행(DFR)은 아래 표와 같이 파일 식별, 파일 검증, 파일 특성 정보, 파일 배포, 텍스트 추출의 5개 기능으로 구성

○ 파일 식별, 검증, 특성 정보 추출, 배포 기능, 텍스트 추출을 포맷별로 모듈화하여 포맷의 추가적인 지원에 대비할 수 있도록 설계되었으며 식별, 배포, 검증, 특성 기능을 중심으로 플러그인 형식으로 추가된 각 포맷에 대한 모듈들을 모듈 관리자가 관리하는 구조로 설계

<표 3-41> 기술정보은행(DFR) 기능 구성 요소

| 기능 항목 | 기능 설명 |
|---|---|
| 파일 식별 | 포맷의 확장자 및 바이트스트림 내의 서명 정보를 이용하여 포맷을 확인하는 기능 |
| 파일 검증 | 디지털 기록물 내에 오류가 있는지 확인하는 기능 |
| 파일특성 정보 | 디지털 기록물의 중요한 특성 정보를 제공하는 기능 |
| 파일 배포 | 특정 디지털 기록물의 접근에 필요한 소프트웨어 및 하드웨어 등의 정보를 제공하는 기능 |
| 텍스트 추출 | 파일 포맷 데이터베이스 정보를 이용하여 사용자가 원하는 텍스트 추출 |

❒ 국가기록원 기술정보은행(DFR)은 파일의 식별 기능과 더불어 검증 및 특성 정보 등 메타데이터를 추출하는 대표적인 도구로서 뉴질랜드 국립도서관 Metadata Extraction Tool, JHOVE, 국가기록원 기술정보은행(DFR)과 비교할 수 있는데, 뉴질랜드 국립도서관 Metadata Extraction Tool은 단독 추출 도구이며, JHOVE은 식별 도구와 함께 제공되지만, 국가기록원 기술정보은행(DFR)은 레지스트리 기능으로 제공됨

<표 3-42> 추출 도구 비교(Comparison of extraction tools)

| 구분 | Metadata Extraction Tool | JHOVE | 기술정보은행(DFR) |
|---|---|---|---|
| 개발 주체 | 뉴질랜드 국립도서관 | California Digital Library (CDL), Portico, Stanford University<br>* JHOVE1.0은 JSTOR와 하버드 대학교 도서관 | 대한민국 국가기록원 |
| 라이센스 | 공개 소스 | 공개 소스 | 비공개 |
| 지원 포맷 | 약 20종 | JHOVE2.0 약 10종 | 약 18종 |
| 포맷 지원 방법 | 포맷별 모듈화 | 포맷별 모듈화 | 포맷별 모듈화 |
| 실행 방법 | 단독 도구 | 식별, 검증, 평가 도구와 함께 제공 | 레지스트리, 식별, 검증 도구와 함께 제공 |
| 실행 방법 | 그래픽인터페이스 명령어 라인 | 명령어 라인 | 웹서비스 |

❒ 현행 기술정보은행(DFR) 구조[98]

○ 현행 기술정보은행에는 “포맷에 대한 기술 정보”와 “소프트웨어에 대한 기술 정보”로 구성되어 있음

98) 디지털 포맷 및 애플리케이션 기술정보은행(DFR) 프로토타입 개발, 국가기록원 ,2008

<표 3-43> 기술정보은행 정보 구성 요소

| 포맷에 대한 기술 정보 | 소프트웨어에 대한 기술 정보 |
|---|---|
| 자동 생성되는 관리 번호 | 내부 시스템의 식별자 |
| 포맷 등록 시스템 내 식별자 | 소프트웨어 상의 외부 식별자 |
| **포맷명** | **소프트웨어 이름** |
| **버전** | **소프트웨어의 버전 정보** |
| 다른 포맷명 | 소프트웨어 별칭 |
| 포맷 유형 | 소프트웨어 유형 |
| 포맷 상태 | 소프트웨어 상태 |
| 지적재산권 | **소장여부 및 소장 위치** |
| 개발자 | **처리 가능한 포맷** |
| 포맷 지원/유지 기관 | 소프트웨어 의해 지원된 언어 |
| 포맷 발행일 | 소프트웨어의 일반적인 특징 |
| 포맷 지원 종료일 | 썸네일 이미지 |
| 포맷 공개 수준 | 지적재산권 |
| Text기반/Binary기반 | 개발자 |
| 관련 포맷 | 소프트웨어 개발회사 |
| **포맷의 기술 문서** | 지원/유지 기관 |
| **외부 서명 항목(확장자)** | 소프트웨어 발행일 |
| 내부 서명 항목 | 지원 종료일 |
| 기술적인 환경 | 관련 소프트웨어 |
| **Note Well-formed에 관한 설명 혹은 문서 위치** | **소프트웨어의 기술 문서** |
| **Note Validity에 관한 설명 문서 위치** | **소프트웨어 지원에 필요한 운영시스템과 기타 S/W 요건** |
| 포맷 메타데이터 | **소프트웨어 지원에 필요한 하드웨어 요건** |
| 정보주기 | 소프트웨어 저장 매체 유형 |
| | 정보주기 |

❒ 포맷에 대한 기술 정보

○ 파일 포맷 관리에 필요한 일반적인 내용만 포함하고 있음

- 포맷에 대한 일반적인 요소(식별자, 개발자, 개발회사, 지원정보 등)
- 포맷의 기술적인 요소(포맷공개수준, 스펙 문서 리스트, 내외부서명항목 등)
- 포맷의 관리에 관한 요소

○ Technical>InteralSignature에서 파일 포맷 식별에 필요한 정보를 담고 있음.

○ Technical>Specification>Document에 “스펙문서를 위한 Identifier 값”을 가지고 있으나, 스펙문서를 위한 기술정보는 구축되어 있지 않음

<표 3-44> 파일 포맷 구성 요소

| Element | 설명 |
|---|---|
| General | 포맷의 일반적인 요소 |
| **FormatID** | 자동 생성되는 관리번호 |
| **Identifier** | 포맷 등록 시스템내에서 식별자 |

| | |
|---|---|
| **FormatName** | 포맷명 |
| **FormatVersion** | 포맷버전 |
| **FormatAlias** | 다른 포맷명 |
| **FormatType** | 포맷유형(파일 포맷 온톨로지 분류체계) |
| **Status** | 포맷상태(현용/위험상태/노후화/기타) |
| IPR | 지적 재산권 |
| IPRType | 유형(저작권/특허/기타) |
| Owner | 소유권자 |
| Name | 이름 |
| Address | 주소 |
| Tel | 전화번호 |
| FAX | 팩스번호 |
| E-mail | 이메일 |
| Web | 웹페이지 주소 |
| Available | 유효기간 |
| Note | 주기 |
| Developer | 개발자 정보 |
| Name | 이름 |
| Address | 주소 |
| Tel | 전화번호 |
| FAX | 팩스번호 |
| E-mail | 이메일 |
| Web | 웹페이지 주소 |
| Support | 포맷지원/유지기관 |
| Name | 이름 |
| Address | 주소 |
| Tel | 전화번호 |
| FAX | 팩스번호 |
| E-mail | 이메일 |
| Web | 웹페이지 주소 |
| ReleaseDate | 포맷 발행일 |
| WithdrawDat | 포맷 지원 종료일 |
| Note | 정보주기 |
| Technical | 포맷의 기술적인 요소 |
| FormatDisclose | 포맷 공개 수준 |
| Orientation | Text기반/Binary기반 |
| RelatedFormat | 관련 포맷 |
| Equality | 동등관계 |
| RelatedVersion | 이전/이후 버전 |
| RelatedType | 상위/하위 유형 |
| Etc | 기타 |
| Specification | 포맷의 스펙 문서 리스트 |
| Document | 포맷의 스펙문서 |
| Document Title | 타이틀 |
| Document Type | 타입 |
| Author | 저자 리스트 |
| Name | 이름 |
| Address | 주소 |
| Tel | 전화번호 |
| FAX | 팩스번호 |
| E-mail | 이메일 |
| Web | 웹페이지 주소 |
| Publisher | Publisher 리스트 |

| | |
|---|---|
| Name | 이름 |
| Address | 주소 |
| Tel | 전화번호 |
| FAX | 팩스번호 |
| E-mail | 이메일 |
| Web | 웹페이지 주소 |
| Document Edition | 문서의 Edition 정보 |
| Document Date | 문서 생성 일자 |
| Document Enumeration | Serial voulm and number |
| Document page | 문서 전체 페이지 |
| Identifier | 스펙문서를 위한 identifier 정보 |
| Value | 값 |
| Type | 타입 |
| Note | 설명 |
| Document Note | 문서 정보주기 |
| ExternalSignature | 외부 서명 항목 |
| **InternalSignature** | 내부 서명 항목 |
| **ByteSequence** | 포맷 식별이 가능한 바이트 시퀀스 |
| **SubSequnce** | 각각의 바이트 시퀀스는 다수의 서브 시퀀스로 구성 |
| **Sequence** | 구체적인 식별 항목 |
| **DefaultShift** | BOF, EOF로부터 shift |
| **Shift** | 시퀀스를 이루는 각각의 byte값의 shift 정보 |
| **LeftFragment** | 시퀀스를 중심으로 왼쪽으로 나타낼수 있는 바이트 값 |
| **RightFragment** | 시퀀스를 중심으로 오른쪽으로 나타낼수 있는 바이트 값 |
| TechnicalEnvironment | 기술적인 환경 |
| Well-formed Note | Well-formed에 관한 설명 |
| Validity Note | Validity에 관한 설명 문서 위치 |
| Metadata | 파일 포맷 메타데이터를 담고 있음 |
| name | |
| value | |
| child | |
| Note | 정보주기 |
| Management | 포맷관리에 대한 요소 |
| Source | 정보제공자 |
| Name | 이름 |
| Address | 주소 |
| Tel | 전화번호 |
| FAX | 팩스번호 |
| E-mail | 이메일 |
| Web | 웹페이지 주소 |
| SourceDate | 포맷등록시스템 등록일 |
| LastUpdate | 최종 업데이트일 |
| Type | 포맷관리 상태 |
| Note | 정보주기 |

○ 파일 포맷 구성 요소에서 가장 중요한 Element는 Format 관련 Element와 포맷 식별을 위한 InternalSignatre Element이고 그 외의 Element는 부가적인 정보를 기술하기 위한 Element로서 포맷 식별에 크게 영향을 미치는 요소가 아님

❒ 소프트웨어에 대한 기술 정보

○ 소프트웨어 관리에 필요한 일반적인 내용만 포함하고 있음

- 소프트웨어에 대한 일반적인 요소(식별자, 개발자, 개발회사, 지원/유지기관 정보 등)
- 소프트웨어의 기술적인 요소(관련소프트웨어, 스펙 문서 리스트, 운영시스템과 기타요건, 저장매체 유형 등)
- 포맷의 관리에 관한 요소

○ Technical>MediaFormat의 경우, 국가기록원은 CAMS에 모두 디지털화 되어서 HDD 혹은 Storage에 저장되기 때문에 저장 매체 유형은 불필요

**<표 3-45> 소프트웨어 구성요소**

| Element | 설명 |
|---|---|
| General | 포맷의 일반적인 요소 |
| **SoftwareID** | 내부시스템 식별자 |
| **Identifier** | 외부식별자 |
| **SoftwareName** | 이름 |
| **SoftwareVersion** | 버젼 |
| **SoftwareAlias** | S/W 별칭 |
| **SoftwareType** | S/W 타입 |
| **Status** | S/W 상태(현용/노후화/사용되지않음/기타) |
| Location | 소장여부 및 소장 위치 |
| ProcessFormat | 처리 가능한 포맷 |
| Default | default 포맷 |
| Create | 생성 가능한 포맷 |
| Render | 렌더링 가능한 포맷 |
| Validate | validate 가능한 포맷 |
| etc | 기타 포맷 |
| Language | 지원된 언더 |
| Feature | 일반적인 특징 |
| Image | 쎔네일 이미지 |
| IPR | 지적 재산권 |
| IPRType | 유형(저작권/특허/기타) |
| IPROwner | 소유권자 |
| Name | 이름 |
| Address | 주소 |
| Tel | 전화번호 |
| FAX | 팩스번호 |
| E-mail | 이메일 |
| Web | 웹페이지 주소 |
| Available | 유효기간 |
| Note | 주기 |
| Developer | 개발자 정보 |
| Name | 이름 |
| Address | 주소 |
| Tel | 전화번호 |
| FAX | 팩스번호 |

| | |
|---|---|
| E-mail | 이메일 |
| Web | 웹페이지 주소 |
| Corperation | 개발회사 정보 |
| Name | 이름 |
| Address | 주소 |
| Tel | 전화번호 |
| FAX | 팩스번호 |
| E-mail | 이메일 |
| Web | 웹페이지 주소 |
| Support | 지원/유지 기관 정보 |
| Name | 이름 |
| Address | 주소 |
| Tel | 전화번호 |
| FAX | 팩스번호 |
| E-mail | 이메일 |
| Web | 웹페이지 주소 |
| ReleaseDate | 발행일 |
| WithdrawDate | 지원 종료일 |
| Note | 정보주기 |
| Technical | 포맷의 기술적인 요소 |
| RelatedSoftware | 관련 소프트 웨어 |
| Equality | 동등관계 |
| ReleatedVersion | 이전/이후 버전 |
| ReleatedType | 상위/하위 유형 |
| Etc | 기타 |
| Document | 소프트웨어 스펙문서 |
| DocumentTitle | 문서의 제목 |
| DocuementType | 문서의 타입 |
| Author | 문서의 저자 리스트 |
| Name | 이름 |
| Address | 주소 |
| Tel | 전화번호 |
| FAX | 팩스번호 |
| E-mail | 이메일 |
| Web | 웹페이지 주소 |
| Publisher | Publisher 리스트 |
| Name | 이름 |
| Address | 주소 |
| Tel | 전화번호 |
| FAX | 팩스번호 |
| E-mail | 이메일 |
| Web | 웹페이지 주소 |
| DocumentEdition | 문서의 Edition 정보 |
| DocumentDate | 문서 생성 일자 |
| DocumentEnum | serial volume and number |
| DocumentPage | 문서 전체 페이지 |
| **Identifier** | **스펙문서를 위한 identifier 정보** |
| **Value** | **Identifier 값** |
| **Type** | **identifier 타입** |
| **Note** | **identifier 설명** |
| Document Note | 문서 정보 주기 |
| SoftwareRequirement | S/W 지원에 필요한 운영시스템과 기타 S/W 요건 |
| HardwareRequirement | S/W 지원에 필요한 H/W 요건 |

| | | |
|---|---|---|
| | MediaFormat | S/W 저작 매체 유형 |
| Management | | 소프트웨어 관리에 대한 요소 |
| | Source | 정보제공자 |
| | Name | 이름 |
| | Address | 주소 |
| | Tel | 전화번호 |
| | FAX | 팩스번호 |
| | E-mail | 이메일 |
| | Web | 웹페이지 주소 |
| | SourceDate | 포맷등록시스템 등록일 |
| | LastUpdate | 최종 업데이트일 |
| | Type | 포맷관리 상태 |
| | Note | 정보주기 |

○ 소프트웨어 구성 요소에서 가장 중요한 Element는 볼드체로 표시된 Software 관련 Element와 소프트웨어 식별을 위한 Identifier Element이고 그 외의 Element는 부가적인 정보를 기술하기 위한 Element임

### (2) 국가기록원 기술정보은행(DFR)에 등록된 XML 데이터 분석

❒ 국가기록원 기술정보은행(DFR)에 등록된 한/글 포맷 정보

<표 3-46> 국가기록원 기술정보은행(DFR)에 등록된 한/글 포맷 정보

```
<?xml version="1.0" encoding="EUC-KR"?>
<FileFormat xmlns:droid="http://pronom.nationalarchives.gov.uk">
    <General>
        <FormatID>821</FormatID>
        <PUID-TI>x-fmt/430</PUID-TI>
        <MIMEType>application/haansoft-hangul</MIMEType>
        <FormatName>Haansoft Hangul</FormatName>
        <FormatVersion>97-2007</FormatVersion>
        <FormatAliases/>
        <FormatType/>
        <Status>        </Status>
        <Support>
            <Individual>
                <Name>(주)한글과컴퓨터</Name>
                <Address>서울시 광진구 구의동 프라임 센터 20층</Address>
                <Tel>+82 02-3424-3400</Tel>
                <Fax>+82 02-3424-4062</Fax>
                <E-mail>hncxml@haansoft.com</E-mail>
                <Web>http://www.haansoft.com</Web>
            </Individual>
        </Support>
        <ReleaseDate></ReleaseDate>
        <WithdrawDate></WithdrawDate>
        <Note>        </Note>
        <IRPList>
            <IRP>



            </IRP>
        </IRPList>
        <Description/>
    </General>
    <Technical>
        <FormatDisclosure/>
```

```
        <Orientation/>
        <RelatedForamtList>
            <RelatedVersion/>
            <Equality/>
            <RelatedType/>
            <Etc/>
        </RelatedForamtList>
        <ExternalSignature>
            <ExtensionName>hwp</ExtensionName>
        </ExternalSignature>
                                    ☜ 가장 중요한 InternalSignature 가 공란으로 비어 있음
        <TechnicalEnvironment/>
        <Well-formedNote/>
        <ValidityNote/>
        <Metadata>
            <Name/>
            <Value/>
        </Metadata>
        <Note/>
        <Specification>
            <Document>
                <Title/>
                <Type/>
                <Author>
                    <Individual>
                        <Name/>
                        <Address/>
                        <Tel/>
                        <Fax/>
                        <E-mail/>
                        <Web/>
                    </Individual>
                </Author>
                <Publisher>
                    <Individual>
                        <Name/>
                        <Address/>
                        <Tel/>
                        <Fax/>
                        <E-mail/>
                        <Web/>
                    </Individual>
                </Publisher>
                <Edition/>
                <Date/>
                <Enumeration/>
                <Page/>
                <Identifier/>
                <Note/>
            </Document>
        </Specification>
    </Technical>
    <Management>
        <Source>
            <Individual>
                <Name>박동현</Name>
                <Address>서울시 광진구 구의동 546-4 프라임센터 20층 (주)한글과컴퓨터</Address>
                <Tel>02-3424-0534</Tel>
                <Fax>02-3424-5983</Fax>
                <E-mail>dh.park@haansoft.com</E-mail>
                <Web>http://www.haansoft.com</Web>
            </Individual>
        </Source>
        <Date>2008-11-19 15:43:43</Date>
        <LastUpdate>2008-11-19 15:43:43</LastUpdate>
        <Type/>
        <Note/>
    </Management>
</FileFormat>
```

○ 한글 포맷에 대한 XML 정보를 살펴 보면 가장 중요한 태그 정보인 InternalSignature를 비롯하여 대다수 태그 정보가 공란으로 되어 있는 것을 확인할 수 있었으며, 다양한 태그에 대한 정의는 되어 있으나 기술정보은행(DFR)

에 등록된 XML 정보에는 대부분 누락되어 있음을 확인함.

○ 구성 요소 정의가 중요한 것이 아니라 정확한 데이터가 등록되어 관리되어야 하나, 현재 국가기록원 기술정보은행(DFR)에 저장된 포맷 및 소프트웨어에 대한 XML 데이터를 분석한 결과 가장 중요한 Element 정보가 누락된 상태이므로 레리스트리 역할이 불가능한 것으로 파악됨

❒ 국가기록원 기술정보은행(DFR)에 등록된 PDF 포맷 정보

<표 3-47> 국가기록원 기술정보은행(DFR)에 등록된 PDF 포맷 정보

```
<?xml version="1.0" encoding="EUC-KR" ?>
<FileFormat xmlns:droid="http://pronom.nationalarchives.gov.uk">
    <General>
        <FormatID>637</FormatID>
        <MIMEType>application/pdf</MIMEType>
        <PUID-TI>fmt/20</PUID-TI>
        <FormatName>Portable Document Format</FormatName>
        <FormatVersion>1.6</FormatVersion>
        <FormatAliases>PDF (1.6)</FormatAliases>
        <FormatType>Page Description</FormatType>
        <Status />
        <Developer>
            <Individual>
                <Name />
                <Address />
                <Tel />
                <Fax />
                <E-mail />
                <Web />
            </Individual>
        </Developer>
        <Support>
            <Individual>
                <Name />
                <Address />
                <Tel />
                <Fax />
                <E-mail />
                <Web />
            </Individual>
        </Support>
        <ReleaseDate>01 Jan 2004</ReleaseDate>
        <WithdrawDate />
        <Note />
        <IRPList>
            <IRP>
                <Type />
                <Owner>
                <Individual>
                    <Name />
                    <Address />
                    <Tel />
                    <Fax />
                    <E-mail />
                    <Web />
                </Individual>
                </Owner>
            <Available />
            <Note />
            </IRP>
        </IRPList>
        <Description>PDF(Portable Document Format)은 Adobe Systems Incorporated에서 개발된 포맷화된 문서들을 표현하기 위한 플랫폼 독립적인 포맷이다. PDF는 Arcobat 7.0 릴리즈에서 사용된 버전 1.6으로, Adobe사의 Arcobat 제품군을 위한 네이티브 포맷이다. PDF는 PostScript 페이지 기술 언어에 기반을 하고 있으며, 같은 이미징 모델을 공유한다. PDF 파일은 Header 섹션과 문서를 구성하는 객체들을 포함한 Body 섹션, Cross Reference Table, Trailer 섹션으로 구성되어 있다. PDF 파일은 텍스트, 이미지, 비디오, 오디오 등 다양한 컨텐츠들을 포함할 수 있다.</Description>
```

```
    </General>
    <Technical>
        <FormatDisclosure />
        <Orientation />
        <RelatedForamtList>
            <RelatedVersion />
            <Equality />
            <RelatedType />
            <Etc />
        </RelatedForamtList>
        <ExternalSignature>
            <ExtensionName>pdf</ExtensionName>
        </ExternalSignature>
        <InternalSignature>
            <SignatureID>21</SignatureID>
            <SignatureName>PDF 1.6a</SignatureName>
            <SignatureNote>Header, footer and trailer dictionary</SignatureNote>
            <ByteSequence>
                <ByteSequenceID>113</ByteSequenceID>
                <PositionType>Absolute from BOF</PositionType>
                <Offset>0</Offset>
                <MaxOffset />
                <IndirectOffsetLocation />
                <IndirectOffsetLength />
                <Endianness />
                <ByteSequenceValue>255044462D312E36(0D|0A|0D0A)</ByteSequenceValue>
            </ByteSequence>
            <ByteSequence>
                <ByteSequenceID>114</ByteSequenceID>
                <PositionType>Absolute from EOF</PositionType>
                <Offset>0</Offset>
                <MaxOffset />
                <IndirectOffsetLocation />
                <IndirectOffsetLength />
                <Endianness />
                <ByteSequenceValue>2525454F46(0D|0A|0D0A)</ByteSequenceValue>
            </ByteSequence>
            <ByteSequence>
                <ByteSequenceID>115</ByteSequenceID>
                <PositionType>Variable</PositionType>
                <Offset />
                <MaxOffset />
                <IndirectOffsetLocation />
                <IndirectOffsetLength />
                <Endianness />

<ByteSequenceValue>747261696C6572(0D|0A|0D0A)3C3C(2F|202F|0D2F|0A2F|0D0A2F)53697A6520*3E3E(0
D|0A|0D0A)737461727478726566(0D|0A|0D0A)</ByteSequenceValue>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature>
            <SignatureID>189</SignatureID>
            <SignatureName>PDF 1.6b</SignatureName>
            <SignatureNote>Header and Document Catalog Version, Footer and Trailer
Dictionary</SignatureNote>
            <ByteSequence>
                <ByteSequenceID>241</ByteSequenceID>
                <PositionType>Absolute from BOF</PositionType>
                <Offset>0</Offset>
                <MaxOffset />
                <IndirectOffsetLocation />
                <IndirectOffsetLength />
                <Endianness />

<ByteSequenceValue>255044462D312E(30|31|32|33|34|35)(0D|0A|0D0A)*2F56657273696F6E2F312E36</Byt
eSequenceValue>
            </ByteSequence>
            <ByteSequence>
                <ByteSequenceID>242</ByteSequenceID>
                <PositionType>Absolute from EOF</PositionType>
                <Offset>0</Offset>
                <MaxOffset />
                <IndirectOffsetLocation />
                <IndirectOffsetLength />
                <Endianness />
                <ByteSequenceValue>2525454F46(0D|0A|0D0A)</ByteSequenceValue>
            </ByteSequence>
```

```
                <ByteSequence>
                    <ByteSequenceID>243</ByteSequenceID>
                    <PositionType>Variable</PositionType>
                    <Offset />
                    <MaxOffset />
                    <IndirectOffsetLocation />
                    <IndirectOffsetLength />
                    <Endianness />

<ByteSequenceValue>747261696C6572(0D|0A|0D0A)3C3C(2F|202F|0D2F|0A2F|0D0A2F)53697A6520*3E3E(0
D|0A|0D0A)737461727478726566(0D|0A|0D0A)</ByteSequenceValue>
                </ByteSequence>
            </InternalSignature>
            <TechnicalEnvironment />
            <Well-formedNote />
            <ValidityNote />
            <Metadata>
                <Name />
                <Value />
            </Metadata>
            <Note />
            <Specification>
                <Document>
                    <Title>PDF Reference, fifth edition: Adobe Portable Document Format, Version
1.6</Title>
                    <Type>Authoritative</Type>
                    <Author>
                        <Individual>
                            <Name />
                            <Address />
                            <Tel />
                            <Fax />
                            <E-mail />
                            <Web />
                        </Individual>
                    </Author>
                    <Publisher>
                        <Individual>
                            <Name />
                            <Address />
                            <Tel />
                            <Fax />
                            <E-mail />
                            <Web />
                        </Individual>
                    </Publisher>
                    <Edition />
                    <Date>14 Nov 2004</Date>
                    <Enumeration />
                    <Page />
                    <Identifier>43</Identifier>
                    <Note />
                </Document>
            </Specification>
        </Technical>
        <Management>
            <Source>
                <Individual>
                    <Name>구인영</Name>
                    <Address>서울특별시 용산구 효창원길 숙명여자대학교</Address>
                    <Tel>02-710-9379</Tel>
                    <Fax>02-710-9704</Fax>
                    <E-mail>k-920@hanmail.net</E-mail>
                    <Web>http://www.sookmyung.ac.kr</Web>
                </Individual>
            </Source>
            <Date>2008-11-19 15:43:43</Date>
            <LastUpdate>2008-11-19 15:43:43</LastUpdate>
            <Type />
            <Note />
        </Management>
</FileFormat>
```

○ PDF 포맷에 대한 XML 정보에서는 Internal Signature 데이터는 등록되어 있

으나 Byte Sequence 정보를 비롯하여 대다수 태그 정보가 공란으로 되어 있는 것을 확인할 수 있음

❒ 국가기록원 기술정보은행(DFR)에 등록된 한/글 소프트웨어 정보

<표 3-48> 국가기록원 기술정보은행(DFR)에 등록된 한/글 소프트웨어 정보

```
<?xml version="1.0" encoding="EUC-KR"?>
<Software xmlns:droid="http://pronom.nationalarchives.gov.uk">
    <General>
        <SoftwareID>1000</SoftwareID>
        <PUID-TI>x-sfw/1000</PUID-TI>
        <SoftwareName>Haansoft Hangul</SoftwareName>
        <SoftwareVersion>2007</SoftwareVersion>
        <SoftwareAlias/>
        <SoftwareType>Word processing</SoftwareType>
        <Status/>
        <Location/>
        <ProcessFormat>
            <Default>
                <FileFormat>
                    <RelatedForamtID>821</RelatedForamtID>
                    <FormatName>Haansoft Hangul</FormatName>
                </FileFormat>
            </Default>
            <Create>
                <FileFormat>
                    <RelatedForamtID>8</RelatedForamtID>
                    <FormatName>Microsoft Word for  Macintosh Document</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>105</RelatedForamtID>
                    <FormatName>Microsoft Word for  Macintosh Documenat</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>106</RelatedForamtID>
                    <FormatName>Microsoft Word for  Macintosh Document</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>11</RelatedForamtID>
                    <FormatName>Microsoft Word for  Macintosh Document</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>405</RelatedForamtID>
                    <FormatName>Microsoft Word for MS-DOS Document</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>406</RelatedForamtID>
                    <FormatName>Microsoft Word for MS-DOS Document</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>407</RelatedForamtID>
                    <FormatName>Microsoft Word for MS-DOS Document</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>408</RelatedForamtID>
                    <FormatName>Microsoft Word for MS-DOS Document</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>733</RelatedForamtID>
                    <FormatName>Microsoft Word for Windows Document</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>734</RelatedForamtID>
                    <FormatName>Microsoft Word for Windows Document</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>688</RelatedForamtID>
                    <FormatName>Microsoft Word for Windows Document</FormatName>
                </FileFormat>
                <FileFormat>
```

```
                    <RelatedForamtID>690</RelatedForamtID>
                    <FormatName>Microsoft Word for Windows Document</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>76</RelatedForamtID>
                    <FormatName>Microsoft Word Template</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>626</RelatedForamtID>
                    <FormatName>Rich Text Format</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>627</RelatedForamtID>
                    <FormatName>Rich Text Format</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>628</RelatedForamtID>
                    <FormatName>Rich Text Format</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>629</RelatedForamtID>
                    <FormatName>Rich Text Format</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>630</RelatedForamtID>
                    <FormatName>Rich Text Format</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>631</RelatedForamtID>
                    <FormatName>Rich Text Format</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>632</RelatedForamtID>
                    <FormatName>Rich Text Format</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>633</RelatedForamtID>
                    <FormatName>Rich Text Format</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>821</RelatedForamtID>
                    <FormatName>Haansoft Hangul</FormatName>
                </FileFormat>
            </Create>
            <Render>
                <FileFormat>
                    <RelatedForamtID>734</RelatedForamtID>
                    <FormatName>Microsoft Word for Windows Document</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>688</RelatedForamtID>
                    <FormatName>Microsoft Word for Windows Document</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>690</RelatedForamtID>
                    <FormatName>Microsoft Word for Windows Document</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>626</RelatedForamtID>
                    <FormatName>Rich Text Format</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>627</RelatedForamtID>
                    <FormatName>Rich Text Format</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>628</RelatedForamtID>
                    <FormatName>Rich Text Format</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>629</RelatedForamtID>
                    <FormatName>Rich Text Format</FormatName>
                </FileFormat>
                <FileFormat>
                    <RelatedForamtID>630</RelatedForamtID>
                    <FormatName>Rich Text Format</FormatName>
                </FileFormat>
```

```
                    <FileFormat>
                        <RelatedForamtID>631</RelatedForamtID>
                        <FormatName>Rich Text Format</FormatName>
                    </FileFormat>
                    <FileFormat>
                        <RelatedForamtID>632</RelatedForamtID>
                        <FormatName>Rich Text Format</FormatName>
                    </FileFormat>
                    <FileFormat>
                        <RelatedForamtID>633</RelatedForamtID>
                        <FormatName>Rich Text Format</FormatName>
                    </FileFormat>
                    <FileFormat>
                        <RelatedForamtID>821</RelatedForamtID>
                        <FormatName>Haansoft Hangul</FormatName>
                    </FileFormat>
                </Render>
                <Validate>
                </Validate>
                <etc>
                </etc>
            </ProcessFormat>
            <Language>Korean</Language>
            <Feature/>
            <Image/>
            <Corporation>
                <Individual>
                    <Name>(주)한글과컴퓨터</Name>
                    <Address>서울시 광진구 구의동 프라임 센터 20층</Address>
                    <Tel>+82 02-3424-3400</Tel>
                    <Fax>+82 02-3424-4062</Fax>
                    <E-mail>hncxml@haansoft.com</E-mail>
                    <Web>http://www.haansoft.com</Web>
                </Individual>
            </Corporation>
            <Support>
                <Individual>
                    <Name>(주)한글과컴퓨터</Name>
                    <Address>서울시 광진구 구의동 프라임 센터 20층</Address>
                    <Tel>+82 02-3424-3400</Tel>
                    <Fax>+82 02-3424-4062</Fax>
                    <E-mail>hncxml@haansoft.com</E-mail>
                    <Web>http://www.haansoft.com</Web>
                </Individual>
            </Support>
            <ReleaseDate>31 May 2001</ReleaseDate>
            <WithdrawDate>
          </WithdrawDate>
            <IRPList>

            </IRPList>
            <Note/>
            <Description/>
        </General>
        <Technical>
            <RelatedSofftware/>
            <SoftwareRequirement/>
            <HardwareRequirement/>
            <MediaFormat/>
            <Specification>
                <Document>
                    <Title/>
                    <Type/>
                    <Author>
                        <Individual>
                            <Name/>
                            <Address/>
                            <Tel/>
                            <Fax/>
                            <E-mail/>
                            <Web/>
                        </Individual>
                    </Author>
                    <Publisher>
                        <Individual>
                            <Name/>
                            <Address/>
```

```
                        <Tel/>
                        <Fax/>
                        <E-mail/>
                        <Web/>
                    </Individual>
                </Publisher>
                <Edition/>
                <Date/>
                <Enumeration/>
                <Page/>
                <Identifier/>
                <Note/>
            </Document>
        </Specification>
    </Technical>
    <Management>
        <Source>
            <Individual>
                <Name>신은주</Name>
                <Address>서울특별시 용산구 효창원길 숙명여자대학교</Address>
                <Tel>02-710-9379</Tel>
                <Fax>02-710-9704</Fax>
                <E-mail>kiki75@gmail.com</E-mail>
                <Web>http://www.sookmyung.ac.kr</Web>
            </Individual>
        </Source>
        <Date>2008-11-19 15:43:43</Date>
        <LastUpdate>2008-11-19 15:43:43</LastUpdate>
        <Type/>
        <Note/>
    </Management>
</Software>
```

## 다. 영국 국가기록원의 PRONOM & DROID 구조 분석[99)]

### (1) PRONOM & DROID 구성 정의

❒ PRONOM 개요

- ○ 영국 국가기록원(The National Archive : TNA)에서 다양한 포맷의 전자 기록을 장기 보존하는데 활용하기 위하여 2002년 3월 내부 사용을 위해 개발한 후 2004년 2월 무료 온라인 서비스로 공개되었음. 계속 데이터베이스 및 소프트웨어가 업데이트 되고 있으며 2011년 9월 현재 약 820개의 포맷 정보가 저장되어 있으며 http://www.nationalarchives.gov.uk/PRONOM/Default.aspx 를 통해 서비스 되고 있음
- ○ PRONOM 사이트에서 확장자나 소프트웨어명으로 관련 포맷명과 버전을 검색할 수 있으며, 특정 날짜에 지원되거나 지원되지 않는 포맷을 찾아볼 수 있음.
- ○ 미국 Harvard 대학교 도서관과 NARA(The US National Archives and Records Administration) 그리고 OCLC(Online Computer Library Center)의 협력 하에 추진되어 온 GDFR(Global Digital Format Registry)와 통합하여 UDFR(Unified Digital Formats Registry)를 구축하려는 제안이 2009년에 계획되어 2012년까지 추진 예정임. 구축될 UDFR의 포맷 데이터베이스는 PRONOM

99) 국가기록원, 전자기록물 검증 기술 및 차세대 그린 전자기록관리체제 인프라 응용 기술 연구 완료보고서

의 데이터를 기반으로 구축될 예정

❒ PRONOM 구성 요소

○ PRONOM을 구성하는 3개 요소 '포맷 관련 데이터베이스', 'DROID', 'PUID' 임

○ DROID(Digital Record Object Identification) : PRONOM에서 제공되는 포맷 정보를 이용하여 파일의 포맷을 식별하는 별도의 소프트웨어

○ PUID(Persistent Unique Identifier) : PRONOM 레지스트리의 기록을 지속적이고 유일하며 명확히 식별하기 위한 식별 체계

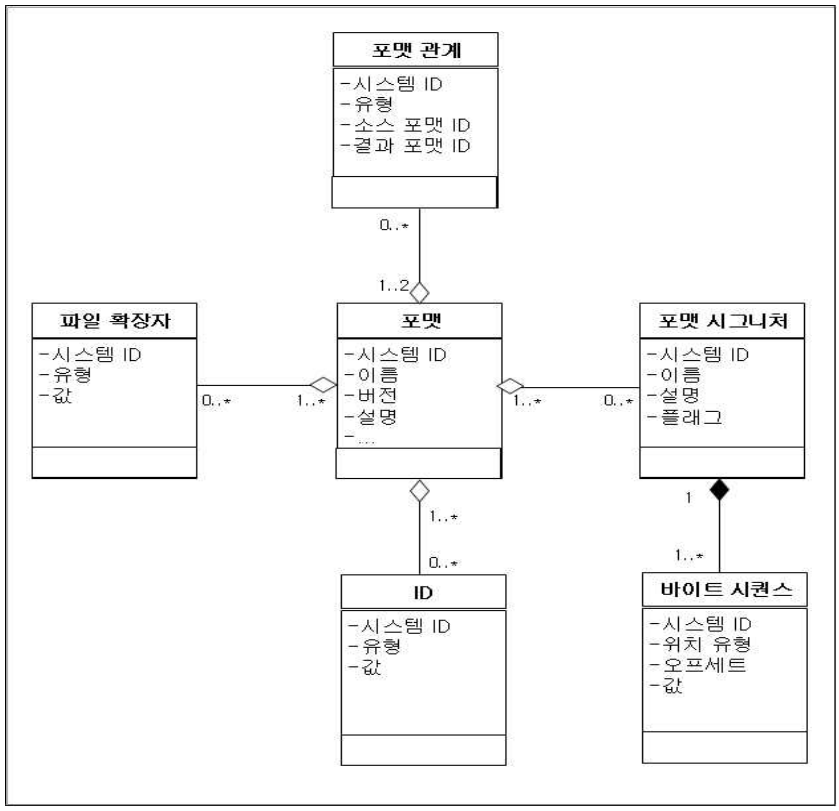
❒ 포맷과 시그니처 관계(Relationship between formats and signature)

○ DROID에서는 포맷 특성 정보가 비트스트림 외부에 존재하느냐 내부에 존재하느냐에 따라 외장 시그니처와 내장 시그니처로 구분됨

○ 파일 외부에 존재하는 외장 시그니처란 디지털 객체의 비트스트림 바깥에 존재하는 포맷 특성 데이터로서 Windows의 경우 파일 확장자, MacOS의 경우 데이터 폭스(data forks)가 외장 시그니처에 해당됨

○ 반면 내장 포맷 시그니처란 디지털 객체의 비트스트림 속의 특정 위치에 존재하는 포맷에 대한 특성 값임

○ 포맷과 시그니처와의 관계를 살펴보면, 각 포맷은 다수의 내장 시그니처 또는 파일 확장자를 가지며 각 내장 시그니처는 1개 이상의 바이트 시퀀스로 구성됨. 포맷에는 식별자가 부여되고 포맷과 포맷 사이에는 수많은 관계를 가질 수 있음




<그림 3-128> 포맷과 시그니처 관계
(Relationship between formats and signature)

❒ 파일 확장자

○ 디지털 객체의 비트스트림으로부터 얻어야 하는 포맷 시그니처(DROID에서는 내장 시그니처)와 달리 외부에 존재하는 특성 정보로서 윈도우나 DOS의 확장자가 여기 해당됨

○ 특성 값을 얻기는 쉬우나 파일 확장자는 파일 포맷 식별 정보로서 불충분한데, 첫째, 파일 확장자와 포맷의 관계는 1:1이 아니기 때문에 특정 포맷을 정의하는 유일한 수단이 될 수 없고, 둘째, 상세히 포맷 버전을 표시할 수 없는 문제가 있으며, 셋째, 사용자가 확장자를 임의대로 바꿀 수 있기 때문에 신뢰성이 떨어짐

○ 따라서 파일 확장자는 포맷을 대략적으로 감지하는 용도로 적당하며 그 이상 신뢰할 수 없음

❒ 포맷 시그니처

○ 동일한 포맷 디지털 객체의 비트스트림 내에는 일관되게 특정 구조가 존재하는데 이 구조를 판별해 냄으로써 포맷을 판단할 수 있음. 이 구조를 포맷 시그니처(DROID에서는 내장 시그니처)라고 하며 연속된 16진수 값과 정규식으로 이루어진 1개 이상의 바이트 열로 구성됨

○ 바이트 열은 비트스트림 내에서의 시작 위치와 그 위치에 존재하는 16진수 값으로 정의함. 시작 위치에는 2가지 유형이 있음

- 절대 위치 : 비트스트림 내 고정된 위치에서 바이트 열 시작. 파일의 BOF(Begin of File)이나 EOF(End of File) 중 한 지점으로부터 몇 번째 바이트인지로 표현되며 해당 바이트로 이동하여 값을 얻어냄
- 유동 위치 : 비트스트림 내 어디에서든 바이트 열이 시작될 수 있으므로 전체 비트스트림을 체크해야 찾아 낼 수 있음

○ 파일 확장자는 포맷 식별에 불충분하며 식별 도구는 포맷 시그니처를 이용해야 함. 따라서 식별 도구가 포맷 시그니처에 대한 정보를 신속히 획득하고 최신 정보로 유지하기 위한 수단을 갖추고 있느냐 역시 식별 도구 개발에서 고려해야 할 것임

○ 국내외 포맷 레지스트리 및 식별 도구 사례에서 보듯이 각 포맷별 포맷 시그니처에 대한 정보는 포맷 레지스트리에 저장하여 관리하며 식별 도구는 레지스트리로부터 이 정보를 얻어 식별에 사용하는 체계를 갖추고 있음

