

Deskriptorenportal

(Abbildung von Personendescriptoren mit der Records in Contexts-Ontology)

Name: Gionathan Diani

Studiengang: MSc Data Visualization

E-Mail: mail@gionathandiani.name

Modul: Knowledge Engineering and Extraction

Referent: Prof. Dr. habil. Albert Weichselbraun

Abgabedatum: 15.01.2025

Inhaltsverzeichnis

Abbildungsverzeichnis	II
Tabellenverzeichnis	III
Abkürzungsverzeichnis	IV
Programmcodeverzeichnis	V
1 Einleitung	1
2 Datenaggregation und -modellierung	2
2.1 Datenaggregation	2
2.1.1 Export AIS	2
2.1.2 Aggregation der Deskriptorrelationen	2
2.2 Datenmodell	4
3 Software	7
3.1 Datenaggregation und -aufbereitung	7
3.1.1 scraper.py	7
3.1.2 sling.py	8
3.1.3 converter.py	8
3.1.4 reasoner.py	9
3.2 qllever	9
3.3 Webapplikation	9
3.3.1 Suche	9
3.3.2 Zeitstrahl	9
3.3.3 Deskriptorendetailansicht	10
4 Diskussion	12
4.1 Datenmodellierung	12
4.2 Reasoner	12
4.3 Visualisierung	14
5 Fazit	15
Literaturverzeichnis	16
A Datenmapping	18

Abbildungsverzeichnis

Abb. 1:	Datenmodell / Datenmapping der Daten mittels RiC-O	6
Abb. 2:	C4 Container-Diagramm des Systemkontexts «Data-Factory»	7
Abb. 3:	Screenshot Deskriptorensuche im Archivkatalog der Burgerbibliothek Bern	8
Abb. 4:	C4 Container-Diagramm der Webapplikation.	9
Abb. 5:	Screenshot der Suchmaske	10
Abb. 6:	Screenshot der Visualisierung «Zeitstrahl»	10
Abb. 7:	Screenshot der Deskriptorendetailansicht zu Berchtold Haller	11

Tabellenverzeichnis

Tab. 1:	Beschreibung Datenfelder aus AIS-Export	2
Tab. 2:	Auflistung und Beschreibung der aggregierten Relationstypen.	3
Tab. 3:	Übersicht Namensräume und Präfixe	5
Tab. 4:	Schema in welches die Relationstypen abgelegt werden.	8
Tab. 5:	Übersicht Datenmapping	19

Abkürzungsverzeichnis

AIS	Archivinformationssystem
ARK	Archival Resource Key
BBB	Bürgerbibliothek Bern
EGAD	Expert Group on Archival Description
ICA	International Council of Archives
RiC	Records in Contexts
RiC-O	Records in Contexts-Ontology

Programmcodeverzeichnis

Programmcode 1	Auszug der RiC-O; Definition von rico:hasSibling	13
Programmcode 2	SPARQL-Abfrage zur Ermittlung von Geschwister	13
Programmcode 3	Implementation des Identifiers Scope-ID	20
Programmcode 4	Beispiel Identifier ARK	20
Programmcode 5	Beispiel für den Eventtyp «Baptism»	20
Programmcode 6	Beispiel für den Eventtyp «Burial»	20
Programmcode 7	Beispiel demografische Gruppe «SexFemale»	20
Programmcode 8	Beispiel demografische Gruppe «SexMale»	21
Programmcode 9	Beispiel Lebenspartner «hasOrHadLifePartner»	21
Programmcode 10	Beispiel Person	21

1. Einleitung

Records in Contexts (RiC) ist ein archivarischer Verzeichnungsstandard welcher die amtierenden Standards ISAD(G), ISAAR(CPF), ISDF und ISDIAH ablöst (EGAD, 2024). Bestandteil des Standards ist die Records in Contexts-Ontology (RiC-O), welche als Blaupause zur Erstellung von RDF-Datensätzen im archivischen Kontext zu verstehen ist (ICA EGAD, 2024). Die Burgerbibliothek Bern (BBB) führt u. a. Thesauri zu Personen, Orten oder Sachen und verwendet Deskriptoren¹ bei der Verzeichnung ihrer Archivalien. Glarner (2024, S. 539, 551) stellt fest, dass die in der BBB erstellten Deskriptoren über Charakteristiken von RDF-Tripeln aufweisen, wobei aufgrund der Beschränkungen des aktuellen Archivinformationssystems (AIS), deren volles Potenzial, insbesondere in der Vermittlung, nicht ausgeschöpft werden. Die vorliegende Arbeit verwendet die Deskriptoren aus dem Thesaurus «Personen» der Burgerbibliothek Bern, um einen RDF-Datensatz anhand der RiC-O zu modellieren (Kap. 2.2). Die Daten wurden dazu einerseits direkt über einen CSV-Dateiexport und andererseits mittels Web Scraping des Archivkatalogs aggregiert (Kap. 2.1). Anschliessend erfolgte eine Aufbereitung der Daten mithilfe der Python-Bibliotheken RDFLib und OWL-RL zu einem RDF-Datensatz (Kap. 3.1). Zur Exploration des Datensatzes wurde eine Webapplikation erstellt, welche eine einfache Suche und die Visualisierung von Lebensdaten und Verwandtschaftsbeziehungen umfasst (Kap. 3.3). In der abschliessenden Diskussion werden Herausforderungen im Zusammenhang der Umsetzung besprochen (Kap. 4).

¹Unter «Deskriptoren» sind Einträge eines Thesaurus zu verstehen.

2. Datenaggregation und -modellierung

Nachfolgendes Kapitel behandelt das zweistufige Datenaggregationsverfahren (Kap. 2.1) und die Datenmodellierung (Kap. 2.2) anhand der Records in Contexts-Ontology (RiC-O).

2.1. Datenaggregation

Die Datenaggregation erfolgte in zwei Schritten: Angaben zur Person – bspw. Bezeichnung oder Lebensdaten – konnten über das AIS im CSV-Dateiformat exportiert werden (Kap. 2.1.1). Relationen zwischen den Deskriptoren wurden mittels Web-Scraping aggregiert (Kap. 3.1.1).

2.1.1. Export AIS

Die Deskriptoren der Burgerbibliothek Bern werden im proprietären AIS «ScopeArchive» verwaltet. Das AIS ermöglicht den Export von bis zu 50 000 Einträgen im CSV-Dateiformat. Vorgenommener Export besteht aus Einträgen des Thesaurus «Natürliche Personen», umfasst 49 242 Einträge und verfügt über die in Tabelle 1 ersichtlichen Datenfelder.

Tabelle 1

Beschreibung Datenfelder aus AIS-Export

Bezeichnung	Beschreibung
ARK	Archival Resource Key; persistenter Identifier des Deskriptors.
ID-Nr.	Entspricht dem Primärschlüssel im AIS; der ganzzahlige Wert wird bei der Erstellung eines Deskriptors automatisch erstellt und ist unveränderlich.
ID-Name	Der ID-Name ist einzigartig und wird automatisch anhand des Felds «Bezeichnung» erstellt. Der Wert kann verändert werden.
Beschreibung	Ein Freitextfeld mit unbeschränkter Zeichenanzahl, welches der Beschreibung von Deskriptoren dient.
Geschlecht	Angabe zum biologischen Geschlecht. Kann den Wert «männlich» oder «weiblich» enthalten.
Geburtsdatum	Die Felder sind als Freitextfeld konfiguriert und sind zur Eingabe von einem Datum vorgesehen. Als Konvention, gelten die nachfolgenden Schreibweisen: YYYY.mm.dd, YYYY.mm. oder YYYY. Vielfach ist aber auch eine Zeichenkette wie z. B. «1. Jhr.» hinterlegt.
Taufdatum	
Todesdatum	
Begräbnisdatum	

2.1.2. Aggregation der Deskriptorrelationen

Relationen zwischen den Deskriptoren, welche bspw. Verwandtschaftsbeziehung beschreiben, sind nicht über das AIS exportierbar. Die Möglichkeit, die Daten direkt über die dahinter-

liegende Datenbank zu exportieren, entfällt aufgrund fehlender Dokumentation über das Datenbankschema. Angesichts dessen erfolgte deren Aggregation mittels Web-Scraping des Online-Archivkatalogs (mehr dazu in Kap. 3.1.1). Resultat sind 82 027 weitere Einträge, wobei sich diese in die gem. Tab. 2 aufgeführten Relationstypen gliedern. Aufgrund fehlerhaft verknüpfter Deskriptoren sind Relationstypen aufgelistet, welche nicht auf Personendescriptoren verweisen dürfen (z. B. «Eigentum von»).

