지상(紙上) 블로그

MARC 포맷의 기본 개념 (MARC Format Basics)

심 경
정보학박사
한국도서관협회 평생회원
(주)아이리스닷넷 대표
shim@irisnet.co.kr

MARC은 도서관에서 이용자 서비스를 위한 가장 기초가 되는 작업이며, 도서관 간 상호협력의 시작점이라고 한다. 그런데 MARC가 뭐길래 그럴 수 있을까?

지금은 국내 거의 모든 도서관에서 MARC 포맷을 사용하고, 대학에서도 배울 뿐 아니라, 이에 관련된 책들도 적지 않게 나와있어, 다들 MARC을 잘 안다고 생각할 것이다. 그러나 MARC에 대한 인상과 경험은 사람마다 다 다를 수도 있다. 우선 필자처럼 1970~80년대에 대학을 다닌 사서들은 학교에서 MARC 포맷에 대하여 상세히 배울 기회가 없었다. 그보다는 도서관 현장에서 실무를 하면서 직접 배우고 익히는 것이 더 많았다. 그런 한편으로 신세대 사서들은 학교에서는 MARC에 대하여 배울 기회가 많았겠지만, 정작 최근의 도서관 현장에서는 일일이 시간 "낭비"하고 만들 필요 없이, 책과 더불어 서점이 납품하는 그 무엇이라는 인상을 가질 수도 있다

필자가 MARC 레코드를 처음 본 것은 1980년대 중반 미국에서 문헌정보학 대학원에 입학한 직후 돈도 벌고, 실무경력도 쌓겠다는 목표로 얻은 직장에서였다. 한국에서 몇 년 동안 대학도서관 사서로 일하다 공부를 하러 간 차였지만, 사실 MARC이란 걸 써본 적도 없었고, 그걸 사용하는 자동화시스템도 낯설긴 마찬가지였다. 다만 무엇이든 열심히 배우고 만져보아 언젠가는 내가 직접 이러한 시스템을 구축해보겠다는 꿈 하나를 품고 부딪혀 보았을 뿐이다. 아무튼 눈치로도 하고, 매뉴얼을 보면서 제법 많은 MARC 레코드를

구축했으면서도 필자 자신도 고정장의 용도가 무엇인지, 또한 고정장의 발행연도와 같이 레코드 기술(記述)부의 내용과도 중복되는 것이 왜 필요한지도 모르고 어느 정도는 기계적으로 입력하였다. 이를 시작으로 목록 관련 업무를 십여 년 하면서 그 용도를 추정할 수 있었지만, 사실 이에 대한 설명이 나온 자료를 접한 것은 극히 최근의 일이다[1]. 그러면서 국내에서 만난 많은 사서들이 MARC에 대한 이론적 지식과 현장에서의 적용 가치 사이에 적지 않은 틈을 보이는 것을 알게 되었다. 따라서 이 글에서는 MARC의 기원, 종류, 구조와 용도에 대해 한번 "복습"함으로써 그에 대한 이해를 넓혀보기로 했다.

MARC은 무엇인가?

MARC이라는 이름의 기원은 반세기 이전으로 거슬러 올라간다. 1950년대 말부터 내부업무에 자동화 기법을 도입하기 위한 연구를 착수했던 미국의회도서관이 1965년 12월에 도서관자원위원회(Council on Library Resources: CLR)로부터 지원을 받아 기계가독형 목록데이터의 배포에 대한 가능성과 효용성 실험을 위한 프로젝트를 수행하였다. 이 프로젝트의 이름이 바로 MAchine Readable Cataloging의 두문자인 MARC이었으며, 오늘날 MARC이라는 이름은 여기에서 기원하였다[2]. MARC구조는 국제표준인 ISO 2709를 구현한 것이며, 또한 우리에게 친숙한 Z39.50처럼 Z39.2라는 이름을 가진 ANSI/NISO 표준이기도 하다.

