

平成 22 年度  
新 ICT 利活用サービス創出支援事業  
(電子出版の環境整備)  
メタデータ情報基盤構築事業  
報告書

平成 23 年 3 月 31 日  
代表提案者: 国立大学法人 筑波大学  
契約主体: インフォコム株式会社

## はじめに

### (本事業の目的)

現在の我が国では、様々な種類のメタデータが個別に流通している状況であり、どのようなメタデータが流通しているのか、把握できない状況となっている。これは、標準規格やルールといったものは特に存在せず、それぞれの機関が独自にメタデータの記述に関する取り決め(以下、メタデータ記述規則という)を決めているからである。

そこで、本事業では、メタデータの相互運用性と利用性の高度化を目的として、メタデータ記述規則等の標準仕様と、オープンなメタデータ情報基盤の仕組み(以下、メタデータ情報基盤システムという)を検証・実証する。さらに、検証・実証の結果を踏まえ、メタデータ情報の共有のためのガイドラインを策定する。

### (事業目標)

メタデータ情報基盤システムの開発を行い、各種機関(図書館、公文書館、博物館、美術館等)のメタデータ記述規則を収集し、蓄積管理した実証環境を構築する。その上でメタデータ記述規則に関する情報の公開、共有、維持管理、それを利用したメタデータ記述規則の定義、メタデータの相互運用性について実証し、メタデータ情報基盤のサービス展開について検証する。

メタデータ情報基盤システムの開発にあたって、中核になる技術として、セマンティックウェブ技術とメタデータスキーマレジストリ技術を用いる。前者はメタデータの WWW 上での流通のための標準技術、後者はメタデータ記述規則に関する情報の蓄積共有サービスのための技術である。

収集してきたメタデータ記述規則を、こうした技術を利用してインターネット上での流通性を高めることで、既存のサービスの効率化や高度化を進め、新しいサービスの創出(「ICTを活用した新しいサービスの概要」を参照)を促進する基盤作りを行う。また、メタデータ記述規則を保存するシステムの存在は、デジタルコンテンツ及びメタデータのメンテナンス性を高め、デジタルコンテンツの長期利用性向上にも寄与する。

また、本事業の基盤となる技術の確立と標準化を目標とし、分野の違いを超えネットワーク上で共有できる幅広いメタデータを対象として、「①メタデータ記述規則を定義する標準方式の開発」、「②メタデータ記述規則を提供するための標準方式の開発」、「③メタデータ記述規則を収集、提供、維持管理するための体制の整備」、④「①～③に対する専門家による調査・検討」を実施する。

また、前述の①～④の成果を元に「(1) ICT を利活用した新しいサービスの創出に向けた開発・実証」を実施した上で、メタデータ情報の共有のためのガイドラインを策定する。本ガイドラインは

デジタルコンテンツに関わる様々な機関(図書館、公文書館、美術館、博物館、サービス事業者、出版社等)におけるメタデータの相互利活用のために広く利用されるものである。

## 本書の構成

はじめに.....	2
本書の構成.....	4
第1部 メタデータの基礎知識とメタデータ基盤の必要性.....	11
1. プロジェクトについて.....	11
1.1 背景と狙い.....	11
1.1.1 プロジェクトの背景.....	11
1.1.2 プロジェクトの狙い.....	13
1.1.3 検討の経緯.....	14
1.2 提案の概要.....	17
1.2.1 新ビジネス分野の基盤技術の確立・標準化、ガイドラインの策定.....	17
1.2.2 ICT を利活用した新しいサービスの創出に向けた開発・実証.....	21
1.3 プロジェクト実施体制と活動.....	24
1.3.1 プロジェクトの実施体制.....	24
1.3.2 プロジェクトの活動実績.....	31
2. メタデータとは何か.....	33
2.1 背景.....	33
2.2 メタデータとは.....	33
2.3 メタデータの活用と相互運用の重要性.....	34
2.3.1 デジタルコンテンツにおけるメタデータ.....	34
2.3.2 メタデータ利用の現状.....	35
2.3.3 メタデータの相互運用に向けて.....	35
2.4 メタデータ情報とスキーマ.....	37
2.4.1 メタデータを共有・利用するための情報.....	37
2.4.2 メタデータ記述規則の基本要素.....	37
2.4.3 メタデータ語彙.....	38
2.4.4 メタデータ語彙とアプリケーション・プロファイル.....	38
3. メタデータ情報の基盤とニーズの現状調査.....	41
3.1 アンケート調査.....	41
3.2 シンポジウム招聘者会議.....	45
3.3 現状のまとめ.....	48
第2部 基盤となる技術の確立・標準化とガイドライン.....	49
4. データ表現方式の標準化動向.....	49

4.1 構造化記述と共通語彙の模索.....	49
4.1.1 公開メタデータとマーク付け.....	49
4.1.2 XML による構造化記述.....	50
4.1.3 標準語彙ダブリンコア.....	52
4.2 データモデルの標準化: RDF.....	53
4.2.1 メタデータ記述フォーマットの源流.....	53
4.2.2 対象の記述とトリプル.....	54
4.2.3 URI とリソース.....	55
4.2.4 RDF バス.....	56
4.2.5 RDF の構文.....	57
4.3 RDF スキーマと OWL.....	57
4.3.1 RDF スキーマ.....	57
4.3.2 OWL.....	60
4.4 その他のデータモデル「トピックマップ」.....	62
4.4.1 トピックマップの概要.....	63
4.4.2 標準化動向.....	64
5. 各分野でのメタデータ記述の現状.....	71
5.1 MLA 領域のメタデータ記述規則.....	71
5.1.1 図書館分野のメタデータ記述規則.....	71
5.1.2 博物館分野のメタデータ記述規則.....	73
5.1.3 文書館分野のメタデータ記述規則.....	76
5.2 コンテンツ流通とメタデータ.....	78
5.2.1 電子書籍のメタデータ.....	78
5.2.2 音楽データ.....	79
5.2.3 メタデータの交換と流通.....	80
5.2.4 管理のためのメタデータ.....	82
5.3 メタデータ記述の代表的語彙.....	84
5.3.1 セマンティック・ウェブとメタデータ記述語彙.....	84
5.3.2 ダブリンコア.....	85
5.3.3 FOAF.....	88
5.3.4 SKOS.....	91
6. メタデータ記述規則を定義するための標準方式.....	93
6.1 メタデータ記述規則定義の現状.....	93
6.1.1 国立国会図書館書誌記述スキーム.....	93
6.1.2 国立国会図書館デジタルアーカイブシステム.....	94
6.1.3 東京国立博物館.....	96

6.1.4	国立美術館	97
6.1.5	国立公文書館	99
6.1.6	学術機関リポジトリの相互運用メタデータ指針	99
6.2	メタデータ・スキーマ定義方法の標準化動向	101
6.2.1	アプリケーション・プロファイル	102
6.2.2	DCMI シンガポール・フレームワーク	102
6.2.3	DCMI 記述セットプロファイル	103
6.2.4	OWL を用いた制約記述	106
6.3	メタデータ記述規則定義言語の提案	106
6.3.1	メタデータ記述規則定義言語の要件と考え方	107
6.3.2	OWL クラス定義によるレコード記述規則の表現	108
6.3.3	プロパティ制約による項目記述規則の表現	109
6.3.4	管理情報の表現	111
6.4	メタデータ記述規則の簡易表現	112
6.4.1	記述規則と OWL-DSP	112
6.4.2	データ項目と記述規則	112
6.4.3	簡易 DSP の記述例	113
6.4.4	OWL-DSP への変換	114
7.	メタデータ情報共有のためのガイドラインの策定	116
7.1	ガイドライン作成の準備と検討	116
7.1.1	ガイドラインの事例検討	116
7.1.2	作成の経過	118
7.2	ガイドラインの基本方針	120
7.2.1	ガイドラインの目的と役割	120
7.2.2	このガイドラインの構成	121
7.3	基本ガイドライン	122
7.3.1	スキーマの選択・設計と公開の指針	123
7.3.2	メタデータ記述の推奨指針	126
7.3.3	メタデータの公開と交換・利用に関する指針	131
7.3.4	運用に関する指針	134
8.	専門家による調査・検討	137
8.1	検討会での報告と議論	137
8.1.1	メタデータの設計とスキーマの現状について	137
8.1.2	RDF とウェブ上でのメタデータについて	137
8.1.3	読みに関する問題とダブリンコアの定義	138
8.1.4	ガイドラインについて	138

8.2 研究会での報告と議論.....	138
8.2.1 問題提起.....	139
8.2.2 研究会およびレジストリの継続的運用について.....	139
第3部 新しいサービスの創出に向けた開発と実証.....	140
9. メタデータの共有とスキーマ・レジストリ.....	140
9.1 メタデータ共有の課題.....	140
9.2 レジストリの役割.....	143
9.2.1 コンテンツ提供者＝メタデータ記述者にとってのレジストリ.....	144
9.2.2 コンテンツ利用者にとってのレジストリ.....	145
9.3 メタデータ・レジストリの事例.....	146
9.3.1 DCMI Metadata Registry.....	146
9.3.2 Open metadata registry.....	153
9.4 メタデータ・スキーマ・レジストリの課題と方向性.....	160
10. レジストリの機能要件.....	161
10.1 考え方とユースケース.....	161
10.1.1 既存の記述規則を標準的な形で表現・公開する.....	161
10.1.2 メタデータを記述するために利用可能な記述規則を探す.....	162
10.1.3 公開されているスキーマを参考に独自のスキーマを定義する.....	163
10.1.4 異なるメタデータ形式間での相互運用.....	163
10.1.5 複数のメタデータをマッシュアップする.....	164
10.2 機能要件:スキーマ登録に関する要件.....	165
10.2.1 RDF グラフとして表現されるスキーマを格納、管理できる.....	165
10.2.2 RDF 形式の語彙定義を確認しながら登録できる.....	166
10.2.3 表などの半形式的定義をもつ記述規則を対話的に登録できる.....	166
10.2.4 スキーマファイルを解析して変換の前処理を行なう.....	167
10.2.5 登録スキーマの更新ができる.....	167
10.3 機能要件:コンテンツ・メタデータ提供者のレジストリ利用に関する要件..	168
10.3.1 登録されている記述規則、語彙定義の一覧、詳細を表示できる.....	168
10.3.2 登録されている記述規則、語彙定義を検索し、該当するものを表示できる.....	169
10.3.3 記述規則、語彙定義を複数フォーマットで取得できる.....	170
10.3.4 独自記述規則の作成を支援できる.....	170
10.3.5 複数記述規則間の関連を調べ、分かりやすく示すことができる.....	171
10.3.6 選択した記述規則に基づくメタデータ記述の支援できる.....	172
10.3.7 コンテンツの長期保存・利用のためのメタデータ記述標準.....	173
10.4 機能要件:サービス提供者の高度なメタデータ利用に関する要件.....	173

10.4.1	レジストリを検索し、語彙、記述規則を調べることができる	173
10.4.2	メタデータ変換の支援	173
10.4.3	コンピュータによる問い合わせインターフェース	174
10.5	簡易 DSP から OWL-DSP への変換	175
10.5.1	名前空間と管理情報	175
10.5.2	レコード記述規則の表現	175
10.5.3	項項目記述規則からクラス制約の生成	177
11.	レジストリの設計	179
11.1	開発言語・ライブラリの選定	179
11.1.1	開発言語: Java	179
11.1.2	データベース : PostgreSQL	179
11.1.3	ライブラリ、フレームワーク	179
11.2	基本設計	181
11.2.1	機能一覧	181
11.2.2	アプリケーション構成図	183
11.3	ユーザインターフェース設計	184
11.3.1	語彙定義登録機能	184
11.3.2	記述規則登録機能	187
11.3.3	語彙定義検索およびファイル出力機能	190
11.3.4	記述規則検索およびファイル出力機能	192
11.3.5	メタデータインスタンスの変換機能	194
11.4	API 設計	198
12.	レジストリの開発作業	203
12.1	第1フェーズの開発	204
12.2	第2フェーズの開発	206
12.3	第3フェーズの開発	207
12.4	開発上の課題と解決	208
12.5	今後の展望	215
13.	サービス(レジストリ)の実証	216
13.1	実証実験の立案	216
13.1.1	実証実験の方針	216
13.1.2	実証実験の内容と実証者の選定	216
13.1.3	評価基準の設定	217
13.2	実証実験 1:スキーマの登録	219
13.3	実証実験 2:スキーマの検索・作成	221
13.4	実証実験 3:スキーマの変換	222

13.5 実験結果のまとめ.....	223
第4部 今後の展望.....	224
14. メタデータ情報基盤システムのサービス展開.....	224
14.1 実証実験 1(スキーマの登録)の評価と課題.....	224
14.1.1 簡易記述規則の登録.....	224
14.1.2 記述規則メタデータの登録.....	225
14.1.3 未登録語彙の登録・公開.....	226
14.1.4 登録機能全般についての評価.....	226
14.2 実証実験 2(スキーマの検索・作成)の評価と課題.....	228
14.2.1 14.2.1 語彙の検索.....	228
14.2.2 記述規則の検索.....	228
14.2.3 語彙・記述規則の詳細情報.....	229
14.2.4 登録済み規則を雛形としての新規規則作成.....	229
14.2.5 検索・作成機能全般についての評価.....	230
14.3 実証実験 3(メタデータの変換と再利用)の評価と課題.....	230
14.3.1 CSV 形式データの RDF への変換.....	230
14.3.2 トピックマップとしての出力.....	231
14.3.3 シンプル DC への変換.....	232
14.4 全体的な評価と今後の課題.....	233
14.4.1 システム全体に関する評価.....	233
14.4.2 関心のある機能.....	235
14.4.3 今後の課題.....	236
15. メタデータ記述規則を収集、提供、維持管理するための体制の整備.....	237
15.1 レジストリの維持運用.....	237
15.1.1 メタデータ・スキーマの登録と共有.....	237
15.1.2 維持運用のための検討課題.....	238
15.1.3 レジストリの継続的開発.....	238
15.1.4 維持運用の組織と計画.....	239
15.2 記述規則定義言語とガイドラインの周知・普及.....	240
15.2.1 記述規則定義言語を用いたスキーマの共有.....	241
15.2.2 ガイドラインの周知と運用.....	242
16. メタデータ共有がもたらす可能性.....	244
16.1 メタデータの応用と領域を超えたサービス.....	244
16.1.1 百科事典から WWW へ.....	244
16.1.2 検索サービスとメタデータ.....	245
16.1.3 ウェブサービスを結ぶメタデータ.....	246

16.1.4 MLA メタデータの広がり .....	251
16.2 リンクするデータ .....	254
16.2.1 リンクするデータとは .....	254
16.2.2 Linking Open Data プロジェクト .....	256
16.2.3 BBC のリンクするデータ .....	257
16.2.4 リンクするデータのハブ .....	260
16.2.5 日本におけるリンクするデータ .....	262
16.2.6 リンクするデータの広がりとレジストリの役割 .....	263
16.3 セマンティック・マーク付けとレジストリ .....	264
16.3.1 セマンティック・マーク付け .....	264
16.3.2 検索エンジンとセマンティック・マーク付け .....	268
16.3.3 セマンティック・マーク付けとスキーマ .....	269
16.4 トピックマップがもたらす可能性 .....	271
16.4.1 可能性を広げるためのシステム構造 .....	271
16.4.2 可能性の切り口 .....	276
16.4.3 利用イメージ .....	280
16.4.4 ビジネス機会 .....	281

(別冊)メタデータ情報基盤構築事業報告書 Appendix

・付録資料

- A1. メタデータ・レジストリの機能要件
- A2. メタデータ・スキーマ・レジストリ設計書
- A3. メタデータ共有のためのガイドライン
- A4. メタデータ・スキーマ定義言語
- A5. Topic Maps
- A6. メタデータ情報提供に関するアンケート
- A7. メタデータに関するアンケート
- A8. 第1回検討会(2010-11-11)
- A9. 第2回検討会(2010-12-07)
- A10. 国際シンポジウム(2010-12-08)
- A11. 第3回検討会(2011-01-31)
- A12. 第4回検討会(2011-02-21)
- A13. 第5回検討会(2011-03-23)
- A14. メタデータ基盤システム利用者操作説明書
- A15. メタデータ情報基盤構築事業ホームページ

## 第1部 メタデータの基礎知識とメタデータ基盤の必要性

### 1. プロジェクトについて

#### 1.1 背景と狙い

現在、インターネット上には、分野毎に造られた様々なメタデータが流通している。これらのメタデータは、主に個々の分野において記述規則の標準化が図られているため、その分野内での相互運用性は向上していても、複数分野をまたいでの相互運用となると容易ではないケースが大半を占める。

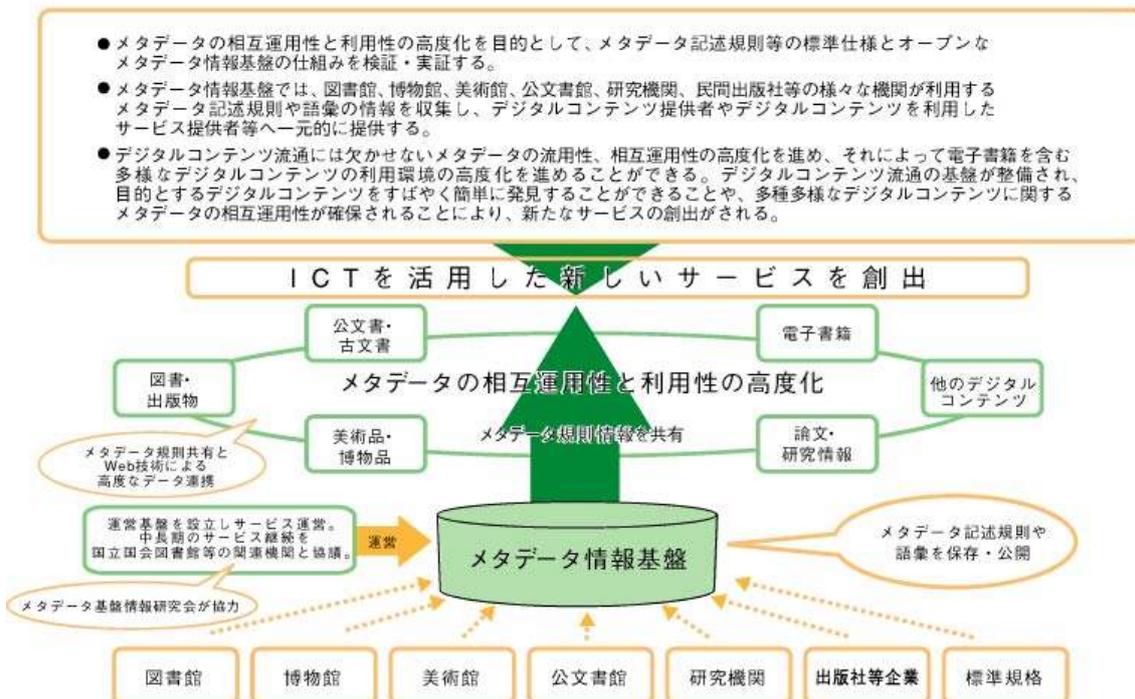
仮に、このようなメタデータ規則をネット上で公開し、複数分野でも共有できる環境を整えば、現状の低い相互運用性という課題も自ずと解消に向かう。さらに、インターネット指向の国際標準規格に基づく形でメタデータの規則を機械処理可能な形式で表し、これを蓄積・提供することができれば、インターネット上でのメタデータ規則の共有性や再利用性は格段に向上するはずである。

こうした観点に基づき、本事業では、WWW の標準規格や Dublin Core Metadata Initiative(DCMI)のメタデータ相互運用性向上のためのフレームワークを基盤として、メタデータ規則と関連情報を収集・蓄積・提供するメタデータ・スキーマ・レジストリを核とするメタデータ情報基盤の構築を進める。これにより、メタデータの利用性や相互運用性の向上を図り、コンテンツの流通性向上に資することを目的としている。

##### 1.1.1 プロジェクトの背景

ネットワークを介したコンテンツ流通においては、利用者は自分でコンテンツを探し、用途や目的に合っているか吟味した上で利用・購入することになる。利用者がより効率的に目的の情報へと辿り着ける環境を提供するには、コンテンツにアクセスする上での窓口となる情報、すなわちメタデータを適切に準備する必要がある。

たとえば、図書館や書店で検索に用いられる図書のデータ(書誌データ)は典型的なメタデータであり、機械可読形式目録(MARC)として広く用いられている。これに対し、WWW 上ではよりシンプルな書誌データの標準体系が提案されている。また、アクセシビリティや想定読者に関する情報は書誌データとは別に標準化されており、分類や主題を表す語彙や権利管理情報もメタデータの一つとしての開発が進んでいる。



◆ 図 1.1: ICT の利活用におけるメタデータ情報基盤の役割

現状では、こうした様々なメタデータが個別に流通しているため、どのような種類が存在するのかさえ把握し難い状況に陥っている。これは、標準規格やルール等が存在せず、記述に関する規則もそれぞれの機関が独自に決定しているからである。

そもそも、従来型のメタデータ記述規則からして、我が国に限らず業界や用途ごとに作られてきた経緯がある。加えて、インターネット上で流通する事を前提に作成されていないため、メタデータを用いたコンテンツ流通が妨げられる一因となっている。

そこで、本事業では、メタデータの相互運用性と利用性の高度化を目的として、記述規則等の標準仕様とオープンな情報基盤の仕組みを検証・実証する。さらに、そこで得た結果を踏まえ、メタデータ情報の共有のためのガイドラインを策定する。

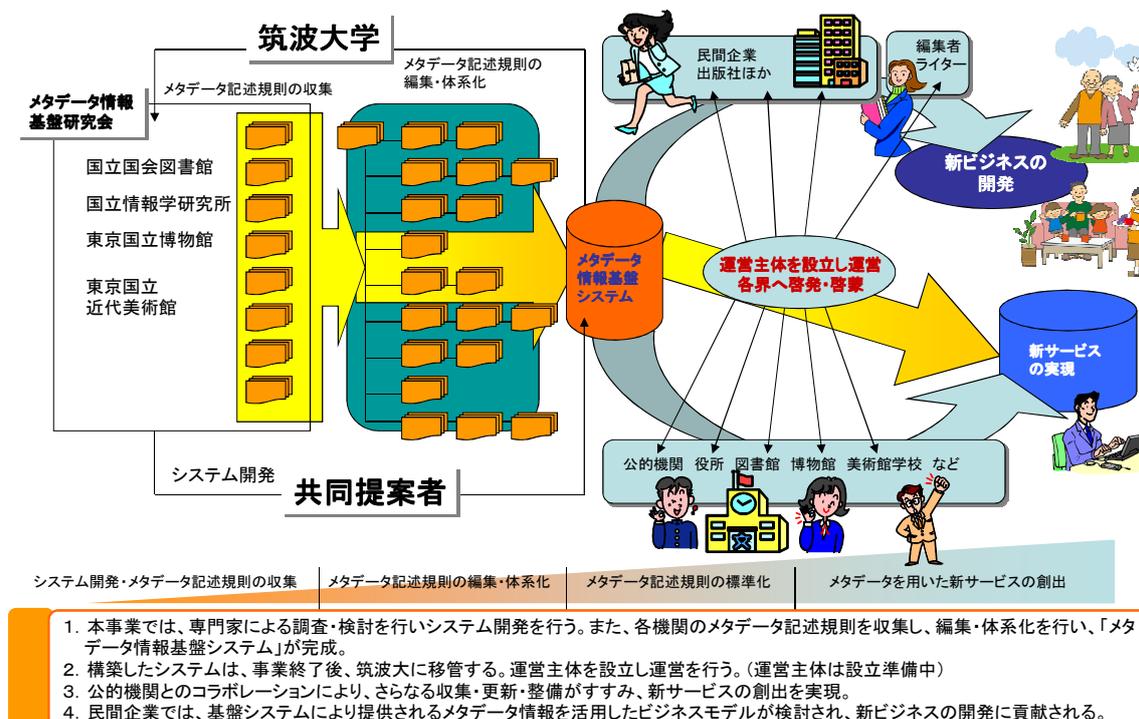
なお、本事業における検証作業では、WWW コンソーシアムが 2000 年ごろから進めてきたセマンティック・ウェブ技術と、筑波大学やメタデータ情報基盤研究会の協力機関が進めてきたメタデータ記述規則の蓄積共有のための経験とノウハウを活用する。

### 1.1.2 プロジェクトの狙い

本事業は、デジタルコンテンツ提供者や関連サービスの提供者らに対し、各機関のメタデータ記述規則や語彙の情報を、メタデータ情報基盤システムを通じて一元的に提供できる情報インフラの構築を目指すものである。

メタデータ情報基盤システムとは、図書館や公文書館、博物館、美術館、出版流通事業者といった様々な機関が利用する分野個別のメタデータ記述規則を収集蓄積し、分野横断的な記述規則等の標準方式を定めた上で、インターネット上で公開・共有・維持管理するシステムである。メタデータ・スキーマ・レジストリとして活用することで、産学官民の誰もが利用できるサービス基盤の実現を目指している。

本基盤システムの特長は、利用者(人)だけでなく、セマンティック・ウェブ技術を用いて「システム(コンピュータ)自体も利用できるサービス」を想定していることにある。よって、本基盤システムの運用段階では、以下のような利用イメージとなる。



