Fabian Steeg und Adrian Pohl

Ein Protokoll für den Datenabgleich im Web am Beispiel von OpenRefine und der Gemeinsamen Normdatei (GND)

1 Einordnung

1.1 Normdaten und die Qualität der Inhaltserschließung

Normdaten spielen speziell im Hinblick auf die Qualität der Inhaltserschließung bibliografischer und archivalischer Ressourcen eine wichtige Rolle. Ein konkretes Ziel der Inhaltserschließung ist z. B., dass alle Werke über Hermann Hesse einheitlich zu finden sind. Hier bieten Normdaten eine Lösung, indem z. B. bei der Erschließung einheitlich die GND-Nummer 11855042X für Hermann Hesse verwendet wird. Das Ergebnis ist eine höhere Qualität der Inhaltserschließung vor allem im Sinne von Einheitlichkeit und Eindeutigkeit und, daraus resultierend, eine bessere Auffindbarkeit.

Werden solche Entitäten miteinander verknüpft, z. B. Hermann Hesse mit einem seiner Werke, entsteht ein *Knowledge Graph*, wie ihn etwa Google bei der Inhaltserschließung des Web verwendet (Singhal 2012). Die Entwicklung des *Google Knowledge Graph* und das hier vorgestellte Protokoll sind historisch miteinander verbunden: OpenRefine wurde ursprünglich als Google Refine entwickelt, und die Funktionalität zum Abgleich mit externen Datenquellen (Reconciliation) wurde ursprünglich zur Einbindung von Freebase entwickelt, einer der Datenquellen des *Google Knowledge Graph*. Freebase wurde später in Wikidata integriert. Schon Google Refine wurde zum Abgleich mit Normdaten verwendet, etwa den Library of Congress Subject Headings (Hooland et al. 2013).

1.2 Reconciliation als Teil der Inhaltserschließung

Bei der Verwendung der GND zur Inhaltserschließung findet bei den Erfassenden ein Abgleich zwischen den vorliegenden Daten (z. B. der Zeichenkette *Hermann Hesse*) und den Normdaten statt. Über ein System zum Abfragen der GND wird etwa der Eintrag zum Schriftsteller und Nobelpreisträger (11855042X), zum russischen Staatsrat und Arzt (137565259), oder zur Hesse-Biografie von Hugo Ball (4592695-5) zur Verknüpfung ausgewählt. In diesem Sinn ist hier der

https://doi.org/10.1515/9783110691597-013

Begriff des Datenabgleichs bzw. der Reconciliation zu verstehen: als Abgleich von Namen (einer Person, eines Ortes, eines Schlagworts etc.) mit jeweils einem Identifikator innerhalb einer Normdatei.

Der Prozess der Reconciliation selbst kann so als Form oder Teil der Inhaltserschließung gesehen werden. Die Werkzeugunterstützung, Automatisierung und Standardisierung dieses manuellen Schrittes ist Gegenstand dieses Beitrags. Dabei wird dargestellt, wie Prozesse des manuellen Abgleichs (z. B. die Disambiguierung von Hermann Hesse als Schriftsteller) beim automatischen Verfahren formal kodiert werden, etwa durch strukturierte Zusatzinformationen zu Beruf und Lebensdaten.

So erweitert sich der beschriebene Anwendungsbereich über die manuelle Erschließung (gedruckter) bibliografischer und archivalischer Ressourcen in den Bereich der halb- oder vollautomatischen Erschließung mittels Batch-/Stapelverarbeitung, etwa von elektronischen Ressourcen, digitalen Editionen oder sonstigen Forschungsdaten. Ziel des in diesem Beitrag beschriebenen Protokolls ist in diesem Sinn also auch, Normdaten als zentrales Element der Inhaltserschließung für neue Anwendungsfälle zugänglich zu machen.

1.3 Terminologie und Abgrenzung

Die beschriebenen Konzepte sollen im Folgenden kurz von verwandten Begriffen und Verfahren abgegrenzt werden. Es handelt sich bei der Reconciliation im engeren Sinn wie oben beschrieben um das Identifizieren eindeutiger Entitäten in einer Wissensbasis (Knowledge Base) durch die Ermittlung von Identifikatoren für vorliegende Eigennamen. Die identifizierten Entitäten selbst haben Attribute, die sie zum Teil wiederum mit weiteren Entitäten verbinden (wenn diese Attribute als Werte Identifikatoren enthalten, in der GND z. B. die Verbindung von Hermann Hesse mit seinem Geburtsort *Calw*). In diesem Sinn handelt es sich bei der Reconciliation um die Verortung, und damit Disambiguierung, von Daten in einem Knowledge Graph.

