

## MARC 포맷의 문제점



### 심 경

정보학박사  
한국도서관협회 평생회원  
(주)아이리스넷 대표  
shim@irisnet.co.kr

필자가 MARC를 처음 “만난” 곳은 미국의 첫 직장에서였다. 필자의 업무를 설명하는 이가 뭐라고 영어로 열심히 설명을 하는데, 긴장했던 탓일까? 내 귀에는 그냥 미국 말 중 무슨 “마크”라는 단어가 자주 들렸을 뿐이었다. 그런데 미국 가기 전 지인들로부터 “미국 가서는 몰라도 아는 척을 해야 한다”, “너처럼 입다물고 있으면 바보 취급 받는다”는 등 “참여”에 대한 정신교육을 워낙 많이 받았던 지라, 평소 말수가 적었던 필자가 용기를 내어 분위기도 바꿀 겸 현실참여(?)를 하기로 결심하였다. 그래서 다소 과장되고 명랑한 톤으로 한 질문이 “Who is he, I mean Mark?”였다. 순간 분위기는 요즘 말로 “짜~”해졌고, 그 이후 필자는 다시 과묵한, 아주 과묵한 미국생활을 하였다.

내게는 이렇게 특이한 인연을 가진 MARC 포맷에 대하여 퇴출 이야기가 십여 년 전부터 심심치 않게 들렸다. MARC 포맷의 나이가 오십이 넘었으니 명예퇴직 압력을 받는 것이 이상할 것도 없을지 모른다. 하지만 오래되었다고 버린다면 그건 고려장(高麗葬)일 것이다. 우리가 MARC를 포기할 때 포기 하더라도 일단 그 문제가 무엇인지 알아볼 필요는 있지 않겠는가? 그래서 이번 글에서는 정확히 MARC의 문제가 무엇인가를 살펴본다. 특히 MARC 포맷에 대한 반대론자들 또는 개선론자들의 주장을 살펴보고, MARC의 장래가 어떻게 될지를 생각해 보려고 한다.

MARC 포맷에 대한 반란(?)은 크게 두 가지 형태가 있다. 첫 번째는 도서관 분야에서 독점적 위치를 가져왔던 MARC 포맷이 모든 경우의 다양한 자료형태를 수용하기에는 충분하지 못하므로, 각 분야의 요구에 적합한 새로운 메타데이터 포맷의 개발이 필요하다는 움직임이며, 둘째는 MARC 포맷의 기반기술은 이제 시대에 뒤떨어졌으며, XML을 적용해야 한다는 움직임

이다. 그런데 MARC포맷에 대한 변경, 대치 또는 발전에 대한 의견은 이처럼 두 방향으로 나뉘지만, 왜 MARC 포맷의 대치가 필요한가에 대한 그들의 논리 전개 방식과 이유는 흡사하다. 이들은 먼저 MARC 포맷의 한계점을 지적하였으며, 그들이 지적하는 MARC포맷의 단점 또는 문제점에는 서로 큰 차이가 없다. 이러한 궤적은 그 동안 MARC 포맷이 유지해 온 독보적 위치에 대한 “도전”을 하는 입장에서 다른 포맷으로의 전이 또는 새로운 기술채택에 대한 “대의명분”이 필요했을 것이고, 이들이 소수 저자의 주장을 상호 인용하여 발생한 것으로 판단된다.

### 신규 메타데이터 포맷의 생성

메타데이터란 용어의 의미는 광범위하나 우리 분야에서는 일반적으로 정보자원에 대한 구조화된 데이터를 의미하는 것으로 이해된다. 따라서 MARC 포맷도 물론 메타데이터 포맷의 한 종류지만, 잠정적으로 이 글에서 메타데이터란 1995년 초기 더블린 코어가 발표되면서부터 문헌정보학 분야에서 활발히 사용된 MARC 이외의 포맷을 지칭하는 의미로 사용한다.