MARC은 하나의 메타데이터 포맷이다. 쉽게 말하여 서지레코드를 저장하는 틀이다. 그래서 우리는 MARC 레코드라고 뒤에 레코드라는 말을 붙여 흔히 사용한다. 왜냐하면 틀에 내용을 입력한 후에야 우리가 주로 사용하는 서지레코드가 되기 때문이다. 따라서 우리는 서지레코드를 구성하기 위하여 MARC라는 포맷과 그 포맷에 내용을 입력하는 방법을 지시하는 목록규칙이라는 두 가지 요소를 필요로 한다. 우리 말로 MARC은 기계가독형목록이라고 주로 표현되지만 이는 서지레코드라는 의미로서 목록보다는 기계 즉, 컴퓨터가 읽을 수 있는 형태로 서지레코드를 담을 수 있는 용기(container)로 이해하는 것이 정확하다.

---

1) Fritz, D. A. & Fritz, R. J. (2003). *MARC21 for everyone: a practical guide*. Chicago: American Library Association. 이전 어느 자료에서도 고정장이 무엇이라는 내용은 있어도 그에 대한 용도를 짧게라도 설명한 책을 필자는 보지 못하였다.

2) Averan, H. D. (1975). *MARC: its history and implications*. Washington, D.C.: Library of Congress.

그럼 서지레코드를 담는 용기로서 MARC은 왜 필요한 것일까? 그 대답은 MARC의 정의에서 찾을 수 있다. Furrie는 MARC은 "정보교환 활성화를 위한 업계표준(an industry-wide standard whose primary purpose is to foster communication of information)"이라 하였고[3], Cooper는 "기관 간 또는 시스템 간 서지 데이터 교환을 위한 표준 포맷(a standard format for interchanging bibliographic data between organizations or systems)"[4]이라고 정의하였다. 또한 MARC의 공식자료인 MARC21 Concise에는 "서지정보의 운반체(carrier)"라고 하였으며, "MARC 포맷은 기계가독형 서지 데이터의 표현(representation)과 교환(exchange)을 위한 표준"이라고 정의한다. 그러므로 MARC은 서지레코드 교환을 위한 표준포맷이지 그 자체가 서지레코드는 아닌 것이다. 이쯤 되면 우리도 서지레코드를 교환하기 위하여 왜 이런 노력이 필요한지를 쉽게 추측할 수 있다. MARC 포맷에 입력되는 서지레코드, 즉 MARC 레코드는 "서지레코드의 기능상의 요건(FRBR)"에 정의된 구현형(manifestation)에 대한 "진실된 정보"를 공유하기 위함이다. 도서관들이 동일한 자료에 대한 서지레코드를 중복구축하기 보다는 이 정보를 서로 공유함으로써 시간과 예산을 절감할 수 있다는 것이다. 그러면서도 MARC 포맷은 개별 소장도서관의 용도를 위한 태그를 마련하고 있다. 이들은 대개 9자를 포함한 것으로 59X[5], 9XX와 지금은 사용하지 않는 69X[6] 등을 들 수 있다. 엄밀히 말하여 이는 서지레코드 교환 시 제외대상이며, USMARC에는 "교환을 위하여 59X 필드에 적용되는 사용법은 데이터 교환을 하는 양방의 양해를 필요로 한다"[7]고 명시함으로써 이 필드가 특정 자료의 모든 구현형에 해당하는 정보라기 보다는 해당 도서관에 특화된 정보임을 주지하도록 하고 있다.

MARC의 종류는?

우리는 서지 데이터용 MARC 포맷만을 주로 사용하지만, 실제 MARC 포맷은 5가지 종류가 있다. 서지 데이터(MARC21 Format for Bibliographic Data), 소장 데이터(… Holdings Data), 전거 데이터(…Authority Data), 분류 데이터(… Classification Data), 커뮤니티 정보 데이터(… Community Information Data)가 그들이다.

---

3) Furrie, B. (2003). *Understanding MARC Bibliographic: Machine-Readable Cataloging*. Retrieved November 22, 2007, from http://www.loc.gov/marc/umb/

4) Cooper, M. D. (1996). *Design of library automation systems: file structures, data structures, and tools*. New York: Wiley. p. 286.