### (2) PRONOM에 등록된 XML 데이터 분석

❒ PRONOM에 등록된 PPT 포맷 정보

<표 3-49> PRONOM에 등록된 PPT 포맷 정보

```
<?xml version="1.0" encoding="UTF-8"?>
<FFSignatureFile DateCreated="2013-08-28T10:17:01" Version="70"
xmlns="http://www.nationalarchives.gov.uk/pronom/SignatureFile">
    <InternalSignatureCollection>
        <InternalSignature ID="133" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>4175746F4341442042696E617279204458460D0A1A00</Sequence>
                    <DefaultShift>23</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="0A">3</Shift>
                    <Shift Byte="0D">4</Shift>
                    <Shift Byte="1A">2</Shift>
                    <Shift Byte="20">8</Shift>
                    <Shift Byte="41">17</Shift>
                    <Shift Byte="42">14</Shift>
                    <Shift Byte="43">18</Shift>
                    <Shift Byte="44">7</Shift>
                    <Shift Byte="46">5</Shift>
                    <Shift Byte="58">6</Shift>
                    <Shift Byte="61">11</Shift>
                    <Shift Byte="69">13</Shift>
                    <Shift Byte="6E">12</Shift>
                    <Shift Byte="6F">19</Shift>
                    <Shift Byte="72">10</Shift>
                    <Shift Byte="74">20</Shift>
                    <Shift Byte="75">21</Shift>
                    <Shift Byte="79">9</Shift>
                </SubSequence>
                <SubSequence MinFragLength="0" Position="2" SubSeqMinOffset="0">
                    <Sequence>0053454354494F4E00024845414445200</Sequence>
                    <DefaultShift>18</DefaultShift>
                    <Shift Byte="00">1</Shift>
```

```
                    <Shift Byte="02">8</Shift>
                    <Shift Byte="41">5</Shift>
                    <Shift Byte="43">14</Shift>
                    <Shift Byte="44">4</Shift>
                    <Shift Byte="45">3</Shift>
                    <Shift Byte="48">7</Shift>
                    <Shift Byte="49">12</Shift>
                    <Shift Byte="4E">10</Shift>
                    <Shift Byte="4F">11</Shift>
                    <Shift Byte="52">2</Shift>
                    <Shift Byte="53">16</Shift>
                    <Shift Byte="54">13</Shift>
                </SubSequence>
                <SubSequence MinFragLength="0" Position="3" SubSeqMinOffset="0">
                    <Sequence>092441434144564552000141433130303600</Sequence>
                    <DefaultShift>19</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="01">8</Shift>
                    <Shift Byte="09">18</Shift>
                    <Shift Byte="24">17</Shift>
                    <Shift Byte="30">3</Shift>
                    <Shift Byte="31">5</Shift>
                    <Shift Byte="36">2</Shift>
                    <Shift Byte="41">7</Shift>
                    <Shift Byte="43">6</Shift>
                    <Shift Byte="44">13</Shift>
                    <Shift Byte="45">11</Shift>
                    <Shift Byte="52">10</Shift>
                    <Shift Byte="56">12</Shift>
                </SubSequence>
                <SubSequence MinFragLength="0" Position="4" SubSeqMinOffset="0">
                    <Sequence>00454E4453454300</Sequence>
                    <DefaultShift>9</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="43">2</Shift>
                    <Shift Byte="44">5</Shift>
                    <Shift Byte="45">3</Shift>
                    <Shift Byte="4E">6</Shift>
                    <Shift Byte="53">4</Shift>
                </SubSequence>
            </ByteSequence>
            <ByteSequence Reference="EOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>00454F4600</Sequence>
                    <DefaultShift>-6</DefaultShift>
                    <Shift Byte="00">-1</Shift>
                    <Shift Byte="45">-2</Shift>
                    <Shift Byte="46">-4</Shift>
                    <Shift Byte="4F">-3</Shift>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="134" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>4175746F4341442042696E617279204458460D0A1A00</Sequence>
                    <DefaultShift>23</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="0A">3</Shift>
                    <Shift Byte="0D">4</Shift>
                    <Shift Byte="1A">2</Shift>
                    <Shift Byte="20">8</Shift>
                    <Shift Byte="41">17</Shift>
                    <Shift Byte="42">14</Shift>
                    <Shift Byte="43">18</Shift>
                    <Shift Byte="44">7</Shift>
                    <Shift Byte="46">5</Shift>
                    <Shift Byte="58">6</Shift>
                    <Shift Byte="61">11</Shift>
                    <Shift Byte="69">13</Shift>
                    <Shift Byte="6E">12</Shift>
                    <Shift Byte="6F">19</Shift>
                    <Shift Byte="72">10</Shift>
                    <Shift Byte="74">20</Shift>
                    <Shift Byte="75">21</Shift>
```

```
                    <Shift Byte="79">9</Shift>
                </SubSequence>
                <SubSequence MinFragLength="0" Position="2" SubSeqMinOffset="0">
                    <Sequence>0053454354494F4E00024845414445520 0</Sequence>
                    <DefaultShift>18</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="02">8</Shift>
                    <Shift Byte="41">5</Shift>
                    <Shift Byte="43">14</Shift>
                    <Shift Byte="44">4</Shift>
                    <Shift Byte="45">3</Shift>
                    <Shift Byte="48">7</Shift>
                    <Shift Byte="49">12</Shift>
                    <Shift Byte="4E">10</Shift>
                    <Shift Byte="4F">11</Shift>
                    <Shift Byte="52">2</Shift>
                    <Shift Byte="53">16</Shift>
                    <Shift Byte="54">13</Shift>
                </SubSequence>
                <SubSequence MinFragLength="0" Position="3" SubSeqMinOffset="0">
                    <Sequence>0924414341445645520001414331303039 00</Sequence>
                    <DefaultShift>19</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="01">8</Shift>
                    <Shift Byte="09">18</Shift>
                    <Shift Byte="24">17</Shift>
                    <Shift Byte="30">3</Shift>
                    <Shift Byte="31">5</Shift>
                    <Shift Byte="39">2</Shift>
                    <Shift Byte="41">7</Shift>
                    <Shift Byte="43">6</Shift>
                    <Shift Byte="44">13</Shift>
                    <Shift Byte="45">11</Shift>
                    <Shift Byte="52">10</Shift>
                    <Shift Byte="56">12</Shift>
                </SubSequence>
                <SubSequence MinFragLength="0" Position="4" SubSeqMinOffset="0">
                    <Sequence>00454E4453454300</Sequence>
                    <DefaultShift>9</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="43">2</Shift>
                    <Shift Byte="44">5</Shift>
                    <Shift Byte="45">3</Shift>
                    <Shift Byte="4E">6</Shift>
                    <Shift Byte="53">4</Shift>
                </SubSequence>
            </ByteSequence>
            <ByteSequence Reference="EOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>00454F4600</Sequence>
                    <DefaultShift>-6</DefaultShift>
                    <Shift Byte="00">-1</Shift>
                    <Shift Byte="45">-2</Shift>
                    <Shift Byte="46">-4</Shift>
                    <Shift Byte="4F">-3</Shift>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="135" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>4175746F4341442042696E617279204458460D0A1A00</Sequence>
                    <DefaultShift>23</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="0A">3</Shift>
                    <Shift Byte="0D">4</Shift>
                    <Shift Byte="1A">2</Shift>
                    <Shift Byte="20">8</Shift>
                    <Shift Byte="41">17</Shift>
                    <Shift Byte="42">14</Shift>
                    <Shift Byte="43">18</Shift>
                    <Shift Byte="44">7</Shift>
                    <Shift Byte="46">5</Shift>
                    <Shift Byte="58">6</Shift>
                    <Shift Byte="61">11</Shift>
```

```
                    <Shift Byte="69">13</Shift>
                    <Shift Byte="6E">12</Shift>
                    <Shift Byte="6F">19</Shift>
                    <Shift Byte="72">10</Shift>
                    <Shift Byte="74">20</Shift>
                    <Shift Byte="75">21</Shift>
                    <Shift Byte="79">9</Shift>
                </SubSequence>
                <SubSequence MinFragLength="0" Position="2" SubSeqMinOffset="0">
                    <Sequence>0053454354494F4E000248454144455200</Sequence>
                    <DefaultShift>18</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="02">8</Shift>
                    <Shift Byte="41">5</Shift>
                    <Shift Byte="43">14</Shift>
                    <Shift Byte="44">4</Shift>
                    <Shift Byte="45">3</Shift>
                    <Shift Byte="48">7</Shift>
                    <Shift Byte="49">12</Shift>
                    <Shift Byte="4E">10</Shift>
                    <Shift Byte="4F">11</Shift>
                    <Shift Byte="52">2</Shift>
                    <Shift Byte="53">16</Shift>
                    <Shift Byte="54">13</Shift>
                </SubSequence>
                <SubSequence MinFragLength="0" Position="3" SubSeqMinOffset="0">
                    <Sequence>0924414341445645520001414331303132 00</Sequence>
                    <DefaultShift>19</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="01">8</Shift>
                    <Shift Byte="09">18</Shift>
                    <Shift Byte="24">17</Shift>
                    <Shift Byte="30">4</Shift>
                    <Shift Byte="31">3</Shift>
                    <Shift Byte="32">2</Shift>
                    <Shift Byte="41">7</Shift>
                    <Shift Byte="43">6</Shift>
                    <Shift Byte="44">13</Shift>
                    <Shift Byte="45">11</Shift>
                    <Shift Byte="52">10</Shift>
                    <Shift Byte="56">12</Shift>
                </SubSequence>
                <SubSequence MinFragLength="0" Position="4" SubSeqMinOffset="0">
                    <Sequence>00454E4453454300</Sequence>
                    <DefaultShift>9</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="43">2</Shift>
                    <Shift Byte="44">5</Shift>
                    <Shift Byte="45">3</Shift>
                    <Shift Byte="4E">6</Shift>
                    <Shift Byte="53">4</Shift>
                </SubSequence>
            </ByteSequence>
            <ByteSequence Reference="EOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>00454F4600</Sequence>
                    <DefaultShift>-6</DefaultShift>
                    <Shift Byte="00">-1</Shift>
                    <Shift Byte="45">-2</Shift>
                    <Shift Byte="46">-4</Shift>
                    <Shift Byte="4F">-3</Shift>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
            <InternalSignature ID="885" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="57" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">

<Sequence>25252B2070726F637365742041646F62655F496C6C7573747261746F72</Sequence>
                    <DefaultShift>30</DefaultShift>
                    <Shift Byte="20">18</Shift>
                    <Shift Byte="25">28</Shift>
                    <Shift Byte="2B">27</Shift>
                    <Shift Byte="41">17</Shift>
                    <Shift Byte="49">11</Shift>
```

```
                        <Shift Byte="5F">12</Shift>
                        <Shift Byte="61">4</Shift>
                        <Shift Byte="62">14</Shift>
                        <Shift Byte="63">22</Shift>
                        <Shift Byte="64">16</Shift>
                        <Shift Byte="65">13</Shift>
                        <Shift Byte="6C">9</Shift>
                        <Shift Byte="6F">2</Shift>
                        <Shift Byte="70">25</Shift>
                        <Shift Byte="72">1</Shift>
                        <Shift Byte="73">7</Shift>
                        <Shift Byte="74">3</Shift>
                        <Shift Byte="75">8</Shift>
                        <LeftFragment MaxOffset="512" MinOffset="1"
Position="1">2525446F63756D656E744E6565646565645265736F75726365733A</LeftFragment>
                        <LeftFragment MaxOffset="512" MinOffset="16"
Position="2">252150532D41646F62652D322E30</LeftFragment>
                        <RightFragment MaxOffset="0" MinOffset="0"
Position="1">415F414933</RightFragment>
                        <RightFragment MaxOffset="0" MinOffset="0"
Position="1">5F414933</RightFragment>
                    </SubSequence>
                </ByteSequence>
            </InternalSignature>
            <InternalSignature ID="886" Specificity="Specific">
                <ByteSequence Reference="BOFoffset">
                    <SubSequence MinFragLength="59" Position="1"
                        SubSeqMaxOffset="0" SubSeqMinOffset="0">

<Sequence>25252B2070726F637365742041646F62655F496C6C7573747261746F72</Sequence>
                        <DefaultShift>30</DefaultShift>
                        <Shift Byte="20">18</Shift>
                        <Shift Byte="25">28</Shift>
                        <Shift Byte="2B">27</Shift>
                        <Shift Byte="41">17</Shift>
                        <Shift Byte="49">11</Shift>
                        <Shift Byte="5F">12</Shift>
                        <Shift Byte="61">4</Shift>
                        <Shift Byte="62">14</Shift>
                        <Shift Byte="63">22</Shift>
                        <Shift Byte="64">16</Shift>
                        <Shift Byte="65">13</Shift>
                        <Shift Byte="6C">9</Shift>
                        <Shift Byte="6F">2</Shift>
                        <Shift Byte="70">25</Shift>
                        <Shift Byte="72">1</Shift>
                        <Shift Byte="73">7</Shift>
                        <Shift Byte="74">3</Shift>
                        <Shift Byte="75">8</Shift>
                        <LeftFragment MaxOffset="512" MinOffset="1"
Position="1">2525446F63756D656E74537570706C6965645265736F75726365733A</LeftFragment>
                        <LeftFragment MaxOffset="512" MinOffset="16"
Position="2">252150532D41646F62652D322E30</LeftFragment>
                        <RightFragment MaxOffset="0" MinOffset="0"
Position="1">415F414933</RightFragment>
                        <RightFragment MaxOffset="0" MinOffset="0"
Position="1">5F414933</RightFragment>
                    </SubSequence>
                </ByteSequence>
            </InternalSignature>
            <InternalSignature ID="887" Specificity="Specific">
                <ByteSequence Reference="BOFoffset">
                    <SubSequence MinFragLength="15" Position="1"
                        SubSeqMaxOffset="0" SubSeqMinOffset="0">

<Sequence>252543726561746F723A2041646F626520496C6C7573747261746F7228544D2920666F722057
696E646F77732C2076657273696F6E20342E30</Sequence>
                        <DefaultShift>58</DefaultShift>
                        <Shift Byte="20">4</Shift>
                        <Shift Byte="25">56</Shift>
                        <Shift Byte="28">29</Shift>
                        <Shift Byte="29">26</Shift>
                        <Shift Byte="2C">13</Shift>
                        <Shift Byte="2E">2</Shift>
                        <Shift Byte="30">1</Shift>
                        <Shift Byte="34">3</Shift>
                        <Shift Byte="3A">48</Shift>
```

```
                        <Shift Byte="41">46</Shift>
                        <Shift Byte="43">55</Shift>
                        <Shift Byte="49">40</Shift>
                        <Shift Byte="4D">27</Shift>
                        <Shift Byte="54">28</Shift>
                        <Shift Byte="57">20</Shift>
                        <Shift Byte="61">33</Shift>
                        <Shift Byte="62">43</Shift>
                        <Shift Byte="64">17</Shift>
                        <Shift Byte="65">10</Shift>
                        <Shift Byte="66">24</Shift>
                        <Shift Byte="69">7</Shift>
                        <Shift Byte="6C">38</Shift>
                        <Shift Byte="6E">5</Shift>
                        <Shift Byte="6F">6</Shift>
                        <Shift Byte="72">9</Shift>
                        <Shift Byte="73">8</Shift>
                        <Shift Byte="74">32</Shift>
                        <Shift Byte="75">37</Shift>
                        <Shift Byte="76">11</Shift>
                        <Shift Byte="77">15</Shift>
                        <LeftFragment MaxOffset="128" MinOffset="1"
Position="1">252150532D41646F62652D332E30</LeftFragment>
                    </SubSequence>
                </ByteSequence>
            </InternalSignature>
            <InternalSignature ID="888" Specificity="Specific">
                <ByteSequence Reference="BOFoffset">
                    <SubSequence MinFragLength="142" Position="1"
                        SubSeqMaxOffset="0" SubSeqMinOffset="0">
                        <Sequence>254149355F46696C65466F726D61742031</Sequence>
                        <DefaultShift>18</DefaultShift>
                        <Shift Byte="20">2</Shift>
                        <Shift Byte="25">17</Shift>
                        <Shift Byte="31">1</Shift>
                        <Shift Byte="35">14</Shift>
                        <Shift Byte="41">16</Shift>
                        <Shift Byte="46">8</Shift>
                        <Shift Byte="49">15</Shift>
                        <Shift Byte="5F">13</Shift>
                        <Shift Byte="61">4</Shift>
                        <Shift Byte="65">9</Shift>
                        <Shift Byte="69">11</Shift>
                        <Shift Byte="6C">10</Shift>
                        <Shift Byte="6D">5</Shift>
                        <Shift Byte="6F">7</Shift>
                        <Shift Byte="72">6</Shift>
                        <Shift Byte="74">3</Shift>
                        <LeftFragment MaxOffset="1024" MinOffset="128"
Position="1">252150532D41646F62652D332E30</LeftFragment>
                    </SubSequence>
                </ByteSequence>
            </InternalSignature>
        </InternalSignatureCollection>
        <FileFormatCollection>
            <FileFormat ID="133" MIMEType="application/vnd.ms-powerpoint"
                Name="Microsoft Powerpoint Presentation" PUID="x-fmt/88" Version="4.0">
                <InternalSignatureID>181</InternalSignatureID>
                <Extension>ppt</Extension>
                <HasPriorityOverFileFormatID>135</HasPriorityOverFileFormatID>
                <HasPriorityOverFileFormatID>767</HasPriorityOverFileFormatID>
            </FileFormat>
            <FileFormat ID="134" MIMEType="application/vnd.ms-powerpoint"
                Name="Microsoft Powerpoint Presentation" PUID="fmt/125" Version="95">
                <InternalSignatureID>171</InternalSignatureID>
                <Extension>ppt</Extension>
                <HasPriorityOverFileFormatID>135</HasPriorityOverFileFormatID>
                <HasPriorityOverFileFormatID>767</HasPriorityOverFileFormatID>
            </FileFormat>
            <FileFormat ID="135" MIMEType="application/vnd.ms-powerpoint"
                Name="Microsoft Powerpoint Presentation" PUID="fmt/126" Version="97-2002">
                <InternalSignatureID>172</InternalSignatureID>
                <Extension>ppt</Extension>
                <HasPriorityOverFileFormatID>767</HasPriorityOverFileFormatID>
            </FileFormat>
            <FileFormat ID="885" Name="Microsoft PowerPoint for Macintosh"
                PUID="fmt/179" Version="4.0">
```

```
            <Extension>ppt</Extension>
        </FileFormat>
        <FileFormat ID="886" Name="Microsoft PowerPoint for Macintosh"
            PUID="fmt/180" Version="98">
            <Extension>ppt</Extension>
        </FileFormat>
        <FileFormat ID="887" Name="Microsoft PowerPoint for Macintosh"
            PUID="fmt/181" Version="2001">
            <Extension>ppt</Extension>
        </FileFormat>
        <FileFormat ID="888" Name="Microsoft PowerPoint for Macintosh"
            PUID="fmt/182" Version="X">
            <Extension>ppt</Extension>
        </FileFormat>
        <FileFormat ID="941" MIMEType="application/vnd.ms-powerpoint"
            Name="Microsoft Powerpoint for Windows" PUID="fmt/215" Version="2007 onwards">
            <Extension>pptx</Extension>
        </FileFormat>
        <FileFormat ID="1274" Name="Macro Enabled Microsoft Powerpoint"
            PUID="fmt/487" Version="2007 Onwards">
            <Extension>pptm</Extension>
        </FileFormat>
        <FileFormat ID="1281" Name="Microsoft Office Encrypted Document"
            PUID="fmt/494" Version="2007 Onwards">
            <Extension>docx</Extension>
            <Extension>pptx</Extension>
            <Extension>xslx</Extension>
        </FileFormat>
    </FileFormatCollection>
</FFSignatureFile>
```

○ PRONOM에 등록된 XML 데이터는 기술정보은행(DFR)의 XMLSchema와 비교할 때 정의된 태그의 개수는 적지만 InternalSignature와 ByteSequence 태그 정보가 빠짐없이 등록되어 있음을 확인

○ 정보 수집 및 관리의 용이성을 위해 태그에 대한 정의를 확장하는 것보다 정확한 정보를 등록 관리하는 것이 중요하다고 판단됨

❒ PRONOM에 등록된 XLS 포맷 정보

<표 3-50> PRONOM에 등록된 XLS 포맷 정보

```
<?xml version="1.0" encoding="UTF-8"?>
<FFSignatureFile DateCreated="2013-08-28T10:17:01" Version="70"
xmlns="http://www.nationalarchives.gov.uk/pronom/SignatureFile">
    <InternalSignatureCollection>
        <InternalSignature ID="677" Specificity="Specific">
            <ByteSequence Endianness="Little-endian" Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>49492A00</Sequence>
                    <DefaultShift>5</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="2A">2</Shift>
                    <Shift Byte="49">3</Shift>
                    <RightFragment MaxOffset="4080" MinOffset="4" Position="1">AF870300
</RightFragment>
                    <RightFragment MaxOffset="1" MinOffset="1" Position="2">000000
</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="678" Specificity="Specific">
            <ByteSequence Endianness="Little-endian" Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
```

```
                    <Sequence>49492A00</Sequence>
                    <DefaultShift>5</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="2A">2</Shift>
                    <Shift Byte="49">3</Shift>
                </SubSequence>
            </ByteSequence>
            <ByteSequence Endianness="Little-endian" Reference="EOFoffset">
                <SubSequence MinFragLength="4" Position="1"
                    SubSeqMaxOffset="4092" SubSeqMinOffset="4">
                    <Sequence>AF870300</Sequence>
                    <DefaultShift>-5</DefaultShift>
                    <Shift Byte="00">-4</Shift>
                    <Shift Byte="03">-3</Shift>
                    <Shift Byte="87">-2</Shift>
                    <Shift Byte="AF">-1</Shift>
                    <RightFragment MaxOffset="1" MinOffset="1" Position="1">000000
</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="679" Specificity="Specific">
            <ByteSequence Endianness="Big-endian" Reference="BOFoffset">
                <SubSequence MinFragLength="8" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>87AF0003000000</Sequence>
                    <DefaultShift>8</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="03">4</Shift>
                    <Shift Byte="87">7</Shift>
                    <Shift Byte="AF">6</Shift>
                    <LeftFragment MaxOffset="4080" MinOffset="4" Position="1">4D4D002A
</LeftFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="680" Specificity="Specific">
            <ByteSequence Endianness="Big-endian" Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>4D4D002A</Sequence>
                    <DefaultShift>5</DefaultShift>
                    <Shift Byte="00">2</Shift>
                    <Shift Byte="2A">1</Shift>
                    <Shift Byte="4D">3</Shift>
                </SubSequence>
            </ByteSequence>
            <ByteSequence Reference="EOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="4094" SubSeqMinOffset="5">
                    <Sequence>87AF0003000000</Sequence>
                    <DefaultShift>-8</DefaultShift>
                    <Shift Byte="00">-3</Shift>
                    <Shift Byte="03">-4</Shift>
                    <Shift Byte="87">-1</Shift>
                    <Shift Byte="AF">-2</Shift>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="681" Specificity="Specific">
            <ByteSequence Endianness="Little-endian" Reference="BOFoffset">
                <SubSequence MinFragLength="10" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>FE00040001000000</Sequence>
                    <DefaultShift>9</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="01">4</Shift>
                    <Shift Byte="04">6</Shift>
                    <Shift Byte="FE">8</Shift>
                    <LeftFragment MaxOffset="4084" MinOffset="6" Position="1">49492A00
</LeftFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="682" Specificity="Specific">
            <ByteSequence Endianness="Little-endian" Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
```

```
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>49492A00</Sequence>
                    <DefaultShift>5</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="2A">2</Shift>
                    <Shift Byte="49">3</Shift>
                </SubSequence>
            </ByteSequence>
            <ByteSequence Endianness="Little-endian" Reference="EOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="4088" SubSeqMinOffset="0">
                    <Sequence>FE00040001000000</Sequence>
                    <DefaultShift>-9</DefaultShift>
                    <Shift Byte="00">-2</Shift>
                    <Shift Byte="01">-5</Shift>
                    <Shift Byte="04">-3</Shift>
                    <Shift Byte="FE">-1</Shift>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="683" Specificity="Specific">
            <ByteSequence Endianness="Big-endian" Reference="BOFoffset">
                <SubSequence MinFragLength="10" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>00FE000400000001</Sequence>
                    <DefaultShift>9</DefaultShift>
                    <Shift Byte="00">2</Shift>
                    <Shift Byte="01">1</Shift>
                    <Shift Byte="04">5</Shift>
                    <Shift Byte="FE">7</Shift>
                    <LeftFragment MaxOffset="4084" MinOffset="6" Position="1">4D4D002A
</LeftFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="684" Specificity="Specific">
            <ByteSequence Endianness="Big-endian" Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>4D4D002A</Sequence>
                    <DefaultShift>5</DefaultShift>
                    <Shift Byte="00">2</Shift>
                    <Shift Byte="2A">1</Shift>
                    <Shift Byte="4D">3</Shift>
                </SubSequence>
            </ByteSequence>
            <ByteSequence Endianness="Big-endian" Reference="EOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="4088" SubSeqMinOffset="0">
                    <Sequence>00FE000400000001</Sequence>
                    <DefaultShift>-9</DefaultShift>
                    <Shift Byte="00">-1</Shift>
                    <Shift Byte="01">-8</Shift>
                    <Shift Byte="04">-4</Shift>
                    <Shift Byte="FE">-2</Shift>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="685" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1">
                    <Sequence>3026B2758E66CF11A6D900AA0062CE6C</Sequence>
                    <DefaultShift>17</DefaultShift>
                    <Shift Byte="00">4</Shift>
                    <Shift Byte="11">9</Shift>
                    <Shift Byte="26">15</Shift>
                    <Shift Byte="30">16</Shift>
                    <Shift Byte="62">3</Shift>
                    <Shift Byte="66">11</Shift>
                    <Shift Byte="6C">1</Shift>
                    <Shift Byte="75">13</Shift>
                    <Shift Byte="8E">12</Shift>
                    <Shift Byte="A6">8</Shift>
                    <Shift Byte="AA">5</Shift>
                    <Shift Byte="B2">14</Shift>
                    <Shift Byte="CE">2</Shift>
                    <Shift Byte="CF">10</Shift>
```

```
                    <Shift Byte="D9">7</Shift>
                    <RightFragment MaxOffset="12" MinOffset="12" Position="1">0102
</RightFragment>
                </SubSequence>
            </ByteSequence>
            <ByteSequence>
                <SubSequence MinFragLength="0" Position="1" SubSeqMinOffset="30">
                    <Sequence>9107DCB7B7A9CF118EE600C00C205365</Sequence>
                    <DefaultShift>17</DefaultShift>
                    <Shift Byte="00">6</Shift>
                    <Shift Byte="07">15</Shift>
                    <Shift Byte="0C">4</Shift>
                    <Shift Byte="11">9</Shift>
                    <Shift Byte="20">3</Shift>
                    <Shift Byte="53">2</Shift>
                    <Shift Byte="65">1</Shift>
                    <Shift Byte="8E">8</Shift>
                    <Shift Byte="91">16</Shift>
                    <Shift Byte="A9">11</Shift>
                    <Shift Byte="B7">12</Shift>
                    <Shift Byte="C0">5</Shift>
                    <Shift Byte="CF">10</Shift>
                    <Shift Byte="DC">14</Shift>
                    <Shift Byte="E6">7</Shift>
                    <RightFragment MaxOffset="8" MinOffset="8"
Position="1">C0EF19BC4D5BCF11A8FD00805F5C442B</RightFragment>
                    <RightFragment MaxOffset="65" MinOffset="65" Position="2">57564331
</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
            <InternalSignature ID="877" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>09000400</Sequence>
                    <DefaultShift>5</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="04">2</Shift>
                    <Shift Byte="09">4</Shift>
                    <RightFragment MaxOffset="2" MinOffset="2" Position="1">4000
</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="878" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>09020600</Sequence>
                    <DefaultShift>5</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="02">3</Shift>
                    <Shift Byte="06">2</Shift>
                    <Shift Byte="09">4</Shift>
                    <RightFragment MaxOffset="2" MinOffset="2" Position="1">4000
</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="879" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>09040600</Sequence>
                    <DefaultShift>5</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="04">3</Shift>
                    <Shift Byte="06">2</Shift>
                    <Shift Byte="09">4</Shift>
                    <RightFragment MaxOffset="2" MinOffset="2" Position="1">4000
</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="880" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
```

```
                    <SubSequence MinFragLength="0" Position="1"
                        SubSeqMaxOffset="0" SubSeqMinOffset="0">

<Sequence>252150532D41646F62652D322E3020455053462D312E320D252543726561746F723A41646F6
26520496C6C7573747261746F7228544D2920312E30</Sequence>
                        <DefaultShift>60</DefaultShift>
                        <Shift Byte="0D">36</Shift>
                        <Shift Byte="20">4</Shift>
                        <Shift Byte="21">58</Shift>
                        <Shift Byte="25">34</Shift>
                        <Shift Byte="28">8</Shift>
                        <Shift Byte="29">5</Shift>
                        <Shift Byte="2D">40</Shift>
                        <Shift Byte="2E">2</Shift>
                        <Shift Byte="30">1</Shift>
                        <Shift Byte="31">3</Shift>
                        <Shift Byte="32">37</Shift>
                        <Shift Byte="3A">26</Shift>
                        <Shift Byte="41">25</Shift>
                        <Shift Byte="43">33</Shift>
                        <Shift Byte="45">44</Shift>
                        <Shift Byte="46">41</Shift>
                        <Shift Byte="49">19</Shift>
                        <Shift Byte="4D">6</Shift>
                        <Shift Byte="50">43</Shift>
                        <Shift Byte="53">42</Shift>
                        <Shift Byte="54">7</Shift>
                        <Shift Byte="61">12</Shift>
                        <Shift Byte="62">22</Shift>
                        <Shift Byte="64">24</Shift>
                        <Shift Byte="65">21</Shift>
                        <Shift Byte="6C">17</Shift>
                        <Shift Byte="6F">10</Shift>
                        <Shift Byte="72">9</Shift>
                        <Shift Byte="73">15</Shift>
                        <Shift Byte="74">11</Shift>
                        <Shift Byte="75">16</Shift>
                    </SubSequence>
                </ByteSequence>
            </InternalSignature>
            <InternalSignature ID="881" Specificity="Specific">
                <ByteSequence Reference="BOFoffset">
                    <SubSequence MinFragLength="24" Position="1"
                        SubSeqMaxOffset="0" SubSeqMinOffset="0">
                        <Sequence>2525446F63756D656E7450726F63536574733A</Sequence>
                        <DefaultShift>20</DefaultShift>
                        <Shift Byte="25">18</Shift>
                        <Shift Byte="3A">1</Shift>
                        <Shift Byte="44">17</Shift>
                        <Shift Byte="50">9</Shift>
                        <Shift Byte="53">5</Shift>
                        <Shift Byte="63">6</Shift>
                        <Shift Byte="65">4</Shift>
                        <Shift Byte="6D">13</Shift>
                        <Shift Byte="6E">11</Shift>
                        <Shift Byte="6F">7</Shift>
                        <Shift Byte="72">8</Shift>
                        <Shift Byte="73">2</Shift>
                        <Shift Byte="74">3</Shift>
                        <Shift Byte="75">14</Shift>
                        <LeftFragment MaxOffset="2096" MinOffset="10"
Position="1">252150532D41646F62652D322E30</LeftFragment>
                        <RightFragment MaxOffset="0" MinOffset="0" Position="1">2041
</RightFragment>
                        <RightFragment MaxOffset="0" MinOffset="0" Position="1">41</RightFragment>
                        <RightFragment MaxOffset="0" MinOffset="0"
Position="2">646F62655F496C6C7573747261746F725F312E</RightFragment>
                        <RightFragment MaxOffset="0" MinOffset="0" Position="3">30</RightFragment>
                        <RightFragment MaxOffset="0" MinOffset="0" Position="3">31</RightFragment>
                        <RightFragment MaxOffset="0" MinOffset="0" Position="3">32</RightFragment>
                    </SubSequence>
                </ByteSequence>
            </InternalSignature>
            <InternalSignature ID="882" Specificity="Specific">
                <ByteSequence Reference="BOFoffset">
                    <SubSequence MinFragLength="25" Position="1"
                        SubSeqMaxOffset="0" SubSeqMinOffset="0">
```

```
                    <Sequence>41646F626520496C6C7573747261746F72</Sequence>
                    <DefaultShift>18</DefaultShift>
                    <Shift Byte="20">12</Shift>
                    <Shift Byte="41">17</Shift>
                    <Shift Byte="49">11</Shift>
                    <Shift Byte="61">4</Shift>
                    <Shift Byte="62">14</Shift>
                    <Shift Byte="64">16</Shift>
                    <Shift Byte="65">13</Shift>
                    <Shift Byte="6C">9</Shift>
                    <Shift Byte="6F">2</Shift>
                    <Shift Byte="72">1</Shift>
                    <Shift Byte="73">7</Shift>
                    <Shift Byte="74">3</Shift>
                    <Shift Byte="75">8</Shift>
                    <LeftFragment MaxOffset="0" MinOffset="0" Position="1">3A</LeftFragment>
                    <LeftFragment MaxOffset="0" MinOffset="0" Position="1">3A20</LeftFragment>
                    <LeftFragment MaxOffset="0" MinOffset="0" Position="2">252543726561746F72
</LeftFragment>
                    <LeftFragment MaxOffset="16" MinOffset="1"
Position="3">252150532D41646F62652D322E30</LeftFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="1">20383828544D29
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="1">28544D29203838
</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="883" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="33" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">

<Sequence>2525446F63756D656E7450726F63536574733A41646F62655F496C6C7573747261746F725F38
38</Sequence>
                    <DefaultShift>40</DefaultShift>
                    <Shift Byte="25">38</Shift>
                    <Shift Byte="38">1</Shift>
                    <Shift Byte="3A">21</Shift>
                    <Shift Byte="41">20</Shift>
                    <Shift Byte="44">37</Shift>
                    <Shift Byte="49">14</Shift>
                    <Shift Byte="50">29</Shift>
                    <Shift Byte="53">25</Shift>
                    <Shift Byte="5F">3</Shift>
                    <Shift Byte="61">7</Shift>
                    <Shift Byte="62">17</Shift>
                    <Shift Byte="63">26</Shift>
                    <Shift Byte="64">19</Shift>
                    <Shift Byte="65">16</Shift>
                    <Shift Byte="6C">12</Shift>
                    <Shift Byte="6D">33</Shift>
                    <Shift Byte="6E">31</Shift>
                    <Shift Byte="6F">5</Shift>
                    <Shift Byte="72">4</Shift>
                    <Shift Byte="73">10</Shift>
                    <Shift Byte="74">6</Shift>
                    <Shift Byte="75">11</Shift>
                    <LeftFragment MaxOffset="2096" MinOffset="10"
Position="1">252150532D41646F62652D322E3020455053462D312E32</LeftFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="884" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="33" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">

<Sequence>2525446F63756D656E7450726F63536574733A2041646F62655F496C6C7573747261746F72</
Sequence>
                    <DefaultShift>38</DefaultShift>
                    <Shift Byte="20">18</Shift>
                    <Shift Byte="25">36</Shift>
                    <Shift Byte="3A">19</Shift>
                    <Shift Byte="41">17</Shift>
                    <Shift Byte="44">35</Shift>
                    <Shift Byte="49">11</Shift>
```

```
                        <Shift Byte="50">27</Shift>
                        <Shift Byte="53">23</Shift>
                        <Shift Byte="5F">12</Shift>
                        <Shift Byte="61">4</Shift>
                        <Shift Byte="62">14</Shift>
                        <Shift Byte="63">24</Shift>
                        <Shift Byte="64">16</Shift>
                        <Shift Byte="65">13</Shift>
                        <Shift Byte="6C">9</Shift>
                        <Shift Byte="6D">31</Shift>
                        <Shift Byte="6E">29</Shift>
                        <Shift Byte="6F">2</Shift>
                        <Shift Byte="72">1</Shift>
                        <Shift Byte="73">7</Shift>
                        <Shift Byte="74">3</Shift>
                        <Shift Byte="75">8</Shift>
                        <LeftFragment MaxOffset="2096" MinOffset="10"
Position="1">252150532D41646F62652D322E3020455053462D312E32</LeftFragment>
                        <RightFragment MaxOffset="0" MinOffset="0" Position="1">3838
</RightFragment>
                        <RightFragment MaxOffset="0" MinOffset="0" Position="1">5F3838
</RightFragment>
                    </SubSequence>
                </ByteSequence>
            </InternalSignature>
        </InternalSignatureCollection>
        <FileFormatCollection>
            <FileFormat ID="677" MIMEType="image/jpeg"
                Name="Still Picture Interchange File Format" PUID="fmt/113" Version="2.0"/>
            <FileFormat ID="678" MIMEType="application/vnd.ms-excel"
                Name="Microsoft Excel 2.x Worksheet (xls)" PUID="fmt/55" Version="2">
                <InternalSignatureID>72</InternalSignatureID>
                <Extension>xls</Extension>
            </FileFormat>
            <FileFormat ID="679" MIMEType="application/vnd.ms-excel"
                Name="Microsoft Excel 3.0 Worksheet (xls)" PUID="fmt/56" Version="3">
                <InternalSignatureID>73</InternalSignatureID>
                <Extension>xls</Extension>
            </FileFormat>
            <FileFormat ID="680" MIMEType="application/vnd.ms-excel"
                Name="Microsoft Excel 4.0 Worksheet (xls)" PUID="fmt/57" Version="4S">
                <InternalSignatureID>74</InternalSignatureID>
                <Extension>xls</Extension>
            </FileFormat>
            <FileFormat ID="681" MIMEType="application/vnd.ms-excel"
                Name="Microsoft Excel 4.0 Workbook (xls)" PUID="fmt/58" Version="4W">
                <InternalSignatureID>75</InternalSignatureID>
                <Extension>xlw</Extension>
            </FileFormat>
            <FileFormat ID="682" MIMEType="application/vnd.ms-excel"
                Name="Microsoft Excel 5.0/95 Workbook (xls)" PUID="fmt/59" Version="5/95">
                <InternalSignatureID>76</InternalSignatureID>
                <Extension>xls</Extension>
                <Extension>xlw</Extension>
                <HasPriorityOverFileFormatID>767</HasPriorityOverFileFormatID>
            </FileFormat>
            <FileFormat ID="683" MIMEType="application/vnd.ms-excel"
                Name="Microsoft Excel 95 Workbook (xls)" PUID="fmt/60" Version="7">
                <Extension>xls</Extension>
                <HasPriorityOverFileFormatID>767</HasPriorityOverFileFormatID>
            </FileFormat>
            <FileFormat ID="684" MIMEType="application/vnd.ms-excel"
                Name="Microsoft Excel 97 Workbook (xls)" PUID="fmt/61" Version="8">
                <InternalSignatureID>78</InternalSignatureID>
                <Extension>xls</Extension>
                <Extension>xlw</Extension>
                <HasPriorityOverFileFormatID>682</HasPriorityOverFileFormatID>
                <HasPriorityOverFileFormatID>767</HasPriorityOverFileFormatID>
            </FileFormat>
            <FileFormat ID="685" MIMEType="application/vnd.ms-excel"
                Name="Microsoft Excel 2000-2003 Workbook (xls)"
                PUID="fmt/62" Version="8X">
                <InternalSignatureID>78</InternalSignatureID>
                <Extension>xls</Extension>
                <Extension>xlw</Extension>
                <HasPriorityOverFileFormatID>767</HasPriorityOverFileFormatID>
            </FileFormat>
```

```
    <FileFormat ID="877" Name="Microsoft Excel for Macintosh"
            PUID="fmt/172" Version="3.0">
            <Extension>xls</Extension>
        </FileFormat>
        <FileFormat ID="878" Name="Microsoft Excel for Macintosh"
            PUID="fmt/173" Version="4.0">
            <Extension>xls</Extension>
        </FileFormat>
        <FileFormat ID="879" Name="Microsoft Excel for Macintosh"
            PUID="fmt/174" Version="98">
            <Extension>xls</Extension>
        </FileFormat>
        <FileFormat ID="881" Name="Microsoft Excel for Macintosh"
            PUID="fmt/175" Version="2001">
            <Extension>xls</Extension>
        </FileFormat>
        <FileFormat ID="882" Name="Microsoft Excel for Macintosh"
            PUID="fmt/176" Version="2002">
            <Extension>xls</Extension>
        </FileFormat>
        <FileFormat ID="883" Name="Microsoft Excel for Macintosh"
            PUID="fmt/177" Version="2004">
            <Extension>xls</Extension>
        </FileFormat>
        <FileFormat ID="884" Name="Microsoft Excel for Macintosh"
            PUID="fmt/178" Version="X">
            <Extension>xls</Extension>
        </FileFormat>

        <FileFormat ID="940" MIMEType="application/vnd.ms-excel"
            Name="Microsoft Excel for Windows" PUID="fmt/214" Version="2007 onwards">
            <Extension>xlsx</Extension>
        </FileFormat>
    </FileFormatCollection>
</FFSignatureFile>
```

❒ PRONOM에 등록된 PDF 포맷 정보

## <표 3-51> PRONOM에 등록된 PDF 포맷 정보

```
<?xml version="1.0" encoding="UTF-8"?>
<FFSignatureFile DateCreated="2013-08-28T10:17:01" Version="70"
xmlns="http://www.nationalarchives.gov.uk/pronom/SignatureFile">
    <InternalSignatureCollection>
     <InternalSignature ID="613" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>1D7D000070</Sequence>
                    <DefaultShift>6</DefaultShift>
                    <Shift Byte="00">2</Shift>
                    <Shift Byte="1D">5</Shift>
                    <Shift Byte="70">1</Shift>
                    <Shift Byte="7D">4</Shift>
                    <RightFragment MaxOffset="120" MinOffset="120" Position="1">7D001D
</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="614" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>31BE000000AB0000000000000000</Sequence>
                    <DefaultShift>15</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="31">14</Shift>
                    <Shift Byte="AB">9</Shift>
                    <Shift Byte="BE">13</Shift>
                    <RightFragment MaxOffset="82" MinOffset="82" Position="1">[!0000]
</RightFragment>
```

```
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="615" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>32BE000000AB0000000000000000</Sequence>
                    <DefaultShift>15</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="32">14</Shift>
                    <Shift Byte="AB">9</Shift>
                    <Shift Byte="BE">13</Shift>
                    <RightFragment MaxOffset="82" MinOffset="82" Position="1">[!0000]
</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="616" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>23646566696E6520</Sequence>
                    <DefaultShift>9</DefaultShift>
                    <Shift Byte="20">1</Shift>
                    <Shift Byte="23">8</Shift>
                    <Shift Byte="64">7</Shift>
                    <Shift Byte="65">2</Shift>
                    <Shift Byte="66">5</Shift>
                    <Shift Byte="69">4</Shift>
                    <Shift Byte="6E">3</Shift>
                    <RightFragment MaxOffset="30" MinOffset="1" Position="1">5F776964746820
</RightFragment>
                    <RightFragment MaxOffset="16" MinOffset="1" Position="2">23646566696E6520
</RightFragment>
                    <RightFragment MaxOffset="30" MinOffset="1" Position="3">5F68656967687420
</RightFragment>
                    <RightFragment MaxOffset="300" MinOffset="1" Position="4">73746174696320
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="5">63686172
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0"
Position="5">756E7369676E65642063686172</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="6">20
</RightFragment>
                </SubSequence>
                <SubSequence MinFragLength="0" Position="2" SubSeqMinOffset="0">
                    <Sequence>5F626974735B5D203D207B</Sequence>
                    <DefaultShift>12</DefaultShift>
                    <Shift Byte="20">2</Shift>
                    <Shift Byte="3D">3</Shift>
                    <Shift Byte="5B">6</Shift>
                    <Shift Byte="5D">5</Shift>
                    <Shift Byte="5F">11</Shift>
                    <Shift Byte="62">10</Shift>
                    <Shift Byte="69">9</Shift>
                    <Shift Byte="73">7</Shift>
                    <Shift Byte="74">8</Shift>
                    <Shift Byte="7B">1</Shift>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="617" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>23646566696E6520</Sequence>
                    <DefaultShift>9</DefaultShift>
                    <Shift Byte="20">1</Shift>
                    <Shift Byte="23">8</Shift>
                    <Shift Byte="64">7</Shift>
                    <Shift Byte="65">2</Shift>
                    <Shift Byte="66">5</Shift>
                    <Shift Byte="69">4</Shift>
                    <Shift Byte="6E">3</Shift>
                    <RightFragment MaxOffset="30" MinOffset="1" Position="1">5F776964746820
</RightFragment>
```

```
                    <RightFragment MaxOffset="16" MinOffset="1" Position="2">23646566696E6520
</RightFragment>
                    <RightFragment MaxOffset="30" MinOffset="1" Position="3">5F68656967687420
</RightFragment>
                </SubSequence>
                <SubSequence MinFragLength="0" Position="2" SubSeqMinOffset="0">
                    <Sequence>737461746963207 56E7369676E65642073686F727420</Sequence>
                    <DefaultShift>23</DefaultShift>
                    <Shift Byte="20">1</Shift>
                    <Shift Byte="61">20</Shift>
                    <Shift Byte="63">17</Shift>
                    <Shift Byte="64">8</Shift>
                    <Shift Byte="65">9</Shift>
                    <Shift Byte="67">11</Shift>
                    <Shift Byte="68">5</Shift>
                    <Shift Byte="69">12</Shift>
                    <Shift Byte="6E">10</Shift>
                    <Shift Byte="6F">4</Shift>
                    <Shift Byte="72">3</Shift>
                    <Shift Byte="73">6</Shift>
                    <Shift Byte="74">2</Shift>
                    <Shift Byte="75">15</Shift>
                    <RightFragment MaxOffset="30" MinOffset="1"
Position="1">5F626974735B5D203D207B</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="618" Specificity="Specific">
            <ByteSequence Endianness="Big-endian" Reference="BOFoffset">
                <SubSequence MinFragLength="4" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>00000007000000</Sequence>
                    <DefaultShift>8</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="07">4</Shift>
                    <LeftFragment MaxOffset="1" MinOffset="1" Position="1">000000
</LeftFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="1">00
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="1">01
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="1">02
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="2">000000[01:20]
</RightFragment>
                    <RightFragment MaxOffset="12" MinOffset="12" Position="3">000000
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="4">00
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="4">01
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="5">000000
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="6">08
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="6">10
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="6">20
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="7">000000
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="8">00
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="8">01
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="9">000000
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="10">08
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="10">10<
/RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="10">20
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="11">000000
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">01
</RightFragment>
```

```
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">02
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">03
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">04
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">05
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">06
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">07
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">08
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">09
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">0A
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">0B
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">0C
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">0D
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">0E
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">0F
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">10
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">18
</RightFragment>
                    <RightFragment MaxOffset="0" MinOffset="0" Position="12">20
</RightFragment>
                    <RightFragment MaxOffset="4" MinOffset="4" Position="13">000000[00:05]
</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="637" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1">

<Sequence>00000000000000000FFFF0100100043486172726973 4D6174726978446F63</Sequence>
                    <DefaultShift>31</DefaultShift>
                    <Shift Byte="00">17</Shift>
                    <Shift Byte="01">20</Shift>
                    <Shift Byte="10">18</Shift>
                    <Shift Byte="43">16</Shift>
                    <Shift Byte="44">3</Shift>
                    <Shift Byte="48">15</Shift>
                    <Shift Byte="4D">9</Shift>
                    <Shift Byte="61">8</Shift>
                    <Shift Byte="63">1</Shift>
                    <Shift Byte="69">5</Shift>
                    <Shift Byte="6F">2</Shift>
                    <Shift Byte="72">6</Shift>
                    <Shift Byte="73">10</Shift>
                    <Shift Byte="74">7</Shift>
                    <Shift Byte="78">4</Shift>
                    <Shift Byte="FF">21</Shift>
                </SubSequence>
            </ByteSequence>
            <ByteSequence Reference="EOFoffset">
                <SubSequence MinFragLength="0" Position="1">
                    <Sequence>00000001000000</Sequence>
                    <DefaultShift>-8</DefaultShift>
                    <Shift Byte="00">-1</Shift>
                    <Shift Byte="01">-4</Shift>
                    <LeftFragment MaxOffset="1" MinOffset="1" Position="1">02266E00
</LeftFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="770" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
```

```
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">

<Sequence>FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF</Sequence>
                    <DefaultShift>33</DefaultShift>
                    <Shift Byte="FF">1</Shift>
                    <RightFragment MaxOffset="2016" MinOffset="2016" Position="1">504344
</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="787" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>43575309</Sequence>
                    <DefaultShift>5</DefaultShift>
                    <Shift Byte="09">1</Shift>
                    <Shift Byte="43">4</Shift>
                    <Shift Byte="53">2</Shift>
                    <Shift Byte="57">3</Shift>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="788" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>43575310</Sequence>
                    <DefaultShift>5</DefaultShift>
                    <Shift Byte="10">1</Shift>
                    <Shift Byte="43">4</Shift>
                    <Shift Byte="53">2</Shift>
                    <Shift Byte="57">3</Shift>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="789" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>00010000000200010005000200002C0</Sequence>
                    <DefaultShift>16</DefaultShift>
                    <Shift Byte="00">3</Shift>
                    <Shift Byte="01">8</Shift>
                    <Shift Byte="02">2</Shift>
                    <Shift Byte="05">6</Shift>
                    <Shift Byte="C0">1</Shift>
                    <RightFragment MaxOffset="527" MinOffset="527" Position="1">50726F2033
</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="790" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>00010000000200010005000200002C0</Sequence>
                    <DefaultShift>16</DefaultShift>
                    <Shift Byte="00">3</Shift>
                    <Shift Byte="01">8</Shift>
                    <Shift Byte="02">2</Shift>
                    <Shift Byte="05">6</Shift>
                    <Shift Byte="C0">1</Shift>
                    <RightFragment MaxOffset="527" MinOffset="527" Position="1">50726F2035
</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="791" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="0" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>060E2B34020501010D0102010102</Sequence>
                    <DefaultShift>15</DefaultShift>
                    <Shift Byte="01">2</Shift>
                    <Shift Byte="02">1</Shift>
                    <Shift Byte="05">9</Shift>
```

```
                    <Shift Byte="06">14</Shift>
                    <Shift Byte="0D">6</Shift>
                    <Shift Byte="0E">13</Shift>
                    <Shift Byte="2B">12</Shift>
                    <Shift Byte="34">11</Shift>
                    <RightFragment MaxOffset="70" MinOffset="70"
Position="1">060E2B34040101010D0102010101</RightFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="818" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="53" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>000200</Sequence>
                    <DefaultShift>4</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="02">2</Shift>
                    <LeftFragment MaxOffset="52" MinOffset="52" Position="1">02</LeftFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
        <InternalSignature ID="819" Specificity="Specific">
            <ByteSequence Reference="BOFoffset">
                <SubSequence MinFragLength="53" Position="1"
                    SubSeqMaxOffset="0" SubSeqMinOffset="0">
                    <Sequence>000300</Sequence>
                    <DefaultShift>4</DefaultShift>
                    <Shift Byte="00">1</Shift>
                    <Shift Byte="03">2</Shift>
                    <LeftFragment MaxOffset="52" MinOffset="52" Position="1">03</LeftFragment>
                </SubSequence>
            </ByteSequence>
        </InternalSignature>
    </InternalSignatureCollection>
    <FileFormatCollection>
        <FileFormat ID="613" MIMEType="application/pdf"
            Name="Acrobat PDF 1.0 - Portable Document Format"
            PUID="fmt/14" Version="1.0">
            <InternalSignatureID>123</InternalSignatureID>
            <Extension>pdf</Extension>
        </FileFormat>
        <FileFormat ID="614" MIMEType="application/pdf"
            Name="Acrobat PDF 1.1 - Portable Document Format"
            PUID="fmt/15" Version="1.1">
            <InternalSignatureID>124</InternalSignatureID>
            <Extension>pdf</Extension>
        </FileFormat>
        <FileFormat ID="615" MIMEType="application/pdf"
            Name="Acrobat PDF 1.2 - Portable Document Format"
            PUID="fmt/16" Version="1.2">
            <InternalSignatureID>125</InternalSignatureID>
            <Extension>pdf</Extension>
        </FileFormat>
        <FileFormat ID="616" MIMEType="application/pdf"
            Name="Acrobat PDF 1.3 - Portable Document Format"
            PUID="fmt/17" Version="1.3">
            <InternalSignatureID>23</InternalSignatureID>
            <Extension>pdf</Extension>
        </FileFormat>
        <FileFormat ID="617" MIMEType="application/pdf"
            Name="Acrobat PDF 1.4 - Portable Document Format"
            PUID="fmt/18" Version="1.4">
            <InternalSignatureID>20</InternalSignatureID>
            <Extension>pdf</Extension>
            <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
        </FileFormat>
        <FileFormat ID="618" MIMEType="application/pdf"
            Name="Acrobat PDF 1.5 - Portable Document Format"
            PUID="fmt/19" Version="1.5">
            <InternalSignatureID>22</InternalSignatureID>
            <Extension>pdf</Extension>
            <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
```

```
            <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
        </FileFormat>
        <FileFormat ID="637" MIMEType="application/pdf"
            Name="Acrobat PDF 1.6 - Portable Document Format"
            PUID="fmt/20" Version="1.6">
            <InternalSignatureID>21</InternalSignatureID>
            <Extension>pdf</Extension>
            <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
        </FileFormat>
        <FileFormat ID="770" MIMEType="application/pdf"
            Name="Acrobat PDF/A - Portable Document Format"
            PUID="fmt/95" Version="1a">
            <InternalSignatureID>264</InternalSignatureID>
            <Extension>pdf</Extension>
            <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
        </FileFormat>
        <FileFormat ID="787" MIMEType="application/pdf"
            Name="Acrobat PDF/X - Portable Document Format - Exchange 1:1999"
PUID="fmt/144">
            <InternalSignatureID>692</InternalSignatureID>
            <Extension>pdf</Extension>
        </FileFormat>
        <FileFormat ID="788" MIMEType="application/pdf"
            Name="Acrobat PDF/X - Portable Document Format - Exchange 1:2001"
PUID="fmt/145">
            <InternalSignatureID>693</InternalSignatureID>
            <Extension>pdf</Extension>
        </FileFormat>
        <FileFormat ID="789" MIMEType="application/pdf"
            Name="Acrobat PDF/X - Portable Document Format - Exchange 1a:2003"
PUID="fmt/146">
            <InternalSignatureID>694</InternalSignatureID>
            <Extension>pdf</Extension>
        </FileFormat>
        <FileFormat ID="790" MIMEType="application/pdf"
            Name="Acrobat PDF/X - Portable Document Format - Exchange 2:2003"
PUID="fmt/147">
            <InternalSignatureID>695</InternalSignatureID>
            <Extension>pdf</Extension>
        </FileFormat>
        <FileFormat ID="791" MIMEType="application/pdf"
            Name="Acrobat PDF/X - Portable Document Format - Exchange 3:2003"
PUID="fmt/148">
            <InternalSignatureID>696</InternalSignatureID>
            <Extension>pdf</Extension>
        </FileFormat>
        <FileFormat ID="818" MIMEType="application/pdf"
            Name="Acrobat PDF/X - Portable Document Format - Exchange 1a:2001"
PUID="fmt/157">
            <InternalSignatureID>581</InternalSignatureID>
            <Extension>pdf</Extension>
            <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>788</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>1100</HasPriorityOverFileFormatID>
        </FileFormat>
        <FileFormat ID="819" MIMEType="application/pdf"
            Name="Acrobat PDF/X - Portable Document Format - Exchange 3:2002"
PUID="fmt/158">
            <InternalSignatureID>582</InternalSignatureID>
            <InternalSignatureID>597</InternalSignatureID>
            <Extension>pdf</Extension>
            <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>1100</HasPriorityOverFileFormatID>
        </FileFormat>
        <FileFormat ID="1016" MIMEType="application/pdf"
            Name="Acrobat PDF 1.7 - Portable Document Format"
            PUID="fmt/276" Version="1.7">
            <InternalSignatureID>289</InternalSignatureID>
            <Extension>pdf</Extension>
        </FileFormat>
```

```
<FileFormat ID="1100" MIMEType="application/pdf"
    Name="Acrobat PDF/A - Portable Document Format"
    PUID="fmt/354" Version="1b">
    <InternalSignatureID>521</InternalSignatureID>
    <Extension>pdf</Extension>
    <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>637</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>1016</HasPriorityOverFileFormatID>
</FileFormat>
<FileFormat ID="1263"
    Name="Acrobat PDF/A - Portable Document Format"
    PUID="fmt/476" Version="2a">
    <InternalSignatureID>750</InternalSignatureID>
    <Extension>pdf</Extension>
    <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>637</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>1016</HasPriorityOverFileFormatID>
</FileFormat>
<FileFormat ID="1264"
    Name="Acrobat PDF/A - Portable Document Format"
    PUID="fmt/477" Version="2b">
    <InternalSignatureID>751</InternalSignatureID>
    <Extension>pdf</Extension>
    <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>637</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>1016</HasPriorityOverFileFormatID>
</FileFormat>
<FileFormat ID="1265"
    Name="Acrobat PDF/A - Portable Document Format"
    PUID="fmt/478" Version="2u">
    <InternalSignatureID>752</InternalSignatureID>
    <Extension>pdf</Extension>
    <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>637</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>1016</HasPriorityOverFileFormatID>
</FileFormat>
<FileFormat ID="1266"
    Name="Acrobat PDF/A - Portable Document Format"
    PUID="fmt/479" Version="3a">
    <InternalSignatureID>753</InternalSignatureID>
    <Extension>pdf</Extension>
    <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>637</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>1016</HasPriorityOverFileFormatID>
</FileFormat>
<FileFormat ID="1267"
    Name="Acrobat PDF/A - Portable Document Format"
    PUID="fmt/480" Version="3b">
    <InternalSignatureID>754</InternalSignatureID>
    <Extension>pdf</Extension>
    <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
    <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
```

```
            <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>637</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>1016</HasPriorityOverFileFormatID>
        </FileFormat>
        <FileFormat ID="1268"
            Name="Acrobat PDF/A - Portable Document Format"
            PUID="fmt/481" Version="3u">
            <InternalSignatureID>755</InternalSignatureID>
            <Extension>pdf</Extension>
            <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>637</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>1016</HasPriorityOverFileFormatID>
        </FileFormat>
        <FileFormat ID="1275"
            Name="Acrobat PDF/X - Portable Document Format - Exchange PDF/X-4"
PUID="fmt/488">
            <InternalSignatureID>762</InternalSignatureID>
            <Extension>pdf</Extension>
            <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>637</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>1016</HasPriorityOverFileFormatID>
        </FileFormat>
        <FileFormat ID="1276"
            Name="Acrobat PDF/X - Portable Document Format - Exchange PDF/X-4p"
PUID="fmt/489">
            <InternalSignatureID>763</InternalSignatureID>
            <Extension>pdf</Extension>
            <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>637</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>1016</HasPriorityOverFileFormatID>
        </FileFormat>
        <FileFormat ID="1277"
            Name="Acrobat PDF/X - Portable Document Format - Exchange PDF/X-5g"
PUID="fmt/490">
            <InternalSignatureID>764</InternalSignatureID>
            <Extension>pdf</Extension>
            <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>637</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>1016</HasPriorityOverFileFormatID>
        </FileFormat>
        <FileFormat ID="1278"
            Name="Acrobat PDF/X - Portable Document Format - Exchange PDF/X-5pg"
PUID="fmt/491">
            <InternalSignatureID>766</InternalSignatureID>
            <Extension>pdf</Extension>
            <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>637</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>1016</HasPriorityOverFileFormatID>
        </FileFormat>
        <FileFormat ID="1279"
            Name="Acrobat PDF/X - Portable Document Format - Exchange PDF/X-5n"
PUID="fmt/492">
```

```
            <InternalSignatureID>765</InternalSignatureID>
            <Extension>pdf</Extension>
            <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>637</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>1016</HasPriorityOverFileFormatID>
        </FileFormat>
        <FileFormat ID="1280"
            Name="Acrobat PDF/E - Portable Document Format for Engineering PDF/E-1"
PUID="fmt/493">
            <InternalSignatureID>761</InternalSignatureID>
            <Extension>pdf</Extension>
            <HasPriorityOverFileFormatID>613</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>614</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>615</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>616</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>617</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>618</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>637</HasPriorityOverFileFormatID>
            <HasPriorityOverFileFormatID>1016</HasPriorityOverFileFormatID>
        </FileFormat>
    </FileFormatCollection>
</FFSignatureFile>
```

○ PRONOM에 등록된 XLS, PPT, PDF 포맷 정보는 포맷 식별을 위해 사용할 수 있도록 다양한 포맷 버전에 대한 XML 정보가 잘 정의되어 활용되고 있음

### 라. 기술정보은행(DFR)과 PRONOM의 포맷 기술 정보 비교 분석

<표 3-52> 기술정보은행(DFR)과 DROID의 포맷 Element 비교

| Element (기술정보은행(DFR)) | Element (DROID) |
|---|---|
| General | |
| FormatID | FileFormat ID |
| Identifier | |
| **PUID-TI** | FileFormat PUID |
| FormatName | FileFormat Name |
| FormatVersion | FileFormat Version |
| FormatAlias | |
| FormatType | |
| Status | |
| IPR | |
| Developer | |
| Support | |
| ReleaseDate | |
| WithdrawDat | |
| Note | |
| Technical | |
| FormatDisclose | |
| Orientation | |
| RelatedFormat | HasPriorityOverFileFormatID |
| Equality | |
| RelatedVersion | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | RelatedType | | | | | | |
| | | Etc | | | | | | |
| | Specification | | | | | | | |
| | | Document | | | | | | |
| | ExternalSignature | | | | | | | |
| | | ExtensionName | | | Extension | | | |
| | InternalSignature | | | | | | | |
| | | SignatureID | | | | InternalSignatureID | | |
| | | ByteSequence | | | | ByteSequence | | |
| | | | SubSequnce | | | | SubSequence | |
| | | | | Sequence | | | | Sequence |
| | | | | DefaultShift | | | | DefaultShift |
| | | | | Shift | | | | Shift |
| | | | | LeftFragment | | | | LeftFragment |
| | | | | RightFragment | | | | RightFragment |
| | TechnicalEnvironment | | | | | | | |
| | Well-formed Note | | | | | | | |
| | Validity Note | | | | | | | |
| | Metadata | | | | | | | |
| | Note | | | | | | | |
| Management | | | | | | | | |
| | Source | | | | | | | |
| | SourceDate | | | | | | | |
| | LastUpdate | | | | | | | |
| | Type | | | | | | | |
| | Note | | | | | | | |

○ 기술정보은행(DFR)의 XML Element와 PRONOM에서 정의된 XML Element를 상호 비교해 보면 PRONOM의 Element 구조는 포맷 기본 정보를 등록하는 FormatID, FormatName, FormatVersion, FormatPUID Element와 InternalSignature와 관련된 SignatureID, ByteSequence, SubSequence Element가 공통으로 정의되어 있으나 기술정보은행(DFR)의 나머지 태그는 정의는 되어 있으나 실제 데이터가 채워져 있지 않은 것으로 파악됨

○ PRONOM에 등록된 XML 데이터는 기술정보은행(DFR)의 XMLSchema와 비교할 때 정의된 태그의 개수는 적지만 InternalSignature와 ByteSequence 태그 정보가 빠짐없이 등록되어 있음을 확인

○ 기술정보은행(DFR)을 표준 레지스트리로 활용하기 위해서는 Element에 대한 정의가 중요한 것이 아니라 정의된 Element에 대한 데이터가 정확하게 등록되어 관리되는 것이 필요한데, 기술정보은행(DFR)에서는 다양한 Element를 정의했지만 데이터가 전혀 확보되지 않은 상태이므로 향후 포맷 정보 보존을 위해 기술정보은행(DFR)에 보다 상세하고 정확한 데이터를 등록하여야 함.

