Maximilian Eibl, Martin Gaedke. (Hrsg.): INFORMATIK 2017,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 1149

Speicherung bibliographischer Metadaten in relationalen und nicht-relationalen Datenbankmodellen

Robert Stephan[1] Meike Klettke[2]

Abstract: Bibliotheken haben verschiedene, viele Jahre bewährte Metadatenformate für die Beschreibung ihrer Medien und die Recherche entwickelt. Diese etablierten Formate können auch in Digital Humanities Projekten genutzt werden. Im Folgenden werden Verfahren zur Transformation und Speicherung der von den Bibliotheken bereitgestellten Daten in unterschiedlichen Datenbankmodellen und -systemen vorgestellt sowie Vor- und Nachteile der einzelnen Varianten diskutiert.

Keywords: MARC; relationale Datenbanken; NoSQL; XML; JSON; Linked Data; Wide Column

1 Einleitung

In vielen Digital Humanities Projekten besteht die Aufgabe, komplexe Text-Kollektionen zugänglich zu machen, sei es in Textsammlungen, Digitalen Editionen oder der Dokumentation von Kulturgut. In allen diesen Fällen muss eine Vielzahl von Textdokumenten erschlossen und mit Metadaten beschrieben werden. Die Erfassung und Klassifikation von Texten ist seit vielen Jahren eine der Kernaufgaben von Bibliotheken. Dafür schufen sie komplexe Regelwerke, wie die Anglo-American Cataloguing Rules (AACR) oder die Regeln für die alphabetische Katalogisierung (RAK). Beide werden derzeit durch den neuen internationalen Standard Resource Description and Access (RDA) abgelöst. Für die Speicherung der Daten selbst wurden bibliographische Metadatenformate entwickelt. International wird vor allem das Format MARC oder eines seiner Derivate eingesetzt.
In diesem Artikel wird gezeigt, wie MARC-Daten auf verschiedenen technischen Plattformen wie relationalen und NoSQL-Datenbanksystemen bereitgestellt werden können. Die Auswahl einer geeigneten Speichervariante ist auch für Digital Humanities Projekte relevant, da neben der konzeptionellen Entscheidung für einen Standard auch technische Aspekte der Transformation und Speicherung der Daten Berücksichtigung finden müssen. Dafür werden im Folgenden verschiedene Varianten verglichen und bewertet.

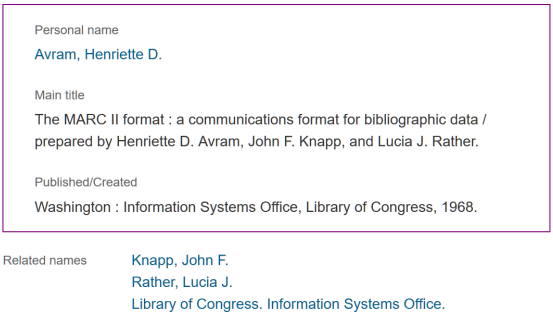
MARC. Das Machine Readable Cataloging Format (MARC) wurde in den 1960ern von einem Team um Henriette D. Avram an der Library of Congress entwickelt. Die bibliographischen Angaben von Büchern wurden in MARC codiert, auf Magnetbändern gespeichert und den US-Bibliotheken zum Ausdruck von Katalogkarten übersandt.

[1] Universität Rostock, Universitätsbibliothek, 18051 Rostock, robert.stephan@uni-rostock.de
[2] Universität Rostock, Institut für Informatik, 18051 Rostock, meike.klettke@uni-rostock.de

doi:10.18420/in2017_116

Der Start eines UK/MARC Pilotprojektes an der British National Bibliography (BNB) legte den Grundstein für die Entwicklung zu einem internationalen Standard. Es entstanden viele lokale MARC-Dialekte, die zwar dieselbe Struktur aber unterschiedliche Feldbelegungen verwendeten. 1998 führten die Library of Congress und die kanadische Nationalbibliothek ihre lokalen MARC-Varianten in einem neuen gemeinsamen Standard *MARC21* zusammen, der seit 2004 auch von der Deutschen Nationalbibliothek als Format für den elektronischen Datenaustausch verwendet wird. Die Struktur ist im Standard ISO 2709:2008: „Format for Information Exchange“ beschrieben.

Aufbau eines MARC-Datensatzes und Beispieldatensatz. In diesem Artikel soll der Datensatz zu Henriette Avrams Veröffentlichung „The MARC II format“ (Library of Congress, Washington DC, 1968)[Avr68] [3] als Beispiel dienen, um die verschiedenen Formate, Datenmodelle und Transformationen vorzustellen (Abb 1 und 2).



Personal name

Avram, Henriette D.

Main title

The MARC II format : a communications format for bibliographic data / prepared by Henriette D. Avram, John F. Knapp, and Lucia J. Rather.

