

# WWW10 調査報告書(別紙)

2001年5月21日

セマンティック WEB タスクフォース  
清水委員長(NEC)、袴田委員(富士通)

2001年5月1日(火)

## 1) TF4 Metadata

### (1) 説明者

Stuart Weibel (Director of the Dublin Core Metadata Initiative)

### (2) 要点

#### ・ウェブの状況

検索システムは、機能よりビジネスモデルが、先行している。

インデックスのカバレッジは、予測できず、限られている。

同じものが多く、精度が低い。

インデックススパムに犯されている。

そのリソースは、変転が、激しい。

バージョン、エディション、バック問題は、どうなるのか？

アーカイビング(記録保管庫)にできない。

信用できるサービスやサービス品質の高いものは一部である。

IPRの管理が、難しい。

#### ・Metadata とは

データに対する構造化データである。

- カオスを解消する手助けになる。

- 自動発見操作を可能にする。

色々な局面を変えうる。

- スペシャリゼーション(特化)

- ディセントラリゼーション(非集中化)

- デモクラティゼーション(平等化)

・インターネットは、色々なコミュニティを包含している。

・インターオペラビリティのためには次の規則が必要である。

意味(セマンティクス)

- 構成要素の意味

構造(ストラクチャー)

- ヒューマンリーダーダブル
- マシンパーサブル

構文(シンタックス)

