

RDF Schema に基づくメタデータレジストリ

永森光晴, Thomas Baker[†], 阪口哲男, 杉本重雄, 田畑孝一

図書館情報大学,
ドイツ国立情報処理研究所[†]

Email: {nagamori, saka, sugimoto, tabata}@ulis.ac.jp, thomas.baker@gmd.de[†]

概要

インターネットの普及と共に、ネットワーク上には様々な分野のコミュニティが形成されている。それらコミュニティでは、扱う情報資源に合わせたメタデータの記述方法を決めている。メタデータの記述形式はメタデータスキーマによって定義される。異なるメタデータスキーマが存在する場合、メタデータスキーマの違いを越えてメタデータの流通性、相互利用性、長期利用性を高める必要がある。

我々は、メタデータの流通性、相互利用性、長期保存性という観点からメタデータスキーマを登録し、利用者がより多くの情報資源を発見することを助けるメタデータレジストリを提案する。本稿では、メタデータレジストリの役割と機能について述べる。また本稿では、メタデータレジストリのプロトタイプとして RDF Schema を利用した Dublin Core のためのメタデータレジストリを作成した。

A Metadata Registry Based on RDF Schema

Mitsuharu Nagamori, Thomas Baker[†], Tetsuo Sakaguchi,
Shigeo Sugimoto, Koichi Tabata

University of Library and Information Science,
GMD German National Research Center for Information Technology[†]

Email: {nagamori, saka, sugimoto, tabata}@ulis.ac.jp, thomas.baker@gmd.de[†]

Abstract

Metadata registries are formal systems that can disclose authoritative information about the semantics and structure of the data elements. Registries would typically define the semantics of metadata elements, give information on any local extensions, and provide mappings to other metadata schemas. This paper discusses basic requirements for metadata schema registries, and proposes a multilingual metadata registry based on the Dublin Core Metadata Element Set. The registry uses the Resource Description Framework(RDF) Schema which is an emerging standard for expressing metadata schemas.

1 はじめに

インターネットの普及と共に、ネットワーク上にはデジタル図書館、博物館、学会など様々な分野のコミュニティが形成されている。コミュニティの大きさは、国際的な広がりを持つものから、地理的・分野的に小さなものまでである。また、そこで使われている言語には、英語をはじめ日本語、独語、仏語など様々なものがある。現在では、これらコミュニティにおいて、それぞれの目的に合わせたメタデータの記述方法を定義するメタデータスキーマが作成されている。例えば Dublin Core は、ネットワーク上の情報資源のためのメタデータスキーマで、Dublin Core Metadata Initiative (DCMI) によって基本的な 15 の要素 (element)¹ が定義されている [1]。IMS では、学習に関する情報資源のメタデータスキーマを IEEE LOM (Learning Object Metadata) に基づいて定義している [3][4]。

複数のメタデータスキーマが存在する場合、異なるメタデータスキーマ間 (例えば Dublin Core と IMS) で横断検索をおこなうことができれば、利用者はより多くの情報資源を発見することができる。

また、さまざまなメタデータスキーマを蓄積し参照することができれば、メタデータレコード作成やコミュニティが新たにメタデータスキーマを作成する際に有用であると考えられる。更にメタデータスキーマを蓄積・管理することにより、メタデータレコードの長期的利用が可能になる。

メタデータレジストリは、このようなメタデータの相互利用性 (interoperability) や長期利用性 (longevity) を高めるために多種多様なメタデータスキーマを蓄積・管理し、利用者やアプリケーションがメタデータスキーマを参照することや、各要素間の関連を知ることを目的としたシステムである。

本稿では、メタデータレジストリの目的とそ

¹Dublin Core の 15 要素: Title, Creator, Subject, Description, Publisher, Contributor, Date, type, Format, Identifier, Source, Language, Relation, Coverage, Rights

の機能と、Resource Description Framework(RDF) Schema を用いた Dublin Core のためのメタデータレジストリについて述べる。

2 メタデータレジストリ

2.1 メタデータレジストリの機能

メタデータスキーマでは、コミュニティの扱う情報資源に合わせた要素を決め、各要素を記述するために必要な値を定義する。例えば Dublin Core では、Subject 要素を図 1 のように定義している。現在では、Dublin Core や IEEE LOM 以外に、BIBLINK [5], GILS [6], MARC21 [7] などコミュニティの目的に合わせて様々なメタデータスキーマが定義されている。また、DCMI では Dublin Core Qualifier として、Dublin Core の各要素に詳細な役割を与えた要素 (element refinement) や各要素で用いる値のフォーマットや語彙 (encoding scheme) を決めている [2]。

