

CA1845

動向レビュー

ResourceSync : OAI-PMH の後継規格

はやし ゆたか*
林 豊*

1. はじめに

本稿では、米国情報標準化機構 (NISO) と Open Archives Initiative (OAI) によって策定が進められている ResourceSync⁽¹⁾ の概要を紹介する。

ResourceSync は、2つのウェブサーバの間でコンテンツの同期⁽²⁾を行うためのプロトコルである。2011年に OAI-PMH (CA1513 参照) の後継規格として検討が始まった。ワーキンググループには OAI-PMH の策定に深く関わったロスアラモス国立研究所のソンプル (Herbert van de Sompel) も名を連ねている。一連の規格のなかでコアとなる ResourceSync Framework Specification (ANSI/NISO Z39.99-2014)⁽³⁾ のバージョン 1.0 が 2014 年 4 月にリリースされた。

OAI-PMH は、機関リポジトリなどで使われているメタデータハーベスティング (収集) のためのプロトコルである。その歴史は 1999 年まで遡り、2001 年にバージョン 1.0 が、2002 年に現時点でも最新版となるバージョン 2.0 がリリースされている。図書館ではデジタルコンテンツの露出機会を増やすために外部のサービスにメタデータを提供することがしばしばある。外部との連携方法は一般に様々であり、何らかの標準的な方法が存在すれば連携が容易になる。OAI-PMH は現在でも国際標準規格にはなっていないが、図書館界ではメタデータ収集の方法としてデファクトスタンダードであり、成功していると評価できるだろう。

それではなぜ ResourceSync という新しい規格が必要となったのだろうか。

2. OAI-PMH の限界

その問いに答えるために、まず、OAI-PMH の問題点について説明したい。

2.1. 収集のプロセス

OAI-PMH では、「サービスプロバイダ」が「データプロバイダ」のメタデータを収集するための方法を定めている⁽⁴⁾。サービスプロバイダがデータプロバイダ上の特定の URL にアクセスし、それに対してデータプロバイダが XML 形式でメタデータを返す、というのが基本的なしくみである。

例えば、九州大学の機関リポジトリのメタデータを収集する際の URL は以下ようになる。

http://api.lib.kyushu-u.ac.jp/opac/mmd_api/oai-pmh/?verb=ListRecords&metadataPrefix=junii2&from=2015-01-05&until=2015-01-12

このうち「?」以前の部分はベース URL と呼ばれ、データプロバイダによって異なる。サービスプロバイダは収集したいデータプロバイダのベース URL を何らかの方法で事前に知っておく必要がある。

「?」以後の部分の書式は OAI-PMH の仕様で標準化されている。verb はデータプロバイダへの要求内容を指示するもので、ListRecords (指定した条件に合致するレコードを全て取得する) のほか、Identify、ListMetadataFormats、ListSets、ListIdentifiers、ListRecords、GetRecord の全 6 種類が用意されている。from と until によって、特定の期間に作成・更新されたメタデータのみを抽出できる。

また、応答時の XML では、削除されたメタデータについて status="deleted" という情報が出力される⁽⁵⁾。

2.2. 活用例

例えば、現在 OAI-PMH は次のようなシーンで活用されている。

(1) デジタルリポジトリ

OAI-PMH は、DSpace などの主要なリポジトリソフトウェアで実装されている。機関リポジトリのメタデータを収集するサービスには、国立情報学研究所 (NII) の学術機関リポジトリデータベース (IRDB)⁽⁶⁾、OAIster、RePEc などがある。数物系のプレプリントサーバ arXiv は OAI-PMH によるメタデータ提供を行っている⁽⁷⁾。

(2) デジタルアーカイブ

国立国会図書館 (NDL) 東日本大震災アーカイブ「ひなぎく」、Europeana、米国デジタル公共図書館 (DPLA)、米国議会図書館 (LC) の American Memory などのデジタルアーカイブが、メタデータの収集や提供の方法として OAI-PMH を採用している⁽⁸⁾⁽⁹⁾。特にこのケースでは図書館界以外の機関・企業のサービスも収集対象になることがある点に注意しておきたい。

