デジタルアーカイブと
その社会的活用

第9回

立命館大学 映像学部講義
福田一史

https://scrapbox.io/fukudakz/21デジタルアーカイブとその社会的活用

manabaRにもリンクがあります

コンテンツ

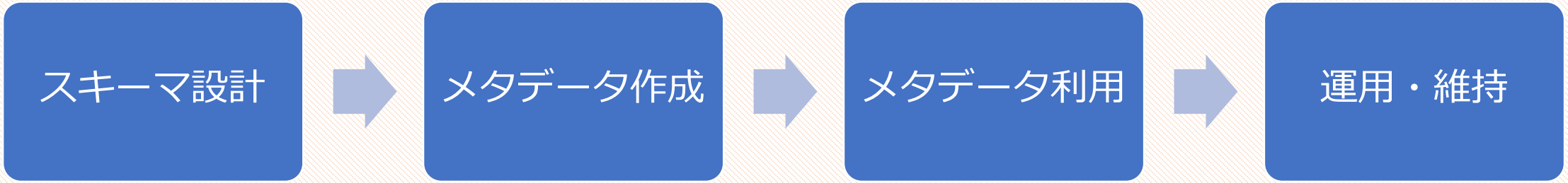
1. メタデータ作成フロー
2. メタデータ・LODの事例
3. メタデータの検索と利用

メタデータ作成フロー

メタデータの設計・作成・公開・維持

メタデータのライフサイクル

- メタデータは下記のフローで生成・利用される。
- 一貫して共通性や標準性を検討する必要がある。
- 具体的にどのような論点に注意すべきか、について以下で論じる。



スキーマの設計

- スキーマの選択・設計と公開の指針（メタデータ情報基盤構築事業 2011, p. 9） ※括弧内のA, B, Cは優先度
  1. スキーマを相互運用可能な形で選択・設計する（A）
  2. 新たなスキーマを設計する場合、仕様語彙や参照記述規則の定義を尊重し、メタデータを相互運用できるよう設計する（A）
  3. 独自スキーマを設計する場合も、特定領域の知識なしに理解し、交換可能なフォーマットに変換するための情報・規則を用意する（B）
  4. スキーマ定義を、コンピュータ処理可能な標準方法でも表現し、公開する（C）

メタデータの仕様

- メタデータの仕様を共通化するための方式として、DSP（記述セットプロファイル）がある。ここでは簡易DSPについて紹介する。
- 簡易DSPは、記述規則ブロックと名前空間宣言ブロックで構成される。

メタデータの仕様

- メタデータの仕様を策定・公開するための方式として、DSP（記述セットプロファイル）がある。ここでは簡易DSPについて紹介する。
- 簡易DSPは、記述規則ブロックと名前空間宣言ブロックで構成され、タブ区切りファイル（.tsv）で記述される（次々ページ）。

メタデータの仕様

- 項目記述規則
  - 項目規則名、プロパティの修飾名、最小出現回数、最大出現回数、値タイプ、値制約、コメント（説明）、といった要素で構成される。

項目規則名	プロパティの修飾名	最小出現回数	最大出現回数	値タイプ	値制約	コメント
タイトル	schema:name	1	1	文字列		リソースのタイトル／名前
ISBN	schema:isbn	0	1	文字列		リソースのISBN
発行日	schema:datePublished	1	1	文字列	dcterms:W3CDTF	リソースが発行された日付
著者	schema:creator	0	-	構造化	foaf:Agent	リソースの作者
出版者	schema:publisher	0	-	構造化	foaf:Agent	リソースの公開者

[@ns]
schema: http://schema.org/

[Book]

#項目規則名	プロパティの修飾名	最小出現回数	最大出現回数	値タイプ	値制約	コメント
タイトル	schema:name	1	1	文字列		リソースのタイトル／名前
ISBN	schema:isbn	0	1	文字列		リソースのISBN
発行日	schema:datePublished	1	1	文字列	dcterms:W3CDTF	リソースが発行された日付
著者	schema:creator	0	-	構造化	foaf:Agent	リソースの作者
出版者	schema:publisher	0	-	構造化	foaf:Agent	リソースの公開者

メタデータ記述

- メタデータ記述の推奨指針（メタデータ情報基盤構築事業 2011, p. 9） ※括弧内のA, B, Cは優先度
  1. リソースにグローバルな識別子（**URI**）を与える（A）
  2. **人間に理解可能なラベル**を標準的な方法で与える（A）
  3. 標準的で再利用可能な形で、コンテンツの作者を記述する（B）
  4. 曖昧さのない**標準形式**で日時、位置情報を付与する（B）
  5. 可能ならばキーワードを統制語彙で付与する（B）
  6. ラベルに読みを与える場合は、**言語タグ**を用いて区別するか、ラベルを構造化して記述する（C）
  7. リテラル値のデータ型、言語タグは、目的が明確な場合に限って用い、スキーマで仕様を宣言して一貫した形で与える（C）

