Master-Thesis

zur Erlangung des akademischen Grades
Master

Technische Hochschule Wildau

Fachbereich Wirtschaft, Informatik, Recht

Studiengang Bibliotheksinformatik (M.Sc.)

Thema (deutsch): ETL-Prozesse für bibliothekarische Metadaten: Die Migration lokaler Katalogisate im GBV

Thema (englisch): ETL Processes for Library Metadata: The Migration of Local Records at GBV Libraries

Autor: Ursula Klute
Matrikelnr.: 050029398
Seminargruppe: BIM / 15

Betreuer: Stefan Lohrum
Gutachter: Andreas Krausz

Reg.-Nr.: BIM 523/17

Eingereicht am: 20.05.2018

Abstract

Bei allen IT-gestützten Anwendungen spielt die Datenhaltung eine zentrale Rolle. Verfahren für eine Migration der gespeicherten Daten werden aus unterschiedlichen Gründen benötigt; dazu zählen technische Weiterentwicklungen der verwendeten Hard- oder Software, ein Wechsel des Anbieters der genutzten Anwendung sowie der Wunsch nach Zusammenführung und Vereinheitlichung von Daten aus unterschiedlichen Quellen, z.B. in einem Data Warehouse. Hierbei kommen ETL-Prozesse zur Anwendung, die die Schritte Datenextraktion, Datentransformation und das Laden der Daten in die Zielumgebung umfassen.

Im bibliothekarischen Umfeld werden ETL-Verfahren u. a. projektbasiert bei einem Wechsel des Bibliotheksmanagementsystems praktiziert; als Routineverfahren ist in zahlreichen Bibliotheken die Verarbeitung von Metadaten aus heterogenen Quellen implementiert. Dabei kommt in Bibliotheken der Analyse und Bereinigung von Metadaten eine zentrale Bedeutung im Migrationsprozess zu.

Diese Grundsätze werden auf ein Migrationsprojekt für bibliografische Metadaten im Kontext von mehr als 100 Verbundbibliotheken im Gemeinsamen Bibliotheksverbund (GBV) angewendet. Im Rahmen des Projekts sollen sog. lokale Katalogisate, die bisher außerhalb der kooperativen Verbundkatalogisierung im jeweiligen lokalen Bibliotheksmanagementsystem LBS erfasst wurden, in die zentrale Verbunddatenbank überführt werden. Als Herausforderung stellt sich dabei die große Diversität und Quantität der zu migrierenden Daten dar. Zur Durchführung des Migrationsprojekts durch die Verbundzentrale des GBV (VZG) ist daher ein Verfahren erforderlich, das der Daten-Heterogenität und den technischen Gegebenheiten Rechnung trägt. Die vorliegende Arbeit untersucht, inwieweit sich die grundlegenden ETL-Prozesse auf dieses Projekt abbilden lassen.

Aufgrund der proprietären Form der Speicherung von Metadaten in LBS-Datenbanken lassen sich standardisierte Verfahren und Werkzeuge zur Extraktion von Daten nicht anwenden. Für den Zugriff auf LBS-Titeldaten werden im Rahmen der Arbeit daher die unterschiedlichen spezifischen Verfahren für den Datenzugriff vorgestellt und für den Anwendungszweck adaptiert.

Es wird exemplarisch gezeigt, wie mit Hilfe der erstellten Tools die Daten analysiert und ggf. vor der eigentlichen Umsetzung bereinigt werden können.

Data management plays a key role in software applications. At some point in the application's lifecycle it may become necessary to migrate the data to a new environment, e.g. because the application is moved to an advanced hardware or software platform, or when separate data sources are consolidated and merged, e.g. in a data warehouse. The procedures applied to data migration are referred to as ETL (data extraction, data transformation and loading the data into the target environment).

In libraries, ETL processes are usually implemented when a legacy library management system is migrated to a different vendor or in order to merge heterogeneous metadata from various sources into a joint search index. Especially with regard to bibliographic metadata, it is of paramount importance to perform a thorough data analysis and data cleansing before starting the actual migration process.

This thesis applies the baseline concepts of ETL processes to real life metadata migration project for more than 100 libraries in the Common Library Network (GBV). Library specific bibliographic records which are held only in the local library database LBS are to be transferred to the central union database (CBS). The high variety and quantity of the data to be migrated poses a major challenge, so the implementation of the migration project by the GBV's head office (VZG) has to take account both of the technical implementation and the complex data.

Due to the proprietary form of storing metadata records in the LBS database, standardized methods and tools for extracting data cannot be used. Therefore to access the LBS data, specific procedures are discussed and adapted for the application purpose. This thesis describes how, with the help of these tools, the data can be analyzed and, if necessary, cleaned up before the actual data transfer.

Inhalt

Abbildungsverzeichnis

Tabellenverzeichnis

Verzeichnis von Datenbeispielen

Abkürzungen

ACQ3 / ACQ4	Aquisitie (niederländ.); Erwerbungsmodul von LBS3 bzw. LBS4
AG KVA	Arbeitsgruppe Kooperative Verbundanwendungen der AG Verbundsysteme
API	Application Programming Interface; Schnittstelle zur Anwendungsprogrammierung
ASEQ	Aleph-SEQuential; auf MAB2 basierendes Aleph-Internformat
B3Kat	Gemeinsamer Verbundkatalog des Bibliotheksverbundes Bayern (BVB) und des Kooperativen Bibliotheksverbundes Berlin-Brandenburg (KOBV)
BABSY	Bochumer Ausleih- und Verbuchungs-System; Eigenentwicklung (2008 abgelöst)
BASE	Bielefeld Academic Search Engine
BK	Basisklassifikation; von niederländischen Bibliotheken entwickelte hierarchische Dezimalklassifikation
BMS	Bibliotheksmanagementsystem
BSZ	Bibliotheksservice-Zentrum (Sitz: Konstanz); Verbundzentrale des Südwestdeutschen Bibliotheksverbundes SWB
BVB	Bibliotheksverbund Bayern (Sitz: München)
CAT4	Katalogisierungsmodul von LBS4
CBS	Centraal Bibliotheek System; (Verbund-)Katalogisierungsdatenbank der PICA-Verbünde sowie DNB und ZDB
CC0	Bedingungslose Creative-Commons-Lizenz
CPAN	Comprehensive Perl Archive Network; Online-Repository für Perl-Module
CRM	Customer-Relationship-Management; Informationssystem zur Kundenpflege
CSV	Comma-separated values; Dateiformat
D-A-CH	Akronym für den deutschen Sprachraum (Deutschland, Österreich, Schweiz)
DAIA	Document Availability Information API; Schnittstelle für Verfügbarkeitsinformationen
DB	Datenbank
DBI	DataBase Interface; Datenbankschnittstelle
DBMS	Datenbankmanagementsystem
DDB	Deutsche Digitale Bibliothek; Portal für Metadaten aus deutschen Kultur- und Wissenschaftseinrichtungen
DDC	Dewey Decimal Classification
DFG	Deutsche Forschungsgemeinschaft
DIN	Deutsches Institut für Normung
DNB	Deutsche Nationalbibliothek (Sitz: Frankfurt a. M. und Leipzig)
DOAJ	Directory of Open Access Journals
DOI	Digital Object Identifier
DTD	Document Type Definition; Validationsverfahren für XML-Dokumente
DW	Data Warehouse
ekz	Dienstleistungsunternehmen für Bibliotheken (Sitz: Reutlingen)
EPN	Exemplar-Produktionsnummer (10-stelliger String)
ERM	Electronic Resource Management; System zur Verwaltung elektronischer Ressourcen
ETH	Eidgenössische Technische Hochschule
ETL	Extract – Transform – Load
FID	Fachinformationsdienst
fno	File number; Nummer des (logischen) Datenbestandes einer Bibliothek in einer LBS-Datenbank
FOLIO	Akronym für The Future of Libraries is Open; Open-Source-Bibliothekssystem-Plattform (in Entwicklung)

FTP	File Transfer Protocol
GBV	Gemeinsamer Bibliotheksverbund; Verbundzentrale VZG (Sitz: Göttingen)
GND	Gemeinsame Normdatei; für Personen, Körperschaften, Kongresse, Geografika, Sachschlagwörter und Werktitel
GOKb	The Global Open Knowledgebase; kooperativ gepflegte Knowledgebase für elektronische Ressourcen
GOSSIP	Good Old Server for Standard Interchange Protocol; Schnittstelle zur Ablösung von SIP2 mit erweiterten Funktionen
GREL	General Refine Expression Language; Transformationssprache in OpenRefine
GVI	Gemeinsamer Verbünde-Index
GVK	Gemeinsamer Verbundkatalog (des GBV)
hbz	Hochschulbibliothekszentrum des Landes Nordrhein-Westfalen (Sitz: Köln)
HeBIS	Hessischer Bibliotheksverbund (Sitz: Frankfurt a. M.)
ID	Identifier; Kennzeichen zur eindeutigen Identifizierung eines Objekts oder Datensatzes
IFLA	International Federation of Library Associations and Institutions (Sitz: Den Haag)
IKT	Index Key Type; interne Nummer des Index, die extern durch die Mnemo-Bezeichnung des Suchschlüssels repräsentiert wird. Beispiel (GBV): ALL=1016, TIT=4
IMD	Zusammenfassende Bezeichnung für die RDA-Standardelemente Inhaltstyp, Medientyp, Datenträgertyp
IPN	Internal Production Number; PPN ohne Prüfziffer; 32-Bit-Integer-Feld
IQ	Informationsqualität
ISBD	International Standard Bibliographic Description; Standard zur Beschreibung von Bibliotheksmaterialien
ISIL	International Standard Identifier for Libraries and Related Organizations; internationales System für Bibliothekssigel
ISO	International Organization for Standardization
ISSN / e-ISSN	International Standard Serial Number; zuweilen als e-ISSN zur Kennzeichnung der elektronischen Ausgabe eines fortlaufenden Sammelwerks
JDBC	Java Database Connectivity; Datenbankschnittstelle für Java-Anwendungen
JISC	Joint Information Systems Committee; britische Dienstleistungsorganisation für Bibliotheken
JSON	JavaScript Object Notation; Datenformat
JSTOR	Journal STORage; Anbieter eines kostenpflichtigen Online-Archivs mit älteren Ausgaben ausgewählter Fachzeitschriften
KBART	Knowledge Bases And Related Tools; Standardformat für Verlagsdaten
K-Int	Knowledge Integration; Softwarefirma (Sitz: Sheffield, UK) mit Schwerpunkt Entwicklungen im Bibliothekskontext
KOBV	Kooperativer Bibliotheksverbund Berlin-Brandenburg (Sitz: Berlin)
LAS:eR	Lizenz-Administrationssystem für e-Ressourcen; Projekt zur Entwicklung eines ERM-Systems
LBS / LBS3 / LBS4	Lokaal Bibliotheek System (niederländ.); u. A. in den Bibliotheken der Bibliotheksverbünde GBV und HeBIS verwendetes lokales Bibliotheksmanagementsystem von OCLC
LDAP	Lightweight Directory Access Protocol; auch genutzt für die Verzeichnisdienste selbst
LOAN4	Schnittstelle zur Anbindung des OPACs an LBS4-Ausleihfunktionen
LoC	Library of Congress (Sitz: Washington, DC)
M+M	Match and Merge; Verfahren zur Identifizierung von Datensatzdubletten
MAB / MAB2	Maschinelles Austauschformat für Bibliotheken; im deutschen Sprachraum genutztes bibliografisches Format für den Datenaustausch

MARC / MARC 21 / MARCXML	MAchine-Readable Cataloging; internationales bibliothekarisches Datenformat, vor allem für den Datenaustausch
MODS	Metadata Object Description Schema; XML-basierter Metadatenstandard
MTM	Mehrteilige Monografien; RDA-Bezeichnung für mehrbändige Werke
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting; Protokoll zum Einsammeln von Metadaten
OBVSG	Österreichische Bibliothekenverbund und Service GmbH; Verbundzentrale des OBV (Sitz: Wien)
OCLC	Online Computer Library Center; Anbieter der Bibliotheksmanagementsysteme LBS und WMS
OCN	OCLC Control Number; Identifier eines Datensatzes im WorldCat
ODBC	Open Database Connectivity; Datenbankschnittstelle für den Zugriff per SQL
OLAP	Online Analytical Processing
OLC	Online Contents; bibliografische Datenbank mit Aufsätzen aus wissenschaftlichen und anwendungsorientierten Zeitschriften
OLE	Open Library Environment; Gemeinschaftsprojekt von Bibliotheken; Ziel: Einsatz von Open-Source-Software
OLF	Open Library Foundation; gemeinnützige Organisation zur Förderung von Open-Source-Projekten für Bibliotheken mit den Projekten FOLIO, OLE und GOKb
OLTP	Online Transaction Processing
ONIX	ONline Information eXchange; XML-basiertes Austauschformat für Verlage und Buchhandel
OPAC	Online Public Access Catalogue
OPC4	OPAC-Modul des Bibliothekssystems LBS
OUM	Online Update Mechanism; Verfahren für das Replizieren von CBS-Daten ins LBS
OUS3 / OUS4	Opslag en Uitleen Systeem; Ausleihmodul von LBS3 bzw. LBS4
OWC	Online Werkcatalogus (niederländ.); Katalogisierungsmodul von LBS3
PI	Preußische Instruktionen; bibliothekarisches Regelwerk zur Katalogisierung (überholt)
Pica / PICA	Project voor GeIntegreerde Catalogus Automatisering; ursprünglich „Stichting Pica“ (1969), 2007 vollständig übernommen von OCLC; Anbieter des im GBV genutzten Verbundsystems und Lokalsystems
PICA3 / PICA+ / PICA XML	Erfassungsformat / Speicherformat / Austauschformat in PICA-Datenbanken
POLLUX	Fachinformationsdienst für die Politikwissenschaft
PPN	(external) Pica Production Number; Identifier einer PICA-Datenbank (10-stelliger String)
PSI	Pica Search and Index; Software zur Indexierung, genutzt für Webkataloge von PICA-Datenbanken
RAK	Regeln für die alphabetische Katalogisierung (überholt)
RDA	Internationales Regelwerk zur Katalogisierung in Bibliotheken
RDF	Resource Description Framework; Metadatenkonzept des Semantic Web
REST	Representational State Transfer; zustandsloses Datenübertragungsprotokoll
RSWK	Regeln für den Schlagwortkatalog; im D-A-CH-Raum genutzt
SaaS	Software as a Service; Vertriebsmodell für Software in der Cloud
SLUB	Sächsische Landesbibliothek – Staats- und Universitätsbibliothek
SOA	Serviceorientierte Architektur; Softwareparadigma, das sich an Geschäftsprozessen orientiert
SQL	Structured Query Language; Datenbankabfragesprache

SRU	Search/Retrieve via URL; technischer Standard für HTTP-basierte Suchanfragen für bibliografische Datenbanken mit Suchindizes und Suchbegriffen; Weiterentwicklung des Z39.50-Protokolls
subito	Dokumentlieferdienst für wissenschaftliche Bibliotheken
SuUB	Staats- und Universitätsbibliothek
SWB	Südwestverbund; Verbundzentrale BSZ (Sitz: Konstanz)
TSV	Tab-Separated Values; Dateiformat
unAPI	REST-basierte Schnittstelle zum Abruf einzelner Datensätze
URICA	Integriertes Bibliothekssystem; ursprünglicher Anbieter: McDonnell Douglas Information Systems (MDIS)
URL / URN	Uniform Resource Locator / Uniform Resource Name
VM	Virtuelle Maschine
VZG	Verbundzentrale des GBV (Sitz: Göttingen)
W3C	World Wide Web Consortium
WinIBW	Katalogisierungs-Client für PICA-Datenbanken mit integrierten JavaScript- oder VBScript-Funktionen zur Anwenderunterstützung
WMS	WorldShare Management Services; cloudbasiertes Bibliotheksmanagementsystem (Anbieter: OCLC)
WSUL	Washington State University Libraries
XML	Extensible Markup Language
XSD	XML Schema Definition; Validationsverfahren für XML-Dokumente
Z39.50	Verbindungsorientiertes Netzwerkprotokoll für die Suche in bibliographischen Informationssystemen
ZBW	Zentrale Bibliothek für Wirtschaftswissenschaften (frühere Bezeichnung); heute: ZBW - Leibniz-Informationszentrum Wirtschaft
ZDB	Zeitschriftendatenbank; zentrale Datenbank für Titel- und Besitznachweise fortlaufender Sammelwerke in Deutschland

1 Einleitung / Motivation

Seit Bibliotheken in den 90er Jahren ihre analogen Kartenkataloge digitalisiert haben, sind mehr als 20 Jahre vergangen. Seitdem hat sich wie in anderen Wirtschaftszweigen zwangsläufig die Notwendigkeit ergeben, die digitalen Daten und Prozesse auf von veralteten Plattformen auf neue Systeme zu migrieren. Die vorliegende Arbeit stellt die grundlegenden Prozesse bei der Migration von Daten vor und untersucht die Frage, inwieweit sich diese Standards und Best Practices auf das Management von bibliografischen Metadaten in Bibliotheken übertragen lassen.

Bei allen IT-gestützten Anwendungen kommt der Datenhaltung eine zentrale Bedeutung zu. Insbesondere bei großen Datenbeständen ist die Speicherung in einem datenbankgestützten System das übliche Vorgehen. Regelmäßig entsteht die Notwendigkeit, die im Quellsystem erzeugten Daten in ein anderes System zu überführen, z. B. anlässlich eines Systemwechsels oder um die Daten in einem anderen Kontext nachzunutzen. Hierbei kommen sog. ETL-Prozesse zur Anwendung, die die Schritte Datenextraktion aus dem Quellsystem, Datentransformation und das Laden der Daten in die Zielumgebung umfassen.

Der Begriff ETL stammt ursprünglich aus dem Data-Warehouse-Umfeld; daher werden einleitend anhand von Data-Warehouse-Anwendungen, auch im bibliothekarischen Kontext, die wesentlichen Merkmale von ETL-Prozessen herausgearbeitet. Diese Arbeit zeigt, dass ETL-Prozesse gleichermaßen im breiten Spektrum des bibliothekarischen Datenmanagements anzutreffen sind. Aus diesem Grund wird hier der Begriff ETL verallgemeinernd für Verfahren der Datenbereitstellung und -integration verwendet.[1]

Essenziell für das Verständnis der Abläufe im Metadatenmanagement ist die Kenntnis von Datenformaten aus dem Bibliotheksumfeld. Bibliografische Metadaten sind in starkem Maße von den Besonderheiten der bibliothekarischen Datenformate und Regelwerke abhängig. Daher werden vor allem MARC- und PICA-Formate eingehend analysiert.

Im Datenmanagement ist die Qualität der zu verarbeitenden Daten von wesentlicher Bedeutung. Deshalb wird neben einer generellen Auseinandersetzung mit Informations- und Datenqualität der Themenkomplex Datenanalyse und -bereinigung im bibliothekarischen Kontext umfassend erörtert. In diesem Zusammenhang werden exemplarisch Verfahren zur Identifizierung und Behandlung von Dubletten vorgestellt.

Ein weiteres Anwendungsszenario für den Einsatz von ETL-Prozessen im Bibliothekskontext ist die Integration von Metadaten aus heterogenen Quellen; hierzu werden exemplarische Anwendungen genauer betrachtet.

Dabei spielt die Nutzung von geeigneten Tools für den Umgang mit bibliografischen Metadaten eine große Rolle. Zu diesem Zweck erfolgt eine Evaluation verbreiteter Software aus dem Kontext bibliothekarischer Anwendungen, insbesondere im Hinblick auf die Eignung als Tool für die Nutzung in dem praktischen Anwendungsfall der Migration von lokalen Katalogisaten.

[1] Vgl. Rossak 2013, S. 37.

Bei Migrationen im Bibliotheksbereich handelt es sich in der Regel um komplexe Vorhaben. Hier stellt sich die Frage, inwieweit solche Migrationen (branchen)spezifischen Bedingungen unterliegen. Daher ist es unabdingbar, sich vorab mit grundlegenden Prinzipien von Migrationsvorhaben und exemplarischen Migrationsstrategien zu beschäftigen. Dabei werden die Gründe für die Migration von Informationssystemen beleuchtet und insbesondere die strukturellen Abläufe hinsichtlich der Anwendbarkeit auf bibliothekarische Migrationsvorhaben analysiert.

Bibliotheken, die vor der Ablösung ihres alten Bibliotheksmanagementsystems stehen, haben mehrere Optionen zur Auswahl eines Nachfolgesystems. Es wird untersucht, inwieweit die Besonderheiten der Systeme Auswirkungen auf den Ablauf der Migration haben. So unterscheidet sich die Vorgehensweise bei einem Systemwechsel innerhalb des GBV-Ökosystems deutlich von der Migration zu einem der cloudbasierten Systeme.

Im Gemeinsamen Bibliotheksverbund (GBV) erfolgt die kooperative Katalogisierung für alle GBV-Bibliotheken in der Verbunddatenbank CBS.[2] Für bisher knapp 200 Bibliotheken wurde durch die Verbundzentrale des GBV (VZG) ein Lokalsystem eingerichtet, genutzt wird das Bibliotheks-managementsystem LBS[3] von OCLC. Den LBS-Bibliotheken steht nun die Verlagerung der Katalogisierungsfunktionen für sog. lokale Katalogisate vom Lokalsystem in die Verbundumgebung bevor; damit verbunden ist die Migration der im LBS erfassten Datensätze in die Verbunddatenbank. Die Aktivitäten hierzu sind im Projekt „Migration lokaler Katalogisate in die Verbunddatenbank des GBV“ (kurz: LOK-Projekt) gebündelt. Hierzu gehört die Entwicklung von Software zur Unterstützung der Datenanalyse und -bereinigung in den betroffenen Bibliotheken.

Anhand der gewählten Verfahren und Arbeitsschritte des LOK-Projekts wird abschließend untersucht, inwieweit sich die allgemeinen Grundsätze und Verfahren von ETL-Prozessen und Datenmigration auf ein Projekt aus der bibliothekarischen Praxis anwenden lassen.

Im Folgenden wird aus Gründen der besseren Lesbarkeit ausschließlich die männliche Form benutzt. Es können dabei sowohl männliche als auch weibliche Personen gemeint sein.

[2] Zur Katalogisierung im GBV vgl. Diedrichs/Sandholzer 2018, S. 4–7.
[3] https://www.oclc.org/de/lbs.html (14.05.2018). Einzelne GBV-Bibliotheken setzen andere Bibliothekssysteme ein, vor allem öffentliche Bibliotheken.

2 Data-Warehouse-Systeme

Im kommerziellen Umfeld sind vielfältige betriebliche Anwendungen mit getrennter Datenhaltung im Einsatz: Verwaltung von Kunden- und Lieferantendaten, Warenwirtschaftssysteme, Finanzbuchhaltungssysteme u. a. m. Diese operativen Informationssysteme dienen der Durchführung betrieblicher Abläufe; der Schwerpunkt der Datennutzung liegt auf Aktualisierung bzw. Abfrage der Stamm- und Bewegungsdaten innerhalb des betreffenden Funktionsbereichs. Um eine globale Sicht auf die heterogenen Daten der verschiedenen Anwendungssysteme zu erhalten, werden die jeweiligen Datenbestände in einem analytischen Informationssystem, dem Data Warehouse (DW), zusammengeführt. Zusätzlich können Daten aus externen Quellen den Umfang des Data Warehouse erweitern.

Ziel ist die Schaffung eines Datenpools zur Ermittlung entscheidungsrelevanter Informationen für das Management, um daraus unternehmensstrategische Maßnahmen ableiten zu können; nur eine quantitative Grundlage ermöglicht nachvollziehbare und transparente Entscheidungen. Dies gewinnt ebenfalls in Bibliotheken zunehmend an Bedeutung. Dabei ist ein effizientes Datenmodell wichtig, um performante Zugriffe realisieren zu können.[4]

Charakteristisch für eine Data-Warehouse-Architektur ist die Speicherung von Daten in einer mehrdimensionalen Datenstruktur (Hypercube). Bei der Datenselektion und -aufbereitung werden OLAP-Funktionalitäten zum Navigieren in den Dimensionen der Datenwürfels genutzt, ebenfalls für die Ermittlung aggregierter Kennzahlenwerte.[5] Im Gegensatz zu operativen Systemen mit einer großen Zahl an kleinteiligen Lese-/Schreibzugriffen per OLTP auf die jeweilige Datenbank erfolgt der Zugriff auf das Data Warehouse nur lesend im Zuge von Auswertungen.

Ein Data Warehouse basiert normalerweise auf einer relationalen Datenbank mit Faktentabellen der Basisdaten aus den Quellsystemen sowie Dimensionentabellen zur Speicherung von Analysekriterien wie z. B. Zeit, Produkt oder geografischem Bezug. Damit bei Abfragen unterschiedlich stark verdichtet werden kann, sind die Daten der Dimensionentabellen hierarchisch gegliedert.[6]

Zur Bildung von Data Marts als fachlichen Ausschnitten werden Daten aus der Data-Warehouse-Datenbank themenspezifisch geladen; hierbei können bereits Datenaggregationen erfolgen. Der Zugriff auf ein Data-Warehouse-System erfolgt in der Regel nur lesend, zum Generieren von Reporten werden die Daten der Data Marts ausgewertet. Ergänzend können materialisierte Views (Datensichten) mit voraggregierten Werten die Antwortzeiten reduzieren.[7] Direkte SQL-Abfragen auf die Daten im Data Warehouse sind ggf. zusätzlich möglich.

Das Auswertungstool eines Data Warehouse bietet spezifische Funktionen für bereichsübergreifende Analysen. Hierzu gehören die schrittweise Verfeinerung von Analyseergebnissen (*Drill-Down*) und deren Aggregation sowie die Bildung von Teilmengen (*Slicing* und *Dicing*).[8] Zur Präsentation der Analyseergebnisse können die Kennzahlen aus den verdichteten Unternehmensinformationen

[4] Vgl. Kempter 2017, S. 142.
[5] Vgl. Kempter 2017, S. 239.
[6] So werden für die Dimension „Zeit“ beispielsweise Tag, Monat, Quartal und Jahr abgebildet. Hierdurch kann bei Abfragen der Zeitaspekt in unterschiedlicher Granularität berücksichtigt werden, je nach gewünschter Fragestellung; vgl. Farkisch 2011, S. 13–17.
[7] Vgl. Farkisch 2011, S. 82–84.
[8] Vgl. Alpar et al. 2014, S. 241.

grafisch aufbereitet werden, beispielsweise die Ausleihfrequenz in Abhängigkeit von Signaturengruppen.

Neben zahlreichen kommerziellen ETL-Lösungen im Data-Warehouse-Bereich sind Open-Source-Produkte verfügbar. Im Bibliothekskontext werden vorwiegend Eigenentwicklungen genutzt, die teilweise als Open-Source-Lösung eine Nachnutzung ermöglichen.[9]

Der Begriff „Data Warehouse" wurde Anfang der 1990er Jahre eingeführt:

> *„A data warehouse is a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management's decisions."*[10]

Ein Data Warehouse zeichnet sich somit durch folgende Eigenschaften aus:

Themenorientierung
Datenstruktur und Inhalte sind unabhängig von Geschäftsprozessen und betrieblicher Anwendungslogik, sie orientieren sich am thematischen Informationsbedarf.[11]

Integrierte Datensammlung
Bei der Zusammenführung der heterogenen Quelldaten erfolgt eine Vereinheitlichung von Datenmerkmalen; so sollte beim Transformationsschritt beispielsweise eine Normalisierung der in den Anwendungen ggf. unterschiedlichen Datumsformate erfolgen. Zu den Maßnahmen zur Qualitätssicherung zählen u. a. Plausibilitätsprüfungen.[12]

Persistente Datenhaltung
Ein Data Warehouse enthält historische Verlaufsdaten mit dauerhafter Speicherung, es findet keine Löschung oder nachträgliche Aktualisierung der gespeicherten Datensätze statt.[13] Die prozessbezogenen Daten aus operativen Anwendungen werden periodisch hinzugefügt.

Zeitbezug der Daten
Die Beständigkeit der Daten ist für Auswertungen mit Zeitdimension von Bedeutung. Während transaktionsorientierte Datenbanken den aktuellen Stand der Einzeldaten widerspiegeln, wird bei der Speicherung von Datensätzen im Data Warehouse zusätzlich der Zeitkontext festgehalten; dies erlaubt Auswertungen über verschiedene Zeiträume zum Erkennen von Trends.[14]

Der Datenbestand des Data Warehouse wird periodisch aktualisiert; hierbei kommt ein Workflow zum Einsatz, der als ETL-Prozess bezeichnet wird.

[9] U. a. Catmandu, siehe Kap. 3.3.1.
[10] Inmon 1995, zitiert nach: Inmon 2005, S. 29.
[11] Vgl. Kempter 2017, S. 146.
[12] Weitere Maßnahmen siehe Abschnitt *Transformation* in Kap. 2.1.
[13] Vgl. Kemper et al. 2010, S. 21.
[14] Vgl. Inmon/Hackathorn 1994, S. 68.

2.1 ETL-Prozesse im Data Warehouse

> *„Der ETL-Prozess beschreibt den Vorgang, Daten aus bestehenden Datenquellen zu extrahieren, mittels geeigneter Transformationsregeln zu homogenisieren, nach bestimmten Vorschriften zu bereinigen und ggf. anzureichern und in ein separates Ziel zu laden. Der integrierte Datenbestand existiert damit materialisiert in einem eigenständigen System, die Quellsysteme bleiben unverändert bestehen.“*[15]

Diese kompakte Zusammenfassung verdeutlicht den generellen Ablauf von ETL-Prozessen, die vor allem im Kontext von Data-Warehouse-Systemen bekannt sind, allerdings auch in anderen Zusammenhängen wie der Integration heterogener Metadaten in einem Rechercheportal Anwendung finden.

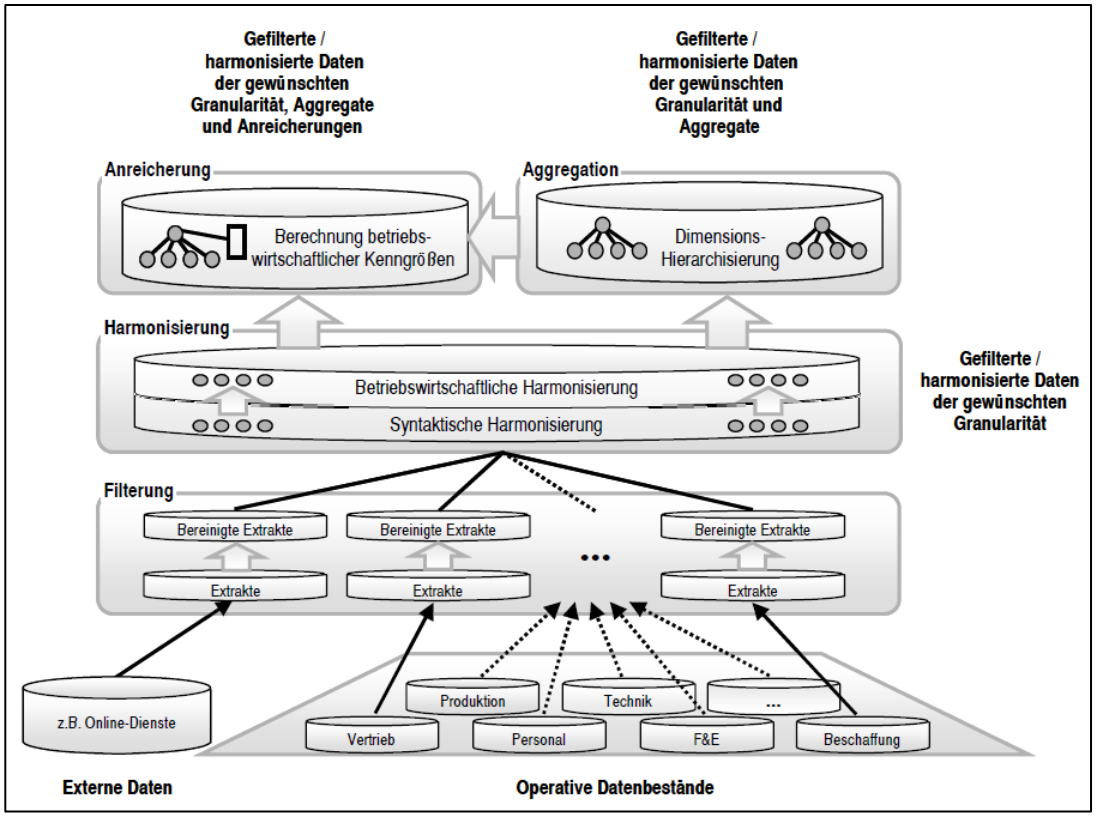


Abb. 1: Data Warehouse: Schritte des ETL-Prozesses[16]

Es folgt die exemplarische Darstellung der Prozessschritte am Muster eines Data Warehouse, angereichert um Beispiele aus dem Bibliotheksbereich.[17]

Auswahl der Quelldaten

Vor dem eigentlichen ETL-Prozess steht die Auswahl der zu übernehmenden Daten, daher muss ein Data-Warehouse-System vielfältige Dateiformate verarbeiten können, u. a. Flat Files, Daten im XML-Format, vor allem Inhalte von Datenbanktabellen der operativen Systeme.[18] Zusätzlich können externe Datenquellen wie z. B. Besucherzähler berücksichtigt werden.[19] So kann in Bibliotheken durch die Analyse von Daten mit entsprechender zeitlicher Granularität ein Zusammenhang zwischen

[15] Rossak 2013, S. 37.
[16] Vgl. Kemper et al. 2010, S. 38.
[17] Zu bibliothekarischen Data-Warehouse-Anwendungen siehe Kap. 2.2.
[18] Vgl. Kimball/Caserta 2004, S. XXIX.
[19] Vgl. Köppen et al. 2014, S. 24.

Bibliotheksbesuchen und Ausleihzahlen hergestellt werden, was z. B. Rückschlüsse zur Gestaltung der Öffnungszeiten erlaubt.

Nicht alle Daten eines operativen Systems sind für eine spätere Auswertung von Bedeutung, daher werden über die Konfiguration des ETL-Tools diejenigen Inhalte der Datenquelle festgelegt, die bei der Extraktion berücksichtigt werden sollen. Bei Datenbanken werden die für das Data Warehouse relevanten Tabellen bzw. Attribute ausgewählt, z. B. bei Daten von Bibliotheksbenutzern die Elemente Geburtsdatum, Geschlecht und Nutzergruppe, nicht aber Name und E-Mail-Adresse.[20]

Die Relevanz der zu berücksichtigenden Datenquellen ist in Abhängigkeit von den geplanten Auswertungen zu bewerten. Maßgeblich ist ebenfalls die Datenqualität, da inkonsistente, fehlerhafte oder unvollständige Daten die Auswertungsergebnisse verfälschen können.[21] Daher ist es sinnvoll, bereits in den Quellsystemen sog. Cleansingverfahren durchzuführen.[22] In jedem Fall finden Datenmodifikationen beim Transformationsschritt des ETL-Prozesses statt.

Extraktion

Im Rahmen der Extraktion werden die Quelldaten in die sog. *Staging Area* des DW-Systems kopiert. Dieser Arbeitsbereich dient als temporärer Zwischenspeicher für die Rohdaten, hier findet die Datentransformation statt.

Zwei Varianten der Extraktion sind möglich: der komplette Abzug des Datenbestandes des Quellsystems (*Snapshot*) oder eine inkrementelle Extraktion, d. h. der Export von Daten, die seit dem vorherigen Extraktionsvorgang im Quellsystem hinzugefügt oder geändert wurden. Ein Delta-Update berücksichtigt alle Änderungen zwischen dem aktuellen und dem vorherigen Extraktionsvorgang.

Für den Zugriff auf externe Datenquellen sind verschiedene Verfahren geeignet, vor allem durch

- Nutzung entsprechender Schnittstellen des operativen Systems;
- Zugriff über Anwendungsprogramme mit Datenbank-Schnittstelle (z. B. per ODBC oder JDBC);
- Import-/Export-Mechanismen: Export aus dem Quellsystem mit Bereitstellung der Daten für den Import in die Staging Area, z. B. auf einem FTP-Server.

Die Selektion der Quelldaten erfolgt meist in festgelegten Intervallen zu einem bestimmten Zeitpunkt. Dabei sind für die Extraktion von Daten eines operativen Systems Zeiträume geeignet, in denen der Zugriff das System nicht übermäßig belastet, also nachts oder am Wochenende. Für jede Datenquelle ist ein spezifisches Extraktionsverfahren festzulegen.

Der Vorgang der Extraktion ist als erste Phase der Filterung anzusehen, da hier bereits eine Auswahl der Daten des Quellsystems vorgenommen wird.

Transformation

Ziel der Transformation ist eine konsistente Datenbasis für die Übernahme in das Data Warehouse; hierfür werden mehrere Phasen durchlaufen.

[20] Zum Schutz personenbezogener Daten.
[21] Die generelle Signifikanz von Datenqualität wird in Kap. 3.2 vertieft.
[22] Vgl. Farkisch 2011, S. 60.

Filterung

Zunächst sind zur Gewährleistung der Datenqualität syntaktische Mängel wie unvollständige Daten zu identifizieren. Probleme auf semantischer Ebene, wie z. B. ungültige Datenfelder, können durch Plausibilitätskontrollen und Wertebereichsüberprüfungen erkannt werden. Eine Protokollierung derartiger Datenfehler ist sinnvoll, damit entsprechende Bereinigungen im operativen Quellsystem durchgeführt werden können; dies ist allerdings nur bei unternehmensinternen Datenquellen möglich.[23] Nicht zuletzt ist bei unterschiedlichen Zeichensätzen in den Quelldaten eine Vereinheitlichung herzustellen, dies kann i. d. R. über eine Zeichenkonvertierung zu UTF-8 erreicht werden.

Harmonisierung

Eine Standardisierung von Datentypen, Codierungen und Maßeinheiten ist für die Auswertung eines Data Warehouse essenziell. Für eine solche Harmonisierung[24] sind entsprechende Mapping-Tabellen implementiert.[25] Darüber hinaus sind Synonyme und Homonyme bei der Bezeichnung von Tabellenattributen in unterschiedlichen Quelldatenbanken zu berücksichtigen,[26] beispielsweise „Studierende“ vs. „Studenten“. Ebenfalls ist eine Eliminierung von Duplikaten bei der Zusammenführung von Daten unterschiedlicher Quellen erforderlich.

Aggregation

Die gewünschte Granularität wird über eine Aggregation der Daten erreicht. Dies ist z. B. der Fall, wenn für tagesaktuell vorliegende Werte eine Monatsauswertung angestrebt ist.[27] Hierfür ist eine Dimensionshierarchie zu entwickeln, z. B. Tag – Monat – Quartal – Jahr.

Laden

Die konsolidierten Daten aus der Staging Area werden in die Datenbank des Data-Warehouse-Systems geladen. In einem weiteren Ladevorgang werden die Data Marts befüllt, hierbei werden auch Views aktualisiert.

Bei periodischen Aktualisierungen wird ein inkrementelles Laden (*Delta Load*) bevorzugt, bei dem lediglich neue und geänderte Daten berücksichtigt werden.

ELT statt ETL

Mit ELT wird ein Prozess bezeichnet, bei dem die extrahierten Daten zunächst in das Zielsystem geladen werden; dort findet anschließend die Datentransformation statt. Diese Variante ist geeignet, wenn das Zielsystem eine Transformation direkt per SQL bzw. mit Hilfe geeigneter Software ermöglicht. Insbesondere bei großen Datenmengen ist es empfehlenswert, die Datenbank des Zielsystems während des Transformationsvorgangs für andere schreibende Zugriffe zu sperren, um Dateninkonsistenzen zu vermeiden. Zusätzlich ist dafür Sorge zu tragen, dass die Datenbank des Zielsystems keine übermäßigen Performanceeinbußen erleidet.[28]

[23] Vgl. Kemper et al. 2010, S. 28–31.
[24] Auch: Homogenisierung; u. a. verwendet bei Köppen et al. 2014.
[25] Auch: Crosswalk. Beispielsweise müssen unterschiedliche Codierungen für das Attribut „Geschlecht“ standardisiert werden, wenn in den Datenquellen einerseits [m|w] und andererseits [0|1] verwendet wird.
[26] Vgl. Kemper et al. 2010, S. 34–35.
[27] In der ZBW werden keine tagesaktuellen Daten benötigt, sodass beim täglichen Laden der Daten in das Data Warehouse eine direkte Aggregation auf Monatsbasis stattfindet.
[28] Vgl. Schreib 2013, S. 42.

Ein solches ELT-Verfahren wird zuweilen im Zusammenhang mit der Migration von Bibliotheksdaten in die GBV-Verbunddatenbank praktiziert.[29] Im LOK-Projekt ist ebenfalls eine nachträgliche Datentransformation vorgesehen, da die Verfahren zur Datenmanipulation im Zielsystem erheblich effizienter sind.[30]

[29] Siehe Kap. 4.4.2.
[30] Siehe Kap. 5.5.3.

2.2 Data-Warehouse-Anwendungen in Bibliotheken

Umfangreiches Datenmaterial zur strategischen Steuerung ist ebenso in Bibliotheken nachgefragt, beispielsweise zur Festlegung von Erwerbungsschwerpunkten. Klassische Bibliotheksmanagementsysteme (BMS) bieten hierfür häufig ein Statistikmodul an, das über vorgefertigte oder selbst erstellte Reporte einen Zugriff auf die Daten des Bibliothekssystems ermöglicht. Dabei sind Umfang und Detaillierungsgrad abhängig von den in der BMS-Datenbank vorhandenen Statistik-Zählern. Weitere Auswertungen bilden lediglich den aktuellen Zustand des Systems ab. Demgegenüber haben Data-Warehouse-Anwendungen den Vorteil, jederzeit einen Zugriff auch auf historische Daten zu gewährleisten und diese nach unterschiedlichen Kriterien gruppieren zu können.

Bei der Entwicklung von Bibliotheksmanagementsystemen der neuen Generation wurde berücksichtigt, dass aufgrund der gestiegenen Anforderungen an die Ermittlung von Kennzahlen herkömmliche Reporting-Tools zumindest in größeren Bibliotheken nicht mehr als ausreichend angesehen werden. Bei den Cloudsystemen Alma[31] (Ex Libris) und WMS[32] (OCLC) wurden daher Data-Warehouse-Lösungen integriert.

Die vorgestellten Lösungen werden vor allem hinsichtlich der Einbindung bibliografischer Metadaten geprüft, da dies für das Reporting in Bibliotheken unverzichtbar ist. Medienbezogene Listen gehören zum bibliothekarischen Arbeitsalltag; auf Neuerwerbungslisten oder Listen zur Evaluation des Lehrbuchbestands sind bibliografische Angaben wie Autor, Titel sowie Angaben zu Auflage und Erscheinungsjahr unerlässlich. Des Weiteren ist im Hinblick auf Maßnahmen zur Datenqualität von Metadaten die möglichst vollständige Ausgabe der gewünschten Informationen entscheidend.

Im Kontext dieser Arbeit sind ETL-Prozesse zur Datenbereitstellung im Data Warehouse von Belang; das Vorgehen zu Konfiguration und Erstellung von Datenauswertungen wird daher vernachlässigt.

2.2.1 Klassische Reporting-Verfahren

Reporting-Funktionen sind nicht bei allen Bibliotheksmanagementsystemen integriert, dennoch ist es i. d. R. möglich, per SQL direkt auf die relationale Datenbank zuzugreifen. So erzeugen die Systemverantwortlichen in LBS-Systemen für ihre Bibliotheken regelmäßig Auswertungen für die Bereiche Ausleihe und Erwerbung, z. B. Listen aktueller Mahnfälle oder ein Zugangsjournal. Hinzu kommen Ad-Hoc-Auswertungen, die auf Anfrage individuell erzeugt werden, z. B. Listen der meistausgeliehenen Medien oder zur Bestandsrevision.

Bis zum Versionsupgrade auf LBS3-Port im Jahr 2007 gab es keine Möglichkeit, direkt auf bibliografische Informationen über ein entsprechendes Tabellenattribut der LBS-Datenbank zuzugreifen. Stattdessen war für einen verbesserten Nutzen solcher Listen das Parsen von Datenbanktabellen mit dem binären PICA+-Speicherformat[33] erforderlich. Seitdem die LBS-Datenbank um zwei Tabellen für Metadaten-Elemente erweitert wurde, können per SQL erzeugte Listen mit den wesentlichen bibliografischen Informationen angereichert werden und erhalten so eine höhere Aussagekraft. Auch wenn es um den Export großer Titelmengen geht (z. B. bei der Qualitätssicherung

[31] http://www.exlibrisgroup.com/products/alma-library-services-platform/ (14.05.2018).
[32] https://www.oclc.org/de/worldshare-management-services.html (14.05.2018).
[33] Siehe Kap. 3.1.3 sowie Anhang 3 mit einer ausführlichen Formatbeschreibung.

in der Signaturerfassung), skaliert die direkte Selektion per SQL stabiler als die entsprechenden Funktionen im Katalogisierungsclient WinIBW[34].

2.2.2 BibControl als Data-Warehouse-Lösung für Bibliotheken

Die Abfragemöglichkeiten per SQL in der Datenbank des Bibliothekssystems erfüllen häufig nicht die bibliothekarischen Anforderungen, weil damit nur statische Auswertungen möglich und komplexe Abfragen nicht umsetzbar sind. Diesen Bibliotheken bietet eine Data-Warehouse-Anwendung den Zugang zu komfortablen Datenanalysen, insbesondere weil weitere Datenquellen wie die Metadaten des Verbundsystems oder die Nutzungsdaten von Online-Ressourcen mit den BMS-Daten verknüpft und die Analyseergebnisse grafisch aufbereitet werden können. Ebenso können Bibliotheken oder Mitarbeiter ohne direkten Datenbankzugang zu ihrem Bibliothekssystem mit einer Business-Intelligence-Lösung Anfragen in Bezug auf Inhalt, Umfang und Rhythmus selbst gestalten und sind für die Erstellung von Auswertungen und Listen nicht von ihrem Systemverantwortlichen bzw. BMS-Dienstleister abhängig. Darüber hinaus kann durch die Auslagerung der gewünschten Daten in ein externes System die Belastung des Bibliothekssystems durch Abfragen reduziert werden. Die Auswertung von Daten in ihrem zeitlichen Kontext ist nur mit Data-Warehouse-Lösungen möglich, da die Datenbank des operativen Bibliothekssystems i. d. R. nur den aktuellen Stand widerspiegelt.

Obwohl das Angebot von DW-Produkten generell recht groß ist[35], existiert auf dem deutschen Markt nur ein Produkt, das speziell auf den Bedarf von Bibliotheken ausgerichtet ist. BibControl[36], ursprünglich als separates Statistik-Modul des Bibliothekssystems BIBLIOTHECA[37] (OCLC) vertrieben, ist inzwischen ebenfalls bei Bibliotheken mit anderen BMS im Einsatz.[38]

Über einen lokal zu installierenden Client besteht Zugriff auf die Inhalte der Data Marts in Form von Würfeln und Views. Für die Bibliotheken der Jade Hochschule[39] ist u. a. ein Würfel *Bestand* eingerichtet, über den eine Qualitätskontrolle der bibliografischen Metadaten erfolgt. Eine Liste der *Laufenden Titel* wird aus einer entsprechenden View generiert. Als Informationsportal für Bibliotheksmitarbeiter ist das *Online-Cockpit* gedacht, das über einen Web-Browser Zugriff auf Views („Berichte“), Dokumente mit automatisch generierten Auswertungen wie Neuerwerbungslisten sowie visualisierte Kennzahlen gewährt. Mit dem ETL-Tool können die Abläufe zum Extrahieren, Transformieren und Laden bibliothekarischer Daten in das DW konfiguriert werden. Bibliotheks-spezifische Erfassungsmasken lassen sich mit dem *Application Builder* erstellen; dies wird in der ZBW[40] u. a. für die Erfassung der Metadaten von Publikationen der Bibliotheksmitarbeiter genutzt.

Die Schritte des ETL-Prozesses werden über die *Jobverwaltung* gesteuert. Ebenfalls über *Jobs* erfolgen die Aktualisierung von Würfeln und das Erstellen von Statistiken. Bei einer Anbindung von BibControl an LBS-Datenbanken erfolgen die Updates inkrementell für Tabellen mit Eingabedatum und/oder Änderungsdatum, Tabellen ohne solche Datumsattribute werden immer komplett

[34] https://www.gbv.de/bibliotheken/verbundbibliotheken/02Verbund/02Verbundsystem/02WinIBW (14.05.2018). Aktuelle Version ist WinIBW 3.7.0.2.
[35] Vgl. Humm/Wietek 2005, S. 12–13.
[36] http://www.triangle-solutions.de/loesungen/tim4bib-biinderbibliothek/index.html (14.05.2018).
[37] https://www.oclc.org/de/bibliotheca.html (14.05.2018).
[38] Die Informationen zu BibControl stammen, wenn nicht anders angegeben, vom BibControl-Anwendertreffen am 14.11.2017 in Göttingen sowie aus dem internen Wiki des Anbieters.
[39] Informationen zu BibControl in der Jade Hochschule vgl. Schulz 2018.
[40] Informationen zu BibControl in der ZBW vgl. Mauder 2018.

selektiert. Da LBS ohne Tages(abschluss)lauf auskommt, ist der Importplan für einen Zeitpunkt am Abend nach Bibliotheksschließung terminiert.

Neben Nutzungsdaten von Online-Ressourcen können weitere externe Datenquellen über Schnittstellen[41] eingebunden werden. Die Metadaten der GBV-Verbunddatenbank (CBS) werden in der ZBW u. a. zur Anreicherung von Auswertungen der Statistikdaten der GBV-Fernleihe benötigt. Zunächst vor allem für Datenqualitätsmaßnahmen wurde die BibControl-Implementierung in der Jade Hochschule genutzt. Die Verfügbarkeit elementarer Metadatenfelder in der LBS-Datenbank ermöglichte es der Bibliothek, Datenfehler u. a. bei Signaturen und Standortangaben zu identifizieren.

Bereits im Rahmen der Implementierung richtet der Anbieter Würfel und Views ein; er orientiert sich dabei an generellen Anforderungen von Bibliotheken. Bei der Definition der Data-Mart-Tabellen wird das Mapping festgelegt, also die Zuordnung von Feldern der Quelldatenbank zu Feldern der Zieldatenbank. Verschiedene konfigurierbare Datentransformationen werden beim Laden der Daten in das Data Warehouse durchlaufen.

Nicht alle wünschenswerten Datentransformationen sind in der BibControl-Implementierung realisiert; so wird zumindest in der Instanz für die ZBW der Inhalt des Feldes „Erscheinungsjahr“ aus den bibliografischen Daten offenbar nicht normalisiert. Als Ergebnis sind z. B. für das Jahr 1990 unter der entsprechenden Dimension die Werte „1990“, c1990, „[1990]“, „(1990)“ u. a. m. vorhanden, da die im CBS vorhandenen heterogenen Jahresangaben bei der Datentransformation in BibControl nicht vereinheitlicht wurden.

Über *Entladeregeln* werden Abfragen auf die operative Datenbank definiert. Festzulegen sind u. a. die zu berücksichtigenden Tabellen und -felder sowie die Bedingungen für die Selektion von Datensätzen aus einer Tabelle (WHERE-Klausel). Bei der Datenselektion kann bereits eine erste Transformation der Daten erfolgen, beispielsweise kann auch nur ein Teil des Tabellenfeld-Inhalts extrahiert werden (SUBSTRING-Funktion). Bei Aktivierung der *Delta Verarbeitung* werden nur die neuen Einträge seit der vorherigen Extraktion für die Selektion berücksichtigt.

Bibliotheken mit einer Lizenz für das ETL-Tool können die Einstellungen der Ersteinrichtung anpassen und ergänzen. Anderenfalls müssen Anpassungen kostenpflichtig beauftragt werden, dies gilt genauso für die Einbeziehung zusätzlicher externer Datenquellen.

BibControl ist ein mächtiges und flexibles Werkzeug, das allerdings einen hohen personellen und finanziellen Aufwand erfordert, wenn der Kunde die Komplexität eines Data Warehouse ausschöpfen will. Dies gilt insbesondere für die Einbindung weiterer Datenquellen zusätzlich zu den Daten des Bibliothekssystems. Deshalb bleiben die Implementierungen in der bibliothekarischen Praxis häufig hinter ihren Möglichkeiten zurück.

[41] Aktuell sind dies: Webservices, ODBC, strukturierte Flat Files (XML, CSV).

2.2.3 Analytics-Module in Bibliothekssystemen der neuen Generation

Alma Analytics

In Alma, dem cloudbasierten Bibliothekssystem von Ex Libris, steht für Reporte und Auswertungen das sog. *Analytics*-Modul zur Verfügung, dem ein Data Warehouse zugrunde liegt; als technische Basis dient *Oracle Business Intelligence Enterprise Edition*[42].

Der ETL-Prozess für Transaktionsdaten aus Alma läuft jede Nacht, die entsprechenden Daten in Alma Analytics sind daher jeweils tagesaktuell; Titeldaten hingegen werden monatlich geladen.[43] Die Daten werden inkrementell extrahiert.[44]

Bei der Inbetriebnahme von Alma stehen einige Standard-Reporte, sog. *Out-of-the-Box Reports*[45], zur Verfügung, zusätzlich können eigene Reporte erstellt werden. Der CSV-Export berücksichtigt bis zu 500.000 Reportzeilen.[46]

Alle Reporte greifen auf die von Ex Libris implementierten Data Marts (in Alma: „Subject Areas“) zu, die als Star-Schema modelliert sind.[47] Jede *Subject Area* umfasst eine Faktentabelle mit den Transaktionsdaten aus Alma und mehrere Dimensionstabellen. Über die *Subject Area* „Titles“ besteht Zugriff auf eine Reihe bibliografischer Daten, ergänzt um Metadaten des Katalogisates wie das letzte Änderungsdatum des Datensatzes. Auch Reporte, die auf eine andere *Subject Area* wie z. B. „Fulfillment“ (Ausleihe) zugreifen, können bibliografische Felder der übergreifend nutzbaren Dimensionentabelle „Bibliographic Details“ einbinden.[48]

Bibliografische Metadaten können außerdem direkt aus Alma exportiert werden, z. B. zur Datenanalyse oder für eine beliebige Weiterverarbeitung; die Ausgabe erfolgt im MARC-Format[49] (wahlweise binär oder MARCXML). Für den Export wird zunächst über die reguläre Alma-Suche ein Set der gewünschten Datensätze zusammengestellt und anschließend ein Offline-Job gestartet, der die Daten dieses Sets exportiert und auf einen FTP-Server kopiert.[50]

Da Bibliotheken über Alma Analytics Zugriff auf die Daten aller genutzten Ex-Libris-Produkte haben, lassen sich vielfältige Auswertungen durchführen. Dies sind neben Daten aus Alma und dem Discovery-System auch Kennzahlen zu Lizenzen. Das Spektrum der abrufbaren Daten erfüllt vermutlich die Anforderungen von Bibliotheken sehr gut, auch weil für das Erstellen eigener Reporte keine zusätzliche (kostenpflichtige) Lizenz erforderlich ist. Der Export von aktuellen bibliografischen Metadaten aus Alma erscheint umständlich, weil das Ausführen von Jobs Mitarbeitern mit einer besonderen Berechtigung vorbehalten ist. Metadatenbasierte Reporte hingegen sind lediglich monatsaktuell. Weitere externe Datenquellen lassen sich nicht einbinden, das manuelle Einpflegen von Daten ist ebenfalls nicht möglich. Aufgrund der zentralen Bereitstellung als Cloudlösung ist eine individuelle Konfiguration des ETL-Prozesses nicht vorgesehen.

[42] http://www.oracle.com/technetwork/middleware/bi-enterprise-edition/overview/index.html (14.05.2018).
[43] Vgl. Ex Libris 2017b, S. 397.
[44] Vgl. Kortick 2015, Folie 4.
[45] Vgl. Ex Libris 2017b, S. 104–108.
[46] Vgl. Ex Libris 2017b, S. 16.
[47] Vgl. Ex Libris 2017b, S. 5.
[48] Vgl. Ex Libris 2017b, S. 113.
[49] Ausführliche Beschreibung siehe Kap. 3.1.2.
[50] Vgl. Ex Libris 2018a, S. 1–3.

WorldShare Analytics

Das cloudbasierte Bibliotheksmanagementsystem von OCLC, WorldShare Management Services (WMS), nutzt zu Reportingzwecken ebenfalls ein Data Warehouse. Als Datenbasis dient WMS einschließlich der Module Licence Manager[51] und Collection Manager sowie WorldCat.

Die Data Marts werden periodisch aktualisiert: Daten der Module Ausleihe und Erwerbung sind tagesaktuell, bibliografische Metadaten werden monatlich auf den neuesten Stand gebracht.[52] Wie bei Alma hat der Anwender keinen Einfluss auf Auswahl und Umfang der Tabellen bzw. Felder in den Data Marts.

Neben kennzahlenbasierten Reporten sind Listen zum physischen Bestand und zu E-Ressourcen abrufbar, beispielsweise Neuerwerbungslisten oder Reporte zum Abgleich von Titeln in unterschiedlichen Paketen der Knowledgebase.[53] Wesentliches Merkmal der Funktion „Collection Evaluation“ sind bibliografische Daten, die allerdings bei Vorhandensein mehrerer Exemplarsätze pro Besitznachweis unvollständig und daher nur bedingt zur Bestandsevaluation geeignet seien. Zumindest könnten daraus Erkenntnisse für spezifische Datenqualitätsmaßnahmen gewonnen werden, z. B. zur Bereinigung von Konvoluten.[54]

Für alle Reporte gilt aus Performancegründen eine Größenbeschränkung von 50.000 Zeilen.[55] Insbesondere bei größeren Bibliotheken dürfte dies eine komplette Ausgabe der gewünschten Elemente (z. B. Metadaten des physischen Bestands) verhindern. Da die Selektionskriterien der Standard-Listen von OCLC vorgegeben sind, wäre es evtl. praktikabel, über mehrere Teilselektionen eine Gesamtliste zu erzeugen und diese z. B. in Excel nach eigenen Kriterien zu filtern; ein solches Vorgehen wäre vermutlich im Hinblick auf Datenqualitätsmaßnahmen geeignet.

Nachteilig erscheint bei WorldShare Analytics vor allem, dass für den Zugriff auf die Reporte vier verschiedene Zugänge zu nutzen sind, jeweils für eine festgelegte Report-Auswahl.[56] Das Erstellen eigener Reporte ist nur mit einer kostenpflichtigen Zusatzlizenz möglich.[57] Nicht zuletzt ist vor allem die Nutzbarkeit von Metadaten-Reporten aufgrund der Ausgabemengenbeschränkung insbesondere für Bibliotheken mit umfangreichen Beständen beeinträchtigt. Wie bei Alma ist aufgrund der zentralen Bereitstellung als Cloudlösung eine individuelle Konfiguration des ETL-Prozesses nicht vorgesehen.

Für WorldShare wie Alma gilt, dass bibliotheksspezifische Anpassungen von Data Marts, die Konfiguration des ETL-Prozesses (u. a. Uhrzeit und Mapping) sowie die Implementierung von Zusatzfunktionen wie das Einbinden weiterer Datenquellen nicht möglich sind. BibControl ist dagegen grundsätzlich für zusätzliche Anforderungen offen; Anwender können Serviceleistungen wie die Einbindung zusätzlicher Datenquellen beim Anbieter kostenpflichtig hinzubuchen.

[51] Die verfügbaren Daten sind jedoch offenbar unvollständig (Stand Februar 2017), vgl. http://lists.eril-l.org/pipermail/eril-l-eril-l.org/2017-February/003108.html (14.05.2018).
[52] Die Daten der Standard-Reporte zur Ausgabe neuer bzw. gelöschter Titel werden täglich aktualisiert, vgl. OCLC 2018a, S. 4.
[53] Liste der verfügbaren Reporte vgl. OCLC 2018b.
[54] Vgl. Edwards et al. 2017, Folien 4-8.
[55] Vgl. https://help.oclc.org/Library_Management/WorldShare_Reports/Available_standard_reports (14.05.2018).
[56] *WorldShare reports*, *STATS reports*, *FTP reports*, *Adobe Analytics reports*; vgl. OCLC 2018b.
[57] Vgl. https://www.oclc.org/de/worldshare-report-designer.html (14.05.2018).

3 ETL-Prozesse im Bibliothekskontext

Das Berufsbild der traditionell als Katalogisierer bezeichneten Bibliotheksmitarbeiter hat sich in den letzten Jahren verändert: gefragt sind jetzt Metadaten-Manager mit erweitertem Tätigkeitsfeld. Die Erschließungsarbeit verlagert sich zunehmend von der manuellen Erfassung einzelner Datensätze im Katalog zur Übernahme und Konvertierung von Datenpaketen aus verschiedenen Quellen. Zum Spektrum der neuen Aufgaben gehört neben dem Umgang mit unterschiedlichen Datenformaten ebenso die Anwendung von Tools für das Metadatenmanagement über den Katalogisierungsclient hinaus.

Um Fremddaten lokal nutzen zu können oder eigene Daten anderen Institutionen zur Verfügung zu stellen, sind ETL-Prozesse etabliert. Dabei sind insbesondere bei der Nutzung unterschiedlicher Standards in Quell- und Zielsystem, bei Abweichungen vom formalen Standard oder Einschränkungen der Datenqualität ggf. umfangreiche Datentransformationen erforderlich. Dies gilt namentlich für die Zusammenführung von Daten in einem Informationssystem, das Anwendern den Vorteil des Zugriffs auf Daten unterschiedlicher Herkunft unter einer Suchoberfläche bietet. Hier stehen Bibliotheken vor den besonderen Herausforderungen bei der Integration heterogener Datenquellen.

3.1 Bibliothekarische Datenformate

Auch vor der Einführung IT-gestützter Verfahren in Bibliotheken existierten Regelwerke für die Erfassung von bibliografischen Daten zur Verzeichnung in Bibliothekskatalogen. Bereits nach den 1899 veröffentlichten *„Instruktionen für die alphabetischen Kataloge der preussischen Bibliotheken und für den preussischen Gesamtkatalog*“ (kurz: Preußische Instruktionen (PI)) sind die formalen Elemente einer Ressource zu identifizieren und *„in eine feststehende Ordnung*“[58] zu bringen. Nicht zuletzt wegen der grammatischen Ordnung bei Sachtitelschriften sind die PI für eine maschinelle Sortierung in Online-Katalogen nicht geeignet.

Der Umstieg auf die 1977 publizierten *„Regeln für die alphabetische Katalogisierung (RAK)*“ war in vielen wissenschaftlichen Bibliotheken mit der EDV-Einführung im Bereich der Katalogisierung gekoppelt. In der Ausgabe für wissenschaftliche Bibliotheken von 1983 wird die *„Schematisierung der Titelaufnahmen* [betont, da] *in den Regeln festgelegte Formalia von der EDV übernommen werden*“[59].

Das 2010 veröffentlichte Regelwerk zur Formalerschließung *„Resource Description and Access*“ (RDA)[60] hat den Anspruch, zeitgemäße Anforderungen hinsichtlich der formalen Beschreibung von Ressourcen für alle Medientypen zu erfüllen und ist auf eine internationale Anwendung ausgelegt.[61]

Im Unterschied zu einem Regelwerk, das Vorgaben für die Identifizierung und Erfassung der Bestandteile einer Ressource macht, legen (standardisierte) Datenformate die Struktur und die Felder fest, in denen diese Bestandteile erfasst bzw. gespeichert werden. Es existiert eine Reihe von Datenformaten zur Erfassung und Speicherung von Katalogisaten in den verschiedenen Bibliotheks-systemen. Allen Formaten gemein ist die Strukturierung der Metadatenelemente. Die Erfassung von

[58] Instruktionen für die alphabetischen Kataloge der preussischen Bibliotheken und für den preussischen Gesamtkatalog 1899, § 10.
[59] Deutsches Bibliotheksinstitut / Kommission für Alphabetische Katalogisierung 1983, S. VIII.
[60] https://wiki.dnb.de/display/RDAINFO/RDA-Info (14.05.2018).
[61] Vgl. Wiesenmüller/Horny 2017, S. 15–16.

bibliografischen Metadaten erfolgt elementweise, wobei je nach System die Kategorien[62] mit Codes und/oder den entsprechenden verbalen Bezeichnungen gekennzeichnet sind. Für den Datenaustausch haben sich weitere Datenformate bewährt. Voraussetzung für einen effektiven Datenaustausch ist die Nutzung standardisierter Metadaten-Schemata zur Beschreibung der strukturellen Eigenschaften von Datenformaten; dies ermöglicht eine syntaktische Prüfung der Daten.[63] Die Interoperabilität der Metadaten ist gewährleistet, wenn die Vorgaben der Metadaten-Standards eingehalten werden.[64]

Zu den verbreiteten Anwendungsformaten für bibliographische Inhalte gehören die feldbasierten Formate MAB, MARC und PICA. Bibliothekarische Daten für den Datenaustausch und zur Nutzung in Webanwendungen sind in der hierarchischen Auszeichnungssprache XML codiert; dafür existieren u. a. die Codierungen MARCXML und PICA XML[65].

Datenformate unterliegen einem fortwährenden Wandel. Bei lokal gepflegten Datenformaten, wie den PICA-Formaten im GBV, führen Anforderungen aus dem Anwenderkreis zuweilen zu einer Anpassung bzw. Erweiterung des Datenformats. Insbesondere Änderungen des genutzten Regelwerks haben teilweise gravierende Änderungen zur Folge. So verursachte die RDA-Einführung in den deutschsprachigen Bibliotheken und Bibliotheksverbünden (D-A-CH-Raum) umfangreiche Modifikationen der Datenformate für Katalogisierung und Datentausch.[66] Betroffen waren außerdem die Anwendungsregeln zur Festlegung der Semantik der Daten, die die Vorschriften des RDA-Regelwerks in der Ausprägung des jeweiligen Datenformats konkretisieren.[67]

3.1.1 MAB / MAB2

Das Maschinelle Austauschformat für Bibliotheken (MAB) war 1973 für den Datenaustausch im deutschen Sprachraum entwickelt worden und wurde für den Magnetbanddienst der Deutschen Nationalbibliothek (DNB)[68] genutzt; ab 1976 konnten Institutionen wöchentlich mit Datensätzen von Katalogisaten des Wöchentlichen Verzeichnisses der Deutschen Bibliographie beliefert werden. Die überarbeitete Version MAB2 wurde 1995 veröffentlicht und wird bis heute in Bibliothekssystemen als Erfassungs- und/oder Internformat eingesetzt. Entwicklung und Pflege von MAB2 wurden 2006 eingestellt.[69]

Die Gruppe der MAB-Formate umfasst Festlegungen für bibliografische Titeldaten, aber auch für Lokal- und Normdaten. Die Aufteilung der Felder in Feldgruppen (Segmenten) erfolgt nach inhaltlichen Gesichtspunkten. Über die Satzarten Hauptsatz (Code h), Untersatz für Bandaufführungen (Code u) und Untersatz für die Aufführung von Abteilungen (Code y) lassen sich Hierarchien abbilden.[70]

---

[62] Der Begriff *Feld* anstelle von *Kategorie* ist im deutschen Sprachgebrauch ebenfalls üblich.
[63] In der Katalogisierungsdatenbank des GBV über Validationsroutinen realisiert.
[64] Vgl. Rühle 2012, S. 8–9.
[65] Schreibweise nach http://format.gbv.de/pica/xml (14.05.2018).
[66] Mai 2012: Beschluss zur RDA-Einführung im D-A-CH-Raum; Januar 2016: RDA-Einführung im GBV.
[67] Die GBV-Katalogisierungsrichtlinie enthält Vorgaben hinsichtlich der im PICA-Format zu erfassenden Inhalte, siehe http://swbtools.bsz-bw.de/cgi-bin/help.pl?cmd=index&verbund=GBV&regelwerk=RDA und https://www.gbv.de/bibliotheken/verbundbibliotheken/02Verbund/01Erschliessung/02Richtlinien/01KatRicht/inhalt.shtml (beide 14.05.2018).
[68] Seinerzeit unter dem Namen „Deutsche Bibliothek“.
[69] Vgl. Meßmer/Müller 2014, S. 398.
[70] Vgl. http://www.dnb.de/DE/Standardisierung/Formate/MAB/mab_node.html (14.05.2018).

Nach wie vor genutzt wird MAB2 beispielsweise in der deutschen Aleph[71]-Auslieferung, deren Internformat Aleph SEQuential (ASEQ) auf MAB2 basiert.[72] Damit wenden die Berliner Alma-Bibliotheken[73] weiterhin das (angepasste) MAB-Format bei der Katalogisierung in der Aleph-Verbunddatenbank B3Kat an; zur Übernahme der Metadaten in die Alma-Umgebung ist eine Datenkonvertierung von MAB2 zu MARC 21 implementiert.[74] MAB2 als Export- und Austauschformat ist ebenfalls noch immer im Einsatz. Daher mussten im Zuge der RDA-Einführung die MAB2-Exportschnittstellen angepasst werden, um Lokalsysteme weiterhin mit regelwerksgerechten Daten beliefern zu können.[75]

3.1.2 MARC / MARC 21

Als internationaler Standard für die Darstellung und Übermittlung bibliografischer und verwandter Informationen in maschinenlesbarer Form hat sich das MARC-Format (*Machine-Readable Cataloging*) etabliert, es bildet die internationale Norm ISO 2709 *Format for information exchange*[76] ab. Im Unterschied zum MAB-Format, bei dem inhaltlich zusammengehörende Elemente in Segmenten angeordnet sind, orientiert sich das MARC-Format bei der Reihenfolge der Felder an der ISBD-Abfolge. Überdies sind im MARC-Format Deskriptionszeichen zu erfassen.