Tabelle 2

Auflistung und Beschreibung der aggregierten Relationstypen.

Bezeichnung	Beschreibung
Bekannte Person	Verweis auf einen Personendescriptor; Person waren oder sind einander bekannt.
Bürger / Bürgerin von (Ort)	Verweis auf einen Ortsdescriptor; Person war oder ist ein:e Bürger:in des Ortes.
Bürgerort von	Inverse zu «Bürger / Bürgerin von (Ort)»; fehlerhafte Verknüpfung.
Ehepartner / Ehepartnerin	Verweis auf einen Personendescriptor; Personen waren oder sind miteinander verheiratet.
Eigentümer / Eigentümerin	Verweis auf einen Sachdescriptor; Person war oder ist ein:e Eigentümer:in der Sache.
Eigentum von	Inverse zu «Eigentümer / Eigentümerin»; fehlerhafte Verknüpfung.
Eltern	Verweis auf einen Personendescriptor; Person ist das Kind von der Person, auf welche verwiesen wird.
Erwähnung von	Inverse zu «Literatur»; fehlerhafte Verknüpfung.
geboren in (Ort)	Verweis auf einen Ortsdescriptor; die Person ist an diesem Ort geboren.
Geburtsort von	Inverse zu «geboren in (Ort)»; fehlerhafte Verknüpfung.
gestorben in (Ort)	Verweis auf einen Ortsdescriptor; Ort an welchem Person verstarb.
Geschwister	Verweis auf einen Personendescriptor; die Personen sind Geschwister.
hat gewirkt in (Ort)	Verweis auf einen Ortsdescriptor; Person hat an diesem Ort gewirkt.

hat gewohnt in (Ort)	Verweis auf einen Ortsdeskriptor; Person wohnt oder hat an diesem Ort gewohnt.
Kind	Verweis auf einen Personendeskriptor; Person ist ein Elternteil, von der Person, auf welche verwiesen wird.
Lebenspartner / Lebenspartnerin	Verweis auf einen Personendeskriptor; die Personen waren oder sind in einer romantischen Beziehung, ohne verheiratet (gewesen) zu sein.
Literatur	Verweis auf einen Sachdeskriptor; weiterführende Literatur zur Person.
siehe auch	Verweis auf Deskriptor, mit weiterführenden Inhalten.
siehe unter	Inverse zu «verwendet für»
Todesort von	Inverse zu «gestorben in (Ort)»; fehlerhafte Verknüpfung.
Verwandte Person	Verweis auf einen Personendeskriptor; Die Personen sind miteinander verwandt.
verwendet für	Entspricht der nicht bevorzugten Bezeichnungsvariante eines Deskriptors.
Wirkungsort von	Inverse zu «hat gewirkt in (Ort)»; fehlerhafte Verknüpfung.
Wohnort von	Inverse zu «hat gewohnt in (Ort)»; fehlerhafte Verknüpfung.

2.2. Datenmodell

Die Datenmodellierung (Abb. 1) erfolgte anhand des Records in Contexts Standards (RiC) resp. der Records in Contexts-Ontology (RiC-O). RiC ist ein vom International Council of Archives (ICA) verabschiedeter Standard, welcher der Beschreibung von Archivalien dient und von der Expert Group on Archival Description entwickelt wird (EGAD, 2024). RiC-O ist als generische Blaupause zur Erstellung von RDF-Datensätzen für den Gebrauch im archivalischen Kontext entworfen und insofern auf Erweiterbarkeit ausgelegt (ICA EGAD, 2024, Abs. «RiC-O design principles»). Während Umsetzungen von Systemen oder Fallstudien, welche RiC-O nutzen, existieren – z. B. Spec (2023) oder De Coulon (2024) – fehlt eine etablierte Standardimplementation. Die Datenmodellierung erfolgte unter Anwendung der in Tab. 3 aufgeführten Namensräumen und Präfixe:

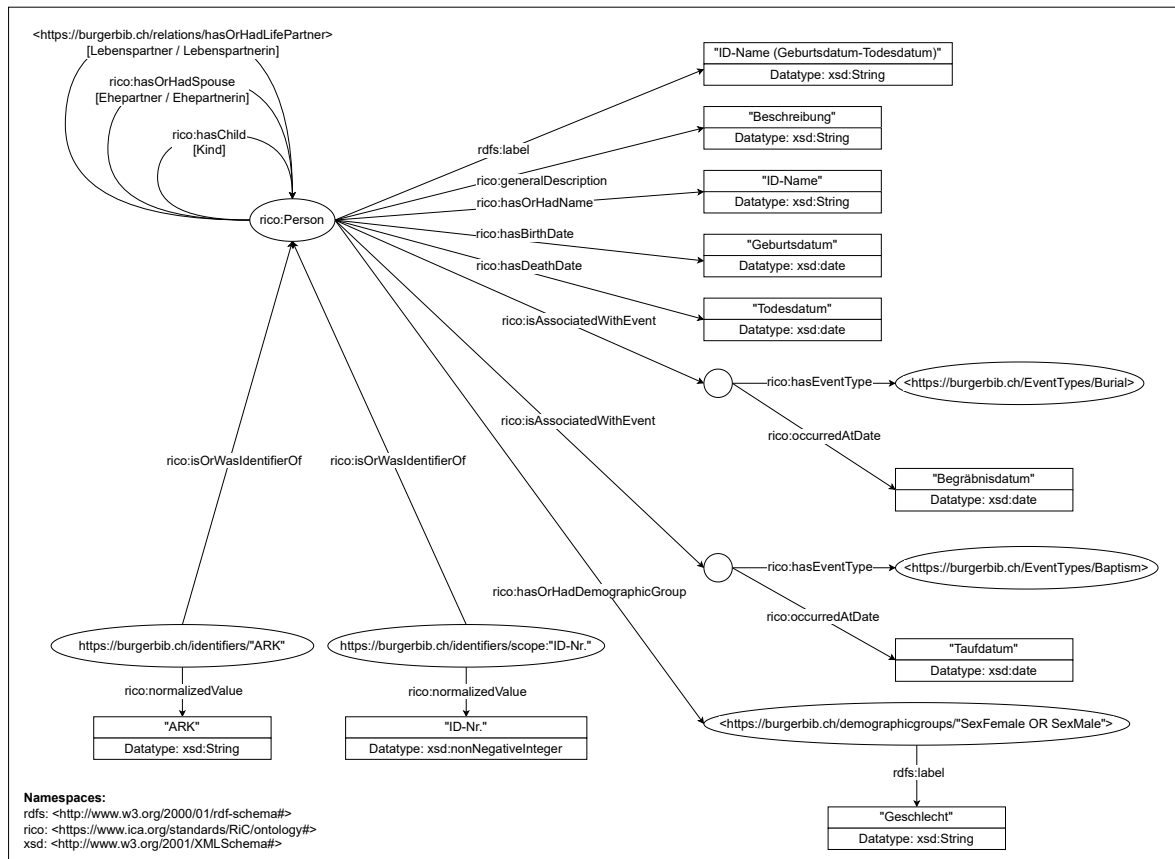
Tabelle 3*Übersicht Namensräume und Präfixe*