メタデータスキーマは、例えば以下のように利用されることが考えられる。

(1) 参照

メタデータレコード作成者がレコード作成の際に、メタデータスキーマにどのような要素があり、各要素にどのような値を記述すればよいか情報を得る。

Name:	Subject and Keywords
Identifier:	Subject
Definition:	The topic of the content of the resource.
Comment:	Typically, a Subject will be expressed as keywords, key phrases or classification codes that describe a topic of the resource. Recommended best practice is to select a value from a controlled vocabulary or formal classification scheme.

図 1 Dublin Core の Subject 要素の定義

(2) 検索

メタデータスキーマ作成者、メタデータレコード作成者、メタデータデータベース利用者などが既存のメタデータスキーマを調べ、同じ目的を持つものがあれば、そのメタデータスキーマをそのまま利用したり、必要な部分だけを利用する。

(3) Crosswalk

異なるメタデータスキーマに含まれる要素について、同様の役割を持つ要素間の関連を作成しておき、メタデータのデータベースの横断検索などに利用する。

以上のことから、利用者やアプリケーションの利便性を考え、様々なメタデータスキーマを登録するレジストリが必要であると考えられる。このレジストリをメタデータレジストリと呼ぶ。メタデータレジストリは、単にメタデータスキーマを蓄積するだけのリポジトリではなく、上に述べたメタデータスキーマの(1)参照、(2)検索、(3)Crosswalkの機能を持つ必要がある。更に(4)長期に渡ってメタデータスキーマを管理(登録・変更・削除)する機能が必要である。

現在 DCMI では、レジストリ working group においてメタデータレジストリに関する議論をおこなっている。UKOLN では、EU の DESIRE II プロジェクトの一環として、メタデータレジストリの開発が行われている。ISO 11179 は、メタデータレジストリを作成するための基準として、メタデータスキーマの管理方法を決めている。

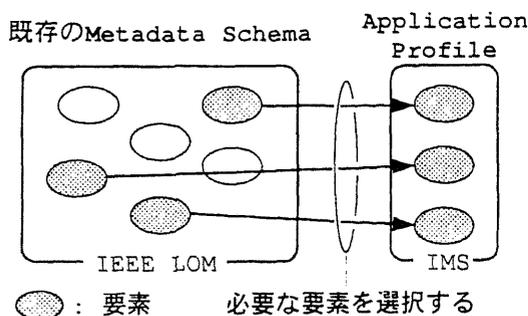


図2 Application Profile の概念

2.2 メタデータレジストリに登録される情報

メタデータレジストリの基本的な役割は、登録されているメタデータスキーマを利用者やアプリケーションに提供することである。メタデータレジストリは単にメタデータスキーマの辞書ではなく、前節で挙げたメタデータスキーマの参照、検索、Crosswalk のサポート、管理などの機能が必要である。これら機能を実現するために、メタデータレジストリには以下の情報を登録する。

(1) メタデータスキーマの参照記述 (Reference Description)

参照記述は、メタデータスキーマの各要素の定義を記述したものである。参照記述は、少なくとも、Title や Subject といったメタデータを構成する要素を言語によらずユニークに識別する ID と人間が読むことができる要素名、要素の役割の解説から構成される。必要であれば、メタデータレコードに記述する値のフォーマットや語彙を決めたり、他の要素との関連を定義する。例えば、サブエレメントを定義する場合、親エレメントの ID を記述してエレメント間の関連を定義する。図1は、Dublin Core の Subject 要素の定義である。Dublin Core では参照記述として、Name, Identifier, Definition, Comment を記述している。

(2) Application Profile

Application Profile は、既存の1つまたは複数のメタデータスキーマの中から、コミュニティや研究プロジェクトなどの目的に合う要素を選択した集合である(図2参照)。例えばIMSは、IEEE LOM というメタデータスキーマから学習に関するメタデータレコードを記述するために必要な要素を選択した Application Profile である。

