

# 多様な記述メタデータの交換・共有スキーマとして BIBFRAME 2.0 は適切か

谷口 祥一 (慶應義塾大学文学部)  
taniguchi@z2.keio.jp

[抄録] BIBFRAME の最新版である 2.0 について、RDA メタデータを超えて、情報資源に対する多様な記述メタデータの交換・共有スキーマとしての観点から、その適切性を検討した。設定した前提条件の下で、特にプロパティの定義域設定に関して BIBFRAME の方式に加えて、他の可能な複数の方式を設定し、それぞれの方式の特徴等を相互に比較した。異なるスキーマからのメタデータの変換容易さと変換されたメタデータ間での照合処理の複雑さなどがいずれの方式でも相反するが、BIBFRAME 2.0 の方式の適切性には疑問が残ることを確認した。

## 1. はじめに

2016 年 4 月に公開された BIBFRAME の最新版である 2.0<sup>1)</sup>は、これまでのバージョン 1.0 からかなり修正された部分も見受けられるとともに、Linked Data の原則に従って RDF を採用するという原則や方向性などを含めて、変更されていない部分も多い。現在、最新バージョンに合わせたツールの開発や実証実験が実施されている段階にあり、最終的な姿は未だ確定していない。

こうした段階にあって、その要求定義・要件定義に沿って、a)図書館目録のメタデータである RDA メタデータを作成し交換・共有するためのスキーマとしての観点、および b)RDA メタデータを超えて、出版流通業界作成のメタデータや利用者作成のメタデータなど、情報資源に対する多様な記述メタデータの受け皿すなわち多様なメタデータからの変換先そして交換・共有スキーマとしての観点という両者から BIBFRAME を検討することが求められている。発表者はこのような観点、特に前者 a)からの検討結果の概要を既に報告しているが<sup>2),3)</sup>、本発表では後者の b)の観点に限定した検討をさらに展開する。

本発表では BIBFRAME を含めて複数の方式の適切性を検討するが、その際の前提条件は下記の通りとする。

- a) 主たるクラスである Work 等の定義、すなわち何を以て Work と見なすか、他のクラスとどのように区別するのか、さらにはその実現値の単位設定はどのようなものかなどについては共通理解を求めない。
- b) 変換前のメタデータとそのスキーマは、対象とする情報資源を階層のないフラットな捉え方をしているものから複数階層で表現しているものまで、多様とする。
- c) 多様なメタデータを単一のスキーマで表現する、すなわち当該スキーマに変換できること、

併せてそこではすべてのプロパティの使用は任意かつ繰り返し可とする。

d) 変換後のメタデータにおいては、照合とそれによる統合を企図する。ただし、その容易さや複雑さの程度には幅があることを前提とする。

## 2. BIBFRAME 2.0 におけるプロパティの定義域設定

BIBFRAME 2.0 では、クラス Work (著作 ; <http://id.loc.gov/ontologies/bibframe/Work>)、Instance (インスタンス)、Item (アイテム ; 個別資料) が定義されている (図 1)。Work は FRBR/RDA の「著作」に、Instance は「体現形」に、Item は「個別資料」にそれぞれ対応するとされている。多様なメタデータの受け皿となることを意図したスキーマにおいて、こうしたコンテンツ (内容) とその物理的実現形という区分を採用することは稀である。たとえば、ダブリンコアにおいては上記のようなクラスの区分はない。その点で BIBFRAME は画期的であるが、それに伴う課題も大きい。

また、BIBFRAME はクラス Agent とそのサブクラス Person、Family、Organization (組織・団体)、Jurisdiction (行政区分)、Meeting (会議) を登録している。先の Work 等のリソースとの関連を表すために、プロパティ contribution (値域はクラス Contribution) が定義され、さらにクラス Contribution はプロパティ role と agent をとるとされている。

Work、Instance、Item の特徴を表すため相当数のプロパティが現時点で登録されている。これらプロパティ設定の粒度と網羅性の問題も当然検討されなければならない事項であるが、本発表ではこの点は取り上げない。

Work、Instance、Item のクラスの区分と、これらクラスに直接関係する個々のプロパティにおける定義域の設定について見てみると、BIBFRAME では大きく 3 つの定義域設定