- 意味と構造とを運搬するための文法

- Metadata を用いた成果はないのか？

MARC STANDARDS がある。

一連の MARC 標準は、世界で唯一の最も成功しているリソース記述標準である。

- MARC カタロギングとは

MARC-AACR2 は、本当にカタロギングできるのか？

- MARC は、通信形式(ストラクチャーとシンタックス)である。

- AACR2(Anglo-American Cataloging Rules)は、カタロギング規則を規定している。

- このモデルをウェブに適用するのは、何故まずいのか？

高価である。

- 複雑すぎる
- カタログ作業のプロが、必要である。

図書目録に限定されている。

- 変化のないリソース
- リソースの増加とリソース間の関連付けの操作が、未完成である。

Anglo-centric である。

- MARC21 は、MARC レコードの 3/4 を考慮しているが、沢山あるその他の物について考慮していない。

- Metadata は色々なことに適用できる。

リソースディスカバリー

文書管理

権利管理

内容格付け

セキュリティと監査

記録状況

プロダクトとサービス

データベーススキーマ

プロセス制御もしくは記述

- Metadata において挑戦すべき事項

色々なメタデータとの共存

テンション：機能と単純性

テンション：拡張性と相互互換性

人間と機械での生成と利用

コミュニティ特有の機能、生成、管理、アクセス手段

- Warwick Framework: モジュール Metadata

Warwick Metadata Workshop(DC-2)で導入された Metadata の概念アーキテクチャ。

モジュール Metadata の仕様、収集、エンコーディング及び交換を可能にする為の概念アーキテクチャ

Metadata efforts(Dublin Core を含む)の為に裏方作業を行う。

- 包括的要素集合のブラックホールを回避する。

- パッケージレベルでの相互互換問題に焦点を当てる。

- モジュール化により、分散管理を実現する。

熟練者(ソフトウェアベンダの事ではない)は、次に責任をもつこと、

- セマンティクスの定義

- 管理

- アクセス管理

- データの権威付け

- 共有と分配

Dublin Core に付いて

- Dublin Core の歴史

1994 年：ウェブ記述の為の簡単なタグを開発

1995 年：Dublin Core は、必要とされた沢山の語彙 (“Warwick Framework”) の一つ。

1996 年：テキストとイメージの記述の為、13 要素を 15 要素に拡張

1997 年：WF は、リソースディスクリプションフレームワーク (RDF)の為、形式記述を必要とする。

2000 年：Dublin Core Metadata Initiative は、修飾子の推奨を行い、その組織の視野を core 以外にも拡張

- Dublin Core Workshop Series

- US DC1,3.6

- UK DC2

- Australia DC4

- Finland DC5
- Germany DC7
- Canada DC8

- Tokyo(2001) 10月22日～26日

・デジタルツーリストのピジン言語

Metadata は言語である。

Dublin Core は、諸々の境界を越えてリソースを見つける為の単語の数が少して簡単な言語 - ピジン言語 -

異なる言葉話す人は、自然に、コミュニケーションにピジン化された言葉を使う。

我々は、全員、インターネット上のツーリストである。

・Dublin Core の文法

<http://www.dlib.org/dlib/october00/baker/10baker.html>

各人の母国語のように難解なものではなく、覚えるのが簡単で、実用的になるよう設計されている。

ピジン言語：少ない語彙（Dublin Core：15 の特別な名詞とたくさんの形容詞）

単純な文法：文(ステートメント)は、単純かつ固定の形式

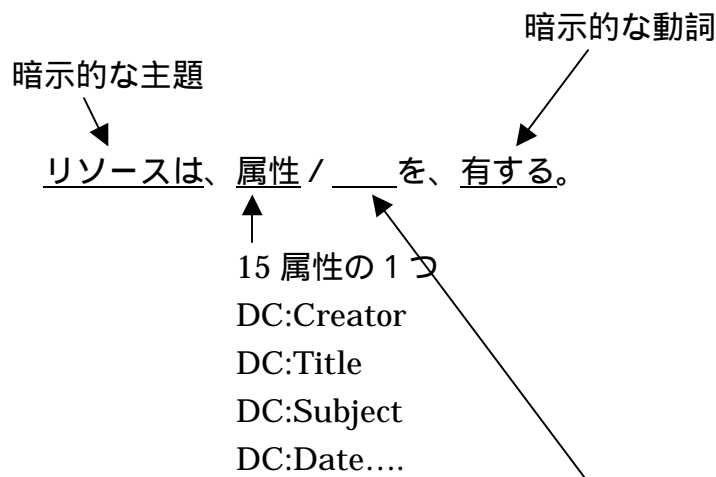
・Dublin Corer ステートメントの例

リソースは'Dublin Core の文法'という表題を有する。

リソースは'Tom Baker'によって作成された。

リソースは'Metadata'という主題を有する。

リソースは'http://foo.org/file.htm'と関連している。



## 属性値(適切なリテラル)

- 15 の特別名詞(属性)

Creator	Title	Subject
Contributor	Date	Description
Publisher	Type	Format
Coverage	Rights	Relation
Source	Language	Identifier

- 修飾詞の色々：要素の洗練化

要素の意味をより絞り、特化する。

- "作成日"に対し、"変更日"

- "ある関連により置換される"に対し、"関連を置換する"

もしも、処理ソフトが、修飾詞を理解できないなら、其れを無視可能である。

- 修飾詞の色々：値の符号化スキーマ

値は、次のようだという。

- コントロールされた語彙か用語である。(例：議会における議題の見出しライブラリ)

- 標準的な方法で形成された文字列(例："2001-05-02"は 5 月 2 日を意味し、2 月 5 日ではない)

もしも、あるスキーマが、ソフトウェアに認知されないとしても、その値は、適切なものであるべきであり、そうならば、リソース検索に有用となる。

- 修飾詞の"サルでも分かる原則"

15 の要素は、修飾詞の有無に拘わらず、理解でき、且つ、有用である。

修飾詞は、意味を明確にする。(しかし、理解が難しいものであってはならない。)

名詞は修飾語句なしで、意味をなさなくてはならない。

もしも、処理ソフトが、不明な修飾詞に出会ったら、その意味を捜すか、又は、単純に無視すべし。

- 新たな用語のプロポーザルの平等なレビュー

DCMI は、新たな DCMI 用語(修飾詞および要素)のプロポーザルのレビューをボードで行う。

文法原則を考慮してプロポーザルを評価する。(修飾詞は無視可能か否か?)

