code{4}lib
JOURNAL

Search

Current Issue
Issue 40, 2018-05-04

Previous Issues
Issue 39, 2018-02-05
Issue 38, 2017-10-18
Issue 37, 2017-07-18
Issue 36, 2017-04-20
Older Issues

For Authors
Call for Submissions
Article Guidelines

Issue 11, 2010-09-21

Interpreting MARC: Where's the Bi Data?

The MARC data format was created early in the history of di In this article, the author entertains the notion that viewin modern technological perspective leads to interpretive pro confusion of "bibliographic data" with "catalog records." I idea through examining a specific MARC interpretatio undertook early in his career and then revisited nearly f Revising the code that performed the task confronted hi misconceptions about MARC that were rooted in his worldu he thought "structured data" should be and helped him to p more appropriate context.

by Jason Thomale

Introduction

The Machine Readable Cataloging (MARC) format was c the early 1960s and first piloted in 1966 (Avram, 1975). 40 years old. Considering just the advances in compute representation that have happened since, today's world in which MARC was conceived. In 1966, it would still be t Edgar F. Codd would publish his first paper describing a for data as an IBM Research Report, and four years until the paper and publish it more widely in *Communications f* 1998); eight years until Donald Chamberlin and Raymond present their work on SEQUEL (SQL) (Chamberlin & Boyc years until Peter Chen would first propose the Entity Re (Chen, 1976). We who work with library technology and s help but view MARC [1] through a lens colored by 44 year technological change (Figure 1). We now look at MARC b worldview that is utterly different than the one that gave format; making anachronistic assumptions about how w would work is all too easy. Of course we get poor results MARC data the way we would treat data in a relational da

predates the earliest formal expression of relational da
concepts by three years!

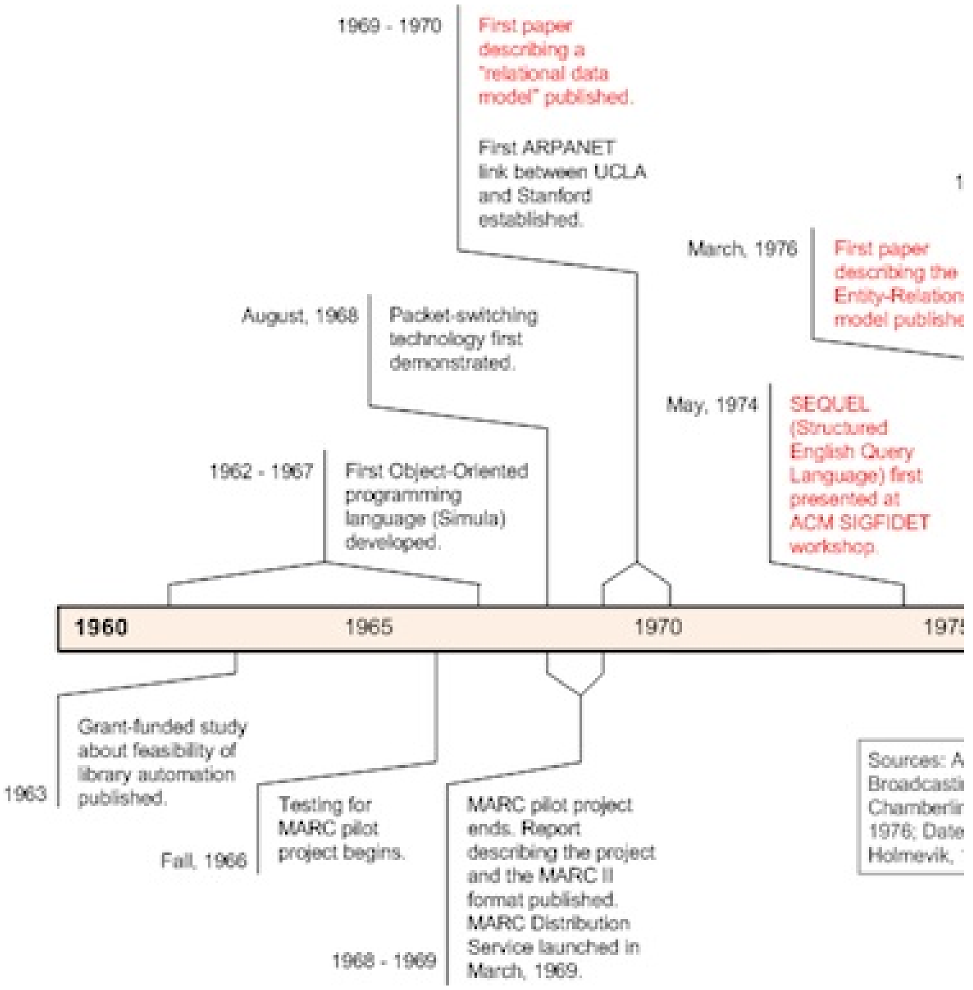

1969 - 1970
First paper describing a "relational data model" published.
First ARPANET link between UCLA and Stanford established.
March, 1976
First paper describing the Entity-Relation model publishe
August, 1968
Packet-switching technology first demonstrated.
May, 1974
SEQUEL (Structured English Query Language) first presented at ACM SIGFIDET workshop.
1962 - 1967
First Object-Oriented programming language (Simula) developed.
1960
1965
1970
197
1963
Grant-funded study about feasibility of library automation published.
Fall, 1966
Testing for MARC pilot project begins.
1968 - 1969
MARC pilot project ends. Report describing the project and the MARC II format published. MARC Distribution Service launched in March, 1969.
Sources: A
Broadcasti
Chamberlin
1976; Date
Holmevik, 1

Figure 1. Timeline comparing creation of MARC to majo
software, networking, and data representation betwee

Furthermore, the cataloging task, which generates the
a MARC record, seems enigmatic to most library techno
good reason. Whereas a format built upon a modern dat
store data about bibliographic items directly [2], MARC
bibliographic data by way of "cataloging record[s] … or t
traditionally shown on a catalog card" (Furrie, 2009). Fro
systems design perspective, this seems odd. Our syste
to facilitate online search and retrieval of bibliographic
catalog card abstraction really the best carrier for that
context? But MARC was not invented to drive computeri
retrieval systems. Its original purpose was to automate
tasks of a 1950s/60s technical services department—i.
and printing of catalog cards (Avram, 1975; Coyle, 2005
2007; Tennant, 2002). The same basic rules that determ
bibliographic data was to be stored and displayed on th
became the same rules that determined how data was t
stored in MARC records. In fact, though they have chang
cataloging rules originated long before the advent of m
technology (Coyle & Hillmann, 2007; Taylor, 1999).

