

機関リポジトリデータ移行時の修復プログラム作成とエラー分析

齋木 匠[†] 小林 裕太^{††} 武内八重子^{††} 荒井 幸代^{†††} 檜垣 泰彦^{††††}

[†] 千葉大学 大学院融合理工学府 〒263 – 8522 千葉市稲毛区弥生町 1 – 33

^{††} 千葉大学附属図書館 〒263 – 8522 千葉市稲毛区弥生町 1 – 33

^{†††} 千葉大学大学院工学研究院 〒263 – 8522 千葉市稲毛区弥生町 1 – 33

^{††††} 千葉大学アカデミック・リンク・センター 〒263 – 8522 千葉市稲毛区弥生町 1 – 33

E-mail: †afma6906@chiba-u.jp

あらまし 日本国内において、各大学の機関リポジトリのメタデータは共通のメタデータスキーマで IRDB によって収集されている。近年、オープンサイエンスの展開を見据えた新しいメタデータスキーマ・JPCOAR スキーマが策定され、各大学が対応を進めている。千葉大学でも、それまで利用していた junii2 スキーマから JPCOAR スキーマへの移行を検討したが、システム更新時のデータ移行設計の不備により、現状のメタデータでは JPCOAR スキーマへの対応が困難であると判明した。本研究では、JPCOAR スキーマへの対応のため、著者順の並び順修正およびデータ上で機械的に修正可能なデータ登録時に生じたミスの修正を行うプログラムを作成した。

キーワード 機関リポジトリ, データ修復, JPCOAR, junii2

Development of repair program for institutional repository data migration and error analysis

Takumi SAIKI[†], Yuta KOBAYASHI^{††}, Yaeko TAKEUCHI^{††}, Sachiyo ARAI^{†††}, and Yasuhiko HIGAKI^{††††}

[†] Graduate School of Science and Engineering, Chiba University 1 – 33 Yayoi-cho, Inage-ku, Chiba-shi, 263 – 8522 Japan

^{††} Chiba University Library 1 – 33 Yayoi-cho, Inage-ku, Chiba-shi, 263 – 8522 Japan

^{†††} Graduate School of Engineering, Chiba University 1 – 33 Yayoi-cho, Inage-ku, Chiba-shi, 263 – 8522 Japan

^{††††} Academic Link Center, Chiba University 1 – 33 Yayoi-cho, Inage-ku, Chiba-shi, 263 – 8522 Japan

E-mail: †afma6906@chiba-u.jp

Abstract In Japan, IRDB collects metadata of each university's institutional repositories with a standard metadata schema. In recent years, a new metadata schema, JPCOAR schema, has been developed in anticipation of the development of open science, and each university has been working on its adaptation. Chiba University also considered migrating from the previously used junii2 schema to the JPCOAR schema. However, it was challenging to support the JPCOAR schema with the current metadata due to inadequate data migration design at the time of the system upgrade. In this study, we developed a program that corrects the order of authors and corrects errors in data registration, which can be corrected mechanically on the data, in order to support the JPCOAR schema. We also examined the classification and causes of the errors that could not be repaired.

Key words institutional repository, data recovery, JPCOAR, junii2

1. はじめに

日本の多くの大学では学生、教員の研究成果を保存、公開する機関リポジトリを運用している。国内において、各大学が運用する機関リポジトリに登録されたコンテンツのメタデー

タは、共通のメタデータスキーマで IRDB によって収集され、CiNii 等の様々なシステムに提供されている [1]。現時点で最新のメタデータスキーマは、JPCOAR スキーマ Version2.0 である [2]。

千葉大学では、2005 年に機関リポジトリの運用を開始し、

2006年に策定された junii2 スキーマ [3] によるメタデータ収集に対応してきた。しかし、2016年に行ったシステム更新のデータ移行設計に不備があり、著者の並び順が入れ替わる等の問題が発生した。それにより、JPCOAR スキーマへの移行を検討した際、現状のメタデータでは JPCOAR スキーマへの対応が困難であると判明した。さらに、メタデータの分析を進めたところ、データ登録時のミスに起因すると思われるエラーも確認された。

そこで本研究では、プログラムの修正可能な問題を分類した上で、著者並び順の復元とデータの修正を行うプログラムを作成した。また、修復しきれないデータについて、その種類と発生原因を分析することとした。