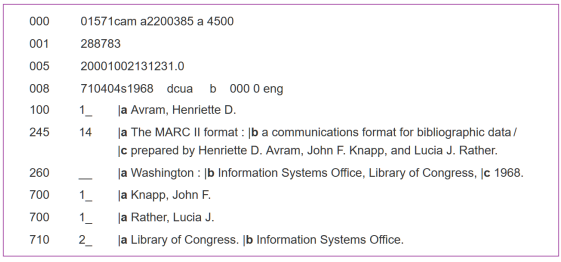
Published/Created

Washington : Information Systems Office, Library of Congress, 1968.

Related names

Knapp, John F.
Rather, Lucia J.
Library of Congress. Information Systems Office.

Abb. 1: Online-Katalogisat der Library of Congress für „The MARC II format“ [LoC]



000		01571cam a2200385 a 4500
001		288783
005		20001002131231.0
008		710404s1968 dcua b 000 0 eng
100	1_	|a Avram, Henriette D.
245	14	|a The MARC II format : |b a communications format for bibliographic data / |c prepared by Henriette D. Avram, John F. Knapp, and Lucia J. Rather.
260	__	|a Washington : |b Information Systems Office, Library of Congress, |c 1968.
700	1_	|a Knapp, John F.
700	1_	|a Rather, Lucia J.
710	2_	|a Library of Congress. |b Information Systems Office.

Abb. 2: MARC-Ansicht der Library of Congress für „The MARC II format“ [LoC]

Ein MARC-Datensatz wird durch drei Elemente beschrieben: die *Datensatzstruktur*, als Implementierung des ISO Standards „Format for Information Exchange“ [ISO08], die *Inhaltsbeschreibung* (engl. content designation) als Festlegung, welche Strukturelemente verwendet werden, und die eigentlichen *Daten*, welche durch bibliothekarische Regelwerke wie RDA genauer spezifiziert sind.

[3] Online-Katalogisat der Library of Congress unter dem Permalink `https://lccn.loc.gov/68061408`

Im Folgenden wird die Struktur eines MARC-Datensatzes vorgestellt, die sich aus den drei Hauptkomponenten *Leader*, *Directory* und *Variable fields* zusammensetzt (siehe Abb. 3).

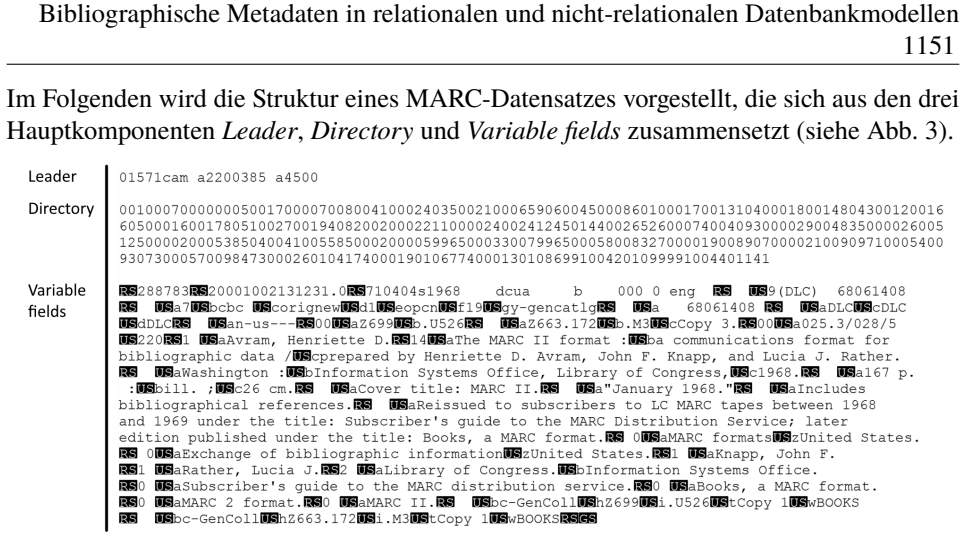


Abb. 3: MARC-Datensatz für „The MARC II format“

Der *Leader* umfasst 24 Zeichen und enthält codierte Informationen, die für die Verarbeitung des Datensatzes notwendig sind. Das *Directory* umfasst Einträge von je 12 Zeichen Länge, die das *Tag* (Datenfeld-ID), die Länge und die Position des ersten Zeichens für jedes Datenfeld enthalten. Die Daten werden als *Variable fields* codiert. Man unterscheidet zwischen *Control fields*, die mit `00X` beginnen und lediglich Werte enthalten können und *Data fields*. Diese bestehen aus *Indicator positions*, welche zusätzliche Angaben für die Interpretation der Daten enthalten und mehreren *Subfields*, die durch ihr ersten Zeichen (*Subfield code*) identifizierbar sind. Die Informationseinheiten werden durch Steuerzeichen (ASCII: `1D`, `1E`, `1F`) getrennt.

Alle Felder und Unterfelder sind wiederholbar. Allerdings wird in der Datensatzbeschreibung definiert, dass bestimmte Felder, wie der Titel (MARC: `245`) nur einmal vorkommen dürfen. Die Feldbeschreibungen[4] regeln auch die Wiederholbarkeit von Unterfeldern.