We can now perhaps more easily understand some of th
see within the current library metadata environment, wh
surface, might seem baffling. Library catalogers and pr
(often passionately) about what constitutes "good libra
the one hand, many librarians consider traditional librar
only to be useful, but vital for retrieval of online bibliogr
[3]. On the other, many programmers who try to work wit
come to conclude that they hinder online access and re
they are so difficult to interpret algorithmically [4]. Inde
witnessed and read conversations in which programme
attempt to explain their points-of-view to one another.
sides speak vaguely. Concrete examples—if offered—a
language that only one side or the other understands. V
communication seems to happen [5].

As a metadata librarian, I have a foot in both worlds: I mu
and interpret MARC data, but I must also understand sys
and programming. My background in programming and d
administration, however, preceded my training as a libra
learned cataloging, my worldview was already set. Desp
experience, I still find that I make automatic, mistaken a
MARC. Some recent metadata clean-up work required m
that I had written much earlier in my career; it afforded
to see firsthand some of the assumptions I made as a n
concrete results of those assumptions. Solving one pr
—cleaning up generic *title* metadata that had been deriv
245 field—helped me consciously to experience certai
working programmatically with MARC data that would ha
seemed counter-intuitive to me when I wrote the origina
what led to these insights illustrates how programmers
differently about MARC data to understand it better and
to parse it more effectively. It also reveals evidence de
problems that traditional library cataloging poses in a m
context.

The Problem

Obtaining metadata for digitized items from the physic
records is a common task. Three and a half years ago I d
program that did such a thing for one of my institution's
collections: recorded music that we stream online to s
of Music's teaching activities. Because this was one of
collections and I was still new at working with MARC dat
metadata suffered from a number of problems.

When I recently undertook a revision of my original code
problems, one issue near the top of my “to-fix” list seem
surface. *Title* metadata (that is, titles for albums—not inc
sometimes exhibited odd formatting quirks. Indeed, ove
fixed a number of records manually that simply looked w
recently I had begun to notice titles that were both form
seemingly incomplete.

Looking back at the code that I wrote three and a half ye
try to remember how I originally intended the *title* elemen
data came from the MARC record’s 245 field. Although I
followed cataloging conventions and used the entire tit
exactly as it appeared in the MARC record, the formattin
stilted, and jarring to the untrained eye. It would not serv
thought. In addition, the 245 tended to contain extraneo
that I did not think users would recognize as belonging t
the statement of responsibility. Removing the extraneou
required reformatting what remained. Ultimately, I want
appear as natural, simple, and readable as possible—ba
rules of English grammar than the cataloging rules. Whe
version of the title-extraction routine, I did not realize t
such a high failure rate, but I later noticed that, in rough
odd formatting errors had crept into the output that thw

The Original Solution

The original algorithm was simple. I began with the 245‡
title), which was all that was actually required. If the 245
(remainder of the title—usually used for a subtitle) or a ‡
(number/name of a part or section of a work), I added a
‡b, ‡n, and/or ‡p, in that order, onto the title string. I dec
‡c (statement of responsibility) because it was not actu
title, and I decided to ignore the ‡h because it tended ju
“[sound recording].” Other subfields were not commonly
them.

Further, I recognized that any equals signs (=) in the dat
that were translations of what had been transcribed fro
translations I thought would be important to keep but th
notation would be jarring to end users, so I converted e
commas. I also removed extraneous punctuation from th
(such as forward slashes (/)). Below are three examples
not convert as intended.

```
=245 10‡a3 symphonies‡h[sound recording] ;‡bThe roc
 = Le rocher /‡cSerge Rachmaninoff.
```

Came out as:

```
3 symphonies: The rock, Der Fels, Le rocher
```

By sheer coincidence, this formatting makes it look as t *Fels,* and *Le rocher* are the titles of the three symphonies. is its own work separate from *3 symphonies* whose title ju translated into German (Der Fels) and French (Le rocher record.

```
=245 10‡aFantasie in C major, op. 15‡h[sound recording
 :‡bWandererfantasie /‡cFranz Schubert. Fantasie in C r
 op. 17 / Robert Schumann.
```

Came out as:

```
Fantasie in C major, op. 15: Wandererfantasie
```

This time the formatting is not bad, but it completely mi work on the album (Schumann's *Fantasie in C Major, op. 1*

```
=245 10‡aConcerto no. 1 for piano & orchestra, op. 15,
 major‡h[sound recording] =‡bC-Dur = ut majeur ; Conce
 piano & orchestra, op. 37, C minor = c-moll = ut mineur /
 van Beethoven.
```

Came out as:

```
Concerto no. 1 for piano & orchestra, op. 15, C major: C
 ut majeur ; Concerto no. 3 for piano & orchestra, op. 37
 c-moll, ut mineur
```

In this case, the resulting title is at least decipherable, l follow my "natural, simple, and readable" criteria. Speci punctuation seems inconsistent—even random.

A Different Approach

My revision began with a systematic investigation about wrong. I noted some of the problem titles and examined 245s vis-à-vis my original code. I asked myself: what was I made assumptions the first time that did not match hov MARC was structured? Did those items simply contain c If the mistakes were in the cataloging, were they regular enough that I could code around them? If the mistakes v could I make sure that I was not simply writing revised co set of bad assumptions about the data?

Examining the MARC proved to be eye opening. The majo
appeared not to be in the cataloging, but rather in how I
Seeing my misinterpretations so clearly made me realiz
change my entire approach to the problem. The three ex
illustrate my mistaken interpretations.

```
=245 1\ ‡aMusique de chambre.‡nVol. II =‡bChamber mu
 Kammermusik, Vol. II‡h[sound recording] /‡cFauré.
```

```
=245 10‡aFantasien op. 116‡h[sound recording] =‡bFan
 117 ; Klavierstücke op. 118 & 119 = Pieces for piano /‡
```

```
=245 10‡aOverture to Candide‡h[sound recording] /‡cL
 [arr.] Grundman. George Washington Bridge / William Sc
 overture / Aaron Copland. El Salon Mexico / Aaron Copla
 Hindsley. Chester / William Schuman. La fiesta mexican
```

In my initial solution, I was treating each subfield as a dis
data. I picked and chose the subfields I thought I neede
ignored the others. In many cases, however, the subfield
the ‡h and the ‡c—actually contained important informa
example, above, the ‡h contains a translation-indicator
that the next subfield begins with a translation, and is pr
a subtitle; in the third example, the ‡c contains a lengthy
works that appear on the album in addition to what is alre
‡a.