(仮)承認状態の結合モデル：提案、確認、推奨、廃棄  
最初の修飾詞は、2000年の7月に推奨された。

<http://pur.org/DC/documents/rec/dcmes-qualifiers-20000711.htm>

・ Dublin Core に対する公開質問

15 属性の適切な値とは何ですか？それらは、どのように境界を超越した検索に使えるのですか？

DCMI は、どのように Dublin Core 実用化をするのですか？

より複雑な Metadata でリソース記述をするとき、AP は、ピジン言語である DC をどのように使う事ができるのですか？

・ 検索桶対記述

曖昧検索桶として DC 要素を想定した場合、

- 異なる桶の為には、それにそれぞれに適切な異なるデータの型が必要：URL、日付文字列、用語文字列、名前
- 三島由紀夫について書いた本と、三島由紀夫が書いた本の分離

検索桶：リソース検索のため

しかし、一般に、曖昧分類は、リソース記述に充分でない。

- 検索後、画面上に更に詳細な説明の表示

・ DCMI は、そのミッションを拡大(2000年10月)

DCMI のミッションは、次の活動を通じて、インターネットを用いたリソース検索を簡単にする。

- 境界を超越した検索の為の Metadata 標準を開発する。(例：Dublin Core)
- Metadata 集合の互換性のため枠組みを定義する。
- コミュニティの発展を促す、または、規範となる特別な Metadata 集合の開発を促す。

・ DCMI の組織の見直し

ミッションの拡大

- エージェント(もしくは、事象)の為の中核 Metadata 要素
- 多くの領域固有の Metadata を統合するための枠組み

組織見直しモデル

- W3C または Unicode コンソーシアムのような組織の会員
- 合意ドリブンモデルの維持
- 国際的視野
- より良い教育、文書化、普及活動

・ DCMI 活動

標準の開発と保守

Metadata 保管と基盤

技術作業グループと定期的なワークショップ

教育資料と利用者の手引き

教育と訓練

オープンソースソフトウェア

他の標準または、利用者コミュニティとの連携

• Dublin Core の公的認知

CEN ワークショップ合意

- Dublin Core の要素を CWA13874 として保証
- 欧州業界に利用ガイドラインとして提供

NISO Z39.85

- National Information Standards Organization  
ANSI 会員
- 現在投票中

• DC-2001

東京で DC-2001 を開催

- 10月22日～26日
- 3本のトラックがある。
- 技術作業グループ会議
- 実装報告と研究発表
- 一般の人に対する紹介と講習

• Dublin Core の背景

もしも、Dublin Core が、DCMI の中核ならば、それ以外のものとは何か？

もしも、Dublin Core が単純なピジン言語ならば、Metadata 言語の広い展望とは何か？

ピジン言語がどのように、より複雑なモデルや”アプリケーションプロファイル”に関連するのか？

• 他の語彙と一緒に DC を利用する場合

特定アプリケーションプロファイル(政府情報、教育、数学)は、以下のものを必要とするかも知れない。

- Dublin Core 要素の General-purpose 利用
- より領域固有の標準で作られた要素の利用
- 特定の局所的利用の為に DC 要素の狭い標準定義
- 既存標準の範囲外にあるローカルな要素の新設

- ・ 名前空間とアプリケーションプロファイル  
 名前空間は、用語と定義とを宣言する。
  - Dublin Core 名前空間=Dublin Core 標準
 アプリケーションプロファイルは名前空間から用語を再利用する。
  - たくさんの名前空間から用語を取り込むかも知れない。
  - 局所的な目的のため定義を付け加えるかも知れない。
  - 総べての用語は、名前空間の中に定義されねばならない。
  - 局所的に定義された名前空間があるかも知れない。
- ・ 多重名前空間の例