以下、2節では千葉大学におけるこれまでのメタデータ運用について述べる。3節では、メタデータスキーマ移行で最も問題となる著者データについて説明する。4節では、メタデータスキーマ移行の際に生じた著者順序の乱れについて述べる。5節では、データ修復の手法について述べ、6節では修復時のデータ分析結果を示す。7節では、結論と今後の課題について述べる。

2. 機関リポジトリにおけるメタデータスキーマの運用

国内の機関リポジトリにおいては、IRDB に提供するための統一的なメタデータスキーマと、各機関の特性に合わせた機関リポジトリの内部で使用しているメタデータスキーマの二つを同時に利用する必要がある。通常、機関リポジトリに登録されているメタデータを、内部で使用しているメタデータスキーマから、統一的なメタデータスキーマに変換し、IRDB に提供することとなる。この流れを図 1 に示す。以下、統一的なメタデータスキーマとして、junii2 スキーマ、JPCOAR スキーマ、千葉大学の機関リポジトリシステム内部で用いられている curator スキーマについて説明する。

2.1 junii2 スキーマ

junii2 スキーマは、国立情報学研究所が機関リポジトリの相互運用性確保のために策定したメタデータ・フォーマット [3] である。各要素はすべての情報をフラットに記述する構造となっている [4]。

2.2 JPCOAR スキーマ

JPCOAR スキーマは、今後のオープンサイエンスの展開を見据えた junii2 の抜本的な改訂が必要とされたため、オープンアクセスリポジトリ推進協会により 2017 年に策定された新たなメタデータスキーマである [4]。関連する情報を階層化して記述できるようになっている。

2.3 curator スキーマ

千葉大学の機関リポジトリ「CURATOR」では、junii2 スキーマを拡張したスキーマ (以下、curator スキーマ) をシステム内部で使用している。

curator スキーマは、junii2 ガイドラインでは定義されていない要素を追加したスキーマである。具体的には、著者名のヨミ、英語表記、著者 ID 等を追加している。

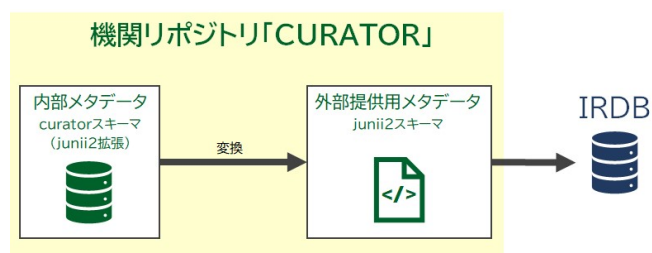


図 1 メタデータが IRDB に収集されるまでの流れ

curator スキーマのメタデータを junii2 スキーマに変換したメタデータが IRDB に収集されており、その際これらの追加項目については、出力対象から除外されている。

3. 著者データの要素

curator スキーマの著者データは、以下の要素、属性によって表現される。

3.1 Creator 要素

Creator 要素には、データ上に記載された著者の名前が含まれている。日本語の論文であれば漢字で、英語論文の場合はアルファベット表記で記載されている。

3.2 Creator.NC 要素

Creator.NC 要素は、Creator 要素と同様、データ上の著者の表記が含まれている。Creator 要素との違いは、AuthorityID.Creator の有無である。各著者は基本的に、Creator か Creator.NC のどちらか一方を持つ。

3.3 Creator.Transcription 要素

Creator.Transcription 要素には、著者の名前をカタカナ表記にし、言語によらない読みの表記が含まれる。Creator.Transcription 要素は複数持つことができる。

3.4 Creator.Transcription.NC 要素

Creator.Transcription 要素と同様、著者の名前をカタカナ表記にし、言語によらない読みの表記が含まれる。Creator.Transcription 要素との違いは、AuthorityID.Creator の有無である。

3.5 Creator.Alternative 要素

Creator.Alternative 要素には、英語論文における日本人著者の漢字表記など、著者の別表記が含まれる。

3.6 lang 属性

lang 属性は、Crator, Creator.NC, Creator.Transcription, Creator.Alternative の各要素に付与されており、その要素が表記された言語を示している。

3.7 AuthorityID.Creator 要素

AuthorityID.Creator 要素は、原則、Creator.NC 要素とその Creator.Transcription.NC に付与されており、著者の ID を示している。