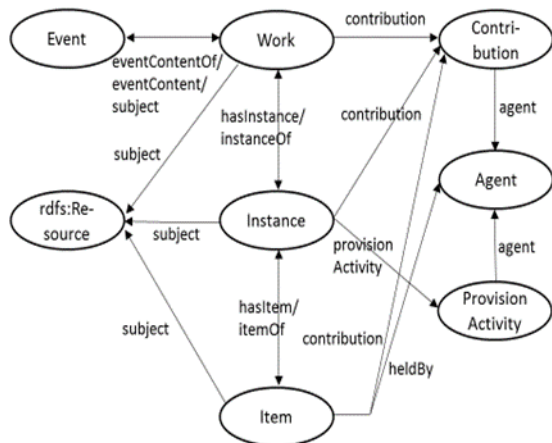


図1 主要なクラスと関連づけプロパティ

タイプに分けることができる。a)いずれか1つのクラスを定義域とする、b)すべてまたは複数のクラスを定義域とする、すなわち指定した、いずれのクラスにも適用可とする、c)定義域を未定義とする、という3タイプである。ただし、BIBFRAMEの付属ドキュメントでは、このような定義域の指定も推奨程度であり、拘束する意図はないと明記されている。

本発表では、上記の定義域設定に注目し、多様なメタデータの交換・共有スキーマとして適切であるかを検討する。なお、検討を単純化するため、これ以降はWorkとInstanceという2クラスのみを対象とし、Itemについては取り上げない。表1の「方式2」としてBIBFRAMEの設定方法をまとめている。

BIBFRAMEでは、originDate(著作の成立日付)、historyOfWork(著作の来歴)などWorkに特有の事項に相当するプロパティ①の定義域はWorkに限定され、同様にresponsibilityStatement(責任表示)やcarrier(キャリアタイプ)などInstanceに特有の事項に相当するプロパティ②および③はいずれもInstanceを定義域とする。

WorkとInstanceに共通する特性と目されるプロパティ④⑤⑦⑧については、その定義域をWorkまたはInstanceとし、いずれにも適用可能としている。

natureOfContent(著作の特性)、intendedAudience(想定利用者)、dissertation(学位論文)、summary(内容要旨)、notation(表記法)、contentAccessibility(内容へのアクセシビリティ)などのプロパティ⑦は、その趣旨からいけば本来はWorkを定義域とすることになるが、BIBFRAMEではWorkと

Instanceの両者を定義域としている。主題を表現するsubjectとclassificationについても同様である。一方、本来はInstanceを定義域とすべきプロパティmedia(メディアタイプ)、frequency(刊行頻度)、custodialHistory(所有者歴)、acquisitionTerms(入手条件)、acquisitionSource(入手源)など⑧も、同様にWorkとInstanceの両者を定義域としている。これらは定義域をWorkとInstanceのいずれかに限定して採用することには、利用者層の多様性の点で困難、あるいは適用の容易さを優先した判断の現れといえよう。

⑨メタレベル情報を示すadminMetadata(メタデータの管理情報)やsource(情報源)、⑩汎用項目としてのdateやplace等は、定義域を未定義とし、WorkやInstanceに限定せず、いかなるクラスのリソースにも適用可能としている。また、⑥identifiedBy(識別子)は、汎用項目の1つと捉え、その定義域を未定義としている。

hasInstanceとInstanceOfというWorkとInstance間以外の、リソース間の関連づけを指示するhasPart(部分をもつ)、partOf(部分である)、accompaniedBy(付属をもつ)、accompanies(付属する)などのプロパティ⑪は、定義域と値域の両者をWorkまたはInstanceとしており、Work間の関連にもInstance間の関連にも適用可能としている。結果的には、WorkとInstance間の関連ともなりうる。

以上のようなプロパティの定義域設定の結果、BIBFRAMEへのメタデータ変換において、a)元のメタデータからWorkとInstanceというリソースへの適切な分割あるいは統合が必要になる。分割後に新設するリソースのURIは、元のリソースがもっていたURIをベースにして新設することになる。加えて、b)titleや汎用項目をいずれのクラスに対応づけるのか決定する必要がある。Workのみを定義域とするプロパティは限定されているため、結果的にはInstanceのみで十分なメタデータとなりうることになる。