ISO8601

- 日付と時刻の標準形式
- 年月日の順番は地域により違う場合がある。例えばアメリカでは月-日-年の順番で日付を記す場合が多く、日本で広く用いられる形式と違う。
  - e.g. June 19, 2021
- 年月日を「YYYYMMDD」（基本方式）や「**YYYY-MM-DD**」（拡張方式）により記述する。視認性や年の明記のため拡張方式が頻繁にもちいられる。
  - e.g. 2028-01-19
- **時刻**や**期間の指定**なども定義される。
- ref. https://ja.wikipedia.org/wiki/ISO_8601

ISO639

- 言語の表記（略号）の標準形式
- リソース（e.g. 図書）やデータなど言語を指定すべき機会は多いが、例えば「日本語」を示す表記にも複数のものが存在しややこしい。ISO639では、2文字や3文字のアルファベットで言語を指定するルールを定義している。
  - e.g. ja （日本語）, jpn（日本語）, de（ドイツ語）, ger（ドイツ語）
- ref. https://ja.wikipedia.org/wiki/ISO_639-1コード一覧

メタデータの公開

- メタデータの公開と交換・利用に関する指針（メタデータ情報基盤構築事業 2011, p. 9） ※括弧内のA, B, Cは優先度
  1. メタデータの公開には、**標準的なデータ形式としてRDFを用いる**（A）
  2. メタデータを**正しく理解・利用するためにスキーマを参照**し、必要に応じてプロパティの整合調整を行う（B）
  3. データを公開用などに変換する場合は、情報が失われないように構造と粒度を保ち、利用者がダムダウンする。主要プロパティはあらかじめ単純化値を提供する（B）
     - ダムダウン＝プロパティ単純化

メタデータの運用

- 運用に関する指針（メタデータ情報基盤構築事業 2011, p. 9-10） ※括弧内のA, B, Cは優先度
  1. スキーマの管理データを明示し、**バージョン管理**を行う（A）
  2. メタデータには作者、作成日時、準拠スキーマなどの**管理データを付与**する（A）
  3. データを集約して格納する場合、由来情報とあわせて管理する（B）
  4. スキーマを**公開レジストリに登録**し、利用者の発見を助けるとともに、**最新版、旧版を確認できるようにする**（B）
  5. メタデータを作成・公開する場合、スキーマの記述規則と矛盾がないか検証する（C）

メタデータ・LODの事例

公開されるLODデータセット

CiNii

- https://ci.nii.ac.jp/
- 国立情報学研究所（NII）が運営する日本の論文データベース
- 書誌データがDublin Coreなどを用いたLOD形式で提供される
  - https://support.nii.ac.jp/ja/cinii/api/api_outline#RDF
  - 詳細ページのURLの末尾に「.rdf」か「.json」をつけてアクセスすることで、LODデータにアクセス可能。
    - e.g. https://ci.nii.ac.jp/naid/170000151256.rdf
  - その他に、OpenSearch やRSSによるデータ提供も行われる

Japan Search

- https://jpsearch.go.jp/
- Schema.orgを用いた直接記述と、独自語彙（JPS）で定義される構造化記述で構成されるデータモデル
- 日本のデジタルアーカイブのポータルサイトであり、国立国会図書館の全国書誌のほか、多数のデータベースのメタデータが登録される
- SPARQLエンドポイントのほか、EasySPARQLも提供
  - https://jpsearch.go.jp/rdf/sparql-explain/

DBPedia

- http://ja.dbpedia.org/ （日本語版）
- WikipediaのLOD化プロジェクトその1
- Wikipediaから情報を抽出して構造化データを生成する
- 独自のメタデータ語彙で、基本的にWikipedia記事に基づき記述される

Wikidata

- https://www.wikidata.org/
- WikipediaのLOD化プロジェクトその2
- 2012年からウィキメディア財団により新たなプロジェクトとして開始される、とりわけWikipediaの事実データの構造化と言語間リンクに注力する点が特徴。
  - DBpediaとの違いについては以下の論文などに詳しい
    - 加藤文彦. 2017. DBpediaの現在：リンクトデータ・プロジェクト. 情報管理. 60(5), 307-315. https://doi.org/10.1241/johokanri.60.307
- 独自のメタデータ語彙からなるオントロジー・データモデルで記述される

