

Dublin Coreの最近の話題から

# Overview

- Introduction : Dublin Coreの基本概念
  - Simple DCとQualified DC
  - Dumb-Down原則
  - Application Profile
  - RDFのモデル
- Dublin Core Grammar
- Dublin Coreメタデータの記述形式
- メタデータ規則の流通性

# Dublin Core

- インターネット上の多種多様な情報資源 (resource) の発見のためのメタデータ (目録)
  - Dublin Core Metadata Element Set (DCMES)  
多様な資源に共通な要素
- (実際には) 応用に合わせていろいろな拡張, 使い方がされている。
- 草の根参加者による開発
  - 図書館からの参加は多いが、多くの分野で利用されている。たとえば, 政府情報、教育学習資源情報、など
- WWWコンソーシアムとの連携

# 基本15 エレメント- 一覧

内容	知的財産権関連	具現化
• 時間的・空間的範囲 (Coverage )	• 作成者 (Creator )	• 日付 (Date )
• 内容記述 (Description )	• 寄与者 (Contributor )	• 形式 (Format )
• 関係 (Relation )	• 公開者 (Publisher )	• 資源識別子 (Identifier )
• 出処 (Source )	• 権利管理 (Rights )	• 言語 (Language )
• 主題 (Subject )		
• タイトル (Title )		
• 資源タイプ (Type )		

# メタデータを考えてみる(2/5)

- **論文集** <http://www.nii.ac.jp/dc2001/proceedings/>
- **タイトル** : DC-2001 Online Proceedings または DC-2001 Proceedings of International ...
- **作成者** :Keizo Oyama, Hironobu Gotoda (編集)
- **公開者** :National Institute of Informatics
- **言語** :英語
- **関係** : (この中に論文が掲載されている)
- **日付** : 2001-10-24
- **識別子** :  
(URI) <http://www.nii.ac.jp/dc2001/proceedings/>  
(ISBN) 4-924600-98-9

# Simple DCとQualified DC

- Simple DC:  
基本 15 エLEMENTのみによる記述
- Qualified DC:  
意味をより詳細に表すための記述子qualifier  
を用いた記述
  - Date (日付) : 公開された日付、無効になる日付
  - Coverage (範囲) : 時間的範囲、空間的範囲

# 限定子 (Qualifier) について(1/2)

- 限定子の意味
  - エレメントの意味や記述形式をより詳細に指示する。
  - 訳 :限定子、制約子
- 限定子の種類 (以前からの議論)
  - Element Structure/Structured Value (エレメントの構造。  
例 :著者の名前、所属)
    - Dublin Coreでは扱っていない
  - Element Refinement (エレメントの内容の詳細化・特化。  
例 :日付 作成された日付、公開された日付)
  - Encoding Scheme (エレメントの値の語彙、記述形式。  
例 :LCSH、ISO 8601)
- Dumb-down原則 :限定子を取り除いたときにエレメントの意味と矛盾してはならない。

# 限定子について(2/2)

- 推奨限定子の種類
  - Element Refinement (エレメントの内容を詳細化・特化)
  - Encoding Scheme (エレメントの値の語彙、記述形式)
- Structured Value (エレメントの構造を表す)は除外された。
  - Dumb-down原則に基づく議論
  - 構造をもつ値を書いてはいけない、というのではない。
- Typeエレメントに用いる値の統制語彙(DCMI Type Vocabulary)を決めた。

# DCMI Type Vocabulary

- 下記の9個の値のセット
  - Collection
  - Dataset
  - Event
  - Image
  - Interactive Resource
  - Service
  - Software
  - Sound
  - Text
- Imageに関する議論
  - 動画と静止画