Präfix	IRI Namensraum
	https://burgerbib.ch/indexterms/ark:36599/
	https://burgerbib.ch/DemographicGroups/
	https://burgerbib.ch/EventTypes/
	https://burgerbib.ch/IdentifierTypes/
	https://burgerbib.ch/relations/
bio	http://purl.org/vocab/bio/0.1/
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
rico	https://www.ica.org/standards/RiC/ontology#
wd	http://www.wikidata.org/wiki/
xsd	http://www.w3.org/2001/XMLSchema#

Eine tabellarische Übersicht des Datenmappings und Beispiele in der Turtle-Syntax sind Ahg. A zu entnehmen. Die Deskriptoren werden im Namensraum `<https://burgerbib.ch/indexterms/ark:36599/>` als Instanzen von der Klasse `«rico:Person»` instantiiert. Die Modellierung der Identifier `«ID-Nr.»` und `«ARK»`, als `«rico:IdentifierTypes»`, ist dem Matterhorn RDF Metadatenmodell nachempfunden und folgt der dortigen Beispielimplementation einer `«URN»` (docuteam AG, n. d.). Der Identifiertyp `«ARK»` wird zudem mittels der Annotations-eigenschaft `«rico:closeTo»`, welche dem Verweisen auf Elemente von weiteren geeigneten Ontologien oder Modelle dient, mit einem Wikidata-Eintrag, welcher den ARK beschreibt, verknüpft (`«wd:Q2860403»`). Für das Abbilden demografischer Merkmale stellt RiC-O die Klasse `«rico:DemographicGroup»` zur Verfügung. Die Implementation von `«Geschlecht»` erfolgt durch Definition entsprechender Subklassen in einem eigens definierten Namensraum für demografische Gruppen (`«https://burgerbib.ch/DemographicGroups/SexFemale»` und `«https://burgerbib.ch/DemographicGroups/SexMale»`). Das Geburts- und Todesdatum sind über die Eigenschaften `«rico:hasBirthDate»` und `«rico:hasDeathDate»` realisiert. Die Abbildung von Tauf- und Begräbnisdatum ist mittels der Definition korrespondierender Subklassen von `«rico:EventTypes»` (`«https://burgerbib.ch/EventTypes/Baptism»` und `«https://burgerbib.ch/EventTypes/Burial»`). Zusätzlich verweisen die Eventtypen über `«rico:closeTo»` auf die analogen Einträge des `«BIO vocabulary»` – eine Ontologie zur Beschreibung von Biografien (Davis & Galbraith, 2011) – `«bio:Baptism»` und `«bio:Burial»`. Für die Relation `«Lebenspartner / Lebenspartnerin»` wird eine Subproperty `«https://burgerbib.ch/relations/hasOrHadLifePartner»` von `«rico:hasFamilyAssociationWith»` definiert, angelehnt an die in RiC-O bestehende Eigenschaft `«rico:hasOrHadSpouse»`, welche für die Relation `«Ehepartner / Ehepartne-`

rin» verwendet wird. Nicht alle aggregierten Daten werden gemappt. Als einzige Verwandtschaftsbeziehung wird definiert, wer Kind von einer Person ist («rico:hasChild»). Weitere Verwandtschaftsgrade werden anschliessend mittels Inferenzbildung ermittelt (z. B. die Inverse zu «rico:hasChild» ist «rico:isChildOf»); vgl. Kap. 4.2). Die Modellierung von Orts- oder Sachdeskriptoren ist nicht als Teil des Projekts vorgesehen, weshalb Relationen zwischen Personendeskriptor und Orts- oder Sachdeskriptoren nicht berücksichtigt werden.

Abbildung 1
Datenmodell / Datenmapping der Daten mittels RiC-O



Anmerkung: Eigene Darstellung.

3. Software

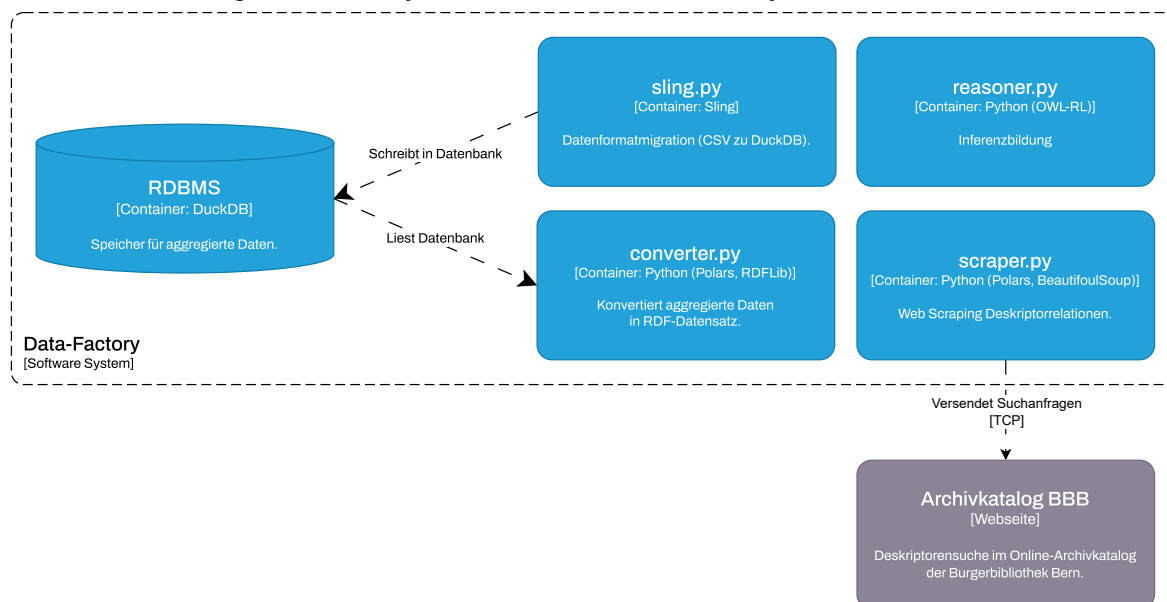
Der Programmcode des Projekts gliedert sich in die zwei Hauptbestandteile «Datenaggregation und -aufbereitung» (Kap. 3.1) sowie «Webapplikation» (Kap. 3.3). Das folgende Kapitel behandelt die Softwarearchitektur der Bestandteile. Sämtlicher Programmcode steht unter der URL <https://purl.org/giodi/deskriptorenportal> zur Verfügung.

3.1. Datenaggregation und -aufbereitung

Der Systemkontext «Data-Factory» enthält den Programmcode, für das Aggregieren und Aufbereiten der Daten (Abb. 2). Nachfolgende Kapitel beschreiben die Bestandteile des Kontexts.

Abbildung 2

C4 Container-Diagramm des Systemkontexts «Data-Factory»



Anmerkung: Eigene Darstellung.

3.1.1. scraper.py

Der Online-Archivkatalog der Burgerbibliothek Bern verfügt über eine dedizierte Deskriptorensuche (Abb. 3). Dabei entspricht das Eingabefeld «Bezeichnung» in der Suchmaske dem «ID-Namen» (vgl. Tab. 1). Über die Suchbeschränkung, nur im Thesaurus «Personen» zu suchen und den Vergleichsoperator «ist gleich» zu nutzen, sind Suchabfragen mit exakten Treffern möglich. Die HTTP-Antwort auf eine entsprechende Suchabfrage ist ein HTML-ähnlicher Code vom Inhaltstyp «text/plain; charset=utf-8», welcher die Suchresultate enthält. Das Programm verwendet die «ID-Name»-Einträge aus dem AIS-Export (vgl. Kap. 2.1.1), um iterativ Suchanfragen zu versenden. Die HTTP-Antwort wird mithilfe der Python-Bibliothek «Beautiful Soup» (Richardson, 2024) als HTML geparkt, nach relevanten HTML-Elementen durchsucht und die benötigten Angaben extrahiert. Die Ergebnisse werden gem. dem in Tab. 4 ersichtlichen Schemas in eine CSV-Datei gespeichert.