Im Kontext von maschineller Sprachverarbeitung und Information Retrieval spricht man innerhalb des weiten Feldes der Informationsextraktion von der Eigennamenerkennung (Named Entity Recognition). Für den über die bloße Erkennung eines Eigennamens (z. B. *Hermann Hesse* ist ein Name) hinausgehenden Fall der Verknüpfung mit Normdaten (z. B. *Hermann Hesse* ist 11855042X) hat sich der Begriff *(Named) Entity Linking* etabliert. Durch die eindeutige Verknüpfung mit einer Entität handelt es sich bei dieser zugleich um eine Form von Wortsinndisambiguierung (Word Sense Disambiguation), hier z. B. von *Hermann Hesse* als Schriftsteller und Nobelpreisträger (11855042X) gegenüber

anderen Bedeutungen wie dem russischen Staatsrat und Arzt (137565259) oder dem Werk von Hugo Ball (4592695-5). Daher wird hier auch von *Named Entity Disambiguation* (Slawski 2015) oder *Named Entity Normalization* (Khalid, Jijkoun und Rijke 2008) gesprochen. Hier schließt sich der Kreis zur bibliothekarischen Terminologie: Das im Folgenden beschriebene Protokoll dient zur Normalisierung von Entitäten mittels Normdaten.

1.4 Protokolle und Standards

Sowohl in klassischen Bibliothekssystemen als auch im Web spielen standardisierte Datenformate und Protokolle eine zentrale Rolle. Formate wie MAB oder MARC und Protokolle wie Z39.50 ermöglichen einen institutions- und systemübergreifenden Datenaustausch und damit die Nachnutzung und Zusammenführung, etwa in Verbundkatalogen. Formate wie HTML oder JSON und Protokolle wie HTTP ermöglichen den weltweiten Datenaustausch im Web. So ist es erstrebenswert, den hier beschriebenen Datenabgleich über ein standardisiertes Protokoll durchzuführen bzw. ein solches zu entwickeln (Delpeuch et al. 2020), um institutionsübergreifend einheitlich auf zentrale Normdaten zuzugreifen. In diesem Sinn werden im nächsten Abschnitt die Details des Protokolls für den Datenabgleich im Web am Beispiel von OpenRefine und der GND dargestellt. Der verwendete Reconciliation-Dienst von lobid-gnd (Steeg, Pohl und Christoph 2019) basiert auf den als RDF publizierten GND-Daten der Deutschen Nationalbibliothek (Hauser 2014).

2 Protokoll

Der folgende Abschnitt beschreibt die einzelnen Elemente des Protokolls für den Datenabgleich im Web und ihre Verwendung am Beispiel von OpenRefine und der GND. Das Protokoll hat seinen Ursprung in der Implementierung der Netzwerkkommunikation in OpenRefine. Ausgehend davon soll es im Rahmen des W3C standardisiert werden (Delpeuch et al. 2020). Dies hat gegenüber dem umgekehrten Ansatz (zuerst wird ein Protokoll standardisiert, dann wird es implementiert) den Vorteil, dass das Protokoll erwiesenermaßen praxistauglich ist. Es entspricht in diesem Aspekt auch der in der Internet Engineering Task Force (IETF) entstandenen pragmatischen Grundhaltung der Internet-Standardisierung (Alvestrand und Lie 2009). Die folgende Beschreibung des Protokolls

kann anhand der Implementierung in lobid-gnd (Steeg, Pohl und Christoph 2019) praktisch nachvollzogen werden.[1]

2.1 JSON

Das im Folgenden beschriebene Protokoll basiert auf JSON, dem bereits ab 2005 sehr populären und spätestens nach seiner Standardisierung 2013 weit etablierten Format für Web-basierten Datenaustausch (Target 2017). Grundelement von JSON ist eine Attribut-Wert Zuordnung, z. B.:

```
{
  "Attribut_1": "Wert_1",
  "Attribut_2": "Wert_2"
}
```

Die Metadaten, Anfragen und Antworten dieses Protokolls werden mit JSON formuliert.

2.2 Service

Ein *reconciliation service*[2] beschreibt sich selbst in einem *service manifest*. Dieses JSON-Dokument definiert mindestens einen Namen in `name`, ein Präfix zur Identifikation der gelieferten Entitäten (z. B. zur Identifikation einer GND-Nummer wie `118624822` als `https://d-nb.info/gnd/118624822`) in `identifierSpace` und den Typ der gelieferten Entitäten (und damit die Ontologie samt verfügbaren Properties für die Entitäten) in `schemaSpace`:

```
{
  "name": "GND reconciliation for OpenRefine",
  "identifierSpace": "https://d-nb.info/gnd/",
  "schemaSpace": "https://d-nb.info/standards/elementset/gnd#Authori-
tyResource"
}
```

1 Eine Dokumentation der API von lobid-gnd findet sich unter https://lobid.org/gnd/api (4.12.2020).

2 Die Beschreibung des Protokolls verwendet die englischen Begriffe (kursiv gesetzt) aus dem Spezifikationsentwurf (Delpeuch et al. 2020).

2.3 Reconciliation queries

Ein Dienst mit einem solchen *service manifest* kann in einem *reconciliation client* eingebunden werden und steht dann für *reconciliation queries* zur Verfügung (s. Abb. 1). Das *service manifest* kann weitere, optionale Hilfsdienste und Unterstützung für *data extension* deklarieren (s. unten).