4. 機関リポジトリ上のデータ表現とデータ移行に伴う著者順序の乱れ

4.1 移行前のデータ表現

千葉大学で従来用いられてきた curator スキーマは、xml 形

Algorithm 1 データ修復の流れ

```
1: データを階層型の辞書に変換
2: if 名寄せ可能なら then
3:   Alternative 要素から Creator 要素と Creator.NC 要素の並び順を
   修復
4: if N(Creator) = 0 or N(Creator.NC) = 0 then
5:   順序を修復せず名寄せ (整列済み)
6: else if N(Creator) ≥ 0 or N(Creator.NC) ≥ 0 かつ N(Creator) +
   N(Creator.NC) ≥ N(Alternative) then
7:   名寄せ
8:   Creator.Alternative の順序に従って, Creator 要素と Creator.NC 要
   素の順序を修復
9: else 順序の修復せず名寄せ (順序修復は不可)
10: 辞書をフラット化
11: lang 属性の修正
12: キーを短縮
```

式で表記され、階層的に要素を含んでいる。図2に、本来の並び順のデータ例を示す。

4.2 メタデータスキーマ移行時の問題点と著者順序の乱れ

junii2 スキーマから JPCOAR スキーマへの移行に影響する大きな変更点は、すべての要素がフラットに記述する構造から、関連する情報を階層化して記述できるようになった点である。また、junii2 では定義されていなかった、著者名のヨミ・英語表記、著者 ID 等が JPCOAR スキーマでは「該当する場合は必須」となっている [2]。

千葉大学では、2016年に CURATOR のシステム更新を行った。その際、データ移行設計に不備があり、移行前に区別していなかった著者 ID を持たない Creator と著者 ID を持つ Creator が Creator と Creator.NC という別の要素に分かれた。その結果、著者名の並び順が変更され、これらの要素がデータの末尾に移動されたため、順序の乱れが生じた。図3に、著者順番の乱れたデータ例を示す。著者順番の乱れたデータを元に戻すことが、本研究の目的となる。

4.3 納品データ形式

機関リポジトリの xml 形式は直接編集できないため、表1に示す、各要素を列とした表形式で納品する必要がある。

5. データ修復

本研究では、python を用いてデータ修復の流れを Algorithm 1 に示す。以下、各修復手順を説明する。

5.1 データの階層化

Creator 要素をはじめとする各要素には lang 属性が附属している。リポジトリ上のデータでは、図2に示す通り、lang 属性は Creator 要素の中に入れ子になっているため、Creator の順序並び変えがあっても lang 属性の順番を気にする必要はない。しかし今回納入するデータ形式は表1に示すフラットなデータ形式になっているため、ある lang 属性がどの Creator データのものであるかは、データの順序によってのみ示される。部分的に順序を維持したまま名寄せや順番変更を実施するのは困難でことから、データを階層型の辞書に変換することとした。

データの階層化の例を図4に示す。著者の名前データに対応する key が存在しないため、仮の key として Creator.name を付与した。

5.2 姓名区切りの修復

データの中には、名字の区切りが ”,”(半角カンマ + 半角空白)ではなく” ”(半角空白のみ)や” ”(全角空白)で区切られているデータが存在した。更に、組織名の読みは,””(半角空白)で分かち書きされていた。そこで、名前を””(半角空白)で区切った際に二つに分割された場合のみ,”,”を補完する処理をした。

5.3 名寄せ

順序修復の前段階として Creator 要素、Creator.NC 要素に対して Creator.Alternative 要素や Creator.Transcription 要素のデータを名寄せし、各著者データを整理する必要がある。機関リポジトリに登録されている文献では、著者は人とは限らず、組織の場合もある。以下、それぞれの名寄せについて説明する。

5.3.1 人 名

ひらがな、カタカナ、アルファベットについては、Python の `jaconv` ライブラリ^(注1)を利用する。このライブラリでは、ひらがなとカタカナの変換、全角と半角の変換、カタカナとローマ字の変換が可能である。しかし、名前のローマ字表記については、個人で異なる場合がある。そこで、カタカナからローマ字への変換後、考えられる別のローマ字表記を候補に加えることとした。変換を表3に示す。