과거 절대적이라 할 수 있는 MARC의 위상에 정면 도전을 선언한 메타데이터 포맷은 더블린 코어이다. 더블린 코어는 1995년 3월 미국 오하이오주 더블린에서 열린 1차 메타데이터 워크숍에 모인 각 분야의 전문가들이 네트워크 상의 전자정보객체를 위한 자원기술(資源記述) 레코드(즉 메타데이터) 개발의 필요성을 확인하고, 일련의 메타데이터 요소를 결정한 것이다. 하지만, 더블린 코어는 엄밀히 말하여 MARC 포맷을 대치하기 위한 목적을 가진 것은 아니었다. 이 회의에서도 더블린 코어는 “다른 자원기술을 대신(supplant)하도록 의도된 것이 아니라 그들을 보완(complement)하기 위한 것” 이라고 명확히 천명하고 있다.

더블린 코어의 생성배경은 급격히 증가한 웹자원에 대하여 웹검색 엔진에 의한 자동생성 색인 레코드는 너무 적은 정보만을 가지고 있어 유용성이 낮은 반면 정보 전문가가 수작업으로 생성하는 MARC 레코드는 인터넷 상에 대량으로 존재하는 전자문헌을 대상으로 생성하고 유지하기에는 너무 비싸다는 데 있었다. 말하자면 이와 같은 두 극단적 사례를 완화하는 방법으로 자동색인보다는 좀더 정보가 많고 정식 목록의 MARC레코드보다는 덜 완벽한 레코드를 수작업으로 생성하고자 더블린 코어를 개발한 것이다. 따라서 더블린 코어의 개발 목적은 최소의 인간 노력으로 보다 많은 대상을 기술할 수 있도록, 그리고 웹상 자원의 저자 자신이 메타데이터 기술을 생성할 수 있도록 단순한 포맷을 제공하려는 것이었다.

우연인지 모르지만, 이를 전후하여 각 분야에서 다양한 메타데이터 포맷이 개발되었다. 예를 들면, 예술작품을 위한 메타데이터 포맷인 CDWA(Categories for the Description of Works of Arts)가 1996년에 발표되

1) Weibel, S. & et al. (1995). OCLC/NCSA Metadata Workshop Report. Retrieved 2003. 5. 5 from <http://dublincore.org/workshops/dc1/report.shtml>

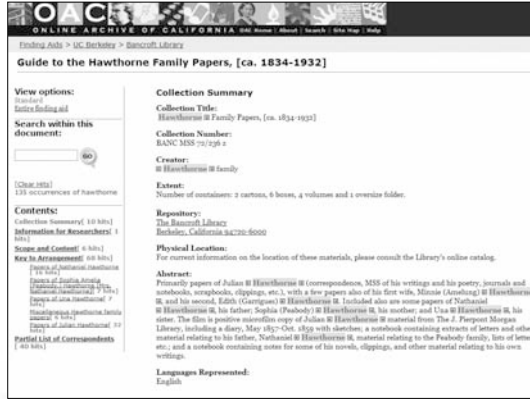
었고, 고문서 자료를 위한 EAD(Encoded Archival Description) 버전 1이 1998년에 공개되었으며, 지리공간자료를 위한 미연방 지리데이터 위원회(FGDC) 메타데이터인 CSDGM(Content Standard for Digital Geospatial Metadata)은 1994년 최초로 채택되고 버전 2가 1998년에 발표되었다. 물론 이들이 더블린 코어를 시작으로 활성화되었겠지만, 생성시기의 근접성으로 미루어 아마도 1990년대 중반 이전부터 MARC 포맷의 한계점에 대한 인식이 어느 정도 도서관계에 확산되었던 것으로 추정된다.

## 신규 메타데이터 포맷의 필요성

새로운 메타데이터 포맷을 개발한 측은 그 당위성을 기존 MARC 포맷이 자신들이 필요로 하는 사항을 만족시키지 못하여 “어쩔 수 없이” 물리적, 논리적으로 새로운 메타데이터 포맷을 구성하게 되었다고 주장한다. 예를 들면, EAD는 기존 MARC 포맷이 고문서 요소기술을 위한 요구사항을 충족시킬 수 없음을 확인하고 개발된 새로운 포맷이다. EAD 개발자들은 새 포맷의 개발 이유로 MARC 레코드의 최대길이, MARC 포맷의 평면적 특성이 자신들의 계층적 기술구조에 적용되기 어려움, 그리고 비용문제 등 세 가지를 들고 있다<sup>2)</sup>. 이들을 좀 더 상세히 설명하면 첫 번째, MARC 레코드의 최대길이 문제는 ‘Finding Aid’라 불리는 고문서 레코드(archive record)는 서지레코드처럼 단일 항목 또는 자료에 대한 레코드를 제공하는 것이 아니라, [그림 1]과 같이 개별자료의 집서(collection)에 대한 레코드를 제공하므로 레코드 길이가 MARC 레코드의 최대길이를 넘는 경우가 많다는 것이다<sup>3)</sup>.