#### ◆ 図 1.2: 運用段階での利用イメージ

(1) デジタルアーカイブやデジタルコンテンツ流通のシステムの設計開発者、運営者によるメタデータ記述規則の探索・閲覧(人間による利用)

(2) デジタルアーカイブ「システム」やデジタルコンテンツ流通「システム」自体が行う、メタデータ交換やメタデータ相互利用等のためのメタデータ記述規則の参照(コンピュータによる利用)

前者はシステム開発の効率化とシステム間の相互運用性向上に役立つ。後者は、ネットワーク上でのメタデータ相互運用の柔軟性を高め、コンテンツの流通性を高めるほか、コンテンツに関わる種々の付加価値ビジネスの基盤として利用できる。

さらに、メタデータ情報基盤システムは、新たなサービスの発生を促進・支援することも可能である。具体的には、以下のようなサービスが生まれることが期待できる。

- ① 博物館、美術館、図書館、文書館等によるコンテンツやサービスの連携（MLA 連携）  
⇒ サービス提供者間連携
- ② 地域、領域、利用者、利用環境に応じたコンテンツ提供を行う付加価値サービス  
⇒ サービス対象の特化ビジネス
- ③ 図書とビデオや Web ページ等、いろいろな情報資源を組み合わせたサービス  
⇒ 異形態資源の連携サービス
- ④ デジタルコンテンツの維持管理のための情報を長期にわたって維持するサービス  
⇒ コンテンツの長期利用性高度化支援サービス

つまり、メタデータ情報基盤システムは、メタデータ記述規則の情報提供を通じて、新たなサービスを構築・提供する基盤の提供を可能にするのである。また、今後発展が望まれるコンテンツ提供サービスのクラウド化に不可欠なインフラともなり得る。

以上のようなメタデータ情報基盤システムの機能は、産学官民を問わずデジタルコンテンツの流通にかかわる人々（デジタルコンテンツの提供者、デジタルコンテンツ利用したサービス提供者、コンテンツ流通に関わる事業者ら）からの高い需要が見込まれる上に、デジタルコンテンツの増加に伴いメタデータ記述規則の増加・乱立が予想されることから、可及的速やかに構築・提供する必要があると言える。

### 1.1.3 検討の経緯

平成 22 年 3 月、総務省・文部科学省・経済産業省の副大臣・大臣政務官による共同懇談会として、「デジタル・ネットワーク社会における出版物の利活用推進に関する懇談会」（以下三省共同懇談会という。）が開催された。電子出版サービス事業において先行する諸外国との利便性における格差を解消し、デジタル・ネットワーク社会に対応して広く国民が出版物にアクセスできる環境を整備することを目指すとしており、以下のような内容を検討課題に挙げている。

- ① デジタル・ネットワーク社会における出版物の収集・保存の在り方
- ② デジタル・ネットワーク社会における出版物の円滑な利活用の在り方

### ③ 国民の誰もが出版物にアクセスできる環境の整備 等

三省共同懇談会の開催に先駆けて、同年 5 月 11 日には政府の高度情報通信ネットワーク社会推進戦略本部においても「新たな情報通信技術戦略」が決定されている。新市場の創出と国際展開の実現に向けた「デジタルコンテンツ市場の飛躍的拡大」を掲げ、「電子書籍の普及に向け、非商業分野におけるデジタルアーカイブ化を促進するとともに、商業分野において民間による標準規格の策定、権利処理ルールやビジネスモデル形成、ソフト・ハード(端末)の開発の取り組みを支援する」こととしている。

こうした課題や目標は、同年 5 月 21 日に知的財産戦略本部が発表した「知的財産推進計画 2010」にも同趣旨の内容が明記されている。また、総務省においても、同年 5 月 6 日に「新たな成長戦略ビジョンー原口ビジョン IIー」を発表し、電子出版に関する技術的課題(フォーマット化)の解消に向けた検討に早急な着手を促すなど、インターネット上のコンテンツ流通環境の整備は各方面で急務と認識されるに至っている。

各機関での動きが活発化する中、同年 6 月 8 日に行われた三省共同懇談会の第 2 回懇談会では、技術ワーキングチームが 18 項目にわたって取りまとめられた施策案が提示された。「オープン型電子出版環境の実現と知のインフラへのアクセス環境の整備に向けて求められる具体的施策」の中で、同ワーキングチームは「メタデータの相互運用性の確保に向けた環境整備」の必要性を提言している。

出版物単位での検索やアクセスのみならず、字句、記事、目次、頁等のより細粒度な単位での相互参照を可能とし、関連情報・文献の検証や記録を容易にするためには、メタデータ情報基盤の確立が不可欠であるという認識が広く共有されつつあることを踏まえ、本事業では同年 7 月、関係機関・民間会社に基盤研究プロジェクトへの参加・協力を依頼し、その翌々月には計画を策定するに至った。

#### <メタデータ情報基盤研究会発足までの経緯>

平成 22 年 4 月以前	協力機関である国立国会図書館のセマンティック・ウェブ対応への支援、Dublin Core Metadata Initiative(メタデータに関する国際組織)との連携によるメタデータ・レジストリの開発等を実施し、基盤構築に関する検証を実施。
平成 22 年 6 月	デジタル・ネットワーク社会における出版物の利活用の推進に関する懇談会技術ワーキングチームの議論において、メタデータ情報基

盤の重要性が認められる。

- |              |                               |
|--------------|-------------------------------|
| 平成 22 年 6 月  | メタデータ情報基盤のあるべき姿の具体的検討開始       |
| 平成 22 年 7 月  | 基盤研究に対しての各機関や民間会社の協力への参加・協力依頼 |
| 平成 22 年 9 月  | 本事業に関する計画策定                   |
| 平成 22 年 11 月 | メタデータ情報基盤研究会発足                |

※本事業に協力する研究会が設立予定。

## 1.2 提案の概要

### 1.2.1 新ビジネス分野の基盤技術の確立・標準化、ガイドラインの策定

#### (1) 本事業の成果目標

本事業の基盤となる技術の確立と標準化を目標とし、分野の違いを超えてネットワーク上で共有可能な幅広いメタデータを対象としたガイドラインの策定を検討するに当たり、成果目標を以下の通りに掲げている。

- ① メタデータ記述規則を定義する標準方式の開発
- ② メタデータ記述規則を提供するための標準方式の開発
- ③ メタデータ記述規則を収集、提供、維持管理するための体制の整備
- ④ 上記①～③に対する専門家による調査・検討

また、上記の4つの成果をもとに、開発・実証作業(1.2.2 にて内容を詳述)を実施した上で、メタデータ情報の共有のためのガイドラインを策定する。

なお、ガイドライン策定にあたっては、デジタルコンテンツに関わる様々な機関(図書館、公文書館、美術館、博物館、サービス事業者、出版社等)におけるメタデータの相互利活用のために広く利用されることを念頭に置く。

#### (2) 実施内容

##### ① メタデータ記述規則を定義するための標準方式の開発

様々な機関で利活用できるメタデータ記述規則を定義するために、標準方式を開発する。開発にあたっては、国際標準や業界標準を取り入れ、国内外で広く共通に利用できる方式の確立を目指すものとする。

##### ② メタデータ記述規則を提供するための標準方式の開発

ア. 蓄積したメタデータ記述規則を利用者(人およびコンピュータによる利用)へインターネットを通じて提供するための標準方式を開発する。また、新たにメタデータ記述規則を作成する機関に必要な提供方法や、メタデータ記述規則の作成支援の方法、コンテンツの長期利用の際にもメタデータ記述規則を安定して利用できるようにするための維持管理の方法等も併せて開発する。

イ. メタデータ記述規則を収集、提供、維持管理するための体制を整備する。

ウ. メタデータ基盤情報システムが公開するメタデータ記述規則を長期にわたり継続的に収集、提供、維持管理するための体制整備を行う。

エ. 上記ア. ～ウ. に対する専門家により調査・検討する。

ア. ～ウ. の開発段階および開発した結果について、専門家による調査・検討を行い開発に反映する。専門家の選任はメタデータ情報基盤研究会の助言と協力を受け実施する。

(3)メタデータ情報の共有のためのガイドラインの作成

上記ア. ～エ. で得た成果をもとに開発・実証作業(1.2.2にて内容を詳述)を実施した上で、メタデータ情報の共有のためのガイドライン策定に臨む。

成果	メタデータ情報基盤システム	メタデータ記述規則を定義・提供するための標準方式 メタデータ記述規則を収集、提供、維持管理するための体制整備
課題・背景	ネットワークを介したデジタルコンテンツ流通においてメタデータが重要な役割を担っている。様々なメタデータが流通しており、どのようなメタデータが流通しているか把握できない状況となっている。	各機関のメタデータ記述規則は文書形式で記述されたものがほとんどであり、積極的な共有は進められていない。一部分野でのメタデータ記述規則の標準化は図られているが、分野を越えた標準化は実施されていない。
	<ul style="list-style-type: none"> <li>・メタデータに関する取り決め(メタデータ記述規則)のルールや標準規格がない事が原因</li> <li>・ルールや標準規格があれば、メタデータ記述規則に統制がとれて、語彙が氾濫しない</li> <li>・標準化にあたって、より高度化(異なる分野でも利用できる形式)を進めていかねばならない</li> </ul>	
目的	<ul style="list-style-type: none"> <li>・メタデータの相互運用性と利用性の高度化</li> <li>・デジタルコンテンツおよびメタデータの長期利用性の確保</li> </ul>	<ul style="list-style-type: none"> <li>・メタデータの相互運用性と利用性の高度化</li> <li>・メタデータ情報の共有</li> </ul>
内容	<ul style="list-style-type: none"> <li>・メタデータスキーマレジストリの開発</li> <li>・メタデータスキーマレジストリに収集蓄積するメタデータ記述規則の収集</li> <li>・収集したメタデータ記述規則を標準方式に準拠した形式に変換しメタデータスキーマレジストリで蓄積管理。</li> </ul>	<ul style="list-style-type: none"> <li>・メタデータ記述規則を定義・提供するための標準方式</li> <li>・メタデータ記述規則を収集、提供維持管理するための体制整備</li> <li>・上記について専門家による調査・検討</li> </ul>

↓

実施の成果を踏まえて、メタデータ情報の共有のためのガイドラインを策定する。

◆ 図 1.3: メタデータ情報の共有のためのガイドライン 課題と目的について

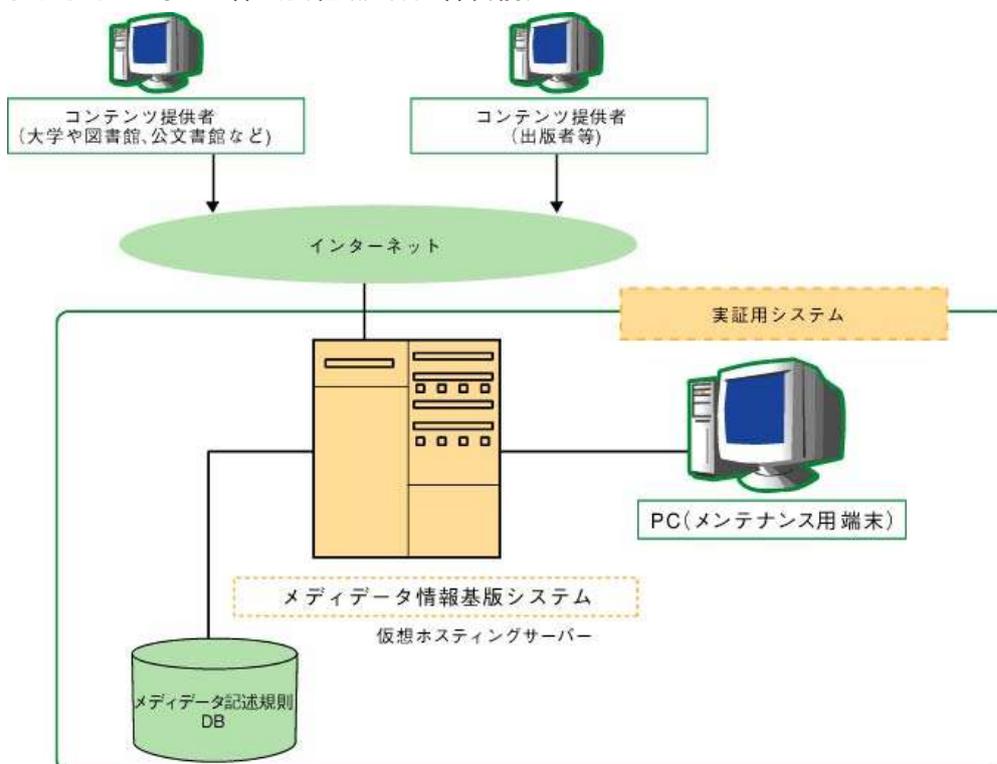
(4)実施場所

① 主な開発場所

筑波大学知的コミュニティ基盤研究センター(茨城県つくば市)  
 およびインフォコム株式会社(渋谷区神宮前)

② 実証実施場所

筑波大学知的コミュニティ基盤研究センター(茨城県つくば市)  
 およびインフォコム株式会社(渋谷区神宮前)



◆ 図 1.4:実証展開イメージ

(5)実施期間

平成22年11月～平成23年3月の5ヶ月間と定めた。期間内の実施スケジュール野詳細は以下の通りである。

	9	10	11	12	1	2	3
委託先候補の決定、委託契約締結							
(1) ICTを利活用した新しいサービスの創出に向けた開発・実証							
①技術開発・システム構築							
ア. メタデータレジストリの開発							
イ. 蓄積するメタデータ規則の収集							
ウ. メタデータ規則の高度化							
エ. 上記開発と基盤維持管理のための体制等に関する検討及び国内外の調査							
②サービスの実証							
ア. 実証の準備							
イ. フィールド実証の実施							
ウ. 実証データの取得・分析							
③サービス展開の検証							
(2) 新しいビジネス分野の基盤となる技術の確立、技術標準化、運用ガイドラインの策定等の実現							
①メタデータ規則記述を定義するための標準方式の開発							
②メタデータ規則を提供するための標準方式の開発							
③メタデータ規則を収集、提供、維持管理するための基盤整備							
④①～③に対する専門家による調査・検討							
⑤メタデータ情報の共有のためのガイドライン作成							
③プロジェクト管理							
①報告書の作成・修正							
②メタデータ情報基盤事業検討会の開催			▲	▲		▲	▲
中間報告、成果報告				▲			▲

◆ 図 1.5:実施スケジュールについて

6)プロジェクト終了時における利用者への対応予定

平成 23 年 4 月以降、本事業で開発したメタデータ情報基盤システムは、共同提案者により別途提供するサーバ機器にインストールし、筑波大学知的コミュニティ基盤研究センターにて運用する。

本事業での実証結果をもとに、メタデータ情報基盤システムの運営基盤(コンソーシアム等)を設立しサービスの運営を行う。本運営基盤に対しては引き続き、筑波大学が主催するメタデータ情報基盤研究会が助言を行う。

なお、中長期にわたるサービスの継続に関しては、国立国会図書館等の関連機関との協議を行い、安定したサービスを国民に提供できるようにする。

主体	現状の課題	プロジェクト実施後	
		メリット	負担
コンテンツ利用者	必要なコンテンツを入手する為に、インターネットを介して情報収集を行うが、必要な情報が氾濫する情報量の中で埋もれており、ネットに詳しくない人間にとって見つけ出すのが容易でない。	必要なコンテンツを見つけるためのメタデータの相互連携が促進され、新たなサービスが生まれる。 これによってこれまで発見できなかった情報を発見できる。	新たなサービスの操作習得。
コンテンツ提供者 (大学、図書館、公文書館等)	各機関ごとに、それぞれで独自のメタデータを作成している場合が多く、メタデータの相互利用が容易ではない。 各機関のメタデータ記述規則を容易に入手する仕組みが無く、メタデータ記述規則の再利用を行うための情報収集が困難である。	メタデータ記述規則の情報入手が容易になり、標準的なメタデータ記述規則に沿ったメタデータ作成が可能となる。 作成したメタデータのメタデータ記述規則が広く公開されることで、相互利用が促進する。その結果、作成したメタデータの流通が増えそれに伴いコンテンツの利用が増える。	メタデータ記述規則に関する知識習得。
コンテンツ提供者 (企業)	メタデータ情報基盤が未整備なため、各種メタデータの相互利用が困難。メタデータの相互連携による新サービスの提供は困難である。	メタデータの相互連携が容易となり、新たなサービスの提供が可能となる。	新サービスへの開発投資。
メタデータ情報基盤サービス提供者	メタデータ情報基盤のサービス提供者は現状存在しない。 メタデータのインフラとなる基盤情報の提供となる為、公共機関や非営利目的の研究会での運営が望ましい。	メタデータの相互運用性が向上し、メタデータ利用の高度化や新たな民間サービスが創出される。	メタデータ情報基盤サービスの普及推進のための活動。
メタデータ規則提供者	各機関毎に独自の方法でメタデータ記述規則を公開・提供している。広く統一的に公開するメタデータ情報基盤が未整備の為、メタデータ記述規則を公開しても利用されにくい状況である。	メタデータ記述規則を広く統一的に公開することが出来る。それに伴い、作成したメタデータが相互利用されやすくなることで、作成したメタデータの流通が増え、それに伴いコンテンツの利用が増える。	メタデータ記述規則に関する知識習得。 メタデータ記述規則に則ったメタデータ記述規則の作成。 メタデータ情報基盤への登録更新等のメンテナンス

◆ 図 1.6:プロジェクト実施後のメタデータ利用環境イメージ

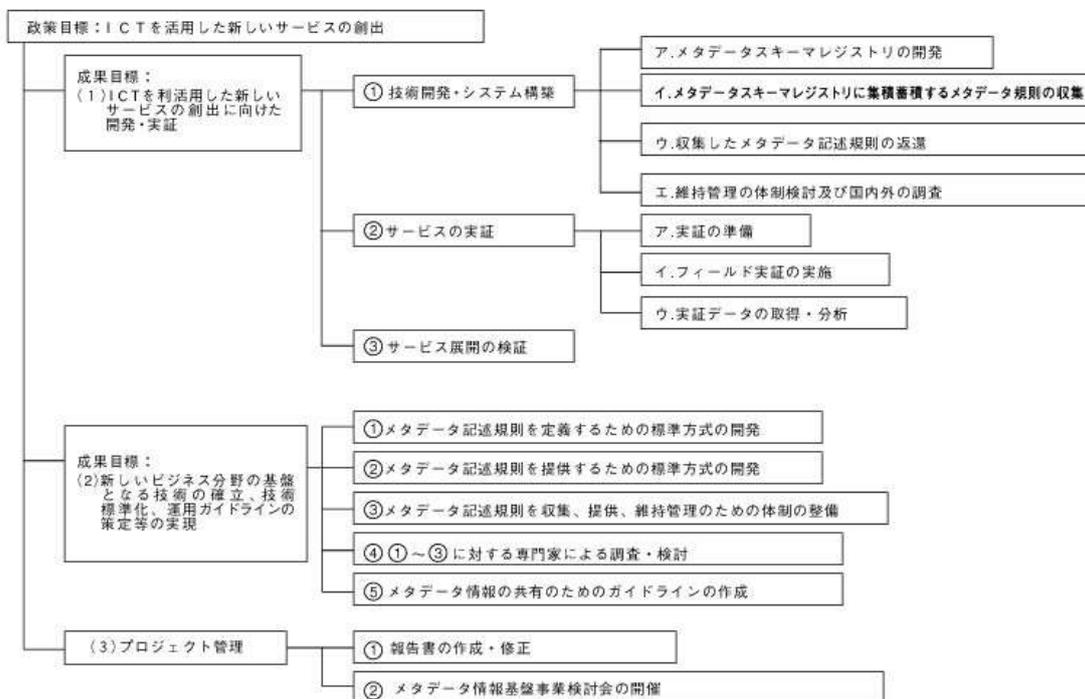
## 1.2.2 ICT を利活用した新しいサービスの創出に向けた開発・実証

### (1) 開発・実証の成果目標

メタデータ情報基盤システムの開発を行い、各種機関(図書館、公文書館、博物館、美術館等)のメタデータ記述規則を収集し蓄積管理した実証環境を構築する。その上でメタデータ記述規則に関する情報の公開・共有・維持管理、またそれを利用したメタデータ記述規則の定義や相互運用性について実証し、メタデータ情報基盤のサービス展開について検証する。

メタデータ情報基盤システムの開発にあたって、中核になる技術として、セマンティック・ウェブ技術とメタデータ・スキーマ・レジストリ技術を用いる。前者はメタデータの WWW 上での流通のための標準技術、後者はメタデータ記述規則に関する情報の蓄積共有サービスのための技術である。

こうした技術を利用し、収集したメタデータ記述規則のインターネット上での流通性を高めることで、既存のサービスの効率化や高度化を進め、新しいサービスの創出を促進する基盤作りを行う。また、メタデータ記述規則を保存するシステムの存在は、デジタルコンテンツ及びメタデータのメンテナンス性を高め、デジタルコンテンツの長期利用性向上にも同時に寄与することにもなる。



◆ 図 1.7:ICT を利活用した新しいサービスの創出に向けた開発・実証 実施イメージ

(2)開発・実証の実施内容

① 技術開発・システム構築

メタデータ情報基盤システムの中核を担うメタデータ・スキーマ・レジストリを開発する。レジストリとは「登録簿」を意味し、文字通り、様々な機関のメタデータ記述規則を蓄積・管理・公開するシステムを指す。

本システムにより、これまで様々な機関が独自に作成していたメタデータ記述規則を参照する事が可能となる。具体的な実施内容は以下の通りである。

ア.メタデータ・スキーマ・レジストリの開発

イ.メタデータ・レジストリに蓄積するメタデータ記述規則の収集

メタデータ情報基盤研究会の支援を受け、各種機関(国立国会図書館、国立公文書館、国立情報学研究所、東京国立博物館、東京近代美術館等を予定)のメタデータ記述規則を収集する。

ウ.収集したメタデータ記述規則の変換

RDF(Resource Description Framework)や DCAP(Dublin Core Application profile)等に基づいて、メタデータ記述規則をWWWでの流通に適した形式への変換(セマンティック・ウェブ化)

## エ. 維持管理の体制検討及び国内外の調査

目的	サービス利用者にとってサービスの使い勝手の検証、課題抽出 インターネット上での利用を前提としたメタデータ規則記述の検証、課題抽出		
仮説	<ul style="list-style-type: none"> <li>サービス利用者は、サービス利用前と比較してメタデータ記述規則の情報収集と解析に必要な時間が1件のメタデータ記述規則あたり32時間(1日8時間で4日と想定)短縮できる。</li> <li>サービス利用者が定義するメタデータの項目は、サービス利用前と比較して、項目の出自が明確化されることにより、相互運用性が高いものとなる。</li> <li>サービス利用者は、メタデータ記述規則の記述が標準化されることにより異なるメタデータ記述規則の理解が容易となる。</li> </ul>		
時期	平成23年2月～平成23年3月		
場所	主な開発場所: インフォコム株式会社及び株式会社インフォコム西日本 実証実験場所: インターネット空間(アクセスは主として筑波大学知的コミュニティ基盤研究センター)		
参加者	主体	一般モニター／関係者	人数(規模)
	コンテンツ提供者 (大学や図書館公文書館など)	関係者(メタデータ情報基盤研究会) 一般モニター	関係者(国立国会図書館、国立情報学研究所、筑波大学、国立東京博物館、国立近代美術館)(6機関、12名) 一般モニター(3機関、3名)
	コンテンツ利用者	関係者(メタデータ基盤研究会) 一般モニター	関係者(インフォコム株式会社、ナレッジシナジー株式会社、合資会社ゼノン)(3社、6名) 一般モニター(3機関、3名)

### ◆ 図 1.8: 実証作業の概要

## ② サービスの実証

### ア. 実証の準備

ホスティングサーバ環境を利用し、上記「①技術開発・システム構築」を経て開発したメタデータ情報基盤システムの実証環境を構築する。本実証環境は、より多くの機関による実証が可能となることを目的とするため、WWW 経由での利用を可能とする。但し、利用に際しては、ID とパスワードによる利用者認証を行いセキュリティに十分考慮したものとする。

### イ. フィールド実証の準備

主にメタデータ記述規則の提供機関、メタデータ情報基盤研究会への協力機関等より本システムの実証に関して協力を得る。

### ウ. 実証データの取得・分析

本事業の提案者(共同提案者含む)、実証協力機関による実証結果を取得し分析を行う。主にメタデータ記述規則に関する情報の公開、共有、維持管理の手法や利便性・操作性、実証環境を利用したメタデータ記述規則の構築手法、メタデータの相互運用性について実証する。

## ③ サービス展開の検証

上記②「サービスの実証」で得た結果をもとに、新しいサービスとしての事業化に向けた検討を中心にメタデータ情報基盤システムのサービス展開について検証を行う。

## 1.3 プロジェクト実施体制と活動

### 1.3.1 プロジェクトの実施体制

<プロジェクト実施体制図>

代表責任者:○杉本 重雄(筑波大学)

(1) ICT を利活用した新しいサービスの創出に向けた開発・実証

#### ① 技術開発・システム構築

##### ・ 担当

開発リーダー:岩杉 大輔(インフォコム株式会社)

開発・実証員:鳥越 直寿(インフォコム株式会社)

開発・実証員:栗原 浩 (インフォコム株式会社)

開発・実証員:河合 正樹(インフォコム株式会社)