RCGSコレクション

- https://collection.rcgs.jp
- 立命館大学ゲーム研究センターの所蔵資料のオンライン目録

メタデータの検索と活用

SPARQLを用いたRDFデータの活用

SPARQL

- RDFで記述されたメタデータは、問い合わせ言語である「**SPARQL**（SPARQL Protocol and RDF Query Language）」を用いることで、リッチな検索や識別やデータ分析を可能とする。
  - ブラウザでアクセスするウェブのGUIでは達成できない、もしくは非常にコストがかかる機能を、数多くかつ容易に達成できる。
- 2008年よりVer. 1.0が、2013年にVer. 1.1がW3C勧告
  - SPARQL 1.1 Query Language (W3C), 日本語版
- SPARQLはRDFに出現する**パターン**の組み合わせや、フィルタリング、文字列指定などで必要とするデータの指定が可能。
- PHP, JavaScript, Perl, Python, Ruby, Rなど**複数のプログラム言語で**SPARQLを実装するための**ライブラリが公開**されている。

RDFストア

- RDFストアとは、**膨大なRDFデータ**（数十万～数億トリプル）**を登録し検索する**ためのデータベースである。トリプルストアとも呼ばれる。
- 複数のRDFストアが公開されている
  - e.g. Apache Jena, Virtuoso
  - **Comparison of triplestores – Wikipedia**

SPARQLエンドポイント

- SPARQLによるRDFデータの検索や分析の機能を提供するインターフェイス。
- ウェブでは数多くのSPARQLエンドポイントが公開されており、これらからSPARQLを用いたデータ検索が可能となっている。
  - e.g. Wikidata Query Service
  - e.g. Snorql for Japan Search