(3) Crosswalk と dumb-down のためのマッピング

Crosswalk と dumb-down は、横断検索でメタデータの相互利用性を高めたり、version が異

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:my="http://www.dl.ulis.ac.jp/myrdf#">
<rdf:Description ID="Subject">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:label>Subject and Keywords</rdf:label>
  <rdfs:comment>The topic of the content of the resource.</rdfs:comment>
  <my:comment>Typically, a Subject will be expressed as keywords, key
    phrases or classification codes that describe a topic of the resource.
    Recommended best practice is to select a value from a controlled
    vocabulary or formal classification scheme.</my:comment>
</rdf:Description>
</rdf:RDF>

```

図 3 RDF Schema を用いた Subject 要素の参照記述

なるメタデータスキーマの間でメタデータレコードの変換をおこなうことによりメタデータの長期利用性を高めるための重要な機能である。

Crosswalk を実現するために、異なるメタデータスキーマに含まれる要素について、同様の役割を持つ要素間の関連を記述し蓄積する。例えば、「Dublin Core と IEEE LOM の Title 要素は互換性がある」や「Dublin Core に IEEE LOM の Educational 要素に対応する要素は存在しない」といったマッピングを記述する。

dumb-down は、より詳細な役割を与えられたメタデータの要素を一般的な要素に変換することである。Dublin Core の場合、dumb-down は qualifier 付きの要素を qualifier の無い Simple Dublin Core の要素に変換することである。例えば、Dublin Core Qualifier の中では、Dublin Core の Date 要素に対して、Created, Valid, Available, Issued, Modified という element refinement qualifier を定義している [2]。以下は、qualifier を取り除くだけの最も簡単な dumb-down の例である。

qualifier 付きの要素:

```

<Date>
  <Created>2000-5-1</Created>
</Date>

```

dumb-down 後:

```

<Date>2000-5-1</Date>

```

(4) ユーザガイド

メタデータレジストリの利用者のために、新たにメタデータスキーマを作成する方法、登録されているメタデータスキーマを用いて記述したメタデータレコードのサンプル、目的のメタデータスキーマを探すための補助など様々なユーザガイドを蓄積する。

3 RDF Schema による Dublin Core 参照記述

前節までにメタデータスキーマの役割やメタデータレジストリの機能について述べてきた。ここでは、レジストリに登録するメタデータスキーマの参照記述の記述方法を Dublin Core と Dublin Core Qualifier を例に述べる。

レジストリに登録する情報やメタデータレコードの記述は、文書構造を明確にし機械的に処理できる必要がある。本稿では、XML を利用してメタデータを記述するための枠組である RDF (Resource Description Framework) [8] を Dublin Core メタデータレコードの記述に用いた。また、Dublin Core のメタデータスキーマの記述には、RDF のスキーマを定義するための言語である RDF Schema [9] を用いた。

図 3 は、RDF Schema を利用した Dublin Core の Subject 要素の参照定義である。図 1 の属性名と図 3 に示した XML のタグは、Name と label, Identifier と ID, Definition と rdfs:comment,

```

<rdf:Property rdf:ID="created">
  <rdfs:label>Created</rdfs:label>
  <rdfs:comment>Date of creation of the resource.</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/elements/1.1#date"/>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/dc/documents/dcmes-qualifiers"/>
</rdf:Property>

```

図 4 Element Refinement の例: RDFS を用いた Created Qualifier の定義