開発・実証員:阿川 裕之(インフォコム株式会社)

開発・実証員:吉田 勇二(インフォコム株式会社)

開発・実証員:松本 幸夫(インフォコム株式会社)

開発・実証員:日高 政志(インフォコム株式会社)

開発・実証員:金野 義和(インフォコム株式会社)

開発・実証員:三須 信幸(インフォコム株式会社)

開発・実証員:加藤 昌也(インフォコム株式会社)

開発・実証員:小嶋 将士(インフォコム株式会社)

開発・実証員:竹中 勇人(インフォコム株式会社)

開発・実証員:小林 俊久(インフォコム株式会社)

開発・実証員:近藤 義照(インフォコム株式会社)

請負開発:株式会社インフォコム西日本

請負開発:合資会社ゼノン

請負開発:株式会社ナレッジ・シナジー

請負開発:ロジカルウェブ株式会社

請負開発:株式会社ジオ・ブレーン

#### ② サービスの実証

## ・ 担当

開発リーダー: 岩杉 大輔(インフォコム株式会社)  
開発・実証員: 鳥越 直寿(インフォコム株式会社)  
開発・実証員: 栗原 浩 (インフォコム株式会社)  
開発・実証員: 河合 正樹(インフォコム株式会社)  
開発・実証員: 阿川 裕之(インフォコム株式会社)  
開発・実証員: 吉田 勇二(インフォコム株式会社)  
開発・実証員: 松本 幸夫(インフォコム株式会社)  
開発・実証員: 日高 政志(インフォコム株式会社)  
開発・実証員: 金野 義和(インフォコム株式会社)  
開発・実証員: 三須 信幸(インフォコム株式会社)  
開発・実証員: 加藤 昌也(インフォコム株式会社)  
開発・実証員: 小嶋 将士(インフォコム株式会社)  
開発・実証員: 竹中 勇人(インフォコム株式会社)  
開発・実証員: 小林 俊久(インフォコム株式会社)  
開発・実証員: 近藤 義照(インフォコム株式会社)  
実証協力: メタデータ情報基盤研究会

## ③ サービス展開の検証

## ・ 担当

開発リーダー: 岩杉 大輔(インフォコム株式会社)  
開発・実証員: 鳥越 直寿(インフォコム株式会社)  
開発・実証員: 栗原 浩 (インフォコム株式会社)  
開発・実証員: 河合 正樹(インフォコム株式会社)  
開発・実証員: 阿川 裕之(インフォコム株式会社)  
開発・実証員: 吉田 勇二(インフォコム株式会社)  
開発・実証員: 松本 幸夫(インフォコム株式会社)  
開発・実証員: 日高 政志(インフォコム株式会社)  
開発・実証員: 金野 義和(インフォコム株式会社)  
開発・実証員: 三須 信幸(インフォコム株式会社)  
開発・実証員: 加藤 昌也(インフォコム株式会社)  
開発・実証員: 小嶋 将士(インフォコム株式会社)  
開発・実証員: 竹中 勇人(インフォコム株式会社)  
開発・実証員: 小林 俊久(インフォコム株式会社)  
開発・実証員: 近藤 義照(インフォコム株式会社)  
実証協力: メタデータ情報基盤研究会

(2) 新しいビジネス分野の基盤となる技術の確立、技術標準化、運用ガイドラインの策定等の実現

① メタデータ記述規則を定義するための標準方式の開発

・ 担当

開発リーダー: 杉本 重雄(筑波大学)  
開発・実証員: 永森 光晴(筑波大学)  
開発リーダー: 岩杉 大輔(インフォコム株式会社)  
開発・実証員: 鳥越 直寿(インフォコム株式会社)  
開発・実証員: 河合 正樹(インフォコム株式会社)  
開発・実証員: 阿川 裕之(インフォコム株式会社)  
開発・実証員: 日高 政志(インフォコム株式会社)  
開発・実証員: 金野 義和(インフォコム株式会社)  
開発・実証員: 近藤 義照(インフォコム株式会社)  
技術協力: メタデータ情報基盤研究会

② メタデータ記述規則を提供するための標準方式の開発

・ 担当

開発リーダー: 杉本 重雄(筑波大学)  
開発・実証員: 永森 光晴(筑波大学)  
開発リーダー: 岩杉 大輔(インフォコム株式会社)  
開発・実証員: 鳥越 直寿(インフォコム株式会社)  
開発・実証員: 河合 正樹(インフォコム株式会社)  
開発・実証員: 阿川 裕之(インフォコム株式会社)  
開発・実証員: 日高 政志(インフォコム株式会社)  
開発・実証員: 金野 義和(インフォコム株式会社)  
開発・実証員: 近藤 義照(インフォコム株式会社)  
技術協力: メタデータ情報基盤研究会

③ メタデータ記述規則を収集、提供、維持管理のための体制の整備

・ 担当

開発リーダー: 杉本 重雄(筑波大学)  
開発・実証員: 永森 光晴(筑波大学)  
開発リーダー: 岩杉 大輔(インフォコム株式会社)  
開発・実証員: 鳥越 直寿(インフォコム株式会社)

開発・実証員:河合 正樹(インフォコム株式会社)  
開発・実証員:阿川 裕之(インフォコム株式会社)  
開発・実証員:日高 政志(インフォコム株式会社)  
開発・実証員:金野 義和(インフォコム株式会社)  
開発・実証員:近藤 義照(インフォコム株式会社)  
技術協力:メタデータ情報基盤研究会

④ ①～③に対する専門家による調査・検討

・ 担当

開発リーダー:杉本 重雄(筑波大学)  
開発・実証員:永森 光晴(筑波大学)  
開発リーダー:岩杉 大輔(インフォコム株式会社)  
開発・実証員:鳥越 直寿(インフォコム株式会社)  
開発・実証員:河合 正樹(インフォコム株式会社)  
開発・実証員:阿川 裕之(インフォコム株式会社)  
開発・実証員:日高 政志(インフォコム株式会社)  
開発・実証員:金野 義和(インフォコム株式会社)  
開発・実証員:近藤 義照(インフォコム株式会社)  
技術協力:メタデータ情報基盤研究会

⑤ メタデータ情報の共有のためのガイドラインの作成

・ 担当

開発リーダー:杉本 重雄(筑波大学)  
開発・実証員:永森 光晴(筑波大学)  
開発リーダー:岩杉 大輔(インフォコム株式会社)  
開発・実証員:鳥越 直寿(インフォコム株式会社)  
開発・実証員:河合 正樹(インフォコム株式会社)  
開発・実証員:阿川 裕之(インフォコム株式会社)  
開発・実証員:日高 政志(インフォコム株式会社)  
開発・実証員:金野 義和(インフォコム株式会社)  
開発・実証員:近藤 義照(インフォコム株式会社)  
技術協力:メタデータ情報基盤研究会

(3) プロジェクト管理

① 報告書の作成・修正

- ・ 担当
  - 開発リーダー: 杉本 重雄(筑波大学)
  - 開発リーダー: 岩杉 大輔(インフォコム株式会社)
  - 開発・実証員: 鳥越 直寿(インフォコム株式会社)
  - 開発・実証員: 河合 正樹(インフォコム株式会社)
  - 開発・実証員: 阿川 裕之(インフォコム株式会社)
  - 開発・実証員: 日高 政志(インフォコム株式会社)
  - 開発・実証員: 金野 義和(インフォコム株式会社)
  - 開発・実証員: 近藤 義照(インフォコム株式会社)

## ② メタデータ情報基盤構築事業検討会の開催

- ・ 担当
  - 開発リーダー: 杉本 重雄(筑波大学)
  - 開発・実証員: 岩杉 大輔(インフォコム株式会社)
  - 開発・実証員: 鳥越 直寿(インフォコム株式会社)
  - 開発・実証員: 松本 幸夫(インフォコム株式会社)
  - 開発・実証員: 日高 政志(インフォコム株式会社)
  - 開発・実証員: 金野 義和(インフォコム株式会社)
  - 開発・実証員: 近藤 義照(インフォコム株式会社)
  - 検討会事務 開発・実証補助員(派遣会社 1 名)

### <各組織の役割>

	名称	役割
代表提案組織	筑波大学	プロジェクト全体の管理 成果目標「(2) 新しいビジネス分野の基盤となる技術の確立、技術標準化、運用ガイドラインの策定等の実現」に関するプロジェクト管理 メタデータ記述規則を定義するための標準方式の調査・検討・開発 メタデータ記述規則を提供するための標準方式の調査・検討・開発 メタデータ記述規則を収集、提供、維持管理のための体制の整備・調査・検討

		ガイドラインの作成
共同提案者	インフォコム株式会社	成果目標「(1)ICT を利活用した新しいサービスの創出に向けた開発・実証」に関するプロジェクト管理 全体プロジェクト管理の支援 技術開発・システム構築 サービスの実証 サービス展開の検証 メタデータ記述規則を定義するための標準方式の調査・検討・開発 メタデータ記述規則を提供するための標準方式の調査・検討・開発 メタデータ記述規則を収集、提供、維持管理のための体制の整備・調査・検討 ガイドラインの作成
共同提案組織	株式会社インフォコム西日本	技術開発・システム構築
共同提案組織	合資会社ゼノン	技術開発・システム構築
共同提案組織	株式会社ナレッジ・シナジー	技術開発・システム構築
共同提案組織	ロジカルウェブ株式会社	技術開発・システム構築
共同提案組織	株式会社ジオ・ブレン	技術開発・システム構築
協力組織	メタデータ情報基盤研究会	本事業に対する、技術支援、メタデータ記述規則の提供を行う。

## ② 委員会委員

名称:メタデータ情報基盤構築事業検討会

### (1) 目的

本事業の遂行に必要な知識・情報・意見等の収集、交換、検討を目的とする検討会である。メタデータ情報基盤構築事業検討会は、筑波大学が主催するメタデータ情報基盤研究会(※)の研究結果を活用する。また、国内外の有識者を招聘してのシンポジウムを実施する。

### (2) 開催期間・頻度

2010年11月～2011年3月

### (3) 開催場所

筑波大学 知的コミュニティ基盤研究センター会議室

又は インフォコム株式会社 会議室

(4) 構成員等

下表の通り。

	所属・役職	氏名	備考
座長	筑波大学 図書館情報メディア研究科教授	杉本 重雄	メタデータ技術に関する学識経験者
構成員	筑波大学 図書館情報メディア研究科講師	永森 光晴	メタデータ技術に関する学識経験者
	インフォコム株式会社 デジタルアーカイブシステム部	近藤 義照	共同提案のシステム開発業者
	インフォコム株式会社 デジタルアーカイブシステム部 システム営業グループ上級主任	松本 幸夫	共同提案のシステム開発業者
	インフォコム株式会社 デジタルアーカイブシステム部 テクニカルクリエイショングループ 課長	鳥越 直寿	共同提案のシステム開発業者
	インフォコム株式会社 デジタルアーカイブシステム部 テクニカルクリエイショングループ 副課長	岩杉 大輔	共同提案のシステム開発業者
	インフォコム株式会社 デジタルアーカイブシステム部 テクニカルクリエイショングループ 主任	日高 政志	共同提案のシステム開発業者
	株式会社ナレッジ・シナジー 代表取締役	内藤 求	共同提案のシステム開発業者 トピックマップに関する有識者
	合資会社ゼノン 代表	神崎 正英	共同提案のシステム開発業者 XML 技術、セマンティック・ウェブ技術に関する有識者
海外招聘	DCMI Chief Information Officer	Thomas Baker	メタデータ技術に関する

	(アメリカ)		る有識者
海外招聘	onebiglibrary.net (アメリカ)	Daniel Chudnov	メタデータ技術に関する有識者
海外招聘	コンサルタント (アメリカ)	Stuart Weibel	メタデータ技術に関する有識者
海外招聘	コンサルタント (オーストラリア)	Elizabeth (Liddy)Nevile	メタデータ技術に関する有識者
海外招聘	ニューヨーク大学図書館 (アメリカ)	Corey Harper	メタデータ技術に関する有識者
国内招聘	国立情報学研究所 情報学プリンシプル研究系 教授	武田 英明	メタデータ技術に関する有識者
国内招聘	国立大学法人 京都大学 地域研究統合情報センター 教授	原 正一郎	メタデータ技術に関する有識者
事務局	メタデータ情報基盤構築事業事務局		

### 1.3.2 プロジェクトの活動実績

#### (1) ICT を利活用した新しいサービスの創出に向けた開発・実証

##### ① 技術開発・システム構築

メタデータ情報基盤システムの中核を担うメタデータ・スキーマ・レジストリを開発

##### ② サービスの実証

ア. 実証の準備

イ. フィールド実証の準備

ウ. 実証データの取得・分析

##### ③ サービス展開の検証

上記②の実証結果をもとに、新しいサービスとしての事業化に向けた検討を中心とし、メタデータ情報基盤システムのサービス展開について検証

#### (2) 新しいビジネス分野の基盤となる技術の確立、技術標準化、運用ガイドラインの策定等の実現

##### ① メタデータ記述規則を定義するための標準方式の開発

様々な機関で利用できるメタデータ記述規則を定義するための標準方式を開発

##### ② メタデータ記述規則を提供するための標準方式の開発

蓄積したメタデータ記述規則を利用者(人およびコンピュータによる利用)へインターネットを通じて提供するための標準方式を開発。また、新たにメタデータ記述規則を作成する機

関に必要な提供方法や、メタデータ記述規則の作成支援の方法、コンテンツの長期利用のためにメタデータ記述規則を長期に安定して利用できるようにする維持管理の方法等を開発。

- ③ メタデータ記述規則を収集、提供、維持管理するための体制の整備  
メタデータ基盤情報システムが公開するメタデータ記述規則を長期にわたり継続的に収集、提供、維持管理するための体制整備。
- ④ ①～③に対する専門家による調査・検討  
①～③の開発段階および開発した結果について、専門家による調査・検討を行い本事業の開発に反映。専門家の選任はメタデータ情報基盤研究会の助言と協力を受け実施。
- ⑤ メタデータ情報の共有のためのガイドラインの作成  
前述の①～④の成果を元に、上記「(1) ICT を利活用した新しいサービスの創出に向けた開発・実証」の作業を実施した上で、メタデータ情報の共有のためのガイドラインを作成。

### (3)実施場所

- ① 主な開発場所  
筑波大学知的コミュニティ基盤研究センター(茨城県つくば市)  
およびインフォコム株式会社(渋谷区神宮前)
- ② 実証実施場所  
筑波大学知的コミュニティ基盤研究センター(茨城県つくば市)  
およびインフォコム株式会社(渋谷区神宮前)

## 2. メタデータとは何か

### 2.1 背景

電子出版、デジタルアーカイブ等、デジタルコンテンツの重要性が高まる中で、総務省・文部科学省・経済産業省による「デジタル・ネットワーク社会における出版物の利活用の推進に関する懇談会」が開催され、1) 広く国民が出版物にアクセスできる環境を整備すること、2) 国民の知る権利の保障をより確かなものとする、3) 知の拡大再生産につなげることを通して、デジタル・ネットワーク社会における出版物の利活用の推進に向けた検討が行われてきた。

民間においては出版物の電子書籍化が進み、大学・研究機関では論文等の学術系機関リポジトリが整備されつつある。また、国立国会図書館、国立公文書館等においては明治期以降の資料についてデジタル化を進めている。諸外国に比べるとまだまだ進んでいないと思われるが、このように、日本でもデジタルコンテンツを提供する基盤ができつつある。

### 2.2 メタデータとは

メタデータとは、一般的に「データに関するデータ(Data about Data)」あるいは「データに関する構造化されたデータ(Structured Data about Data)」と定義される。言い換えると、メタデータとは記述対象となる情報資源に関して、決められた属性についてその属性値を書き表したものである<sup>1</sup>。

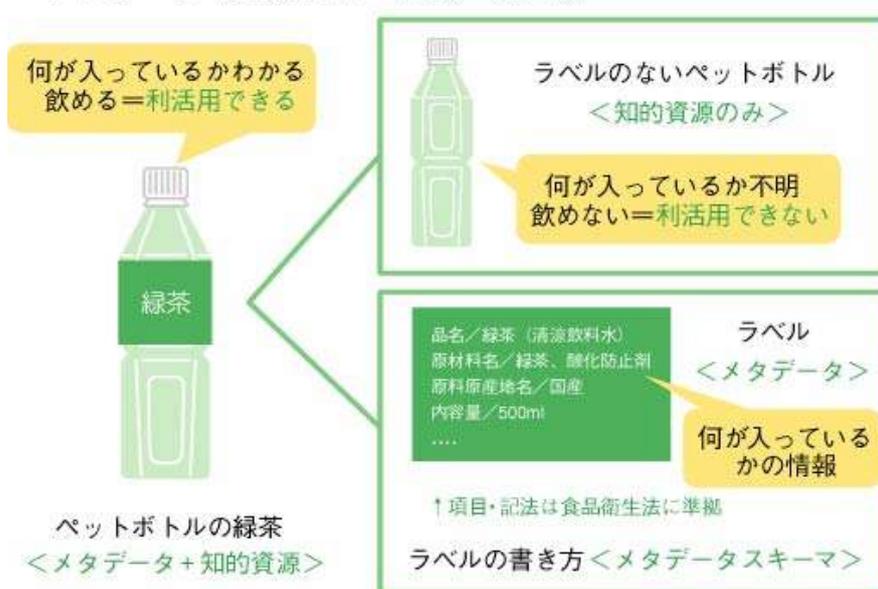
メタデータの説明をする際にペットボトルを例にとることがある。通常、ペットボトルには商品名や説明が印字されたラベルがついており、消費者はラベルを見て中身を判断している。そのラベルをはがしてしまうと、中に何が入っているかわからなくなる。何が入っているのかわからないと、人は飲むこと躊躇するだろう(図 2.1)

ここで液体の入ったペットボトルをデジタルコンテンツ(知的資源)としてとらえると、メタデータはそのラベルであり、ラベルの表記方法がメタデータ・スキーマであるといえる。デジタルコンテンツの場合、コンテンツに関する記述、すなわちメタデータがないとコンテンツが何であるのか、どのようにすれば閲覧できるのかわからない。ネットワーク上ではURIによってコンテンツへのアクセスは可能であっても現実には利用できない。

メタデータは、あらゆるところに存在する。スーパーマーケットの商品には、商品名、原産地、価格、重さなどが表示されている。おもちゃであればパッケージに製造社名、製造場所、対象年齢などが記載されているだろう。

店の看板などもメタデータである。看板があることによって、その店で何を売っているかを推測することが可能となる。商品の情報だけでなく、利用対象者、利用環境など、細かい属性をメタデータとして設定することで、商品がどういうものかがわかり、ユーザは商品を選択するのである。

<sup>1</sup>杉本重雄, 「Dublin Core の現在」, デジタル図書館 No. 36, Mar. 2009



◆ 図 2.1: メタデータはペットボトルのラベルであり、ラベルの表記方法がメタデータスキーマ

●メタデータ例  
 <ペットボトル>  
 <品名>緑茶(清涼飲料水)</品名>  
 <原材料名>緑茶、酸化防止剤</原材料名>  
 <原料原産地名>国産</原料原産地名>  
 <内容量>500ml</内容量>  
 </ペットボトル>

## 2.3 メタデータの活用と相互運用の重要性

### 2.3.1 デジタルコンテンツにおけるメタデータ

情報化社会においてメタデータはさらに重要なものとなる。これまでは商品を手にとって、商品に関する情報を見るだけであった。しかしインターネットが普及した現在、商品を手にとって情報を見るのではなく、ユーザがほしい商品を検索することからスタートする。

買い物する場合、ユーザは価格.comで最安値を調べ、Amazonや楽天といったオンラインストアを利用する。もちろん購入者による評価・クチコミも購入する際に考慮される。こういった購買行動において事前にフィルタリングがなされるため、商品情報、価格情報、評価といったメタデータがさらに重要性を増すのである。例えば子供向けの電子書籍をさがそうとした場合に、大きく分けても、1) 探すべき本の内容 2) 電子書籍の利用環境 3) 利用対象者 というように多様なメタデータが必要となる。

メタデータが整備されることによって、デジタルコンテンツ商品も細かい単位で選択されるようになる。たとえば、itunesでは一曲単位で楽曲を購入することが可能である。また、Reference-tree<sup>2</sup>では、教科書を1章ごとに購入することが出来、ユーザはほしい情報を探せるだけでなく、ほしい情報のみを購入することが出来るようになる(図 2.2)。



◆ 図 2.2: Reference-tree (リファレンストリー)

### 2.3.2 メタデータ利用の現状

デジタルコンテンツを実際に探す場合には、現在、Googleなどの検索エンジンの利用が中心である。検索エンジンを用いて検索されるウェブ上の情報は実に多様であり、信頼性に欠ける情報も多々存在する。多くのデータがあるにもかかわらず、より信頼度の高い情報を得るためには、国立国会図書館のホームページ上の目録データなど個別のデジタルアーカイブが持つデータに頼るなど、各分野のサイトにおいて専用の検索フォームを利用するしかない。

このようにそれぞれの領域・コミュニティに応じた探し方で情報を利用するだけでは、動物園のデータベースと植物園のデータベース、水族館のデータベースがあったとしても、例えば日本に生息する動物、植物、魚といったものを同時に探すことは検索することはできない。こういった多様なニーズ、領域を超えて横断的に検索するニーズがあるのは確かである。

### 2.3.3 メタデータの相互運用に向けて

今後インターネット上の情報量が増大していくなかで、特に高品位なメタデータを流通させること

<sup>2</sup>Save Money-Just Buy Your Chapters Online, <http://www.reference-tree.com/>

が、ユーザの利便性を高め、コンテンツの差別化につながっていくと考えられる。

(a) **相互運用と個別の要求** 新たなサービスを創出する上で、メタデータを相互利用可能にすることによって得られるメリットは大きい(「16.メタデータ共有がもたらす可能性」にて詳述)。その一方、どこでもつかえる共通のメタデータ記述規則を作ることは現実的とは言えない。

メタデータ記述規則は、本質的に目的の領域における要求要件に応じて決められるべきものである。相互運用が求められる場合であっても、領域毎に求められる記述項目は準備すべきであり、むやみに相互運用のための項目の縮退や意味の変更を行ってはならない。したがって、ネットワーク上のメタデータ規則には、個別の要求を満たすことと相互運用性を高めることという相互に矛盾する要求を満たすことが求められる。

(b) **メタデータの信頼性** また、領域を超えて信頼性の高い検索を行っていくためには、信頼性の高いメタデータが必要である。検索エンジンのように機械的に創りだすメタデータも多いが、一般的に高度に意味的な内容を含むメタデータは人の手によって作らざるを得ない。そのためには図書館目録がこれまで行ってきたように、メタデータがこういったポリシーでこういった記述のされ方をすべきなのかという、記述のための規則・ガイドラインが必要となる。

たとえば「件名」「タイトル」「題目」すべて日本語で記した場合、コンピュータによってこれらが同じ意味の項目であるということを自動的に判断することは容易ではない。一方、こういった記述項目を集めたデータベースを作成して、情報を蓄積していくことで、これらが同一のものであると認識させ、共通部分を増やしていくことが可能となる。

(c) **メタデータ情報の共有と再利用** 本プロジェクトでは、こういったメタデータに関する情報をネット上で共有できるように、再利用できるようにしたいと考えている。一定のメタデータ規則を流通させることが出来れば、これからメタデータを作る人たちは、既存のものを再利用することができるようになる。

もちろんメタデータをカスタマイズして利用することもあれば、新たなメタデータ項目を設定することもあり得るだろう。しかし主要なメタデータ規則をネット上で再利用することで、おのずと応用毎のメタデータ規則の間に共通部分が出てきて、相互運用がしやすくなる。

データベース同士の相互運用はスキーマが同じでないとやりづらいが、記述項目をできる限り独自につくらず、既存のものを再利用することで、互換性を向上していくことは可能であろう。こういった基本的なメタデータ相互運用の考え方が今後重要となっていくと考えられる。

## 2.4 メタデータ情報とスキーマ

### 2.4.1 メタデータを共有・利用するための情報

メタデータは一般に

- ・**識別**:どんな資源があり、それが具体的に何を指すのかを、他のものと区別して示す
- ・**記述**:現物を見なくてもそれが何であるかが分かるような概要を記述する
- ・**発見**:多数の資源の中から、特定のもの、あるいは条件に合致するものを見つける手段を提供する
- ・**入手**:求める資源がどこにあるか、どうやったら入手できるかの手段を示す
- ・**管理**:資源の構造や管理上の特徴等を示し、資源の維持管理のための情報と提供する

という目的のために作られる。個々のコンテンツ保持者が、自らの資源を管理・検索する場合は、メタデータは組織内で共有・理解できればよい。しかしコンテンツを流通させ、多角的な利用を可能とするためには、メタデータも外部から利用でき、理解可能でなければならない。

メタデータを第三者が利用・理解できるようにするためには、メタデータを公開するだけでなく、それがどのような規則で記述されているかも明示する必要がある。

### 2.4.2 メタデータ記述規則の基本要素

メタデータを複数の担当者が分担して作成する場合、一貫したメタデータの作成のためには、記述の規則を定める必要がある。この規則は、記述に用いる項目と項目値のための統制語(メタデータ語彙)の定義の上に、それぞれの記述項目は必須か任意かなどといった取り決め、すなわちメタデータの構造的制約を明示的に定義する。さらに、メタデータを実現するには、何らかの具体的実現形式を決めて、システム化する。