漢字表記の名前に対しては、`jinmei-dict`^(注2)を用いて、読みの候補を抽出した上で、それぞれの読みに対して表3を用いてローマ字表記の候補を求める。また、名は読み方のバリエーションが多く、網羅されているデータベースも存在しないため、今回は姓のみで名寄せすることとし、同姓の著者が二人以上存在するデータは修復の対象外とした。

5.3.2 外 国 人

中国人、韓国人などの著者は、同じ漢字であってもアルファベット、カタカナへ直すルールが異なるため、日本人用の人名辞書では対応できない。そこで、日本人でない可能性の高い名字のリストを作成し、リストに含まれる名字を持つ著者のデータは修復の対象外とした。

5.3.3 組 織 名

大学図書館など組織全体として著者となった文献の場合、姓名が存在しないだけでなく、アルファベット表記も英訳あるいは略称などの特殊な名前となるため、そのままでは名寄せできない。そこで、組織など特殊な名前の辞書を作成した。辞書の例を表4に示す。

5.4 整 列

NC 要素の存在により、著者並び順が変更されている可能性があるため、著者順序の整列を行なう必要がある。はじめに、Creator 要素と Creator.NC 要素の両方が存在するかを確認する。各要素内の並び順はデータ上の並び順と一致しているため、ど

(注1) : <https://github.com/ikegami-yukino/jaconv/>

(注2) : <https://github.com/slr-J/jinmei-dict>

```

<Creator lang="ja">
  佐藤, 一郎
  <AuthorityID.Creator>DAXXXXXXXX</AuthorityID.Creator>
</Creator>
<Creator lang="ja">田中, 二郎</Creator>
<Creator lang="ja">鈴木, 三郎</Creator>
<Creator.Transcription lang="ja">
  サトウ, イチロウ
  <AuthorityID.Creator_Transcription> DAXXXXXXXX </AuthorityID.Creator_Transcription>
</Creator.Transcription>
<Creator.Transcription lang="ja">タナカ, ジロウ </Creator.Transcription>
<Creator.Alternative lang="en">SATO, Ichiro</Creator.Alternative>
<Creator.Alternative lang="en">TANAKA, Jiro</Creator.Alternative>
<Creator.Alternative lang="en">Suzuki, Saburo</Creator.Alternative>

```

図2 機関リポジトリ内部のデータ形式例

```

<Creator lang="ja">田中, 二郎</Creator>
<Creator lang="ja">鈴木, 三郎</Creator>
<Creator.Transcription lang="ja">タナカ, ジロウ </Creator.Transcription>
<Creator.Alternative lang="en">SATO, Ichiro</Creator.Alternative>
<Creator.Alternative lang="en">TANAKA, Jiro</Creator.Alternative>
<Creator.Alternative lang="en">Suzuki, Saburo</Creator.Alternative>
<Creator.NC lang="ja">
  佐藤, 一郎
  <AuthorityID.Creator>DAXXXXXXXX</AuthorityID.Creator>
</Creator.NC>
<Creator.Transcription_NC lang="ja">
  サトウ, イチロウ
  <AuthorityID.Creator_Transcription> DAXXXXXXXX </AuthorityID.Creator_Transcription>
</Creator.Transcription_NC>

```

図3 データ移行後に発生した、著者順序の乱れ

表1 機関リポジトリ納品時の中間形式

id	/Creator#1	/Creator#1@lang	/Creator.Transcription#1	/Creator.Transcription#1@lang	Creator#2	/Creator#2@lang	...
1002450	佐藤, 一郎	ja	サトウ, イチロウ	ja	田中, 二郎	ja	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	

表2 データの記法

記法	意味
N(Creator)	Creator 要素の数
N(Creator.NC)	Creator.NC 要素の数
N(Alternative)	Alternative 要素の数

ちらかの要素のみが存在する場合、整理済みとして扱える。

もし Creator 要素と Creator.NC 要素の両方がある場合、Creator.Alternative 要素の数によって整列の可否が変わる。Creator.Alternative 要素の順序は、Creator 要素と Creator.NC 要素の両方を保持している。さらに、Creator.Alternative 要素には AuthorityID が含まれないため、データ移行の際に

表3 カタカナ-ローマ字変換表

日本語例	ローマ字候補
サト(ウ)	ou / o / oh
(オオ)ノ	oo / o / oh
ナカ(ジ)マ	ji / zi
ナ(ン)バ	n / m
(ツ)ムラ	tsu / tu
ツ(チ)ダ	chi / ti