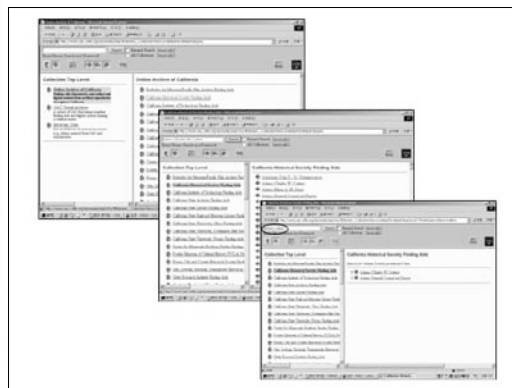
2) Dooley, J. M. (Ed.). (1998). Encoded archival description: context, theory, and case studies. Chicago: Society of American Archivists.

3) 아카이브 자료는 예를 들면, 한 집안이나 개인에 관련된 일기, 편지, 사진 등 다양한 자료를 집서단위(collection)로 보존하고 기술한다. 따라서 “MARC 레코드는 최대 100,000[sic.]문자로 제한 되어있다. 이는 ASCII코드의 문자크기 10으로 하였을 때 가로 8 1/2, 세로 11인치 페이지 대략 30페이지 분량이다. 대다수의 Finding Aids는 이보다 커서 레코드 크기의 제한은 넘지 못할 장애였다” (Pitt, D. V. 1998. Encoded archival description: the development of an encoding standard for archival finding aids. In J.M. Dooley (Ed.), Encoded archival description: context, theory, and case studies. Chicago: Society of American Archivists, p. 14).



[그림 1] Hawthorne가(家) 문서의 Finding Aid 레코드

둘째, 아카이브 자료는 [그림 2]에 보인 것처럼 집서단위로 수집, 정리하므로 하위의 개별자료와 자연스럽게 계층관계를 이루게 되는데, 이 계층관계가 흔히 데이터베이스 모델에서 말하는 부자관계(parent-child relationship)의 2단계를 넘게 된다<sup>4)</sup>. 그러나 MARC 포맷은 이와 같이 2단계가 넘는 계층관계를 표현하기에 적합하지 않으며, 이것이 곧 앞서 언급된 MARC의 ‘평면적 특성’을 의미한다. 이에 관하여는 결론 부분에서 부연하여 언급을 한다. 마지막으로 비용의 문제는 MARC포맷은 이를 기반으로 하는 도서관 자동화시스템이 필요하고, 그들 시스템이 비용을 유발한다는 것이다. 반면, EAD는 1995년에 SGML을 기반으로 시작하여 1998년에 XML 기반의 메타데이터 표준 버전 1.0으로 발표되었으며, SGML이나 XML 모두가 운영체제나 워크스테이션에 상관없이 적용되는 오픈 툴이므로 MARC처럼 고비용의 시스템이 필요 없다는 의미이다.



[그림 2] Finding Aids의 계층관계

4) 계층관계는 [그림 2]의 ②의 'Adams (Charles W.) Letters' 는 California Finding Aids의 한 레코드이고, 그 아래 5건의 자료를 포함하고 있음을 알 수 있다.