## 마. 기능성 보존을 위한 XML Schema 개선안

### (1) 기능성 보존의 필요성

❒ 현재의 국가기록원 기술정보은행(DFR) 정보가 실제 포맷 정보가 아니라 포맷을 설명하기 위한 메타 정보로만 구성되어 있고 소프트웨어 정보 역시 소프트웨어에 대한 기본적인 정보만을 포함하고 있으므로 동적인 렌더링 기능성 요소에 대한 속성을 포함하도록 기술정보은행(DFR) 구조 개선에 대한 연구가 필요

❒ 본 연구에서는 아래아한글의 OLE 개체, 엑셀의 수식 및 그래프 기능성에 대한 포맷 정보를 조사하여 국가기록원의 기술정보은행(DFR)에 추가하기 위한 자료구조 및 기술정보은행(DFR) 구조 개선 방향을 제시하고자 하는데 현재 아래아한글 및 마이크로소프트사에서 배포한 기술 문서를 분석하였음. 한/글 파일구조 분석을 위해 “HWP Document File Formats revision 1.0:20100630” 기술 문서를 기반으로 HWP Document File V3.00과 HWP Document File V5.00에 대한 분석을 수행중이고, 엑셀 파일구조 분석을 위해 “Excel Binary File Format(.xls) Structure, Release July 30, 2013” 과 “Excel(.xlsx) Extensions to the Office Open XML SpreadSheetML File Format, Release July 30, 2013” 기술문서를 분석

❒ 각 사에서 배포한 기술 문서는 소프트웨어 기능에 대한 정리부터 자사의 포맷에 포함된 기능성의 데이터 형식까지 상세한 내용을 기술하고 있으므로 소프트웨어 버전별 기능 목록을 우선 검토하여 기능성 요소를 포함하는 기능을 1차 정리한 후 동적 요소를 저장하는 데이터 형식을 상세 검토하여 기술정보은행(DFR) 구조 개선 방향을 제시

### (2) 기능성 보존을 위한 XML Schema 정의

❒ 아래 표에는 3.3.2절에서 조사한 한/글, 파워포인트, 엑세 포맷을 문서보존포맷(PDF/A)로 변환 시 소실되는 기능성을 보존하기 위해 3.3.3절에서 분석 검토한 기능성의 속성 정보를 보존하기 위한 구성요소와 XML Schema를 제시함

- ○ 기능성 보존을 위한 구성요소 Element는  현재 국가기록원 기술정보은행(DFR)의 포맷에 대한 기술 정보에 추가하여 Element 요소를 확장하도록 제안함
- ○ 3.3.3절에서 제시한 Characterization을 위한 구성요소와 본 절에서 제시한 기능성 보존을 위한 구성요소는 모두 현재의 기술정보은행(DFR)에 추가하여 확장

하는 구조를 제시하였으며, 향후 표준 레지스트리의 역할을 위해서는 제시된 Element에 정확한 데이터를 등록 관리하기 위한 작업이 필요함.

○ 앞 절에서 기술정보은행(DFR)과 DROID의 Element를 비교하면서 시사점으로 밝힌 바와 같이 Element에 대한 정의가 중요한 것이 아니라, 정의된 Element에 정확한 데이터를 등록 관리하는 것이 표준 레지스트리의 역할이기 때문에 국가기록원 기술정보은행(DFR)이 국가 표준 레지스트리로 자리잡기 위해서는 Element에 대한 정확한 정보 등록 및 관리가 지속적으로 필요함.

<표 3-53> 기능성 보존을 위한 구성요소

| Element | 설명 |
|---|---|
| General | 포맷의 일반적인 요소 |
| Rendering | 내부시스템 식별자 |
| ID | 고유한 식별자 |
| containername | Container의 이름 |
| version | container에 대한 설명에 대한 버젼 정보 |
| relatedID | 연관된 렌더링을 위한 컨테이너 |
| referenceID | 해당 Container를 참조하고 아이디 |
| Format | container를 포함하고 있는 포맷에 대한 정보 |
| name | Container를 포함하고 있는 포맷의 이름 |
| version | Container를 포함하고 있는 포맷의 버젼 |
| extension | 포맷의 확장자 |
| application | 확장자와 관련된 애플리케이션 |
| name | 애플리케이션의 이름 |
| version | 애플리케이션의 버젼 |
| softwarealias | 소프트웨어를 부르는 별칭 |
| corporation | 소프트웨어를 개발한 회사 |
| hwrequirement | 소프트웨어를 설치하기 위한 하드웨어 조건 |
| swrequirement | 소프트웨어를 설치하기 위한 os 조건 |
| Explanation | 렌더링을 위한 기술적인 정보 |
| container | 컨테이너의 정보 |
| name | 컨테이너의 이름 |
| property | 컨테이너의 속성, 그리고 컨테이너 전체를 설명할 때에는 생략 |
| explain | 설명을 위한 데이터가 들어가는 부분 |
| type | 저장된 데이터의 형태 txt/binary |
| formatinfo | binary로 저장된 경우, 포맷 정보 |
| explaindata | 실제 데이터가 저장된 곳 |

<표 3-54> 기능성 보존을 위한 추가 XML Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http//www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
```

```
<xs:element name="FileRender">
    <xs:annotation>
        <xs:documentation>Root Element</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <!-- 포맷에 관한 ID, 버젼과 같은 일반 적인 정보 저장 -->
            <xs:element ref="General"/>
            <!-- 렌더링을 하는데 필요한 설명 자료 -->
            <!-- Container에 대한 설명 혹은 Container의 속성에 대한 설명 -->
            <xs:element ref="Explanation"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>    <!-- <xs:element name="FileRender"> -->
<!-- 렌더링을 위한 일반 정보 -->
<xs:element name="General">
    <xs:annotation>
        <xs:documentation>렌더링을 위한 일반 정보</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Rendering">
                <!-- 고유한 값 -->
                <xs:attribute name="ID"/>
                <!-- Container의 이름 -->
                <xs:attribute name="containername"/>
                <!-- container에 대한 설명에 대한 버젼 정보 -->
                <xs:attribute name="version"/>
                <!-- 연관된 렌더링을 위한 컨테이너 -->
                <xs:element name="relatedID"/>
                <!-- 해당 Container를 참조하고 아이디 -->
                <xs:element name="referencedID"/>
            </xs:element>    <!-- <xs:element name="Rendering"> -->
        </xs:sequence>
    </xs:complexType>
    <xs:complexType>
        <xs:sequence>
            <!-- container를 포함하고 있는 포맷에 대한 정보 -->
            <xs:element name="Format">
                <xs:complexType>
                    <!-- Container를 포함하고 있는 포맷의 이름 -->
                    <xs:attribute name="name"/>
                    <!-- Container를 포함하고 있는 포맷의 버젼 -->
                    <xs:attribute name="version"/>
                    <!-- 포맷의 확장자 -->
                    <xs:element name="extension"/>
                    <!-- 확장자와 관련된 애플리케이션 -->
                    <xs:element name="application">
                        <!-- 애플리케이션의 이름 (MS Word, 한/글 등등) -->
                        <xs:attribute name="name"/>
                        <!-- 애플리케이션의 버젼 -->
                        <xs:attribute name="version"/>
                        <xs:complexType>
                            <!-- 소프트웨어를 부르는 별칭 -->
                            <xs:element name="softwarealias"/>
                            <!-- 소프트웨어를 개발한 회사 -->
                            <xs:element name="corporation"/>
                            <!-- 소프트웨어를 설치하기 위한 하드웨어 조건 -->
                            <xs:element name="hwrequirement"/>
                            <!-- 소프트웨어를 설치하기 위한 os 조건 -->
                            <xs:element name="swrequirement"/>
                        </xs:complexType>
                    </xs:element>    <!-- <xs:element name="application"> -->
                </xs:complexType>
            </xs:element>    <!-- <xs:element name="Format"> -->
        </xs:sequence>
    </xs:complexType>
</xs:element>    <!-- <xs:element name="General"> -->
<!-- 렌더링을 위한 일반 정보 -->
```