(3) ディスカバリーサービス

国立国会図書館サーチ (CA1762 参照) では、メタデータの収集・提供の方法として OAI-PMH が採用されている⁽¹⁰⁾。また、ウェブスケールディスカバリーサービス (CA1772 参照) では図書館の OPAC や出版社のデータベースからセントラルインデクスにメタデータを取り込む方法として使われている (E1604 参照)。

2.3. 問題点

OAI-PMH は仕様の簡潔さもあって、図書館界で広く普及してきた。しかし、以下のようにいくつかの課

*九州大学附属図書館 e リソースサービス室

題を抱えていることも事実である。

(1) 収集対象がメタデータのみである

OAI-PMH はメタデータの記述対象であるコンテンツ自身の収集に対応していない。多様なアクセスポイントを提供するだけならば問題ないが、長期保存のためのバックアップや全文検索機能の提供を目的とする場合にはコンテンツ自身の収集が必要となる⁽¹¹⁾。また、アクセス負荷分散のためにミラーサイトを設置したい場合にも OAI-PMH では不十分である⁽¹²⁾⁽¹³⁾。

(2) 図書館界以外では普及していない

例えば、Google は利用の少なさを理由として 2008 年に Google Sitemaps における OAI-PMH のサポートを終了している⁽¹⁴⁾。図書館界に閉じた範囲で収集している分には問題ないが、震災アーカイブのように広く一般のサービスを対象とする場合には課題を感じることもあるだろう。策定から 12 年以上が経過した OAI-PMH は、現在のウェブでは標準的なスタイルといえない。

(3) 収集がリアルタイムではない

OAI-PMH はプル型のしくみであり、収集の主導権はサービスプロバイダにある。データプロバイダ側でコンテンツが生成されたことをサービスプロバイダに通知（プッシュ）する手段を設けていないため、収集までにタイムラグが生じがちである。メタデータだけでなくコンテンツの収集も行うならば、リアルタイム性は重要になるだろう。

3. ResourceSync Framework Specification

このような OAI-PMH の限界を乗り越えるべく、ResourceSync の開発が始まった⁽¹⁵⁾。本節では、そのコアである ResourceSync Framework Specification (バージョン 1.0) の要点を紹介したい。

3.1. 特徴

ResourceSync の同期の対象は、URI を持つあらゆるウェブ上のリソース⁽¹⁶⁾であり、OAI-PMH のようにメタデータには限定されない。

その仕様は、Google などの検索エンジンに対してクローラすべき URL を知らせるための標準的な仕様である Sitemap プロトコル⁽¹⁷⁾をベースにしている。つまり、検索エンジン最適化 (SEO) として既に Sitemap ファイルを公開していれば、そこに多少の変更を加えることで ResourceSync による同期が実現できる⁽¹⁸⁾。

ResourceSync のコアの仕様は、OAI-PMH と同様にプル型である。ただし、関連規格の ResourceSync Notification がコンテンツの生成などを通知するプッシュ型のしくみを定めており、これによって同期のタイムラグを減らすことが可能になる。

ResourceSync は、コンテンツ数が数百万件を超え

る大規模環境や、秒単位で頻繁に更新されるようなコンテンツを想定し、スケーラビリティに配慮した設計を目指している⁽¹⁹⁾。

3.2. 同期のプロセス

次に ResourceSync による同期の基本的なプロセスを説明したい。詳しくは「仕様書」⁽²⁰⁾の §5 を参照いただきたい。

まず、同期の対象となるリソースを持つサーバは「ソース (Source)」、同期を行うシステムは「デステイネーション (Destination)」と呼ばれる。

同期を行うために必要なソースの機能 (capability) として、Resource List、Resource Dump、Change List、Change Dump の 4 種類が定められている⁽²¹⁾。いずれも何らかの XML ファイルを出力するもので、それらのファイルも同様の名称で呼ばれる。例えば、Resource List は、ある時点におけるソースのリソースを全て列挙するものである。Resource Dump は、ある時点におけるリソース全体をパッケージングした ZIP ファイル⁽²²⁾の場所を示すものである。Change List および Change Dump は、それぞれ Resource List および Resource Dump に対応するもので、ある時点からの変化を記述するものである。