Es handelt sich generell um ein flaches Datenformat ohne hierarchische Datensatzstruktur. Im Datenmodell des D-A-CH-Raums erhält jedoch, abweichend zur Standard-Anwendung von MARC 21, jeder Band einer mehrteiligen Monografie (MTM) einen eigenen Datensatz; beim Datenaustausch sind entsprechende Konvertierungen durchzuführen.[77]

Neben dem Format für bibliografische Daten existieren weitere MARC-Formate für Normdaten (*Authority Format*), Bestandsdaten (*Holdings Format*), Veranstaltungsdaten (*Community Information Format*) und Klassifikationen (*Classification Data Format*).

Das unter Federführung der Library of Congress (LoC) entwickelte Format wurde 1966 erstmals im Rahmen einer Pilotanwendung als Magnetbanddienst zur Lieferung von LoC-Katalogisaten an US-amerikanische Bibliotheken eingesetzt. Zunächst nur für Metadaten von Büchern konzipiert, wurde das Format später für andere Medientypen erweitert.[78] Aufgrund der für damalige Verhältnisse großen zu übermittelnden Datenmengen war ein Format gewählt worden, das die bibliografischen Metadaten in kompakter, speicherplatzsparender Form abbildet.

Vor allem im angloamerikanischen Raum wurden Formate aus der MARC-Familie zunächst für den Datentausch und das Erstellen von Buchkatalogen und Katalogkarten genutzt, z. T. in länderspezifischen Varianten. Ebenfalls war das MARC-Format von Anbietern kommerzieller Bibliothekssysteme bald auch als Erfassungs- und Speicherformat implementiert worden. Dies geschah offenbar,

[71] http://www.exlibrisgroup.com/products/aleph-integrated-library-system/ (14.05.2018).
[72] Vgl. Scholz/Labner 2014, Folie 7.
[73] Die Bibliotheken der Freien Universität, Humboldt-Universität, Technischen Universität und der Universität der Künste.
[74] Weitere Ausführungen, auch zur Kreiskonvertierung MARC – MAB - MARC siehe Kap. 4.4.4.
[75] Das BSZ verspricht die Pflege der MAB-Schnittstelle noch bis 2020, vgl. https://wiki.bsz-bw.de/doku.php?id=v-team:daten:datendienste:mab2 (14.05.2018).
[76] https://www.iso.org/standard/41319.html (14.05.2018)
[77] Vgl. Helmkamp/Oehlschläger 2008, S. 26.
[78] Vgl. Avram 1975, S. 10.

weil die Konvertierung in ein abweichendes lokales Format die Rechenleistung damaliger Computer zu sehr beeinträchtigt hätte.[79]

Mit der Veröffentlichung von MARC 21 wurden 1999 die bisherigen Formatvarianten USMARC und CAN/MARC vereinheitlicht,[80] seither haben weitere Nationalbibliotheken ihr nationales MARC-Format durch MARC 21 abgelöst.[81] Einige länderspezifische Formatvarianten[82] sowie das von der IFLA entwickelte UNIMARC sind allerdings weiterhin in Gebrauch, ebenfalls USMARC, u. a. als Exportformat des Bibliotheksmanagementsystems BIBLIOTHECA.

Der Beschluss zur „*einheitliche*[n] *Anwendung von MARC21* [sic!] *als Austauschformat für alle deutschsprachigen Bibliotheken*“[83] erfolgte 2004. Nachdem in den Folgejahren die Vorbereitungen für den Formatwechsel getroffen wurden, ist MARC 21 seit 2009 als Austauschformat in deutschen Verbünden und der DNB im Einsatz. Bibliotheksmanagementsysteme mit MARC 21 als Erfassungsformat werden ebenfalls hierzulande genutzt, u. a. Koha und Alma.[84]

Im Gemeinsamen Bibliotheksverbund (GBV) waren im Rahmen des Umstiegs von MAB2 auf MARC 21 im D-A-CH-Raum nur geringfügige Änderungen am Katalogisierungsformat und am Internformat erforderlich, da hierfür die PICA-Formate[85] genutzt werden. Allerdings mussten die Import- und Exportschnittstellen angepasst werden, um weiterhin den Datentausch zwischen den Verbünden sowie weiteren Datenlieferanten bzw. -empfängern zu gewährleisten. Als Vorteil gegenüber MAB2 war für die VZG insbesondere die Formatänderung in Bezug auf Datensatz-Verknüpfungen von Belang, namentlich der Wegfall des Satztyps für Abteilungen innerhalb einer MTM-Datensatzhierarchie.[86]

Wie andere strukturierte Datenformate können MARC-Daten per XML-Serialisierung verarbeitet werden.[87]

Die in Anhang 1 dargestellte Datenstruktur beschreibt das sog. *Marc Communications Format* für den Datenaustausch. Da solche `.mrc`-Dateien lediglich maschinenlesbar sind, ist für das manuelle Prüfen oder Editieren von MARC-Daten eine Konvertierung in das MARC-Textformat[88] (`.mrk`) erforderlich; hierfür existieren mehrere Tools, u. a. das verbreitete MarcEdit.[89]

[79] Vgl. Kommentar von Michelle Newbury am 24.05.2009 zum Blogbeitrag „Who needs MARC?“ https://commonplace.net/2009/05/who-needs-marc/ (14.05.2018).
[80] Vgl. https://www.loc.gov/marc/annmarc21.html (14.05.2018).
[81] U. a. in Großbritannien, vgl. http://www.bl.uk/bibliographic/nbsils.html (14.05.2018), und Norwegen, vgl. https://bibliotekutvikling.no/ressurser/kunnskapsorganisering/verktoykasse-for-kunnskapsorganisering/marc-formater/ (14.05.2018).
[82] U. a. Dänemark mit DanMARC2, vgl. https://slks.dk/om-slots-og-kulturstyrelsen/organisation/raad-naevn-og-udvalg/bibliografisk-raad/resource-description-and-access/ (14.05.2018), und Japan mit JAPAN/MARC, vgl. http://www.ndl.go.jp/en/data/jm.html (14.05.2018).
[83] Arbeitsstelle für Standardisierung (AfS) 2004, S. 7.
[84] Zur Katalogisierung in Koha vgl. https://koha-community.org/manual/17.11/en/html/06_cataloging.html (14.05.2018) und Alma vgl. Ex Libris 2018c. (14.05.2018).
[85] Siehe Kap. 3.1.3.
[86] Vgl. Block 2007, Folie 9.
[87] Siehe dazu Abschnitt *MARCXML* in Kap. 3.1.4.
[88] In der angloamerikanischen Literatur: *text file*, zuweilen auch: *tagged format* oder *mnemonic file*.
[89] Siehe auch Kap. 3.3.4.

Trotz grundsätzlich einheitlicher Anwendung von MARC 21 als Austauschformat unterscheiden sich die Exportformate der deutschen Bibliotheksverbünde, sowohl im Leader als auch in den Blöcken der allgemeingültigen Felder.[90] Die „Vereinbarungen zum Datentausch in MARC 21“[91] der Arbeitsgruppe Kooperative Verbundanwendungen (AG KVA) aus dem Jahr 2014 werden daher zurzeit überarbeitet. Aktuell erfolgt eine Bestandsaufnahme der Unterschiede in den jeweiligen MARC-21-Exportformaten.[92]

Daraus folgt, dass die spezifische Ausprägung des MARC-Formats, je nach verwendeter Schnittstelle bzw. Datenquelle, bei der Weiterverarbeitung von MARC-Daten entsprechend zu berücksichtigen ist. Dies gilt insbesondere bei der Integration von Metadaten aus heterogenen Datenquellen im Rahmen von ETL-Prozessen.[93]

3.1.3 PICA3 / PICA+

Die drei deutschen Bibliotheksverbünde GBV, HeBIS und SWB sowie die Deutsche Nationalbibliothek und die Zeitschriftendatenbank (ZDB) betreiben das von der niederländischen Pica-Stiftung entwickelte Bibliothekssystem CBS für die (Verbund-)Katalogisierung.[94] In nahezu allen Bibliotheken der Verbünde HeBIS und GBV wird LBS als Lokalsystem eingesetzt. Beide Systeme verwenden die proprietären Datenformate PICA3 zur Erfassung und PICA+ als Internformat. Über Mappingtabellen werden die PICA3-Felder beim Speichern eines neu angelegten oder bearbeiteten Datensatzes in das Internformat umgewandelt. Die Reihenfolge der PICA3-Felder folgt im Wesentlichen der ISBD-Struktur.

In der CBS-Implementierung des GBV sind weitere Formate konfiguriert, neben MAB2, UNIMARC und MARC 21 können weitere präsentationsorientierte Formate zur Anzeige bzw. Download von Datensätzen genutzt werden.

Die PICA-Datenstruktur sieht verschiedene Arten der Datensatzverknüpfung vor. Zusätzlich zu hierarchischen Titel-Titel-Verknüpfungen (z. B. Aufsatz → Zeitschrift) ist eine Verknüpfungsmöglichkeit zu Normdaten (z. B. GND) implementiert. An der Kategorie für die Materialart ist der Satztyp (u. a. Monographie, Überordnung einer MTM, Zeitschrift) erkennbar.

PICA3 verwendet vierstellige numerische Feldnummern („Kategorien“), während das Internformat PICA+ vierstellige alphanumerische Codes für die Feldbezeichnung vorsieht und wiederholbare Felder durch eine sog. *Occurrence*-Angabe[95] kennzeichnet.[96] Bedeutung und Gebrauch der Felder im GBV sind mit Angabe von PICA3- und PICA+-Kategorie in der Katalogisierungsrichtlinie dokumentiert. Aufgrund lokaler Anpassungen bzw. Erweiterungen unterscheidet sich das PICA-Format der deutschen Anwenderinstitutionen. Daher wird für die Zusammenlegung der Katalogisierungsdatenbanken von GBV und SWB im K10plus-Projekt[97] eine gemeinsame Formatbeschreibung erarbeitet; beim Laden der jeweiligen Verbunddaten in die K10plus-Datenbank muss eine entsprechende Datenkonvertierung erfolgen.

[90] Unterschiede bestehen u. a. bei der Speicherung von GND und ISBN.
[91] Vgl. AG Kooperative Verbundanwendungen der AG der Verbundsysteme 2014.
[92] Vgl. interne Projektdokumentation https://info.gbv.de/display/KNEU/Home (14.05.2018).
[93] Siehe Kap. 3.4.
[94] Durch die vollständige Übernahme von Pica im Jahr 2007 ist OCLC der Systemanbieter von CBS und LBS.
[95] Mit Schrägstrich abgetrennte 2-stellige Zahl.
[96] Weitere Informationen zu Besonderheiten im PICA3-/PICA+Format siehe Anhang 2.
[97] https://www.bszgbv.de/services/k10plus/ (14.05.2018).

Wie bei MARC 21 ist der Inhalt eines Feldes ggf. auf mehrere Unterfelder verteilt. Im PICA+-Format werden die Unterfelder ausnahmslos durch das Unterfeld-Einleitungszeichen $ gefolgt vom Unterfeld-Code dargestellt, während in PICA3 die Unterfelder zum Teil durch Deskriptionszeichen nach ISBD gekennzeichnet sind, siehe Beispiel:

PICA+	ƒfSUB-LSƒaH Germ 231/31: 21ƒdfƒx00
PICA3	!SUB-LS!H Germ 231/31: 21 @ f

Allerdings wurde im GBV im Zusammenhang mit der Anfang 2016 erfolgten Formatanpassung für RDA in den meisten Feldern der Titelebene die bisherige Unterfeldkennzeichnung durch Deskriptionszeichen zugunsten der in PICA+ üblichen Darstellung aufgegeben.

Ebenso wie beim MARC-Format existiert für PICA+ neben dem textbasierten Format, das u. a. im Katalogisierungsclient WinIBW für Erfassung, Präsentation und Download genutzt wird, eine Variante für die physische Speicherung in der Datenbank. Der Aufbau des internen PICA+-Speicherformats für bibliografische Metadaten ist in Anhang 3 anhand der LBS-Datenbanktabelle `titles_global` exemplarisch dargestellt.

3.1.4 XML-basierte Datenformate

Zu den bekannten Formaten für bibliografische Metadaten zählen Dublin Core und MODS; beide treten vor allem als XML-Serialisierung in Erscheinung. Ebenfalls basieren zahlreiche Schnittstellen auf XML, z. B. SRU[98] und OAI-PMH[99]. Nicht zuletzt ist für den Umgang mit Daten in den Formaten MARCXML und PICA XML ein Verständnis für XML unerlässlich.

Datenformat XML

XML (Extensible Markup Language) ist eine erweiterbare Metasprache, deren erste Spezifikation[100] 1998 vom World Wide Web Consortium (W3C) als Standard veröffentlicht wurde. Als Textdatei gespeicherte Daten im strukturierten XML-Format lassen sich sowohl von Menschen lesen als auch von XML-Prozessoren (Parsern) verarbeiten. Im Unterschied zu HTML existieren keine festgelegten Tags, daher sind anwendungsspezifische Strukturen darstellbar. So ist es möglich, die Semantik der Datenfelder über Tags auszudrücken.

Bei XML handelt es sich um eine branchenübergreifend einsetzbare, hersteller- und plattformunabhängige Technologie, die einen webbasierten interoperablen Datentausch erlaubt.

> *„Because XML makes it possible to exchange data in a standard format, independent of storage, it has become the de-facto standard for representing metadata descriptions of resources on the Internet."* [101]

Um Dateien im XML-Format verarbeiten zu können, ist zunächst deren Wohlgeformtheit sicherzustellen. Eine XML-Datei ist wohlgeformt, wenn die syntaktischen XML-Regeln eingehalten werden. Dazu gehört das Vorhandensein von Start- und Ende-Tags sowie eine korrekte Verschachtelung der XML-Tags. Ebenfalls müssen die Regeln zur formalen Schreibweise, zur Bildung von Elementnamen und zur Definition von Attributen eingehalten werden.

[98] Search/Retrieve via URL: http://www.loc.gov/standards/sru/ (14.05.2018); auch: Search/Retrieval via URL.
[99] Open Archives Initiative Protocol for Metadata Harvesting: https://www.openarchives.org/pmh/ (14.05.2018).
[100] Spezifikation für XML 1.0: https://www.w3.org/TR/xml/ (14.05.2018).
[101] Hunter 2003, S. 321.

Gerade weil die Baumstruktur einer XML-Datei grundsätzlich frei gestaltbar ist, sollten Formatdefinitionen hinterlegt werden, um die Konsistenz von XML-Dokumenten zu gewährleisten. Zur Prüfung auf Gültigkeit eines XML-Dokuments sind die Validationsmethoden *Document Type Definition* (DTD) und *XML Schema Definition* (XSD) verbreitet. Eine DTD enthält eine schematische Beschreibung von XML-Dokumenten, damit wird die Baumstruktur der XML-Dokumente exakt vorgegeben. XML-Schema ist wesentlich flexibler und mächtiger, es können beispielweise Inhalte validiert und Namensräume berücksichtigt werden. Während DTD eine eigene Syntax verwendet, werden XSD in der XML-Syntax ausgedrückt. Die zur Validation genutzte Formatdefinition wird innerhalb des Wurzelelements in der XML-Datei deklariert. Mit Hilfe solcher Validationsverfahren kann sichergestellt werden, dass XML-Dokumente identisch aufgebaut sind. Dies ist insbesondere für die Weiterverarbeitung von XML-Dateien erforderlich, wenn diese über Exportroutinen aus einer Anwendung erzeugt werden.

Mit XML als Metasprache können beliebige in (semi-)strukturierter Form vorliegende Daten übermittelt werden. XML agiert dabei als Containerformat, das ebenfalls bei MARC 21 und PICA+ Verwendung findet.

MARCXML

MARCXML wurde 2002 von der LoC entwickelt, um das weltweit verbreitete MARC-Format für moderne Datenübertragungsprotokolle nutzbar zu machen.[102] MARCXML ist eine Implementierung des MarcXchange-Standards (ISO 25577), der die Anforderungen an einen XML-basierten Datenaustausch bibliografischer und anderer Metadaten spezifiziert.[103]

MARCXML-Daten lassen sich verlustfrei zu MARC 21 (und umgekehrt) konvertieren.[104] Eine Datenumsetzung in andere XML-basierte Formate wie Dublin Core, MODS oder ONIX ist ebenfalls leicht durchführbar. Für MARCXML-Daten wird der Unicode-Zeichensatz (codiert in UTF-8) verwendet.

Die Struktur von MARC 21 bleibt in MARCXML erhalten, da alle Informationen zu Feld und Subfield als Attribut eines XML-Tags gespeichert sind. Daher kann auf das Directory verzichtet werden. Die XML-Struktur ist in der Schemadatei `MARC21slim.xsd` festgelegt.[105] Auf Grund des Overheads durch XML-Tags benötigt eine MARCXML-Datei mehr als dreimal so viel Speicherplatz wie eine kompakte `.mrc`-Datei.[106]

Zu den zahlreichen Anwendungen im Bibliothekskontext zählen vor allem die von Verlagen bereitgestellten Metadaten für E-Ressourcen, die Bibliotheken für den Import in die Suchmaschine ihres Discovery-Systems nutzen, teilweise über automatisierte ETL-Verfahren. Ebenfalls stellen Bibliotheksverbünde ihre Daten im Format MARCXML unter CC0-Lizenz zur freien Nachnutzung zur Verfügung, u. a. die Daten des B3Kat[107] und der Verbunddatenbank des SWB[108].

[102] Vgl. Contessi/Gadea Raga o.J., S. 2.
[103] Vgl. https://www.iso.org/standard/62878.html (14.05.2018).
[104] Hierfür existieren zahlreiche Tools; eine Auswahl wird in Kap. 3.3 behandelt.
[105] Für eine übersichtliche Darstellung der Elemente vgl. http://www.loc.gov/standards/marcxml/xml/spy/spy.html (14.05.2018).
[106] Getestet mit 812 Datensätzen: im MARC-Format 8,9 MB, als MARCXML 27,9 MB.
[107] Vgl. https://www.bib-bvb.de/web/b3kat/open-data (14.05.2018).
[108] Vgl. https://wiki.bsz-bw.de/doku.php?id=v-team:daten:openaccess:swb (14.05.2018).

PICA XML

Wird das Exportformat `picaxml` beim Zugriff per unAPI- oder SRU-Schnittstelle gewählt, so ist die vollständige Ausgabe der Kategorien eines Datensatzes gewährleistet.[109] Im Unterschied dazu fehlen beim MARCXML-Export einige für das LOK-Projekt relevante Kategorien wie Materialart oder Verbuchungsnummer.

Eine proprietäre XML-Schnittstelle für PSI-Datenbanken[110] ist ebenfalls verfügbar. Es kann zwischen zwei Ausgabeformaten gewählt werden: entweder das in XML verpackte binäre bzw. „normalisierte" PICA+-Format als lesbarer Text mit binären Steuerzeichen oder eine in XML verpackte HTML-Darstellung der OPAC-Präsentation mit verbalen Feldbezeichnungen. Diese Schnittstelle wird von den im Zuge des LOK-Projekts erstellten Perl-Programmen verwendet.[111]

[109] Zur Evaluation von unAPI und SRU im LOK-Projekt siehe Kap. 5.2.2.
[110] Datenbanken mit *Pica Search and Index Software* (PSI) im GBV: https://www.gbv.de/gsomenu (14.05.2018).
[111] Ausführliche Darstellung der XML-Schnittstelle siehe Kap. 5.2.2.

3.2 Datenqualitätsmanagement

Die Beschäftigung mit Daten- und Informationsqualität ist branchenübergreifend von Belang. So thematisieren zahlreiche White Paper und Studien das Problem des mangelnden Bewusstseins für Datenqualität aus kommerzieller Sicht. Auch der Sammelband „Daten- und Informationsqualität“[112] nimmt sich des Themas unter unterschiedlichen Aspekten an.

„Der Erfolg eines Unternehmens hängt elementar von der Qualität der im Unternehmen vorhandenen und genutzten Daten ab.“[113] Diese Aussage unterstreicht eine Studie aus dem Jahr 2015, der zufolge 62% aller deutschen Unternehmen der Einschätzung sind, *„dass die größten Probleme im Unternehmen durch unvollständige oder fehlende Daten verursacht werden*.“[114]

Dies müsste insbesondere auf den Bibliotheksbereich übertragbar sein, da hier das Erzeugen und Verarbeiten von Daten als zentrales Geschäftsfeld anzusehen ist. Allerdings setzen sich Bibliotheken häufig erst im Zusammenhang mit der Migration ihres Bibliothekssystems mit der Qualität ihrer Daten auseinander, da der Migrationserfolg maßgeblich von der Qualität des Datenmaterials abhängig ist. Im Regelbetrieb hingegen wird in Bibliotheken die Einbeziehung von Metadaten in das Qualitätsmanagement zumeist vernachlässigt. Vielfach sind Methoden und Techniken nicht bekannt, die die Diskrepanz zwischen positiver Selbsteinschätzung und tatsächlicher Datenqualität aufdecken könnten.

Die Nutzbarkeit einer Anwendung wird erheblich gesteigert, wenn die zugrundeliegenden Daten fehlerfrei und vollständig sind. Beispielsweise ist die Auswertung von Daten in einem Data-Warehouse-System nur dann hinreichend aussagekräftig, wenn die Daten vollständig, korrekt und konsistent sind. Generell ist im Kontext von ETL-Prozessen die Datenqualität von essenzieller Bedeutung, da namentlich der Transformationsschritt erheblich erleichtert wird, wenn konsistente Daten vorausgesetzt werden können; dazu ist eine Kontrolle der extrahierten Daten erforderlich. Bei fehlenden Informationen oder formal falschen Daten ist eine Korrektur in den Ausgangssystemen sinnvoll. Diese sollten daher beispielsweise über die Möglichkeit der Validation bei der Dateneingabe verfügen, um eine hohe Datenqualität zu gewährleisten.

Zur Einschätzung der Qualität ist eine Analyse der Daten unter verschiedenen Kriterien erforderlich. Dabei erscheint es sinnvoll, zunächst eine Klassifizierung der Datenfehler vorzunehmen. Generell zu unterscheiden sind einerseits Fehler in einem einzelnen Datensatz bzw. einem einzelnen Datenfeld und andererseits Mängel unter Betrachtung der Gesamtdatenmenge. Demzufolge greifen unterschiedliche Maßnahmen zur Datenbereinigung.

[112] Hildebrand et al. 2015.
[113] Weigel 2015, S. 71.
[114] Experian Marketing Service 2016, S. 5.

Die von Rahm/Do[115] erarbeitete Klassifizierung von Datenfehlern wurde in der folgenden Übersicht auf den Bibliothekskontext übertragen.

Fehlerart	**Beispiel**
Datenfehler in einem einzelnen Datensatz	
Fehlender Inhalt	Zweiter Vorname nicht angegeben
Schreibfehler	*Biblothekartag*
Inhalt formal falsch	Konferenzdaten im falschen Format: *21.-24.03.2006* (falsch) statt *2006.03.21-24* (korrekt)
Widersprüchliche Werte	Verantwortlichkeitsangabe in Kategorie `4000` enthält einen anderen Namen als Kategorie `3000` (Verfasser).
Inhalt in falscher Kategorie bzw. falschem Subfield	Verbuchungsnummer in Kategorie `8100` (Zugangsnummer)
Fehlende Kategorie bzw. Subfield	Ohne Sprachcode (Kategorie `1500`)
Duplikate	identischer Inhalt in mehreren Datenfeldern: *1131 !105825778!Hochschulschrift ; ID: gnd/4113937-9* *1131 !105825778!Hochschulschrift ; ID: gnd/4113937-9*
Schreibvarianten	bei Verlagsangaben: *Springer-Fachmedien-Wiesbaden-GmbH* und *Springer-Fachmedien Wiesbaden*
Veraltete Daten	Nicht mehr gültiger Link zum Digitalisat
Datenfehler bezogen auf die Gesamtheit der Datensätze	
Duplikate	*Die Kinder aus Bullerbü*, Ausgabe 1986
Fehlende Verknüpfungen	Personenname nicht mit GND-Normsatz verknüpft
Falsche Verknüpfungen	Teil einer MTM ist mit der übergeordneten Aufnahme einer anderen Auflage verknüpft
Unterschiedliche Genauigkeit	Durch Datenimport erzeugte Duplikate, wenn die umfangreichere Zielaufnahme nicht als Matching-Kandidat erkannt wurde
Unterschiedliche Datensatzstrukturen	Bandsätze im LBS-Ausleihmodul vs. Av-Sätze im CBS[116]

Tab. 1: Klassifikation von Datenfehlern im CBS[117]

Eine Einordnung dieser Qualitätseinbußen wird im Zusammenhang mit allgemeingültigen Kriterien für Informationsqualität in Kap. 3.2.1 vorgenommen; ebenso wird beispielhaft aufgezeigt, inwieweit die Kriterien auf den Bibliotheksbereich übertragbar sind. Kap. 3.2.2 beleuchtet potenzielle Maßnahmen zur Verbesserung der Datenqualität unter zahlreichen Gesichtspunkten. Beispiele für die Einbeziehung von Datenqualitätsmaßnahmen im bibliothekarischen Metadatenmanagement zeigt Kap. 3.2.3 auf. Schließlich wird in Kap. 3.2.4 der Themenkomplex Dublettenerkennung und -behandlung in der bibliothekarischen Praxis betrachtet.

[115] Vgl. Rahm/Do 2000, S. 4–8.
[116] Als Repräsentation eines Zeitschriftenjahrgangs.
[117] Mit Beispielen aus der Katalogisierungsdatenbank des GBV (Stand: 27.02.2018).

3.2.1 Informationsqualität

Informationsqualität (IQ) lässt sich nach verschiedenen Kriterien beurteilen. Auf Grundlage des Konzepts von Wang/Strong[118] wurden 15 sog. IQ-Dimensionen von der Deutschen Gesellschaft für Informations- und Datenqualität e.V. (DGIQ) für den deutschen Sprachraum aufbereitet.[119]

Für die bibliothekarische Praxis sind vorrangig diese sechs Dimensionen relevant:

1. Relevanz
2. Vollständigkeit
3. Fehlerfreiheit
4. Einheitliche Darstellung
5. Aktualität
6. Wertschöpfung

Relevanz

Bei der Auswahl von Ausgangsdaten für ein Data Warehouse ist die Relevanz der Daten zu berücksichtigen. Dies gilt insbesondere auch für die Integration von Daten in einem Discovery-System. Die Relevanz ist jeweils im Hinblick auf das Zielsystem und dessen Anwender einzuschätzen. Von den verfügbaren Datenquellen und -inhalten sind nur diejenigen zu berücksichtigen, die für die Anwender relevante Informationen enthalten.

Beispiel: Das Portal des Fachinformationsdiensts (FID) Politikwissenschaft POLLUX berücksichtigt Nachweise zu politikwissenschaftlicher Fachliteratur. Hierzu werden die aus verschiedenen Quellen stammenden Metadaten entsprechend gefiltert.[120]

Vollständigkeit

Je nach Datentyp gelten unterschiedliche Kriterien für das Merkmal Vollständigkeit. Inwieweit ein Datensatz als vollständig betrachtet werden kann, hängt von den Vereinbarungen der Anwendercommunity ab. Zur Verbesserung der Datenqualität im Hinblick auf Vollständigkeit wäre eine Prüfroutine für das Vorhandensein von Pflichtfeldern geeignet.

Beispiel: Bei der Erfassung von Metadaten für Aufsätze im CBS ist bei fehlender Jahresangabe die Speicherung des Datensatzes nicht möglich; es wird eine entsprechende Meldung ausgegeben, da dem Erscheinungsjahr eine große Bedeutung für die Identifikation der Ressource zugemessen wird.

Problematisch in Bezug auf das Merkmal „Vollständigkeit“ ist die Änderung der Vorgaben im laufenden Betrieb, wenn diese nicht auch rückwirkend umgesetzt werden, sondern nur für neu erfasste Daten gelten. Dadurch sind Altdaten ggf. nicht mehr formal oder inhaltlich vollständig. Dies kommt besonders bei bibliografischen Daten regelmäßig vor, da die Regelwerke zur Formalerfassung des Öfteren angepasst werden und eine Änderung von u. U. Millionen von bereits vorhandenen Datensätzen technisch und inhaltlich nicht ohne Weiteres möglich ist.

[118] Vgl. Wang/Strong 1996.
[119] Vgl. Rohweder et al. 2015, S. 26–31.
[120] Vgl. https://wikis.sub.uni-hamburg.de/webis/index.php/Politikwissenschaft (14.05.2018).

Fehlerfreiheit
„*Informationen sind fehlerfrei, wenn sie mit der Realität übereinstimmen.*“[121] Die Informationsqualität ist beeinträchtigt, wenn aufgrund von Datenfehlern die Information verfälscht wird.

Beispiel: Ist im Bibliothekssystem eine E-Mail-Adresse nicht korrekt erfasst, kann die betreffende Mitteilung ihren Empfänger nicht erreichen. Dies betrifft u. a. Mahnungen für Bibliotheksnutzer oder Bestellungen beim Buchhändler.

Einheitliche Darstellung
Insbesondere für die Vergleichbarkeit von Daten hat die Konsistenz einen besonderen Stellenwert. So sind gleichartige Informationen einheitlich zu erfassen, anderenfalls ist die Nutzung der Daten bzw. deren Auswertung eingeschränkt. Quelldaten aus unterschiedlichen Ausgangssystemen können eine vollständige Konsistenz im jeweiligen Quellsystem aufweisen; für die Zusammenführung, z. B. in einem Data Warehouse oder in einem Discovery-System, sind bestimmte Informationen jedoch formal zu vereinheitlichen. Ebenfalls ist die Konsistenz von Datensätzen untereinander zu betrachten; hier dürfen keine logischen Widersprüche auftreten.[122]

Beispiel: Bei der Katalogisierung im CBS ist auf eine einheitliche Erfassung von Signaturen zu achten; anderenfalls führt eine OPAC-Recherche zu unerwarteten Ergebnissen und das Auffinden von Literatur im Bibliotheksbestand wird folglich beeinträchtigt.

Aktualität
Aktuelle Informationen bilden die Eigenschaft des Objekts zeitnah ab.[123] Daraus folgt, dass bei der Nutzung von Daten zwischenzeitliche Änderungen am Datenbestand berücksichtigt werden.

Beispiel: Für die in einem OPAC oder Discovery-System präsentierten Nachweise elektronischer Ressourcen müssen Änderungen der Lizenzbedingungen (z. B. Ablauf des Online-Zugriffs) zeitnah nachgeführt werden.

Wertschöpfung
Hierunter fällt auch der Bereich der Redundanzen, da bei einem Duplikat bzw. einer Dublette keine wertschöpfende Eigenschaft gegeben ist.

Als Dubletten werden mehrere Datensätze bezeichnet, die dasselbe reale Objekt beschreiben, sich allerdings in (meist) mehreren Datenfeldern unterscheiden.[124]

Beispiel: Vor der Erfassung eines Katalogisats im CBS ist das Vorhandensein einer entsprechenden Aufnahme mittels geeigneter Recherchen zu überprüfen, da anderenfalls ggf. eine Dublette angelegt werden würde.

Im Bibliotheksbereich genutzte Verfahren zum Umgang mit Dubletten werden in Kap. 3.2.4 behandelt.

[121] Rohweder et al. 2015, S. 37.
[122] Vgl. Apel et al. 2015, S. 8.
[123] Vgl. Rohweder et al. 2015, S. 41.
[124] Vgl. Rohweder et al. 2015, S. 43.

3.2.2 Datenanalyse und -bereinigung

Zur Verbesserung der Qualität bibliothekarischer Daten ist zunächst die Analyse der Daten erforderlich. Hierdurch ist eine Kategorisierung der identifizierten Probleme möglich, beispielsweise nach den sechs in Kap. 3.2.1 vorgestellten Kriterien. In diesem Zusammenhang sind ebenfalls die Ursachen der Datenfehler zu ermitteln. Nach einer Priorisierung der Fehlergruppen zielen die weiteren Überlegungen darauf ab, die Fehlerquellen zu beseitigen. Schließlich müssen jeweils geeignete Verfahren für die Fehlerbereinigung entwickelt werden.

Dieser Ablauf zur Fehlerkorrektur wird bei Datenmigrationen meist nur einmalig durchlaufen. Vorteilhaft sind Bereinigungen vor der Migration im Altsystem, so können die im Kontext des Altsystems bekannten und bewährten Verfahren angewendet werden. Auch im Zuge der eigentlichen Datenkonvertierung können Datenfehler berichtigt werden. Ebenso sind periodisch wiederholte Datenkorrekturen eingeführte Praxis bei der Integration heterogener Metadaten. Hierbei liegt allerdings der Schwerpunkt auf der Überführung des Datenschemas der zu importierenden Daten an die Datenstruktur des Zielsystems, weniger auf einer Fehlerkorrektur.

Zur Datenanalyse können verschiedenen Verfahren zum Einsatz kommen; in Bibliotheken werden u. a. die Tools Catmandu, OpenRefine und Mable+/MARCel genutzt.[125] Erste Hinweise auf unsachgemäß genutzte Datenfelder bieten die Häufigkeitswerte einer Feldstatistik. Mit Hilfe der Filterfunktionen in OpenRefine lassen sich statistische Ausreißer und damit potenzielle Datenfehler erkennen. Recht einfach ist dabei die Identifizierung fehlerhafter Eingaben bei Feldern mit codierten oder formalisierten Inhalten (z. B. Sprachcodes bzw. ISSN und Datumsangaben). Eine im CBS bewährte Kontrollmöglichkeit, u. a. nach dem Import von Bibliotheksdaten, bietet das Scan-Kommando, das eine geordnete Liste der Indexeinträge für den angegebenen Index präsentiert. Damit können sich auch bibliothekarische Anwender einen strukturierten Überblick über die Inhalte von (indexierten) Datenfeldern verschaffen.

In erster Linie gelten effektive Korrekturmaßnahmen den Stammdaten, also Daten mit Grundinformationen über die im jeweiligen Zusammenhang gespeicherten Objekte. Im bibliothekarischen Kontext sind hier vor allem Daten mit normativem Charakter wie GND- oder ZDB-Daten zu nennen, aber auch Datensätze einer lokalen oder Verbunddatenbank. Im Gegensatz dazu haben Bewegungsdaten wie die zu einer Ausleihe gespeicherten Informationen eine begrenzte Speicherdauer; eine Datenkorrektur erfolgt hier i. d. R. nur punktuell und anlassbezogen.

Für Datenfehler gibt es zahlreiche Ursachen. In bibliothekarischen Datenbanken sind vor allem die Datenflüsse maßgeblich, resp. die manuelle Erfassung und das automatisierte Einspielen von Metadaten aus heterogenen Quellen wie Verlagen, anderen Bibliotheken und Verbünden. Während beispielsweise in der Verbunddatenbank des GBV bei der Online-Erfassung Kontrollmechanismen wie Dublettenprüfung und eine komplexe Validation von Datenfeldern aktiviert sind, ist die Validation beim Datenimport ausgeschaltet, um die Quote der automatisch eingespielten Datensätze zu erhöhen. Diese teilweise fehlerbehafteten Katalogisate müssen vom Bibliotheksmitarbeiter bereinigt werden, wenn ein solcher Datensatz für die eigene Bibliothek nachgenutzt werden soll, da bei diesem Vorgang wieder die Online-Validation greift und der fehlerhafte Datensatz nicht ohne Korrektur abgespeichert werden kann.

[125] Siehe auch Kap. 3.3.

Gelegentlich werden in der CBS-Verbunddatenbank Katalogisate nicht regelgerecht erfasst, um die lokale Kataloganzeige zu optimieren. Auch im LBS werden auf Grund der fehlenden Möglichkeit zur Erweiterung des Datenschemas zuweilen Datenfelder für andere als die eigentlich vorgesehenen Zwecke genutzt.

Die zu ergreifenden Maßnahmen zur Fehlerkorrektur sind entsprechend vielfältig; ergänzend sollten Überlegungen zur zukünftigen Reduzierung fehlerhafter Daten angestellt werden.

Nach der Aufstellung der vorhandenen Fehlertypen ist eine Priorisierung der möglichen Bereinigungsmaßnahmen vorzunehmen. Dabei ist primär die Relevanz der Daten im jeweiligen Kontext zusammen mit der Effektivität der Maßnahmen zu bewerten. Während Daten, die für die Benutzerrecherche im Katalog von Bedeutung sind, vordringlich zu bereinigen sind, können Daten mit geringer Relevanz vergleichsweise nachrangig behandelt werden.

Nach Zwirner[126] sind bei einer Klassifizierung von Datenfehlern deren Häufigkeiten im Zusammenhang mit der Bedeutung der Datenmenge zu bewerten. Die Änderungsfrequenz der Daten sollte Einfluss auf die Art potenzieller Bereinigungsmaßnahmen haben, dabei ist die Grundlage der Anforderungen an die Datenqualität im Blick zu behalten. Der generelle Aufwand für Datenqualitätsmaßnahmen schließt auch Maßnahmen zur Vermeidung neuer Fehler mit ein.

Wenngleich diese Aufstellung seinen Ursprung im betrieblichen Umfeld hat, lassen sich die Merkmale von Datenfehlern gleichwohl auf die Bibliothekssparte übertragen.

<u>Bedeutung der Daten</u>
Relevanz der betroffenen Daten für die Bibliothek und deren Geschäftsprozesse.

- Fehlerhafte Angaben in Datenfeldern, die zur Recherche bei der Vorakzession herangezogen werden, führen ggf. zu einer unnötigen Beschaffung mit entsprechenden Kosten.
- Fehler bei der Erfassung von Signaturen oder URLs erschweren bzw. verhindern die Zugänglichkeit eines Mediums.
- Eine fehlende oder falsche Angabe der Umfangsangabe im Katalogisat hat kaum Konsequenzen.

<u>Grundlage der Anforderungen an die Datenqualität</u>
Bedeutung von institutionellen Vorgaben; Konsequenzen bei Missachtung.

- Für die Katalogisierung im CBS gelten die Erfassungsregeln nach RDA, die jeweiligen Kategorien eines Datensatzes sind gemäß Katalogisierungsrichtlinie zu belegen. Beim Beitritt zum GBV verpflichtet sich eine Bibliothek, die jeweils gültigen Katalogisierungsregeln des Verbundes einzuhalten; dies ist in der sog. Bearbeitungsrichtlinie[127] festgehalten. Diese Vorgaben haben normativen Charakter, bei Nichteinhaltung findet keine Sanktionierung statt. Von den Vorgaben abweichende Katalogisate werden i. d. R. von nachnutzenden Bibliotheken verbessert.

<u>Art der Fehler</u>
Klassifizierung in technische oder fachliche Fehler, um so Rückschlüsse auf die Art der möglichen Bereinigungsmaßnahmen ziehen zu können.

[126] Zwirner 2015, S. 104–107.
[127] Vgl. Verbundzentrale des GBV 2012.

- Erfolgen in der aktuellen Version von LBS4 mehrere Änderungen im Nutzerdatensatz in einem Schritt, so wird bei einer bestimmten Konstellation die Ausweisnummer in der LBS-Datenbank auf (`null`) gesetzt. Solche technisch induzierten Datenfehler können nur direkt in der Datenbank per SQL bereinigt werden.
- Wird im CBS die Verbuchungsnummer nicht in dem dafür vorgesehen Feld erfasst, kann das betreffende Medium im Ausleihmodul OUS nicht verbucht werden. In der überwiegenden Zahl solcher Fälle wird versehentlich das Feld für die Zugangsnummer verwendet, sodass über den Zugangsnummern-Index diese Fälle ermittelt werden können.

Art der möglichen Bereinigungsmaßnahmen

Abhängig vom Fehlerfall; Unterscheidung zwischen manuellen, automatischen und gemischten Korrekturverfahren.

- Bei einer kleinen Menge zu bereinigender Fälle ist eine manuelle Korrektur angemessen. Im GBV wird dies beispielsweise bei falsch erfassten Verbuchungsnummern praktiziert.
- Bei der Datenkonvertierung für eine Bibliothek mit LBS-Service liegt der Fokus auf der Umsetzung der bibliografischen Daten. Zuweilen stellt sich jedoch erst bei der Einrichtung des Ausleihmoduls heraus, dass bestimmte ausleihrelevante Inhalte in den Daten geändert werden müssen, um die gewünschte Funktionalität des Ausleihmoduls zu gewährleisten. Je nach Umfang werden hierfür automatisierte Verfahren, auch in Eigenregie der Bibliothek, angewandt (per WinIBW-Funktion „sucheErsetze“[128]) oder es erfolgt das Erstellen eines spezifischen Bereinigungsprogramms und dessen Ausführung durch Mitarbeiter der VZG.
- Als gemischtes Korrekturverfahren stellt sich die Dublettenbereinigung im CBS dar. Der Mitarbeiter verlinkt den als Dublette erkannten Datensatz mit der Zielaufnahme, zur Unterstützung des Ablaufs kann eine WinIBW-Funktion verwendet werden.[129] Die eigentliche Umlenkung der Exemplare erfolgt offline durch ein Programm der VZG.

Änderungshäufigkeit der Daten

Unterscheidung zwischen statischen und dynamischen Daten, daraus resultiert die Frequenz von Bereinigungsmaßnahmen.

- Bei dynamischen Daten sollte aus Gründen der Effizienz das Augenmerk auf Maßnahmen zur Fehlervermeidung gerichtet werden. Vorteilhaft ist beispielsweise die Verwendung von normiertem Vokabular über Auswahllisten anstelle einer freien Texteingabe.
- Zu den eher statischen Daten im Bibliotheksbereich zählen die Normdatensätze der GND. Eine Datenbereinigung kann hier in größerem zeitlichem Abstand erfolgen; allerdings haben Änderungen an einem GND-Satz Auswirkungen auf eine ggf. sehr große Anzahl verknüpfter Titelsätze und daher eine hohe Priorität.

Anzahl der Datenfehler

Das zu wählende Korrekturverfahren ist abhängig von der Anzahl der Fehler je Fehlerkategorie.

- Bei einer großen Zahl an Erfassungsfehlern ist eine initiale Bereinigung sinnvoll. Häufig wiederkehrende Datenfehler müssen regelmäßig beobachtet und korrigiert werden. Formal falsche E-Mail-Adressen im LBS-Ausleihmodul werden zur Bereinigung an die betreffende Bibliothek weitergeleitet. Infolge des wiederholten Vorkommens wird hierfür ein täglich gestartetes Shellscript genutzt.

[128] Vgl. Zeitschriftendatenbank 2015, S. 18–20. Im GBV nicht dokumentiert.
[129] Vgl. https://verbundwiki.gbv.de/display/VZG/Dubletten+umlenken (14.05.2018).

- Bei den Überlegungen zur Datenbereinigung ist die Anzahl der Datenfehler in Relation zur Relevanz der betreffenden Daten für den Geschäftsprozess zu betrachten. Ein durch Nutzung eines fehlerhaft erstellten Makros in zahlreichen Datensätzen vorkommendes zusätzliches Leerzeichen innerhalb des Signaturfeldes hat keine Auswirkungen auf die Recherche und die Identifizierung des betreffenden Mediums. Lediglich in einer Revisionsliste mit Sortierung nach Signatur kann dies nachteilig sein.

Maßnahmen zur Vermeidung neuer Fehler

Im Zusammenhang mit der Bereinigung einer Fehlerkategorie sind Methoden zur Beseitigung der Fehlerquellen mit dem Ziel der zukünftigen Vermeidung dieser Fehler zu entwickeln.

- Nicht nur für ungeübte Mitarbeiter ist im CBS die Nutzung von Textbausteinen sowie Titel- und Exemplarmasken bei der Anfertigung von Katalogisaten eine gute Unterstützung.[130] Die Vorgabe bestimmter Felder bzw. deren Vorbelegung sorgt für eine einheitliche Erfassung im Sinne der Katalogisierungsregeln.

Aufwand für Datenqualitätsmaßnahmen

Für alle Aktivitäten mit dem Ziel der Qualitätsverbesserung ist der jeweilige Aufwand zu bewerten. Dieser hat die o. g. Punkte zu berücksichtigen, vor allem Relevanz und Umfang der fehlerhaften Daten sowie die Möglichkeiten einer automatischen Bereinigung. Vorbeugende Maßnahmen sind insbesondere effizient, wenn hiermit häufig auftretende Fehler relevanter Daten vermieden werden können.

Datenbereinigungen im GBV per „Reparatur"-Programm

Während bei einer Migration von bibliografischen Metadaten im Zusammenhang mit der Verbundteilnahme die Datenkonvertierung[131] im Vordergrund steht, ist es im laufenden Betrieb regelmäßig geboten, mit Hilfe von sog. Reparaturprogrammen Datenbereinigungen durchzuführen. Handelt es sich dabei um Änderungswünsche von Bibliotheken, werden vorwiegend Daten der Lokal- und Exemplarebene bearbeitet. Weiterhin werden Korrekturen durchgeführt, die die Datenqualität der Titeldaten verbessern und somit allen Verbundbibliotheken zu Gute kommen. Hierzu gehören auch die aufgrund der RDA-Einführung erforderlichen Anpassungen und ebenso vorbereitende Maßnahmen für K10plus, der Zusammenlegung der Katalogisierungsdatenbanken von GBV und SWB. Die hierzu genutzten Reparaturprogramme sind in der Programmiersprache C geschrieben. Unter Nutzung einer CBS-spezifischen Programmbibliothek[132] werden die betreffenden Datensätze über eine Suchanfrage identifiziert, die Kriterien zur Datensatzauswahl werden per XML-Datei an das Programm übergeben. Nachdem ein Datensatz eingelesen ist, werden die Datenmanipulationsfunktionen auf diesen Datensatz angewendet, anschließend wird der Datensatz wieder zurück in die Datenbank geschrieben. In der Praxis wiederholen sich bestimmte Bereinigungsaufträge (auch in Varianten), daher sind Standardfunktionen[133] zur Bearbeitung einer spezifischen Aufgabe ausgelagert und können je nach Bedarf in das Reparaturprogramm eingebunden werden.

[130] Seite „Eingabehilfen für die Katalogisierung" im WinIBW3-Handbuch: https://verbundwiki.gbv.de/pages/viewpage.action?pageId=16711726 (14.05.2018).
[131] Siehe Kap. 4.4.2.
[132] Library CSDBC, vgl. Smit/Sutherland 2002.
[133] Z. B. zum Ändern der Standortangabe oder zur Berechnung von Prüfziffern.

Die folgende Tabelle führt eine Auswahl von Datenbereinigungsmaßnahmen[134] der VZG auf, mit Zuordnung von IQ-Dimensionen nach der DGIQ-Strukturierung.[135] Bei den aufgeführten Maßnahmen wurden zwischen einigen Tausend und 1–2 Mio. Datensätze bearbeitet.

Maßnahme	Nutznießer	Anlass Dimensionen der Datenqualität
Entfernen von Leerzeichen am Ende von Unterfeldern in der Signaturkategorie	Bibliothek	Auftrag von Bibliothek; unbedeutender Datenfehler, aber einfach umzusetzen Dimension: Fehlerfreiheit; Einheitliche Darstellung
Änderung der Standortangabe wegen Umzug eines Teilbestands in ein Außenmagazin	Bibliothek	Auftrag von Bibliothek; kein Datenfehler; Dimension: Aktualität
Löschen der Signaturkategorie bei MTM-Überordnung und Schriftenreihen	Bibliothek	Nach abgeschlossener Datenkonvertierung zur Einhaltung der Katalogisierungsrichtlinie Dimension: Einheitliche Darstellung
Änderung des Ausleihindikators für Rara-Bestand einer Bibliothek	Bibliothek	Im Zusammenhang mit der Einrichtung des Ausleihmoduls, um die gewünschte Ausleihpolitik für diesen Teilbestand realisieren zu können Dimension: Wertschöpfung, Fehlerfreiheit
Ergänzen eines Entsäuerungsvermerks in Titel- und Exemplarsatz anhand von PPN- und EPN-Listen	Bibliothek	Auftrag von Bibliothek; kein Datenfehler; Zur Verbesserung der Datenqualität durch Anreicherung von Informationen Dimension: Wertschöpfung
Änderung der URL von *http* zu *https* bei Online-Ressourcen eines Anbieters	Verbund	Zur Aufrechterhaltung des Onlinezugriffs für Bibliotheksbenutzer Dimension: Aktualität
Korrektur alter zweistelliger Ländercodes	Verbund	Datenfehler; Anpassung an die aktualisierte Katalogisierungsrichtlinie Dimension: Einheitliche Darstellung
Ersetzen der Zeichen "/l" bzw. "/L" durch die korrekte Zeichenrepräsentation mit Oktal-Wert 261 bzw. 241.	Bibliothek Verbund	Datenfehler; nach abgeschlossener Datenkonvertierung; Erfassungsmängel im Altsystem (falscher Zeichensatz bei einzelnen Zeichen) Dimension: Fehlerfreiheit
Ergänzen von IMD-Feldern	Verbund	Datenfehler; Einhaltung des Regelwerks RDA Dimension: Vollständigkeit
Ändern des Materialart-Codes bei Noten, dabei Ergänzung eines Codes für „Veröffentlichungsart und Inhalt“	Verbund	Absprache für K10plus; zur Einhaltung des zukünftigen Regelwerks Dimension: Relevanz; Einheitliche Darstellung

Tab. 2: VZG-Bereinigungsmaßnahmen mit Dimensionen der Datenqualität

Datenanalyse und -bereinigungen bei Bibliothekssystem-Migrationen

Die Washington State University Libraries (WSUL) hatten bereits 2008 damit begonnen, fehlende bzw. doppelte OCLC-Nummern (OCN) in MARC-Feld `001` zu bereinigen. In Vorausschau auf die geplante Migration des Bibliothekssystems wurde ein Jahr vor Vertragsunterzeichnung eine Arbeitsgruppe zur Durchführung von speziellen Datenkorrektur-Projekten eingesetzt. Problematisch waren die sog. Hausregeln bei der Katalogisierungspraxis und die daraus folgenden Abweichungen vom nationalen Katalogisierungsstandard. Insgesamt beschäftigten sich die WSUL-Bibliotheken über einen

[134] Die Bereinigungsaufträge werden über das Ticketsystem JIRA (Atlassian) verwaltet.
[135] Siehe Kap. 3.2.1.

Zeitraum von zwei Jahren mit Datenbereinigungsmaßnahmen, um die Datenmigration zu Alma bestmöglich vorzubereiten.[136]

Auch die Bibliotheken des Österreichischen Bibliothekverbundes (OBV) nutzen die im Projektablauf vorgesehene 6-monatige „Getting-Ready-Phase" für Datenbereinigungen im Altsystem. Eine erste Testmigration der Daten war bereits erfolgt, sodass in mehreren Arbeitsgruppen die Daten in Aleph und in Alma analysiert werden konnten. Daraus resultierten umfangreiche Bereinigungsarbeiten und Datenanpassungen, um die Erfordernisse des Zielformats bestmöglich erfüllen zu können. Zusätzlich konnten so die Vorgaben für die endgültige Datenmigration weiter spezifiziert werden.[137]

3.2.3 Qualitätsmanagement für Metadaten in Bibliotheken

Dem Datenqualitätsmanagement wird in Bibliotheken offenbar keine große Bedeutung beigemessen, zumindest ist dies selten Thema in der deutschsprachigen bibliothekarischen Fachpresse, wenn man von entsprechenden Hinweisen zu Maßnahmen im Zusammenhang mit einer Bibliothekssystem-Migration absieht. Dies mag auch damit begründet sein, dass es einer kritischen Beurteilung der eigenen Arbeit bedarf, um eine Notwendigkeit von Kontrollverfahren zu erkennen.

In Zeiten großer Datenmengen, sei es in Verbunddatenbanken oder Discovery-Systemen, fallen Erfassungsfehler nur selten auf. Gerade rechercherelevante Datenmängel bleiben häufig unentdeckt, da in der Regel ausreichend Treffer gefunden werden. Daher sind formalisierte Verfahren mit standardisierten Tools notwendig, um Bibliotheksdaten einer genaueren Prüfung zu unterziehen. Vorab sollte eine Bibliothek Kriterien festlegen, nach denen die Daten beurteilt werden.

Eine Ausnahme stellen daher die Aktivitäten der ZBW - Leibniz-Informationszentrum Wirtschaft dar, dort sind Datenqualitätsmaßnahmen als Routineaufgabe etabliert. Des Weiteren ist das Projekt „Metadatenmanagement" in der Bibliothek der ETH Zürich zu nennen, das u. a. Datenanalysen und -bereinigungen zum Gegenstand hatte.

ZBW Kiel/Hamburg[138]

In der ZBW sind in der Gruppe Abteilung „Integrierte Erwerbung und Katalogisierung" mehrere Stellen für Metadaten-Manager angesiedelt, zu deren Aufgaben u. a. „*Datenanalysen für das Qualitätsmanagement von Metadaten* [… und die] *Unterstützung bei der Fehleranalyse in Informationssystemen*"[139] gehören.

Als Grundlage der Maßnahmen zur Datenqualität dient ein 2015 entwickeltes „Konzept zum Datenqualitätsmanagement mit Empfehlungen für die künftige Arbeit mit Metadaten in der ZBW"[140]. Zu diesem Zweck wird dem Team „Metadaten-Management" monatlich ein Abzug der im CBS als RDA-Aufnahme gekennzeichneten Datensätze mit ZBW-Besitznachweis zur Verfügung gestellt. Die Textdatei mit Daten im PICA+-Format wird von den Mitarbeitern mit Hilfe regulärer Ausdrücke auf

[136] Vgl. Zhu/Spidal 2015, S. 258 und 268.
[137] Vgl. Kann 2017, S. 568; Köpf 2017, Folie 13.
[138] Informationen zum Datenqualitätsmanagement in der ZBW vgl. Hemme 2017.
[139] ZBW 2017b.
[140] ZBW 2017a, S. 12.

Datenfehler überprüft.[141] Hierfür wurde eine umfangreiche Liste der anzuwendenden Suchkriterien entwickelt.

Es werden die Inhalte einzelner Felder geprüft, z. B. das Vorhandensein zulässiger bzw. bibliotheksintern vereinbarter Selektionsschlüssel[142] bei unselbständigen Werken. Die Nutzung der Facette „Sprache" in einem Discovery-System führt nur dann zu repräsentativen Ergebnissen, wenn gewährleistet ist, dass bei (fast) allen Datensätzen eine Dokumentsprache angegeben wurde, daher wird in der ZBW gezielt nach Datensätzen mit fehlendem Sprachcode gesucht. Ebenso können Kategorien auf Abhängigkeiten untereinander untersucht werden.[143]

Mit Hilfe dieser Verfahren wird sichergestellt, dass bei Katalogisaten, die nach RDA erstellt wurden, alle RDA-Kernelemente im Datensatz enthalten sind, um die Vorgabe der Arbeitsstelle für Standardisierung (AfS) als Mindeststandard für die Erschließung nach RDA[144] zu erfüllen. Ebenfalls wird für eine Auswahl rechercherelevanter Felder das Einhalten der GBV-Katalogisierungsrichtlinie auf gültige Felder und Feldinhalte überwacht, soweit dies mit technischen Verfahren möglich ist.

Die ermittelten Problemfälle werden einer Mitarbeiterin zur Korrektur übermittelt. Weitere Personen werden bei Bedarf für spezifische Bereinigungsarbeiten herangezogen, z. B. für Felder mit elektronischen Adressen wegen der Besonderheiten von Persistent Identifiern wie DOI oder URN.

Das Metadatenmanagement in der ZBW umfasst nicht nur das Generieren von bibliografischen Daten, sondern ebenso eine routinemäßige Qualitätskontrolle. Diese Schwerpunktsetzung hat zum Ziel, den Bibliothekskunden mittels qualitativ hochwertiger Metadaten eine optimale Auffindbarkeit der für sie relevanten Informationen zu ermöglichen.[145]

Bibliothek der ETH Zürich[146]

Im Jahr 2014 startete die ETH-Bibliothek das dreijährige Pilotprojekt „Metadatenmanagement" mit dem Ziel, *„mögliche Aufgaben zu definieren, Abläufe zu automatisieren, neue Lösungen zu entwickeln und künftige Tätigkeiten auszuloten"*[147]. Das Projektteam ermittelte zunächst potenzielle Aktivitäten zum Datenqualitätsmanagement und führte diese anschließend durch. Die Datenanalyse fand primär im seinerzeit genutzten Bibliothekssystem Aleph statt, hier konnten mit Hilfe vielfältiger Recherchemöglichkeiten Unstimmigkeiten erkannt werden.

Mit Hilfe eines Exports im Aleph-Internformat (ASEQ) konnten umfangreiche Datenbereinigungen durchgeführt werden. Ebenfalls mit Zugriff auf das Betriebssystem wurden *„komplexere und schnellere Datenanalysen"*[148] durchgeführt, hiermit könnten direkte Datenbankabfragen gemeint sein.

[141] Es wird der Editor *EditPad Pro* (https://www.editpadpro.com/de.html, 14.05.2018) in Kombination mit dem Tool *RegExBuddy* (http://www.regexbuddy.com/index.html, 14.05.2018) eingesetzt; damit lassen sich auch sehr große Dateien auf komfortable Weise unter Nutzung komplexer regulärer Ausdrücke analysieren.
[142] Vgl. Verbundzentrale des GBV 2006.
[143] Beispielsweise wird geprüft, ob bei Vorhandensein der Kategorie für „Art des Inhalts" mit dem Text „Festschrift" auch eine Personenkategorie mit Beziehungskennzeichen „GefeierteR" vorhanden ist.
[144] Für den D-A-CH-Raum: vgl. Deutsche Nationalbibliothek 2018.
[145] Vgl. ZBW 2014, S. 12.
[146] Vgl. Pfister et al. 2017; Cavegn-Pfister et al. 2018; Bissegger/Wittwer 2016.
[147] Pfister et al. 2017, S. 22.
[148] Pfister et al. 2017, S. 23.

Im weiteren Projektverlauf wurde Datenmappings erstellt, außerdem ein Datenflussdiagramm zur Visualisierung der Metadaten-Datenflüsse zwischen den zahlreichen in der ETH-Bibliothek genutzten Systemen erarbeitet. Weitere Maßnahmen zur Datenanalyse und -bereinigung waren im Rahmen der Einführung von RDA und GND erforderlich. Zu diesem Zweck wurden verschiedene Metadaten-management-Tools getestet, u. a. d:swarm, Metafacture, MarcEdit und OpenRefine.[149]

Resümee aus der Pilotphase war vor allem die Erkenntnis, dass für die vielfältigen Arbeiten im Metadatenmanagement hochwertige Verfahren zur Datenanalyse unverzichtbar sind. Ebenfalls erkannte die Bibliothek die Relevanz eines professionellen Metadatenmanagements im Zusammenhang mit neuen bibliothekarischen Arbeitsfeldern und etablierte daher die Aufgaben des Projektteams dauerhaft in einem abteilungsübergreifenden „Netzwerk Metadatenmanagement“.[150]

3.2.4 Dublettenerkennung und -behandlung

Als Folge mangelnder Vollständigkeit, Fehlerfreiheit oder Konsistenz können in einem Informations-system redundante Informationen gespeichert sein. Die Problematik von Mehrfachnachweisen für bibliothekarische Metadaten (vereinfachend: Dubletten[151]) hat sich mit der Zunahme IT-gestützter Verfahren in Bibliotheken und der daraus resultierenden Datenmenge erheblich verstärkt.[152] Mit Dubletten sind im Folgenden mehrere Datensätze mit unterschiedlichen Inhalten für dieselbe Ressource gemeint. Allgemeiner formuliert handelt es sich um *„semantisch äquivalente Datensätze […], die das gleiche Realweltobjekt darstellen.“*[153]

Für die Existenz von Dubletten in einer bibliografischen Datenbank kommen verschiedene Ursachen in Betracht. Bei der intellektuellen Erfassung von Katalogisaten spielen eine nicht ausreichende vorherige Recherche, das Nicht-Erkennen von übereinstimmenden Merkmalen, eine individuelle Auslegung der Katalogisierungsregeln sowie Tippfehler bzw. Schreibvarianten eine Rolle. Als Lösung bieten sich problemorientierte Mitarbeiterschulungen und eine adäquate Werkzeugunterstützung an. Bei einer Datenintegration, resp. dem Offline-Import von Metadaten, können (zu) strenge oder zu wenige Dublettenkriterien die Erkennung von Dublettenkandidaten verhindern.

Sind Dubletten vorhanden, entspricht die Zahl der entsprechenden Datensätze nicht der Anzahl tatsächlich vorhandener Objekte. Überdies ist in einem Verbundkatalog die gezielte Recherche erschwert, wenn für das Ermitteln des Besitznachweises einer bestimmten Bibliothek mehrere gleichartige Titelrepräsentationen durchgeblättert werden müssen.

Eine Datenintegration findet i. d. R. in drei Schritten[154] statt. Im ersten Schritt werden die Daten einer Datenquelle strukturell in das Zielformat überführt. Nach dieser Datenkonvertierung erfolgt eine Dublettenprüfung, dabei werden ggf. weitere Repräsentationen desselben (Daten-)Objekts ausfindig gemacht. Zumeist findet eine abschließende Datenfusion statt, bei der der Zieldatensatz mit Attributen aus dem Quelldatensatz angereichert wird. Auf eine Fusion kann bei geringer Qualität der Datenquelle ggf. verzichtet werden, so bei dem auf S. 35 vorgestellten Projekt der UB Klagenfurt.

[149] Siehe Kap. 3.3.
[150] Cavegn-Pfister et al. 2018.
[151] Nicht gemeint sind mehrfach vorhandene Exemplare eines Werkes in einer Bibliothek.
[152] Vgl. Reichart/Mönnich 1994, S. 193.
[153] Köppen et al. 2014, S. 88.
[154] Vgl. Bleiholder/Schmid 2015, S. 122.

Zur Identifizierung von Dubletten in bibliografischen Metadaten ist das Vorhandensein eindeutiger Identifier in den Daten von Vorteil. Verbreitet ist die Auswertung der Standardnummern ISBN, ISSN, ZDB und GND als eindeutiges Merkmal. Allerdings können einem Objekt mehrere Standardnummern zugeordnet sein, daher ist für eine Dublettenerkennung die Auswertung weiterer Datenfelder sinnvoll.

Bereits Mitte der 90er Jahre sind Dublettenerkennungs-Mechanismen in Bibliotheksdatenbanken im Einsatz. Bei dem 1993 für die UB Karlsruhe entwickelten Verfahren[155] werden beim Speichern eines Datensatzes die Inhalte bestimmter Felder mit dem bereits vorhandenen Datenbestand abgeglichen. Die Steuerung der Felder und ihrer Gewichtung erfolgt über eine Parameterdatei. Zur Überprüfung zweier Dublettenkandidaten werden die jeweiligen Feldinhalte in kategoriespezifisch normierter Form miteinander verglichen.

Die dahinter stehende grundsätzliche Idee wird ebenfalls bei anderen Dublettencheck-Verfahren im Bibliotheksbereich angewendet, so auch für das bei Lohrum, Schneider et al.[156] beschriebene Eliminieren von Dubletten bei einer verteilten Suche. Wesentlich sind dabei diese attributspezifischen Normierungsschritte:[157]

- Feldspezifische Normierung
- Feldspezifischer Vergleich (auf Gleichheit bzw. auf Ähnlichkeit)
- Feldspezifische Vergabe von Gewichten
- Berechnung des Gesamtgewichts

Als Dublette wird ein Datensatz erkannt, wenn dessen Gesamtgewicht einen vorher festgelegten Schwellwert erreicht bzw. überschreitet.

Ähnlich wird beim Import von Bibliotheksdaten in die hbz-Verbunddatenbank verfahren. Dem Abgleich vorgeschaltet ist für jeden Datensatz die Vergabe eines Kennbuchstabens, der die Satzart[158] repräsentiert. Im sog. Match-and-Merge-Verfahren (M+M) werden nur Datensätze mit demselben Kennbuchstaben miteinander abgeglichen. Für den Vergleich werden je nach Satzart unterschiedliche Felder herangezogen, für die individuelle Match-Kriterien festgelegt sind. Auch hier erfolgt eine Normierung der Feldinhalte zur Berücksichtigung von Erfassungsvarianten.[159]

Das in der Verbundzentrale des GBV genutzte Match-and-Merge-Verfahren dient ebenfalls der Reduzierung bzw. Vermeidung von Dubletten beim Datenimport in den Verbundkatalog, es verläuft zweistufig. Im ersten (Match-)Schritt werden die zu verwendenden Selektionskriterien festgelegt, um für jede zu importierende Aufnahme eine Liste von potenziellen Dublettensätzen im CBS zu erzeugen. Zusätzlich werden zwei Schwellwerte (*match limit* und *candidate limit*) für den späteren Vergleich definiert. Das Evaluationsmodul[160] berechnet für jeden Datensatz aus der Menge der Dublettenkandidaten dessen Ähnlichkeit mit der Quellaufnahme. Hierbei werden die bibliografischen Datenpaare unter Nutzung einer Konfigurationstabelle verglichen, in der materialspezifisch die zu berücksichtigenden Felder bzw. Feldgruppen definiert sind. Für diese Elemente werden über feldspezifische Vergleichsfunktionen mit Normierung des Feldinhalts zunächst die einzelnen

[155] Vgl. Reichart/Mönnich 1994, S. 207.
[156] Vgl. Lohrum et al. 1999.
[157] Vgl. Rusch 1999, S. 3.
[158] Z. B. Monografie, Mehrteilige Monografie, Monografische Reihe/Zeitschrift u. a.
[159] Vgl. Block 2017, S. 2–4.
[160] Vgl. Sutherland et al. 2009.

Elementähnlichkeiten ermittelt. Außerdem wird pro Element auf Grund des Vergleichs ein Rückgabestatus mit einem Gewichtungsfaktor gesetzt. Die Summe der gewichteten Elementähnlichkeiten ergibt den endgültigen Ähnlichkeitswert für den Vergleich zweier Datensätze. Die Evaluationssoftware ermittelt einen Ähnlichkeitswert (zwischen 0 und 1), der vom Programm für den Datenimport ausgewertet wird. Wenn der beste Kandidat das *match limit* überschreitet, wird er als Treffer-Dublette für die Quellaufnahme gewertet. Im zweiten (Merge-)Schritt werden für das gefundene Datensatzpaar anhand einer Konfigurationstabelle diejenigen Kategorien ermittelt, die aus dem Quelldatensatz in den Zieldatensatz übernommen werden. Dessen bibliografische Felder können dabei ergänzt oder ersetzt werden. Liegt die Ähnlichkeit des Dublettenkandidaten unter dem *candidate limit*, wird der Quelldatensatz als Novum importiert. Zur späteren manuellen Prüfung kann ein Datensatz als sog. B-Novum mit entsprechender Kennzeichnung eingespielt werden, wenn der ermittelte Ähnlichkeitswert zwischen den beiden Schwellwerten liegt.

Im kommerziellen Bereich, insbesondere bei der Nutzung von CRM-Systemen oder beim Einsatz eines Data Warehouse, ist das Erkennen von Dubletten bei (Kunden-)Adressdaten von großer Bedeutung. Hier sind regelmäßig weitere Algorithmen zur Dublettenerkennung implementiert, so z. B. die Editier-Distanz nach Levenshtein als Maß für die Ähnlichkeit zweier Zeichenketten.[161] Berücksichtigt wird dabei die minimale Anzahl der Operationen bei der Überführung einer Zeichenkette in eine andere, und zwar durch Einfügen, Löschen oder Ersetzen eines einzelnen Zeichens. Der Aufwand bei diesem Verfahren ist abhängig von der Länge der Zeichenketten und die Laufzeit daher $O(m * n)$.[162]

Ein anderer mathematischer Ansatz für den Titelabgleich wurde in einem Projekt der UB Klagenfurt zur Dublettenerkennung angewandt, bei dem zunächst 3.100 zu ladende Datensätze gegen einen Bestand von ca. 400.000 Datensätzen abgeglichen werden sollten.[163] Als Dublettenkriterien wurden die Angaben zu ISSN bzw. ISBN, Jahres- und Auflagenzahl gewählt, weitere Unterschiede wurden bewusst ignoriert. Das Verfahren berücksichtigt Ähnlichkeitsmaße, wie sie auch im Relevanzranking von Suchmaschinen verwendet werden: den Jaccard-Koeffizienten und die euklidische Distanz von zwei Zeichenketten-Vektoren. In einer zweiten Versuchsanordnung kam der beim Online-Retrieval in der KOBV-Suchmaschine verwendete Ähnlichkeitswert für bibliografische Datensätze mit kategoriespezifischer Gewichtung der euklidischen Distanz zum Einsatz. Ergänzend zu den im Bibliotheksbereich verwendeten Verfahren, bei denen feldweise normalisierte Strings miteinander verglichen werden, werden hier zusätzlich Trigramme genutzt. Hintergrund ist der Ansatz, dass aus der Ähnlichkeit von Trigrammen auf die Ähnlichkeit der repräsentierten Zeichenketten geschlossen werden kann. Hierfür werden die beiden miteinander zu vergleichenden Zeichenketten jeweils schrittweise in Einheiten zu je 3 Zeichen zerlegt, dabei rückt der Beginn eines Trigramms jedes Mal eine Zeichenposition nach rechts. Die Anzahl der Trigramme pro Zeichenkette entspricht somit der Anzahl Zeichen im String. Die Liste der Trigramme für eine Zeichenkette bildet bei den folgenden Ähnlichkeitsberechnungen einen Vektor. Ist ein empirisch festzulegender Schwellwert überschritten, sind die beiden Datensätze als identisch anzusehen. Da Trigramme in linearer Zeit zu berechnen sind[164], ist der Zeitaufwand bei diesem Verfahren erheblich geringer als unter Berücksichtigung der

[161] Vgl. Leser/Naumann 2007, S. 335.
[162] Für m und n als Länge zweier Zeichenketten, vgl. Szott 2016, Folie 15.
[163] Vgl. Jele 2009. Es sollten Erfahrungen für die Integration von ca. 10.000 Daten einer Institutsbibliothek gewonnen werden. In einem weiteren Durchgang wurde ein Abgleich gegen 4,5 Mio. Datensätze durchgeführt.
[164] Vgl. Szott 2016, Folie 22.