5) 1990년에 출간된 KORMARC에는 이 필드의 정의가 빠져있으며, 2000년에 나온 '고서용' 포맷과 2006년 '통합서지용' 에는 포함되어있음.

6) 이 필드는 1990년에 출간된 USMARC에 정의된 적이 있으나, MARC21이나 KORMARC에는 반영되지 않음.

7) Network Development and MARC Standards Office (1990). *USMARC Format for Bibliographic Data: including Guidelines for Content Designation*, v. 2. Washington, D.C.: Library of Congress.

서지 포맷이나 전거 포맷에 대하여는 대략 이해를 하고 있지만 소장 포맷은 약간 고개가 갸우뚱해지고, 분류 데이터 포맷 또는 커뮤니티 정보 데이터 포맷에 대하여는 무엇일까 하는 사람들이 많을 것이다. 이들을 간략히 살펴보면 다음과 같다.

● **소장 포맷** : 모든 형태 자료의 소장(holdings)과 소재(location) 데이터에 적절한 데이터 요소 인코딩을 위한 포맷 명세를 포함한다. 소장 데이터 포맷이 저장하는 정보는 FRBR에서 정의한 구현형에 대한 정보가 아니라 개별자료(item)에 대한 정보를 의미하며, 쉽게 말하면 우리가 기존 KORMARC에서 049 태그에 입력하던 정보가 저장되는 포맷이다[8]. 따라서 이 정보는 앞서 말한 MARC 레코드의 이념인 구현형에 대한 "진실된 정보"를 포함하도록 의도한 서지레코드에 포함될 수 없는 정보인 셈이다. 이 포맷은 서지레코드 내에 내장(embedded)되거나 독립적(separate)으로 사용되도록 정의되었으며, 그 내용이 서지레코드의 "MARC보기"처럼 이용자에게 보여지는 경우는 거의 없으며, 주로 소장 데이터를 반입/반출할 때 유용하다.

● **분류 데이터 포맷** : 분류 번호 및 그와 연관된 설명(captions)에 관련된 데이터 요소를 인코딩 하기 위한 포맷 명세를 포함하고 있다. 이 포맷은 주로 미의회도서관분류표와 듀이십진분류표에 관한 인코딩 방법을 기술하는데, MARC21 Classification의 개요에 의하면 "분류데이터는 OPAC검색시스템이나 도서관분류를 위한 컴퓨터를 이용한 분류와 같은 온라인시스템, 분류표의 관리와 개발을 위한 시스템, 서지레코드에 부여된 분류번호의 확인(validation)과 MARC 전거레코드로의 연계 등을 위하여 활용될 수 있다"고 그 용도를 밝히고 있다[9]. 이를 활용한 시스템 중 우리에게 잘 알려진 것은 OCLC가 서비스하는 웹기반 듀이십진분류표인 WebDewey를 들 수 있다.

● **커뮤니티 정보 데이터 포맷** : 이 포맷은 지난 호 글에서 코일(Coyle)이 MARC의 오용(誤用)이라며 비판한 것으로 행사, 프로그램, 서비스 등에 관한 정보를 포함하는 레코드를 위한 포맷에 대한 명세를 제공한다. 커뮤니티 정보의 예를 들면, 지역의 노숙자 쉼터나 연말정산도움기관 등을 기록하는 것으로 엄밀히 말하여 우리가 말하는 목록과는 연관이 없는 정보를 저장하기 위한 것이다. 하지만 이용자 서비스를 위한 데이터로서는 유용할 수 있다.

---

8) 049태그에는 연속간행물의 소장정보를 모두 기입할 수 없고, 이는 표준 MARC 태그가 아닌 OCLCMARC의 태그를 KORMARC이 차용한 것이다.