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/22-rdf-syntax-ns#">
```

```
  xmlns:dc="http://pur.org/dc/elements/1.1/"
```

```
  xmlns:sy="http://pur.org/rss/1.0/modules/syndication/"
```

```
  xmlns:co="http://pur.org/rss/1.0/modules/company">
```

```
<dc:publisher> The O'Reilly Network </dc:publisher>
```

```
<dc:creator> Rael Dormfest(mailto:rael@oreilly.com) </dc:creator>
```

```
<dc:rights> Copyright © 1999, 2000 O'Reilly & Associates, Inc </dc:rights>
```

```
<dc:date> 2000-01-01T12:00+00:00 </dc:date>
```

```
<dc:subject> XML </dc:subject>
```

```
<sy:updatePeriod> hourly </sy:updatePeriod>
```

```
<sy:updateFrequency> 2 </sy:updateFrequency>
```

```
<co:name> XML.com </co:name>
```

```
<co:market> NASDAQ </co:market>
```

```
<co:symbol> XML </co:symbol>
```

```
</rdf:RDF>
```

- ・ 局所的利用の為の標準定義の付加  
 Dublin Core 名前空間の場合
  - DC:Title – ある要素のマシンリーダーブル名
  - "Title : リソースに与えられた名前" – ヒューマンリーダーブルな名前と定義



コレクション記述プロファイル(UKOLN)の場合

- DC:Title – DC 名前空間と同一
  - "Title:コレクションに与えられた名前
- 定義はアプリケーションの状況により変化する。

・局所的利用の為の DC:Title 付加の例

正式な Dublin Core 名前空間の定義では、

- "Title: リソースに与えられた名前"

英国の"アプリケーションプロファイル"の定義では、

- "Title:コレクションに与えられた名前"

この場合、定義が狭い。

・プロファイルは、多重実体のモデルの場合もある。

次の情報を持つ実体としてのリソース(物)

- 主題(dc:title)
- 作成日(dc:date dcq:created)
- 識別子(dc:identifier)

次の情報を持つエージェント(一人の人)

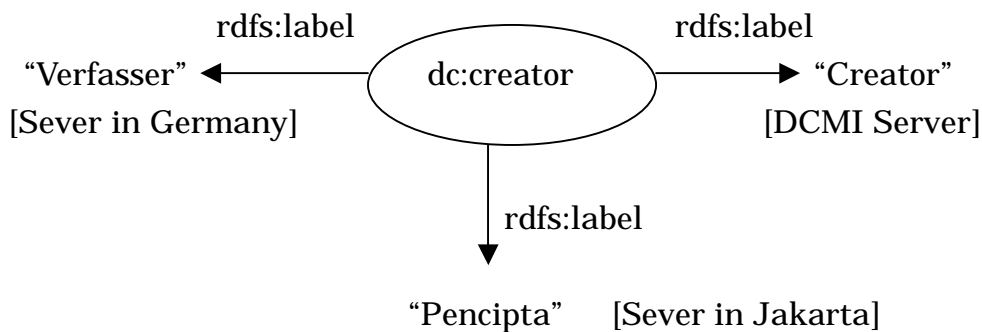
- 名前(vcard:fn)
- 誕生日(vcard:bday)
- 識別子(dc:identifier)

・名前空間の翻訳

Dublin Core は、26ヶ国語に翻訳されている。

- マシンリーダブルトークンは全部で共有されている。
- ヒューマンリーダブルラベルは、各国毎に定義されている。
- 翻訳は、たくさんの国々で配布され、保守されている。
- 更に、DCMI レジストリにリンクされている。

・1つのトークン 多国語の中のラベル



・RDF より強力な文章パターン

### Dublin Core ステートメント

- リソースの作成者は"Tom Baker"である。

- リソースは識別子として <http://foo.org/bar.html> を持つ。

RDF の 3 元(triples)は、同じ事を言うにも、より強力な方法である。

- <http://foo.org/bar.htm> の作成者は、"Tom Baker"である。