Levenshtein-Distanz. Im Unterschied zu der im Bibliotheksbereich gängigen Praxis wurde in diesem Projekt auf das Einmischen ergänzender Angaben aus dem Quell-Datensatz verzichtet.

Eine Herausforderung bei der Behandlung von Dubletten stellt die Zusammenführung bibliografischer Metadaten aus heterogenen Quellen in einem Discovery-System dar, insbesondere bei unterschiedlicher Datenqualität. Hierzu folgen weitere Ausführungen in Kap. 3.4.

Sind Daten als Repräsentation eines Objekts in mehreren Ausgangsdatenbanken gespeichert, muss für Überschneidungen bei den Datenfeldern eine Priorisierung festgelegt sein. Dies gilt genauso für Daten von Bibliotheksnutzern, die sowohl im Bibliotheksmanagementsystem als auch im hochschuleigenen Informationssystem gespeichert sind. Beim Transfer von Daten in das jeweils andere System müssen die Informationen des führenden Systems Vorrang haben.

3.3 Tools für das Datenmanagement

Bibliotheken sind heutzutage verstärkt im Bereich digitaler Dienste bzw. Dienstleistungen wie Discovery-Systeme oder ERM aktiv. Dabei müssen Metadaten in verschiedenen Formaten (z. B. Excel, KBART, MAB2, MARC 21, MODS und PICA) über unterschiedliche Schnittstellen (u. a. SRU, Z39.50, OAI-PMH) extrahiert, aufbereitet und im Zielsystem bereitgestellt werden. Zur Unterstützung dieses ETL-Prozesses werden Verfahren zur Verarbeitung und Analyse von Metadaten benötigt. Hierfür ist im Bibliotheksbereich häufig Open-Source-Software im Einsatz.

Eine solche Software soll drei Kriterien erfüllen: Sie soll zur „*Validierung und einfache*[n] *Analyse von Datenlieferungen* [geeignet sein, die] *Anpassung von Datenlieferungen* [ermöglichen und ebenso das] *Durchführen eines ETL-Prozesses*“[165]. Mit Hilfe der Tools soll es vor allem möglich sein, Unstimmigkeiten in den Daten zu erkennen und konsistente Feldbelegungen zu gewährleisten; statistische Analysen der genutzten Felder sind hierfür hilfreich. Ebenfalls gehört dazu das Filtern von Datensätzen und Anpassen von Feldinhalten, wenn z. B. Daten aus einem Bibliotheksystem extrahiert und in einen Suchmaschinenindex geladen werden.

Einige Open-Source-Software-Tools erscheinen für den Einsatz im LOK-Projekt geeignet und werden daher exemplarisch einer genaueren Untersuchung unterzogen. Vorrangig geprüft werden die Möglichkeiten der Verarbeitung von Daten im PICA+-Format, da beim Datenexport per MARC 21 einige Kategorien nicht berücksichtigt werden, die für das LOK-Projekt von Bedeutung sind.[166] Nicht einbezogen in die Tests sind Softwarefunktionen, die nicht projektrelevant erscheinen.

3.3.1 Catmandu

Ein in der deutschen Bibliothekscommunity verbreitetes Toolkit zur Verarbeitung von Metadaten ist Catmandu[167], es wurde 2012 im Kontext der LibreCat-Kooperation[168] der Universitäten Ghent, Lund und Bielefeld entwickelt. Zusätzlich sind weitere Mitarbeiter aus anderen Institutionen an der Entwicklung beteiligt. Catmandu stellt eine große Anzahl von Perl-Modulen zur Verfügung, die die Abläufe des ETL-Prozesses bei der Verarbeitung von Metadaten vereinfachen.

Exemplarische Anwendungen aus den Bereichen Bibliothek, Museum und Archiv zeigen die Vielseitigkeit der potenziellen Einsatzgebiete.[169] Mitarbeiter der ZDB setzen Catmandu für Datenanalysen und -selektionen ein; dazu werden halbjährlich die ZDB-Daten in eine MongoDB geladen. Daraus können u. a. Zeitschriften einer Bibliothek mit Alleinbesitz ermittelt und als CSV-Datei exportiert werden. Darüber hinaus erfolgt die Aufbereitung von Daten für den ZDB-Import über Catmandu.[170]

Aus den angebotenen Downloadmöglichkeiten wurde für den Test eine virtuelle Maschine (VM) auf *Lubuntu*[171]-Basis gewählt.

---

[165] Pfeffer 2016, Folie 6. In Bibliotheken verwendet wird häufig Software, die auch für den Einsatz in der Lehre geeignet ist, da sie ohne spezielle Programmierkenntnisse bzw. großen technischen Aufwand genutzt werden kann.
[166] U. a. Selektionsschlüssel in Kategorie `70xx` und Verbuchungsnummer in `8200`.
[167] http://librecat.org/ (14.05.2018).
[168] Zweck ist die gemeinschaftlich betriebene Erstellung von Open-Source-Anwendungen für Bibliotheken.
[169] Vgl. http://librecat.org/use-cases.html (14.05.2018).
[170] Vgl. Rolschewski 2018.
[171] https://lubuntu.net/ (14.05.2018).

Die Verwaltung der zzt. mehr als 80 sog. „Distributionen“[172] erfolgt i. d. R. über CPAN, die Quellen sind auf GitHub verfügbar. Die Distributionen enthalten Module für

- *Importer* (Pakete für den Import von Daten aus einem Quellsystem),
- *Exporter* (Pakete für den Export von Daten in ein Zielsystem),
- *Stores* (Pakete zur Speicherung in einer Datenbank oder Suchmaschine wie MongoDB, LDAP oder ElasticSearch),
- *Fixes* (Kommandos zur Datentransformation).

Anwendung

Ausgeführt wird Catmandu über die Kommandozeile, alternativ können die Funktionen über ein Perl-Programm gesteuert werden. Das zentrale Kommando `convert` wird für das Herunterladen von Daten aus einer externen Datenquelle und das Transformieren von Datensätzen genutzt. Die Angaben zu *Importer* und *Exporter* können über spezifische Optionen erweitert werden. Zur Vereinfachung ist die Auslagerung der Kommando-Optionen in eine Konfigurationsdatei (`catmandu.yml`) möglich.

Daten aus einer externen Quelle können über verschiedene Schnittstellen wie OAI-PMH oder SRU heruntergeladen werden. Zur Datentransformation existieren zahlreiche Module für unterschiedliche Datenformate, u. a. PICA+, MARC, MAB, XML, JSON oder CSV, auch RDF kann generiert werden. Über Catmandu lassen sich Metadaten in Datenbanken wie MongoDB oder Volltextindizes wie Solr speichern, auslesen oder löschen.[173] Außerdem wird der Zugriff auf relationale DBMS per DBI unterstützt. Eine Konvertierung in das CSV-Format ist allerdings nur für Daten mit homogener Struktur sinnvoll, da lediglich die Felder des ersten Datensatzes aus der Datenmenge berücksichtigt werden. Für Bibliotheksdaten mit Wiederholungsfeldern und optionalen Kategorien bzw. Subfields ist eine CSV-Ausgabe daher nicht praktikabel.

Zentrales Element einer Datentransformation sind die Transformationsvorschriften, die sog. *Fixes*. Ein Fix kann beim Kommandoaufruf explizit angegeben werden, dies ist allerdings nur für einfache Transformationsregeln sinnvoll. Komplexe Regeln speichert man in einer separaten Datei und gibt diesen Dateinamen als Kommandozeilenparameter beim Aufruf mit.

Mit Hilfe der *Fix Language* sind komplexe Datenmanipulationen möglich, vor allem:

- Einfügen und Löschen von Feldern (`add_field`, `remove_field`),
- Setzen von Feld-Inhalten (`set`),
- Datenmanipulation (für Strings und andere Datenformate),
- Verarbeitung von Hashs und Arrays,
- Bedingungen (`if`, `unless`),
- Schleifen (do),
- Suche in Dateien und Datenbanken (`lookup`).[174]

[172] Gefolgt wird der Catmandu-Terminologie, die Pakete von Perl-Modulen als „Distribution“ bezeichnet. Vgl. http://librecat.org/distributions.html (14.05.2018).
[173] Liste der verfügbaren Export- und Importformate bzw. Schnittstellen vgl. ebd.
[174] Die Fix-Funktionen sind ausführlich in einem Cheat Sheet dokumentiert, vgl. Hochstenbach 2017. Zahlreiche Beispiele enthält die ältere Fassung unter http://librecat.org/catmandu/2013/06/21/catmandu-cheat-sheet.html, ebenfalls die Dokumentation auf GitHub https://github.com/LibreCat/Catmandu/wiki/Fixes-Cheat-Sheet (beide 14.05.2018).

Eine *Breaker*-Distribution[175] ermöglicht die TSV-Ausgabe von Identifier, Feldbezeichnung und Feldinhalt. Für eine verwertbare Ausgabe von MARC- oder PICA-Daten ist beim Aufruf ein spezieller Handler als Option anzugeben, da das im Hintergrund genutzte JSON-Format die Feldbezeichnungen als Bestandteil der Daten speichert. Die Ausgabe erfolgt zeilenweise je Kategorie bzw. Subfield, bei PICA-Daten auch getrennt nach Occurrence.[176]

Ausgehend von der *Breaker*-Ausgabe generiert die *Stat*-Distribution[177] eine Statistik über die in der Datenmenge vorhandenen Datenfelder inkl. Subfield (bei PICA+ ggf. zusätzlich die Occurrence-Angabe). Für das LOK-Projekt sind vor allem diese Informationen relevant:

- `count` Anzahl Datensätze mit diesem Feld/Subfield
- `zeros` Anzahl Datensätze ohne dieses Feld/Subfield
- `min` Minimale Häufigkeit innerhalb eines Datensatzes
- `max` Maximale Häufigkeit innerhalb eines Datensatzes

Anhand dieser Statistik lassen sich auch Kategorien identifizieren, die innerhalb eines PICA-Datensatzes mehrfach vorkommen.[178] Gerade zur Vorbereitung von ETL-Prozessen kommt der gründlichen Datenanalyse ein große Bedeutung zu; so liefert die *Stat*-Funktion wesentliche Aufschlüsse über die Bandbreite der genutzten Kategorien und damit zur Komplexität der lokalen Katalogisate einer Bibliothek.[179]

PICA-spezifische Funktionen

Die Catmandu-Distribution zur Verarbeitung von PICA-Daten[180] umfasst nur wenige Module, daher sind die Anwendungsmöglichkeiten beschränkt. Zusätzlich existiert neuerdings ein Handler für PICA-Daten im *Breaker*-Modul, sodass darauf aufsetzend eine Statistik der in der Datenmenge vorhandenen PICA-Kategorien erzeugt werden kann.

Die *Importer*- bzw. *Exporter*-Funktionen „PICA“ bieten diese Formatvarianten:

- `plain` menschenlesbare PICA+-Darstellung mit „$” zur Subfield-Kennzeichnung[181]
- `binary` mit binären Subfield-, Feldende- und Datensatzende-Kennzeichen
- `plus` „normalisiertes“ PICA+; wie `binary`, aber pro Datensatz auf eigener Zeile[182]
- `xml` PICA XML (Default-Einstellung)
- `ppxml` PICA XML-Variante der DNB

Nach Tests mit den verfügbaren Formatvarianten erscheint das `plain`-Format für die Zwecke im LOK-Projekt am besten geeignet, ebenso wird das PICA-XML-Format ggf. nutzbar sein. Für die Ausgabevarianten mit binären Trennzeichen wäre ein entsprechendes Programm zum Parsen der Dateiinhalte erforderlich.

[175] https://metacpan.org/pod/Catmandu::Breaker (14.05.2018).
[176] Siehe *Beispiel 1: Anwendung des Breaker-Moduls mit PICA-Daten* in Anhang 4.
[177] https://metacpan.org/pod/Catmandu::Stat (26.05.2018).
[178] Für Transformationen im LOK-Projekt von Bedeutung, siehe Kap. 5.3.
[179] Siehe *Beispiel 2: Anwendung der Stat-Distribution mit PICA-Daten* in Anhang 4.
[180] https://metacpan.org/pod/Catmandu::PICA (14.05.2018).
[181] Entspricht der Formatsyntax „P“ in CBS und LBS.
[182] Siehe *Beispiel 3: SRU-Download mit Exporterformat PICA, Typ plus* in Anhang 4.

Zur Transformation und Datenmanipulation sind vor allem diese speziell auf das PICA-Format zugeschnittenen *Fix*-Funktionen geeignet:

- `pica_each` Bestandteil einer do-Schleife: die *Fix*-Kommandos werden innerhalb des Feld-Kontextes ausgeführt
- `pica_match` Bestandteil einer `if`-Bedingung: prüft Vorhandensein eines Subfields
- `pica_map` kopiert Feldinhalt mit neuer Feld-Bezeichnung
- `pica_add` fügt ein weiteres Subfield ein
- `pica_set` überschreibt des Inhalt eines Subfields

Damit lassen sich einfache Transformationen ausführen. Verschiedene für das LOK-Projekt nützliche Verarbeitungsschritte konnten getestet werden, unter anderem die Ausgabe von Datensätzen mit einem bestimmten Inhalt in einer bestimmten Kategorie; damit können z. B. Erfassungsfehler in der Kategorie für den Selektionsschlüssel erkannt werden, die ggf. die gewünschte Unterdrückung des Datensatzes für die OPAC-Anzeige verhindern. Die Ausgabe der PPNs aller Datensätze ist für eine Nutzung in weiteren Anwendungen dienlich. Wegen der besonderen Bedeutung von Fernleihtiteln für die Datenbereinigung im LOK-Projekt[183] wurde die Anzahl der Datensätze mit dem Begriff „Fernleihe“ im Titel ermittelt. Ebenfalls war es möglich, Datensätze, deren Titel mit „Fernleihe“ beginnt, aus der Zieldatenmenge auszuschließen.

Die Catmandu-Webseite dokumentiert alle Funktionen mit Angabe von Beispielen, allerdings sind viele der dort aufgeführten Funktionen nur für die Bearbeitung von Formaten wie JSON oder Solr-Daten geeignet, für die bibliothekarischen Datenformate MARC und PICA jedoch nicht anwendbar. Eine große Anzahl von Beispielkommandos zur Anwendung von *Mapping Rules* bei MARC-Daten ist dokumentiert[184], die teilweise auch auf PICA-Daten übertragbar sind. Auf den einzelnen Modulseiten zur Distribution `Catmandu::PICA` im CPAN-Repository sind nützliche Beispiele zu finden, die allerdings teils fehlerhafte Angaben enthalten. Dass weitere Funktionen ebenfalls auf PICA-Daten angewendet werden können, wie `Count` (als *Exporter*) zum Zählen der Datensätze einer Datei, ist den Dokumentationen nicht zu entnehmen.[185]

Catmandu ist ein vielseitiges, durch eine engagierte Community[186] getragenes Werkzeug. Gerade die *Fix*-Funktionen sind sehr mächtig, erfordern aber auch die Einarbeitung in eine eigene Meta-Sprache. Catmandu-Funktionen lassen sich darüber hinaus als Module in komplexen Workflows zusammen mit anderen Softwaretools einsetzen.

Gerade für die bibliothekarischen Datenformate PICA und MARC werden leider noch nicht alle Funktionen zur Transformation unterstützt. Um herauszufinden, welche *Fix*-Funktionen tatsächlich mit PICA-Daten genutzt werden können, sind zeitaufwendige Tests notwendig. Die Informationen zur praktischen Anwendung von Transformationen sind verstreut auf zahlreichen Webseiten zu finden.[187]

[183] Siehe Kap. 5.4.
[184] Vgl. https://github.com/LibreCat/Catmandu-MARC/wiki/Mapping-rules (14.05.2018).
[185] Vgl. https://metacpan.org/release/Catmandu (14.05.2018).
[186] Da für die Verarbeitung von PICA-Daten nur wenige Module existieren, werden Anregungen zur Erweiterung des Funktionsumfangs durch die Entwickler zeitnah aufgegriffen und umgesetzt. So wurde innerhalb weniger Tage die Erweiterung der *Breaker*-Distribution um einen PICA-Handler realisiert.
[187] Voß thematisiert in seinem Vortrag „Catmandu Documentation“ (vgl. Voß 2016) die unübersichtlichen Informationsquellen und macht Verbesserungsvorschläge zur Strukturierung der Catmandu-Dokumentation.

Der Einsatz von Catmandu im LOK-Projekt ist denkbar, falls der SRU-Zugriff zusätzlich auf die sog. OWC-Kataloge von LBS-Bibliotheken im GBV ausgeweitet würde.[188] Sind diese Rahmenbedingungen gegeben, lässt sich für die Nutzung von Catmandu resümieren:

Vorteile

- Spezifische Recherche zur Titelauswahl über SRU-Kommandosyntax möglich;
- Wenige Schritte zum Erzeugen der PICA+-Datei;
- Flexible Steuerung des Ablaufs durch einfache Anpassung der Fix-Datei.

Nachteile

- Funktion `remove_field` noch nicht verfügbar, um nicht benötigte PICA-Kategorien aus der Ergebnismenge entfernen zu können;[189]
- Unklar, ob die erforderlichen (String-)Konvertierungen möglich sind;
- Linux als technische Basis erforderlich; alternativ: Windows mit Perl-Implementierung.

3.3.2 OpenRefine

Umfangreiche Funktionen zur Datenanalyse und -transformation enthält die lokale Web-Anwendung OpenRefine[190]. Primäres Anwendungsszenario ist die Datenbereinigung, aber auch das Aufbereiten und Anreichern von statistischen und textbasierten Daten. Zum Anwenderkreis gehören vor allem Bibliothekare, (Daten-)Journalisten und Wissenschaftler.

Die browserbasierte Oberfläche ähnelt klassischen Tabellenverarbeitungsprogrammen. Das Programm wird lokal installiert und ist unter Windows, Linux und Mac nutzbar, es benötigt eine Java JRE. OpenRefine wurde unter Windows in Version 2.7 getestet. Der Programmstart aktiviert einen lokalen Webserver, dessen HTTP-Kommandos in einem Konsolenfenster verfolgt werden können. Automatisch öffnet sich ein Browser-Tab mit der Anwendung.[191] Beim Import von Daten wird ein sog. *Project* angelegt; die zugehörigen Projektdateien sind im *Workspace*, im lokalen Userverzeichnis, gespeichert.

Der Funktionsumfang lässt sich durch Erweiterungen ausbauen, u. a. zur Nutzung im Zusammenspiel mit der GOKb[192] oder zur Ausgabe statistischer Informationen zu den Werten einer Spalte[193]. Der Zugriff auf OpenRefine-Projekte ist ebenfalls über eine API möglich.[194]

Zur Verarbeitung großer Dateien ist der Standardwert von 1 GB RAM nicht ausreichend[195], dann ist eine Vergrößerung des zugewiesenen Arbeitsspeichers nötig. Hierfür muss Java in einer 64-bit-Umgebung installiert sein.[196]

---

[188] Anzeigevariante des LBS-OPACs mit Präsentation der im Publikumskatalog unterdrückten Datensätze; erforderlich für den Zugriff auf Lokale Katalogisate, die häufig als „nicht sichtbar" gekennzeichnet sind. Vgl. den Abschnitt zu *SRU* in Kap. 5.2.2.

[189] Die Entwicklung der Funktion `pica_remove` ist in Planung, vgl. https://github.com/gbv/Catmandu-PICA/issues/60 (14.05.2018).

[190] http://openrefine.org/ (14.05.2018); unter diesem Namen seit 2012, vormals *Freebase Gridworks* bzw. ab 2010 *Google Refine*, vgl. http://kb.refinepro.com/2012/10/from-freebase-gridworks-to-google.html (14.05.2018).

[191] Zur technischen Architektur vgl. https://github.com/OpenRefine/OpenRefine/wiki/Architecture (15.05.2018).

[192] https://gokb.org/ (15.05.2018). Zur Integration heterogener Metadaten am Beispiel GOKb siehe Kap. 3.4.

[193] https://github.com/sparkica/refine-stats (15.05.2018).

[194] https://github.com/OpenRefine/OpenRefine/wiki/OpenRefine-API (15.05.2018).

Für den Datenimport sind verschiedene Quellen wählbar. Neben Dateien auf der lokalen Festplatte oder Google-Drive-Tabellen können auch Daten einer Webseite per URL integriert werden. OpenRefine kann zahlreiche Datenformate verarbeiten: XML, JSON, MARC, Excel u. a. m., das Datenformat wird i. d. R. automatisch erkannt. Zum Import von PICA+-Daten ist das Format *Line-based text files* geeignet. Binäre MARC-Daten werden beim Import automatisch in MARCXML umgewandelt, für Dateien im `.mrk`-Format bietet sich der Import als *Fixed-width field text files* an. Zur Festlegung einer Zeichencodierung kann aus einer umfangreichen Liste ausgewählt werden.

OpenRefine arbeitet spaltenbasiert. Auf die Zelleninhalte einer oder mehrerer Spalten können Filter und Facetten angewendet werden; dies ermöglicht eine komfortable Datenanalyse, da spezifische Zelleninhalte über reguläre Ausdrücke[197] identifiziert werden können. Datenänderungen können über die gesamte Datenmenge oder über die Ergebnismenge von Facetten bzw. Filtern durchgeführt werden. Mit Hilfe der elementaren Programmfunktionen Sortieren, Filtern, Clustern und Splitten bzw. Zusammenführen von Zelleninhalten lassen sich Daten in eine andere Struktur überführen. Zur Bearbeitung von Zelleninhalten und genauso bei der Facettierung wird standardmäßig die Transformationssprache GREL (General Refine Expression Language) verwendet.[198] Auch komplexe Datenmanipulationen können damit ohne Nutzung einer Programmierumgebung realisiert werden. Die durch die Transformation bewirkten Änderungen werden zunächst in einer Vorschau dargestellt, sodass die Auswirkungen des Transformationsausdrucks direkt erkennbar sind.

Alle Transformationsschritte werden projektbezogen gespeichert. Über die *Undo-/Redo*-Funktion können Aktionen rückgängig gemacht werden. Aufgrund der Möglichkeit, Transformationsschritte im JSON-Format zu exportieren und auf ein anderes OpenRefine-Projekt anwenden zu können, ist OpenRefine optimal für wiederkehrende Abläufe zur Bearbeitung von Daten geeignet.[199]

Zur automatisierten Anwendung in einem ETL-Prozess kann OpenRefine über eine HTTP-API angesprochen werden, dies ist für verschiedene Programmiersprachen realisiert. Unter Nutzung einer Python-Bibliothek lässt sich der Ablauf über ein Shellscript steuern, zuvor muss eine Datei mit den gewünschten Transformationsregeln erstellt werden. Per Script können dann die Dateien eines Eingabeverzeichnisses mit Hilfe der Transformationsregeln verarbeitet und in das gewünschte Ausgabeformat exportiert werden.[200]

Eine umfangreiche Dokumentation ist im Wiki des GitHub-Projekts zu finden. Sog. *Recipes* enthalten annotierte Beispiele zu GREL-Ausdrücken für konkrete Anwendungsfälle, beispielsweise zum Entfernen von Dubletten oder zur Umwandlung von ISBN-10 zu ISBN-13.[201] Zur praktischen Nutzung von OpenRefine mit Daten aus dem Bibliothekskontext gibt es ebenfalls Hilfestellungen.[202]

---

[195] In der OpenRefine-Community wurden hierzu Benchmark-Tests durchgeführt, vgl. https://groups.google.com/d/msg/openrefine/-loChQe4CNg/eroRAq9_BwAJ (15.05.2018).
[196] Vgl. https://github.com/OpenRefine/OpenRefine/wiki/FAQ%3A-Allocate-More-Memory (15.05.2018).
[197] Nach Java-Syntax (`java.util.regex`).
[198] https://github.com/OpenRefine/OpenRefine/wiki/General-Refine-Expression-Language (15.05.2018).
[199] U. a. bei der Aktualisierung von Titellisten für GOKb, siehe Kap. 3.4.
[200] Vgl. https://github.com/openculturecons ulting/openrefine-batch (15.05.2018).
[201] https://github.com/OpenRefine/OpenRefine/wiki (15.05.2018).
[202] Vgl. https://github.com/felixlohmeier?utf8=%E2%9C%93&tab=repositories&q=openrefine (15.05.2018).

Interessant wäre die Möglichkeit des direkten Zugriffs auf eine Datenbank, gerade im Zusammenhang mit bibliothekarischen Daten. Für das LBS kommen bibliografische Daten aufgrund der Datenstruktur kaum in Betracht, aber zur Bereinigung von z. B. Benutzerdaten könnte OpenRefine ggf. nützlich sein. Im Rahmen einer finanziellen Zuwendung von Google Inc. News Lab an die zu diesem Zweck gegründete OpenRefine Foundation wird u. a. eine Erweiterung für die Anbindung von Datenbanken per JDBC entwickelt.[203] Der schriftlichen Vereinbarung zufolge ist der Auftrag bereits erledigt,[204] allerdings ist zzt. lediglich ein GitHub-Repositoy für den Datenimport aus einer Datenbank nach OpenRefine aufzufinden.[205] Als Workaround kann die *Template*-Funktion genutzt werden; damit ist das Erzeugen von entsprechenden SQL-Statements möglich.[206]

Beim Einsatz von OpenRefine im LOK-Projekt spielt eine direkte Bereinigung der Ausgangsdaten keine Rolle, da ein Re-Import von (bereinigten) bibliografischen Metadaten ins LBS nur mit unverhältnismäßig hohem Aufwand möglich wäre. Dafür hat sich OpenRefine bei der Bereitstellung von Excel-Dateien mit lokalen Katalogisaten zur Prüfung durch LBS-Bibliotheken bewährt. Das im LOK-Projekt angewendete Verfahren wird in Kap. 5.3 dargestellt.

OpenRefine bietet vielfältige Möglichkeiten zur Analyse und Manipulation von Daten. Die Einarbeitung in die Transformationssprache GREL wird durch praxisnahe Beispiele in der Dokumentation sowie in zahlreichen im Web verfügbaren Tutorials erleichtert. Die breite Palette an Funktionen ermöglicht komplexe Datentransformationen. Die tabellarische Darstellung und insbesondere die Vorschaufunktion für Transformationen erlaubt auch Anwendern ohne Programmierkenntnisse eine effektive Nutzung des Programms. Insbesondere für Daten in tabellarischen Formaten wie CSV sind keine besonderen Vorarbeiten beim Datenimport erforderlich; hingegen ist bei bibliothekarischen Metadaten im MARC- oder PICA-Format aufgrund ihrer Besonderheiten (u. a. wiederholbare Felder) eine Reihe von Transformationsschritten nötig, bis die Daten im Projekt sinnvoll strukturiert und für eine Datenanalyse vorbereitet sind. Sind die Transformationsregeln einmalig erstellt, kann OpenRefine als effektives Werkzeug für die Anwendung im LOK-Projekt genutzt werden.

3.3.3 Mable+ und MARCel

Die beiden Kommandozeilen-Tools zur automatischen Daten- und Fehleranalyse, Mable+[207] und MARCel[208], wurden im Zusammenhang mit der 2007 geschlossenen „Vereinbarung zur Strategischen Allianz zwischen BVB und KOBV“ entwickelt. Anlass war die Überführung der Daten von KOBV-Bibliotheken in die seitdem von beiden Verbünden gemeinsam genutzte Verbunddatenbank B3Kat. Mit Hilfe von Mable+ konnten die zur Migration der Bibliotheksdaten erforderlichen Datenanalysen durchgeführt werden.

Beide Programme können als Java-gestützte Open-Source-Software sowohl unter Windows als auch einem Unix-basierten Betriebssystem installiert werden.

[203] Vgl. https://github.com/OpenRefine/OpenRefine/pull/1394 (15.05.2018).
[204] Vgl. https://docs.google.com/document/d/1UwoT1nFk9zwqSlH8rmqKPmiLS2Liw7-KM5HTKD2VVi8 (15.05.2018).
[205] Vgl. https://github.com/tcbuzor/openrefine-db-extension (15.05.2018).
[206] Vgl. http://kb.refinepro.com/2014/04/prepare-sql-update-query-in-openrefine.html (15.05.2018).
[207] https://www.kobv.de/entwicklung/software/mable/ (15.05.2018).
[208] https://www.kobv.de/entwicklung/software/marcel/ (15.05.2018).

Mable+

Mit Mable+ können Daten im MAB2-Bandformat oder als MABXML analysiert werden. Dafür sind zahlreiche Prüfungen implementiert: generelle MAB2-Formatfehler, nicht erlaubte Zeichen, fehlende Felder, Verstöße in Bezug auf Wiederholungsfaktoren von Feldern sowie für einen Satztyp nicht zugelassene Felder. Das Programm gibt zusätzlich zur Fehlerprotokollierung eine Statistik über gefundene Fehler und Felder aus.[209]

MAB-Daten standen für einen Test nicht zur Verfügung, daher wurde auf die Installation von Mable+ verzichtet.

Mable+ wird trotz der weitgehenden Ablösung von MAB als Austauschformat weiter zur Datenanalyse genutzt, wenn sich Bibliotheken dem KOBV anschließen und deren Datenbestände in den B3Kat migriert werden. Das Programm kommt ebenso bei der Prüfung von Bibliotheksdaten für das KOBV-Portal[210] zum Einsatz.

MARCel

MARCel wird ebenfalls zur Unterstützung der Integration von Bibliotheksdaten in das KOBV-Portal genutzt, die Daten müssen dazu im Format MARCXML vorliegen. Neben der formalen Prüfung der Katalogdaten erfolgt eine statistische Auswertung über die Verteilung der MARC-Felder. Genutzt wurde MARCel in der Version 0.1 vom 09.04.2015.

Formale Fehler werden in einer CSV-Datei protokolliert. So wurde z. B. bei einem Test mit Daten eines SRU-Exports aus einem LBS-OPAC im Format MARCXML für jeden Datensatz ein fehlerhafter Leader bemängelt. Ursache ist die Verwendung von Buchstaben als Dummy-Angabe für Datensatzlänge und Startadresse des ersten Kontrollfeldes.[211] SRU-Exporte der DNB enthalten stattdessen „00000“ als Dummy-Wert und werden daher von MARCel als fehlerfrei erkannt. Bei Daten, die direkt als MARC 21 aus dem CBS exportiert und anschließend zu MARCXML konvertiert wurden, sind im Leader naturgemäß die tatsächlichen Werte eingetragen und somit ebenfalls einwandfrei.

Eine Protokolldatei im Textformat enthält neben der Anzahl der verarbeiteten Datensätze und der Feldstatistik (Abschnitt *„Field allocations“*) eine Liste von Datensatz-IDs mit fehlenden Feldern. Bei den Tests wurde allerdings das MARC-Feld 245a in allen Datensätzen als fehlend erkannt, obwohl das Feld in den verarbeiteten Dateien enthalten ist, dies ist auch der Feldstatistik zu entnehmen. Die Bedeutung des Protokollabschnitts *„a2 allocations“* blieb unklar, da bei den Tests keine Eintragungen generiert wurden und die Dokumentation keine entsprechenden Erläuterungen enthält.

Die statistische Auswertung enthält für jedes in der Datenmenge vorhandene Feld bzw. Subfield:

- Absolute Anzahl Vorkommen in der Datenmenge,
- Anzahl Datensätze mit diesem Feld,
- Angabe in % bezogen auf die Datenmenge,
- Anzahl Datensätze mit Mehrfachauftreten des Feldes,
- Angabe in % bezogen auf die Gesamtdatenmenge.

[209] Vgl. https://www.kobv.de/entwicklung/software/mable/dokumentation/ (15.05.2018).
[210] Regionales Bibliotheksportal für Berlin und Brandenburg: https://www.kobv.de/services/recherche/portal/ (15.05.2018).
[211] Es werden die Zeichenketten „xxxxx“ bzw. „yyyyy“ verwendet.

Zusätzlich ermöglicht eine Datenbankimport-Funktion für MySQL[212] inhaltliche Datenanalysen. Voraussetzung ist lediglich eine eingerichtete Datenbank[213], da beim Import die Datenbankstruktur erstellt und u. a. für jedes MARC-Feld eine eigene Tabelle angelegt wird.[214] Damit lassen sich spezifische Prüfungen auf Feld- bzw. Subfieldebene durchführen. Insbesondere durch die Nutzung regulärer Ausdrücke können Unregelmäßigkeiten in den Daten aufgedeckt werden. KOBV-intern wird ein weiteres Programm „MARCel-Statistik“ zur statistischen Auswertung der Daten in der MySQL-Datenbank genutzt.

Die Steuerung von Programmausgabe und Datenbankanbindung ist über Kommandozeilenoptionen möglich. Diese sind auf GitHub dokumentiert, dort finden sich zusätzlich eine Installationsanleitung und weitere Hinweise zur Anwendung von MARCel sowie der Sourcecode.[215]

Da der SRU-Export von LBS-Daten im MARCXML-Format nur eine Teilmenge der Exemplar-Kategorien berücksichtigt, sind für die Nutzung im LOK-Projekt nicht alle wünschenswerten Datenanalysen durchführbar. Das ist allerdings nicht MARCel anzulasten, sondern der beschränkten Export-möglichkeit von Datensätzen, die im LBS erfasst wurden. Eine Alternative wäre Catmandu, das auch Datenexporte im Format PICA XML verarbeitet.

Die Verarbeitung großer Datenmengen stellte bei der Exploration ein Problem dar: Ein Test mit allen 232 Dateien eines Verzeichnisses (zusammen 2,5 GB) führte zum Programmabbruch.[216] Eine Teilmenge von 99 Dateien (1,1 GB) mit 655.044 Datensätzen (Alma-Exemplardaten) wurde hingegen in 1:37 Minuten erfolgreich verarbeitet.

MARCel ist ein unkompliziert zu bedienendes Werkzeug zur Analyse von MARC-Daten mit ausgezeichneter Performance. Nützlich ist die Speicherung der Daten in einer Datenbank für weitere Datenanalysen. Der im Unterschied zu Catmandu eingeschränkte Funktionsumfang vereinfacht allerdings die Einarbeitung und Handhabung erheblich.

3.3.4 Weitere Tools

Zur Unterstützung des Metadatenmanagements in Bibliotheken stehen eine Reihe weiterer Tools zur Verfügung. Dabei ist zu unterscheiden zwischen Programmen für den Umgang mit MARC-Daten einerseits und Software zur Datenanalyse bzw. zum Abbilden von ETL-Prozessen andererseits.

Die folgende Übersicht stellt exemplarisch (überwiegend) Open-Source-Software vor, die im Bibliothekskontext eine größere Verbreitung gefunden hat bzw. aus den Reihen der Bibliotheks-community entwickelt wurde. Der Funktionsumfang wurde vor allem im Hinblick auf eine Nutzung im LOK-Projekt untersucht.

[212] Getestet wurde mit MariaDB in Version 10.1.16.
[213] Für den Test wurde phpMyAdmin verwendet.
[214] Vgl. Datenbankschema unter https://github.com/bvb-kobv-allianz/marcel/wiki/Datenbankschema (15.05.2018).
[215] https://github.com/bvb-kobv-allianz/marcel (15.05.2018).
[216] Fehlermeldung: `java.lang.OutOfMemoryError: Java heap space`.

Software für den Umgang mit MARC-Daten

MarcEdit[217]

MarcEdit, ein Programm mit einem breit gefächerten Funktionsumfang, ist frei verfügbar, allerdings kein Open-Source-Produkt.[218] Es wurde im LOK-Projekt vor allem zur Formatkonvertierung von Dateien genutzt. Es werden zahlreiche Formate unterstützt: u. a. binäres und textbasiertes MARC-Format, MARCXML, JSON, Dublin Core. Ein Datenimport ist per Z39.50, SRU und OAI-PMH möglich. Beim Export von MARC-Daten in eine TSV-Datei können die gewünschten MARC-Felder und -Subfields spezifiziert werden.

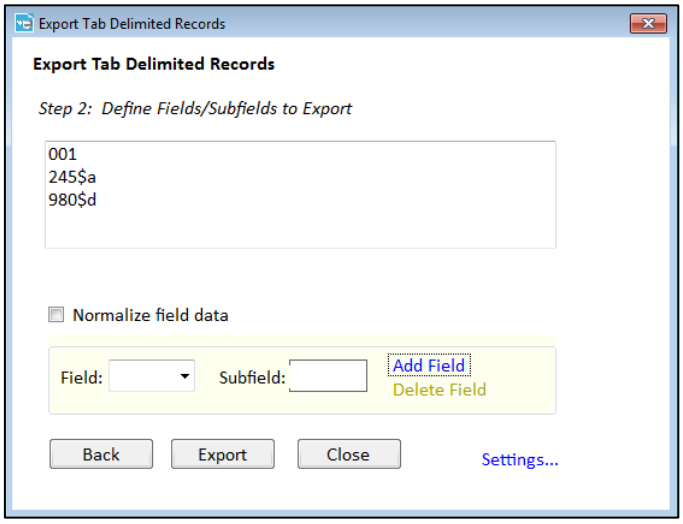


Abb. 2: MarcEdit: Auswahl von MARC-Feldern für den TSV-Export

Der integrierte Hex-Viewer kann genauso für Nicht-MARC-Formate genutzt werden, die Funktion zur Zeichenkonvertierung hingegen nur für MARC-Dateien. Zu den Statistikfunktionen gehört die „Field Count"-Funktion, die für jedes MARC-Feld bzw. -Subfield die Anzahl Vorkommen in der Datenmenge ausgibt und der „Material Type Report" mit einer Statistik über die vorhandene Materialarten sowie einer materialartspezifischen Filteroption. Verschiedene Validationen sind durchführbar, u. a. für die formale Datensatzstruktur oder für einzelne Felder (z. B. ISBN). Zusätzlich können die Feldgruppen 1XX, 6XX und 7XX mit externen Normdateien abgeglichen werden. Weitere Funktionen ergänzen IMD-Felder, DDC-Notationen oder Subject Headings der LoC in den Datensätzen. Das pauschale Ersetzen kann auf Feldbezeichnungen, Indikatoren und Subfields angewendet werden. In einer „Task List" können mehrere Transformationsvorgänge hinterlegt und wiederholt ausgeführt werden. Damit ist MarcEdit ein vielseitiges Tool für das Datenmanagement mit MARC-Daten. Die einzelnen Schritte eines ETL-Prozesses lassen sich gut abbilden.

[217] http://marcedit.reeset.net/ (12.05.2018).
[218] Vgl. http://marcedit.reeset.net/is-marcedit-open-source (15.05.2018).

C# MARC Editor[219]

Sehr komfortabel ist das Bearbeiten von MARC-Dateien mit dem Open-Souce-Tool C# MARC Editor. Das Programm beschränkt sich auf die Verarbeitung von Dateien in den generischen MARC-Formaten und in MARCXML, Daten können auch per Z39.50/SRU importiert werden. Für den Export kann aus verschiedenen MARC-Formaten gewählt werden. Die CSV-Export-Datei ist um Validations-Fehlermeldungen angereichert und kann so den Bearbeiter bei der (Fehler-)Analyse unterstützen. Beim Laden von MARC-Daten in die Anwendung werden die Daten in einer lokalen SQLite-Datenbank gespeichert, daher ist eine Bearbeitung zeitversetzt und schrittweise durchführbar.

Die übersichtliche Programmoberfläche ist in mehrere Bereiche aufgeteilt: Für einen gewählten Datensatz ist zusätzlich zur Anzeige im textbasierten MARC-Format die Liste der Felder mit Indikatoren sichtbar. Bei Auswahl des gewünschten MARC-Feldes wird dieser mit allen Subfields zur Bearbeitung präsentiert.

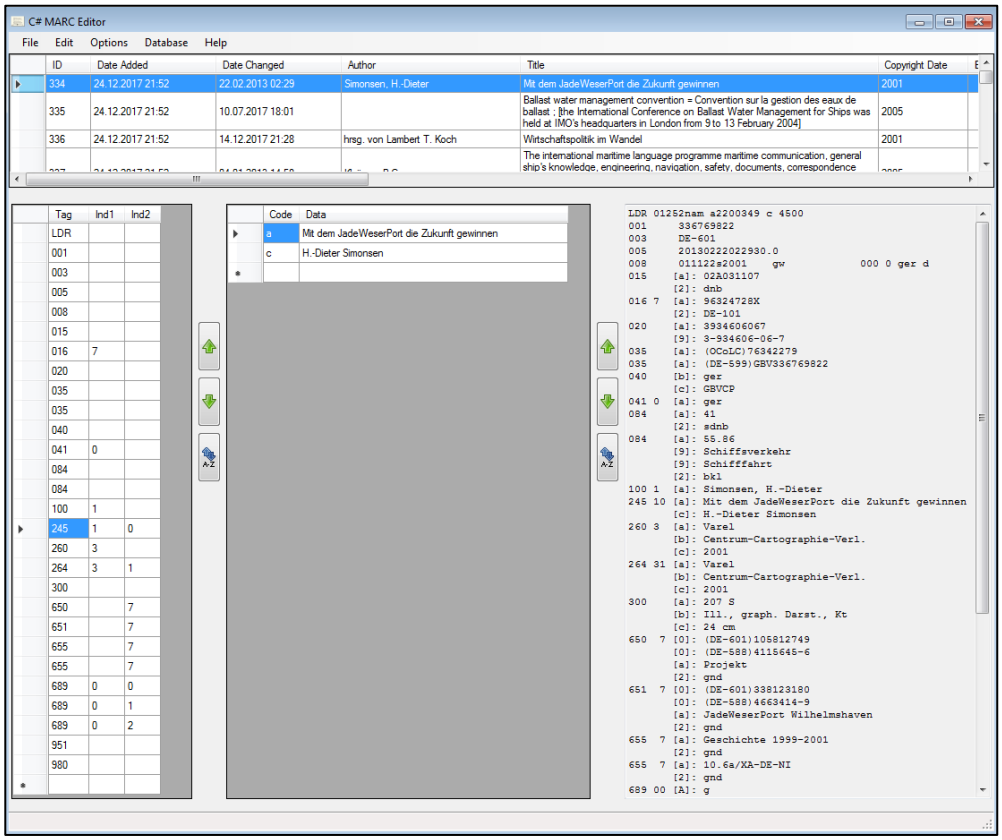


Abb. 3: C# MARC Editor: Bearbeitungsfenster

Die Suchfunktion berücksichtigt reguläre Ausdrücke; es lassen sich sogar Datensätze finden, die ein bestimmtes Feld/Subfield nicht enthalten. Unter den Statistikfunktionen ist eine Auswertung der Copyright-Jahresangabe aus MARC-Feld 008, visualisiert als Tortendiagramm. Hiermit lassen sich intuitiv Datenfehler in Form von Ausreißern erkennen. Besonders nützlich ist die Funktion „Advanced Batch Edit“, mit der in Abhängigkeit vom Inhalt eines Feldes bzw. Subfields MARC-Felder ergänzt, gelöscht oder geändert werden können.

Mit dem C# MARC Editor lassen sich auch große Dateien auf komfortable Weise bearbeiten.[220] Im LOK-Projekt wurden damit die MARC-Felder von SRU-Ausgaben im Format MARCXML kontrolliert.

[219] https://csharpmarc.net/ (15.05.2018).

MARCView[221]

MARCView ist ein einfacher Open-Source-Dateibetrachter für MARC-Daten, das Bearbeiten von Dateien ist nicht möglich. Die als Standard eingestellte Grenze von 100.000 Datensätzen für das Laden in den Editor ist in den Optionen änderbar, der vorbelegte Grenzwert für die Anzeige der Titelliste (linker Fensterbereich in Abb. 4) kann allerdings nicht beliebig erhöht werden.[222]

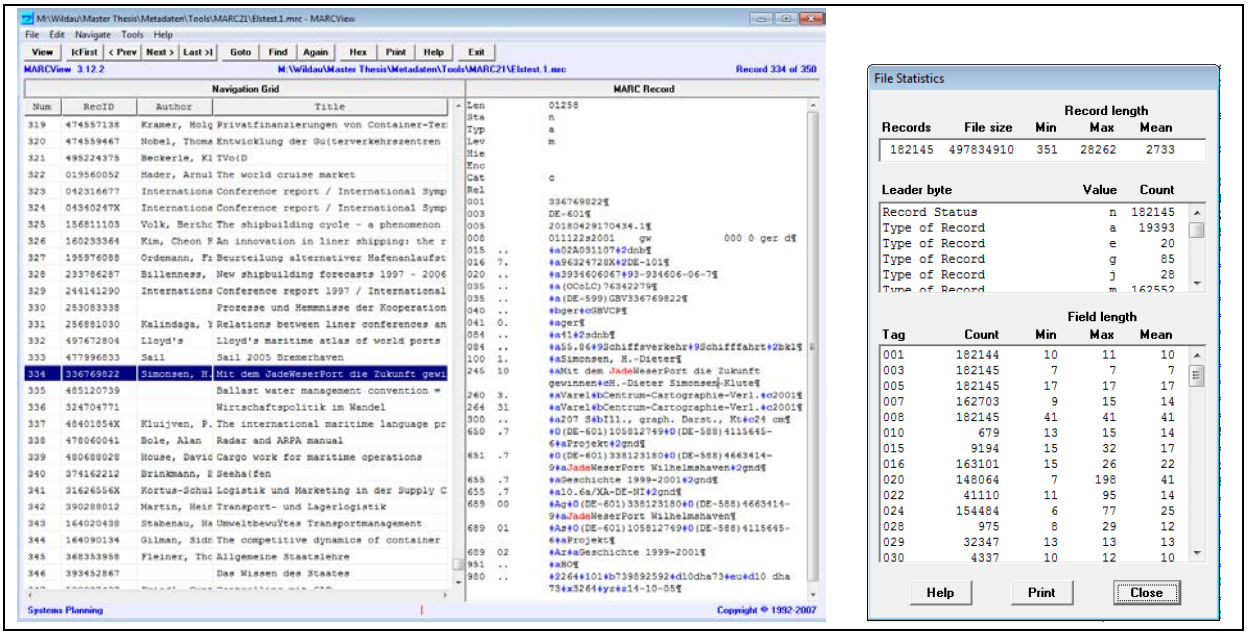


Abb. 4: MarcView: Datenpräsentation und Dateistatistik

Trotz der genannten Einschränkungen ist das Programm nützlich: Ein Datensatz kann in der Hex-Codierung angezeigt und zusätzlich ausgedruckt werden. Ein besonderes Feature ist die Dateistatistik, die für alle Datensätze den Inhalt der Leader-Elemente auswertet sowie für jedes in der Datenmenge vorhandene MARC-Feld die Anzahl sowie die Feldlänge ausgibt.

Software zur Datenanalyse und zum Abbilden eines ETL-Prozesses

Metafacture[223]

Im Rahmen des Culturegraph-Projektes[224] zur Vernetzung von Datenbeständen wurde das Open-Source-Toolkit Metafacture zur Verarbeitung von (vorwiegend Bibliotheks-)Metadaten entwickelt. Metafacture kann über die Kommandozeile gestartet oder als Java-Bibliothek in anderen Anwendungen genutzt werden, es enthält zahlreiche Module zum Lesen, Transformieren und Schreiben von Daten. Das Lesen einer Eingabedatei erfolgt streambasiert, das ist insbesondere bei großen Dateien vorteilhaft. Mit Hilfe der XML-basierten anwendungsspezifischen Sprache *Metamorph* werden Transformationsanweisungen erstellt. Unterstützung beim ETL-Prozess bietet die Scriptsprache *Flux*, damit lassen sich Module zu Pipelines verketten und so komplexe Abläufe bei der Metadatenverarbeitung abbilden. Metafacture ist erweiterbar um weitere Module; u. a. sind zzt. Plugins zur Anbindung an unterschiedliche Datenbanktypen und eine Erweiterung für *Eclipse*

[220] Getestet mit gut 182.000 Datensätzen.
[221] https://github.com/OCLC-Developer-Network/MARCView-Convert (15.05.2018).
[222] Standardeinstellung: 5.000 Datensätze.
[223] https://github.com/metafacture/metafacture-core (15.05.2018).
[224] http://www.culturegraph.org/ (15.05.2018).

vorhanden.[225] Viele Anwender nutzen Metafacture zur Konversion ihrer Katalogdaten nach RDF, u. a. die Deutsche Nationalbibliothek.[226]

Metafacture kann beliebige (semi)strukturierte Datenformate verarbeiten. Das *Metamorph*-Datenmodell bildet die logische Struktur der Ausgangsdaten ab, dazu werden Schlüssel-Wert-Paare (hier „Entitäten“ und „Literale“ genannt) gebildet. Die Schlüsselbezeichnungen werden dabei aus den Benennungen der Datenelemente generiert.

Datensätze im PICA+-Format lassen sich einfach in das Metamorph-Datenmodell überführen, wie das folgende Beispiel zeigt:

```
003@      $0785495819
021A      $aWeniger schlecht programmieren
028A      $9512977364$aPassig$dKathrin
```

Entität	Literal: Schlüssel	Literal: Wert
003@	0	785495819
021A	a	Weniger schlecht programmieren
028C	9	512977364
	a	Passig
	d	Kathrin

Tab. 3: PICA+-Datensatz im Metamorph-Datenmodell mit Entitäten und Literalen

Eine Transformation des Kategorie-/Subfield-Bezeichners für die Titelkategorie würde entsprechend so durchgeführt:

```
<?xml version="1.0" encoding="UTF-8"?>

<metamorph xmlns="http://www.culturegraph.org/metamorph"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   version="1" entityMarker="."> <- Separator for entities and literal names

      <rules>
         <data source="021A.a" name="Title" />
      </rules>

</metamorph>

Name of the literal to listen for
Name of the literal that is output
```

Abb. 5: Metafacture: Beispiel für Metamorph-Transformation[227]

Die Transformationsregeln wirken auf die Daten im internen Datenmodell und sind somit unabhängig vom Import- bzw. Exportformat. Damit können MARC-21- und MARCXML-Dateien mit denselben Transformationsregeln verarbeitet werden.

Metafacture ist für komplexe Datentransformationen im ETL-Prozess geeignet und wird vor allem im Linked-Data-Kontext eingesetzt. Auch für vergleichsweise einfache Transformationen ist ein hoher Einarbeitungsaufwand nötig. Aus diesen Gründen wurde auf eine Nutzung im LOK-Projekt verzichtet.

[225] https://github.com/metafacture/metafacture-core/wiki/Plugins-and-Tools (15.05.2018).
[226] Vgl. http://www.dnb.de/DE/Service/DigitaleDienste/LinkedData/linkeddata_node.html (15.05.2018).
[227] Böhme 2013, Folie 13.

d:swarm[228]

Die Plattform d:swarm ist ein Open-Source-Projekt unter Federführung der SLUB Dresden zur webbasierten Modellierung von ETL-Prozessen für das Management bibliothekarischer Daten. Es ist eine Serverinstallation erforderlich, da d:swarm als Middleware zwischen vorhandenen Systemen (z. B. Bibliothekssystem und Discovery-System) konzipiert ist. Aktiv genutzt wird das Tool zur Normalisierung von Metadaten aus heterogenen Datenquellen mit anschließendem Import in einen Solr-Index sowie zur Konvertierung von Bibliotheksdaten für die Bereitstellung von Linked Open Data.[229] In einer öffentlichen Demo-Instanz[230] kann die Funktionsweise von d:swarm ausprobiert werden.

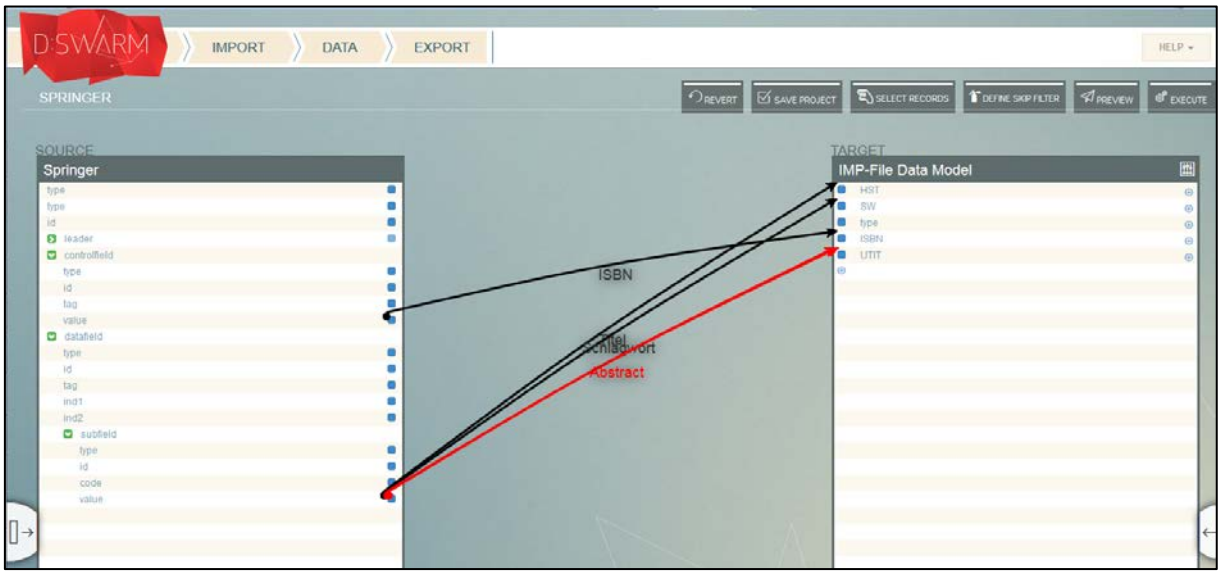


Abb. 6: d:swarm: Erstellen des Mappings[231]

Über die grafische Oberfläche werden die Schritte des ETL-Prozesses abgebildet. Es können Daten unterschiedlicher Formate (CSV, JSON, verschiedene XML-Formate) importiert werden, seit einiger Zeit ebenfalls Daten im Format PICA XML. Das Erstellen eines Mapping zur Datentransformation ist grundsätzlich intuitiv per Drag & Drop durchführbar und wird mit zahlreichen vorkonfigurierten Funktionen[232] unterstützt, dennoch ist für die Nutzung die Dokumentation[233] zu Rate zu ziehen. Die Daten werden in einer Graphdatenbank gespeichert. Für den Datenexport stehen JSON, XML und Linked-Data-Formate wie Turtle oder RDF XML zur Verfügung. Die zu einem Regelset zusammengefassten Transformationsanweisungen sind in der Anwendung gespeichert und können nachgenutzt werden.

Die Einstiegshürde bei d:swarm ist gegenüber Metafacture um einiges höher, allerdings ist die Schritt-für-Schritt-Anleitung in der Dokumentation didaktisch gut aufbereitet. Da die Anwendung eher für komplexe Einsatzszenarien mit Schwerpunkt Datentransformation und einer XML- bzw. oder RDF-Ausgabe gedacht ist, wurde eine Nutzung im LOK-Projekt verworfen.

[228] http://www.dswarm.org/de/ (15.05.2018). Projekteigene Schreibweise uneinheitlich, auch: D:SWARM.
[229] Vgl. http://www.dswarm.org/de/los-gehts/anwendungsfaelle/ (15.05.2018).
[230] http://demo.dswarm.org/#/data/ (15.05.2018).
[231] Beispiel „Springer“ aus Demo-Instanz (kein Direktlink verfügbar): ebd.
[232] Es wird dabei auf Metamorph-Funktionen von Metafacture zurückgegriffen.
[233] https://github.com/dswarm/dswarm-documentation/wiki (15.05.2018).

3.4 Integration heterogener Metadaten

Integrierte Informationssysteme sind aus der Bibliothekswelt nicht mehr wegzudenken. Gemeinsames Merkmal ist die Zusammenführung von Informationen aus unterschiedlichen Datenquellen in ein Zielsystem mit einheitlicher Datenstruktur. Das ermöglicht den Anwendern eine einheitliche Sicht auf die Datenbestände.[234] Im Folgenden werden Anwendungsfälle für die Metadatenintegration vorgestellt.

Gemeinsamer Verbünde-Index (GVI)[235]

Ein Beispiel für die Integration heterogener bibliografischer Metadaten ist der Gemeinsame Verbünde-Index (GVI) als gemeinschaftliches Projekt der AG Verbundsysteme. Der GVI enthält die Daten aller deutschen Bibliotheksverbünde sowie die Daten von DNB und ZDB. Die Daten werden (mehrheitlich) täglich aktualisiert und in einer Solr-Cloud vorgehalten. Die Daten in den Quellsystemen sind in unterschiedlichen Datenformaten vorgehalten (PICA, MARC, MAB/ASEQ). Um die Daten im GVI in einem einheitlichen MARC-21-Format zusammenzuführen, ist ein Mapping erforderlich. Der GVI wird in erster Linie von drei Bibliotheksverbünden zur Vereinfachung der Fernleihverfahren genutzt.

Deutsche Digitale Bibliothek (DDB)[236]

Die DDB versteht sich als nationales Zugangsportal mit dem Ziel der Integration von Metadaten aus allen deutschen Kultur- und Wissenschaftseinrichtungen. Für die Europäische Digitale Bibliothek (EUROPEANA) betätigt sich die DDB als nationaler Datenaggregator. Aufgrund der Heterogenität der beteiligten Institutionen ist neben MARCXML eine Reihe weiterer XML-basierter Formate für Datenlieferungen zugelassen. Andere Datenformate werden ggf. von der DDB-Servicestelle konvertiert. Der Datenworkflow[237] beginnt mit einer Datenanalyse, auf deren Grundlage dann die spezifischen Mappings erstellt werden. Die Datentransformation wird zunächst anhand von Testdaten in einem Testsystem erprobt; dabei durchlaufen die Schritte Ergebniskontrolle und Überarbeitung des Mapping einen iterativen Prozess. Verläuft die Ergebniskontrolle zufriedenstellend, beginnt der ETL-Prozess für die Gesamtmenge der vom Datengeber zur Verfügung gestellten Daten. Die Institutionen liefern ihre Daten per FTP; insbesondere bei regelmäßig aktualisierten Datenbeständen ist die Bereitstellung per OAI-PMH zu bevorzugen. Als Werkzeug für Datenimport und -transformation wird der vom Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme entwickelte *Augmented SIP-Creator* verwendet, der sowohl Metadaten als auch Binärdaten digitaler Objekte verarbeiten kann.[238]

[234] Vgl. Leser/Naumann 2007, S. 4.
[235] Auch: Gemeinsamer Fernleihindex.
[236] https://www.deutsche-digitale-bibliothek.de/ (15.05.2018).
[237] Vgl. Hartmann/Schulze 2013, Folie 51.
[238] Vgl. https://pro.deutsche-digitale-bibliothek.de/faq (15.05.2018).

Global Open Knowledgebase (GOKb)

Für den Einsatz eines ERM-Systems ist die zentrale Verzeichnung in einer sog. Knowledgebase erforderlich. Dabei werden neben rudimentären bibliografischen Daten Identifier (ISSN), Plattform, Host, verfügbare Jahrgänge und URL erfasst.[239] Zu den eigenständigen Open-Source-Lösungen mit kollaborativer Beteiligung zählen die von der JISC betriebene KB+[240] und die GOKb.

Ursprünglich war GOKb als globale Austauschplattform für Metadaten elektronischer Ressourcen eine Gemeinschaftsproduktion der OLE[241]-Partner und der JISC mit finanzieller Unterstützung der Andrew W. Mellon Foundation. Verantwortlich für die technische Entwicklung war die Firma Knowledge Integration Ltd (K-Int). Die Anwendung wurde prototypisch seit 2014 als Knowledgebase in OLE eingebunden. Organisatorisch ist die GOKb als Projekt in der Open Library Foundation (OLF) angesiedelt, die Projektleitung hat im September 2017 die ZDB übernommen.[242] Im LAS:eR-Projekt ist die Anbindung der GOKb an das ERM-System bereits realisiert. Ebenfalls berücksichtigen die ERM-Aktivitäten im FOLIO-Projekt[243] die Einbeziehung der GOKb als zentraler Knowledebase, auch hier ist K-Int an der Entwicklung beteiligt.

Als Referenzdatenbank für Metadaten von elektronischen Ressourcen verwaltet GOKb die Metadaten zu Paketen, Titeln und Anbietern.[244] Die Datenpflege erfolgt kooperativ, dabei übernehmen Bibliotheken eine „Patenschaft“ für einzelne Pakete. Bis jetzt sind nur Daten von elektronischen Zeitschriften enthalten, angedacht ist eine Erweiterung um E-Book-Pakete. Die Daten sind in einem speziellen GOKb-Format gespeichert und stehen unter CC0-Lizenz zur freien Nachnutzung zur Verfügung. Im Zuge des LAS:eR-Projektes werden zunächst Paket- und Titeldaten von National- und Allianzlizenzen in die GOKb-Datenbank eingebracht.

Zur Anzeige und Bearbeitung einzelner Datensätze wird die GOKb-Weboberfläche verwendet. Für den direkten Import von Titellisten oder den Datenexport stehen verschiedene APIs zur Verfügung.[245]

Die Normalisierung der Daten und der anschließende GOKb-Import erfolgen über OpenRefine[246] mit einer speziellen Erweiterung zur Anbindung an GOKb (im Folgenden kurz „GOKb-Erweiterung“).[247] Zunächst wird eine vom Verlag bzw. Aggregator erhaltene Titelliste (meist im KBART-Format) manuell in OpenRefine importiert, dabei wird ein Projekt angelegt. Die GOKb-Erweiterung führt automatisch eine Validation durch und gibt ggf. entsprechende Fehlermeldungen aus.

Das Zielformat erwartet standardisierte Spaltenbezeichnungen, daher müssen ggf. die Spaltennamen entsprechend geändert werden. Für den GOKb-Import sind weitere Spalten manuell zu ergänzen, und zwar mindestens für Plattform/Host, Paketname und Verlag. Dabei kann über die GOKb-Erweiterung direkt auf die Datenbank-IDs dieser Elemente zugegriffen werden, falls diese bereits in GOKb vorhanden sind.

---

[239] Vgl. den KBART-Standard der NISO: https://www.uksg.org/kbart/s5/guidelines/data_format (15.05.2018).
[240] https://www.kbplus.ac.uk/kbplus/ (15.05.2018).
[241] Open-Source-Bibliothekssystem, ursprünglich unter dem Dach der Kuali Foundation, vgl. Kemner-Heek 2016, S. 16.
[242] Seit 1.9.2017, vgl. Horn/Kemner-Heek 2017, Folie 4.
[243] https://www.folio.org/ (15.05.2018).
[244] Zur Speicherung bibliotheksspezifischer Lizenz- und Holdinginformationen ist ein ERM-System geeignet.
[245] Vgl. https://github.com/k-int/gokb-phase1/wiki/API (15.05.2018).
[246] Siehe Kap. 3.3.2.
[247] Geplant ist die Integration der Titellisten-Bearbeitung in die GOKb-Anwendung, vgl. Horn/Kemner-Heek 2017, Folie 13.

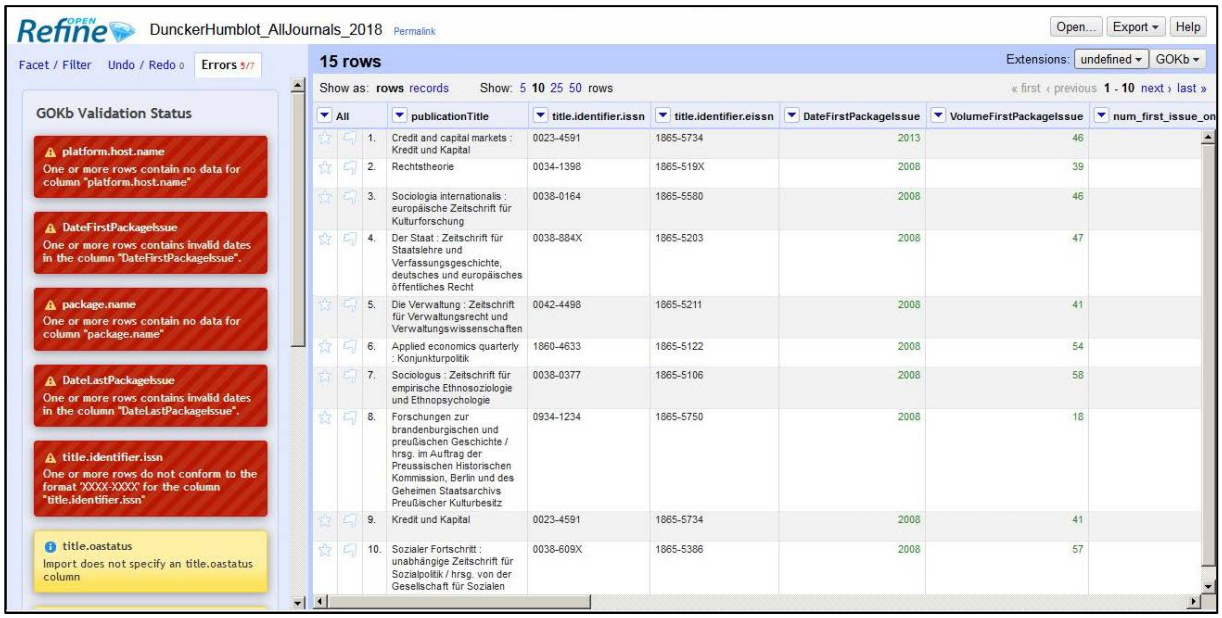


Abb. 7: GOKb-Erweiterung in OpenRefine mit Fehlermeldungen

Für einige Spalten müssen die Werte an die GOKb-Syntax angepasst werden, das betrifft u. a. Datumsfelder. Die Validationsroutine prüft ebenfalls die Eindeutigkeit von ISSN und e-ISSN, da neue Datensätze auf Grundlage der ISSN als eindeutigem Identifier beim Import eingemischt werden. Dieser Schritt ist besonders aufwendig, da durchaus Datensätze mit derselben ISSN im Paket enthalten sein können und zunächst geprüft werden muss, bei welchem Datensatz die ISSN erhalten bleiben kann.[248] Zur Speicherung kundenspezifischer Angaben ist eine Spalte für lokale Identifier vorgesehen.[249] Zur Unterstützung bei der Bearbeitung sind in der OpenRefine-Erweiterung zahlreiche GOKb-spezifische Makros hinterlegt, die zur Bearbeitung der Titellisten verwendet werden können. Ebenfalls lassen sich paketspezifische Transformationsregeln speichern und für die periodische Aktualisierung einer Titelliste nachnutzen. Dies ist vor allem für Titellisten in einem vom KBART-Format abweichenden Datenformat sinnvoll, da hier der Aufwand der Anpassungen per OpenRefine erheblich sein kann.

Sind alle Meldungen zu formalen Validationsfehlern behoben, kann das OpenRefine-Projekt per Upload-Funktion zur GOKb übertragen werden. Beim Import in die GOKb-Datenbank werden ggf. inhaltliche Unstimmigkeiten festgestellt[250], die in GOKb als sog. *Review Tasks* auf dem Dashboard präsentiert werden. Jeder Review Task muss einzeln geprüft und bearbeitet werden. Sind alle Tasks abgearbeitet, kann die Paketbearbeitung abgeschlossen werden.

Besonders vorteilhaft hinsichtlich der Bearbeitung über OpenRefine erscheint die Möglichkeit, die Datenanpassung einer Titelliste unterbrechen und zu einem späteren Zeitpunkt wieder aufnehmen zu können, da die Speicherung von OpenRefine-Projekten lokal erfolgt und erst nach Abschluss der Bearbeitung die Titelliste in die GOKb hochgeladen wird.[251]

[248] Dies ist z. B. bei separaten Datensätzen für Vorgänger und Nachfolger von Zeitschriften der Fall.
[249] Hierzu gehören IDs des Anbieters, z. B. eine ID des Springer-Verlags.
[250] U. a. wenn derselbe Zeitschriftentitel schon in einem anderen Paket enthalten ist.
[251] Vgl. https://openlibraryenvironment.atlassian.net/wiki/spaces/GOKB/pages/656219/Tutorial+GOKb+Data+Ingest+Using+OpenRefine (15.05.2018).

Titelverläufe von Zeitschriften (Titelsplits, Vorgänger/Nachfolger) können in der GOKb abgebildet werden. Problematisch ist dabei, dass die entsprechenden Informationen nicht immer identisch zu den in der ZDB vorhandenen Daten sind. Dies erschwert den Abgleich beim Datenfluss.

Discovery-System / FID-Portal

Zu den gängigen integrierten Informationssystemen im Bibliotheksbereich zählen mittlerweile auch Discovery-Systeme[252] als Ablösung der bisherigen OPACs. Wesentliche funktionale Eigenschaft ist die Aggregation weiterer bibliografischer Metadaten zusätzlich zu den Daten des lokalen Bibliotheks-systems. Der Suchindex bietet aufgrund der genutzten Suchmaschinentechnologie erweiterte Funktionen wie eine Relevanzsortierung oder das Filtern einer Treffermenge nach unterschiedlichen Kriterien. Vor allem Daten von elektronischen Ressourcen und Zeitschriftenaufsätzen, die von Verlagen, Aggregatoren und Open-Access-Repositorien bezogen werden, erweitern den Suchraum für die Bibliotheksnutzer erheblich. Der zentrale Suchindex K10plus-Zentral führt beispielsweise knapp 200 Mio. Datensätze aus unterschiedlichen Datenquellen zusammen: den Verbund-datenbanken von GBV und SWB, dazu Daten der Online Contents (OLC), aus JSTOR, DOAJ und zahlreichen weiteren Datenquellen.

Bei Discovery-Systemen kommerzieller Anbieter ist der eigene kostenpflichtige Suchindex integriert, der neben den Metadaten Zugriff auf freie und lizenzpflichtige Volltexte bietet. Daneben existieren Open-Source-Lösungen für Discovery-Systeme wie VuFind[253] oder Lukida[254].

Im Mittelpunkt der im Rahmen eines DFG-Förderprogramms etablierten „Fachinformationsdienste für die Wissenschaft“ (FID)[255] stehen ebenfalls Suchportale zur Bereitstellung wissenschaftlich relevanter Fachinformationen für Wissenschaftler des jeweiligen Fachgebiets.