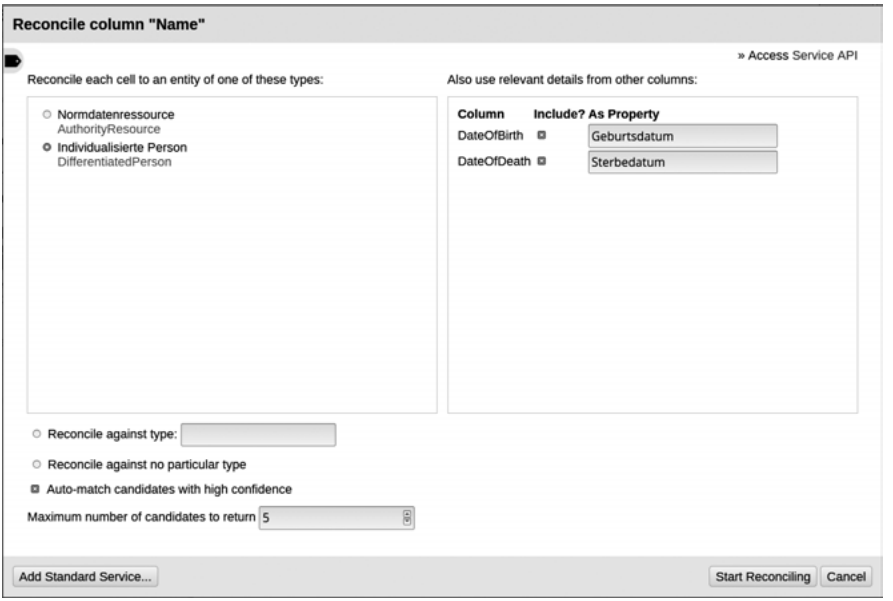

Abb. 1: Der Reconciliation-Dialog mit zahlreichen Konfigurationsmöglichkeiten in OpenRefine

2.3.1 Einfache Anfragen

In der einfachsten Form werden nur Namen an den Dienst geschickt. Dies erfolgt für alle abzugleichenden Werte in einer einzigen Anfrage. Auf Ebene der OpenRefine-Oberfläche bedeutet dies die Auswahl der entsprechenden Spalte (z. B. *Name* in Abb. 1).

Auf Ebene des Protokolls handelt es sich um ein JSON-Objekt, bei dem jeder Wert durch einen eindeutigen Schlüssel (hier `q1` und `q2`) identifiziert wird. Der Wert eines Attributs muss selbst keine Zeichenkette sein (wie `"Wert_1"` im ersten JSON-Beispiel oben), sondern kann selbst wieder JSON sein, z. B.

`{ "query": "Hans-Eberhard Urbaniak" }`. So kann mit JSON eine geschachtelte Struktur ausgedrückt werden.

Für eine Spalte bzw. Anfrage mit zwei Werten stellt sich eine minimale Anfrage dann so dar:

```
{
  "q1": { "query": "Hans-Eberhard Urbaniak" },
  "q2": { "query": "Ernst Schwanhold" }
}
```

Eine solche minimale *reconciliation query* lässt sich etwa mit folgender URL im Browser durchführen:

```
https://lobid.org/gnd/reconcile/?queries={"q1":{"query":"Twain, Mark"}}
```

Die im Browser ausgelieferte Antwort ist ein JSON-Dokument. Zur komfortableren Anzeige im Browser, etwa mit Syntax-Coloring und einklappbaren Unterabschnitten, existieren diverse JSON-Browser-Plugins. Auf der Kommandozeile können die Daten etwa mit dem vielseitigen Werkzeug `jq` verarbeitet werden:[3]

```
curl --data 'apo;queries={"q1":{"query":"Twain, Mark"}}' \
https://lobid.org/gnd/reconcile/ | jq
```

Die Antwort besteht aus einer Reihe von Vorschlägen (in JSON als Array innerhalb von `[` und `]` ausgedrückt) für jedes Element der Anfrage (hier gekürzt: nur `q1` und die ersten zwei Vorschläge):

```
{
  "q1": {
    "result": [
      {
        "id": "118624822",
        "name": "Twain, Mark",
        "score": 84.15378,
        "match": true,
        "type": [{"id": "DifferentiatedPerson", "name": "Individualisierte
        Person"}]
      },
```

3 Für Details und weitere Beispiele siehe https://shapeshed.com/jq-json/ (4.12.2020).

```
    {
      "id": "1045623490",
      "name": "Bezirkszentralbibliothek Mark Twain. Schreibwerkstatt",
      "score": 78.29902,
      "match": false,
      "type": [{"id": "CorporateBody", "name": "Körperschaft"}]
    }
   ]
  }
}
```

Als erster Vorschlag erscheint hier also Mark Twain selbst (`118624822`, Typ `Individualisierte Person`), als zweiter Vorschlag eine `Körperschaft` mit dem Identifikator `1045623490`. Die Eindeutigkeit der Identifikatoren ergibt sich durch den im *service manifest* angegebenen `identifierSpace` (s. Abschnitt *Service*). Der gemeinsame Namensraum `https://d-nb.info/gnd/` für GND-Nummern ermöglicht so die Interoperabilität verschiedener Dienste auf Basis der GND.

Neben Identifikator und Typ enthalten die Vorschläge die Felder `name`, `score` und `match` als Details zum jeweiligen Vorschlag. `Score` ist ein Maß des Dienstes für die Übereinstimmung des Vorschlages mit der Anfrage (d. h. je höher der `score`, desto besser der Vorschlag) und `match` drückt per Wahrheitswert aus, ob der Vorschlag nach internen Kriterien des Dienstes als Treffer zu dem Vorschlag bewertet wird.