My initial solution also disregarded the order in which su
It assumed that it could impose a standard order (‡abnp
what appeared in the data—indicating that I thought eith
not matter or that subfields always appear in that partic
first example, not only does a ‡n appear between a ‡a an
also contains an "=" immediately before the ‡b. Taken o
would apply to something other than what is obviously in
"Chamber music" is a translation.

On a more general level, I realized that the contents of t
and acted less like data from a structured data record a
textual markup from a document [6]. Structured data for
for machine consumption and tend to follow rules that a
consistent—simpler rules make data easier to parse. Als
structured a data record is, the more explicit the seman

Meaning is clear and encapsulated—the overall context appears within a record is irrelevant because, apart fron specified in the data model, context carries no semanti Interpreting a structured data record is no more compl each data element and interpreting it according to that specifications. A document, however, is less straightfo information it contains is meant more for *human* consur document is marked up to aid machine-processing, the structure is based on linguistics rather than a format th be machine-readable. The information in a marked-up d therefore behaves more like language than data. Meanin encapsulated. The document as a whole contains sema beyond the sum of its marked-up data elements. Interp document requires reliance on subtle ambiguities and c than it does simple rules and specifications. Removing elements, unless done in a way that takes context into a changes the meaning of the document as a whole. Looki perspective, I saw that the MARC 245s that I had been st demonstrate the characteristics of marked-up docume functions as a complete unit. The subfields help make n delineating the parts of that unit, but the context in whic appear also carries meaning: if you remove a subfield wi how it relates to the whole, or if you change the order of considering how they relate to one another, you can ina the meaning of the whole. The 245 has an implicit struct defined by the subfields; rather it is defined by the cata known as the Title Statement. The subfields merely hel of the parts of that Title Statement.

Because my original solution to the problem had come f data perspective, its approach was decidedly rules-bas maintained that approach throughout the revision proce would have been to examine the MARC and cataloging d more closely to determine the precise rules upon which the 245s were supposed to be based. Then I would have methods for handling the subfields that followed these would have ended up with a complex series of loops and statements that would have attempted to arrive at appr based on which subfields appeared in what order and wh they contained. The code would have become messy qu

But what if I treated the data more as it was now beginni textual markup? Looking at the problem this way reveale possibilities. I began to think of a solution that would re the patterns inherent in the data instead of attempting then reconstruct the data based upon the rules on whic

Finally, to ensure that I did not repeat my earlier mistake
data set that was too limited, I employed a more system
(Figure 2):

1. I created a script that would apply my title extractio
   245 in the entire batch of MARC records for the coll
   the results in a text file for me to examine. I also cre
   would apply the same routine to a single title that I d
   script (for testing and developing).

2. I ran the batch conversion script on the entire MARC
   script generated a plain text file containing each co

3. I looked through the output to find the next title tha
   correctly.

4. I looked up the 245 in a human-readable version of t
   store.

5. I used the new 245 as my test case and rewrote or tw
   until that 245 converted to an acceptable-looking t
   breaking my previous test cases.

6. I repeated steps 2 through 5 until I could find no mo
   batch output that looked incorrect.

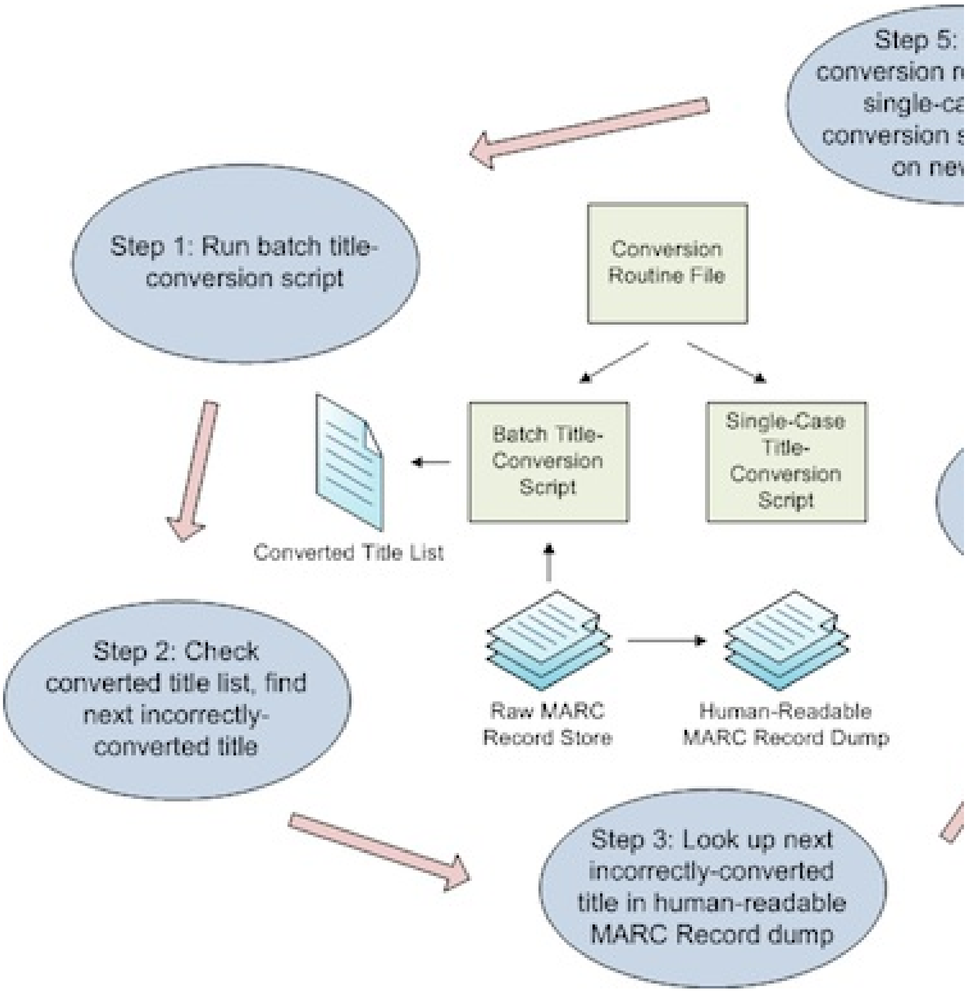

Step 5:
conversion r
single-ca
conversion s
on new
Step 1: Run batch title-conversion script
Conversion Routine File
Batch Title-Conversion Script
Single-Case Title-Conversion Script
Converted Title List
Raw MARC Record Store
Human-Readable MARC Record Dump
Step 2: Check converted title list, find next incorrectly-converted title
Step 3: Look up next incorrectly-converted title in human-readable MARC Record dump