また、変換後のBIBFRAMEメタデータに対して、URIの一致によるリソースの同一性判定以外に、WorkとInstanceの両者に対するプロパティ値に基づく照合と統合を試みる際には、上記のいずれかのクラスに限定しない処理が基本的に必要となる。同一プロパティがリソースによってはWorkとInstanceのいずれか一方または両方に出現するからである。

### 3. BIBFRAME 以外の可能な方式群とそれらとの相互比較

最初に設定した条件下で BIBFRAME の適切性を明らかにするため、プロパティの定義域設定に関して BIBFRAME の方式以外の他の可能な複数の方式を設定し、それぞれの方式における特徴等を相互に比較し検討する。ここでは、プロパティの定義域制約が強いものから順に方式 1, 2, 3 とし、方式 5 までを設定した。方式 2 が BIBFRAME 2.0 の方式を指す。なお、いずれの方式においても、`rdf:type` を併用して対象リソースのクラス (`Work`, `Instance`、または新たな定義クラス) を指定することになる。

**方式 1:** 原則として個々のプロパティは `Work` または `Instance` のいずれかのみを定義域とする。これは RDA Vocabulary における RDA プロパティ定義に準拠した方式である。

`Work` と `Instance` に共通する事項⑦と⑧であっても、本来いずれのクラスのリソースに特有かとの点に立脚し、`Work` または `Instance` の一方のプロパティとしている。④`title`、⑤`contribution`、⑥`identifiedBy` については、`Work` を定義域とするプロパティと `Instance` を定義域とするプロパティにそれぞれ分割設定することになる。たとえば、`title` は `workTitle` と `instanceTitle` などに分ける。なお、メタレベル情報に該当するプロパティ⑨については定義域を未定義とし、いずれのクラスのリソースにも適用できるようにする。

こうした方式のスキーマに元のメタデータから変換する際には、BIBFRAME (方式 2) の場合と同様、`Work` と `Instance` のリソースへの分割等が必要となる。`title` 等についてもいずれのクラスのプロパティに対応づけて変換するのかが決定しなければならず、変換仕様策定の負荷が大きい。それに対して、変換されたメタデータでは、`Work` 間の照合と `Instance` 間の照合を区別することができる。制約が厳しい分だけ、変換後のメタデータの品質管理を指向する場合に適しているといえよう。ただし、個々のメタデータがこれら 2 つのクラスを正しく区別できているという保証はない。

**方式 3:** `Work` と `Instance` に加え、共通する部分を抽出し独立させたクラス (仮に `BibRes` と名付ける) を設け、プロパティはこれら 3 つのいずれか、またはそのすべてを定義域とする。`Work` と `Instance` の共通する部分は `BibRes` のプロパティとなるが、`Work` と `Instance` のそれぞれ独自の値をそれらクラスにもたせる

こともできる。`BibRes` から `Work` と `Instance` へのプロパティの継承はない。こうしたプロパティ設定の結果、対象リソースのメタデータにおいて、上記 3 つのクラスではなく、`BibRes` と `Instance` で十分とされる場合もありうる。

`BibRes` の導入によって、変換先のクラスの決定という負荷が減少するが、逆に変換後のメタデータの照合においては 3 つのクラスすべてを対象とした横断的な照合が必要となる。

**方式 4:** `Instance` に代えて、クラス `BibRes` を `Work` と `Instance` を足し合わせたもの、すなわち `Work` のプロパティはすべて `BibRes` に継承されるかたちでクラスを定義する。形式上は `Work` が `BibRes` の上位クラスとなる。

プロパティの定義域は、`Work` と `BibRes` の両方とするものと、`BibRes` のみとするものに分かれる。`Work` に適用できるものはそのまま `BibRes` にも適用する (①、④~⑦) が、その値は `BibRes` において異なるものとなることを妨げない。よって、`BibRes` のみですべてのプロパティを適用することも可能であり、`Work` のリソースを必要としない場合もありうる。