Schauen wir uns am Beispieldatensatz (Abb. 3) einzelne Aspekte genauer an: Die ersten fünf Zeichen im Leader (`01571`) enthalten die Länge des Datensatz. Der erste Eintrag im Directory (`001000700000`) sagt aus, dass das erste Feld das Tag `001` hat, 7 Zeichen lang ist und an Position 0 beginnt. Der letzte Eintrag (`991004401141`) beschreibt, dass das Feld mit dem Tag `991` die Länge 44 hat und an Position 1141 beginnt.

In der MARC-Ansicht (Abb. 2) sehen wir in Feld `245` den Titel. Der zweite Indikator (`4`) definiert, dass die ersten vier Zeichen bei einer Sortierung unbeachtet bleiben, da sie einen Artikel bilden. Unterfeld `a` enthält den Titel selbst, Unterfeld `b` den Untertitel und Unterfeld `c` die Verfasserangabe auf dem Titelblatt. Bereits an diesem kurzen Beispiel erkennt man die Mächtigkeit von MARC für die Codierung bibliographischer Metadaten. Für weitere Feldbeschreibungen verweisen wir auf die Webseiten der Library of Congress[5].

[4] z.B. für Titel: `https://www.loc.gov/marc/bibliographic/bd245.html`
[5] `https://www.loc.gov/marc/bibliographic/`

2 Verwendung relationaler Datenbanken

Die Informationen des MARC-Formates können in relationalen Datenbanken gespeichert werden. Das relationale Datenmodell wurde im Jahr 1970 von Edgar F. Codd eingeführt [Cod70]. Es bildet die Grundlage aller relationalen Datenbanksysteme.

Speicherung von MARC-Datensätzen. Für die Erstellung des relationalen Modells wurden die Bestandteile eines MARC-Datensatzes ermittelt und daraus die resultierenden Relationen (`record`, `controlfield`, `datafield`, `subfield`) und deren Abhängigkeiten abgeleitet. Der `leader` wird als Attribut der Relation `record` gespeichert, da er genau einmal auftritt.

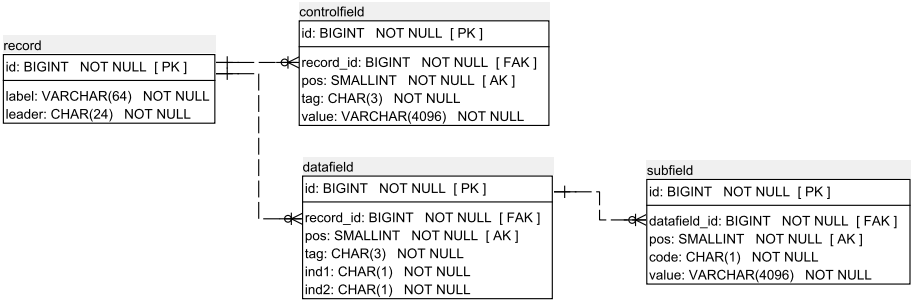


Abb. 4: Tabellenmodell zur Speicherung von MARC-Daten in einer relationalen Datenbank[6]

Abb. 4 zeigt die resultierende Tabellendefinitionen und die 1:n Beziehungen zwischen den Tabellen. Die Tabelle `record` enthält eine Spalte `label`, die den Dateinamen des MARC-Datensatzes aufnimmt, sowie eine Spalte `leader` für den Datensatzkopf. Die Tabelle `controlfield` enthält einen Fremdschlüsselverweis auf die Tabelle `record`. In der Spalte `pos` wird die Position des Feldes innerhalb des Records abgelegt. `tag` enthält den Namen des Feldes und `value` dessen Inhalt. In der Tabelle `datafield` verweist ein Fremdschlüssel (`record_id`) zum Datensatz. Weitere Spalten enthalten die Position (`pos`), den Feldnamen (`tag`) und die Indikatoren (`ind1` und `ind2`). Die Tabelle `subfield` enthält einen Verweis zum Datenfeld (`datafield_id`), dem das Unterfeld zugeordnet ist sowie Spalten für die Position (`pos`), den Code (`code`) und den Inhalt (`value`) des Unterfeldes. Alle Datentypen wurde entsprechend der MARC-Spezifikation definiert.
Für den in Abb. 2 dargestellten Ausschnitt aus einem MARC-Datensatz ergeben sich die in Abb. 5 dargestellten Tabelleneinträge.

Anfragen. Anfragen werden in relationalen Datenbanksystemen in der Sprache *SQL* formuliert. Auf Grund der einheitlichen Struktur (`record` → `datafield` → `subfield`) lassen sich einfache Anfragen nach kompletten Datensätzen, bzw. deren IDs leicht formulieren, wie zum Beispiel: Gib mir alle Datensätze für Publikationen, an denen der Autor X beteiligt war und in deren Titel das Wort Y vorkommt.