```
    <xs:element name="Explanation">
        <xs:annotation>
            <xs:documentation>렌더링을 위한 기술적인 정보</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="container">
                    <!-- 컨테이너의 이름 -->
                    <xs:attribute name="name"/>
                    <xs:complexType>
                        <!-- 컨테이너의 속성, 그리고 컨테이너 전체를 설명할때에는 생략
-->
                        <xs:element name="property"/>
                        <!-- 설명을 위한 데이터가 들어가는 부분 -->
                        <xs:element name="explain">
                            <!-- 저장된 데이터의 형태 txt/binary -->
                            <xs:attribute name="type"/>
                            <!-- binary로 저장된 경우, 포맷 정보 -->
                            <xs:attribute name="formatinfo"/>
                            <!-- 실제 데이터가 저장된 곳 -->
                            <xs:element name="explaindata"/>
                        </xs:element>    <!-- <xs:element name="explanation"/> -->
                    </xs:complexType>
                </xs:element>    <!-- <xs:element name="container"> -->
            </xs:sequence>
        </xs:complexType>
    </xs:element>    <!-- <xs:element name="Explanation"> -->
</xs:schema> <!-- 끝 -->
```

❒ 국가기록원 기술정보은행(DFR)에 렌더링을 위한 기능성 요소를 추가한 형태로 DFR 확장 필요

- ○ 렌더링을 위한 일반 정보(General)와 컨테이너와 속성에 대한 설명하는 부분(Explanation)으로 구성
- ○ General에는 Rendering, Format으로 구성되어 있고, ID와 연관된 컨테이너 정보, 컨테이너를 포함하고 있는 포맷에 대한 확장자, 애플리케이션과 같은 생산에 관련된 정보 저장
- ○ Explanation에서 렌더링을 하기 위한 컨테이너와 속성에 대한 설명 그리고 화면 표현을 위한 설명으로 구성
- ○ 컴포넌트를 화면에 표현하기 위해서는 해당하는 컨테이너와 속성 하나하나에 대해서 상세하게 표현하여야 함
- ○ 예를 들면 한/글 소프트웨어가 사라진 경우, 화면에 어떻게 표현되는지 어떻게 표현하여야 하는지 알 수가 없고, 한/글 문서를 에뮬레이션 하는 새로운 애플리케이션을 구축하는 경우를 예측하여 저장하여야 함

## 3.4. 에뮬레이션 제안 및 프로토타입 구현

### 3.4.1 에뮬레이션 방안 연구

❒ 국가기록원에서는 “중앙영구기록물관리시스템(CAMS)”에 장기(30년, 준영구, 영

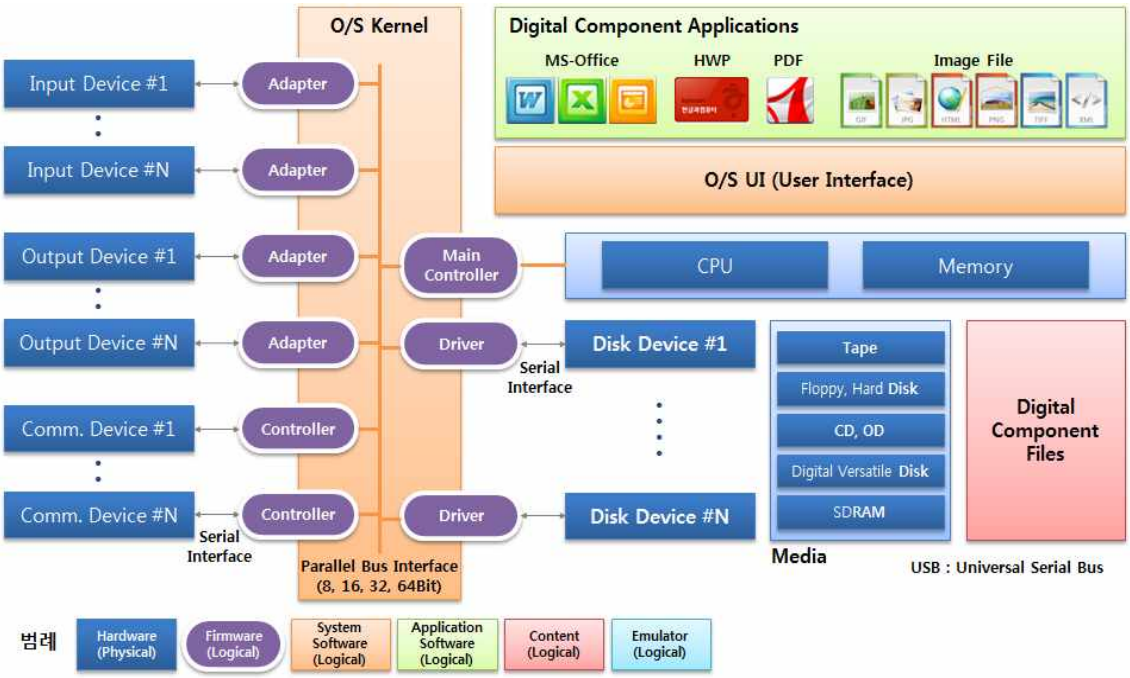
구) 보존이 필요한 공공기관 전자기록물을 등록 관리하는데, 이러한 전자기록물에 포함된 디지털 컴포넌트의 렌더링 기능성를 조회할 수 있는 에뮬레이션 전략 수립이 필요

❒ 본 절에서는 구 형식의 전자기록물을 조회하기 위한 다양한 에뮬레이션 기술 동향과 해외 에뮬레이션 사례에서 적용한 에뮬레이션 방식을 분석한 후 한국형 에뮬레이션 전략을 제시하고자 함

## 가. 에뮬레이션 대상 연구

### (1) 에뮬레이션 대상 분석

❒ PC 기본 구조




<그림 3-129> PC 기본 구조

○ 현재까지 개발되어 사용 중인 폰노이만 방식의 PC는 위의 그림과 같이 중앙처리장치, 주기억장치, 입출력장치 및 보조기억장치가 버스 구조로 연결되어 있으며 운영체제(O/S)를 통해 사용자의 제어가 가능하도록 구성됨

○ 디지털 컴포넌트를 구동하는 애플리케이션과 데이터는 모두 보조기억장치에 저장되어 있으므로 기억장치를 에뮬레이션 하기 위한 검토가 필요

○ 운영체제(Kernel)는 하드웨어를 구동하기 위한 가장 중요한 요소로서 다양한 디바이스를 제어하기 위해 Adapter, Driver, Controller 등의 펌웨어 및 소프트

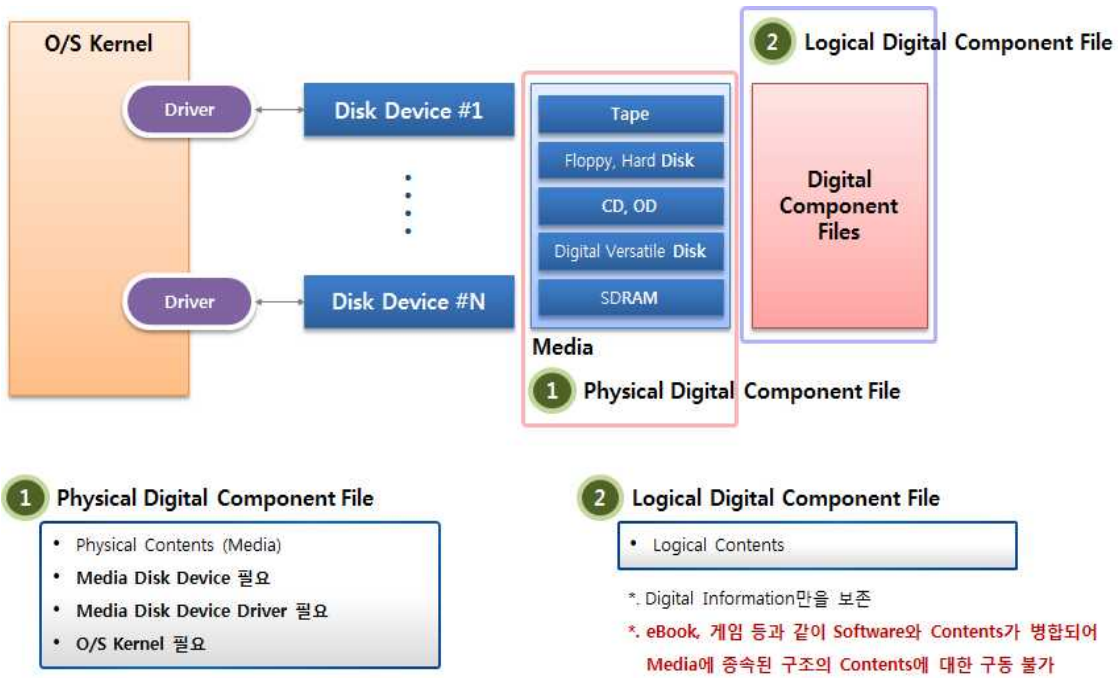
웨어가 필요함

○ 현재 사용 중인 운영체제에 필요한 디바이스 Driver나 Adapter는 제조회사에서 제공되지만 구형의 디바이스에 대한 Driver나 Adapter는 제공되지 않으므로 구형의 디바이스를 구동하기 위해서는 에뮬레이션이 필요

○ 즉, 신형 하드웨어가 출시되면서 운영체제가 업그레이드되거나 새로운 운영체제가 등장하면 구형의 하드웨어 및 운영체제에서 사용되던 디바이스 및 애플리케이션 소프트웨어는 더 이상 사용할 수 없으므로 이를 위한 대책이 필요

❒ 디지털 컴포넌트의 보존 방식

○ 아래 그림은 디지털 컴포넌트를 저장하여 보존하는 방안에 대한 검토로서 ①번 방안은 Media에 저장된 전자기록물을 Media 단위로 보존하는 방안이고 ②번 방안은 Media에 관계없이 전자기록물을 보존하는 방안임




<그림 3-130> 디지털 컴포넌트 보존 방식 비교

○ 국가기록원의 전자기록물은 “중앙영구기록물관리시스템(CAMS)”의 Archiving Storage를 통해 모든 전자기록물을 ②번 방식으로 보존하므로 Logical 방식의 디지털 컴포넌트 보존 방식임을 확인

○ 향후 국가기록원의 한국형 에뮬레이션 방안은 “중앙영구기록물관리시스템(CAMS)”에 등록된 디지털 컴포넌트를 대상으로 정의함

### (2) 에뮬레이션 해외 사례 비교

❒ 해외 전자기록물 에뮬레이션 사례 프로젝트 비교

<표 3-55> 해외 에뮬레이션 프로젝트 비교

| 항목 | | CAMiLEON | KB | Planets | KEEP |
|---|---|---|---|---|---|
| 프로젝트 개요 | 명칭 | Creating Creative Archiving at Michigan and LEEDS | Koninklike Bibliotheek | Preservation and Long term Access through NETworked Services | Keeping emulation environments portable |
| | 년도 | 1992 ~ 2002 | 2005 ~ 2007 | 2006 ~ 2010 | 2009 ~ 2012 |
| | 주관 기관 | 미시건 & 리즈대학 | 네덜란드국립도서관 & 네덜란드국립아카이브 | British Library (대영도서관) | 파리국립도서관 |
| | 내용 | 에뮬레이션을 중심으로 한 디지털매체의 장기적 보존 방법론 개발 연구 프로젝트 | 에뮬레이션 기반의 새로운 보존전략에 관한 연구 프로젝트 | 디지털 형태의 문화 및 과학 자산들에 대한 장기적인 접근을 돕기 위한 실제적인 서비스와 도구 개발<br>디지털 객체를 관리하기 위한 일련의 도구들을 개발하고 시험하는데 사용할 수 있는 견고한 프레임워크와 인프라를 제공하려는 지향점을 가짐 | 에뮬레이션에 기반한 영구적인 접근 전략 연구<br>렌더링 기반의 에뮬레이션 접근 플랫폼(emulation access platform) 개발 |
| | 금액 | | | 206억 | 58억 |
| | 적용 사례 | BBC Domesday 프로젝트: BBC와 Acorn Computer Ltd, Philips, Logica가 공동으로 조사한 영국 국민들의 삶에 대한 기록 | 하드웨어 아키텍처의 구성요소를 에뮬레이션한 2007년 Dioscuri란 명칭으로 개발하였는데, 이 에뮬레이터는 Planets 프로젝트에 의해 계속 연구를 수행 | Open Planets Foundation을 통해서 그 결과물들이 유지 운영 | Open Planets Foundation을 통해서 그 결과물들이 유지 운영 |
| | URL | http://www2.si.umich.edu/CAMILEON/index.html | http://www.kb.nl/en/expertise/e-depot-and-digital-preservation | http://www.Planets-project.eu/ | http://www.keep-project.eu/ |
| | 정책 | • 소프트웨어 장기 지속 에뮬레이션<br>• 마이그레이션 온 리퀘스트(이용 시점에 객체를 마이그레이션하는 도구와 함께 원래의 바이트스트림을 유지하여 보다 정확하고 비용 효율적이 되도록 함) | • 장기보존 전략으로서의 다양한 에뮬레이션 전략<br>• 에뮬레이션 기술의 지속성/유연성 | • 네트워크 서비스를 통한 장기보존 및 접근 방법<br>• 구형화 된 디지털 자산의 에뮬레이션 툴, 서비스 평가 및 테스트<br>• 보존 계획을 위한 테스트베드 제공 | • 디지털 컴포넌트 포맷, 플랫폼, 운영체제의 에뮬레이션<br>• 에뮬레이션 서비스<br>• 법적 이슈 및 저작권 관리 |

| | | | | | |
|---|---|---|---|---|---|
| | 산출물 | • 다양한 수준의 에뮬레이션 비교자료<br>• 도서관용 보존 툴<br>• 마이그레이션 및 에뮬레이션 전략 가이드라인<br>• 다양한 디지털 객체의 속성 정의<br>• 디지털 객체 보존 전략 | • 에뮬레이션 방법론 4종<br>• 모듈 에뮬레이터: Dioscuri<br>• 에뮬레이터 명세 문서(ESD) 제시 | • Planets 상호운용성 프레임워크<br>• 보존계획 툴 : Plato(의사결정 지원서비스)<br>• 에뮬레이션 툴 : GRATE | • 이관 툴<br>• KEEP-EF(Emulation Framework)<br>• TOTEM(Trustworthy Online Technical Metadata Database)이라는 이름으로 웹에서 기술 메타데이터를 담은 데이터베이스를 시범적으로 구축 |

○ 1992년부터 시작한 CAMiLEON 프로젝트부터 2012년까지 수행된 KEEP 프로젝트의 목적과 내용 산출물을 비교 검토한 결과 CAMiLEON과 KB 프로젝트는 방법론 정립 중심의 프로젝트인 반면 Planets과 KEEP 프로젝트를 통해서는 다양한 산출물 및 시스템이 구축됨

○ CAMiLEON 프로젝트는 장기적 보존 방법론에 대한 연구를 통해 마이그레이션 및 에뮬레이션 방법론을 제시하였고, KB 프로젝트는 에뮬레이션 기반의 새로운 보존 전략을 위해 에뮬레이터(Dioscuri)를 개발하였음

○ Planets 프로젝트를 통해 본격적인 에뮬레이터 3종(Dioscuri, QEMU, UVC) 가 개발되어 GRATE 서비스가 제공되었는데 3종의 에뮬레이터는 운영체제에 대한 에뮬레이터로 정의됨

○ 가장 최근에 수행된 KEEP 프로젝트에서는 렌더링 기반의 에뮬레이션 프레임워크(KEEP-EF)에 Qemu(x86), Dioscuri(x86), WinUAE(Amiga), Vice(c64), JavaCPC (Amstrad/Schneider), BeebEm(BBCmicro), Thomson(Thomson T07) 등 7종의 에뮬레이터를 제공하는데 모두 운영체제 에뮬레이터임을 확인

❒ 해외 전자기록물 에뮬레이션 시스템 비교

○ 해외 에뮬레이션 프로젝트를 통해 개발된 시스템을 분석한 결과 Planets 프로젝트에서 개발한 Plato는 에뮬레이션 정책 관련 시스템이고, KEEP 프로젝트에서 제공하는 KEEP-TTF, TOTEM은 메타데이터 관리 시스템이며 그 외에는 모두 에뮬레이션 프레임워크 및 운영체제 에뮬레이터임을 확인함

**<표 3-56> 해외 에뮬레이션 시스템 비교**

| 항목 | | CAMiLEON | KB | Planets | KEEP |
|---|---|---|---|---|---|
| 결과 시스템 | 정책 | ✖ | ✖ | **Plato** | ✖ |
| | 디바이스 에뮬레이션 | ✖ | Dioscuri | GRATE<br>• Dioscuri | KEEP-EF<br>• Qemu(x86)<br>• Dioscuri(x86) |

| | | | | | |
|---|---|---|---|---|---|
| | 운영체제<br>에뮬레이션 | ✖ | | • QEMU<br>• UVC | • WinUAE(Amiga)<br>• Vice(c64)<br>• JavaCPC(Amstrad/Schneider)<br>• BeebEm(BBCmicro)<br>• Thomson(Thomson T07) |
| | 애플리케이션<br>에뮬레이션 | ✖ | ✖ | ✖ | ✖ |
| | 메타데이터<br>관리 | ✖ | ✖ | DROID<br>PRONUM | DROID<br>PRONUM<br>KEEP-TTF<br>TOTEM |

❒ 본 연구에서는 해외 에뮬레이션 시스템 및 국내외 기술 동향의 분석 결과를 바탕으로 구식 디지털 컴포넌트를 조회할 수 있는 3가지 에뮬레이션 방안을 검토함

○ 디바이스 에뮬레이션 방안

○ 운영체제 에뮬레이션 방안

○ 애플리케이션 에뮬레이션 방안

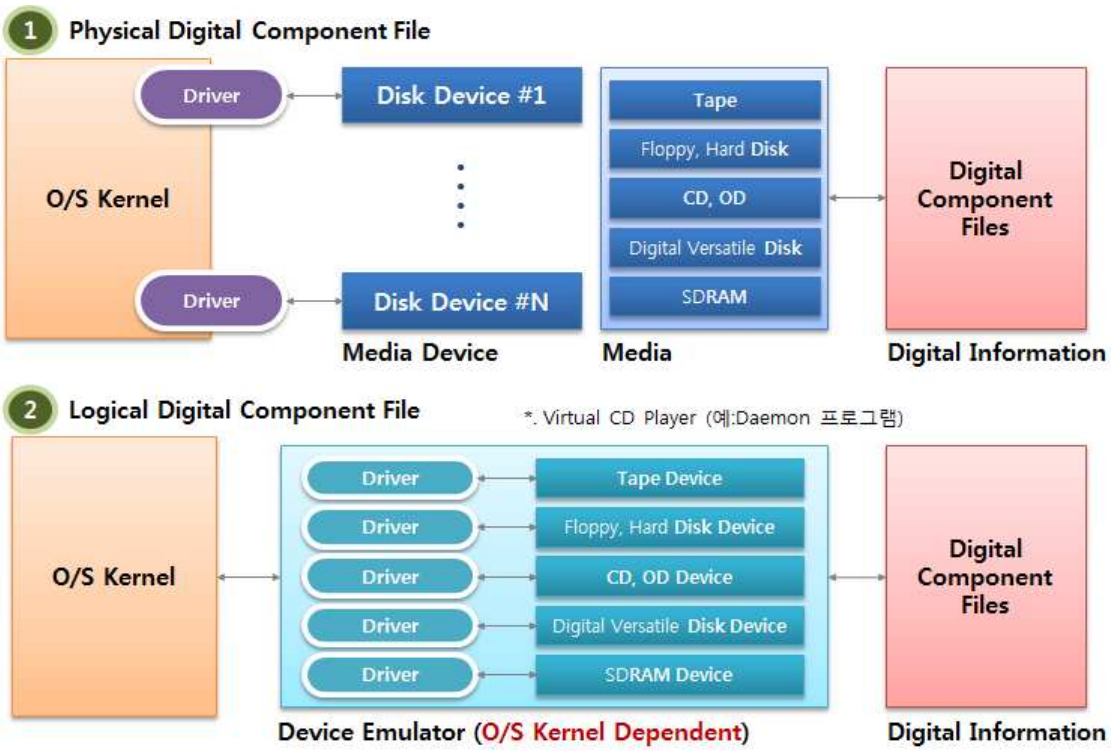
## 나. 에뮬레이션 방안 비교

### (1) 디바이스 에뮬레이션 방안

❒ 미디어 플레이어에 종속된 디지털 컴포넌트를 조회하기 위해서는 테이프, 플로피 디스크 등 미디어 디바이스 및 드라이버에 대한 에뮬레이션이 필요하므로 디바이스에 대한 에뮬레이터를 제공하여 미디어에 저장된 디지털 컴포넌트를 조회하는 방식의 에뮬레이션

❒ 디바이스 에뮬레이션 개념도

○ 미디어에 종속적인 형태로 저장된 디지털 컴포넌트(ISO 포맷)는 조회를 위해 반드시 미디어에 등록된 형태로 조회가 가능하나, Virtual CD Player (예:Daemon 프로그램)와 같이 디바이스 에뮬레이터를 통해 조회가 가능하도록 서비스하는 개념




<그림 3-131> 디바이스 에뮬레이션 개념

○ 디바이스 에뮬레이션은 운영체제에 종속적이며 신규 운영체제에 대한 지속적인 업그레이드가 필요하므로 보통의 경우 운영체제 에뮬레이션과 통합되어 서비스되고 있음

### (2) 운영체제 에뮬레이션

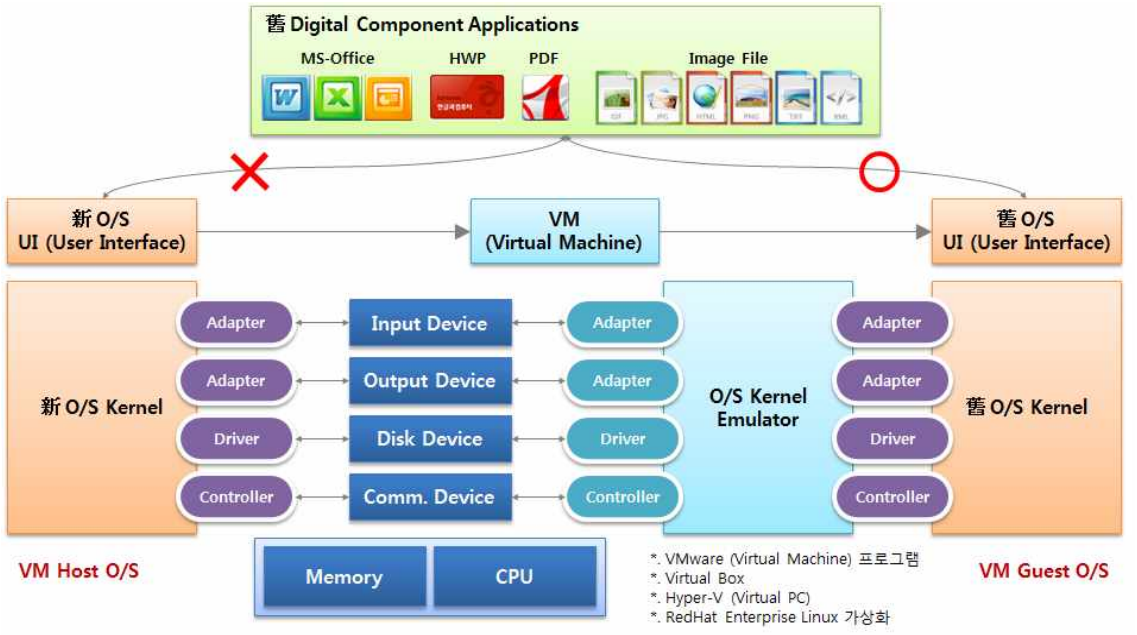
❒ 신형 하드웨어에서 구형 운영체제 기반의 애플리케이션을 구동하기 위해 가상머신(VM : Virtual Machine) 형태의 운영체제 에뮬레이션 지원

○ 운영체제 에뮬레이터는 가상 머신(VM:Virtual Machine)으로 불리는데, 현실에 존재하지 않지만 마치 존재하는 것과 같이 만든 가상의 기계 또는 장치를 의미함

○ 운영체제 에뮬레이터에서 Host OS는 에뮬레이터를 설치할 수 있는 운영체제를 의미하고 Guest OS는 에뮬레이터 기반에서 설치하여 사용 가능한 운영체제를 의미함

❒ 운영체제 에뮬레이션 개념도

○ 구형 애플리케이션을 사용하기 위해 신형 하드웨어에 설치된 신형 운영체제에 에뮬레이터(VM) 및 구형 운영체제를 설치하여 사용하는 방식




<그림 3-132> 운영체제 에뮬레이션 개념

❒ 가상 머신(VM:Viratual Machine)이 실행되기 위해서는 컴퓨터를 구성하는 장치에 대한 에뮬레이션 필요하며 다음의 장치는 반드시 에뮬레이션이 되어야 함

○ CPU, RAM, HDD : 현재 컴퓨터의 CPU, RAM, HDD 성능(용량)의 영향을 굉장히 크게 받음

○ VGA : 100% 가상 장치로써 현재 컴퓨터의 VGA 성능과는 무관하지만 성능에 분명한 한계가 있음

○ Network(Lan) : 100% 가상 장치로써 내부 네트워크끼리의 연결은 가상 장치의 성능이 그대로 나오지만 외부 네트워크와의 연결은 실제 장치를 반드시 거쳐야 하기 때문에 해당 성능을 뛰어넘을 수는 없음

○ 기타 장치 : 100% 가상 장치들이며 나머지 장치들은 성능적인 부분에는 큰 영향 없음

❒ 현재 사용되는 가상 머신(VM:Viratual Machine) 분석

○ VMware

- 1999년 버전 1.0이 출시된 후 2012년 8월 9.0 버전이 출시됨
- 현재 가장 폭 넓은 Host OS를 지원하여 가장 많이 사용되고 있는 에뮬레이터임
- Host OS (VMware 설치 가능한 운영 체제)
  - Windows Xp 이상
  - Windows 2000 Server이상
  - Asianux 3.0 CentOS 5.0이상

- Mandriva Linux 4.0이상
- OpenSUSE 10.0 이상
- Oracle Linux 5.0이상
- OSX 10.4 이상
- RedHat Enterprise Linux 2.1 이상
- RedHat Linux 7 이상
- SUSE 8이상
- SUSE Linux Exterprise 8이상
- Ubuntu 5.0 이상

• Guest OS (VMware에서 설치를 지원하는 운영 체제)
- Windows Server 2012
- Windows 8
- Windows Server 2008 R2
- Windows 7
- Windows Server 2008
- Windows Vista
- Windows Server 2003
- Windows XP
- Windows 2000
- Windows NT 4.0
- Windows ME
- Windows 98
- Windows 95
- MS-DOS 6.22 and Windows 3.1x

○ VirtualBox
• 2007년 이노텍 Virtual OSE(Open Source Edition)에서 처음 출시한 후 2013년 7월 4.2.16 버전을 출시함
• 썬 마이크로시스템즈는 VirtualBox를 개발한 회사인 이노텍(Innotek)을 인수하였으나, 2009년 4월 20일 오라클이 썬 마이크로시스템즈를 인수하면서 현재 Oracle VM VirtualBox로 배포하고 있으며 특별한 기능은 다음과 같음
- VMWare, Virtual PC에 비해서 기능적으로 떨어짐
- 원격 데스크톱 프로토콜 (RDP), iSCSI 지원, RDP를 거치는 원격 장치의 USB 지원과 같이 원격으로 가상 컴퓨터를 제어하는 기능임
- 인텔의 하드웨어 가상화 VT-x와 AMD의 AMD-V를 지원함
- 하드 디스크는 VDI (가상 디스크 이미지:Virtual Disk Image) 디지털 컴포넌

트로 불리는 특별한 컨테이너 포맷으로 에뮬레이트함

- Host OS (VirtualBox 설치 가능한 운영 체제)
  - Windows
  - Linux
  - Macintosh
  - Solaris
- Guest OS (VirtualBox에서 설치를 지원하는 운영 체제)
  - Windows NT 4.0
  - Windows 2000
  - Windows XP
  - Windows Server 2003
  - Windows Vista
  - Windows 7
  - DOS/Windows 3.x
  - Linux (2.4 and 2.6)
  - Solaris
  - OpenSolaris
  - OS/2
- Open BSD

○ Hyper-V (Virtual PC)

- Connectix에서 2006년 7월 처음 출시된 Hyper-V는 MS사에 인수되었으며 특징은 다음과 같음
  - USB 지원
  - 리다이렉션 : 플래시 드라이브, 디지털 카메라와 같은 주변 기기를 연결하고 게스트 운영체제에서 호스트 운영체제의 프린터로 인쇄 가능
  - 보이지 않는 애플리케이션 표시 및 실행 : 윈도 XP 모드의 애플리케이션을 윈도 7 바탕 화면에서 직접 실행
  - 멀티스레딩 지원 : 여러 개의 가상 머신을 동시에 실행하며 각 스레드에서 처리하므로 성능과 안정성을 개선
  - 스마트 카드 리다이렉션 : 호스트에 연결된 스마트 카드를 사용
  - 윈도 탐색기 통합 : 모든 가상 머신을 하나의 탐색기 폴더 (%USER%\Virtual Machines)에서 관리
- Host OS (Hyper-V 설치 가능한 운영 체제)
  - Windows 8
  - Window server 2008 이상

- OS X 10 (맥OS X)

• Guest OS (Hyper-V에서 설치를 지원하는 운영 체제)
- Windows Server 2008 R2
- Windows 7
- Windows Server 2008
- Windows Vista
- Windows Server 2003
- Windows XP

○ RedHat Enterprise Linux 가상화

• Host OS (RedHat Enterprise Linux 가상화 설치 가능한 운영 체제)
- RHEL 6 64비트 x86-64(KVM)
- RHEV-H 6
- RHEL 5 64비트 x86-64 (KVM)1
- RHEV-H 5
- RHEL 5 64비트 x86-64(Xen)
- RHEL 5 32비트 x86-32(PAE, Xen)
- RHEL 5 IA64

• Guest OS (RedHat Enterprise Linux 가상화에서 설치 지원하는 운영 체제)
- Windows Server 2008 R2
- Windows 7
- Windows Server 2008
- Windows Vista
- Windows Server 2003
- Windows XP

### (3) 애플리케이션 에뮬레이션

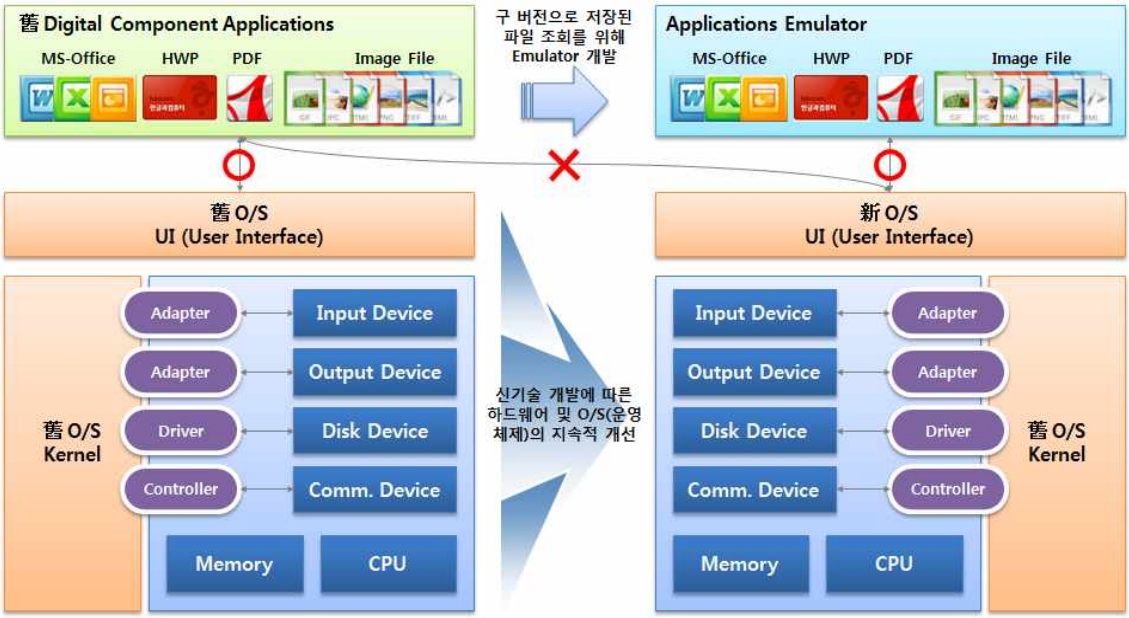
❒ 구형 하드웨어 및 구형 운영체제 기반의 디지털 컴포넌트에 대한 전자기록물을 조회할 수 있도록 신형 운영체제 기반에서 구동되는 뷰어 소프트웨어 제공 방식

○ 구형 운영체제 기반에서 저장된 디지털 컴포넌트 조회를 위해 디지털 컴포넌트에 대한 새로운 뷰어 개발

○ 현재 스마트폰이나 스마트패드에서 제공하는 뷰어는 인쇄 렌더링을 통한 단순 이미지 조회 방식을 다수 제공하고 있으나 디지털 컴포넌트에 포함된 렌더링 기능성에 대한 속성 정보 조회가 가능하도록 문서의 원문 포맷을 조회하는 방식으로 제공되어야 함

❒ 애플리케이션 에뮬레이션 개념도

○ 신형 하드웨어에 설치된 신형 운영체제에서 실행이 가능한 에뮬레이터(뷰어)를 개발하여 디지털 컴포넌트를 조회하는 방식




<그림 3-133> 애플리케이션 에뮬레이션 개념

❒ 현재 사용되는 뷰어 분석

○ 현재 윈도우 운영체제를 제외한 타 운영체제(안드로이드, iOS, Linux, MacOS)에서 개발된 한/글(HWP), 파워포인트(PPT), 엑셀(XLS) 포맷에 대한 뷰어를 조사한 결과 아래의 표와 같음

○ 다음 표에 조사된 뷰어 외에도 다양한 뷰어가 제공되는데 대부분의 뷰어는 원문을 인쇄 렌더링한 문서보존포맷(PDF/A) 형태로 제공됨을 확인함

<표 3-57> 애플리케이션 뷰어

| 포맷 | 안드로이드 用 | iOS 用 | linux 用 | Mac PC 用 |
|---|---|---|---|---|
| HWP | • 한컴오피스 모바일 뷰어 안드로이드 에디션 | • 한컴오피스 모바일 뷰어iOS 에디션 | • 한컴오피스 뷰어 | • 한컴오피스 뷰어 |
| | • 사이냅 뷰어 아래아한글 에디션 | | | |

| | | | | |
|---|---|---|---|---|
| | | • 폴라리스 오피스 | | |
| PPT | | | | • MS Office |
| | | | • Apache OpenOffice 임프레스 | • Apache OpenOffice 임프레스 |
| | • 씽크프리 오피스 모바일 뷰어 | • 씽크프리 오피스 모바일 뷰어 | | |
| | | • 폴라리스 오피스 | | |
| XLS | | | | • MS Office |
| | | | • Apache OpenOffice 캘크 | • Apache OpenOffice 캘크 |
| | • 씽크프리 오피스 모바일 뷰어 | • 씽크프리 오피스 모바일 뷰어 | | |
| | | • 폴라리스 오피스 | | |

### (4) 에뮬레이션 방안 비교

❒ 운영체제 에뮬레이션 방식과 애플리케이션 에뮬레이션 방식의 장단점 비교

<표 3-58> 에뮬레이션 방안 비교

| 항목 | 디바이스 & 운영체제 에뮬레이션 | 애플리케이션 에뮬레이션 |
|---|---|---|
| 방식 | • 신규 운영체제기반에서 CAMS에 등록된 디지털 컴포넌트를 조회할 수 있도록 구형 운영체제를 구동할 수 있는 VM(VirtualMachine) 제공 | • 신규 운영체제 기반에서 CAMS에 등록된 디지털 컴포넌트를 조회 할 수 있도록 DigitalComponent에 대한 Emulator 제공 |
| 장점 | • 애플리케이션 Format에 대한 정보 관리 불필요<br>• 애플리케이션 종류 및 수에 관계없이 에뮬레이션 가능 | • 애플리케이션 S/W에 대한 보존 불필요<br>• 상대적으로 폰트 관리가 용이하여 문서 형식(원본성) 유지가 가능<br>• 전자기록물에 포함된 디지털 컴포넌트 조회 기능 및 구성이 상대적으로 간단함 |
| 문제점 | • 운영체제에 종속된 애플리케이션 S/W(라이선스,버전) 관리가 필요하며 폰트 관리가 복잡하여 원본 형식 유지가 어려움<br>• 중첩(Stacked)방식의 가상화(VM) 전략을 사용할 경우 전자기록물에 포함된 디지털 컴포넌트 조회기능이 상대적으로 복잡하며 시간이 지날수록 속도 저하 | • 신규 운영체제에 대하여 DigitalComponent별Emulator를 개발하므로 개발 대상 증가<br>• 애플리케이션 Format에 대한 정확한 정보와 관리 필요<br>• 애플리케이션 개발사의 적극적인 협조 필요 |

○ 각 방안에 대한 장단점을 분석하면 운영체제 에뮬레이터를 적용하면 다양한 응용 소프트웨어에 대한 에뮬레이션 서비스가 가능하지만 하드웨어 성능이 현저히 저하되는데 반해 애플리케이션 에뮬레이터는 개별 애플리케이션에 대한 뷰어를 개발하기 때문에 개발 대상이 증가하지만 하드웨어 성능에 영향을 미치지 않는다는 점이 가장 큰 특징으로 볼 수 있음

○ 따라서, 국가기록원에서 에뮬레이션 서비스가 필요한 대상에 대하여 보다 구체적인 조사 분석을 통해 에뮬레이션 방식을 선정할 필요가 있음

## 다. 국가기록원의 한국형 에뮬레이션 대상 연구

❒ 국가기록원의 에뮬레이션 전략 수립 대상

○ 2015년부터 공공기관의 기록관리시스템(RMS)에서 10년이 경과한 전자기록물에 대하여 장기(30년, 준영구, 영구) 보존이 필요한 전자기록물을 국가기록원의 “중앙영구기록물관리시스템(CAMS)로 이관함

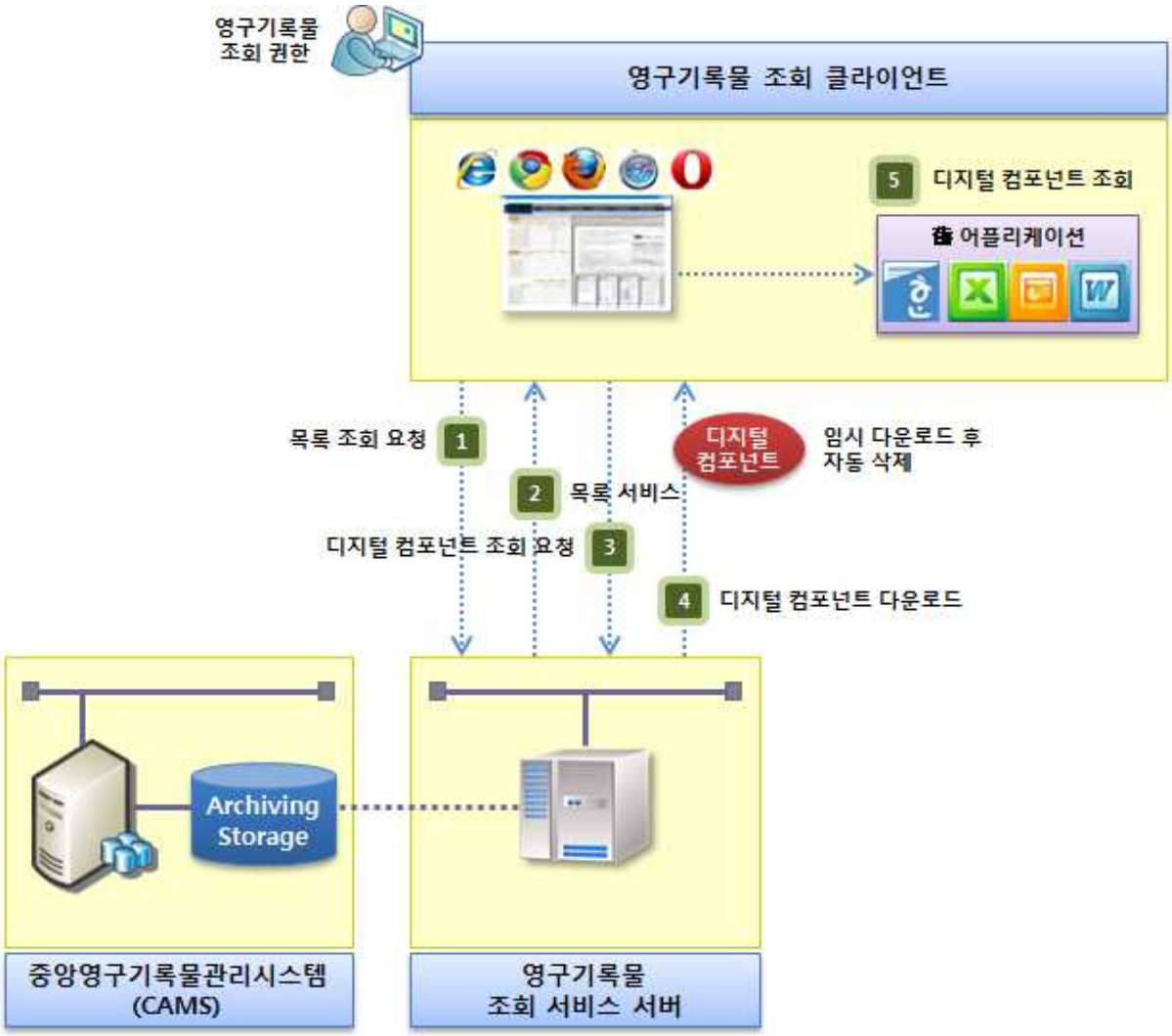
○ 본 연구의 에뮬레이션 전략 수립 대상은 2015년부터 공공기관에서 이관되어 “중앙영구기록물관리시스템(CAMS)”에 등록되는 전자기록물을 대상으로 하며 전자기록물에 포함된 디지털 컴포넌트의 원문과 문서보존포맷(PDF/A) 중 원문에 대한 에뮬레이션을 대상으로 정의

○ 마지막으로 렌더링 기능성을 포함한 디지털 컴포넌트에 대한 에뮬레이션을 대상으로 하며 이미지 형식의 디지털 컴포넌트를 비롯하여 표준 포맷 문서는 제외함

❒ 본 연구를 통해 제안하는 “에뮬레이션 기반의 디지털 컴포넌트 조회 서비스”는 목록 정보 및 원문의 조회 권한이 부여된 국가기록원 내부 사용자를 대상으로 하며 일반인을 대상으로 한 공개 서비스는 별도의 에뮬레이션 서비스 전략이 필요

❒ 앞서 살펴본 바와 같이 에뮬레이션 전략은 크게 운영체제 에뮬레이션 방안과 애플리케이션 에뮬레이션 방안이 존재하는데 운영체제 에뮬레이션 방안은 에뮬레이터 설치 위치(서버, 클라이언트)에 따라 서비스 방식이 상이하므로 본 연구에서는 현행 영구기록물 조회 서비스를 위한 3가지 형식의 에뮬레이션 방안을 제시하고자 함

❒ 다음 그림은 현재 “중앙영구기록물관리시스템(CAMS)”에 등록된 디지털 컴포넌트에 대한 조회 서비스 구성도임




<그림 3-134> 현행 디지털 컴포넌트 조회 서비스

○ 중앙영구기록물관리시스템의 Archiving Storage에 등록된 전자기록물의 디지털 컴포넌트를 조회하기 위해서는 전자기록물에 대한 조회 권한을 부여받은 사용자에 한하여 다음과 같은 절차로 서비스되며 임시 다운로드 된 디지털 컴포넌트는 삭제되어야 함

① CAMS 서버에 전자기록물 목록 요청
② CAMS 서버에서 전자기록물 목록 서비스
③ 전자기록물에 포함된 디지털 컴포넌트 조회 요청
④ 선택된 디지털 컴포넌트 '임시' 다운로드
⑤ 사용자 PC에서 디지털 컴포넌트 조회 후 자동 삭제

○ 전자기록물에 포함된 디지털 컴포넌트의 대외 유출을 막기 위해 문서보존포맷(PDF/A)으로 변환된 디지털 컴포넌트는 뷰어를 통하여 서비스됨

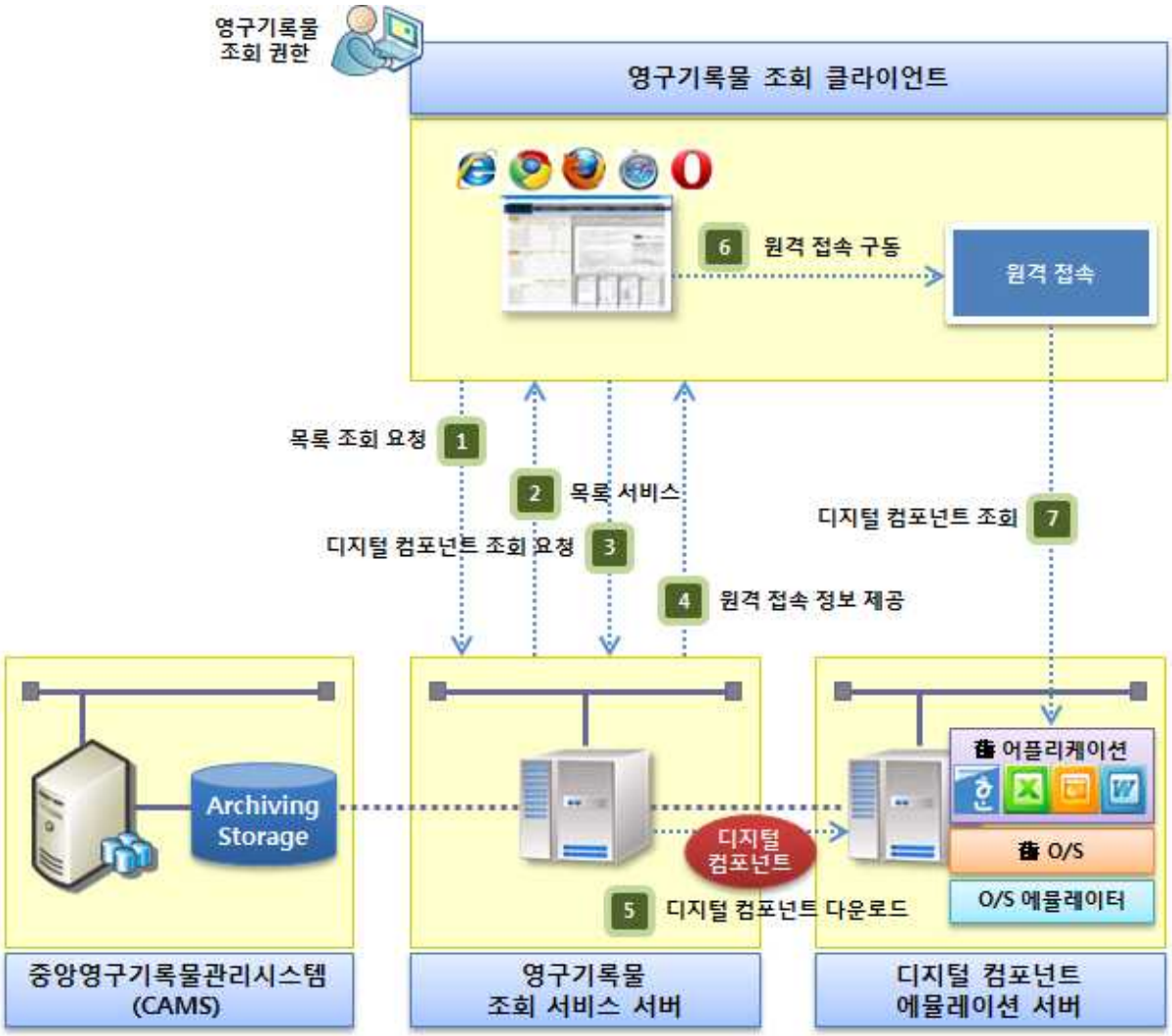
### 라. 국가기록원의 한국형 에뮬레이션 방안

#### (1) 서버 기반의 운영체제 에뮬레이션 방안

❒ 중앙영구기록물관리시스템의 Archiving Storage에 등록된 전자기록물을 서버 기반의 운영체제 에뮬레이션 방식으로 디지털 컴포넌트를 조회하기 위해서는 전자기록물에 대한 조회 권한을 부여받은 사용자에 한하여 다음 그림과 같은 절차로 서비스되며 임시 다운로드 된 디지털 컴포넌트는 삭제되어야 함

① CAMS 서버에 전자기록물 목록 요청
② CAMS 서버에서 전자기록물 목록 서비스
③ 전자기록물에 포함된 디지털 컴포넌트 조회 요청
④ 디지털 컴포넌트 에뮬레이션 서버의 원격 접속 정보 제공
⑤ 디지털 컴포넌트 에뮬레이션 서버로 선택된 디지털 컴포넌트 '임시' 다운로드
⑥ 디지털 컴포넌트 에뮬레이션 서버에 원격 접속
⑦ 원격 접속 상태에서 디지털 컴포넌트 조회 후 자동 삭제

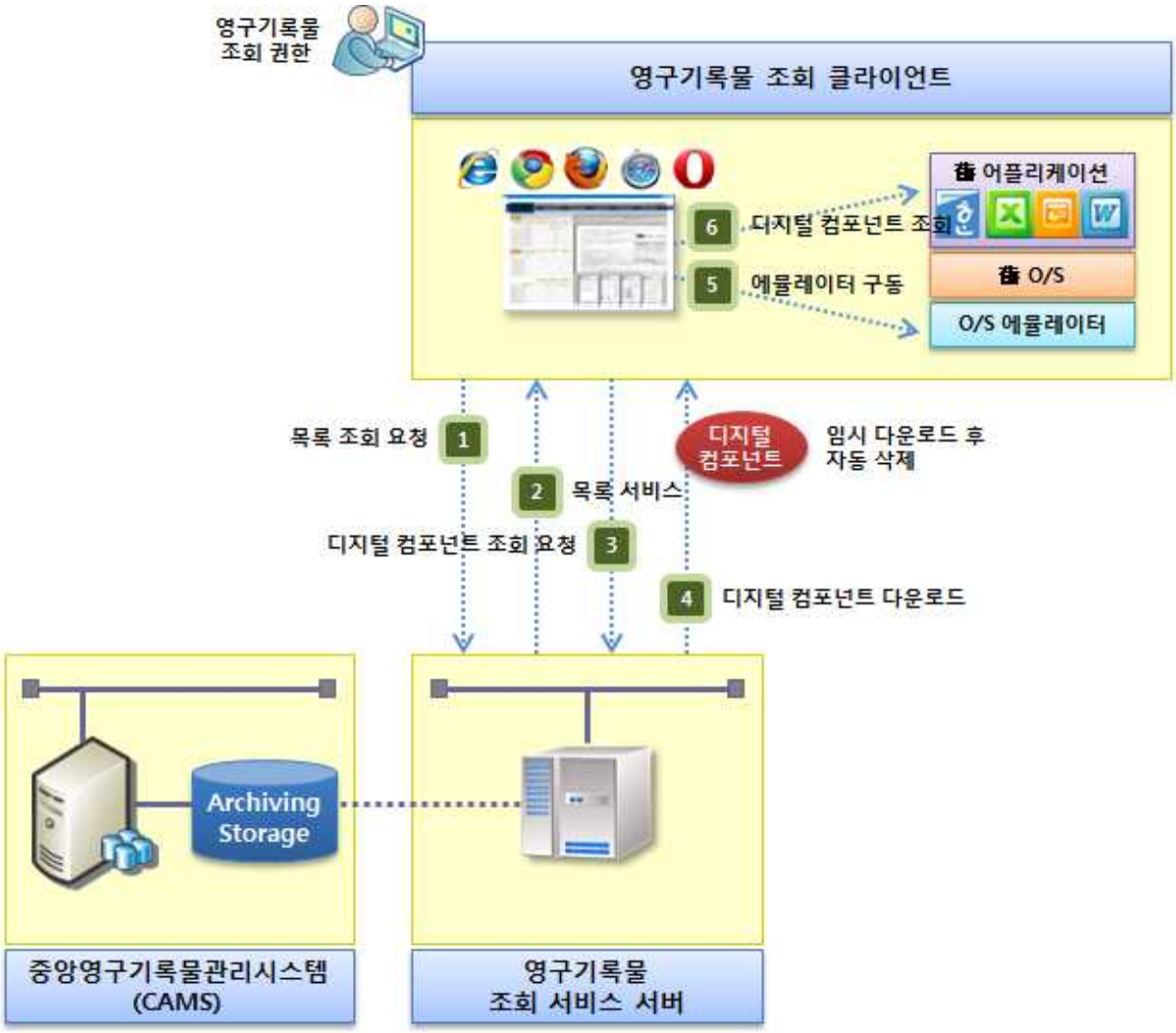
○ 디지털 컴포넌트 에뮬레이션 서버는 구형 애플리케이션의 실행을 지원하기 위해 운영체제 에뮬레이터 및 구형 운영체제가 설치되어야 함
○ 서버에 구형 운영체제를 설치하기 위해서는 1:1로 에뮬레이터를 설치하기 때문에 에뮬레이션 대상 운영체제의 수가 증가하면 설치하는 에뮬레이터의 수가 비례하여 증가
○ 운영체제 에뮬레이터 방식에서는 서버의 '디지털 컴포넌트 Configuration 서비스' 기능과 클라이언트의 '원격 접속 서비스' 기능이 부가적으로 필요함




<그림 3-135> 서버 O/S 에뮬레이터 기반의 디지털 컴포넌트 조회 서비스

### (2) 클라이언트 기반의 운영체제 에뮬레이션 방안

❒ 중앙영구기록물관리시스템의 Archiving Storage에 등록된 전자기록물을 클라이언트 기반의 운영체제 에뮬레이션 방식으로 디지털 컴포넌트를 조회하기 위해서는 전자기록물에 대한 조회 권한을 부여받은 사용자에 한하여 위의 그림과 같은 절차로 서비스되며 임시 다운로드 된 디지털 컴포넌트는 삭제되어야 함




<그림 3-136> 클라이언트 O/S 에뮬레이터 기반의 디지털 컴포넌트 조회 서비스

① CAMS 서버에 전자기록물 목록 요청
② CAMS 서버에서 전자기록물 목록 서비스
③ 전자기록물에 포함된 디지털 컴포넌트 조회 요청
④ 선택된 디지털 컴포넌트 '임시' 다운로드
⑤ 클라이언트에서 에뮬레이터 및 구형 운영체제 실행
⑥ 에뮬레이터 기반에서 디지털 컴포넌트 조회 후 자동 삭제

○ 구형 운영체제를 설치하기 위해서는 1:1로 에뮬레이터를 설치하기 때문에 에뮬레이션 대상 운영체제의 수가 증가하면 설치하는 에뮬레이터의 수가 비례하여 증가
○ 구형 애플리케이션의 실행을 지원하기 위해 사용자는 자신의 PC에 운영체제 에뮬레이터 및 구형 운영체제를 설치하여야 하므로 사용이 복잡할 뿐만 아니라

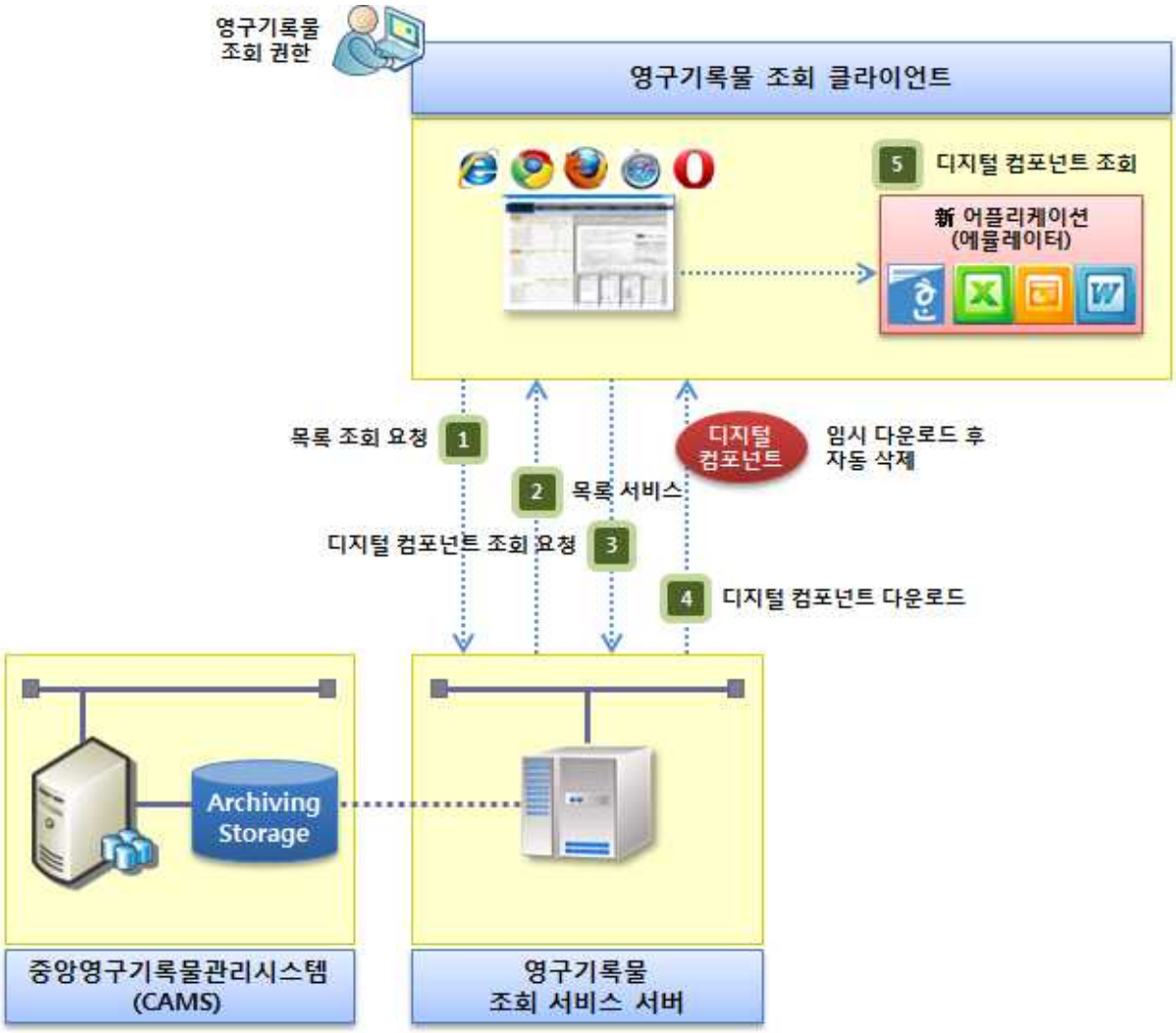
설치에 필요한 라이센스 정책 수립 필요

### (3) 애플리케이션 에뮬레이션 방안

❒ 현재까지 개발된 대부분의 애플리케이션은 클라이언트에서 실행되는 방식이므로 클라이언트에 에뮬레이터(뷰어)를 설치하는 방식으로 제안

❒ 애플리케이션에 대한 에뮬레이터(뷰어)는 현재 개발되어 서비스되는 뷰어와 같이 하나의 뷰어에서 다양한 포맷의 디지털 컴포넌트를 조회하도록 개발되어야 함

❒ 중앙영구기록물관리시스템의 Archiving Storage에 등록된 전자기록물을 애플리케이션 에뮬레이션 방식으로 디지털 컴포넌트를 조회하기 위해서는 전자기록물에 대한 조회 권한을 부여받은 사용자에 한하여 다음 그림과 같은 절차로 서비스되며 임시 다운로드 된 디지털 컴포넌트는 삭제되어야 함




<그림 3-137> 애플리케이션 에뮬레이터 기반의 디지털 컴포넌트 조회 서비스

① CAMS 서버에 전자기록물 목록 요청
② CAMS 서버에서 전자기록물 목록 서비스
③ 전자기록물에 포함된 디지털 컴포넌트 조회 요청
④ 선택된 디지털 컴포넌트 '임시' 다운로드
⑤ 사용자 PC에서 에뮬레이터를 통해 디지털 컴포넌트 조회 후 자동 삭제

○ 신형 운영체제가 출시되면 구형의 디지털 컴포넌트를 조회할 수 있도록 에뮬레이터 개발이 필요하므로 구형 디지털 컴포넌트에 대한 포맷 정보에 대한 보존 관리가 필요
○ 구형 애플리케이션의 실행을 지원하기 위해 사용자는 자신의 PC에 에뮬레이터를 설치하여야 하므로 설치에 필요한 라이센스 정책 수립 필요

❒ 그러나, 현재 다양한 연구 및 개발이 진행되는 클라우드 컴퓨팅 개념의 애플리케

이션에 대한 에뮬레이터는 서버에 설치하는 개념으로 구성 필요

## 마. 한국형 에뮬레이션 추진 전략 제안

### (1) 한국형 에뮬레이션 전략 개요

❒ 한국형 에뮬레이션 대상 선정

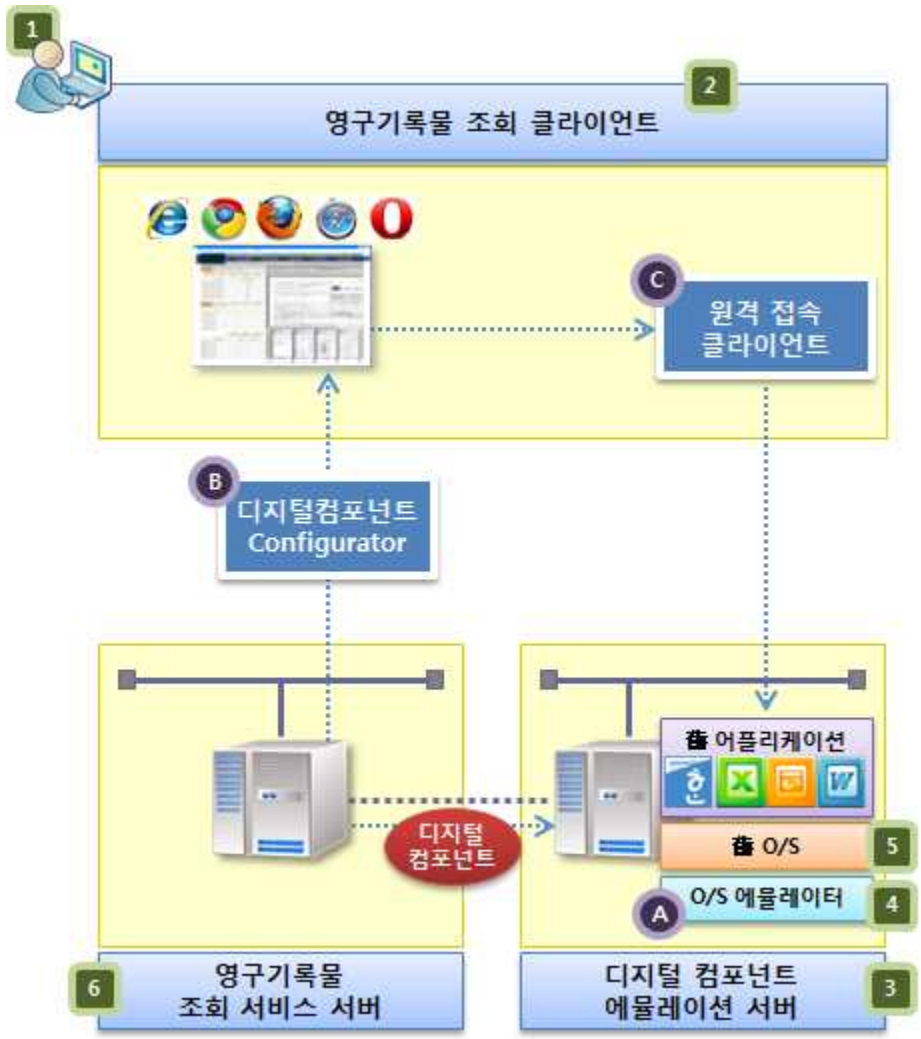
○ 2015년부터 공공기관의 기록관리시스템(RMS)으로부터 이관되어 국가기록원의 “중앙영구기록물관리시스템(CAMS)”에 장기 보존되는 전자기록물에 포함된 디지털 컴포넌트의 에뮬레이션 필요
○ 본 연구의 한국형 에뮬레이션 전략은 국가기록원 “중앙영구기록물관리시스템(CAMS)”의 Archiving Storage에 등록된 디지털 컴포넌트에 대한 에뮬레이션을 대상으로 함
○ 해외 사례 및 기술 정보 동향을 기반으로 다양한 에뮬레이션 방안에 대하여 연구하였으나 그 중 “서버 기반의 운영체제 에뮬레이션 방안”과 “애플리케이션 에뮬레이션 방안”을 중심으로 국가기록원의 한국형 에뮬레이션 구축 전략을 제시하고자 함

❒ 앞에서 살펴본바와 같이 에뮬레이션 방안은 각각의 장단점을 갖고 있으므로 각 방안에 대한 추진 전략을 검토한 후 에뮬레이션이 필요한 시점에서 상세한 조사 분석을 통해 한국형 에뮬레이션 전략을 확정하는 것이 바람직하다고 판단

### (2) 한국형 에뮬레이션 추진 전략 제안

❒ “서버 기반의 운영체제 에뮬레이션” 추진 전략

○ 본 연구에서는 아래 그림과 같이 사용자, 클라이언트 운영체제, 서버 운영체제, 에뮬레이션이 필요한 구형 운영체제, 구형 애플리케이션 및 대상 디지털 컴포넌트 등 6개 항목에 대한 전제 조건을 제시하고 이에 대한 추진 전략 수립




<그림 3-138> 서버 기반의 운영체제 에뮬레이션 구성도

○ 본 연구에서는 아래의 표에 제시된 항목별 범위에 대한 에뮬레이션 개발 범위를 제안하고자 함

<표 3-59> 서버 기반의 운영체제 에뮬레이션 범위

| 번호 | 항목 | | 범위 | 비고 |
|---|---|---|---|---|
| 1 | 에뮬레이션 사용자 | | 국가기록원 내부 사용자 | |
| 2<br>3<br>4 | 운영 체제<br>(O/S) | 클라이언트 | 윈도우 7 이상 | |
| | | 서버 | 윈도우 8.1 이상 | Host O/S |
| | | 에뮬레이션 | 윈도우 7 | Guest O/S |
| 5 | 애플리케이션 | 종류 | 한/글, 파워포인트, 엑셀, 워드 | |
| | | 버전 | 한/글 97, 한/글 2007<br>파워포인트 2003, 파워포인트 2007<br>엑셀 2003, 엑셀 2007 | |

| | | | | |
|---|---|---|---|---|
| | | | 워드 2003, 워드 2007 | |
| | | 포맷 | HWP 3.0, HWP 5.0<br>PPT, PPTX<br>XLS (BIFF 8.x), XLSX<br>DOC, DOCX | |
| 6 | 디지털 컴포넌트 | 보존 매체 | Archiving Storage (CAMS) | |
| | | 생성 기한 | 2004년 이후 생성된 디지털 컴포넌트 | |

○ 서버 기반의 운영체제 에뮬레이션을 위해서는 다음 표와 같이 에뮬레이션 서버 도입과 운영체제 에뮬레이터, 디지털 컴포넌트 Configurator의 개발이 필요

**<표 3-60> 서버 기반의 운영체제 에뮬레이션 구성**

| 도입 또는 개발 항목 | | | 방식 | 내용 | 라이센스 |
|---|---|---|---|---|---|
| 서버 | | 하드웨어 | 도입 | | |
| | | 신형 운영체제 | 도입 | | 서버 라이센스 필요 |
| | | 구형 운영체제 | 도입 | | 서버 라이센스 필요 |