```

<rdf:Description rdf:ID="http://purl.org/dc/elements/1.1#subject">
  <rdfs:range rdf:resource="SubjectScheme"/>
</rdf:Description>
<my:DataType rdf:ID="SubjectScheme">
  <rdfs:label>Subject Encoding Schemes</rdfs:label>
  <rdfs:comment>A set of subject encoding schemes and/or formats</rdfs:comment>
</my:DataType>
<my:DataType rdf:ID="LCSH">
  <rdfs:label>LCSH</rdfs:label>
  <rdfs:comment>Library of Congress Subject Headings</rdfs:comment>
  <rdfs:subClassOf rdf:resource="SubjectScheme"/>
</my:DataType>
<my:DataType rdf:ID="MESH">
  <rdfs:label>MeSH</rdfs:label>
  <rdfs:comment>Medical Subject Headings</rdfs:comment>
  <rdfs:subClassOf rdf:resource="SubjectScheme"/>
</my:DataType>

```

図 5 Encoding Scheme の例: RDFS を用いた Subject に関する scheme の定義

Comment と my:comment がそれぞれ対応している。

Dublin Core では qualifier として, element refinement と element encoding scheme の 2 種類を定義している。図 4 と図 5 は, Dublin Core Qualifier の中の, Subject 要素と Date 要素に関連する qualifier の参照記述の一部である。

element refinement では, 要素を適用する範囲に制限を加えたり, 詳細な意味を与えた要素を定義する。図 4 では, Date 要素を refine して, 情報資源が作成された日付を問わず Created 要素を定義している。

encoding scheme では, 要素に記述する値のフォーマットや語彙などを定義する。図 5 では, Subject 要素を記述する値に用いることができる scheme として, LCSH と MeSH を定義している。図 6 は RDF を用いたメタデータレコードの記述例である。

4 Dublin Core 多言語レジストリ

我々は, Dublin Core 参照記述を登録したメタデータレジストリのプロトタイプを作成した [11]。このレジストリには, 14 種類の言語²で記述した Dublin Core 参照記述を登録してある。本稿では, このメタデータレジストリを DCML (Dublin Core in Multiple Languages) レジストリと呼ぶ。

利用者は, Web ブラウザを介して DCML レジストリに登録されている参照記述を閲覧することができる。タイ語やギリシャ語など, ブラウザ側にフォントが無い場合でも MHTML アプレット [10] を利用することで, 文字化けすることなく参照記述を表示することができる。

まず利用者は図 7 や図 8 に示したインターフェー

²14 種類の言語: 英語, フィンランド語, 仏語, 独語, ギリシャ語, ノルウェー語, ポルトガル語, スペイン語, 中国語 (Chinese, Taiwanese), インドネシア語, 韓国語, タイ語, 日本語

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcq="http://purl.org/dc/documents/dcmes-qualifiers"
  xmlns:my="http://www.dl.ulis.ac.jp/myrdf#">
  <rdf:Description about="http://www.dl.ulis.ac.jp/DCML/">
    <dc:title>Dublin Core in Multiple Language</dc:title>
    <dc:creator>Shigeo Sugimoto</dc:creator>
    <dc:subject>a prototype RDF registry</dc:subject>
    <dcq:created my:scheme="W3CDTF">2000-08-19</dcq:created>
  </rdf:Description>
</rdf:RDF>

```

図 6 RDF を用いた Dublin Core メタデータレコードの記述例

スを使い、それぞれの言語で記述された Dublin Core の要素を選択する。図 7 は列に言語、行に要素を持つマトリクスである。利用者は閲覧したい言語で書かれた要素に当てはまるマスを一つ一つ選択する。図 8 は、言語と要素をそれぞれ箇条書きにしたものである。図 8 では、図 7 のように閲覧する要素をマトリクスの中から一つ一つ選択するのではなく、利用者が選択した言語と要素に当てはまる定義を検索して表示する。例えば図 8 は、英語、ノルウェー語、タイ語で書かれた Creator 要素と Subject 要素の定義を選択し、表示させる例である。利用者が選択した結果は図 9 のように表示される。図 9 は、Dublin Core の Title 要素の定義を英語、日本語、タイ語で表示した例である。

5 今後のメタデータレジストリの課題

メタデータスキーマは、メタデータの流通性、相互利用性、長期利用性のために標準 (standard) として定義されるものである。したがって、メタデータレジストリの基本的役割は標準を誰もが見られるようにすることである。ところが現実には、

- (1) コミュニティの目的や分野に応じた要素や qualifier の出現、
- (2) 地域や言語、文化に応じた要素や qualifier の出現、

(3) 時間的経過による新版の出現

などに対応していかなければならない。

今後メタデータレジストリでは、これらメタデータスキーマの違いを越えて、メタデータの相互利用性や長期的利用性を高めるために、以下の機能が必要であると考えられる。

- qualifier から Simple Dublin Core への dumb-down を実現するためのモデル
- 異なる要素や qualifier 間で crosswalk を実現するためのモデル
- メタデータスキーマのバージョン管理のモデル
- 言語によらずメタデータを表現するためのモデル

6 おわりに

本稿では、メタデータスキーマの役割、メタデータレジストリの機能、RDF Schema を利用した Dublin Core の参照記述の記述方法について述べた。また、参照記述を検索して閲覧することができるメタデータレジストリを作成した。