デステイネーションの取りうるプロセスには、ベースライン同期 (Baseline Synchronization)、増分同期 (Incremental Synchronization)、監査 (Audit) の 3 種類がある。ベースライン同期は、初回同期時などに行われるもので、Resource List や Resource Dump によって実現される。増分同期は、前回同期時から変化したリソースのみを収集するもので、Change List や Change Dump によって実現される。監査は、同期が正しく行われているかをチェックするもので、ソースが各機能にリソースのハッシュ値などの固定化情報 (fixity) を含めることで可能となる。

実際に同期を開始する前に、ソースはこれらの機能が出力する XML ファイルの URI をデステイネーションに発見 (discovery) してもらう必要がある。その方法として、ソースの robots.txt に記述するなどの 3 種類が定められている⁽²³⁾。

3.3. XML の構文

ResourceSync の XML は Sitemap プロトコルをベースに独自要素を追加したものである⁽²⁴⁾。Resource List、Resource Dump、Change List、Change Dump はそれぞれ Sitemap ファイルとして記述される。

以下、いくつかの記述例を紹介する。説明を簡単にするために <rs:ln> など一部の必須要素を省略している。詳しくは仕様書の §7、§10 ~ §13 を参照いただきたい。

	Baseline Synchronization	Incremental Synchronization	Audit
<ul style="list-style-type: none"> URI Metadata fixity links 	Resource List	Change List	Resource List fixity Change List fixity
<ul style="list-style-type: none"> URI Bitstream Metadata fixity links 	Resource Dump	Change Dump	Resource Dump fixity Change Dump fixity

図：各プロセスで必要となる機能

出典：仕様書 Figure 3

● Resource List

Sitemap プロトコルと同様、<urlset> と <url> を基本構造とする。独自要素 <rs:md> の capability 属性で機能の種類を、at 属性でいつの時点の状態かを示している。リソースの数だけ <url> が繰り返され、<loc> でその URI、<lastmod> で最終更新日時、<rs:md> でハッシュ値やファイル形式を記述している。

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset
  xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:rs="http://www.openarchives.org/rs/terms/"
  <rs:md capability="resourcelist"
    at="2015-01-03T09:00:00Z"/>
  <url>
    <loc>http://ex.com/content.pdf</loc>
    <lastmod>2015-01-02T13:00:00Z</lastmod>
    <rs:md hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      type="application/pdf"/>
  </url>
  <url>
    .....
  </url>
  .....
</urlset>
```

● Resource Dump

<loc> でリソースをまとめたパッケージの URI を示している。独自要素 <rs:ln> ではパッケージに含まれるファイルの一覧 (Resource Dump Manifest と呼ばれる) の所在を表している。パッケージは複数に分割することもでき、必要ならば <url> を繰り返す。

```
<rs:md capability="resourcedump"
  at="2015-01-03T09:00:00Z"/>
<url>
  <loc>http://ex.com/package-1.zip</loc>
  <rs:md type="application/zip" length="4765"
    at="2013-01-03T09:00:00Z"/>
  <rs:ln rel="contents"
    href="http://ex.com/manifest-1.xml"
    type="application/xml"/>
</url>
```

● Change List

<rs:md> の from 属性で、いつの時点からの変化なのかを示している。変化のあったリソースを <url> で列挙し、それぞれの <rs:md> の change 属性で変化の種類 (新規作成・更新・削除) を表している。

```
<rs:md capability="changelist"
  from="2015-01-03T00:00:00Z"/>
<url>
  <loc>http://ex.com/content1.html</loc>
  <lastmod>2015-01-03T11:00:00Z</lastmod>
  <rs:md change="created"/>
</url>
<url>
  <loc>http://ex.com/content2.pdf</loc>
  <lastmod>2015-01-03T13:00:00Z</lastmod>
  <rs:md change="updated"/>
</url>
<url>
  <loc>http://ex.com/content3.tiff</loc>
  <lastmod>2015-01-03T18:00:00Z</lastmod>
  <rs:md change="deleted"/>
</url>
```

なお、Sitemap プロトコルでは、1つの Sitemap ファイルのサイズは5万 URL または 10MB までという制限があり、それを超える場合は、複数の Sitemap ファイルに分割し、別途作成した Sitemap index ファイルからリンクすることになっている⁽²⁵⁾。このような階層構造によって1つの Sitemap index ファイルで最大で 25 億 URL のリソースを記述できる。ResourceSync もこの方法に従っており、Resource List や Change List に対する Sitemap index ファイルを Resource List Index や Change List Index と呼んでいる⁽²⁶⁾。

3.4. メタデータハーベスティング

もちろん ResourceSync はメタデータのみの収集にも対応している⁽²⁷⁾。一般のリソースの同期と同様、Resource List などの機能で実現されるが、コンテンツと同じようにメタデータも URI を持ったひとつのリソースでなければならない点に注意が必要である。コンテンツとメタデータの関係は rel 属性で記述できる。コンテンツに対しては rel="describedby" で、メタデータに対しては rel="describes" で相手との関係を表現できる。