Abbildung 3
Screenshot Deskriptorensuche im Archivkatalog der Burgerbibliothek Bern

Deskriptorensuche
Suchen Sie in einem Thesaurus nach Deskriptoren. Klicken Sie auf die Anzahl Verknüpfungen um direkt nach den damit verknüpften Archivalien zu suchen. Alternativ fügen Sie einzelne Begriffe einer Liste hinzu, um später nach den Archivalien zu suchen die mit diesen Deskriptoren verknüpft sind.

1. Schritt: Suchen der Deskriptoren	2. Schritt: Suchen der Archivalien																								
<p>Thesaurus: <input type="text" value="Personen"/></p> <p>Bezeichnung: <input type="text" value="ist gleich"/> <input type="text" value="Hallwyl, Kaspar von (14??-1564)"/></p> <p><input type="button" value="Suchen"/></p> <p>Klicken Sie auf Hinzufügen, um die Deskriptoren in der Liste rechts hinzuzufügen.</p>	<p>Es sind noch keine Deskriptoren ausgewählt.</p> <p>Suchen Sie zuerst nach Deskriptoren und fügen Sie diese dann der Suchauswahl hinzu.</p>																								
<table border="1"><thead><tr><th>Bezeichnung</th><th>Anz. Verknüpfungen</th><th>Hinzufügen</th></tr></thead><tbody><tr><td>Hallwyl, Kaspar von (14??-1564) (Personen\Natürliche Personen\H)</td><td>1</td><td><input type="button" value="Hinzufügen"/></td></tr><tr><td>Literatur: HBL S. (Bibliographie\Nachschlagewerke\H)</td><td>130</td><td><input type="button" value="Hinzufügen"/></td></tr><tr><td>Katalog, Galerie Fischer (Bibliographie\Nachschlagewerke\K)</td><td>8</td><td><input type="button" value="Hinzufügen"/></td></tr><tr><td>Ehepartner / Ehepartnerin: Hallwyl, Barbara von (?-1531) (Personen\Natürliche Personen\H)</td><td>0</td><td><input type="button" value="Hinzufügen"/></td></tr><tr><td>Kind: Hallwyl, Burkhard III. von (1535-1598) (Personen\Natürliche Personen\H)</td><td>4</td><td><input type="button" value="Hinzufügen"/></td></tr><tr><td>Eltern: Hallwyl, Sigonia von (?-?) (Personen\Natürliche Personen\H)</td><td>0</td><td><input type="button" value="Hinzufügen"/></td></tr><tr><td>Hallwyl, Dietrich von (14??-1509) (Personen\Natürliche Personen\H)</td><td>0</td><td><input type="button" value="Hinzufügen"/></td></tr></tbody></table>		Bezeichnung	Anz. Verknüpfungen	Hinzufügen	Hallwyl, Kaspar von (14??-1564) (Personen\Natürliche Personen\H)	1	<input type="button" value="Hinzufügen"/>	Literatur: HBL S. (Bibliographie\Nachschlagewerke\H)	130	<input type="button" value="Hinzufügen"/>	Katalog, Galerie Fischer (Bibliographie\Nachschlagewerke\K)	8	<input type="button" value="Hinzufügen"/>	Ehepartner / Ehepartnerin: Hallwyl, Barbara von (?-1531) (Personen\Natürliche Personen\H)	0	<input type="button" value="Hinzufügen"/>	Kind: Hallwyl, Burkhard III. von (1535-1598) (Personen\Natürliche Personen\H)	4	<input type="button" value="Hinzufügen"/>	Eltern: Hallwyl, Sigonia von (?-?) (Personen\Natürliche Personen\H)	0	<input type="button" value="Hinzufügen"/>	Hallwyl, Dietrich von (14??-1509) (Personen\Natürliche Personen\H)	0	<input type="button" value="Hinzufügen"/>
Bezeichnung	Anz. Verknüpfungen	Hinzufügen																							
Hallwyl, Kaspar von (14??-1564) (Personen\Natürliche Personen\H)	1	<input type="button" value="Hinzufügen"/>																							
Literatur: HBL S. (Bibliographie\Nachschlagewerke\H)	130	<input type="button" value="Hinzufügen"/>																							
Katalog, Galerie Fischer (Bibliographie\Nachschlagewerke\K)	8	<input type="button" value="Hinzufügen"/>																							
Ehepartner / Ehepartnerin: Hallwyl, Barbara von (?-1531) (Personen\Natürliche Personen\H)	0	<input type="button" value="Hinzufügen"/>																							
Kind: Hallwyl, Burkhard III. von (1535-1598) (Personen\Natürliche Personen\H)	4	<input type="button" value="Hinzufügen"/>																							
Eltern: Hallwyl, Sigonia von (?-?) (Personen\Natürliche Personen\H)	0	<input type="button" value="Hinzufügen"/>																							
Hallwyl, Dietrich von (14??-1509) (Personen\Natürliche Personen\H)	0	<input type="button" value="Hinzufügen"/>																							
Resultate 1 - 1 von 1																									

Anmerkung: Eigene Aufnahme der Webseite <https://katalog.burgerbib.ch/deskriptorensuche.aspx> vom 16.12.2024. Ersichtlich ist das Suchresultat einer Deskriptorensuche mit dem Suchterm «Hallwyl, Kaspar von (14??-1564)».

Tabelle 4
Schema in welches die Relationstypen abgelegt werden.

Bezeichnung	Beschreibung
desk_id	ID-Nr. des Deskriptors welcher abgefragt wurde.
role	Relationstyp (z. B. «Kind»)
entity_name	ID-Name des gefundenen Deskriptors
entity_id	ID-Nr. des gefundenen Deskriptors

3.1.2. sling.py

Das Skript sling.py dient als Wrapper zum Ausführen des Kommandozeilenprogramms «slingcli» (Larco, 2024). Das Programm migriert die CSV-Dateien mit den Daten zu den Deskriptoren in eine relationale Datenbank. Als Datenbankmanagementsystem wird «DuckDB» verwendet (Raasveldt & Muehleisen, 2024). Dieser Schritt erleichtert einerseits die Sichtung der Daten, etwa durch die Verfügbarkeit von SQL, und erleichtert die Programmierung performanter Programme in nachfolgenden Schritten (Kap. 3.1.3).

3.1.3. converter.py

Das Skript «converter.py» dient der Zusammenstellung des RDF-Datensatzes gemäss dem Datenmodell (vgl. Kap. 2.2). Dazu wird die Python-Bibliothek «RDFLib» (RDFLib Team, 2024) verwendet. Der Datensatz wird im Turtle-Dateiformat serialisiert.

3.1.4. reasoner.py

Das Programm lädt die Ontologien RiC-O und den RDF-Datensatz, bildet Inferenzen und serialisiert den expandierten Datensatz im Turtle-Dateiformat. Als reasoning engine wird die auf RDFLib aufbauende Python-Bibliothek «OWL-RL» (Herman, 2014) verwendet.

