

Dublin Core の最近の話題から

杉本重雄

筑波大学・図書館情報学系
知的コミュニティ基盤研究センター
sugimoto@slis.tsukuba.ac.jp

概要

ここでは、Dublin Core の開発に関連して 2001 年 10 月に東京で開催した国際会議 DC-2001¹以後この 1 年くらいの間の話題について述べる、具体的には、2001 年 10 月にイタリアのフィレンツェで開催された DC-2002 に関する話題、Usage Board による新しい限定子の認定に関する話題、筆者も開発にかかわっているメタデータスキーマレジストリに関する話題、および Dublin Core Metadata Initiative (DCMI) に関して、筆者にとって印象に残っていることについて述べる。

1. はじめに

2001 年 10 月に東京で開催した国際会議 DC-2001²は Dublin Core ワークショップの歴史の中でも特筆すべきものであったと思う。第 1 の理由は、Dublin Core の開発に直接かかわる議論を行うワークショップに加えて、研究や開発成果に関する論文発表を主体とする会議と Dublin Core に関して学びたい人のためのチュートリアルを加えた本格的な会議としたことである。加えて、初めてのアジア地域での開催であったことも理由のひとつである。

今年フィレンツェで開催された DC-2002 はフィレンツェのイタリア国立図書館他いくつかの組織によって主催された。DC-2002 の参加者は 200 名強で、日本からは 9 名の参加であった。DC-2002 では、DC-2001 と同様に投稿論文の中から採択された論文発表のセッション（口頭発表 22 件、ポスター発表 16 件）、ワークショップとチュートリアルのセッションがもたれた。基調講演はロスアラモス研究所の Van de Sompel 氏であった。このほか、WWW コンソーシアムの Semantic Web のリーダーである Eric Miller 氏による講演、ヨーロッパの国立図書館による共同の取り組みである TEL に関する報告などがあった。ワークショップでは、いくつかのテーマに分かれたセッションが持たれたが、政府行政情報や Semantic Web に関するセッションなどが注目を集めていたようである。

Dublin Core は草の根参加者によって開発されてきたメタデータ規則であり、開発組織である DCMI は誰に対しても開かれた組織の形態をとってきた。しかしながら、長期にわたる Dublin Core の維持管理を安定的に行えるようにするために、実質的には数年前から DCMI の再構成に関する議論が続けられてきた。現在、DCMI の現在の組織は、Directorate を中心に、Board of Trustees、Advisory Board、Usage Board、および Working Group と Interest Group からなっている。これまで DCMI は、DCMI Directorate のメンバーや WWW サーバを提供する OCLC を最大のスポンサーとして支えられてきたが、今後長期にわたって DCMI が活動を続けるには組織として収入を得る仕組みを作り出すことが求められており、それに関する議論が進められている。

2001 年頃以降の DCMI の大きな組織の変更は、少人数による Usage Board が立ち上げられたことのほかに、Board of Trustees (BoT) が構成されることになったことであろう。BoT は DC-2001 の直後には最初の小さなミーテ

¹ International Conference on Dublin Core and Metadata Applications 2001

² 会議の主催者は直接ホストした国立情報学研究所に加えて、国立国会図書館、日本科学技術事業団(JST)、通信総合研究所のほか図書館情報大学、DCMI である。

イングが持たれ、その後2002年4月に現在のメンバーによる最初の会議が開かれ、DC-2002で第2回目の会議が開かれた。筆者もメンバーの一人としてこの会議に参加しているが、Board of Trusteesの一番の役割は、今後の長期にわたるDCMIを支える仕組みを考えていくことである。

Dublin Core Metadata Element Setの開発は1995年に始められ、15エレメントからなるSimple Dublin Core (Simple DC)と限定子を含むQualified Dublin Core (Qualified DC)が開発されてきている。Simple DCは固定したエレメントセットであるのに対し、Qualified DCには新たなエレメントや限定子が加えられてきており、今後も加えられていく。Simple DCはこれまでアメリカ(NISO)、ヨーロッパ(CEN/ISSS)での標準化が終わり、現在ISOのFast Trackによる投票が行われている段階である。Simple DCがDublin Coreの根とすると、Qualified DCは根から上に伸びる幹や枝の部分であると言える。Dublin Coreを利用するコミュニティは、必要に応じて新しいエレメントや限定子を使うことになる。一方、Dublin Coreと呼ばれるメタデータ規則の下でのメタデータの相互利用性(Interoperability)を保つためには無秩序な拡張は許されない。Usage Boardの役割は、応用毎のワーキンググループから出される新しいエレメントや限定子の提案を議論し、特定のコミュニティだけでなくグローバルなコミュニティにとって有用であると判断したものを新たなエレメント、限定子として認定することである。DC-2002の前後に、CreatorエレメントがContributorエレメントを詳細化したものであると再定義することに関する議論が行われた。現時点ではこの再定義は行われないうであるが、1998年の第6回Dublin Coreワークショップで議論されたCreator、Contributor、PublisherをAgentエレメントにまとめるという議論以来の大きな話題であると思われる。