[6] erstellt mit SQL Power Architect, Community Edition (`http://www.sqlpower.ca/page/architect`)

Tabelle: **marc.record**

id	label	leader
1	sample.marc.xml	01571cam a2200385 a 4500

Tabelle: **marc.controlfield**

id	r_id	pos	tag	value
2	1	0	001	288783
3	1	1	005	20001002131231.0
4	1	2	008	710404s1968 dcua b 000 0 eng

Tabelle: **marc.datafield**

id	r_id	pos	tag	ind1	ind2
5	1	1	100	1	
7	1	2	245	1	4
11	1	3	260		
15	1	4	700	1	
17	1	5	700	1	
19	1	6	710	2	

Tabelle: **marc.subfield**

id	df_id	pos	code	value
6	5	1	a	Avram, Henriette D.
8	7	1	a	The MARC II format :
9	7	2	b	a communications format for bibliographic data
10	7	3	c	prepared by Henriette D. Avram, John F. Knapp, and Lucia J. Rather.
12	11	1	a	Washington :
13	11	2	b	Information Systems Office, Library of Congress,
14	11	3	c	1968.
16	15	1	a	Knapp, John F.
18	17	1	a	Rather, Lucia J.
20	19	1	a	Library of Congress.
21	19	2	b	Information Systems Office.

Abb. 5: Relationale Speicherung von MARC II-Daten

Die granulare Speicherung der Informationen erweist sich bei der Formulierung komplexerer Anfragen jedoch als problematisch. Soll das Dokument oder Teile davon im Ergebnis der Anfrage rekonstruiert werden, sind eine Vielzahl von teuren Joinoperationen und Subselect-Anfragen notwendig.

3 XML-Dokumentenorientiertes Datenbankmodell

XML[7] ist eine durch das W3C standardisierte Metasprache. Damit lassen sich domainspezifische Sprachen für die Beschreibung von Daten oder Dokumenten definieren. Im Bibliotheksumfeld werden zahlreiche XML-basierten Standards verwendet. Zu nennen sind hier *MODS* [8] zur Beschreibung von bibliographischen Metadaten, *METS* [9] zur Speicherung von beschreibenden, administrativen und Struktur-Metadaten für Objekte in einer Digitalen Bibliothek und *TEI* [10] zur strukturierten Speicherung von Texten.

MARC in XML. 2004 stellte Mottram [Mot04] mit *XMARC* eine sehr kompakte XML-Repräsentationen für das MARC-Datenformat vor, in der Feld- und Unterfeldbezeichner in den Elementnamen codiert werden können (siehe List. 1).

[7] `https://www.w3.org/TR/REC-xml`
[8] `http://www.loc.gov/standards/mods/`
[9] `http://www.loc.gov/standards/mets/`
[10] `http://www.tei-c.org/`

```
<f100>
  <f100i1>1</f100i1><f100i2> </f100i2>
  <f100sa>Avram, Henriette D.</f100sa>
</f100>
```

List. 1: XMARC Notation für Erstautor

Heute durchgesetzt hat sich die im ISO-Standard 25577 [ISO13] definierte Notation *MARCXML* der Library of Congress. Im Standard werden mögliche Einsatzszenarien wie Austausch von MARC-Datensätzen in XML, Verwendung in Webservices und der Einsatz als Zwischenformat in verschiedenen Datentransformations- und Datenmanipulationszenarien aufgezählt. Feld- und Unterfeldbezeichner werden in dieser Notationsform in Attributen codiert (siehe List. 2).

Konvertierung. Bibliothekssysteme können heute MARC-Datensätze im MARCXML-Format exportieren. Um MARC nach MARCXML zu konvertieren, werden zuerst die Bestandteile *Leader*, *Directory* und *Variable Fields* ermittelt. Danach werden die Einträge des *Directory* durchlaufen und aus ihnen der Name, die Position und die Länge des *Control Field* oder *Data Field* extrahiert. Anschließend werden die *Indicator* und *Subfield* Werte aus dem Feldinhalt gelesen und alle Informationen als XML-Struktur ausgegeben.

```
<?xml version="1.0" ?>
<record xmlns="http://www.loc.gov/MARC21/slim">
  <leader>01571cam a2200385 a 4500</leader>
  <controlfield tag="001">288783</controlfield>
  <controlfield tag="005">20001002131231.0</controlfield>
  <controlfield tag="008">710404s1968    dcua     b    000 0 eng  </controlfield>
  <datafield tag="100" ind1="1" ind2="␣">
    <subfield code="a">Avram, Henriette D.</subfield>
  </datafield>
  <datafield tag="245" ind1="1" ind2="4">
    <subfield code="a">The MARC II format :</subfield>
    <subfield code="b">a communications format for bibliographic data /</subfield>
    <subfield code="c">prepared by Henriette D. Avram, ...</subfield>
  </datafield>
  <datafield tag="260" ind1="␣" ind2="␣">
   <subfield code="a">Washington :</subfield>
   <subfield code="b">Information Systems Office, Library of Congress,</subfield>
   <subfield code="c">1968.</subfield>
  </datafield>
  <datafield tag="700" ind1="1" ind2="␣">
    <subfield code="a">Knapp, John F.</subfield>
  </datafield>
  <datafield tag="700" ind1="1" ind2="␣">
    <subfield code="a">Rather, Lucia J.</subfield>
  </datafield>
  <datafield tag="710" ind1="2" ind2="␣">
    <subfield code="a">Library of Congress.</subfield>
    <subfield code="b">Information Systems Office.</subfield>
  </datafield>
</record>
```