Diese Vorschläge können den Nutzenden im *reconciliation client* angezeigt werden (s. Abb. 2, pro Name sehen wir die entsprechenden Vorschläge, bzw. den Namen in Fett bei der automatisch als Treffer gewerteten Entität).

2.3.2 Weitere Metadaten

Dadurch, dass mit jedem Element der Anfrage (z. B. `q1` oben) wieder JSON assoziiert ist, können jedem Element der Anfrage zusätzlich zum Namen weitere Metadaten hinzugefügt werden, etwa der gesuchte Entitätstyp in `type` oder eine Begrenzung der vom Dienst gelieferten Vorschläge in `limit`:

```
{
 "q0": {
   "query": "Christel Hanewinckel",
   "type": "DifferentiatedPerson",
   "limit": 5
```

```
 },
 "q1": {
  "query": "Franz Thünnes",
  "type": "DifferentiatedPerson",
  "limit": 5
 }
}
```

Diese können in einem *reconciliation client* bei der Nutzung konfiguriert werden (s. Abb. 1, Typauswahl oben links: *Reconcile each cell to an entity of one of these types*, Beschränkung unten links: *Maximum number of candidates to return*).

Für eine höhere Transparenz der oben beschriebenen Bewertung in der Antwort (`score`, `match`) kann ein *reconciliation service* für jedes `result` auch spezifische `features` zurückgegeben, die eine differenziertere Bewertung der Vorschläge durch den *reconciliation client* ermöglichen, z. B. eine separate Bewertung der Übereinstimmung des Namens (in `name_tfidf`) bzw. des angeforderten Typs (in `type_match`):

```
"features": [
 {
  "id": "name_tfidf",
  "value": 334.188
 },
 {
  "id": "type_match",
  "value": 13.78
 }
]
```

Auf dieser Basis könnte ein *reconciliation client* neben der Auswertung des `match`-Wertes (s. Abb. 1, links unten: *Auto-match candidates with high confidence*) selbst entscheiden, ob etwa eine Übereinstimmung beim angeforderten Typ (d. h. ein hoher Wert für `type_match`) eine geringere Übereinstimmung des Namens (d. h. einen niedrigen Wert für `name_tfidf`) ausgleicht und doch als Treffer zu werten ist.

2.3.3 Zusätzliche Daten

Neben dem abzugleichenden Namen und den oben beschriebenen Metadaten können weitere Daten mitgeschickt werden, um die Qualität des Abgleichs zu erhöhen, d. h. um mit höherer Wahrscheinlichkeit den korrekten Identifikator vom Dienst angeboten zu bekommen. Dies können bei Personen etwa Lebensdaten oder Berufe sein. Auf Ebene der OpenRefine-Oberfläche sind diese weiteren Daten zusätzliche Spalten der Tabelle (z. B. Spalten Beruf, Geburtsjahr, Sterbejahr; s. Abb. 1, oben rechts: *Also use relevant details from other columns*). Auf Ebene des Protokolls werden diese als `properties` abgebildet:[4]

```
"properties": [
 {
  "pid": "professionOrOccupation",
  "v": "Politik*"
 },
 {
  "pid": "affiliation",
  "v": "http://d-nb.info/gnd/2022139-3"
 }
]
```

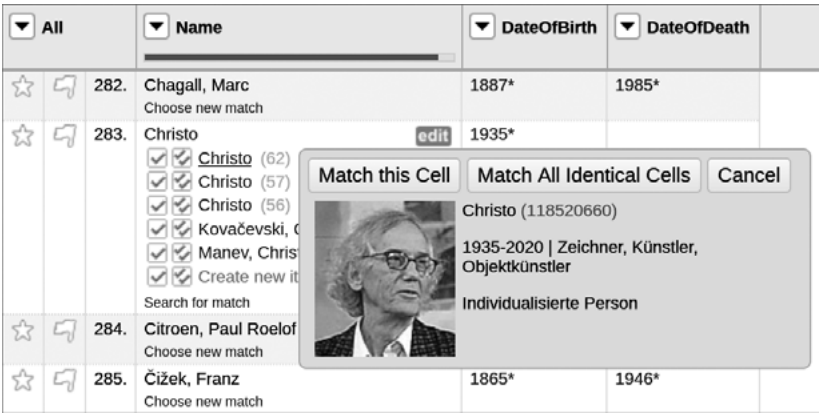

Abb. 2: Ergebnisse der Reconciliation mit Vorschlägen und Vorschau zur Auswahl von Kandidaten

4 Details zu diesem Beispiel finden sich unter https://blog.lobid.org/2019/09/30/openrefine-examples.html#occupations-and-affiliations (4.12.2020).

2.4 Hilfsdienste

Neben der zentralen Funktionalität der *reconciliation queries* kann ein *reconciliation service* weitere Dienste anbieten. Dies sind zum einen Hilfsdienste, die die Kernfunktionalität erweitern, insbesondere in Form von Vorschauen und Vorschlägen, sowie zum anderen Dienste zur Verwendung der abgeglichenen Entitäten zur Datenanreicherung der lokalen Datensätze (*data extension*).