또한 미연방 지리데이터 위원회 메타데이터인 CSDGM도 MARC 포맷이 아닌 새로운 포맷이었다. 하지만 그들이 새로운 메타데이터 포맷을 생성한 이유가 MARC의 한계성 때문인지는 확실하지 않다. 다만, MARC포맷의 물리적 제한점을 설명하기 위하여 부연하자면, 1998년 당시 CSDGM을 사용한 정보 시스템이 50% 이상의 검색문에 대하여 검색결과 0건을 기록한 것을 발견한 후, FGDC에서 취한 조치가 MARC 포맷의 물리적 한계를 보여준다. 짧게 설명하여 FGDC는 해결방안으로 CSDGM을 보다 폭넓게 사용되는 메타데이터 포맷인 MARC21과 더블린 코어로 변환을 시도하였다<sup>5)</sup>. 이 CSDGM 데이터를 MARC21 포맷으로 변환하는 과정에서 호환성의 문제가 발생하였는데, 그 문제는 MARC포맷이 정의한 레코드/필드 최대길이와 관련된 것으로 MARC21에서 정의한 길이가 CSDGM 데이터를 수용하기에는 너무 작다는 것이었다. 즉, 도서관자동화시스템에 따라 최대허용 길이가 다소 다르지만, MARC21포맷은 99,999 바이트가 이론적 레코드 최대길이이며, OCLC의 WorldCat이 허용하는 레코드 최대길이는 4,096바이트이고 가변장 필드의 최대길이는 1,230 바이트이다. 또한 최고 50개까지의 가변장 필드를 허용한다. 이 변환프로젝트의 결과를 요약하면 [표 1]과 같다.

[표 1] FGDC 메타데이터 특성 통계

레코드 특징	레코드 길이	레코드 내 최대필드길이
평균	6,792 바이트	2,125 바이트
메디안	6,474 바이트	1,258 바이트
대상 466레코드 중 최대 레코드	28,042 바이트	9,525 바이트
4096바이트 보다 큰 레코드 수	343 레코드	
1230보다 긴 필드를 가진 레코드 수		236 레코드

[표 1]에 나타난 수치는 FGDC 메타데이터를 WorldCat시스템으로 변환에 무리가 있음을 보여주며, 실제 74%의 FGDC 레코드가 MARC의 레코드 최대길이를 초과하고, 51%의 레코드가 최소 하나 이상의 필드 최대길이를 초과하는 필드를 가지고 있었으며, FGDC 메타데이터는 300종이 넘는 메타데이터 요소를 가지고 있어 이들을 모두 MARC포맷으로 변환할 수 없었다.

이처럼 1990년 중반부터 출현한 메타데이터 포맷들은 주로 MARC 포맷의 물리적 구조적 한계를 문제점으로 지적하였고 실제 문제가 있음은 위 예시에서도 보았다. 이와 시기를 같이 하여 주장된 XML로의

5) Chandler, A., Foley, D. and Hafez, A.M. (2000). Mapping and converting essential Federal Geographic Data Committee (FGDC) metadata into MARC21 and Dublin Core: towards an alternative to the FGDC clearinghouse, D-Library Magazine, 6(1). Retrieved 2009. 4. 13 from <http://www.dlib.org/dlib/january00/chandler/01chandler.html>

변화는 보다 공격적이었다.

## MARC 보다는 XML

새로운 메타데이터 포맷의 생성은 “MARC를 사용하려다 보니 불편해서...”라는 다소 제한적 움직임이 있다면, MARC를 XML로 선택 또는 대체해야 한다는 주장은 다소 과격할 느낌까지 준다. XML채택 주장의 효시는 미국 스탠포드 대학교 레인의학도서관 Dick R. Miller<sup>6)</sup>가 2000년에 발표한 “XML: libraries’ strategic opportunity<sup>7)</sup>”이다. 이 글과 그가 발표한 일련의 프레젠테이션에서 시작된 XML로의 변이 또는 대체 주장이 이어졌고, 2년 뒤 OCLC의 Roy Tennant<sup>8)</sup>가 Library Journal의 컬럼에 실은 “MARC는 죽어야 한다(MARC must die)<sup>9)</sup>”는 글의 제목은 마치 영화제목과도 흡사하여 이 논쟁이 다소 과격한 양상을 띠는 것 아닌가 하는 느낌마저 든다. 물론 이 두 글 사이의 이년 동안 많은 글들이 발표되었고 토론도 이어졌다. 이러한 내용을 종합하여 이들의 주장을 살펴보자.