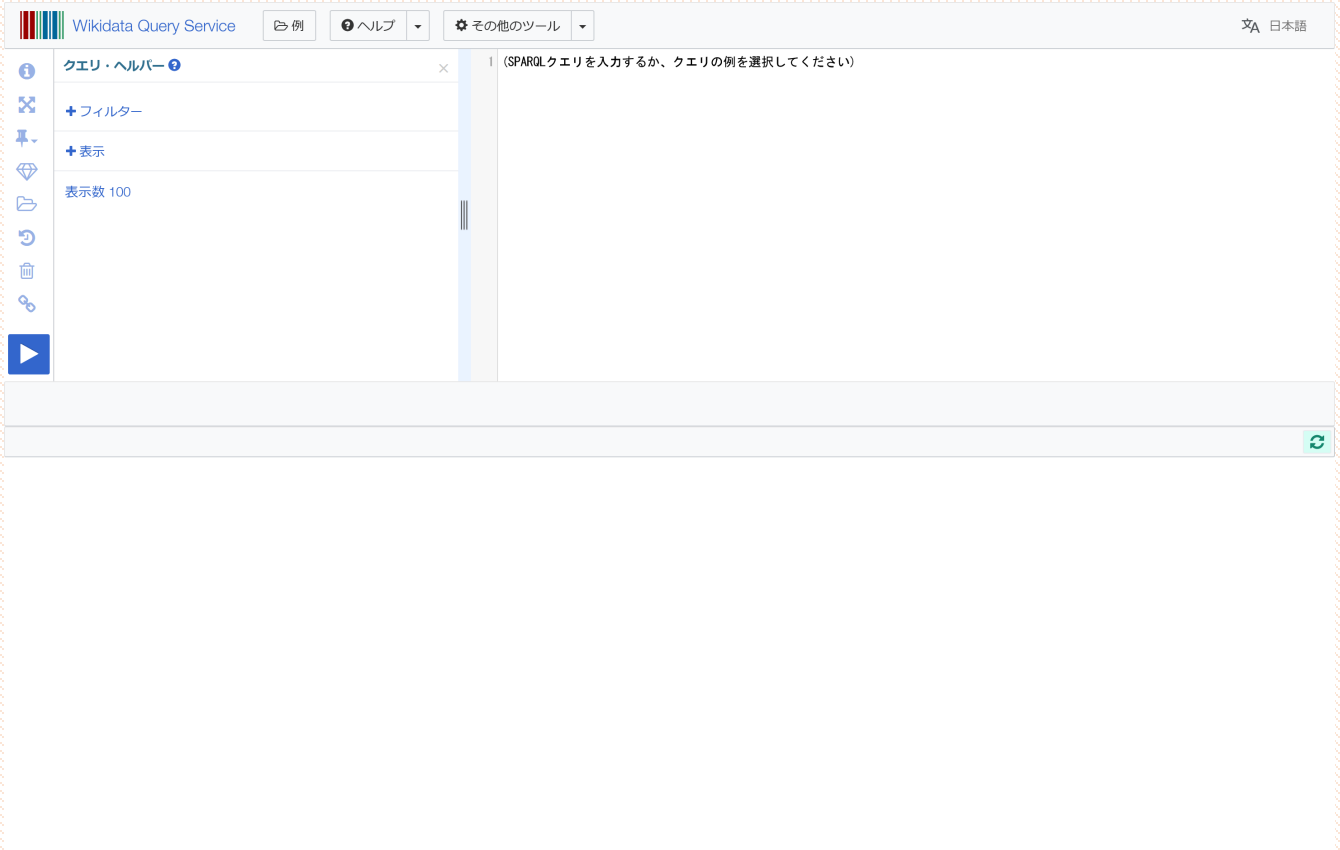


図. Wikidata Query Service

SPARQLを用いた検索サンプル

- SPARQLを用いて、高度な検索が可能となる
- 様々なSPARQLエンドポイントのクエリ（問い合わせ）のサンプルがウェブ上で公開されている。
  - e.g. Wikidata:SPARQL query service/queries/examples
  - e.g. RCGS SPRQLクエリサンプル

SPARQLを用いたデータの指定

- 右のクエリは指定のURIを主語とするリソースのRDFグラフを取得する
  - DESCRIBEは特定のリソースのRDFグラフを応答する
  - 上：ジャパンサーチでの問い合わせ
  - 下：Wikidataでの問い合わせ

```
DESCRIBE <https://jpsearch.go.jp/data/michi-D0004990094_00000>
```

```
DESCRIBE <http://www.wikidata.org/entity/Q24862683>
```

SPARQLを用いた検索

- 右はSPARQLの基本的な検索パターンに基づくサンプルクエリ（ジャパンサーチのSPARQLエンドポイントでの検索）
  - PREFIXは**URIを省略するための接頭辞**を定義する
  - SELECTは**変数を定義**する
  - WHEREはRDFの**記述パターンを指定**する
  - LIMITは結果の件数の上限の指定
- 本クエリで指定した変数の値をテーブル形式により取得できる
  - 検索結果
  - エンドポイントの機能にもよるが、JSONやCSVなどの構造化データのファイル形式でデータを取得することも可能

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?s ?label
WHERE {
  ?s rdf:type type:刀剣 .
  ?s rdfs:label ?label .
}
LIMIT 10
```

データ検索演習（１）

1. 2つ前のページのDESCRIBEを試して、レスポンスを取得してみよう
2. 前ページのクエリを試して、レスポンスを取得してみよう
3. 2) を元にジャパンサーチの「type:版画」のリソースとラベルのレスポンスを取得してみよう

- Scrapboxの「SPARQLクエリリスト」にコマンドが貼ってあります。名前の右のファイルボタンを押すとコピーできます。

SPARQLを用いた検索

- 前ページの検索より少し高度なクエリ
  - OPTIONALは値がある場合のみデータを返す
  - 「;」（セミコロン）でTurtleのように繰り返しの主語を省略できる
  - 亀甲括弧 [] で入れ子による構造的記述の値を指定できる

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX schema: <http://schema.org/>

SELECT ?s ?label ?thumb ?provider
WHERE {
  ?s rdf:type type:刀剣 ;
     rdfs:label ?label .
  OPTIONAL { ?s schema:image ?thumb ;
                jps:accessInfo [ schema:provider ?provider ]
           }
}
LIMIT 10
```

SPARQLを用いた分析

- 検索結果の件数をリスティング、ソートした結果を示す。
  - countは件数をカウントする
  - order by は表示順序のルールを指定する

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT (count(?provider) as ?number) ?provider
WHERE {
  ?s rdf:type type:刀剣 ;
     rdfs:label ?label ;
     jps:accessInfo [ schema:provider ?provider ] .
}
order by desc(?number)
```

データ検索演習（２）

1. 「type:絵画」の所蔵者のリストと所蔵数をレスポンスとして取得してみよう。
2. これまでに得た知識で、新しいSPARQLのクエリを作ってみよう。サンプルクエリも参考にしてください。
   - 作成したクエリについて、**1) どのようなクエリであるかの説明**と、**2) クエリ結果が生成されたページのURL**を、コミュニケーションノートのコメントとして投稿してください。
     - 投稿の例）
       所在地ごとのジャパンサーチ登録機関件数
       https://jpsearch.go.jp/rdf/sparql/easy/?query=SELECT+%28count%28%3Fprovider%29+as+%3Fnumber%29+%3Fprovider%0D%0AWHERE+%7B%0D%0A++%3Fs+rdf%3Atype+type%3A%E5%88%80%E5%89%A3+%3B%0D%0A+++++rdfs%3Alabel+%3Flabel+%3B%0D%0A+++++jps%3AaccessInfo+%5B+schema%3Aprovider+%3Fprovider+%5D+.%0D%0A%7D%0D%0Aorder+by+desc%28%3Fnumber%29%0D%0A