このように、メタデータ記述規則は

- (1) メタデータ語彙 (プロパティの語彙、クラスの語彙の総称、「2.4.3 メタデータ語彙」で詳述)
- (2) 構造的制約 (属性値記述の省略可能性や繰返条件などメタデータの構造的な制約)
- (3) 具体的表現形式 (システム上でのメタデータの具体的表現形式)

からなっている。さらに、メタデータの実際の記述の際に、(4)記述者に与えられるべきメタデータ記述のガイドラインが必要である。大まかに言って、メタデータ記述規則はこれら(1)～(4)で成り立っているといえる。

### 2.4.3 メタデータ語彙

メタデータを記述し、相互に理解可能とするためには、メタデータ記述に用いる項目名や情報タイプ名(語彙)を明確にする必要がある。

(a) **プロパティ**    **メタデータの記述対象が持つ特徴の一つに名前をつけて表現するもの。**

メタデータは対象の何らかの特徴を捉えて記述することで、対象の何たるかをコンパクトに表現する。したがってメタデータの作成のためには、まずどのような特性(タイトル、作者、作成日、主題など)によって対象を表現するかを定義しなければならない。この特性は、表形式カタログの項目、データベースのカラムあるいはフィールドに相当する。これはまたRDF(「4.2.データモデルの標準化:RDF」を参照)では「**プロパティ**」と呼ばれる。

(b) **クラス**    **共通の特徴を持つメタデータ記述対象の分類に名前をつけたもの。**

メタデータの記述対象にはいろいろな対象(これを情報資源、あるいは資源、リソースと呼ぶ)がある。情報資源には、書籍、人物などいろいろなものがあるので、それらを分類、もしくは種類分けすることが必要である。この分類は、一般に、汎化・特化関係等を用いて体系化される。この分類のことを「**クラス**」と呼ぶ。たとえば、ある図書を対象とした記述対象の場合、その記述対象のクラスは「**図書**」であると考えることができる。

プロパティやクラスは語として定義される。メタデータを記述するために必要なプロパティとクラスの集合をそれぞれプロパティの語彙、クラスの語彙と呼ぶ。これらを総称してメタデータ語彙と呼ぶ。

メタデータ語彙は、何らかの目的で用いるメタデータ、あるいは何らかの領域のコンテンツを記述することを念頭に置いて定義されることが一般的である。たとえば住所録を作成するためには「郵便番号」「住所」「氏名」「電話番号」という項目(プロパティ)を定めて「住所録記述語彙」を定義することになる。また、Simple Dublin Core と呼ばれる標準では、タイトルや作成者などの 15 項目からなる語彙を定めている。

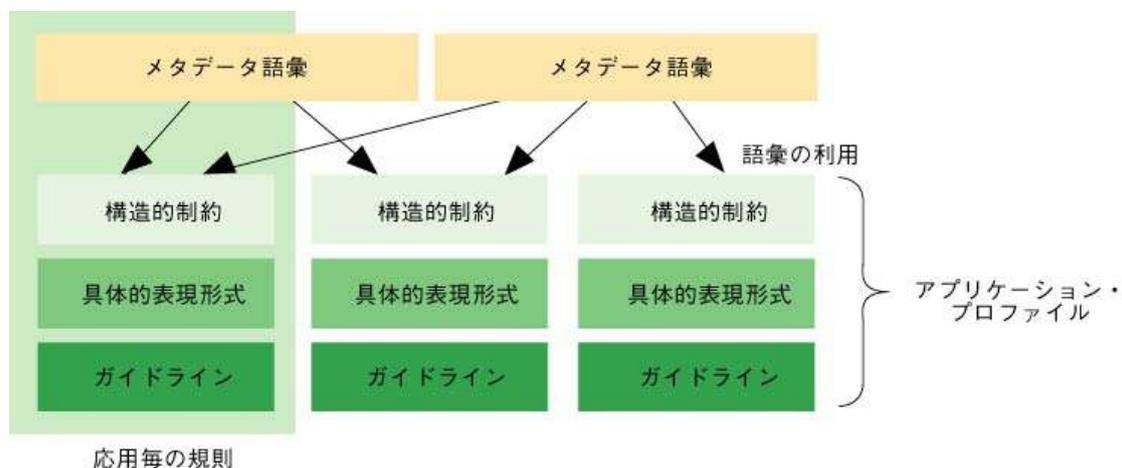
メタデータ語彙の標準化動向については、「5.各分野でのメタデータ記述の現状」(特に「5.3 メタデータ記述の代表的語彙」)で詳述する。

### 2.4.4 メタデータ語彙とアプリケーション・プロファイル

先に述べたように、メタデータ記述規則は(1)メタデータ語彙、(2)構造的制約、(3)具体的表現形式、そして(4)記述上のガイドラインからなり、これらは応用毎(直す語彙のアプリケーションとしての規約毎)に決められるべきものである。その一方、メタデータ語彙を応用間で共有することにより、メタデータの相互運用性が高まる。

そのため、(1)～(4)の内、(1)と(2)以降を分離してメタデータ記述規則を階層的にとらえることで、語彙の共通性を高められることになる。すなわち、既存の標準的な語彙をできるだけ利用し、応用

に特化した語彙の必要性を最小化しやすくなる。また、(2)～(4)に関しても、新たに規則を作る場合に既存のものを参考にしやすい、規則間の違いを明確にとらえることができるために規則間の相互運用の可能性を的確に判断しやすい、といった長所を持つ。



◆ 図 2.3:メタデータスキーマの階層構造

本プロジェクトでは、以上のように、メタデータ記述規則のうち、メタデータ語彙とそれ以外の要素を分離してとらえる(図 2.3)。これは Dublin Core Metadata Initiative が提案した概念に基づくものであり、(1)をメタデータ語彙、(2)以降を「**アプリケーション・プロファイル(Application Profile)**」と呼ぶことにする。(アプリケーション・プロファイルについては、「6.2.1.アプリケーション・プロファイル」で改めてとりあげる。)

本報告書では、独自あるいは標準の語彙、およびアプリケーション・プロファイルを総称して、メタデータ記述の「スキーマ」と呼ぶこととし、メタデータ記述規則を構成する(1)～(4)のうち(2)構造的制約、および(3)具体的表現形式を中心として報告を行うものとする。

(a) **スキーマの標準化** メタデータ・スキーマの多くは、データベースごと、組織ごとに定義されている。しかし住所録、書誌カタログなどのように同じ情報を各地で記述し、交換できることが望ましいものは、標準的な語彙、規則を定義して公開、共有する場合が多い。

標準的な語彙の例としては、住所録データをアドレス帳アプリケーションで交換するための VCard、人に関わる記述項目を決める FOAF (Friend of a Friend)、情報資源の発見と記述のための Dublin Core などがある。また、学術機関リポジトリで用いられるメタデータ・スキーマや国立国会図書館がデジタル情報資源向けに開発したメタデータ・スキーマなど、標準的なスキーマもある。

メタデータを作成する場合、独自の語彙や規則を定めるよりもこうした標準スキーマを用いるほうが、作業面からも相互運用性の面からも望ましいといえる。そのためには、目的にあった標準スキーマを容易に見出すための手段、そしてそれを実際に利用する際の指針が必要である。

標準的スキーマの作成や発見を支援するレジストリの開発については「第 3 部 新しいサービスの創出に向けた開発と実証」で、また指針となるガイドラインについては「7.メタデータ情報共有のためのガイドラインの策定」で詳しく報告する。

### 3. メタデータ情報の基盤とニーズの現状調査

#### 3.1 アンケート調査

本事業では、メタデータに関する現状についてより詳細に把握するために、公的機関や民間企業等を対象に随時調査を行っている。以下は、平成22年末から平成23年1月中旬にかけて行ったアンケート調査の結果である。

##### (1)メタデータ・スキーマ提供に関するアンケート

調査対象： 博物館、公文書館、研究機関等

調査期間： 平成22年12月27日(月)～平成23年1月13日(木)

##### 主な質問内容と回答

質問内容 回答者	提供可能な メタデータ・ スキーマが あるか？	何を対象とし たものか	記述形式	運用手順書あ るいはガイドラ インの有無と公 開状況	サンプルの 提供可否	サンプルの 一般公開 可否
国立情報学 研究所①	ある	書誌	CATP 形式	目録情報の基 準 第4版 ・コーディングマ ニュアル ・その他のマニ ュアル等	可	可
国立情報学 研究所②	ある	論文書誌、論 文著者情報	HTML ページと して公開 ・論文詳細表示 の RDF につい て ・著者詳細表示 の RDF につい て	なし	可	可

質問内容 回答者	提供可能な メタデータ・ スキーマが あるか？	何を対象とし たものか	記述形式	運用手順書あ るいはガイドラ インの有無と公 開状況	サンプルの 提供可否	サンプルの 一般公開 可否
国立情報学 研究所③	ある	機関リポジ タリのメタデー タをハーベス ト(収集)する ため	XML Schema 形 式 ・文書形式の日 本語版 ・英語版	junii2 ガイドラ イン(バージョン 1.0)を公開	可	可
国立情報学 研究所④	ある	研究プロジェ クト、研究者 情報	文書	なし	可	可
国立公文書 館	ある	歴史公文書 等	国立公文書館 EAD 定義(第 1.07版)	国立公文書館 EAD 定義(第 1.07版)	可	可
農林水産研 究情報総合 センター	ある	図書書誌、文 献書誌	RDF 形式、 XML Schema 形 式	ガイドライン有 り、一般にも公 開されている	可	可
国立国会図 書館	ある	書誌	RDF 形式 「国立国会図書 館ダブリンコア メタデータ記述 (DC-NDL)」	Application Profile 内に、 RDF/XML によ る記述例あり。	可	可
東京国立近 代美術館	ある	収蔵品	Excel で提供予 定	ない(ただし、非 公開ながら作 品管理システム の業務マニユア ルはあり)	可	不可
東京国立博 物館	ある	人文系ミュー ジアムの収蔵 品	RDF 形式	Web サイトにて 公開	不可	不可

## (2)メタデータ・スキーマ提供に関するアンケート

調査対象： 民間企業・団体

調査期間： 平成22年12月20日(月)～平成23年1月7日(金)

質問	A社(情報・通信関連)回答	B社(図書館関連)回答	C社(出版関連)回答
組織内でメタデータ を利用しているか、 または利用経験があ るか	ある	ある	ない
(利用がある場合) 利用データと内容	商品 ID, 製作者 ID, メディア 種別, 販売形態, 販売情報, 書誌情報, 目次データ, ファ イル形式, 版数, など	書誌情報をオンライン書店で 使用	—
組織内で公開・流通 を目的としたメタデー タを作成しているか	作成している	作成している	—
(作成している場合) その内容	システムへの入稿、ストアで の公開を目的に作成。他の 目的への利用は、現状、考 慮していない。	書誌データを作成・販売。現 在ではそのノウハウも活かし 電子書籍のメタデータも作成 し、ハイブリッド型総合書店 への提供を開始。また他に 典拠・目次・雑誌データも販 売している。	—
メタデータ・スキーマ は公開しているか、 またはその可能性が あるか	予定はない	公開している	—
(公開の場合)その 内容と公開方法	—	販売しているため、利用者 には PDF で提供。	—
(公開しない場合) その理由	現状は実験的な作成・利用 の段階であり、外部に公開で きる内容となっていない。将 来的に、事業としてのメリッ トがあると判断すれば外部に		—

	も公開する可能性はある。		
メタデータ情報構築 基盤事業についての 感想	メタデータ記述規則が体系化・標準化されることで、業者間で流通するメタデータの作成頻度が減少し（流用が可能となり）、費用の削減につながると共に、利用する側もレジストリ等により利便性の向上につながると思う。	（無回答）	書籍の情報はいろいろなところに散らばっており、集めての活用が困難であったため、非常に期待できる。
レジストリ活用の可能性についての見解	コンテンツ入稿時のメタデータ生成のガイドラインやチェックツールの作成などに活用することができるのではないかと思う。	（無回答）	誰でも簡単に活用できるのであれば、発展する可能性は非常に高いと思う。
今後のメタデータ利用に関する展望	コンテンツの内容や書誌情報的なメタデータだけでなく、権利関係の記述やクリアランスに活用可能なメタデータが標準化され、DRM などの互換性が確保されることで、コンテンツ業界全体の発展に寄与できると思う。また、このような取り組みが、出版物だけでなく他のメディアにも拡大されると尚有益と考えている。	（無回答）	商品（書籍）の販売のため、各社のもつ書誌情報などを上手く活用しプロモーションに役立てていきたいと考えている。

### 3.2 シンポジウム招聘者会議

平成 22 年 12 月 8 日、本事業の一環として、内外の専門家を招いて国際シンポジウムを開催した。メタデータを取り巻く現状課題と将来展望を探るためのシンポジウムで、約 70 名の一般参加者も交えて議論を交わした。

#### (1) 基調講演

シンポジウムでは、開会にあたり主催側より筑波大学大学院 図書館情報メディア研究科 教授 杉本重雄氏、来賓として総務省 情報流通行政局 情報流通振興課 統括補佐 松田昇剛氏が挨拶。インフォコム株式会社 デジタルアーカイブシステム部 鳥越直寿氏によるプロジェクト概要説明を経て、以下の基調講演が行われた。

<メタデータ・レジストリについて(Introduction to Metadata Schema Registries)

永森光晴氏(筑波大学大学院 図書館情報メディア研究科)>

永森氏は、DCMI Registry WG に参加し、DCMI Metadata Schema Registry の開発に携わった経験を持つ。講演の趣旨は以下の通り。

メタデータ・スキーマやメタデータの相互運用性・再利用性の向上に関心を持つメタデータ・レジストリは各種情報を流通させるためのハブとなる位置づけである。複数の場所に格納されているコンテンツを、所蔵場所に関係なく相互的に運用する際にはメタデータが必須であり、これを支えるのがメタデータ・レジストリとなる。

それぞれのコミュニティに合わせて作成されたメタデータを、機関や組織の違いを越えて利用したいというニーズは、ネット上で共有することで実現する。

メタデータ・レジストリでは、

- ① 収集・蓄積したメタデータ・スキーマをネット上で提供することで、再利用・共通化を促す。
- ② WWW の標準形式(RDF、TopicMaps)を活用することで流通性を高める。

信頼できるメタデータ・スキーマを誰もが簡単に調べることができるようにするには、人だけではなくコンピュータ(ソフトウェア)を対象に含め、メタデータを利用したサービスを容易に構築できる環境を第三者に提供する必要がある。メタデータ変換やマッシュアップを可能にする環境の構築も重要である。

- ⑤ メタデータ・レジストリに蓄積する必要があるのは以下の要素である。

- ア. メタデータ語彙(メタデータ記述項目の語彙。統制語彙)
- イ. Application Profile(構造制約、データ交換形式、メタデータ作成ガイドライン等)
- ウ. その他のメタデータ関連情報

また、必要な機能や条件等は以下の通り。

- ア. メタデータ・スキーマの登録、管理、検索、表示機能
- イ. 複数スキーマ間の関連性調査
- ウ. WWW の標準形式の利用環境
- エ. サービス提供者(第三者として)の高度なメタデータ活用(メタデータ・スキーマ調査、メタデータ変換支援、問い合わせ API)

(2) 国内外招聘者による講演

基調講演に続き、メタデータならびにインターネット上の情報流通の専門家5名による講演が行われた。Thomas Baker、Daniel Chudnov、Corey Harper、Liddy Nevile、武田英明の各氏から、それぞれの専門分野の視点から見た課題の指摘や提言があり、質疑応答も活発に行われた。

① Dublin Core: Linked Data の環境における基盤的メタデータ(Dublin Core: basic metadata in a linked data environment)

Thomas Baker 氏(Dublin Core Metadata Initiative)

② 4.2 Web における Linked Data(Linking Library Data in the Web)

Daniel Chudnov 氏(onebiglibrary.net)

③ Metadata Registries: 語彙の管理と流通サービス(Metadata Registries: Vocabulary Management and Dissemination Services)

Corey Harper 氏(New York University)

④ 複数の領域にまたがるメタデータの利用: ローカルな要求への特化とグローバルな相互運用性(Cross-domain metadata use: local specificity and global interoperability)

Liddy Nevile 氏(La Trobe University)

Linked Data における日本の課題(Challenges for Linked Data in Japan)

武田英明氏(国立情報学研究所)

(3) パネルディスカッション

杉本氏をモデレーターに、永森氏および招聘者5氏をパネリストとしてパネルディスカッションを行った。事前に会場に質問用紙を配布しており、その結果を基にディスカッションが進められた。また、会場との質疑応答も実施した。

冒頭では、杉本氏から「メタデータは情報のソース(対象資源)を明確化させることが重要。ディープ Web(検索エンジンにヒットしないサイト, 情報等)では特に顕著」との指摘があった。参加者からは、「メタデータ間の同一視の方法をどうするのか」といった概念的な質問から、「粒度の異なるメタデータの取り扱いはどうすればよいのか」「組織として MLA 連携においてメタデータをどう取り扱うべきか」といった具体的な運用の質問まで広範囲にわたる質問が寄せられた。これに対し、パネリストらはそれぞれの事例等の紹介を交え、活発な意見が交わされた。

#### (4) 参加者の声

国際シンポジウム会場では、一般参加者に向けてアンケートを実施した。以下は寄せられた回答からの抜粋・要約である。

・メタデータの相互運用、統合的な利用を進めることで、何が実現できるのか。メリット・具体的利用例、長期的運用と世界とのつながりについて考えたい。

・Linked Data の「信頼性」について、粒度の異なるデータの統合は可能なのか。基礎となるメタデータの確立が重要ではないか。

・異なるソースのエレメント同士が同じであると判断する基準について知りたい。

・Linked Data が広まるほど玉石混交となり、結局は現状の Web と同じになるのでは。検索によりコンテンツが多数ヒットすればするほど混乱する。適切にフィルタリングする方法が必要ではないだろうか。

・公共機関が出すデータは信頼性が認められ、共有が進む一方で、小さな機関はつぶれていく…といった二極分化を危惧している。

・Application Profile を作るためのツールに興味がある。

・例えば e book におけるリンク(パスとしての ID)に適したものは何か。

・Linked Data ではない状態から Linked Data に変換するために何をすべきか考える必要がある。

・identity の問題を議論すべき。

- ・記述規則について、制度化・標準化・規格化は必須である。
- ・クラウドコンピューティングの方向を考えると、集中と分散のバランスや重みが重要になると思う。二極化し過ぎても問題なので、その意味でメタデータ(DB)が中継になるように感じる。特に「誰が何のために情報を使うのか」を明確にしないと、専門家や興味を持つ人以外には使われないのではないか？

- ・特許、公共政策、人権、倫理などの難解さが指数関数的に増加しているのではないだろうか。

#### (5) 付記事項

プロジェクト自体は今年度末までのものであっても、メタデータ情報基盤は「より長期の視点に立って継続的に作り上げていかねばならないもの」という認識が、一般参加者も含めた会場の総意と言える。なお、本シンポジウムに関する詳細な資料は、以下の URL にて公開している。

本事業運営サイト <http://meta-proj.jp/ev-1.html>

### 3.3 現状のまとめ

ICT の専門家も、公的機関や民間企業の現場の人々も、そしてシンポジウムに集まった一般参加者も、メタデータの必要性とそれが生み出す可能性については意見が一致するところである。メタデータ情報の基盤構築と利活用へのニーズは極めて高いことは、本事業に対して各方面から寄せられた声を見れば一目瞭然と言える。

公的機関や民間企業では活用も進んでいるが、その一方で、それは分野や組織毎の壁の中でのことであり、公に共有することの難しさも浮き彫りとなっている。しかしながら、この壁を打破しない限り、メタデータの利用性や相互運用性、コンテンツの流通性の向上はないと言える。それは、産学官民の情報が遮断されたままとなることを意味し、社会・文化、経済・産業、そして家庭生活にまで及ぶ ICT の恩恵を十分に受けることができないという危惧に晒されることになる。

その半面、各種アンケートやヒアリングの結果からは、メタデータにまつわる知識の吸収意欲や積極的な問題提起姿勢が見られた。ICT を活用した新しいサービスの創出への期待感は極めて高いだけに、メタデータ情報基盤の構築を急ぐことが何よりも重要であろう。

## 第2部 基盤となる技術の確立・標準化とガイドライン

### 4. データ表現方式の標準化動向

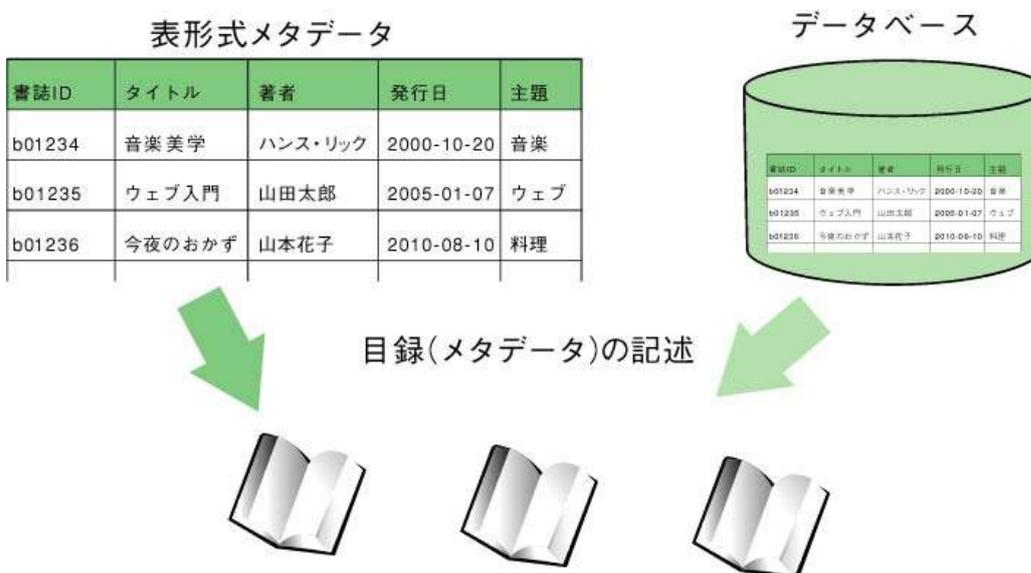
さまざまなメタデータを交換・共有するためには、まずデータの表現方法(基底データモデル<sup>1)</sup>の標準的な枠組みが求められる。メタデータ共有の基盤となる技術として、まずこのデータ表現方法を標準化する流れの背景と動向を概観する。

#### 4.1 構造化記述と共通語彙の模索

ネットワークの発達、特に WWW の普及によって、さまざまな情報が公開され、広く入手可能になった。HTML によるウェブページの記述だけでなく、XML を用いたデータの構造化記述が進み、領域を超えた相互運用のために共通の語彙が模索されることになる。

##### 4.1.1 公開メタデータとマーク付け

メタデータはコンテンツの保有者がリソース管理のために作成し、表やデータベースの形で保持されてきた(図 4.1)。



◆ 図 4.1:表形式、DB によるメタデータ

1989 年に WWW が誕生し、さまざまな文書が HTML で公開されるようになると、コンテンツを保有する組織もその紹介をウェブページで行なうようになる。もっとも、多くの場合はコンテンツを人間の読者向けに紹介することが目的であるため、公開されるメタデータは部分的で、HTML のテーブルなどで整形するのみであった。ここからメタデータを収集するためには、HTML を解析して

<sup>1</sup>個々のデータ構造を、共通の方法で表現し計算処理可能とするための、記述の要素とその意味論を定義するデータモデル。

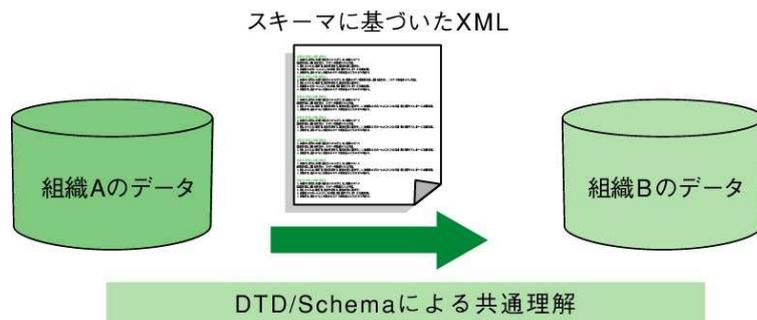
データ構造を推測、抽出する「スクリーン・スクレイピング」が行なわれた。しかし、ページの構造は見た目を優先するものが大半で、提供組織ごとに HTML の記述法もまちまちであったため、メタデータの共有は非常に困難であった(図 4.2)。



◆ 図 4.2:ウェブページと、スクリーン・スクレイピングによるデータ収集

#### 4.1.2 XML による構造化記述

データを構造的に記述するマーク付けとしては、1980 年代から SGML が用いられてきたが、WWW 誕生以前の規格であり、仕様が複雑で組織ごとに部分的な実装が用いられるなど、インターネットでメタデータを公開、共有するためには使いにくい面が多かった。そこで W3C において、SGML のサブセットとして仕様をコンパクトにし、ウェブ上での利用を念頭に置いたマーク付け仕様の XML が 1998 年に規格化、勧告された。