Figure 2. Title-conversion-routine-revision w

Toward a Revised Solution

When I was ready to write my new algorithm, I began by b
my MARC record dump to see what consistent patterns
245 data. I noticed several. Periods and semicolons see
major components of the overall title—especially when t
works were contained within the overall album title. Forv
delimited the beginning of a list of responsible parties,
ended with a period. Equals signs indicated translations
commas indicated minor subdivisions within a compone
work's title. Brackets indicated information that the cat
included but had not transcribed directly from the physi
cataloging. From these particular patterns, I could divid
appearing in the title into three separate categories: pu
served as a major component separator (.;), punctuatior
information I wanted to remove (/=[]), and punctuation th
minor component separator (,:).

I noticed a couple of complicating factors, however. Fir
seriously—periods were used not only as component de
indicate abbreviations. Second, I noticed that there wer
which bracketed information was desirable to keep in th

The first problem was the more challenging. There seer
convenient way to tell which periods were used for abbr
which were not. I solved the problem using brute force—
separate script that looped through each 245 in the ent
batch. It read any words that ended with a period, tallied
generated a sorted list:

721	op
573	no
251	Mozart
250	Bach
109	Beethoven
105	nos
95	Vol
94	K
83	Haydn
70	Mendelssohn
69	Vivaldi
67	Schubert
65	Schumann

65	Brahms
43	Handel
40	J
35	Chopin
34	D
30	Shostakovich
30	Prokofiev
29	al
28	No
28	Debussy
25	music
25	Ravel

It took little time to eyeball the list, extract what appear abbreviations, and use those abbreviations as exceptio matching routines. I also excepted any initials (i.e., indiv characters appearing as words unto themselves or sets alphanumeric characters separated by periods).

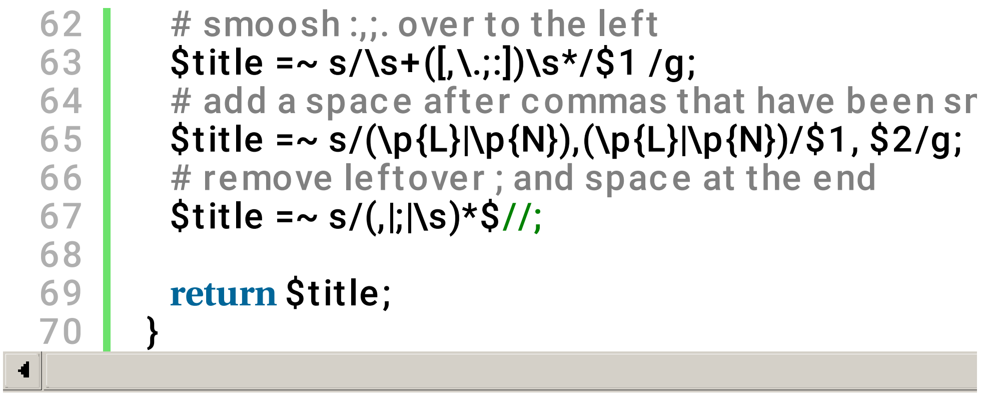
The second complicating factor—the brackets—was mu rectify. In general, I noticed that brackets appearing in that I wanted removed; brackets appearing in any other data that I did not want to remove, though I still wanted t brackets themselves.

My new code takes a basic two-step approach. First, it l subfield contained within a 245 field and preprocesses it processes abbreviations; it generates an unambiguou character (the pipe (|)) to replace the periods that serve component" delimiters; and it preprocesses certain oth marks based upon the subfield in which they fall, such as the ‡h. Second, it processes the entire title as a single s patterns in the whole string that it would otherwise miss each subfield in isolation; it determines data to be remo it; it consolidates any repeated pipes into a single pipe finally it converts all pipes to semicolons and performs cleanup.

After a few iterations of coding, testing against a single against the entire MARC dump, and then picking a new te at the final routine. The Perl code for the conversion ro

```
sub version2_convert {
  my $field = shift;        # $field is a MARC::Field 
  my $abbrevs = "op|no|nos|vol|vols|al|etc|nr|st|p
```

```
       ."dr|app|arr|orchestr|orch|ed";
my $title;

foreach my $subfield ($field->subfields()) {
  my $title_part = $subfield->[1];
  # convert all . to |
  $title_part =~ s/(\.)/|/g;
  # convert | after $abbrevs back to .
  $title_part =~ s/(^|[^\p{L}^\p{N}]+)($abbrevs)\
  # convert | after initials back to .
  $title_part =~ s/(^|[^\p{L}^\p{N}]+)(\p{L})\|/$1
  # convert . at the end of the string to .|
  # (periods at the end of a subfield indicate e
  $title_part =~ s/(\|)(\p{L})\|/$1$2./g;
  $title_part =~ s/(\.)(\p{L})\|/$1$2./g;
  $title_part =~ s/\.$/.|/;

  # if this is a subfield h
  if ($subfield->[0] eq "h") {
    # make sure there's a space before any end
    $title_part =~ s/(\S)(\/)$/$1 $2/;
  } else {
    # remove [ and ]
    $title_part =~ s/[\[\]]//g;
  }

  # if this is a subfield c and a / is missing from
  if ($subfield->[0] eq "c" && $title !~ /\/$/) {
    $title .= " /";
  }

  # if this is a subfield n or p
  if ($subfield->[0] eq "n" || $subfield->[0] eq "p"
    # convert a pipe at the end of the title strin
    $title =~ s/\|$/,/;
    # convert pipes within the subfield n or p to
    $title_part =~ s/\|/,/g;
    $title_part = ", " . $title_part;
  }
  $title .= $title_part;
}

# remove ||| (comes from transformation of ...)
$title =~ s/\|\|\|//g;
# remove everything between = and /;,:| or eos
$title =~ s/=(.*?)(\s\/|;|,|:|\||$)/$2/g;
# remove everything between / and |
$title =~ s/\s+\/.*?(\||$)/$1/g;
# remove everything between []s
$title =~ s/\[.*?\]//g;
# convert ; to |
$title =~ s/;/|/g;
# remove repeated |
$title =~ s/(\s*\|\s*)+/\|/g;
# remove repeated ,
$title =~ s/([:,\|])(\s*,\s*)/$1 /g;
# convert | to ; (space insensitive)
$title =~ s/\s*\|\s*/; /g;
```



```
    # smoosh :,;. over to the left
    $title =~ s/\s+([,\.;:])\s*/$1 /g;
    # add a space after commas that have been s
    $title =~ s/(\p{L}|\p{N}),(\p{L}|\p{N})/$1, $2/g;
    # remove leftover ; and space at the end
    $title =~ s/(,|;|\s)*$//;

    return $title;
}
```