2.4.1 Anzeige

Im Wesentlichen gibt es zwei Dienste zur Anzeige von Entitäten. Zum einen kann der *reconciliation service* in seinem *service manifest* deklarieren, wo Entitäten auf Basis eines Identifikators angezeigt werden können. Dies erfolgt über einen Eintrag `view` mit einer URL, die einen Platzhalter für den Identifikator enthält, z. B.:

```
"view": {
  "url": "https://lobid.org/gnd/{{id}}"
}
```

Ein *reconciliation client* kann damit einen Link zu einer Entität erzeugen, indem in `https://lobid.org/gnd/{{id}}` die Zeichenkette `{{id}}` durch den eigentlichen Identifikator ersetzt wird, um z. B. oben den ersten Vorschlag zu Mark Twain mit einem Link zu `https://lobid.org/gnd/118624822` zu hinterlegen (s. Abb. 2, Vorschläge sind mit Links hinterlegt).

Für eine engere Integration in einen *reconciliation client* gibt es darüber hinaus die Möglichkeit, eine Vorschau zu liefern. Der *reconciliation service* definiert dazu in vergleichbarer Form einen Service in seinem *service manifest*:

```
"preview": {
  "height": 100,
  "width": 320,
  "url": "https://lobid.org/gnd/{{id}}.preview"
}
```

Im Unterschied zum `view` wird hier eine Größe definiert, so dass der Client einen entsprechend großen Vorschaubereich erzeugen kann. Hier liefert der Dienst wie bei `view` HTML, das direkt angezeigt werden kann, allerdings muss dafür bei `preview` keine neue Seite verlinkt werden, sondern die Vorschau kann

etwa in einem Popup angezeigt werden (s. Abb. 2, Vorschau für Kandidaten mit Bild, Lebensdaten, Beruf und Typ).

2.4.2 Vorschläge

Die `suggest`-Hilfsdienste dienen zur Anzeige von Vorschlägen an verschiedenen Stellen des Datenabgleichs in einem *reconciliation client*. Vorgeschlagen werden können Entitäten, Properties und Typen. Dazu wird jeweils im *service manifest* der eigentliche Vorschlagsdienst, sowie analog zum `preview` oben, ein sogenannter Flyout-Dienst für kleine, integrierte Darstellungen deklariert:

```
"suggest": {
   "property": {
     "service_url": "https://lobid.org/gnd/reconcile",
     "service_path": "/suggest/property",
     "flyout_service_path": "/flyout/property?id=${id}"
   },
   "entity": {
     "service_url": "https://lobid.org/gnd/reconcile",
     "service_path": "/suggest/entity",
     "flyout_service_path": "/flyout/entity?id=${id}"
   },
   "type": {
     "service_url": "https://lobid.org/gnd/reconcile",
     "service_path": "/suggest/type",
     "flyout_service_path": "/flyout/type?id=${id}"
   }
}
```

Alle Vorschlagsdienste erwarten einen Query-Parameter `prefix`, in dem die bisher von den Nutzenden eingegebene Zeichenkette übergeben wird. Dies dient im Client etwa dazu, vorzuschlagen, mit welcher GND-Property mitgeschickte Daten assoziiert werden (s. oben, Zusätzliche Daten). Wird im Client etwa an der entsprechenden Stelle `beruf` eingegeben, wird intern folgende Anfrage an den `property`-Hilfsdienst gesendet:

```
https://lobid.org/gnd/reconcile/suggest/property?prefix=beruf
```

Die Antwort zu dieser Anfrage lautet (kann wie oben im Browser oder über das Kommandozeilenwerkzeug `curl` nachvollzogen werden):


```
{
 "code": "/api/status/ok",
 "status": "200 OK",
 "prefix": "beruf",
 "result": [
  {
   "id": "professionOrOccupation",
   "name": "Beruf oder Beschäftigung"
  },
  {
   "id": "professionOrOccupationAsLiteral",
   "name": "Beruf oder Beschäftigung (Literal)"
  },
  {
   "id": "professionalRelationship",
   "name": "Berufliche Beziehung"
  }
 ]
}
```


Die drei gelieferten `properties` können dann den Nutzenden zur Auswahl vorgeschlagen werden (s. Abb. 1, rechts oben: *Also use relevant details from other columns* sowie Abb. 3, links: *Add Property*). Analog kann vor dem Abgleich ein spezifischer Typ vorgeschlagen werden (`type`-Suggest-Dienst, s. Abb. 1, unten links: *Reconcile against type*), oder nach dem Abgleich gezielt nach einem Treffer gesucht werden (*entity suggest service*, s. Abb. 2, unterhalb der Vorschläge: *Search for match*).

2.5 Data Extension

Das Protokoll zur *data extension* ermöglicht eine Datenanreicherung auf Basis der abgeglichenen Treffer. Es besteht aus zwei wesentlichen Teilen: erstens der Kommunikation über die zur Datenanreicherung verfügbaren *properties* (s. Abb. 3, linker Bereich) und zweitens der eigentlichen Anreicherung mit den Werten der ausgewählten *properties* (s. Abb. 3, rechter Bereich).