◆ 図 4.3:スキーマに基づくXMLによりデータの相互運用性が高まる

XML では、組織のニーズに応じたタグを定義し、その構造制約を DTD や XML スキーマとして明示できる。これを用いることで、HTML では十分表現できなかったデータ構造を的確に示し、コンピュータ処理可能なメタデータを、相互運用可能な形で公開できるようになった(図 4.3)。

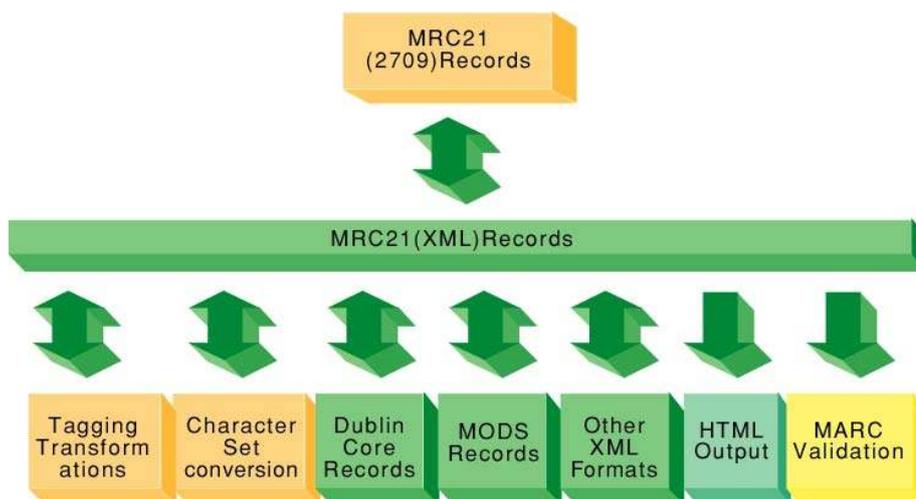
(a) **領域の標準XML** XMLのデータは、DTDもしくはスキーマを参照することで互いに確実な交換が可能だが、個々の組織がそれぞれ独自のスキーマを定義していると、多数のソースからのデータ統合が複雑になる。

そこで、業界や領域、あるいは目的ごとに、標準的なスキーマを定義して、データを共通化する動きが出た。企業間の情報交換のための ebXML や RosettaNet、財務諸表などのビジネスレポートを電子化する XBRL、データ放送のための BML などがある例である。

(b) **書誌レコードとXML** 古くは粘土板や巻物にまでさかのぼる図書館の書誌は、冊子体目録、カード型目録を経て、1960年代にはコンピュータ処理可能なMARC(MAchine-Readable Cataloging)となった。

MARCは磁気テープ時代のデータ形式が基本にあり、特有のフォーマットを用いているため、データの変換や多目的利用にコストがかかる。そこで MARC データを XML で表現する MARCXML が 2003 年に提唱された<sup>2</sup>。また、並行して開発された MODS (Metadata Object Description Schema) は、MARC の数値タグではなく titleInfo、name、originInfo など分かりやすい名前をタグに用い、要素をグループ化して階層構造を導入した<sup>3</sup>。

MARCXML、MODS の構造は XML スキーマで定義されている。ツールを用いて容易にほかのフォーマットに変換可能であり、領域を超えたメタデータ交換のための重要な一歩となった(図



◆ 図 4.4: MARCXML を介して、MARC レコードはさまざまなフォーマットと相互変換できる

<sup>2</sup> MARC 21 XML Schema, <http://www.loc.gov/standards/marcxml/>

<sup>3</sup> Metadata Object Description Schema:MODS(Library of Congress), <http://www.loc.gov/standards/mods/>

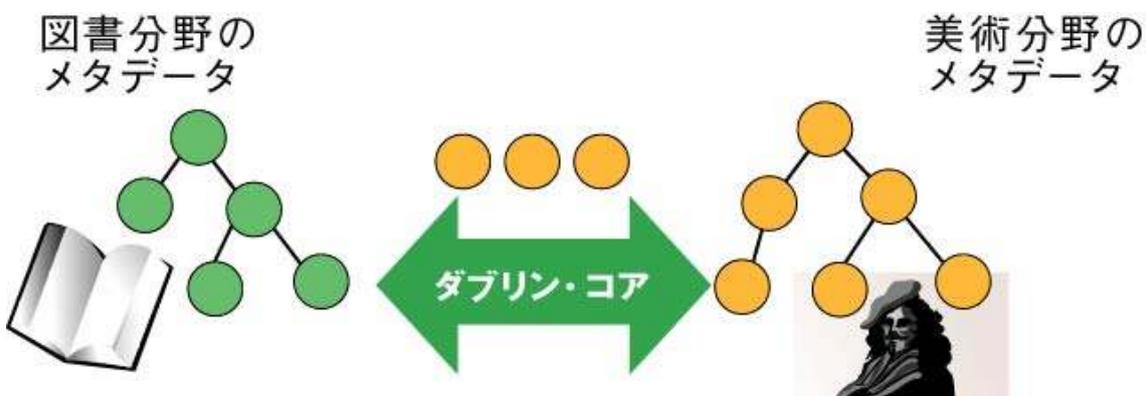
4.4)。

(c) **共通の語彙と枠組みの必要性** こうしたXML応用により、領域特有のデータ交換は標準化が進んだ。しかし、領域を超えて流通するコンテンツのメタデータ標準としては、まだ十分とはいえない。データをAPI経由で取得できるサービスも増えたが、サービスごとに異なる語彙、フォーマットを用いたメタデータが提供され、コンテンツ・メタデータの交換や再利用は個別対応が必要な状況である。

領域を超えたメタデータの共有のためには、分野をまたいで共通に利用可能な語彙、そして個別のデータ構造に特化せずに交換を可能とする基底モデルの枠組みが必要である。

#### 4.1.3 標準語彙ダブリンコア

1995 年 3 月に、図書館、コンテンツ、マーク付けなどの分野の専門家が米国オハイオ州のダブリンに集まって、「ネットワーク情報資源の記述、組織化、発見、アクセス」のための方法と標準を検討するワークショップが開催された。一連のワークショップでは、できるだけ幅広い領域のリソースを記述できる「コア」なメタデータ要素、**ダブリンコア**(DC, Dublin Core)の策定が目指され、その成果として Dublin Core Metadata Element Set(DCMES)が定義された(「5.3.2.ダブリンコア」参照)。



◆ 図 4.5: 異なる領域のメタデータと、橋渡しとなるダブリンコア

DCMES は、さまざまな分野に共通するプロパティ(データ項目)を目録専門家でも使えるように整備し、分野を越えた意味の相互運用性(semantic interoperability)を得ることを狙いとする。そのため要素はシンプルで基本的(コア)なものに限定しており、必要ならこの共通項を基盤にして拡張ができる柔軟性を提供した。

領域やサービスごとに異なるメタデータも、ダブリンコアにマッピングすることで、最小限の相互運用性を持たせることができる。こうして語彙のレベルでは、交換のための基礎となる標準は用意された。

## 4.2 データモデルの標準化：RDF

ダブリンコアのような共通の語彙を用いてメタデータを交換するためには、そのデータモデルの記述方法も共通化する必要がある。現在メタデータ記述の標準として広く支持されている RDF の概要と背景を説明する。

### 4.2.1 メタデータ記述フォーマットの源流

ウェブ・コンテンツのメタデータを記述する標準フォーマットには、2つの源流が考えられる。

そのひとつは、コンテンツ内容に未成年に不適切なものが含まれているかどうかを記述し、フィルタリングを可能にしようという試みである。1996年10月にW3C勧告となったPICS(Platform for Internet Content Selection)規格は、コンテンツ格付けの語彙を定めるのではなく、いつ(on)誰が(by)何を(for)どう格付けしたか(ratings)といった「記述の枠組み」のみを定めた。

```
(PICS-1.1 "http://www.gcf.org/v2.5" labels
  on "1994.11.05T08:15-0500"
  exp "1995.12.31T23:59-0000"
  for "http://www.greatdocs.com/foo.html"
  by "George Sanderson, Jr."
  ratings (suds 0.5 density 0 color/hue 1))
```

PICSはHTMLのmeta要素での記述を念頭においていたが、この「枠組みとしてのメカニズムを定義する」という考え方が、メタデータ一般を表現するモデルに発展する。

もうひとつの源流に、コンテンツの概要をHTML本体とは別に流通させて、新しい情報を素早く提供するという考えがある。つまりコンテンツ・メタデータを流通させるための標準フォーマット案である。ニュースのチャンネルやサイトマップを記述するためのフォーマットMCF(Meta Content Framework)が、AppleやNetscapeの技術をベースに開発され、1997年6月にW3Cに提案された。

これらのメタデータ表現技術をもとに、リソース一般の記述方法の枠組みとして1999年にW3Cから勧告されたのが、Resource Description Framework(RDF)である。さらにその後の実装経験などを踏まえて、RDFで記述された情報の厳密な意味論を含めた改定が2004年に勧告され、メタデータ記述方式の国際的な標準となっている<sup>4</sup>。

<sup>4</sup> RDF Current Status, <http://www.w3.org/standards/techs/rdf>

#### 4.2.2 対象の記述とトリプル

RDF は、リソースについての記述を主語(対象)—述語(記述項目)—目的語(項目値)の三つ組み(トリプル)として表現するモデルである。トリプルは、主語リソースと目的語リソースを述語をラベルとした矢印で結ぶ「有向ラベル付グラフ」として表される<sup>5</sup>(図 4.6)。

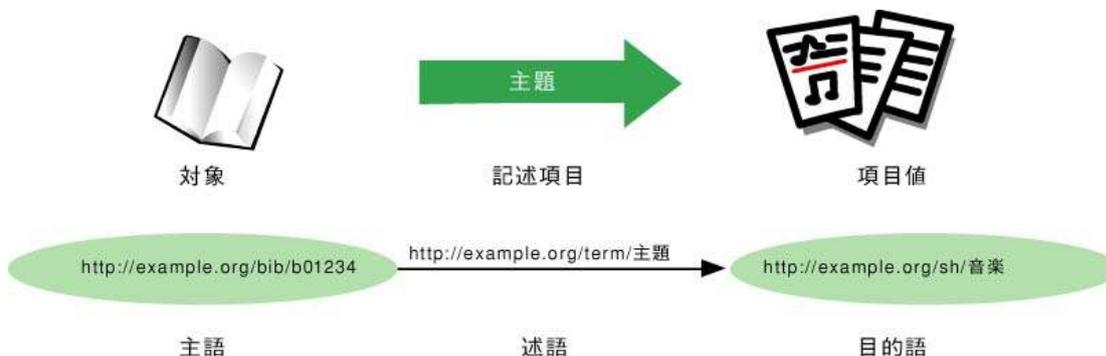
RDF のトリプルの集合を「グラフ」と呼ぶ。また、グラフ中の述語を「プロパティ」とも呼ぶ。

RDF によるリソースの記述は、一つのデータベース内で保持されるだけでなく、ウェブを通して広く共有、交換されることを念頭においている。そのため、トリプルとして記述する主語、述語、目的語は、グローバルに識別が可能な形で表現されなければならない。RDF はこれらの表現に Uniform Resource Identifier (URI) を用いる<sup>6</sup>。

URI、特に http:スキームによる URI はすでに WWW でのウェブページアドレスとして広く利用されており、多くのアプリケーションが処理可能という利点がある。またドメイン名システムとの組み合わせにより

- ・ウェブ全体で、衝突の恐れのない一意の識別子を提供する
- ・ユーザが任意の識別子を自由に作成できる分散型システム

という 2 つの重要な機能を持ち、どんな組織においても、確実にグローバルに一意な識別子を与えることができる。

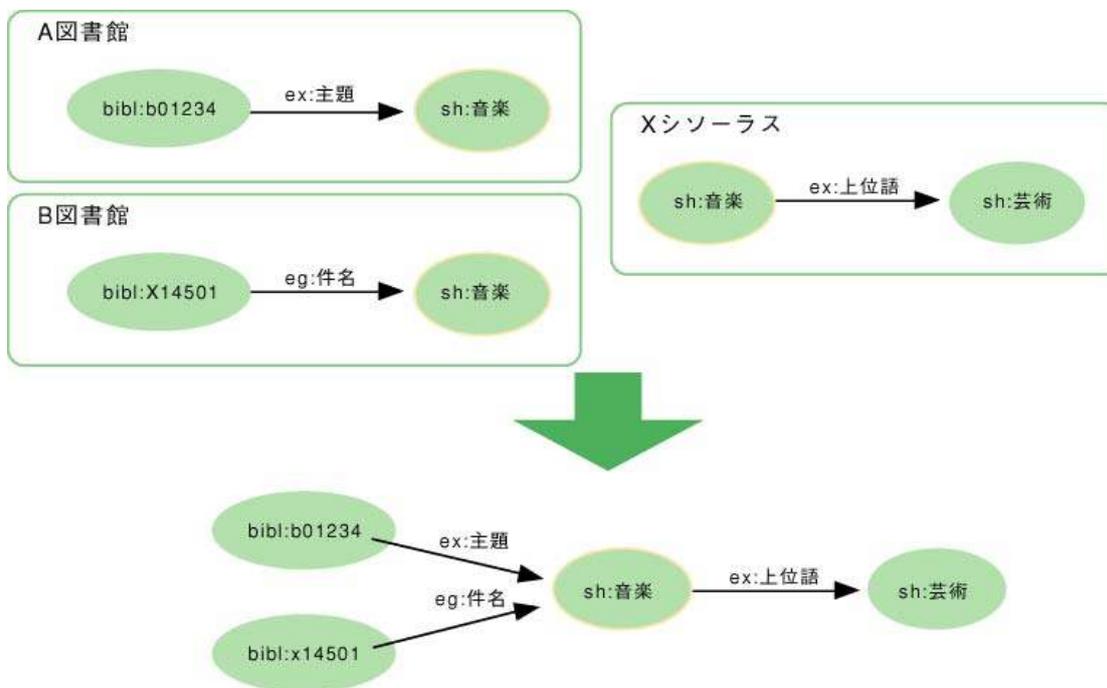


◆ 図 4.6: RDF は主語—述語—目的語トリプルでリソースを記述する

<sup>5</sup> この円と矢印のグラフにおいて、円(主語、目的語)を「ノード」、矢印(述語)を「アーク」とも呼ぶ

<sup>6</sup> ここでの URI は、多言語文字に対応する Internationalized Resource Identifier (IRI) と同等の意味で用いている。

グローバルに一意であるということは、同じ URI を用いて名前付けされたリソースは、同一とみなせることを意味する。従って、異なる組織で別々に記述された RDF グラフであっても、共通の URI を介して連動させることが可能となる(図 4.7)。



◆ 図 4.7: RDF グラフは同じ URI を介して連結(併合)することができる(以下「<http://example.org/term/>主題」のような URI を、XML 修飾名を用いて「ex:主題」のように短縮表記する)

(a) **リテラルと空白ノード** またRDFトリプルの目的語は、URI以外にリテラル値を持つことができる。これは、URIのように何らかのリソースを名前付けして参照する値ではなく、その値文字列そのものがリソースであるタイプの目的語である。たとえばある人の「名前」プロパティ(述語)の目的語として「山田太郎」があるとき、この値は'山' '田' '太' '郎'という文字の並び自身であって、文字列を介して何か別のものを表しているわけではない。リテラル値は、URIと違ってグローバルな識別ができず、主語や述語に用いることはできない。

さらに、トリプルの主語と目的語は、URIで名前付けしない匿名の“あるリソース”とすることができる。通常これを「空白ノード」と呼ぶ。空白ノードは、数式における変数とよく似た役割を果たす。

### 4.2.3 URI とリソース

RDF の記述対象となる「リソース」は、URIによって名前付け(識別)される。ウェブ文書のようにネットワーク経由でアクセスできるものに限らず、書籍、人物、概念など、名前付け可能なものすべてをリソースとして扱うことができる。

(図 4.6)において、主語となる書籍を「<http://example.org/bib/b01234>」と URI で表しているが、

この URI を辿っても書籍そのものがネットワークで送られてくるわけではない。また、目的語となる主題件名を「http://example.org/sh/音楽」としているのも同様で、「音楽」のような具体的な形を持たない概念も URI で表現される。

主語と目的語の関係(項目名=プロパティ)も、「http://example.org/term/主題」という URI となる。項目名を URI で表現することで、異なる組織の項目名を的確に共有したり区別したりできる。さらに、項目名 URI を主語にした RDF を記述することで、項目名に関する説明や、異なる組織の項目名同士の関係をも説明することが可能になる。

たとえば、A 図書館の「主題」項目と B 図書館の「件名」項目が、同じ意味を表しているということを、項目名同士の関係として表現できる(図 4.8)。



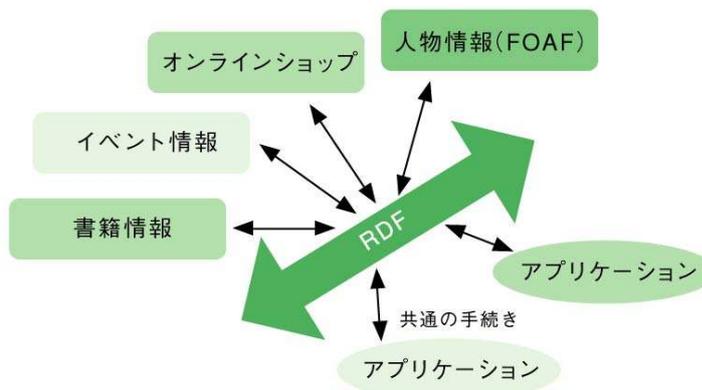
◆ 図 4.8: RDF は関係(述語)も URI で名付けるので、述語間の関係を記述することもできる

#### 4.2.4 RDF パス

RDF のグラフは柔軟なモデルを表現できるので、表形式データはもちろん、関係データベースの複数テーブルに格納した複雑なデータであっても、RDF による記述に変換することができる。いったん RDF に変換したデータは、URI によって記述対象やデータ値を集約することができ、また項目名(プロパティ)間の関係も定義できるので、データの交換が容易になる。

異なる組織間でのデータ交換のためには、それぞれのデータ項目や構造のマッピングが必要である。この交換が、1 対 1 ではなく複数対複数になると、必要なマッピングは幾何級数的に増加する。

RDF は多様なデータを柔軟に表現でき、URI を介してデータを集約したり関連付けることができる。これを利用し、各組織のデータ交換の中間項として RDF を用いれば、変換マッピングは各組織と RDF の間だけでよい。RDF がデータの交換と共有のための橋渡しを行なう。RDF はデータ流通の共通基盤である「パス」として機能するのである(図 4.9)。



◆ 図 4.9:異なる組織やアプリケーションのデータを交換・共有するための「バス」として RDF が機能する

#### 4.2.5 RDF の構文

RDF はデータのモデルをグラフとして表現するもので、それをプログラム処理できる情報として記述する構文は、一つに限定されていない。しばしば用いられる構文としては、グラフを XML の木構造に翻訳する形で記述する RDF/XML 構文<sup>7</sup>、より端的に主語・述語・目的語を列挙する形で記述する Turtle 構文<sup>8</sup>がある。

本報告書で RDF の例を示すときは、主として Turtle 構文を用いて記述する。

#### 4.3 RDF スキーマと OWL

RDF による記述において、各記述に用いる語彙を定義したり推論を可能にするための仕組みが導入されている。

独自に記述したデータを URI を介して結び付けて利用するとき、データをつなぐ述語(プロパティ)の意味や、データのグループ(クラス)を第三者も理解できるようにする必要がある。これらの「語彙」の定義も RDF を用いて行なわれる。

##### 4.3.1 RDF スキーマ

RDF スキーマ(RDF Schema)<sup>9</sup>は、記述対象の分類を可能にする「クラス」と、記述の述語に用いる「プロパティ」を名前付けして定義する。一つのファイルなどでまとめて定義されるクラス、プロパティの総体を「語彙」(Vocabulary)と呼び、語彙に属する個々のクラス、プロパティを「ターム」と

<sup>7</sup> RDF/XML Syntax Specification (Revised), <http://www.w3.org/TR/rdf-syntax-grammar/>

<sup>8</sup> Turtle - Terse RDF Triple Language, <http://www.w3.org/TeamSubmission/turtle/>

<sup>9</sup> RDF Vocabulary Description Language 1.0: RDF Schema, <http://www.w3.org/TR/rdf-schema/>

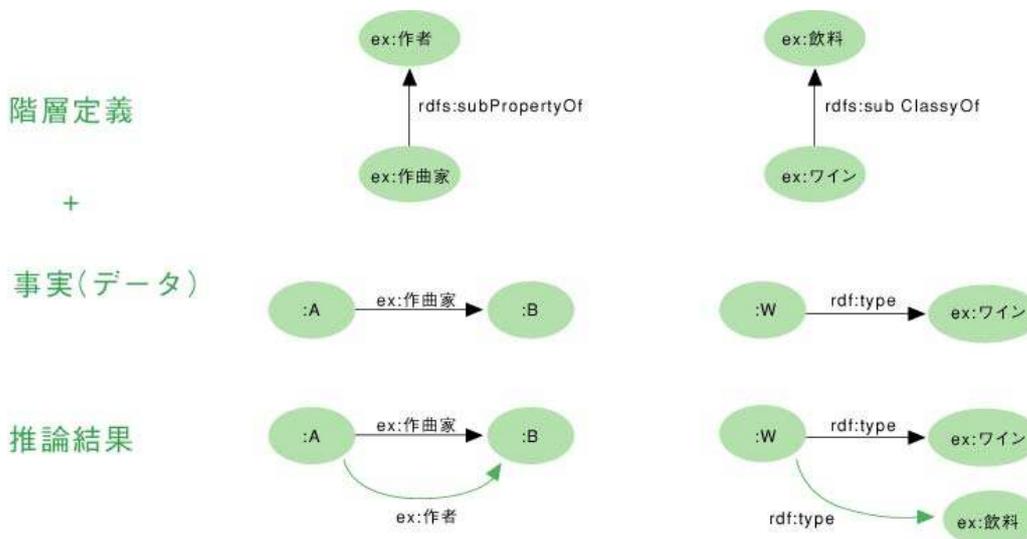
呼ぶ。

一般にこれらのタームの名前は、語彙全体を表す名前空間 URI と、語彙の中で個々のタームを識別するローカル名を結合した URI として与えられる。また RDF スキーマでは、タームを URI で名前付けするだけでなく、人間の利用者向けの簡易名(ラベル)と説明(コメント)を、それぞれ `rdfs:label`、`rdfs:comment` を用いて記述できる。

さらに RDF スキーマでは、クラス、プロパティの階層的定義、およびプロパティとクラスの関係定義を行なう。これらは XML スキーマのような制約の定義ではなく、メタデータ自身に直接記述されない暗黙の関係を導く情報として用いられることが大きな特徴である。

(a) **階層的語彙定義** たとえば、「作曲家」と「作者」というプロパティを階層関係として(「作曲家」は「作者」の《サブプロパティ》である)定義すると、「AはBの作曲家である」というメタデータから、暗黙的に「AはBの作者である」という関係を導くことができる。同様に、「ワイン」と「飲料」というクラスを階層関係として(「ワイン」は「飲料」の《サブクラス》である)定義すると、「Wはワインである」という記述から「Wは飲料である」という関係を導くことができる(図 4.10)。

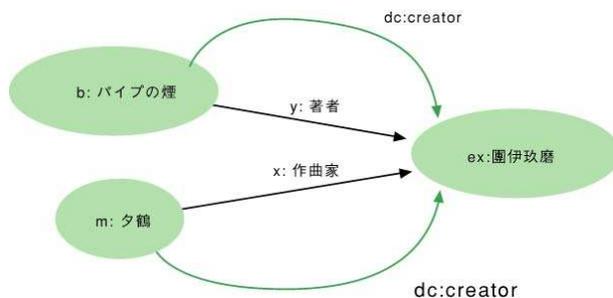
サブプロパティ、サブクラス関係は、それぞれ RDF スキーマの `rdfs:subPropertyOf`、`rdfs:subClassOf` を用いて記述する。



◆ 図 4.10: プロパティ、クラスの階層関係を用いて、データから未知の事実を推論できる

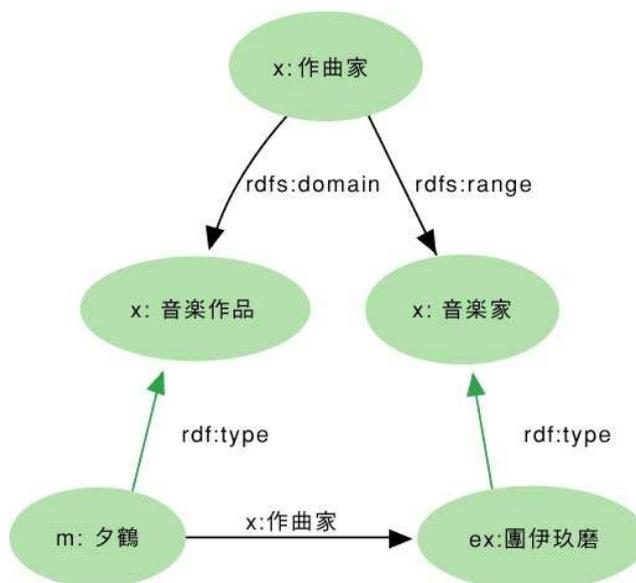
あるクラスのメンバーである(クラスに属する)リソースを、クラスの「インスタンス」と呼ぶ。「W は飲料である」という関係は、この用語を用いて「W は飲料クラスのインスタンスである」と表現する。インスタンスの関係は、`rdf:type` を用いて記述する。