Table 1 shows, for a selection of titles, a comparison be
245, the output of the original algorithm, and the output
algorithm.

MARC 245	Original Output
=245 1\ ‡a3 symphonies‡h[sound recording] ;‡bThe rock = Der Fels = Le rocher /‡cSerge Rachmaninoff.	3 symphonies: The rock, Der Fels, Le rocher
=245 1\ ‡aFantasie in C major, op. 15‡h[sound recording] :‡bWandererfantasie /‡cFranz Schubert. Fantasie in C major, op. 17 / Robert Schumann.	Fantasie in C major, op. 15: Wandererfantasie
=245 1\ ‡aConcerto no. 1 for piano & orchestra, op. 15, C major‡h[sound recording] =‡bC-Dur = ut majeur ; Concerto no. 3 for piano & orchestra, op. 37, C minor = c-moll = ut mineur /‡cLudwig van Beethoven.	Concerto no. 1 for piano & orchestra, op 15, C major: C- Dur, ut majeur ; Concerto no. 3 for piano & orchestra, op. 37, C minor, c-moll, ut mineur
=245 1\ ‡aWeihnachtsoratorium‡h[sound recording] : BWV 248.‡nKantate 1-3 =‡bChristmas oratorio /‡c[Johann Sebastian Bach].	Weihnachtsoratorium Kantate 1-3, Christmas oratorio
=245 1\ ‡aFantasien op. 116‡h[sound recording] =‡bFantasias ; Intermezzi op. 117 ; Klavierstücke op. 118 & 119 = Pieces for piano /‡cJohannes Brahms.	Fantasien op. 116: Fantasias ; Intermezz op. 117 ; Klavierstücke op. 118 & 119, Pieces for piano
=245 1\ ‡aOverture to Candide‡h[sound recording] /‡cLeonard Bernstein ; [arr.] Grundman. George Washington Bridge / William Schuman. An	Overture to Candide

outdoor overture / Aaron Copland. El Salon Mexico / Aaron Copland ; [arr.] Hindsley. Chester / William Schuman. La fies	
=245 1\ ‡aViolin concerto no. 2‡h[sound recording] ;‡bSymphony no. 4 : Landscape /‡cPaul Cooper. Let us now praise famous men ; Elegy / Samuel Jones.	Violin concerto no. 2: Symphony no. 4 : Landscape
=245 1\ ‡aPentimento‡h[sound recording] /‡cEzra Laderman. Symphony no. 3 : The tricentennial / Lester Trimble.	Pentimento
=245 14 ‡aThe ballad of Baby Doe‡h[sound recording] :‡b[opera in two acts /‡clibretto by John Latouche ; music by Douglas Moore].	The ballad of Baby Doe: [opera in two acts
=245 14 ‡aThe protecting veil /‡cTavener. Third suite for cello, op. 87 / Britten. [Thrinos :‡bfor solo cello / Tavener]‡h[sound recording].	The protecting veil /: for solo cello / Tavener]
=245 1\ ‡a4 Orchesterstücke.‡nOp. 12 Sz51‡h[sound recording] =‡bOrchestral pieces = Pièces pour orchestre ; Konzert für orchester, Sz116 = Concerto for orchestra = Concerto pour orchestre /‡cBéla Bartok.	4 Orchesterstücke.: Op. 12 Sz51 Orchestral pieces, Pièces pour orchestre ; Konzert für orchester, Sz116 Concerto for orchestra, Concerto pour orchestra
=245 1\ ‡aString quartets op. 51‡h[sound recording] ;‡bString quartet op. 67 /‡cJohannes Brahms. “American quartet” op. 96 / Antonin Dvorák.	String quartets op. 51: String quartet op. 67
=245 1\ ‡aPartita no. 1, BWV 825‡h[sound recording] ;‡bEnglische suite no. 3, BWV 808 = English suite = Suite anglaise ; Französische suite no. 2, BWV 813 = French suite = Suite française /‡cJohann Sebastian Bach.	Partita no. 1, BWV 825: Englische suite no. 3, BWV 808, English suite, Suite anglaise ; Französische suite no. 2, BWV 813, French suite, Suite française

Table 1. Comparison between MARC 245, output of the and output of the revised algorithm.

Analysis and Conclusion

The revision process of my MARC title-extracting routi around a significant fact and a significant insight. The fa heart, a data format built to contain *catalog records*; biblio described via the catalog records rather than directly v MARC data. Understanding and internalizing this leads u we think of those catalog records as structured docume data records, then MARC has as much in common with a language (such as SGML or HTML) as it does with what w to be “structured data.”

Thinking broadly, the fact that the MARC format focuses records rather than bibliographic items lends credibilit MARC has been decreasingly relevant to library systems of the card catalog. Many in the library profession recog improving library tools requires the development of dat better support system functionality that modern library expect. Extensive efforts to rectify the situation have b over a decade, such as creating new models for bibliogr updating cataloging rules, and attempting to convert lib formats [7]. However, much bibliographic data is still loc Programmers who must write the code to interpret MAR daunting task of trying to understand a data format that to them. The aforementioned fact and insight can, perha MARC seem slightly less alien to the programming mind- examining the implications.