以下では、エレメントや限定子に関する話題、メタデータスキーマレジストリの役割、DCMIの組織としての今後に関して述べたい。

2. Dublin Coreの基本的な考え方について エレメント、限定子、その他の語彙

基本的にDublin Coreはメタデータ規則に用いる語彙を決めるだけで、構文的な規則は決めていない。敢えて言うとSimple DCは「すべてのエレメントが繰り返し可能、かつ省略可能」という構文規則的なルールを持っているが、これ以外はエレメントや限定子の名前とその意味を決めているのみである。また、XMLやHTMLでの表記方法に関する推奨を決めているが、基本的に表現方法は利用者に任されている。

Dublin Coreが決める語彙の種類は大別してエレメント、限定子および情報資源(リソースと呼ぶ)のタイプを表す統制語彙であるDCMI Type Vocabularyである。厳密に言うとこれ以外にもエレメントや限定子の定義のために用いられる少数の用語があるが、ここでは無視することにする。Dublin Coreで表現するメタデータは、大雑把に言って「属性と属性値の対の組」である。エレメントはリソースが持つ属性を表現するためのものである。限定子には2種類あり、エレメントの意味を詳細化するもの(エレメント詳細化限定子・element refinement qualifier)とエレメントの値を記述する祭の語彙の名前あるいは記述形式の名前を表すもの(コード化形式限定子・encoding scheme qualifier)がある。もともと、エレメント詳細化限定子は属性をより詳細に限定する修飾語として導入されたものであり、コード化形式限定子は属性値の種類(クラス)を限定するために導入されたものである。限定子導入の原則として、「限定子を加えた記述から限定子を取り除いたとしても、その記述が矛盾してはならない」というDumb-down原則がある。たとえば、日付エレメントのエレメント詳細化限定子である「Created」をつけた記述「Created Date: 2002-11-17」から「Created」を取り除いて「Date: 2002-11-17」としても矛盾しない。コード化形式限定子の場合も同様である。この原則は、限定子を「形容詞」、エレメントを「名詞」として捉えるというDublin Core Grammar[1]につながっている。

Resource Description Framework (RDF)スキーマに基づくエレメントや限定子の形式的定義が進むに連れ、エレメントや限定子の定義がより明確になる一方、「限定子」の概念との食い違いも生じてきている。タイトルエレメン

トのエレメント詳細化限定子である alternative を例にとって説明してみよう。この限定子は「alternative title」を意味するように導入された。一方、すべてのエレメントは RDF Schema の中では Property(属性)として定義された。エレメント詳細化限定子は、RDF Schema に基づく定義の際に、ある属性の下位の属性を意味する sub-property を用いて表されることとなったため、それ自身が属性となり、エレメントと同じ性質のものであると定義されることになった(あるいは「なってしまった」)。「alternative」の場合、「alternative-title」という意味の詳細化された一つのエレメント(=属性)を意味することになった。この定義は、上に示した「限定子は形容詞、エレメントは名詞」というモデルと、ある意味で矛盾することになった。この観点からは、エレメント詳細化限定子ということばはあまり適切ではなく、「詳細化されたエレメント(Refined Element)」と呼ぶほうが適切なのかもしれず、それに関する議論も進められているようである。コード化形式限定子のほうにはこうした議論はないが、エレメント詳細化限定子のほうが、「限定子」でなくなれば、敢えて限定子ということばを使う必要がなくなるのかもしれない。