9) Library of Congress (2006). *MARC21 Format for Classification Data, Introduction*. Retrieved October 5, 2008, from http://www.loc.gov/marc/classification/cdintro.html

MARC의 구조

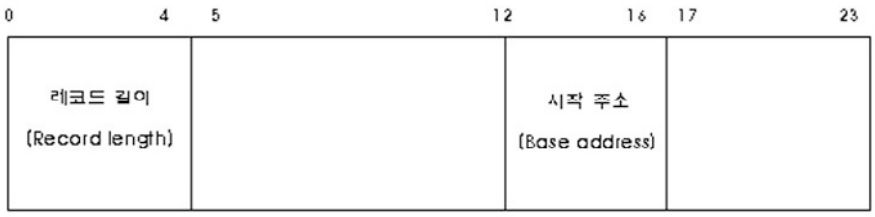
MARC이 가변장(variable-length, 可變長) 레코드라는 것쯤은 누구나 알고 있는 사실이다. 그럼 가변장 레코드라는 말은 무엇일까? 또한 레코드는 가변장인데, 레코드 안에 존재하는 필드는 고정장도 있고 가변장도 있다고 하니 단순하지만 가끔 혼돈을 초래한다. 먼저 가변장 레코드의 의미를 살펴보자.

MARC 포맷이 서지레코드를 가변장 레코드로 정의한 까닭은 서지레코드의 특성상 그 길이가 일정하지 않으므로 레코드 저장공간을 절약하기 위함이다. 왜냐하면 MARC 포맷이 처음 발명되었던 당시인 40여 년 전에는 컴퓨터의 저장장치는 아주 고가였기 때문이다. 사실 서지레코드의 길이가 일정하지 않은 것은 서지항목은 물론 개별항목의 길이도 레코드 마다 차이가 있기 때문이다. 예를 들면, 거의 모든 서지레코드는 저자명(1XX), 서명(2XX) 필드 등을 포함하지만 주기 필드(5XX)나 주제명 필드(6XX)를 반드시 포함하는 것이 아니므로 서지레코드의 길이가 일정하지 않을 것이라는 점은 짐작하기에 어려움이 없다. 또한 흔히 1XX필드에 저장되는 저자명의 경우, "Kennedy, John F. ≠q(John Fitzgerald)"와 "심경"은 명백하게 레코드 내에 차지하는 공간에 큰 차이를 가진다[10]. 흔히 데이터베이스 디자인 기초에서는 이와 같은 경우 80% 법칙을 적용하여 필드길이를 고정하지만, MARC 포맷은 공간의 낭비를 줄이기 위한 유연성을 가진 가변장의 형태를 택하였다. 이로써 최대한 데이터 손실을 초래하지 않는 한편 저장공간을 절약할 수 있는 구조를 제공하고 있는 것이다. 하지만 MARC 레코드가 무작정 길어질 수는 없다. 왜냐하면 가변장이긴 해도 레코드와 필드 길이의 최대 허용치를 규정하고 있기 때문이다. 설명이 점점 복잡해 지는 듯 하지만, 학창시절 시험을 위하여 무작정 외었을 법한 이 부분에 대하여 MARC레코드의 구조를 살펴보면서 한번 "이해"를 해 보자.

MARC에 관한 책을 보면 항상 등장하는 구조의 요소는 (1) 리더(Leader), (2) 디렉토리(Directory), (3) 제어필드(Control Fields)와 (4) 가변장필드(Variable Fields)이며, 마지막의 제어필드와 가변장필드는 가변장제어필드(Variable Control Fields)와 가변장 데이터필드(Variable Data Fields)가 원래 명칭이다(〈그림 1〉 참조). 흔히 리더는 레코드 자체에 대한 중요한 정보를 저장한다고 설명하는데, 도대체 중요한 정보가 무엇일까 하는 궁금증이 생긴다. 리더는 24자리를 차지하는 고정장이며, 〈그림 2〉와 같이 항상 동일한 위치에 자리잡는 두 요소가 있다. 0에서 4번째 자리까지(다섯 자리)의 문자위치에 자리한 "레코드 길이" 필드는 그 레코드의

10) 한글문자는 저장 시 2바이트를 차지하고, 영문은 1바이트를 차지한다는 사실을 감안해도 레코드 수가 대규모로 확장되면 이들 공간의 낭비는 크다.