3.2. qlever

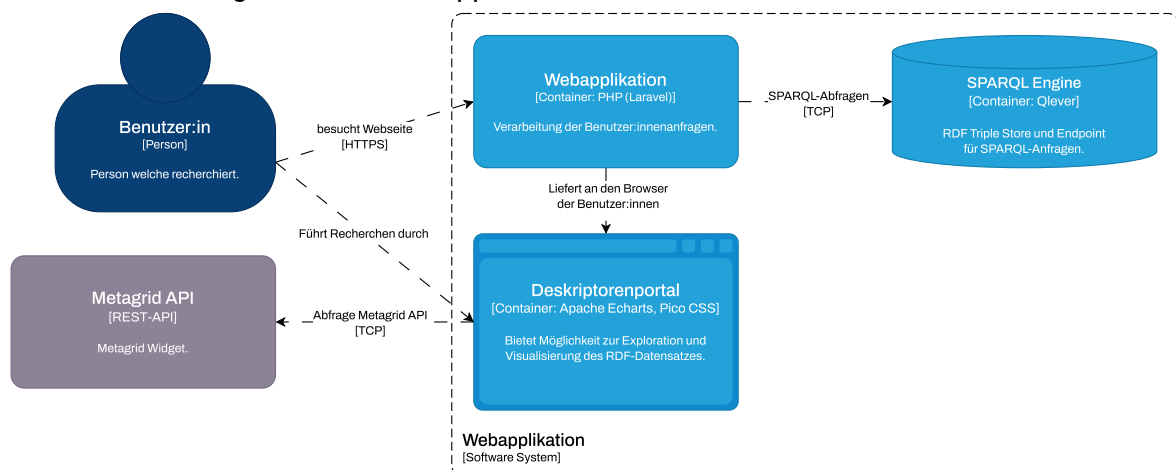
Als RDF-Triplestore / SPARQL-Engine wird «Qlever» eingesetzt (Bast et al., 2022). Die Konfiguration von Qlever wird mithilfe des Python-Skripts «qlever-control» vorgenommen (Bast et al., 2024).

3.3. Webapplikation

Die Webapplikation (Abb. 4) fungiert der Exploration und Visualisierung des RDF-Datensatzes. Das Backend ist mithilfe der PHP-Bibliothek «Laravel» umgesetzt (Otwell, 2024). Für das Frontend wird die CSS-Bibliothek «pico.css» (Larroche, 2024) und die JavaScript-Bibliothek «Apache ECharts» (Li et al., 2018) genutzt. Die nachfolgenden Kapitel beschreiben die Funktionalität der Webapplikation.

Abbildung 4

C4 Container-Diagramm der Webapplikation.



Anmerkung: Eigene Darstellung.

3.3.1. Suche

Die Suchfunktion bietet die Möglichkeit nach Deskriptoren anhand der Bezeichnung, Geburts- und Todesdatum (resp. Geburtsjahr, -monat, -tag und Todesjahr, -monat, -tag) und dem Kriterium, ob der Deskriptor über Nachkommen verfügt, zu suchen (Abb. 5). Die Suchkriterien sind mittels «UND» Verknüpfung kombiniert. D. h. die Suchtreffer entsprechen Einträgen, auf welche alle Suchkriterien zutreffen.

3.3.2. Zeitstrahl

Der Zeitstrahl ist ein Liniendiagramm, welches in aufsteigender Reihenfolge alle Deskriptoren, bei welchen das Geburtsdatum in der gewählten Zeitspanne liegt, auflistet (Abb. 6). Es werden nur Deskriptoren angezeigt, welche sowohl über ein Geburts- und Todesdatum

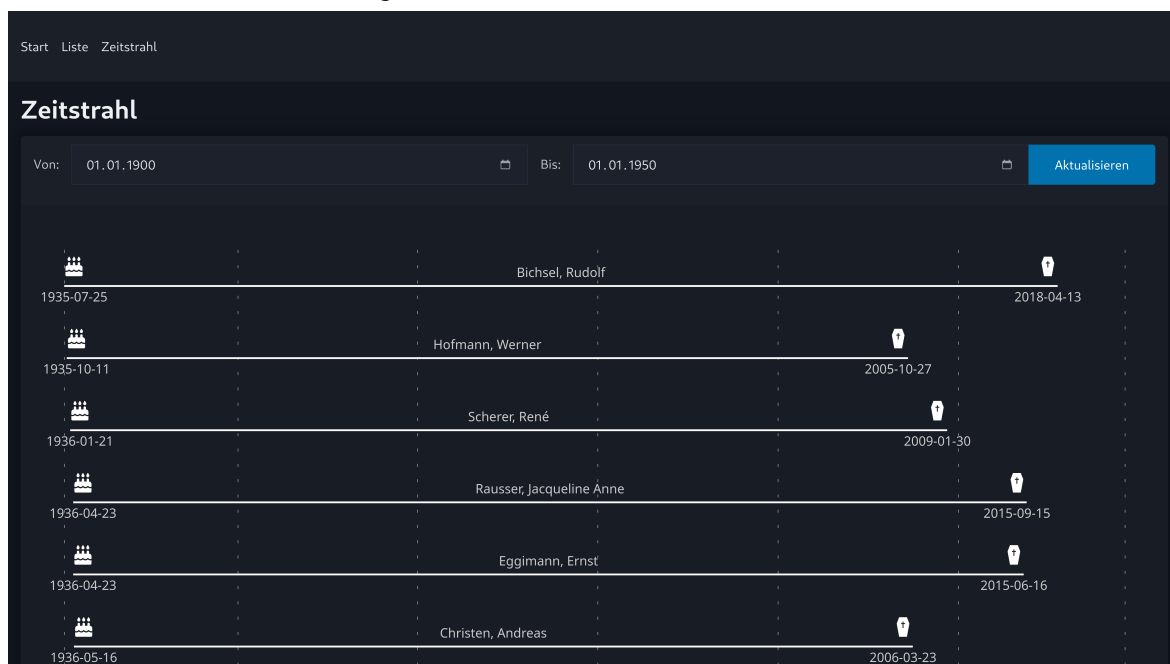
Abbildung 5
Screenshot der Suchmaske

The screenshot shows a search interface with a dark theme. At the top, there is a search bar with a magnifying glass icon and the text 'Suche', followed by a blue 'Search' button. Below the search bar, there is a section titled 'Erweiterte Suchoptionen' with a dropdown arrow. Under this section, there is a checkbox labeled 'hat Nachfahren'. Below the checkbox, there are six input fields arranged in two rows and three columns. The first row contains 'Geburtsjahr' (YYYY), 'Geburtsmonat' (mm), and 'Geburtstag' (dd). The second row contains 'Todesjahr' (YYYY), 'Todesmonat' (mm), and 'Todesstag' (dd). Each input field has a small circular icon with a downward arrow on its right side.

Anmerkung: Eigene Darstellung. Es ist möglich, mehrere oder nur einzelne Felder auszufüllen.

verfügen. Das Diagramm stellt bis zu 100 Einträge dar. Bei Suchresultaten mit mehr als 100 Einträgen setzt eine Paginierung ein und die nachfolgenden Einträge sind mittels Blättern anzeigbar. Alle Einträge im Diagramm sind mit der jeweiligen Deskriptorendetailsansicht (Kap. 3.3.3) verlinkt.

Abbildung 6
Screenshot der Visualisierung «Zeitstrahl»



Anmerkung: Eigene Darstellung. Auf dem Zeitstrahl ersichtlich sind Deskriptoren, welche zwischen 1900 und 1950 geboren sind.

3.3.3. Deskriptorendetailsansicht

Die Deskriptorendetailsansicht (Abb. 7) führt alle verfügbaren Angaben zu einem Deskriptor auf. Dies umfasst Name, biologisches Geschlecht, nächste Verwandten (Eltern, Geschwister, Kinder, Lebens- und Ehepartner:innen), Geburts-, Tauf-, Begräbnis- und Todes-

datum. Hyperlinks zu weiteren Portalen, mit Angaben zum Deskriptor, werden über die Metagrid REST-API – ein Projekt zur Vernetzung von Daten zu historischen Personen (Dodis & SAGW, n. d.) – geladen. Verfügt der Deskriptor über Nachkommen, sind diese über ein Baumstrukturdiagramm erkundbar. Die Knotenbeschriftungen des Diagramms sind mit der jeweiligen Deskriptorendetailansicht verlinkt.