次に、メタデータ記述の枠組みを少し整理してみたい。メタデータの記述を枠組みを整理すると次のような要素からできていると筆者は理解している。

- (1) メタデータ記述のための語彙定義：エレメントや限定子の名前と意味の定義
- (2) Application Profile あるいは特定の表現形式に依存しない構文定義(抽象構文定義)：エレメントごとの必須・省略可能性の区別、繰り返し回数の定義、エレメントの値(属性値)の型あるいは形式の定義
- (3) 具体的な表現形式：RDF、XML 上での表現形式、HTML 上での表現形式など。データベースのスキーマであってもかまわないが、その場合はデータ交換のための表現形式の定義も定義する必要がある。

この中で Dublin Core が直接決めているものは(1)のみであることが理解できる。(Simple DC の「全てのエレメントは省略可能かつ繰り返し可能」という規則は、(2)に当てはまる。(3)に関して、XML、HTML、RDF での記述形式は「推奨形式」を与えている。)

1998 年の第 6 回ワークショップでは Creator、Contributor、Publisher をまとめて Agent と呼ぶエレメントにしてはどうかという提案があった。Library Application Profile[2]では Contributor と Creator の利用方法について言及している。筆者は Library Application Profile では Contributor の方を優先して利用しているように理解している。最近、MARC Relator コードを Dublin Core に導入するという議論から、Creator を Contributor の詳細化エレメントとする、すなわち Creator は Contributor の sub-property であると定義するという提案がなされた。DC-2002 直後の Advisory Committee の場では Usage Board から Creator を Contributor の sub-property とするように Usage Board は判断したとの報告があったが、それに関する議論はまだ続いており、筆者の理解では現時点ではまだ結論は出ていない。メタデータの語彙を定めるモデルの議論からはこの結論は望ましいかもしれないが、Simple DC では 15 エレメントの間に関係を決めておらず、かつ Simple DC が Dublin Core として広く受け入れられているということから、Creator を Contributor の sub-property とすることが現時点で認められる可能性は少ないようである。次の段落で、Creator と Contributor に関する議論に関連して、筆者の理解する点をまとめてみたい。

(1) Simple Dublin Core と Qualified Dublin Core

これまで Qualified Dublin Core の説明として、Simple Dublin Core を基礎にして限定子を加えたものとするものが多かった。しかしながら、現在開発が進められている Dublin Core、すなわち Qualified Dublin Core と呼ばれるもの、あるいは DC1.1[3]以降に定義された語彙に基づく Dublin Core には、16 番目のエレメントとしての Audience があり、Simple DC を基礎とした限定子の拡張だけではない。したがって、現時点で開発が進められている Dublin Core を Qualified DC と呼ぶことは必ずしも正確ではないといえよう。Simple DC の 15 エレメントを核とした、エレメントと限定子の両面での拡張がなされ続けており、この進化を続けるメタデータ語彙セットのことを Dublin

Core Metadata Element Set と呼べばよいのであろう³。

(2) Contributor と Creator の間の関係に関連して

Contributor と Creator の意味の違いが必ずしも明確ではないという指摘は以前からあった。また、Simple DC の中には Source と Relation エlement のように意味的な包含関係が明らかにあるものもある。Simple DC はリソースの発見のために有用な 15 の属性を集めたもので、属性間の関係は視野の外であると理解すればよいのであろう。Simple DC の定義を進めている段階では、意味的な包含関係に関する議論はなされているが、メタデータのモデルに関する議論は十分に進んでいたとは言えず、15 エlement 間の関係は Qualified DC に関する議論とともにより明らかになってきたと言ってよい。筆者は、Interoperability が Dublin Core の最も基本的な目標であると理解しており、その観点からは Dumb-Down 原則は非常に重要な概念である。Dumb-Down をどのレベルまで行うかは応用に任される問題であり、Dublin Core にとっては属性間の関係をできるだけ正確に決めておくことが求められると思われる。その意味では、Creator を Contributor の下位属性とし、Creator 以外にも詳細化された Element を定義する道筋（あるいは他で定義された属性を詳細化された Element として用いる道筋）を作ることは重要であると、筆者には思える。Creator - Contributor に限らず他の Element についても同様なことがいえる。