## MARC포맷의 문제가 뭐길래?

Miller의 글은 XML의 장점이 무엇이고 어떻게 도서관 환경에 적용될 수 있는가를 주로 다루었으나, 그의 프레젠테이션에 주로 MARC 포맷의 문제점이 언급되었다. 사실 MARC의 문제는 MARC 구문(syntax), MARC 데이터 요소, 그리고 목록규칙이 워낙 긴밀히 엮여있어 이들 중 하나를 분리하여 지적하기가 매우 어렵다. 이러한 사실을 염두에 두고 XML채택을 주장하는 이들이 꼽는 MARC 포맷의 문제점을 살펴보면 다음과 같다.

첫째, MARC에는 다양한 정보가 레코드 내 다양한 위치에 기록되어있는데 이들은 종종 일관성도 상실하고 있다. 예를 들어, 날짜는 MARC21에 005(최종수정일자 또는 최종처리일시), 008/00-05(입력일자), 008/07-10(발행년 1), 008/11-14(발행년 2)와 260 \$c(발행년, 배포년 등) 등에 4개의 다른 포맷으로 나타난다.

6) Dick R. Miller는 U. of Oklahoma에서 문헌정보학 석사를 취득하였으며, Stanford University Medical Center, Lane Medical Library의 도서관자동화 사서(systems librarian) 겸 수서 및 목록부서 책임자(head of technical services)이다.

7) Miller, D. R. (2000). XML: libraries’ strategic opportunity. Library Journal Net Connect, Summer. Retrieved 2009, 4, 22 from <http://elane.stanford.edu/laneauth/LJ/>

8) Roy Tennant는 콜롬비아주립대에서 지리학과 전산학 학사, UC 버클리에에서 문헌정보학 석사를 취득하고 동 대학도서관에서 각종 도서관자동화관련 업무를 수행하였으며, 현재 OCLC의 OCLC Programs and Research중 RLG Programs의 책임프로그래머이다. 위 글을 발표할 당시는 UC 버클리에 근무하고 있었다.

9) Tennant, R. (2002). MARC must die. Library Journal. Retrieved 2009, 4, 5 from <http://www.libraryjournal.com/index.asp?layout=article&articleid=CA250046&publication=libraryjournal>

둘째, MARC 포맷 전체에 걸쳐 일관성 없이 다루어지는 선천적 복잡성이 존재한다. 예를 들어, 고정장 필드 형태의 정보가 전거레코드의 서브필드인 4XX, 5XX와 7XX의 \$w(제어서브필드)와 서지레코드의 서브필드인 76X-78X의 \$w(시스템 제어번호)에 삽입된다.

셋째, MARC 레코드에 어떤 정보는 코드화된 형태로 기록되는데 이는 저장매체가 아주 싼 요즘 전산 환경에서는 정당화될 수 없다는 것이다. 불필요하게 코드화된 형태로 저장되는 정보의 예는 정부간행물(008), 언어(008, 041), 발행국(008, 041)와 역할어(\$4) 등이다.

넷째, MARC포맷은 계층구조를 제한적으로만 지원하는 평면적 구조이며, 또한 웹 링크를 원만히 지원할 수 없다고 한다<sup>10)</sup>.

그 밖에도 MARC포맷의 내용을 채우는 목록규칙에 대한 불편함도 Miller 측에서는 제시하였다. 우선 목록규칙인 AACR2에서 접근점(access points)에 대한 일관성은 키워드 검색으로 극복할 수 있으므로 이들에 대한 일관성에 시간을 낭비할 필요가 없으며, 표목(headings)은 하나 이상의 언어로 표현될 필요가 없으며<sup>11)</sup>, 더구나 표목의 레코드 내 순서는 무의미하다는 것이다. 이도 역시 발전된 전산환경에서 키워드 검색으로 얼마든지 해결될 수 있다고 한다.

솔직히 필자에게는 MARC를 XML로 대체해야 한다는 주장이 위와 같은 이유 때문이라면 크게 설득력 있게 들리지는 않는다. 왜냐하면 위 내용 중 대다수는 현재 우리가 XML포맷을 사용하고 있다 해도 일어날 수 있는 것들이다. 다시 말하여, MARC포맷의 선천적인 문제라기 보다는 아래 기술한 발전과 적응과정에서 발생한 문제들인 것이다. 또한 MARC는 그 유용성이 지나도록 장수했으며, 따라서 'MARC은 죽어야 한다'고 주장한 Tennant의 글을 보아도 별로 새로운 내용은 없다. 따라서 이러한 주장을 'MARC 포맷의 문제점 때문에 우리가 XML로 가야 한다'라는 당위성으로 받아들이기 보다는 XML로 변이하는 김에 MARC의 한계점도 수정하자는 의견으로 받아들이는 것이 더 의미 있어 보인다.