今後は、前節で述べたようなコミュニティによるメタデータスキーマの違いを表現するためのデータモデルを作成し、メタデータスキーマの登録、修正、バージョン管理、crosswalk の機能を持つメタデータレジストリを実現する。

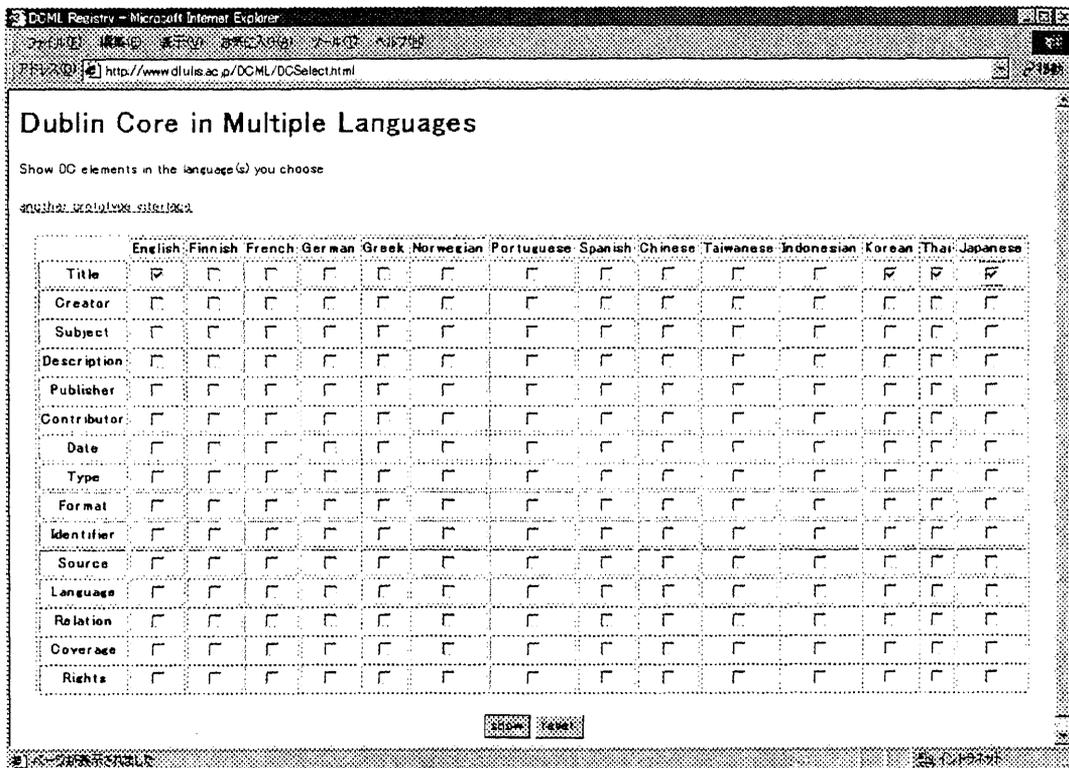


図 7 選択画面 (1)