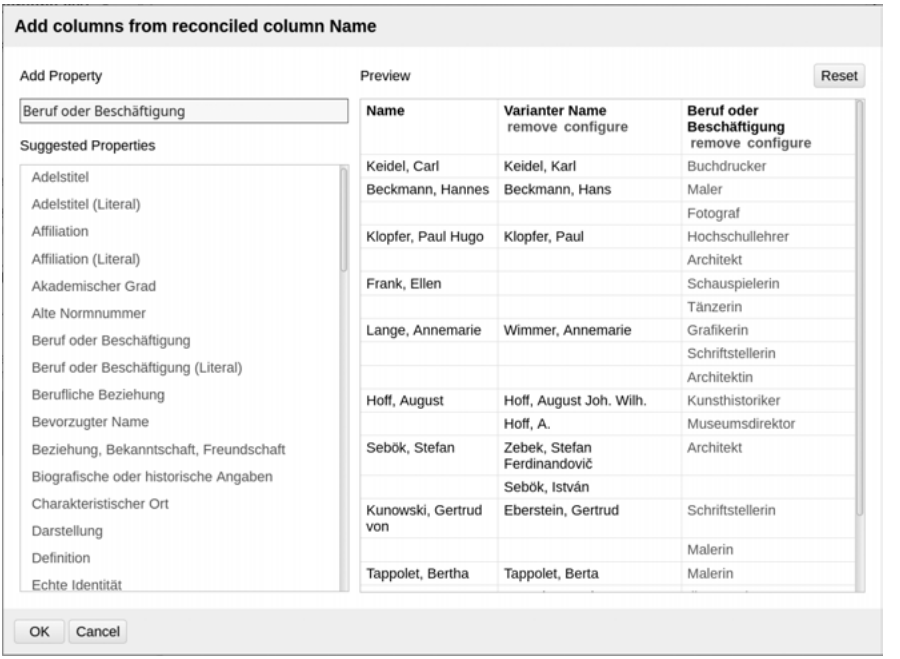

Abb. 3: Dialog zur *data extension* mit verfügbaren *properties* und Vorschau für ergänzte Spalten

2.5.1 Property-Vorschläge

Zunächst muss wieder das *service manifest* die Unterstützung für Property-Vorschläge deklarieren:

```
"propose_properties": {
  "service_url": "https://lobid.org",
  "service_path": "/gnd/reconcile/properties"
}
```

Ein solcher Dienst liefert die für einen bestimmten Typ (z. B. Work) verfügbaren *properties*:

```
https://lobid.org/gnd/reconcile/properties?type=Work
```

Hier eine gekürzte Antwort:


```
{
 "type": "Work",
 "properties": [
  {
   "id": "abbreviatedNameForTheWork",
   "name": "Abgekürzter Name des Werks"
  },
  {
   "id": "firstAuthor",
   "name": "Erste Verfasserschaft"
  },
  {
   "id": "preferredName",
   "name": "Bevorzugter Name"
  },
  {
   "id": "relatedConferenceOrEvent",
   "name": "In Beziehung stehende Konferenz oder Veranstaltung"
  }
 ]
}
```


Aus diesen *properties* können dann von Nutzenden diejenigen ausgewählt werden, die zur eigentlichen Datenanreicherung verwendet werden sollen (s. Abb. 3, links: *Suggested Properties*).

2.5.2 Extension-Anfragen

In der einfachsten Form werden bei der eigentlichen Anfrage zur *data extension* die Identifikatoren der zu verwendenden Entitäten sowie die gewünschten *properties* geschickt, z. B.:


```
{
  "ids": [
   "1081942517",
   "4791358-7"
  ],
```

```
  "properties": [
   {"id": "preferredName"},
   {"id": "firstAuthor"}
  ]
}
```

Als vollständige Anfrage etwa:

```
https://lobid.org/gnd/reconcile/?extend={"ids":
["1081942517","4791358-7"],"properties":[{"id":"preferredName"},
{"id":"firstAuthor"}]}
```

Die Antwort liefert die entsprechenden Daten:

```
{
 "meta": [
  {"id": "preferredName", "name": "Bevorzugter Name"},
  {"id": "firstAuthor", "name": "Erste Verfasserschaft"}
 ],
  "rows": {
   "1081942517": {
  "preferredName": [{"str": "Autobiography of Mark Twain"}],
  "firstAuthor": [{"id": "118624822", "name": "Twain, Mark"}]
  },
  "4791358-7": {
   "preferredName": [{"str": "Die größere Hoffnung (1960)"}],
   "firstAuthor": [{"id": "118501232","name": "Aichinger, Ilse"}]
  }
 }
}
```

Es erscheinen zunächst, unter `meta`, Informationen zu den angereicherten Daten: die Identifikatoren und Namen für die angereicherten Properties. Dies ist vergleichbar mit der Header-Zeile einer Tabelle (s. Abb. 3, rechts, fett gedruckte Kopfzeile). Im Folgenden erscheinen die einzelnen Datensätze in `rows`, jeweils unter dem Identifikator der Entität (hier das Werk), z. B. `1081942517`, die jeweiligen Properties (hier `preferredName` und `firstAuthor`). Die Struktur der Werte unterscheidet sich, da die Ansetzungsformen aus `preferredName` einfache Zeichenketten sind, während in `firstAuthor` GND-Entitäten enthalten sind, die jeweils wieder einen Identifikator und einen Namen haben. Diese Daten können

dann im *reconciliation client* den lokalen Daten hinzugefügt werden (s. Abb. 3, rechts).