Die Verfahren für das Einbringen von heterogenen Metadaten in ein FID-Portal sind vergleichbar mit den Abläufen bei ETL-Prozessen im Data Warehouse[256]:

- Abruf der Metadaten bei den Anbietern,
- Filterung auf fachlich relevante Inhalte,
- Konvertierung der in unterschiedlichen Ausgangsformaten vorliegenden Daten in das einheitliche Zielformat,
- Bereitstellung der Daten für den Suchmaschinenindex.

In diesem Zusammenhang erscheint der FID Politikwissenschaft POLLUX[257] erwähnenswert, da zur Kontrolle der ETL-Prozesse ein Tool „Nightwatch“ entwickelt wird, das die Abläufe für jede Datenquelle visualisieren und somit auch für (Projekt-)Mitarbeiter ohne Serverzugriff nachvollziehbar machen soll.

[252] Auch: Portale.
[253] https://vufind.org/vufind/ (15.05.2018).
[254] https://www.lukida.org/ (15.05.2018).
[255] http://www.dfg.de/foerderung/programme/infrastruktur/lis/lis_foerderangebote/fachinformationsdienste_wissenschaft/ (15.05.2018).
[256] Siehe Kap. 2.1.
[257] https://www.pollux-fid.de/ (15.05.2018).

Da Nightwatch noch nicht in Betrieb ist, werden zunächst die Abläufe in der aktuell betriebenen Form vorgestellt.[258]

Der Suchraum der zentralen Suchmaschine (Solr/Lucene) soll relevante elektronische Verlagspublikationen, wissenschaftliche Open-Access-Dokumente, Forschungsdaten sowie Inhalte aus Fachdatenbanken und Zeitungsarchiven umfassen, außerdem gedruckte fachspezifische Publikationen. Dieses breite Spektrum verdeutlicht die Herausforderungen bei der Implementierung der Prozesse.

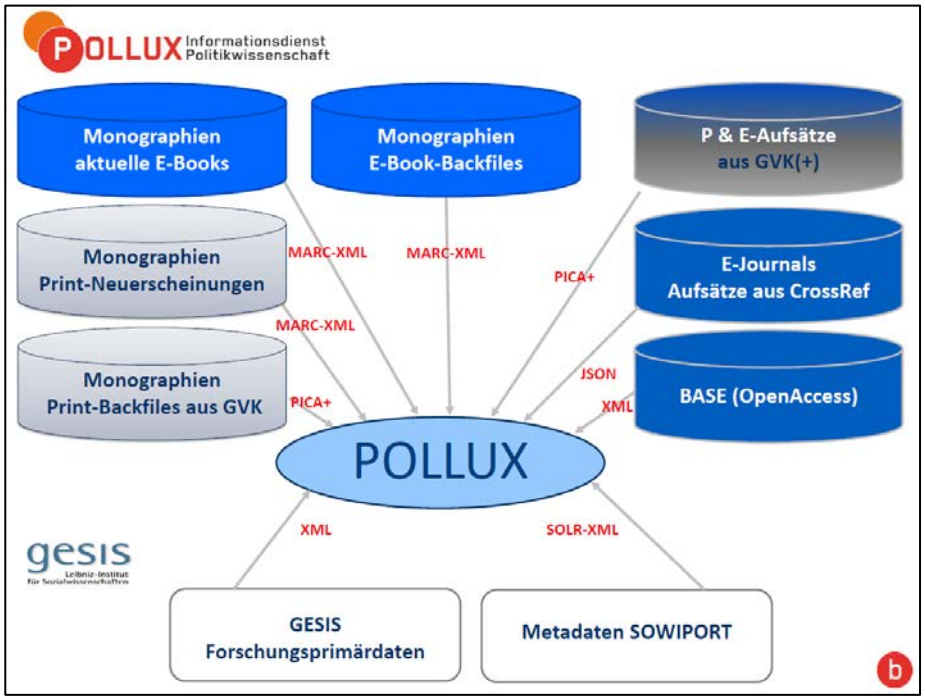


Abb. 8: Einbeziehung von Datenquellen in POLLUX[259]

Zu Beginn steht die Auswahl der relevanten Quellen. In Absprache mit der Fachcommunity wurden Verlage mit politikwissenschaftlichem Angebot identifiziert und fachliche E-Book-Pakete lizenziert. Diese Daten brauchen im weiteren Verarbeitungsprozess nicht weiter inhaltlich gefiltert zu werden. Die Metadaten von Print-Monografien aus der GBV-Verbunddatenbank (CBS) werden bereits gefiltert zur Verfügung gestellt, Auswahlkriterien sind geeignete Titelstichwörter, DDC-Notationen sowie die potenzielle subito-Verfügbarkeit. Eine regelmäßige Evaluation der gewählten Auswahlkriterien ist Bestandteil des Auswahlprozesses. Um Metadaten von Zeitschriftenaufsätzen fachlich zuordnen zu können, wurden vorab die im CBS enthaltenen ZDB-Zeitschriftentiteldaten auf Relevanz untersucht und anhand geeigneter Sacherschließungsmerkmale eine Liste passender Zeitschriften erstellt. Deren ISSNs werden für die Auswahl der Aufsatzdaten aus Crossref[260], OLC und dem ehemaligen Sowiport-Portal[261] verwendet. Zur Auswahl der BASE[262]-Daten werden eine Whitelist und eine Blacklist herangezogen.

[258] Vgl. Mönkediek 2018.
[259] Ursprüngliche Planung. Opitz/Haake 2017, Folie 4.
[260] https://www.crossref.org/ (15.05.2018).
[261] https://git.gesis.org/open-data/solis-sofis (15.05.2018).
[262] https://www.base-search.net/ (15.05.2018).

Je nach Datenquelle erfolgt die Extraktion der Daten auf unterschiedliche Weise. Zum Teil werden die Daten vom Anbieter per FTP zur Verfügung gestellt, oder es werden entsprechende Titellisten direkt von den Plattformen der Verlage bzw. Aggregatoren heruntergeladen. Der Zugriff auf Crossref-Daten erfolgt über eine REST-Schnittstelle.[263]

Zum Zwecke der Transformation wurde für jede Datenquelle ein eigenes Konvertierungsprogramm (mehrheitlich in Python unter Windows) erstellt. Mit Stand März 2018 werden Daten in den Formaten JSON, PICA+, XML, MARCXML und CSV verarbeitet. In den Konvertierungsprogrammen erfolgt eine Filterung der gewünschten Datenfelder aus der Quelldatei; die Affiliation eines Autors wird beispielsweise nicht übernommen. Zusätzlich findet eine Harmonisierung der Feldinhalte statt, so werden Datumsfelder normalisiert oder XML-Tags im Abstract-Feld entfernt.

Für den ETL-Schritt Laden werden die Dateien mit den konvertierten Daten schließlich an die Indexierungsroutine der Suchmaschine übergeben und stehen damit den Nutzern von POLLUX für eine übergreifende Suche zur Verfügung.

Bei der Zusammenführung von Metadaten im Suchindex ist die Behandlung von Dubletten eine Herausforderung; das gilt auch für Datenlöschungen, z. B. bei einer zurückgezogenen Lizenz.

Mangels spezifischer ETL-Tools für bibliografische Metadaten wurde Nightwatch konzipiert, mit dem Ziel der Visualisierung aller für POLLUX eingerichteten Prozesse mit linearer Datenflussarchitektur. Da die Beschaffung und Bereitstellung der Metadaten durch verschiedene Abteilungen erfolgt, soll das Tool als Workflowmanagement-Werkzeug für die Beteiligten aus IT, Erwerbung und POLLUX-Geschäftsstelle dienen. Die Überwachung der Datenflüsse von Datenquellen (Pipelines) soll realisiert werden, indem für jede Pipeline der Status aller Prozessschritte[264] erkennbar ist, ohne dass Zugriff auf Protokolldateien auf Serverebene notwendig wäre. Ebenfalls soll eine Steuerung der Prozesse über die webbasierte Oberfläche ermöglicht werden, indem ein (Re-)Start von Prozessen einer Pipeline angestoßen werden kann.[265] Die Bereitstellung als Open Source in GitHub ist geplant.

[263] Vgl. https://github.com/CrossRef/rest-api-doc (15.05.2018).
[264] Beispielsweise *abgebrochen*, *abgeschlossen* oder *fehlgeschlagen.*
[265] Vgl. Opitz/Haake 2017, Folien 14-16.

4 Migration: Grundlagen und Strategien

Ein Informationssystem *„dient […] der rechnergestützten Erfassung, Speicherung, Verarbeitung, Verwaltung, Pflege usw. von Information“*[266] und umfasst Hardware, Betriebssystem, Datenbanksystem, Systemschnittstellen und Anwendungssoftware mit Benutzerschnittstellen. Informationssysteme lassen sich einteilen in operative Systeme, Managementsysteme für Planung und Controlling sowie Querschnittssysteme[267]. Beispiele für operative Systeme einer Bibliothek sind das lokale Bibliotheksmanagementsystem oder Anwendungen zur Unterstützung des Digitalisierungsworkflows.

Als Legacy-System[268] wird ein Informationssystem bezeichnet, wenn es sich um *„ein altes, etabliertes System in der Firmensoftware* [handelt]*, welches immer noch im Gebrauch ist, aber in weiten Teilen nicht mehr den neuesten Entwicklungen entspricht.“*[269]

Legacy-Systeme sind häufig unflexibel und erlauben keine Anpassungen an geänderte Geschäftsanforderungen; das ist insbesondere problematisch, wenn es sich um geschäftskritische Anwendungen handelt.[270] So setzen derzeit viele Bibliotheken Legacy-Bibliotheksmanagementsysteme ein, die keine oder nur eingeschränkte Möglichkeiten zur Verwaltung elektronischer Ressourcen bieten.

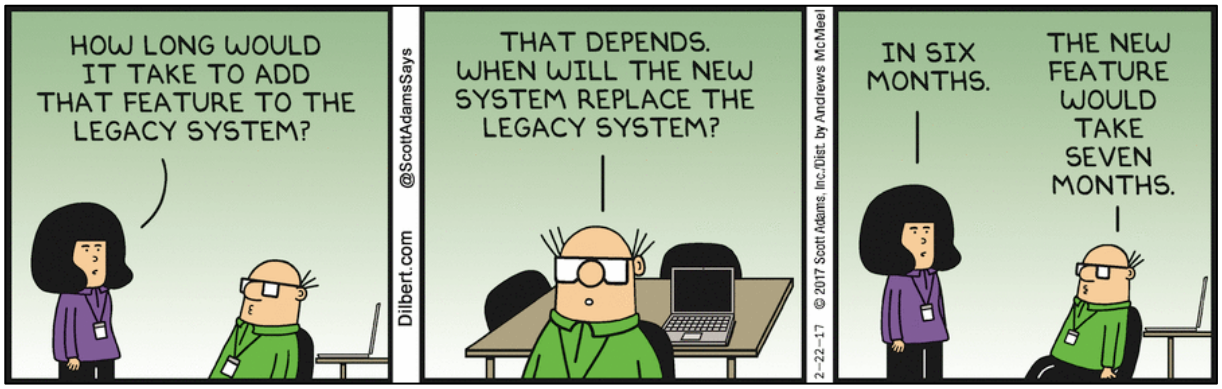


Abb. 9: Dilbert-Cartoon *How long for New Feature*[271]

Beim Umgang mit einem solchen Altsystem können zwei Strategien verfolgt werden. Soll die bisherige Technologie beibehalten werden, kann mit der klassischen Systemwartung (*Maintanance*) oder per *Refactoring* bzw. *Reengineering* eine Verbesserung des Altsystems erreicht werden, sofern sich Kosten und Risiko kontrollieren lassen. Ist der Wechsel auf ein Zielsystem unter Nutzung neuer Technologien geplant, so ist grundsätzlich der Umstieg („Migration“) auf eine Neuentwicklung oder geeignete Standardsoftware möglich.

In der Literatur wird der Begriff „Migration“ uneinheitlich verwendet. Abweichend zur gängigen Interpretation beschränke sich eine Softwaremigration auf *„die Überführung eines Softwaresystems in eine andere Zielumgebung oder in eine sonstige andere Form, wobei die fachliche Funktionalität unverändert bleibt.“*[272] Nur so seien vergleichende Funktionstests von Alt- und Neusystem möglich.

[266] Mandl 2009, S. 2.
[267] Darunter sind Systeme für Bürokommunikation oder kollaboratives Arbeiten und Dokumentmanagementsysteme zu verstehen.
[268] Auch: Altsystem.
[269] http://www.itwissen.info/Legacy-System-legacy-system.html (15.05.2018).
[270] Vgl. Brodie/Stonebraker 1995, S. 30; Masak 2006, S. 2.
[271] http://dilbert.com/strip/2017-02-22 (15.05.2018).
[272] Sneed et al. 2010, S. 25.

Im Gegensatz dazu sei das Ändern oder Hinzufügen einer fachlichen Funktion als Neuentwicklung aufzufassen.[273] Ebenso ist der Wechsel zu Standardsoftware nach dieser engen Definition nicht als Migration zu verstehen.

Andere Autoren kommen zu dem Schluss, dass nach einer Migration im Zielsystem die wesentlichen Funktionen und Daten aus dem Altsystem zur Verfügung stehen.[274] Dieser Einschätzung folgt die vorliegende Arbeit. Demnach bezeichnet Migration die technische Transformation von Informationssystemen in eine andere Zielumgebung. Es handelt sich dabei um eine *„Ablösung bestehender ‚Alt-Systeme' durch neue Systeme mit gleicher oder erweiterter Funktionalität"*[275] und ist durch eine *„wesentliche Veränderung der vorhandenen Systemlandschaft"*[276] gekennzeichnet.

Demnach kann anstelle einer Neuentwicklung genauso die Inbetriebnahme von Standardsoftware das Ziel einer Migration sein. Dies gilt insbesondere für die heutzutage bei Bibliotheksmanagementsystemen übliche Migration zu einem auf dem Markt befindlichen Produkt.[277] In der Praxis wird eine Migration zu Standardsoftware immer auch funktionale Änderungen nach sich ziehen, da es keine 100%ige Abdeckung geben wird, während man sich bei einer Eigenentwicklung auf die Implementierung bestehender Funktionen auf zeitgemäßer technologischer Basis beschränken kann.

In einem Migrationsprojekt ist die jeweilige Migrationsart zu berücksichtigen. Man unterscheidet generell Hardwaremigrationen, die ggf. auch ohne eine begleitende Systemmigration durchgeführt werden, und Anwendungsmigrationen, die in der Regel außer der Migration der Software ebenfalls eine Datenmigration umfassen. Eine vollständige Systemmigration umfasst die Migration aller Komponenten des Altsystems; hierzu gehören Benutzerschnittstelle, Systemschnittstellen, Anwendungen und Datenbank. Eine Migration des Betriebssystems kann von Anpassungen an weiteren Systemkomponenten begleitet sein. Entsprechend den Bestandteilen des Legacy-Systems müssen ggf. für die Komponenten der Softwarearchitektur (Benutzeroberfläche, Schnittstellen, Programme und Daten) separate Migrationspfade entwickelt werden.

Für eine Datenmigration ist eine spezifische Herangehensweise erforderlich; hier kommen die in Kap. 2.1 dargestellten ETL-Prozesse zur Anwendung. Im Allgemeinen sind vor der meist erforderlichen Konvertierung der Daten zusätzlich Aktivitäten zum Datenqualitätsmanagement einzuplanen (siehe Kap. 3.2).

Die Motive für die Migration eines Informationssystems sind vielfältig. Kap. 4.1 stellt systemimmanente Beweggründe und strategische Argumente vor. Für die Beteiligten ist die Durchführung einer Migration eine große Herausforderung, die Konzeption einer Projektstruktur kann dabei unterstützen (siehe Kap. 4.2). In einer frühen Projektphase verständigen sich die Beteiligten auf eine geeignete Migrationsstrategie, dabei wird generell zwischen einer inkrementellen und einer Stichtagsumstellung unterschieden (siehe Kap. 4.3).

[273] Vgl. Sneed et al. 2010, S. 26.
[274] Vgl. Brodie/Stonebraker 1995, S. 7.
[275] Dippold et al. 2005, S. 117.
[276] Die Beauftragte der Bundesregierung für Informationstechnik 2012, S. 6.
[277] Zur Migration lokaler Bibliotheksmanagementsysteme siehe Kap. 4.4.

4.1 Gründe für eine Migration

Die Entscheidung für die Migration eines Legacy-Systems lässt sich in den meisten Fällen auf die folgenden Gründe[278] zurückführen.

Erweiterung der Funktionalität

Ziel ist ein verbesserter Nutzen für die Anwender, auch durch Fehlerbehebungen.

- Integration aller Geschäftsgänge und Abteilungen im Bibliotheksmanagementsystem (BMS)
  Beispiel: Migration von der Eigenentwicklung BABSY zu SunRise (OCLC) in der UB Bochum
- Integration der Verwaltung elektronischer Ressourcen im BMS
  Beispiel: Migration der UB Mannheim zu Alma (Ex Libris)

Mangelnde Unterstützung des Altsystems

- Keine sicherheitsrelevanten Updates, auch wegen veralteter Technologie
  Beispiel: Ablösung der Eigenentwicklung für den Versand von Benachrichtigungen aus dem Ausleihmodul, bisher auf einer Workstation mit Windows 2000 betrieben
- Kein geschultes Personal mehr
  Beispiel: Ablösung von URICA (McDonnell Douglas) in der UB Oldenburg
- Fehlende Wartung durch Auslaufen des Supports
  Beispiel: Migration von LBS3 zu LBS4 in den Bibliotheken des GBV
  Beispiel: Transfer von bisher lokal erfassten Katalogisaten in die GBV-Verbunddatenbank wegen fehlender Unterstützung des LBS4-Katalogisierungsmodul CAT4 in zukünftigen LBS4-Versionen[279]

Strategische Gründe

- Hohe Betriebskosten durch Lizenzgebühren und Personalkosten
  Beispiel: Migration der Commerzbibliothek Hamburg von einer lokalen Alephino-Instanz (Ex Libris) zum LBS-Service der VZG
- Ablösung von unübersichtlichen und umfangreichen Daten- und Dokumentensammlungen auf verschiedenen Plattformen zum Zwecke der Verwaltung von Lizenzen elektronischer Ressourcen
  Beispiel: Einführung des ERM-Systems RMS (Semper Tools) in der UB Kassel
- Teilnahme am Bibliotheksverbund mit kooperativer Katalogisierung und weltweiter Sichtbarkeit der Bestände
  Beispiel: Migration der Bibliothek des Wissenschaftskollegs zu Berlin von BIBLIOTHECA zur GBV-Verbundlösung
- Wechsel zu Open-Source-Lösungen zur Vermeidung des Vendor Lock-in[280]
  Beispiel: Migration der Bibliotheken der Goethe-Institute zu Koha
- Cloud-basierte SaaS-Lösung statt lokaler Serverbetreuung
  Beispiel: Migration des Max-Planck-Instituts für Bildungsforschung (Berlin) von einer lokal betriebenen Allegro-Implementierung zum LBS-Service der VZG

[278] Vgl. Die Beauftragte der Bundesregierung für Informationstechnik 2012, S. 21; Wachter/Zaelke 2015, S. 2; Dippold et al. 2005, S. 116–117; Albrecht 2012, S. 72–74.
[279] Dies ist der Anlass für das Projekt zur Migration der lokalen Katalogisate, siehe Kap. 5.
[280] Abhängigkeit vom Anbieter, insbesondere wenn der Wechsel zu einem alternativen Anbieter unwirtschaftlich ist.

Die Nutzung moderner Technologien als Motivation für eine Softwaremigration, beispielsweise zu einer auf serviceorientierter Architektur (SOA) basierenden Anwendung, wird bei Sneed kritisch hinterfragt. Er verweist dabei auf den Einfluss der IT-Industrie, die Anwendern regelmäßig neue technologische Entwicklungen aufdränge, um damit die Entwicklungskosten zu finanzieren, auch wenn die Anwender keinen direkten Nutzen davon hätten.[281] Im Umkehrschluss ist bei der Nutzung von Informationssystemen eine strikte Verweigerung technischer Innovationen sicher nicht angemessen; in der Praxis sind vermutlich branchenspezifisch unterschiedliche Positionen anzutreffen. Es ist im Einzelfall abzuwägen, ob im Sinne der Zukunftssicherheit eine Migration sinnvoll ist. Marketing-Buzzwords wie „Cloud" entheben den Kunden nicht einer kritischen Bewertung des Zielsystems.

Für den spezielleren Fall einer Datenmigration, häufig Bestandteil eines umfangreichen Migrationsszenarios, ist in vielen Fällen ein Anbieter- bzw. Produktwechsel der operativen Anwendung das auslösende Ereignis. Bei der Zusammenführung und Vereinheitlichung von Daten aus bislang getrennter Datenhaltung wie aktuell im K10plus-Projekt sind umfangreiche Maßnahmen zur Bewältigung der Datenmigration erforderlich. Außerdem ist die Weiterentwicklung der technischen Plattform zumeist Anlass für eine Datenmigration, z. B. beim Wechsel von Hardware, Betriebssystem oder DBMS.

[281] Vgl. Sneed et al. 2010, S. 2–4.

4.2 Migrationsvorhaben als Projekt

Auf Grund ihrer Komplexität werden Migrationsvorhaben i. d. R. als Projekt realisiert. Hierbei handelt es sich um ein *„Vorhaben, das im Wesentlichen durch Einmaligkeit der Bedingungen in ihrer Gesamtheit gekennzeichnet ist“*.[282] Weitere Kriterien für die Einordnung als Projekt sind:

- Neuartigkeit,
- Zeitliche Befristung,
- Spezifische Zielvorgabe,
- Begrenzte Ressourcen (finanziell, personell, technisch),
- Abgrenzung gegenüber anderen Vorhaben,
- Risiko/Unsicherheit,
- Projektspezifische Organisation.

Im Gegensatz dazu ist von einem Prozess auszugehen, wenn es sich um einen wiederholbaren Ablauf von Tätigkeiten handelt, die zudem standardisiert sind. Daher ist eine implementierte Abfolge von Extract – Transform – Load als Prozess einzustufen, während das Erstellen von Transformationsregeln vor der Inbetriebnahme eines ETL-Prozesses wegen der o. g. Projekteigenschaften als Arbeitspaket eines Projekts eingeordnet werden kann.

Insbesondere die Migration bibliografischer Daten ist ein komplexes Vorhaben; das lässt sich formal anhand der folgenden Kriterien[283] belegen.

Kriterium für Komplexität	Charakteristikum im Bibliothekskontext
Vielzahl der zu migrierenden Elemente	Große Anzahl Datensätze
Verschiedenartigkeit der Elemente	Diversität bei Satzarten und Datenfeldern
Vieldeutigkeit	Eine Gesamtmenge bibliografischer Daten ist zumeist nicht homogen. Die Daten stammen z. B. aus mehreren Generationen von Regelwerken zur Formalerfassung bibliografischer Daten (PI, RAK, RDA).
	In Bezug auf die Einhaltung von Regeln und Richtlinien lassen sich keine Voraussagen treffen, daher können dieselben Sachverhalte in Datenelementen (Kategorien) unterschiedlich ausgedrückt werden; dies begünstigt das Anlegen von Dubletten.
Veränderlichkeit	Gerade bei bibliografischen Daten kann es auch bei einer sorgfältigen Datenanalyse zu Projektbeginn vorkommen, dass sich die Ausgangsdaten durch eine Regelwerksänderung, im Rahmen der Sachbearbeitung oder durch zentrale Datenumsetzungen bis zur endgültigen Datenmigration unvorhergesehen geändert haben, wenn die Datenbasis während der Projektlaufzeit nicht eingefroren wurde.
Menge und Vernetzung der Beziehungen zwischen Elementen	Insbesondere zutreffend im Hinblick auf die Erscheinungsformen MTM, monografische Reihe, Zeitschriftenaufsatz oder Vorgänger-Nachfolger-Beziehung bei Zeitschriften.

Tab. 4: Kriterien für Komplexität von Migrationsprojekten

Für den Erfolg eines Migrationsvorhabens ist insbesondere bei komplexen Datenmigrationen eine gute Projektplanung unabdingbar.

[282] DIN 69901-5:2009-01, S. 11.
[283] Vgl. Reiß 1993, S. 57–59.

Projektphasen bei Datenmigrationen

Für die Datenmigration innerhalb eines bibliothekarischen Migrationsprojektes ist die Anwendung des Standardvorgehensmodells der öffentlichen Verwaltung zur Abwicklung von IT-Projekten nur bedingt geeignet, da sich das sog. V-Modell XT[284] als deutsches Referenzmodell vorrangig für die (Neu-)Entwicklung von Softwaresystemen versteht. Die gemäß Modell anstehende Make-or-Buy-Entscheidung wird bei der Migration eines BMS heutzutage auch ohne explizite Wirtschaftlichkeitsanalyse immer zugunsten eines Standardproduktes ausfallen.[285] Ausführungen zu Datenmigrationen sind nicht vorhanden, es heißt lediglich:

> *„Die Datenmigration ist detailliert zu planen. Der Datenfluss von den Quelldatenbanken zu den Zieldatenbanken wird festgelegt. Zusätzlich werden alle notwendigen Datentransformationen definiert.“*[286]

Grundsätzlich auf den Bibliotheksbereich übertragbar erscheint dagegen der standardisierte Ansatz zur Bewältigung von Datenmigrationen von Lüssem/Harrach[287], da sich die beschriebene Projektstruktur mit Migrationsprojekten aus der bibliothekarischen Praxis deckt.

Für Datenmigrationen sind die Aktivitäten demnach in diese zwölf Phasen zu gliedern:

	Projektphase	**Inhalt**
A	Initiierung des Projektes	Einsetzen der Projektleitung mit Verantwortung für die Organisation und Durchführung der folgenden Projektphasen
B	Planungsphase	- Festlegung der Arbeitspakete gemäß der funktionalen Anforderungen - Festlegung einer Migrationsstrategie - Festlegung der nicht zu migrierenden Daten - Erstellen eines Archivierungskonzepts für diese Daten - Ermittlung von Abhängigkeiten in Quell- und Zielsystem
C	Datenanalyse	Zur Sicherstellung einer hohen Datenqualität im Zielsystem ist die Einbeziehung von Personal mit Fachwissen erforderlich. - Entwicklung eines Verfahren zum Export der Quelldaten und zur Unterstützung der Datenanalyse - Vorbereitung der technischen Infrastruktur - Entwicklung technischer Spezifikationen - Evaluation potenzieller Importverfahren - Festlegung des zukünftigen Datenformats
D	Migrationskonzept	- Einrichten einer Testumgebung - Auswahl geeigneter Migrationstools - Bestimmung kritischer Abhängigkeiten zwischen Altsystem und externen Anwendungen - Abschätzung eines Freeze-Zeitraums während der produktiven Migrationsphase
E	Konzept zur Datentransformation	- Erstellen eines Datenmappings von Quell- und Zielstruktur - Ermittlung von obligatorischen bzw. fakultativen Daten in der Zielstruktur - Erarbeiten von Transformationsregeln

[284] https://www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/vmodell_xt_node.html (15.05.2018).
[285] Vgl. Ausführungen in Kap. 4.4.1.
[286] Verein zur Weiterentwicklung des V-Modell XT e.V. (Weit e.V.) 2017, S. 160.
[287] Vgl. Lüssem/Harrach 2013. Der Ablauf wurde bei einem Migrationsprojekt von Daten einer SAP-Standardsoftware für Versicherungen angewandt.

	Projektphase	Inhalt
F	Programmanpassungen	- Evaluation der Anwendungsumgebung: Workflows, anwendungsspezifische Datenfelder, Anforderungen an die Protokollierung - Ggf. Nachnutzung existierender Tools zum Laden der Daten
G	Vorbereiten der Testumgebung	- Implementierung der Migrationstools - Export aus dem Legacy-System mit Sperrung für den schreibenden Zugriff - Erstellen eines Backup der Altsystem-Daten - Sperre der Anwendungen mit Zugriff auf das Altsystem
H	Datenimport in die Testumgebung	- Datentransfer in die Testumgebung - Prüfung auf Vollständigkeit und Konsistenz - Durchführen von Anwendungstests mit migrierten Daten
I	Abnahme der Testmigration	- Kontrolle des Migrationsergebnisses durch alle Beteiligten - Freigabe der produktiven Migration
J	Produktive Migration	- Import der Daten ins Zielsystem - Aktivierung eines Fallback-Szenarios
K	Datenbereinigung	- Anpassung der Daten an die neue funktionale Umgebung - Löschen alter bzw. nicht mehr benötigter Daten
L	Abschalten des Altsystems	- Archivierung nicht migrierter Daten - Ggf. Bereitstellung des Altsystem mit lesendem Zugriff

Tab. 5: Phasen der Datenmigration[288]

Abweichend zu diesen Überlegungen erscheint eine Datenbereinigung bereits im Zusammenhang mit Phase C (Datenanalyse) sinnvoll, da die Erkenntnisse aus der Datenanalyse ggf. in eine direkt anschließende Datenbereinigung überführt werden können. Hier ist, soweit vom Aufwand vertretbar, im Bibliothekskontext zusätzlich zur Durchführung von Korrekturen eine Dublettenbereinigung vorteilhaft, ebenfalls sollten fehlende Daten im Ausgangssystem ergänzt werden.

Im Zusammenhang mit der Festlegung des Migrationskonzepts (Phase D) müsste ein Fallback-Szenario erarbeitet werden.

Bei komplexen Datenmigrationen, die mehrere Testmigrationen mit Nachbesserungen der Transformationsregeln benötigen, ist eine Sperrung des Legacy-Systems (Phase G) im Zusammenhang mit einem ersten Datenexport(-Test) unangebracht. Die Phasen E und H (Erarbeitung der Transformationsvorgaben und Datenimport in die Testumgebung) werden in der Praxis in einem iterativen Prozess mehrfach durchlaufen, bis eine endgültige Abnahme der Testmigration (Phase I) erfolgen kann. Erst dann kann der endgültige Datenexport aus dem Legacy-System stattfinden, gefolgt von finaler Datenkonvertierung und Import in das Zielsystem.[289]

Trotz der genannten Kritikpunkte lässt sich diese Gliederung in Projektphasen anlässlich einer Datenmigration generell auf das Projekt zur Migration lokaler Katalogisate anwenden. Daher erfolgt in Kap. 5 eine Zuordnung der Schritte im LOK-Projekt zu den o. g. Arbeitspaketen.

Die Projektstruktur bei BMS-Migrationen wird zumeist vom Anbieter des Zielsystems vorgegeben, der den Ablauf möglichst effizient gestalten möchte. Entsprechende Migrationsvorhaben werden in Kap. 4.4 erörtert.

[288] Vgl. Lüssem/Harrach 2013.
[289] So wird in der VZG bei Datenmigrationen verfahren, ebenso z. B. bei Migrationen zum BMS Alma, vgl. Ex Libris 2017a.

4.3 Migrationsstrategien

Bei komplexen Vorhaben wie einer Migration bedarf es einer detaillierten Planung. Hierbei sind Modelle geeignet, die den Migrationsprozess in idealisierter Form beschreiben und als Grundlage zur Ausgestaltung konkreter Projektabläufe dienen.[290]

Bei der Wahl einer geeigneten Migrationsstrategie sind verschiedene Faktoren zu berücksichtigen:[291]

- Art und Menge der Migrationsobjekte,
- Zerlegbarkeit des Migrationsobjekt (modular vs. monolithisch),
- Stellenwert von Ausfallzeiten,
- Vorhandensein einer Kopie des Zielsystems für Test- und Schulungszwecke,
- Anzahl und Umfang der Daten im Altsystem,
- Verfügbares Personal,
- Risiko- und Kostenabschätzung,
- Umfang der Änderungen am Datenschema beim Systemwechsel.

Die gewählte Strategie ist somit immer abhängig von Ausgangslage, Rahmenbedingungen und Komplexität des Migrationsprojektes.

Generell wird bei prozessorientierten Umstellungsmodellen zwischen einer inkrementellen (*Chicken-Little-Strategie*, siehe Kap. 4.3.2) und einer Punktumstellung (*Big Bang Approach*, siehe Kap. 4.3.1) unterschieden. Bei einer schrittweisen Umstellung werden einzelne Komponenten separat migriert, während bei einer Punktumstellung die Übergabe[292] aller Migrationsobjekte zu einem bestimmten Termin erfolgt.

Eine weitere bekannte Strategie ist der phasenorientierte *Butterfly*-Ansatz, der speziell für die Migration unternehmenskritischer Systeme mit einem großen Datenbestand konzipiert wurde, bei denen die Datenmigration zu längeren Ausfallzeiten führen würde. Der hohe technische Aufwand aufgrund der Nutzung von temporären Datenspeichern lässt einen Einsatz nur in kommerziellen Unternehmen bei sehr großem Datenvolumen sinnvoll erscheinen.[293]

4.3.1 Big Bang

Als „Big Bang“ (auch: *Cold Turkey*) wird das Migrationsverfahren bezeichnet, bei dem die Ablösung des Altsystems als Gesamtheit mit dem Umstieg auf das neue System in vollem Funktionsumfang an einem vorher festgelegten Stichtag erfolgt. Die Umstellung geschieht für alle Systemkomponenten gleichzeitig.

Im Vorfeld sind umfangreiche Tests des neuen Systems erforderlich. Zum Umstiegszeitpunkt wird das Ausgangssystem deaktiviert und es erfolgt die Datenmigration mit den in Kap. 2.1 beschriebenen Schritten eines ETL-Prozesses. Nach der Überführung der Daten wird das neue System in Betrieb genommen.

[290] Vgl. Sneed et al. 2010, S. 49.
[291] Vgl. Sneed et al. 2010, S. 114–117.
[292] Auch: *Cutover.* - Schreibweise nach *Cambridge Business English Dictionary* https://dictionary.cambridge.org/de/worterbuch/englisch/cutover (15.05.2018).
[293] Vgl. Wu et al. 1997; Kaps 2017, S. 107–108.

Bewertung des Big-Bang-Ansatzes[294]

Vorteile

- Einfachere organisatorische Umsetzung;
- Kürzere Dauer der eigentlichen Umstiegsphase, insbesondere der kritischen Sperrzeit für Änderungen am Altsystem (*Freeze*) bis zur Inbetriebnahme des Zielsystems (*Go live*);
- Testmöglichkeit des Zielsystems in seiner Gesamtheit vor Inbetriebnahme, inkl. Abhängigkeiten einzelner Module;
- Kein Parallelbetrieb von Legacy- und Zielsystem, dadurch keine Entwicklung von temporären Schnittstellen erforderlich.

Nachteile

- Komplexes Projektmanagement bei einer großen Zahl an Migrationspaketen;
- Hohes Projektrisiko auf Grund erhöhter Komplexität, insbesondere bei größeren Projekten;
- Ggf. längere Ausfallzeit der Produktionsumgebung, abhängig von der Dauer der Datenmigration bzw. der Menge der zu migrierenden Daten;
- Geeignete Notfallpläne für das Scheitern der Umstellung erforderlich (*fallback policy*).

Bei kommerziellen geschäftskritischen Anwendungen wird auf Grund der genannten Nachteile von einer Stichtagsumstellung abgeraten.

Die folgenden Kriterien kennzeichnen die Eignung für eine Big-Bang-Migration:

- Projekte mit eher geringem Datenvolumen;
- Systemwechsel mit umfangreichen Änderungen am Datenschema;
- Migrationen mit Terminvorgabe, z. B. bei Außerbetriebstellung der bisherigen Hard- oder Software;
- Anwendungen, die nicht unternehmenskritisch sind bzw. für die eine entsprechende Ausfallzeit akzeptabel ist.

Daher werden Migrationen im Bibliothekskontext zumeist als Big Bang praktiziert.

4.3.2 Chicken Little

Die Chicken-Little-Strategie basiert auf einer inkrementellen Vorgehensweise. Dabei wird das Gesamtsystem in voneinander unabhängige Module bzw. Komponenten zerlegt, die jeweils in einem überschaubaren Zeitraum migriert werden.[295]

Eine der Herausforderungen in der Planungsphase bei der Umsetzung der Chicken-Little-Strategie besteht darin, für die Bestandteile des Informationssystems geeignete Migrationspakete zu bestimmen. Jedes Migrationspaket wird separat in die Zielumgebung überführt, der Ablauf ist in jeweils elf Schritte unterteilt. Schritte, die für ein Paket irrelevant sind, werden übersprungen. Beim Fehlschlag eines Schrittes braucht nur dieser wiederholt zu werden. Für eine größtmögliche Effizienz müssen die Migrationsschritte nicht sequenziell nacheinander ablaufen, sondern können teilweise parallel erfolgen. Ebenfalls kann der Umstieg von Komponenten simultan oder aufeinander folgend ablaufen, so wie es den lokalen Erfordernissen am besten gerecht wird.[296]

[294] Vgl. Dippold et al. 2005, S. 127–128; Sneed et al. 2010, S. 14–15; Klettke/Thalheim 2011, S. 401–402; Glöckle 2007, S. 17.
[295] Vgl. Sneed et al. 2010, S. 16.
[296] Vgl. Brodie/Stonebraker 1995, S. 36–37.

Auch bei einer Aufteilung in unabhängige Komponenten sind für die Migrationsphase ggf. vorhandene Wechselwirkungen zu bedenken, da die Komponenten des Zielsystems jeweils einzeln in Betrieb genommen werden. Hierfür werden sog. Gateways genutzt, die als funktionale Kopplung Anfragen und Daten über die Schnittstellen der Komponenten übersetzen und austauschen. Ziel ist die Gewährleistung der Interoperabilität, da während der Migrationsphase das alte und das neue System nebeneinander existieren. Die Gateways werden nur temporär für die Dauer des Parallelbetriebs benötigt.[297]

Bewertung der Chicken-Little-Strategie[298]

Vorteile

- Überschaubar, auch hinsichtlich des Personalaufwandes für jedes Paket bzw. jeden Schritt;
- Bessere Planung von Fallback-Szenarien;
- Kontrollierbares Risiko;
- Geringere Komplexität der einzelnen Migrationsschritte;
- Fehler sind schneller erkennbar.

Nachteile

- Höherer Planungs- und Koordinationsaufwand;
- Erhöhter Aufwand für die Implementierung von Gateways;
- Dadurch erhöhte Komplexität der Synchronisation von Daten und Prozessen;
- Lange operative Parallelphase mit entsprechender Betreuung beider Systeme, dadurch höherer Aufwand.

4.3.3 Migrationsstrategien im Bibliotheksumfeld

Bei der Wahl der geeigneten Migrationsstrategie ist die Analyse des Altsystems von großer Bedeutung. Ein inkrementelles Vorgehen ist geeignet für gut strukturierte Systeme, bei denen Anwendungslogik, Präsentation und Datenhaltung voneinander getrennt sind. Berücksichtigt werden müssen die jeweiligen Rahmenbedingungen und Anforderungen. Dabei spielen die Vielfalt der zu migrierenden Komponenten und die Datenmenge sowie die Akzeptanz einer Ausfallzeit eine entscheidende Rolle. Zusätzlich sind die personellen und technischen Anforderungen einer Migrationsstrategie zu bedenken.

Diese Kriterien sind auf bibliothekarische Migrationsvorhaben übertragbar; hier ist die Anzahl der beteiligten Stakeholder ebenfalls mitentscheidend bei der Festlegung des Migrationskonzeptes.

Bibliothekarische Migrationsvorhaben werden überwiegend als Big Bang praktiziert. Bei umfangreichen Migrationsprojekten, beispielsweise unter Beteiligung mehrerer Bibliotheken wie bei der Alma-Migration im Österreichischen Bibliothekenverbund (OBV)[299], ist jedoch die Aufteilung in Migrationspakete erforderlich. Ähnliches gilt für die Migration von LBS3 zu LBS4, die pro Bibliothek modulweise durchgeführt wird, sodass sich die inkrementelle Migration des Gesamtprojekts als eine Kette von Big-Bang-Migrationen für jeweils ein Migrationspaket darstellt.[300]

[297] Vgl. Brodie/Stonebraker 1993, S. 6.
[298] Vgl. Sneed et al. 2010, S. 16.
[299] Auf die Migration im OBV wird in Kap. 4.4.4 genauer eingegangen.
[300] Weitere Ausführungen hierzu siehe Kap. 4.4.3.

4.4 Migration lokaler Bibliotheksmanagementsysteme

Die deutschsprachige Bibliothekslandschaft befindet sich aktuell im Umbruch. Die Client-Server-Systeme aus den 90er Jahren des vorigen Jahrhunderts werden vermehrt durch cloudbasierte SaaS-Systeme abgelöst. Zahlreiche Vorträge beim 106. Deutscher Bibliothekartag 2017 belegen dies: sie thematisierten Migrationsvorhaben in unterschiedlichen Stadien.[301] Hier wurde deutlich, dass neben den Veränderungen bei bibliothekarischen Workflows insbesondere die Qualität der Datenmigration eine wesentliche Rolle für die Nutzbarkeit des neuen Systems spielt.

Der Ablauf einer Datenmigration beim Wechsel des Bibliothekssystems folgt den Schritten eines ETL-Prozesses:

- Export der Daten aus dem Altsystem;
- Datenkonvertierung, ggf. auch Zeichensatz-Konversion;
- Import der Daten in das Zielsystem.

Dem Datenexport vorgeschaltet ist i. d. R. eine Phase umfangreicher Datenbereinigungsaktivitäten. Nach der Migration sind ebenfalls Korrekturarbeiten einzuplanen.

Die Behandlung von nicht zu migrierenden Daten ist bei der Planung ebenso zu berücksichtigen. Dies könnten beispielsweise Benutzerdaten mit abgelaufener Nutzungsberechtigung sein oder lokale Katalogisate mit temporärer Gültigkeit. Ggf. ist auch die Migration von Erwerbungsdaten technisch nicht möglich oder wäre mit zusätzlichen Kosten verbunden. Übliche Praxis ist das Erzeugen von Listen der betroffenen Datensätze im Altsystem nach dem Cutover.

Die Konvertierung bibliografischer Metadaten wird immer notwendig sein, da auch mit MARC 21 als Standardformat Felder bibliotheksspezifisch belegt werden. Daher muss die Bibliothek dafür sorgen, dass die Exportdatei den Anforderungen des Anbieters entspricht. Ein einfaches Mapping der Felder ist nicht ausreichend, wenn Export- und Zielformat strukturell nicht kompatibel sind. So werden beim USMARC-Datenexport aus BIBLIOTHECA die Verknüpfungen zwischen Stücktiteln und übergeordneten Werken nicht abgebildet, sodass der Export im BIBLIOTHECA-eigenen Format erfolgen muss, wenn diese Informationen ins Zielsystem übernommen werden sollen.[302] Bei einem Export im MAB2-Format tritt ebenfalls ein Datenverlust auf, daher bevorzugt die VZG Daten im BIBLIOTHECA-Exportformat für eine möglichst verlustfreie Datenkonvertierung.[303] Eine weitere Herausforderung bei Datenmigrationen sind unterschiedliche Datenstrukturen der lokalen Ebene (mit oder ohne Holding-Level zur Kennzeichnung des Besitznachweises zwischen Titelebene und Exemplarsatz).

In den meisten Fällen übernimmt der Anbieter des zukünftigen Bibliothekssystems die Datenkonvertierung; das kann eine kommerzielle Firma sein oder aber die Verbundzentrale eines Bibliotheksverbundes. Die Projektplanung wird dabei ebenfalls vom Anbieter vorgegeben.

Im Folgenden wird neben Beispielen für die Migration eines Bibliothekssystems auch der Umstieg von GBV-Bibliotheken von LBS3 zu LBS4 im Hinblick auf Migrationsablauf und -strategie untersucht.

[301] In den Vortragsblöcken „Die Qual der Wahl: Neue Bibliothekssysteme“ https://opus4.kobv.de/opus4-bib-info/solrsearch/index/search/searchtype/collection/id/16591, „Alma im Verbund“ https://opus4.kobv.de/opus4-bib-info/solrsearch/index/search/searchtype/collection/id/16588 und „Alma in der Anwendung“ https://opus4.kobv.de/opus4-bib-info/solrsearch/index/search/searchtype/collection/id/16589 (alle: 15.05.2018).
[302] Vgl. Becker 2018.
[303] Vgl. Rzehak 2018.

4.4.1 Der Begriff „Migration“ im Bibliothekskontext

Im Kontext von Bibliothekssystemen bezeichnet Migration i. d. R. den Wechsel vom bisher eingesetzten Altsystem zu einem bereits auf dem Markt befindlichen Produkt. Der Markt für BMS weist dabei einige spezifische Charakteristika auf. Die Eigenentwicklungen aus den 1970er Jahren sind mittlerweile abgelöst durch kommerzielle Produkte oder Open-Source-Lösungen. Ausschlaggebend für diese Entwicklung war vor allem die Planbarkeit der finanziellen und personellen Aufwendungen; ebenfalls sind die Erwartungen an den Funktionsumfang einschließlich der Implementierung von Schnittstellen sowie an User Interface und User Experience gestiegen. Begrenzte Entwicklerkapazitäten vor Ort lassen die Entwicklung und Pflege solch komplexer Systeme nicht zu.

Bei den heute verfügbaren kommerziellen Produkten WMS und Alma findet eine Migration zum Zielsystem „*as it is*“ statt. Zusätzlich zeichnet die BMS der neuen Generation ein agiler Entwicklungszyklus aus mit monatlichen automatischen Versionsaktualisierungen. Damit ist eine Bibliothek gefordert, kontinuierlich die lokalen Workflows an das System anzupassen. Über Meldungen per Ticketsystem können Bibliothekskunden Einfluss auf die Programmentwicklung nehmen. Die Priorisierung und Umsetzung der Kundenwünsche obliegt allerdings dem kommerziellen Anbieter, sodass de facto die Einflussmöglichkeit der Bibliotheken eher gering ist. Dies gilt insbesondere für Belange deutscher Bibliotheken aufgrund der überwiegend US-amerikanischen Kundschaft.[304]

Zu einem Zeitpunkt, als die Anbieter der cloudbasierten Systeme Alma und WMS ihre Marketingoffensive in Deutschland begannen, plante ein Konsortium wissenschaftlicher Bibliotheken in Baden-Württemberg die gemeinsame Ablösung ihres bisherigen Bibliothekssystems Horizon (SirsiDynix). Ziel war die Einführung eines landesweit genutzten einheitlichen lokalen Bibliotheksmanagementsystems, das sich unaufwendig in ein Verbundszenario integrieren lässt. 2010 startete der Pilotbetrieb von aDIS/BMS (aStec), die Migrationsphase konnte 2013 abgeschlossen werden. aDIS/BMS basiert auf herkömmlicher Client-Server-Architektur und gehört daher zu den klassischen Bibliothekssystemen, hebt sich jedoch von anderen BMS dieser Gattung durch einen erweiterten Funktionsumfang ab. Dazu gehören die Verwaltung elektronischer Ressourcen und ein Statistikmodul. Daneben werden zahlreiche Schnittstellen bedient. Die inhabergeführte Firma aStec entwickelt das System kontinuierlich nach Kundenanforderungen weiter und passt die Implementierung des Kunden an die lokalen Bedürfnisse an. Für die Bibliotheken in Baden-Württemberg erfolgt der technische Betrieb per SaaS-Hosting durch das Zentrum für Datenverarbeitung der Universität Tübingen, die Systemadministration wird für einen Großteil der Bibliotheken vom BSZ übernommen.[305] Alternativ kann aDIS/BMS auch lokal betrieben werden. Zum Anwenderkreis zählen neben den knapp 50 baden-württembergischen wissenschaftlichen Bibliotheken weitere Bibliotheken unterschiedlicher Sparten, überwiegend aus Deutschland.

Die Open-Source-Lösung Koha[306] wird als Alternative zu kommerziellen Produkten auch in einer Reihe deutschsprachiger Bibliotheken genutzt. Koha kann lokal implementiert oder aber als Hosting-Lösung beim BSZ[307] bzw. einem kommerziellen Dienstleister betrieben werden. Die aktive Community entwickelt das Produkt stetig weiter. Aufgrund der Quelloffenheit ist es möglich, Funktionen zur Lösung bibliothekseigener Erfordernisse in Form von Programmerweiterungen zu

[304] Das automatische Erzeugen von Heftmahnungen für Printzeitschriften, ein dringendes Desiderat deutscher Bibliotheken, ist in Alma noch nicht realisiert (Stand 08.05.2018).
[305] Vgl. https://www.bsz-bw.de/bibliothekssysteme/adis/ (15.05.2018).
[306] https://koha-community.org/ (15.05.2018).
[307] Vgl. https://www.bsz-bw.de/bibliothekssysteme/koha.html (15.05.2018).

ergänzen. Bibliotheken ohne entsprechendes Know-how können dies über Dienstleister realisieren. Somit kann eine Migration zu Koha als Standardprodukt mit einer (Eigen-)Entwicklung gekoppelt werden, um den Funktionsumfang an die Bedürfnisse der Bibliothek anzupassen.

Eine weitere potenzielle Alternative zu kommerziellen Cloudsystemen stellt FOLIO dar. Das unter dem Dach der Open Library Foundation stehende Projekt ist eine komplette Neuentwicklung eines Bibliotheksmanagementsystems, das wahlweise in der Cloud oder lokal betrieben werden kann. Dabei erfolgt die Entwicklung der FOLIO-Plattform und der Basismodule durch kommerzielle Firmen und die OLE Community, unterstützt u. a. durch das FOLIO-Team von VZG und hbz[308]. In thematischen Arbeitsgruppen erstellen Bibliothekare mit entsprechendem Fachwissen die funktionalen Vorgaben für die Programmentwicklung. Die modulare Systemarchitektur und die Integration der Community in den Entwicklungsprozess ermöglichen die Berücksichtigung von lokalen Anforderungen. Geplant ist die Unterstützung verschiedener Support-Modelle: wie bei Koha wird neben einer lokalen Implementierung auch ein Hostingbetrieb möglich sein, mit Unterstützung durch eine Verbundzentrale oder einen kommerziellen Dienstleister. Die technische Plattform bietet die Basis zur Entwicklung zusätzlicher Services und Funktionalitäten als sog. *Apps*.[309]

4.4.2 Datenmigration bei GBV-Verbundteilnahme

Die meisten Bibliotheken, die einen Vertrag über die Teilnahme am GBV abschließen, nutzen bereits ein Bibliothekssystem, vielfach Allegro oder BIBLIOTHECA. In diesen Fällen ist eine Migration der Legacy-Daten erforderlich.

Der Ablauf einer Migration von bibliografischen Metadaten folgt in der VZG einem iterativen Prozess, der hier grafisch wiedergegeben ist:

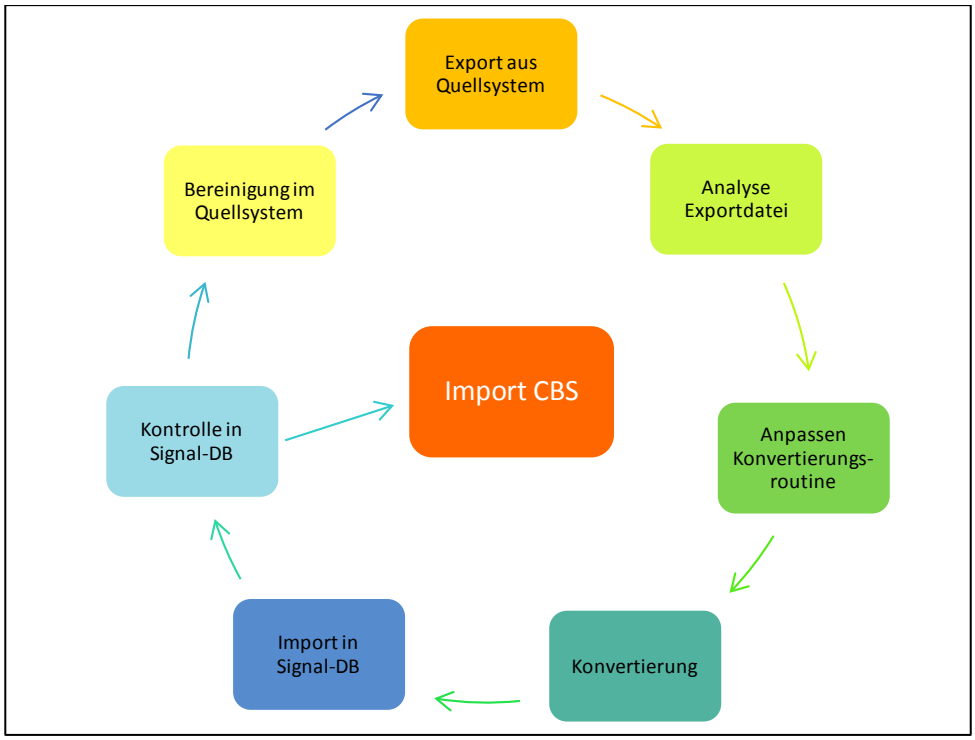


Abb. 10: Iterativer Prozess bei Datenmigrationen in der VZG

[308] https://www.folio-bib.org/ (15.05.2018)
[309] Vgl. Kemner-Heek/Schomburg 2017.

Die konvertierten Daten werden zunächst in eine Testdatenbank als Staging-Area (sog. Signal-Datenbank) geladen. Bibliotheks- und VZG-Mitarbeiter können per WinIBW-Client auf die Daten zugreifen und verschiedene Prüfungen durchführen. Falls eine Implementierung des LBS-Ausleihmoduls geplant ist, werden insbesondere auch ausleihrelevante Daten der Exemplarebene kontrolliert. Ebenfalls müssen Datensätze mit Verknüpfungen geprüft werden, beispielsweise Datensätze für unselbständige Werke. In allen Datensätzen wird die bisherige Identifikationsnummer des Altsystems gespeichert, dies ermöglicht den direkten Vergleich eines Datensatzes in Altsystem und CBS. Die Bibliotheksmitarbeiter kontrollieren die Daten meist stichprobenartig, potenzielle Problemfälle werden systematisch untersucht. Hierbei fallen regelmäßig Datenqualitätsmängel auf, die daraufhin im bisherigen BMS bereinigt werden können.[310] Dies ist beispielsweise der Fall, wenn Angaben im Altsystem per Freitext erfasst wurden, die bei der Konvertierung in einen Code umgesetzt werden. Nach der Prüfphase von meist mehr als acht Wochen wird eine zweite Datenkonvertierung mit zwischenzeitlich angepassten Konvertierungsroutinen durchgeführt. Es folgt ein erneutes Laden in die Testdatenbank. Ist das Ergebnis zufriedenstellend, kann der finale Export der Daten aus dem bisherigen BMS erfolgen.

In der Regel erfolgt die Freigabe durch die Bibliothek nach dem zweiten Laden der Signal-DB. Bei sehr komplexen oder auch unzulänglichen Quelldaten werden die Schritte „Anpassen der Konvertierungsroutine – Konvertieren – Einspielen in die Signal-DB“ noch einige Male wiederholt, wenn sich herausstellt, dass das Ergebnis der Datenumsetzung noch nicht zufriedenstellend ist. Auf einen weiteren Durchlauf wird allerdings verzichtet, wenn absehbar ist, dass der Aufwand für eine weitere Überarbeitung der Konvertierungsroutinen unverhältnismäßig hoch ist. Das ist beispielsweise dann der Fall, wenn die erforderlichen Änderungen eine intellektuelle Prüfung benötigen.

Für den Import in die CBS-Verbunddatenbank wird nicht die Datei mit den konvertierten Daten verwendet; stattdessen sieht das Verfahren ein Verschieben der Daten aus der Signal-DB vor, getrennt nach Datensatz-Typen. Begonnen wird dabei stets mit unverknüpften Monografien, die eine Standardnummer wie ISBN enthalten, da bei diesen im Regelfall bereits eine Verbundaufnahme existiert und lediglich die Exemplardaten bei der vorhandenen Aufnahme ergänzt werden müssen; als Zielaufnahme werden Datensätze mit DNB-Nummer bevorzugt. Es folgen monografische Datensätze ohne Standardnummer. Bei verknüpften Datensätzen wird zunächst die übergeordnete Aufnahme verarbeitet. Bei der Datenumsetzung wird die Altsystem-ID des übergeordneten Satzes zum Zweck der späteren Zuordnung in ein temporäres Unterfeld der Verknüpfungskategorie abgelegt, daher werden die untergeordneten Datensätze vorläufig ohne Verknüpfung eingespielt. Schließlich folgen Aufsatzdaten und Zeitschriften-Bandsätze. Ein weiteres Import-Paket sind Datensätze für Online-Ressourcen; für ein mögliches Matching wird über DOI, URN bzw. URL nach bereits vorhandenen Aufnahmen gesucht. Bei dieser Verfahrensweise erfolgt immer auch ein Dublettenabgleich nach dem in Kap. 3.2.4 beschriebenen Match-and-Merge-Verfahren.

Nachdem alle Datensätze in die Verbunddatendank eingespielt sind, werden über das sog. „Relate“-Programm die Datensatzverknüpfungen hergestellt. Hierzu wird die im temporären Unterfeld gespeicherte Altsystem-ID im CBS gesucht und die PPN des gefundenen Datensatzes im entsprechenden Unterfeld der Verknüpfungskategorie gespeichert. Das temporäre Unterfeld wird nun nicht mehr benötigt und aus dem Datensatz gelöscht.

[310] Tatsächlich unterbleibt dies in den meisten Fällen, sodass in der Konvertierungsroutine uneinheitlich erfasste Daten entsprechend berücksichtigt werden müssen.

Da die im CBS eingespielten Datensätze über einen Online-Update-Prozess direkt in die entsprechende LBS-Datenbank geladen werden, erfolgt das Einspielen in mehreren Blöcken, falls die einzuspielende Datenmenge zu groß für eine zeitnahe Verarbeitung durch das Online-Update-Programm ist; die Grenze liegt hier bei ca. 50.000 Datensätzen.

Die Schritte des ETL-Prozesses bei Datenmigrationen in der VZG sind hier tabellarisch wiedergegeben.

ETL-Schritt	Aktivität	Akteur
Datenkontrolle und -bereinigung	Kontrolle und Bereinigung im Quellsystem (wünschenswert, aber meist nicht erfolgt)	Bibliothek
Extract	Datenexport aus Quellsystem	Bibliothek
	Bereitstellung der Exportdatei auf FTP-Server	
Transform	Download der Exportdatei auf lokalen PC	VZG
	Analyse der Exportdatei	
	Anpassung einer vorhandenen Konvertierungsroutine bzw. Erstellen einer neuen Routine (bei bisher unbekanntem Format) gemäß Datenanalyse und Vorgaben der Bibliothek	
	Datenkonvertierung mit Ausgabe im PICA+-Importformat	
	Ggf. Zeichensatz-Konvertierung zu UTF-8	
Load – Test	Einspielen der PICA+-Datei in Signal-Datenbank mit Nachführen von Datensatzverknüpfungen	VZG
Datenkontrolle und -bereinigung	Kontrolle der Datenumsetzung in der Signal-DB	Bibliothek + VZG
	Datenbereinigung im Quellsystem (selten; mehrheitlich ausleihrelevante Daten)	Bibliothek
Extract	Finaler Datenexport aus Quellsystem; Beginn der Cutover-Phase	Bibliothek
	Bereitstellung der Exportdatei auf FTP-Server	
Transform	Download der Exportdatei auf lokalen PC	VZG
	Endgültige Anpassung der Konvertierungsroutine	
	Endgültige Datenkonvertierung mit Ausgabe im PICA+-Importformat	
	Ggf. Zeichensatz-Konvertierung zu UTF-8	
Load – Test	Endgültiges Einspielen der PICA+-Datei in Signal-DB mit Nachführen von Datensatzverknüpfungen	VZG
Load – Produktion	Verschieben der Daten aus Signal-DB in die CBS-Verbund-datenbank, getrennt nach Art der Datensätze: zuerst unverknüpfte Monografien, danach Datensätze mit Verknüpfungen Zum Abschluss: Nachführen von Datensatzverknüpfungen (sog. „Relate“)	VZG
Datenbereinigung	Ggf. diverse Nacharbeiten in der CBS-Verbunddatenbank, u. a. ausleihrelevante Angaben im Exemplarsatz	Bibliothek + VZG

Tab. 6: Schritte des ETL-Prozesses bei Datenmigrationen in der VZG

Die farbig markierten Schritte werden bei Bedarf wiederholt, wenn bei der Datenkontrolle Mängel identifiziert werden, die durch eine Überarbeitung der Konvertierungsroutine behoben werden können.

Die aufgeführten Aktivitäten sind auch dann erforderlich, wenn die betreffende Bibliothek ihr bisheriges lokales System lediglich zur Verwaltung der bibliografischen Metadaten genutzt hat und/oder zukünftig kein Lokalsystem einsetzen wird. Die ausleihbezogenen Datenprüfungen und -bereinigungen können dann unterbleiben.

In Einzelfällen werden bestimmte Bereinigungen nach Abschluss der Konvertierung per WinIBW-Script durchgeführt, weil sich die Änderungen auf Exemplarkategorien beziehen und außerdem die betroffene Datenmenge gering ist. In diesen Fällen wird also eine nachträgliche Datentransformation nach dem Laden in das Zielsystem praktiziert, wie es bei ELT (siehe Kap. 2.1) der Fall ist. Das CBS bietet hierfür verschiedene Möglichkeiten zur automatisierten Datentransformation. Der Ablauf bei Datenmigrationen in die CBS-Verbunddatenbank stellt sich dann genaugenommen als ETLT (Extract – Transform – Load – Transform) dar.

Zu den Nacharbeiten in der CBS-Verbunddatenbank gehören automatisierte Datenkorrekturen wie das pauschale Einfügen einer Standortangabe oder die Änderung des Ausleihindikators für einen bestimmten Signaturenbereich; dies erfolgt überwiegend durch Mitarbeiter der VZG unter Nutzung der Funktion „sucheErsetze“ im Katalogisierungs-Client WinIBW. Weitere Korrekturen sind von der Bibliothek manuell zu erledigen, z. B. die Bereinigung von doppelt erfassten Verbuchungsnummern oder das Hinzufügen eines Konvolut-Kennzeichens, wenn dies bei der Konvertierung nicht zweifelsfrei identifiziert werden konnte. Ebenfalls müssen fehlende Datensatzverknüpfungen manuell nachgeführt werden, falls die entsprechenden Beziehungen aus den Quelldaten nicht eindeutig hervorgegangen sind und demzufolge bei der Konvertierung nicht berücksichtigt werden konnten. Arbeitsaufwendig sind Korrekturen, die aufgrund gravierender Unterschiede bei der Datenstruktur zwischen Quellsystem und CBS und entsprechend abweichender Erfassungspraxis notwendig sind. Im Ausnahmefall wird für Nacharbeiten ein speziell dafür angepasstes („Reparatur“-)Programm verwendet, wenn für die erforderlichen Korrekturen die vollständige Übernahme der Altdaten inkl. Nachführen von Datensatzverknüpfungen abgeschlossen sein muss.[311]

Für Zeitschriften existiert ein separater Ablauf, da im GBV Zeitschriftentitel direkt in der ZDB erfasst werden und die VZG Zeitschriftenbestände aus den Konvertierungsdaten direkt an die ZDB meldet.

Die Dauer der Datenmigrationsphase beträgt i. d. R. 4–6 Monate. Ab dem Zeitpunkt des finalen Datenexports aus dem Altsystem können die Bibliotheken mit der Katalogisierung von Neuerwerbungen im CBS beginnen. Das gilt allerdings nur für einbändige Monografien, für die keine Exemplardaten im bisherigen System vorhanden sind. Weitere Teile einer bereits vorhandenen MTM können ebenfalls schon katalogisiert werden. Der Freeze-Zeitraum zur Sicherstellung der Datenintegrität gilt entsprechend nur für die Bearbeitung von zum Zeitpunkt des finalen Datenexports bereits vorhandenen Katalogisaten.

Die Datenmigration bei einer Verbundteilnahme wird demnach als Big Bang durchgeführt; zeitlich unabhängig davon ist der Wechsel des Katalogisierungsclients zu WinIBW3 für die Erfassung neuer Katalogisate bzw. Besitznachweise in der CBS-Verbunddatenbank. In Einzelfällen wurden übergangsweise, bis zur Bereitstellung des LBS-Ausleihmoduls, Medien zusätzlich im Altsystem katalogisiert, wenn solche Datensätze für eine Ausleihe im Altsystem benötigt werden.[312]

[311] Zu den in der VZG genutzten Reparaturprogrammen siehe Kap. 3.2.2.

[312] Die OUS-Implementierung erfolgt meist zeitgleich zur Konvertierung; das Ausleihmodul kann jedoch erst in Betrieb genommen werden, nachdem die Datenkonvertierung abgeschlossen ist.

4.4.3 Migration von LBS3 zu LBS4

Für das im GBV 1993 eingeführte lokale Bibliotheksmanagementsystem LBS3[313] wurde Anfang der 2000er Jahre vom Anbieter ein Nachfolgesystem unter dem Namen LBS4 entwickelt. Der Sprung in der Versionsbezeichnung kennzeichnet die einschneidenden technischen Änderungen gegenüber der Vorgängerversion. Seit 2003 wird für neue Verbundbibliotheken ein LBS4-Lokalsystem eingerichtet, für die bestehenden Bibliotheken mit LBS3 musste ein Migrationskonzept entwickelt werden.

Bei der Migration von LBS3 zu LBS4 lassen sich die in Kap. 4.3.2 genannten Charakteristika der Migrationsstrategie Chicken Little ableiten. Es kommen ein neuer Anwendungsclient und systemseitig ein neuer Anwendungsserver sowie neue Schnittstellen zum Einsatz; eine Datenmigration ist nicht erforderlich.[314]

Seit 2007 begleitet die LBS-Gruppe der VZG den Umstiegsprozess von LBS3 auf LBS4 in den Bibliotheken des GBV.[315] Die Migration erfolgt modulweise: In vielen Fällen wird zunächst der Umstieg des Ausleihmoduls auf OUS4 durchgeführt, während die Erwerbungsabläufe weiterhin mit dem Erwerbungsmodul ACQ3 praktiziert werden. Für den Wechsel zu ACQ4 wird ein weiterer Umstellungszeitpunkt verabredet. Grund für dieses Verfahren ist zum einen die Inanspruchnahme des LBS-Systemverwalters, der umfangreiche Umstellungsarbeiten durchzuführen hat. Zum anderen ist in zahlreichen Bibliotheken eine Trennung in modulspezifische Arbeitsbereiche nicht gegeben. Aufgrund des inkrementellen Migrationsverfahrens können sich die Mitarbeiter zeitlich versetzt mit dem jeweils neuen Modul vertraut machen.

Erst nachdem alle Bibliotheken eines LBS-Standortes[316] auf LBS4 umgestellt sind, kann ein Wechsel zu LOAN4, der LBS4-Ausleihschnittstelle des OPACs, erfolgen. Bis dahin werden von den ausleihspezifischen OPAC-Funktionen wie (Magazin-)Bestellung, Vormerkung und Verlängerung noch die betreffenden LBS3-Systemeinstellungen ausgewertet. Dies gilt auch für die Funktionalität von Selbstverbuchungsgeräten o. ä., solange die entsprechende GOSSIP-Schnittstelle mit LBS4-Anbindung als Ersatz für SIP2 noch nicht fertig entwickelt ist. Während des Parallelbetriebs von LBS3 und LBS4 ist u. a. die Zetteldruck-Steuerung für beide LBS-Versionen vorzuhalten, da die Anwendungen in den beiden LBS-Versionen verschiedene Dateipfade und einen unterschiedlichen Zeichensatz nutzen.

Das Migrationsvorhaben ist zeitversetzt durchführbar, weil die Anwendungen LBS3 und LBS4 auf dieselbe Datenbank zugreifen. Mit Einführung von LBS4 wurde die Datenbankstruktur beibehalten, es sind lediglich einige Tabellen und Tabellenfelder hinzugekommen, die von den LBS3-Anwendungen OUS3 und ACQ3 nicht genutzt werden. Daher können die ca. 120 LBS3-Bibliotheken in standortspezifischen Migrationsprojekten bibliotheks- bzw. modulweise auf LBS4 migrieren.

Die Migration von LBS3 zu LBS4 im GBV erfolgt demnach inkrementell, auch wenn der Umstieg eines einzelnen Migrationspakets (Migration eines Moduls in einer Bibliothek) als Stichtagsumstellung praktiziert wird.

---

[313] Vgl. Bossers 2005, S. 81.

[314] Allerdings ist der Transfer von Daten eines ausleihbezogenen Tabellenattributs notwendig, weil in der LBS4-Anwendung ein unter LBS3 verwendetes Kommentarfeld nicht nutzbar ist. Diese Datenmigration wird einmalig im Rahmen der Cutover-Phase durchgeführt.

[315] Der lange Zeitraum von mehr als 10 Jahren ist vor allem den Vorgaben der Bibliotheken hinsichtlich des gewünschten Migrationszeitraums geschuldet.

[316] Entspricht einer LBS-Installation und umfasst meist mehrere Bibliotheken.

4.4.4 Migration zu Cloud-Systemen: Alma

Die Migration eines lokalen Bibliothekssystems zu einem New-Generation-System in der Cloud bedeutet für eine Bibliothek nicht nur eine Umstellung der Anwendung, sondern den Wechsel zu einer gänzlich anderen Systemphilosophie. Bei vielen Bibliotheken ist eine Migration mit der Ablösung der selbst betriebenen Serverlösung verbunden, für Betrieb und Wartung ist dann der Cloudanbieter zuständig. Eine solche Übertragung der Zuständigkeiten ist allerdings auch bei der Wahl eines klassischen BMS im Hostingbetrieb gegeben. Der zentrale Unterschied besteht vor allem in der Möglichkeit des Datenzugriffs: bei einer Cloudlösung ist der Zugriff auf die Datenbank gekapselt und lediglich über Reporte oder APIs möglich.