| | | 구형 애플리케이션 | 도입 | | 뷰어는 무상 지원 |
| | A | 에뮬레이터 | 개발 | 구형 운영체제 설치를 위한 VM (Virtual Machine) 서비스 | |
| | B | 디지털 컴포넌트 Configurator | 개발 | CAMS에 보존된 디지털 컴포넌트의 포맷 정보를 식별하여 에뮬레이션 대상 유무와 대상 애플리케이션 연계 | |
| 클라이언트 | C | 원격 접속 클라이언트 | 도입 | 에뮬레이션 서버에서 애플리케이션 실행이 가능하도록 원격 접속 서비스 | 운영체제에서 무상 지원 |

○ 서버 기반의 운영체제 에뮬레이터 개발을 위한 비용은 다음 표와 같음

- (조건 1) 현재 사용중인 운영 체제를 기반으로 한 에뮬레이터는 기 개발되어 무상으로 배포되고 있으며 소스 프로그램까지 자유롭게 활용이 가능하므로 대상에서 제외하고 마이크로소프트사의 윈도우 운영체제와 전혀 다른 운영체제가 출시되어 PC에서 운용되는 경우를 대상으로 함
- (조건 2) 개발 대상 에뮬레이터의 Host O/S는 신규 출시된 하드웨어 및 운영체제이고 Guest O/S는 마이크로소프트사의 윈도우 운영체제 중 1종을 대상으로 함
- (조건 3) 구형 애플리케이션 뷰어와 원격 접속 클라이언트는 무상 지원

- (조건 4) 대상 어플리케이션은 현재 가장 많이 사용되는 한/글과 마이크로소프트 오피스 어플리케이션을 대상으로 선정
- (조건 5) 모든 소프트웨어는 초기 개발 비용과 업그레이드를 위한 추가 비용이 발생하나 본 비용 계산에서는 초기 개발 비용만을 고려함

**<표 3-61> 서버 기반의 운영체제 에뮬레이터 개발 비용**

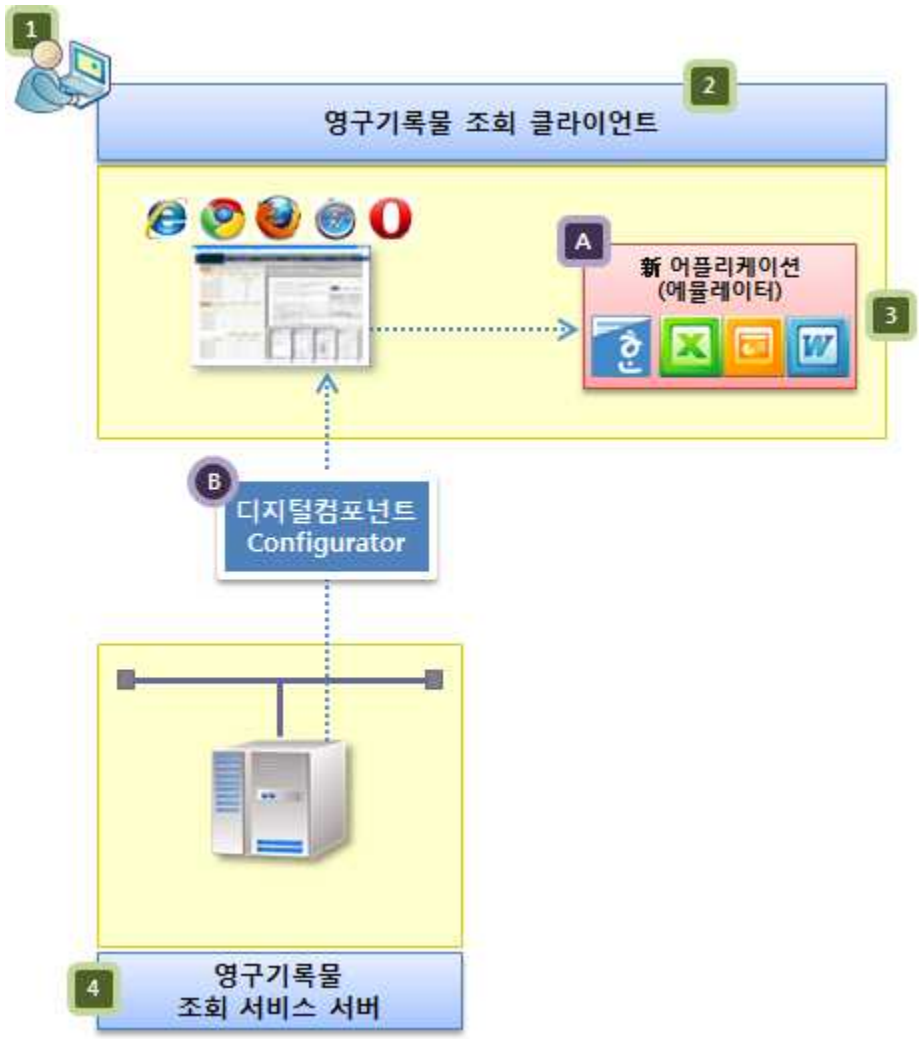
| 개발 항목 | | 개발 세부 기능 | 비용 | 비고 |
|---|---|---|---|---|
| 서버 | 에뮬레이터 | | 1,000,000,000 원 이상 | 초기 개발 비용 |
| | 디지털 컴포넌트 Configurator | 레지스트리 구축 | 500,000,000 원 | DFR 개선 |
| | | 포맷 식별기 | 300,000,000 원 | |
| | | 어플리케이션 연계 | 200,000,000 원 | 시스템 연계 |

※ 새롭게 출시된 하드웨어 및 운영 체제가 현재 가장 많이 사용되는 윈도우 운영 체제와 얼마만큼 호환성을 유지하는지가 에뮬레이터 개발 비용의 핵심 요소이기 때문에 차세대 하드웨어 구조와 운영 체제를 알지 못하는 현 시점에서 개발 비용 산정은 무의미하나 개략적인 비용을 추정하는 의미로 해석 필요.

※ 위에 제시한 5개의 조건을 기반으로 VirtualBox와 유사한 에뮬레이터를 개발할 때의 추정 비용을 위의 표에 나타냄

❒ “어플리케이션 에뮬레이션” 추진 전략

○ 본 연구에서는 아래 그림과 같이 사용자, 클라이언트 운영체제, 신형 애플리케이션 및 대상 디지털 컴포넌트 등 4개 항목에 대한 전제 조건을 제시하고 이에 대한 추진 전략 수립




<그림 3-139> 어플리케이션 에뮬레이션 구성도

○ 본 연구에서는 아래의 표에 제시된 항목별 범위에 대한 개발 범위를 제안하고자 함

<표 3-62> 어플리케이션 에뮬레이션 범위

| 번호 | 항목 | | 범위 | 비고 |
|---|---|---|---|---|
| 1 | 에뮬레이션 사용자 | | 국가기록원 내부 사용자 | |
| 2 | 운영 체제 (O/S) | 클라이언트 | 윈도우 7 이상 | |
| 3 | 어플리케이션 | 종류 | 한/글, 파워포인트, 엑셀, 워드 | |
| | | 버전 | 한/글 97, 한/글 2007<br>파워포인트 2003, 파워포인트 2007<br>엑셀 2003, 엑셀 2007<br>워드 2003, 워드 2007 | |
| | | 포맷 | HWP 3.0, HWP 5.0 | |

| | | | | |
|---|---|---|---|---|
| | | | PPT, PPTX<br>XLS (BIFF 8.x), XLSX<br>DOC, DOCX | |
| 4 | 디지털<br>컴포넌트 | 보존 매체 | Archiving Storage (CAMS) | |
| | | 생성 기한 | 2004년 이후 생성된 디지털 컴포넌트 | |

○ 어플리케이션 에뮬레이션을 위해서는 다음 표와 같이 클라이언트에 설치하는 에뮬레이터(뷰어)와 디지털 컴포넌트 Configurator 개발이 필요

<표 3-63> 어플리케이션 에뮬레이션 구성

| 도입 또는 개발 항목 | | | 방식 | 내용 | 라이센스 |
|---|---|---|---|---|---|
| 클라이언트 | A | 에뮬레이터 | 개발 | 하나의 에뮬레이터(뷰어)에서 다양한 포맷의 디지털 컴포넌트를 조회하도록 개발 | |
| 서버 | B | 디지털 컴포넌트 Configurator | 개발 | CAMS에 보존된 디지털 컴포넌트의 포맷 정보를 식별하여 에뮬레이션 대상 유무와 대상 애플리케이션 선정 | |

○ 어플리케이션 에뮬레이터 개발을 위한 비용은 다음 표와 같음

- (조건 1) 현재 사용중인 운영 체제를 기반으로 한 어플리케이션 에뮬레이터는 기 개발되어 무상으로 배포되고 있으므로 대상에서 제외하고 마이크로소프트사의 윈도우 운영체제와 전혀 다른 운영 체제가 출시되어 PC에서 운용되는 경우를 대상으로 함
- (조건 2) 개발 대상 에뮬레이터는 신규 출시된 하드웨어 및 운영 체제에서 실행되는 것을 조건으로 함
- (조건 3) 대상 어플리케이션은 현재 가장 많이 사용되는 한/글과 마이크로소프트 오피스 어플리케이션을 대상으로 선정
- (조건 4) 모든 소프트웨어는 초기 개발 비용과 업그레이드를 위한 추가 비용이 발생하나 본 비용 계산에서는 초기 개발 비용만을 고려함

<표 3-64> 어플리케이션 에뮬레이터 개발 비용

| 개발 항목 | | 개발 세부 기능 | 비용 | 비고 |
|---|---|---|---|---|
| 서버 | 에뮬레이터 | 한/글 에뮬레이터 | 500,000,000 원 | 초기 개발 비용 |
| | | MS 오피스 에뮬레이터 | 500,000,000 원 | 초기 개발 비용 |
| | 디지털 컴포넌트 Configurator | 레지스트리 구축 | 500,000,000 원 | DFR 개선 |

| | | | | |
|---|---|---|---|---|
| | | 포맷 식별기 | 300,000,000 원 | |
| | | 어플리케이션 연계 | 200,000,000 원 | 시스템 연계 |

※ 새롭게 출시된 하드웨어 및 운영 체제가 현재 가장 많이 사용되는 윈도우 운영 체제와 얼마만큼 호환성을 유지하는지가 에뮬레이터 개발 비용의 핵심 요소이기 때문에 차세대 하드웨어 구조와 운영 체제를 알지 못하는 현 시점에서 개발 비용 산정은 무의미하나 개략적인 비용을 추정하는 의미로 해석 필요.

※ 위에 제시한 4개의 조건을 기반으로 한/글과 MS 오피스 디지털 컴포넌트에 대한 뷰어와 유사한 에뮬레이터를 개발할 때의 추정 비용을 위의 표에 나타냄

### 3.4.2 에뮬레이터 프로토타입 개발

❒ 에뮬레이터 프로토타입 개발 방향
- ○ 해외 에뮬레이션 구축 사례인 KEEP-EF에 대한 실행 환경 분석
- ○ 한국형 에뮬레이션 전략에 기반한 에뮬레이션 프로토타입 시나리오 구성
- ○ 공공기관 전자기록물에 포함된 디지털 컴포넌트 중에서 렌더링 기능성을 포함한 한/글과 마이크로소프트사의 오피스 디지털 컴포넌트에 대한 에뮬레이션을 위해 운영체제 에뮬레이션 방안과 애플리케이션 에뮬레이션 방안에 대한 프로토타입 개발

#### 가. 해외 에뮬레이터 구동 사례

#### (1) KEEP-EF 에뮬레이터 개요

❒ 한국형 에뮬레이션 프로토타입 구현을 위해 먼저 유럽에서 선행되어 있는 KEEP-EF에 대한 실행 사례 분석을 수행함

❒ KEEP-EF(Emulation Framework)는 자신의 고유 환경에서 디지털 컴포넌트와 컴퓨터 프로그램을 렌더링 할 수 있고 현재의 컴퓨터 시스템에서 독립적으로 디지털 컴포넌트를 볼 수 있는 Look&Feel 가능성을 제공하며 지원 가능한 잠재적 컴퓨터 플랫폼 및 애플리케이션의 스펙트럼은 거의 무제한으로 설계되어 있음

❒ KEEP-EF를 구동하기 위해서는 EA(Emulator Archive)와 SWA(SoftWare Archive) 데이터베이스가 먼저 구축되어 있어야 함

○ EA 데이터베이스에는 Hardware 정보, Image Format(FAT12, FAT32등)정보, Emulator 정보, Emulator와 Hardware 연결 정보, Emulator과 Image Format 연결정보를 가지고 있음

○ SWA 데이터베이스에는 파일포맷, Application 정보, OS 정보, 플랫폼 정보, Image Format 정보, Software Image 정보와 파일 포맷과 Application, Operation System, 플랫폼을 연결하는 정보, Application과 Operating System, Software Image 정보, Operating System과 플랫폼 그리고 Software Image를 연결하는 정보를 가지고 있음

### (2) KEEP-EF 실행 사례[100]

○ KEEP-EF 기반에서 에뮬레이션을 실행하기 위해서는 먼저 에뮬레이션을 위한 환경을 구성(서버 및 클라이언트 실행)한 후 대상 디지털 컴포넌트를 선택하여 에뮬레이션을 실행하여야 함

❒ 에뮬레이션을 위한 환경 구성

① KEEP-EF EA(Emulation Archive) 서버 실행

② KEEP-EF SWA(SoftWare Archive) 서버 실행

③ KEEP-EF 에뮬레이션 클라이언트 실행

❒ 디지털 컴포넌트에 대한 에뮬레이션 실행

④ 에뮬레이션 대상 디지털 컴포넌트 선택

⑤ 대상 디지털 컴포넌트에 대한 Configuration 실행

- 선택된 디지털 컴포넌트에 대한 식별, 검증, 특성 정보를 추출
- 특성 정보를 이용하여 디지털 컴포넌트를 구동하기 위한 H/W와 S/W를 결정
- EF에서 가능한 최적의 환경을 선택
- EA(Emualator Archive)로부터 에뮬레이터를 선택한 후 디지털 컴포넌트를 포함한 디스크 이미지와 소프트웨어를 첨부 (Dioscuri, QEMU, VICE, UAE, Java CPC, BeebEm 등)
- SWA(SoftWare Archive)로부터 소프트웨어를 선택하고 주어진 디지털 컴포넌트를 디스크 이미지 형태로 변환. (FreeDOS, Damn Small Linux, Puppy Linux 등)

⑥ 선택된 에뮬레이터 실행

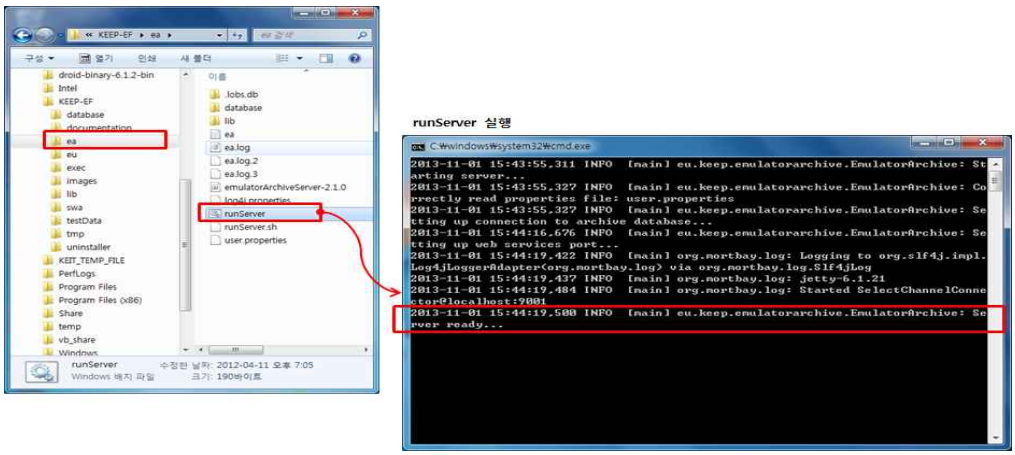
---

100) Architectural Design Document Emulation Framework Release 2.0.0 (February 2012)- Bram Lohman (Tessella), Edo Noordermeer (Tessella), System Maintenance Guide Emulation Framework Release 2.0.0 (February 2012)-Bram Lohman, Tessella), David Michel (Tessella), Edo Noordermeer (Tessella), System User Guide Emulation Framework Release 2.0.0 (February 2012)- Bram Lohman (Tessella), Jeffrey van der Hoeven (National Library of the Netherlands), Edo Noordermeer (Tessella), Test description and results document for Emulation Framework(January, 2012) Bart Kiers, Jeffrey van der Hoeven

⑦ 에뮬레이터(VM) 기반에서 구 운영체제 및 애플리케이션 실행하면 준비된 에뮬레이션 환경에서 디지털 컴포넌트 렌더링

❒ KEEP-EF EA(Emulation Archive) 서버 실행

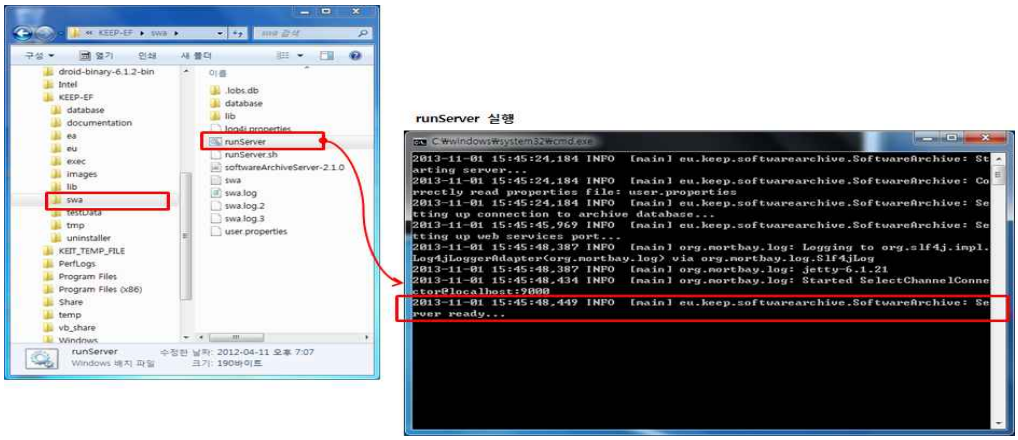
○ KEEP-EF\EA\runServer (배치파일)을 실행하면 EA 서버가 Background에서 실행




<그림 3-140> KEEP-EF EA(Emulation Archive) 서버 실행

❒ KEEP-EF SWA(SoftWare Archive) 서버 실행

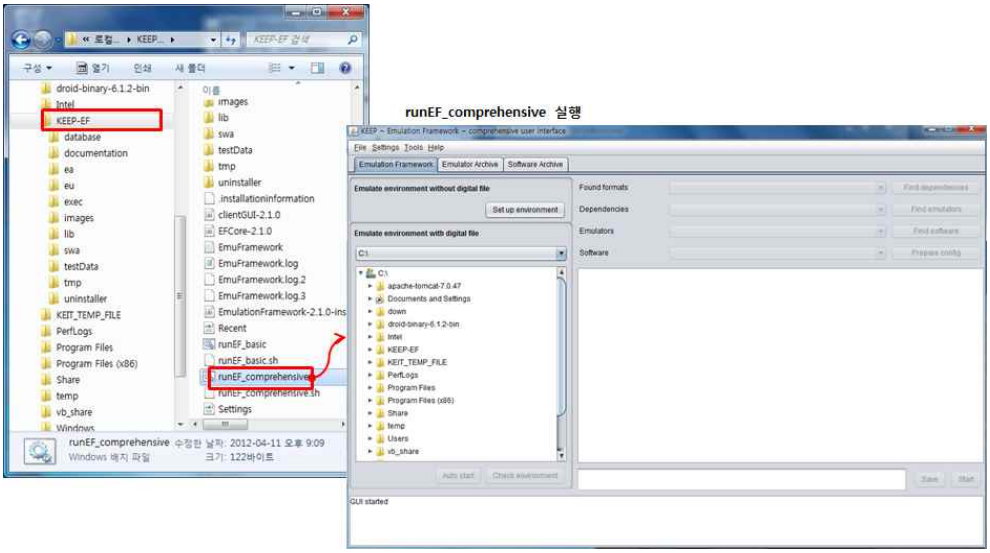
○ KEEP-EF\swa\runServer (배치파일)을 실행하면 SWA 서버가 Background에서 실행




<그림 3-141> KEEP-EF SWA(SoftWare Archive) 서버 실행

❒ KEEP-EF 에뮬레이션 클라이언트 실행

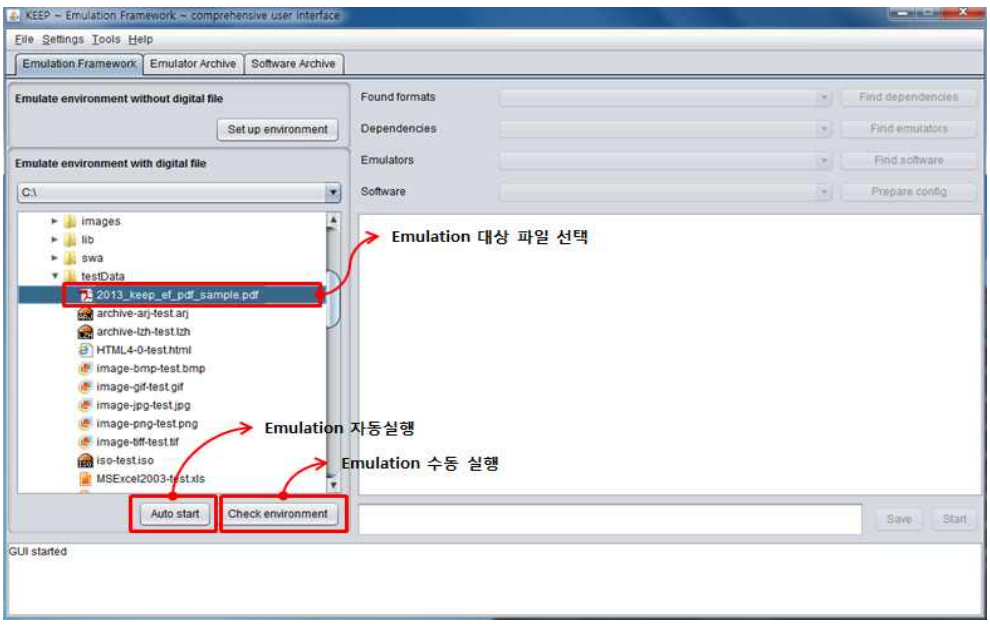
○ 에뮬레이션 클라이언트는 KEEP-EF\runEF_Comprehensive(배치파일)을 실행




<그림 3-142> KEEP-EF 에뮬레이션 클라이언트 실행

❒ 에뮬레이션 대상 디지털 컴포넌트 선택

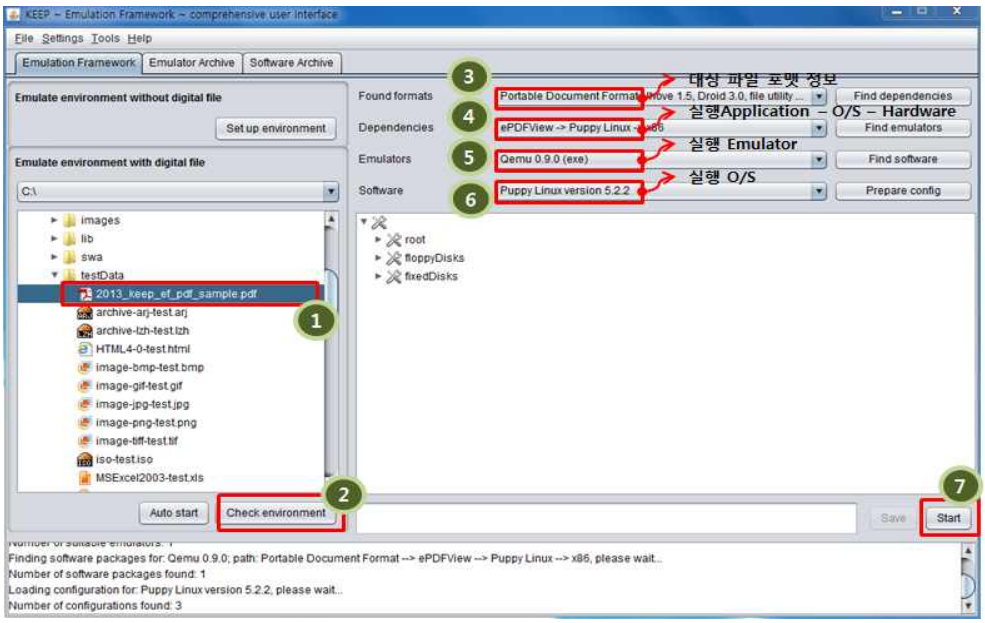
○ 에뮬레이션 대상 디지털 컴포넌트를 선택한 후 Auto start를 선택하면 에뮬레이션 과정을 자동으로 진행하고, Check environment 버튼을 선택하면 Configuration을 단계적으로 진행




<그림 3-143> 에뮬레이션 대상 디지털 컴포넌트 선택 화면

❒ 대상 디지털 컴포넌트에 대한 Configuration 실행

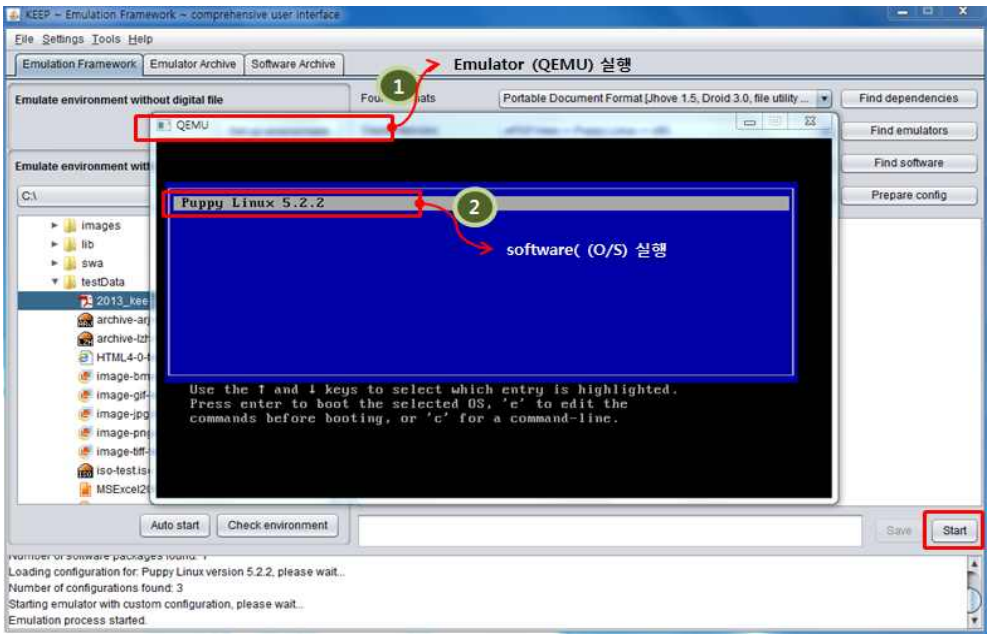
○ 대상 디지털 컴포넌트의 포맷 정보를 식별한 후 가장 적합한 Application - O/S - Hardware를 검색한 후 Emulator와 O/S를 추천함




<그림 3-144> 대상 디지털 컴포넌트에 대한 Configuration 실행 화면

❒ 선택된 에뮬레이터 실행

○ Configuration을 통해 추천한 Emulator 중에서 선택한 Emulator 실행

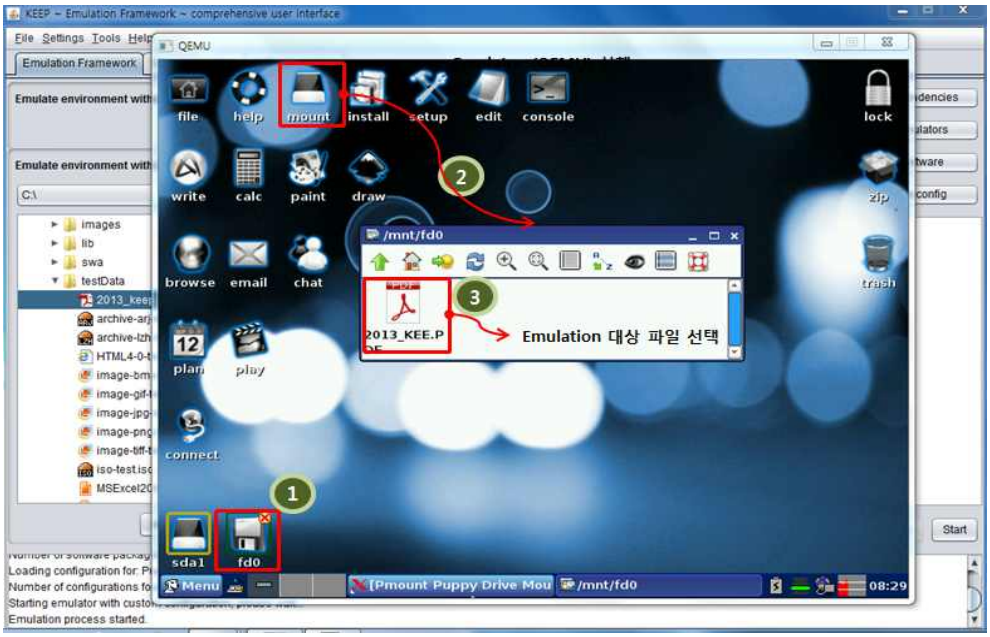



<그림 3-145> 선택된 에뮬레이터 실행 화면

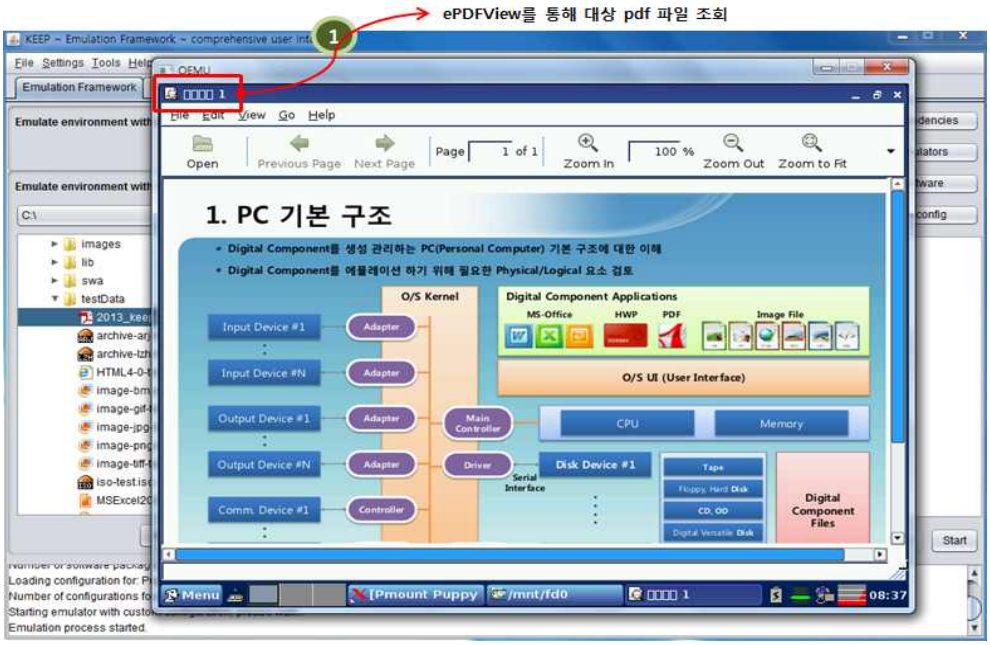
❒ 에뮬레이터(VM) 기반에서 구 운영체제 및 애플리케이션 실행

○ Emulator 기반에서 구 O/S 및 Application을 실행하여 해당 디지털 컴포넌트를 조회할 수 있도록 에뮬레이션 제공

○ 에뮬레이션 환경에서 아래 그림과 같이 해당 디지털 컴포넌트를 선택하면 조회가 가능




<그림 3-146> 구 운영체제 및 애플리케이션 실행 화면




<그림 3-147> 애플리케이션 실행 결과 화면

## 나. 한국형 에뮬레이터 프로토타입 개발

### (1) 프로토타입 개요

❒ 프로토타입 시스템 구성

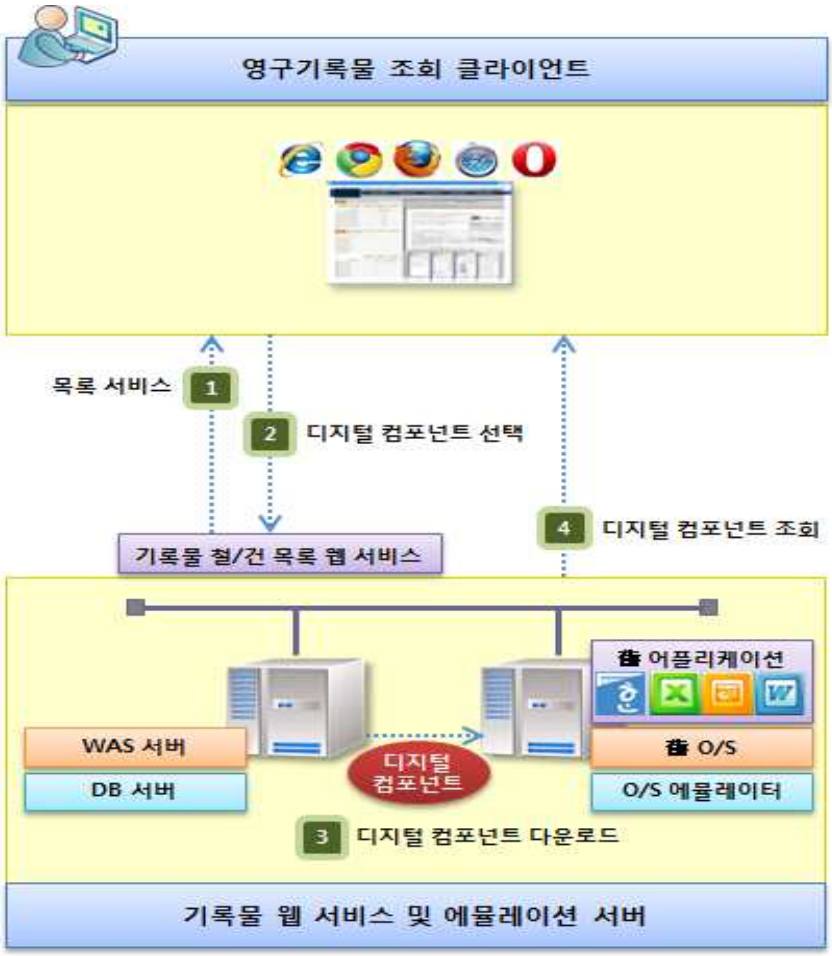
- ○ 프로토타입에서는 운영체제 에뮬레이션 방식과 애플리케이션 에뮬레이션 방식에 대한 서비스 제공
- ○ 운영체제 에뮬레이션은 윈도우7(Host O/S)에 Oracle의 VirtualBox를 설치한 후 윈도우XP(Guest O/S)에서 디지털 컴포넌트에 대한 조회 서비스를 구현
- ○ 애플리케이션 에뮬레이션은 '엑셀' 포맷에 대한 뷰어를 개발하여 서비스를 구현
- ○ 디지털 컴포넌트 Configurator는 MySQL, Tomcat 기반의 웹 서비스 형태로 구성되었으며 구형 운영체제와의 통신을 위해 VMServer를 개발함
- ○ 프로토타입 구축을 위해 도입 및 개발한 항목은 아래 표와 같음

<표 3-65> 프로토타입 에뮬레이션 구성

| 도입 또는 개발 항목 | | 프로토타입 | 방식 |
|---|---|---|---|
| 운영체제 에뮬레이션 | 신형 운영체제 | 윈도우 7 | 도입 |
| | 구형 운영체제 | 윈도우 XP | 도입 |
| | 구형 애플리케이션 | 한/글<br>파워포인트<br>엑셀<br>워드<br>훈민정음 | 뷰어 도입<br>(무상) |
| | 에뮬레이터 | VirtualBox (Oracle) | 도입 |
| | 디지털 컴포넌트 Configurator | | 개발 |
| 애플리케이션 에뮬레이션 | 에뮬레이터 (뷰어) | 엑셀 | 개발 |
| | 디지털 컴포넌트 Configurator | | 개발 |

### (2) 프로토타입 구성 시나리오

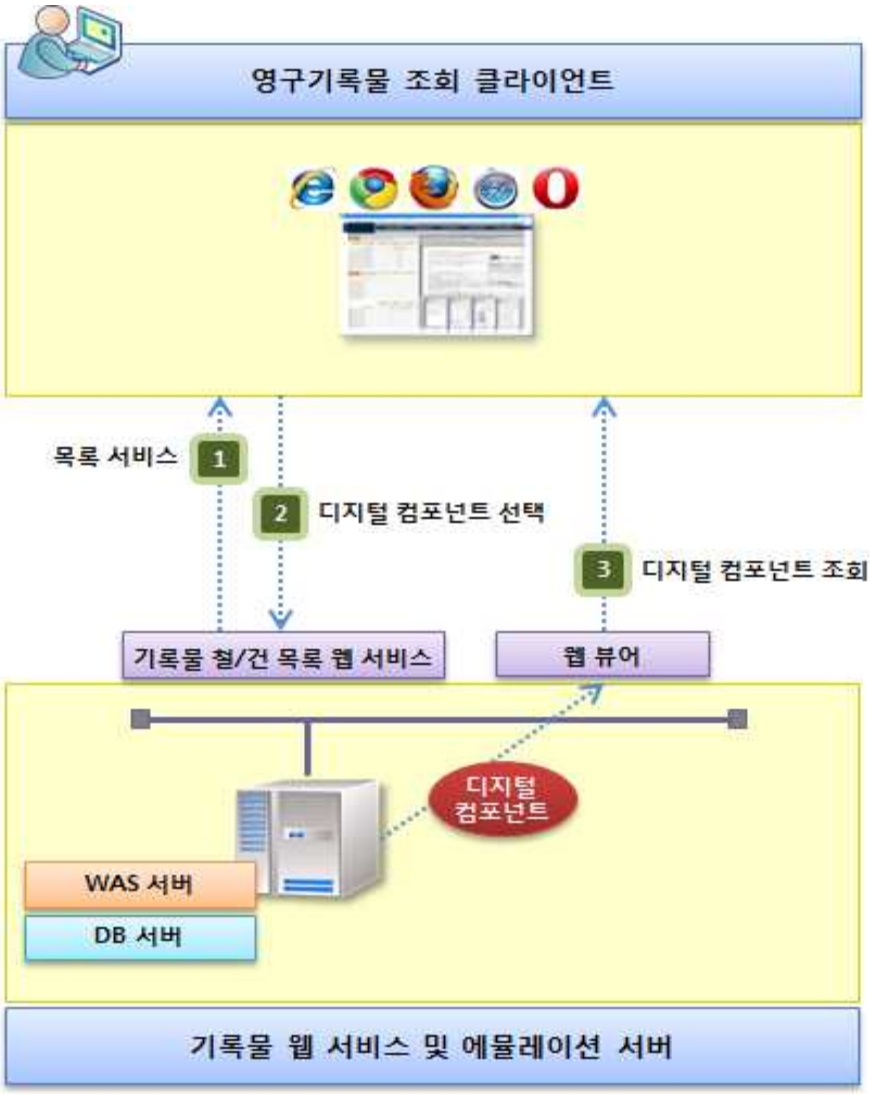
- ○ 다음 그림과 같이 프로토타입 시스템에서는 Oracle사의 VirtualBox를 도입하여 에뮬레이션 환경을 구축하였으며 기록물 철/건 목록에서 디지털 컴포넌트를 선택하였을 때 구형 애플리케이션을 자동으로 구동하기 위해 디지털 컴포넌트 Configurator와 VMServer를 개발하여 환경을 구축함




<그림 3-148> 운영체제 에뮬레이터 프로토타입 구성

○ 또한 다음 그림과 같이 프로토타입 시스템에서는 애플리케이션에 대한 에뮬레이션 실행을 위해 엑셀 포맷에 대한 웹 뷰어와 기록물 철/건 목록에서 디지털 컴포넌트를 선택하였을 때 웹 뷰어 에뮬레이터를 자동으로 구동하기 위해 디지털 컴포넌트 Configurator를 개발하여 환경을 구축함

○ 프로토타입은 운영체제 에뮬레이션과 애플리케이션 에뮬레이션에 대한 실행이 하나의 서버에서 구동되도록 구축하였는데 향후 웹 서비스 서버와 에뮬레이션 서버로 분리 가능함




<그림 3-149> 애플리케이션 에뮬레이터 프로토타입 구성

### (3) 프로토타입 실행 사례

❒ 에뮬레이션 서버 실행

① Oracle VirtualBox 실행

② 윈도우 XP(Guest O/S) 실행

③ VMServer 실행

❒ 웹 서비스 서버 실행

① 웹 서비스 DB 서버 실행 (MySQL)

② 웹 서비스 WAS 서버 실행 (Tomcat)

③ 클라이언트 브라우저(Chrome)를 통해 웹 서비스

❒ 운영체제 기반의 에뮬레이션 서비스

① 기록물에 포함된 디지털 컴포넌트 선택
② 디지털 컴포넌트에 대한 Configuration 실행 후 적합한 에뮬레이션 환경 추천
③ 에뮬레이션 환경을 선택하면 디지털 컴포넌트에 대한 조회 서비스

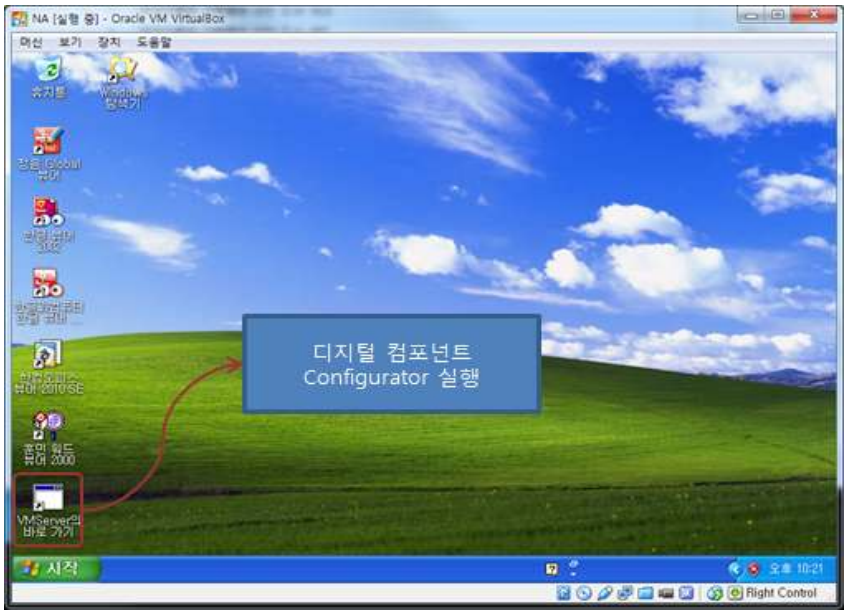
❒ 애플리케이션 뷰어 기반의 에뮬레이션 서비스
① 기록물에 포함된 디지털 컴포넌트 선택
② 뷰어 기반의 디지털 컴포넌트에 대한 조회 서비스

❒ 에뮬레이션 서버 실행
○ 운영체제 에뮬레이션을 위하여 신형 운영체제 (윈도우 7) 에서 VirtualBox 실행한 후 구형 운영체제 선택



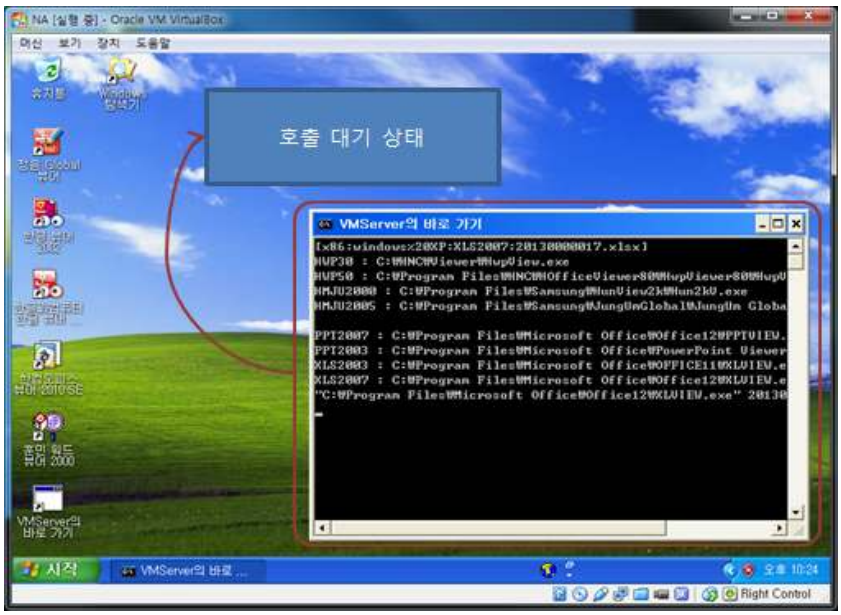

<그림 3-150> VirtualBox 실행 화면

○ Oracle VirtualBox 기반에서 구형 운영체제 (윈도우 XP) 실행한 후 웹 서비스 프로그램과의 통신을 위해 VMServer 프로그램을 실행




<그림 3-151> VirtualBox에서 윈도우 XP 실행 화면

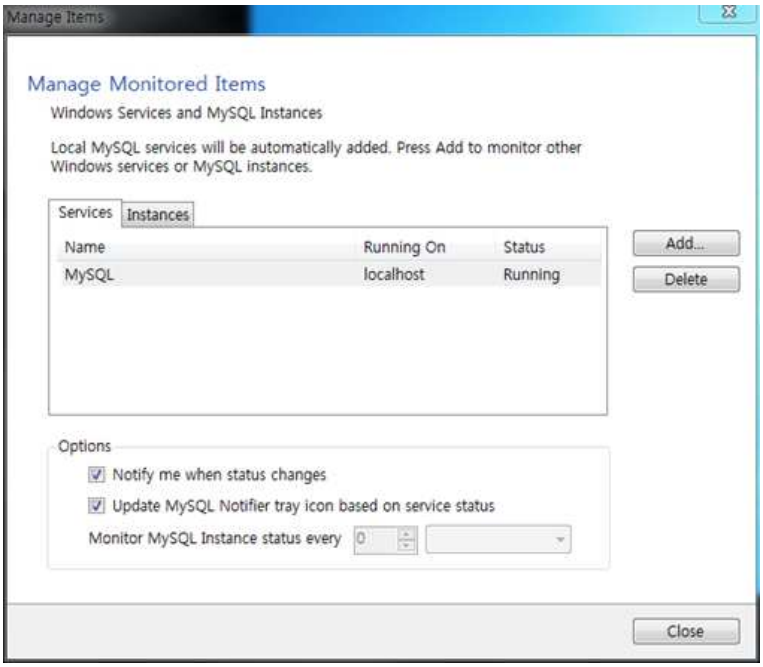
○ VMServer가 실행되면 웹 서비스 서버에서 선택된 디지털 컴포넌트에 대한 애플리케이션 실행 요청을 수신하여 애플리케이션을 실행함




<그림 3-152> VMServer 실행 화면

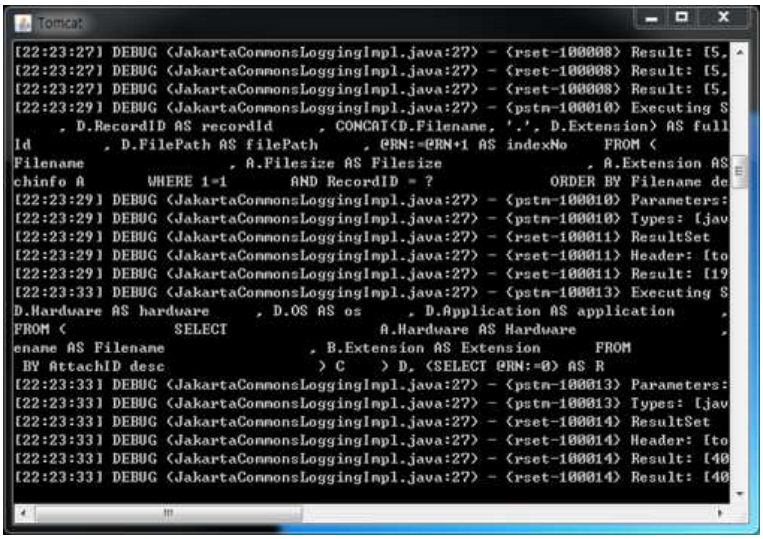
❒ 웹 서비스 서버 실행

○ 기록물 목록에 대한 웹 서비스를 위해 DB 서버 실행



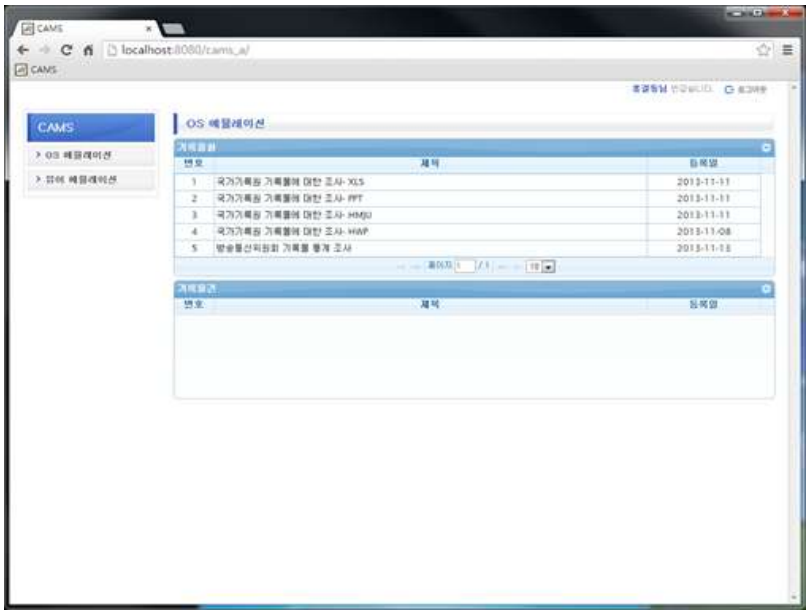

<그림 3-153> MySQL 실행 화면

○ 기록물 목록에 대한 웹 서비스를 위해 WAS 서버 실행




<그림 3-154> Tomcat 실행 화면

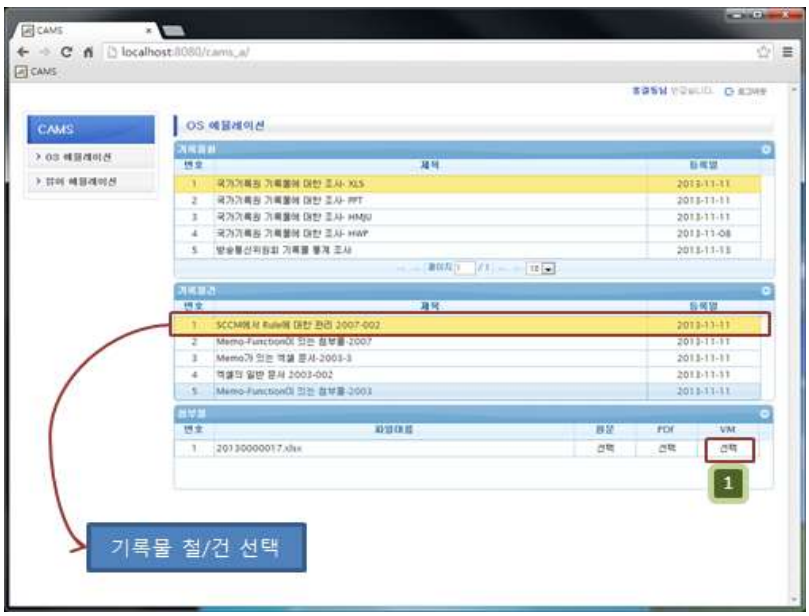
○ 클라이언트에서 브라우저를 통해 웹 서비스 접근




<그림 3-155> 웹 서비스 초기 화면

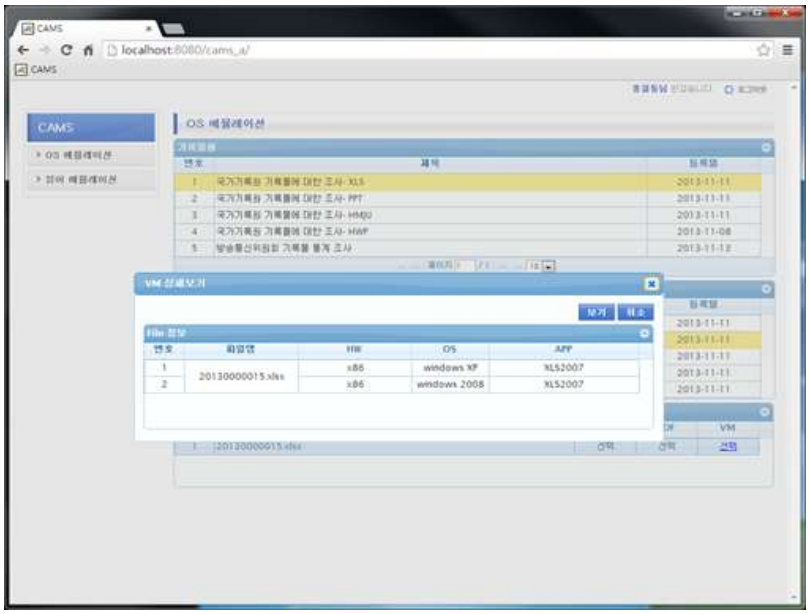
❒ 운영체제 기반의 에뮬레이션 서비스

○ 운영체제 에뮬레이션 서비스 화면에서 기록물 철/건을 선택하면 포함된 디지털 컴포넌트 목록이 나타나는데, 다음 그림과 같이 뷰어를 선택하면 에뮬레이터 기반에서 디지털 컴포넌트가 조회됨



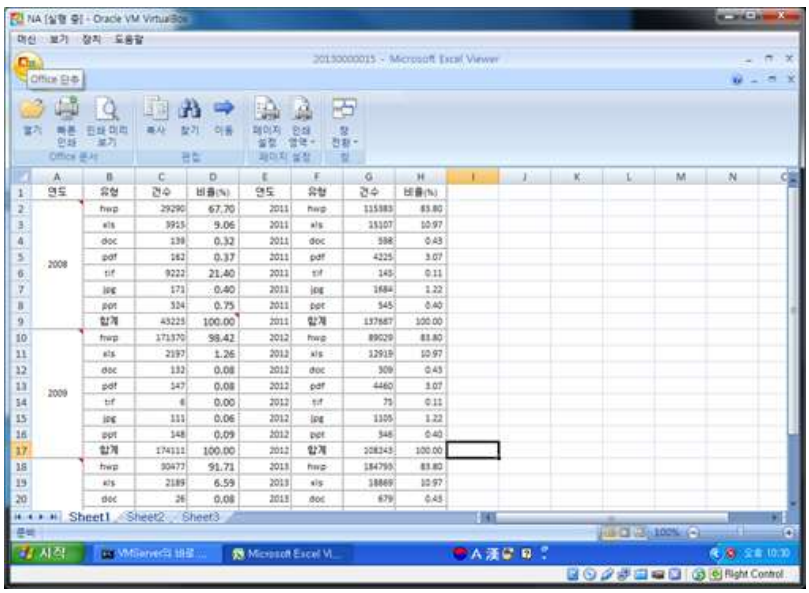

<그림 3-156> 운영체제 에뮬레이션 서비스 화면

○ 선택된 디지털 컴포넌트를 에뮬레이션하기 위한 환경 (하드웨어, 운영체제, 애플리케이션)에 대한 Configuration을 수행하여 가장 적합한 에뮬레이터 환경을 추천함



<그림 3-157> 디지털 컴포넌트 Configuration

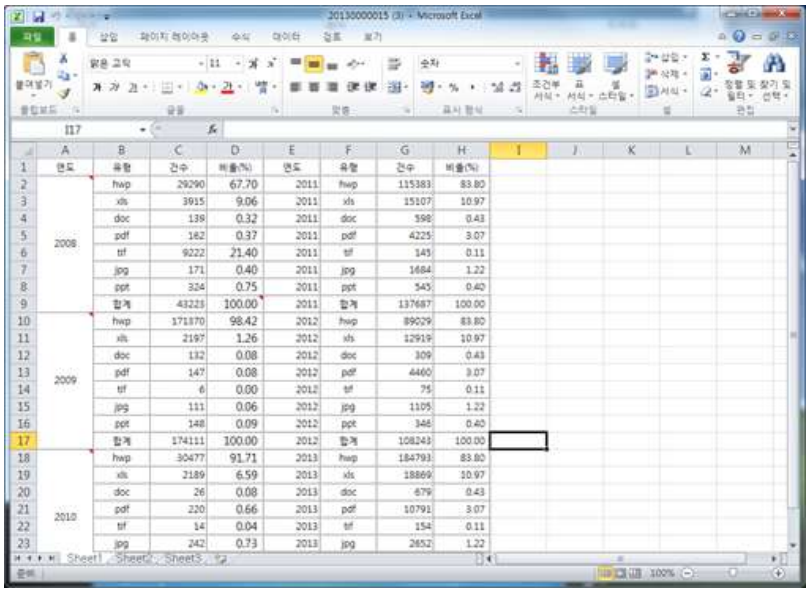
○ 에뮬레이터 기반의 애플리케이션을 통해 조회한 화면



<그림 3-158> 에뮬레이터 기반에서 조회한 화면

○ 원문을 선택하면 해당 애플리케이션으로 문서를 조회하며 PDF는 문서보존포맷(PDF/A)로 변환된 문서를 PDF 뷰어로 조회함

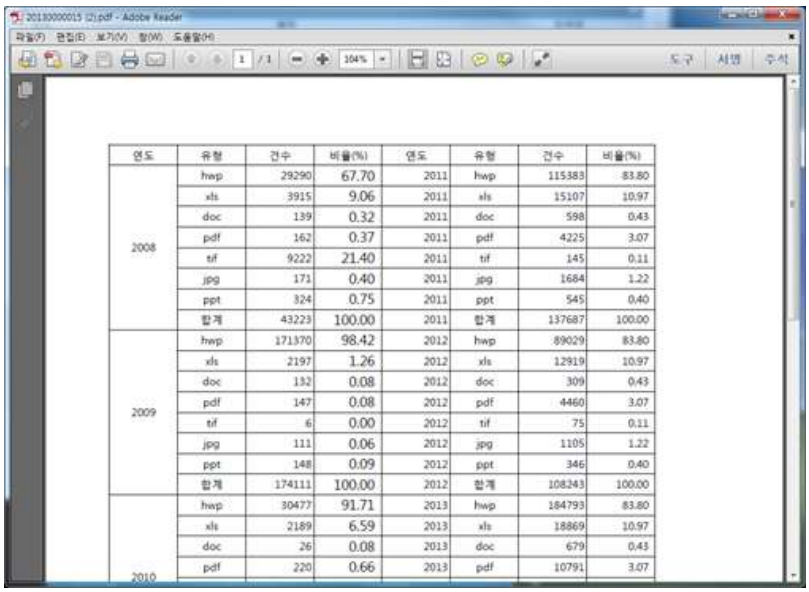
○ 엑셀 포맷 원문을 엑셀 애플리케이션으로 조회한 화면



| 연도 | 유형 | 건수 | 비율(%) | 연도 | 유형 | 건수 | 비율(%) |
|---|---|---|---|---|---|---|---|
| 2008 | hwp | 29290 | 67.70 | 2011 | hwp | 115383 | 83.80 |
| | xls | 3915 | 9.06 | 2011 | xls | 15107 | 10.97 |
| | doc | 139 | 0.32 | 2011 | doc | 598 | 0.43 |
| | pdf | 162 | 0.37 | 2011 | pdf | 4225 | 3.07 |
| | tif | 9222 | 21.40 | 2011 | tif | 145 | 0.11 |
| | jpg | 171 | 0.40 | 2011 | jpg | 1684 | 1.22 |
| | ppt | 324 | 0.75 | 2011 | ppt | 545 | 0.40 |
| | 합계 | 43223 | 100.00 | 2011 | 합계 | 137687 | 100.00 |
| 2009 | hwp | 171370 | 98.42 | 2012 | hwp | 89029 | 83.80 |
| | xls | 2197 | 1.26 | 2012 | xls | 12919 | 10.97 |
| | doc | 132 | 0.08 | 2012 | doc | 309 | 0.43 |
| | pdf | 147 | 0.08 | 2012 | pdf | 4460 | 3.07 |
| | tif | 6 | 0.00 | 2012 | tif | 75 | 0.11 |
| | jpg | 111 | 0.06 | 2012 | jpg | 1105 | 1.22 |
| | ppt | 148 | 0.09 | 2012 | ppt | 346 | 0.40 |
| | 합계 | 174111 | 100.00 | 2012 | 합계 | 108243 | 100.00 |
| 2010 | hwp | 30477 | 91.71 | 2013 | hwp | 184793 | 83.80 |
| | xls | 2189 | 6.59 | 2013 | xls | 18869 | 10.97 |
| | doc | 26 | 0.08 | 2013 | doc | 679 | 0.43 |
| | pdf | 220 | 0.66 | 2013 | pdf | 10791 | 3.07 |
| | tif | 14 | 0.04 | 2013 | tif | 154 | 0.11 |
| | jpg | 242 | 0.73 | 2013 | jpg | 2652 | 1.22 |

<그림 3-159> 엑셀로 조회한 화면

○ 엑셀 포맷 원문을 문서보존포맷(PDF/A)로 변환하여 PDF 뷰어로 조회한 화면

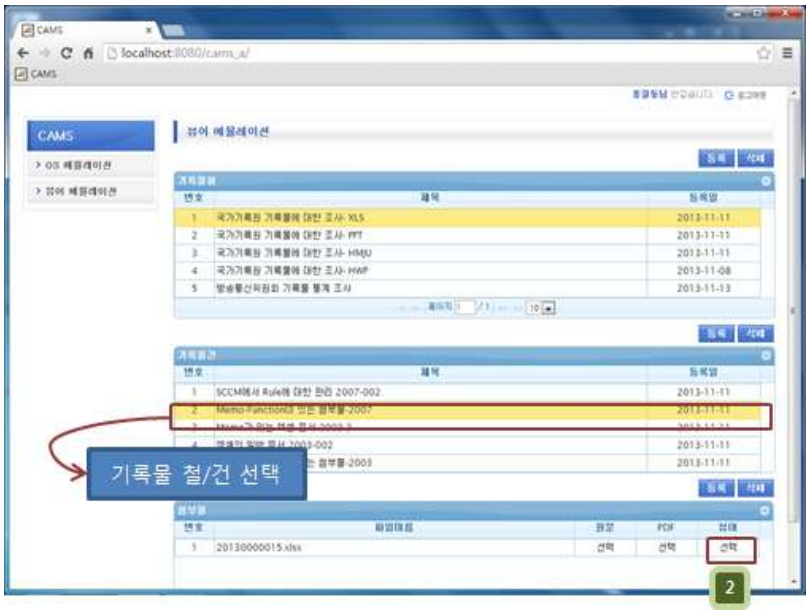


| 연도 | 유형 | 건수 | 비율(%) | 연도 | 유형 | 건수 | 비율(%) |
|---|---|---|---|---|---|---|---|
| 2008 | hwp | 29290 | 67.70 | 2011 | hwp | 115383 | 83.80 |
| | xls | 3915 | 9.06 | 2011 | xls | 15107 | 10.97 |
| | doc | 139 | 0.32 | 2011 | doc | 598 | 0.43 |
| | pdf | 162 | 0.37 | 2011 | pdf | 4225 | 3.07 |
| | tif | 9222 | 21.40 | 2011 | tif | 145 | 0.11 |
| | jpg | 171 | 0.40 | 2011 | jpg | 1684 | 1.22 |
| | ppt | 324 | 0.75 | 2011 | ppt | 545 | 0.40 |
| | 합계 | 43223 | 100.00 | 2011 | 합계 | 137687 | 100.00 |
| 2009 | hwp | 171370 | 98.42 | 2012 | hwp | 89029 | 83.80 |
| | xls | 2197 | 1.26 | 2012 | xls | 12919 | 10.97 |
| | doc | 132 | 0.08 | 2012 | doc | 309 | 0.43 |
| | pdf | 147 | 0.08 | 2012 | pdf | 4460 | 3.07 |
| | tif | 6 | 0.00 | 2012 | tif | 75 | 0.11 |
| | jpg | 111 | 0.06 | 2012 | jpg | 1105 | 1.22 |
| | ppt | 148 | 0.09 | 2012 | ppt | 346 | 0.40 |
| | 합계 | 174111 | 100.00 | 2012 | 합계 | 108243 | 100.00 |
| 2010 | hwp | 30477 | 91.71 | 2013 | hwp | 184793 | 83.80 |
| | xls | 2189 | 6.59 | 2013 | xls | 18869 | 10.97 |
| | doc | 26 | 0.08 | 2013 | doc | 679 | 0.43 |
| | pdf | 220 | 0.66 | 2013 | pdf | 10791 | 3.07 |

<그림 160> PDF 뷰어로 조회한 화면

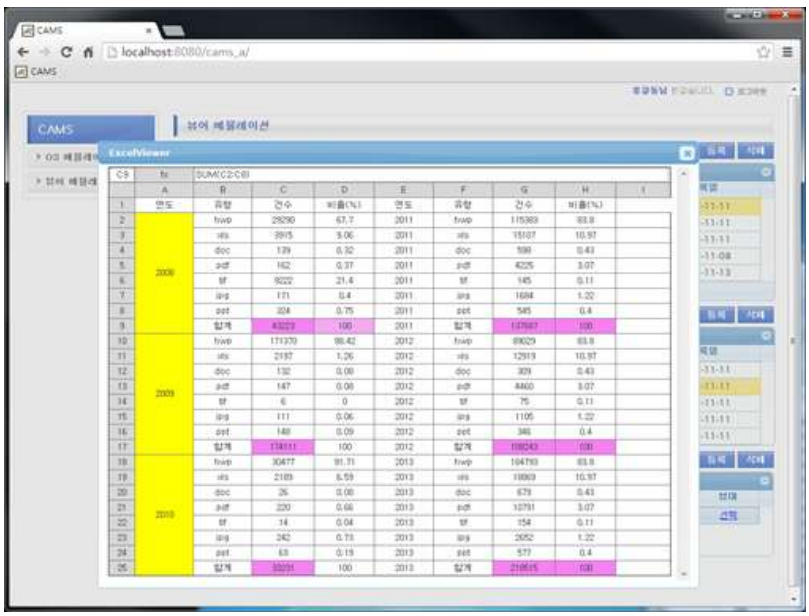
❒ 애플리케이션 뷰어 기반의 에뮬레이션 서비스

○ 애플리케이션 에뮬레이션 서비스 화면에서 기록물 철/건을 선택하면 포함된 디지털 컴포넌트 목록이 나타나는데, 다음 그림과 같이 뷰어를 선택하면 애플리케이션 뷰어에서 디지털 컴포넌트가 조회됨




<그림 3-161> 애플리케이션 뷰어 서비스 화면

○ 애플리케이션 뷰어를 통해 조회한 화면




<그림 3-162> 애플리케이션 뷰어 화면

# 제4장 세부연구개발과제의 연구결과 고찰 및 결론

❒ 전자기록물 장기보존 전략으로서 에뮬레이션이 국내 기록관리 영역에 본격적으로 인식되기 시작한 것은 OAIS 참조모델을 통해 마이그레이션과 더불어 대표적인 보존 행동으로서 소개된 것이었음. 그러나 표준에서는 개념적인 설명에 머물고 있을 뿐 실행 방법으로서 에뮬레이션을 어떻게 실제 기록관리 환경에서 사용할 수 있을지에 관한 내용은 찾아보기 어려웠음. 그리고 2012년 OAIS 개정판이 나왔고 에뮬레이션 전략에 관해서는 여전히 답보상태로 보임

❒ 에뮬레이션 전략은 기술의 복잡성이 높고 구현 비용의 추산이 어려워 효과적인 에뮬레이션이 현실적으로 적용가능한지 판단이 어려움. 해외의 경우에도 장기간, 대규모 예산을 들인 공동 연구를 통해서 계속적인 검토와 연구를 진행하고 있지만 현실적으로 각국에서 이러한 에뮬레이션 전략을 성공적으로 도입한 사례는 아직 파악되고 있지 않음

❒ 그러나 다양한 기능성을 보유한 디지털 컴포넌트들이 계속적으로 생산되고 있고 원래의 기록 그대로를 보려고 하는 인간의 욕구 또한 변함없을 것이며 에뮬레이션 전략이 아니고서는 재현이 불가능한 디지털 컴포넌트들의 기능성들은 여전히 존재할 것임. 즉, 장기보존에 있어서 에뮬레이션이 필요한 영역이 분명하고 각국의 환경에 근거한 에뮬레이션 전략의 필요성 자체는 변함없음

❒ 본 연구는 국내 행정기관의 환경에서 에뮬레이션 전략의 적용이 필요한 사례를 찾고 한국형 에뮬레이터 개발의 필요성에서 시작되었음. 이에, 해외의 에뮬레이션 전략에 관한 주요 동향을 고찰하고 국내 행정기관에서 에뮬레이션 대상이 될 수 있는 구체적인 디지털 컴포넌트를 파악해냈으며 국내 행정기관에서 생산되고 있는 대표적인 디지털 컴포넌트들이 보유하는 기능성 분석과 아울러 에뮬레이션을 위해서 특성화(Characterization)가 필요함을 확인하였음. 이러한 연구결과를 토대로 한국형 에뮬레이션 전략을 제안하고 그 중에서 엑셀에 대한 애플리케이션 에뮬레이터 프로토타입을 설계·구현하였음

❒ 한국형 에뮬레이션 전략을 제안함에 있어서 본 연구에서는 “운영체제 에뮬레이션 방안”과 “애플리케이션 에뮬레이션 방안”이라는 두 가지의 접근법을 취했음. 아울러 최대한 마이그레이션 전략을 구사하되 에뮬레이션에 특화된 영역을 국내 환경에 근거해서 구체화하여 에뮬레이션의 대상과 범위를 한정하고 이러한 제한된 생산 환경과 포맷으로 에뮬레이션의 복잡성과 비용을 낮출 수 있다고 보았음. 이러한 에뮬레이션 전략을 바탕으로 이후로 실제적인 구현과 적용 시도를 통해서

법적 문제, 비용 구조 등도 우리 상황에 맞게 만들어갈 수 있을 것이라 보았음

❒ 한편, 에뮬레이션 전략을 구체화하고 실현하기 위해서는 몇 가지의 과제가 존재할 것으로 보고 있음

○ 첫 번째로, 보다 포괄적인 장기보존계획 속에서 에뮬레이션 전략을 하이브리드하게 적용해나가기 위한 계획이 필요함. 이와 관련 전자기록물의 기능성 재현이라는 맥락에서 에뮬레이션 할 수 밖에 없는 영역이 명백히 존재한다고 보았을 때 보유하고 있는 에뮬레이션 대상 및 구체적인 특성에 근거한 전략이 필요하며 기본적으로 생산된 시점과 가깝게, 해당 기술 환경 및 도구가 사라지기 전에 고안되어야 할 것임

○ 두 번째로, 특성화(Characterization)를 위한 준비가 필요함. 이것은 디지털 컴포넌트 객체 속성 정보를 기반으로 디지털 컴포넌트의 비트스트림에서 랜더링 객체의 직접 추출 되는 방식으로 구현되어야 할 것임. 본 연구를 통해서 국가기록원 기술정보은행(DFR)에 랜더링 객체에 대한 정보를 보완할 수 있도록 XML Schema 개선안이 제시되었으나 보다 본격적으로 자동화된 특성화 툴에 관한 연구가 진행될 필요가 있음

○ 세 번째로, 에뮬레이션을 위해 필요한 기술정보 관리를 위한 준비가 필요함. 본 연구에서는 인쇄 렌더링으로 소실되는 렌더링 객체를 기술정보은행(DFR)의 레지스트리에 저장할 수 있도록 개선된 XML Schema 구조를 제안하였음. 이러한 레지스트리는 장차 디지털 보존 아키텍처에서 일반적으로 핵심적인 요소로서 인정받고 있는 재현정보 레지스트리(Representation Information Registries)로 확장 발전시켜야 할 것임

- 재현정보 레지스트리는 디지털 보존 활동에 있어서 디지털 객체에 접근하는 것을 지원하기 위한 기술적 환경의 필요사항이 지식화 하여 저장된 것임. 이러한 개념을 처음으로 포괄적으로 다룬 것이 OAIS의 RI(Representation Information)임
- RI의 범위와 양은 잠재적으로 광범위함. OAIS는 구조적(structure)RI, 의미적(sementic)RI, 기타(other)RI로 구분해놓고 있음
- RI는 참고정보(reference), 특성화(characterization), 보존계획(preservation planning), 보존행동(preservation action), 입수(ingest), 보급(dissemination) 등의 디지털 보존 활동에서 활용되고 있음
- 국내의 경우 본격적인 RI가 구축되어 있지는 않지만 현재의 DFR이 구조적 RI에 해당하는 파일 포맷 스펙 등 구조적인 정보들을 포함하고 있으며 장차 디지털 보존을 위한 RI가 구축된다면 DFR은 RI의 핵심적인 구성요소가 될 수 있을 것임
- RI의 구축은 에뮬레이션 활동 뿐 아니라 장기보존 활동을 위한 지식 베이스

를 마련하는 차원에서 필요함

○ 마지막으로, 생산 포맷의 표준화가 필요함. 생산 포맷에 표준 포맷이 적용되고 장기적으로 이 포맷에 대한 뷰어가 계속적으로 만들어진다면 에뮬레이션 서비스의 제공이 용이해 질 수 있음

# 제5장 세부연구개발과제의 연구성과

## 5.1 활용성과

| 세부과제명 | 행정기관 전자기록물 재현기술 연구 및 프로토타입 개발 |
|---|---|
| 세부과제책임자 | 임진희 / 명지대학교 기록정보과학 전문대학원 / 컴퓨터공학 |

가. 연구논문

| 번호 | 논문제목 | 저자명 | 저널명 | 집(권) | 페이지 | Impact factor | 국내/국외 | SCI여부 |
|---|---|---|---|---|---|---|---|---|
| 1 | 전자기록 장기보존 전략으로써의 에뮬레이션 사례분석 | 김명훈, 오명진 외 | 기록학 연구 | 제 38호 | 265~309 | | 국내 | |
| 2 | 디지털 컴포넌트의 렌더링 객체 보존 방안 연구(예정) | 이재영, 최주호 외 | 기록학 연구 | | | | 국내 | |

나. 학술발표

| 번호 | 발표제목 | 발표형태 | 발표자 | 학회명 | 연월일 | 발표지 | 국내/국제 |
|---|---|---|---|---|---|---|---|
| 1 | 전자기록 장기보존 전략으로써의 에뮬레이션 사례분석 | 구두 | 이재홍 | 제5회 전국 기록인대회 | 2013. 11. 2 | 제 5회 전국 기록인 대회 발표 자료집 | 국내 |
| 2 | 에뮬레이션 기반의 전자기록물 재현 방안 연구 | 구두 | 최주호 | 제5회 전국 기록인대회 | 2013. 11. 2 | 제 5회 전국 기록인 대회 발표 자료집 | 국내 |

다. 지적재산권

| 번호 | 출원/등록 | 특허명 | 출원(등록)인 | 출원(등록)국 | 출원(등록)번호 | IPC분류 |
|---|---|---|---|---|---|---|
| 1 | 출원(예정) | 국가기록원 디지털 컴포넌트의 에뮬레이션 모델 | 국가기록원 | 한국 | | |

라. 정책활용

| |
|---|
| 전자기록물의 장기보존 전략 수립시 디지털 컴포넌트 레벨에서 에뮬레이션 대상에 대한 평가가 이루어 질 수 있도록 정책 수립의 기반 마련 |

마. 타연구/차기연구에 활용

| |
|---|
| 국가기록원의 “장기보존 전자기록물의 에뮬레이션 기술 연구”의 2차년도 과제로 2014년 수행될 예정인 “전자기록물 재현을 위한 통합 에뮬레이터 연구 개발” 과제의 디지털 컴포넌트 유형별 렌더링 도구 및 에뮬레이터 개발과 DFR(Digital Format Registry) 관리도구 개발의 연구 기반으로 활용 |

바. 언론홍보 및 대국민교육

| |
|---|
| 해당사항 없음 |

사. 기타

| |
|---|
| 해당사항 없음 |

## 5.2 활용계획

- ❒ 디지털 컴포넌트 유형별 렌더링 방법 마련 및 에뮬레이션 전략의 수행
- ❒ 한국형 에뮬레이터를 이용한 본 디지털(born digital) 기록물을 원본 기반의 장기보존 및 활용성 고취

# 제6장 기타 중요변경사항

(없음)

# 제7장 참고문헌


국가기록원, 『전자기록 관리와 보존을 위한 국제협력 아젠다 개발』, 국가기록원, 2007

국가기록원, 『전자기록물 장기보존 전략 연구 및 테스트베드 구축』, 국가기록원, 2008

국가기록원, 『웹기록물 아카이빙 기반기술 연구 및 적용시험』, 2009

오삼균외, 「전자기록물 장기보존을 위한 마이그레이션 전략에 관한 연구」, 『한국기록관리학회지』9(2), 한국기록관리학회, 2009

이규철 외, 「공공기관 심층 웹 기록물 아카이빙을 위한 머타데이터 설계」, 한국전자거래학회지 제 14권 제4호, 2009

조은희, 임진희, 「행정정보 데이터세트 기록의 기준 및 절차연구」, 『기록학연구』 제 19호, 2009

송주형, 「기록관리 대상으로서 SNS연구」, 한국외국어대학교 석사학위 논문, 2012

한글과컴퓨터, Hwp Document File Formats revision 1.0:20100630, 2010

Digital Preservation Testbed, Emulation : Context and Current Status, Digital Preservation Testbed White Paper, 2003

National Archives of Australia, Digital Recordkeeping : Guidelines for Creating, Managing and Preserving Digital Records[Exposure Draft], 2004

Stewart Granger, “Emulation as a Digital Preservation Strategy", D-Lin Maganize 6(10), 2000.