Neben dieser Definition von Properties allein über ihre Identifikatoren (z. B. oben `{"id":"preferredName"}`) besteht die Möglichkeit, Properties zu konfigurieren. So kann etwa die Zahl der Werte für ein bestimmtes Feld bei der Anreicherung eingeschränkt werden, wenn z. B. nicht alle, sondern nur die ersten fünf Namensvarianten zurückgegeben werden sollen. Die Unterstützung für eine solche Konfiguration deklariert der Dienst zunächst etwa so:

```
"property_settings": [
 {
   "name": "limit",
   "label": "Limit",
   "type": "number",
   "default": 0,
   "help_text": "Maximum number of values to return per row (0 for no limit)"
 }
]
```

Neben der freien Eingabe soll die Konfiguration oft auch mit festen Werten erfolgen. So können für bestimmte Felder in der GND etwa Identifikatoren oder Namen geliefert werden. Damit Nutzende dies je nach Bedarf im *reconciliation client* auswählen können, definiert der Dienst zusätzlich zu den Werten oben `choices`, z. B.:

```
"property_settings": [
 {
   "name": "content",
   "label": "Content",
   "type": "select",
   "default": "literal",
   "help_text": "Content type: ID or literal",
   "choices": [
    { "value": "id", "name": "ID" },
    { "value": "literal", "name": "Literal" }
   ]
 }
]
```

Die in `choices` beschriebenen Optionen können dann in einem *reconciliation client* bei der Auswahl und Konfiguration der zur Datenanreicherung zu verwendenden Properties angezeigt werden, z. B. in Form eines Auswahlmenüs für die oben im *service manifest* deklarierten Werte (s. Abb. 3, rechts, jeweils: *configure* öffnet einen Konfigurationsdialog).

Bei der entsprechenden *data extension* Anfrage wird die jeweilige Konfiguration dann mitgeschickt, z. B.:

```
{
 "ids": [
  "10662041X",
  "1064905412"
 ],
 "properties": [
  {
   "id": "variantName",
   "settings": {
    "limit": "5"
   }
  },
  {
   "id": "professionOrOccupation"
  },
  {
   "id": "geographicAreaCode",
   "settings": {
    "limit": "1",
    "content": "id"
   }
  }
 ]
}
```

Hier werden für zwei Entitäten jeweils drei Properties angefordert: erstens `variantName` (konfiguriert mit einem `limit` von 5), zweitens `professionOrOccupation` (ohne Konfiguration) und schließlich drittens `geographicAreaCode` mit einem `limit` von 1 und als `content` den Identifikator (über die im *service manifest* deklarierte Option mit `value: id`). Hier werden die Daten also mit maximal fünf Namensvarianten, allen verfügbaren Berufen und einem Ländercode angereichert.

3 Ausblick

Auf Basis dieser Darstellung des Protokolls und seiner Verwendung in *OpenRefine* sollen die folgenden zwei Abschnitte einen Ausblick auf die Arbeiten der Entity Reconciliation Community Group des World Wide Web Consortiums (W3C) und das weitergehende Ökosystem rund um die Reconciliation-API geben.

3.1 W3C Community Group

Aufgabe der *W3C Entity Reconciliation Community Group* ist die Entwicklung einer Web-API, mit der Datenanbietende einen Abgleich von Drittdaten mit den eigenen Identifikatoren ermöglichen können. Ausgangspunkt der Community Group bildet die beschriebene Implementierung in OpenRefine. Diese API soll zunächst dokumentiert und dann auf Basis ihrer Nutzung (s. Delpeuch 2019) zu einem Standard weiterentwickelt werden.[5] Als ein Beispiel sei hier eine Diskussion zur Frage der Nachvollziehbarkeit von Algorithmen zur Bewertung (*Scoring*) von Reconciliation-Kandidaten genannt,[6] die anschließend zu einer Erweiterung des Protokolls geführt hat.[7]

Diese Entwicklung einer Web-API durch die Community Group umfasst die Arbeit an den eigentlichen Spezifikationen, der Satzung der Gruppe, einer Webanwendung zum Testen von Reconciliation-Diensten sowie der Erfassung des Reconciliation-Ökosystems aus Diensten, Clients und sonstiger Software rund um die Reconciliation-API. Die konkrete Arbeit findet innerhalb einer GitHub-Organisation[8] statt. Der Aufgabenbereich der Gruppe schließt neben den etablierten Anwendungsfällen des Zusammenführens von Daten aus verschiedenen Quellen auch ähnliche Anwendungsfälle wie die Deduplizierung von Daten aus einer einzigen Quelle ein. Die Gruppe steht allen Interessierten offen und freut sich über jegliche Art von Mitarbeit und Unterstützung.

5 Die vollständige Charter der Gruppe findet sich unter https://reconciliation-api.github.io/charter/ (4.12.2020).

6 https://lists.w3.org/Archives/Public/public-reconciliation/2020Jul/0000.html (4.12.2020).

7 https://github.com/reconciliation-api/specs/pull/38 (4.12.2020).

8 https://github.com/reconciliation-api (4.12.2020).