〈그림 1〉 MARC 레코드 구조



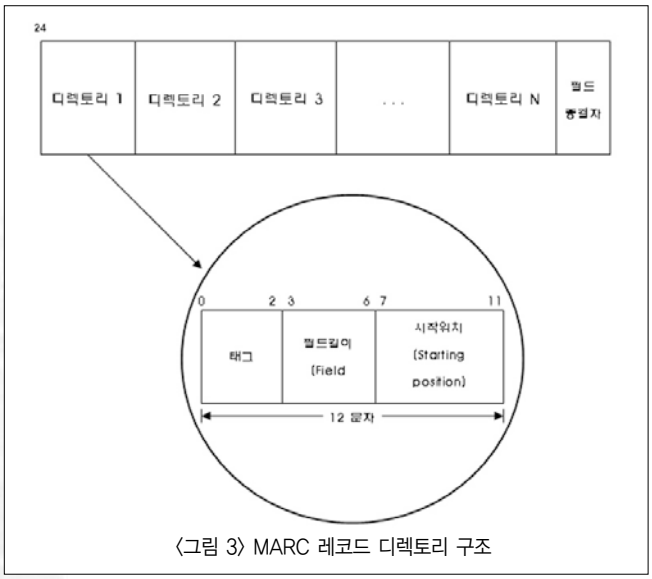
〈그림 2〉 MARC 레코드 리더 구조

길이를 저장하고, 12에서 16번째 자리(다섯 자리)에 있는 "시작주소"는 그 레코드에 출현한 최초 데이터필드 또는 제어필드의 시작위치를 저장한다. 그 밖에 빈 칸들은 우리가 서지레코드 입력화면에서 고정장 요소 중 "레코드 상태(리더/05 자리)", "레코드 형태(리더/06 자리)", "서지수준(리더/06 자리)", "입력수준(리더/17 자리)", "목록기술형식(리더/18자리)" 등이 저장된다.

그런데 〈그림 2〉의 레코드 길이와 시작주소가 최대 5자리를 차지하도록 정의한 것은 왜일까? 이는 MARC 레코드의 최대길이와 관련이 있다. 모든 MARC 레코드의 길이는 5자리 숫자를 초과할 수 없고, 그 의미는 5자리 숫자 중 가장 큰 99,999자가 MARC 레코드가 가질 수 있는 최대길이라는 것이다. 그래서 MARC 레코드의 최대길이가 바로 99,999캐릭터 또는 바이트이며, 그 범위 내에서 가변하는 길이를 가질 수 있다는 것이다.

리더 뒤에 바로 따라오는 정보는 디렉토리로서 〈그림 3〉에 보인 것처럼 서지레코드 내에 저장된 각 태그(즉, 필드)의 색인으로서 해당 레코드의 태그 수만큼 존재한다(〈그림 4〉 참조). 그러므로 디렉토리의 개별 요소는 고정장이지만 디렉토리 자체는 해당 서지레코드에 몇 개의 필드가 저장되는가에 따라 변한다. 각 태그에 대한 정보를 저장하는 디렉토리는 각 태그에 대한 길이와 그 시작 위치를 저장한다. 디렉토리에서 필드길이는 3에서 6째 자리까지 4자리를 차지하는데, 이는 위 리더구조의 설명에 사용한

논리를 그대로 적용하면 MARC 레코드 내 한 필드의 최대길이는 9,999 캐릭터 또는 바이트를 초과할 수 없음을 알 수 있다.