Abbildung 7
Screenshot der Deskriptorendetailansicht zu Berchtold Haller

Start
Liste
Zeitstrahl

Haller, Berchtold

Deutscher Pfarrer in Moudon 1767, Pfarrer in Rüegsau 1774.

Weblinks

GND

[hallerNet](#)

[HFLS](#)

Links via Metagrid – die Vernetzungsinitiative der SAGW

Geschlecht (biologisch)

Männlich

Geburtsdatum

1728

Todesdatum

1778

Ehepartner:innen

[Haller, Johanna Salome](#)

Kinder

[Sprüngli, Salome Katharina](#)

[Haller, Karl Gabriel](#)

[Haller, Berchtold Friedrich](#)

[Bourgeois, Marianne](#)

Verzeichnungseinheiten

[Zeitstrahl](#)

Nachkommenschaft

Anmerkung: Eigene Darstellung. Im oberen Abschnitt sind Angaben und weiterführende Links zu sehen. Der untere Abschnitt zeigt die Nachkommenschaft Hallers in einem Baumstrukturdiagramm.

4. Diskussion

Herausforderungen in der Umsetzung betreffen insbesondere die Datenmodellierung (Kap. 4.1), die Inferenzbildung (Kap. 4.2) und Visualisierung von Stammbäumen (Kap. 4.3).

4.1. Datenmodellierung

RiC-O ist flexibel und für das Abbilden von Daten sind mehrere gültige Herangehensweisen möglich. Mit Matterhorn RDF steht ein Metadatenmodell für die Beschreibung digitaler Archivalien zur Verfügung (docuteam AG, n. d.). Um das Potenzial von RiC auszuschöpfen und Archive verschiedener Institutionen effizient zu vernetzen, bedarf es weiterer, auf der RiC-O aufbauender Datenmodelle. Dazu sieht die EGAD (2024) als vierten Teil des RiC-Standards die Records in Contexts-Application Guidelines vor, eine Richtlinien zur Implementation von RiC in Softwaresystemen. Zudem listet die EGAD als einer der nächsten Schritte in der Weiterentwicklung von RiC-O das Prüfen möglicher Mappings zu anderen Ontologien auf (ICA EGAD, 2024, Abs. «Next steps»).

Komplex gestaltete sich die Definition eines Tauf- und Begräbnisdatums anhand der RiC-O. Die Umsetzung nutzt dazu Instanzen der Klasse «rico:EventTypes». Dabei bleibt die Frage offen, ob das Geburts- und Todesdatum ebenfalls als Eventtypen modelliert werden sollten, um eine konsistente Modellierung der Lebensdaten zu erreichen. Die verwendeten Eigenschaften «rico:hasBirthDate» und «rico:hasDeathDate», folgen einer anderen Logik resp. kennt RiC-O keine vorgefertigten Eventtypen zu Lebensdaten. Eine Alternative besteht in der vollständigen Modellierung biografischer Angaben, mit einer darauf spezialisierten Ontologie, wie das BIO Vocabulary (Davis & Galbraith, 2011).

4.2. Reasoner

Die Auswahl an Reasoner-Engines ist überschaubar (Abicht, 2023). Die Inferenzenbildung erfolgt unter Einsatz der Software OWL-RL, welche auf dem Softwarepaket RDFLib aufbaut. Dabei erlaubt RDFLib das Bilden von Tripel mit Literalen in der Subjektposition. Während die RDF-Spezifikation dies nicht untersagt (W3C, 2014, Kap. 7), wird dies von SPARQL-Engines (bspw. Qlever oder Fuseki) zumeist nicht unterstützt. In der Konsequenz wurden die Lösungsvorschläge von al-Dosari (2023) und Car (2023) implementiert. Tripel mit Literalen in der Subjektposition werden hierbei abgefangen und anschliessend wieder vom Graph entfernt. Ein weiterer Versuch bestand in der Verwendung des Reasoners «reasonable» (Fierro & Paulson, 2022). Zwar liegt die Verarbeitungsdauer im Vergleich zu OWL-RL um ein Vielfaches tiefer, das Resultat war aber fehlerbehaftet: Teils wiesen Tripel anstatt der korrekten Daten einen beliebigen Zahlenwert auf. Zudem bestand die gleiche Problematik der Tripel mit Literalen in der Subjektposition.

Bei der Datenmodellierung wurde als einziger Verwandtschaftsgrad «rico:hasChild» definiert (Person wird mit einem Kind verbunden). Die Erwartung, dass dadurch sämtliche weitere Verwandtschaftsbeziehungen per Inferenz gebildet werden, erfüllte sich nicht. Generierte Verwandtschaftstripel beschränkten sich auf Aussagen zu Vor- oder Nachfahren

(rico:hasAncestor, rico:isSuccessorOf, rico:isChildOf). Hingegen wurden Tripel, welche Aussagen, dass Personen in einer Geschwisterbeziehung zueinander stehen (rico:hasSibling), nicht gebildet. Ein Blick in die Ontologie zeigt, dass die Property Chain der Eigenschaft, Geschwister nicht über gemeinsame Eltern definiert, sondern dadurch, dass die Personen über einen Relationsknoten miteinander verbunden sind (Programmcode 1, Zeilen 7–14). Für die Webapplikation wurden die Geschwister stattdessen über eine SPARQL-Abfrage ermittelt (Suche Elternteil einer Person und vom Elternteil alle Kinder; Programmcode 2).

Programmcode 1

Auszug der RiC-O; Definition von rico:hasSibling

```

1 <owl:ObjectProperty rdf:about="https://www.ica.org/standards/RiC/ontology#
  hasSibling">
2   <rdfs:subPropertyOf
3     rdf:resource="https://www.ica.org/standards/RiC/ontology#
      hasFamilyAssociationWith"/>
4   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#SymmetricProperty"/>
5   <rdfs:domain rdf:resource="https://www.ica.org/standards/RiC/ontology#Person
    "/>
6   <rdfs:range rdf:resource="https://www.ica.org/standards/RiC/ontology#Person"
    />
7   <owl:propertyChainAxiom rdf:parseType="Collection">
8     <rdf:Description
9       rdf:about="https://www.ica.org/standards/RiC/ontology#
        thingIsConnectedToRelation"/>
10    <rdf:Description
11      rdf:about="https://www.ica.org/standards/RiC/ontology#
        siblingRelation_role"/>
12    <rdf:Description rdf:about="https://www.ica.org/standards/RiC/ontology#
      relationConnects"
13    />
14  </owl:propertyChainAxiom>
15  ...
16 </owl:ObjectProperty>

```

Programmcode 2

SPARQL-Abfrage zur Ermittlung von Geschwister

```

1 PREFIX rico: <https://www.ica.org/standards/RiC/ontology#>
2 SELECT DISTINCT ?ark ?name {
3   <https://burgerbib.ch/indexterms/ark:36599/pv194dk27hx> rico:isChildOf ?parent
4   .
5   ?parent rico:hasChild ?child .
6   ?child rico:hasOrHadIdentifier ?id ;
7     rico:hasOrHadName ?name .
8   ?id rico:hasIdentifierType <https://burgerbib.ch/IdentifierTypes/ARK> ;
9     rico:normalizedValue ?ark .
10  FILTER (?ark != 'ark:36599/pv194dk27hx')
11 }

```

4.3. Visualisierung

Für die Visualisierung von Stammbäumen verfügt Apache Echarts keinen standardmäßigen Diagrammtyp. Software aus dem akademischen Umfeld für die Darstellung von Stammbäumen geht meist aus der Domäne der Bioinformatik hervor. Ein Beispiel dafür ist «pedigree.js», ein Visualisierungswerkzeug und Editor für die Darstellung der Familienhistorie in der Genomforschung (Carver et al., 2018). Die Gestaltung des Stammbaums verwendet dabei fachspezifische Symbole und bietet nur wenig Konfigurationsmöglichkeiten für die Darstellungsweisen. Eine kommerzielle Lösung ist die JavaScript-Bibliothek «FamilyTree JS» (BALKAN App, 2024), für welche der Erwerb einer kostenpflichtigen Lizenz nötig ist. Die Suche nach quelloffener, kostenfreier und aktiv entwickelter Software mit einem permissiven Lizenzmodell blieb ergebnislos.