3.2 Reconciliation-Ökosystem

Die *Reconciliation service test bench*, eine Webanwendung zum Testen von Reconciliation-Diensten, ist zugleich ein Werkzeug für die Entwicklung eines eigenen Reconciliation-Dienstes und in Form einer zentralen Instanz[9] eine Übersicht der in Wikidata verzeichneten[10] Reconciliation-Dienste mit ihren jeweils unterstützen Features.

Über diese Übersicht der Dienste hinaus, gibt es im Rahmen der Community Group eine Erfassung des Reconciliation-Ökosystems von Diensten, Clients und sonstiger Software rund um die Reconciliation-API.[11] Hier findet sich etwa eine Übersicht alternativer Clients, die statt OpenRefine mit einem Reconciliation-Dienst kommunizieren, z. B. das Mapping-Tool Cocoda (Balakrishnan 2016) oder die Alma-Refine-App[12] mit GND-Integration.[13]

Im Reconciliation-Ökosystem finden sich also reichlich Dokumentation, Werkzeuge und Beispiele für die Entwicklung eigener Reconciliation-Dienste. Über einen solchen Dienst können Datenanbietende, die eigene Identifikatoren prägen, ihre Daten über eine einheitliche API zur Integration durch Dritte zur Verfügung stellen. So wird, auch über den Bereich der Inhaltserschließung hinaus, durch das beschriebene Protokoll eine einheitliche Erfassung von Entitäten ermöglicht, ohne dass diese in allen vorliegenden Datenquellen einheitlich identifiziert sein müssen.

4 Literaturverzeichnis

Alvestrand, Harald Tveit und Håkon Wium Lie: Development of Core Internet Standards: The Work of IETF and W3C. In: Internet Governance – Infrastructure and Institutions. Hrsg. von Lee A. Bygrave und Jon Bing. S. 126–146. Oxford University Press 2009. http://dblp.uni-trier.de/db/books/collections/BB2009.html#AlvestrandL09 (4.12.2020).

Balakrishnan, Uma: DFG-Projekt Coli-Conc: Das Mapping Tool „Cocoda“. In: o-bib. Das Offene Bibliotheksjournal/Herausgegeben vom VDB (2016) Bd. 3 Nr. 1. S. 11–16. https://doi.org/10.5282/o-bib/2016H1S11-16.

9 https://reconciliation-api.github.io/testbench/ (4.12.2020).

10 https://reconciliation-api.github.io/census/services/#how-to-add-a-service-to-the-test-bench (4.12.2020).

11 https://reconciliation-api.github.io/census/ (4.12.2020).

12 https://developers.exlibrisgroup.com/blog/how-to-install-and-use-the-alma-refine-cloud-app-2-2/ (4.12.2020).

13 https://developers.exlibrisgroup.com/blog/how-to-use-the-alma-refine-cloud-app-for-service-gnd/ (4.12.2020).

Delpeuch, Antonin: A survey of OpenRefine reconciliation services. 2019. arXiv:1906.08092. http://arxiv.org/abs/1906.08092.

Delpeuch, Antonin, Adrian Pohl, Fabian Steeg, Thad Guidry Sr. und Osma Suominen: Draft: Reconciliation Service API – a Protocol for Data Matching on the Web. https://reconciliation-api.github.io/specs/latest/ (4.12.2020).

Hauser, Julia: Der Linked Data Service der Deutschen Nationalbibliothek. Dialog mit Bibliotheken (2014) H. 1. S. 38–42. https://d-nb.info/1118655494/34 (4.12.2020).

Hooland, Seth van, Ruben Verborgh, Max De Wilde, Johannes Hercher, Erik Mannens und Rik Van de Walle: Evaluating the Success of Vocabulary Reconciliation for Cultural Heritage Collections. In: Journal of the Association for Information Science and Technology (2013) Bd. 64 Nr. 3. S. 464–479. https://doi.org/10.1002/asi.22763.

Khalid, Mahboob A., Valentin Jijkoun und Maarten de Rijke: The Impact of Named Entity Normalization on Information Retrieval for Question Answering. In: Advances in Information Retrieval. 30th European Conference on Information Retrieval (ECIR 2008). Hrsg. v. Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven und Ryen W. White. Berlin, Heidelberg: Springer 2008. S. 705–710. https://doi.org/10.1007/978-3-540-78646-7_83.

Singhal, Amit: Introducing the Knowledge Graph: Things, Not Strings. 2012. https://www.blog.google/products/search/introducing-knowledge-graph-things-not/ (4.12.2020).

Slawski, Bill: How Google Uses Named Entity Disambiguation for Entities with the Same Names. 2015. https://www.seobythesea.com/2015/09/disambiguate-entities-in-queries-and-pages/ (4.12.2020).

Steeg, Fabian, Adrian Pohl und Pascal Christoph: lobid-gnd – Eine Schnittstelle zur Gemeinsamen Normdatei für Mensch und Maschine. In: Informationspraxis (2019) Bd. 5 Nr. 1. https://doi.org/10.11588/ip.2019.1.52673.

Target, Sinclair: The Rise and Rise of JSON. 2017. https://twobithistory.org/2017/09/21/the-rise-and-rise-of-json.html (4.12.2020).