〈그림 3〉 MARC 레코드 디렉토리 구조

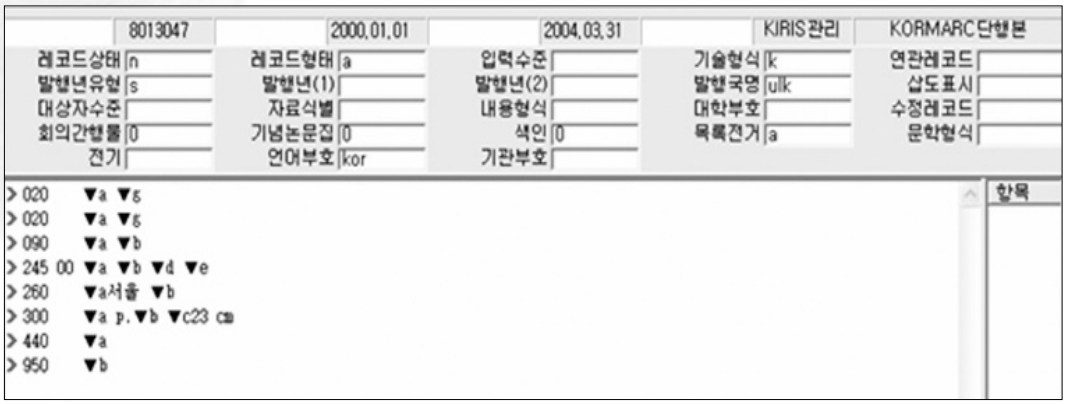
태그 (Tag)	필드길이 (Length)	시작위치 (starting position)
001	0016	00000
003	0008	00016
005	0017	00024
008	0041	00041
010	0019	00082
020	0031	00101
035	0025	00132
035	0017	00157
040	0012	00174
100	0025	00186
245	0076	00211
250	0022	00287
260	0054	00309
300	0042	00363
504	0033	00405
650	0037	00438
650	0049	00475
651	0035	00524
952	0073	00559

〈그림 4〉 디렉토리의 엔트리

리더와 디렉토리 뒤에 오는 것은 제어필드인데, MARC 태그로 001에서 008태그에 저장되는 정보를 의미한다. 이들은 시스템에 의하여 자동으로 부여되는 001과 같이 시스템 제어번호나 005와 같은 레코드 최종갱신일시, 그리고 008태그처럼 MARC 레코드 화면에서 고정장이라 불리는 곳에서(〈그림 5〉 참조) 입력하는 언어, 발행연도 유형, 발행국명, 대상자 수준 등을 들 수 있다.

고정장, 가변장, 제어필드, 그 차이는?

더 진행하기 전에 MARC 레코드 내에서 고정장 필드와 가변장 필드에 대하여 확실한 정의를 내리고 가면 도움이 될 것이다. 우리가 흔히 MARC 포맷으로 서지레코드를 구축할 때 화면 상단에 위치한 요소들을 "고정장"이라고 칭한다(〈그림 5〉 참조). 왜냐하면 그 곳에 입력되는 정보는 리더나 제어필드에 저장되며, 해당 필드의 길이가 고정장이거나 해당 데이터요소가 코드화되어 요소의 길이가 고정장인 경우이기 때문이다. 하지만 엄밀한 의미에서 이들 제어필드 모두가 고정장 필드는 아니다. 다시 말하여 이 요소들이 실제 MARC 포맷에 저장될 때는 위에 리더의 요소들을 설명한 것처럼 조합되어 하나의 필드를 구성하는데(이들 코드는 해당 필드 내의 상대적 위치에 저장된다), MARC 포맷에서 명확히 고정장 필드라고 정의한 것은 005, 006, 008뿐이며, 007태그도 자료형태에 따라 필드길이가 변하므로 엄밀한 의미에서는 고정장 필드라고 하기에는 다소 제약이 따른다. 따라서 MARC 레코드에서 고정장 필드는 〈그림 1〉에 보인 리더와 위에 언급한 몇몇 제어필드뿐이다.