Paul Wheatley, "Migration: A CAMiLEON Discussion Paper", Ariadne 29, 2001

Jeffrey van der Hoeven, Hilde van Wijngaarden, Remoco Verdegem, Jacqueline

Slats, Emulation: A Viable Preservation Strategy, The National Library of the Netherlands & Nationaal Archief of the Netherlands, 2005

Jeff Rothenberg, "An Experiment in Using Emulation to Preserve Digital Publications", Koninklijke Bibliotheek, The Hague, 2000

Andrew Jackson, Cost of emulation, emulationSercives FAQ, 2010

PP7-D1. Report on Comparison of Planets with OAIS. 2007. 6.; PP7-D3-4. Report on The Planets Functional Model. 2009

Adam Farquhar & Helen Hockx-Yu, Planets: Integrated Services for Digital Preservation, The International Journal of Digital Curation, Issue 2, Volume 2, 2007

Erik Oltmans, Nanda Kol, “A Comparison between Migration and Emulation in Terms of Costs”, RLG DigiNews 9(2), 2005

Digital Preservation Testbed, From Digital Volatility to Digital Permanence: Preserving Databases(version 1.0), Dutch National Archives, The Hague, December 2003

Planets, An Emerging Market, Planets White Paper, 2010

Microsoft Corporation, [MS-PPT] – v20130726 PowerPoint (.ppt) Binary File Format Copyright © 2013 Microsoft Corporation Release:, 2013

Microsoft Corporation, [MS-XLS] – v20130726 Excel Binary File Format (.xls) Structure Copyright © 2013 Microsoft Corporation. Release:, 2013

Bram Lohman, Edo Noordermeer, Architectural Design Document Emulation Framework Release 2.0.0, 2012

Bram Lohman, David Michel, Edo Noordermeer, System Maintenance Guide Emulation Framework Release 2.0.0, 2012

Bram Lohman, Jeffrey van der Hoeven, Edo Noordermeer, System User Guide

Emulation Framework Release 2.0.0, 2012

Bart Kiers, Jeffrey van der Hoeven, Test description and results document for Emulation Framework, 2012

# 제8장 첨부서류

## 8.1 컴퓨터 프로그램 관련 저작권 법률자문 및 상담

### 가. 컴퓨터 프로그램 관련 저작권 법률 자문 및 상담 요청사항

### 1. 저작권법 상 컴퓨터 프로그램관련 법률

❒ 저작권 법상 컴퓨터 프로그램의 정의

○ 저작권법 제1장(총칙), 제2조(정의), 16항에 따르면 "컴퓨터프로그램저작물"은 특정한 결과를 얻기 위하여 컴퓨터 등 정보처리능력을 가진 장치(이하 "컴퓨터"라 한다) 내에서 직접 또는 간접으로 사용되는 일련의 지시·명령으로 표현된 창작물을 말한다. 라고 정의 하고 있다.

○ 제2장 (저작권)의 제1절 (저작물)의 제 4조(저작물의 예시 등)에서 본 저작권법이 말하는 저작물의 예시로 9항의 "컴퓨터 프로그램 저작물" 이 저작물의 범주에 포함됨을 알 수 있다.

❒ 저작권 법상 저작재산권의 제한 사항

○ 제 2관 저작재산권의 제한에서는 재판절차 등에서의 복제(제23조), 정치적 연설 등의 이용(24조), 학교교육 목적 등의 이용(제25조), 시사보도를 위한 이용(제26조), 시사적인 기사 및 논설의 복제(제27조), 공표된 저작물의 인용(제28조), 영리를 목적으로 하지 아니하는 공연·방송(제29조), 사적이용을 위한 복제(제30조), 도서관 등에서의 복제등(제31조), 시험문제로서의 복제(제32조), 시각장애인 등을 위한 복제 등(제33조), 방송사업자의 일시적 녹음 · 녹화(제34조), 미술저작품등의 전시 또는 복제(제35조), 번역 등에 의한 이용(제36조), 출처의 명시(제36조), 출처의 명시(제37조)으로 법률상 명시하고 있으나 37조의 2의 경우 프로그램에 관해서는 제 23조(재판절차 등에서의 복제), 제25조(학교교육 목적 등의 이용), 제30조(사적이용을 위한 복제), 및 제32조(시험문제로서의 복제)는 적용치 아니한다고 명시하고 있다.

❒ 저작권 법상 컴퓨터 프로그램의 저작재산권의 존속기한

○ 제 3관 저작재산권의 보호기간의 제 42조 영상저작물 및 프로그램의 보호기간의 경우 "영상 저작물 및 프로그램의 저작재산권은 제 39조 및 40조에도 불구하고 공표한 때부터 50년간 존속한다. 다만 창작한 때부터 50년 이내에 공표되지 아니한 경우에는 창작한 때무터 50년간 존속한다."라고 명시하고 있다.

❒ 도서관 등에서의 복제

○ 제31조(도서관등에서의 복제 등) ①「도서관법」에 따른 도서관과 도서·문서·기록 그 밖의 자료(이하 "도서등"이라 한다)를 공중의 이용에 제공하는 시설 중 대통령령이 정하는 시설(당해시설의 장을 포함한다. 이하 "도서관등"이라 한다)은 다음 각 호의 어느 하나에 해당하는 경우에는 그 도서관등에 보관된 도서등(제1호의 경우에는 제3항의 규정에 따라 당해 도서관등이 복제·전송받은 도서등을 포함한다)을 사용하여 저작물을 복제할 수 있다. 다만, 제1호 및 제3호의 경우에는 디지털 형태로 복제할 수 없다.

1. 조사·연구를 목적으로 하는 이용자의 요구에 따라 공표된 도서등의 일부분의 복제물을 1인 1부에 한하여 제공하는 경우

2. 도서등의 자체보존을 위하여 필요한 경우

3. 다른 도서관등의 요구에 따라 절판 그 밖에 이에 준하는 사유로 구하기 어려운 도서등의 복제물을 보존용으로 제공하는 경우

②도서관등은 컴퓨터를 이용하여 이용자가 그 도서관등의 안에서 열람할 수 있도록 보관된 도서등을 복제하거나 전송할 수 있다. 이 경우 동시에 열람할 수 있는 이용자의 수는 그 도서관등에서 보관하고 있거나 저작권 그 밖에 이 법에 따라 보호되는 권리를 가진 자로부터 이용허락을 받은 그 도서등의 부수를 초과할 수 없다. <개정 2009.4.22>

③도서관등은 컴퓨터를 이용하여 이용자가 다른 도서관등의 안에서 열람할 수 있도록 보관된 도서등을 복제하거나 전송할 수 있다. 다만, 그 전부 또는 일부가 판매용으로 발행된 도서등은 그 발행일로부터 5년이 경과하지 아니한 경우에는 그러하지 아니하다. <개정 2009.4.22>

④도서관등은 제1항제2호의 규정에 따른 도서등의 복제 및 제2항과 제3항의 규정에 따른 도서등의 복제의 경우에 그 도서등이 디지털 형태로 판매되고 있는 때에는 그 도서등을 디지털 형태로 복제할 수 없다.

⑤도서관등은 제1항제1호의 규정에 따라 디지털 형태의 도서등을 복제하는 경우 및 제3항의 규정에 따라 도서등을 다른 도서관등의 안에서 열람할 수 있도록 복제하거나 전송하는 경우에는 문화체육관광부장관이 정하여 고시하는 기준에 의한 보상금을 당해저작재산권자에게 지급하여야 한다. 다만, 국가, 지방자치단체 또는 「고등교육법」 제2조의 규정에 따른 학교를 저작재산권자로 하는 도서등(그 전부 또는 일부가 판매용으로 발행된 도서등을 제외한다)의 경우에는 그러하지 아니하다. <개정 2008.2.29>

⑥제25조제5항 내지 제9항의 규정은 제5항의 보상금의 지급 등에 관하여 준용한다.

⑦제1항 내지 제3항의 규정에 따라 도서등을 디지털 형태로 복제하거나 전송하는 경우에 도서관등은 저작권 그 밖에 이 법에 따라 보호되는 권리의 침해를 방지하기 위하여 복제방지조치 등 대통령령이 정하는 필요한 조치를 하여야 한다.

⑧ 「도서관법」 제20조의2에 따라 국립중앙도서관이 온라인 자료의 보존을 위하여 수집하는 경우에는 해당 자료를 복제할 수 있다. <신설 2009.3.25>

❒ 컴퓨터 프로그램의 저작재산권의 제한 사항

○ 제 5장의 2 프로그램에 관한 특례의 제 101조의 3 프로그램의 저작재산권의 제한의 조항의 경우에는 제 1항 “다음 각 호의 어느 하나에 해당하는 경우에는 그 목적상 필요한 범위에서 공표된 프로그램을 복제 또는 배포할 수 있다. 다만, 프로그램의 종류·용도, 프로그램에서 복제된 부분이 차지하는 비중 및 복제의 부수 등에 비추어 프로그램의 저작재산권자의 이익을 부당하게 해치는 경우에는 그러하지 아니하다.”

1. 재판 또는 수사를 위하여 복제하는 경우

2. 「유아교육법」, 「초·중등교육법」, 「고등교육법」에 따른 학교 및 다른 법률에 따라 설립된 교육기관(상급학교 입학을 위한 학력이 인정되거나 학위를 수여하는 교육기관에 한한다)에서 교육을 담당하는 자가 수업과정에 제공할 목적으로 복제 또는 배포하는 경우

3. 「초·중등교육법」에 따른 학교 및 이에 준하는 학교의 교육목적을 위한 교과용 도서에 게재하기 위하여 복제하는 경우

4. 가정과 같은 한정된 장소에서 개인적인 목적(영리를 목적으로 하는 경우를 제외한다)으로 복제하는 경우

5. 「초·중등교육법」, 「고등교육법」에 따른 학교 및 이에 준하는 학교의 입학시험이나 그 밖의 학식 및 기능에 관한 시험 또는 검정을 목적(영리를 목적으로 하는 경우를 제외한다)으로 복제 또는 배포하는 경우

6. 프로그램의 기초를 이루는 아이디어 및 원리를 확인하기 위하여 프로그램의 기능을 조사·연구·시험할 목적으로 복제하는 경우(정당한 권한에 의하여 프로그램을 이용하는 자가 해당 프로그램을 이용 중인 때에 한한다)

❒ 저작재산권의 기증

○ 제135조(저작재산권 등의 기증) ①저작재산권자등은 자신의 권리를 문화체육관광부장관에게 기증할 수 있다.

②문화체육관광부장관은 저작재산권자등으로부터 기증된 저작물등의 권리를 공정하게 관리할 수 있는 단체를 지정할 수 있다.

③제2항의 규정에 따라 지정된 단체는 영리를 목적으로 또는 당해저작재산권자

등의 의사에 반하여 저작물등을 이용할 수 없다.

④제1항과 제2항의 규정에 따른 기증 절차와 단체의 지정 등에 관하여 필요한 사항은 대통령령으로 정한다.

## 2. 전자기록물 재현을 위한 컴퓨터 프로그램 저작권 질문

❒ 저작권 법 제, 개정과 관련한 질문

○ 제 31조의 1항의 경우 도서 · 문서· 기록 그 밖의 자료를 공중의 이용에 제공하는 시설로 도서관을 명시하고 있다. 여기에 기록관리 법에서 명시하고 있는 기록물 관리기관을 포함 시켜 기록물관리법이나 저작권법에 기록관의 저작재산권의 제한 조항을 추가로 재정하는 방법은 어떠한가?

○ 현재 정보통신기술의 발전으로 인해 전자적으로 도서, 문서, 기록을 생산하는 추세인데 앞서 31조의 1항에서 명시하고 있는 도서 등에 도서, 문서, 기록 등에 전자적으로 생산되는 도서, 문서, 기록의 경우 이를 디스플레이 해 줄 수 있는 프로그램역시 포함해야 하는 조항이 추가되어야 하며, 공중의 목적으로 이용하는 기관에서는 전자적 기록을 활용할 수 있게 하는 프로그램에 대한 저작권 제한의 조항을 추가적으로 제정하거나 기존의 조항들을 개정할 수 있는 가능성이 있는가?

○ 도서 문서 기록을 재현할 수 있는 구형 프로그램에 대한 저작권을 문화체육 관광부에 기증을 하여 저작권법상의 컴퓨터 프로그램의 저작재산권의 존속기한인 50년이 지나지 않은 구형 컴퓨터 프로그램에 한하여 개발사로부터 문화체육관광부로 비영리의 목적으로 저작재산권을 기증 할 수 있는 방법으로 디지털 납본등과 같은 제도를 도입할 수 있는 방법에 대한 제안은 어떠한가?

❒ 프로젝트 팀 시나리오 질문

- ①공공기관이 과거 구입해서 가지고 있는 옛날버전의 소프트웨어를 내부적으로 장기간 보유하는 것(소프트웨어의 아카이빙)은 저작권법에 저촉되는가?

- ②국가기록원에서 구 버전의 도스용 한글을 다수 가지고 있는데 기관 내 활용 및 대국민 서비스하기 위해서는 구형의 소프트웨어의 설치가 필요하다. 현재 (주)한글과 컴퓨터 사와의 해당 소프트웨어 라이선스 계약이 끝난 상황에서 해당 소프트웨어를 재설치 하여 사용 시 저작권법에 저촉되는 것인가?

- ③현재 한글과 컴퓨터의 공공기관용 라이선스 프로그램의 경우 라이선스 계약이 아닌 패키지 구입시 구매시점부터 영구적으로 사용되는 권한을 갖게 된다. 현재 이와같은 라이선스 정책을 펼치고 있는데, 이 경우 현재까지 생산된 모든 소프트웨어를 포함하는 것인가?

- ④현재 (주)한글과 컴퓨터 사의 최신 소프트웨어를 라이선스를 계약하여 사용하고 있는데 기존에 구 버전의 경우 사용권이 없는 것인가?
- ⑤내부 즉, 기관 내에서 사용할 뿐만 아니라 외부 사용 즉 대민 서비스를 할 경우 해당 소프트웨어의 반출은 저작권법에 저촉되는 것인가?

- ⑥국가기록원의 경우 공공기관으로써 대민서비스의 경우 공공서비스로 분류되는데 이러한 경우 공적인 사용으로도 저작권법에 저촉이 되는가?

- 공공기관에서 위와같은 시나리오를 수행 시 문제가 있을 경우 어떠한 저작권과 관련한 법적인 문제가 있는 것인가?

- 현행법으로 보았을 때 불법일 경우 현행법을 개정하기 위해서는 어떠한 법을 개정해야 하는 것인가?

## 나. 한국 소프트웨어 저작권 협회 소속 변호사 법률 자문 결과

******

# 의 견 서

## Ⅰ. 사안의 개요 및 질의의 요지

귀 연구소는 전자기록물을 장기간 보관하기 위한 프로젝트 수행 중, 이를 위한 방안으로 전자기록물 재현을 위한 저작권법 등 관련법의 개정 문제(개정가능성 및 방법) 및 재현 과정에서 발생할 수 있는 저작권법 저촉 우려 문제에 대해 질의하셨습니다.

구체적인 질문 내용은 다음과 같습니다.

### □ 저작권법 제, 개정과 관련한 질문

1. 저작권법 제31조의 제1항의 경우 도서·문서·기록 그 밖의 자료를 공중의 이용에 제공하는 시설로 도서관을 명시하고 있다. 여기에 기록관리법에서 명시하고 있는 기록물 관리기관을 포함시켜 기록관리법이나 저작권법에 기록관의 저작재산권의 제한 조항을 추가로 제정하는 방법은 어떠한가?
2. 현재 정보통신기술의 발전으로 인해 전자적으로 도서, 문서, 기록을 생산하는 추세인데 앞서 저작권법 제31조 제1항에서 명시하고 있는 도서 등에 도서, 문서, 기록 등에 전자적으로 생산되는 도서, 문서, 기록의 경우 이를 디스플레이 해 줄 수 있는 프로그램 역시 포함해야 하는 조항이 추가되어야 하며, 공중의 목적으로 이용하는 기관에서는 전자적 기록을 활용할 수 있게 하는 프로그램에 대한 저작권 제한의 조항을 추가적으로 제정하거나 기존의 조항들을 개정할 수 있는 가능성이 있는가?
3. 도서, 문서, 기록을 재현할 수 있는 구형 프로그램에 대한 저작권을 문화체육관광부에 기증을 하여 저작권법상의 컴퓨터프로그램의 저작재산권의 존속기한인 50년이 지나지 않은 구형 컴퓨터프로그램에 한하여 개발사로부터 문화체육관광부로 비영리의 목적으로 저작재산권을 기증할 수 있는 방법으로 디지털 납본 등과 같은 제도를 도입할 수 있는 방법에 대한 제안은 어떠한가?

### □ 프로젝트 팀 시나리오 질문

4. ① 공공기관이 과거 구입해서 가지고 있는 옛날버전의 소프트웨어를 내부적으로 장기간 보유하는 것(소프트웨어의 아카이빙)은 저작권법에 저촉되는가?

5. ② 국가기록원에서 구버전의 도스용 훈글을 다수 가지고 있는데 기관 내 활용 및 대국민 서비스 하기 위해서는 구형의 소프트웨어의 설치가 필요하다. 현재 (주)한글과 컴퓨터 사와의 해당 소프트웨어 라이선스 계약이 끝난 상황에서 해당 소프트웨어를 재설치 하여 사용시 저작권법에 저촉되는 것인가?

6. ③ 현재 한글과 컴퓨터의 공공기관용 라이선스 프로그램의 경우 라이선스 계약이 아닌 패키지 구입시 구매시점부터 영구적으로 사용되는 권한을 갖게 된다. 현재 이와같은 라이선스 정책을 펼치고 있는데, 이 경우 현재까지 생산된 모든 소프트웨어를 포함하는 것인가?

7. ④현재 (주)한글과 컴퓨터 사의 최신 소프트웨어를 라이선스를 계약하여 사용하고 있는데 기존에 구 버전의 경우 사용권이 없는 것인가?

8. ⑤내부 즉, 기관내에서 사용할 뿐 만아니라 외부 사용 즉 대민 서비스를 할 경우 해당 소프트웨어의 반출은 저작권 법에 저촉되는 것인가?

9. ⑥국가기록원의 경우 공공기관으로써 대민서비스의 경우 공공서비스로 분류되는데 이러한 경우 공적인 사용으로도 저작권 법에 저촉이 되는가?

10. 공공기관에서 위와같은 시나리오를 수행시 문제가 있을경우 어떠한 저작권과 관련한 법적인 문제가 있는 것인가?

11. 현행법으로 보았을 때 불법일 경우 현행법을 개정하기 위해서는 어떠한 법을 개정해야 하는 것인가?

## Ⅱ. 검토의견 요약

### □ 저작권법 제, 개정과 관련한 질문

1. 저작권법(시행령 포함), 도서관법, 기록물관리법의 규정 내용 및 저작자의 재산권 등을 종합적으로 고려해 볼 때, 기록물관리기관(특히 기록관)을 저작권법 제31조의 “도서관 등”에 포함시켜 저작권법의 도서관 등의 저작재산권 제한 조항에 기록관의 저작재산권 제한 조항을 추가로 제정하는 것은 도서관의 범위를 확대하는 것 뿐만 아니라 저작권자의 권리를 제한하는 범위를 확대하는 것이기도 하므로 쉽지 않을 것으로 판단됩니다. 이는 기타 법률의 제·개정을 통한 추가도 마찬가지일 것입니다. 다만 저작권법 제35조의3의 적용가능성에 대해 2번에서 상술하도록 하겠습니다.

2. 저작권법 제31조 제1항의 해석 및 제정 취지에 비추어 동조의 ‘도서등에’ 이를 디스플레이 해 줄 수 있는 프로그램 역시 포함되어야 한다는 의견은 공감의 여지가 있으나, 저작권법상 “도서등”은 이와 같은 도서등을 활용하기 위한 도구로서 기능하는 프로그램과는 전혀 성격을 달리하는 것으로, 위 “도서등”의 개념에 이를 디스플레이 해 줄 수 있는 프로그램을 포함해야 한다는 견해는 문언의 해석을 넘어서 수용되기 어려울 것으로 사료됩니다.

   다만, 이와 관련하여 특히 공중 즉, 불특정 다수인에게 제공하기 위한 목적으로 이용하는 기관에서 전자적 기록을 활용할 수 있게 하는 프로그램에 대한 저작권 제한의 조항의 필요성을 논의하고자 한다면 **저작권법 제35조의3 포괄적 공정이용 조항**에 대한 검토는 가능할 것 같습니다.

   전자기록물을 공중에게 제공하기 위한 전제로서 동 기록물 재현을 위한 프로그램의 이용이라는 목적을 고려해 볼 때, 관련 판례 및 그 외 규정상의 다른 요소들을 함께 고려한다는 것을 감안하면 저작자의 포괄적 공정이이용 조항의 범위 안에 포섭될 가능성은 제안해 볼 수 있을 것입니다(상세한 것은 후술 검토의견 참조).

3. 관련 제도 및 관련법을 종합하여 살펴보면, 디지털 자료의 경우에는 납본제도 추진 노력에도 불구하고 인쇄 자료와 달리 납본을 의무화하지 않고 일정한 경우 정당한 보상을 조건으로 수집·보존하도록 규정한 개정법의 취지에 비추어, 현행법 하에서는 도서등을 재현할 수 있는 구형 프로그램에 한하여 기증할 수 있는 제도를 도입하는 것이 쉽지는 않을 것으로

사료됩니다.

다만 저작자의 재산권과의 조화를 고려해 볼 때, 현재 판매하지 않고 있어 구입이 불가능한 프로그램의 경우에 한하여 문제되고 있는 당해 기록을 재현할 목적으로만 한정하여 제안하는 것은 설득력이 있어 보입니다.

□ **프로젝트 팀 시나리오 질문**

4. "공공기관"이 구입하여 가지고 있는 소프트웨어라면 구입과 동시에 국유재산에 해당하기 때문에 우선하는 관련법에 의해 처리되어야 할 것입니다. 따라서 기간을 정하여 라이선스 계약을 체결한 경우라고 하더라도, 이는 내부적으로 장기간 보관하는 것이 말 그대로 사용을 전제하지 않은 백업·보관의 목적인지 혹은 사용이나 공개·서비스 등을 목적으로 하는 것인지에 따라 저작권법 적용 여부가 달라질 수 있습니다. 일률적으로 저작권법에 저촉된다고 보기는 어려울 것으로 보입니다(상세한 것은 후술 검토의견 참조).

5. 현재 해당 소프트웨어 라이선스 계약이 끝난 상황에서 재계약 등을 통한 권리자 허락 없이 재설치·사용하는 것은 저작권법 제46조 이용허락 조항을 위반한 것임과 동시에 재설치 과정에서 복제권을 침해한 것이므로 저작권법에 저촉됩니다. (저작권법 제136조 제1항 제1호 참조)

   물론 저작권자가 기존에 라이선스 계약을 체결했던 사용자에게 대국민서비스 등의 공익적 성격을 갖는 특정한 목적에 한정하여 위와 같은 허락 없는 재설치를 허용한다고 하면 당연히 허용된다고 할 것입니다.

6. 패키지 구입시 영구적인 사용권한을 갖게 된다는 것의 의미는 일반적인 경우 구입한 버전에 한정하는 것으로 이해됩니다. 따라서 당시에 구입한 버전에 대해서는 영구적인 사용권을 취득하였다고 볼 수 있을 것이나, 이것을 (주)한글과컴퓨터가 현재까지 생산한 모든 소프트웨어를 포함한다는 의미로는 볼 수 없습니다.

7. 구 버전에 대한 사용권이 있을 수도 있고, 없을 수도 있습니다. 저작권자가 이를 허락하느냐 허락하지 않느냐의 문제이므로, 언제나 기존 구 버전에 대한 사용권이 없는 것은 아닙니

다(상세한 것은 후술 검토의견 참조).

8. 내부 사용은 물론 외부 사용의 경우에도 대여권을 갖는 저작자의 허락 없이 소프트웨어를 반출하면 저작자의 대여권을 침해(저작권법 제136조 제1항 제1호 참조)하게 됩니다. 다만, 이와 관련하여서는 '**공공대출권**' 개정(도입) 가능성은 제안해 볼 수 있다고 사료됩니다(상세한 것은 후술 검토의견 참조).

9. 공적인 사용이라고 해서 저작물의 자유이용이 당연히 허용되는 것은 아닙니다. 국가기록원의 대민서비스의 경우 공표된 저작물의 인용(법 제28조) 또는 저작물이 공정한 이용(제35조의3)과 같은 저작재산권 제한 사유를 검토해 볼 수 있을 것입니다(상세한 것은 후술 검토의견 2번과 9번 참조).

10. 이용허락계약 위반에 따른 민사상 손해배상책임, 저작자의 복제권 등을 침해한 경우라면 형사처벌을 받을 수 있습니다(상세한 것은 후술 검토의견 참조).

11. 궁극적으로 소프트웨어 반출(대여)을 의도하는 개정의 경우라면 후술하는 바와 같이 그 필요성과 목적에 비추어 저작권법에 공공대출권의 도입가능성을 제안해 볼 수 있을 것으로 사료됩니다(상세한 것은 후술 검토의견 8번과 11번 참조).

## Ⅲ. 검토의견

**1. 저작권법 제31조의 제1항의 경우 도서·문서·기록 그 밖의 자료를 공중의 이용에 제공하는 시설로 도서관을 명시하고 있다. 여기에 기록관리법에서 명시하고 있는 기록물 관리기관을 포함시켜 기록관리법이나 저작권법에 기록관의 저작재산권의 제한 조항을 추가로 제정하는 방법은 어떠한가?**

저작권법 제31조 제1항은 도서관법에 따른 도서관과 도서·문서·기록 그 밖의 자료를 공중의 이용에 제공하는 시설 중 대통령령이 정하는 시설은 도서자체의 보존과 조사·연구를 목적으로 하는 경우 등에는 그 도서관 등에 보관된 도서 등을 사용하여 저작물을 복제할 수 있다고 규정하고 있습니다.

그리고 동법 시행령 제12조 제1호와 제2호에서는 법 제31조의 대통령령이 정하는 시설이라 함은 「도서관법」에 따른 국립중앙도서관·공공도서관·대학도서관·학교도서관·전문도서관(영리를 목적으로 하는 법인 또는 단체에서 설립한 전문도서관으로서 그 소속원만을 대상으로 도서관 봉사를 하는 것을 주된 목적으로 하는 도서관은 제외한다)이나 국가, 지방자치단체, 영리를 목적으로 하지 아니하는 법인 또는 단체가 도서·문서·기록과 그 밖의 자료(이하 "도서등"이라 한다)를 보존·대출하거나 그 밖에 공중의 이용에 제공하기 위하여 설치한 시설을 말한다고 규정하고 있습니다.

또한 도서관법 제2조 제1호에서 "도서관"이라 함은 도서관자료를 수집·정리·분석·보존하여 공중에게 제공함으로써 정보이용·조사·연구·학습·교양·평생교육 등에 이바지하는 시설을 말한다고 규정하고 있습니다.

그런데 '공공기록물 관리에 관한 법률'(이하 '기록물관리법') 제3조 제4호에서는 "기록물관리기관"이라 함은 일정한 시설 및 장비와 이를 운영하기 위한 전문인력을 갖추고 기록물관리 업무를 수행하는 기관을 말하며, 영구기록물관리기관, 기록관 및 특수기록관으로 구분한다고 규정하고 있고, 동법 제13조 제1항, 제2항에서는 대통령령으로 정하는 공공기관은 공공기관의 기록물을 효율적으로 관리하기 위하여 기록관을 설치·운영하여야 하고, 대통령령으로 정하는 공

공기관은 동법 시행령 제10조에서 규정한 다음 기관이 해당됩니다.

1. 「정부조직법」 제2조에 따른 중앙행정기관
2. 감사원, 국가인권위원회, 방송통신위원회, 국가정보원, 국무조정실, 국무총리비서실
3. 국민권익위원회, 공정거래위원회, 금융위원회
4. 중앙행정기관의 소속기관 중 지방보훈청, 지방국세청, 서울세관, 인천공항세관, 부산세관, 인천세관, 대구세관, 광주세관, 고등검찰청, 지방검찰청 및 지청, 지방교정청, 지방병무청, 지방경찰청, 지방우정청, 지방식품의약품안전청, 유역환경청, 지방환경청 및 국립환경과학원, 지방고용노동청, 중앙노동위원회, 지방국토관리청, 중앙토지수용위원회, 지방해양항만청, 지방항공청, 지방해양경찰청
5. 시·도
6. 시·군·구(지방자치단체인 구를 말한다) 및 「제주특별자치도 설치 및 국제자유도시 조성을 위한 특별법」 제15조제2항에 따라 제주자치도에 두는 행정시
7. 시·도 교육청 및 「지방교육자치에 관한 법률」 제34조에 따른 지역교육청
8. 국방부장관이 중앙기록물관리기관의 장과 협의하여 정하는 직할 군 기관
9. 육군·해군·공군 참모총장이 중앙기록물관리기관의 장과 협의하여 정하는 군 기관
10. 제3조 각 호의 어느 하나에 해당하는 공공기관 중 관리하여야 하는 기록물의 양이 안전행정부령이 정하는 기준을 초과하는 공공기관
11. 「고등교육법」 제2조에 따른 학교 중 국립·공립대학
12. 그 밖에 영구기록물관리기관의 장이 기록관 설치가 필요하다고 인정되어 지정한 공공기관

저작권법 및 동법 시행령, 도서관법, 기록물관리법의 규정내용을 종합해 보면, 귀 연구소에서 제안하는 바와 같이 기록물관리법에서 명시하고 있는 기록물관리기관(특히 기록관)을 저작권법 제31조의 “도서관 등”에 포함시키는 경우 도서관의 범위가 폭넓게 확대되는 결과를 가져오게 됩니다. 그러나 도서관의 범위를 확대한다는 것은 저작권자의 권리를 제한하는 범위를 확대

한다는 것과 마찬가지의 결과를 가져오게 됩니다. 따라서 이와 같은 내용으로 관련법을 해석하거나 제·개정을 제안하는 것은 신중한 논의가 필요할 뿐 아니라 저작권자의 충분한 양해가 전제되어야 하는 것으로서, 제안이 불가능한 것은 아닐 것이나 반영되기는 쉽지 않을 것으로 생각됩니다. 또한, 저작권을 제한하는 것은 헌법이 보장하는 재산권에 중대한 제한을 초래하는 사항이 될 수 있으므로, 권리자의 권리를 규율한 저작권 법 외 다른 법률에 저작재산권의 제한 규정을 추가하는 것은 법체계에 비추어 볼 때, 수용되기는 어려울 것으로 사료됩니다.

즉, 기록물관리법은 “공공기관”의 투명하고 책임 있는 행정 구현과 공공기록물의 안전한 보존 및 효율적 활용을 위하여 공공기록물 관리에 필요한 사항을 정함을 목적(법 제1조)으로 하고 있어 그 적용 대상이 “공공기관”에 한정됩니다. 또한 동법 제3조 제2호에서 규정하는 “기록물”이란 공공기관이 업무와 관련하여 생산하거나 접수한 문서·도서·대장·카드·도면·시청각물·전자문서 등 모든 형태의 기록정보 자료와 행정박물(行政博物)을 말합니다.

반면 저작권법 제2조 제1호는 저작물을 인간의 사상 또는 감정을 표현한 창작물로 규정하고 있어, 기록물관리법의 규율대상인 기록물과는 그 의미 및 범위에 있어 상당한 차이가 있습니다. 물론 기록물에 일부 저작물이 포함되는 경우가 있을 것이나, 도서관 등의 저작재산권 제한 사유는 가장 강력한 저작재산권 제한 사유 중 하나에 해당되므로, 이와 같은 저작권법 규정에 기록물 전체를 포괄시키는 제한 조항을 추가한다면 저작물의 공공성과 사회성을 감안하더라도 저작자의 재산권에 중대한 침해를 가져 올 가능성도 배제할 수 없습니다.

다만, 궁극적으로 전자적 기록을 활용할 수 있게 하는 프로그램에 대한 저작권 제한 즉 자유이용의 가능성을 살펴보고자 한다면 **저작권법 제35조의3 포괄적 공정이용 조항**에 대한 검토는 가능할 것 같습니다. 이는 다음 2번 문항에서 상술하도록 하겠습니다.

**2. 현재 정보통신기술의 발전으로 인해 전자적으로 도서, 문서, 기록을 생산하는 추세인데 앞서 저작권법 제31조 제1항에서 명시하고 있는 도서 등에 도서, 문서, 기록 등에 전자적으로 생산되는 도서, 문서, 기록의 경우 이를 디스플레이 해 줄 수 있는 프로그램 역시 포함해야 하는 조항이 추가되어야 하며, 공중의 목적으로 이용하는 기관에서는 전자적 기록을 활용할 수 있게 하는 프로그램에 대한 저작권 제한의 조항을 추가적으로 제정하거나 기존의 조항들을 개정할 수 있는 가능성이 있는가?**

저작권법 제31조 제1항은 복제의 대상을 “도서·문서·기록 그 밖의 자료(이하 ”도서등“이라 한다)”라고 하여 다른 조항들과 달리 예시적으로 광범위하게 규정하고 있으므로, 동 규정상의 “도서등”이라 함은 문서자료 외에도 사진, 음반, 영상물, 전자적으로 생산되는 도서 등 모든 정보매개물을 포함하는 개념으로 볼 수 있습니다.

그런데 전자적으로 생산되는 도서 등의 경우 열람하는 과정에서 이와 같은 도서 등을 보여 줄 수 있는 프로그램의 설치 및 재생이 필연적으로 수반됨에도 불구하고 법에서는 이와 관련한 규정을 두고 있지 않으므로, 도서관 등에서 위와 같은 전자적 도서를 열람을 목적으로 복제하기 위해서는 위와 같은 프로그램 저작자의 별도 허락을 얻는 과정이 필요하게 됩니다. 그리고 이와 같이 별도 허락을 받는 것은 경우에 따라 상당한 비용과 시간을 요구하기 때문에 전자적 도서 등의 이용자가 급증하고 있는 시점에서 상당한 불편이 초래될 수 있습니다.

그러나 저작권법상 “도서등”은 이와 같은 도서등을 활용하기 위한 도구로서 기능하는 프로그램과는 전혀 성격을 달리하는 것입니다. 문의 주신 바와 같이 위 “도서등”의 개념에 이를 디스플레이 해 줄 수 있는 프로그램을 포함해야 한다는 견해는 문언의 해석을 넘어서 위와 같은 프로그램저작자의 권리를 심각하게 해치는 것으로 사료됩니다. 이미 업계에는 도서 등을 디스플레이 하도록 하는 프로그램(혹은 소프트웨어)이 개발되어 판매되고 있고, 관련 시장도 이미 형성되어 있기 때문입니다. 따라서 위와 같은 규정의 미비로 인해 불편함이 발생한다고 하더라도 이것이 저작권법 제31조 제1항 “도서등”과 관련하여 이를 디스플레이 해 줄 수 있는 프로그램까지 포함해야 하는 합리적인 이유가 된다고 보기는 어렵다고 판단됩니다. 오히려 이러한 조항으로 인해 관련 산업을 고사시킬 가능성도 있습니다.

다만, 이와 관련하여 특히 공중 즉, 불특정 다수인에게 제공하기 위한 목적으로 이용하는 기관에서 전자적 기록을 활용할 수 있게 하는 일부의 프로그램에 대한 저작권 제한의 조항의 필요

성을 논의하고자 한다면 **저작권법 제35조의3 포괄적 공정이용 조항**에 대한 검토는 가능할 것 같습니다.

2011년 한·미 FTA에 따라 새롭게 도입된 동 규정의 취지를 살펴보면, 저작물의 디지털화와 유통환경 변화에 따라 기존 저작권법상 열거적인 저작재산권 제한규정으로는 다양한 상황에서의 저작물 이용을 모두 아우르기 어려운 한계가 있으므로, 이러한 환경 변화에 대응하여 적용될 수 있도록 포괄적인 저작재산권 제한규정이 도입된 것입니다. 동 규정에 따르면 특정한 이용행위에 한정하지 않고, 어떠한 이용행위이든지 그것이 공정이용에 해당하는지 여부를 법에 예시되어 있는 네 가지 기준을 고려하여 판단할 수 있습니다.

법 규정은 아래와 같습니다. 이하에서는 동 규정을 중심으로 전자기록물 재현을 위한 저작권법 개정 가능성에 대해 살펴보도록 하겠습니다.

> **제35조의3(저작물의 공정한 이용)** ①제23조부터 제35조의2까지, 제101조의3부터 제101조의5까지 경우 외에 저작물의 통상적인 이용방법과 충돌하지 아니하고 저작자의 정당한 이익을 부당하게 해치지 아니하는 경우에는 보도 · 비평 · 교육 · 연구 **등**을 위하여 저작물을 이용할 수 있다.
>
> ②저작물 이용 행위가 제1항에 해당하는지를 판단할 때에는 다음 각 호의 사항 등을 고려하여야 한다.
>
> 1. 영리성 또는 비영리성 등 이용의 목적 및 성격
> 2. 저작물의 종류 및 용도
> 3. 이용된 부분이 저작물 전체에서 차지하는 비중과 그 중요성
> 4. 저작물의 이용이 그 저작물의 현재 시장 또는 가치나 잠재적인 시장 또는 가치에 미치는 영향

동 조항은 보도·비평·교육·연구를 예시로 들고 있지만 이에 해당하지 않더라도 저작물의 통상적인 이용방법과 충돌하지 아니하고 저작자의 정당한 이익을 부당하게 해치지 아니하는 경우에는 공정이용으로 볼 여지가 있으므로, 이에 대해 살펴보도록 하겠습니다.

먼저 저작물의 이용이 영리성을 갖는다고 하여 언제나 공정이용이 될 수 없는 것은 아니고 특히 귀 연구소는 대학 내에서 학문 또는 연구 활동 목적으로 전자기록물을 재현하고자 하는 것으로 영리성이 있더라도 그 정도가 매우 낮다고 할 것이며, 오히려 그 이용이 가지는 공익적 측면이 강하다고 할 것입니다. 판례도 동 규정이 도입되기 전 저작권법 제28조(공표된 저작물의 인용)를 적용한 사안이기는 하나, 영리적·상업적인 성격을 가지고 있긴 하지만 공익적 측면

이 강하다고 하여 공정한 인용으로 인정한 사례가 있습니다(대법원 2006. 2. 9. 선고 2005도 7793 판결).

그리고 이용 저작물이 전자기록물로서, 그 이용의 범위 또한 타인의 저작물을 전부 복제하였다고 하여 공정이용의 가능성이 전적으로 부정되는 것은 아니고, 이용 목적 달성에 필요한 범위로 한정하여 이용하는 것은 다른 고려요소를 종합적으로 고려할 때 충분히 가능할 것으로 사료됩니다.

마지막으로 동 저작물의 이용이 전자적 기록을 활용할 수 있게 하는 프로그램저작물에 대한 시장 수요를 대체하는 범위까지로 확대된다면 공정이용으로 인정되기 어려울 것이나, 저작자의 정당한 이익을 부당하게 해치지 않은 일부의 경우라고 한다면 공정이용으로 인정될 가능성도 있습니다.

따라서 현행 저작권법 하에서 전자기록물 활용방안 만을 위해서 새로운 저작권 제한 사유를 추가하는 개정은 업계의 공감을 이끌어내기 어려울 것으로 보이고, 다만 개정된 포괄적 공정이용 조항의 해석을 벗어나지 않는 범위 내에서 전자기록물을 활용할 가능성은 충분히 있다고 사료됩니다.

**3. 도서, 문서, 기록을 재현할 수 있는 구형 프로그램에 대한 저작권을 문화체육관광부에 기증을 하여 저작권법상의 컴퓨터프로그램의 저작재산권의 존속기한인 50년이 지나지 않은 구형 컴퓨터프로그램에 한하여 개발사로부터 문화체육관광부로 비영리의 목적으로 저작재산권을 기증할 수 있는 방법으로 디지털 납본 등과 같은 제도를 도입할 수 있는 방법에 대한 제안은 어떠한가?**

먼저 저작권법 개정(2011. 6. 30.개정, **2013. 7. 1.부터 시행**)에 의해 컴퓨터프로그램 저작재산권의 존속기한이 과거 50년에서 70년으로 연장되었습니다. 따라서 컴퓨터프로그램저작물의 경우 일반저작물과 마찬가지로 업무상저작물 등이 아닌 한 원칙적으로 **저작자 사망시로부터 70년의 보호기간**이 인정됩니다.

이를 전제로 문의 주신 디지털 납본 등의 제도 도입에 대해 살펴보도록 하겠습니다.

납본제도는 도서관법 제2조 제8호에 따르면 도서관자료를 발행하거나 제작한 자가 일정 부수를 법령에서 정한 기관에 의무적으로 제출하는 제도로서, 동법 제20조에 의하여 누구든지 도서관자료를 발행 또는 제작한 경우 이를 국립중앙도서관에 납본하여야 합니다. 그러나 이 도서관자료에 온라인 자료는 제외된다고 명시하고 있습니다. 온라인 자료는 동법 제2조 제9호에 따르면 정보통신망(이용촉진 및 정보보호 등에 관한 법률 제2조 제1항 제1호의 정보통신망을 말한다)을 통하여 공중송신(저작권법 제2조 제7호의 공중송신을 말한다)되는 자료를 말합니다.

그리고 디지털 납본 제도(Digital Repository System)는 인쇄자료 뿐만 아니라 온라인으로 생성·유통되는 자료나 문서 즉, 디지털 자료에 대해서도 도서관으로 제출하는 것을 의무화하는 제도를 말합니다.