3. メタデータ・スキーマ・レジストリについて

従来、メタデータ規則、たとえば目録規則を参照するには印刷物が頼りであった。しかしながら、現在では、ネット上で提供されるものの方が、作業環境がネットワークにつながっていれば手っ取り早く利用できる。また、それ以上に重要なことは、メタデータを相互に利用するためにソフトウェアがメタデータ規則を参照する必要が生じることがある。このようにメタデータ規則を人間とソフトウェア（機械）の両方に理解可能な形で提供することが必要である。そのためには、メタデータの規則を登録するための標準的な形式を定義し、それに基づくサービスを実現することが求められる。この役割を担うことが期待されているものにメタデータ・スキーマ・レジストリ (Metadata Schema Registry) がある。インターネット上では多様なメタデータ規則が用いられる。Dublin Core の場合、もともと Element を拡張して利用することが前提とされているので⁴、応用毎に定義した要素の定義を登録し、ネットワーク上で参照できるようにできれば有用である。特に、ソフトウェアによる処理が可能な形式で登録することで、メタデータの収集や横断検索の機能の実現が行いやすくなると考えられる。DCMI では以前から DCMI で定義した語彙に含まれる語の定義を書いた参照記述を提供するレジストリの開発を進めている。このレジストリでは語の定義を RDF Schema 形式で記述することで機械にも理解できる（処理できる）形式で提供することを進めている。

Open Archives Initiative (OAI) では複数の学術リソースのリポジトリ間⁵で共通に検索を行えるようにし、付加価値を高める取り組みを進めている。ここでは、リソースの書誌情報、すなわちメタデータをリポジトリから収集 (Harvest) するためのプロトコルが開発されている。OAI の場合は、各リポジトリが収集用に Simple DC に基づきメタデータ (XML で表される) を提供している。一方、メタデータ規則の違いを越えて横断的なメタデータの利用を進めるには、メタデータの Element 間の意味的な関係を形式的に、すなわち計算機処理可能な形式で蓄積しておくことが重要である。この観点からもレジストリが重要な役割を果たすと考えられる。

³ とは言っても、ここでは以降でもこれまでの慣習でこのメタデータ語彙セットを Qualified DC、あるいは単に Dublin Core と呼ぶ。

⁴ Dumb-Down 原則に反しない範囲で応用ごとの限定子を定義することや、他のメタデータ規則の Element と組み合わせて利用すること。

⁵ プレプリントやテクニカルレポート、学位論文などの学術情報資源を蓄積する提供するサービス。たとえば、ロスアラモス研究所の e-Print Archive やヴァージニア工科大学の NDLTD など、さまざまなものがある。

4. 組織としての DCMI について

はじめに述べたように DCMI は組織の再構成を進めようとしている。再構成を必要としている理由は長期にわたる Dublin Core の維持のための組織としての自立である。Dublin Core は草の根参加者によって開発されてきたメタデータ規則である。その一方、OCLC が大きなスポンサーの役割を果たしてきた。OCLC は DCMI の中心的な役割を果たしている Directorate を支えてきたほか、会議の主催、会議参加者の補助など Dublin Core の開発に対して大きな貢献をしてきた。長期に渡って Dublin Core を維持していくためには、特定の組織による負担にのみ大きく依存することはできないため、組織の形態の再構成を進めようとしている。

組織の再構成のもうひとつの背景として、Dublin Core の利用の広がりに対応するための地域ごとの組織を必要としている点がある。もともと Dublin Core は地域や応用分野による違いを前提としており、地域や分野の必要性に基づく新しい要素（エレメントや限定子など）の提案をグローバルなコミュニティに結び付けていくためのモデルを持っている。また、Dublin Core のエレメントや限定子の参照記述やいろいろな文書の翻訳は地域ごとの取り組みとならざるを得ない。たとえば、エレメントの参照記述は 20 以上の言語に翻訳されているが、現在のところそのほとんどがボランティアによるものである。これをより信頼度の高いものにするには、国ごとの標準化といった時間のかかる作業とは別に、翻訳に責任をもつ組織による「正式な翻訳」を提供することが求められている。たとえば、前節のメタデータ・スキーマ・レジストリを地域に対してサービスするといったことはその一例である。

5. おわりに

ここでは最近の Dublin Core に関する最近の話題から筆者の関心を持つ話題について述べた。Dublin Core の応用は図書館分野以外にも政府行政情報や教育学習情報、環境情報などの分野で広く利用されている。国内の状況を見た場合、たとえば、「これが正式な日本語訳」というものがまだ無い。また、日本国内での応用を進めるに当たってどんな拡張が必要になるのかといった議論をする場も十分ではない。Dublin Core のエレメント定義や利用に関する正確な情報を提供する WWW サイト、議論や意見交換のための組織作りが強く求められている。