5. Fazit

Die Projektarbeit behandelt die Aggregation, Aufbereitung und Visualisierung eines RDF-Datensatzes. Als Datengrundlage dient ein Auszug der Deskriptoren aus dem Thesaurus «Personen» der Burgerbibliothek Bern. Die Daten werden anhand der RiC-O modelliert und serialisiert. Über eine Webapplikation ist das Explorieren des erstellten RDF-Datensatzes möglich. Herausforderungen resp. Forschungsdesiderate zeigten sich in mehrfacher Hinsicht: Mit RiC-O steht ein sehr flexibles Datenmodell zur Beschreibung von Archiven zur Verfügung. Um das Potenzial des Standards auszuschöpfen, braucht es weitere Implementationsstandards, wie Matterhorn RDF oder Datenmappings zu anderen Ontologien wie z. B. das BIO Vocabulary. Beim Bilden von Inferenzen zeigte sich, dass Geschwisterbeziehungen nicht einzig über die Definition von der Eltern-Kind-Beziehung geschlussfolgert werden können. Eine weitere Schwierigkeit zeigte sich bei den Reasoning-Engines, welche das Bilden von Tripeln mit Literalen in der Subjektposition erlauben, was von gängigen SPARQL-Engine nicht unterstützt wird. Im Zusammenhang der Visualisierung von Stammbäumen blieb die Suche nach geeigneter erfolglos. Das Projekt ist ein Beitrag zur Implementation von Personendeskriptor anhand der RiC-O und zeigt den Vorteil eines auf Graphen basierenden Datenmodells, welches neue Ansätze in der Vermittlung von Deskriptoren erlaubt.

Literaturverzeichnis

- Abicht, K. (2023, September). OWL Reasoners still useable in 2023 [arXiv:2309.06888 [cs]]. <https://doi.org/10.48550/arXiv.2309.06888>
- al-Dosari, M. (2023, März). OWL RL creates triples with a Literal as Subject when evaluating PROV-O ontology #50. Zugriff am 28. Dezember 2024 unter <https://github.com/RDFLib/OWL-RL/issues/50#issuecomment-1476897684>
- BALKAN App. (2024, Oktober). FamilyTreeJS. Zugriff am 6. Januar 2025 unter <https://balkan.app/FamilyTreeJS>
- Bast, H., Kalmbach, J., Klumpp, T., Kramer, F., & Schnelle, N. (2022). Efficient and Effective SPARQL Autocompletion on Very Large Knowledge Graphs. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2893–2902. <https://doi.org/10.1145/3511808.3557093>
- Bast, H., Kalmbach, J., Klumpp, T., Kramer, F., & Schnelle, N. (2024). qlever-control. Zugriff am 17. November 2024 unter <https://github.com/ad-freiburg/qlever-control>
- Car, N. (2023, August). Unexpected behaviour of OWL-RL reasoner (Literals in the position of Subjects) #63. Zugriff am 28. Dezember 2024 unter <https://github.com/RDFLib/OWL-RL/issues/63#issuecomment-2283074918>
- Carver, T., Cunningham, A. P., Babb de Villiers, C., Lee, A., Hartley, S., Tischkowitz, M., Walter, F. M., Easton, D. F., & Antoniou, A. C. (2018). pedigreejs: a web-based graphical pedigree editor. *Bioinformatics*, 34(6), 1069–1071. <https://doi.org/10.1093/bioinformatics/btx705>
- Davis, I., & Galbraith, D. (2011, Juni). BIO: A vocabulary for biographical information. Zugriff am 17. Dezember 2024 unter <http://purl.org/vocab/bio/0.1/>
- De Coulon, B. (2024). Déploiement de la norme Records in Contexts pour la gestion des collections de la Fondation SAPA. *Revue électronique suisse de science de l'information (RESSI)*, (24). <https://doi.org/10.55790/journals/ressi.2024.e1511>
- docuteam AG. (n. d.). Matterhorn RDF. Zugriff am 23. Dezember 2024 unter <https://docs.docuteam.ch/introduction/matterhornRDF>
- Dodis & SAGW. (n. d.). Metagrid. Zugriff am 30. Dezember 2024 unter <https://metagrid.ch>
- EGAD. (2024, Juni). Records in Contexts (RiC). Zugriff am 17. Dezember 2024 unter <https://www.ica.org/ica-network/expert-groups/egad/records-in-contexts-ric/>
- Fierro, G., & Paulson, E. (2022, November). reasonable [original-date: 2019-10-31T18:06:06Z]. Zugriff am 6. Januar 2025 unter <https://github.com/gtfierro/reasonable>
- Glarner, N. (2024). Deskriptoren – Erschliessungs- und Vermittlungspraxis in der Burgerbibliothek Bern [Number: 1]. *Informationswissenschaft: Theorie, Methode und Praxis*, 8(1), S. 535–553. <https://doi.org/10.18755/iw.2024.25>
- Herman, I. (2014, Oktober). OWL-RL: OWL-RL: A simple OWL2 RL reasoner on top of RDFLib. <https://doi.org/10.5281/zenodo.14543>

- ICA EGAD. (2024, September). International Council on Archives Records in Contexts Ontology (ICA RiC-O). Zugriff am 2. November 2024 unter https://www.ica.org/standards/RiC/RiC-O_1-0-2.html
- Larco, F. (2024, Dezember). Sling [original-date: 2020-10-15T10:06:58Z]. Zugriff am 17. Dezember 2024 unter <https://github.com/slingdata-io/sling-cli>
- Larroche, L. (2024, März). pico [original-date: 2019-11-15T09:42:54Z]. Zugriff am 23. Dezember 2024 unter <https://github.com/picocss/pico>
- Li, D., Mei, H., Shen, Y., Su, S., Zhang, W., Wang, J., Zu, M., & Chen, W. (2018). ECharts: A declarative framework for rapid construction of web-based visualization. *Visual Informatics*, 2(2), 136–146. <https://doi.org/https://doi.org/10.1016/j.visinf.2018.04.011>
- Otwell, T. (2024). Laravel. Zugriff am 17. November 2024 unter <https://github.com/laravel/laravel>
- Raasveldt, M., & Muehleisen, H. (2024, Dezember). DuckDB. Zugriff am 27. Dezember 2024 unter <https://github.com/duckdb/duckdb>
- RDFLib Team. (2024). RDFLib. Zugriff am 17. November 2024 unter <https://github.com/RDFLib/rdfliib>
- Richardson, L. (2024, Januar). Beautiful Soup: We called him Tortoise because he taught us. Zugriff am 17. Dezember 2024 unter <https://www.crummy.com/software/BeautifulSoup/>
- Spec, J. (2023). From ISAD(G) to Records in Contexts: A new era [Backup Publisher: Fachhochschule Graubünden. Schweizerisches Institut für Informationswissenschaft Issue: 159 Place: Chur Series: Churer Schriften zur Informationswissenschaft. Arbeitsbereich Informationswissenschaft]. https://www.fhgr.ch/fileadmin/fhgr/angewandte_zukunftstechnologien/SII/churer_schriften/sii-churer_schriften_159_spec.pdf
- W3C. (2014, Februar). RDF 1.1 Concepts and Abstract Syntax. Zugriff am 28. Dezember 2024 unter <https://www.w3.org/TR/rdf11-concepts/#section-generalized-rdf>