만약 이와 같은 “디지털” 납본제도가 인정 및 정착된다면, 디지털로만 생산된 국가 문헌이나 정부 자료 등을 종이책과 마찬가지로 국립도서관에 데이터베이스화하는 작업이 의무화되어, 정보에 대한 최소한의 접근성 보장을 통해 국민의 정보 불평등을 해소할 것으로 기대됩니다.

그러나 아직까지 인쇄자료는 납본이 의무화되었으나, 온라인상의 디지털 자료의 경우 납본 의무가 없는 상태입니다[도서관법 제20조의2 (온라인자료의 수집) 참조]. 도서관법은 국립중앙도서관의 경우에는 의무적으로 보존가치가 높은 온라인 자료를 선정하여 수집·보존하도록 하되, 이와 같이 수집되는 온라인 자료의 일부라도 판매용인 경우에는 그 온라인 자료에 대하여 정당한 보상을 할 것을 명시하고 있습니다(제20조의2 제1항, 제5항).

위와 같은 관련 제도 및 관련법을 종합하여 살펴보면, 디지털 자료의 경우에는 납본제도 추진 노력에도 불구하고 인쇄 자료와 달리 납본을 의무화하지 않고 일정한 경우 정당한 보상을 조건으로 수집·보존하도록 규정한 개정법의 취지에 비추어, 현행법 하에서는 도서등을 재현할 수 있는 구형 프로그램에 한하여 기증할 수 있는 제도를 도입하는 것이 쉽지는 않을 것으로 사료됩니다.

물론 정부도 기증 등 공유저작물의 이용활성화를 공감 및 추진하고 있으나, 그러한 정부의 노력에도 불구하고 기증은 사적 자치에 속하는 것으로 권리자가 자유롭게 선택할 문제이지 정부 차원에서 강제할 수 있는 영역은 아닙니다.

또한 프로그램저작물에 대한 저작재산권도 헌법상 보장된 재산권으로 그 재산권의 내용과 한계는 법률로 정하고, 공공필요에 의해 재산권을 수용·사용 또는 제한하기 위해서 법에 명시적인 근거를 요하며, 이와 같이 재산권을 제한하기 위해서는 법률에 근거하여 그에 대한 정당한 보상을 지급하여야 합니다(헌법 제23조).

따라서 도서관법을 개정하여 디지털 납본 제도를 도입하기 위해서는 우선적으로 공공필요를 입증하여야 할 것입니다. 즉 디지털 납본제도가 1) 전자기록물 재현을 위한 목표 달성에 적합한 수단에 해당하고, 2) 이와 같은 목적 달성을 위해 디지털 납본 제도를 도입하는 것이 가장 적은 침해를 가져오는 수단이고, 3) 동 제도를 도입함으로 인하여 얻어지는 공익과 이로 인하여 침해되는 저작자의 사익(재산권 침해)을 비교형량하여 공익이 더 큰다는 것을 입증하여야 할 것입니다. 쉽지는 않겠지만 만약 이와 같은 요건(1)~3))들을 입증할 수 있다면 납본제도 도입도 가능할 수 있습니다.

다만 그 도입 제안은 같은 여러 가지 상황을 종합하여 고려해 볼 때, 현재 판매하지 않고 있어 구입이 불가능한 프로그램의 경우에 한하여 문제되고 있는 당해 기록을 재현할 목적으로만 한정하여야 할 것으로 보입니다.

**4. ① “공공기관”이 과거 구입해서 가지고 있는 옛날버전의 소프트웨어를 내부적으로 장기간 보유하는 것(소프트웨어의 아카이빙)은 저작권법에 저촉되는가?**

질문의 괄호에서 소프트웨어 아카이빙을 어떠한 의미로 사용한 것인지 명확하지는 않으나, 단순히 “장기간 보유하는 것”에 초점을 맞추어 살펴보도록 하겠습니다.

과거 구입해서 가지고 있는 구 버전의 소프트웨어에 대해 기간제한 없는 라이선스 계약을 체결했던 경우라면 영구적인 이용권을 갖는다고 볼 수 있으므로 장기간 보유하는 것은 전혀 문제가 없습니다.

그렇지만 기간을 정하여 계약을 체결한 경우라면 기간만료시 소프트웨어를 폐기하는 것이 일반적일 것입니다. 따라서 이를 위반하여 장기간 보유하는 것은 저작권자가 허락한 이용

허락의 범위를 넘어 저작권법에 저촉될 것입니다(제46조).

다만 문의 주신 바와 같이 “공공기관”이 구입하여 가지고 있는 소프트웨어라면 구입과 동시에 국유재산에 해당하기 때문에 우선하는 관련법에 의해 처리될 것입니다.

또한 기간을 정하여 라이선스 계약을 체결했다고 하더라도 내부적으로 장기간 보관하는 것이 말 그대로 사용을 전제하지 않은 백업·보관의 목적인지 혹은 사용이나 공개·서비스 등을 목적으로 하는 것인지에 따라 저작권법 적용 여부가 달라질 수 있습니다.

만일 언급하신 소프트웨어 아카이빙이 백업이나 보존을 넘어 대내외적인 이용을 전제로 하는 것이라면 원칙적으로는 저작권자로부터 허락을 얻어야 할 것이나, 그 이용이 저작권법 제35조의3 공정이용에 해당할 수 있는 경우라면 저작권 침해라고는 볼 수 없을 것입니다. 따라서 일률적으로 보유하는 것만으로 저작권법에 저촉된다고 보기는 어렵습니다.

**5. ② 국가기록원에서 구버전의 도스용 훈글을 다수 가지고 있는데 기관 내 활용 및 대국민 서비스 하기 위해서는 구형의 소프트웨어의 설치가 필요하다. 현재 (주)한글과 컴퓨터 사와의 해당 소프트웨어 라이선스 계약이 끝난 상황에서 해당 소프트웨어를 재설치 하여 사용 시 저작권법에 저촉되는 것인가?**

해당 저작권자마다 라이선스 정책이 다르므로 각 권리자의 라이선스 정책에 대해 확인을 하는 것이 가장 안전합니다.

다만 문의 주신 (주)한글과컴퓨터의 경우 라이선스 계약 체결을 통해 자신의 소프트웨어를 사용하도록 허락한 경우이므로, 현재 해당 소프트웨어 라이선스 계약이 끝난 상황에서 재계약 등을 통한 권리자의 허락 없이 재설치·사용하는 것은 저작권법 제46조 이용허락 조항을 위반한 것임과 동시에 재설치 과정에서 복제권을 침해(저작권법 제136조 제1항 제1호 참조)하여 저작권법에 저촉된다 할 것입니다.

물론 타인의 저작물은 저작권자와 사용자간의 개별적이고 구체적인 계약에 따라 저작권자가 허락한 이용 방법 및 조건의 범위 안에서 이용(법 제46조 제2항 참조)할 수 있는 것이기 때문에, 저작권자인 ㈜한글과컴퓨터가 귀 기관에 대하여 계약이 만료된 이후라도 대국민서비스 등

의 공익적 성격을 갖는 특정한 목적에 한정하여 소프트웨어를 계속 사용하도록 허락하였다면 복제권 등 저작권침해의 문제는 없습니다. 그러나 이는 양사의 계약에 관한 사항이므로 기관 내 활용 및 대국민 서비스에 대해 사전에 협의가 되어 있었어야 하고, 업계의 관행으로 볼 때 '기관 내 활용'의 범위가 업무용 활용에 가깝다면 이용허락을 받아야 할 것으로 생각됩니다.

만일 반드시 구버전의 도스용 한글 프로그램을 재설치하여 사용하여야만 제공할 수 있는 서비스라면, 이는 현재 판매되지 않는 제품에 해당하는 것이고 그 목적 또한 제한적인 것이라고 생각됩니다. 따라서 저작권자인 ㈜한글과컴퓨터와 별도 계약을 체결하거나 기증을 제안하는 것도 한 방법일 것입니다.

**6. ③ 현재 한글과 컴퓨터의 공공기관용 라이선스 프로그램의 경우 라이선스 계약이 아닌 패키지 구입시 구매시점부터 영구적으로 사용되는 권한을 갖게 된다. 현재 이와 같은 라이선스 정책을 펼치고 있는데, 이 경우 현재까지 생산된 모든 소프트웨어를 포함하는 것인가?**

앞서 말씀 드린 바와 같이 해당 권리자마다 라이선스 정책이 다르므로 각 권리자에게 확인하는 것이 가장 정확합니다.

다만 패키지 구입시 영구적인 사용권한을 갖게 된다는 것의 의미는 일반적인 경우 구입한 버전에 한정하는 것으로 이해됩니다. 따라서 (주)한글과컴퓨터가 현재까지 생산된 모든 소프트웨어를 포함한다고 보기는 어렵습니다.

만일 귀 기관이 영구적인 사용권을 제공하는 패키지 소프트웨어를 구입한 경우라면, 해당 버전의 소프트웨어는 이용허락계약의 범위 내에서 영구적으로 사용할 수 있습니다.

**7. ④현재 (주)한글과 컴퓨터 사의 최신 소프트웨어를 라이선스를 계약하여 사용하고 있는데 기존에 구 버전의 경우 사용권이 없는 것인가?**

저작권법 제5조에 의하면 원저작물을 번역·편곡·변형·각색·영상제작 그 밖의 방법으로 작성한 창작물인 2차적저작물은 독자적인 저작물로 보호되고, 제22조에 의해 저작자는 2차적저작물을 작성하여 이용할 권리가 있습니다. 질문에서 기술한 한글 프로그램으로 예를들어 설명한다면, 일반적으로는 '한글 1.5' 버전이나 '한글 2012'버전이나 동일한 소프트웨어(컴퓨터프로그램 저작물)로 생각될 수 있이나, 엄밀하게 법적으로 구분한다면 '한글 1.5' 버전과 '한글 2012' 버전 두 개의 소프트웨어는 2차적저작물로 볼 수 있어 각각 다른 저작물에 해당하게 되는 것입니다.

따라서 새로운 버전이 있다고 해서 구 버전을 당연히 사용할 수 있다고는 할 수 없습니다. 서로 다른 저작물이기 때문입니다.

그러나 계약에 의해서 권리자가 구 버전에 대한 사용을 허락하였다면 해당 저작물을 이용할 수 있습니다. 이때도 전체 수량은 동일해야 합니다. 즉 기존에 라이선스 계약을 체결하여 사용하였던 소프트웨어의 수가 새롭게 사용하고 있는 소프트웨어의 수와 구 버전의 수에서 최신 버전의 수를 제외한 나머지를 합한 수에 일치하는 범위 내에서 기존 구 버전의 사용권이 있습니다. 예를 들어 (주)한글과컴퓨터의 구 버전 소프트웨어 10개에 대해 라이선스 계약을 체결하여 사용하였고, 그 후 최신 소프트웨어를 5개 사용하는 경우 전체 10개 중에서 최신 버전 5개를 제외한 나머지 5개의 한도 내에서 구 버전 소프트웨어를 사용하는 것은 전혀 문제가 없습니다. 최신 버전 5개와 구 버전 5개를 함께 사용하는 것은 전체 10개를 사용하는 것과 수가 일치하기 때문입니다. 따라서 만약 최신 버전을 10개 사용하고 있는 경우라면 이미 10개를 사용하고 있기 때문에 구 버전에 대한 사용권이 없게 됩니다. 다만, 이와 같은 사용권의 범위도 저작권자마다 다를 수 있기 때문에 해당 권리자에게 확인을 하는 것이 가장 정확합니다.

일반적으로 ㈜한글과컴퓨터의 경우 구버전의 사용은 허용하여 주는 것으로 알려져 있으니, 하위버전의 사용이 필요하면 저작권자와 협의하면 됩니다.

## 8. ⑤내부 즉, 기관내에서 사용할 뿐 만 아니라 외부 사용 즉 대민 서비스를 할 경우 해당 소프트웨어의 반출은 저작권법에 저촉되는 것인가?

문의하신 해당 소프트웨어의 반출은 저작재산권의 하나인 "대여권"(저작권법 제21조)과 관련된 문제입니다. 도서 등의 경우 권리자에게 대여권이 없기 때문에 도서 반출이 법적으로 아무

런 문제가 없지만, 판매용 프로그램의 경우 판매용 음반과 함께 저작물이 판매된 후에도 저작자가 대여권 갖습니다.

따라서 내부 사용은 물론 외부 사용의 경우에도 대여권을 갖는 저작자의 허락 없이 소프트웨어를 반출하면 저작자의 대여권을 침해(저작권법 제136조 제1항 제1호 참조)하게 됩니다. 이러한 점에서 다른 저작물과 구별됩니다.

다만, 저작권법은 판매용 프로그램을 영리를 목적으로 대여할 권리를 가진다고 규정하고 있으므로 만약 위와 같은 반출이 '영리적 목적을 가지지 아니한 개인적인 대여로 인정'될 수 있다고 보는 시각도 있습니다. 다만 이 대여가 설치하여 사용(복제)한 후 삭제하지 않고 반납하는 형태의 대여는 아니라는 것에 유의해야 합니다.

그리고 입법론으로는 아직 우리나라에서는 도입하고 있지 않지만 대여권과 구별되는 개념으로 **'공공대출권'** 의 도입을 논의해 볼 수 있을 것 같습니다. 공공대출권제도는 1946년에 덴마크에서 시작된 이래 영국·독일·네덜란드·오스트레일리아 등에서 성공적으로 시행되고 있는 제도로, 도서관소장의 도서나 음반녹음물을 **공중**에게 대출하는 것을 인정하되 이용된 분만큼 저작자는 판매의 기회를 잃어 재산적 이익을 잃게 되므로 보상금을 지급하여야 한다는 것을 내용으로 합니다. 국내에서도 일부에서는 오프라인 도서 등 뿐만 아니라 온라인 파일에 대해서도 공공대출권을 도입할 것을 제안하고 있습니다.

따라서 대민서비스의 경우 그 목적을 고려할 때 위 제도의 개정(도입) 가능성을 제안해 볼 수 있을 것으로 사료됩니다.

## 9. ⑥국가기록원의 경우 공공기관으로써 대민서비스의 경우 공공서비스로 분류되는데 이러한 경우 공적인 사용으로도 저작권법에 저촉이 되는가?

공적인 사용이라고 해서 저작물의 자유이용이 당연히 허용되는 것은 아닙니다. 그러한 사용이 저작권법에 저촉되지 않기 위해서는 법이 규정하는 저작재산권 제한 사유 중 어느 하나에 해당되어야 합니다.

국가기록원의 대민서비스의 경우 공표된 저작물의 인용(법 제28조) 또는 저작물이 공정한 이용(제35조의3) 규정과 같은 저작재산권 제한 사유를 검토해 볼 수 있고, 대민서비스의 목적 및 범위가 광범위하더라도 그 목적과 범위를 구체적으로 특정할 수 있다면, 2번 문항에서 상술한 바와 같이 포괄적 공정이용 조항의 적용가능성을 검토할 여지는 있습니다.

> 저작권법 제28조(공표된 저작물의 인용) 공표된 저작물은 보도·비평·교육·연구 등을 위하여는 정당한 범위 안에서 공정한 관행에 합치되게 이를 인용할 수 있다.
>
> 제35조의3(저작물의 공정한 이용) 저작물의 통상적인 이용 방법과 충돌하지 아니하고 저작자의 정당한 이익을 부당하게 해치지 아니하는 경우에는 보도·비평·교육·연구 등을 위하여 저작물을 이용할 수 있다.

## 10. 공공기관에서 위와 같은 시나리오를 수행시 문제가 있을 경우 어떠한 저작권과 관련한 법적인 문제가 있는 것인가?

기본적으로 구매자와 저작권자 사이에는 우리가 흔히 라이선스계약이라고 부르는 이용허락계약이 존재합니다. 따라서 앞서 말씀 드린 바와 같이 공공기관에서 공적인 목적을 위해 사용한다는 이유만으로 별도의 라이선스 계약을 다시 체결하지 않고 임의로 구 버전의 소프트웨어를 재설치하여 사용한다거나 반출하는 것은 당사자 사이에 존재하는 계약을 위반한 것으로 **민사상 손해배상 책임**(저작권법 제125조)을 질 수 있습니다.

권리자의 입장에서는 제123조에 따라 침해할 우려가 있는 자에 대하여 침해의 예방 또는 손해배상의 담보를 청구할 수 있고, 침해자에 대하여는 침해의 정지청구와 침해행위에 의해 만들어

진 물건의 폐기나 기타 필요한 조치를 청구할 수 있으며, 이와 같은 본안소송 이전에 임시로 가처분 즉, 침해행위의 정지 또는 침해행위로 말미암아 만들어진 물건의 압류 그 밖의 필요한 조치를 명할 수 있습니다.

그리고 경우에 따라 복제권 등의 침해가 있고 권리자인 저작권자의 고소 등이 있는 경우라면 **형사처벌**(징역 또는 벌금형)을 받을 수 있습니다(이하 구체적 침해행위에 따른 벌칙의 내용은 제136조 이하 벌칙 규정 참조).

## 11. 현행법으로 보았을 때 불법일 경우 현행법을 개정하기 위해서는 어떠한 법을 개정해야 하는 것인가?

궁극적으로 소프트웨어 반출(대여)을 의도하는 개정의 경우라면 다음과 같은 접근이 가능할 것 같습니다.

먼저 8번 문항에서 상술한 바와 같이 위와 같은 대여가 영리를 목적으로 하지 아니한 개인적인 대여의 개념에 포섭될 수 있다면 저작권법 제21조의 대여권을 침해하지 않는 것으로 볼 수 있을 것입니다.

그렇지만 대국민서비스의 경우 만약 영리를 목적으로 하지 아니한 경우로 본다고 하더라고 “개인적인” 대여로 인정되지 않을 가능성이 크기 때문에, 전술한 바와 같이 그 필요성과 목적에 비추어 저작권법에 공공대출권의 도입가능성을 제안해 볼 수 있을 것으로 사료됩니다.

끝.

## 다. 한국 저작권 위원회 법률 상담 결과

❒ 프로젝트 팀 시나리오 질문

- 질문 : ①공공기관이 과거 구입해서 가지고 있는 옛날버전의 소프트웨어를 내부적으로 장기간 보유하는 것(소프트웨어의 아카이빙)은 저작권법에 저촉되는가?

- **답변 : 공공기관이 과거에 구입한 소프트웨의 라이선스가 저작권사로부터 영구라이선스의 구매일 경우 해당 소프트웨어를 장기보유(소프트웨어 아카이빙)은 가능하나 임대라이선스일 경우에는 저작권사마다 다를 수도 있으나 한정적인 예외사항을 각기 라이선스 정책으로 표명하고 있기 때문에 저작권과 관련한 법적인 분쟁일 발생할 요지가 있음**

- 질문 : ②국가기록원에서 구버전의 도스용 훈글을 다수 가지고 있는데 기관내 활용 및 대국민 서비스 하기 위해서는 구형의 소프트웨어의 설치가 필요하다. 현재 (주)한글과 컴퓨터 사와의 해당 소프트웨어 라이선스 계약이 끝난 상황에서 해당 소프트웨어를 재설치 하여 사용 시 저작권법에 저촉되는 것인가?

- **답변 : ① 질문과 라이선스 관련한 문제는 같은 맥락에서 이해할 수 있으며, 구형 소프트웨어의 설치시 공공기관에서 구매하지 않은 경우 현재 절판된 소프트웨어의 경우 저작권사인 개발사로부터 해당 소프트웨어 구입과 관련한 논의가 필요할 것임**

- 질문 : ③현재 한글과 컴퓨터의 공공기관용 라이선스 프로그램의 경우 라이선스 계약이 아닌 패키지 구입시 구매시점부터 영구적으로 사용되는 권한을 갖게 된다. 현재 이와 같은 라이선스 정책을 펼치고 있는데, 이 경우 현재까지 생산된 모든 소프트웨어를 포함하는 것인가?

- **답변 : 소프트웨어 라이선스 정책은 저작권사별로 각기 다르지만 일반적으로는 소프트웨어 버전별, 종류별로 상세히 구분지어서 라이선스 정책을 펼치고 있음, 라이선스 정책과 관련한 문제의 경우 저작권사 별로 다양한 정책을 펼치고 있기 때문에 해당 저작권사와 협의가 필요함**

- ④현재 (주)한글과 컴퓨터 사의 최신 소프트웨어를 라이선스를 계약하여 사용

하고 있는데 기존에 구 버전의 경우 사용권이 없는 것인가?

- **답변 : ①, ③ 답변과 같은 맥락 이해할 수 있음**

- ⑤내부 즉, 기관내에서 사용할 뿐 만아니라 외부 사용 즉 대민 서비스를 할 경우 해당 소프트웨어의 반출은 저작권 법에 저촉되는 것인가?

- **답변 : ①답변의 영구라이선스를 획득한 기관 내에서 설치하는 경우에는 해당 컴퓨터 프로그램의 저작권이 기관에 영구적으로 귀속되어 있기 때문에 문제가 없다. 이는 대부분의 저작권사들이 그러할 것이나, end-user 즉 최종 이용자가 개인 pc이용자인지, 해당기관 인원인지, 공공 서비스 대상인 불특정 다수의 국민에 이르기 까지 다양한 계층의 이용자 대상을 전부 포함할 지에 대한 부분은 해당 저작권사의 라이선스 정책을 확인해보아야 함.**
- **외부 서비스의 경우 해당 컴퓨터 프로그램의 영구라이선스를 획득하였다고 하더라고 외부적인 반출의 경우 이는 배포권 침해로 법적인 분쟁이 발생할 수 있음**

- ⑥국가기록원의 경우 공공기관으로써 대민서비스의 경우 공공서비스로 분류되는데 이러한 경우 공적인 사용으로도 저작권 법에 저촉이 되는가?

- **답변 : 저작재산권의 제한사항의 경우에도 예를 들어 교육목적으로 인해 복제를 허용한다고 하였을 경우에 최소한의 경우에만 복제를 허용한 것이지 전체복제 및 다수복제는 불가능함, 대민서비스, 공중의 이용에도 해당 저작권사와의 협의 없이는 법적인 분쟁이 발생할 가능성이 농후함**

- 질문 : 공공기관에서 위와같은 시나리오를 수행시 문제가 있을경우 어떠한 저작권과 관련한 법적인 문제가 있는 것인가?

- **답변 : 현행 법의 제, 개정을 위해 저작권사와의 협상시 한글과 컴퓨터 사의 경우 국내 소프트웨어 개발기업이라 말이 통할 수 도 있으나 마이크로소프트의 경우 한국마이크로소프트에서는 힘이 없다. 라이선스 정책도 미국 마이크로소프트 본사에서 내려주는 것이기 때문에 마이크로 소프트나, 어도비와 같은 해외 유명 소프트웨어 개발 본사와의 협상이 추가적으로 필요하며, 구형이든 신형이든 프로그램의 저작권과 관련해서는 강경한 정책을 펼치고 있기 때문에 난관이 예상 될 것임**

- 한미 fta 체결로 인해 기존의 저작물의 보호기간이 50년에서 70년으로 연장되는 것으로 현행법이 개정되었으며, 개발사에서 생산한 컴퓨터 프로그램의 경우 업무상 저작물로써 공표한 이후 70년, 개인개발의 경우 저작권자 사후 70년으로 개정되었음

8. 2 제 38호 『기록학연구』 투고 논문

논문

# 전자기록 장기보존 전략으로서의 에뮬레이션 사례 분석*

김 명 훈** · 오 명 진***
이 재 홍**** · 임 진 희*****

1. 머리말
2. 장기보존 전략으로서의 에뮬레이션 분석
  1) 에뮬레이션의 개념 및 특징
  2) 에뮬레이션의 유형
3. 에뮬레이션 전략 사례
  1) CAMiLEON 프로젝트의 에뮬레이션 방식 분석
  2) KB의 에뮬레이션 방식 분석
  3) Planets 프로젝트의 에뮬레이션 방식 분석
  4) KEEP 프로젝트의 에뮬레이션 방식 분석
4. 시사점과 전망
  1) 사례 분석의 시사점
  2) 우리나라의 적용을 위한 전망

* 본 연구는 "안전행정부 국가기록원의 기록물 보존기술 연구 개발사업"의 일환인 "2013년도 행정기관 전자기록물 재현기술 연구 및 프로토타입 개발 연구사업"의 재정지원을 받아 수행되었음.
** 한국외국어대학교 경보 · 기록관리학과 겸임교수(주저자). 주요 논저 : 『전자기록 평가론』, 신진리탐구, 2009; 「전자기록 환경에서의 '업무친화적' 기록관리 방향성 분석」, 『정보관리연구』 제38권 제4호, 2007.
*** 한국외국어대학교 경보 · 기록관리학과 강사(공동저자). 주요 논저 : 「공공영역의 2004년 대통령 탄핵사건 기록」, 『기록학연구』 제33호, 2012.
**** 명지대학교 디지털아카이빙연구소 연구원(공동저자).
***** 명지대학교 기록정보과학대학원 교수(교신저자). 주요 논저 : 「영구기록물관리를 위한 기록물 데이터베이스 스키마 개발 방향(공저)」, 『기록학연구』 제34호, 2012; 「대량기록물 처리를 위한 영구기록물관리시스템의 디지털저장소 배치형상 연구(공저)」, 『기록학연구』 제32호, 2012.
•투고일 : 2013년 9월 23일 •최초심사일 : 2013년 9월 28일 •게재확정일 : 2013년 10월 25일.

8. 3 <제 5회 전국 기록인대회> 발표자료



## ◈ 제5회 전국기록인대회 세션별 프로그램 / 2013년 11월 2일(토)

| 장소 / 시간 | 제1회의장 [사범관 422호] | 제2회의장 [사범관 423호] | 제3회의장 [사범관 424호] | 제4회의장 [사범관 516호] | 제5회의장 [사범관 517호] |
|---|---|---|---|---|---|
| 10:00 – 12:00 | **한국기록학회**<br>**세션명 : 공공기록에서 발견하는 개인들의 삶**<br>사회 : 김명훈(한국외대)<br>•발표1 : 감시와 증명: 진실화해위원회 기록 속에 담긴 국가와 개인 / 김선호(김영삼민주센터)<br>•토 론 : 최동운(서울시청)<br>•발표2 : 공공기록으로 살펴보는 한국전쟁기 개인생애사와 한국전쟁의 역사적 실체 재조명 / 신기철(금정굴인권평화재단)<br>•토 론 : 심성보(한국국가기록연구원) | **부산대학교 · 성공회대학교 SSK 산업변동과 로컬리티 연구단**<br>**세션명 : 로컬리티 기록화 – 지방기록 관리를 위한 도전**<br>사회 : 이주연(부산대)<br>•발표1 : 로컬리티 기록의 탐구와 기록화 전략 / 송정숙(부산대)<br>•토 론 : 위영(국가기록원)<br>•발표2 : 지역 노동기록의 의미와 노동사 아카이브: 성공회대 SSK 연구팀의 사례 / 이재성(성공회대 노동사연구소)<br>•토 론 : 최재희(이화여대)<br>•발표3 : '기록으로 보는 부산의 교통공간' 설계 사례 / 현문수(부산대)<br>•토 론 : 이상봉(부산대) | **한국문헌정보기술(주)**<br>**세션명 : 표준 RMS 도입, 그 이후…**<br>사회 : 김화경(한국문헌정보기술(주))<br>•발표1 : 보유목록관리시스템 소개: 비전자 기록물 관리와 평가 관리를 위한 보유목록관리시스템의 개발 이야기 / 이성재(한국문헌정보기술(주))<br>•발표2 : 디지털 자산관리 시스템 DAMS(Digital Asset Management System) / 정헌배(한국문헌정보기술(주))<br>•발표3 : 2013년 한국문헌정보기술 주요 프로젝트 성과 소개 / 김화경(한국문헌정보기술(주)) | **명지대학교 박사과정 연구팀**<br>**세션명 : 기록관리현장의 오늘과 실제**<br>사회 : 이은영(명지대)<br>•발표1 : 검찰청 특수기록관 설립배경 및 기대효과 / 이현정(대검찰청)<br>•발표2 : 지질분야 연구기관의 연구기록 관리 사례 / 이미영(한국지질자원연구원)<br>•발표3 : 대학 기록관리 매뉴얼 소개 / 김성겸(국가기록원) | **목천재단 건축아카이브**<br>**세션명 : "목천김정식문화재단 건축아카이브"의 컬렉션과 향후 과제**<br>사회 : 임진희(명지대)<br>•발표1 : 목천건축아카이브 구축현황과 역할: 민간기관의 기록활동 / 조준배(앤드건축사사무소)<br>•발표2 : 목천건축아카이브 구축사례: 아날로그아카이브와 디지털아카이브 구축사례를 바탕으로 / 김태형(목천김정식문화재단)<br>•종합토론 : 박석훈(한국문헌정보기술(주)) |
| 13:30 – 15:30 | **한국기록관리학회 추계학술대회**<br>**세션명 : 민간아카이브 공유와 이용촉진을 위한 기술(Description)과 기술(Technology)**<br>사회 : 설문원(부산대)<br>•발표1 : 민간 아카이브를 위한 기술(description) / 현문수(부산대)<br>•토 론 : 이미영(한국지질자원연구원)<br>•발표2 : 메타데이터를 통한 공공문화정보 통합관리 및 서비스 방안 / 홍선희(한국문화정보센터)<br>•토 론 : 조은희(한국가스안전공사) | **한국외대 기록학연구센터**<br>**세션명 : 기록학의 지평 확산**<br>사회 : 이정연(한국외대)<br>•발표1 : 사회적 기억 구축을 위한 기록관리의 역할 / 김명훈(한국외대)<br>•토 론 : 심성보(한국국가기록연구원)<br>•발표2 : 기록의 전시와 디지털 사진기록 다시 생각하기 / 조민지(한국외대)<br>•토 론 : 이영남(한신대) | **이화여대 기록관리교육원**<br>**세션명 : 국내외 기록관리 현황과 발전 방향**<br>사회 : 최재희(이화여대 기록관리교육원)<br>•발표1 : 국내 지방기록물관리기관 연구동향 분석 / 송슬기(이화여대 기록관리교육원)<br>•토 론 : 이주연(부산대)<br>•발표2 : 일본의 디지털 아카이브 / 박경희(이화여대 기록관리교육원)<br>•토 론 : 현문수(부산대)<br>•발표3 : 대통령기록관 전시 · 교육 서비스 프로그램 연구 / 김혜진(이화여대 기록관리교육원)<br>•토 론 : 김령은(UNIST도서관) | **한국기록전문가협회**<br>**세션명 : 특별지방행정기관의 기록관리 현황과 과제**<br>사회 : 박종연(한국기록전문가협회)<br>•발표 : 특별지방행정기관의 기록관리 현황 및 개선과제 / 남경호(국가보훈처)<br>•토론 : 중앙행정부처, 유관기관, 국가기록원 기록연구사 4명 | **행정기관 전자기록물 재현기술 연구 및 프로토타입 개발 연구팀**<br>**세션명 : 에뮬레이션 전략 해외 사례 및 국내적용 방안과 이슈**<br>사회 : 오명진(한국외대)<br>•발표1 : 전자기록 장기보존 전략으로써의 에뮬레이션 사례 분석 / 이재홍(명지대 디지털아카이빙연구소)<br>•발표2 : 에뮬레이션 기반의 전자기록물 재현 방안 연구 / 최주호((주)이산티에스) |
| 15:50 – 17:50 | **명지대학교 인간과기억아카이브**<br>**세션명 : "인간과기억아카이브"의 탄생과 협력적 성장 모형**<br>사회 : 임진희(명지대)<br>•발표1 : 협력을 통한 "5월 12일 일기" 수집, 그리고 일상아카이브 / 황진현(명지대 디지털아카이빙연구소)<br>•발표2 : 기록인 기증 컬렉션과 명지대 아카이브 수집 전략 / 성효주(명지대 기록정보대학원)<br>•발표3 : 명지대학교 인간과기억아카이브의 설립과 AtoM시스템 구축 사례 / 안대진(명지대 디지털아카이빙연구소) | **전북대학교 BK21+ 지역문화콘텐츠 융복합전문인력양성사업단**<br>**세션명 : 지역 문화콘텐츠 아카이빙 전략**<br>사회 : 김길훈(전북대)<br>•발표1 : 지역문화콘텐츠 융복합전문인력양성사업단 / 김 용(전북대)<br>•발표2 : 지역성의 발견과 보존: 전주 영화제 기록 관리 / 윤은하(전북대)<br>•발표3 : 지역 기록물과 지역 아카이브의 과제 : 전라북도 일반동산문화재 조사를 중심으로 / 이선아(전북대) | **한국국가기록연구원**<br>**세션명 : 민간 아카이브 발전 전략의 모색**<br>사회 : 이진렬(영남대)<br>•발표1 : 해외 민간 아카이브 쇼케이스 / 이상민(한국국가기록연구원)<br>•발표2 : 민간아카이브 발전을 위한 단계별 전략 / 곽건홍, 김해중(한남대)<br>•발표3 : 민간 아카이브 활성화를 위한 첫걸음 모색 / 심성보(한국국가기록연구원)<br>•종합토론 : 설문원(부산대), 이경용(국가기록원), 임종철(한국문헌정보기술(주)) | **한국외국어대학교 박사과정 연구팀**<br>**세션명 : ICA AtoM으로 아카이브 시스템 구축하기**<br>사회 : 최윤진<br>•발표1 : ICA AtoM 소개 및 개발이력 / 최윤진<br>•발표2 : ICA AtoM 설치 및 이용법 / 김형희<br>•발표3 : ICA AtoM 아카이브시스템으로 이용하기: 수집형 아카이브 / 최동운<br>•발표4 : ICA AtoM 아카이브시스템으로 이용하기: 이관형 아카이브 / 박민영<br>•발표5 : 해외 구축 사례 및 ICA AtoM KOREA Forum 2013 / 이보람<br>•발표6 : ICA AtoM의 국내 활용을 위한 과제 / 황진현 | **대통령기록관리 연구팀**<br>**세션명 : 대통령기록관리 위기와 과제**<br>사회 : 김유승(중앙대)<br>•발표 : 박종연(한국기록전문가협회), 정진임(투명사회를 위한 정보공개센터)<br>•토론 : 김유승(중앙대), 전진한(투명사회를 위한 정보공개센터), 이상민(한국국가기록연구원) |

제5회 전국기록인대회 세션별 프로그램

기록관리, 지평의 확대

# 행정기관 전자기록물 재현기술 연구 및 프로토타입 개발 연구팀/ 에뮬레이션 전략 해외 사례 및 국내적용 방안과 이슈

사회: 오명진(한국외대)

11월 2일(토)

13:30 – 15:30


전자기록 장기보존 전략으로써의 에뮬레이션 사례 분석

/ 이재홍(명지대 디지털아카이빙연구소) ····· 285

에뮬레이션 기반의 전자기록물 재현 방안 연구

/ 최주호((주)이산티에스) ····· 291

| NO.37 |

# 전자기록 장기보존 전략으로써의 에뮬레이션 사례 분석

이 재 홍

명지대 디지털 아카이빙 연구소

국가기록원 National Archives of Korea

# 전자기록 장기보존 전략으로써의 에뮬레이션 사례 분석

2013. 11. 02

명지대 디지털 아카이빙 연구소 이 재 홍



명지대학교 디지털 아카이빙 연구소 DIGITAL ARCHIVING INSTITUTE

## 1. 에뮬레이션의 개념 및 특징(1/2)

### 에뮬레이션의 개념

*"디지털 정보를 생산한 시점에서 사용된 하드웨어, 매체, 운영 체제, 소프트웨어의 운용을 그대로 흉내 내어(emulate) 그 내용을 읽어내는 프로그램을 통하여 재현하는 보존 전략"*

『기록학 용어사전』

- 에뮬레이션은 소프트웨어의 원래 기능이 현재의 컴퓨터상에서 재현될 수 있도록 하기 위해 디지털 정보가 생산.활용된 원래의 운영 환경을 재생산하는 것을 의미
- 에뮬레이션에서는 현재의 컴퓨터 환경에서 다른 컴퓨터 환경을 실제적으로 재현할 수 있도록 하기 위해 에뮬레이터란 소프트웨어를 필요로 함

### 장기보존 전략으로써의 에뮬레이션

- 시간의 경과에 따른 시스템 내지 하드웨어, 소프트웨어의 노후화에 따라 디지털 정보들을 수 차례 변환시킬 필요가 없는 관계상, 매체 변환 및 특정 포맷에 대한 세부 사항들을 파악하는데 소요되는 시간과 비용을 절감
- 디지털 정보의 기타 보존 전략들에서는 디지털 정보의 포맷 및 유형, 기타 개별적 특성에 따라 분산화된 복잡한 보존 전략들을 채택해야 함에 반해, 에뮬레이션은 일관된 보존 전략을 유지할 수 있음
- 에뮬레이션은 디지털 객체가 보유하고 있는 고유한 '기능성'(functionality)을 원본 그대로 재현할 수 있는 가능성을 보여줌으로써 장기보존 전략으로서 큰 잠재력을 지님

| NO.38 |

# 에뮬레이션 기반의 전자기록물 재현 방안 연구

최 주 호

(주)이산티에스 대표

# "에뮬레이션 기반의 전자기록물 재현 방안 연구"

2013년 11월 02일

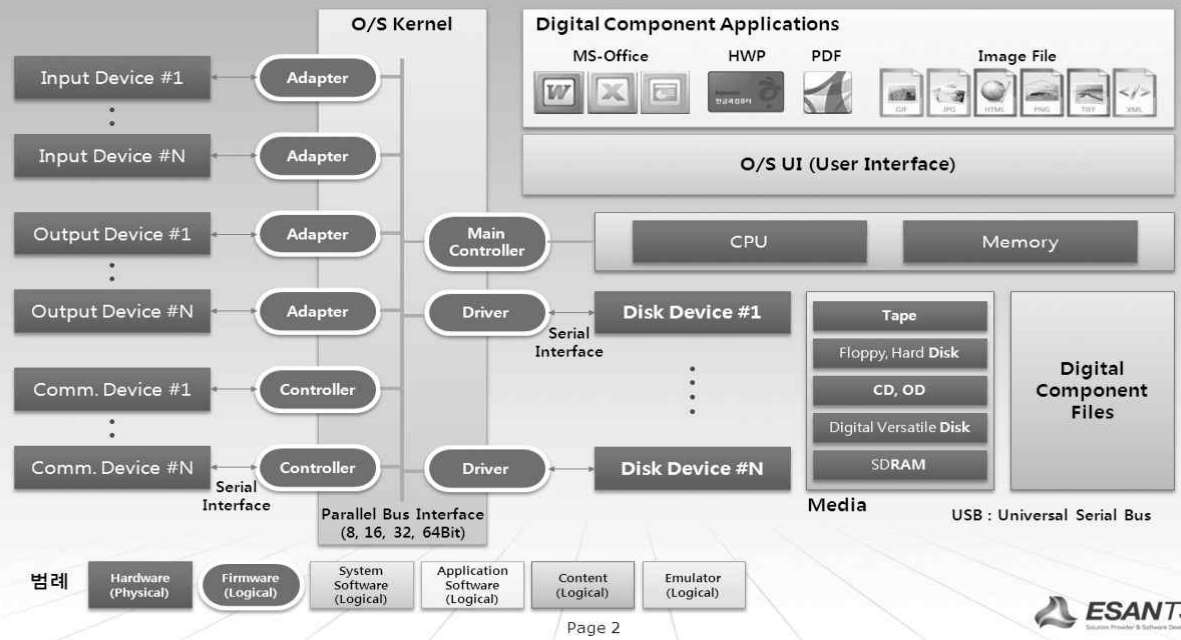
(주)이산티에스 최 주 호

이산티에스 Solution Provider & Software Developer

국가기록원 National Archives of Korea

## 1. PC 기본 구조

- Digital Component를 생성 관리하는 PC(Personal Computer) 기본 구조에 대한 이해
- Digital Component를 에뮬레이션 하기 위해 필요한 Physical/Logical 요소 검토




Page 2

ESANTS Solution Provider & Software Developer

## 8. 4 추가 해외사례 자문

# 미국과 영국의 전자기록 장기보존 전략 사례

개 요

1. 미국의 전자기록 장기보존전략 ERA
2. ERA 시스템 개요와 시스템 요건
3. 전자기록 장기보존전략으로서의 ERA에 대한 평가
4. 영국 TNA 디지털기록 장기보존전략
5. 미국 주정부 디지털기록보존연합 컨소시엄의 기존 디지털보존 전략 평가

## 1. 미국의 전자기록 장기보존전략 ERA

○ 미국 국가기록관리처 (NARA)는 전자기록아카이브즈 Electronic Records Archives (ERA)라는 전자기록장기보존이용 프로젝트를 2000년부터 시작했고, 2005년부터 2011년까지 실제 애플리케이션 개발했다. 전자특수매체기록서비스과에서 주도한 이 ERA 사업은 2000년 1월 공식적으로 출범하여 2004년에 2억불의 예산을 승인받아 본격적으로 추진되기 시작했다. 개발사로는 록히드사가 선정되었다.

○ ERA 사업의 배경: 1998년의 미국 연방정부에서의 전자기록관리의 상황을 보아야 한다. 당시 전 세계적으로 전자기록물의 보존과 이용에 관한 해결책과 경험이 전무하며 전자기록관리 분야에서 증명된 보존 및 열람제공 방법이 없었다는 인식에서 ERA 사업이 출발했다.

○ 기록관리 선진국들의 기록학계에서도 그 성공을 보장할 수 있는 정책 전략 표준을 제공하지 못했다. 이로 인해 NARA는 관련학계와 함께 다양한 연구 프로젝트를 수행했다. DOCT 연구 프로젝트, "Electronic File Wrapper", SDSC의 대량파일 관리와 장기 보존에 관한 연구계획들, OAIS 개발 프로젝트와 협력, NSF(NPACI)와의 협력, InterPARES와의 협력 프로젝트가 그 대표적인 연구 프로젝트이다. 이러한 연구프로젝트에서의 공통 과제는 디지털기록의 호환 작용성(Interoperability)과 정보자산의 항구성이었다.

○ ERA는 기본적으로 인캡슐레이션 전략을 선택했고, 기록포맷 전환 기술적 기능을 포함한다. ERA는 미국 의회도서관(Library of Congress)과 기술적 협력에 의해 수행되었다. 따라서 미국의 ERA 개념을 이해하려면 2000년부터 시작된 미국 의회도서관의 디지털 정보 장기보존 전략 개발 과정과 결과를 이해해야 한다.

○ ERA는 미국연방정부에서 수천 개의 포맷으로 생산되는 전자기록물을 하드웨어나 소프트웨어에 관계없이 진본성과 불변성을 해치지 않고, 보존 이용할 수 있게 하려는 NARA의 디지털기록 장기보존전략이다.

○ ERA로 인해 "NARA의 사명은 모든 형태의 전자기록물의 보존과 즉각적인 이용을 보장하는 것이다"라고 NARA의 비전이 변화했다.

○ ERA로 연방기관의 전자기록물이 이관되기 위해서는 연방기관의 전자기록시스템이 DoD 5015.2-STD 표준을 준수하고 그 기능요건을 충족하는 국방부 승인을 받은 것이어야 한다(2008년 NARA 공지). 또한 이메일 등을 ERA로 이관하기 위해서는 이메일 아키빙 애플리케이션에 대한 NARA의 지침을 준수해야 한다.

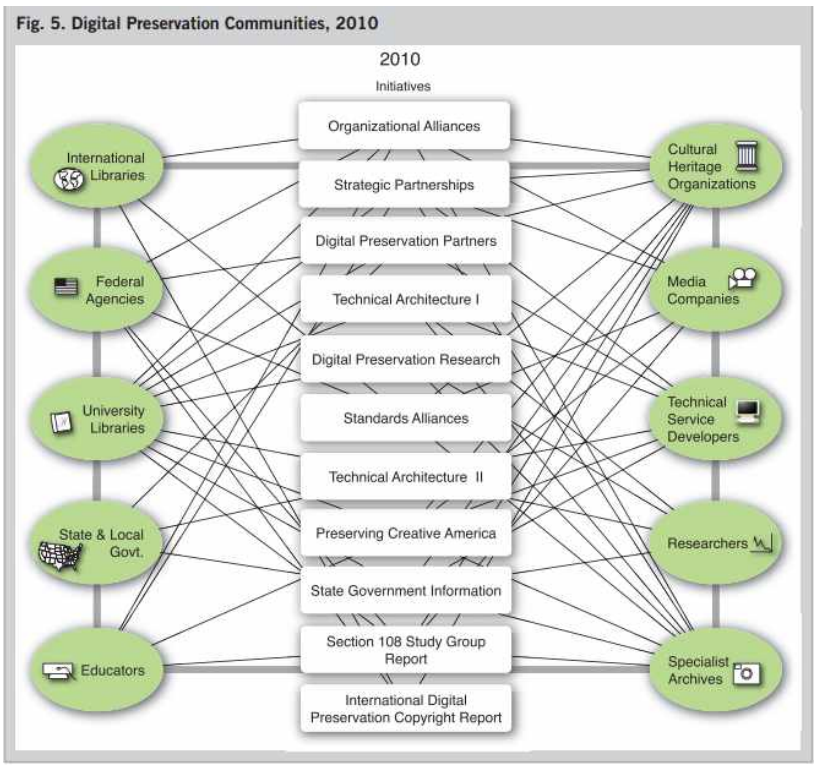
### 1.1 미국 의회도서관의 디지털기록 장기보존 전략

○ 2000년 미국 의회는 '국가디지털정보인프라스트럭처와 보존프로그램'(NDIIPP)법을 제정(PL106-554). 의회는 이 목적을 위해 1억달러의 예산을 승인했다. 미국 외회도서관이 이 법정 업무에 따라 전국의 디지털 정보의 수집, 보존, 이용에 관한 업무를 조직하고 국가적인 전략을 수립하고 이행하게 함.

- NDIIPP에 따라 수백개의 파트너십 기관이 참여하여 디지털 정보의 보존에 관한 표준, 최선의 실무, 도구와 서비스를 개발하고 제공함. 법령에 언급한 디지털 정보에는 중요한 공공정책과 관련한 지리위치정보, 웹사이트, 시청각 기록물, 이미지와 전자 텍스트 기록을 포함함.
- 의회가 제정한 법령에 의해 NARA를 포함한 연방기관은 의회와 협력하여 디지털 보존 국가전략을 수립해야 했음.
- 2003년에 의회는 '디지털유산 보존: NDIIPP의 계획'을 승인.
- 2007년에 예술 분야의 디지털 정보 보존을 위해 '창조적 아메리카 보존'(Preserving Creative America)를 결성하고, 2008년에는 23개의 주정부를 포괄하여 '주정부 보존 이니시에이티브'를 조직하여 4개의 주요 프로젝트를 지원했다.
- 2011년에 '우리의 디지털 유산 보존: 국가디지털정보인프라스트럭처와 보존프로그램 2010 보고서'를 발표했다. 이 보고서에 미국 의회의 디지털 정보 보존에 관한 업무 성과와 결과물이 요약되어 있다.[101]