も並び順が変更されなかったと考えられる。したがって $N(\text{Alternative}) \geq N(\text{Creator}) + N(\text{Creator.NC})$ (全ての Creator と Creator.NC 要素に、対応する Creator.Alternative 要素が存在) が成り立つ場合は、Creator.Alternative 要素の並び順を基準に Creator

```

{"Creator#1": {"name": "佐藤, 一郎", "lang": "ja",
               "AuthorityID.Creator": "DAXXXXXXX"},
 "Creator#2": {"name": "田中, 次郎", "lang": "ja"},
 "Creator.Transcription#1": {"name": "サトウ, イチロウ", "lang": "ja",
                              "AuthorityID.Creator_Transcription": "DAXXXXXXX"},
 "Creator.Transcription#2": {"name": "タナカ, ジロウ", "lang": "ja"},
 "Creator.Alternative#1": {"name": "SATO, Ichiro", "lang": "en"},
 "Creator.Alternative#2": {"name": "TANAKA, Jiro", "lang": "en"}
}

```

図4 名寄せ, 順序修復前の中間形式

表4 組織の名前に対する辞書の例

カタカナ	漢字
チバダイガク フゾク トショカン	千葉大学附属図書館
チバイガッカイ	千葉医学会

表5 キーの短縮ルール

短縮前キー	短縮後キー
/Creator.NC#1	/Creator#1
/Creator#1./Creator.Transcription#1	/Creator#1.Transcription#1
/Creator#1./Creator.Alteranative#1	/Creator#1.Alternative#1

要素と Creator.NC 要素の並び順を修正できると考えられる。

5.5 フラット化

階層化されたデータは, 最終的な納品形式であるフラットなデータ形式に変換する必要がある。フラット化は, 階層化されたデータを再帰的に探索しキーを作成することで実現する。

5.6 キーの短縮

最終的にデータをフラット化した際にキーが長くなりすぎするため, 短縮する必要がある。短縮ルールを表5に示す。

5.7 修復後のデータ構造

全ての修復が終わり, 再度機関リポジトリに登録される際には, 順序の乱れが生じないように, 各要素に通し番号を振って管理する。修復後に JPCOAR に移行した際のメタデータ例を図5に示す。修復段階では, Creator.Alternative や Creator.Transcription 要素が存在するが, JPCOAR スキーマでは, 全て name 要素として管理される。

6. データ分析

5.節で提案したプログラムを用いて, 機関リポジトリのデータを修復した。

6.1 対象データ

対象となるデータは, 以下の二種類である。

- 著者順序が乱れる原因は NC の付いた要素であることから, 全データ中いずれかの要素に NC を持つデータを抽出した。
- 元のデータの時点で Creator.Transcription や Creator.Alternative が Creator および Creator.NC と一対一対応していないデータについても, 順序がずれているので抽出対象

とし, 名寄せの機能を使って修復することとした。

対象としたデータについて, 名寄せと順序の修復ができたもの, 名寄せのみできたもの, 機械的には修復が困難なものに分けて, 分析した。結果を表6に示す。

6.2 アクセスできないデータ

データ抽出時, 機関リポジトリシステムのデータベースには存在するが取り下げたデータやテストデータなど, 機関リポジトリ Web サイト上からはアクセスできないデータも一緒に抽出された。そこで第一段階として各データの ID を用いてアクセス可能かどうかを確認し, アクセスできないデータを除外した。

6.3 不十分な Creator.Alternative 要素

Creator 要素と Creator.NC 要素の両方が存在するデータでは, 正しい並び順がわからなくなっている。修復の手掛かりとなるのは, Creator.Alternative 要素である。

その両方に対応する Creator.Alternative 要素が存在しない場合, 正確な並び順が不明であるため, 修復の対象外とした。

6.4 姓が不明なデータ

姓名の区切り文字が一切存在せず, 「姓名」と表記されるデータが存在した。これは単純なデータ登録時のミスと考えられる。名寄せのためにどこまでが姓か判別できないので, 修復対象外とした。

6.4.1 旧姓表記

また, 名字が不明なデータの中では, 著者の旧姓を「佐藤(旧姓 田中), 一郎」のように記載しているデータが存在した。この表記の場合, 旧姓, 現在の姓のどちらについても単純なルールでは抽出できない。問題の生じたデータについて pdf を確認すると, pdf 上でも同様の名前表記をしていたため, これに倣ったと考えられる。しかし, 検索の際にはこれは著者の別表記として存在するべきであり, データ登録に際して旧姓表記に基準がないことが問題と考えられる。