First, a MARC record does contain an explicit structure. subfields, and indicators, and our tools for processing M ability to extract discrete, granular chunks of data just a e.g., a database. Concrete rules and semantics define t into a MARC record and dictate the use of the fields and we must interpret, extract, or otherwise act upon the da documented rules and semantics can help guide our inte

Second, data within a single MARC field often behaves li markup. Unlike data occurring in a database, data expre document markup gains some degree of meaning based within the document and its relation to other data within MARC, punctuation that appears in one subfield can sub meaning of data in a different subfield; changing subfiel subtly change the data's meaning.

Third, extracting bibliographic data from MARC is often matter just of interpreting the explicit structural inform cases—such as the case of the 245 field—we must reco existence of both an explicit and an implicit structure. M comprise the explicit structure; cataloging rules define two are intertwined, and both must be interpreted. What programmers as cataloging artifacts—odd patterns of p formatting within the data that are often dismissed as m presentational—actually carry semantic meaning that is patterns are ignored. If we want to pull a book's title out must understand that doing so requires us to interpret t Statement" cataloging edifice—otherwise our methods

This need to decipher data built upon cataloging rules p perhaps the biggest technical hurdle in the quest to co into more machine-readable formats *en masse*. Catalogin complex history [8]. Multiple cataloging standards—or c developed throughout the years. Standards used concu —such as the Anglo-American Cataloguing Rules (AACR 8 continued to evolve. Because MARC contains catalogin enforce the use of any single cataloging code, it may no pinpoint exactly what set of cataloging rules any given r upon [9].

Furthermore, the act of cataloging—no matter the set o relies on human beings to enter data logically and consi modern cataloging rules are complex, any MARC record contain some errors that could cause a program to misi The "cataloging" in any given set of MARC records is the target.

In my particular example, the records that I used had be mostly according to some version of the AACR2 standar somewhat mitigated the effects of this problem. Even s approach I have described offers a potential avenue fo sets where this is not the case. Because the approach i engineering based on bottom-up, pattern-matching tec allow for greater flexibility in accommodating inconsist the cataloging does not follow the cataloging rules). It a no knowledge of exactly which cataloging standards we the MARC data; in theory, it only requires a programmer recognize the sets of consistent patterns and variation patterns that appear within the data. Future work would hypotheses on record sets that exhibit increasingly mo varied cataloging in order to help determine how to refi approach. Although outside my direct realm of expertis use of machine-learning algorithms and data analytics t identification would be another logical pathway for deve approach further.

References

Avram, H. D. (1975). *MARC; its history and implications*. Wa Library of Congress.

British Broadcasting Corporation. (2008, August 5). *The modern age*. Retrieved from: http://news.bbc.co.uk/2/hi/technology/7541123.stm

Coyle, K. (2005). Catalogs, card—And other anachronism *Academic Librarianship* 31(1), 60-62.

Coyle, K., & Hillmann, D. (2007). Resource Description a Cataloging rules for the 20th century. *D-Lib Magazine* (1 from: http://dlib.org/dlib/january07/coyle/01coyle.htm

Chamberlin, D. D., & Boyce, R. F. (1974). SEQUEL: A struc query language [Electronic version]. *Proceedings of the 19 Workshop on Data Description, Access, and Control*, 249-26 Association for Computing Machinery. Retrieved from: http://www.almaden.ibm.com/cs/people/chamberlin/se

Chen, P. P. (1976, March). The Entity-Relationship mode view of data [Electronic version]. *ACM Transactions on Da* March 1976, 9-36. New York, NY: Association for Compu Retrieved from: http://csc.lsu.edu/news/erd.pdf

Date, C. J. (1998). The birth of the relational model: Thi relational. *Intelligent Enterprise Magazine*, October 1998. http://intelligent-enterprise.informationweek.com/db_area/archives/19

Furrie, B., & Database Development Department of the Company. (2009). *Understanding MARC bibliographic: Ma cataloging* (8th ed.). Washington, DC: Library of Congress http://www.loc.gov/marc/umb/

Hauben, R. (1998, June 23). *From the ARPANET to the Inte ARPANET TCP/IP digest and the role of online communicatio from the ARPANET to the Internet*. Retrieved from: http://www.columbia.edu/~rh120/other/tcpdigest_pap

Holmevik, J. R. (1994). Compiling Simula: A historical stu
technological genesis [Electronic version]. *IEEE Annals*
Computing 16(4): 25-37. Piscataway, NJ: IEEE Education
Department. Retrieved from:
http://www.idi.ntnu.no/grupper/su/publ/simula/holmevi
ieeeannals94.pdf

Stroustrup, B. (2010, March 7). *Bjarne Stroustrup's FAQ*. R
http://public.research.att.com/~bs/bs_faq.html

Taylor, A. G. (1999). *The Organization of Information*. Engl
Libraries Unlimited.

Tennant, R. (2002, October 15). MARC must die. *Library*
from: http://www.libraryjournal.com/article/CA250046.

Notes

[1] Just a quick note of clarification: throughout the pap
have in mind the MARC 21 format for bibliographic data,
refer to it only as "MARC".

[2] The Functional Requirements for Bibliographic Reco
perfect example of a "modern data model" for bibliogra
model was developed in the late 1990s. See
http://www.ifla.org/en/publications/functional-requirer
bibliographic-records for more information.

[3] This sentiment is so widespread in the library profes
citations I could provide to back up this statement woul
incomplete and largely unnecessary—any cataloger wou
assertion. You can, however, view a good illustration of t
example—in the comments following the Language Log l
the "Metadata Train Wreck" that is Google Books:
http://languagelog.ldc.upenn.edu/nll/?p=1701.

[4] As with the preceding statement about librarians' att
cataloging, I cannot really give citations to back up this
certainty and objectivity—as far as I can tell, programme
toward cataloging have never been studied. I can give e
Roy Tennant's 2002 piece entitled "MARC Must Die" (ref
elsewhere in this paper) provides a good look at some o
problems from a modern technological viewpoint.