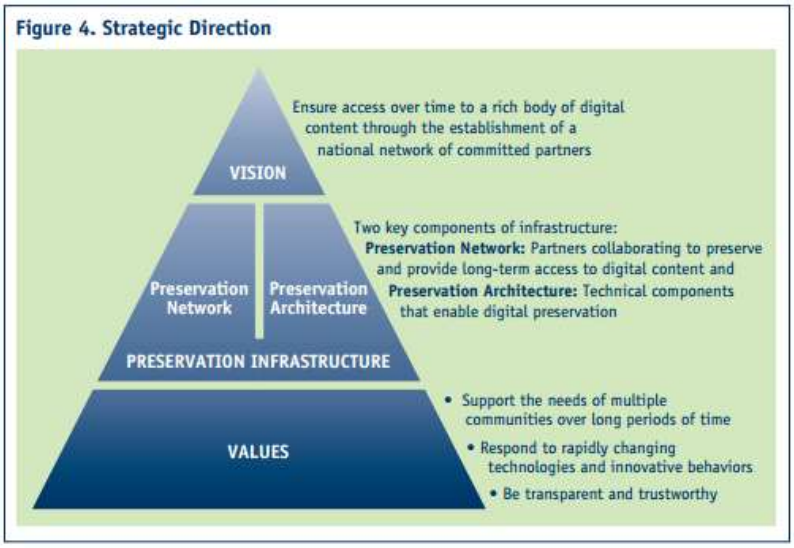
○ ‘국가디지털정보인프라스트럭쳐와 보존프로그램’(NDIIPP)가 추구하는 비전은 전국적 관련기관 네트워크를 설립하여 풍부한 디지털 컨텐츠를 장기간에 걸쳐 접근 가능하게 하는 것이다. 이를 뒷받침하는 사회적 가치는 다양한 조직/공동체에게 장기간에 걸쳐 각각의 디지털 기록정보의 보존 필요를 지원하는 것과 급격히 변화하는 기술과 혁신에 대응하는 것, 투명하고 신뢰성이 있는 디지털 보존 프로그램을 만드는 것이다.

○ ‘디지털보존 인프라스트럭쳐’는 ‘디지털보존 네트워크’와 ‘디지털정보 아키텍쳐’로 구성된다. ‘디지털정보 아키텍쳐’는 디지털 정보가 파트너십 네트워크 전체에 디지털 보존을 가능하게 하는 시스템의 구조, 논리 구성요소, 논리관계를 규정하는(기술적 스펙을 부여하는) 기술적 프레임워크를 말한다.




<그림 1> 국가디지털정보보존 네트워크 (p.23)

101) U.S., Library of Congress, *Preserving Our Digital Heritage: The National Digital Information Infrastructure and Preservation Program 2010 Report,* 2011.




<그림 2> '국가디지털정보인프라스트럭처와 보존프로그램'(NDIIPP) 전략 방향

○ '국가디지털정보인프라스트럭처와 보존프로그램'(NDIIPP)에 의해 '전국디지털스튜워드십연맹' (National Digital Stewardship Alliance)이 결성되어 디지털 기록(컬렉션)을 보존하고 '배분 디지털보존 인프라스트럭처'를 제공한다.

○ 이 '국가디지털정보 인프라스트럭처와 보존'에서 고수하는 **'보존 아키텍처 개발 원칙'**은 다음과 같다.

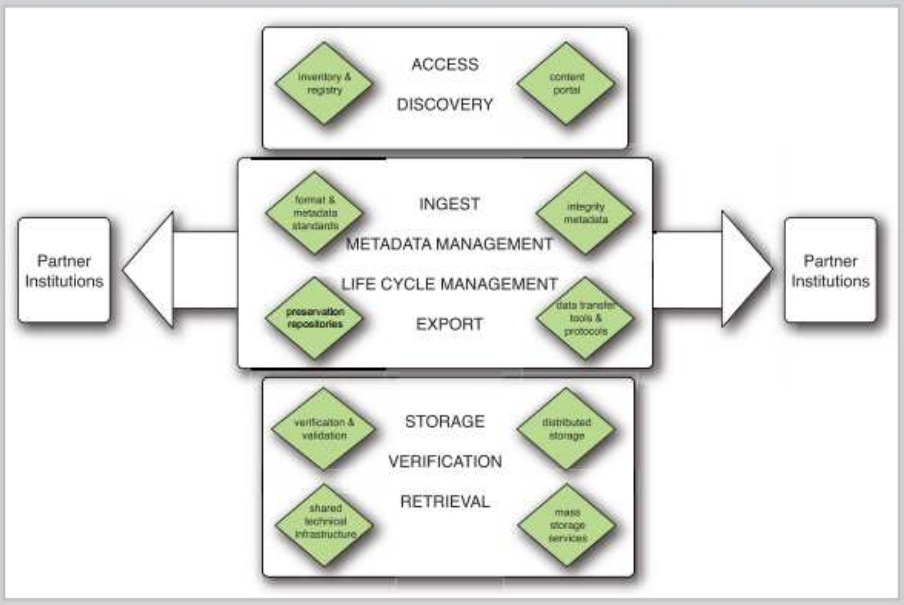
디지털 보존 아키텍처는;

- 기관간의 관계를 지원해야 한다.
- 접근 문제를 보존문제와 분리해서 취급할 수 있게 해야 한다.
- 이용가능한 기존 기술과 성과를 이용해서 모듈 방식으로 수립해야 한다.
- 단 한번에 다 구축하는 것이 아니라 장기간에 걸쳐 조합될 수 있게 해야 한다.
- 전체 시스템을 해체/재구축함이 없이 부분적으로 업그레이드 될 수 있어야 한다.
- 널리 사용가능한 프로코콜을 사용해서 스펙을 결정해야 한다.

○ 이를 위해 제안된 아키텍처는 4개의 레이어를 갖는데, 리포지토리 레이어. 게이트웨이 레이어, 컬랙션 레이어, 인터페이스 레이어를 갖는다.

**Fig. 8. Technical Architecture for Digital Preservation**

The three-layer architectural framework tested and modeled by digital preservation partners in a variety of projects has proven to be flexible and extensible across a diverse community of partners.




<그림 3> 디지털 보존 기술 아키텍쳐 (p.28)

## 2. ERA 시스템 개요와 시스템 요건

### 2.1 ERA 시스템 개요

○ ERA는 전자기록을 포함한 모든 연방기록의 기록처분일정 수행을 지원하고, 기록을 이관받는 과정을 지원하는 시스템이다. ERA는 인터넷으로 접속해서 업무를 수행하는 시스템이며 연방기관에서 전자적 형태로 생산되는 다양한 기록을 관리하는 시스템이다.

○ ERA는 기본적으로 OAIS 정보 모형을 기반으로 설계되었다. OAIS 모형의 외부 디지털 데이터 상호관계를 그림으로 제시한다.

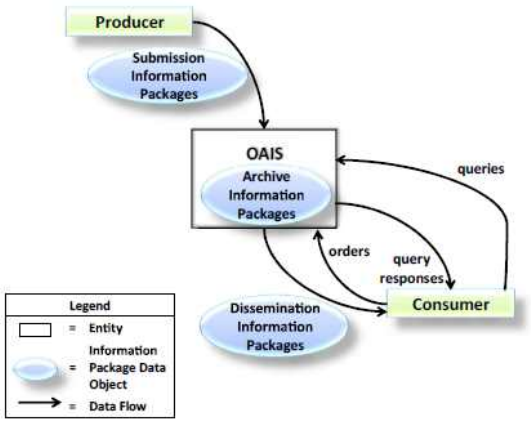
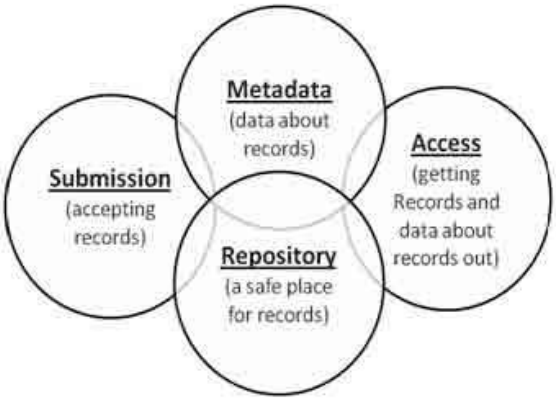



Figure 2-4: OAIS Archive External Data

<그림 4> OAIS 표준의 외부 디지털관계 모형

○ ERA는 4개의 시스템이 결합된 복합시스템이다(system of sytems). ERA는 1. 기록제출 시스템, 2.메타데이터 시스템, 3. 보존서고 시스템, 4. 접근시스템으로 구성되었다. 각 시스템별로 각기 다른 법적 규제적 환경에 맞추어 시스템을 운영한다.




<그림 5> ERA 4개의 시스템 모형

○ 1. 기록제출 (Submission) 시스템,: 연방정부기관이 전자기록과 메타데이터를 ERA에 제출하기 위해 이 시스템과 이 기능을 사용. 즉 모든 연방기관이 이 시스템을 이용할 수 있음. 국가기록관리처는 이 시스템을 이용하여 기록을 인수받아 입고처리한다. 연방기관은 모든 포맷의 전자기록을 매체 전환 없이 NARA에 이관한다. 즉, ERA는 모든 포맷의 전자기록을 ERA의 애플리케이션을 통해 에뮬레이션으로 원본 포맷 기록를 이용하게 하는 시스템이다.

○ 2.메타데이터 (Metadata) 시스템: 미국 국가기록관리처의 영구기록 정보, 영구기록 생산기관 정보, 영구기록으로 선별된 사유 정보, 기록처리과정이 적용된 내역이 메타데이터에 수록되어 이용하고 작성할 수 있다.

○ 3. 보존서고 (Repository) 시스템: 전자기록은 이 전자기록보존서고에 저장되고 검색 이용된다. 비종이기록(시청각기록 포함)의 디지털화된 기록을 장기보존할 수 있게 태생적 전자기록과 장기보존을 병행 수행할 수 있는 시스템으로 개발했다.

○ 4. 접근 (Access)시스템: 이 시스템을 사용해서 일반 시민이 장기보존되고 있는 전자기록을 검색해서 이용할 수 있다. 이 시스템 안에서 아키비스트가 비공개부분을 처리할 수 있다.

○ ERA 이용자인 연방기관의 ERA 이용 방식:
- 연방기관 아이디를 부여받아 웹으로 로그인하여 온라인상으로 기록처분일정 승인을 요청할 수 있다.
- 즉, ERA의 기록처분일정 결정 기능을 통해 기록물 평가를 수행한다.
- 연방기관의 영구기록 이관을 요청할 수 있고, 원래 포맷 상태의 기록을 이관할 수 있다.

- 연방기관의 영구기록을 ERA에 제출하여 저장 보존할 수 있다.
- 다음과 같은 기록처분업무 전자양식을 온라인상으로 이용한다.
  - Records Schedules (RS)
  - Records Schedule Items (RSI)
  - Legacy Records Schedule (LS)
  - Legacy Records Schedule Item (LSI)
  - Transfer Request (TR)
  - Transfer Processing Results (TPR)
  - Legal Transfer Instrument (LTI)
  - Folders
  - Transfer Plan (TP)
  - Data Type Template (DTT)
  - Code List (CL)

○ ERA 이용자인 NARA 기록관리직원의 ERA 이용 방식:
- 연방기관을 대신하여 레거시 전자기록을 패키지화하여 ERA에 이관하여 보존할 수 있다.
- 디지털기록을 온라인상으로 ERA에 인수하여 입고하는 업무를 수행한다(Ingest).
- 연방기관 기록처분일정을 전자적으로 검토하고 승인한다. 예전처럼 종이 양식을 보내고 처리할 필요가 없다.

○ ERA를 본격적으로 가동한 2008년 이후 연방기관의 전자기록과 대통령실의 전자기록이
ERA에 입수되었다. 2011년에 16.6 TB의 연방기록, 82 TB의 대통령기록물이 입수되었다. 입수된 전자적 형태의 부시 대통령기록물에는 인덱스가 만들어진 2억5천만 건의 전자기록물, 2억 건의 이메일, 3백만 건 이상의 디지털 사진, 3천만 건 이상의 기타 전자기록이 포함되었다. 그리고 15.1 TB의 미국 의회기록(상원 및 하원)이 입수 보존되고 있다. 2012년 1월 현재 131 TB의 전자기록이 ERA에서 보존되고 있다.

○ 입수된 대통령 전자기록을 ERA에서 처리하고, 입수, 보존 업무를 수행하고, 기록에 대한 접근을 제공하기 위해 약 30명의 아키비스트들이 일하고 있다. 2008년 이후 ERA를 통해 66,000건의 대통령기록물 조사가 수행되었다.

## 2.2 ERA 시스템을 통한 기록물 선별평가-기록처분 개요

○ 'ERA 기록처분일정 결정 워크플로우'는 ERA를 사용한 연방기관 영구기록 선별 프

로세스를 나타내준다. 즉 ERA안에는 연방기록물의 처분일정을 결정할 수 있는 기능이 탑재되어 있으며, ERA에서 가장 중요한 기능 중의 하나이다. ERA를 사용하여 기록물이 ERA에 입고(ingest)되기 전에 무슨 기록이 인수되어야 할 것인지를 결정할 뿐 아니라 한시기록의 처분일정 결정에 따른 기록물 폐기 여부의 승인도 이루어진다.

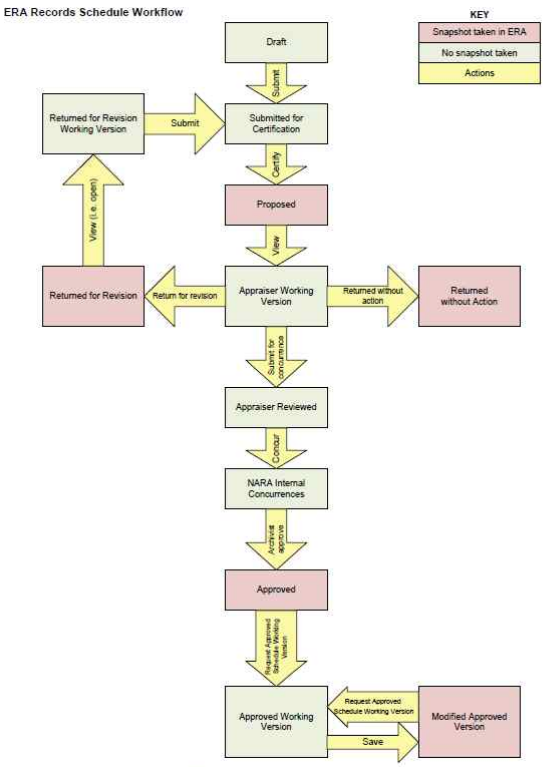
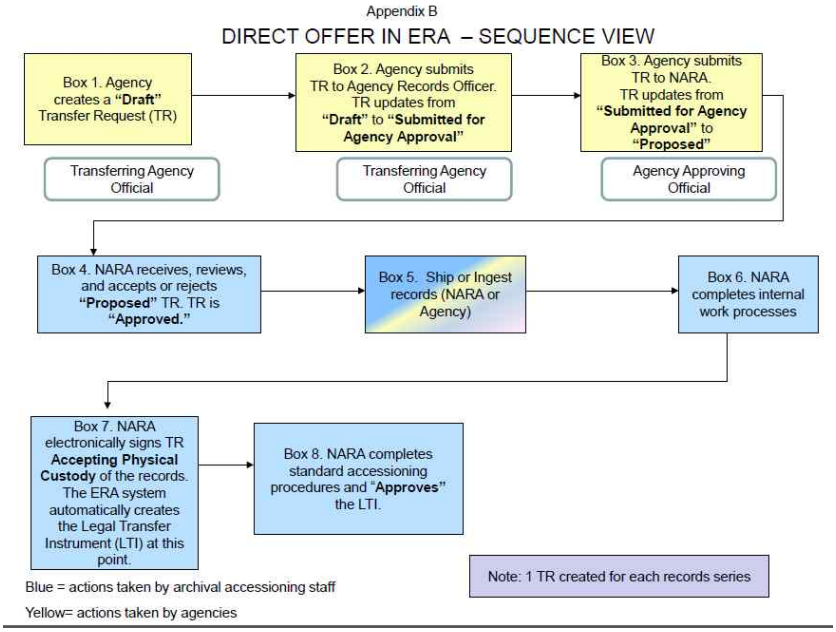
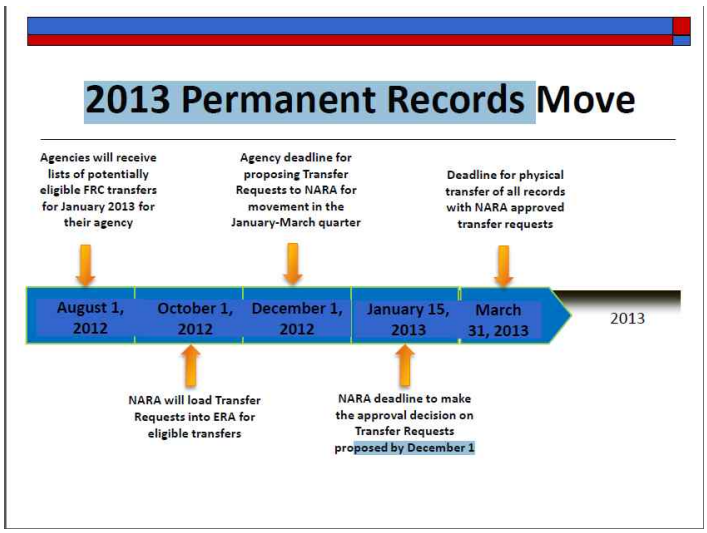



*Figure 6a – ERA Records Schedule Workflow Diagram*

<그림 6> ERA 기록선별평가 결정 프로세스, 출처: NARA, *ERA Agency User Manual*




< 그림 7> ERA 기록이관 과정 워크플로우, 출처: NARA, *ERA Agency User Manual*




<그림 8> 2013년 미국 연방정부기관의 영구전자기록 ERA 이관 일정

## 2.3 전자기록 보존 및 이뮬레이션 관련 ERA 시스템 요건[102)]

○ ERA 시스템 요건 'ERA8'의 전자기록 보존 요건(8.2)에 원래의 디지털 포맷으로 입수할 수 있어야 하고 보존할 수 있어야 한다고 요건이 규정되어 있다.
- 8.3 입수되는 데이터 파일의 속성을 회득
- 8.4 입수 데이터 파일과 각 전자 파일의 관계를 구체적으로 나타냄
- 8.5 전자기록을 다른 포맷으로 전환하는 기능 요건 부과
- 8.6 전자기록을 지속적(persistent) 포맷으로 전환하는 기능 요건 부과 (하드웨어와 소프트웨어로부터 독립적인 포맷으로 전환하는 기능 요건)
- 8.7 파일 아웃풋을 보존 목적으로 저장하는 기능 요건 부과

○ ERA 시스템 요건 'ERA15'에 모든 형태(types)의 전자기록을 입수할 수 있어야 한다는 기능 요건 부과. 데이터 유형 식별 기능,

○ ERA 시스템 요건 'ERA9'의 전자기록 보존 요건은 전자기록을 자동적으로 다른 매체에 이전할 수 있게 하는 매체관리 기능 요건을 부과.

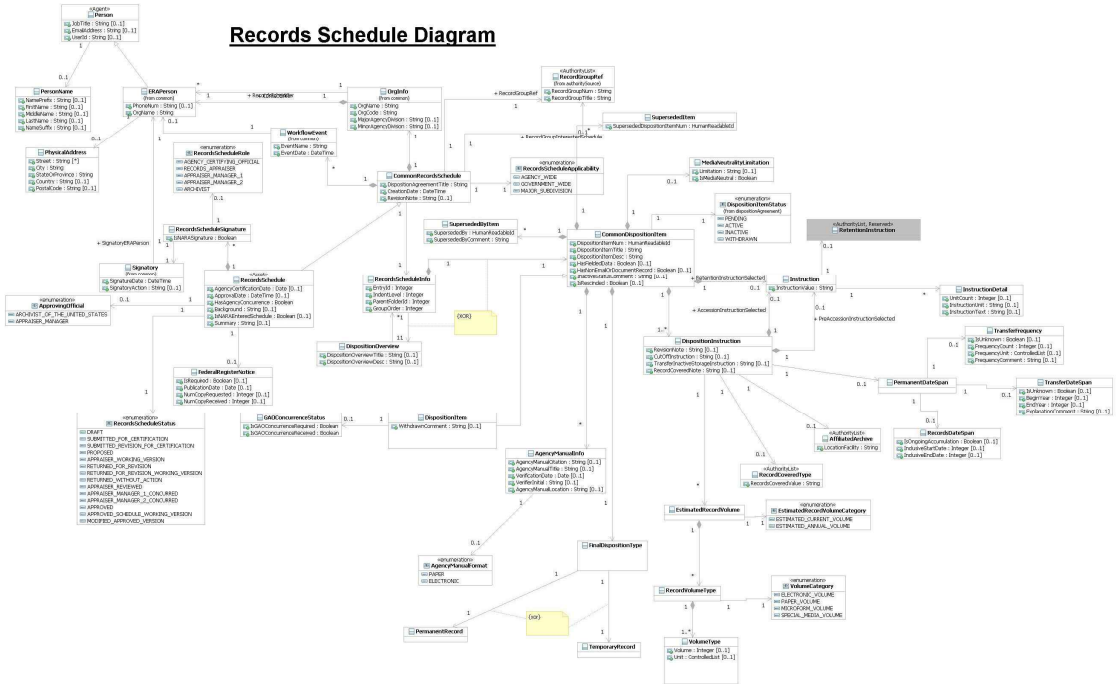
○ 특히 ERA 시스템 요건 'ERA20'은 ERA 시스템이 모든 형태의 전자적 정보 자산(모든 형태의 전자기록, 20.1)에 접근할 수 있어야 하고 전자적으로 제시할 수 있어야 한다는 기능 요건을 부과하고 있다.
- 20.4 그 전자기록이 생산되었던 하드웨어와 독립적으로 아웃풋을 제공할 수 있어야 한다고 기능 요건 부과.
- 20.5 그 전자기록이 생산되었던 소프트웨어와 독립적으로 아웃풋을 전자적으로 제공할 수 있어야 한다(records presentation)고 기능 요건 부과.

102) U.S. NARA, *ELECTRONIC RECORDS ARCHIVES REQUIREMENTS DOCUMENT (RD v4.0)*, 2010.

## 2.4 ERA 논리데이터 모델(Logical Data Model, DLM)

1. 기록처분일정 논리데이터 모델 (Records Schedule LDM)

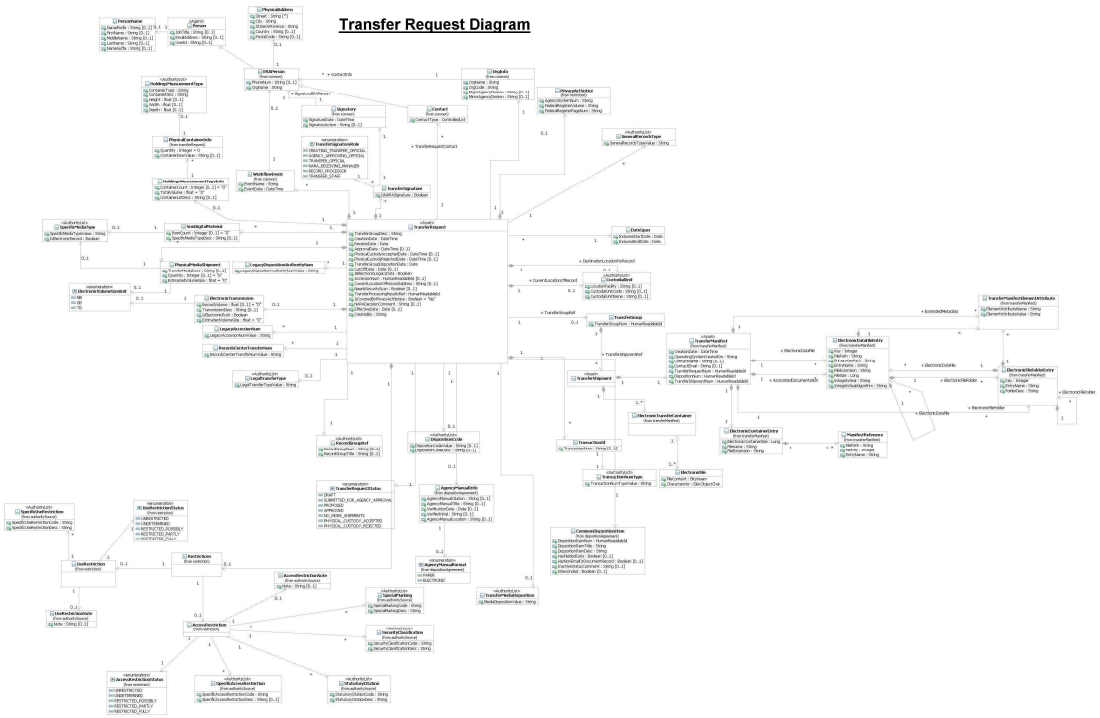
○ 기록처분일정 논리데이터 모델 (Records Schedule LDM)은 ERA 연방기록물의 기록처분일정의 데이터 클래스와 요소를 나타낸 그림이다. 기록처분 논리 설계에 유용한 도표이다.




<그림 9> 기록처분일정 논리데이터 모델 (Records Schedule LDM)

2. 기록물 이관 신청 논리데이터 모델 (Transfer Request LDM)

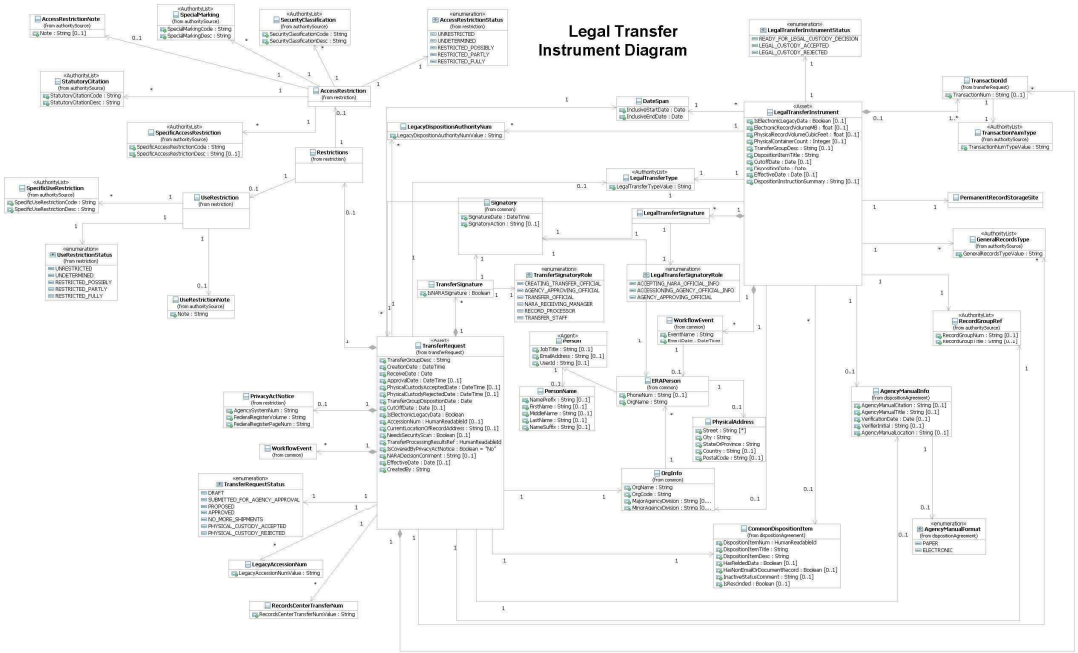
○ 기록물 이관 신청 논리데이터 모델 (Transfer Request LDM)은 ERA 연방기록물의 기록물 이관 신청 디지털 객체의 데이터 클래스와 요소를 나타낸 그림이다. ERA에서 연방기관의 기록물이 NARA로 이관되는 과정을 나타내준다. 이 이관 과정은 물리적 이관을 포함한다.




<그림 10> 기록물 이관 신청 논리데이터 모델 (Transfer Request LDM)

3. 법적 이관 도구 논리데이터 모델 (Legal Transfer Instrument LDM)

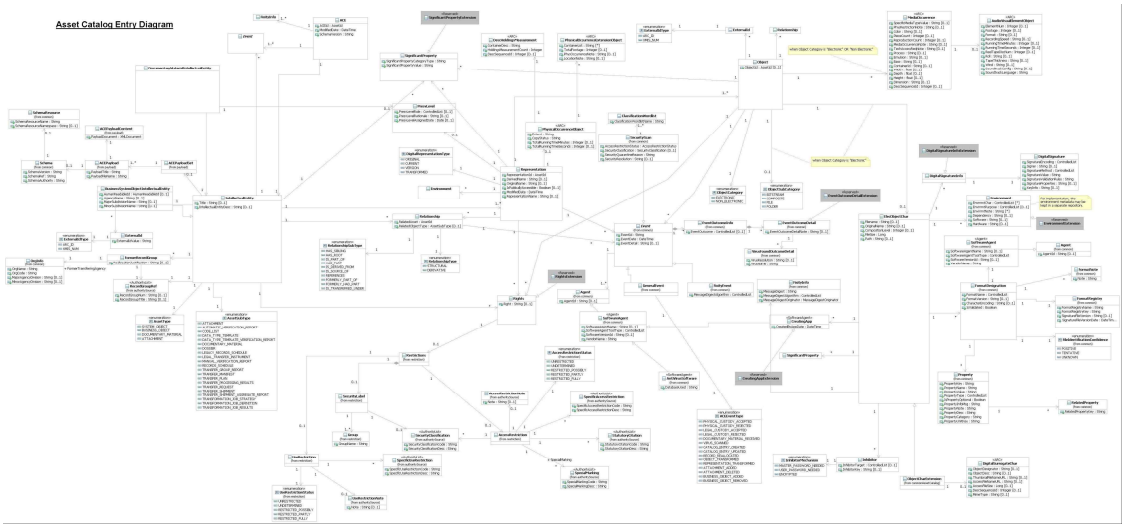
○ 법적 이관 도구 논리데이터 모델 (Legal Transfer Instrument LDM)은 ERA 연방 기록물의 기록물이 법적 이관 도구의 객체의 데이터 클래스와 요소를 나타낸 그림이다. ERA에서 연방기관의 기록물이 NARA로 법적으로 이관되는 과정을 나타내준다.




<그림 11> 법적 이관 도구 논리데이터 모델 (Legal Transfer Instrument LDM)

4. 자산 캐탈로그 엔트리 논리데이터 모델 (Asset Catalog Entry LDM)

○ 자산 캐탈로그 엔트리 논리데이터 모델 (Asset Catalog Entry LDM)는 ERA에서 ‘자산 캐탈로그 엔트리(ACE)’의 데이터 클래스와 요소를 나타낸 그림이다. ERA로 입수되는 각각의 연방 전자기록은 ‘자산 캐탈로그 엔트리(ACE)’라고 하는 관련된 메타데이터 파일을 갖고 있다. 이 ACE 구조는 디지털보존 메타데이터 PREMIS 표준에 기반을 두고 있다.




<그림 12> 자산 캐탈로그 엔트리 논리데이터 모델 (Asset Catalog Entry LDM)

## 3. 전자기록 장기보존전략으로서의 ERA 장기보존전략의 평가

○ ERA는 연방기관이 온라인상으로 기록처분일정을 수립하고 국가기록관리처로부터 승인받고 전자기록을 이관할 수 있게 개발했다. 즉 연방기록관리를 웹베이스로 온라인으로 기록처분하고 이관할 수 있게 해주는 시스템이다. 이러한 접근 방식은 연방기관에게 편리한 업무방식이다. 특히 연방기관의 영구기록 평가선별을 지원하는 기록평가업무 지원이 탁월하다. 이를 통해 ERA는 연방기관의 기록관리를 효과적으로 지원한다.

○ ERA는 2012년 1월 현재 131 TB의 연방전자기록을 지원한다. 2011년 9월에 개발이 완료되었으나 기술적으로 아직 완결된 시스템이 아니라서 더욱 많은 자원을 들여 추가 개발을 요하지만, 과도한 비용을 요하는 과제임으로 인해 현재 추가 개발이 중단된 상태이다.

○ ERA는 전자기록을 원래 포맷으로 저장 보존하는 인캡슐레이션 방식을 채택하고 있다. 이것은 미국 의회도서관의 디지털 객체 장기보존 방식이다. ERA의 주 목적은 원래 포맷의 기록을 안전하게 보존하는 것이다. 이와 더불어 기술적 메타데이터, 출처 메타데이터, 이력 및 기록관리 메타데이터를 잘 획득해서 보존하게 하는 기본적 표준을 수행하는 것을 목적으로 한다.

○ ERA는 현재 연방기록을 이용하게 해주는 검색·열람 포탈인 NARA의 OPA(Online Public Access, 온라인 기록검색 접근)에 부분적으로 연결되어 있다. 미국연방정부의 전자기록이 기록검색 포털에서 충분히 검색될 수 없다는 문제를 갖고 있다. OPA는 현재 NARA 소장 전자기록의 95%의 기술(description)을 제공하고 있다. 현재 백만건의 ERA 전자기록이 OPA에서 검색 이용될 수 있다. 따라서 이관된 모든 전자기록의 기록 내용을 통한 검색 접근하는 것은 불가능하다.

○ ERA는 전자기록 전환 기능(Transformation Framework)을 갖추고는 있지만 현재 그 기능을 활용하고 있지는 않는다고 한다. 즉 기록 포맷을 전환할 수는 있으나 전환하지는 않고 있다. ERA는 종이기록의 디지털화를 지원하지도 않는다. 다만 디지털화된 기록을 입수 보존할 수 있다.

○ ERA는 원래 형태의 전자기록 보존뿐 아니라 기록획득의 기술적 정보, 출처 정보, 이벤트 이력관리 메타데이터를 생성 보존할 수 있다. ERA는 탄력적이고 적용이 쉬운 표준에 기반한 XML 메타데이터 캐탈로그를 채택하고 있으며 이것은 메타데이터 보존을 위해 PREMIS 요소를 포함한다.

○ ERA는 ERA에 저장된 기록 내용을 기반으로 검색하여 기록을 찾을 수 없다는 단점을 가지고 있다. ERA는 기록에 첨부된 기술(description)을 보존하고 이것을 이용해서 기록을 찾을 수 있게 한다.

○ ERA 자체에서 기록에 대한 접근 재검토 리뷰나 부분 비공개처리를 수정할 수 없음으로 해서 기록이용체제로서 불충분하거나 불편하다는 점이 있다. 다만 대통령기록에 대해서는 이용제한을 재검토하는 기능과 부분삭제(redaction) 기능을 부여하여 대통령기록에 대한 정보자유 청구에 대응할 수 있게 하고 있다.

○ ERA는 비밀 전자기록과 비공개 전자기록을 보존할 수 있다. 이러한 전자기록의 접근 제한을 수행할 수 있다.

## 4. 영국의 '디지털 연속성' 전자기록장기보존 전략과 '디지털기록 인프라스트럭쳐' (Digital Records Infrastructure)

○ 영국 국가기록관(TNA)은 디지털보존연합 (Digital Preservation Coalition), 정보기록관리협회 (Information and Records Management Society), JISC, 디지털큐레이션센터 (Digital Curation Centre), 정보기술관리학회 (Society of Information Technology Management), 영국기록협회 (Archives and Records Association)과 협력하여 영국의 가치 있는 디지털 정보가 장기간에 걸쳐 이용될 수 있게 추진하고 있다. 이러한 디지털 정보보존 노력이 영국정부의 행정, 설명책임성, 투명성을 보장하는데 필요한 것임을 국가기록관 '21세기 발전전략'에서 천명하고 있다.[103]

○ 영국 국가기록관(TNA)은 2001년부터 '디지털 보존부'를 설립하여 영국정부의 전자기록 보존 및 이용 방법을 개발했다. 국가기록관은 2003년부터 디지털아카이브를 설립, 운영했다. 국가기록관의 디지털보존서고는 민간회사인 테셀라(Tessella)와 공동으로 구축했으며 '여왕기업혁신상'을 수상했다. 이들이 개발한 Safety Deposit Box는 현재 전세계 여러나라에서 사용되고 있다. 국가기록관은 '디지털 리포지토리 시스템'에서 제공하는 '온라인전자기록'(Electronic Records Online) 서비스를 통해 2005년부터 전자기록를 이용할 수 있게 하고 있다. '디지털 리포지토리 시스템'은 2013년에 개발된 '디지털기록인프라'의 기초를 제공했다.

○ 영국 국가기록관(TNA)은 '디지털 컨티뉴이티'(Digital Continuity) 전략을 수립했다. 이것은 공공기관에서 디지털 정보를 필요한 기간만큼 필요한 방식으로 업무에 이용할 수 있게 하는 전략으로 확립되었다. 영국 국가기록관(TNA)은 각급 공공기관에서 전자기록을 생산하고 계속 안정적으로 이용할 수 있게 하는 전략을 중점적으로 추진하고 있다. 즉 장기보존 이전에 전자기록의 확실한 생산 보장과 획득 보장을 최우선 과제로 인식하고 있다.

○ 영국 국가기록관(TNA)은 최소의 비용으로 하는 전자기록 보존을 도모한다. 따라서 고비용이 드는 전자기록 포맷 전환이나 인캡슐레이션 전략을 우선적으로 고려하지 않고 있다. TNA의 주요 중점은 기록전환이나 인캡슐렝션을 통한 기록 재연이 아니라 기록 메타데이터와 기록 구조의 획득과 전환이다. 그 과정을 통해 TNA의 '카탈로그'에 기록이 목록화 될 수 있게 한다. 그 결과로서 전자기록을 안전하게 보존하고, 그 전자기록을 찾을 수 있게 한다.

○ 영국 국가기록관(TNA)은전자기록 포맷의 낙후성에 의한 접근이용 불가능성에 대해

103) U.K., The National Archives, *Archives for the 21st century in action: refreshed 2012-15*, 2012

회의적인 입장을 취한다. 이러한 시각으로 인해 2013년 영국 TNA는 디지털기록인프라(Digital Records Infrastructure, DRI)라는 디지털보존서고를 설립했다. 이 보존방식에 따르면 전자기록보존시스템이 중점적으로 지원하고 수행할 작업은 “프로세스”와 “결과물”의 관점에서 디지털기록인프라를 구성하는 것이다.[104] DRI는 시맨틱 디지털 아카이브이다. DRI 카탈로그는 디지털 보존서고에 있는 전자기록의 인벤토리이다. 반면에 국가기록관 ‘카탈로그’(TNA Catalogue)는 종이기록과 디지털기록의 종합적인 입고, 편집관리, 일반 열람접근 제공을 수행하게 해주는 시스템이다. DRI 카탈로그의 메타데이터 정보 사본 1부는 TNA의 일반 접근 검색 포털인 ‘발견’(Discovery)에 보내져서 디지털 기록 검색에 이용된다.

○ 현재 영국 정부에는 10억개의 전자기록이 존재하며. 3.2 페가바이트 기록을 입고 2020년까지 6페가바이트가 입고될 것으로 예상하고 있다. 현재 250개의 영국정부기관이 전자기록을 영국 국가기록관(TNA)에 이관한다.

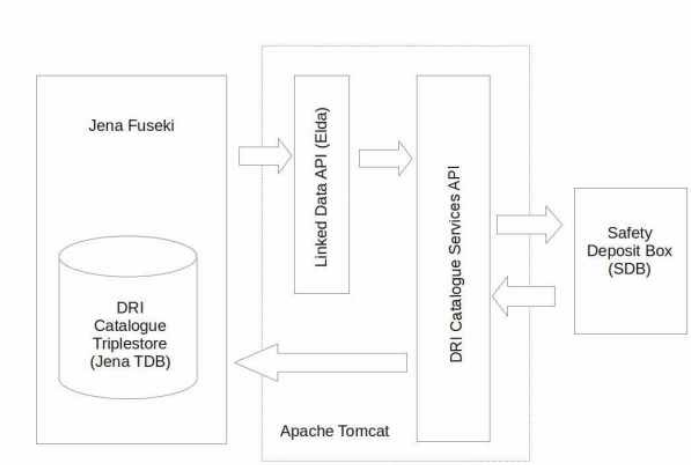
### 4.1 디지털기록인프라(DRI)의 개요

○ 디지털기록인프라의 중점은 1. 전자기록 이관과 이관 준비, 2. 안전한 보관, 3. 이관과 입고, 4, 보존시 고려 사항, 5. 인출(export)에 있다. 디지털기록인프라는 대용량 스토리지 수단으로 로보트 테이프 라이브러리를 사용하는데 수천 테라바이트의 용량처리를 가능하게 한다.

○ 디지털기록인프라(DRI) 채택 기술:

- 툴로써 RDF (D2RQ, Apache Jena를 사용, 이것은 우수한 Java API 이다.
- Turtle, SPARQL 같은 우수한 표준 포맷과 랭귀지 사용
- 개방된 open source 와 open standards 사용
- DRI는 Java 기반 시스템이며, 채택된 상용 애플리케이션도 Safety Deposit Box인데 이것도 Java system이다.

104) Rob Walpole, *The National Archives Digital Records Infrastructure Catalogue: First Steps to Creating a Semantic Digital Archive*, 2013




<그림 13> Digital Records Infrastructure(DRI) 오픈 소스 시스템 개요도

○ 영국 국가기록관(TNA)은 디지털기록인프라를 일반시민이 이용하는 기록검색 포털인 Discovery와 연결하고 있다. 이를 위해 시맨틱 웹 기술을 사용하고 있다.

○ 국가기록관(TNA)은 DROID( Digital Record Object Identification) 툴을 사용하여 전자기록의 파일 포맷을 식별한다. DROID는 250개 이상의 전자기록 포맷을 식별할 수 있게 하는 무료 소프트웨어이다. 시그니쳐 정보를 사용하는 파일도 인식한다.

○ 국가기록관(TNA)은 PRONOM이라는 디지털 파일 포맷 레지스트리(데이터베이스)를 유지하여 이를 사용하고 있으며 다른 기관도 사용할 수 있게 허용하고 있다. PRONOM은 포맷, 소프트웨어, Compression, Character 인코딩, 추가정보, 공급자, 배포 일자 등의 정보를 제공함. 이것은 실제 규격과도 같다.

○ 국가기록관(TNA) 디지털기록인프라의 기본 디지털 기록 장기보존 방식:
- 이관되는 기록과 생성되는 메타데이터를 인캡슐레이션하여 테이프 아카이브에 3개의 원본/사본을 보존한다.
- 이 3개의 원본/사본은 2개의 각기 다른 미디어/매체에 보존된다.
- 그 중 1개의 원본/사본은 접근 등 이용에 직접 사용되고, 나머지 2개의 원본/사본은 백업용으로 설계되었다.
- 백업용 2개 중 1개의 원본/사본은 재난시 복구 목적으로 다른 별개의 장소에 보존된다.

○ 디지털기록인프라 기능 개요

- **기능 1. 전자기록 이관과 이관 준비**: 주요 중점은 메타데이터와 기록 구조의 전환이다. 그 과정을 통해 TNA의 '카탈로그'에 기록이 목록화 될 수 있게 한다. 그 결과로서 전자기록을 안전하게 보존하고, 그 전자기록을 찾을 수 있게 한다.
- **기능 2. 안전한 보관:** 원 전자기록과 생성된 메타데이터를 이중 3중으로 보존하여 안전하게 보존하는 것이 중점이다.
- **기능 3. 이관과 입고**: 전자기록이 입고되면 5년~10년간 전자기록을 이전하거나 이뮬레이션을 하기 위한 처리를 않는다. 웹베이스로 해서 디지털 기록을 이관 입고 인수 할 수 있게 한다(마이크로 서비스).
- **기능 4. 보존 고려 사항**: 구조 메타데이터 , 기술 메타데이터, 기술 메타데이터를 보존한다. 이를 위해 영국 국가기록관은 '포맷 이전'과 '이뮬레이션'을 병용한 테셀라사의 상용 애플리케이션인 Safety Deposit Box를 채용했다. 전자기록에 대한 접근, 이용, 그리고 이에 대한 보고와 같은 마이크로서비스 제공
- **기능 5. 인출**

## 4.2 영국 TNA의 디지털기록 보존 애플리케이션 Safety Deposit Box(SDB)의 채택

○ 영국 국가기록관은 디지털 기록 장기 보존을 위해 테셀라(Tessella)사의 상업 애플리케이션인 Safety Deposit Box(SDB)를 채택했다. SDB의 특징은 다음과 같다.

- 웹베이스 디지털 아키빙솔류션
- 시스템의 확장성을 좋게하기 위해 마이크로 서비스를 채택
- '마이크로 서비스'에 디지털기록의 보존, 저장, 접근, 보고, 진본성과 무결성 확인 및 보호, 시스템의 확장성 포함
- SDB는 디지털 객체의 기본 개념, 정보 객체 개념, 구조-기술(descriptive)-기술(technical) 메타데이터 개념을 채택
- 스위스, 네덜란드, 핀랜드, 에스토니아 등도 SDB을 디지털 기록 보존 애플리케이션으로 채택

○ 디지털 기록 장기보존 애플리케이션으로서의 SDB 기능

- Ingest (기록 내용이 포털에서 업로드됨)
- Data Management (컨텐츠 관리)
- Preservation (이전)
- Registry (파일 포맷과 전환 패스웨이의 리소스 데이터베이스)

- Storage (포털이 스토리지 어댑터로 연결)
- Administration (시스템 프로세스 포털)
- Search (구글형 검색 포털)
- Explorer (주 접근 영역 포털)
- Access (접근 포털로서 디지털 기록 컨텐츠가 아키브에서 이출되거나 다운로드되는 것을 통제)

○ SDB는 이른바 '능동적 보존' 접근방식을 취하는데, 이 능동적 접근방식은 기술이 바뀌어도 디지털 기록에 대한 접근을 보장한다. 능동적 접근에는 2가지가 있는데, 첫째는 기록형태(포맷)을 현재에 가능한 형태로 전환하는 것이고, 둘째는 오래된 정보를 읽을 수 있게 하는 방식이다.

○ 첫 번째, '관리되는 이전' (managed migration)방식은 기록 포맷을 새로운 포맷으로 전환하는 방식인데 이것은 놀랍게도 복잡한 방식이며, 정당화되어야 하고 검증되어야 할 전문적 접근법을 요한다. 두 번째 '시뮬레이트된 컴퓨터 환경'(simulated computer environment) 방식은 오래된 OS와 애플리케이션을 오늘날의 컴퓨팅 플랫폼에서 동작시키는 방법인데 이또한 복잡한 방식이다. SDB를 개발한 테셀라사는 이 두가지 기술을 제공해야 진정하게 디지털기록을 장기보존할 수 있다고 주장한다.[105]

○ 테셀라사는 2004년부터 네델란드의 국립도서관 KB, 네델란드국가기록관과 함께 이뮬레이션 방식을 개발해왔다. 이들이 함께 개발한 모듈화된 에뮬레이터인 디오스쿠리(Dioscuri)는 2007년 7월 이후 오픈소스 에뮬레이터로 제공되기 시작했다. 테셀라는 EU의 플래넛(Planets) 프로젝트의 일환으로 이뮬레이션 디지털보존 방식의 문제점을 개선했다. 테셀라사는 KEEP (Keeping Emulation Environments Portable) 컨소시엄에도 참여하여 이뮬레이션 보존방식, 즉 디오스쿠리를 SDB에 포함시켜 상업적 애플리케이션으로 제공하고 있다.

105) http://www.digital-preservation.com/solution/active-preservation [2013.11.20]

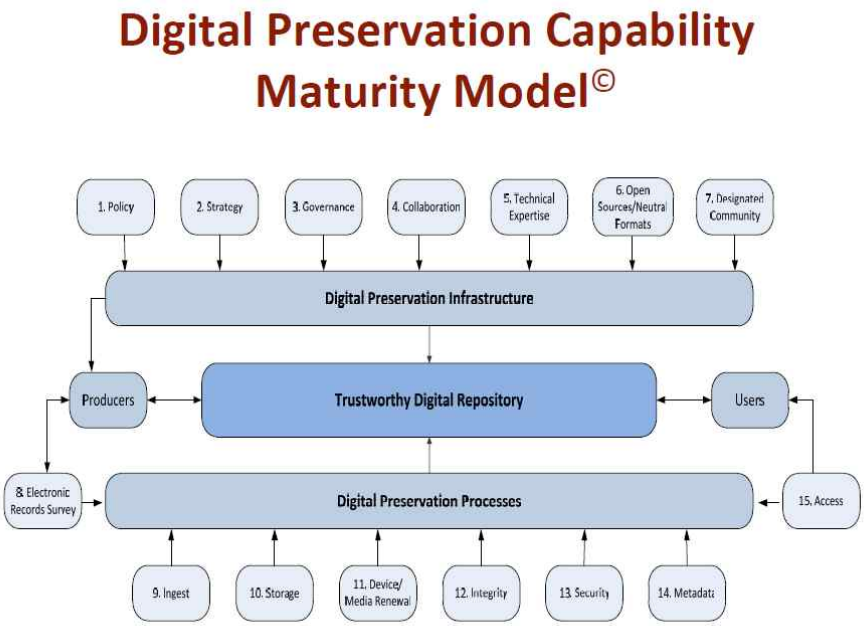
## 5. 미국 주정부 디지털기록보존연합 컨소시엄의 기존 전략 평가

○ 미국의 다수 주정부의 입법 디지털기록 장기보존을 위한 해결책을 찾기 위한 미네소타역사학회 프로젝트 The Minnesota Historical Society project (2008-2012) ***A Model Technological and Social Architecture for the Preservation of State Government Digital Information*** 를 수행. 프로젝트 연구결과 ***Preserving State Government Digital Information: NDIIPP Final Report,*** February 29, 2012 참조 (참여 주정부: Illinois, Kansas, Mississippi, Tennessee, and Vermont, Arkansas, Nebraska, North Dakota. 참여 단체 및 연구기관 California Digital Library, the National Conference of State Legislatures, the California Legislative Counsel)

○ Kansas Enterprise Electronic Preservation (KEEP) 시스템 프로젝트, Merritt 상업 애플리케이션, Tessella의 Safety Deposit Box (SDB) 등을 기술적으로 검토.

○ 찰즈 달러와 로리 애쉴리가 개발한 <크로스탭 디지털 보존 능력 성숙도 모델> (Cross Tabbed Digital Preservation Capability Maturity Model (DPCMM))을 사용하여 디지털기록 보존능력을 평가할 것을 권장.

○ <크로스탭 디지털 보존 능력 성숙도 모델>은 최선의 실무 표준과 기록관리 원칙, 정보 거버넌스에 기반하여 프로세스와 수행 성과 프레임워크을 제공. 15개의 고차원 요건을 제시함.




<그림 14> 크로스탭 디지털 보존 능력 성숙도 모델

# 총괄 연구과제 요약

| 과제 고유번호 | 자동부여 | | 공개가능여부 | |
|---|---|---|---|---|
| 사업명 | 2013년 기록보존기술 연구개발 사업 | | | |
| 과제명 | 행정기관 전자기록물 재현기술 연구 및 프로토타입 개발 | | | |
| 연구책임자 | 성 명 | 임 진 희 | | |
| | 소속 기관명 | 명지대학교 기록정보과학 전문대학원 | | |
| | 전자우편 | ****** | 전화번호 | ****** |

○ 연구목표 (400 ~600자)

국가기록원의 '2013년 기록보존기술 연구개발 사업'의 과제인 '행정기관 전자기록물 재현기술 연구 및 프로토타입 개발'은 행정기관에서 생산되는 다양한 유형의 전자기록물에 대한 장기보존 전략인 에뮬레이션 전략을 중심으로 한 전자기록 장기보존 방법론을 모색하고 이를 기반으로 한국형 에뮬레이션 프로토타입의 개발을 수행하였다.

국내 전자기록관리 환경에 적합한 에뮬레이션 전략의 수립에 앞서 전자기록물 에뮬레이션 국내외 연구동향을 분석하고, 특정 행정기관을 대상으로 행정기관의 전자기록물의 유형별 보존대상을 정의하고 현행 보존전략 수행 시 소실되는 디지털 컴포넌트 렌더링 객체의 유형을 정의하고 기능성을 보존할 수 있는 방안을 제시하여 이를 기반으로 한국형 에뮬레이터의 프로토타입을 설계하는 것을 목표로 본 연구과제를 수행하였다.

○ 연구내용 (1000~1200자)

국가기록원의 디지털 컴포넌트에 대한 에뮬레이션 전략을 수립하기 위해 해외 에뮬레이션 사례를 분석한 후 방송통신위원회를 대상으로 기록관리시스템(RMS)에 등록된 전자기록물의 디지털 컴포넌트 포맷 유형 및 버전에 대한 조사 분석을 실시하였다. 이 결과를 바탕으로 문서보존포맷(PDF/A)으로 변환 시 소실되는 렌더링 기능성 보존을 위한 다양한 에뮬레이션 방법에 대한 연구를 통해 한국형 에뮬레이션 전략을 제시하였다.

먼저 해외 에뮬레이션 프로젝트인 CAMiLEON, KB, Planets, KEEP 프로젝트의 연구 결과와 시스템을 분석하였는데 CAMiLEON과 KB 프로젝트는 에뮬레이션 방법론 연구 중심의 프로젝트이며 Planets과 KEEP는 에뮬레이터 개발을 통해 에뮬레이션 서

비스를 목적으로 개발되었다. 본 연구에서는 가장 최근에 개발된 KEEP-EF (에뮬레이션 프레임워크)를 구성하는 에뮬레이터(7종)에 대한 구성 방식을 검토하였다.

다음으로 공공기관의 전자기록물에 포함된 디지털 컴포넌트 포맷 유형 및 버전에 대한 조사를 위하여 방송통신위원회의 기록관리시스템(RMS)를 대상으로 모든 디지털 컴포넌트에 대한 포맷 및 버전 추출 작업을 수행하여 포맷 유형 및 버전 정보를 수집하였는데 한/글과 엑셀 포맷이 약 94% 차지함을 확인하였으며 전체 디지털 컴포넌트의 99%가 7종(HWP, XLS, PDF, PPT, DOC, TIF, JPG)의 포맷으로 분포됨을 확인하였다.

이러한 조사 결과를 토대로 비표준 포맷 문서인 한/글, 엑셀, 파워포인트 애플리케이션에서 문서보존포맷(PDF/A)으로 변환할 때 소실되는 렌더링 기능성을 조사하여 약 16개의 기능성을 파악하였고 애플리케이션 제조사에서 제공하는 포맷 기술문서를 바탕으로 각 기능성의 속성 정보를 분석하여 향후 Characterization 분석에 활용할 수 있도록 렌더링 속성에 대한 XML Schema를 정의하였다.

또한 다양한 에뮬레이터 개발 방식에 대한 조사 분석을 통해 운영체제 에뮬레이션 및 애플리케이션 에뮬레이션에 대한 구체적인 방법을 연구하였고 이를 기반으로 한국형 에뮬레이션 전략을 제시하였다.

마지막으로 제시한 에뮬레이션 전략을 검증하기 위하여 VM 기반의 운영체제 에뮬레이터와 렌더링 기능성에 대한 조회가 가능하도록 뷰어 형식의 애플리케이션 에뮬레이터 프로토타입을 개발하였으며 프로토타입 에뮬레이터를 통해 다양한 디지털 컴포넌트에 대한 에뮬레이션의 가능성을 확인하였다.

○ 연구성과(응용분야 및 활용범위포함) (400 ~600자)

본 연구를 통해 제시된 렌더링 기능성에 대한 XML Schema와 일반 속성 정보 보존을 위한 XML Schema를 기반으로 국가기록원 기술정보은행(DFR)의 XML Schema를 개선하여 디지털 컴포넌트 보존 및 Characterization 연구 기반을 제공하였다.

또한, 한국형 에뮬레이션 전략 방안으로 제시된 2가지 방안을 기반으로 개발된 프로토타입 시스템에 대한 심도 있는 연구를 통해 중앙영구기록물관리시스템(CAMS)에 등록된 장기 보존(30면, 반영구, 영구) 대상 디지털 컴포넌트에 대한 국가기록원의 체계적인 에뮬레이션 전략 수립에 활용할 수 있다.

또한 본 연구에서 개발한 VM 기반의 운영체제 에뮬레이터 (디지털 컴포넌트 Configurator)는 Oracle사에서 개발하여 제공하는 VirtualBox를 기반으로 개발하였는데 VirtualBox는 현재 소스 프로그램까지 제공하므로 향후 국가기록원의 에뮬레이터 개발 기반으로 활용할 수 있다.

○ 총괄 참여연구원

| 성 명 | 소속/직위 | 성 명 | 소속/직위 |
|---|---|---|---|
| 임진희 | 명지대학교 기록과학전문대학원 | 최주호 | 이산티에스 |
| 김명훈 | 한국외국어대학교 대학원 | 이재영 | 이산티에스 |
| 오명진 | 명지대학교<br>디지털아카이빙연구소 | 문종배 | 이산티에스 |
| 황진현 | 명지대학교<br>디지털아카이빙연구소 | 김양자 | 이산티에스 |
| 이재홍 | 명지대학교<br>디지털아카이빙연구소 | 안대진 | 명지대학교<br>디지털 아카이빙 연구소 |


| | | |
|---|---|---|
| ***Keywords (5개 내외)*** | ***한글*** | 전자기록 장기보존 전략, 에뮬레이션 프레임워크, 디지털 컴포넌트 포맷, DFR, 에뮬레이터 프로토타입 |
| | ***영문*** | Electronic records Long-term Preservation Strategies, Emulation Framework, Digital Component Format, DFR(Digital Format Registry), Emulator Prototype |

## 주 의

1. 이 보고서는 국가기록원에서 시행한 용역연구개발사업의 연구결과 보고서입니다.

2. 이 보고서 내용을 발표할 때에는 반드시 국가기록원에서 시행한 용역 연구개발사업의 연구결과임을 밝혀야 합니다.

3. 국가과학기술 기밀유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 아니 됩니다.