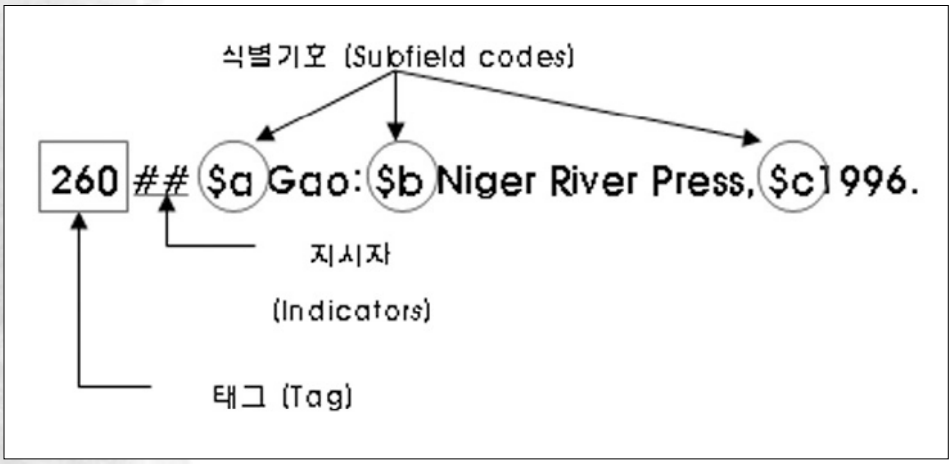
〈그림 5〉 MARC 레코드 구조

앞서 언급한 것처럼 원래 가변장필드의 이름은 "가변장 제어필드(variable control fields)"와 "가변장 데이터필드(variable data fields)"[11]인데 편의상 제어필드와 데이터필드라고 한다. 그리고 종종 데이터필드만을 가변장필드라고 칭하기도 한다. 이 둘을 구분하는 가장 명확한 방법은 예를 들어 005와 같이 태그 시작에 0이 두 개가 붙은 필드를 제어필드라고 하고, 090이나 245와 같이 0이 하나만 있거나 아예 없는 필드를 가변장 필드라고 하는 것이다. 또 다른 구분방법은 제어필드에는 데이터필드와는 달리 지시기호(indicators)나 식별기호(subfield codes)가 사용되지 않는 것이다(〈그림 6〉 참조).

왜 MARC 포맷은 이렇게 복잡한가?

MARC에서 리더나 디렉토리를 우리는 눈으로 본 적이 없고, 제어필드의 대부분도 코드화되어 있다. 그럼 왜 이렇게 복잡하게 만들었을까? 그 이유는 MARC의 이름이 시사하듯이 기계, 즉 컴퓨터가 쉽게 인식하도록 하기 위한 부분이 많기 때문이다. 앞서 MARC 포맷은 서지레코드 교환을 위한 표준이라고 하였다. 그러므로 교환 또는 다운로드된 MARC 레코드는 도서관자동화시스템에 반입되면서 일단 필요한 부분에 따라 모두 해체된다. 이 절차의 초기에 필요한 부분이 리더라고 할 수 있다. 즉 "글을 읽지 못하는" 컴퓨터가 복수의 MARC 레코드가 들어있는 파일을 개별 레코드로 나누어 읽어 들이기 위해서는 리더에 명시된 해당 레코드의 레코드 길이를 알아야 하기 때문이다[12]. 그 다음, 디렉토리는 특정 MARC 레코드 내에서 개별 태그의 시작위치와 길이를 알려주는데, 이는 우리가 검색을 위하여 저자명 또는 서명을 색인하는데 그들이 저장되어 있는 물리적 위치를 컴퓨터에게 정확히 알려주는 역할을 한다. 이 밖에도 아래 예와 같이 태그 내에서는 식별기호를 사용하여 하위 요소를 컴퓨터가 인식할 수 있도록 해 주고 있다(〈그림 6〉 참조). 결국 이들은 컴퓨터를 위한 것이지 일반 이용자나 사서를 위한 것은 아니다.

11) Network Development and MARC Standard Office (1988). *USMARC Format for Bibliographic Data: including Guidelines for Content Designation*, v. 1. Washington, D.C.: Library of Congress. p. 3.
12) 물론 MARC에는 레코드를 구분하기 위한 레코드 종결자(record terminator)라는 특수문자조합도 사용된다.



〈그림 6〉 MARC 레코드 구조

"고정장" 정보의 용도는?