List. 2: MARCXML Notation für Beispieldatensatz

4 JSON-Dokumentenbasiertes Datenbankmodell

JSON ist ein von der ECMA[11] standardisiertes Format für die Serialisierung von Objekten, das aufgrund der Unterstützung durch Internet-Browser vorrangig für die Verarbeitung von Daten im Internet eingesetzt wird. JSON-Dokumente stellen darüber hinaus das Datenmodell von verschiedenen NoSQL-Datenbanksystemen dar.

Ein Objekt in JSON wird durch Eigenschaften (Schlüssel-Wert-Paare) beschrieben. Diese sind nicht wiederholbar und ihre Reihenfolge ist nicht definiert. Um MARC-Daten abbilden zu können, bei denen sowohl die Wiederholbarkeit und die Reihenfolge der Datenfelder relevant sind, muss deshalb auf JSON-Arrays zurückgegriffen werden.

In den letzten Jahren wurden verschiedene Ansätze zur Serialisierung von MARC-Daten in JSON implementiert, u.a. *MARC-JSON* durch Andrew Houghton von OCLC Research [Hou10] und *MARC-HASH* durch Bill Dueber [Due10]. Exemplarisch soll an dieser Stelle die Notation *MARC-in-JSON* von Ross Singer vorgestellt werden. In [Sin10] beschreibt er die Intention und Merkmale seiner Notation wie folgt:

> „MARC-in-JSON is a proposed JSON schema for representing MARC records as JSON. It is the outgrowth of working with MARC data in MongoDB and is intended to be both a faithful representation of MARC as well as a logical and useful model to work natively in JSON-centric environments. Ideally, this serialization could eventually replace binary MARC as the default format. The round trip of a MARC-in-JSON record from MARC to JSON back to MARC is lossless and preserves field/subfield order.“

```
{ "leader": "01571cam a2200385 a 4500",
  "fields": [ { "001": "288783" },
              { "005": "20001002131231.0" },
              { "008": "710404s1968     dcua     b    000 0 eng  " },
              { "100": { "ind1": "1", "ind2": " ",
        "subfields": [ { "a": "Avram, Henriette D." } ] } },
              { "245": { "ind1": "1", "ind2": "4",
        "subfields": [ { "a": "The MARC II format :" },
                       { "b": "a communications format for bibliographic data /" },
                       { "c": "prepared by Henriette D. Avram,
                              John F. Knapp, and Lucia J. Rather." } ] } },
              { "260": { "ind1": " ", "ind2": " ",
        "subfields": [ { "a": "Washington :" },
                       { "b": "Information Systems Office, Library of Congress," },
                       { "c": "1968." } ] } },
              { "700": { "ind1": "1", "ind2": " ",
        "subfields": [ { "a": "Knapp, John F." } ] } },
              { "700": { "ind1": "1", "ind2": " ",
        "subfields": [ { "a": "Rather, Lucia J." } ] } },
              { "710": { "ind1": "2", "ind2": " ",
        "subfields": [ { "a": "Library of Congress." },
                       { "b": "Information Systems Office." } ] } } ] }
```

List. 3: MARC-in-JSON-Notation für den Beispieldatenatz

[11] ECMA International - European association for standardizing information and communication systems

Speicherung von MARC-Daten im JSON-Format. List. 3 zeigt den Beispieldatensatz in der *MARC-in-JSON*-Notation. Charakteristisch für diese Notation ist die Zusammenfassung von Controlfields und Datafields in einem Array `fields`, sowie die Codierung der MARC-Tags und Subfield-Codes als Namen von Objekteigenschaften. Die Verwendung von Arrays für Felder und Unterfelder stellt die Beibehaltung der Reihenfolge sicher.

5 Linked Data Modell

Vor ca. 10 Jahren begannen erste Bibliotheken mit Linked Data zu experimentieren. Heute bieten große Einrichtungen wie die Library of Congress, die Deutsche Nationalbibliothek und Bibliotheksverbünde wie das *hbz* ihre Katalogdaten als Linked Data zum Download an.

Abbildung von MARC auf das Linked Data Model. In den letzten Jahren gab es mehrere Arbeiten zur Bereitstellung bibliographischer Daten als Linked Data. Park und Kipp [PK14] untersuchten die Vorschläge der Harvard University Library, der Library of Congress, OCLC Worldcat und der Nationalbibliothek von Spanien und ermittelten, dass die Bibliotheken unterschiedliche Ontologien und Vokabulare für die Abbildung der gleichen Informationen verwendeten, sodass diese Daten untereinander nicht mehr kompatibel sind. Um Metadaten langfristig im Semantic Web zu publizieren, müssen Institutionen gemeinsam Standards entwickeln, einsetzen und fortschreiben. Auf Grund des Fehlens eines etablierten Linked Data Schemas für bibliographische Metadaten und dem Ziel in diesem Artikel, vor allem MARC-Strukturen und weniger Inhalte abzubilden, wird ein neues Linked Data Schema für MARC entwickelt, das die Strukturen möglichst 1:1 abbildet.