(b) **スキーマと横断検索** 4.2.3 で述べたように、RDF ではプロパティもURI で名前付けされるため、この関係記述は異なる領域の語彙間でも定義できる。これを利用することで、異なる領域で定義されるメタデータ・スキーマを関連付け、複数の情報源からのメタデータを相互変換したり連動させることが可能になる。たとえば「x:作曲家」「y:著者」などの異なる語彙のプロパティをダブリンコアの「dc:creator」のサブプロパティとして関連付けておけば、音楽メタデータと書籍メタデータを、同じ項目名を用いて横断検索できる(図 4.11)。



◆ 図 4.11 サブプロパティ関係を用いて異なるメタデータを同じ検索項目で横断検索できる

(c) **定義域と値域** RDFスキーマでは、プロパティの「定義域」「値域」も定義できる。あるプロパティを用いてトリプルを記述したとき、前者は主語リソースのクラスを、後者は目的語リソースのクラスを暗黙的に示す。例えば、「x:作曲家」の定義域を「x:音楽作品」、値域を「x:音楽家」と定義すると、「A x:作曲家 B」というトリプルから、Aは音楽作品、Bは音楽家である(インスタンスである)と理解できる。これを利用することで、様々な情報源から集めたメタデータに記述されているリソースを、自動的に分類することが可能になる(図 4.12)。



◆ 図 4.12: 定義域、値域を用いて、リソースのクラス(タイプ)を推論できる

定義域、値域はそれぞれ `rdfs:domain`、`rdfs:range` を用いて定義する。

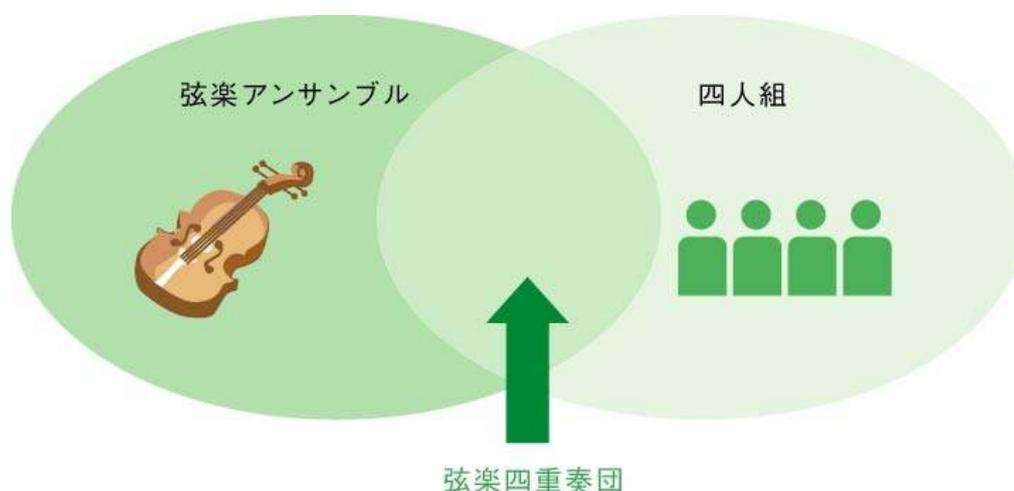
プロパティの値域は、メタデータの記述規則における項目値の制約条件と対比させることができる。値制約は項目値として許される値の規則、RDF の値域は目的語(項目値)の属するクラスを推論するものと役割は異なるが、いずれもプロパティの持つ値に関する情報を提供する定義という点で共通する。

#### 4.3.2 OWL

論理学や人工知能の研究で発達してきたオントロジー(領域知識の体系的記述)や述語論理に基づく推論をウェブにも応用するため、RDF とオントロジーを組み合わせる方法が検討されてきた。この成果として 2004 年に W3C から勧告されたのが、OWL(Web Ontology Language)である<sup>10</sup>。

OWL は、RDF スキーマよりさらに詳細なクラス定義と、プロパティの論理的性質の定義を行なう。また語彙定義の「オントロジー」自身のメタデータを記述し、相互のインポートやバージョン管理するための機能も導入する。

(a) **クラスの組み合わせ** OWL では、既存のクラスの組み合わせによって新たなクラスを定義することができる。たとえば「弦楽アンサンブル」というクラスと「四人組」というクラスがあらかじめ定義されているとして、その積集合(共通部分)を利用して「弦楽四重奏団」クラスを定義するという具合である(図 4.13)。



◆ 図 4.13: 弦楽アンサンブルと四人組の共通部分として弦楽四重奏団クラスを定義できる

(b) **プロパティ制約によるクラス定義** また OWL のクラスは、プロパティがどのような条件を持つか(プロパティ制約)という視点でも定義できる。たとえば、「演奏する」というプロパティについて、その値を「弦楽器」とする主語リソースは「弦楽器奏者」、値を「ピアノ」とする主語リソースは「ピアニスト」という形で、プロパティの使われ方によってクラスを定義する。

<sup>10</sup> OWL Web Ontology Language Current Status, [http://www.w3.org/standards/techs/owl#w3c\\_all](http://www.w3.org/standards/techs/owl#w3c_all)

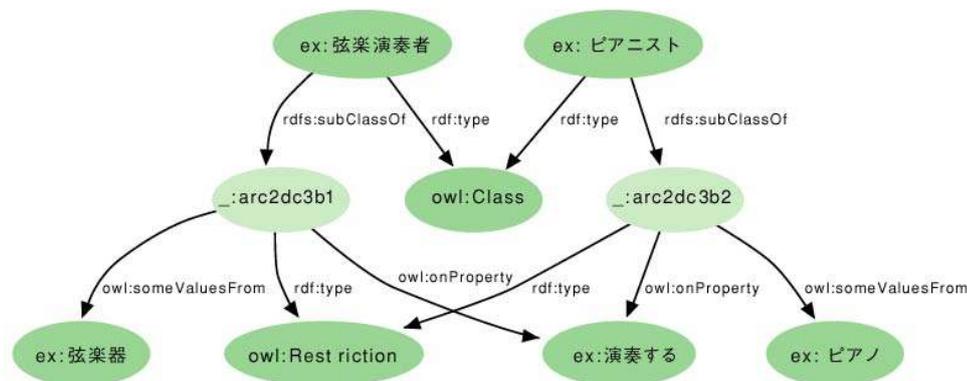
```

ex:弦楽器奏者
  a owl:Class ;
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty ex:演奏する ;
    owl:someValuesFrom ex:弦楽器
  ] .
    
```

```

ex:ピアニスト
  a owl:Class ;
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty ex:演奏する ;
    owl:someValuesFrom ex:ピアノ
  ] .
    
```

これは、OWL で用意されている owl:Restriction を用いて「owl:onProperty で示すプロパティの値が少なくともひとつは owl:someValuesFrom クラスのインスタンスである」という制約クラスを表し、対象クラス (ex:弦楽器奏者、ex:ピアニスト) がその制約クラスの下位クラスであると表現することでクラスに属する条件を示している。この RDF グラフを図示すると図 4.14 のようになる。



◆ 図 4.14: プロパティ「演奏する」の制約条件によってクラスを定義する

値の制約ではなく、プロパティの出現回数による制約で定義する場合は、owl:cardinality を用いる。

```

ex:四人組
  a owl:Class ;
  rdfs:subClassOf [
    
```

```
a owl:Restriction ;  
  owl:onProperty ex:メンバー ;  
  owl:cardinality 4  
]
```

この例は、「四人組」は「メンバー」プロパティが 4 回出現する（メンバーが 4 人いる）という特徴で定義できることを示している。最小、最大出現回数を制約する場合は、owl:cardinality に代えて owl:minCardinality、owl:maxCardinality を用いる。

RDF スキーマでの値域は、主語リソースに関係なくすべての目的語を一定のクラスに属するものとして定義するのに対し、OWL のプロパティ制約は、より柔軟で詳細な定義方法を提供している。

(c) **推論と制約記述** プロパティ制約によるクラス定義は、基本的には推論（ピアニストが A を演奏する → A はピアノであるといえる）を行なうものであるが、視点を変えてこれを検証のための制約（ピアニストが A を演奏する → A がピアノでなければ誤り）として利用することもできる<sup>11</sup>。これを応用すると、メタデータの記述規則を OWL のプロパティ制約をもつクラスとして表現することで、メタデータを規則に照らして検証することが可能となる。

(d) **オントロジー管理** OWL では、オントロジー自身を owl:Ontology クラスのインスタンスとして表現し、そのバージョンの記述 (owl:versionInfo)、古いバージョンへの参照 (owl:priorVersion)、後方互換性があること (owl:backwardCompatibleWith) もしくはないこと (owl:incompatibleWith) をヘッダに記述してバージョン管理を行うことができる。また、オントロジー自身のファイルの所在地 URI を owl:versionIRI で記述できる。

これらは、メタデータ・スキーマのバージョン管理や、メタデータの長期保存を実践する上で、非常に重要な情報を提供するためのツールとなる。

#### 4.4 その他のデータモデル「トピックマップ」

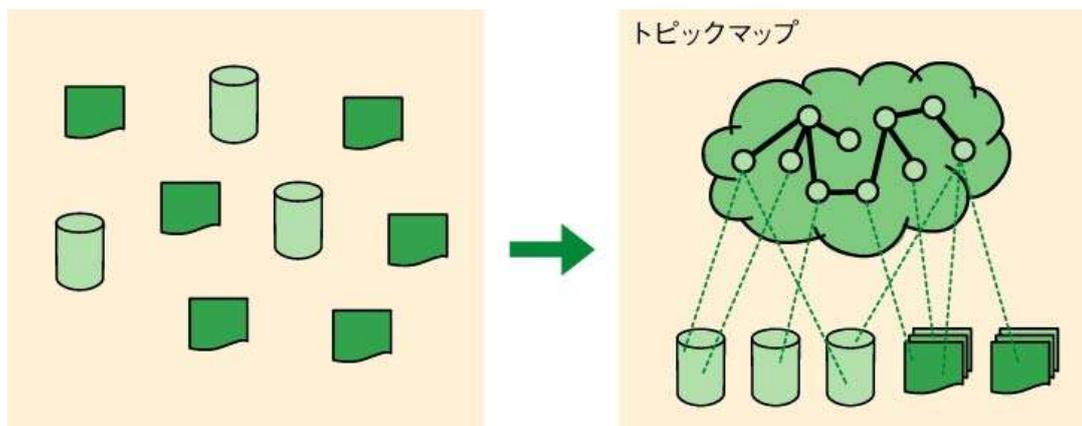
トピックマップの起源は、1991 年の Davenport グループでの作業だとされている。トピックマップの国際標準化は、当初、ISO/IEC JTC1/SC18/WG8 で、そして、現在では、ISO/IEC JTC1 SC34/WG3 で検討されている。

トピックマップは、Library science (図書館学) から多くを学んでいる。トピックマップは、コンピュータ・ネットワーク上の情報空間に対する索引の一種に例えることができる。

<sup>11</sup>ウェブのような未知の情報があり得る環境ではなく、閉じた世界で情報が既知であるという前提の上で検証することになる。

#### 4.4.1 トピックマップの概要

(a) **トピックマップ** トピックマップは、情報/知識を利用者の持つ概念体系に合わせて分類、整理、体系化、組織化し見つけやすくする。サイバースペースにおいて高度に構造化された索引のような役割をする。知的活動をこれまでよりも人に優しく支援することを目指す情報技術である。図 4.15 にそのイメージを示す。トピックマップは、情報実体から独立しており、それ自体を独立して編集、交換、共有することができる。



◆ 図 4.15:トピックマップによる情報/知識の構造化、体系化、組織化

トピックマップは、問題領域に存在する主題(概念)をトピック(Topic)、主題間の関係を関連(Association)、そして、情報リソースとの関係を出現(Occurrence)という構成要素で表現し、問題領域をモデル化してコンピュータ処理可能にする。トピック、関連、出現は、それぞれの頭文字をとって、"TAO of Topic Maps" と呼ばれている。また、問題領域を主題中心(Subject-centric) にモデル化するものであり、これまでの文書中心(Document-centric) の計算機処理からのパラダイムシフトと考えられている。

図 4.16 にトピックマップの基本的なモデルを示す。図 4.16 は司馬遼太郎の世界を問題領域とした場合の単純な例であり、トピック、関連、出現として以下のものが描いてある。

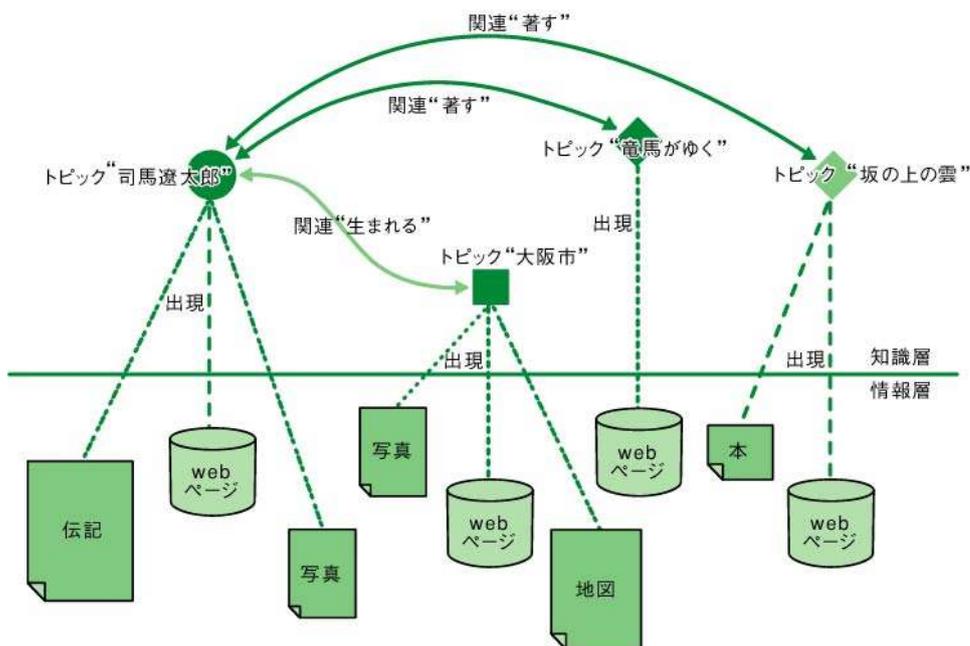
トピック: 司馬遼太郎、大阪市、竜馬がゆく、坂の上の雲

関連 : 著す、生まれる

出現 : 伝記、Web ページ、写真、地図、本へのリンク

そして、トピックと関連で以下の記述が表現されている。

- "司馬遼太郎" が "竜馬がゆく" を著す
- "司馬遼太郎" が "坂の上の雲" を著す
- "司馬遼太郎" は "大阪" に生まれる



◆ 図 4.16: トピックマップの基本モデル: TAO of Topic Maps

(b) 公開主題識別子 (Published Subject Identifier) トピックマップ中のトピックには、主題を同定/識別するための主題識別子 (Subject Identifier) を割当てることができる。主題識別子として、主題ごとにユニークな IRI (Internationalized Resource Identifier) を割当てて公開したものを PSI (Published Subject Identifier) と呼ぶ。PSI を表す記号として、PSI (ピー・エス・アイ) と発音が似ているということで、しばしば、 $\Psi$  (プサイ) が用いられる。IRI が指し示すアドレス上に置かれ、主題が何を意味しているのか人が理解できるように記述した情報リソースを、Published Subject Indicator、または、PSD (Published Subject Descriptor) と呼ぶ。

PSI を利用することにより、名前や別名を持つ Synonym (同意語)、Homonym (同音異義語)、Polysemy (多義性) などの問題に煩わされることなく、ユニークな IRI による正確な主題の識別/同定が可能になる。それを前提にして複数のトピックマップのマージや、トピックマップをベースにしたアプリケーション間でのトピックマップフラグメントの交換が可能になる。別々に作成された様々なサイト上の情報/知識を主題中心に結び付けて、情報/知識のネットワークを拡大、豊かにしていくことが可能になる。

#### 4.4.2 標準化動向

(a) 動向、種類 既に記述したように、トピックマップは、ISO/IEC JTC1 SC34 で策定された標準で、2000 年に最初の版、2003 年に第 2 版が国際標準として発行されている。

その後、全面的な見直しが行われ、現時点では以下の標準群から構成されている。

- ISO/IEC 13250: Topic Maps

Part-1: 概観及び基本概念

Part-2: データモデル (TMDM)

Part-3: XML 構文 (XTM)

Part-4: 正準化 (CXTM)

Part-5: 参照モデル (TMRM)

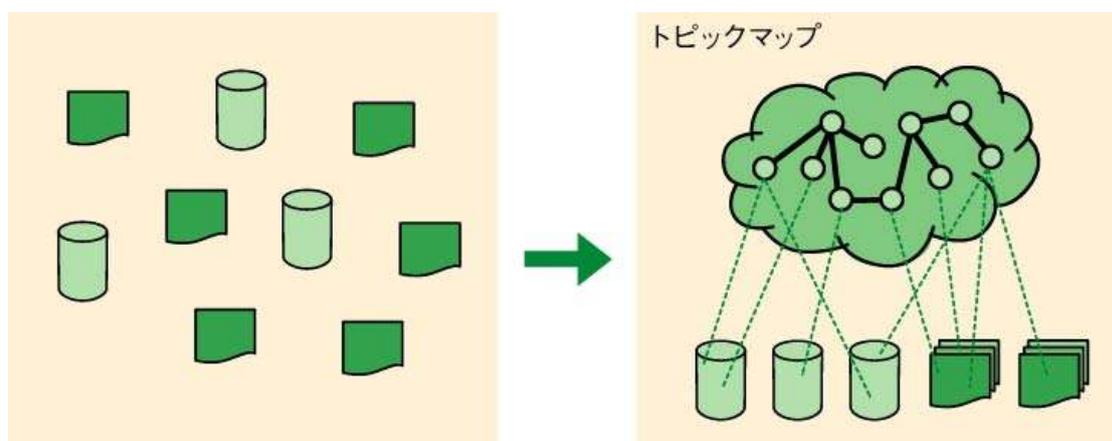
Part-6: 簡潔構文 (CTM)

Part-7: グラフ記法

- ISO/IEC 18048: 問合せ言語 (Topic Maps Query Language: TMQL)

- ISO/IEC 19756: 制約言語 (Topic Maps Constraint Language: TMCL)

このうち、2006 年に TMDM (トピックマップデータモデル)、2007 年に XTM (XML 構文)、2008 年に CXTM (Canonicalization) が国際標準として発行されている。2010 年には、CTM (簡潔構文) も国際標準として承認されている。TMQL (トピックマップ問合せ言語) と TMCL (トピックマップ制約言語) は、議論が大詰めを迎えている。最初の版と、TMDM、XTM、及び、CXTM は、JIS の規格にもなっている。図 4.17 に、各標準の関係を示す。



◆ 図 4.17: トピックマップ標準間の関係

## (b) 各標準の説明 ISO/IEC 13250: Topic Maps

### ISO/IEC 13250 Part-1: 概観及び基本概念

この規格は、ISO/IEC 13250 の各パートの概観と、それらが全体としてどのように組み合わせられるかを記述する。

### ISO/IEC 13250 Part-2: データモデル (TMDM)

この規格は、トピックマップのデータモデルを明記する。データモデルは、XML 情報集合(XML Infoset)形式を用いてトピックマップの抽象構造を、そして、散文を用いて特定の範囲についてその解釈を定義する。基本的な主題識別子と同様に、トピックマップの併合ルールについても定義する。データモデルの目的は、トピックマップ交換構文の解釈を定義することである。それにより、トピックマップの構文や処理環境に依存することなく、トピックマップがもつ情報を維持、共有、交換することが可能になる。さらに、正準化、問合せ言語、制約言語などを支援するための標準を定義するための基礎となることである。XML 情報集合は、XML 文書に含まれている情報の集合を "情報項目(information item)" と 情報項目が持つ 複数の "名前付き特性(named property)" で表現する。データモデルでは、同じ情報表現方法を用いて、トピックマップに含まれている情報を表現する。表 4.1 に、データモデルで定義されている情報項目と名前付き特性を示す。

◆ 表 4.1.データモデルで定義されている情報項目と名前付き特性

情報項目 名前付き特性	トピックマップ topic map	トピック topic	トピック名 topic name	異形 variant	出現 occurrence	関連 association	関連役割 association role
associations	○						
datatype				○	○		
item identifiers	○	○	○	○	○	○	○
occurrences		○					
parent		○	○	○	○	○	○
player							○
reified		○					
reifier	○		○	○	○	○	○
roles						○	
roles played		○					
scope			○	○	○	○	
subject identifiers		○					
subject locators		○					
topic names		○					
topics	○						
type			○		○	○	○
value			○	○	○		
variants			○				

### ISO/IEC 13250 Part-3: XML 構文 (XTM)

この規格は、トピックマップのための XML に基づく交換構文について定義している。交換構文は、[ISO/IEC 13250 Part-2: データモデル (TMDM)]で定義するデータモデルのインスタンスの交換に用いることができる。この規格は、交換構文からデータモデルへの対応付けも定義している。構文は RELAX-NG スキーマによって定義され、データモデルへの対応付けによって、より精密に定義されている。構文の解釈も定義している。XTM は、以下の 18 の XML 要素によって構成されている。

topicMap 要素 topic 要素 itemIdentity 要素 subjectLocator 要素 subjectIdentifier 要素 instanceOf 要素 name 要素 value 要素 variant 要素 scope 要素 type 要素 occurrence 要素 resourceData 要素 resourceRef 要素 association 要素 role 要素 topicRef 要素 mergeMap 要素

#### ISO/IEC 13250 Part-4: 正準化 (CXTM)

この規格では、トピックマップのデータモデルにおけるあらゆる情報集合のソート順と、XML 情報集合(XML Infoset)への変換規則を規定している。トピックマップは、知識を表現できる抽象的なデータモデルを定義している。このトピックマップは、その文法を用いたファイル、データベース内のエンティティ、プログラム中のデータ、または人の頭の中での心的な表現とさまざまな形式で表すことができる。正準化(canonicalization) は、トピックマップのデータ構造を、定められた順序に従い直列化する処理を意味している。定められた順序に従い直列化された二つの異なるデータ構造を比較し、その結果が等しいならば、それらは同一の内容を表しているといえる。正準化により、二つのデータ構造を比較し、同一であるか否かの判断することが可能になる。

#### ISO/IEC 13250 Part-5: 参照モデル (TMRM)

この規格は、データモデル (TMDM) より抽象的なモデルであり、トピックマップを含め、語彙、タクソミー、シソーラス、スキーマ、オントロジー、その他の多様な主題の識別方法に適用可能な共通の形式モデルを提供すること、それにより、それらのマッピングを可能にすることを目指している。いろいろな検討を経て、現時点では、集合モデルを定義するための試行が続けられている。集合モデルとして、まず、キーと値のペアとして特性を定義する。次に主題代用品を定義する。主題代用品は、一つの主題を特定するための一つ以上の特性の集合である。主題代用品 = {特性 0, 特性 1, ..., 特性 n}さらに、主題マップ(subject map) を定義する。主題マップは、一つ以上の主題代用品の集合である。主題マップ = {主題代用品 0, 主題代用品 1, ..., 主題代用品 n}

#### ISO/IEC 13250 Part-6: 簡潔構文 (CTM)

この規格は、データモデル (TMDM) のインスタンスのテキスト形式の記法を定義する。また、この記法からデータモデルへのマッピングも定義する。CTM (Compact Topic Maps) は、ISO/IEC 13250 Part-3: XML 構文 (XTM)で定義されている XML に基づく交換構文を補完する簡潔、軽量の記法を提供する。そして以下の用途が期待される。 - 手作業によるトピックマップの作成 - 人間が読みやすいトピックマップのサンプルの提供 - TMCLおよびTMQLの共通の構文基盤 CTM の主要な設計基準は、簡潔さ、人間による作成の容易さ、そして、読みやす

さである。

#### ISO/IEC 13250 Part-7: **グラフ記法 (GTM)**

この規格は、トピックマップのグラフ記法を定義する。グラフ記法は、設計、説明や教育など、構文以外のトピックマップ表現が有効と思われる場合の使用を目的としている。現時点で 2 つのレベルのグラフ記法が検討されている。レベル 0 は、スキーマレベルを表現し、レベル 1 は、インスタンスレベルを表現する。