In Deutschland haben inzwischen einige Bibliotheken ihr bisheriges lokales BMS zugunsten von Alma aufgegeben, WMS wird zurzeit lediglich von der FH Münster genutzt. Aus diesem Grund wird exemplarisch der Migrationsablauf zu Alma beschrieben.

Die Ausgestaltung des ETL-Prozesses für die Datenmigration ist davon abhängig, ob es sich bei dem bisherigen BMS um ein Ex-Libris-Produkt handelt. Bei Ex-Libris-Produkten (z. B. Aleph) werden die Schritte *Extract* und *Transform* auf dem Bibliotheksserver des Kunden durchgeführt, hierfür wird der Bibliothek ein entsprechendes Tool zur Verfügung gestellt. Anderenfalls stellt die Bibliothek die eigenen Daten[317] auf einem FTP-Server bereit. Ein Validationstool prüft die extrahierten bzw. exportierten Daten auf Gültigkeit.

Zum besseren Verständnis des neuen Systems stellt Ex Libris seinen Kunden eine Testumgebung („Sandbox“) zur Verfügung, die in der Standardversion ca. 70.000 bibliografische Metadatensätze mit einer Standardkonfiguration enthält. So erhält die Bibliothek Einblick in Systemphilosophie und Datenhaltung und kann daraus Erkenntnisse für die Mapping- und Konfigurationsvorgaben gewinnen, dazu gehören insbesondere Spezifikationen zu Exemplardaten und Bestandsdaten (*Holdings*). Die nach den Vorgaben der Bibliothek konvertierten Daten werden im Zuge einer ersten Testmigration in die zukünftige Produktionsumgebung geladen. Der Migrationsplan sieht generell nur eine Testmigration vor der endgültigen Datenmigration vor[318], allerdings wird meist eine zweite Testmigration[319] durchgeführt, da die Auswirkungen der Migrationsvorgaben seitens der Bibliothek zu diesem Zeitpunkt oftmals nicht überblickt werden können. Auch hier erfolgt die Datenmigration in Iterationen: Stellt die Bibliothek nach einer ersten Migration gravierende Probleme fest, werden die Konvertierungsroutinen angepasst und es folgt eine weitere Testmigration.

Eine Datenbereinigung findet im Rahmen der Konvertierung nicht statt. Daher sollten die Daten im bisherigen BMS vorher in der Weise bereinigt bzw. angepasst werden, dass sie für die zukünftigen Alma-Workflows geeignet sind; Ex Libris macht hierzu umfangreiche Vorschläge.[320]

Für den Freeze von Katalogdaten wird meist ein Zeitraum von ca. 4 Wochen veranschlagt. Der Ausleihbetrieb kann noch weitergeführt werden, bis zum Beginn der Cutover-Phase für Benutzer- und Ausleihdaten. In diesem Zeitraum[321] werden die Benutzer- und die Ausleihdaten erneut an Ex Libris übermittelt und es erfolgt die letzte Konvertierung dieser Daten, die zum Go live ebenfalls im Alma-Produktionssystem zur Verfügung stehen.

[317] Für bibliografische und Bestandsdaten in MARC 21 oder MARCXML, andere Daten als CSV.
[318] Vgl. Ex Libris 2017c, S. 3–4.
[319] Für die Migration der FU Berlin war eine dritte Testmigration erforderlich.
[320] Vgl. Appendix B *Optional Data Preparations* in Ex Libris 2017c, S. 14–16.
[321] Meist werden hierfür zwei Arbeitstage veranschlagt.

Für die Berliner Bibliotheken konnte die ursprünglich geplante Katalogisierung in Alma mit Datentransfer zum B3Kat aufgrund mangelnder Praxistauglichkeit der verlustfreien Kreiskonvertierung nicht realisiert werden, daher findet die Primärkatalogisierung weiterhin im B3Kat per Aleph-Client statt.[322] Zur Aktualisierung der Daten in der *Institution Zone* der jeweiligen Bibliothek ist eine Versorgungsschnittstelle mit Datenkonvertierung von MAB2/ASEQ zu MARC 21 implementiert, ebenfalls kann eine manuelle Datenübernahme per Z39.50 erfolgen. Auch bei den Alma-Bibliotheken aus CBS-Verbünden erfolgt die Primärkatalogisierung nach wie vor per WinIBW-Katalogisierungsclient in der jeweiligen Verbunddatenbank. Das hat den Vorteil, dass für das Metadatenmanagement weiterhin die bewährten Verfahren genutzt werden können.

Migration zu Alma im Österreichischen Bibliothekenverbund (OBV)[323]

Eine vergleichbare Ausgangssituation wie im GBV gab es im OBV mit knapp 70 Verbundteilnehmern und einer homogenen Systemstruktur der Fa. Ex Libris: Aleph wird (noch) als Verbundsystem genutzt, die Bibliotheken setz(t)en Aleph (bzw. Alephino) als Lokalsystem ein. Anfang 2016 startete das Projekt „Alma im OBV“ mit dem Ziel, Lokalsysteme und Verbundsystem durch Alma abzulösen. Auch im OBV konnten bzw. wollten die Einrichtungen nicht zeitgleich auf ein neues System umstellen, daher erfolgt die Migration in mehreren Phasen, die ersten 13 Umstiegsbibliotheken wurden dazu in zwei Gruppen („Kohorten“) eingeteilt. Die Migration selbst wird als Big Bang mit einem festen Stichtag für alle Bibliotheken einer Kohorte praktiziert.

Somit ergibt sich eine komplexe Systemstruktur mit Parallelbetrieb während der mehrjährigen Migrationsphase, in der eine konsistente Datenhaltung in den beteiligten Systemen gewährleistet sein muss.

Phase 1 des Parallelbetriebs hatte im August 2017 begonnen: die Bibliotheken der ersten Kohorte arbeiten seitdem mit Alma als Lokalsystem in jeweils einer *Institution Zone*, zur Katalogisierung wurden die Datensätze im Aleph-Verbundsystem erfasst bzw. recherchiert und über einen Z39.50-Abruf nach Alma importiert. Updates von Daten in der Verbunddatenbank wurden automatisch in den lokalen *Institution Zones* nachgeführt. Beim Import eines Datensatzes aus der Aleph-Verbunddatenbank nach Alma setzt ein Aleph-Konverter die MAB2-Daten aus Aleph nach MARC 21 um. Der Umstieg der zweiten Kohorte fand im Januar 2018 statt.

Zum Start von Phase 2 erfolgte im März 2018 die Inbetriebnahme von Alma als primärem System in der neuen Verbundarchitektur, damit sind die Daten der Alma-Bibliotheken aus der bisherigen Aleph-Verbunddatenbank in die Alma *Network Zone* umgesetzt. Die lokalen *Institution Zones* der Alma-Bibliotheken werden mit der *Network Zone* gekoppelt, die nun für diese Bibliotheken als Verbunddatenbank fungiert.[324]

Die noch verbleibenden Aleph-Bibliotheken nutzen weiterhin als Katalogisierungsplattform das bisherige Aleph-Zentralsystem, das die bibliografischen und Bestandsinformationen aller OBV-Bibliotheken in MAB2 enthält. Über eine sog. „Aleph-Bridge“ ist eine Datensynchronisation zwischen Aleph-Datenbank und *Network Zone* in Alma eingerichtet, die Datenkonvertierung von MAB2 zu

[322] Vgl. Christof 2017, Folie 16.
[323] Vgl. Kann 2017, S. 568–572.
[324] https://www.obvsg.at/wir-ueber-uns/aktuelles/news/go-live-der-alma-network-zone/ (15.05.2018).

MARC 21 muss dabei in beiden Richtungen (Kreiskonvertierung) erfolgen.[325] Den Aleph-Bibliotheken bleibt damit ein System- und Formatbruch erspart.

Zusätzlich ist für den Daten-Rückimport von Alma nach Aleph eine sog. Hybridisierung geplant: Datensätze, die nach dem Regelwerk RAK katalogisiert wurden, erhalten dabei eine Formatanreicherung für RDA. Vermutlich trägt dies zu einer weiteren Komplexität im Hinblick auf die Beibehaltung einer konsistenten Datenhaltung bei.

Der Umstieg auf Alma im OBV ist ein komplexes Projekt, da wegen der großen Zahl beteiligter Institutionen eine mehrjährige Phase des Parallelbetriebs erforderlich ist. Ein MAB2-MARC21-Konverter wird aufgrund der unterschiedlichen Datenformate in beide Richtungen benötigt. Damit betreibt die Verbundzentrale des OBV über einen längeren Zeitraum de facto zwei Verbundsysteme, die mit Hilfe einer kontinuierlichen Datensynchronisation aktualisiert gehalten werden müssen.

[325] Vgl. Ex Libris 2018b.

5 Projekt: Migration lokaler Katalogisate in die Verbunddatenbank des GBV

Im GBV findet die Katalogisierung kooperativ in der zentralen Verbunddatenbank CBS statt. Der Zugang zum CBS erfolgt über den lokal zu installierenden Client WinIBW. Zur Erfassung von Metadaten für Materialien, die nicht zum eigentlichen Bestand einer Bibliothek zählen, wird die sog. lokale Katalogisierung im LBS genutzt.

Im Rahmen der langfristigen Versionsplanung hatte OCLC bereits 2014 angekündigt, das Modul für die lokale Katalogisierung CAT4 in zukünftigen Versionen von LBS4 nicht mehr zu unterstützen. Für Version 2.12, mit deren Auslieferung 2018 zu rechnen ist, ist eine CAT4-Integration nicht mehr vorgesehen.[326] Diese Entscheidung war der Auslöser für das Projekt zur Migration der lokalen Katalogisate ins CBS.

Der Wechsel zur Katalogisierungsplattform CBS für lokale Katalogisate ist daher bei allen LBS-Bibliotheken dringend geboten. Derzeit wird in 100 LBS-Bibliotheken die lokale Katalogisierung aktiv genutzt, mit einem Datenvolumen von zusammen mehr als 2,51 Mio. lokalen Katalogisaten (Stand: 15.05.2018). Aufgrund der großen Anzahl betroffener Bibliotheken wird für das Migrationsprojekt ein längerer Zeitraum veranschlagt werden müssen.

Das Projekt zur Migration lokaler Katalogisate in die Verbunddatenbank des GBV gliedert sich in zwei Teile. Das Vorprojekt beinhaltete die Umstellung auf lokale Katalogisierung im CBS. Dazu musste die Erfassung solcher Katalogisate im CBS technisch ermöglicht werden.

Das Hauptprojekt umfasst die Projektphasen für den Transfer der lokalen Katalogisate im LBS ins CBS. Die Arbeitspakete des Projekts lassen sich den Phasen einer Datenmigration in der Gliederung nach Lüssem/Harrach[327] zuordnen. Daher wird bei der Beschreibung der Aktivitäten auf die entsprechende Phase verwiesen.

Der in der VZG gängige Sprachgebrauch im Rahmen des Projekts ist „Transfer“ lokaler Katalogisate (und nicht „Migration“), daher werden hier ebenfalls die Begriffe „Transfer“ und „Transferverfahren“ verwendet.

[326] Bibliotheken, die zzt. noch LBS3 einsetzen, sind hierbei außer Acht gelassen, da von den aktuell 35 LBS3-Bibliotheken bis zum Jahresende voraussichtlich 30 auf LBS4 migriert sein werden und für diese dann auch die LBS4-Bedingungen gelten werden (Stand: 02.05.2018).
[327] Vgl. Lüssem/Harrach 2013, S. 3–4.

5.1 Darstellung der Ausgangssituation

Für die Rahmenbedingungen des LOK-Projektes ist ein Verständnis des Kontexts von Verbundstruktur und Verbundkatalogisierung unerlässlich, daher erfolgt zunächst eine Darstellung der Situation im GBV.

Der GBV verfolgt die Strategie, alle teilnehmenden Bibliotheken mit einem einheitlichen Lokalsystem zu versorgen. Das lokale Bibliotheksmanagementsystem LBS mit den Modulen OPAC, Ausleihe und Erwerbung nutzen vor allem die wissenschaftlichen Bibliotheken im GBV (mit einer Ausnahme). Bei dieser Systemstruktur erfolgt der Datenfluss ausschließlich vom Verbundsystem ins Lokalsystem, die beide von demselben Anbieter stammen. Im Unterschied zu anderen Bibliotheksverbünden werden sämtliche Elemente eines Katalogisats in der zentralen Katalogisierungsdatenbank CBS erfasst, auch lokale Daten (wie bibliotheksspezifische Sacherschließung) und Exemplardaten mit (u. a.) Signatur und Verbuchungsnummer. Die direkte Anbindung des Lokalsystems an das CBS ist für die Verbundbibliotheken vorteilhaft: Die im CBS erfassten Daten gelangen über einen in der LBS-Umgebung implementierten Prozess direkt in die angeschlossenen LBS-Systeme, dasselbe gilt auch für Datensatzkorrekturen und -löschungen. Dieser sog. Online-Update-Mechanismus (OUM)[328] sorgt dafür, dass die im CBS mit einem Besitznachweis einer Bibliothek versehenen Datensätze in der betreffenden LBS-Datenbank ohne zeitliche Verzögerung laufend nachgeführt werden, die Datensätze werden vollständig repliziert, da die Datenhaltung im LBS wegen der Nutzung der lokalen Module Erwerbung, Ausleihe und OPAC erforderlich ist.

Unter den LBS-Bibliotheken sind zzt. 112 Bibliotheken in Länderfinanzierung, diese verteilen sich auf 27 sog. LBS-Standorte, denen jeweils 1-15 Bibliotheken zugeordnet sind. Für jeden LBS-Standort ist eine LBS-Installation eingerichtet, teils auf vor Ort betriebener Hardware, überwiegend als Hosting-Lösung in der VZG mit virtuellen Servern. Die bibliothekarisch-technische Betreuung der Bibliotheken eines LBS-Standortes wird von einem Systemverwalter durchgeführt.[329] Die zzt. 80 Bibliotheken mit kostenpflichtigem LBS-Service sind einer der drei von der VZG betreuten LBS-Installationen zugeordnet, hier übernimmt die LBS-Gruppe der VZG die Systembetreuung. Auch Bibliotheken ohne Ausleih- bzw. Erwerbungsmodul gehören zu einem LBS-Standort, diese nutzen lediglich den lokalen OPAC.

Zur Identifizierung in der CBS-Datenbank sind die Besitznachweise einer Bibliothek einer ILN (Internal Library Number)[330] zugeordnet. Innerhalb einer LBS-Installation ist eine Datenbank eingerichtet, in der die Daten aller Bibliotheken des LBS-Standortes gespeichert sind. Zur Mandantentrennung wird gleichermaßen die ILN verwendet, in einigen Tabellen jedoch auch eine sog. Bestandsnummer (niederländ. *stelsel*), die in den Datenbanktabellen als Attribut *fno* (=File Number) bezeichnet wird.[331]

Hier und im weiteren Verlauf dieser Arbeit ist mit „Bibliothek“ eine Institution mit eigener ILN bezeichnet, tatsächlich sind unter einer ILN auch zweischichtige Bibliotheksysteme mit einer zentralen Bibliothek und bis zu 30 Fakultäts-/Institutsbibliotheken subsumiert.

[328] Diese Funktion wird in Aleph-Verbünden als „Versorgungsschnittstelle“ bezeichnet.
[329] Die Zuständigkeiten sind meist auf mehrere Personen verteilt, dennoch wird im Weiteren die LBS-Betreuung eines Standorts als LBS-Systemverwalter (sing.) bezeichnet.
[330] Und einer ELN (External Library Number) zur Steuerung der Fernleihe.
[331] Im GBV gilt eine 1:1-Zuordnung von ILN und fno.

5.1.1 Lokale Katalogisierung im LBS

Bibliografische Daten und lokaler Bestand werden im CBS erfasst und gepflegt und über das Online-Update sofort in das jeweilige LBS übertragen. Daneben gibt es Bedarf zur Erfassung von Metadaten für Materialien, die nicht zum eigentlichen Bestand einer Bibliothek zählen. Dazu gehören:

- Titel der nehmenden Fernleihe,
- Schließfachschlüssel, Bücherwagen oder Laptops,
- Lizenzen für E-Ressourcen[332],
- Verbrauchsmaterial, Fotokopien, Buchbinderkosten,
- Lokale Fernsehmitschnitte auf Video, selbst angefertigte Diasammlungen,
- Diplom-, Bachelor- und Masterarbeiten,
- Material für den Schriftentausch,
- Dummy-Sammel-Titel, an die vorhandene ACQ-Bestellungen umgehängt werden, bevor ein Exemplarsatz im CBS gelöscht wird, da es ansonsten zu Dateninkonsistenzen in der LBS-Datenbank kommt.

In LBS3 existiert für diese Zwecke das integrierte Modul OWC (niederländ.: *Online Werkcatalogus*, Dienstkatalog), für LBS4 war ein separates Katalogisierungsmodul CAT4 entwickelt worden. Die Katalogisierung sowohl im OWC als auch in CAT4 erfolgt über den WinIBW-Client. Ein Großteil der LBS-Bibliotheken nutzt die lokale Katalogisierung; die Zahl der lokalen Datensätze bewegt sich dabei jeweils zwischen einigen Dutzend und mehr als 750.000.

Das Anlegen einer Katalogaufnahme für diese Materialien ist Voraussetzung für eine Ausleihverbuchung, eine Lieferanten-Bestellung oder die Nutzung der LBS4-Tauschverwaltung.

Ein LBS-OPAC zeigt in der Standardpräsentation für Endbenutzer nur diejenigen Titel bzw. Exemplare, die bei der Katalogisierung entsprechend gekennzeichnet sind. Dazu wird in Kategoriegruppe `70xx` die Position 1 des sog. Selektionsschlüssels ausgewertet, sodass Exemplare mit einem Nichtanzeige-Code für die Anzeige im Publikumskatalog unterdrückt werden. Besitzen alle Exemplare eines Titels einen Nichtanzeige-Code, so wird auch die Ausgabe der Titelinformationen unterdrückt. Ein Großteil der lokalen Katalogisate ist mit einem solchen Nichtanzeige-Code versehen, dies sind vor allem Titel, die zu Fernleihzwecken erfasst wurden. Die Katalogisierungsrichtlinie für Kategoriegruppe `70xx` sieht für diese Fälle die Codes „d“ und „`i`“ vor.[333] Für den bibliotheksinternen Zugriff auf sämtliche Titel bzw. Exemplare im OPAC, auch diejenigen mit Nichtanzeige-Code, ist ein Zugriff auf den sog. OWC-Katalog auf einfache Weise möglich, indem man die Angabe für den (Datenbank-)Bestand in der URL von `DB=<`*`fno`*`>` zu `DB=OWC.<`*`fno`*`>` ändert. So kann auch über HTTP-basierte Schnittstellen auf alle lokalen Katalogisate per OWC-Katalog zugegriffen werden.

In den folgenden Ausführungen bezieht sich der Begriff „lokale Katalogisate“ bzw. „L-Sätze“ auf Datensätze, die direkt im LBS erfasst wurden. Diese sind in der Kategorie `0500` für die Materialart mit „L“ an der 1. Position gekennzeichnet.[334] Die Positionen 2 und 3 wurden von den Bibliotheken individuell belegt, daher gibt es neben der Standardangabe „Luy“ noch zahlreiche weitere wie „Lux“ oder „Lua“.

[332] Dem LBS fehlt ein ERM-Modul.
[333] Vgl. Verbundzentrale des GBV 2006.
[334] Vgl. http://swbtools.bsz-bw.de/cgi-bin/help.pl?cmd=kat&val=0500&regelwerk=RDA&verbund=GBV (15.05.2018).

Zur Unterscheidung werden die gemäß der neuen sog. LOK-Richtlinie[335] im CBS erfassten lokalen Katalogisate als „Lax-Sätze“ bezeichnet, da für diese der Materialart-Code „Lax“ in Kategorie 0500 als verpflichtend festgelegt wurde.

Um eine einheitliche Katalogisierungspraxis der verschiedenen Medientypen im Verbundkatalog zu gewährleisten, soll die lokale Katalogisierung ausschließlich für die in Kap. 5.1.1 genannten Zwecke genutzt werden. Dementsprechend bestehen zur Erfassung von lokalen Katalogisaten im LBS laut OCLC-Dokumentation[336] einige Einschränkungen gegenüber der Verbundkatalogisierung im CBS:

1. Es sind nur Kategorien aus Titel- und Exemplarlevel erlaubt.
2. Es kann nur ein Exemplarsatz pro Titel angelegt werden.
3. Verknüpfungen zu anderen Titel oder Normsätzen sind nicht möglich.
4. Die Materialart in Kategorie 0500 beginnt mit einem „L“, d. h. die im CBS übliche Differenzierung der Medientypen ist nicht möglich.

Allerdings war es dennoch möglich, zumindest die Beschränkungen der Punkte 2 und 3 zu umgehen, aufgrund der Autonomie der Bibliotheken bzw. ihrer Systemverwalter sowie bei Offline-Datenimporten ins LBS durch die VZG. Die Auswirkungen dieser Ausnahmen werden in Kap. 5.3 aufgegriffen.

Klassische lokale Katalogisate verwenden i. d. R. nur einen geringen Umfang der möglichen PICA-Kategorien. In der LBS3-Systemkonfiguration sind über eine Konfigurationsdatei die zulässigen Kategorien bibliotheksspezifisch festgelegt. In CAT4 gelten die genannten Einschränkungen ebenfalls, allerdings ist die Auswahl der nutzbaren Kategorien erheblich größer.

Diese Einschränkungen sollten die Verbundteilnehmer davon abhalten, Titeldaten und Bestand an der kooperativen Verbundkatalogisierung vorbei nur im Lokalsystem zu erfassen. Lokale Katalogisate sollten als Ausnahme nur für die genannten Zwecke genutzt werden. Wie ist es trotzdem zu der großen Anzahl lokaler Katalogisaten in einigen Bibliotheken gekommen?

5.1.2 Lokale Katalogisate in LBS3 und LBS4

Für die große Menge an lokalen Katalogisaten sind im Wesentlichen diese Gründe verantwortlich:

Einspielen von Daten ins LBS

Für einige Bibliotheken sind bei der Migration ihres vorherigen lokalen BMS zum LBS auch Datensätze direkt in die LBS-Datenbank eingespielt worden. Unter den Daten des Legacysystems waren auch Datensätze ohne nennenswerte bibliografische Angaben, sodass ein Import ins CBS als nicht zielführend erachtet wurde. Diese Datensätze wurden beispielsweise mit der fiktiven Titelangabe „o.T.“ („ohne Titel“) ins LBS importiert.

Unterdrückung der Sichtbarkeit im GVK

Andere Bibliothek haben Hochschulabschlussarbeiten, Dienstexemplare oder DIN-Normen als lokale Katalogisate im LBS erfasst, um damit die Sichtbarkeit im Verbundkatalog zu verhindern. Dass derselbe Effekt durch das Setzen eines speziellen Nichtanzeige-Codes im Exemplarsatz erreicht werden kann, ist kaum einer Bibliothek bekannt.

[335] Verbundzentrale des GBV 2017.
[336] OCLC 1995, S. 23.

„Not“-Sätze für die Ausleihe

Bei der Medienausleihe kann aus verschiedenen Gründen eine Verbuchung blockiert sein. Das ist der Fall, wenn z. B. die Verbuchungsnummer (Barcode) im LBS nicht vorhanden oder die im Exemplarsatz erfasste Signatur ungültig ist. Auch ist ein im CBS neu angelegter Exemplarsatz im Falle eines verzögerten Updates vom CBS ins LBS für eine Verbuchung nicht verfügbar. Ebenso kann aus verschiedenen Gründen für das betreffende Medium ein CBS-Katalogisat fehlen. In einer Reihe von Bibliotheken legen die Mitarbeiter der Ausleihtheke dann ein lokales Katalogisat an, damit die Verbuchung des Mediums sofort durchgeführt werden kann.

Zur Identifizierung der Ursache, warum ein Medium nicht verbucht werden kann, sind meist erweiterte Kompetenzen bzw. Systembefugnisse erforderlich, oder aber es fehlt an der nötigen Ruhe für entsprechende Ermittlungen im Tagesgeschäft.

Sinnvoll wäre, diese Fälle spätestens nach der Rückbuchung zu bereinigen; dazu müsste allerdings eine entsprechende Dokumentation des Problemfalls stattfinden. Hierfür fehlt zumeist das Problembewusstsein oder die Kenntnis der entsprechenden Zusammenhänge, sodass solche lokalen Katalogisate dauerhaft im LBS verbleiben.

Fernleihe

Einige LBS-Bibliotheken nutzen für die aktive Fernleihe sog. „rote Pappen“, dabei handelt es sich um selbsterstellte laminierte Pappstreifen mit einem aufgeklebten Barcode-Etikett. Diese Pappstreifen werden wiederverwendet, wie auch das zugehörige Katalogisat bei einem weiteren Fernleihfall nachgenutzt wird. Bei diesem Verfahren bleibt die Menge der zu Fernleihzwecken angelegten lokalen Katalogisate konstant.

Andere Bibliotheken erfassen für jeden Fernleihfall ein neues Katalogisat und weisen diesem jeweils eine neue Verbuchungsnummer zu.[337] Unter LBS3 ist es möglich, solche lokalen Katalogisate mittels eines Systemprogramms `kill_local_titles` zu löschen; ein Automatismus zum Löschen solcher Datensätze existiert allerdings nicht.[338]

Voraussetzung für das Löschen per Programm ist, dass der Exemplarsatz nicht mehr mit dem verbuchungsrelevanten Bandsatz[339] verknüpft ist, daher wird in einigen Fernleihabteilungen nach der Rückbuchung der Bandsatz manuell gelöscht. Aufgabe der Systemadministratoren war es, von Zeit zu Zeit das Löschprogramm zu starten. Solange nur Fernleihtitel als lokales Katalogisat erfasst werden, ist dieses Verfahren unproblematisch. Da als Programmparameter ein PPN-Bereich anzugeben ist, wurde die Nutzung des Programms erschwert, als weitere (vermeintliche) Anlässe zur Erfassung lokaler Katalogisate hinzukamen und daher der korrekte PPN-Bereich nur mit erheblichem Aufwand zu ermitteln war.

Auch aus diesem Grund hatten einige Bibliotheken damit begonnen, die unterschiedlichen Medientypen über spezifische Angaben in Kategorie `0500` zu codieren, da als weiterer obligatorischer Parameter für das Löschprogramm die Materialart anzugeben ist. Somit konnten Bibliotheken für Fernleihtitel die bisher gebräuchliche Materialart „Luy“ weiternutzen, während für andere lokale Katalogisate abweichende Codes als Materialart erfasst wurden, immer jedoch mit „L“ als erstem

[337] Angesichts der Preise für Barcode-Etiketten ist das heutzutage verschmerzbar.
[338] Wünschenswert wäre beispielsweise ein Offline-Programm, das FL-Titel nach einer angemessenen Frist löscht: nach der Rückgabe durch den lokalen Bibliotheksnutzer und Rückversand an die Lieferbibliothek.
[339] Im LBS erfolgt die Verbuchung über den sog. Bandsatz, nicht über den Exemplarsatz.

Buchstaben in Kategorie `0500`. Das LBS3-Löschprogramm konnte somit gezielt auf Fernleihtitel angewendet werden, sofern bei diesen der Bandsatz gelöscht worden war.

Da es aber beim Betrieb des LBS i. d. R. keine spürbare Rolle spielt, ob und wieviel obsolete Datensätze vorhanden sind, ist eine Löschung zumeist unterblieben. Für LBS4 existiert kein entsprechendes Löschprogramm. In der Folge ist in den meisten Bibliotheken die Menge der lokalen Fernleihtitel stetig angewachsen.

5.1.3 Bestandsaufnahme der lokalen Katalogisierung im GBV

Im Unterschied zu dem in der Literatur beschriebenen üblichen Migrationsablauf, bei dem zu Beginn eine Migrationsstrategie festgelegt wird und im weiteren Verlauf eine Datenanalyse erfolgt, ist bei dem geplanten Transfer der lokalen Katalogisate vom LBS ins CBS die unterschiedliche Genese der Daten zu beachten; daher ist eine gründliche Analyse der inhaltlich heterogenen Ausgangsdaten Voraussetzung für die weitere Ausgestaltung des Migrationsverfahrens.[340] Problematisch ist dabei die große Anzahl der LBS-Bibliotheken: Es war unrealistisch, die Erhebung durch die Bibliotheken selbst vornehmen zu lassen, zumal die Werte aufgrund von Neukatalogisaten bzw. Löschungen nicht statisch sind. Zur Einschätzung des Datenvolumens beim Transfer der lokalen Katalogisate wurde daher ein Tool zur Bestandsaufnahme der aktuell vorhandenen lokalen Katalogisate entwickelt.

Das Perl-Programm `opc4_lok_anzahl.pl` ermittelt für jede Bibliothek die Anzahl der lokalen Katalogisate, getrennt nach Materialart, und schreibt die Werte in eine vorbereitete Excel-Tabelle. Als weitere Ausgabe erfolgt eine Protokollierung mit einer Zusammenfassung der ermittelten Werte pro Bibliothek sowie der Anzahl der dabei gefundenen Materialarten. Damit kann auf einfache Weise einer Bibliothek die nur für sie geltende Auswertung übermittelt werden, ohne jeweils die Excel-Datei mit der Gesamtauswertung versenden zu müssen.

Das Programm wird in unregelmäßigen Abständen ausgeführt und die Excel-Ausgabedatei anschließend im internen VZG-Wiki auf einer Seite des LOK-Projektes hinterlegt.[341]

Anlässlich eines Vortrags beim 2. LBS-Systemverwaltertreffen im September 2017 war eine erste umfassende Bestandsaufnahme erfolgt. Zum damaligen Zeitpunkt existierten lokale Katalogisate für 124 LBS-Bibliotheken. Berücksichtigt man nur Bibliotheken mit mehr als 2 L-Sätzen[342], bleiben 102 Bibliotheken mit zusammen mehr als 2,66 Mio. lokalen Katalogisaten. Aufgrund von zwischenzeitlich erfolgten Löschungen sind es derzeit 100 noch Bibliotheken mit mehr als 2,51 Mio. lokalen Katalogisaten (Stand: 02.05.2018).

Weitere Details zum Programm `opc4_lok_anzahl.pl` siehe Anhang 5.

[340] Entspricht Phase C (Datenanalyse) bei Lüssem/Harrach 2013, S. 3.
[341] Vgl. https://info.gbv.de/pages/viewpage.action?pageId=41557513#LOK-Umzug(intern)-StatistischeAuswertungderLokalenKatalogisateninallenLBS-Bibliotheken (15.05.2018).
[342] Unter der Annahme, dass es sich bei einer Menge von 1–2 lokalen Katalogisaten um Testdaten handelt.

5.2 ETL-Verfahren zur Datenanalyse und -bereinigung: Extraktion

Vor der Migration der lokalen Katalogisate ins CBS ist eine Bereinigung der Daten geboten. Ziel ist, die absolute Datenmenge und damit die Laufzeit für den Datentransfer so weit wie möglich zu reduzieren. Auch handelt es sich in vielen Fällen um obsolete Datensätze, die im Geschäftsprozess der nehmenden Fernleihe angelegt wurden oder aus anderen Gründen nur von temporärer Bedeutung sind. Angesichts der Menge der betroffenen Standorte und der zu migrierenden Datensätze kommt der Datenbereinigung eine zentrale Bedeutung im Migrationskonzept zu.

Die Entscheidung, welche Datensätze gelöscht werden können und welche migriert werden sollen, kann nur durch die Bibliotheken erfolgen. Als Entscheidungsgrundlage sollen den Bibliotheken die Metadaten der eigenen lokalen Katalogisate zur Verfügung gestellt werden, da der Zugang per Katalogisierungsclient keinen strukturierten Überblick erlaubt. Ebenfalls können daraus Erkenntnisse für die zukünftige Katalogisierungspraxis gewonnen werden: In vielen Fällen wurden lokale Katalogisate zur Lösung von Katalogisierungs- und Verbuchungsproblemen zweckentfremdet, obwohl es auch andere Lösungen gibt. Daher ist es erforderlich, den Bibliotheken die eigenen L-Sätze in einem angemessenen Format bereitzustellen, sinnvollerweise zeitlich vor dem geplanten Transfer und mit ausreichender Bearbeitungszeit.

Dafür ist ein eigener ETL-Prozess zu implementieren. Das Extrahieren von bibliografischen Metadaten aus dem LBS hat sich dabei als die zentrale Herausforderung erwiesen.

5.2.1 Bibliografische Metadaten im LBS

Lokale Katalogisate werden, wie alle bibliografischen Daten, in der LBS-Datenbank gespeichert. Sowohl CBS als auch LBS verwenden zur Speicherung ihrer Daten das relationale Datenbankmanagementsystem Adaptive Server Enterprise[343]. Gemäß der Struktur einer Katalogaufnahme verteilen sich die Inhalte der bibliografischen Metadaten auf mehrere in beiden Systemen gleichnamige Datenbanktabellen.

Datenbanktabelle[344]	**Inhalt**
`titles_global`	Felder aus Level 0 (Titelbezogene Kategorien)
`titles_local`	Felder aus Level 1 (Lokale Kategorien)
`titles_copy`	Felder auf Level 2 (Exemplar-Kategorien)
`title_overflow`	„Überlauftabelle“: Tabelle für die Aufnahme von Inhalt, der aufgrund der Speicherbeschränkung in den Tabellen `titles_global`, `titles_local` und `titles_copy` dort nicht aufgenommen werden kann

Tab. 7: LBS-Datenbanktabellen mit bibliografischen Metadaten

Die Speicherung aller Daten eines Katalogisats erfolgt in diesen Tabellen, daher würde es sich anbieten, diese Datenbanktabellen für das Extrahieren von L-Satz-Daten zu verwenden. Dementsprechend erfolgte zunächst eine Analyse des internen PICA+-Speicherformats für bibliografische Metadaten, dessen Aufbau in Anhang 3 anhand der LBS-Datenbanktabelle `titles_global` exemplarisch dargestellt ist.

[343] https://www.sap.com/germany/products/sybase-ase.html (15.05.2018); entwickelt von Sybase (seit 2010 Tochtergesellschaft von SAP).
[344] Die Angaben und Beispiele in dieser Arbeit beziehen sich auf LBS-Datenbanken.

Auf einzelne Kategorien kann in dieser Datenstruktur allerdings nicht direkt zugegriffen werden, da die Inhalte binär gespeichert sind. Für jeden Datensatz ist in der Tabelle `titles_global` ein Eintrag vorhanden, der aus aufeinanderfolgenden Abschnitten für die einzelnen PICA+-Kategorien besteht. Jeder Abschnitt teilt sich auf in einen Header mit fester Länge und dem variabel langen Kategorie-Inhalt, bestehend aus 1–n Subfields. Im Header ist die Gesamtlänge des Kategorieabschnitts codiert, ebenfalls die Kategoriebezeichnung. Da das interne Speicherformat ohne Feldende-Kennzeichen auskommt, muss für den Zugriff auf den Inhalt einer bestimmten Kategorie der Datensatz sequenziell gelesen werden. Über die Auswertung der Header-Inhalte ist es möglich, schrittweise zu der gewünschten Kategorie eines Datensatzes zu gelangen.[345] Der Aufwand einer Programmentwicklung zur Extraktion der einzelnen Kategorien aus der Datenstruktur in ein lesbares Format erschien allerdings zu hoch.

Auch die beiden LBS-Datenbanktabellen `acq_copy_cache` und `ous_copy_cache` kommen für eine Datenbereitstellung nicht in Frage, denn sie enthalten nur einige wenige zentrale, für Ausleih- und Erwerbungsfunktionen bedeutsame Kategorien als separate Attribute[346] und sind daher nicht ausreichend für eine Datenanalyse. Dies gilt insbesondere für Bibliotheken, deren lokale Katalogisate aufgrund ihres Einsatzzwecks jenseits der Standardfälle eine große Streuung der verwendeten Kategorien aufweisen.

Zur Auswertung der gespeicherten lokalen Katalogisate einer Bibliothek sind sämtliche Kategorien eines Datensatzes von Bedeutung. Da der Zugriff auf die binären Tabelleninhalte hinsichtlich Selektion als auch Parsing der Ergebnisse sehr aufwendig ist, wurden verschiedene alternative Zugriffsverfahren für LBS-Daten evaluiert.

5.2.2 Evaluation der Extraktionsverfahren

Ziel der Datenextraktion ist ein strukturierter Export der Katalog- und Bestandsdaten von lokalen Katalogisaten einer Bibliothek. Auf Grundlage dieser Daten können Bibliotheken entscheiden, welche Datensätze gelöscht werden können und in welchen Workflows weiterhin lokale Katalogisate benötigt werden.

Bei der Bewertung der in Frage kommenden Verfahren zur Datenextraktion sind mehrere Kriterien zu berücksichtigen. Vorrangig ist die Qualität der Datenausgabe entscheidend, vor allem die Vollständigkeit bzw. der Umfang der ausgegebenen Kategorien sowie eine generelle Fehlerfreiheit. Der Datenexport sollte möglichst in einem Datenformat erfolgen, mit dem die Bearbeiter vertraut sind (z. B. nicht in MARC 21, wenn die Katalogisierung gewöhnlich in PICA3 erfolgt). Hinsichtlich der Effizienz ist der Aufwand für die Erzeugung der Datenausgabe zu bewerten, auch im Hinblick auf die praktische Anwendbarkeit bzw. Zugänglichkeit, sowie die Performance des jeweiligen Verfahrens. Die Nutzbarkeit einzelner Verfahren könnte durch Programmänderungen bzw. Konfigurations-anpassungen verbessert bzw. hergestellt werden. Allerdings wären zusätzlich zur Inanspruchnahme von Mitarbeitern der VZG bzw. OCLC ggf. auch Aktivitäten seitens der LBS-Systemverwalter erforderlich. Stattdessen werden, sowie verfügbar und dokumentiert, alle potenziell nutzbaren Schnittstellen als Out-of-the-box-Lösung evaluiert.

[345] Detaillierte Beschreibung des PICA+-Speicherformats siehe Anhang 3.

[346] Insgesamt stehen neben PPN und EPN lediglich 13 unterschiedliche Elemente eines Katalogisats für den direkten Datenbankzugriff zur Verfügung. Die Datenfelder der Copy-Cache-Tabellen werden für die Präsentation im LBS4-Client und auch für die Druck- bzw. E-Mail-Ausgabe verwendet.

Für den Export von lokalen Katalogisaten kommen grundsätzlich verschiedene Verfahren in Frage, die im Folgenden jeweils anhand der oben genannten Kriterien untersucht und bewertet werden:

- WinIBW-Download aus LBS3/OWC und LBS4/CAT4,
- WinIBW-Funktion „Exceltabelle erstellen“ in LBS3/OWC und LBS4/CAT4,
- OPAC-Screenscraping,
- OPAC-Download,
- unAPI-Schnittstelle (OPAC),
- SRU-Schnittstelle (OPAC),
- CBS-Dienstprogramm `csft_ttlextract`,
- Eigenes Perl-Programm mit Nutzung der XML-Schnittstelle des OPACs.

Methode

Um die Qualität der Ausgabeformate der verschiedenen Schnittstellen vergleichen und bewerten zu können, wurde ein Referenzdatensatz verwendet (PPN 17226832X im Bestand der Jade Hochschule Elsfleth). Einige der untersuchten Verfahren sind für den Einsatz im CBS-Kontext vorgesehen, lassen sich aber grundsätzlich auch im LBS anwenden. Es wurde daher bewusst ein CBS-Datensatz anstatt eines lokalen Katalogisats ausgewählt, um eine Vergleichbarkeit der Funktionalität zu gewährleisten.

Als geeignete Ausgabeformate werden PICA3, PICA+ und MARC 21 in Betracht gezogen, da diese formalisiert und strukturiert sind und somit eine weitere Nachnutzung mit Hilfe von Tools für das Datenmanagement[347] ermöglichen. Beispiele für die Ausgabe der jeweiligen Datensatzexporte sind im Anhang dokumentiert, jeweils mit Angabe der genutzten Schnittstelle.

Beschreibung und Bewertung der Verfahren

WinIBW-Download

Im Katalogisierungsclient WinIBW für PICA-Datenbanken lassen sich Datensätze eines zuvor durch eine Recherche erzeugten Treffersets in eine Textdatei herunterladen. Unter den im GBV zur Verfügung stehenden Formaten sind für das LOK-Projekt die Formate PICA3, PICA+ und MARC 21 wegen der feldorientierten Ausgabe von Interesse.[348] In der folgenden Tabelle sind die Download-möglichkeiten aus OWC, CAT4 und (zum Vergleich) CBS zusammengestellt. Die gewählten Kriterien berücksichtigen das Ausgabeformat und vor allem die Zugriffsoptionen auf lokale Katalogisate. Ebenso sind die Recherchebedingungen von Belang, da sie wesentlichen Einfluss darauf haben, ob das gewünschte Trefferset (Gesamtmenge oder Teilmengen, auch für unterschiedliche Materialien) überhaupt in der benötigten Granularität zu bilden ist.

[347] Siehe Kap. 3.3.
[348] Weitere Downloadformate sind MAB2 und UNIMARC sowie Formate für den Katalogisierungsprozess.

Kriterium	CBS	OWC/LBS3	CAT4/LBS4
Ausgabeformat			
Format PICA3	Format D[349]	Format D	Format D
Format PICA+	Format P	Format P	Format P
Format MARC 21	Format USX	✘	✘
Zugriff auf lokale Katalogisate			
Zugriff auf L-Sätze	✘	✓	✓
Ausgabemenge			
keine Treffermengen-beschränkung	✓	✘ (max. 1000 Treffer)	✓
Recherchemöglichkeiten			
Anzahl Suchschlüssel	Level 0: 236[350] Level 1+2: 25	unterschiedlich je nach Standort: 13-23	je nach Standort: 114-182[351]
Suche über Materialart	✓	✘	✓
Suche über Selektionsschlüssel	✓	✘	✓
Suche über Sonderstandort (SST)	✓	unterschiedlich je nach Standort; generell: ✘ teilw. im Signatur-Index berücksichtigt	✓
Suchschlüssel für Ausleihindikator	✓	✘	✓
Vollständigkeit der Ausgabe			
Daten in Ausgabedatei	vollständig	Zeilen abgeschnitten KO-Kriterium	Zeilen abgeschnitten KO-Kriterium

Tab. 8: WinIBW-Download aus CBS, LBS3 und LBS4

Per WinIBW-Download können lokale Katalogisate nur im OWC bzw. CAT4 ausgegeben werden, in den Formaten PICA3 oder PICA+. Der Suchschlüssel MAK, mit dem eine differenzierte Auswertung der codierten Materialart möglich ist, steht dabei nur in CAT4 zur Verfügung. Fernleihexemplare sind überwiegend mit einem speziellen Sonderstandort gekennzeichnet; dieser ist im OWC grundsätzlich nicht recherchierbar.[352] Weitere Recherchekriterien, z. B. für funktional bedeutsame Angaben wie Ausleihindikator oder Selektionsschlüssel, können ebenfalls nur in CAT4 angewendet werden. Nicht zuletzt behindert die systemseitige Treffermengenbeschränkung in LBS3 auf 1000 Titel den Download lokaler Katalogisate. CAT4 unterliegt nicht dieser Mengenbeschränkung, allerdings werden hier, wie auch im OWC, in der Ausgabedatei Kategorie-Inhalte nicht vollständig ausgegeben, sondern am Zeilenende abgeschnitten.[353] Aus diesen Gründen ist ein WinIBW-Download aus LBS3 bzw. LBS4 nicht zur Ausgabe von Metadaten lokaler Katalogisate geeignet.

[349] Kürzel für CBS-Anzeige- und Downloadformat.
[350] Vgl. Übersicht der CBS-Suchschlüssel: Verbundzentrale des GBV 2016. Suchschlüssel für Normdaten sind nicht berücksichtigt.
[351] Für alle Satzarten (auch Normdaten).
[352] Abweichend von der Standardkonfiguration ist der Sonderstandort in einzelnen LBS-Standorten indexiert.
[353] Für den Beispieldatensatz ist dies bei Kategorie `4000` bzw. `021A` der Fall, siehe Anhang 6.

WinIBW-Funktion „Exceltabelle erstellen“[354]

Diese Funktion gibt für die Daten eines Treffersets eine frei wählbare Auswahl an Kategorien aus und speichert das Resultat in einer Tab-Separated-Value-Datei (TSV), die anschließend in eine Excel-Tabelle umgewandelt werden kann. Wie beim WinIBW-Download stellt das Bilden des Treffersets die größte Hürde dar. Zudem ist die Funktion „Exceltabelle erstellen“ zur Anwendung im CBS programmiert worden, daher kann eine Nutzung im LBS nicht verlässlich erwartet werden. Tatsächlich wird in LBS3 nur die PPN des ersten Titels im Trefferset ausgegeben, und dies für jeden Titel des Treffersets. Eine sinnvolle Nutzung der Funktion im OWC ist daher ausgeschlossen. Bei Anwendung in CAT4 werden lediglich PPN und EPN ausgegeben.[355]

Eine genauere Analyse des in JavaScript vorliegenden Programmcodes könnte u. U. zu einer Verbesserung der Funktionalität im LBS führen, konnte aber aufgrund des Aufwandes im Rahmen dieser Arbeit nicht weiter verfolgt werden.

Allerdings kann die PPN-/EPN-Ausgabe in weiteren Programmen nachgenutzt werden, da CAT4 eine präzise Eingrenzung der gewünschten Datensätze aufgrund der umfangreichen Recherchemöglichkeiten erlaubt. Der Erfolg solcher Recherchen zur Bildung eines geeigneten Treffersets ist jedoch abhängig davon, wie konsistent die Bibliotheken Datensätze für lokale Materialien erfassen.

Eine Möglichkeit der Nachnutzung solcher Excel-Dateien mit PPNs ist das WinIBW-Script `LoksatzLoeschen.vbs` zum Löschen lokaler Katalogisate (siehe Kap. 5.4.2).

Zugriff auf lokale Katalogisate per LBS-OPAC

Eine Alternative zum Zugriff über den Client WinIBW stellt der OPAC der LBS-Systeme dar. Dort stehen grundsätzlich alle bibliografischen Informationen für die Anzeige zur Verfügung. Eine Besonderheit der OPAC-Ausgabe ist die zusätzlich enthaltene (virtuelle) PICA+-Kategorie 201@, deren Kategorie-Inhalt aus der Ausleihschnittstelle des OPACs gespeist wird.[356]

Beispiel 1: Lokales Katalogisat (ausgeliehen) mit Anzeige im Publikumskatalog:

```
201@/01 $aAusleihbar|derzeit ausgeliehen$b2$d24-04-
2018$e992031036$mmon$f1$lhttp://opac.lbs-
rostock.gbv.de:80/loan/RES?EPN=992031036&MTR=mon&BES=1$uAusleihbar$vderzeit
ausgeliehen
```

Beispiel 2: Lokales Katalogisat (ausgeliehen) mit Nichtanzeige-Code (Selektionsschlüssel d):

```
201@/01 $aPraesenzbestand|noch nicht verfuegbar$b0$d03-05-
2018$e970006772$mmon$f11$uPraesenzbestand$vnoch nicht verfuegbar
```

[354] Vgl. WinIBW-Handbuch https://verbundwiki.gbv.de/display/VZG/Excel-Tabelle+erstellen (15.05.2018).
[355] Vgl. Ausgabedatei in Anhang 7.
[356] Die Kategorie 201@ ist seitens OCLC nicht dokumentiert, die Bedeutung der Angaben in den Subfields konnte allerdings empirisch ermittelt werden. Dies war Voraussetzung für die Auswertung der Kategorie durch die *Document Availability Information API* (DAIA), einer Schnittstelle zur Verfügbarkeitsanzeige in Bibliothekskatalogen, vgl. https://verbundwiki.gbv.de/display/VZG/DAIA (15.05.2018). Eine Beschreibung von Kategorie 201@ ist im internen Wiki der VZG vorhanden.

Für ausgeliehene Exemplare mit Nichtanzeige-Code (Beispiel 2) wird anstelle des tatsächlichen Ausleihstatus „derzeit ausgeliehen“ die Information „noch nicht verfuegbar“ ausgegeben. Auch der Code in Subfield $b mit Wert „0“ entspricht nicht dem realen Ausleihstatus. Allerdings ist das Leihfristende in Subfield $d angegeben; dies ist also ein verlässliches Kennzeichen auch bei denjenigen lokalen Katalogisaten, die nicht für die Anzeige im OPAC vorgesehen sind. Bei allen OPAC-bezogenen Zugriffsverfahren stehen somit auch aktuelle Ausleihinformationen zur Verfügung.

Das Präsentationsformat des LBS-OPACs wird über einen URL-Parameter gesteuert, als Standard ist eine gefelderte Darstellung mit verbalen Labels für die Katalogdaten hinterlegt. Es ist aber auch die Anzeige der Katalogdaten im PICA3-Format möglich. Da jeder Datensatz im LBS-OPAC über den sog. Zitierlink eindeutig referenziert werden kann, lassen sich lokale Katalogisate einer Bibliothek einzeln im PICA3-Format aufrufen, wenn der Identifier (PPN) bekannt ist.

OPAC: Screenscraping

Um die Metadaten aller L-Sätze oder einer Teilmenge zu erhalten, wird eine Liste der gewünschten PPNs benötigt. Eine solche Liste kann mit Hilfe der WinIBW-Funktion „Exceltabelle erstellen“ (s. o.) erzeugt werden. Zur weiteren Verwendung der angezeigten Metadaten ist ein Screenscraping der HTML-Katalogseite notwendig, bei dem die benötigten Daten aus dem HTML-Code der Seite extrahiert werden. Ein solches Verfahren ist allerdings anfällig für Änderungen an den HTML-Templates des zugrundeliegenden OPACs, daher wurde diese Idee nicht weiter verfolgt.

OPAC-Download

Eine Download-Funktion ist für LBS-OPACs ebenfalls implementiert. Welche Download-Formate angeboten werden, ist je nach OPAC-Konfiguration der LBS-Installation unterschiedlich, da hierfür der jeweilige Systemverwalter verantwortlich ist. So werden in den OPACs des Standorts Emden lediglich die Formate *Vollanzeige* und *Kurzanzeige* angeboten. Das Format *Vollanzeige* ist identisch mit der feldorientierten Titelpräsentation im OPAC, die *Kurzanzeige* entspricht dem ISBD-Format für Titel, also ohne Lokal- bzw. Exemplarkategorien. In den meisten LBS-Standorten sind weitere Formate für den Download konfiguriert (jeweils eine Teilmenge der im Folgenden aufgeführten Formate):

- Literaturverwaltung: Citavi, Reference Manager (RIS)
- Literaturverwaltung: Endnote
- Literaturverwaltung: BibTeX
- Bibliographisches Format: PICA3

Das PICA3-Format kann bei ca. 40% aller LBS-Standorte für den OPAC-Download genutzt werden. Die Ausgabe der Literaturverwaltungsformate ist ebenfalls feldorientiert, allerdings auf die Nutzung in Literaturverwaltungsprogrammen zugeschnitten; es fehlen daher insbesondere exemplarspezifische Informationen, die für eine umfassende Datenanalyse erforderlich wären.[357] In einigen LBS-Standorten haben die LBS-Systemverwalter zur Abdeckung bibliothekspezifischer Bedürfnisse Anpassungen an den Standardformaten vorgenommen oder eigene Formate entwickelt.

[357] Datumsfelder, Ausleihindikator, Selektionsschlüssel u. a. m.

Eine weitere Einschränkung der Nutzbarkeit des OPAC-Downloads ist die Begrenzung des „Download-Maximums“, also der Maximalzahl herunterladbarer Titelsätze pro Downloadvorgang. Unabhängig von der auf der OPAC-Webseite[358] angegebenen Grenze von 500 bzw. 1000 (bei wenigen Standorten) liegt die tatsächliche Grenze bei 500; die Treffer 1-1000 können dennoch en bloc heruntergeladen werden, alle weiteren in Blöcken zu je 500 Titeln. Ein Download der lokalen Katalogisate per OPAC ist demnach nur sinnvoll, wenn die Anzahl der L-Sätze für eine Bibliothek nicht größer als 1000 ist. Bei einer größeren Anzahl lokaler Katalogisate müsste das Herunterladen in Blöcken von je 500 Titeln erfolgen.

Da das PICA3-Format bei mehr als der Hälfte aller LBS-Standorte nachträglich als Download-Option eingerichtet werden müsste, kommt ein Download per OPAC als allgemeingültiges Verfahren für den Export lokaler Katalogisate nicht in Frage. Zudem behindert das Download-Maximum einen vertretbaren Arbeitsablauf.[359]

OPAC: unAPI-Schnittstelle

Mit unAPI[360], einem REST-basierten Webservice, lassen sich einzelne Datensätze in verschiedenen Datenformaten abrufen; für die Nutzung mit PSI-Datenbanken sind u. a. die Formate PICA+, PICA XML, MARC 21 und MARCXML konfiguriert. Die Schnittstelle liefert jeweils einen einzelnen, per Identifier referenzierten, Datensatz aus, eine Suche über den Datenbestand ist nicht möglich.[361] Für den Abruf eines Datensatzes sind ein Datenbankkürzel (*unAPI-key*), die PPN und das Format anzugeben. Beim folgenden Beispiel für den unAPI-Abruf[362] mit Ausgabe im PICA+-Format handelt es sich um ein lokales Katalogisat der UB Braunschweig:

```
http://unapi.gbv.de/?id=owc-de-84:ppn:981559603&format=pp
```

Zu beachten ist, dass in der PICA+-Ausgabe Unterfelder mit dem einleitenden Steuerzeichen „$“ (Dollar) gekennzeichnet sind; daher wird ein $-Zeichen als Bestandteil des Kategorie-Inhalts bei der Ausgabe verdoppelt. Dies betrifft die meisten Bibliotheken mit LBS-Ausleihmodul, da Verbuchungsnummern im GBV häufig ein $-Zeichen enthalten: `209G/01 $a897E$$087491`. Bei der Weiterverarbeitung von PICA+-Ausgaben der unAPI-Schnittstelle muss dies entsprechend berücksichtigt werden.

Die Ausgabe in den Formaten PICA+ und PICA XML umfasst sämtliche Kategorien eines Datensatzes, daher ist ein unAPI-Abruf grundsätzlich für den Export von lokalen Katalogisaten geeignet. Da jeweils nur ein einzelner Datensatz abgerufen werden kann, ist das vorherige Erstellen einer Liste der gewünschten PPNs Voraussetzung. Eine solche Liste lässt sich mit Hilfe der WinIBW-Funktion „Exceltabelle erstellen“ (s. o.) erzeugen.

Die Nutzung der unAPI-Schnittstelle muss für jeden OWC-Katalog einzeln konfiguriert werden; zum untersuchten Zeitpunkt fehlten die Konfigurationseinträge bei knapp 100 LBS-Bibliotheken, so dass ein flächendeckender Einsatz der unAPI-Schnittstelle für das Projekt nicht realistisch umzusetzen war.

[358] Abbildung der Download-Seite im OPAC siehe Anhang 8.
[359] Daher wird auf Beispiele für die Darstellung der Ausgabeformate verzichtet.
[360] Spezifikation s. https://web.archive.org/web/20150117162606/http://unapi.info/specs/ (15.05.2018).
[361] Vgl. https://verbundwiki.gbv.de/display/VZG/unAPI (15.05.2018).
[362] Die Ausgabe dieses unAPI-Abrufs sowie weitere in den Formaten PICA XML, MARC 21 und MARCXML siehe Anhang 9.

OPAC: SRU-Schnittstelle[363]

Search/Retrieve via URL (SRU) ist ein HTTP-basiertes Protokoll zur Abfrage bibliografischer Datenbanken mit Ausgabe in einem XML-basierten Format. Die SRU-Schnittstelle für PICA-Kataloge verwendet die Katalogzugänge der unAPI-Konfiguration, sodass auch hier aktuell lediglich zwei OPACs für den Abruf lokaler Katalogisate genutzt werden können. Im Unterschied zu unAPI können per SRU alle konfigurierten Suchschlüssel genutzt und auch mehrere Titel über eine Anfrage abgerufen werden. Bei PICA-Katalogen sind alternativ zu den verbalen Suchschlüsseln auch die entsprechenden numerischen sog. IKTs nutzbar. Die in einer OPAC-Installation definierten IKTs lassen sich über die XML-Schnittstelle des OPACs ermitteln.[364] Für den Abruf von Datensätzen sind der bibliotheksspezifische *unAPI-key*, die Suchkriterien, die Maximalzahl auszugebender Datensätze und das Format anzugeben. Die folgende Beispiel-Abfrage exportiert maximal 200 Braunschweiger Fernleihtitel (Suche mit `mak luy`) im Format PICA XML:

```
http://sru.gbv.de/owc-de-84?version=1.1&operation=searchRetrieve&query=pica.mak%3Dluy&maximumRecords=200&recordSchema=picaxml
```

Anstelle des Suchschlüssels `mak` könnte auch die entsprechende IKT-Nummer `8600` angegeben werden. Da der URL-Parameter `maximumRecords` obligatorisch ist, müsste für den Abruf von Metadaten lokaler Katalogisate einer Bibliothek vorher die (ungefähre) Anzahl ermittelt werden. Ob alternativ eine pauschal gesetzte 6-stellige Zahl, entsprechend der größten Menge L-Sätze einer Bibliothek, für einen SRU-Abruf praktikabel ist, wurde nicht getestet. Für jede Bibliothek kann die Anzahl ihrer lokalen Katalogisate der Ausgabedatei des Perl-Programms `opc4_lok_anzahl.pl` entnommen werden.[365]

Auch mit der SRU-Schnittstelle können Metadaten von lokalen Katalogisaten im PICA+- bzw. PICA-XML-Format exportiert werden. Es gilt allerdings dieselbe Einschränkung wie bei unAPI: Die SRU-Schnittstelle wäre für den flächendeckenden Einsatz nur geeignet, wenn für alle LBS-Bibliotheken auch die betreffenden OWC-Kataloge im VZG-internen Datenbankverzeichnis hinterlegt würden.

Programm csft_ttlextract[366]

Mit dem CBS-Dienstprogramm `csft_ttlextract` werden Datensätze direkt aus einer PICA-Datenbank gelesen und im PICA+-Format ausgegeben. Datensätze können anhand einer Liste von PPNs oder IPNs[367] oder eines IPN-Bereichs ausgewählt und nach ILN gefiltert werden. Da das interne Speicherformat von CBS und LBS für bibliografische Metadaten identisch ist[368], kann das Programm ebenso für das Extrahieren von LBS-Daten genutzt werden. Hierzu ist grundsätzlich die Installation auf dem Server des jeweiligen LBS-Standorts erforderlich.

Das Programm `csft_ttlextract` liefert Datensätze im PICA+-Format zurück.[369] Daher ist es für das Extrahieren von bibliografischen Metadaten aus dem LBS sehr gut geeignet, zumal auch für große

[363] Vgl. https://verbundwiki.gbv.de/display/VZG/SRU (15.05.2018).
[364] Für den OWC-Katalog der UB Braunschweig: http://lhbrs2.gbv.de:8080/DB=OWC.1//XML=1.0/IKTLIST (15.05.2018), vgl. Abschnitt *OPAC: XML-Schnittstelle* in diesem Kapitel.
[365] Siehe Anhang 5 sowie beiliegende CD.
[366] Vgl. Sutherland 2010, S. 5–6.
[367] Eine IPN entspricht der PPN ohne Prüfziffer. In der LBS-Datenbank sind IPNs gespeichert, als Attribut ppn.
[368] Vgl. Anhang 3.
[369] Siehe Beispiel in Kap. 5.5.2, Abschnitt *Schritt 2*.

Datenmengen nur eine sehr kurze Programmlaufzeit benötigt wird.[370] Erforderlich ist allerdings die vorherige Erstellung einer Liste der betreffenden PPNs bzw. IPNs.

Im Rahmen des Transfer-Projekts wird für die Extraktion der lokalen Katalogisate aus der LBS-Datenbank das Programm `csft_ttlextract` eingesetzt. Die Installation erfolgte allerdings lediglich auf einem einzigen LBS-Testserver der VZG, der Zugriff auf die jeweilige Datenbank der 30 LBS-Installationen wird für den Produktivbetrieb über die Anpassung der Sybase-Umgebungsvariablen DSQUERY[371] realisiert, somit werden die Datenbankzugriffe auf den hinterlegten Datenbankserver umgeleitet.

Die installierte Programminstanz wird kontinuierlich vom Projektteam zur Implementierung des ETL-Prozesses und später für den produktiven Datentransfer genutzt, sie steht daher für Datenexporte zwecks Bereitstellung von Dateien zu Analysezwecken nicht zur Verfügung. Alternativ hätte ggf. ein weiterer LBS-Server aufgesetzt werden müssen einschließlich kompletter Sybase- und LBS-Installation, der Aufwand hierfür ist allerdings unverhältnismäßig hoch.

OPAC: XML-Schnittstelle[372]

Als PSI-Datenbank verfügen LBS-OPACs über eine proprietäre sog. XML-Schnittstelle. Entsprechend den per OPAC-Webzugriff ausführbaren Funktionen sind für die XML-Schnittstelle zusätzlich zur Titelrecherche weitere Funktionsaufrufe implementiert. Diese sind als HTTP-GET-Request in der URL mit den entsprechenden Parametern anzugeben, die Ausgabe erfolgt im XML-Format.

Die XML-Schnittstelle bedient u. a. diese Anfragen:

- Ausgabe der konfigurierten Suchschlüssel mit ihrer IKT-Entsprechung,
- Suche mit IKT und Suchbegriff(en),
- Suche mit Inhalt von 70xx = Datum(sbereich) und Selektionsschlüssel,
- Ausgabe aller IPNs eines Treffersets,
- Ausgabe eines Titels aus einem Trefferset (über laufende Nummer im Trefferset spezifiziert),
- Ausgabe eines Titels mit PPN / IPN,
- Ausgabe der Indexliste für eine IKT (entspricht CBS-Kommando *scan*),
- Ausgabe der Suchgeschichte.

Bei der Suche per IKT und Suchbegriff(en) kann zwischen zwei Ausgabeformaten gewählt werden:

Parameter in der URL	XML-Attribut „`format`“	Formatbezeichnung in der Dokumentation	Inhalt
`&PLAIN=ON`	`extpp`	*external PICA+ format*	in XML verpacktes binäres bzw. normalisiertes PICA+-Format (lesbarer Text mit binären Steuerzeichen)
---	`text`	*textual presentation*	in XML verpackte HTML-Darstellung der OPAC-Präsentation

Tab. 9: Ausgabeformate der PICA-XML-Schnittstelle[373]

[370] Bisher umfangreichster Test im LOK-Projekt mit 43.000 lokalen Katalogisaten in 67 Sekunden.
[371] Hier wird der jeweilige Datenbankname hinterlegt.
[372] Vgl. OCLC 2011.
[373] Beispiele hierzu siehe Anhang 10.

Im Gegensatz zu den bisher vorgestellten Verfahren, die nicht oder nur eingeschränkt praktisch nutzbar sind, lassen sich unter Verwendung der XML-Schnittstelle des LBS-OPACs die lokalen Katalogisate so granular extrahieren, dass im Vorfeld des eigentlichen Transfers ins CBS die Mitarbeiter der LBS-Bibliotheken selbst Datenbereinigungen und Analysen zur Vorbereitung der Migration vornehmen können.

Vorteilhaft ist ebenfalls, dass die XML-Schnittstelle uneingeschränkt nutzbar ist, während ein direkter Serverzugriff an einem der LBS-Standorte aufgrund der extremen Sicherheitsvorkehrungen des lokalen Systemverwalters nur für einzelne VZG-Mitarbeiter realisiert ist.

Durchführung: Extraktion von bibliografischen Metadaten lokaler Katalogisate

Aufgrund der vollständigen Datenausgabe und der uneingeschränkten Zugänglichkeit der LBS-OPACs wurde die XML-Schnittstelle des OPACs für die Extraktion der lokalen Katalogisate ausgewählt und zu diesem Zweck das Perl-Programm `opc4_lok_titel.pl` entwickelt. Als Client-Lösung ist dieses Verfahren unabhängig von einer Serverinstallation und somit auch für LBS-Systemverwalter nachnutzbar.

Der Zugriff auf die OPACs aller LBS-Bibliotheken ist jederzeit ohne weitere Absprachen möglich, allerdings muss bei der Ausführung des Perl-Programms berücksichtigt werden, dass i. d. R. sonntags gegen 23:00 Uhr eine Neuindexierung der OPAC-Daten stattfindet und in diesem Zeitraum die Programmausgabe fehlerhaft bzw. unvollständig ist.

In der Ausgabedatei sind die Daten im PICA+-Format gespeichert. Die Protokolldatei enthält auch eine Kategorien-Statistik.

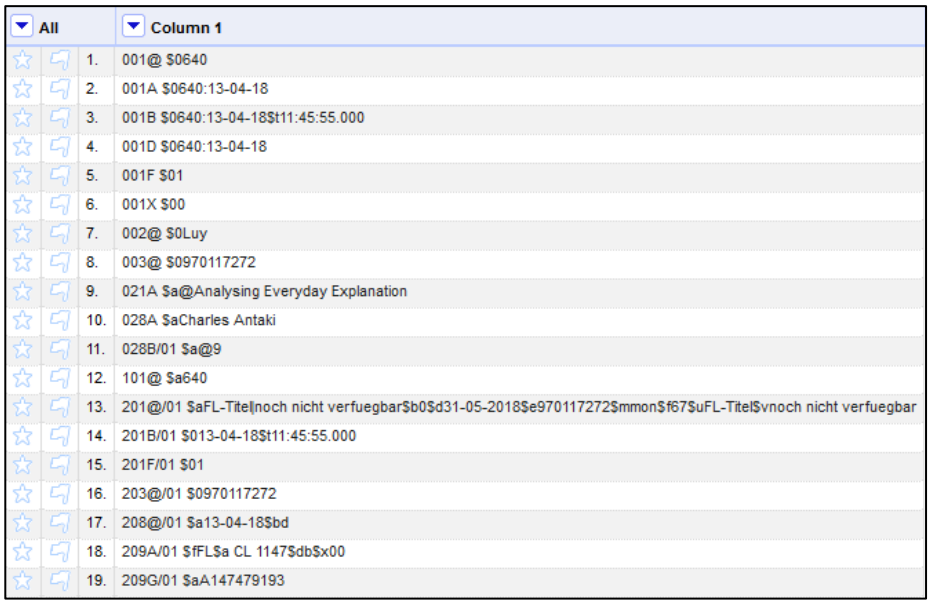
Weitere Details zum Programm `opc4_lok_titel.pl` siehe Anhang 12.

5.3 ETL-Verfahren zur Datenanalyse und -bereinigung: Transformation

Das Perl-Programm `opc4_lok_titel.pl` gibt die Metadaten der lokalen Katalogisate im textbasierten PICA+-Format aus. Eine solche Datei ist zur Durchsicht im Hinblick auf potenzielle Datensatzkorrekturen und -löschungen nicht geeignet, da die Kategorien der Datensätze sequenziell ausgegeben werden und eine Analyse der Daten, z.B. durch Gruppieren oder Filtern ähnlicher Datensätze, für die Zielgruppe der Bearbeiter in den Bibliotheken[374] in diesem Format nicht möglich ist. Daher wurde zur Verbesserung der Handhabung eine Lösung zur weiteren Strukturierung der Daten in Tabellenform benötigt. Da die CSV-Ausgabe von Catmandu wie gezeigt[375] für diese Zwecke nicht geeignet ist, fiel die Wahl auf OpenRefine.

Mittels OpenRefine erfolgt die Verarbeitung der PICA+-Textausgabe mit dem Ziel einer für Bibliotheksmitarbeiter geeigneten Darstellungsform; die Transformationen dienen lediglich zur Aufbereitung der Daten für die Analyse. Ein direkter Re-Import von Datenkorrekturen aus OpenRefine ins LBS ist aus systemtechnischen Gründen nicht möglich.[376]

Zunächst bestand die Herausforderung darin, Daten im PICA+-Format in ein OpenRefine-Projekt zu importieren. In der Datei folgen die Datensätze sequenziell aufeinander, mit einer Leerzeile voneinander abgesetzt; jede Kategorie beginnt auf neuer Zeile. Diese Struktur musste in die tabellenorientierte Struktur von OpenRefine umsetzt werden.[377] Für jede Kategorie soll eine Spalte erzeugt werden, mit der Kategoriebezeichnung als Spaltenüberschrift, und für jeden Datensatz soll eine Tabellenzeile angelegt werden. Als Format der Quelldatei wird daher *Line-based text files* gewählt, dies wird von OpenRefine bereits automatisch erkannt. Zwar kann für den Import in OpenRefine aus einer großen Anzahl Zeichensätze gewählt werden, der proprietäre Pica-Zeichensatz ist jedoch nicht darunter. Daher wurden mit verschiedenen Zeichensätzen experimentiert und ISO 8859-1 für am besten geeignet befunden.



All		Column 1
	1.	001@ $0640
	2.	001A $0640:13-04-18
	3.	001B $0640:13-04-18$t11:45:55.000
	4.	001D $0640:13-04-18
	5.	001F $01
	6.	001X $00
	7.	002@ $0Luy
	8.	003@ $0970117272
	9.	021A $a@Analysing Everyday Explanation
	10.	028A $aCharles Antaki
	11.	028B/01 $a@9
	12.	101@ $a640
	13.	201@/01 $aFL-Titel\|noch nicht verfuegbar$b0$d31-05-2018$e970117272$mmon$f67$uFL-Titel$vnoch nicht verfuegbar
	14.	201B/01 $013-04-18$t11:45:55.000
	15.	201F/01 $01
	16.	203@/01 $0970117272
	17.	208@/01 $a13-04-18$bd
	18.	209A/01 fFLa CL 1147dbx00
	19.	209G/01 $aA147479193

Abb. 11: OpenRefine: Datensatz vor der Transformation

[374] Zur Datenanalyse ist Fach- und Domänenwissen erforderlich, vgl. Lüssem/Harrach 2013, S. 3.
[375] Siehe Kap. 3.3.1.
[376] Ggf. ließe sich hierfür ein zu entwickelndes WinIBW-Script nutzen.
[377] Hilfreich war hier eine Anleitung für Daten in MARC 21, vgl. Lohmeier 2017, S. 80–84.