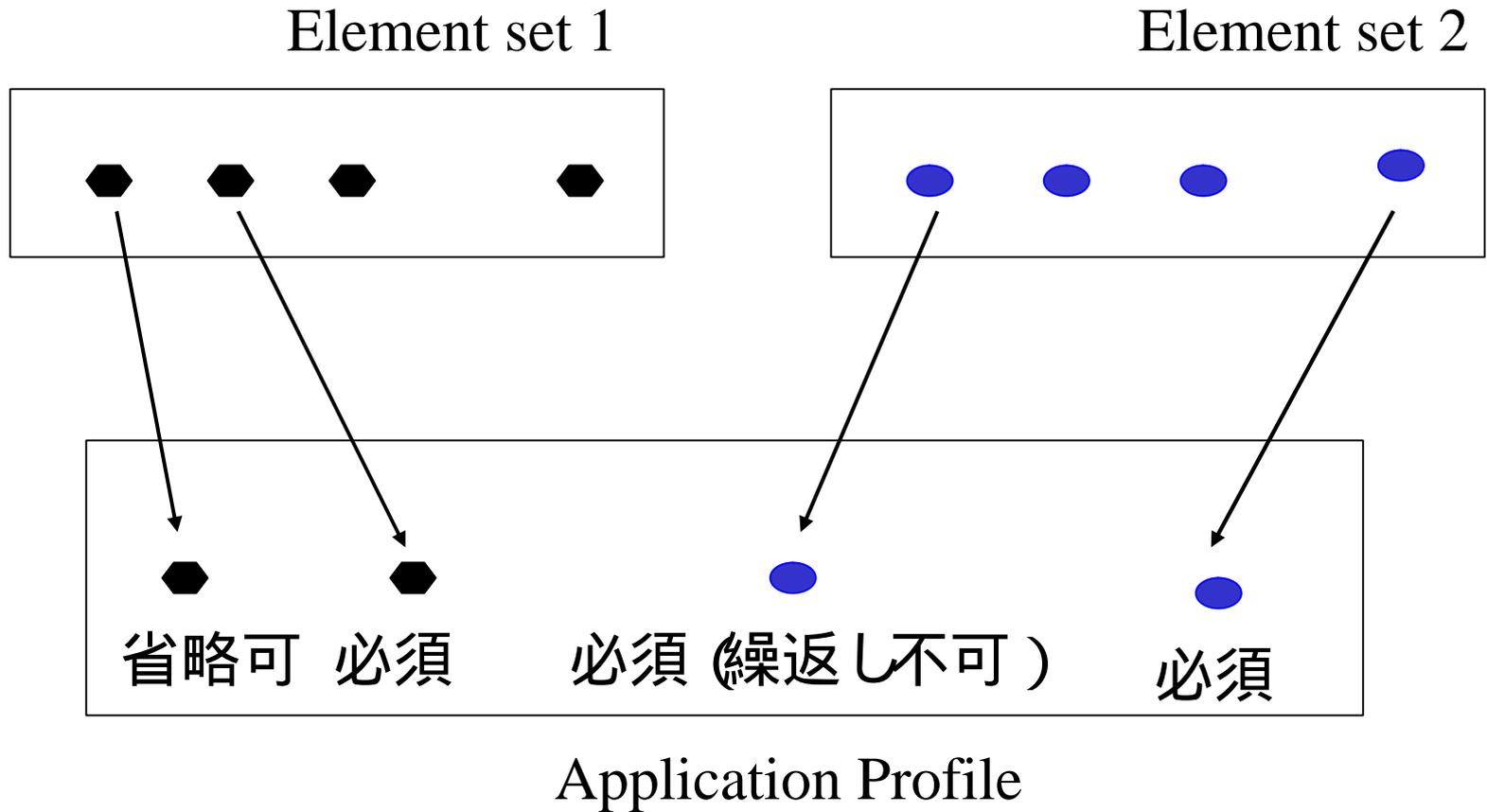
# Element Set と Application Profile (1/3)