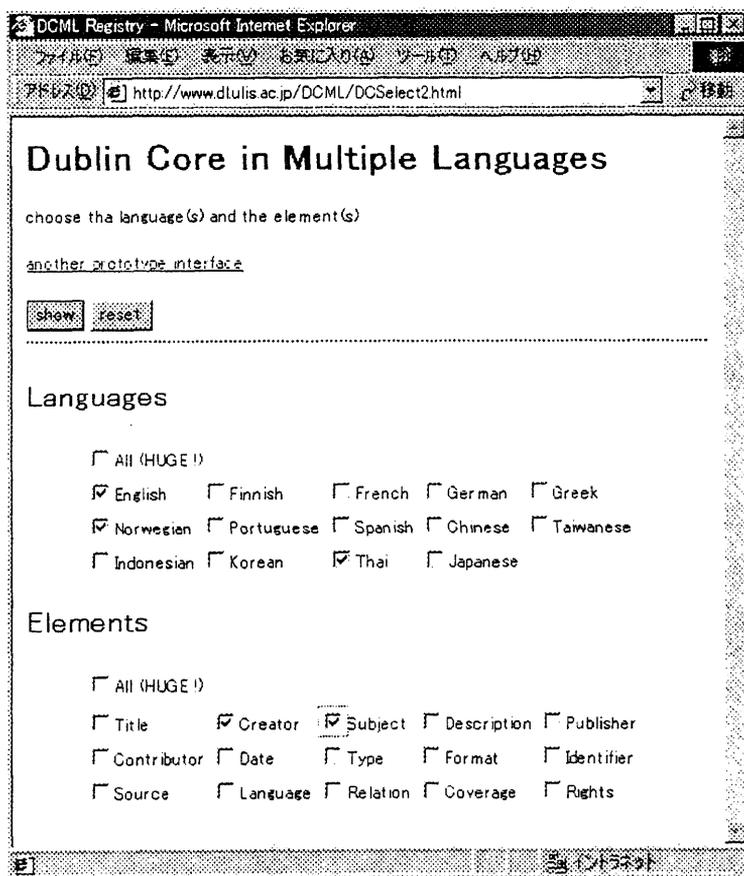


図 8 選択画面 (2)

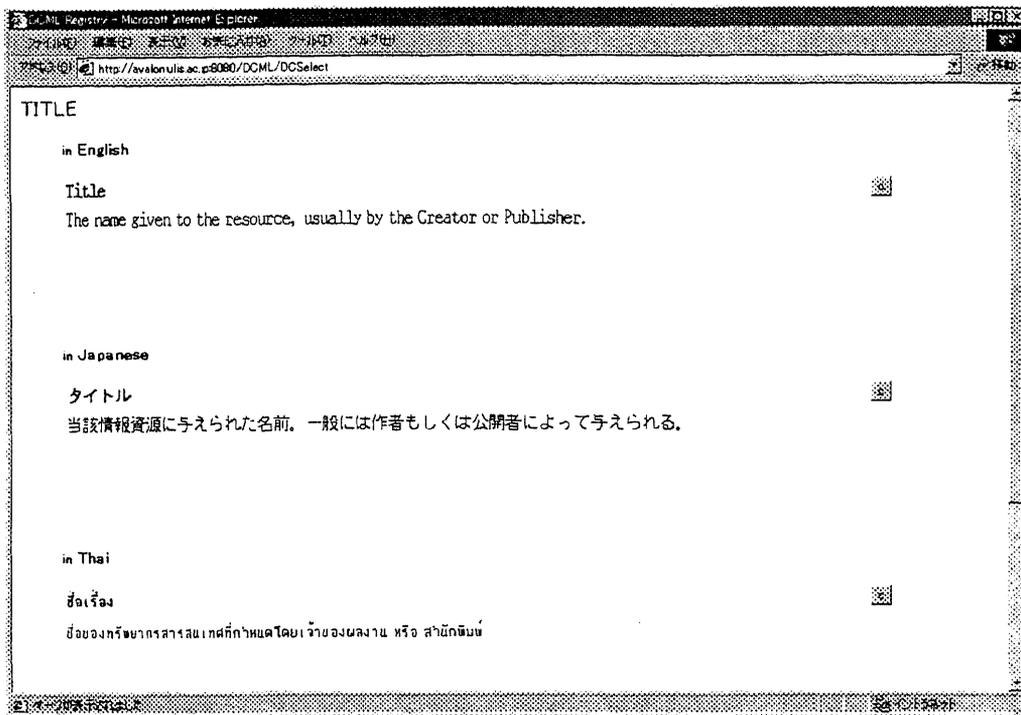


図 9 エレメントの選択結果

参考文献

- [1] Dublin Core Metadata Initiative,
<http://purl.org/dc/>
- [2] Dublin Core Qualifiers,
<http://purl.org/DC/documents/dcmes-qualifiers>
- [3] IMS Meta-data Working Group,
<http://www.imsproject.org/metadatateam.html>
- [4] IEEE P1484.12 Learning Object Meta-data Working Group,
<http://ltsc.ieee.org/wg12/index.html>
- [5] BIBLINK.
<http://hosted.ukoln.ac.uk/biblink/>
- [6] Global Information Locator Service (GILS),
<http://www.gils.net/>
- [7] MARC21 Concise Format for Bibliographic Data,
<http://lcweb.loc.gov/marc/bibliographic/ecbdhome.html>
- [8] Resource Description Framework,
<http://www.w3.org/RDF/>
- [9] RDF Schema,
<http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>
- [10] Multilingual HTML,
<http://mhtml.ulis.ac.jp/>
- [11] Dublin Core 多言語レジストリ,
<http://www.dl.ulis.ac.jp/DC/>