Das durch den Import in OpenRefine erzeugte Projekt enthält die Zeilen der Quelldatei in einer einzigen Spalte (s. Abb. 11). Zur Umsetzung in die Tabellenstruktur sind zahlreiche Transformationsschritte erforderlich. Zunächst muss die Kategoriebezeichnung abgetrennt und in einer separaten Spalte gespeichert werden. Nun kann mit Hilfe der *columnize*-Funktion die Umsetzung zur Darstellung mit einer Spalte pro Kategorie erfolgen.

OpenRefine unterscheidet in der Darstellung zwischen Zeilen (*rows*) und Datensätzen (*records*). Im besten Fall sind die Zahlen von *records* und *rows* identisch, dann existieren in der Quelldatei keine mehrfach in einem Datensatz vorhandenen PICA+-Kategorien. Ist aber eine Kategorie doppelt vorhandenen, werden für einen Datensatz zwei Zeilen ausgegeben (s. Abb. 12). Diese strukturelle Ungleichheit muss aufgelöst werden, bevor die Datei für die Weitergabe im Excel-Format aus OpenRefine exportiert wird. Diffizil ist dabei die unterschiedliche Systematik der Repräsentation von wiederholbaren Kategorien in den Formaten PICA3 und PICA+. Je nach Fall werden die Inhalte in einer Spalte zusammengeführt oder in separaten Spalten abgelegt, hierzu sind weitere Transformationen nötig. Bei der Kategoriengruppe Signaturen beispielsweise entsprechen identische PICA+-Kategorien unterschiedlichen PICA3-Kategorien; die Unterscheidung erfolgt über das Subfield $x.[378]

All		PPN 0100	MAK 0500	Titel 4000	EPN 7800	Datum 7001	SEL 7001	209A/01
1.		969653786	Luy	Chron's Disease IFERNLEIHE	969653417	17-11-17	d	fFLa A145601706dsx00
								$a2015 A 3243 /FERNLEIHE$x01
2.		96965376X	Luy	Das Belvedere in Wien IFERNLEIHE	969653395	17-11-17	d	fFLa A145621456dsx00
								$aK 7.101 Wien /FERNLEIHE$x01
3.		969653751	Luy	Krebs im Kontext IFERNLEIHE	969653387	17-11-17	d	fFLa A14561557Xdsx00
								$aCW 6800 W798 /FERNLEIHE$x01

Abb. 12: OpenRefine: Datensätze mit mehrfach vorhandener Kategorie 209A/01 (Ausschnitt)[379]

Damit OpenRefine Datensätze (im Unterschied zu Zeilen) in der Importdatei korrekt erkennen kann, muss die jeweils erste Zeile eines Datensatzes ein Identifikationsmerkmal enthalten. Bei MARC-Daten ist dies der Leader, in anderen Fällen der Identifier des Datensatzes. Als erste Kategorie der PICA+-Ausgabe findet sich jedoch in den meisten Datensätzen – aber nicht immer – die PICA+-Kategorie `001@`. Daraus resultiert das Problem, dass OpenRefine den Beginn eines Datensatzes nicht zweifelsfrei ermitteln kann und es in der Folge zu falschen Zuordnungen von Kategorien zu Datensätzen kommt, falls mehrfach in einem Datensatz vorkommenden Kategorien in der Datenmenge enthalten sind. Daher wurde das Perl-Programm `opc4_lok_titel.pl` zur Erzeugung der OpenRefine-Importdatei dahingehend angepasst, dass auch vor dem ersten Datensatz eine Leerzeile ausgegeben wird. Zur Lösung des genannten Problems wird nun vor den eigentlichen Transformationen ein beliebiger Inhalt in alle Leerzeilen eingetragen, diese Pseudo-Kategorie kann nach Abschluss der Transformationen wieder gelöscht werden. Nun werden die eigentlichen Transformationen durchgeführt. Bei einigen Kategorien ist es sinnvoll, die Inhalte der enthaltenen Subfields in jeweils eine eigene Spalte umzusetzen, damit man die Daten auch über diese Elemente filtern kann; dies gilt insbesondere für Kategorie `209A` mit Sonderstandort, Signatur und

[378] Die Problematik wird in Anhang 2 durch Beispiele illustriert.
[379] Zur besseren Lesbarkeit wurden bereits einige Transformationsschritte durchgeführt, ebenfalls wurden Spalten umsortiert.

Ausleihindikator. Die PICA+-Kategoriecodes als Spaltenüberschrift werden für relevante Kategorien durch geeignete Bezeichner („PPN 0100“, „SGN 7100“ usw.) ersetzt, da den späteren Empfängern der Datei die PICA+-Benennungen im Allgemeinen nicht bekannt sind. Bei diesen Spalten wird auch jeweils das Subfieldkennzeichen (z. B. $a) entfernt, damit nur die eigentlichen Kategorie-Inhalte bestehen bleiben. Spalten, die für die Bibliotheken unerheblich sind (z. B. ILN), werden entfernt. Schließlich wird die Spalte der Dummy-Kategorie gelöscht und eine Umsortierung der verbleibenden Spalten vorgenommen, damit die Reihenfolge in etwa der PICA3-Anzeige entspricht.

Geplant war die Erstellung grundlegender Transformationsregeln, die auf beliebige Datenmengen angewendet werden können. Wegen der großen Spannbreite der genutzten Kategorien mit Mehrfachvorkommen hat sich allerdings in der Praxis gezeigt, dass für diese Umsetzungen weitere individuelle Transformationen für die jeweilige Datensituation durchgeführt werden müssen. Jedoch kann bei allen Bibliotheken zur Transformation von einheitlich genutzten Kategorien wie PPN, Titel oder Verbuchungsnummer eine Reihe Standard-Transformationsregeln angewendet werden. Die Transformationsschritte für den Standardablauf sind in Anhang 11 aufgeführt, das Ergebnis zeigt Abb. 13.

All		PPN 0100	MAK 0500	PER 3000 Name	PER 3000 Vornar	TIT 4000	EPN 7800	DTM 7001	SEL 7001	EXK 4801	SST 7100	SGN 7100	AUI 7100
1.		969682689	Lab			FL-Test	96968231X	21-02-18	d	Nur eine Verlängerung	FL	1234-1234	d
2.		96967922X	Luy		Bremen$c46	Die missionarische Gesellschaft IFERNLEIHE	969678851	09-02-18	d		FL	A146646533	s
3.		969679211	Luy		Braunschweig$c834	Atlas oder die unruhige fröhliche Wissenschaft IFERNLEIHE	969678843	09-02-18	d		FL	A146646517	s
4.		969679203	Luy		Bremen$c46	Biosemoitica IFERNLEIHE	969678835	09-02-18	d		FL	A14660640X	s
5.		969679181	Luy		Bremen$c46	Führungspraxis in Forscung und Lehre IFERNLEIHE	969678819	09-02-18	d		FL	A146651030	s
6.		969679173	Luy		Göttingen$c7	An introduction to predictive maintenance IFERNLEIHE	969678800	09-02-18	d		FL	A146376293	s
7.		969679165	Luy		Göttingen$c7	Virtuelle Teams IFERNLEIHE	969678797	09-02-18	d		FL	A146594592	s

Abb. 13: OpenRefine: Datensätze nach Anwendung der Standard-Transformation (Ausschnitt)

Zum Abschluss werden die Daten im Excel-Format (`.xlsx`) exportiert und der Bibliothek übermittelt. Damit erhalten Bibliotheksmitarbeiter eine komfortable Möglichkeit zur Sichtung und Analyse ihrer lokalen Katalogisate.

Es wurden zahlreiche Durchläufe von Datenexport und -transformation durchgeführt, in deren Verlauf die Transformationsschritte immer wieder geändert bzw. optimiert wurden, mit dem Ziel einer für die Bibliotheken geeigneten Strukturierung bzw. Darstellung der Daten in der Excel-Exportdatei. Dabei stellte sich heraus, dass es sinnvoll ist, die Daten getrennt nach Materialart bzw. Medientyp zu selektieren und zu verarbeiten, sofern diese über formale Kriterien erkennbar sind. So ist beispielsweise der Aufbau der lokalen Katalogisate von Fernleihtiteln bei allen Bibliotheken vergleichbar, insbesondere die Menge und Auswahl der verwendeten Kategorien hält sich in einem überschaubaren Rahmen. Da besonders Fernleihdatensätze zu den potenziellen Löschkandidaten gehören, wurde der beschriebene Ablauf vorrangig für Bibliotheken praktiziert, deren lokale Katalogisate überwiegend aus Fernleihtiteln bestehen oder aber deren FL-Titel über die verwendete Materialart eindeutig identifizierbar sind.

5.4 Datenbereinigung im LBS

Im Vorfeld des Transfers von lokalen Katalogisaten ins CBS ist es sinnvoll, wenn sich die Bibliotheken einen Überblick über die bisher erfassten Daten verschaffen. Hierzu wird jeder Bibliothek bei Bedarf eine per OpenRefine aufbereitete Datei im Excel-Format zur Verfügung gestellt werden.[380] Mit dem entsprechenden Domänenwissen können Bibliotheksmitarbeiter Aussagen über den Zweck der jeweiligen Datensätze bzw. Datensatz-Gruppen treffen, Datenfehler identifizieren und Konzepte für ggf. erforderliche Datenbearbeitungen erarbeiten.[381]

Eine Offline-Bereinigung von Daten nach dem Vorbild der im CBS für VZG-Mitarbeiter zur Verfügung stehenden Reparaturprogramme[382] ist im LBS technisch nicht möglich. Das im CBS durch Bibliothekare selbst ausführbare WinIBW-Script „sucheErsetze“ funktioniert weder im OWC noch in CAT4. Systematische Korrekturen struktureller Art werden nach dem Abschluss des Transferverfahrens für die Daten einzelner Bibliothek im CBS durchgeführt werden. Daher konzentrieren sich die geplanten programmgestützten Bereinigungen auf das Löschen von Datensätzen, insbesondere von Fernleihtiteln.

5.4.1 Kontrolle und Korrektur lokaler Katalogisate

Einige Problemfelder hinsichtlich der Datenqualität sind genereller Natur und betreffen die meisten Bibliotheken mit lokalen Katalogisaten bzw. können unabhängig vom Dateninhalt formal erkannt werden. Dazu gehören die folgenden Sachverhalte:

Hierarchische Verknüpfungen zu anderen Titeln

Entgegen der Vorgaben der Katalogisierungsrichtlinie[383] wurden in einigen Bibliotheken lokale Katalogisate mit Titelverknüpfungen angelegt, v. a. zu einer Zeitschriftenaufnahme.[384] Teilweise sind die Datensätze mit einem CBS-Katalogisat verknüpft, in anderen Fällen mit einem anderen lokalen Katalogisat. Verknüpfungen zu einem CBS-Titel sind unproblematisch, da die Relation zwischen den beiden Datensätzen auch nach dem Transfer erhalten bleibt. Ist aber ein lokales Katalogisat mit einem anderen L-Satz verknüpft, so wird es hier zu Dateninkonsistenzen kommen, da der Zieldatensatz der Verknüpfung nicht mehr unter der bisherigen PPN vorhanden sein wird.

Die Kategorien-Statistik des Perl-Programms `opc4_lok_titel.pl` wertet allerdings zzt. nicht auf Subfield-Ebene aus; in Subfield $9 gespeicherte Verknüpfungen lassen sich daher nicht gezielt ermitteln. Daher erscheint es angeraten, vor dem Datentransfer eines LBS-Standortes solche Fälle per SQL zu ermitteln und den betreffenden Bibliotheken zwecks manueller Bereinigung zu übermitteln.[385] Die fraglichen Datensätze sollten notwendigerweise vor dem Transfer bearbeitet werden.

---

[380] Siehe Kap. 5.3.
[381] Insbesondere im Vorfeld der Datenmigration wurden Datenbereinigungen als sinnvoll erachtet, im Unterschied zu Phase K (Datenbereinigung) bei Lüssem/Harrach 2013, S. 4.
[382] Siehe Kap. 3.2.2.
[383] Sowohl nach der neuen LOK-Richtlinie zur Erfassung lokaler Katalogisate im CBS als auch der bisher gültigen Richtlinie sind Datensatzverknüpfungen nicht zugelassen.
[384] Eine entsprechende Validation in LBS3 wurde von einigen LBS-Systemverwaltern außer Kraft gesetzt. Die Anpassung der Validationsvorschriften in CAT4 ist erheblich komplexer und daher unterblieben. Für lokale Katalogisate im CBS wäre die Einrichtung einer entsprechenden Validation technisch machbar.
[385] Ggf. wäre zusätzlich eine Erweiterung des Perl-Programms denkbar, damit in diesen Fällen eine entsprechende Meldung protokolliert wird.

Selektionsschlüssel (Kat. 70xx)

Da eine zentrale Abfrage über alle LBS-Datenbanken der LBS-Standorte nicht möglich ist, wurden exemplarisch die tatsächlich genutzten Selektionsschlüssel für einige Standorte ermittelt. Die Stichprobe ergab, dass alle 12 in der Richtlinie genannten Codes bei lokalen Katalogisaten verwendet wurden. Aufgrund der fehlenden Validation für Kategorie `70xx` wurden auch zahlreiche weitere Buchstaben, Ziffern und Sonderzeichen als Selektionsschlüssel-Code ermittelt. Diese Erfassungsfehler können über die per OpenRefine erzeugte Excel-Datei identifiziert werden, eine Bereinigung vor Abschluss des Transferverfahrens für eine Bibliothek muss allerdings manuell im LBS erfolgen. Korrekturen, die den Selektionsschlüssel betreffen, sollten wegen der Auswirkungen auf die Anzeige im lokalen OPAC zeitnah erfolgen.

Ausleihrelevante Daten

Die Angaben in den PICA3-Kategorien `7100` (mit Subfields für Sonderstandort, Signatur, Ausleihindikator und Konvolutindikator) und `8200`[386] (Verbuchungsnummer) sind für die Verbuchung im LBS von entscheidender Bedeutung. Daher sind lokale Katalogisate, die für eine Ausleihe vorgesehen sind, insbesondere hinsichtlich der Angaben in diesen beiden Kategorien zu kontrollieren. Bei der OpenRefine-Transformation wird der Inhalt von Kategorie `7100` auf je eine Spalte pro Subfield aufgeteilt. So kann die Bibliothek in der Excel-Tabelle über Sortier- und Filtermechanismen die Inhalte der einzelnen Elemente kontrollieren. Für Ausleihindikator und Konvolutindikator gelten Vorgaben zur Belegung der Codes, dagegen sind die Angaben zu Sonderstandort und Signatur bibliotheksspezifisch und können daher nur von der Bibliothek selbst sachgerecht geprüft werden. Bei der Kontrolle auf gültige Inhalte werden auch formale Erfassungsfehler sichtbar. So wird der Code für den Ausleihindikator nur dann im korrekten PICA+-Subfield gespeichert und im Ausleihmodul entsprechend berücksichtigt, wenn er bei der PICA3-Erfassung mit dem korrekten Subfield-Kennzeichen „ @ “ eingeleitet wird; dies ist eine verbreitete Fehlerquelle. Derartige Fälle sollten unbedingt behoben werden; die Bibliothek kann diese nun auf einfache Weise über die Excel-Tabelle ermitteln und daraufhin die Datensätze korrigieren.

5.4.2 Löschen lokaler Katalogisate im LBS

Zur Unterstützung der Datenbereinigungen wurde ein VB-Script `LoksatzLoeschen.vbs` entwickelt[387], das per WinIBW ausgeführt wird und Datensätze aus dem LBS löscht; als Eingabedatei dient eine Excel-Datei mit den zu löschenden PPNs. Zur Ausführung des Scripts ist ein LBS3-Zugang erforderlich, da in CAT4 aufgrund eines Programmfehlers beim Löschen die mit dem Exemplarsatz verknüpften Einträge in der LBS-Datenbanktabelle `volume` (sog. Bandsätze für die Ausleihverbuchung) verbleiben. Für die lokalen Katalogisate der beiden LBS4-Bibliotheken ohne zugrundeliegende LBS3-Implementierung kann das Script daher nicht verwendet werden.

Da das Script primär zur Löschung von Fernleihtiteln gedacht ist, wurde es so konzipiert, dass es innerhalb des OUS-Moduls abläuft, da hier beim Löschversuch eine entsprechende Meldung ausgegeben wird, falls ein Bandsatz vorhanden ist. Nur dann brauchen ggf. vorhandene Verknüpfungen zu Ausleihen, Vormerkungen bzw. Forderungen geprüft zu werden. Zur Sicherheit kann ebenfalls auf Verknüpfungen zu ACQ-Bestellungen geprüft werden. Solche Datensätze werden beim Löschen übergangen, wenn dies beim Scriptablauf entsprechend angegeben wird.

[386] Vgl. Verbundzentrale des GBV 2011.
[387] Das Grundgerüst des Scriptes stammt von Jarmo Schrader, UB Hildesheim.

Aufgrund der integrierten Datenbankabfragen wird für die Ausführung des WinIBW-Scripts ein ODBC-Treiber für Sybase auf dem ausführenden PC benötigt. Eine Dokumentation[388] mit Screenshots unterstützt die Anwender bei der Scriptausführung. Script und Dokumentation wurden bisher interessierten LBS-Systemverwaltern auf Anfrage zugänglich gemacht, da wegen des benötigten ODBC-Treibers ohnehin nur ein kleiner Kreis an aktiven Anwendern in Frage kommt.[389] Das Script wurde bereits von einigen LBS-Systemverwaltern aktiv eingesetzt.

Weitere Details zum VB-Script `LoksatzLoeschen.vbs` siehe Anhang 13.

[388] Klute 2018.
[389] Die Systemverwalter von 27 LBS-Standorten sowie VZG-Mitarbeiter.

5.5 Datenmigration

Voraussetzung für den Transfer der lokalen Katalogisate ist die Anpassung der CBS-Umgebung, um die Speicherung solcher Datensätze zu ermöglichen. Zu diesem Zweck musste zunächst im Rahmen eines Vorprojektes unter Einbeziehung der Abteilung „Bibliothekarische Dienste“ ein Konzept für die zukünftige lokale Katalogisierung entwickelt werden, daraus resultierten Konfigurationsänderungen hinsichtlich Indexierung, Validation und Datenpräsentation.[390] Entschieden wurde, dass eine Unterscheidung hinsichtlich der verschiedenen Datensatzarten (Fernleihtitel, Abschlussarbeiten, Online-Lizenzen u. a. m.) zukünftig über Kategorie `8600` getroffen werden kann, als Materialart-Kennzeichen für alle lokalen Katalogisate wurde einheitlich „Lax“ festgelegt.

Die Überlegungen hinsichtlich der neuen Lax-Sätze führten zur Neukonzeption der sog. LOK-Richtlinie[391] mit Informationen für GBV-Anwender zur Erfassung lokaler Katalogisate und deren Besonderheiten bei der Recherche. Im Unterschied zu kooperativ erfassten Datensätzen im CBS sind Lax-Sätze nur für die erfassende Bibliothek sichtbar; entsprechend werden bei einer Titelrecherche nur eigene Lax-Sätze berücksichtigt. Die Richtlinie enthält ebenfalls Beispiele typischer Katalogisate für die nach den neuen Vorgaben zu erfassenden Datensätze. Eine Pilotbibliothek hatte, noch mit provisorischen Vorgaben, im Februar 2017 mit der Erfassung lokaler Katalogisate im CBS begonnen. Zunächst wurden nur einzelne Bibliotheken auf die neue LOK-Richtlinie aufmerksam gemacht, damit im Falle von Nachfragen ausreichend Beratungszeit zur Verfügung stand. Am 11.04.2018 wurden die LBS-Bibliotheken von der VZG offiziell über den Wechsel der Katalogisierungsplattform für lokale Katalogisate und das Projekt zum Transfer der Datensätze informiert.

5.5.1 Konzept

Das Konzept für den eigentlichen Transfer der lokalen Katalogisate ins CBS wurde am 26.09.2017 beim jährlichen Treffen der LBS-Systemverwalter vorgestellt.[392] Das ETL-Verfahren gliedert sich in 6 Schritte:

1.	Im LBS: Daten im PICA+-Format extrahieren	EXTRACT
2.	Im LBS: Daten konvertieren	TRANSFORM
3.	Im CBS: Daten importieren	LOAD
4.	Update der Daten vom CBS ins LBS	
5.	Im LBS: Verknüpfungen zu OUS- und ACQ-Tabellen umhängen	
6.	Im LBS: Alte L-Sätze löschen	

Die Grundidee des Konzepts zur Verlagerung der lokalen Katalogisate ins CBS stammt ursprünglich von OCLC.[393] Dazu waren der VZG Dokumentationen und Shellscripte für einzelne Schritte zur Verfügung gestellt worden.[394] Nach erster Durchsicht der Scripte wurden erhebliche Anpassungsnotwendigkeiten erkannt, so waren u. a. die Datenbanktabellen für das LBS4-Tauschmodul nicht berücksichtigt worden. Daher wurden die Scripte korrigiert und auf die aktuellen GBV-Bedingungen angepasst. Weitere Shellscripte wurden entwickelt, ebenfalls ein SQL-Script zum abschließenden Löschen der obsoleten L-Sätze.

[390] Vorgaben für die lokale Katalogisierung im CBS gemäß Phase E (Konzept zur Datentransformation) nach Lüssem/Harrach 2013, S. 4.
[391] Vgl. Verbundzentrale des GBV 2017.
[392] Vgl. Klute 2017 sowie die zugehörige PPT-Präsentation mit weiteren Details, siehe beiliegende CD.
[393] Vgl. Frans 2014.
[394] Erstellung eines Migrationskonzepts entsprechend Phase D bei Lüssem/Harrach 2013, S. 3.

Nach der ursprünglichen Idee waren jeweils einzelne Scripte auszuführen, einige im LBS, weitere im CBS. Nach einigen Test-Durchläufen wurde daher zur Vereinfachung ein Shellscript entwickelt[395], mit dem der komplette Ablauf für alle Bibliotheken eines Standorts durchgeführt werden kann. Einzig das abschließende Löschen der bisherigen lokalen Katalogisate per SQL in der LBS-Datenbank wird weiterhin separat ausgeführt, um vorher die Ergebnisse des Datentransfers kontrollieren zu können.

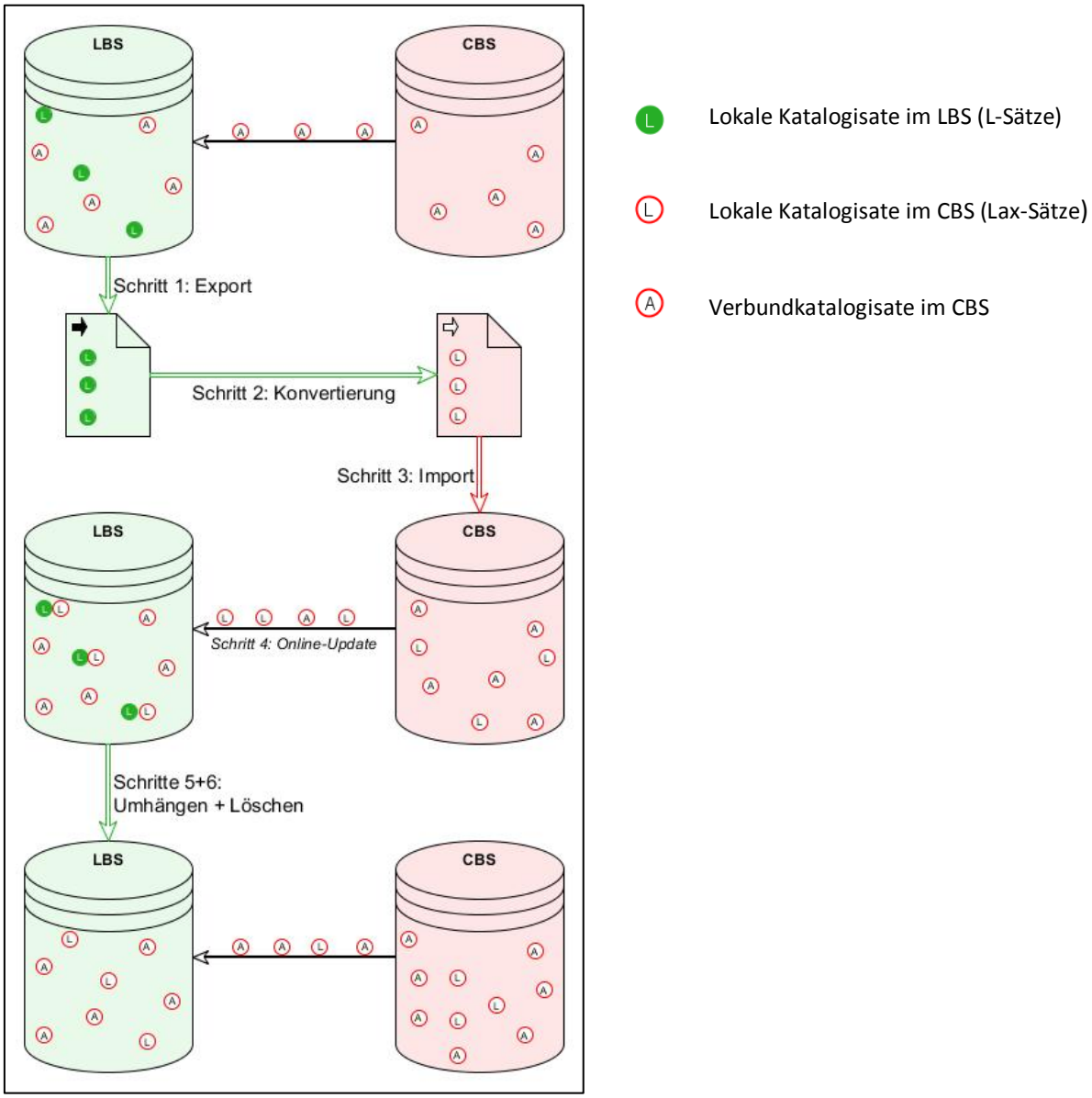


Abb. 14: LOK-Projekt: Ablauf

[395] Entsprechend Phase F (Programmanpassungen) bei Lüssem/Harrach 2013, S. 4.

5.5.2 Kritische Auseinandersetzung mit dem Transferverfahren

Zur Vorbereitung des Transferverfahrens[396] wurde ein LBS-Testserver der VZG ausgewählt, auf dem das CBS-Dienstprogramm `csft_ttlextract` installiert wurde.[397] Im Unterschied zum geplanten produktiven Ablauf, bei dem der Zugriff auf die jeweiligen LBS-Datenbanken über die Umgebungsvariable DSQUERY hergestellt werden soll, wurde zunächst für die erforderlichen Tests die jeweilige LBS-Datenbank auf den Testserver kopiert. Somit erfolgen die Tests in einer anderen Systemkonfiguration als bei dem späteren produktiven Transferverfahren.

Bisher wurden ausschließlich Testläufe durchgeführt. Um das Verfahren zunächst in einem überschaubaren Zeitfenster prüfen zu können und daraus Erfahrungen für weitere (Test-)Läufe zu sammeln, wurde mit den Daten eines LBS-Standorts begonnen, dessen 4 Bibliotheken lediglich zwischen 50 und 5.200 lokale Katalogisate erfasst hatten. Inzwischen wurden weitere Tests für die Daten eines anderen Standorts mit 6 Bibliotheken und bis zu 43.000 lokalen Katalogisaten absolviert.

In der folgenden Darstellung der in Kap. 5.5.1 aufgeführten Schritte werden schwerpunktmäßig die besonderen Herausforderungen des Verfahrens aufgezeigt.[398] Darüber hinaus werden Abschätzungen für den zeitlichen Umfang der Schritte 1–5 und 6 vorgenommen.

Schritt 1: Extraktion der Metadaten

Als Vorbereitung wird zunächst für jede Bibliothek des LBS-Standorts eine IPN-Liste der betreffenden Titel erzeugt. Das Programm `csft_ttlextract` extrahiert anschließend die Daten aus der LBS-Datenbank anhand der IPN-Listen, die Ausgabedatei enthält Titeldaten im PICA+-Format.

Das Konzept von OCLC sieht zur Erstellung der IPN-Listen eine SQL-Abfrage auf die Datenbanktabelle `titles_copy` vor. Für lokale Katalogisate ist ein spezieller IPN-Bereich reserviert, der für den gesamten LBS-Standort gilt; damit können die Datensätze eingegrenzt werden. Problematisch ist allerdings, dass bei den LBS-Standorten unterschiedliche IPN-Bereiche für lokale Katalogisate festgelegt wurden, sodass im Ergebnis für das Transferprojekt der IPN-Bereich zwischen 90.000.000 und 100.000.000 als für alle Standorte gleichermaßen geeignet erachtet wurde. Zusätzlich sind lokale Katalogisate laut OCLC-Dokumentation über das Tabellenattribut `status` mit Wert „1“ erkennbar.[399]

Die Datenbankabfrage zur Erstellung der IPN-Liste lautet demnach (sinngemäß):

```
select ppn from titles_copy where fno=<fno> and ppn>90000000 and
ppn<100000000 and status="1"
```

Um diese Abfrage zu verifizieren, wurde für jeden Standort der tatsächlich genutzte IPN-Bereich ermittelt. Es stellte sich heraus, dass für einen Standort auch lokale Katalogisate im IPN-Bereich 20022435–20412713 vorhanden sind; diese waren vor längerer Zeit durch einen direkten Import ins LBS erzeugt worden. Der Direktor der VZG entschied, dass diese lokalen Katalogisate beim Transferverfahren (zunächst) nicht zu berücksichtigen seien.

[396] Zur Vorbereitung der Testumgebung vgl. Phase G bei Lüssem/Harrach 2013, S. 4.
[397] Vgl. Abschnitt zu `csft_ttlextract` in Kap. 5.2.2.
[398] Kontrolle von Transferablauf und -ergebnissen vgl. Phase I (Abnahme der Testmigration) bei Lüssem/Harrach 2013, S. 4.
[399] Vgl. Frans 2014, S. 3.

Die Festlegung auf Status 1 musste ebenfalls hinterfragt werden. Tatsächlich existieren für einige Bibliotheken auch lokale Katalogisate aus dem betreffenden IPN-Bereich, die trotz Materialkennung „L“ den Status 0 aufweisen. Dieser Status wird für gewöhnlich bei Datensätzen gesetzt, die im CBS erfasst und per Online-Update-Verfahren ins LBS übertragen werden. Auch direkt ins LBS importierte Datensätze haben Status 0, dies ist allerdings für die betreffenden Bibliotheken nie erfolgt. Die Ursache der Dateninkonsistenz ist demnach unklar, daher müssen diese Datensätze mit Status 0 manuell bereinigt werden, da sie beim Transferverfahren nicht berücksichtigt werden.

Schritt 2: Konvertierung der Daten

Die Ausgabedatei des Programms `csft_ttlextract` enthält sämtliche Kategorien der lokalen Katalogisate in normalisiertem PICA+-Format[400].

```
##TTLnumber 98000156
-001A 00000:11-11-97
-001B 00000:11-11-97t13.37.02.110
-001D 09999:99-99-99
-002@ 0Lux
-003@ 0980001560
-011@ a1997n1997-
-021A aSignal transduction catalog & technical resource. Calbiochem
-029A aCalbiochem
-033A pBad Soden/Ts.nCalbiochem-Novabiochem GmbH
-101@ a20
-201B/01 031-01-03t13.24.51.860
-203@/01 0980001560
-208@/01 a11-11-97bd
-209A/01 aZNKx00
-237A/01 aKat. [N], Wegwerfen / Ob, 7.11.97 / 97 99 00 03
```

Datenbeispiel 1: Ausgabe von `csft_ttlextract`

Ziel der Konvertierung ist eine Datei, die als Eingabe für den CBS-Datenbankimport dient. Dabei bleibt der Datensatz in seiner Form erhalten, es werden lediglich die Inhalte einiger Kategorien ersetzt bzw. in eine neue Kategorie verschoben.

```
-001A 00000:11-11-97
-001B 00000:11-11-97t13.37.02.110
-001D 09999:99-99-99
-002@ 0Lax
-011@ a1997n1997-
-021A aSignal transduction catalog & technical resource. Calbiochem
-029A aCalbiochem
-033A pBad Soden/Ts.nCalbiochem-Novabiochem GmbH
-101@ a20
-201B/01 031-01-03t13.24.51.860
-207F/01 lLBS0980001560a980001560
-208@/01 a11-11-97bd
-209A/01 aZNKx00
-209O/01 aLOKMAT: Luxx00
-237A/01 aKat. [N], Wegwerfen / Ob, 7.11.97 / 97 99 00 03
```

Datenbeispiel 2: Datensatz nach Konvertierung

Die Unterschiede vor und nach der Konvertierung sind farbig hervorgehoben.

[400] Mit binären Steuerzeichen, siehe Kap. 3.1.3.

Da die Datensätze beim Import neue PPNs und EPNs erhalten, werden die Inhalte der Kategorien 003@ (PPN) und 203@ (EPN) in einer Kategorie 207F eingefügt; damit besteht auch nach dem Transfer zu Prüfzwecken Zugriff auf die bisherigen Identifier. Die bisherige Materialart in Kategorie 002@ wird in Kategorie 2090 abgelegt, da hierüber zukünftig eine Unterscheidung der Datensatzarten ermöglich wird. Die Materialart Lax wird pauschal für alle Datensätze gesetzt. Die den Kategorien vorangestellte Zeile mit der aktuellen IPN als ##TTLnumber wird ebenfalls entfernt.

Schritt 3: CBS-Datenimport

Zunächst erfolgt eine Zeichenkonvertierung der Eingabedaten zu UTF-8, anschließend werden die Daten in der CBS-Verbunddatenbank gespeichert. Anders als z. B. beim Import von Daten eines Legacy-Bibliothekssystems erfolgt der Import ohne Match-and-Merge direkt in die Verbunddatenbank. Anschließend wird eine PPN-EPN-Konkordanzdatei erstellt, bestehend aus alter PPN, alter EPN, neuer PPN und neuer EPN.

Schritt 4: Online-Update CBS → LBS

Der standardmäßig für jede LBS-Bibliothek konfigurierte Updateprozess extrahiert die neuen Lax-Sätze aus dem CBS und speichert sie im LBS. Da dieser Prozess bei größeren Datenmengen den sofortigen LBS-Zugriff auf normale CBS-Datensatzänderungen (Neuaufnahmen, Korrekturen) verhindert, werden im GBV Datenimporte für große Datenmengen, z. B. für umfangreiche E-Book-Pakete, generell nachts durchgeführt. Für das Einspielen der lokalen Katalogisate müsste das Verfahren möglicherweise ebenfalls entsprechend terminiert werden. Problematisch erscheint dabei, dass das Transferverfahren grundsätzlich für alle Bibliotheken eines Standorts durchgeführt wird. Sind hier mehrere Bibliotheken mit einer größeren Anzahl L-Sätze vertreten, kann die vertretbare Menge von ca. 50.000 Datensätzen pro Nacht leicht überschritten werden. Beim Standort Braunschweig mit 6 Bibliotheken waren bei einem Testlauf zusammen knapp 92.000 Datensätze verarbeitet worden; allerdings benötigte der Updateprozess nur 2:20 Stunden. Dies liegt eventuell an dem eher geringen Umfang von lokalen Katalogisaten im Vergleich zu E-Book-Aufnahmen, die häufig ausführliche Abstracts enthalten.

Trotz des vergleichsweise kurzen Zeitraums bei diesem Testlauf würde der Transferprozess tagsüber vor allem die regulären Bestellkatalogisierungs-Vorgänge beeinträchtigen; diese Datensätze würden erst mit entsprechender Verzögerung im LBS benutzbar sein. Betroffen sind Mitarbeiter der Erwerbungsabteilung, die während des Update-Zeitraums keine Vorgänge im Erwerbungsmodul ACQ für neu angelegte CBS-Katalogisate durchführen können. Daher ist mit den Bibliotheken der voraussichtliche Freeze-Zeitraum abzusprechen. Insbesondere sollten Fernleihabteilungen auf das Anlegen von Lax-Sätzen im CBS während des Transferzeitraums verzichten. Um einen ungestörten Migrationsablauf zu gewährleisten, muss eine entsprechende Ausfallzeit seitens der Bibliotheken in Kauf genommen werden.

Zurzeit sind für 6 LBS-Bibliotheken jeweils mehr als 100.000 lokale Katalogisate vorhanden, mit einer Spannbreite von 140.000 bis 756.000 Datensätzen. Es erscheint sinnvoll, weitere Testläufe für einen der betreffenden Standorte durchzuführen, um eine zeitliche Abschätzung für den Produktivlauf zu erhalten. Vermutlich muss der Datenimport bei diesen Standorten in mehreren Teilen erfolgen; dazu ist zu prüfen, inwieweit die Shellscripte entsprechend angepasst werden können.

Ungeklärt ist derzeit, wie verfahren wird, wenn sich der Update-Prozess „aufhängt“. Auch im normalen Regelbetrieb kann eine unerwartete (Daten-)Konstellation dazu führen, dass der Prozess

für das Online-Update formal weiter läuft, aber keine Daten mehr verarbeitet. Für solche Fälle muss Sorge getragen werden, dass während der Durchführung des LOK-Transfers für einen Standort mit einer großen Menge lokaler Katalogisate „Erste Hilfe“ durch entsprechend instruierte VZG-Mitarbeiter geleistet werden kann. Dies trifft auf voraussichtlich 5 Standorte zu.

Schritt 5: Umsetzen der Verknüpfungen zu ACQ- und OUS-Tabellen in der LBS-Datenbank

Für jedes lokale Katalogisat ist nun zusätzlich ein entsprechender Lax-Satz im LBS vorhanden. Die bisherigen L-Sätze sind i. d. R. durch relationale Verknüpfungen mit (meist mehreren) LBS-Datenbanktabellen verbunden. Zur Ausleihe vorgesehene Datensätze haben einen sog. Bandsatz (Tabelle `volume`), zu Erwerbungszwecken erfasste Katalogisate sind mit der Tabelle `orders` verknüpft, um nur einige der zu berücksichtigenden Tabellen zu nennen. Der Bezug zwischen den Metadaten und den LBS-Tabellen wird über die Tabellenattribute ppn bzw. epn hergestellt.

Abb. 15: LBS-Tabellen vor dem Umsetzen der PPN-/EPN-Verknüpfungen

In den LBS-Datenbanktabellen müssen diese PPNs bzw. EPNs[401] durch die beim Datenimport erzeugten Pendants der neuen Lax-Sätze ersetzt werden. Hierzu wird die bei Schritt 3 erzeugte PPN-EPN-Konkordanzdatei ausgewertet: Die EPN eines bisherigen L-Satzes wird durch die EPN des neuen Lax-Satzes ersetzt, analog wird mit der PPN bei den entsprechenden Tabellen verfahren.

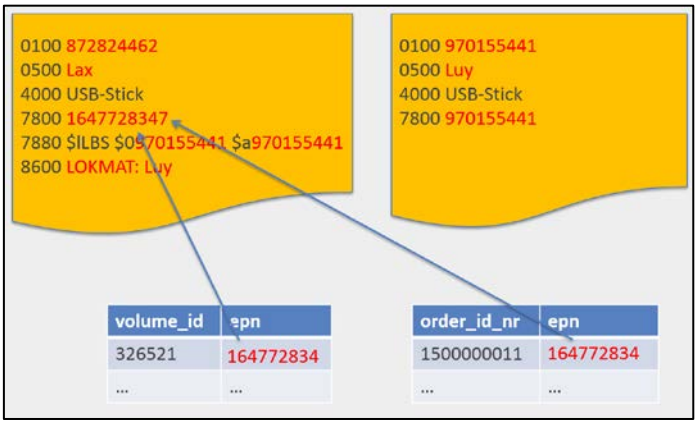


Abb. 16: LBS-Tabellen nach dem Umsetzen der PPN-/EPN-Verknüpfungen

Der Austausch der EPNs bzw. PPNs erfolgt durch entsprechende SQL-Update-Statements, die zunächst in eine Datei geschrieben und anschließend ausgeführt werden. Im Transfer-Script ist je

[401] Zur Vereinfachung wird hier von PPNs und EPNs gesprochen, tatsächlich sind es die jeweils um die Prüfziffer gekürzten internen Nummern, vgl. Abb. 15 (in Kategorie 7800 mit Prüfziffer, in der Datenbank ohne Prüfziffer).

eine Liste von 4 PPN-Tabellen und 8 EPN-Tabellen hinterlegt. Für jedes EPNalt-EPNneu-Paar wird ein Update-Statement für jede EPN-Tabelle erzeugt, entsprechendes gilt für die PPN-Paare. Somit werden für jedes lokale Katalogisat 12 SQL-Statements erzeugt. Das Schreiben dieser Statements in die Datei dauert verhältnismäßig lange: für die Daten einer Bibliothek mit 43.583 lokalen Katalogisaten wurden 55 Minuten für die PPN-bezogenen Statements benötigt und 127 Minuten für die EPN-bezogenen Statements. Die Dauer für das Schreiben der Statements aller 6 Bibliotheken dieses Standorts betrug in der Summe 102 Minuten für PPN-Tabellen und 196 Minuten für EPN-Tabellen.[402] Die Ausführungsdauer der SQL-Statements hingegen war unauffällig (insgesamt 50 Minuten für diesen Standort). Die Gesamtdauer der Schritte 1-5 des Transferprozesses betrug bei diesem Standort 8:57 Stunden für 91.949 Datensätze.

Dieser Schritt der Umsetzung der LBS-Datenbankverknüpfungen birgt erhebliches Optimierungspotenzial. Das aktuelle Verfahren sieht eine (mathematische) Kombination aus Datensätzen und zu berücksichtigenden Tabellen vor, ungeachtet der Tatsache, ob die betreffenden Tabellen von der Bibliothek genutzt werden.

Tabelle	Primärschlüssel EPN / PPN	LBS-Modul
delivery	EPN	ACQ
orders	EPN	ACQ
orders_ppn	EPN + PPN	ACQ
ous_transaction	EPN + PPN	OUS
pamphlet_volume	EPN	OUS
volume	EPN	OUS
reservation	PPN	OUS
acq_shipment_log	EPN + PPN	Tausch
acq_stock	EPN	Tausch

Tab. 10: Beim Umsetzen von Verknüpfungen zu berücksichtigende LBS-Datenbanktabellen

Farbig gekennzeichnete Tabellen sind nur für wenige Bibliotheken relevant.

Für die einzelnen Tabellen ist präzise auszumachen, in welchem funktionalen Zusammenhang sie genutzt werden, insbesondere sind die Tabellen `acq_shipment_log` und `acq_stock` einzig für das LBS4-Tauschmodul von Belang, das nur für eine kleine Zahl Bibliotheken eingerichtet ist. Zusätzlich wäre zu berücksichtigen, ob die betreffenden Tabellen tatsächlich Einträge für lokale Katalogisate haben. Dies betrifft vor allem die Tabellen `pamphlet_volumes` und `reservation`. Die Tabelle `pamphlet_volumes` wird ausschließlich beim Anlegen eines Konvoluts gefüllt. In der Tabelle `reservation` sind Eintragungen mit PPN nur dann vorhanden, wenn die Funktion der Titelvormerkung genutzt wird, dies ist bei der überwiegenden Mehrheit der Bibliotheken nicht der Fall. Dass diese beiden Tabellen EPNs bzw. PPNs von L-Sätzen enthalten, dürfte eine große Ausnahme sein, die gleichwohl beim Transferverfahren berücksichtigt werden muss.

Daher wäre zu überlegen, die Menge der EPN- bzw. PPN-Tabellen je nach Standort zu modifizieren. Dazu könnte im Vorwege für einen Standort per SQL ermittelt werden, ob in den betreffenden Tabellen Einträge vorhanden sind, ggf. weiter eingeschränkt auf lokale Katalogisate. Eventuell bietet

[402] Es handelte sich um 1,2 Mio. SQL-Statements.

es sich im Hinblick auf eine erhebliche Reduzierung der Prozesslaufzeit auch an, bei Standorten mit zahlreichen Bibliotheken und sehr vielen L-Sätzen den Ablauf separat für Gruppen von Bibliotheken zu starten, die dieselbe (Teil-)Menge von Tabellen nutzen.

Schritt 6: Löschen der redundanten alten lokalen Katalogisate

Das Löschen der L-Sätze, die nun keine Verknüpfungen mehr zu LBS-Tabellen haben, wird per SQL-Delete-Statements durchgeführt. Da im vorangegangenen Schritt die Verknüpfungen von diesen Datensätzen zu LBS-Tabellen gelöst wurden, ist dieses Verfahren zum Löschen geeignet.[403]

Unter der Annahme, dass bei lokalen Katalogisaten PPN und EPN identisch sind, werden für jede Bibliothek die alten PPNs aus der bei Schritt 3 entstandenen PPN-EPN-Konkordanzdatei extrahiert und daraus die entsprechenden SQL-Delete-Statements generiert. Die SQL-Statements werden in einer Datei gespeichert und anschließend zur Ausführung gebracht. Bisher wurde nur ein einziger Testlauf durchgeführt.

Die Datenbankabfrage zur Erzeugung der Delete-Statements lautet sinngemäß:

```
delete <tabelle> where [fno=<bestandsnr> and] [ppn|epn]=<ppn>
```

Zurzeit sind in der hinterlegten Liste der zu berücksichtigenden Datenbanktabellen 9 Tabellen eingetragen. Die Anzahl der SQL-Statements berechnet sich aus der Anzahl PPNs multipliziert mit der Menge der Tabellen; für den Standort mit insgesamt 91.949 Datensätzen wären das 827.541 Delete-Statements. Wenn man von derselben Schreibgeschwindigkeit wie bei Schritt 5 ausgeht, ist mit einer Dauer von 224 Minuten zu rechnen.

Bei genauerer Betrachtung der Liste der Datenbanktabellen kann tatsächlich eine Tabelle komplett entfallen. Alle anderen Tabellen speichern metadatenbezogene Elemente und sind somit für alle Bibliotheken relevant, unabhängig von der Nutzung der lokalen Module Ausleihe und Erwerbung. Daraus resultiert allerdings nur eine geringfügige Optimierung dieses Prozessschrittes, und zwar auf geschätzt 199 Minuten Laufzeit für das Schreiben der SQL-Statements in eine Datei. Die Dauer der Ausführung selbst dürfte, wie bei Schritt 5, vernachlässigbar sein.

Damit ist auch die bisherige (falsche) Annahme, PPN und EPN eines L-Satzes seien identisch, bedeutungslos, da die wegfallende Tabelle als Einzige nur über EPN ansprechbar ist. Das Shellscript zum Generieren der SQL-Statements kann daher vereinfacht werden.

Ungeachtet dessen sollte für eine vorsichtige Schätzung der Laufzeit für Schritt 6 bedacht werden, dass in der Tabelle `title_keywords`[404] jeweils mehrere Tupel pro PPN vorhanden sind und gelöscht werden müssen. In welchem Maße sich die Ausführungszeit demzufolge verlängert, ist ungewiss. Für Bibliotheken mit überwiegend Fernleihtiteln als L-Sätze dürfte die Menge der Eintragungen in `title_keyword` pro PPN überschaubar sein. Bei Bibliotheken, die zahlreiche lokale Katalogisate für andere Materialien erstellt haben, ist mit einer erheblich höheren Anzahl an Eintragungen in dieser Tabelle pro PPN zu rechnen.

[403] Hingegen ist bei Datenbereinigungen vorab ungewiss, ob für die betreffenden Datensätze Tabellenverknüpfungen existieren; daher wird vom Löschen per SQL durch LBS-Systemverwalter vor Beginn des Transferverfahrens wegen der Komplexität der Tabellenverknüpfungen generell abgeraten, vgl. Kap. 5.4.2.
[404] Enthält Indexeinträge für die Suche unter LBS3.

Exemplarisch wurden einige Werte für verschiedene Bibliotheken aus unterschiedlichen Standorten ermittelt.

Bibliothek	Anzahl L-Sätze	Anzahl LBS3-Keywords	Keywords pro Datensatz
1	1.640	14.167	8,6
2	18.016	117.242	6,5
3	43.564	867.939	19,9
4	128.970	1.700.495	13,2
5	562.621	5.993.985	10,7
6	5.282	104.533	19,8

Tab. 11: Verhältnis lokale Katalogisate zu LBS3-Indexeinträgen

Da bisher lediglich ein Testlauf für das Löschen von L-Satz-spezifischen Tabelleneinträgen per SQL durchgeführt wurde, steht nur für die Daten von Bibliothek 6 eine verwertbare Zahl zur Verfügung: die Löschdauer betrug 32 Minuten. Auf diesem Wert basieren die weiteren Hochrechnungen.

Vermutlich kann bei Kalkulationen der Laufzeit von Schritt 6 für weitere Bibliotheken nicht die Anzahl L-Sätze zugrunde gelegt werden, sondern eher die Anzahl der Einträge in `title_keywords`[405], da pro Datensatz ein Vielfaches an Indexeinträgen vorhanden ist. Wird für eine Berechnung nur die Anzahl L-Sätze berücksichtigt, so ergibt sich für das Löschen der L-Sätze von Bibliothek 5 eine voraussichtliche Dauer von 1.835 Minuten, also mehr als 30 Stunden. Unter Einbeziehung der Anzahl Keywords pro L-Satz reduziert sich die geschätzte Lösch-Zeit auf ca. 16,5 Stunden.

Diese Berechnungen können allerdings lediglich die Problematik von potenziell sehr langen Laufzeiten aufzeigen, da mit nur einem einzigen Ausgangswert keine belastbaren Prognosen möglich sind.

Ebenso ist ungewiss, ob sich die Laufzeiten der Schritte 1, 2, 5 und 6 verändern, wenn die betreffende Datenbank physisch auf einem anderen Server gespeichert ist und der Zugriff darauf per DSQUERY erfolgt, insbesondere bei den 5 Servern auf Hosts außerhalb der VZG. Weiteren Einfluss könnten die VZG-Routinen zur Datenbankoptimierung für die Mehrzahl der produktiven LBS-Systeme haben.

Trotz der voraussichtlich langen Laufzeit wegen der Speicherung der SQL-Delete-Statements in einer Datei hat dieses Verfahren den Vorteil, dass die Ausführung der SQL-Statements ggf. in Teilen erfolgen kann.

Bei den zahlreichen Durchläufen des Programms `opc4_lok_titel.pl` wurden Datensätze mit zwei PPNs entdeckt. Wie damit im Transfer-Ablauf umgegangen wird, wurde sorgfältig untersucht. Tatsächlich wird das erste Vorkommen der PICA+-Kategorie `003@` als PPN interpretiert; hierbei handelt es sich jedoch um die PPN desjenigen Datensatzes, den ein Mitarbeiter beim Anlegen des L-Satzes als „Kopiervorlage“ genutzt hatte. Dies hat im weiteren Verlauf gravierende Auswirkungen, da

[405] Zusätzlich wäre zu bedenken, dass die Anzahl Indexdefinitionen in den LBS-Standorten nicht einheitlich ist; für die in Tab. 11 aufgeführten Bibliotheken schwankt sie zwischen 13 und 18. Ebenfalls ist in einigen Standorten bereits der Prozess zum Generieren von LBS3-Indexeinträgen abgeschaltet worden. Für die Bibliotheken 2 und 3 ist dies seit Januar 2017 der Fall; seitdem dürften dort keine weiteren Einträge in `title_keywords` hinzugekommen sein.

dann die Verknüpfungen eines falschen Datensatzes umgehängt werden (Schritt 5) und dieser falsche Datensatz bei der abschließenden Löschung per SQL (Schritt 6) entfernt wird. Diese Fälle treten vermutlich recht selten auf, dennoch sollten Maßnahmen zur Vermeidung einer versehentlichen Löschung von unbeteiligten Datensätzen ergriffen werden. Angedacht ist eine entsprechende Erweiterung des für die Konvertierung genutzten Perl-Scripts (Schritt 2).

5.5.3 Schlussbetrachtung

Die hier vorgelegte Analyse des Projekts zur Migration der lokalen Katalogisate die CBS-Verbundumgebung zeigt, dass die Ausgangslage der zu migrierenden Daten von Bibliothek zu Bibliothek sehr heterogen und komplex ist. Dafür konnten verschiedene Ursachen identifiziert werden.

So ist die Datensituation nicht präzise vorhersagbar; daher muss damit gerechnet werden, dass weitere, bisher noch nicht identifizierte Datenkonstellationen durch die ETL-Routinen nicht berücksichtigt sind. Die erhebliche Menge der zu berücksichtigenden Datensätze lässt sich nur mit erheblichem Aufwand im Vorhinein einer Plausibilitätsprüfung unterziehen. Neben dem Datenvolumen spiegelt sich die große Zahl der beteiligten Bibliotheken in der Diversität der Daten wider.

Das Migrationskonzept sieht vor, dass alle lokalen Katalogisate eines LBS-Standortes zum Stichtag pauschal ins CBS migriert werden. Aufgrund der Heterogenität der Daten werden nach Abschluss des Transferverfahrens bibliotheksspezifisch weitere Bereinigungen erforderlich sein.

Dazu sollte das VB-Script `LoksatzLoeschen.vbs` dahingehend angepasst werden, dass es zum Löschen von Lax-Sätzen im CBS geeignet ist, zumal auch zukünftig beim Löschen lokaler Katalogisate ggf. vorhandene Verknüpfungen zu Ausleih- oder Erwerbungsdaten in der LBS-Datenbank zu berücksichtigen sind. Ein entsprechendes performanteres Offline-Programm im CBS-Kontext zur Löschung von Daten kann dies nicht leisten, so dass zur Vermeidung von Dateninkonsistenzen eine vorherige Kontrolle per SQL auf Tabellenverknüpfungen in der LBS-Datenbank erforderlich ist.

Für spezifische Konstellationen kann dennoch sinnvoll sein, die weitere Bereinigung per CBS-Reparaturprogramm[406] durchzuführen. In einem Fall müssen 240.000 Datensätze für Zeitschriftenbände ohne Stücktitel zu einer regulären CBS-Aufnahme als sog. Av-Satz konvertiert werden. In einem anderen Fall ist für 5.000 Dissertationen, für die zusätzlich zur regulären CBS-Aufnahme ein lokales Katalogisat für den Nachweis des Archivexemplars angelegt worden war, die Zusammenführung der Exemplare mit dem CBS-Katalogisat erforderlich.

Eine bibliotheksinterne Koordination hinsichtlich der Handhabung lokaler Katalogisate, die in unterschiedlichen Abteilungen erfasst werden, fand bisher in der Regel nicht statt. Hier könnte das LOK-Projekt Anreize für abteilungsübergreifenden Absprachen bieten.

[406] Entsprechend Phase K (Datenbereinigung) nach Lüssem/Harrach 2013, S. 4.

6 Zusammenfassung und Ausblick

Bei den in dieser Arbeit betrachteten Themenfeldern wurde deutlich, dass die Prinzipien von ETL-Prozessen auf den bibliothekarischen Kontext übertragbar sind und sich dort in vielfältigen Ausprägungen wiederfinden.

Es wurde gezeigt, dass die Migration von Bibliotheksmanagementsystemen oder die Aggregierung von Daten in Data-Warehouse-Systemen in Bibliotheken grundsätzlich den gleichen Prinzipien wie bei Anwendungsfällen in anderen Wirtschaftsbereichen folgen. Bibliografische Daten unterscheiden sich allerdings von strukturierten Daten aus anderen Anwendungsfeldern. Auch wenn auf eine Vielzahl von Tools zur Konvertierung und Bearbeitung von bibliografischen Metadaten zurückgegriffen werden kann, ist zu beachten, dass bibliografische Daten schwächer strukturiert und normiert sind. Durch die Wechselwirkung mit den bibliothekarischen Regelwerken zur Formalerschließung ist über die verschiedenen Datenquellen eine große Diversität festzustellen, die den Einsatz von automatisierten Verfahren erschwert.

An dem vorgestellten LOK-Projekt zur Migration lokaler Katalogisate in die zentrale Verbunddatenbank des GBV lassen sich die vorgestellten Grundsätze und Mechanismen von ETL-Prozessen exemplarisch nachweisen. So ist es schon bei der Planung wichtig, sich über die gewählte Strategie („Big Bang“ oder „Chicken Little“) und die daraus folgenden Konsequenzen Gedanken zu machen.

Die vorliegende Arbeit hat gezeigt, dass eine strukturierte und gründliche Datenanalyse wesentlicher Bestandteil von Migrationsprozessen ist. Dafür sind neben Domänenwissen zusätzlich Kompetenzen im Umgang mit bibliothekarischen Schnittstellen und Datenformaten unerlässlich, da nur mit einem möglichst genauen Bild der Datenausgangslage angemessene Entscheidungen für geeignete Strategien und die konkrete Umsetzung gefällt werden können.

Im konkreten Anwendungsfall des LOK-Projekts (und bei zukünftigen Projekten) stellt allerdings der Zugriff auf Datenquellen zum Zweck einer effizienten Datenanalyse eine zentrale Herausforderung dar. Zwar stehen generell verschiedene Methoden für den Zugriff auf LBS-Daten zur Verfügung, aktuell nutzbar ist ausschließlich die proprietäre XML-Schnittstelle für PSI-Datenbanken, sofern die vollständige Ausgabe aller PICA-Kategorien benötigt wird. Dies wurde mit dem Programm `opc4_lok_titel.pl` realisiert. Für die weiteren potenziellen Verfahren wurden die jeweiligen Probleme und Schwachstellen aufgezeigt. Hier wäre vor allem die Ergänzung der OWC-Kataloge aller LBS-Bibliotheken im GBV-Datenbankverzeichnis anzustreben, um den Zugriff auf die Daten einer Bibliothek auch über die Standardschnittstelle SRU zu ermöglichen.

Die im Rahmen dieser Arbeit beschriebenen Verfahren für den Zugriff auf die lokalen Katalogisate sind umso wichtiger, als bei den ursprünglichen Planungen für die Umsetzung der lokalen Katalogisate in das CBS eine Datenanalyse zunächst nicht vorgesehen war; der ETL-Prozess wurde dadurch auf die technische Umsetzbarkeit der Datenmigration reduziert. Es wurde jedoch nach einer ersten Bestandsaufnahme mit Hilfe des Perl-Programms `opc4_lok_anzahl.pl` deutlich, dass ein Teil der vorhandenen lokalen Katalogisate nicht für den Transfer ins CBS geeignet ist. Deshalb ist vorab ein Bereinigungsprozess erforderlich.

Um Bibliotheken bei ihren Bereinigungsvorhaben bestmöglich zu unterstützen, wurde ein Verfahren zur Aufbereitung der PICA+-Datenausgabe des Programms `opc4_lok_titel.pl` konzipiert. Zur Datenanalyse vor Ort ist eine tabellarische Ausgabe als niederschwelliger Zugang zu den Metadaten

besser geeignet; daher wurden für die Weiterverarbeitung der Ausgabedateien verschiedene Tools für das Metadatenmanagement evaluiert. Im Ergebnis ist OpenRefine für diesen Zweck am besten geeignet. Mit dessen Ausgabedateien werden Bibliotheken in die Lage versetzt, Fehlerquellen und Fälle unsachgemäßer Erfassung zu identifizieren und abzustellen und dadurch die Datenqualität der zu migrierenden Datensätze zu erhöhen. Darüber hinaus verringert sich der Aufwand für Nachbearbeitungen und verbessert die Qualität für zukünftig zu erstellende Daten. Ebenfalls lassen sich so Datensätze ermitteln, die von temporärer Bedeutung sind oder aus anderen Gründen gelöscht werden sollten. Dies ist nun mit Hilfe eines programmgestützten Löschverfahrens in Form des VB-Scriptes `LoksatzLoeschen.vbs` durchführbar, das es Bibliotheken erlaubt, vor der Migration die z. T. beträchtliche Anzahl lokaler Katalogisate auf tatsächlich benötige Datensätze zu reduzieren.

Inzwischen können Bibliotheken, die entsprechenden Bedarf anmelden, mit einer tabellarischen Übersicht über ihre lokalen Katalogisate versorgt werden. Damit jede Bibliothek ausreichend Zeit für eine Datenanalyse und -bereinigung hat, bevor das Transferverfahren für den betreffenden Standort durchgeführt wird, sollen diese Aktivitäten mit dem Projektteam für die technische Umsetzung koordiniert werden.

Im zeitlichen Rahmen der Arbeit ließen sich nicht alle für das Projekt zur Migration lokaler Katalogisate notwendigen oder wünschenswerten Arbeitspakete umsetzen. Zum einen sollte der Programmlaufzeit der eigentlichen Datenmigration besondere Beachtung geschenkt werden, hierzu wurden einige Ideen zur Laufzeitverbesserung skizziert. Zum anderen wurden problematische Datenkonstellationen aufgedeckt, die beim Migrationsablauf zu Datenbankinkonsistenzen führen können. Die betroffenen Datensätze müssen für eine Fehlerbehandlung identifiziert werden. Es bietet sich an, die Funktionalität des Perl-Programms `opc4_lok_titel.pl` um weitere Funktionen zu erweitern (z. B. Feldstatistik auf Subfield-Ebene, oder die Protokollierung von Datensätzen mit Verknüpfungen zu anderen Datensätzen, da diese Links im CBS nicht u. U. nicht mehr konsistent sind).

Auch nach Abschluss des Transferverfahrens für die LBS-Bibliotheken im GBV wird die VZG für die Bibliotheken bei weiteren Projekten zur Migration und Bereinigung von Metadaten Unterstützung leisten müssen. Hierzu werden die im Rahmen dieser Arbeit erworbenen Kenntnisse des wissenschaftlichen Kontexts von ETL-Prozessen genauso wie die in der Praxis eines Migrationskonzept erworbenen Erfahrungen und Fertigkeiten sowie die entwickelten Programme nützlich sein.

Literaturverzeichnis

AG Kooperative Verbundanwendungen der AG der Verbundsysteme (Hg.) [2014]: Vereinbarungen zum Datentausch in MARC 21, 2014. http://www.dnb.de/SharedDocs/Downloads/DE/DNB/wir/marc21VereinbarungDatentauschTeil1.pdf?__blob=publicationFile (16.05.2018).

Albrecht, Jörg [2012]: Integriertes elektronisches Bibliothekssystem. In: Erdmute Lapp (Hg.): *Die Bibliothek als Erfolgsfaktor.* Die Universitätsbibliothek Bochum nach 50 Jahren : 1962-2012. Ruhr-Universität Bochum. Bochum, 2012, S. 72–76. https://repo.ub.rub.de/bibliographie/110791991/Festschrift.pdf (15.05.2018).

Alpar, Paul; Alt, Rainer; Bensberg, Frank et al. [2014]: Anwendungsorientierte Wirtschaftsinformatik: Strategische Planung, Entwicklung und Nutzung von Informationssystemen. 7., aktual. u. erw. Aufl. Wiesbaden: Springer Vieweg, 2014. http://dx.doi.org/10.1007/978-3-658-00521-4 (16.05.2018).

Apel, Detlef; Behme, Wolfgang; Eberlein, Rüdiger et al. [2015]: Datenqualität erfolgreich steuern: Praxislösungen für Business-Intelligence-Projekte. 3., überarb. und erw. Aufl. Heidelberg: dpunkt.verlag, 2015 (Edition TDWI).

Arbeitsstelle für Standardisierung (AfS) [2004]: 9. Sitzung des Standardisierungsausschusses am 15. Dezember 2004: Protokoll, 2004. http://www.dnb.de/SharedDocs/Downloads/DE/DNB/standardisierung/protokolle/pSta20041215v.pdf?__blob=publicationFile (15.05.2018).

Avram, Henriette D. [1975]: MARC: Its history and impllications. Washington, DC: Library of Congress, 1975. https://babel.hathitrust.org/cgi/pt?id=mdp.39015034388556 (16.05.2018).

Becker, Markus [2018]: MARC-Export aus BIBLIOTHECA. Duisburg, 02.01.2018. E-Mail (Persönliche Kommunikation).

Bissegger, Judith; Wittwer, Barbara [2016]: Metadatenmanagement: Die ETH-Bibliothek beschreitet neue Wege. BIS-Kongress. Luzern, 01.09.2016. http://www.slideshare.net/ETH-Bibliothek/metadatenmanagement-die-ethbibliothek-beschreitet-neue-wege (16.05.2018).

Bleiholder, Jens; Schmid, Joachim [2015]: Datenintegration und Deduplizierung. In: Knut Hildebrand, Marcus Gebauer et al. (Hg.): *Daten- und Informationsqualität.* Auf dem Weg zur Information Excellence. 3., erweiterte Auflage. Wiesbaden: Springer Vieweg, 2015, S. 121–140.

Block, Barbara [2007]: Umstieg nach MARC 21: Bericht aus dem GBV. Firmenworkshop der DNB. Deutsche Nationalbibliothek. Frankfurt am Main, 26.09.2007. http://www.dnb.de/SharedDocs/Downloads/DE/DNB/standardisierung/blockMarcWorkshopGbvFormatumstieg2007.pdf?__blob=publicationFile (16.05.2018).

Block, Brigitte [2017]: Verfahren für die Dublettenzusammenführung bei der Einspielung von Daten in die hbz-Verbunddatenbank (M+M-Verfahren). hbz, 2017. https://wiki1.hbz-nrw.de/download/attachments/141721616/Verfahren_Dublettenzusammenfuehrung_2017_M%C3%A4rz_externe+Version+ohne+Fu%C3%9Fzeile.pdf (15.05.2018).

Böhme, Christoph [2013]: Analysis of library metadata with Metafacture. SWIB13. hbz; ZBW, 25.11.2013. http://swib.org/swib13/slides/boehme_swib13_131.pdf (15.05.2018).

Bossers, Anton [2005]: Samenwerkende bibliothecarissen en technische innovaties: Pica van 1969 tot 2002, 2005. http://www.oclcpica.org/content/1496/pdf/Picavan1969tot2002.pdf (16.05.2018).

Brodie, Michael L.; Stonebraker, Michael [1993]: DARWIN: On the Incremental Migration of Legacy Information Systems. University of California, Berkeley, CA, 1993 (Technical Report, TR-0222-10-92-165). http://db.cs.berkeley.edu/papers/S2K-93-25.pdf (15.05.2018).

Brodie, Michael L.; Stonebraker, Michael [1995]: Migrating Legacy Systems: Gateways, Interfaces & the Incremental Approach. San Francisco, CA: Kaufmann Publ, 1995.

Cavegn-Pfister, Erica; Wirth, Andrea; Wittwer, Barbara et al. [2018]: Metadatenmanagement – Wie die ETH-Bibliothek ein neu entstandenes Arbeitsfeld bedient.
In: *arbido: Die Fachzeitschrift für Archiv, Bibliothek und Dokumentation* (2018), Nr. 3.
http://arbido.ch/en/ausgaben-artikel/2018/metadaten-datenqualit%C3%A4t/metadatenmanagement-wie-die-eth-bibliothek-ein-neu-entstandenes-arbeitsfeld-bedient (16.05.2018).

Christof, Jürgen [2017]: Der Umstieg der Berliner Universitätsbibliotheken auf Alma: eine Zwischenbilanz. 106. Deutscher Bibliothekartag. Frankfurt am Main, 01.06.2017.
https://www.slideshare.net/UB_TU_Berlin/der-umstieg-der-berliner-universittsbibliotheken-auf-alma-eine-zwischenbilanz (16.05.2018).

Contessi, Angela; Gadea Raga, Alejandro [o.J.]: The future of the MARC: a conversation with Sally H. McCallum, o.J.
https://gumarc21.unicatt.it/progetti-milan-Belotti_eng.pdf (15.05.2018).

Deutsche Nationalbibliothek [2013]: Umstieg von MAB2 auf MARC 21: MARC-21-Anwenderebene: Festlegungen in MARC 21 für den deutschsprachigen Raum, 2013.
http://www.dnb.de/SharedDocs/Downloads/DE/DNB/standardisierung/marc21FormatumstiegAnwenderebeneDeutscherRaum2013.pdf?__blob=publicationFile (15.05.2018).

Deutsche Nationalbibliothek [2018]: Standardelemente-Set für den deutschsprachigen Raum, 2018.
https://wiki.dnb.de/download/attachments/114430616/Standardelemente-Set_Titeldaten.pdf (16.05.2018).

Deutsches Bibliotheksinstitut / Kommission für Alphabetische Katalogisierung [1983]: Regeln für wissenschaftliche Bibliotheken: RAK-WB. Unter Mitarbeit von Franz Georg Kaltwasser und Irmgard Bouvier. Wiesbaden: Reichert, 1983 (Regeln für die alphabetische Katalogisierung, 1).

Die Beauftragte der Bundesregierung für Informationstechnik [2012]: Migrationsleitfaden: Leitfaden für die Migration von Software. Version 4.0, 2012.
http://www.cio.bund.de/SharedDocs/Publikationen/DE/Architekturen-und-Standards/migrationsleitfaden_4_0_download.pdf?__blob=publicationFile (15.05.2018).

Diedrichs, Reiner; Sandholzer, Ute [2018]: 25 Jahre Katalogisierung im Pica-CBS im GBV.
In: *VZG aktuell* (2018), Nr. 1, S. 4–7.
https://www.gbv.de/Verbundzentrale/Publikationen/broschueren/vzg-aktuell/VZG_Aktuell_2018_01.pdf (16.05.2018).

Dippold, Rolf; Meier, Andreas; Schnider, Walter et al. [2005]: Unternehmensweites Datenmanagement: Von der Datenbankadministration bis zum Informationsmanagement. 4. Aufl. Wiesbaden: Vieweg+Teubner, 2005.
http://dx.doi.org/10.1007/978-3-322-86870-1 (15.05.2018).