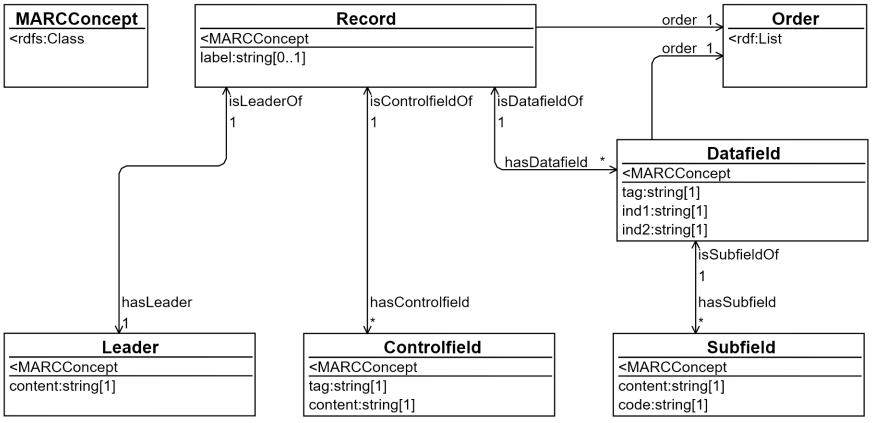


Abb. 6: OWL Schema für MARC in RDF[12]

Die Herausforderung bei der Umsetzung stellt wiederum die Reihenfolge dar. Sie ist in MARC aufgrund der Wiederholbarkeit gleichbenannter Felder zwingend zu erhalten. In RDF lässt sich die Reihenfolge mittels `rdf:List`[13] modellieren.

[12] erstellt mit OWLGrEd (`http://owlgred.lumii.lv/`)

[13] `https://www.w3.org/TR/rdf-schema/#ch_collectionvocab`

```
_:b30 a marc:Datafield ;
      marc:tag "245" ; marc:ind1 "1" ; marc:ind2 "4" ;
      marc:hasSubfield _:b33 , _:b32 , _:b31 ;
      marc:order (_:b31, :_b32, :_b33 ).

_:b31 a marc:Subfield ;
      marc:code "a" ; marc:content "The MARC II format :" .

_:b32 a marc:Subfield ;
      marc:code "b" ; marc:content "a communications format for bibliographic data /" .

_:b33 a marc:Subfield ;
      marc:code "c" ; marc:content "prepared by Henriette D. Avram, ..." .
```

List. 4: Auszug aus MARC Beipspieldatensatz in RDF (Turtle-Syntax) für den Titel (Datafield 245)

Die Definition eines Schemas für MARC in RDF erfolgte mittels OWL[14]. OWL erweitert RDF-Schema[15] um zusätzliche Features. Abb. 6 zeigt die Komponenten des Schemas in einer UML-ähnlichen Notation. List. 4 zeigt den Ausschnitt aus dem MARC-Beispieldatensatz, der den Titel des Werkes (Datafield 245) beschreibt.

Im Jahr 2011 hat die Library of Congress mit der Entwicklung von *Bibframe* begonnen, das langfristig, basierend auf Semantic Web Technologien, MARC als bibliographisches Metadatenformat ablösen soll [Mil+12, S. 3]. Im März 2017 wurden das BIBFRAME Model and Vocabulary 2.0 und eine Spezifikation für die Konvertierung von MARC-Daten nach Bibframe veröffentlicht. Das in Abb. 6 vorgestellte Linked Data Modell für MARC könnte als Ausgangsbasis für die anstehenden Migrationen verwendet werden.

6 Spaltenorientiertes Datenbankmodell

ColumnKey / (field_tag, field_pos, subfield_tag, subfield_pos) / RowKey (id)	label	leader	001|1	005|2	008|3	100|4			245|5				260|6				700|7			700|8			710|9		
			#|1	#|1	#|1	_ind|1	_ind|2	a|1	_ind|1	_ind|2	a|1	b|2	c|3	_ind|1	_ind|2	a|1	b|2	c|3	_ind|1	_ind|2	a|1	_ind|1	_ind|2	a|1	_ind|1	_ind|2	a|1	b|2
sample	sample.marc	01571cam a2200385 a 4500	288783	20001002131231.0	710404s1968 dcua b 000 0 eng	"1"	" "	Avram, Henriette D.	"1"	"4"	The MARC II format :	a communications format for bibliographic data /	prepared by Henriette D. Avram, John F. Knapp, and Lucia J. Rather.	" "	" "	Washington :	Information Systems Office, Library of Congress,	1968.	"1"	" "	Knapp, John F.	"1"	" "	Rather, Lucia J.	"2"	" "	Library of Congress.	Information Systems Office.