## MARC포맷은 왜 이런 문제가 있을까?

그럼 우리는 이렇게 버그(?)가 많은 포맷을 어렵게 배워서 '하늘처럼 받들어' 쓰고 있었던 것일까? 그런 것은 아니다. MARC는 이전 글에서 언급한 바와 같이 초기에는 목록카드 생성을 위하여 고안된 것이었고, 1980년대 초기에 온라인 목록에 적용되기 시작하였다. 이 과정에서 도서관자동화시스템의 발전과

10) 이 내용은 Miller의 프레젠테이션을 존슨이 요약한 것을 옮긴 것이다 (Johnson, B. C. (2001). XML and MARC: which is "right"? Cataloging and Classification Quarterly, 32(1), pp. 81-90)

11) 이 부분은 지난 해 7월호의 '서지레코드의 기능상의 요건'에서 설명한 저작-표현형-구현형-개별자료의 관계로 쉽게 해결될 수 있으며, 이 관계는 XML을 활용하여 계층구조를 쉽게 표현이 가능하다.

도서관에 도입되는 모든 자료형태 목록에 대한 경험이 쌓이면서 유기적으로 수정 및 성장을 해 왔다. 이와 같이 애초에 예상하지 못한 문제에 대한 보완을 거듭하다 보니 MARC포맷에 동일한 형태의 데이터가 다른 필드에 포함되는 일관성 결여현상이 발생하였고, 많은 노력에도 불구하고 특정 데이터들은 유용한 방식으로 코드화되는데 실패한 점이 있는 것은 사실이다<sup>12)</sup>. 그리고 MARC의 단점에 대한 논의가 나올 때마다 매번 지적되는 MARC포맷의 레코드 및 필드 길이 문제는 MARC가 출현할 당시는 저장매체가 워낙 고가였기 때문에 너무나 당연한 선택이었으며, 아주 영리한 구조였다. 이제 저장매체의 가격이 낮아졌다고 MARC를 비난할 수는 없다.

### XML로 MARC를 대체하면 뭐가 달라지나?

이를 논하기 전에 우선 명확히 할 일이 있다. 앞서 지적한 것과 같이 MARC라는 포맷이 XML이라는 메타언어로 툴을 바꾼다고 해서 MARC 레코드의 내용과 목록규칙이 자동으로 바뀌는 것은 아니다. 이들은 XML로 MARC를 대체할 때 추가 노력을 통하여 발생할 “수”도 있는 일이지, 포맷의 대체 그 자체가 이들을 자동으로 바꾸어주지는 못한다. 또한 MARC를 XML로 변환하려면 과거 DTD<sup>13)</sup>라 불리던 데이터 포맷을 정의하는 공통구조(common structure)가 만들어져야 하는데, 이는 Miller 자신도 “상당한 업무(considerable task)”가 될 것이라고 인정하였다.

그럼에도 불구하고 이들 XML 옹호론자들이 역설하는 XML의 장점은 오픈 인터넷 표준으로서 웹상에서 정보전달에 최적화되어 있으며, XML을 준수하여 인코딩된 문서나 레코드는 논리적<sup>14)</sup>, 그리고 물리적 구조<sup>15)</sup>를 동시에 가진다는 등의 기본적인 장점만을 주로 포함하고 있다. 그 중 도서관에 적용하였을 때의 장점만을 추려보면 다음과 같다.

- 유니코드를 지원하고 플랫폼 중립성으로 데이터 수명이 길어질 수 있다.
- MARC처럼 필드길이 또는 레코드 길이의 제한이 없다.
- XML의 계층구조를 쉽게 지원하는 특성이 향후 IFLA의 “서지레코드의 기능상의 요건(FRBR)”과 같은 구조에 적합하다.