```
<url>
  <loc>http://ex.com/content.pdf</loc>
  <rs:ln rel="describedby"
    href="http://ex.com/metadata.xml">
</url>
```

4. 関連する動向

4.1. 関連規格

現在、ResourceSync Framework Specification の関連規格として、ResourceSync Archives⁽²⁸⁾、Relation Types used in the ResourceSync Framework⁽²⁹⁾、ResourceSync Notification⁽³⁰⁾ がドラフト段階にある。このうち ResourceSync Archives と ResourceSync Notification はコアの拡張という位置づけである。

ResourceSync Archives は、ソースから過去の時点におけるリソースのスナップショットを取得するためのしくみで、Resource Dump Archive、Change List Archive、Change Dump Archive という3つの機能について定めている。この規格はソンベルらのウェブアーカイビングプロジェクト Memento (CA1733、E1573 参照) と関わりが深い。

ResourceSync Notification は、ソースにおけるリソースの生成などをデスクトップに対して通知するプッシュ型のしくみについて定めている。一般にウェブでプッシュ通知を実現する方法はいくつかあるが、ここでは PubSubHubbub⁽³¹⁾ が採用されている。

4.2. 実装実験

管見のかぎりでは今のところ ResourceSync を実運用として実装したサービスは見られないが、いくつかの実験は行われてきている。

例えば、コーネル大学図書館のワーナー (Simeon Warner)⁽³²⁾ によって arXiv のデータを用いたシミュレーションが行われ、Resource List や Change List が公開されている。

ロスアラモス国立研究所のクライン (Martin Klein)

ら⁽³³⁾ は、更新の激しいソースとして Wikipedia の Linked Data 版サイトである DBpedia Live⁽³⁴⁾ を取り上げ、プッシュ通知の実験を行っている。

また、ソンベルは、プッシュ通知を使ってリソース同期を行うデモ動画を公開している⁽³⁵⁾。

4.3. ツール

ResourceSync の普及のためには、DSpace のような代表的ソフトウェアで実装されることが有効である。例えば Cottage Labs は、ResourceSync でメタデータハーベスティングを行うための Java ライブラリと DSpace モジュールを公開している⁽³⁶⁾。現在はメタデータに留まり、Resource Dump や Change Dump によるコンテンツの同期は今後の課題となっている。

また、resync という、Python で開発されたクライアント (デスクトップ側) とライブラリ (ソース側、デスクトップ側) が公開されている⁽³⁷⁾。合わせて動作する ResourceSync Simulator というソース側のシミュレータも公開されている⁽³⁸⁾。

ResourceSync PuSH という、Python による ResourceSync Notification の実装も存在する⁽³⁹⁾。

5. おわりに

以上、メタデータハーベスティングの規格として成功した OAI-PMH にもいくつかの課題があること、そして、それを乗り越えるために登場した標準規格 ResourceSync の概要について紹介してきた。

コア以外の仕様も固まりつつある ResourceSync だが、実装は現時点では実験段階に留まり、今後実際に広く活用されていくかは未知数という印象である。しかしながら、電子コンテンツの長期的な保存など、リソース同期の標準的な方法が必要となる場面が存在することは疑いがないだろう。図書館界を超えて広く採用されるかどうか、注目したい。

(※参照 URL の最終確認日はすべて 2015 年 2 月 13 日)

- (1) ResourceSync には関連規格も含めて現在 4 つの仕様が存在する。本稿ではそれらの総称として "ResourceSync" という表現を用いる。
"ResourceSync Framework Specification - Table of Contents". NISO. 2014-07-02.
<http://www.openarchives.org/rs/toc>.
- (2) 原語を尊重して「同期」と表現するが、ResourceSync は OAI-PMH と同様に双方向的な同期を実現するものではない。
- (3) "ResourceSync Framework Specification (ANSI/NISO Z39.99-2014)". NISO. 2014-04-21.
<http://www.openarchives.org/rs/resourcesync>.
- (4) 正確にはサービスプロバイダのハーベスタが、データプロバイダのリポジトリからメタデータを収集する。
- (5) 仕様上は "may" とされ、出力は必須ではない。
- (6) 国立情報学研究所. "IRDB ハーベスタについて". 学術機関リポジトリ構築連携支援事業. 2015-01-23.
http://www.nii.ac.jp/irp/archive/system/irdb_harvest.html.
- (7) "Open Archives Initiative (OAI)". ArXiv.
<http://arxiv.org/help/oa/index>.

- (8) “5 国立国会図書館東日本大震災アーカイブとの連携について”. 国立国会図書館東日本大震災アーカイブ.
<http://kn.ndl.go.jp/static/renkei>.
 “Technical requirements”. Europeana.