参考資料

- [1] Baker, T., "A Grammar of Dublin Core", D-Lib Magazine, Vol.6, No.10, 2000.10,
<http://www.dlib.org/dlib/october00/baker/10baker.html>
- [2] Guenther, R., "Library Application Profile", 2002.9,
<http://dublincore.org/documents/2002/09/24/library-application-profile/>
- [3] "Dublin Core Metadata Element Set, Version 1.1: Reference Description", 1999.7,
<http://dublincore.org/documents/1999/07/02/dces/>

その他の参考資料

- ・ 杉本重雄, "Dublin Core について (第1回、第2回)", 情報管理, Vol.45, No.4, pp.241-254, No.5, pp.321-335
- ・ "Dublin Core に関する情報のページ", <http://avalon.ulis.ac.jp/DublinCore/> (実験的に作成中のページ)

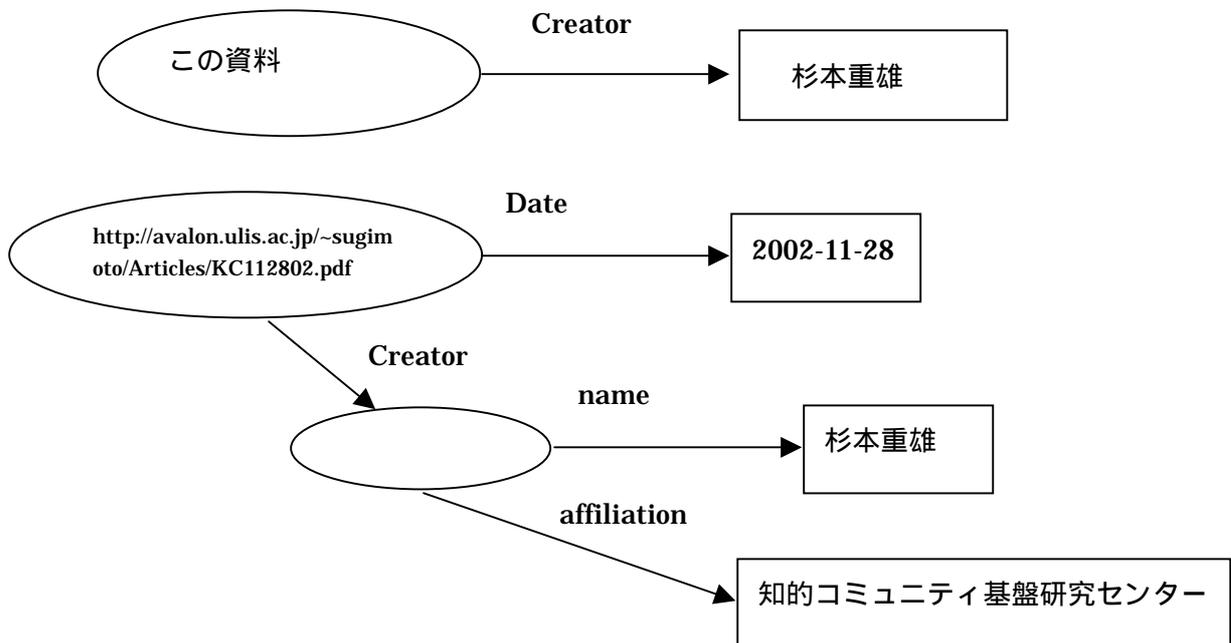
参考 : Resource Description Framework(RDF)の基本モデル

RDF の基本モデル



3 つ組み (triple) で表す。 { 資源 , 属性 , 値 }

値はなんらかのオブジェクト(すなわち資源もしくはリテラル)。資源は楕円 , リテラルは長方形で表す。



RDF Schema

RDF Schema は RDF 記述に利用する語 (たとえば、Creator や Date) の意味を定義する

Property: リソースが持つ属性。

上位・下位関係 = Property - Sub-property 関係

Creator や Date は Property として定義される。

Class: オブジェクトのの種類を表す。

上位・下位の関係 = Class - Subclass 関係

Encoding Scheme (LCSH や URI) は Class として定義される。

RDF は XML で表現する。(詳細は <http://www.w3.org/RDF/> を参照。セマンティックウェブ (<http://www.w3.org/2001/sw/>) も参照すると良い。)