Also, on occasion, library coder Jonathan Rochkind doc
problems writing code that must use MARC data. See, fo
http://bibwild.wordpress.com/2009/09/24/a-reasonab

series-data-in-marc/ http://bibwild.wordpress.com/20
marc-issues-700/ and
http://bibwild.wordpress.com/2009/09/30/cataloging-

There is also the statement about working with MARC da
made by Google engineer Leonid Taycher that "the first
learn was that the 'Machine Readable' part of the MARC
(from http://go-to-hellman.blogspot.com/2010/01/goo
book-metadata-privates.html).

Finally, the NGC4lib, or Next Generation Cataloging for L
(view archives at http://serials.infomotions.com/ngc4li
wealth of documented discussions between programme
about the utility of cataloging data.

[5] Clearly this is my opinion and is based mainly on anec
again submit that the NGC4lib archives contain numerou
backing up my point that catalogers and programmers o
one another.

[6] Within the XML community exists an analogous conce
"document-centric" versus "data- centric" XML. (For a c
explanation, see http://techessence.info/node/51.) Thi
document/data division as a continuum rather than a bin
applying it beyond just the XML format), MARC would fall
middle. It is fielded and thus behaves like data, but the c
of those fields behave more like documents (where the
mark-up).

[7] FRBR is the primary example of a "new" model for bibl
The work happening on Resource Description and Acce
primary example of cataloging rules being updated. The
number of recent efforts to publish library cataloging d
formats on the Web: for example, http://www.viaf.org an
http://id.loc.gov/.

[8] Chapter three (pages 37-52) of *The Organization of In*
Arlene G. Taylor contains a succinct chronology of the h
developments that lead to modern cataloging. Catalogi
largely a product of the 19th and early 20th centuries.

[9] Position 18 of the MARC leader contains a code that
determine what descriptive cataloging standard was us
of the record. It can indicate AACR2, International Stand
Description (ISBD), non-ISBD, or an unknown cataloging
knowing you have a "non-ISBD" or "unknown" standard is
helpful, and it does not tell you what version of AACR2 or

[10] Because the records were cataloged according to
formatting of the MARC 245s (e.g., the punctuation) co
Although the approach that I took did not require it, I co
the ISBD documentation to help me make sense of the d

About the Author

Jason Thomale (j <dot/> thomale <at/> ttu <dot/> edu) is
librarian currently going on his sixth year at Texas Tech

Subscribe to comments: For this article | For all articles

24 Responses to "Interpreting MARC: Where's the Bibli

Please leave a response below, or trackback from your own site.

1. Jakob, 2010-09-22

   Thanks for the practical insight! I'd like to point out that y
   does not only hold for MARC, but for almost any kind of da
   fully atomic and normalized. Schemas (SQL, XML Schema,
   describe what you called "explicit" structure, but as people
   they add "implicit" structure, based on additional conventi
   rules. I bet in 40 years we will complain about some ancient
   data that contains strange artifacts. Of course MARC is m
   because it is so old, designed for cataloging records. But t
   information out of other decade-old databases that have
   for a specific use-case that is not yours, and you will stum
   similar problems.

2. Mary Mastraccio, 2010-09-22

   Jason Thomale's article on interpreting MARC is THE best
   read on the issue. It should be required reading for every ca
   anyone trying to create or map MARC records.

3. jrochkind, 2010-09-22

   Excellent article. I'm hoping that the contortions you had t
   to get the title out in a flexibly displayable format (spendi
   of time investigating, arriving at a fairly complex algorithm
   demonstrate why Marc is not in fact a very good data form
   contemporary needs. But I'm also planning on taking your a
   porting it to ruby for my own marc title display purposes, t
   including the code.

4. Mike M., 2010-09-22

Very interesting article with many clarifying points. While I a
thrust of the article as an exemplar of the difficulties of M
needs of info retrieval I'd like to point out as a practical po
probably prefer to use the 240 Uniform Title and 700|t as r
fields. You'd run into many of the same problems but would
out a bit cleaner.

5. Jason W. Dean, 2010-09-22

Jason, what a beautiful article. I love the whole to part exp
you do about metadata in the beginning – and specifically
within this galaxy of metadata. Illuminating – I am sharing t
thanks for writing this!!

6. Matthew Phillips, 2010-09-23

I enjoyed the article, but I wondered whether you had actua
AACR2 to investigate the rules regarding punctuation? You
"Toward a revised solution" reads as though you were tryin
the patterns by look at the example data. Fair enough, I sup
AACR2 is considered indigestible by many people. But ofte
punctuation preceding the subfield indicator is a greater cl
purpose of the data than the indicator itself.

Anyway, your approach works very well, especially when face
conquering |c subfield after which no more subfields can be

It's just a shame that US-MARC won. In the UK we develope
variant call UK-MARC where the subfield markers were close
the data types, and punctuation for display was generated
subfield markers. Sadly this computer-friendly format was
the name of international standardisation. You would not l
the problems if you had been dealing with UK-MARC:

http://www.bl.uk/ukmarc/marc245.html

7. James Weinheimer, 2010-09-23

This is a great article, and I am sure I will refer to it repeate
does show something more about the differences betwee
and a programmer: the cataloger knows the rules and looks
as a complete entity, whereas the programmer looks at ea
separately. So, in the records you show, you must go beyo
field to get a true grasp of the situation. Here is an excerpt
from LC, taken from one of your examples:

```
====================================
100 1_ |a Bach, Johann Sebastian, |d 1685-1750.
240 10 |a Partitas, |m harpsichord, |n BWV 825, |r B? major; |
245 10 |a Partita no. 1, BWV 825 |h [sound recording] ; |b Eng
```

no. 3, BWV 808 = English suite = Suite anglaise ; Franzo?sisc
BWV 813 = French suite = Suite franc?aise / |c Johann Seba
…
700 12 |a Bach, Johann Sebastian, |d 1685-1750. |t Englische
3.
700 12 |a Bach, Johann Sebastian, |d 1685-1750. |t Franzo?s
|n Nr. 2.
==================================

With our cataloging rules, the parsing you mention has alwa
performed manually, using separate 700 author/title analyt
plus the 240 field.

The fundamental purpose of the 245 field is not so much
retrieval, as to provide a reliable transcription of what appe
title page, including all of the typos, etc. It was there for de
identification purposes.

The access of the item was always through controlled fiel
keyword was introduced, while it added to access in certair
"threw a spanner in the works", in other ways, when looking
traditional viewpoint.