6.4.2 外部データの不適切なインポート

一部のデータでは, 著者名が存在するはずの Creator.Transcription 要素に, AuthorityID のみが記載されていた。機関リポジトリ上で実際のページを確認したところ, 学外のオープンアクセスジャーナルのページからインポートしたものと判明した。機械的なインポートの際にデータ順番がずれ

図5 JPCOAR スキーマ準拠に修正したメタデータの例

```

<creator>
  <creatorName lang="ja"> 佐藤, 一郎 </creatorName>
  <creatorName lang="ja-kana"> サトウ, イチロウ </creatorName>
  <creatorName lang="en">SATO, Ichiro</creatorName>
  <nameIdentifier>DAXXXXXXXX</nameIdentifier>
</creator>
<creator>
  <creatorName lang="ja">田中, 二郎</creatorName>
  <creatorName lang="ja-kana">タナカ, ジロウ</creatorName>
  <creatorName lang="en">TANAKA, Jiro</creatorName>
</creator>
<creator>
  <creatorName lang="ja">鈴木, 三郎</creatorName>
  <creatorName lang="en">Suzuki, Saburo</creatorName>
</creator>

```

表6 エラー分析結果

データ総合計	アクセス不可	修正成功	Creator.Alternative 不十分	同姓	日本人以外の名前	名字不明	名前なし	名寄せ残り	除外
8611	157	5147	1373	37	266	21	12	1515	8

たと考えられる。

6.5 名寄せしきれなかったデータ

名寄せしきれなかったデータでは、外国人の名前や、名字登録されているが読みのないデータなどがあった。例えば、Smith をカタカナ表記 (スミス) する際には、日本語のローマ字変換ルールが使えないため修復の対象外となる。

6.5.1 姓名の逆転

名前のアルファベット表記は、論文上の表記によらず「姓、名」と表記する決まりになっている。しかし、アルファベット表記の際に逆転している表記が見られた。このうち一部の表記では「Taro, SATO」のように姓を大文字で表記しているものがあり、この場合のみ姓と名の逆転が文字列から判定できるため修復対象とした。

6.5.2 不正確な表記

一部のデータにおいて、前述した姓名の逆転だけでなく、名前をイニシャル表記しているデータが存在した。これは、姓名の逆転と同様、登録時のルール徹底が不十分だったことが原因と考えられる。

6.5.3 Creator 要素と Creator.NC 要素両方のデータ存在

一部のデータにおいて、本来重複しないはずの Creator 要素と Creator.NC 要素が両方存在していた。これは単純な重複登録が原因と考えられるが、同姓同名の可能性もあるため、修復の対象外とした。

7. ま と め

本研究では、大学の機関リポジトリにおいてデータ移行の際に生じた著者並び順のエラーについて分析し、可能なものについて自動で並び順を修復するプログラムを作成した。また、同時にデータ登録時のミスについても分析し、その原因を

考察した。

今後の課題として、(1) 外国人名の扱いは今後が増えると考えられることから、例外処理でない方法を考案する必要性 (2) 人手によるデータ登録が続く限り、エラーは存在するため、エラーを適切に弾き、可能な限り修正する必要性の二つが挙げられる。

文 献

- [1] “学術機関リポジトリデータベースサポート 2024-02-19 閲覧,” <https://support.irdb.nii.ac.jp/ja>.
- [2] “JPCOAR スキーマ項目の説明 | JPCOAR スキーマガイドライン 2024-02-19 閲覧,” <https://schema.irdb.nii.ac.jp/ja/schema>.
- [3] “学術機関リポジトリ構築連携支援事業 — ドキュメント — システム情報 — メタデータ・フォーマット junii2 2024-02-19 閲覧,” <https://www.nii.ac.jp/irp/archive/system/junii2.html>.
- [4] 大園隼彦, 片岡朋子, 高橋菜奈子, 田口忠祐, 林 豊, 南山泰之, “JPCOAR スキーマの策定: 日本の学術成果の円滑な国際的流通を目指して,” 情報管理, vol.60, no.10, pp.719–729, 2018.