앞서 기술한 리더에 입력되는 고정장 요소와 제어필드에 입력되는 코드화된 정보는 이용자나 사서를 위한 것이 아니라 컴퓨터를 위한 것이라고 하였다. 부연하면, 이들이 주로 활용되는 분야는 OPAC의 검색화면에서 흔히 볼 수 있는 자료유형(이전에는 자료별 MARC 유형, 통합 후 리더 내 요소 활용), 언어(008/35-37 언어부호), 발행년도(008/6 발행년도 유형, 008/07-10 발행년도 1과 008/11-14 발행년도 2), 또는 이용자 수준(008/22 대상자 수준) 등 검색범위를 제한하는 용도이다. 그 밖에 사서들을 위하여 각종 통계를 생성하는 용도에도 이들이 주로 사용된다. 예를 들면, 한 해 동안 도서관이 정리한 자료의 수를 형태별로 통계를 생성한다던가, 올해 입수된 동양서와 서양서의 통계를 추출하는데 이와 같은 코드화된 정보가 사용되는 것이다. 재미있는 것은 도서관에서 일반적으로 동양서와 서양서 통계를 필요로 하는데, 그 정의가 명확하지 않다는 점이다. 동양서를 발행국에 따라 결정할 것인가(008/15-17 발행국명) 아니면 자료언어로 결정할 것인가(008/35-37 언어부호)에 따라서 사용되는 데이터(즉, 필드)가 달라지고 그 결과 수치가 변화할 것은 당연하다. 사실 MARC 레코드에 저장된 코드화된 정보는 도서관에서 일반적으로 제시하는 통계나 검색인터페이스에 아직 사용되지 않는 요소들도 있지만, 어쨌든 코드는 사람에게는 불편하지만 컴퓨터에게는 그 보다 처리가 편한 것이 없기 때문이다.

결언

여기에서는 MARC 포맷의 역사, 구조, 그리고 그 용도에 대하여 가능하면 다른 자료에서 언급하지 않은 내용들을 설명하려고 노력을 하였다. 우리가 MARC 레코드에서 막연히 입력하던 "고정장"이라는 부분이 어떻게 활용되는지 보다 명확히 이해함으로써 최초에 MARC 포맷이 만들어진 목적과 지금까지도 대부분의 도서관들이 이를 이용하게 된 배경까지도 이해할 수 있다. 말하자면 고정장의 기능, 그리고 그곳에 입력되는 하나 하나의 코드가 도서관 서지레코드의 처리, 공유와 도서관 간 협력을 위한 의미 있는 내용을 반영하기 때문이다. 도서관자동화는 어떤 의미에서는 MARC 포맷에서 시작되었다고도 할 수 있다. 그러나 여기서 강조하고 싶은 것은 MARC 포맷이 "전부"가 아니라는 사실이다. 보다 중요한 것은 MARC 포맷이라는 '용기' 안에 무엇을 담는가이고, 그 무엇을 담는가는 바로 목록을 얼마나 잘하는가에 달려있다는 점이다. 정확하고 세심한 목록작업에서 얻어진 데이터를 MARC 포맷에 담지 않는다면, 그 또한 "Garbage In, Garbage Out"의 결과를 가져올 뿐이다.

흔히 사람들은 필자를 도서관자동화 전문가라고 하지만 그 바탕은 십여 년이 넘는 목록사서로서의 경험에서 시작되었다. 물론 목록만 열심히 해서 도서관자동화 전문가가 될 수는 없다. 다소의 전산 지식을 필요로 한다. 따라서 예비사서들은 자료조직론 과목에서 목록의 기초를 다지고 전산 관련 기초과목들을 몇 가지 수강함으로써, 도서관자동화에서 필요한 데이터 생성과 처리의 기본지식을 갖추는 것이 바람직하다고 하겠다. 우리는 프로그래밍 실력을 필요로 하지 않지만, MARC의 구조와 같이 도서관자동화의 기반이 되는 원리를 알 때 단순한 기계적 운영이 아니라, 보다 효율적 업무수행이 가능하기 때문이다.