- Element Set (基本エレメントとqualifier)は、メタデータを記述するための属性と値の形式を決めるだけ。
- 応用によっては・・・
  - TitleとCreatorを必須にしたい。
  - Subjectを3回以上繰り返すことを必須にしたい。
  - 自分独自のエレメントとDCのエレメントを組み合わせて使いたい。

# Element Set と Application Profile (2/3)

- 応用毎に決まるメタデータエレメントセットの適用規則をApplication Profileと呼んでいる。
- 応用毎にApplication Profileを持つことになる。
  - Dublin Coreの原則「全てのエレメントは省略可能であり、かつ繰り返し可能である」  
Application Profileの一種と見なせる。

# Element Set と Application Profile (3/3)

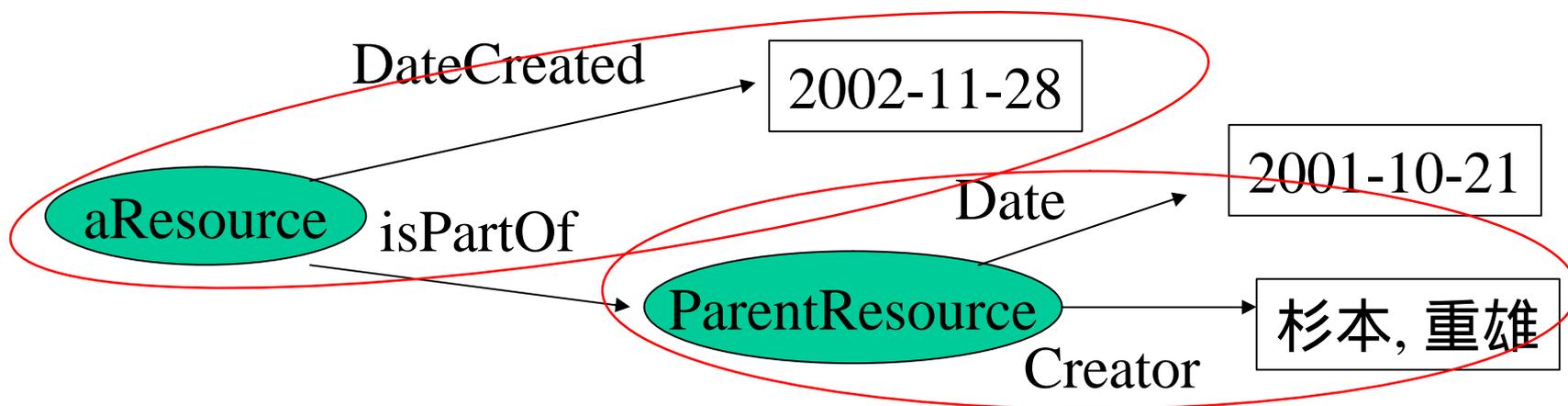


# エレメントや限定子の追加

- 分野毎のWorking Groupからの要求とUsage Boardによる判断
  - Audience Elementの追加
  - いくつかの限定子の追加
    - Recommended, Conforming, Registered
  - Application Profileの認定

# RDFのモデル

- {Resource, Property, Value }の三つ組 (triple )で表す。
- Property = エlement、もしくは詳細化されたElement (Element詳細化限定子付きElement)。たとえば、Creator、DateCreated
- Value=任意の値、統制語彙として決められた値 (例 :NDC, LCSH )、形式を決められた値 (例 :言語コード、URI)



# RDFのモデルとDCMES

- DCMESの開発
  - = Propertyや統制語彙として使う語を決めること
  - = 語の名前とその意味を決めること
  - = メタデータ・ボキャブラリの定義 (語彙定義)  
構文規則は第一義的には含まれていない。
- RDF schemaに基づくDCMESの語彙定義
  - エlement、詳細化されたElementはProperty (詳細化されたElementは何らかのpropertyのsub-property)として定義。
  - 値 (値を表現する語彙、形式)は値の種類を表すClassとして定義。