<http://pro.europeana.eu/technical-requirements>.
 “DPLA Participation Instructions”. DigitalNC.
<http://www.digitalnc.org/about/dpla/dpla-instructions/>.
 “OAI-harvestable records for digitized historical collections”. Library of Congress. 2008-05-23.
<http://memory.loc.gov/ammem/oamh/index.html>.
- (9) 国内公共図書館でも例がある。
 “大阪府立図書館 Web-API 情報”. 大阪府立図書館. 2014-02-21.
<http://www.library.pref.osaka.jp/site/e-service/webapi.html>.
- (10) “国立国会図書館サーチが提供する OAI-PMH”. 国立国会図書館サーチ.
http://iss.ndl.go.jp/information/api/oai-pmh_info/.
- (11) Van de Sompel, Herbert. et al. Resource Harvesting within the OAI-PMH Framework. D-Lib Magazine. 2004, 10 (12).
<http://www.dlib.org/dlib/december04/vandesompel/12vandesompel.html>.
- (12) arXiv は 1990 年代から独自の方法でミラーリングを行っている。
 Van de Sompel, Herbert. ResourceSync: A Web-Based Resource Synchronization Framework. 2013-06-19.
<http://www.slideshare.net/hvdsomp/resourcesync-tutorial-oai8>.
- (13) OAI-PMH を用いてコンテンツを同期させる方法が 2004 年に提案されている。後年、普及には至らなかったと振り返られている。
 Van de Sompel, Herbert. et al. Resource Harvesting within the OAI-PMH Framework. D-Lib Magazine. 2004, 10 (12).
<http://www.dlib.org/dlib/december04/vandesompel/12vandesompel.html>.
 Van de Sompel, Herbert. et al. A Perspective on Resource Synchronization. D-Lib Magazine. 2012, 18 (9-10).
<http://www.dlib.org/dlib/september12/vandesompel/09vandesompel.html>.
- (14) Mueller, John. "Retiring support for OAI-PMH in Sitemaps". Google Webmaster Central Blog. 2008-04-23.
<http://googlewebmastercentral.blogspot.jp/2008/04/retiring-support-for-oai-pmh-in.html>.
- (15) Van de Sompel, Herbert. et al. A Perspective on Resource Synchronization. D-Lib Magazine. 2012, 18 (9-10).
<http://www.dlib.org/dlib/september12/vandesompel/09vandesompel.html>.
- (16) Van de Sompel, Herbert. "ResourceSync: A Web-Based Resource Synchronization Framework". 2014-01-24.
<http://www.slideshare.net/OpenArchivesInitiative/resourcesync-tutorial>.
- (17) "What are Sitemaps?" Sitemaps.org.
<http://www.sitemaps.org/>.
 なお、Sitemap の最初のバージョンである 0.84 が登場したのは 2005 年であり、OAI-PMH の策定後である。
 Shivakumar, Shiva. "Webmaster-friendly". Official Google Blog. 2005-06-25.
<http://googleblog.blogspot.jp/2005/06/webmaster-friendly.html>.
- (18) 以下の論文は、Sitemap プロトコルを拡張することの懸念を 3 つ挙げ、実際に Google に処理させて問題が起こらないかを実験している。
 Klein, Martin. et al. "Extending Sitemaps for ResourceSync". 2013-05-21.
<http://arxiv.org/abs/1305.4890>.
- (19) Van de Sompel, Herbert. "ResourceSync: A Web-Based Resource Synchronization Framework". p. 10-11. 2013-06-19.
<http://www.slideshare.net/hvdsomp/resourcesync-tutorial-oai8>.
- (20) "ResourceSync Framework Specification (ANSI/NISO Z39.99-2014)". NISO. 2014-04-21.
<http://www.openarchives.org/rs/resourcesync>.