このようなスキーマへの変換仕様の策定は容易であり、変換先のクラスを決定することの負荷が少ない。逆に、変換後のメタデータの照合においては、`BibRes` と `Work` の両者に対する照合が必要となる。同一プロパティが `BibRes` と `Work` のいずれか一方または両方に出現し、かつ両方に出現したときにも同一値とは限らないからである。

**方式 5:** すべてのプロパティの定義域を未定義とし、`rdf:type` を併用して対象リソースのクラス (`Work` または `Instance`) のみを指定する。従って、意味的には適切でないクラスとプロパティの組み合わせも生じうる。OCLC による Schema.org を適用した Linked Data の公開は、基本的にこの方式を採用している。

変換先のクラスを決定することの負荷が最小化されるが、その分多様性を許容する、最も緩やかな方式となり、変換後のメタデータの照合等は `Work` および `Instance` の両者に対して必要となる。

以上の複数の方式の検討結果から、多様なメタデータの交換・共有用スキーマとして BIBFRAME 2.0 の方式が適切かについては疑問が残る。変換後のメタデータの照合や統合等を重視するのであれば方式 1 が適切となり、異なるメタデータからの変換の容易さを重視するならば方式 4 や 5 が適しているといえよう。

なお、元の多様なプロパティを判別可能なよ

うに残して変換後も受け入れる方式もあるが、統一されたプロパティ群での表現とはならず、意図する目的が異なることになる。たとえば、Europeana のメタデータでは、元の詳細かつ多様なプロパティを上位プロパティ edm:hasMet と edm:hasType とでラッピングして受け入れることができるようにしている。

## 注

1) Library of Congress. BIBFRAME Model, Vocabulary, Guidelines, Examples, Notes,

Analyses.

<http://www.loc.gov/bibframe/docs/index.html>

2) 谷口祥一. BIBFRAME 2.0 の意義を問い直す. 2016 年度三田図書館・情報学会研究大会発表論文集. 2016, p.5-8.

[http://www.mslis.jp/am2016yoko/02\\_taniguchi.pdf](http://www.mslis.jp/am2016yoko/02_taniguchi.pdf)

3) Shoichi Taniguchi. Examining BIBFRAME 2.0 from the viewpoint of RDA Metadata Schema. *Cataloging & Classification Quarterly*, 2017, in press.

表 1 プロパティの種別と方式ごとの定義域設定

プロパティの種別	方式 1 (RDA 相当)	方式 2 (BIBFRAME)	方式 3	方式 4
<b>特定クラスに特有の事項</b>				
①originDate, originPlace, historyOfWork, content, musicMedium, instrument など	Work	Work	Work	Work / BibRes
②responsibilityStatement, editionStatement, provisionActivityStatement, seriesStatement, provisionActivity など	Instance	Instance	Instance	BibRes
③carrier, extent, dimensions, digitalCharacteristic, systemRequirement など	Instance	Instance	Instance	BibRes
<b>Work と Instance に共通する事項</b>				
④title	Work を定義域とするプロパティと Instance を定義域とするものに分割設定	Work / Instance	BibRes / Work / Instance	Work / BibRes
⑤contribution (Agent とのリンク)	同上	Work / Instance	BibRes / Work / Instance	Work / BibRes
⑥identifiedBy	同上	(未定義)	BibRes / Work / Instance	Work / BibRes
⑦natureOfContent, intendedAudience, dissertation, summary, capture, notation, awards, review, tableOfContents, genreForm, subject など	Work	Work / Instance	BibRes / Work	Work / BibRes
⑧media, frequency, issuance, copyrightDate, custodialHistory, acquisitionTerms, acquisitionSource, usageAndAccessPolicy など	Instance	Work / Instance	BibRes / Instance	BibRes
<b>汎用項目、メタレベル情報</b>				
⑨adminMetadata, source など	(未定義)	(未定義)	(未定義)	(未定義)
⑩date, place, note, language など	(未定義)	(未定義)	(未定義)	(未定義)
<b>リソース間の関連づけ (Work-Instance 間以外)</b>				
⑪hasPart, partOf, accompaniedBy, accompanies, hasDerivative, derivativeOf など	Work / Instance	Work / Instance	BibRes / Work / Instance	Work / BibRes