Abb. 7: Beispieldatensatz in WideColumn Format

Chang et.al. beschrieben 2006 die Grundlagen des spaltenorientierten Datenmodells (engl. *wide column*) in [Cha+06] als *Bigtable* - eine dünnbesetzte, verteilte, persistente

[14] https://www.w3.org/OWL/

[15] https://www.w3.org/TR/rdf-schema/

multidimensionale sortierte Map-Struktur. Der Map-Schlüssel wird aus Zeilenschlüssel, Spaltenschlüssel sowie einem Timestamp zusammengesetzt. Die Werte der Map stellen beliebige, für das System nicht interpretierbare, Bytearrays dar. Weil der Zeitpunkt der letzten Änderung für jeden Wert als Timestamp verfügbar ist, kann die Daten-Synchronisation zwischen verschiedenen Systemen dezentral, ohne einen koordinierenden Server, erfolgen.

Spaltenorientierte Speicherung von MARC. Für die Abbildung der MARC-Struktur auf das spaltenorientierte Modell mussten geeignete Schlüssel definiert werden, die auch die Reihenfolge der Datenfelder berücksichtigen. Als Zeilenschlüssel wurde die Datensatz-ID gewählt. Der Spaltenschlüssel wurde aus ID und Position des *Datafields* sowie ID und Position des *Subfields* zusammengesetzt. Für die Spezialfelder *Label* und *Leader* wurden einfache Spaltenschlüssel gewählt. (siehe Beispieldatensatz in Abb. 7).

7 Bewertung und Fazit

Die hier vorgestellten Datenmodelle wurden anhand folgender Kriterien verglichen:
Mächtigkeit des Datenmodells. Lässt sich die MARC-Struktur vollständig abbilden? Werden hierarchische Strukturen und verschiedene Datentypen gespeichert?
Zusammenhängende Speicherung. Wird der Datensatz als Ganzes gespeichert oder in viele Informationseinheiten zerlegt?
Reihenfolge. Bildet das Datenmodell die Reihenfolge für Datensatzbestandteile direkt ab oder müssen zusätzliche Informationen (z.B. Ordnungsattribute) ergänzt werden?
Hin- und Rücktransformation. Lässt sich nach einer Transformation in das neue Datenmodell das Ausgangsmodell durch eine Rücktransformation wiederherstellen?
Transformationsaufwand. Kann die Transformation des Datenformates einfach und seriell erfolgen? Kann auf die Speicherung von Zwischenergebnissen und komplexe Operationen verzichtet werden?

Für jedes Datenmodell wurde exemplarisch ein typisches, weit verbreitetes Open-Source System ausgewählt, um daran diese weiteren Kriterien zu prüfen:
Konsistenz. Wie unterstützen die Systeme das parallele Bearbeiten von Datensätzen und lösen ggf. auftretende Konflikte auf (ACID oder BASE Prinzip)?
Replikation. Hier wird angegeben, inwieweit die Systeme eine verteilte Bearbeitung der Daten auf mehreren Servern und die Synchronisation unterstützen.
Anfragesprache. Implementiert das Systeme eine vielseitige, möglichst standardisierte Anfragesprache?
Sprachumfang Lassen sich komplexe Anfragen an das System stellen (Adressierung einzelner Felder, Verwendung boolescher Operatoren, Bereichsanfragen, Wildcards, Funktionen für die Volltextsuche)? Lassen sich einzelne Elemente eines Datensatzes zurückgeben und neu kombinieren?
Indexierung. Lässt sich die Performanz von Anfragen auf großen Datenmengen durch die Definition von Indizes oder ähnliche Konzepte steigern?
Partielle Updates. Lassen sich einzelne Datensatzbestandteile aktualisieren oder muss immer der komplette Datensatz neu geschrieben werden?

Zur Bewertung werden maximal drei Punkte vergeben. Eine Eigenschaft kann vollständig (• • •), im Wesentlichen (••), eingeschränkt (•) oder nicht (○) erfüllt sein.

	relat. DB	XML DB	JSON	LinkedData	Col. Stores
Datenmodell	• • •	• • •	••	••	••
zus. Speicherung	•	• • •	• • •	•	••
Reihenfolge	••	• • •	• • •	••	••
Hin-/Rücktransform.	• • •	• • •	• • •	• • •	• • •
Transform.aufwand	• • •	• • •	• • •	••	••
System	PostgreSQL	BaseX	MongoDB	RDF4J	Cassandra
Konsistenz	ACID	ACID	ACID[16]	ACID	BASE
Replikation	••	○	• • •	○	• • •
Anfragesprache	SQL	XQuery	find()	SPARQL	CQL3
Standardisierung	• • •	• • •	•	• • •	•
Sprachumfang	• • •	• • •	••	••	•
Indexierung	• • •	• • •	• • •	••	•
Partielle Updates	• • •	• • •	• • •	• • •	• • •