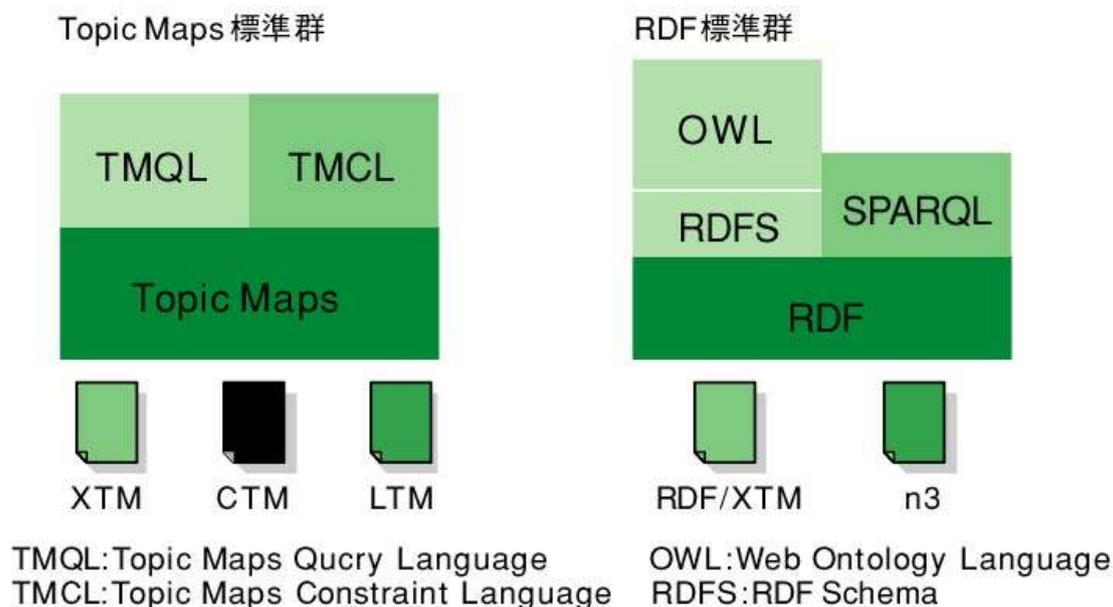
#### ISO/IEC 18048: **問合せ言語 (Topic Maps Query Language: TMQL)**

この規格は、トピックマップの問合せ言語を定義する。TMQL は、例えば、関係データベースにおける問合せ言語 SQL と同様の役割を果たすことを目指している。TMQL により、トピックマップの構造に基づいた問合せが可能になり、トピックマップの中に含まれている情報(内容)にアクセスすることが可能になる。現時点で、問合せ式の構文は、Path Expressions(経路式)、SELECT Expressions(SELECT 式)、FLWR Expressions(FLWR 式)の 3 種類が検討されていて、3 種類の式を組合わせて使用することも可能としている。経路式は、XPath(XML Path Language) 風、SELECT 式は、SQL の SELECT 文風、また、FLWR 式は、XQuery(XML Query Language) の FLWOR 式(FLWOR Expressions)風の構文と機能を持つ。

#### ISO/IEC 19756: **制約言語 (Topic Maps Constraint Language: TMCL)**

この規格は、トピック型、名前型、出現型、関連型、役割型から構成されるトピックマップスキーマ、および、スキーマの構成要素についての制約を正確で機械可読な形式で定義することを可能にする。制約には、例えば、トピックのインスタンスにおける名前や出現の出現回数、出現のデータ型、トピックのインスタンスがある関連において持つことができる役割などがある。トピックマップが TMCL スキーマおよび制約に適合しているかどうか確認することが可能になり、より厳密なトピックマップの作成が可能になる。TMCL は、それ自身は独自の構文をもっていないが、XTM や CTM などのトピックマップ構文を使用して記述する。利用を容易にするために、CTM 構文のテンプレートを複数定義している。

(c) **RDF とトピックマップの相互運用性** トピックマップは ISO で作成され、RDF は W3C で作成されている。ともに複数の標準から構成されていて、多くの類似点もあるが、異なる点もある。W3C の SWBPD (Semantic Web Best Practices and Deployment Working Group) の中に、トピックマップと RDF の相互運用のためのタスクフォースが作られ、2006 年 2 月 10 日付けで、"RDFTM: Survey of Interoperability Proposals"が公開されている。



◆ 図 4.18: トピックマップ標準群と RDF 標準群

RDF/トピックマップ間のマッピング方法の例として、Lars Marius Garshol 氏を提唱する方法を紹介する。RTM (RDF to topic maps mapping) という RDF ボキャブラリを使用して、RDF データとして両構成要素間の対応関係を指定する方法である。RTM は以下に示す RDF プロパティとリソースから構成される。

#### rtm:maps-to プロパティ

RDF プロパティとトピックマップ構成要素とのマッピングを定義する。

#### rtm:type プロパティ

マッピングによって作成されるトピックマップ構成要素の型を指定する。

#### rtm:in-scope プロパティ

マッピングによって作成されるトピックマップ構成要素のスコープを指定する。

#### rtm:subject-role

RDF ステートメントを、トピックマップの構成要素の一つである"関連"にマッピングする際に、RDF ステートメントの主語に該当する"関連"の役割の型を示す。

#### rtm:object-role

RDF ステートメントを、トピックマップの構成要素の一つである"関連"にマッピングする際に、

RDF ステートメントの目的語に該当する"関連"の役割の型を示す。

#### リソース

マッピング先のトピックマップ要素を指定する。リソースには、以下のものがある。

1. rtm:basename
2. rtm:occurrence
3. rtm:association
4. rtm:instance-of
5. rtm:subject-identifier
6. rtm:subject-locator
7. rtm:source-locator

## 5. 各分野でのメタデータ記述の現状

一貫したメタデータの作成のためには、記述に用いる語彙を定め、さらにそれぞれの項目の値をどのように記述するか、また項目は必須か任意かなどといった取り決めも明示する必要がある。こうしたメタデータの記述規則は、領域によって規範の状況がさまざまに異なる。主要なコンテンツの分野について、記述の規則と語彙定義の現状を報告する。

### 5.1 MLA 領域のメタデータ記述規則

公共性の高い資源を収集、保管、公開する博物館・図書館・文書館(MLA)では、それぞれの領域ごとにメタデータ記述規則の国際的な標準化が進められている。主要なものを概観する。

#### 5.1.1 図書館分野のメタデータ記述規則

図書館分野では古くから目録の整備が進んでおり、目録作成の規則も、19 世紀半ばに大英博物館のパニッツィによる 91 条目録規則、米スミソニアン協会図書館のジュエットによる目録規則、またカッター(Charles Ammi Cutter)の辞書体目録規則などが定められてきた。

(a) **パリ原則と ISBD** 現在の図書館目録は、タイトル、著者、出版社、出版年など資料に関する客観的な属性を記録する「記述」(Description)と、目録の内容を利用者が探するための手がかりとなる「標目」(Heading)という 2 つの概念で構成される。

前者については、1961 年に IFLA(国際図書館連盟)主催の国際目録原則会議において「パリ原則」が採択された。後者については、1970 年代に IFLA によって「国際標準書誌記述」(ISBD)<sup>1</sup>が策定された。これらを基本とし、言語や出版慣行といった事情を固有の考慮し、それぞれの規則が各国で定められている。日本でも「日本目録規則」(NCR)の 1987 年版が ISBD に準拠したものとなり、各館書誌記述の基本となっている(図 5.1)。

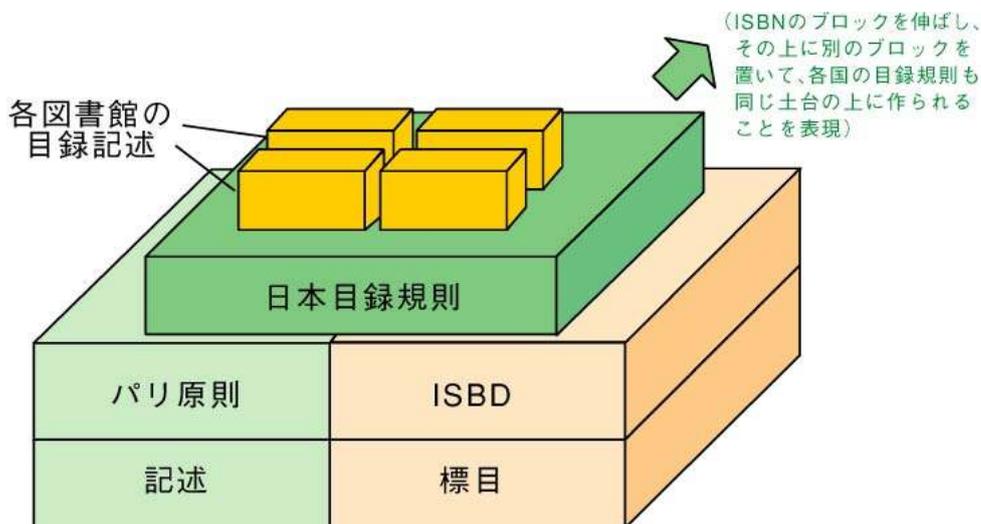
日本目録規則では、何を(記述対象)どの方式で(基本記入方式/記述ユニット方式)どのように(範囲と順序)記述するかを詳細に定義する。範囲と順序は次の項目からなる。

1. タイトルと責任表示に関する事項
2. 版(次)に関する事項
3. 資料(刊行方式)の特性に関する事項
4. 出版・頒布等に関する事項
5. 形態に関する事項
6. シリーズに関する事項
7. 注記に関する事項

<sup>1</sup> ISBD: International Standard Bibliographic Description,  
<http://www.ifla.org/en/publications/ifla-series-on-bibliographic-control-31>

## 8. 標準番号・入手条件に関する事項

さらに、何を必須、標準、オプションとするか(記述の精粗)、分冊や作品集などの場合の記述単位の扱い(書誌階層)、標記のばらつきを統一するための基準(記述／転記の方法)が定められている。



◆ 図 5.1: 記述—パリ原則: 標目—ISBD: これらに基づいて日本目録規則など各国の規則が作られ、さらにそれに準拠して各図書館の目録が作成される

(b) FRBR たとえば『ロミオとジュリエット』のメタデータという場合、シェイクスピアの創作した作品についての情報なのか、その初版本に関する情報なのか、あるいは小田島雄志訳による白水社刊行の本についてなのかが明確に区別されないと、データの精密な扱いができない場合がある(ある本の「作者」は、図書館から見ると作品の著者かもしれないが、博物館の観点では出版社であったり、あるいは装丁者であるかもしれない)。

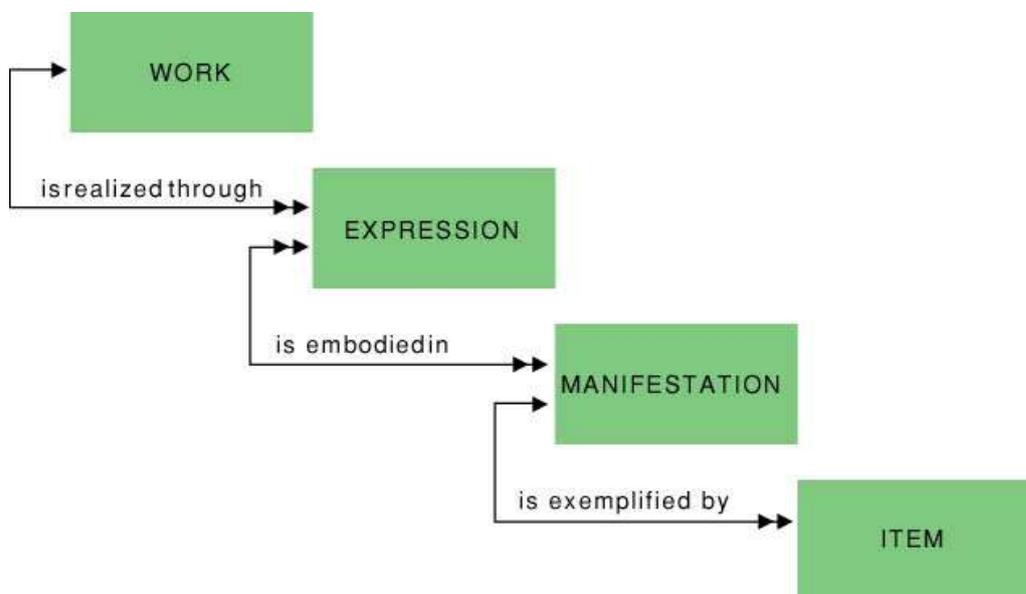
そこで書籍などのメタデータについて、コンテンツ自身としての「作品」(Work)、その原語版や翻訳版、あるいは映画版といった表出手段をふまえた「表現形」(Expression)、そしてそれが印刷書籍、電子ブック、DVD といった具体的な形を取った「体现形」(Manifestation)といったレベルに分けて考える新しいモデルを導入し、それに基づいて記述体系を見直した「書誌レコードのための機能要件」(FRBR)<sup>2</sup>が提案された(図 5.2)。これに基づく新しい英米目録規則「資料の記述とアクセス」(RDA)<sup>3</sup>が 2010 年に発行されている。

所蔵する個々の書物を記録・管理するカードから出発した目録を、FRBR のモデルに従ったものに作り変えるのは膨大な作業であり、容易に実現できるものではないが、『ロミオとジュリエット』という作品を的確に集約して見出すなどの利用のためには、「作品」と「書籍」の関係を明確化する

<sup>2</sup> Functional Requirements for Bibliographic Records, <http://www.ifla.org/publications/functional-requirements-for-bibliographic-records>

<sup>3</sup> RDA: Resource Description & Access, <http://www.rdatoolkit.org/>

ことは不可欠である。またこのモデルは図書館に限らず、メタデータ一般を考える上でも重要な役割を果たすものと考えられる。



◆ 図 5.2: FRBR のモデル

### 5.1.2 博物館分野のメタデータ記述規則

博物館においては、国際博物館会議の国際ドキュメンテーション委員会 (CIDOC, International Committee for Documentation) が文書化やメタデータの標準化に取り組んできている。1970 年代末から博物館内の目録に使う情報分類の議論が進められ、80 年代から 90 年代前半にかけて、美術と考古学のコレクションのための情報カテゴリ、博物館情報のデータモデルが開発された。

これらを元に、1995 年に CIDOC によって「博物館資料情報のための国際ガイドライン」(IGMOI, International Guidelines for Museum Object Information)<sup>4</sup>が、さらに 1998 年に「概念参照モデル」(CRM, Conceptual Reference Model)<sup>5</sup>が提案された。

#### (a) IGMOI IGMOI は

- 資料の詳細を記録する際に使う情報分類
- これらの分類に入れる情報を管理するための形式規則と慣習
- これらの分類に使われる用語

についての指針と説明を提供する。このガイドラインでは、資料の記述を取得情報、状態情報、償却・処分情報、記述情報、画像情報、組織情報など 22 の「情報グループ」に分け、それぞれの

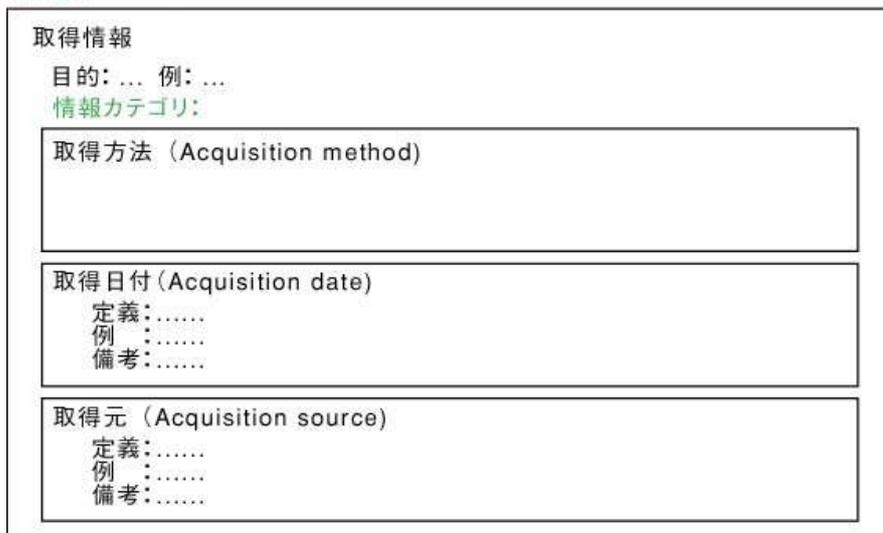
<sup>4</sup> International Guidelines for Museum Object Information: The CIDOC Information Categories June 1995, <http://cidoc.mediahost.org/guidelines1995.pdf>

<sup>5</sup> The CIDOC CRM, <http://www.cidoc-crm.org/>

グループにおいて記述する「情報分類」を定めている。たとえば取得情報グループには取得方法、取得日付、取得元、画像情報グループには画像タイプ、画像参照番号といった情報分類が含まれる(図 5.3)。

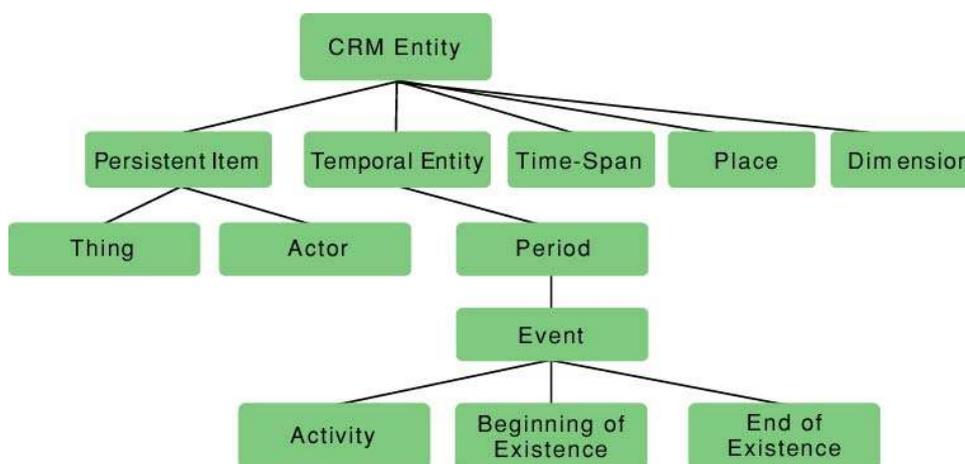
ガイドラインは、これらのグループと分類について、定義と記述例、制約などを提示し、メタデータ記述の基本的な規則を提供している。

情報グループ



◆ 図 5.3:IGMOI の情報グループと情報分類

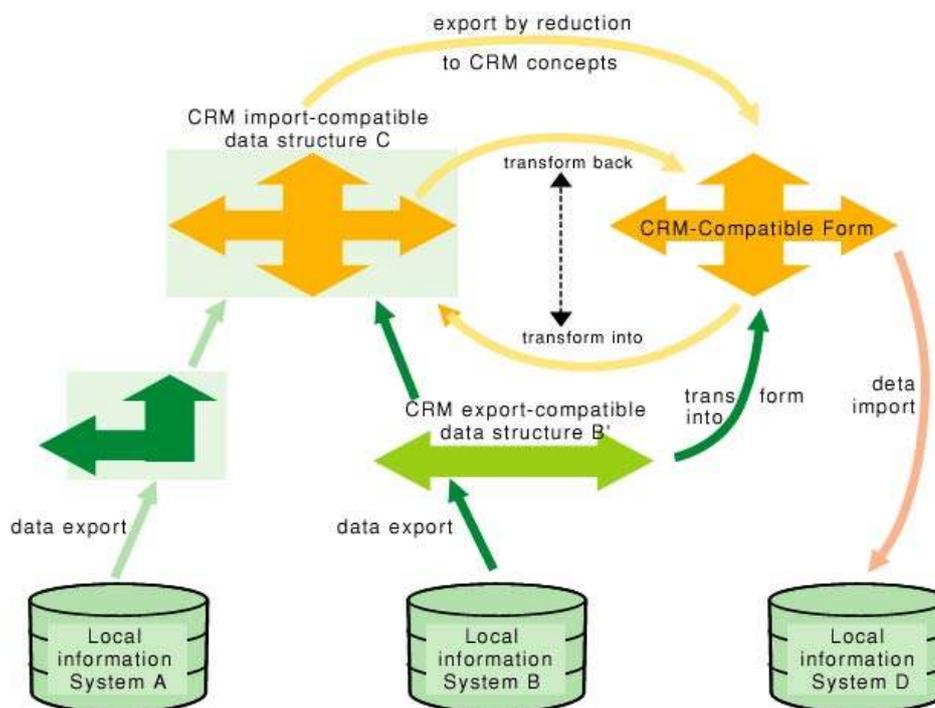
(b) CRM CRM は博物館資料の記述に用いる概念とその関係を定義するための手段を提供する。これは目録の項目や分類を直接提供するというよりも、それらを概念化したクラス階層および関連記述のためのプロパティを定義したオントロジーとなっている(図 5.4)。



◆ 図 5.4:CRM のクラス体系(部分)

このクラス階層においては、時間軸による推移のある実体 (Temporal Entity) と恒久的な実体 (Persistent Item) という大きな 2 つの軸が打ち出され、それに加えて期間 (Time-Span)、場所 (Place)、サイズ形状 (Dimension) といった物理的な表現が並べられている。時間推移する実体とはつまりイベントであり、そこには制作や改作といった事象が含まれる。

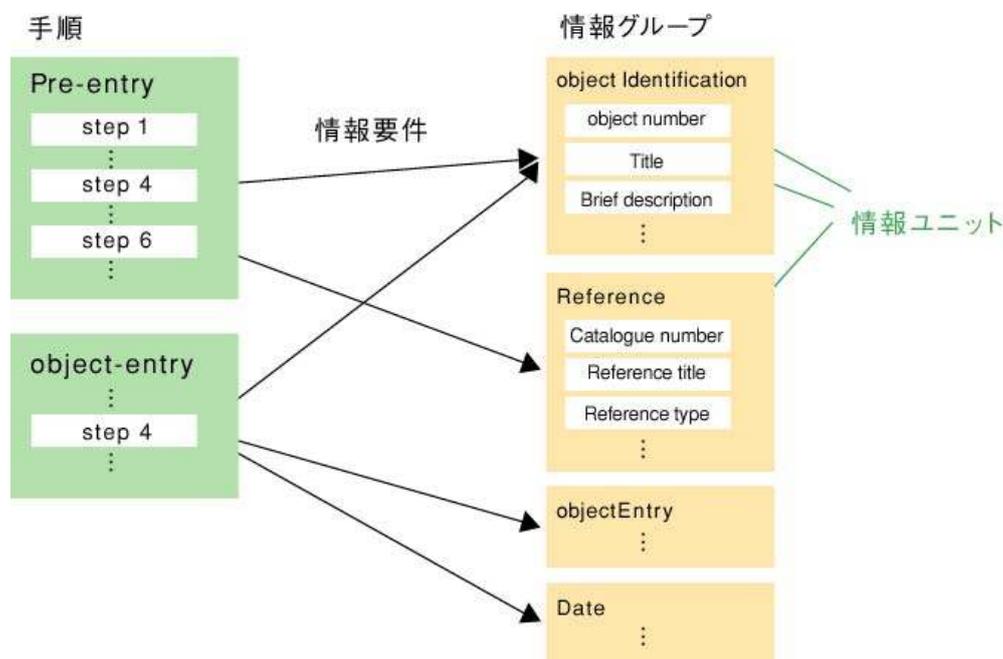
個々の博物館における記述規則は、このオントロジーを用いて形式的に定義したり、他館の記述規則との対応関係を示すことができる (図 5.5)。



◆ 図 5.5: CRM のオントロジーを介して、異なる構造の記述規則間でのやり取りができる

CRM のクラス、プロパティは RDF で表現でき、博物館資料のメタデータを相互運用可能な形で記述することが可能になっている。

(c) **ミュージアム資料情報構造化モデルおよびその他の規格** また、英国博物館ドキュメンテーション協会が 1994 年に定めた標準「SPECTRUM」も広く知られている。これは博物館の資料文書化に必要な要素として 20 以上の「手順 Procedure」と、その記述に用いる 400 以上の「情報ユニット」を定義している。情報ユニットはそれが記述するオブジェクト、イベント、組織などのまとめごと「情報グループ」を構成し、各手順のあるステップの記述にはどの情報グループを必要とするという形で記述規則が示される (図 5.6)。



◆ 図 5.6: SPECTRUM の手順、情報グループ、情報ユニット

そのほか、

- **CDWA** (Categories for the Description of Works of Art) : 米国Getty財団の AITF (The Art Information Task Force) による
  - **VRA Core** : VRA (The Visual Resources Association) による
  - **CIMI Profile** : CIMI (Consortium for the Computer Interchange of Museum Information) による
  - **CHIN** (The Canadian Heritage Information Network) : カナダ文化財情報ネットワークによる
- などの記述標準化項目案がよく知られている。

国内では、2005 年に東京国立博物館の「ミュージアム資料情報構造化モデル」が発表された（「6.1.3.アプリケーション・プロファイル」参照）。これらの国際的な標準モデルを念頭に、日本の資料や博物館の実情を踏まえて定義された記述モデルである。資料の受入単位だけでなく、構成部分を独立した記述単位に分割する階層的な記述を可能としている。資料は「識別・特定」「物理的特性」「履歴」「関連・参照」の 4 つに区分される 34 の属性によって記述される。

### 5.1.3 文書館分野のメタデータ記述規則

文書館分野では、国際文書館評議会 (ICA, International Council on Archives) を中心に記録史料記述の標準化が進められている。1992 年に基礎となる「記録史料記述に関する原則について

の声明」(マドリッド原則)を採択し、1994 年に記述の標準である「国際標準記録史料記述一般原則」(ISAD(G), General International Standard Archival Description)<sup>6</sup>が刊行された。また米国アーキビスト協会(SAA, Society of American Archivists)などが、記録史料記述のためのマーク付け言語「コード化記録史料記述」(EAD, Encoded Archival Description)<sup>7</sup>を 1998 年に公開している。

(a) ISAD(G) ISAD(G)は 26 の記述要素で構成され、各要素は 7 つの領域に分類されている。領域は識別表示、コンテキスト、内容と構造、アクセスと利用条件などの記述内容グループを表し、識別表示領域には参照コード、タイトル、年月日などの記述要素を用いるという形である。