Edwards, Laura; King, Todd; Smith, Kelly [2017]: Navigating WMS Analytics for the Right Data. OCLC Global Community and User Group Meeting. Dublin, Ohio, 27.09.2017.
https://encompass.eku.edu/context/fs_research/article/1192/type/native/viewcontent (16.05.2018).

Ex Libris [2017a]: Alma, Primo/Summon Implementation Methodology, 2017.
https://knowledge.exlibrisgroup.com/@api/deki/files/53786/Alma%252C_Primo-Summon_Implementation_Methodology.pdf?revision=1 (15.05.2018).

Ex Libris [2017b]: Analytics, 2017.
https://knowledge.exlibrisgroup.com/@api/deki/files/40872/Alma_Analytics_Guide.pdf (15.05.2018).

Ex Libris [2017c]: Getting Ready for Alma and Discovery Implementation, 2017.
https://knowledge.exlibrisgroup.com/@api/deki/pages/43963/pdf/Getting%2bReady%2bfor%2bAlma%2band%2bDiscovery%2bImplementation.pdf?stylesheet=default (15.05.2018).

Ex Libris [2018a]: Running Manual Jobs on Defined Sets, 2018.
https://knowledge.exlibrisgroup.com/@api/deki/pages/33393/pdf/Running%2bManual%2bJobs%2bon%2bDefined%2bSets.pdf?stylesheet=default (15.05.2018).

Ex Libris [2018b]: Alma Network Zone Configuration for Integration with Aleph Members, 2018. https://knowledge.exlibrisgroup.com/@api/deki/pages/56675/pdf/Alma%2bNetwork%2bZone%2bConfiguration%2bfor%2bIntegration%2bwith%2bAleph%2bMembers.pdf?stylesheet=default (15.05.2018).

Ex Libris [2018c]: Arbeiten mit Titeldatensätzen, 2018. https://knowledge.exlibrisgroup.com/@api/deki/pages/33834/pdf/Arbeiten%2bmit%2bTiteldatens%25C3%25A4tzen.pdf?stylesheet=default (14.05.2018).

Experian Marketing Service [2016]: Datenqualität und -management: Trends 2016. Düsseldorf, 2016. http://www.experian.de/assets/marketing-services/white-papers/data-management-trend-report-2016-DE.pdf (15.05.2018).

Farkisch, Kiumars [2011]: Data-Warehouse-Systeme kompakt: Aufbau, Architektur, Grundfunktionen. Berlin, Heidelberg: Springer, 2011 (Xpert.press). http://dx.doi.org/10.1007/978-3-642-21533-9 (15.05.2018).

Frans, Wilfred [2014]: Outsourcing Local Titles Manual. [Interne Systemdokumentation]. OCLC, 2014.

Furrie, Betty [2009]: Understanding MARC bibliographic: Machine-readable cataloging. 8. Aufl. Washington, DC, 2009. http://www.loc.gov/marc/umb/ (16.05.2018).

Glöckle, Herbert [2007]: IT-Integration und Migration - Konzepte und Vorgehensweisen. In: *HMD* 44 (2007), Nr. 5, S. 7–19. http://dx.doi.org/10.1007/BF03341120 (15.05.2018).

Hartmann, Sarah; Schulze, Francesca [2013]: Das Europeana Data Model im Kontext der Deutschen Digitalen Bibliothek. KIM Workshop. Mannheim, 25.03.2013. https://wiki.dnb.de/download/attachments/66785366/DDB_KIM-WS2013_final.pptx (16.05.2018).

Helmkamp, Kerstin; Oehlschläger, Susanne [2008]: Firmenworkshop „Umstieg auf MARC 21“: Workshop an der Deutschen Nationalbibliothek am 26. September 2007. In: *Dialog mit Bibliotheken* 20 (2008), Nr. 1, S. 25–28. http://nbn-resolving.de/urn:nbn:de:101-2016110330 (16.05.2018).

Hemme, Felix [2017]: Metadatenmanagement in der ZBW. Kiel, 15.12.2017. E-Mail (Persönliche Kommunikation).

Hildebrand, Knut; Gebauer, Marcus; Hinrichs, Holger; Mielke, Michael (Hg.) [2015]: Daten- und Informationsqualität: Auf dem Weg zur Information Excellence. 3., erweiterte Auflage. Wiesbaden: Springer Vieweg, 2015. http://dx.doi.org/10.1007/978-3-658-09214-6 (16.05.2018).

Hochstenbach, Patrick [2017]: Catmandu Fixes: Cheat Sheet, 2017. http://librecat.org/assets/catmandu_cheat_sheet.pdf (16.05.2018).

Horn, Moritz; Kemner-Heek, Kirstin [2017]: Überlegungen zu ERM im GBV. 2. Systemverwalter-Workshop. Verbundzentrale des GBV. Göttingen, 26.09.2017. https://www.gbv.de/Verbundzentrale/Publikationen/publikationen-der-vzg-2017/pdf/horn_kemner_2692017_ERMimGBV_LBS-Workshop_Goettingen.pdf (16.05.2018).

Humm, Bernhard; Wietek, Frank [2005]: Architektur von Data Warehouses und Business Intelligence Systemen. In: *Informatik Spektrum* 28 (2005), Nr. 1, S. 3–14. http://dx.doi.org/10.1007/s00287-004-0450-5 (15.05.2018).

Hunter, Jane L. [2003]: A Survey of Metadata Research for Organizing the Web. In: *Library Trends* 52 (2003), Nr. 2, Fall 2003, S. 318–344. http://hdl.handle.net/2142/8529 (15.05.2018).

Inmon, William H. [2005]: Building the data warehouse. 4th edition. Indianapolis, IN: Wiley, 2005 (Wiley technology publishing).

Inmon, William H.; Hackathorn, Richard D. [1994]: Using the Data Warehouse. New York, NY: Wiley, 1994 (A Wiley-QED publication).

Instruktionen für die alphabetischen Kataloge der preussischen Bibliotheken und für den preussischen Gesamtkatalog: Vom 10. Mai 1899 [1899]. Berlin: Asher, 1899. http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:bvb:12-bsb00078903-8 (16.05.2018).

Jele, Harald [2009]: Erkennung bibliographischer Dubletten mittels Trigrammen: Messungen zur Performanz. Teil 1. In: *B.I.T. online* 12 (2009), Nr. 3, S. 265–272. http://www.b-i-t-online.de/heft/2009-03/fach2.htm (15.05.2018).

Jong, L. de [2004]: Pica Character Set: Document Type. OCLC, 2004. https://wiki-cbs.oclc.org/wiki/images/Picacharset.pdf (15.05.2018).

Kann, Bettina [2017]: Alma im Österreichischen Bibliothekenverbund: Werkstattbericht. In: *Bibliotheksdienst* 51 (2017), Nr. 7, S. 562–574. http://dx.doi.org/10.1515/bd-2017-0061 (16.05.2018).

Kaps, Stephan [2017]: Migrationsstrategien im Vergleich: Orientierungshilfe. In: *iX Developer - Altlasten im Griff*. Legacy-Software erhalten, 2017, S. 104–108.

Kemner-Heek, Kirstin [2016]: Von OLE zu FOLIO. In: *VZG aktuell* (2016), Nr. 3, S. 16–17. https://www.gbv.de/Verbundzentrale/Publikationen/broschueren/vzg-aktuell/VZG_Aktuell_2016_03.pdf (16.05.2018).

Kemner-Heek, Kirstin; Schomburg, Silke [2017]: FOLIO: Evaluation einer Innocation. 106. Deutscher Bibliothekartag. Frankfurt am Main, 30.05.2017. http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:0290-opus4-29382 (16.05.2018).

Kemper, Hans-Georg; Baars, Henning; Mehanna, Walid [2010]: Business Intelligence - Grundlagen und praktische Anwendungen: Eine Einführung in die IT-basierte Managementunterstützung. 3., überarbeitete und erweiterte Auflage. Wiesbaden: Vieweg+Teubner, 2010 (Studium). http://dx.doi.org/10.1007/978-3-8348-9727-5 (15.05.2018).

Kempter, Hubert [2017]: Betriebliche Informationssysteme: Datenmanagement und Datenanalyse. Stuttgart: Verlag W. Kohlhammer, 2017 (BWL Bachelor Basics).

Kimball, Ralph; Caserta, Joe [2004]: The data warehouse ETL toolkit: Practical techniques for extracting, cleaning, conforming, and delivering data. Indianapolis, IN: Wiley, 2004.

Klettke, Meike; Thalheim, Bernhard [2011]: Evolution and Migration of Information Systems. In: David W. Embley u. Bernhard Thalheim (Hg.): *Handbook of Conceptual Modeling*. Berlin, Heidelberg: Springer, 2011, S. 381–419. http://dx.doi.org/10.1007/978-3-642-15865-0_12 (15.05.2018).

Klute, Uschi [2017]: Lokale Katalogisierung im CBS: Ablösung von CAT4. [GBV-interne Veröffentlichung]. 2. Systemverwalter-Workshop. Verbundzentrale des GBV. Göttingen, 26.09.2017. https://info.gbv.de/download/attachments/61374507/Klute-LOK-Transfer.pptx (15.05.2018).

Klute, Uschi [2018]: Lokale Katalogisate im LBS löschen: Anleitung zur Bereinigung per WinIBW-Script. [unveröffentlicht]. Verbundzentrale des GBV, 2018 (15.05.2018).

Köpf, Silvia [2017]: Alma-Implentierung im OBV: Kohorte 1. 33. Österreichischer Bibliothekartag. Linz, 13.09.2017. http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:0290-opus4-32637 (16.05.2018).

Köppen, Veit; Saake, Gunter; Sattler, Kai-Uwe [2014]: Data Warehouse Technologien. 2. Aufl. Heidelberg: mitp, 2014.

Kortick, Yoel [2015]: Overview and Introduction to Alma Analytics. Ex Libris, 2015. https://knowledge.exlibrisgroup.com/@api/deki/files/57113/Analytics_-_Overview_and_introduction_to_Alma_Analytics.pptx (16.05.2018).

Leser, Ulf; Naumann, Felix [2007]: Informationsintegration: Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen. 1. Aufl. Heidelberg: dpunkt.verlag, 2007.

Lohmeier, Felix [2017]: Skript zum Seminar "(Open) Discovery: Wir bauen uns einen Bibliothekskatalog": (WS 2016/17), 2017.

https://www.gitbook.com/download/pdf/book/felixlohmeier/seminar-wir-bauen-uns-einen-bibliothekskatalog (16.05.2018).

Lohrum, Stefan; Schneider, Wolfram; Willenborg, Josef [1999]: De-duplication in KOBV. Konrad-Zuse-Zentrum für Informationstechnik, 1999 (ZIB-Report, SC 99-05). http://nbn-resolving.de/urn:nbn:de:0297-zib-3931 (15.05.2018).

Lüssem, Jens; Harrach, Hakim [2013]: How to make data migration processes more efficient by using TOGAF: Best practice data migration approach applied to SAP Financial Services-Policy Management. In: *2013 ACS International Conference on Computer Systems and Applications (AICCSA)*. Ifrane, Morocco, 27.-30.05.2013, 2013. http://dx.doi.org/10.1109/AICCSA.2013.6616436 (16.05.2018).

Mandl, Peter [2009]: Masterkurs Verteilte betriebliche Informationssysteme: Prinzipien, Architekturen und Technologien. Wiesbaden: Vieweg+Teubner, 2009. http://dx.doi.org/10.1007/978-3-8348-9262-1 (15.05.2018).

Masak, Dieter [2006]: Legacysoftware: Das lange Leben der Altsysteme ; mit 39 Tabellen. Berlin, Heidelberg: Springer, 2006 (Xpert.press). http://dx.doi.org/10.1007/3-540-30320-0 (16.05.2018).

Mauder, Ines [2018]: BibControl in der ZBW. Hamburg, 24.01.2018. Gespräch (Persönliche Kommunikation).

Meßmer, Gabriele; Müller, Manfred [2014]: Standards in der Formalerschließung gedruckter und elektronischer Ressourcen. In: Rolf Griebel, Hildegard Schäffler et al. (Hg.): *Praxishandbuch Bibliotheksmanagement*. Berlin: de Gruyter Saur, 2014 (De Gruyter Reference), S. 341–356. http://dx.doi.org/10.1515/9783110303261.341 (16.05.2018).

Mönkediek, Yvonne [2018]: Abläufe bei der Integration heterogener Metadaten für POLLUX. Bremen, 16.03.2018. Telefonat (Persönliche Kommunikation).

OCLC [1995]: LBS OWC Implementation. Unveröffentlichtes Dokument. Leiden, 1995 (DE 054/0195).

OCLC [2011]: PSI XML Interface. [Interne Systemdokumentation], 2011.

OCLC [2018a]: Cataloging/Collection reports, 2018. https://help.oclc.org/@api/deki/pages/362/pdf/Cataloging%252FCollection%2breports.pdf?stylesheet=default (15.05.2018).

OCLC [2018b]: Available reports for WMS users, 2018. https://help.oclc.org/@api/deki/pages/4359/pdf/Available%2breports%2bfor%2bWMS%2busers.pdf?stylesheet=default (15.05.2018).

Opitz, Daniel; Haake, Elmar [2017]: Because the night… is full of errors: Effizientes Metadatenmanagement mit Nightwatch. 106. Deutscher Bibliothekartag. Frankfurt am Main, 01.06.2017. http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:0290-opus4-30935 (16.05.2018).

Pfeffer, Magnus [2016]: Open Source Software zur Verarbeitung und Analyse von Metadaten. LIS Workshop. 105. Deutscher Bibliothekartag in Leipzig 2016 = 6. Bibliothekskongress. Leipzig, 16.03.2016. http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:0290-opus4-24490 (15.05.2018).

Pfister, Erica; Wittwer, Barbara; Wolff, Marianne [2017]: Metadaten - Manuelle Datenpflege vs. Automatisieren: Ein Praxisbericht zu Metadatenmanagement an der ETH-Bibliothek. In: *B.I.T. online* 20 (2017), Nr. 1, S. 22–25. http://b-i-t-online.de/heft/2017-01/nachrichtenbeitrag-pfister.pdf (16.05.2018).

DIN 69901-5:2009-01: Projektmanagement - Projektmanagementsysteme - Teil 5: Begriffe.

Rahm, Erhard; Do, Hong Hai [2000]: Data Cleaning: Problems and Current Approaches. In: *Bulletin of the Technical Committee on Data Engineering* 23 (2000), Nr. 4, S. 3–13. http://sites.computer.org/debull/A00dec/A00DEC-CD.pdf (16.05.2018).

Reichart, Markus; Mönnich, Michael W. [1994]: Dublettenkontrolle in bibliographischen Datenbanken. In: *BIBLIOTHEK Forschung und Praxis* 18 (1994), Nr. 2, S. 193–216. http://dx.doi.org/10.1515/bfup.1994.18.2.193 (03.02.2018).

Reiß, Michael [1993]: Komplexitätsmanagement: (I).
In: *Das Wirtschaftsstudium* 22 (1993), S. 54–60.
http://dx.doi.org/10.18419/opus-5594 (15.05.2018).

Rohweder, Jan P.; Kasten, Gerhard; Malzahn, Dirk et al. [2015]: Informationsqualität – Definitionen, Dimensionen und Begriffe.
In: Knut Hildebrand, Marcus Gebauer et al. (Hg.): *Daten- und Informationsqualität.* Auf dem Weg zur Information Excellence. 3., erweiterte Auflage. Wiesbaden: Springer Vieweg, 2015, S. 25–46.
http://dx.doi.org/10.1007/978-3-658-09214-6_2 (15.05.2018).

Rolschewski, Johann [2018]: Verwendung von Catmandu in der ZDB. Kiel, 20.02.2018. E-Mail (Persönliche Kommunikation).

Rossak, Ines (Hg.) [2013]: Datenintegration: Integrationsansätze, Beispielszenarien, Problemlösungen, Talend Open Studio. München: Carl Hanser Verlag, 2013.
http://dx.doi.org/10.3139/9783446434912 (16.05.2018).

Rühle, Stefanie [2012]: Kleines Handbuch Metadaten. KIM Kompetenzzentrum Interoperable Metadaten, 2012.
http://www.kim-forum.org/Subsites/kim/SharedDocs/Downloads/DE/Handbuch/metadaten.pdf?__blob=publicationFile (16.05.2018).

Rusch, Beate [1999]: Normierungen von Zeichenfolgen als erster Schritt des Match: Zur Dublettenbehandlung im Kooperativen Bibliotheksverbund Berlin-Brandenburg. Konrad-Zuse-Zentrum für Informationstechnik, 1999 (ZIB-Report, SC 99-13).
http://nbn-resolving.de/urn:nbn:de:0297-zib-4010 (15.05.2018).

Rzehak, Michael [2018]: Konvertierung von Daten aus BIBLIOTHECA. Göttingen, 18.01.2018. E-Mail (Persönliche Kommunikation).

Scholz, Stephani; Labner, Joseph [2014]: Konvertierung MARC21 - ASEQ (MAB2): Ein kooperatives Projekt der Aleph-Verbünde BVB, hbz, KOBV und OBV. 103. Deutscher Bibliothekartag. Bremen, 06.06.2014.
https://wiki.dnb.de/download/attachments/94672183/MARC%2021_alephseq_MAB_Bibliothekartag%20Bremen%202014_CSS_jl.20140604.pdf (16.05.2018).

Schreib, Gernot [2013]: Applikationsanbindung an das Data Warehouse: ETL vs. ELT.
In: *DOAG News* (2013), Nr. 3, S. 41–45.
https://www.doag.org/formes/pubfiles/4820500/2013-03-News-Gernot-Schreib-Applikationsanbindung-an-das-Data-Warehouse--ETL-vs-ELT.pdf (16.05.2018).

Schulz, Dörthe [2018]: BibControl in der Jade Hochschule. Wilhelmshaven, 31.01.2018. Telefonat (Persönliche Kommunikation).

Smit, Paul; Sutherland, Ryan [2002]: CSDBC DataBase Conversion Library. [Interne Systemdokumentation]. Version 1.1. OCLC, 2002.

Sneed, Harry M.; Wolf, Ellen; Heilmann, Heidi [2010]: Softwaremigration in der Praxis: Übertragung alter Softwaresysteme in eine moderne Umgebung. 1. Aufl. Heidelberg: dpunkt.verlag, 2010.

Sutherland, Ryan [2010]: Tools for offline processing: User Manual. [Interne Systemdokumentation]. OCLC, 2010.

Sutherland, Ryan; Sieburgh, Wouter; van Zutphen, Hetty [2009]: Evaluation Manual: Developing the EVAL table. [Interne Systemdokumentation]. Version 6.0. OCLC. Leiden, 2009.
https://wiki-cbs.oclc.org/wiki/images/Evaluation_Manual.pdf (15.05.2018).

Szott, Sascha [2016]: Korrektur von Schreibfehlern in Anfragetermen: [unveröffentlichte Unterlagen zur Vorlesung Suchmaschinentechnologie], 2016 (18.03.2018).

Verbundzentrale des GBV [2006]: Datum und Selektionsschlüssel: 70xx, 2006 (Katalogisierungsrichtlinie für den GBV).
https://www.gbv.de/bibliotheken/verbundbibliotheken/02Verbund/01Erschliessung/02Richtlinien/01KatRicht/7001.pdf (15.05.2018).

Verbundzentrale des GBV [2011]: Verbuchungsnummer: 8200, 2011 (Katalogisierungsrichtlinie für den GBV).
https://www.gbv.de/bibliotheken/verbundbibliotheken/02Verbund/01Erschliessung/02Richtlinien/01KatRicht/8200.pdf (16.05.2018).

Verbundzentrale des GBV [2012]: Richtlinie zur Bearbeitung von Titelaufnahmen begrenzter Werke im GBVKat, 2012.
https://www.gbv.de/bibliotheken/verbundbibliotheken/02Verbund/01Erschliessung/02Richtlinien/03Bearbeitungsrichtlinie/pdf/Bearbeitungsrichtlinie.pdf (15.05.2018).

Verbundzentrale des GBV [2014]: Signatur: 710x, 2014 (Katalogisierungsrichtlinie für den GBV).
https://www.gbv.de/bibliotheken/verbundbibliotheken/02Verbund/01Erschliessung/02Richtlinien/01KatRicht/7100.pdf (16.05.2018).

Verbundzentrale des GBV [2016]: CBS4-Indexierung: Indexübersicht, 2016.
http://www.gbv.de/vgm/info/mitglieder/02Verbund/02Verbundsystem/01Indexierung/01Indexierung_1298.pdf (15.05.2018).

Verbundzentrale des GBV [2017]: Lokale Katalogisierung: LOK. [GBV-interne Dokumentation], 2017 (Katalogisierungsrichtlinie für den GBV).
https://info.gbv.de/download/attachments/162496517/Richtlinie-LOK_final.pdf (15.05.2018).

Verein zur Weiterentwicklung des V-Modell XT e.V. (Weit e.V.) [2017]: V-Modell XT: Das deutsche Referenzmodell für Systementwicklungsprojekte. Version 2.1. München, 2017.
http://ftp.tu-clausthal.de/pub/institute/informatik/v-modell-xt/Releases/2.1/V-Modell-XT-Gesamt.pdf (15.05.2018).

Voß, Jakob [2016]: Catmandu Documentation. Catmandu Hackaton. Berlin, 11.04.2016.
http://dx.doi.org/10.5281/zenodo.49439 (16.05.2018).

Wachter, Sabine; Zaelke, Thomas [2015]: Systemkonsolidierung und Datenmigration als Erfolgsfaktoren: HMD Best Paper Award 2014. Wiesbaden: Springer Vieweg, 2015 (essentials).
http://dx.doi.org/10.1007/978-3-658-11406-0 (16.05.2018).

Wang, Richard Y.; Strong, Diane M. [1996]: Beyond Accuracy: What Data Quality Means to Data Consumers.
In: *Journal of Management Information Systems* 12 (1996), Nr. 4, S. 5–33.
http://dx.doi.org/10.1080/07421222.1996.11518099 (16.05.2018).

Weigel, Niels [2015]: Datenqualitätsmanagement - Steigerung der Datenqualität mit Methode.
In: Knut Hildebrand, Marcus Gebauer et al. (Hg.): *Daten- und Informationsqualität.* Auf dem Weg zur Information Excellence. 3., erweiterte Auflage. Wiesbaden: Springer Vieweg, 2015, S. 69–86.
http://dx.doi.org/10.1007/978-3-658-09214-6_4 (16.05.2018).

Wiesenmüller, Heidrun; Horny, Silke [2017]: Basiswissen RDA. 2., überarbeitete und erweiterte Auflage. Berlin, Boston: De Gruyter, 2017.
http://dx.doi.org/10.1515/9783110544725 (15.05.2018).

Wu, Bing; Lawless, Deirdre; Bisbal, Jesus et al. [1997]: The Butterfly Methodology: A Gateway-free Approach for Migrating Legacy Information Systems.
In: *Proceedings / Third IEEE International Conference on Engineering of Complex Computer Systems.* Como, 08.-12.09.1997. Los Alamitos, CA: IEEE Computer Soc. Press, 1997, S. 200–205.
http://dx.doi.org/10.1109/ICECCS.1997.622311 (15.05.2018).

ZBW [2014]: Jahresbericht 2013, 2014 (Jahresbericht, 2013).
http://zbw.eu/econis-archiv/handle/11159/713 (15.05.2018).

ZBW [2017a]: In Bewegung, 2017 (Jahresbericht, 2016).
http://hdl.handle.net/11159/716 (15.05.2018).

ZBW [2017b]: Metadatenmanager/in: [Stellenausschreibung], 2017.
https://phpefi.schleswig-holstein.de/stellenausschreibungen/pdf.php?eid=5660 (15.05.2018).

Zeitschriftendatenbank [2015]: ZDB-Skripte für die WinIBW 3, 2015.
http://www.zeitschriftendatenbank.de/fileadmin/user_upload/ZDB/pdf/winibw/ZDB-Skripte_WinIBW3.7_Dokumentation.pdf (15.05.2018).

Zhu, Lihong; Spidal, Debra F. [2015]: Shared Integrated Library System Migration From a Technical Services Perspective.
In: *Technical Services Quarterly* 32 (2015), Nr. 3, S. 253–273.
http://dx.doi.org/10.1080/07317131.2015.1029844 (16.05.2018).

Zwirner, Marcus [2015]: Datenbereinigung zielgerichtet eingesetzt zur permanenten Datenqualitätssteigerung. In: Knut Hildebrand, Marcus Gebauer et al. (Hg.): *Daten- und Informationsqualität.* Auf dem Weg zur Information Excellence. 3., erweiterte Auflage. Wiesbaden: Springer Vieweg, 2015, S. 101–120. http://dx.doi.org/10.1007/978-3-658-09214-6_6 (16.05.2018).

Verzeichnis der Internetquellen

Die aufgeführten Webseiten wurden im Internetarchiv http://archive.org/web/ gespeichert, soweit technisch möglich.

Arbeitsgruppe Kooperative Verbundanwendungen [2017]: Home - Projekt Kooperative Verbundanwendungen, 2017.
https://info.gbv.de/display/KNEU/Home (14.05.2018).

Berufsverband Information Bibliothek [o.D.]: Alma im Verbund, o.D.
https://opus4.kobv.de/opus4-bib-info/solrsearch/index/search/searchtype/collection/id/16588 (15.05.2018).

Berufsverband Information Bibliothek [o.D.]: Alma in der Anwendung, o.D.
https://opus4.kobv.de/opus4-bib-info/solrsearch/index/search/searchtype/collection/id/16589 (15.05.2018).

Berufsverband Information Bibliothek [o.D.]: Die Qual der Wahl: Neue Bibliothekssysteme, o.D.
https://opus4.kobv.de/opus4-bib-info/solrsearch/index/search/searchtype/collection/id/16591 (15.05.2018).

Bibliotheksservice-Zentrum Baden-Württemberg [o.D.]: aDIS/BMS, o.D.
https://www.bsz-bw.de/bibliothekssysteme/adis/ (15.05.2018).

Bibliotheksservice-Zentrum Baden-Württemberg [o.D.]: Koha, o.D.
https://www.bsz-bw.de/bibliothekssysteme/koha.html (15.05.2018).

Bibliotheksservice-Zentrum Baden-Württemberg [2017]: MAB2, 2017.
https://wiki.bsz-bw.de/doku.php?id=v-team:daten:datendienste:mab2 (14.05.2018).

Bibliotheksservice-Zentrum Baden-Württemberg [2018]: Open Access - Daten aus der SWB-Verbunddatenbank, 2018.
https://wiki.bsz-bw.de/doku.php?id=v-team:daten:openaccess:swb (14.05.2018).

Bibliotheksservice-Zentrum Baden-Württemberg; Verbundzentrale des GBV [o.D.]: K10plus – Kooperationsprojekt BSZ und GBV, o.D.
https://www.bszgbv.de/services/k10plus/ (14.05.2018).

Bibliotheksverbund Bayern [2016]: B3Kat - Open Data, 2016.
https://www.bib-bvb.de/web/b3kat/open-data (14.05.2018).

bvb-kobv-allianz [2015]: marcel: Datenbankschema, 2015.
https://github.com/bvb-kobv-allianz/marcel/wiki/Datenbankschema (15.05.2018).

bvb-kobv-allianz [2018]: marcel: Analysewerkzeug für MARCXML Dateien, 2018.
https://github.com/bvb-kobv-allianz/marcel (15.05.2018).

C# MARC Editor [o.D.].
https://csharpmarc.net/ (15.05.2018).

Cambridge University Press [o.D.]: cutover, o.D.
https://dictionary.cambridge.org/de/worterbuch/englisch/cutover (15.05.2018).

Catmandu-1.09 [2018].
https://metacpan.org/release/Catmandu (14.05.2018).

Catmandu::Breaker: Module version: 0.12 [o.D.].
https://metacpan.org/pod/Catmandu::Breaker (14.05.2018).

Catmandu::PICA: Module version: 0.25 [o.D.].
https://metacpan.org/pod/Catmandu::PICA (14.05.2018).

Catmandu::Stat: Module version: 0.13 [o.D.].
https://metacpan.org/pod/Catmandu::Stat (16.05.2018).

Chudnov, Daniel [2006]: unAPI version 1, 2006.
https://web.archive.org/web/20150117162606/http:/unapi.info/specs/ (15.05.2018).

Crossref [2018].
https://www.crossref.org/ (15.05.2018).

CrossRef [2018]: rest-api-doc: Crossref REST API, 2018.
https://github.com/CrossRef/rest-api-doc (15.05.2018).

Culturegraph [2017]: Plattform für Wissensvernetzung, 2017.
http://www.culturegraph.org/Subsites/culturegraph/DE/Home/home_node.html (15.05.2018).

d:swarm: Anwendungsfälle [o.D.].
http://www.dswarm.org/de/los-gehts/anwendungsfaelle/ (15.05.2018).

d:swarm [2016]: dswarm-documentation, 2016.
https://github.com/dswarm/dswarm-documentation/wiki (15.05.2018).

DATACOM Buchverlag GmbH [2012]: Legacy-System, 2012.
https://www.itwissen.info/Legacy-System-legacy-system.html (15.05.2018).

Deutsche Forschungsgemeinschaft (DFG) [2015]: Informationen zum Förderprogramm „Fachinformationsdienste für die Wissenschaft“, 2015.
http://www.dfg.de/foerderung/programme/infrastruktur/lis/lis_foerderangebote/fachinformationsdienste_wissenschaft/ (15.05.2018).

Deutsche Nationalbibliothek [o.D.]: Deutsche Digitale Bibliothek: Kultur und Wissen online, o.D.
https://www.deutsche-digitale-bibliothek.de/ (15.05.2018).

Deutsche Nationalbibliothek [o.D.]: Deutsche Digitale Bibliothek pro: Fragen & Antworten, o.D.
https://pro.deutsche-digitale-bibliothek.de/faq (15.05.2018).

Deutsche Nationalbibliothek [2016]: MAB, 2016.
http://www.dnb.de/DE/Standardisierung/Formate/MAB/mab_node.html (14.05.2018).

Deutsche Nationalbibliothek [2018]: Linked-Data-Service der Deutschen Nationalbibliothek, 2018.
http://www.dnb.de/DE/Service/DigitaleDienste/LinkedData/linkeddata_node.html (15.05.2018).

Deutsche Nationalbibliothek [2018]: RDA-Info, 2018.
https://wiki.dnb.de/display/RDAINFO/RDA-Info (14.05.2018).

Die Beauftragte der Bundesregierung für Informationstechnik [o.D.]: V-Modell XT, o.D.
https://www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/vmodell_xt_node.html (15.05.2018).

Ex Libris [o.D.]: Aleph: Integrated Library System, o.D.
http://www.exlibrisgroup.com/products/aleph-integrated-library-system/ (14.05.2018).

Ex Libris [o.D.]: Alma, o.D.
http://www.exlibrisgroup.com/products/alma-library-services-platform/ (14.05.2018).

FOLIO - The Future of Libraries is Open [o.D.].
https://www.folio.org/ (15.05.2018).

gbv [2018]: Catmandu-PICA: Add Fix command pica_remove, 2018.
https://github.com/gbv/Catmandu-PICA/issues/60 (14.05.2018).

GOKb: The Global Open Knowledgebase [o.D.].
https://gokb.org/ (15.05.2018).

Goyvaerts, Jan [2016]: RegEx Buddy Screen Shots: Create, Edit, Test, Debug, Convert, Save and Use Regex, 2016.
http://www.regexbuddy.com/index.html (14.05.2018).

Goyvaerts, Jan [2018]: EditPad - Text Editor für Windows - Deutsche Version, 2018.
https://www.editpadpro.com/de.html (14.05.2018).

hbz; Verbundzentrale des GBV [o.D.]: Das Projekt OLE/FOLIO, o.D.
https://www.folio-bib.org/ (15.05.2018).

Hochstenbach, Patrick [2017]: Catmandu - Cheat Sheet, 2017.
http://librecat.org/catmandu/2013/06/21/catmandu-cheat-sheet.html (14.05.2018).

How Long For New Feature [2017].
http://dilbert.com/strip/2017-02-22 (15.05.2018).

International Organization for Standardization [2008]: ISO 2709:2008: Information and documentation - Format for information exchange, 2008
https://www.iso.org/standard/41319.html (14.05.2018).

International Organization for Standardization [2013]: ISO 25577:2013: Information and documentation -- MarcXchange, 2013.
https://www.iso.org/standard/62878.html (14.05.2018).

JISC [o.D.]: Knowledge Base+, o.D.
https://www.kbplus.ac.uk/kbplus/ (15.05.2018).

k-int [2015]: gokb-phase1: API, 2015.
https://github.com/k-int/gokb-phase1/wiki/API (15.05.2018).

KOBV-Zentrale [o.D.]: Analysetool MABLE+, o.D.
https://www.kobv.de/entwicklung/software/mable/ (15.05.2018).

KOBV-Zentrale [o.D.]: Analysetool MARCEL, o.D.
https://www.kobv.de/entwicklung/software/marcel/ (15.05.2018).

KOBV-Zentrale [o.D.]: MABLE+: Dokumentation, o.D.
https://www.kobv.de/entwicklung/software/mable/dokumentation/ (15.05.2018).

KOBV-Zentrale [o.D.]: Regionale Recherche im "KOBV-Portal", o.D.
https://www.kobv.de/services/recherche/portal/ (15.05.2018).

Koha Community [2018]: Cataloging — Koha Manual 17.11 documentation, 2018.
https://koha-community.org/manual/17.11/en/html/06_cataloging.html (14.05.2018).

Koha Community [2018]: Koha Library Software, 2018.
https://koha-community.org/ (15.05.2018).

Library of Congress [o.D.]: Schema MARC21slim.xsd, o.D.
http://www.loc.gov/standards/marcxml/xml/spy/spy.html (14.05.2018).

Library of Congress [2016]: SRU: Search/Retrieval via URL, 2016.
http://www.loc.gov/standards/sru/ (14.05.2018).

Library of Congress Network Development and MARC Standards Office [1998]: USMARC and CAN/MARC Become MARC 21, 1998.
https://www.loc.gov/marc/annmarc21.html (14.05.2018).

Library of Congress Network Development and MARC Standards Office [2007]: The MARC 21 Formats: Background and Principles, 2007.
http://www.loc.gov/marc/96principl.html (15.05.2018).

LibreCat [2017]: Catmandu: the data processing toolkit, 2017.
http://librecat.org/ (14.05.2018).

LibreCat [2017]: Catmandu-MARC: Mapping Rules, 2017.
https://github.com/LibreCat/Catmandu-MARC/wiki/Mapping-rules (14.05.2018).

LibreCat [2017]: Distributions, 2017.
http://librecat.org/distributions.html (14.05.2018).

LibreCat [2017]: Use Cases buid Catmandu, 2017.
http://librecat.org/use-cases.html (14.05.2018).

LibreCat [2018]: Catmandu: Fixes Cheat Sheet, 2018.
https://github.com/LibreCat/Catmandu/wiki/Fixes-Cheat-Sheet (14.05.2018).

Lohmeier, Felix [o.D.]: [Projekte zu OpenRefine], o.D.
https://github.com/felixlohmeier?utf8=%E2%9C%93&tab=repositories&q=openrefine (15.05.2018).

lubuntu [o.D.].
https://lubuntu.net/ (14.05.2018).

MarcEditDevelopment [2018]: Current News, 2018.
http://marcedit.reeset.net/ (15.05.2018).

MarcEditDevelopment [2018]: Is MarcEdit Open Source?, 2018.
http://marcedit.reeset.net/is-marcedit-open-source (15.05.2018).

metafacture [2017]: metafacture-core: Plugins and Tools, 2017.
https://github.com/metafacture/metafacture-core/wiki/Plugins-and-Tools (15.05.2018).

metafacture [2018]: metafacture-core: Core package of the Metafacture tool suite for metadata processing, 2018.
https://github.com/metafacture/metafacture-core (15.05.2018).

Nasjonalbiblioteket Oslo [o.D.]: MARC-format, o.D.
https://bibliotekutvikling.no/ressurser/kunnskapsorganisering/verktoykasse-for-kunnskapsorganisering/marc-formater/ (14.05.2018).

National Diet Library [2018]: JAPAN/MARC Manual and Format, 2018.
http://www.ndl.go.jp/en/data/jm.html (14.05.2018).

OBVSG [2018]: Go-Live der Alma Network Zone, 2018.
https://www.obvsg.at/wir-ueber-uns/aktuelles/news/go-live-der-alma-network-zone/ (15.05.2018).

OCLC [2017]: Available standard reports, 2017.
https://help.oclc.org/Library_Management/WorldShare_Reports/Available_standard_reports (14.05.2018).

OCLC [2018]: BIBLIOTHECAplus: Software für Bibliotheken, die begeistern, 2018.
https://www.oclc.org/de/bibliotheca.html (14.05.2018).

OCLC [2018]: Lokales Bibliotheksverwaltungssystem mit integrierten Funktionen zum Katalogisieren und Bestellen, 2018.
https://www.oclc.org/de/lbs.html (14.05.2018).

OCLC [2018]: WorldShare Management Services: Eine integrierte Suite Cloud-basierter Anwendungen zur Bibliotheksverwaltung, 2018.
https://www.oclc.org/de/worldshare-management-services.html (14.05.2018).

OCLC [2018]: WorldShare Report Designer: Gestaltung maßgeschneiderter Berichte und Grafiken für Ihre Bibliothek, 2018.
https://www.oclc.org/de/worldshare-report-designer.html (14.05.2018).

OCLC WMS: harvested counter data needs to be available in analytics module [2017].
http://lists.eril-l.org/pipermail/eril-l-eril-l.org/2017-February/003108.html (14.05.2018).

OCLC-Developer-Network [2017]: MARCView-Convert, 2017.
https://github.com/OCLC-Developer-Network/MARCView-Convert (15.05.2018).

Open Archives Initiative [2016]: Open Archives Initiative Protocol for Metadata Harvesting, 2016.
https://www.openarchives.org/pmh/ (14.05.2018).

Open Library Foundation [2014]: Tutorial: GOKb Data Ingest Using OpenRefine, 2014.
https://openlibraryenvironment.atlassian.net/wiki/spaces/GOKB/pages/656219/Tutorial+GOKb+Data+Ingest+Using+OpenRefine (15.05.2018).

opencultureconsulting [2017]: openrefine-batch: OpenRefine batch processing (openrefine-batch.sh), 2017.
https://github.com/opencultureconsulting/openrefine-batch (15.05.2018).

open-data [2018]: solis-sofis, 2018.
https://git.gesis.org/open-data/solis-sofis (15.05.2018).

OpenRefine: A free, open source, powerful tool for working with messy data [2017].
http://openrefine.org/ (14.05.2018).

OpenRefine [2015]: OpenRefine: General Refine Expression Language, 2015.
https://github.com/OpenRefine/OpenRefine/wiki/General-Refine-Expression-Language (15.05.2018).

OpenRefine [2017]: OpenRefine: Architecture, 2017.
https://github.com/OpenRefine/OpenRefine/wiki/Architecture (15.05.2018).

OpenRefine [2018]: FAQ: Allocate More Memory, 2018.
https://github.com/OpenRefine/OpenRefine/wiki/FAQ%3A-Allocate-More-Memory (15.05.2018).

OpenRefine [2018]: OpenRefine: Home, 2018.
https://github.com/OpenRefine/OpenRefine/wiki (15.05.2018).

OpenRefine [2018]: OpenRefine: OpenRefine Database Import Extension #1394 · OpenRefine, 2018.
https://github.com/OpenRefine/OpenRefine/pull/1394 (15.05.2018).

OpenRefine [2018]: OpenRefine API, 2018.
https://github.com/OpenRefine/OpenRefine/wiki/OpenRefine-API (15.05.2018).

OpenRefine Foundation [2017]: OpenRefine Phase 1 enhancements, 2017.
https://docs.google.com/document/d/1UwoT1nFk9zwqSlH8rmqKPmiLS2Liw7-KM5HTKD2VVi8/edit#heading=h.2gazcsgmxkub (15.05.2018).

Oracle [2016]: Oracle Business Intelligence 12c, 2016.
http://www.oracle.com/technetwork/middleware/bi-enterprise-edition/overview/index.html (14.05.2018).

RefinePro [2018]: From Freebase Gridworks to Google Refine and now OpenRefine, 2018.
http://kb.refinepro.com/2012/10/from-freebase-gridworks-to-google.html (14.05.2018).

RefinePro [2018]: Prepare SQL update where query in OpenRefine, 2018.
http://kb.refinepro.com/2014/04/prepare-sql-update-query-in-openrefine.html (15.05.2018).

SAP AG [o.D.]: SAP Adaptive Server Enterprise, o.D.
https://www.sap.com/germany/products/sybase-ase.html (15.05.2018).

Slots- og Kulturstyrelsen [2017]: RDA, 2017.
https://slks.dk/om-slots-og-kulturstyrelsen/organisation/raad-naevn-og-udvalg/bibliografisk-raad/resource-description-and-access/ (14.05.2018).

SLUB Dresden [2016]: d:swarm: eine quelloffene Datenmanagementplattform für Wissensarbeiter, 2016.
http://www.dswarm.org/de/ (15.05.2018).

SLUB Dresden [2017]: [d:swarm - Demo-Installation], 2017.
http://demo.dswarm.org/#/data/ (15.05.2018).

sparkica [2013]: refine-stats, 2013.
https://github.com/sparkica/refine-stats (15.05.2018).

Stephens, Owen [2017]: Measuring scale/limits of OpenRefine, 2017.
https://groups.google.com/forum/#!msg/openrefine/-loChQe4CNg/eroRAq9_BwAJ (15.05.2018).

SUB Hamburg [2018]: Politikwissenschaft, 2018.
https://wikis.sub.uni-hamburg.de/webis/index.php/Politikwissenschaft (14.05.2018).

SuUB Bremen [2018]: POLLUX, 2018.
https://www.pollux-fid.de/ (15.05.2018).

tcbuzor [2018]: openrefine-db-extension: OpenRefine Database Extension, 2018.
https://github.com/tcbuzor/openrefine-db-extension (15.05.2018).

The British Library [2017]: British Library, MARC 21 and UKMARC, 2017.
http://www.bl.uk/bibliographic/nbsils.html (14.05.2018).

Triangle Solutions [2013]: BIB-Control: Profi-Controlling für die Bibliotheksleitung, 2013.
http://www.triangle-solutions.de/loesungen/tim4bib-biinderbibliothek/index.html (14.05.2018).

UKSG [o.D.]: KBART 5.3.1: data format, o.D.
https://www.uksg.org/kbart/s5/guidelines/data_format (15.05.2018).

Universität Bielefeld [o.D.]: Suchmaschine BASE: Bielefeld Academic Search Engine, o.D.
https://www.base-search.net/ (15.05.2018).

Verbundzentrale des GBV [o.D.]: Datenbanken, o.D.
https://www.gbv.de/gsomenu (14.05.2018).

Verbundzentrale des GBV [o.D.]: Informationen zur Zugangssoftware (WinIBW) und zum CBS, o.D.
https://www.gbv.de/bibliotheken/verbundbibliotheken/02Verbund/02Verbundsystem/02WinIBW (14.05.2018).

Verbundzentrale des GBV [o.D.]: Katalogisierungsrichtlinie für den GBV (RDA - Ansicht), o.D.
http://swbtools.bsz-bw.de/cgi-bin/help.pl?cmd=index&verbund=GBV&regelwerk=RDA (14.05.2018).

Verbundzentrale des GBV [o.D.]: Lukida, o.D.
https://www.lukida.org/ (15.05.2018).

Verbundzentrale des GBV [o.D.]: PICA XML, o.D.
http://format.gbv.de/pica/xml (14.05.2018).

Verbundzentrale des GBV [2014]: Katalogisierungsrichtlinie für den GBV, 2014.
https://www.gbv.de/bibliotheken/verbundbibliotheken/02Verbund/01Erschliessung/02Richtlinien/01KatRicht/inhalt.shtml (14.05.2018).

Verbundzentrale des GBV [2016]: Dubletten umlenken, 2016.
https://verbundwiki.gbv.de/display/VZG/Dubletten+umlenken (14.05.2018).

Verbundzentrale des GBV [2016]: Eingabehilfen für die Katalogisierung, 2016.
https://verbundwiki.gbv.de/pages/viewpage.action?pageId=16711726 (14.05.2018).

Verbundzentrale des GBV [2016]: unAPI, 2016.
https://verbundwiki.gbv.de/display/VZG/unAPI (15.05.2018).

Verbundzentrale des GBV [2017]: Excel-Tabelle erstellen, 2017.
https://verbundwiki.gbv.de/display/VZG/Excel-Tabelle+erstellen (15.05.2018).

Verbundzentrale des GBV [2017]: Katalogisierungsrichtlinie: Bibliographische Gattung und Status. 0500, 2017.
http://swbtools.bsz-bw.de/cgi-bin/help.pl?cmd=kat&val=0500&regelwerk=RDA&verbund=GBV (15.05.2018).

Verbundzentrale des GBV [2017]: SRU, 2017.
https://verbundwiki.gbv.de/display/VZG/SRU (15.05.2018).

Verbundzentrale des GBV [2018]: DAIA, 2018.
https://verbundwiki.gbv.de/display/VZG/DAIA (15.05.2018).

Verbundzentrale des GBV [2018]: LOK-Umzug (intern), 2018.
https://info.gbv.de/pages/viewpage.action?pageId=41557513#LOK-Umzug(intern)-StatistischeAuswertungderLokalenKatalogisateninallenLBS-Bibliotheken (15.05.2018).

Villanova University / Falvey Memorial Library [2018]: VuFind - Search. Discover. Share., 2018.
https://vufind.org/vufind/ (15.05.2018).

W3C [2017]: Extensible Markup Language (XML) 1.0 (Fifth Edition), 2017.
https://www.w3.org/TR/xml/ (14.05.2018).

Who needs MARC? [2011].
https://commonplace.net/2009/05/who-needs-marc/ (14.05.2018).

Anhänge

Anhang 1 Datenformat MARC 21

MARC 21-Datenstruktur[407]

Ein MARC 21-Datensatz ist in drei Abschnitte gegliedert:

- Satzkennung[408] (*Leader*)
- Inhaltsverzeichnis (*Directory*)
- Kontroll- und Datenfelder mit variabler Länge

Der Leader-Abschnitt hat eine feste Länge von 24 Zeichen und enthält codierte Angaben mit Informationen über den Aufbau des Datensatzes. Die Bedeutung eines Codes ist abhängig von dessen Position innerhalb des Leaders. So werden hier u. a. Datensatzlänge, Zeichencodierung und Materialtyp codiert. Auf den Positionen 12-16 ist die erste Zeichenposition des variablen Abschnitts für Kontroll- und Datenfelder angegeben.

Das Directory enthält für jedes enthaltene Kontroll- bzw. Datenfeld einen Block von 12 Zeichen, jeweils mit diesen Bestandteilen:

- Dreistellige numerische Feldnummer (Länge: 3 Bytes),
- Feldlänge (Länge: 4 Bytes),
- Anfangsposition des zugehörigen Datenfeldes
  relativ zum Beginn des ersten variablen Feldes, dessen Position im Leader vermerkt ist (Länge: 5 Bytes).

Den Abschluss bildet ein Feldende-Zeichen (*Field Terminator*, `0x1e`).

[407] Vgl. http://www.loc.gov/marc/96principl.html (15.05.2018) und Furrie 2009, Part XI.
[408] Auch: Vorspann.

Der Abschnitt mit den variablen Feldern beginnt mit Kontrollfeldern (Feldnummern[409] 001-008). Das erste Kontrollfeld (Feldnummer 001), die Kontrollnummer (*control_number_field*), entspricht i. d. R. der Datensatznummer der liefernden Institution. Weitere Kontrollfelder enthalten die ISILs der besitzenden Bibliotheken und den Zeitstempel des Datensatzes. Für jedes Kontrollfeld werden Inhalt und Feldende-Zeichen (0x1e) gespeichert.

Dahinter folgen die Datenfelder (Feldnummern 010-999[410]) mit den Kategorie-Inhalten des Katalogisats. Der Abschnitt für ein MARC-21-Datenfeld beginnt mit zwei feldspezifischen Indikatoren in der Länge je eines Zeichens, die die Bedeutung des Inhalts spezifizieren. So wird bei Feld 100 (Personenname als Haupteintragung) angegeben, ob es sich bei dem folgenden Inhalt um einen klassischen Familiennamen oder einen Persönlichen Namen handelt.

Die Inhalte der Datenfelder werden durch das genutzte Regelwerk für die Katalogisierung bestimmt. Der Inhalt eines Feldes kann sich auf mehrere Unterfelder mit feldspezifischer Bedeutung verteilen, diese werden mit einem Unterfeld-Code eingeleitet. So sind für Feld 100 u. a. Unterfeld $a für Nachname und/oder Vorname und Unterfeld $d für Datumsangaben wie Geburts- und Todesjahr vorgesehen.

Beispiel für den Inhalt der Verfasserangabe in Feld 100:

1#$a Chiang, Kai-shek,**$d**1887-1975.

Hier wird zur besseren Lesbarkeit ein im Datensatz vorhandenes Leerzeichen für den zweiten Indikator als # dargestellt, die Unterfeld-Einleitungszeichen (hier: $) sind im Datensatz mit 0x1f codiert. Die Feldnummer ist ausschließlich im Directory-Abschnitt des betreffenden Feldes gespeichert, nicht im zugehörigen Abschnitt des Datenfeldes.

Jeder Datenfeld-Abschnitt wird mit einem Feldende-Zeichen (0x1e) abgeschlossen. Bis auf wenige Ausnahmen sind alle Felder wiederholbar, können also in einem Datensatz mehrfach vorkommen. Das Ende eines Datensatzes wird mit einem Satzende-Zeichen (*Record Terminator*, 0x1d) markiert.

Somit stellt sich die Datensatzstruktur wie folgt dar[411]:

leader | directory | FT | control_number | FT | control_field_1 | FT | …
control_field_n | FT | data_field_1 | FT | … | data_field_n | FT | RT

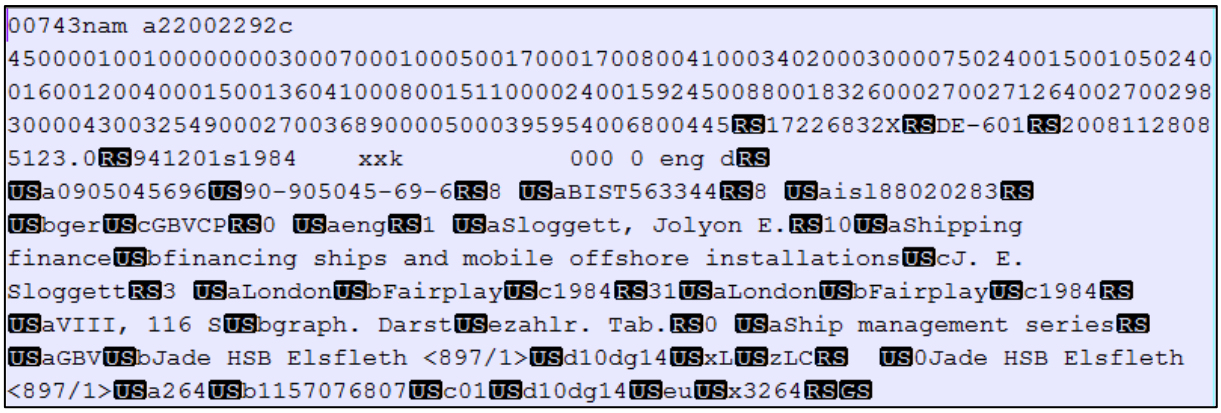
FT	*Field Terminator*	Feldende-Zeichen	0x1e
RT	*Record Terminator*	Datensatzende-Zeichen	0x1d

Diese Art der kompakten Speicherung ist aufgrund des optimierten Speicherplatzbedarfs für den Austausch großer Datenmengen gut geeignet.

[409] Engl.: *tag.*
[410] Das offizielle internationale Format sieht eine regionale Nutzung der Feldbereiche XX9, X9X (mit Ausnahme von Feld 490) und 912-924 für den D-A-CH-Raum vor. Zusätzlich sind die Felder 925-980 den regionalen Verbünden und weiteren Datenlieferanten wie DNB und ekz vorbehalten. Die Felder 980-999 fallen in die jeweilige Zuständigkeit des Datenlieferanten, können also lokal individuell belegt werden.
[411] Die Leerzeichen zwischen den einzelnen Elementen dienen lediglich der besseren Lesbarkeit.

Beispiele

Die nachfolgenden Abbildungen zeigen einen Beispieldatensatz (PPN 17226832X) aus der Verbunddatenbank des GBV in verschiedenen MARC-21-Ausgabevarianten.



Datenbeispiel 3: Speicherformat MARC 21 (per unAPI-Schnittstelle)[412]

Legende für den Screenshot aus Notepad++	RS	Feldende-Zeichen	(0x1e)
	US	Subfield-Kennzeichen	(0x1f)
	GS	Satzende-Zeichen	(0x1d)

```
xxxxxnam a22yyyyy2c 4500
001 17226832X
003 DE-601
005 20130328174229.0
008 941201s1984 xxk 000 0 eng d
020 $a0905045696$90-905045-69-6
024 8 $aBIST563344
024 8 $aisl88020283
035 $a(OCoLC)832268522
035 $a(DE-599)GBV17226832X
040 $bger$cGBVCP
041 0 $aeng
100 1 $aSloggett, Jolyon E.
245 10$aShipping finance$bfinancing ships and mobile offshore installations$cJ. E.
Sloggett
260 3 $aLondon$bFairplay$c1984
264 31$aLondon$bFairplay$c1984
300 $aVIII, 116 S$bgraph. Darst$ezahlr. Tab.
490 0 $aShip management series
900 $aGBV$bJade HSB Elsfleth <897/1>$d10dg14$xL$zLC
954 $0Jade HSB Elsfleth <897/1>$a264$b1157076807$c01$d10dg14$eu$x3264
```

Datenbeispiel 4: Textformat MARC 21 (WinIBW3, Format USX)

[412] http://unapi.gbv.de/?id=opac-de-897-1:ppn:17226832X&format=marc21 (23.02.2018).

```
LDR 00763nam a22002532c 4500
001     17226832X
003     DE-601
005     20171227133321.7
008     941201s1984    xxk           000 0 eng d
020    [a]: 0905045696
       [9]: 0-905045-69-6
024 8  [a]: BIST563344
024 8  [a]: isl88020283
035    [a]: (OCoLC)832268522
035    [a]: (DE-599)GBV17226832X
040    [b]: ger
       [c]: GBVCP
041 0  [a]: eng
100 1  [a]: Sloggett, Jolyon E.
245 10 [a]: Shipping finance
       [b]: financing ships and mobile offshore installations
       [c]: J. E. Sloggett
260 3  [a]: London
       [b]: Fairplay
       [c]: 1984
264 31 [a]: London
       [b]: Fairplay
       [c]: 1984
300    [a]: VIII, 116 S
       [b]: graph. Darst
       [e]: zahlr. Tab.
490 0  [a]: Ship management series
951    [a]: BO
980    [2]: 264
       [1]: 01
       [b]: 1157076807
       [d]: 10dg14
       [e]: u
       [l]: 06542
       [x]: 3264
       [y]: z
       [z]: 27-04-10
```

Datenbeispiel 5: Textformat MARC 21 (aus VZG-Datenexport)

Die Unterschiede in den beiden Textformat-Ausgaben sind blau unterlegt. Die Abweichungen betreffen Felder, die regionalen (Verbund-)Festlegungen (950–954) vorbehalten sind bzw. in die Zuständigkeit der jeweiligen Institution fallen (980–999).[413] Offenbar sind bei diesen beiden Ausgabeformaten unterschiedliche Mapping-Tabellen hinterlegt.

[413] Vgl. Deutsche Nationalbibliothek 2013, S. 2.

Anhang 2 Besonderheiten des PICA3-/PICA+-Formats

Die Datensatzstruktur hinsichtlich der Aufteilung in Titel-, Lokal- und Exemplarkategorien ist in der ersten Position des PICA+-Kategoriecodes abgebildet: Bei Feldern der Titelebene beginnt der Kategorie-Code mit 0, bei Feldern des Lokalbereichs mit 1 und bei Exemplarkategorien mit 2. Während für die ersten drei Positionen nur Ziffern zugelassen sind, sind für den Indikator an vierter Position alle Großbuchstaben sowie das At-Zeichen (@ = „Klammeraffe“) erlaubt.

Eine formale Zuordnung zum Datensatzlevel kann aus den PICA3-Kategoriebezeichnungen nicht geschlossen werden, auch entspricht die Anzeigereihenfolge in PICA3 nicht der numerischen Ordnung.[414]

PICA+	PICA3	Inhalt	Level[415]
011@	1100	Erscheinungsjahr	0
028C/00	3100	Person/Familie als 2. und weiterer geistiger Schöpfer, sonstige Personen/Familien, die mit dem Werk in Verbindung stehen, Mitwirkende, Hersteller, Verlage, Vertriebe	0
147B/05	4715	Allgemeiner Hinweis auf Sekundärausgaben (lokal)	1
145Z/00	6000	Lokale Notation	1
220B/01	4802	Exemplarbezogener Kommentar	2
209A/01	7100	ELN, Exemplarzahl, Sonderstandort, Signatur, Ausleihindikator, Konvolutindikator	2

Tab. 12: Beispiele für PICA3- und PICA+-Kategorien

Die Wiederholbarkeit von Kategorien im PICA-Format ist uneinheitlich realisiert. Im PICA+-Format ist dies generell durch die ergänzende *Occurrence*-Angabe mit einer durch Schrägstrich abgetrennten zweistelligen Zahl umgesetzt (s. Beispiele in Tab. 12). Beim Import von PICA+-Daten in OpenRefine[416] stellen die unterschiedlichen Repräsentationen wiederholbarer Kategorien eine besondere Herausforderung dar, daher werden die verschiedenen Ausprägungen hier ausführlich wiedergegeben.

In einem Katalogisat mehrfach vorkommende Sachverhalte werden in PICA3 meist durch eine Kategoriegruppe abgebildet, dies gilt z. B. für die PICA3-Kategorien der Basisklassifikation (BK). Sind mehrere BK-Notationen vorhanden, werden diese jeweils einzeln in den PICA3-Kategorien 5301–5309 erfasst. Dies entspricht den PICA+-Kategorien 045Q/01 bis 045Q/09.

PICA+	**045Q/01** ƒ9106405047ƒ818.10$jDeutsche Literatur **045Q/02** ƒ9106404555ƒ817.71$jLiteraturgeschichte **045Q/03** ƒ9106402722ƒ817.70$jLiteraturwissenschaft: Allgemeines
PICA3	**5301** !106405047!18.10$jDeutsche Literatur **5302** !106404555!17.71$jLiteraturgeschichte **5303** !106402722!17.70$jLiteraturwissenschaft: Allgemeines

Tab. 13: Mehrere Kategorien einer PICA3-Kategoriegruppe

[414] Die Kategorien in Tab. 12 sind in der PICA3-Anzeigereihenfolge aufgeführt.
[415] In der PICA-Terminologie werden die Ebenen Titel – Lokal – Exemplar als Level bezeichnet.
[416] Siehe Kap. 5.3.

Eine weitere Variante ist die Unterscheidung von Feldern einer Kategoriegruppe über das PICA+-Unterfeld $x, das ist z. B. bei Signaturkategorien der Fall. In PICA3 stehen die Kategorien 7100–7109 für die Erfassung von Signaturen zur Verfügung, dabei ist die in der letzten Ziffer ausgedrückte Zählung in PICA+ als Unterfeld $x abgebildet (s. Beispiel in Tab. 14). Die Elemente dieser Kategoriegruppe werden in den angeschlossenen Systemen unterschiedlich behandelt. So hat ausschließlich der Inhalt aus 7100 in LBS und OPAC eine Steuerungsfunktion, die Inhalte aus 7101–7109 dienen lediglich der Recherche.[417]

PICA+	**209A/01** ƒf8/10ƒaMG-Va-0086 4.Aufl.ƒdiƒx00 **209A/01** ƒaBereich Politikƒx04
PICA3	**7100** !8/10!MG-Va-0086 4.Aufl. @ i **7104** Bereich Politik

Tab. 14: Wiederholbare PICA+-Kategorie mit Kennzeichnung in Subfield $x

Schließlich gibt es wiederholbare PICA+-Kategorien, denen entsprechende wiederholbare PICA3-Kategorien zugeordnet sind. Für bis zu zehn RSWK-Schlagwortfolgen stehen die jeweils wiederholbaren PICA3-Kategorien 5550–5559 zur Verfügung. Dabei wird jedes Element einer Schlagwortfolge in einer eigenen PICA3-Kategorie mit derselben Kategorienummer erfasst. Die Schlagwörter der ersten Schlagwortfolge erhalten jeweils die Kategorie 5550, die Schlagwörter der zweiten Schlagwortfolge die Kategorie 5551 usw. Jedes Vorkommen einer PICA3-Kategorie 5550 wird im PICA+-Format als 044K/00 dargestellt, ein Ordinalzahl-Aspekt ist nicht erkennbar.

PICA+	**044K/00** ƒ9104529369ƒ8Deutsches Sprachgebiet ; ID: gnd/4070370-8 **044K/00** ƒ9104744138ƒ8Schriftsteller ; ID: gnd/4053309-8 **044K/00** ƒSzƒaGeschichte 1900-2000 **044K/00** ƒSfƒaBiographie **044K/01** ƒSsƒ9106083406ƒ8Politische Wissenschaft ; ID: gnd/4076229-4 **044K/01** ƒSsƒ9106230158ƒ8Methode ; ID: gnd/4038971-6 **044K/01** ƒSfƒaAufsatzsammlung
PICA3	**5550** !104529369!Deutsches Sprachgebiet ; ID: gnd/4070370-8 **5550** !104744138!Schriftsteller ; ID: gnd/4053309-8 **5550** |z|Geschichte 1900-2000 **5550** |f|Biographie **5551** |s|!106083406!Politische Wissenschaft ; ID: gnd/4076229-4 **5551** |s|!106230158!Methode ; ID: gnd/4038971-6 **5551** |f|Aufsatzsammlung

Tab. 15: Unterschiedliche Darstellung von wiederholbaren Kategorien (Beispiel fingiert)

[417] Vgl. Richtlinie für die Kategoriegruppe 710x: Verbundzentrale des GBV 2014.

Anhang 3 Datenbankinternes Speicherformat PICA+

Die LBS-Datenbanktabelle `titles_global` enthält 13 Attribute, davon sind 8 Attribute für den Inhalt der titelbezogenen Kategorien aus dem Katalogisat vorgesehen.

Attribut	Datentyp	Länge in Bytes	Inhalt
`fno`	tinyint	1	Nummer des logischen Bestandes für die Daten einer Bibliothek[418]
`ppn`	int	4	laufende CBS-Datensatznummer, ohne Prüfziffer
`status`	char	1	mögliche Werte: 0 Titel aus CBS 1 Lokales Katalogisat 2 logisch gelöschter Titel aus CBS 3 logisch gelöschtes lokales Katalogisat[419]
`opacflag`	bit	1	regelt die Sichtbarkeit im OPAC true im OPAC sichtbar false für die OPAC-Anzeige unterdrückt wird per Selektionsschlüssel in PICA3-Feld 70xx gesetzt
`length`	smallint	2	Gesamtlänge des Datensatzes
`mark1`	varchar	255	enthält die ersten 255 Byte der titelbezogenen Felder im PICA+-Format, in komprimierter Form (ähnlich wie MARC 21)
`mark2`	varchar	255	ggf. Fortsetzung von `mark1`
`mark3`	varchar	255	ggf. Fortsetzung von `mark2`
`mark4`	varchar	255	ggf. Fortsetzung von `mark3`
`mark5`	varchar	255	ggf. Fortsetzung von `mark4`
`mark6`	varchar	255	ggf. Fortsetzung von `mark5`
`mark7`	varchar	255	ggf. Fortsetzung von `mark6`
`mark8`	varchar	255[420]	ggf. Fortsetzung von `mark7`

Tab. 16: Aufbau der Datenbanktabelle `titles_global`

Die Kategorien aus Level 0 sind in der Tabelle `titles_global` in den Attributen `mark1` bis `mark8` mit je 255 Byte gespeichert. Falls die Titelkategorien weiteren Speicherplatz benötigen, werden die restlichen Inhalte in der Datenbanktabelle `title_overflow` abgelegt. Auch hier stehen pro Tupel die Attribute `mark1` bis `mark8` mit je 255 Byte zur Verfügung. Je nach Umfang des Katalogisates werden mehrere Einträge in `title_overflow` gebildet.[421]

[418] Die Bestandsnummer dient der Zuordnung zu einer Bibliothek im Sinne einer Mandantenfähigkeit.
[419] Weitere Ausführungen zum Status siehe Kap. 5.5.2, Schritt 1.
[420] Laut OCLC-Dokumentation im CBS mit 141 Byte.
[421] Diese werden über das Attribut `volgnr` nummeriert.

```
Address   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  Dump
00000000 01 20 15 20 01 20 30 0f 30 37 35 35 3a 30 31 2d . . . 0.0755:01-
00000010 31 32 2d 39 34 20 01 20 23 20 02 20 30 0f 34 33 12-94 . # . 0.43
00000020 35 38 3a 32 38 2d 31 31 2d 30 38 74 0e 30 38 3a 58:28-11-08t.08:
00000030 35 31 3a 32 33 2e 30 30 30 20 01 20 15 20 04 20 51:23.000 . . .
00000040 30 0f 39 39 39 39 3a 39 39 2d 39 39 2d 39 39 20 0.9999:99-99-99
00000050 02 20 0b 20 20 20 30 05 41 61 72 20 03 20 11 20 . .   0.Aar . .
00000060 20 20 30 0b 31 37 32 32 36 38 33 32 58 20 04 20   0.17226832X .
00000070 12 20 01 20 30 0c 30 39 30 35 30 34 35 36 39 36 . . 0.0905045696
00000080 06 20 12 20 19 20 30 0c 42 49 53 54 35 36 33 33 . . . 0.BIST5633
00000090 34 34 06 20 13 20 19 20 30 69 73 6c 38 38 30 32 44. . . 0isl8802
000000a0 30 32 38 33 20 b6 20 0b 20 20 20 61 05 65 6e 67 0283 ¶ .   a.eng
000000b0 20 0b 20 0c 20 20 20 61 06 31 39 38 34 13 20 20  . .   a.1984.
000000c0 20 20 61 07 58 41 2d 47 42 20 15 20 5b 20 01 20   a.XA-GB . [ .
000000d0 61 12 53 68 69 70 70 69 6e 67 20 66 69 6e 61 6e a.Shipping finan
000000e0 63 65 64 33 66 69 6e 61 6e 63 69 6e 67 20 73 68 ced3financing sh
000000f0 69 70 73 20 61 6e 64 20 6d 6f 62 69 6c 20 7c    ips and mobil |
```

Datenbeispiel 6: Attribut `mark1` aus Tabelle `titles_global` im Speicherformat PICA+

Eine PICA+-Kategoriebezeichnung setzt sich den drei Bestandteilen Level, Typ und Indikator zusammen. Diese werden an verschiedenen Positionen im Speicherformat codiert.

Beispiele:

`0|02|C`	Level 0, Typ 02, Indikator C
`2|09|A`	Level 2, Typ 09, Indikator A

Ein **Datensatz** besteht aus aufeinanderfolgenden Abschnitten für die einzelnen PICA+-Kategorien, ggf. angereichert um ein Füllbyte[422] (`0x00`).

Aufbau: `<tag>[<PAD>]{<tag>[<PAD>]}`

Header	Länge: 6 Byte
Kategorie-Inhalt	Länge: 3..n Byte
Füllbyte (0x00)	Länge: 1 Byte

Abb. 17: Aufbau eines Datensatzes im Speicherformat PICA+

Ist die Gesamtlänge von Header und allen Unterfeldern für eine Kategorie ungerade, so wird vor dem Abschnitt für die nachfolgende Kategorie ein Füllbyte eingeschoben, damit alle Kategorien auf gradzahliger Byteposition beginnen.

[422] Auch: *Pad*.

Der Abschnitt für eine **Kategorie** beginnt mit einem Header in fester Länge von 6 Byte, gefolgt von dem variabel langen Bereich für eine beliebige Anzahl Unterfelder.

Aufbau: `<tag_header>{<subfield>}`

Header	Länge: 6 Byte	
Subfield-Typ	Länge: 1 Byte	
Subfield-Länge	Länge: 1 Byte	
Subfield-Inhalt	Länge: 1..253 Byte	

Abb. 18: Aufbau einer Kategorie im Speicherformat PICA+

Der **Header** enthält diese Elemente:

Bestandteil	gültige Werte bzw. Inhalt	Länge in Byte
`tag_type`	Pos. 2+3 der PICA+-Kategoriebezeichnung	2
`tag_length`	Gesamtlänge der Kategorie mit allen Subfields, inkl. Header	2
`tag_lvlind`	Level und Indikator werden bitweise codiert: Das Level (Pos. 1 der PICA+-Kategoriebezeichnung) ist in den beiden *Most Significant Bit* gespeichert, der Indikator (Pos. 4 der PICA+-Kategoriebezeichnung) auf den folgenden 6 Bitpositionen.	1
`tag_occurrence`	Wiederholungsfaktor; im GBV zweistellig[423] (in der externen Darstellung)	1

Tab. 17: Bestandteile des Kategorie-Header im Speicherformat PICA+

Aufbau: `<tag_type><tag_length><tag_lvlind><tag_occurrence>`

Tag-Typ	Länge: 2 Byte	1..99
Tag-Länge	Länge: 2 Byte	6..<maxlenttl>
Tag-Level/-Indikator	Länge: 1 Byte	
Tag-Occurrence	Länge: 1 Byte	0..99

Abb. 19: Aufbau des Kategorie-Headers im Speicherformat PICA+

Das **Byte für die Codierung von Level und Indikator** ist aufgeteilt in 2 Bit für die Darstellung des Levels und 6 Bit für die Darstellung des Indikators.