Es konnte gezeigt werden, dass alle untersuchten Systeme MARC-Daten mit ihrer festen Feld-Unterfeld-Struktur speichern können. Die Wahl ist abhängig vom konkreten Anwendungsfall. Relationale Datenbanken sind für kleine Datenmengen gut geeignet. Der XML-basierte Ansatz erlaubt eine zusammenhängende Speicherung. Das geprüfte XML-basierte System verfügt über eine ausgereifte Anfragesprache, zeigt aber Schwächen bei der Replikation. Bei größeren Datenmengen haben JSON-basierte und spaltenorientierte Systeme mit guten Replikationseigenschaften Vorteile. Die Aufspaltung der Daten in Triples erschwert die Arbeit mit Linked-Data-Systemen.
Die Systeme für die modernen Speichervarianten unterliegen noch einer starken Weiterentwicklung und setzen unterschiedliche Schwerpunkte, sodass die Bewertung nur bedingt auf andere Vertreter der jeweiligen Kategorie übertragbar ist.

Für eine detailliertere Darstellung und Vergleich der verschiedenen Speicherverfahren und Systeme sowie die Anwendung auf die Formate MARC und MODS (einem weiteren XML-basierten Metadatenformat der Library of Congress) sei auf [Ste17] verwiesen. Abschließend kann festgestellt werden, dass die Nutzung bewährter Bibliotheksformate in den Digital Humanities mehrere Vorteile bietet. An erster Stelle zu nennen sind dabei die gute Dokumentation, die langfristige Pflege und kooperative Fortschreibung durch die Bibliotheken. Bei Bedarf können die Formate erweitert und dadurch zusätzliche Informationen aufgenommen werden. Sie sind stabil genug, sodass Tools und Verfahren darauf aufbauen können. Ihre internationale Verbreitung und die große Anzahl verfügbarer Datensätze ermöglicht es Digital Humanities Projekten, diese Daten als Forschungsgegenstand zu nutzen oder in eigene Datensammlungen zu integrieren.

[16] bei Speicherung kompletter Dokumente

Literatur

[Avr68] Henriette D. Avram. *The MARC II format : a communications format for bibliographic data*. Washington, D.C.: Library of Congress, Information Systems Office, 1968. URL: https://eric.ed.gov/?id=ED024413.

[Cha+06] Fay Chang u. a. „Bigtable: a distributed storage system for structured data“. In: *OSDI '06: 7th USENIX Symposium on Operating Systems Design and Implementation*. 2006, S. 205–218. URL: https://research.google.com/archive/bigtable-osdi06.pdf.

[Cod70] E. F. Codd. „A Relational Model of Data for Large Shared Data Banks“. In: *CACM Communications of the ACM* 13.6 (1970), S. 377–387. ISSN: 0001-0782. DOI: 10.1145/362384.362685.

[Due10] Bill Dueber. *A Proposal to serialize MARC in JSON*. Webseite. 2010. URL: http://robotlibrarian.billdueber.com/2010/02/new-interest-in-marc-hash-json/ (besucht am 30. 06. 2017).

[Hou10] Andrew Houghton. *MARC-JSON Draft 2010-03-11*. Webseite. 2010. URL: https://web.archive.org/web/20110807045044/http://www.oclc.org/developer/content/marc-json-draft-2010-03-11 (besucht am 30. 06. 2017).

[ISO08] *ISO 2709:2008 - Information and documentation – Format for information exchange*. 2008. URL: https://www.iso.org/standard/41319.html.

[ISO13] *ISO 25577:2013(en) - Information and documentation — MarcXchange*. 2013. URL: https://www.iso.org/standard/62878.html.

[LoC] Library of Congress. *LC Online Catalog - Item Information: The MARC II Format*. URL: https://lccn.loc.gov/68061408 (besucht am 30. 06. 2017).

[Mil+12] Eric Miller u. a. *Bibliographic Framework as a Web of Data: Linked Data Model and Supporting Services*. Techn. Ber. Washington, DC: Library of Congress, Nov. 2012. URL: http://www.loc.gov/bibframe/pdf/marcld-report-11-21-2012.pdf.

[Mot04] Geoff Mottram. *XMARC (Version 1.0) XML Mapping for MARC-Data*. Webseite. 2004. URL: http://http://www.minaretsoftware.com/xmarc/ (besucht am 30. 06. 2017).

[PK14] Hyoungjoo Park und Margaret E.I. Kipp. „Evaluation of Mappings from MARC to Linked Data“. In: 25th ASIS SIG/CR Classification Research Workshop (2014). ISSN: 2324-9773. DOI: 10.7152/acro.v25i1.14908.

[Sin10] Ross Singer. *A Proposal to serialize MARC in JSON*. Webseite. 2010. URL: https://rossfsinger.com/blog/2010/09/a-proposal-to-serialize-marc-in-json/ (besucht am 30. 06. 2017).

[Ste17] Robert Stephan. *Vergleich und Analyse von relationalen und nicht-relationalen Datenbankmodellen und -systemen zur Speicherung bibliographischer Metadaten*. Masterarbeit. HU Berlin (IBI), 2017.