A. Datenmapping

Tabelle 5
Übersicht Datenmapping

AIS-Export	Subjekt	Prädikat	Objekt
ARK	<https://burgerbib.ch/identifiers/"ARK">	rico:hasIdentifierTyp	<https://burgerbib.ch/identifierTypes/ARK
		rico:isOrWasIdentifierOf	<https://burgerbib.ch/indexterms/"ARK">
		rico:normalizedValue	"ARK"
ID-Nr.	<https://burgerbib.ch/identifiers/scope:"ID-Nr.">	rico:hasIdentifierTyp	<https://burgerbib.ch/identifierTypes/Scope-ID
		rico:isOrWasIdentifierOf	<https://burgerbib.ch/indexterms/"ARK">
		rico:normalizedValue	"ID-Nr."
ID-Name	<https://burgerbib.ch/indexterms/"ARK">	rico:hasOrHadName	"ID-Name"
Beschreibung	<https://burgerbib.ch/indexterms/"ARK">	rico:generalDescription	"Beschreibung"
Geschlecht (weiblich)	<https://burgerbib.ch/indexterms/"ARK">	rico:hasOrHadDemographicGroup	<https://burgerbib.ch/demographicgroups/SexFemale>
	<https://burgerbib.ch/demographicgroups/SexFemale>	rdfs:label	"weiblich"
Geschlecht (männlich)	<https://burgerbib.ch/indexterms/"ARK">	rico:hasOrHadDemographicGroup	<https://burgerbib.ch/demographicgroups/SexMale>
	<https://burgerbib.ch/demographicgroups/SexMale>	rdfs:label	"männlich"
Geburtsdatum	<https://burgerbib.ch/indexterms/"ARK">	rico:hasBirthDate	"Geburtsdatum"
Todesdatum	<https://burgerbib.ch/indexterms/"ARK">	rico:hasDeathDate	"Todesdatum"
Taufdatum	<https://burgerbib.ch/indexterms/"ARK">	rico:isAssociatedWithEvent	_:e1
	_:e1	rico:hasEventType	<https://burgerbib.ch/EventTypes/Baptism>
	_:e1	rico:occuredAtDate	"Taufdatum"
	<https://burgerbib.ch/indexterms/"ARK">	rico:isAssociatedWithEvent	_:e2
	_:e2	rico:hasEventType	<https://burgerbib.ch/EventTypes/Burial>
	_:e2	rico:occuredAtDate	"Begräbnisdatum"
Ehepartner / Ehepartnerin	<https://burgerbib.ch/indexterms/"ARK">	rico:hasOrHadSpouse	<https://burgerbib.ch/indexterms/"ARK">
Lebenspartner / Lebenspartnerin	<https://burgerbib.ch/indexterms/"ARK">	<https://burgerbib.ch/relations/hasOrHadLifePartner>	<https://burgerbib.ch/indexterms/"ARK">
Kind	<https://burgerbib.ch/indexterms/"ARK">	rico:hasChild	<https://burgerbib.ch/indexterms/"ARK">

Programmcode 3

Implementation des Identifiers Scope-ID

```
1 @prefix rico: <https://www.ica.org/standards/RiC/ontology#> .
2 @prefix wd: <http://www.wikidata.org/wiki/> .
3 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
4
5 <https://burgerbib.ch/IdentifierTypes/Scope-ID> a rico:IdentifierType ;
6     rdfs:label "ID from the ScopeArchiv AIS." .
7
8 <https://burgerbib.ch/identifiers/scope:*> rico:hasIdentifierType <https://
    burgerbib.ch/IdentifierTypes/Scope-ID> ;
9     rico:isOrWasIdentifierOf <https://burgerbib.ch/indexterms/ark:36599/*> ;
10    rico:normalizedValue "*"^^xsd:nonNegativeInteger .
```

Programmcode 4

Beispiel Identifier ARK

```
1 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
2 @prefix rico: <https://www.ica.org/standards/RiC/ontology#> .
3 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
4
5 <https://burgerbib.ch/IdentifierTypes/ARK> a rico:IdentifierType ;
6     rico:closeTo wd:Q2860403 .
7
8 <https://burgerbib.ch/identifiers/ark:36599/zzzkz09zv22> rico:hasIdentifierType
    <https://burgerbib.ch/IdentifierTypes/ARK> ;
9     rico:isOrWasIdentifierOf <https://burgerbib.ch/indexterms/ark:36599/
    zzzkz09zv22> ;
10    rico:normalizedValue "ark:36599/zzzkz09zv22"^^xsd:string .
```

Programmcode 5

Beispiel für den Eventtyp «Baptism»

```
1 @prefix bio: <http://purl.org/vocab/bio/0.1/> .
2 @prefix rico: <https://www.ica.org/standards/RiC/ontology#> .
3
4 <https://burgerbib.ch/EventTypes/Baptism> a rico:EventType ;
5     rico:closeTo bio:Baptism .
```

Programmcode 6

Beispiel für den Eventtyp «Burial»

```
1 @prefix bio: <http://purl.org/vocab/bio/0.1/> .
2 @prefix rico: <https://www.ica.org/standards/RiC/ontology#> .
3
4 <https://burgerbib.ch/EventTypes/Burial> a rico:EventType ;
5     rico:closeTo bio:Burial .
```

Programmcode 7

Beispiel demografische Gruppe «SexFemale»

```
1 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
2 @prefix rico: <https://www.ica.org/standards/RiC/ontology#> .
3
4 <https://burgerbib.ch/DemographicGroups/SexFemale> rdfs:label "Female (
    biological sex)" ;
5     rdfs:subClassOf rico:DemographicGroup .
```

Programmcode 8

Beispiel demografische Gruppe «SexMale»

```
1 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
2 @prefix rico: <https://www.ica.org/standards/RiC/ontology#> .
3
4 <https://burgerbib.ch/DemographicGroups/SexMale> rdfs:label "Male (biological
   sex)" ;
5   rdfs:subClassOf rico:DemographicGroup .
```

Programmcode 9

Beispiel Lebenspartner «hasOrHadLifePartner»

```
1 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
2 @prefix rico: <https://www.ica.org/standards/RiC/ontology#> .
3
4 <https://burgerbib.ch/relations/hasOrHadLifePartner> rdfs:label "Connects two
   Persons who are or were in a romantic relationship, as if they are spouses
   but without being legally married." ;
5   rdfs:subPropertyOf rico:hasFamilyAssociationWith .
```

Programmcode 10

Beispiel Person

```
1 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
2 @prefix rico: <https://www.ica.org/standards/RiC/ontology#> .
3 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
4
5 <https://burgerbib.ch/indexterms/ark:36599/7xkk749wklb> a rico:Person ;
6   rdfs:label "Kern, Johann Konrad (1808-1888)"^^xsd:string ;
7   rico:generalDescription "Jurist, Politiker, Diplomat. Thurgauer Grossrat,
   Regierungsrat, Präsident des Obergerichts. National- und Ständerat,
   Präsident des Bundesgerichtes, des eidg. Schulrates und des
   Nationalrates. Gesandter und bevollmächtigter Minister der Schweiz in
   Paris."^^xsd:string ;
8   rico:isAssociatedWithEvent [ rico:hasEventType <https://burgerbib.ch/
   EventTypes/Baptism> ;
9     rico:occurredAtDate "1669-07-08"^^xsd:date ] ;
10  rico:hasBirthDate "1808-06-11"^^xsd:date ;
11  rico:hasDeathDate "1888-04-14"^^xsd:date ;
12  rico:isAssociatedWithEvent [ rico:hasEventType <https://burgerbib.ch/
   EventTypes/Burial> ;
13    rico:occurredAtDate "1669-07-08"^^xsd:date ] .
14  rico:hasOrHadDemographicGroup <https://burgerbib.ch/DemographicGroups/
   SexMale> ;
15  rico:hasOrHadName "Kern, Johann Konrad"^^xsd:string ;
16  rico:hasOrHadSpouse <person> ;
17  rico:hasOrHadLifePartner <person> .
```