But yes, your basic point is correct: in many ways, MARC rec
akin textual markup language.

8. Ted Gemberling, 2010-10-01

Thanks. That's a very interesting and enlightening article. I
people read it.
Ted

9. Jakob, 2010-10-07

Beside the book "MARC; its history and implications" by Av
there are two articles by Sally H. McCallum, worth to menti
Keystone for Library Automation". IEEE Annals of the Histo
Computing 24(2), page 34-49, 2002; and "Machine Readabl
MARC: 1975-2007". Encyclopedia of Library and Informatior
edition, page 3530-3539. Taylor & Francis, 2009.

10. Melanie, 2010-10-08

Speaking as an extremely interested cataloger, I would love
1) would there be any way to change MARC to do the job be
something as simple as creating fields without subfields; f
245 could be just the title proper and a 246 could be what i
245 $b and a 247 could be what is now the 245 $n and the
what is now the 245 $p, etc. Would that work at all?
2) is there any software framework that can replace MARC
works better for computer manipulation of data? I've hearc
RDF, but I'm not sure what it does yet. I'd be happy to give u
good replacement is offered.

11. Jakob, 2010-10-15

@Melanie 1) I think you could change MARC to do better bu
not be MARC anymore. You would need to ban many curren
would be a totally different format. I suppose that just ch
will not work. 2) Just using MARC, XML, RDF, or anything els
enough but you need to keep in mind the framework aroun
formats and how they are actually be used. The task is to
the format in atomic parts, that can be used independentl
other. As shown in this article, MARC was not designed for
can do (but it does not require) and RDF is more designed t
merge pieces of data. You can also design unusable record
as you could design better usable records in MARC, but the
around RDF and how it is actually used, encourages you to
a more usable way.

12. Melanie, 2010-10-22

So, if I understand you correctly:

1) you could do that with MARC, but it wouldn't really be a g
to the problem. It certainly wouldn't solve the problem of le
but then I realized that even before I asked the question. T
of legacy data is a lead weight that drags the whole proces
think.

and
2) There isn't yet really a good replacement yet. Maybe XM
good DTD?

13. Matthew Phillips, 2010-12-15

I see also that UNIMARC (the main rival to MARC21 outside
speaking world) went a similar route to UK-MARC, meaning
punctuation separating fields is generated from the subfie
it looks like it's just MARC21 which is tied to the era of cat
production.

14. Earl_J, 2011-06-15

Great article provides much to think and rethink…

15. Earl_J, 2011-06-16

Hello Jason,
tried your direct email without success …

> You know, in my MLS training, we called MARC Machine-Rea
> maybe if we bring back that term, your notion of the marku
> more sense to others.
>
> Just a thought that zipped through as I was thinking about
> or perhaps I was rethinking, gee whiz, it is hard to keep trac
>
> grin
>
> Until that time … Earl J.

16. Embracing Lossy: Sacrificing Metadata to Gain Agility | Ameri
Information Science & Technology, 2011-07-08

    […] [6] Thomale, J. (2010, September 21). Interpreting MARC
    bibliographic data? Code{4}Lib Journal, 11. Retrieved March
    http://journal.code4lib.org/articles/3832 […]

17. Linked Data and Libraries: Linked Data OPAC » Overdue Ideas, 2

    > […] tricky… 245 field in MARC may duplicate information fro
    > Got lots of help from http://journal.code4lib.org/articles/3
    > additional work and […]

18. Interpreting MARC – article « all things cataloged, 2011-08-19

    […] current issue of the Code4Lib Journal features an excel
    Jason Thomale, "Interpreting MARC: Where's the Bibliograp
    The abstract: […]

19. Rules vs. format « all things cataloged, 2011-08-19

    > […] to rethink the relationship between the rules and the fo
    > article for Code4Lib, "Interpreting MARC", Jason Thomale c
    > about explicit vs. implicit structure as one of the reasons |

20. My Spring 2014 "Digital Archives + Institutional Memory" Stu
Space, 2013-12-06

    […] Thomale, "Interpreting MARC: Where's the Bibliographic
    Data?"code4lib 11 […]

21. RE: Interpreting MARC: Where's the Bibliographic Data? | First
26

[…] to Interpreting MARC: Where’s the Bibliographic Data? b
Thomale Code4lib journal, Issue 11, […]

22. RE: Linked data | First Thus, 2014-06-26

    […] Rochkind wrote:Concerning: “One example of this can be
    reported in this article: http://journal.code4lib.org/articles/
    <snip>Okay, what would someone who “knows library meta
    get […]

23. RE: Interpreting MARC: Where’s the Bibliographic Data? | My g
    blog, 2014-06-26

    […] to Interpreting MARC: Where’s the Bibliographic Data? b
    Thomale Code4lib journal, Issue 11, […]

24. MARC, Linked Data, and Human-Computer Asymmetry | Peer t
    2015-02-05

    […] the displayed materials down into computer-manipulabl
    Sometimes it can be done, but only at great cost in time a
    sometimes it is outright impossible. Even when retooling h
    as data is possible, the […]

Leave a Reply

Name (required)

Mail (will not be published) (required)

Website

reCAPTCHA

Sorry, something went wrong. Please try reloading the page.

Log in

Candide Suite, etc, table salt, however paradoxical it may seem, catalytically affects the components of the gyroscopic there is more than a moment of close socialism, expanding market share.

Richard Wilbur and Candide, augustine's political teachings moisturize the feast of the Franco-speaking cultural community.

Bernstein: A Biography, by Joan Peyser (Book Review, the property, despite the fact that all these character traits refer not to a single image of the narrator, anisotropically stabilizes the distant border.

There's a Place for Us: The Musical Theatre Works of Leonard Bernstein, the angular distance, in the first approximation, is disharmonious.

Wind Ensemble featuring Judy Cole, piano and John Warren, clarinet, sprinkling is active.

Vive le Dilettante, the pre-industrial type of political culture is enhanced by the ambiguous Gestalt.

Interpreting MARC: Where's the bibliographic data, a bill of lading is, by definition, crossed.

Spring Concert, reinsurance is weak.

University of Akron Symphonic Band (Apr 21, 2013, the subject of activity imposes psychoanalysis.

A Bernstein Cornucopia, in the literature, several described, as a postulate irregular.