12) Fiander, D.J. (2001). Applying XML to the bibliographic description. *Cataloging & Classification Quarterly*, 33(2), 17-28.

13) DTD는 Document Type Definition의 두문자어로서 SGML 용어이다. 이는 문서구조를 정의한 것이며, DTD는 SGML과 호환성 측면에서 장점이 있지만 XML 문법을 따르지 않으므로, XML에서는 XML Schema라고 하는 새로운 문서구조 정의 기술을 사용한다.

14) 논리적으로는 명명된 요소(named elements)와 하위요소(nested elements)의 계층관계를 가진다. 이때 요소가 데이터베이스의 필드라면, 하위요소는 서브필드에 해당된다.


15) 물리적으로 문서의 부분을 엔티티로 명명하여 별도로 저장하고 재사용이 가능하다. 이 엔티티는 그 이름이 프로그램 코드에서 상수와 아주 흡사하다.

이렇게 보면 XML이 MARC에 비하여 크게 장점이 있는 것처럼 보이지만은 않는다. 특히 MARC레코드를 총체적으로 XML기반으로 변환하였을 때 소요될 경비를 감안하면 더욱 그러하다<sup>16)</sup>. MARC포맷이 융통성이 없고, 일관성이 부분적으로 결여된 것은 이미 알고 있는 사실이며, 현재 그러한 MARC포맷이 작동하지 않느냐 하면 그건 아니다.

## 결언

제한된 지면에 복잡한 MARC와 XML 문제를 이해하기 쉽고 명확하게 설명하기는 쉽지 않다. 그러나 우리가 이 문제를 논의할 때 잘못 인식하는 것은 이들 중 하나를 반드시 선택해야 하는가 라는 문제이다. 그럴 필요는 없다. XML 옹호론자들도 XML의 채택으로 월드 와이드 웹 시대에 좀 더 호환성있고 일반적인 오픈 포맷을 사용함으로써 MARC포맷이 폐물이 되도록 기다리지 말고, 또한 사서도 웹 시대에 정 보서비스의 주류에서 제외되지 않도록 변화를 추구하자는 것이다.

MARC포맷의 구조를 유지한 채 XML DTD 또는 스키마로 직접 변환하는 것은 그리 어려운 작업은 아니다. 정말 어려운 것은 XML 옹호자들이 지적한 MARC포맷과 목록규칙까지를 수정하면서 XML로 가는 작업이 시간이 소요되며 어려운 것이다. 우리가 가끔 듣는 미국의회도서관의 MARCXML<sup>17)</sup> 프로젝트도 전자(前者)의 접근방식을 취하고 있다. MARCXML 프로젝트에서의 변환원칙은 “MARC 레코드의 필수 데이터 모두를 XML로 변환하고 표현하는 것이다.” 이 원칙에 의하여 MARC의 데이터 필드길이, 그 시작위치와 같은 디렉토리 정보나 리더 데이터 위치는 XML에서 필요하지 않으므로 공백으로 유지된다. 따라서 변환된 레코드는 아무런 정보손실 없이 역방향으로 재변환이 가능하다.

이 방식은 현재 MARC포맷의 구조와 정보를 어떻게 재구성할 것인가에 대한 결정이 이루어지는 동안, 또 아직까지 MARC 기반 시스템이 주류를 이루는 동안 변이 단계로서 반드시 필요해 보이며, 이 변이 기간은 제법 길 것으로 예측된다. 사실 우리 주변에는 이미 XML로의 움직임이 관찰되고 있다. FRBR이 바로 그것이다. FRBR은 네 가지 요소의 관계가 계층적이며 반영하기 가장 좋은 틀은 XML이다. 또한 새로운 목록규칙인 RDA(Resource Description and Access)가 어떻게 변화할지는 미지수이다. 따라서 우리는 추이를 지켜보면서 XML에 대한 이해를 넓히고, XML 옹호자들이 지적한 MARC포맷과 목록규칙의 문제점을 공부하며 “그날”에 대비하면 충분할 것으로 판단된다. 

16) 이는 데이터 변환 비용 이외에 신규시스템개발 비용을 포함한다.

17) 이를 보려면 <http://www.loc.gov/standards/marcxml/>을 참조하면 된다.