Für den Indikator gilt: Gespeichert wird die Differenz, die sich aus der Subtraktion des Wertes für das @-Zeichen (`0x40`) vom Wert des Indikators (z. B. `0x41` für A) ergibt. Somit errechnen sich diese Werte[424]: @ → 0 A → 1 B → 2 … Z → 26

[423] Damit sind max. 99 Exemplare möglich.
[424] Angegeben sind die Dezimalwerte.

Aufbau: `<tag_level> << 6) | (<tag_indicator> - '@'`

- Tag-Level — Länge: 2 Bit — 0..2
- Tag-Indikator — Länge: 6 Bit — @|A..Z

Abb. 20: Aufbau des Bytes für die Codierung von Level und Indikator im Speicherformat PICA+

Jedem **Unterfeld**-Inhalt werden der Subfield-Typ und die Subfield-Länge mit je 1 Byte vorangestellt.

Bestandteil	gültige Werte bzw. Inhalt	Länge in Byte
`subfield_type`	Buchstabe oder Ziffern	1
`subfield_length`	Gesamtlänge des Unterfeldes, inkl. Typ und Länge	1
`content`	Inhalt des Unterfeldes	1..253

Tab. 18: Aufbau von Subfields im PICA+-Speicherformat

Beispiele für die Speicherung im PICA+-Format

Kategorie `002@`

Inhalt: $0Aau ≙ $02000B0000003005416175_{16}$

Header:		0200|0B00|00|00	
Type	0	30	1 Byte
Length	5	05	1 Byte
Inhalt	Aau	41|61|75	3 Byte

Kategorie `002C`

Inhalt: $aText$btxt ≙ $020011000300610654657874620574787 4_{16}$

Header:		0200|1100|03|00	
Type	a	61	1 Byte
Length	6	06	1 Byte
Inhalt	Text	54|65|78|74	4 Byte
Type	b	62	1 Byte
Length	5	05	1 Byte
Inhalt	txt	74|78|74	3 Byte

Ist der Inhalt eines Subfields länger als 253 Byte, so werden die weiteren Zeichen in einem direkt angeschlossenen Folge-Subfield (*continuation subfield*) gespeichert und mit Subfield-Type `0x19` eingeleitet. Je nach tatsächlicher Kategorielänge können weitere Folge-Subfields folgen.

Anhang 4 Catmandu-Beispiele

Der folgende Aufruf nutzt die *Fix*-Kommandos in der Datei `fix_fl.txt`:

```
$ catmandu convert plain --fix fix_fl.txt to plain < bs_luy_plain.txt
> bs_fernleihe.txt
```

Zum Laden eines Datensatzes über die SRU-Schnittstelle einer GBV-Bibliothek kann dieses Kommando verwendet werden:

```
$ catmandu convert SRU --base http://sru.gbv.de/owc-de-84
--query "pica.mak=luy" --recordSchema picaxml --parser picaxml
to PICA --type plain > opac_bs.txt
```

Hier wird der per SRU im PICA-XML-Format heruntergeladene Datensatz vor dem Speichern in das (lesbare) PICA+-Format konvertiert.

Beispiele

Es folgte eine Auswahl von getesteten *Fix*-Funktionen zur Transformation von PICA-Daten im Hinblick auf die Anwendung im LOK-Projekt.

Datensätze mit Doppelpunkt in Subfield $b von Feld 208@ im ersten Exemplar ausgeben:

```
select pica_match("208@[01]b",":")
```

So können Erfassungsfehler in der Kategorie für den Selektionsschlüssel erkannt werden, die ggf. die gewünschte Unterdrückung des Datensatzes für die OPAC-Anzeige verhindern.

Fix-Funktion `fix_ohne_fl.txt`, um Datensätze, deren Titel mit „Fernleihe“ beginnt, aus der Zieldatenmenge auszuschließen:

```
do pica_each()
  if pica_match("021Aa","^Fernleihe*")
    reject()
  end
end
```

Kommando:

```
$ catmandu convert plain --fix fix_ohne_fl.txt to plain < bs_luy_plain.txt
> bs_luy_ohne_fl.txt
```

Fix-Funktion `fix_fl.txt` zur Ermittlung der Anzahl Datensätze mit dem Begriff „Fernleihe“:

```
unless
  pica_match("021Aa","Fernleihe")
  reject()
end
```

Kommando:

```
$ catmandu convert plain --fix fix_fl.txt to Count < bs_luy_plain.txt
```

Fix-Funktion `fix_ppn.txt` zur Ausgabe der PPNs aller Datensätze:

```
pica_map("003@0","PPN");
retain("PPN")
```

Kommando:

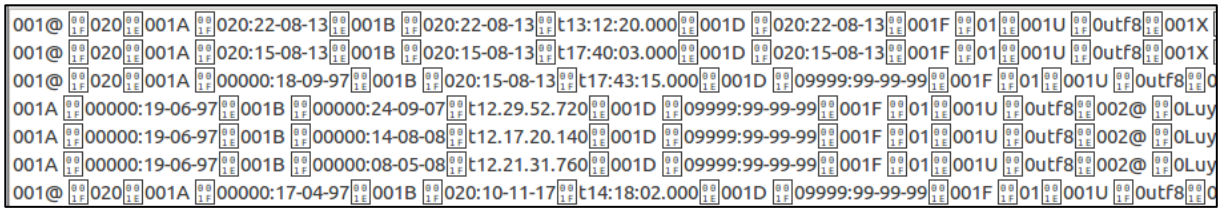
```
$ catmandu convert plain --fix fix_ppn.txt to Text < bs_luy_plain.txt
```

Beispiel 1: SRU-Download mit Exporterformat PICA, Typ plus

Kommando

```
$ catmandu convert opac_bs_pica --query "pica.mak=luy" to PICA --type plus
> bs_luy_plus.txt
```

Ausgabe



Datenbeispiel 7: Ausschnitt aus SRU-Download mit Catmandu-Formattyp `plus`

Beispiel 2: Anwendung des Breaker-Moduls mit PICA-Daten

Kommando

```
$ convert plain to Breaker --handler pica < bs_luy_plain.txt > data.breaker
```

Ausgabe

```
981433251   001@0       20
981433251   001A0       20:22-08-13
981433251   001B0       20:22-08-13
981433251   001Bt       13:12:20.000
981433251   001D0       20:22-08-13
981433251   001F0       1
981433251   001U0       utf8
981433251   001X0       0
981433251   002@0       Luy
981433251   003@0       981433251
981433251   011@a       2014
981433251   021Aa       @test
981433251   101@a       20
981433251   201@[01]a   Ausleihbestand|Standort unbekannt, wenden Sie sich an die Information
981433251   201@[01]b   0
981433251   201@[01]e   981433162
981433251   201@[01]m   mon
981433251   201@[01]f   1
981433251   201@[01]u   Ausleihbestand
981433251   201@[01]v   Standort unbekannt, wenden Sie sich an die Information
981433251   201B[01]0   22-08-13
981433251   201B[01]t   13:12:20.000
981433251   201F[01]0   1
981433251   201U[01]0   utf8
981433251   203@[01]0   981433162
981433251   208@[01]a   22-08-13
981433251   208@[01]b   d
```

Datenbeispiel 8: Breaker-Modul mit PICA-Daten

Beispiel 3: Anwendung der Stat-Distribution mit PICA-Daten

Kommando

```
$ catmandu breaker data.breaker
```

Ausgabe

name	count	zeros	zeros%	min	max	mean	variance	stdev	uniq~	uniq%	entropy
#	179										
001@0	51	128	71.5	0	1	0.284916201117318	0.2	0.5	2	3.9	0.9/7.5
001A0	179	0	0.0	1	1	1	0.0	0.0	55	30.8	5.0/7.5
001B0	179	0	0.0	1	1	1	0.0	0.0	108	60.5	5.5/7.5
001Bt	179	0	0.0	1	1	1	0.0	0.0	178	100.0	7.5/7.5
001D0	179	0	0.0	1	1	1	0.0	0.0	4	2.2	0.1/7.5
001F0	179	0	0.0	1	1	1	0.0	0.0	1	0.6	0.0/7.5
001U0	179	0	0.0	1	1	1	0.0	0.0	1	0.6	0.0/7.5
001X0	51	128	71.5	0	1	0.284916201117318	0.2	0.5	2	3.9	0.9/7.5
002@0	179	0	0.0	1	1	1	0.0	0.0	1	0.6	0.0/7.5
003@0	179	0	0.0	1	1	1	0.0	0.0	179 (!)	100.6 (!)	7.5/7.5 (!)
011@a	2	177	98.9	0	1	0.0111731843575419	0.0	0.1	3 (!)	150.0 (!)	0.1/7.5 (!)
021Aa	179	0	0.0	1	1	1	0.0	0.0	110	61.7	5.4/7.5
021Ah	1	178	99.4	0	1	0.00558659217877095	0.0	0.1	2 (!)	200.0 (!)	0.0/7.5 (!)
028Aa	103	76	42.5	0	1	0.575418994413408	0.2	0.5	48	46.7	3.0/7.5
028Ac	1	178	99.4	0	1	0.00558659217877095	0.0	0.1	2 (!)	200.0 (!)	0.0/7.5 (!)
028Ad	50	129	72.1	0	1	0.279329608938547	0.2	0.4	51 (!)	102.2 (!)	2.4/7.5 (!)
028B[01]a	1	178	99.4	0	1	0.00558659217877095	0.0	0.1	2 (!)	200.0 (!)	0.0/7.5 (!)
028B[01]d	1	178	99.4	0	1	0.00558659217877095	0.0	0.1	2 (!)	200.0 (!)	0.0/7.5 (!)
101@a	179	0	0.0	1	1	1	0.0	0.0	1	0.6	0.0/7.5
101B0	1	178	99.4	0	1	0.00558659217877095	0.0	0.1	2 (!)	200.0 (!)	0.0/7.5 (!)
101Bt	1	178	99.4	0	1	0.00558659217877095	0.0	0.1	2 (!)	200.0 (!)	0.0/7.5 (!)
101U0	1	178	99.4	0	1	0.00558659217877095	0.0	0.1	2 (!)	200.0 (!)	0.0/7.5 (!)
144Za	2	178	99.4	0	2	0.0111731843575419	0.0	0.1	3 (!)	150.0 (!)	0.1/7.5 (!)
201@[01]a	179	0	0.0	1	1	1	0.0	0.0	9	5.0	0.9/7.5
201@[01]b	179	0	0.0	1	1	1	0.0	0.0	3	1.7	0.3/7.5
201@[01]d	28	151	84.4	0	1	0.156424581005587	0.1	0.4	18	64.3	1.2/7.5
201@[01]e	179	0	0.0	1	1	1	0.0	0.0	179 (!)	100.6 (!)	7.5/7.5 (!)
201@[01]f	179	0	0.0	1	1	1	0.0	0.0	1	0.6	0.0/7.5
201@[01]l	9	170	95.0	0	1	0.0502793296089385	0.0	0.2	10 (!)	111.1 (!)	0.4/7.5 (!)
201@[01]m	179	0	0.0	1	1	1	0.0	0.0	1	0.6	0.0/7.5
201@[01]n	5	174	97.2	0	1	0.0279329608938547	0.0	0.2	2	40.0	0.2/7.5
201@[01]u	179	0	0.0	1	1	1	0.0	0.0	3	1.7	0.1/7.5
201@[01]v	179	0	0.0	1	1	1	0.0	0.0	6	3.4	0.9/7.5
201B[01]0	179	0	0.0	1	1	1	0.0	0.0	82	45.9	5.6/7.5
201B[01]t	179	0	0.0	1	1	1	0.0	0.0	175	98.3	7.5/7.5
201F[01]0	179	0	0.0	1	1	1	0.0	0.0	1	0.6	0.0/7.5
201U[01]0	179	0	0.0	1	1	1	0.0	0.0	1	0.6	0.0/7.5
203@[01]0	179	0	0.0	1	1	1	0.0	0.0	179 (!)	100.6 (!)	7.5/7.5 (!)
208@[01]a	179	0	0.0	1	1	1	0.0	0.0	56	31.3	5.0/7.5
208@[01]b	179	0	0.0	1	1	1	0.0	0.0	4	2.2	0.5/7.5
209A[01]a	177	2	1.1	0	1	0.988826815642458	0.0	0.1	177 (!)	100.5 (!)	7.5/7.5 (!)
209A[01]d	3	176	98.3	0	1	0.0167597765363128	0.0	0.1	4 (!)	133.3 (!)	0.1/7.5 (!)
209A[01]f	168	11	6.1	0	1	0.93854748603352	0.1	0.2	3	1.8	0.6/7.5
209A[01]x	177	2	1.1	0	1	0.988826815642458	0.0	0.1	2	1.1	0.1/7.5

Datenbeispiel 9: Catmandu-Feldstatistik über PICA+-Daten

Anhang 5 Perl-Programm `opc4_lok_anzahl` zur Bestandsaufnahme

Das Programm ermittelt für jede Bibliothek die Anzahl der lokalen Katalogisate, getrennt nach Materialart, und schreibt die Werte in eine vorbereitete Excel-Tabelle. Der Datenzugriff erfolgt über die PSI-XML-Schnittstelle.

Der **Programmablauf** wird über diese Kommandozeilenparameter gesteuert:

`-s` Standortkürzel, wie in der VZG üblich als LBS-Systemvariable *site* gesetzt;
alternativ: `all` für alle Standorte.

`-h` Anzeige des Hilfetextes

Als Kommandozeilenparameter ist ein 3-stelliges alphanumerisches Kürzel für den betreffenden LBS-Standort anzugeben.[425] Dann erfolgt die Auswertung für die Bibliotheken des betreffenden Standortes. Mit dem Parameter `all` wird eine Auswertung für alle LBS-Bibliotheken durchgeführt.

In der Excel-Datei ist die Zuordnung von Kürzel zur jeweiligen OPAC-URL hinterlegt. Das Programm liest diese Informationen und führt in dem betreffenden LBS-OPAC einen sog. `Browse`-Aufruf durch. Es wird die Liste der Indexeinträge für den angegebenen Suchbegriff mit der jeweiligen Trefferzahl zurückgegeben; ggf. muss noch eine Folgeseite aufgerufen werden. Diese Informationen (Indexeintrag und Anzahl) werden in die Excel-Tabelle zurückgeschrieben.

Es entstehen außerdem drei **Ausgabedateien**:

`lok-logfile_<timestamp>.txt`
Die Protokolldatei enthält für jede Bibliothek eine Auflistung der ermittelten Materialcodes mit der jeweiligen Anzahl.

```
https://lhbrs2.gbv.de/DB=OWC.3/XML=1.0/CMD?ACT=BRWS&IKT=8600&TRM=\l
Materialcode Lwx  mit   5785 Treffern
Materialcode Luy  mit   9040 Treffern
Materialcode Lbx  mit      1 Treffern
Materialcode Ldx  mit  28783 Treffern
Materialcode Lux  mit      1 Treffern
Anzahl unterschiedliche L-Typen:  5
```

Die Protokolldatei schließt mit einer Verarbeitungsstatistik.

```
+++ Programmstart:  Thu May 10 18:04:27 2018
+++ Programm:       opc4_lok_anzahl.pl
+++   183 Bibliotheken verarbeitet
+++         0 mit Fehler beim OPAC-Zugriff
+++       109 Bibliotheken mit L-Sätzen
+++    50 unterschiedliche L-Typen
+++ Programmende:          Thu May 10 18:05:58 2018
```

[425] Die Kürzel entsprechen i. d. R. den in der LBS4-Systemkonfiguration anzugebenden Standortkürzeln, die in der LBS-Gruppe bzw. bei den betreffenden Systemverwaltern bekannt und gebräuchlich sind. Zur Unterscheidung der identischen LBS-Systemkürzel bei den drei von der LBS-Gruppe betreuten LBS-Servern wurden für das Programm individuelle Kürzel als Kommandozeilenparameter festgelegt.

`lok-search_urls_<timestamp>.txt`
Diese Datei war (prophylaktisch) vorgesehen, falls in einem weiteren Programm die exakten Suchanfragen nachgenutzt würden. Im weiteren Verlauf der Programmierung stellte sich heraus, dass dies nicht notwendig ist, da im Programm `opc4_lok_titel` mit variablen Suchbegriffen per Kommandozeilenparameter gearbeitet wird. Die entsprechenden Zeilen im Programmcode könnten daher entfernt oder auskommentiert werden.

`lok-daten_<timestamp>.txt`
Die Dumper-Ausgabe der geparsten OPAC-Seiten wurde während der Programmentwicklung benötigt. Ggf. wäre das Programm dahingehend zu erweitern, dass diese Datei nur erzeugt wird, wenn ein entsprechender Kommandozeilenparameter vorhanden ist.[426] Alternativ könnten die entsprechenden Zeilen im Programmcode auskommentiert werden.

Die **Fehlerbehandlung** berücksichtigt u. a. diese Fälle:

- HTTP-Fehler: bisher vorgekommen sind die Fehler „`read timeout`“ und „`Can't connect to <serveradresse>`“. Vermutliche Ursache: Wartungsarbeiten am Server.
- Kein Zugriff auf den Index. Ursache: Parallel läuft das OPAC-Systemprogramm „`create_index`“ bzw. „`update_index`“.

Bei diesen Fehlern wird die Verarbeitung für die betreffende Bibliothek abgebrochen. Außerdem trat im Rahmen der Programmtests ein Fehler beim Seitenaufruf auf, weil einer der Materialcodes mit einem Suchschlüssel identisch ist. Daher wurde im Programm die Maskierung des Suchbegriffs in der Aufruf-URL ergänzt.

Anmerkungen

Bei strukturellen Änderungen der Excel-Tabelle muss das Programm angepasst werden, da das erste Vorkommen eines Materialcodes fest codiert ist. Änderungen der Adresse eines LBS-Servers müssen in der Excel-Datei nachgeführt werden.

Seitdem einige Bibliotheken mit der Erfassung von lokalen Katalogisaten im CBS begonnen haben, sind die ermittelten Zahlen für diese Bibliotheken nicht unbedingt aussagekräftig, da das Programm nicht zwischen LBS-Datensätzen mit Materialart „Lax“ und Lax-Sätzen im CBS unterscheidet. Hier wäre eine entsprechende Programmerweiterung sinnvoll. Hierzu müssten die betreffenden Datensätze aufgerufen und geparst werden; entsprechende Funktionen ließen sich vermutlich aus dem Programm `opc4_lok_titel` übernehmen.

Auf der beiliegenden CD im Verzeichnis „1 – Bestandsaufnahme“:

- `opc4_lok_anzahl.pl` Quellcode
- `LOK-Tabelle.xlsx` Eingabe-/Ausgabedatei
- Für zwei Programmläufe mit verschiedenen Kommandozeilenoptionen
  - `lok-logfile_*.txt` Protokolldatei
  - `lok-search_urls_*.txt` Ausgabedatei mit URLs
  - `lok-daten_*.txt` Dumper-Ausgabe

[426] So ist es im Programm `opc4_lok_titel.pl` umgesetzt.

Anhang 6 WinIBW-Download aus OWC/CAT4

```
SET: S1 [1] TIT: 1            PPN: 0017226832X                          SEI: 01.

001A:0755:01-12-94  001B:4358:28-11-08 08:51:23  001D:9999:99-99-99

+00+ [S:0,F:1]
001A ƒ00755:01-12-94
001B ƒ04358:28-11-08ƒt08:51:23.000
001D ƒ09999:99-99-99
002@ ƒ0Aar
003@ ƒ017226832X
004A ƒ00905045696
006Y ƒ0BIST563344
006Y ƒ0isl88020283
010@ ƒaeng
011@ ƒa1984
019@ ƒaXA-GB
021A ƒaShipping financeƒdfinancing ships and mobile offshore installationsƒhJ. E
028A ƒdJolyon E.ƒaSloggett
033A ƒpLondonƒnFairplay
034D ƒaVIII, 116 S
034K ƒazahlr. Tab.
034M ƒagraph. Darst
036E ƒaShip management series
+02+ [I:264,S:0,F:1]
208@/01 ƒa27-04-10ƒbz
201B/01 ƒ027-04-10ƒt13:27:51.000
201D/01 ƒ027-04-10ƒb9812ƒa3264
203@/01 ƒ01157076807
209A/01 ƒa10dg14ƒduƒx00
209C/01 ƒa7581ƒx00
209G/01 ƒa897E$087491
220B/01 ƒa06542
```

Datenbeispiel 10: WinIBW-Download aus OWC im Format PICA+

```
SET: S1 [1] TIT: 1            PPN: 0017226832X                          SEI: 01.

0200:0755:01-12-94  0210:4358:28-11-08 08:51:23  0230:9999:99-99-99

0100 17226832X
0500 Aar
1100 1984
1500 /1eng
1700 /1XA-GB
2000 0-905045-69-6
2199 BIST563344
2199 isl88020283
3000 Jolyon E.@Sloggett
4000 Shipping finance : financing ships and mobile offshore installations / J. E
4030 London : Fairplay
4060 VIII, 116 S
4061 graph. Darst
4063 zahlr. Tab.
4170 Ship management series
7001 27-04-10 : z
7901 27-04-10; 9812/3264
4802 06542
7100 10dg14 @ u
8100 7581
8200 897E$087491
7800 1157076807
```

Datenbeispiel 11: WinIBW-Download aus OWC im Format PICA3

```
SET: S1 [1] TTL: 1            PPN: 17226832X                          SEITE 1.


001A $00755:01-12-94
001B $04358:28-11-08$t08:51:23.000
001D $09999:99-99-99
001U $0utf8
002@ $0Aar
003@ $017226832X
004A $00905045696
006Y $0BIST563344
006Y $0isl88020283
010@ $aeng
011@ $a1984
019@ $aXA-GB
021A $aShipping finance$dfinancing ships and mobile offshore installations$hJ. E
028A $dJolyon E.$aSloggett
033A $pLondon$nFairplay
034D $aVIII, 116 S
034K $azahlr. Tab.
034M $agraph. Darst
036E/00 $aShip management series
208@/01 $a27-04-10$bz
201B/01 $027-04-10$t13:27:51.000
201D/01 $027-04-10$b9812$a3264
201U/01 $0utf8
203@/01 $01157076807
209A/01 $a10dg14$du$x00
209C/01 $a7581$x00
209G/01 $a897E$087491
220B/01 $a06542
```

Datenbeispiel 12: WinIBW-Download aus CAT4 im Format PICA+

```
SET: S1 [1] TTL: 1            PPN: 17226832X                          SEITE 1.


000K utf8
0100 17226832X
0200 0755:01-12-94
0210 4358:28-11-08; 08:51:23.000
0230 9999:99-99-99
0500 Aar
1100 1984
1500 eng
1700 XA-GB
2000 0-905045-69-6
2199 BIST563344
2199 isl88020283
3000 Sloggett, Jolyon E.
4000 Shipping finance$dfinancing ships and mobile offshore installations$hJ. E.
4030 London$nFairplay
4060 VIII, 116 S
4061 graph. Darst
4063 zahlr. Tab.
4170 Ship management series
7001 27-04-10 : z
0248 utf8
4802 06542
7100 10dg14 @ u
7800 1157076807
7901 27-04-10; 9812/3264
7903 27-04-10 13:27:51.000
8100 7581
8200 897E$087491
```

Datenbeispiel 13: WinIBW-Download aus CAT4 im Format PICA3

Anhang 7 WinIBW-Funktion „Exceltabelle erstellen“

"PPN"	"EPN"	"Titel"	"Standort"	"Signatur"	"AUI"	"Verbuchungsnr"
"39345889X"		"970187025"	""	""	""	""
"970196962"		"970196962"	""	""	""	""
"970196571"		"970196571"	""	""	""	""
"970196512"		"970196512"	""	""	""	""
"970196504"		"970196504"	""	""	""	""
"97019515X"		"97019515X"	""	""	""	""
"970195141"		"970195141"	""	""	""	""
"970195133"		"970195133"	""	""	""	""
"970192983"		"970192983"	""	""	""	""
"970192975"		"970192975"	""	""	""	""
"97019241X"		"97019241X"	""	""	""	""
"970190050"		"970190050"	""	""	""	""
"970190042"		"970190042"	""	""	""	""
"970187890"		"970187890"	""	""	""	""
"970183798"		"970183798"	""	""	""	""
"97017618X"		"97017618X"	""	""	""	""

Datenbeispiel 14: Ausgabedatei der WinIBW-Funktion „Exceltabelle erstellen“ in CAT4

Anhang 8 OPAC-Download

Die Speichern-Seite eines von der VZG betreuten LBS-Standorts, mit 5 Ausgabeformaten zur Auswahl:

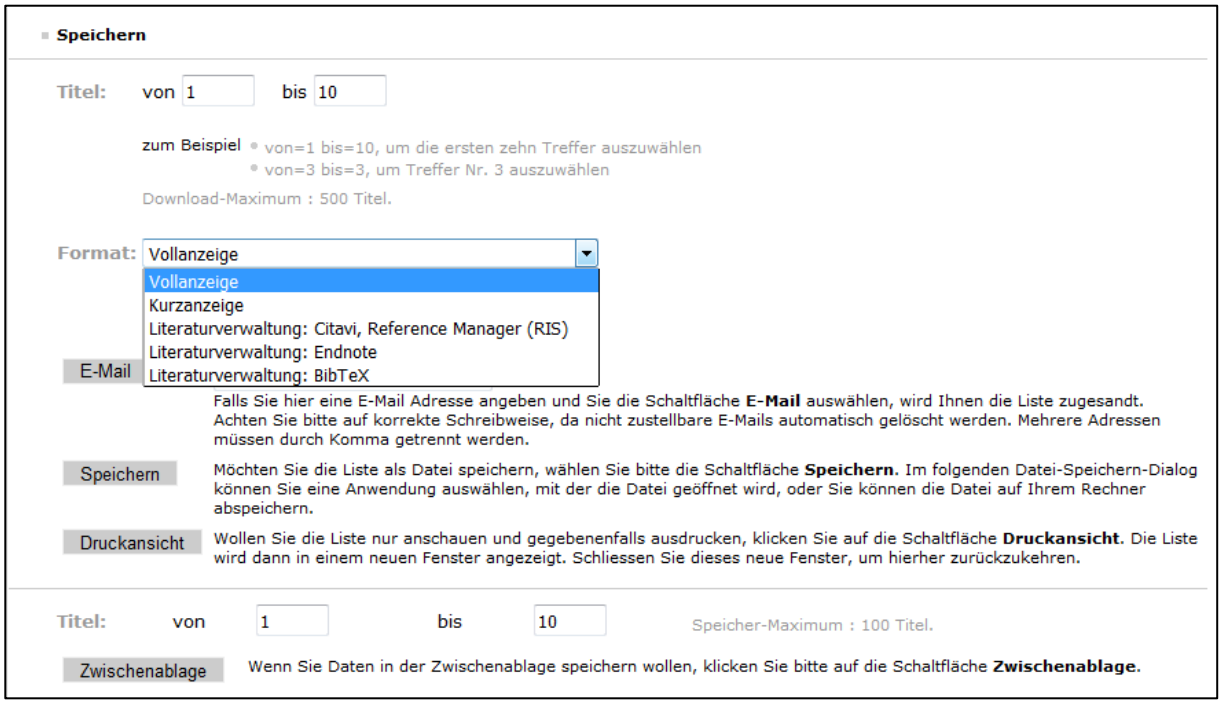


Abb. 21: OPAC-Download: Ausgabeformate

Als *Download-Maximum* sind hier 500 Titel angegeben.

Anhang 9 Ausgabe eines lokalen Katalogisats per unAPI- und SRU-Schnittstelle

Die folgenden Beispiele zeigen jeweils PPN 981559603.

```
001@ $020$e1
001A $020:18-04-17
001B $020:18-04-17$t09:25:08.000
001D $020:18-04-17
001F $01
001U $0utf8
001X $00
002@ $0Lux
003@ $0981559603
010@ $ager
011@ $a2017
021A $a@Zeitlich hochaufgelöste optische und elektrochemische Untersuchungen beim
Gefrieren unterkühlter, wässriger Tropfen$hvon Tillmann Peregrin Buttersack
028A $dTillmann Peregrin$aButtersack$BVerfasserIn$4aut
034D $a1 CD-ROM
037A $a$$dDissertation$$eTechnische Universität  Carolo-Wilhelmina zu
Braunschweig$$f2016
037A $aAuch als Druckausgabe und Online-Ausgabe
037A $aZusammenfassung in deutscher und englischer Sprache
101@ $a20
201@/01 $aAusleihbestand|Noch nicht
verfuegbar$b0$e981559514$mmon$f1$uAusleihbestand$vNoch nicht verfuegbar
201B/01 $018-04-17$t10:20:18.000
201F/01 $01
201U/01 $0utf8
203@/01 $0981559514
208@/01 $a18-04-17$bd
209A/01 $a4209-0324$x00
209G/01 $a84$$042090324
237A/01 $aArchivexemplar
```

Datenbeispiel 15: unAPI-Ausgabe im Format PICA+[427]

[427] http://unapi.gbv.de/?id=owc-de-84:ppn:981559603&format=pp (07.05.2018).

```
<?xml version="1.0" encoding="UTF-8"?>
<record xmlns="info:srw/schema/5/picaXML-v1.0">
  <datafield tag="001@">
    <subfield code="0">20</subfield>
    <subfield code="e">1</subfield>
  </datafield>
  <datafield tag="001A">
    <subfield code="0">20:18-04-17</subfield>
  </datafield>
  <datafield tag="001B">
    <subfield code="0">20:18-04-17</subfield>
    <subfield code="t">09:25:08.000</subfield>
  </datafield>
  <datafield tag="001D">
    <subfield code="0">20:18-04-17</subfield>
  </datafield>
  <datafield tag="001F">
    <subfield code="0">1</subfield>
  </datafield>
  <datafield tag="001U">
    <subfield code="0">utf8</subfield>
  </datafield>
  <datafield tag="001X">
    <subfield code="0">0</subfield>
  </datafield>
  <datafield tag="002@">
    <subfield code="0">Lux</subfield>
  </datafield>
  <datafield tag="003@">
    <subfield code="0">981559603</subfield>
  </datafield>
  <datafield tag="010@">
    <subfield code="a">ger</subfield>
  </datafield>
  <datafield tag="011@">
    <subfield code="a">2017</subfield>
  </datafield>
  <datafield tag="021A">
    <subfield code="a">@Zeitlich hochaufgelöste optische und elektrochemische
Untersuchungen beim Gefrieren unterkühlter, wässriger Tropfen</subfield>
    <subfield code="h">von Tillmann Peregrin Buttersack</subfield>
  </datafield>
  <datafield tag="028A">
    <subfield code="d">Tillmann Peregrin</subfield>
    <subfield code="a">Buttersack</subfield>
    <subfield code="B">VerfasserIn</subfield>
    <subfield code="4">aut</subfield>
  </datafield>
  <datafield tag="034D">
    <subfield code="a">1 CD-ROM</subfield>
  </datafield>
  <datafield tag="037A">
    <subfield code="a">$dDissertation$eTechnische Universität  Carolo-Wilhelmina zu
Braunschweig$f2016</subfield>
  </datafield>
  <datafield tag="037A">
    <subfield code="a">Auch als Druckausgabe und Online-Ausgabe</subfield>
  </datafield>
  <datafield tag="037A">
    <subfield code="a">Zusammenfassung in deutscher und englischer
Sprache</subfield>
  </datafield>
  <datafield tag="101@">
    <subfield code="a">20</subfield>
  </datafield>
  <datafield tag="201@" occurrence="01">
    <subfield code="a">Ausleihbestand|Noch nicht verfuegbar</subfield>
    <subfield code="b">0</subfield>
    <subfield code="e">981559514</subfield>
```

```
    <subfield code="m">mon</subfield>
    <subfield code="f">1</subfield>
    <subfield code="u">Ausleihbestand</subfield>
    <subfield code="v">Noch nicht verfuegbar</subfield>
  </datafield>
  <datafield tag="201B" occurrence="01">
    <subfield code="0">18-04-17</subfield>
    <subfield code="t">10:20:18.000</subfield>
  </datafield>
  <datafield tag="201F" occurrence="01">
    <subfield code="0">1</subfield>
  </datafield>
  <datafield tag="201U" occurrence="01">
    <subfield code="0">utf8</subfield>
  </datafield>
  <datafield tag="203@" occurrence="01">
    <subfield code="0">981559514</subfield>
  </datafield>
  <datafield tag="208@" occurrence="01">
    <subfield code="a">18-04-17</subfield>
    <subfield code="b">d</subfield>
  </datafield>
  <datafield tag="209A" occurrence="01">
    <subfield code="a">4209-0324</subfield>
    <subfield code="x">00</subfield>
  </datafield>
  <datafield tag="209G" occurrence="01">
    <subfield code="a">84$042090324</subfield>
  </datafield>
  <datafield tag="237A" occurrence="01">
    <subfield code="a">Archivexemplar</subfield>
  </datafield>
</record>
```

Datenbeispiel 16: unAPI-Ausgabe im Format PICA XML[428]

```
00841nam a2200193 c
450000100100000000300070001000500170001700800410003404000150007504100080009010 0
005200098245015700150300001300307500008500320500004500405500005600450900006400 5
06954007700570RS981559603RSDE-601RS20170418092508.0RS000000s2017
    000 0 ger dRS  USbgerUScGBVCPRS0 USagerRS1 USaButtersack, Tillmann
PeregrinUSeVerfasserInUS4autRS10USaZeitlich hochaufgelöste optische und
elektrochemische Untersuchungen beim Gefrieren unterkühlter, wässriger
TropfenUScvon Tillmann Peregrin ButtersackRS  USa1 CD-ROMRS
USa$dDissertation$eTechnische Universität  Carolo-Wilhelmina zu
Braunschweig$f2016RS  USaAuch als Druckausgabe und Online-AusgabeRS
USaZusammenfassung in deutscher und englischer SpracheRS  USaGBVUSbUB
Braunschweig <84>USd4209-0324USxLUSzLCUSfArchivexemplarRS  US0UB
Braunschweig
<84>USa20USb981559514USc01USd4209-0324USkArchivexemplarUSx0084RSGS
```

Datenbeispiel 17: unAPI-Ausgabe im Format MARC 21[429]

[428] http://unapi.gbv.de/?id=owc-de-84:ppn:981559603&format=picaxml (07.05.2018).
[429] http://unapi.gbv.de/?id=owc-de-84:ppn:981559603&format=marc21 (07.05.2018).

```
<?xml version="1.0" encoding="UTF-8"?>
<record xmlns="http://www.loc.gov/MARC21/slim">
  <leader>xxxxxnam a22yyyyy c 4500</leader>
  <controlfield tag="001">981559603</controlfield>
  <controlfield tag="003">DE-601</controlfield>
  <controlfield tag="005">20170418092508.0</controlfield>
  <controlfield tag="008">000000s2017                    000 0 ger d</controlfield>
  <datafield tag="040" ind1=" " ind2=" ">
    <subfield code="b">ger</subfield>
    <subfield code="c">GBVCP</subfield>
  </datafield>
  <datafield tag="041" ind1="0" ind2=" ">
    <subfield code="a">ger</subfield>
  </datafield>
  <datafield tag="100" ind1="1" ind2=" ">
    <subfield code="a">Buttersack, Tillmann Peregrin</subfield>
    <subfield code="e">VerfasserIn</subfield>
    <subfield code="4">aut</subfield>
  </datafield>
  <datafield tag="245" ind1="1" ind2="0">
    <subfield code="a">Zeitlich hochaufgelöste optische und elektrochemische
Untersuchungen beim Gefrieren unterkühlter, wässriger Tropfen</subfield>
    <subfield code="c">von Tillmann Peregrin Buttersack</subfield>
  </datafield>
  <datafield tag="300" ind1=" " ind2=" ">
    <subfield code="a">1 CD-ROM</subfield>
  </datafield>
  <datafield tag="500" ind1=" " ind2=" ">
    <subfield code="a">$dDissertation$eTechnische Universität  Carolo-Wilhelmina zu
Braunschweig$f2016</subfield>
  </datafield>
  <datafield tag="500" ind1=" " ind2=" ">
    <subfield code="a">Auch als Druckausgabe und Online-Ausgabe</subfield>
  </datafield>
  <datafield tag="500" ind1=" " ind2=" ">
    <subfield code="a">Zusammenfassung in deutscher und englischer
Sprache</subfield>
  </datafield>
  <datafield tag="900" ind1=" " ind2=" ">
    <subfield code="a">GBV</subfield>
    <subfield code="b">UB Braunschweig &lt;84&gt;</subfield>
    <subfield code="d">4209-0324</subfield>
    <subfield code="x">L</subfield>
    <subfield code="z">LC</subfield>
    <subfield code="f">Archivexemplar</subfield>
  </datafield>
  <datafield tag="954" ind1=" " ind2=" ">
    <subfield code="0">UB Braunschweig &lt;84&gt;</subfield>
    <subfield code="a">20</subfield>
    <subfield code="b">981559514</subfield>
    <subfield code="c">01</subfield>
    <subfield code="d">4209-0324</subfield>
    <subfield code="k">Archivexemplar</subfield>
    <subfield code="x">0084</subfield>
  </datafield>
</record>
```

Datenbeispiel 18: unAPI-Ausgabe im Format MARCXML[430]

[430] http://unapi.gbv.de/?id=owc-de-84:ppn:981559603&format=marcxml (07.05.2018).

```
<?xml version="1.0" encoding="UTF-8"?>
<zs:searchRetrieveResponse
xmlns:zs="http://www.loc.gov/zing/srw/"><zs:version>1.1</zs:version><zs:numberOfRec
ords>1</zs:numberOfRecords><zs:records><zs:record><zs:recordSchema>picaxml</zs:reco
rdSchema><zs:recordPacking>xml</zs:recordPacking><zs:recordData><record
xmlns="info:srw/schema/5/picaXML-v1.0">
  <datafield tag="001@">
    <subfield code="0">20</subfield>
    <subfield code="e">1</subfield>
  </datafield>
  <datafield tag="001A">
    <subfield code="0">20:18-04-17</subfield>
  </datafield>
  <datafield tag="001B">
    <subfield code="0">20:18-04-17</subfield>
    <subfield code="t">09:25:08.000</subfield>
  </datafield>
  <datafield tag="001D">
    <subfield code="0">20:18-04-17</subfield>
  </datafield>
  <datafield tag="001F">
    <subfield code="0">1</subfield>
  </datafield>
  <datafield tag="001U">
    <subfield code="0">utf8</subfield>
  </datafield>
  <datafield tag="001X">
    <subfield code="0">0</subfield>
  </datafield>
  <datafield tag="002@">
    <subfield code="0">Lux</subfield>
  </datafield>
  <datafield tag="003@">
    <subfield code="0">981559603</subfield>
  </datafield>
  <datafield tag="010@">
    <subfield code="a">ger</subfield>
  </datafield>
  <datafield tag="011@">
    <subfield code="a">2017</subfield>
  </datafield>
  <datafield tag="021A">
    <subfield code="a">@Zeitlich hochaufgelöste optische und elektrochemische
Untersuchungen beim Gefrieren unterkühlter, wässriger Tropfen</subfield>
    <subfield code="h">von Tillmann Peregrin Buttersack</subfield>
  </datafield>
  <datafield tag="028A">
    <subfield code="d">Tillmann Peregrin</subfield>
    <subfield code="a">Buttersack</subfield>
    <subfield code="B">VerfasserIn</subfield>
    <subfield code="4">aut</subfield>
  </datafield>
  <datafield tag="034D">
    <subfield code="a">1 CD-ROM</subfield>
  </datafield>
  <datafield tag="037A">
    <subfield code="a">$dDissertation$eTechnische Universität  Carolo-Wilhelmina zu
Braunschweig$f2016</subfield>
  </datafield>
  <datafield tag="037A">
    <subfield code="a">Auch als Druckausgabe und Online-Ausgabe</subfield>
  </datafield>
  <datafield tag="037A">
    <subfield code="a">Zusammenfassung in deutscher und englischer
Sprache</subfield>
  </datafield>
  <datafield tag="101@">
    <subfield code="a">20</subfield>
  </datafield>
```

```
  <datafield tag="201@" occurrence="01">
    <subfield code="a">Ausleihbestand|Noch nicht verfuegbar</subfield>
    <subfield code="b">0</subfield>
    <subfield code="e">981559514</subfield>
    <subfield code="m">mon</subfield>
    <subfield code="f">1</subfield>
    <subfield code="u">Ausleihbestand</subfield>
    <subfield code="v">Noch nicht verfuegbar</subfield>
  </datafield>
  <datafield tag="201B" occurrence="01">
    <subfield code="0">18-04-17</subfield>
    <subfield code="t">10:20:18.000</subfield>
  </datafield>
  <datafield tag="201F" occurrence="01">
    <subfield code="0">1</subfield>
  </datafield>
  <datafield tag="201U" occurrence="01">
    <subfield code="0">utf8</subfield>
  </datafield>
  <datafield tag="203@" occurrence="01">
    <subfield code="0">981559514</subfield>
  </datafield>
  <datafield tag="208@" occurrence="01">
    <subfield code="a">18-04-17</subfield>
    <subfield code="b">d</subfield>
  </datafield>
  <datafield tag="209A" occurrence="01">
    <subfield code="a">4209-0324</subfield>
    <subfield code="x">00</subfield>
  </datafield>
  <datafield tag="209G" occurrence="01">
    <subfield code="a">84$042090324</subfield>
  </datafield>
  <datafield tag="237A" occurrence="01">
    <subfield code="a">Archivexemplar</subfield>
  </datafield>
</record></zs:recordData><zs:recordPosition>1</zs:recordPosition></zs:record></zs:r
ecords><zs:echoedSearchRetrieveRequest><zs:version>1.1</zs:version><zs:query>pica.p
pn=981559603</zs:query><zs:maximumRecords>1</zs:maximumRecords><zs:recordPacking>xm
l</zs:recordPacking><zs:recordSchema>picaxml</zs:recordSchema></zs:echoedSearchRetr
ieveRequest></zs:searchRetrieveResponse>
```

Datenbeispiel 19: SRU-Ausgabe im Format PICA XML[431]

[431] http://sru.gbv.de/owc-de-84?version=1.1&operation=searchRetrieve&query=pica.ppn%3D981559603&maximumRecords=1&recordSchema=picaxml (07.05.2018).

Anhang 10 Ausgabe eines Datensatzes per XML-Schnittstelle

Beispieldatensatz: PPN 17226832X

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESULT>
<SESSION>
<SESSIONVAR name="LNG">DU</SESSIONVAR>
<SESSIONVAR name="DB">OWC.5</SESSIONVAR>
<SESSIONVAR name="SID">9640b9c0-1</SESSIONVAR>
<SESSIONVAR name="SET">4</SESSIONVAR>

<SESSIONVAR name="TITLE">OPC4</SESSIONVAR>
</SESSION>
<SET nr="4" hits="0"/>
<LONGTITLE id="17226832X" set="4" format="text">&lt;TR&gt;Titel: &lt;TD&gt;Shipping finance :
financing ships and mobile offshore installations / J. E. Sloggett&lt;TR&gt;VerfasserIn:
&lt;TD&gt;Sloggett, Jolyon E.&lt;TR&gt;Sprache/n:
&lt;TD&gt;Englisch&lt;TR&gt;Veröffentlichungsangabe: &lt;TD&gt;London : Fairplay,
1984&lt;TR&gt;Umfang: &lt;TD&gt;VIII, 116 S : graph. Darst + zahlr.
Tab.&lt;TR&gt;Schriftenreihe: &lt;TD&gt;Ship management series&lt;TR&gt;ISBN: &lt;TD&gt;0-
905045-69-6&lt;TR&gt;
<br />
&lt;TR&gt;Signatur: &lt;TD&gt;10dg14
<br />
&lt;TR&gt;Ausleihstatus: &lt;TD&gt;Ausleihbestand
<br />
Selbst entnehmen (Freihandbestand)</LONGTITLE>
</RESULT>
```

Datenbeispiel 20: Datensatz im XML-Format `text`[432]

Dies entspricht der in XML verpackten HTML-Darstellung der Titelpräsentation im OPAC[433]:

Titel:	Shipping finance : financing ships and mobile offshore installations / J. E. Sloggett
VerfasserIn:	Sloggett, Jolyon E.
Sprache/n:	Englisch
Veröffentlichungsangabe:	London : Fairplay, 1984
Umfang:	VIII, 116 S : graph. Darst + zahlr. Tab.
Schriftenreihe:	Ship management series
ISBN:	0-905045-69-6
Signatur:	**10dg14**
Ausleihstatus:	Ausleihbestand
	Selbst entnehmen (Freihandbestand)

Abb. 22: Titelpräsentation im LBS-OPAC

[432] https://lhemd.gbv.de/DB=OWC.5/XML=1.0/PPN?PPN=17226832X (23.02.2018).
[433] https://lhemd.gbv.de/DB=OWC.5/PPN?PPN=17226832X (23.02.2018).

```
<?xml version="1.0" encoding="UTF-8" ?>
<RESULT>
<SESSION>
<SESSIONVAR name="LNG">DU</SESSIONVAR>
<SESSIONVAR name="DB">OWC.5</SESSIONVAR>
<SESSIONVAR name="SID">9640b9c0-1</SESSIONVAR>
<SESSIONVAR name="SET">4</SESSIONVAR>

<SESSIONVAR name="TITLE">OPC4</SESSIONVAR>
</SESSION>
<SET nr="4" hits="0"/>
<LONGTITLE id="17226832X" set="4" format="extpp">001A -00755:01-12-94-001B -04358:28-11-08-
t08:51:23.000-001D -09999:99-99-99-001F -00-002@ -0Aar-003@ -017226832X-004A -
00905045696-006Y -0BIST563344-006Y -0isl88020283-010@ -aeng-011@ -a1984-019@ -aXA-GB-021A -
aShipping financedfinancing ships and mobile offshore installations-hJ. E. Sloggett-028A -
dJolyon E.-aSloggett-033A -pLondon-nFairplay-034D -aVIII, 116 S-034K -azahlr. Tab.-034M -
agraph. Darst-036E -aShip management series-101@ -a264-201@/01 -aAusleihbestand|Selbst
entnehmen (Freihandbestand)-b0-e1157076807-mmon-f5-uAusleihbestand-vSelbst entnehmen
(Freihandbestand)-201B/01 -027-04-10-t13:27:51.000-201D/01 -027-04-10-b9812-a3264-201F/01 -
00-203@/01 -01157076807-208@/01 -a27-04-10-bz-209A/01 -a10dg14-du-x00-209C/01 -a7581-
x00-209G/01 -a897E$087491-220B/01 -a06542-</LONGTITLE>
</RESULT>
```

Datenbeispiel 21: Datensatz im XML-Format `extpp`[434]

Kategorieende-Zeichen `0x1e` = - Subfield-Kennzeichen `0x1f` = -

[434] https://lhemd.gbv.de/DB=OWC.5/XML=1.0/PPN?PPN=17226832X&PLAIN=ON (23.02.2018).

Anhang 11 OpenRefine: Transformationsregeln

Für Bibliotheken mit lokalen Katalogisaten kann ein Standardregelset zur Transformation in OpenRefine genutzt werden. Diese Transformationen werden dabei durchgeführt:

Umwandlung von Text- in Spaltenformat

- Text transform on cells in column Column 1 using expression grel:"0000 Dummy"
- Split column Column 1 by separator
- Columnize by key column Column 1 1 and value column Column 1 2 with note column

Nicht relevante Kategorien löschen

- Remove column 001@
- Remove column 001A
- Remove column 001B
- Remove column 001D
- Remove column 001F
- Remove column 001X
- Remove column 101@
- Remove column 201@/01
- Remove column 201F/01

Subfield-Kennzeichen entfernen und Spalte umbenennen

- Text transform on cells in column 002@ using expression grel:replace(value, "$0", "")
- Rename column 002@ to MAK 0500
- Text transform on cells in column 003@ using expression grel:replace(value, "$0", "")
- Rename column 003@ to PPN 0100
- Text transform on cells in column 021A using expression grel:replace(value, "$a", "")
- Rename column 021A to TIT 4000
- Text transform on cells in column 011@ using expression grel:replace(value, "$a", "")
- Rename column 011@ to ERJ 1100
- Text transform on cells in column 033A using expression grel:replace(value,"$p", "")
- Rename column 033A to PUB 4030
- Text transform on cells in column 203@/01 using expression grel:replace(value,"$0", "")
- Rename column 203@/01 to EPN 7800
- Text transform on cells in column 220B/01 using expression grel:replace(value, "$a", "")
- Rename column 220B/01 to EXK 4802
- Text transform on cells in column 237A/01 using expression grel:replace(value, "$a", "")
- Rename column 237A/01 to EXK 4801
- Text transform on cells in column 209G/01 using expression grel:replace(value, "$a", "")
- Rename column 209G/01 to BAR 8200

Personenkategorien aufteilen in je eine Spalte für Nachname und Vorname

- Split column 028A by separator
- Text transform on cells in column 028A 1 using expression grel:replace(value, "$a", "")
- Rename column 028A 1 to PER 3000 Name
- Rename column 028A 2 to PER 3000 Vorname
- Split column 028B/01 by separator
- Rename column 028B/01 2 to PER 3001 Name
- Rename column 028B/01 1 to PER 3001 Vorname

Kategorie 208@ / 7001 aufteilen in Datum und Selektionsschlüssel

- Split column 208@/01 by separator
- Rename column 208@/01 2 to SEL 7001
- Text transform on cells in column 208@/01 1 using expression grel:replace(value, "$a", "")
- Rename column 208@/01 1 to DTM 7001

Kategorie 209A / 71xx aufteilen in Sonderstandort, Signatur, Ausleihindikator

- Split column 209A/01 by separator
- Text transform on cells in column 209A/01 2 using expression grel:replace(value, "$x00", "")
- Rename column 209A/01 2 to AUI 7100
- Split column 209A/01 1 by separator
- Rename column 209A/01 1 2 to SGN 7100
- Text transform on cells in column 209A/01 1 1 using expression grel:replace(value, "$f", "")
- Rename column 209A/01 1 1 to SST 7100
- Text transform on cells in column SGN 7100 using expression grel:replace(value, "$x00", "")

Exemplarsatz-Änderungsdatum: Subfield-Kennzeichen entfernen

- Text transform on cells in column 201B/01 using expression grel:replace(value, "$t", " ")
- Text transform on cells in column 201B/01 using expression grel:replace(value,"$0", "")
- Rename column 201B/01 to DAT 7903

ISBN: Subfieldkennzeichen entfernen

- Text transform on cells in column 004A using expression grel:replace(value,"$0", "")
- Text transform on cells in column 004A using expression grel:replace(value, "$A", "=")
- Rename column 004A to ISB 2000
- Text transform on cells in column 004D using expression grel:replace(value,"$0", "")
- Text transform on cells in column 004D using expression grel:replace(value, "$A", "=")
- Rename column 004D to ISB 2009

Wegen des großen Spektrums genutzter Kategorien sind im Anschluss weitere spezifische Transformationen durchzuführen, insbesondere für multiple Kategorien. Abschließend wird die Dummy-Kategorie 0000 gelöscht und eine Spaltensortierung durchgeführt.

Aufgeführt ist die verbale Beschreibung des jeweiligen Transformationsschritts. Auf der beiliegenden CD befindet sich die zugehörige JSON-Datei (`Transformationsregeln-Standard.json`).

Anhang 12 Perl-Programm `opc4_lok_titel` zur Ausgabe bibliografischer Daten

Unter Nutzung der XML-Schnittstelle für PSI-Kataloge extrahiert das Programm die Metadaten von lokalen Katalogisaten aus LBS-OPACs. Die Ausgabe erfolgt im Format PICA+.

Der Programmablauf wird über mehrere Kommandozeilenparameter gesteuert:

- `-s` Standortkürzel, wie in der VZG üblich als LBS-Systemvariable *site* gesetzt
- `-f` `fno` = Bestandsnummer des logischen Bestands in der LBS-Datenbank
- `-n` Anzahl der zu verarbeitenden Titelsätze (Zahl oder `all`)
- `-l` Suchterm (Inhalt von Kategorie `0500`, auch trunkiert möglich)
  Im Programm codiert ist die IKT des Suchschlüssels `MAK` zur Recherche über die Materialart in Kategorie `0500`
- `-d` mit Protokollierung der XML-Quelldaten in separater Datei
- `-p` mit Protokollierung der PPNs in der Protokolldatei (für Testzwecke nützlich)
- `-h` Anzeige des Hilfetextes

Mit den Angaben von Standort(kürzel) und Bestandsnummer kann der betreffende OPAC identifiziert werden. Über das Standortkürzel ermittelt das Programm die URL des zugehörigen OPACs aus einer JSON-Datei. Dort ist zusätzlich für jeden Standort eine Liste der gültigen Bestandsnummern hinterlegt, sodass das Programm eine Validation der Parametereingaben durchführen kann und einen Programmabbruch veranlasst, wenn die als Parameter angegebene Bestandsnummer für den LBS-Standort nicht (mehr) existiert.

Auszug aus der JSON-Datei (Angaben für den Standort Emden):

```
{
    "site" : "emd",
    "adresse" : "https://lhemd.gbv.de/",
    "stelsel" : [ "1", "2", "4", "5", "6" ]
},
```

Für die Suche nach lokalen Katalogisaten wird der Suchschlüssel MAK (=IKT `8600`) verwendet, es ist der Suchbegriff „`l*`“ als Default hinterlegt. Die URL lautet beispielsweise:

```
https://vzlbs.gbv.de/DB=OWC.26/XML=1.0/CMD?ACT=SRCH&IKT=8600&TRM=\l*
```

Die Angabe des Suchbegriffs in der URL muss maskiert werden, da der Suchbegriff ggf. mit einem Suchschlüssel übereinstimmt.

Ausgehend von der erhaltenen Trefferliste werden nun die einzelnen Datensätze über die laufende Nummer innerhalb des Treffersets (Parameter `&FRST`) abgerufen, die Zuordnung zum vorherigen Trefferset erfolgt über eine Session-ID (`SID`).

```
https://vzlbs.gbv.de/DB=OWC.26/SID=c527276f-0/XML=1.0/SHW?SET=1&FRST=1&PLAIN=ON
```

Das Parsen der HTML-Präsentation erschien deutlich aufwendiger, daher wurde die Ausgabe in normalisiertem PICA+-Format (Parameter `&PLAIN=ON`) gewählt.

Das Programm transformiert die XML-Ausgabe in textbasiertes PICA+, zusätzlich wird eine Statistik der verwendeten Kategorien über alle verarbeiteten Titel erstellt.

Als **Eingabedatei** dient `opacs.json`. Die Datei müsste normalerweise angepasst werden, wenn für an einem Standort eine weitere Bibliothek aufgenommen wird. Da aber für neu einzurichtende LBS-Bestände die Funktionen und Befugnisse für lokale Katalogisierung im LBS nicht mehr freigeschaltet werden, ist hier kein Pflegeaufwand notwendig.

Es sind diese **Ausgabedateien** implementiert:

`<site>-<fno>-pica_<timestamp>.txt`
Die Datei enthält die Katalogisate im Format PICA+.

`<site>-<fno>-logfile_<timestamp>.txt`
In der Protokolldatei ist ggf. zusätzlich für jeden Datensatz die PPN aufgeführt, falls dies über den Parameter `-p` beim Programmaufruf aktiviert wurde.

`<site>-<fno>-daten_<timestamp>.txt`
Das Erzeugen der Dumper-Ausgabe kann per Kommandozeilenoption `-d` aktiviert werden.

Es folgt die Beschreibung einiger **spezieller Programmfunktionen**.

Aufgrund der in den Daten enthalten PICA+-spezifischen Steuerzeichen `0x1F` und `0x1E` zur Kennzeichnung von Subfield und Feldende ist die XML-Ausgabe nicht wohlgeformt. Daher werden diese Steuerzeichen durch andere Zeichen(folgen) ersetzt, damit die XML-Daten verarbeitet werden können.

Inzwischen haben einige Bibliotheken mit der Erfassung von Lax-Sätzen im CBS begonnen; diese werden bei der Standard-Suchanfrage „`l*`“ ebenfalls mit berücksichtigt. Diese Lax-Sätze lassen sich über die IPN identifizieren (>100.000.000) und werden bei der Datenausgabe ausgefiltert. Da einige Bibliotheken auch für im LBS erfasste lokale Katalogisate die Materialart „Lax“ verwendet haben, darf diese Materialart nicht von vornherein als Suchanfrage ausgeschlossen werden.

Da die XML-Schnittstelle die Daten im proprietären Pica-Zeichensatz[435] abruft und daher bei der Weiterverarbeitung die Zeichendarstellung oberhalb von Codepoint 127 nicht korrekt ist, werden zur besseren Lesbarkeit zumindest die Umlaute und das ß-Zeichen für die Datenausgabe nach ISO 8859-1 (Latin1) umgewandelt.

Diese Fälle durchlaufen im Programm eine besondere **Fehlerbehandlung**:

Logisch gelöschte Titel: Beim Löschen von lokalen Katalogisaten per WinIBW bleiben zunächst die Indexeinträge erhalten. Diese werden täglich über ein OPAC-Dienstprogramm aktualisiert (Programm `update_index`); einmal pro Woche findet ein kompletter Index-Neuaufbau (Programm `create_index`) statt. Daher findet das Perl-Programm zuweilen solche bereits logisch gelöschten Titel; die reguläre OPAC-Präsentation zeigt in diesen Fällen als Titelanzeige den Text "TITEL NICHT IN DER DATENBANK VORHANDEN ???". Diese Fälle werden über einen eigenen Zähler erfasst.

Doppelte PPNs: Werden beim Anlegen eines neuen lokalen Katalogisats aus Gründen der Arbeitsersparnis die Kategorien eines bereits vorhandenen Datensatzes in den Datensatzerfassungs-Schirm kopiert, so wird beim Speichern des neuen lokalen Katalogisats im Datensatz zusätzlich zur neu vergebenen PPN auch die PPN des als Kopie genutzten Datensatzes gespeichert. Diese Fälle von doppelt vorhandenen PPN-Kategorien werden protokolliert. Zurzeit kann das Transferverfahren mit

[435] Die Zeichen 0-127 entsprechen dem ASCII-Zeichensatz, der Bereich 128-255 ist abweichend von ISO 8895-1 belegt, vgl. Jong 2004.

diesen Fällen noch nicht korrekt umgehen. Es sind Anpassungen in einem der verwendete Scripte erforderlich, da es anderenfalls zu irrtümlichen Datensatzlöschungen käme.[436]

Mehrere Exemplare: In der Vergangenheit wurden für einige Bibliotheken Daten direkt ins LBS eingespielt; hierbei ist üblicherweise eine sonst aktive Validation außer Kraft gesetzt. Daher sind auch lokale Katalogisate mit mehr als einem Exemplarsatz gespeichert worden. In den betreffenden Standorten sind zwischenzeitlich die meisten Datensätze dieser alten Importe gelöscht. Falls dennoch solche Datensätze gefunden werden, werden diese ebenfalls protokolliert.

Anmerkungen

Nachteilig ist die lange Programmlaufzeit: pro Minute werden <400 Datensätze verarbeitet. Verschiedene Tests zeigten, dass die Laufzeit abhängig von der sonstigen Inanspruchnahme der LBS-Daten sowie der Leistungsfähigkeit des Servers ist. Der Zugriff über HTTP dauert natürlich um ein Vielfaches länger als bei einem Export per `csft_ttlextract` mit direktem Zugriff auf die LBS-Datenbanktabellen. Da aber die Anzahl der lokalen Katalogisate in 80 von 100 Bibliotheken niedriger als 10.000 ist, beträgt die Laufzeit bei einer solchen Datenmenge weniger als 20 Minuten. Daher ist das Programm gut geeignet, nach Bedarf einen Datenexport für diese 80 Bibliotheken durchzuführen.

Die Kategorien-Statistik, die lediglich das Vorkommen der PICA+-Kategorien über die Datenmenge einer Datei zählt, könnte um weitere Funktionen ergänzt werden. Analog zu den entsprechenden Funktionen bei Catmandu und/oder MARCeL wäre insbesondere die Maximalzahl Vorkommen einer PICA+-Kategorie innerhalb eines Datensatzes von Bedeutung. Mit dieser Information könnten zunächst kategoriespezifische OpenRefine-Transformationsregeln erstellt werden, aus deren Menge bei der Transformation einer Datei die passenden Regeln gewählt und ausgeführt werden könnten.

Eine weitere Optimierung wäre die Spezifizierung der Statistik pro PICA+-Kategorie nach Subfields. Damit wäre eine genauere Übersicht über die erfassten Datenelemente gegeben.

Ein Desiderat ist das Entfernen der letzten (leeren) Ausgabezeile, da bei der Weiterverarbeitung in OpenRefine diese Zeile anderenfalls manuell gelöscht werden muss.

Änderungen an der Adresse eines LBS-Servers müssen in der Excel-Datei nachgeführt werden.

Auf der beiliegenden CD im Verzeichnis „2 – ETL“:

- `opc4_lok_titel.pl`
- `opacs.json`
- Für drei Programmläufe (wegen unterschiedlicher Datenkonstellation):
  - `*-pica_20180511-*.txt` Ausgabe: PICA+-Daten
  - `*-logfile_20180511-*.txt` Protokoll
  - `*-daten_20180511-*.txt` XML-Daten (Dumper)
  - `*-pica_20180511-*.xlsx` Export aus OpenRefine
- `Transformationsregeln-Standard.json`

[436] Hierzu wurde für das Projektteam eine ausführliche Fehlerbeschreibung mit Lösungsvorschlag erstellt.

Anhang 13 VB-Script `LoksatzLoeschen.vbs` zum Löschen lokaler Katalogisate

Das Script erwartet eine Excel-Datei mit PPNs oder IPNs. Eine solche Eingabedatei kann entweder per SQL-Statement aus der LBS-Datenbank erzeugt werden (dort sind IPNs gespeichert) oder aber per WinIBW-Funktion „Exceltabelle erstellen“ in CAT4[437] (mit PPNs).

Wie bei WinIBW-Scripten üblich simuliert das Script die Schirmfunktionen mit Hilfe PICA-spezifischer VBS-Funktionen. Dabei werden Kommandos bzw. das Auslösen einer Schaltfläche oder einer Taste an das LBS übermittelt. Das Script wertet die vom LBS zurückgegebenen (Bildschirm-)Meldungen aus.

Vor dem eigentlichen Beginn des Löschvorgangs wird der Anwender durch eine Reihe von Abfragen geführt, zur Steuerung der gewünschten Verarbeitungsbedingungen.

Für jede PPN/IPN wird zunächst ein Löschvorgang angestoßen. Ist ein zur Ausleihverbuchung notwendiger Bandsatz (Tabelle `volume`) vorhanden, wird eine entsprechende Meldung ausgegeben. Falls bei der Scriptsteuerung entsprechend aktiviert, erfolgt eine Prüfung auf ggf. vorhandene Vormerkungen, Forderungen oder eine Ausleihe für diesen Bandsatz; dies ist über entsprechende SQL-Datenbankabfragen realisiert. Im positiven Fall wird lediglich eine entsprechende Meldung in die Excel-Datei geschrieben. Hatte der Anwender auch eine Prüfung auf Verknüpfungen zu einer Bestellung im Erwerbungsmodul angefordert, wird vor dem Löschen noch eine SQL-Abfrage auf die betreffende LBS-Tabelle durchgeführt. Wenn auch hier keine Verknüpfungen gefunden wurden, wird der Löschvorgang fortgesetzt.

Für alle Script-Situationen (*gelöscht* bzw. *nicht gelöscht*) wird eine spezifische Meldung in die Excel-Datei geschrieben, ebenfalls ein Timestamp, über den sich die Laufzeit des Scriptes ablesen lässt. Konnte der Datensatz nicht gelöscht werden, wird zusätzlich die EPN des Datensatzes in die Excel-Datei eingefügt; so kann der Anwender die problematischen Datensätze gezielt über die EPN im LBS recherchieren.[438]

	A	B	C	D	E	F	G	H
1	PPN	EPN	Timestamp	Meldungen				
2	991041585	99104153	13.12.2017 - 19:04:01	ACQ-Verknüpfung vorhanden; L-Satz wurde nicht gelöscht;				
3	991041577	99104152	13.12.2017 - 19:04:02	ACQ-Verknüpfung vorhanden; L-Satz wurde nicht gelöscht;				
4	991041569		13.12.2017 - 19:04:06	Titel im LBS gelöscht;				
5	991109236	99110918	13.12.2017 - 19:04:08	OUS-Verknüpfung vorhanden; L-Satz wurde nicht gelöscht;				
6	991104773	99110472	13.12.2017 - 19:04:09	OUS-Verknüpfung vorhanden; L-Satz wurde nicht gelöscht;				
7	991102789		13.12.2017 - 19:04:13	Titel im LBS gelöscht;				
8	991099842	99109979	13.12.2017 - 19:04:15	OUS-Verknüpfung vorhanden; L-Satz wurde nicht gelöscht;				
9	991094271		13.12.2017 - 19:04:18	Titel im LBS gelöscht;				
10	99107145X	99107140	13.12.2017 - 19:04:20	OUS-Verknüpfung vorhanden; L-Satz wurde nicht gelöscht;				
11	991069749		13.12.2017 - 19:04:24	Titel im LBS gelöscht;				
12	991064771		13.12.2017 - 19:04:28	Titel im LBS gelöscht;				
13	991063147		13.12.2017 - 19:04:31	Titel im LBS gelöscht;				

Abb. 23: Excel-Tabelle mit Ausgabe des VB-Scripts zum Löschen lokaler Katalogisate

Das Script beginnt mit der Verarbeitung in der Zeile mit der ersten leeren Zelle der Timestamp-Spalte, damit kann eine Datei in mehreren Teilen verarbeitet werden.

[437] Siehe Kap. 5.2.2.
[438] OUS-Bandsätze bzw. ACQ-Bestelldatensätze sind im LBS mit dem Exemplarsatz verknüpft.

Hat der Anwender bei der Scriptsteuerung den Simulationsmodus aktiviert, durchläuft das Script alle Verarbeitungsschritte bis auf den tatsächlichen Löschvorgang. Vorwiegend zu Testzwecken ist ein Debugmodus implementiert, dieser muss allerdings im Quellcode des Scripts aktiviert werden. Dann werden für jeden Verarbeitungsschritt Zwischenmeldungen auf dem Bildschirm ausgegeben, teilweise auch die jeweils aktuellen Werte der WinIBW-Status-Variablen bzw. scriptinternen Variablen.

Als Grundlage diente ein Script von Jarmo Schrader, UB Hildesheim. Dessen Scriptversion umfasst 116 Zeilen und ist für den Eigengebrauch gedacht. Es werden IPNs verarbeitet, die über eine SQL-Abfrage ermittelt werden.

Um eine Ausführung durch Systemverwalter der LBS-Standorte zu ermöglichen, musste das Script erheblich erweitert werden, nicht zuletzt um die heterogene Datensituation in den Bibliotheken zu berücksichtigen. Ebenfalls sollte die Verarbeitung von PPN-Listen ermöglicht werden, die per WinIBW-Recherche in CAT4 durch Fachabteilungen wie z. B. Fernleihe erzeugt wurden. Eine Kontrolle auf ggf. vorhandene LBS-Tabellenverknüpfungen ist in CAT4 nicht möglich, im LBS4-Client hingegen sind solche Prüfungen nur auf Einzelfallbasis möglich.

Daher wurden vor allem die folgenden Ergänzungen vorgenommen:

- Strukturierte Variablenbezeichnungen
- Verarbeitung von wahlweise IPNs oder PPNs
- Programmlogik mit interaktiven Abfragen und ausführlicher Fehlerbehandlung
- Datenbankzugriff per ODBC mit Abfragen auf Ausleihen/Vormerkungen/Forderungen und (ACQ-)Bestellungen
- Bildschirmausgabe für Simulationsmodus und Debugmodus erweitert bzw. angepasst
- Ausgabe von EPN und Timestamp im Logfile
- Schlussprotokoll per MessageBox

Die Programmschritte sind umfassend kommentiert, Screenshots der interaktiven Programmausführung sind der Dokumentation zu entnehmen.

Auf der beiliegenden CD im Verzeichnis „3 – Bereinigung“:

- `Dokumentation_Datensätze im LBS löschen.pdf` Dokumentation
- `LoksatzLoeschen.vbs` Quellcode

Verzeichnis der auf der beiliegenden CD gespeicherten Materialien

0 – Dokumentation

- Katalogisierungsrichtlinie für Lokale Katalogisierung `Richtlinie-LOK-FINAL.pdf`
- Präsentation `Klute-LOK-Transfer.pptx`, gehalten beim LBS-Systemverwaltertreffen am 26.09.2018

1 – Bestandsaufnahme

- `opc4_lok_anzahl.pl` Quellcode
- `LOK-Tabelle.xlsx` Eingabe-/Ausgabedatei
- Für zwei Programmläufe mit verschiedenen Kommandozeilenoptionen
  - `lok-logfile_*.txt` Protokolldatei
  - `lok-search_urls_*.txt` Ausgabedatei mit URLs
  - `lok-daten_*.txt` Dumper-Ausgabe

2 – ETL

- `opc4_lok_titel.pl` Quellcode
- `opacs.json`
- Für drei Programmläufe (wegen unterschiedlicher Datenkonstellation)
  - `*-pica_*.txt` Ausgabe: PICA+-Daten
  - `*-logfile_*.txt` Protokoll
  - `*-daten_*.txt` XML-Daten (Dumper)
  - `*-pica_*.xlsx` Export aus OpenRefine
- `Transformationsregeln-Standard.json`

3 – Bereinigung

- `Dokumentation_Datensätze im LBS löschen.pdf` Anleitung für Anwender
- `LoksatzLoeschen.vbs` Quellcode

`README-Inhaltsverzeichnis.txt`

`Klute_Thesis_ETL-Prozesse.pdf` Master Thesis

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Abschlussarbeit selbstständig verfasst und die verwendeten Quellen deutlich gekennzeichnet habe.

Ich stimme einer Veröffentlichung durch das WIT zu.

Wildau, 22. Mai 2018 ----------

Ursula Klute