- (21) ResourceSync の仕様はモジュラー (modular) な設計になっており、例えばベースライン同期を行うだけであれば、Change List や Change Dump の実装は必要ない。
 Van de Sompel, Herbert. "ResourceSync A Quick Overview". 2014-09-30.
<http://www.slideshare.net/hvdsomp/resourcesync-quick-overview>.
- (22) 仕様書では ZIP 形式が使われているが、他の形式でも良いとされている。
- (23) 仕様書 § 6.3 を参照。デスティネーションがソースの機能を発見する方法には、Source Description、Capability List、Resource List など、の 3 種類がある。
- (24) 記述方法として Sitemap、Atom、独自仕様を検討し、最終的に Sitemap が採用された。§ 4 に理由が述べられている。
 Klein, Martin. et al. Technical Framework for Resource Synchronization. D-Lib Magazine. 2013, 19 (1-2).
<http://www.dlib.org/dlib/january13/klein/01klein.html>.
- (25) "Frequently asked questions". Sitemaps.org. 2008-02-27.
http://www.sitemaps.org/faq.html#faq_sitemap_size.
- (26) 仕様書 Figure 4 を参照。
- (27) 仕様書 § 14.5 を参照。他にも § 14 ではリソース間の関係を記述する方法が定められており、OAI-PMH のセット (set) や、OAI-ORE (CA1690 参照) の集合体 (aggregation) についても言及されている。
- (28) Klein, Martin. et al. "ResourceSync Framework Specification - Archives - Beta Draft". 2013-08-21.
<http://www.openarchives.org/rs/0.9.1/archives>.
- (29) Klein, Martin. et al. "Relation Types used in the ResourceSync Framework - Beta Draft". NISO. 2013-09-27.
<http://www.openarchives.org/rs/0.9.1/relationtypes>.
- (30) Klein, Martin. et al. "ResourceSync Framework Specification - Notification - Beta Draft". NISO. 2014-03-24.
<http://www.openarchives.org/rs/notification/0.9/notification>.
- (31) Fitzpatrick, Brad. et al. "PubSubHubbub Core 0.4 - Working Draft".
<https://pubsubhubbub.googlecode.com/git/pubsubhubbub-core-0.4.html>.
- (32) "ResourceSync experiments on resync.library.cornell.edu".
<http://resync.library.cornell.edu/>.
- (33) Klein, Martin. et al. "Real-Time Notification for Resource Synchronization". 2014-02-11.
<http://arxiv.org/abs/1402.3305>.
- (34) "DBpedia Live".
<http://live.dbpedia.org/>.
 "DBpedia Live". DBpedia. 2015-01-08.
<http://wiki.dbpedia.org/DBpediaLive>.
- (35) "ResourceSync Change Notification Demo". 2014-04-07.
https://www.youtube.com/watch?v=H2Le9_Bbkdw.
- (36) "ResourceSync Module for DSpace". Cottage labs.
<http://cottagelabs.com/news/resourcesync-module-for-dspace>.
 "ResourceSync". Cottage labs.
<http://cottagelabs.com/projects/resourcesync>.
- (37) "resync 1.0.0". Python.org.
<https://pypi.python.org/pypi/resync/1.0.0>.
- (38) "resync/resync-simulator". GitHub.
<https://github.com/resync/resync-simulator>.
- (39) "resync/resourcesync_push". GitHub.
https://github.com/resync/resourcesync_push.

[受理 : 2015-02-13]

Hayashi Yutaka.
 ResourceSync: A Successor of OAI-PMH.



本著作 (CA1845) はクリエイティブ・コモンズ 表示 2.1 日本 ライセンスの下に提供されています。ライセンスの内容を知りたい方は
<http://creativecommons.org/licenses/by/2.1/jp/>
 でご確認ください。