記録史料は、たとえば領収書や走り書きのメモのように、それ単体では大きな意味を持たず、コンテキストの中で関連史料と組み合わせた集合体として扱ってはじめて史料価値をもつことが少なくない。そのため、ISAD(G)は記録史料の由来あるいは出所を重視し、同じ人物あるいは組織によって作られた史料の総体を「フォンズ」(Fonds)と呼んで、そのフォンズを頂点に置いた階層的な記述モデルを示している(図 5.7)。

出所となる組織が階層を持てばそれは「サブ・フォンズ」という単位にまとめられ、その下に一連の関連史料をまとめる「シリーズ」、そしてひとつの案件に相当する「ファイル」という階層が置かれ、それ以上分割できない最小単位(一つのメモ、写真など)を「アイテム」として記述する。

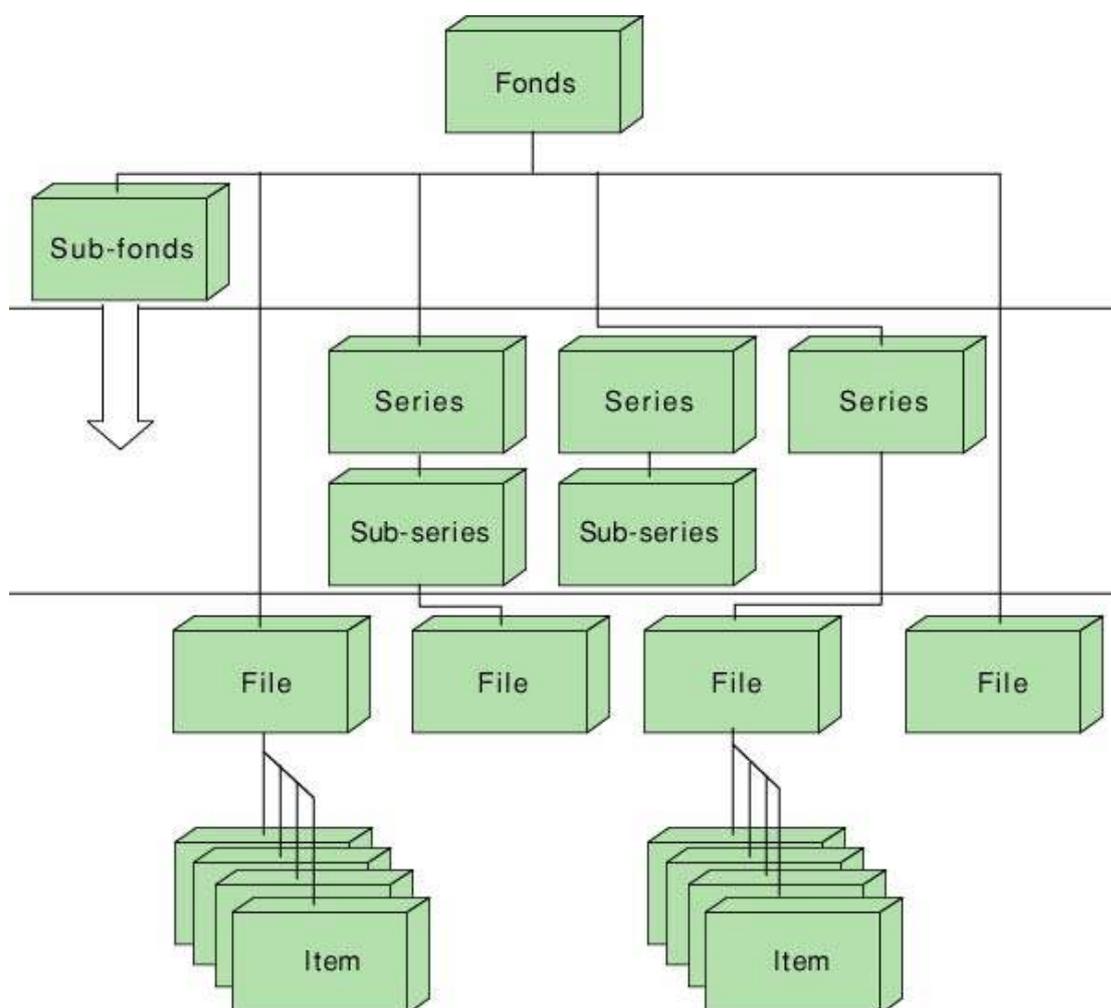
ISAD(G)の記述要素は、この階層のどのレベルにでも用いることができる。各レベルの記述は、コンテキストとしてこの階層のどこに位置づけられるのかを示す。

(b) EAD EAD は、ISAD(G)を意識しながら開発された XML の応用言語で、さまざまな情報を記述するために 140 以上の要素型(タグ)が定義されている。仕様書の付録に ISAD(G)との対応表が掲載され、ISAD(G)の記述要素に沿った情報を、XML として構造化し、記述できる。また付録では書誌記述の MARC21 との対応表も掲載され、領域を超えた相互運用が可能なことを示唆している。

---

<sup>6</sup> INTERNATIONAL COUNCIL ON ARCHIVES CONSEIL INTERNATIONAL DES ARCHIVES,  
[http://www.icacds.org.uk/eng/ISAD\(G\).pdf](http://www.icacds.org.uk/eng/ISAD(G).pdf)

<sup>7</sup> EAD:Encoded Archival Description Version 2002 Official site,Library of Con,  
<http://www.loc.gov/ead/>



◆ 図 5.7: ISAD のフォズ中心モデル

## 5.2 コンテンツ流通とメタデータ

### 5.2.1 電子書籍のメタデータ

書籍の電子的流通フォーマットは、一般に複数のコンテナを統合する形で構成され、本文を記述するファイルのほかにメタデータを記述するコンテナを含んでいる。

(a) EPUB 主要な電子書籍フォーマットのひとつ、EPUB<sup>8</sup>は、すでに第 2 版で、メタデータ記述にダブリンコア語彙を採用している。メタデータやナビゲーション情報を保持するオープン・パッケージング・フォーマット(OPF, Open Packaging Format)ファイルにおいて、metadata 要素内でタイトル、著者、言語などをダブリンコアで記述する。

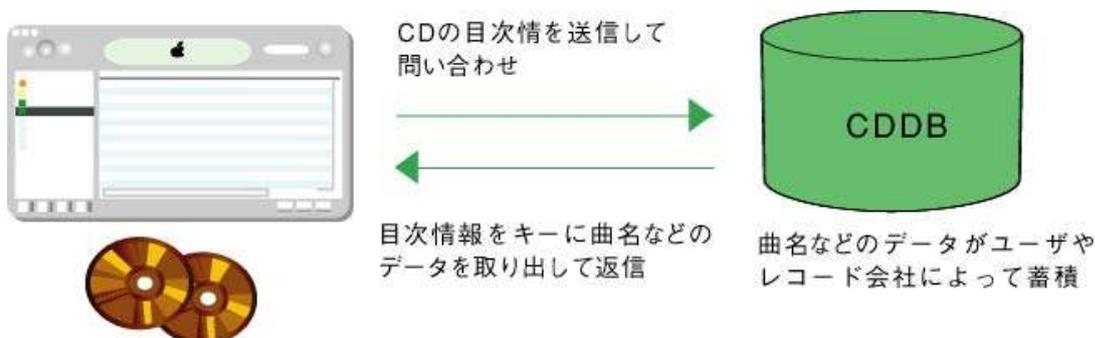
<sup>8</sup> EPUB International Digital Publishing Forum, <http://idpf.org/epub>

EPUB 3.0 では、出版仕様において metadata 要素の記述が定義され、第 2 版から引き継いだダブリンコア要素に加え、RDFa(「16.3.1.セマンティック・マーク付け」参照)を用いて任意のメタデータを記述できるようになる。

(b) **電子書籍交換フォーマット** XMDF は配布フォーマットにおいてタイトル、著者などの書誌情報を加えるようになっている。ドットブックでは、変換用の設定ファイルに記述した書誌情報がメタデータとして組み込まれる。これらの規格を念頭に策定されている「**電子書籍交換フォーマット仕様**」<sup>9</sup>は、書誌データファイルの bibliography 要素内に独自のメタデータを記述するが、その要素群はダブリンコア語彙をベースに、レーティングや表紙画像情報などを加えて拡張したものとなっている。

### 5.2.2 音楽データ

CD にはタイトルなどのメタデータを収録する CD-TEXT 規格もあるが、多くの音楽 CD はこれらのメタデータを含んでいない。そこで CD のパッケージとしての(アルバム)メタデータ、および各トラックの作品メタデータを独自に登録する、CDDDB と呼ばれる第三者サービスが普及している(図 5.8)。



◆ 図 5.8: CD や音楽データを再生するとき CDDDB にアクセスしてメタデータを取得・表示

CDDDB には商用サービスの Gracenote<sup>10</sup>やフリーサービスの FreeDB<sup>11</sup>などがあり、それぞれがアルバムのタイトルやジャンル、収録トラックのタイトル、アーティストなどのメタデータを提供するが、フォーマットはそれぞれ異なる。一般に、CD に記録されているデータから一意の ID を計算し、この ID に対応する形でメタデータが管理される。

また、このデータベースを拡張し、ビデオや映画などのコンテンツも含めたメタデータを提供するサービスもある。

<sup>9</sup> 電子書籍交換フォーマット, <http://ebformat.jp/>

<sup>10</sup> Gracenote.com, <http://www.gracenote.com/>

<sup>11</sup> Freedb.org>Home, <http://www.freedb.org/>

(a) **MusicBrainz** オープンな音楽データベースのひとつである MusicBrainz<sup>12</sup>は、そのメタデータ記述スキーマも公開している。2011 年 3 月現在で XML 形式の MusicBrainz XML Metadata Format が提示されているが、これを RDF 化する LinkedBrainz プロジェクト<sup>13</sup>も進められている。

また MusicBrainz はアルバム、トラック、アーティストの識別子である MusicBrainz Identifier (MBID)も公開しており、例えば CD 評を音楽データベースに結び付けるなど、さまざまな音楽関連情報の連動に利用されている。

(b) **MP3 の ID3 タグ** 音楽データを圧縮するファイルフォーマットの 1 つである MP3 には、「ID3 タグ」と呼ばれる形式でメタデータを持つことができる。これは MP3 再生ソフトが読み取るためのデータで、データのオフセット位置(何バイト目に記録されたデータか)によってデータ内容を識別する。

登録されるデータは、曲名、アーティスト名、アルバム名、日付、コメント文字列、ジャンルコードで、CDDDB のトラックデータとほぼ同様である。

### 5.2.3 メタデータの交換と流通

メタデータの記述は領域によって異なるのはもちろん、同じ領域でも組織ごとに細部に違いがあるため、直接共有することは難しい。それぞれのメタデータをダブリンコアなどの共通語彙に変換したものを取得できれば、最小限のメタデータ共有が可能となる。

メタデータを標準的な方法で公開し、機械的に取得可能にする方法としては、OAI-PMH や OpenSearch などの標準手段が規定されている。

(a) **OAI-PMH** Open Archives Initiative によって、メタデータを自動収集し、交換するためのプロトコル OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting)<sup>14</sup>が開発されている。これは XML のコンテナ内に、メタデータ本体に加えてフォーマットを問わずに共通な ID、日付を埋め込んで提供し、アプリケーションに依存しない相互運用可能な枠組みを提供する仕組みである。

OAI-PMH のレコードは、任意のフォーマットのメタデータを格納する `metadeta` 要素と、アイテムの識別子、日時スタンプ、削除フラグという共通情報を格納する `header` 要素、およびメタデータの権利や由来などのメタ・メタデータを格納するオプションの `about` 要素で構成される(図 5.9)。

利用アプリケーションは、レポジトリ(メタデータ提供サービス)に対してどのフォーマットのメタデータをサポートしているかを問い合わせることができ、フォーマットを指定した上でメタデータを要求する。また、ヘッダのみを取り寄せたり、一定日時以降のレコードのみを要求するといった手段

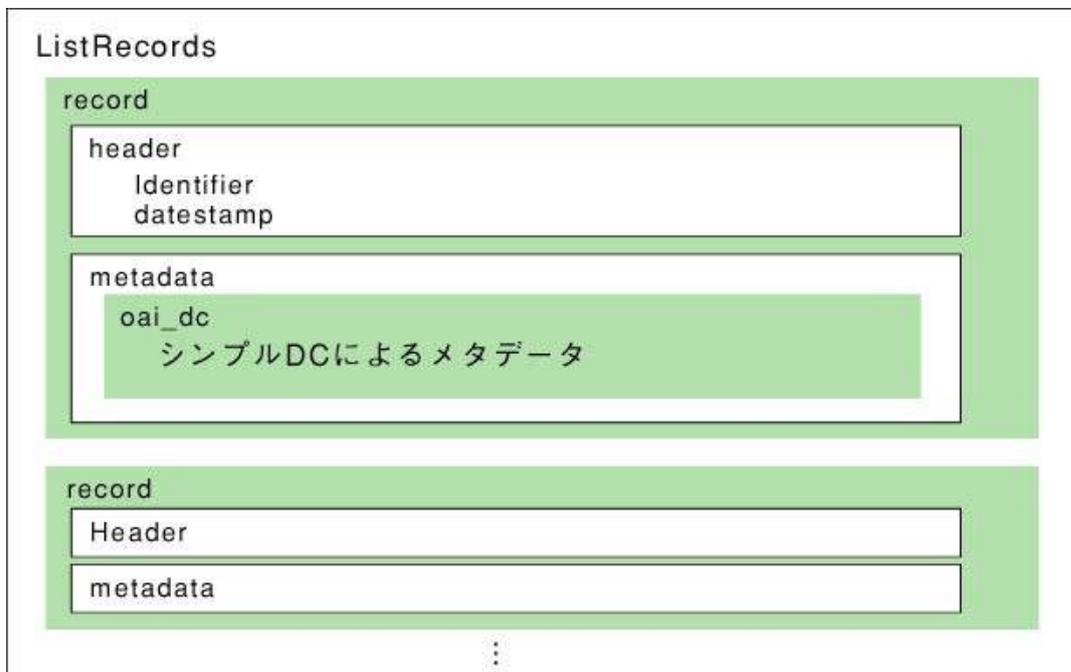
<sup>12</sup> Welcome to MusicBrainz!, <http://musicbrainz.org/>

<sup>13</sup> LinkedBrainz A project to provide MusicBrainz NGS as Linked Data, <http://linkedbrainz.c4dmpresents.org/>

<sup>14</sup> Open Archives Initiative Protocol for Metadata Harvesting, <http://www.openarchives.org/OAI/openarchivesprotocol.html>

が用意されており、累積的なメタデータの収集を可能としている。

メタデータ・フォーマットとしては、少なくともダブリンコアによる oai\_dc をサポートすることとされ、最小限の相互運用性を確保している。



◆ 図 5.9: OAI-PMH では、共通のヘッダとともに、指定したフォーマットのメタデータが得られる

(b) RSS と Atom ささまざまなサイトのコンテンツ更新情報を一定の形で提供し、集約を可能にするための配信フォーマットとして RSS (RDF Site Summary<sup>15</sup>もしくは Really Simple Syndication<sup>16</sup>) や Atom 規格<sup>17</sup>が用いられる。

いずれもコンテンツを item 要素 (Atom では entry 要素) で表現し、その中にタイトル、作者、日付、リンク先などを示す要素を持つ。複数のコンテンツ情報が channel 要素 (Atom では feed 要素) にまとめられ、購読アプリケーションが定期的アクセスすることで、最新の情報が得られる仕組みになっている。

(c) OpenSearch 米アマゾン子会社の A9.com 社が提唱する OpenSearch<sup>18</sup>は、検索エンジンへの問い合わせと結果のフォーマットを標準化する規格で、情報の検索を複数サイトに対して行ない、その結果を組み合わせて利用することを可能にする。

OpenSearch に対応する検索サイトは、どのような方法で問い合わせをすればよいかなどを記述した OpenSearch description document (記述文書) を提供する。そこに示されたテンプレートに従っ

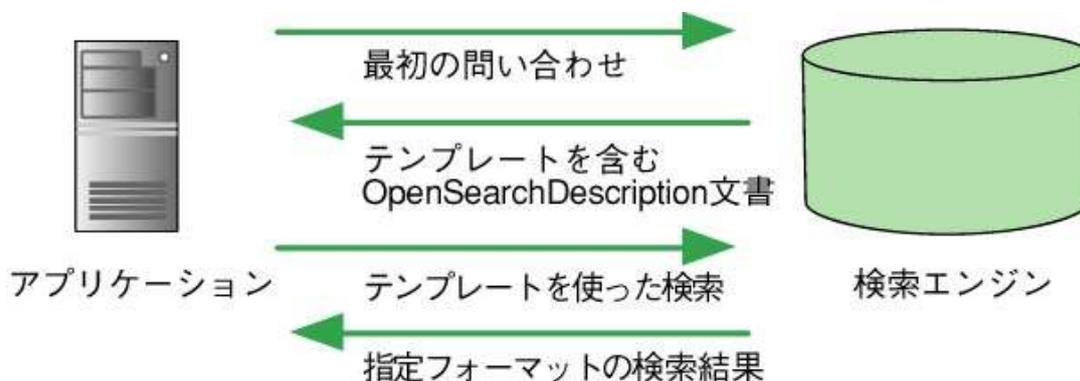
<sup>15</sup> RDF Site Summary (RSS) 1\_0, <http://web.resource.org/rss/1.0/spec>

<sup>16</sup> RSS Advisory Board, <http://www.rssboard.org/>

<sup>17</sup> RFC 4287 - The Atom Syndication Format, <http://tools.ietf.org/html/rfc4287>

<sup>18</sup> Home - OpenSearch, <http://www.opensearch.org/>

て問い合わせを送ると、RSS、Atom などのフォーマットで検索結果が返される。サイトごとに検索パラメータなどの違いが存在するが、記述文書を取り出す共通方法を用意することで、異なる検索サービスに対して同じ問い合わせを行なうことが可能となっている(図 5.10)。



◆ 図 5.10:OpenSearch description を用いることで、異なるパラメータを必要とする検索サイトに一定の手順で照会できる

#### 5.2.4 管理のためのメタデータ

メタデータには、タイトルや作者などの「記述メタデータ」以外に、資料の保存やアクセス制御などに関する「管理メタデータ」がある。特に資料の長期保存や権利マネジメントの観点から、管理メタデータの重要性が高まっている。

(a) Creative Commons 権利やライセンスに関するメタデータは、ダブリンコアの dc:rights プロパティをはじめ、電子書籍などのメタデータ記述規則にも含まれている。しかしネットワーク上のさまざまなコンテンツを利用するために、それぞれ異なる権利メタデータをチェックしたり、その内容を人間が確認する必要があるのでは、扱いにくい。

クリエイティブ・コモンズ (CC, Creative Commons)<sup>19</sup>は、知的創作物の“著作権”を尊重しつつ、その成果をより広く容易に利用するための新しい形を提唱する活動である。そのプロジェクトのひとつ Licensing Project において「一定の条件の下に創作物を自由に利用してよい」というライセンスを用意し、作者が適当なライセンスを選択して自分の著作物に適用できるようにしている。これは、

- クレジットの明示 (Attribution)
- 商用利用 (Commercial Use)
- 作品の翻訳、編曲などの改編、派生作品 (Modification)

の3つをどう扱うかを組み合わせる形でライセンスを定義する。ライセンスは一般利用者向けのライセンス“証書”(Commons Deed)、法的なライセンス文書 (Legal Code)、機械可読なメタデータ

<sup>19</sup> Creative Commons, <http://creativecommons.org/>

(Digital Code)の3つで構成され、メタデータは RDF によって記述されている。CC ライセンスに基づいて提供されるコンテンツは、次のようなアイコンでライセンスを示すこともできる(図 5.11)。



◆ 図 5.11: Creative Commons のライセンスアイコン

(b) DRM 機械可読な権利メタデータという意味では、デジタル著作権管理(DRM, Digital Rights Management)もその一種になる。ただし DRM は、一般にコンテンツの広い流通とは逆に無制限な複製を抑制する目的であり、また特定のメーカーに占有された技術となっている。

(c) OAIS 参照モデルと PREMIS、METS 一般にデジタルコンテンツは、印刷物などに比べて長期保存の方法が確立していない。たとえば印刷した書物であれば、図書館が資料の閲覧だけでなく保存の役割も担っている。一方で電子ジャーナルのようなデジタルコンテンツの場合、利用者は(図書館を経由する場合でも)出版社などのコンテンツ保有者にアクセスしており、コンテンツ保有者以外にその資料を保存する機関はない。デジタルコンテンツを長期にわたって利用可能とするためには、その保存に関する標準が必要となる。

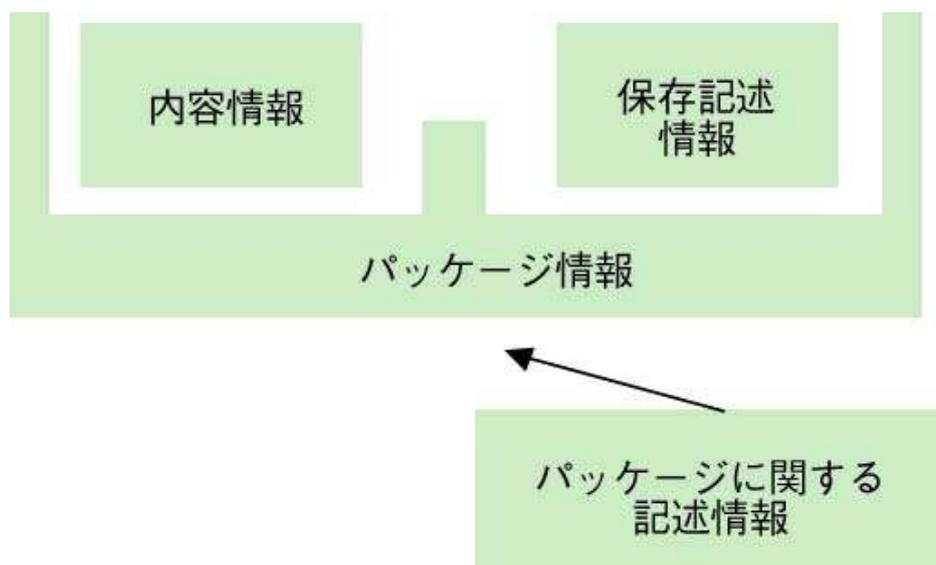
デジタル情報の長期保存システム構築に関する国際標準としては、宇宙データシステム諮問委員会(CCSDS, Consultative Committee for Space Data Systems)の開発した OAIS 参照モデル(Reference Model for an Open Archival Information System)がある<sup>20</sup>。

このモデルは、保存するコンテンツの内容情報と保存記述情報から成る「情報パッケージ」という概念を持つ。保存記述情報は

- 内容情報の由来(Provenance)
- 他の情報との関係(Context)
- 内容情報を同定するための参照(Reference)
- 内容情報が変更されていないこと(Fixity)

の4つからなり、これに加えて内容情報に含まれる解釈・提示のための表示情報が保存メタデータ(管理メタデータ)となる(記述メタデータはパッケージ情報には含まれず、外部から記述される。(図 5.12)。

<sup>20</sup> ユーザー名とパスワードが必要, <http://public.ccsds.org/publications/archive/650x0b1.PDF>



◆ 図 5.12: OAIS の情報パッケージと内容情報と保存記述情報、および記述メタデータ

これを発展させる形で、OCLC(Online Computer Library Center,Inc.)と RLG(Research Libraries Group)が作業部会 PREMIS(PREservation Metadata: Implementation Strategies)を発足させた。PREMIS では、メタデータが表す情報を、知的実体、オブジェクト、エージェント、権利、イベントの 5 つのタイプの実体に分ける抽象データモデルを構築し、知的実体を除く実体のメタデータを記述するコア要素(意味単位)を PREMIS データ辞書<sup>21</sup>として定義している(知的実体のメタデータは記述メタデータであり、ダブリンコアなどを用いて記述できる)。

また電子図書館連合(DLF, Digital Library Federation:)は、この参照モデルに準拠した具体的な XML スキーマ規格である METS(Metadata Encoding and Transmission Standard)<sup>22</sup>を開発した。METS は、ヘッダ、記述メタデータ、管理メタデータ、ファイルセクション、構造マップ、構造リンク、動作記述の 7 セクションで構成される。これらのセクション内容は、PREMIS の要素をはじめとする任意の語彙を用いて記述できる

### 5.3 メタデータ記述の代表的語彙

#### 5.3.1 セマンティック・ウェブとメタデータ記述語彙

RDF は、文書に限らず非常に多様なものをコンピュータ理解可能な形で記述可能とする。セマンティック・ウェブは、人物プロフィールやその知人関係、概念シソーラス、権利関係など、さまざまな資源のメタデータを RDF を用いて記述し、「データのウェブ」(「16.2.1.リンクするデータとは」参

<sup>21</sup> premis-dd-2-1, <http://www.loc.gov/standards/premis/v2/premis-dd-2-1.pdf>

<sup>22</sup> Metadata Encoding and Transmission Standard (METS) Official Web Site, <http://www.loc.gov/standards/mets/>

照)を構築しようとする。

こうした広範な資源記述に用いられる RDF 語彙としては、人物を中心とした表現のための FOAF(Friend of a friend)、シソーラス表現の SKOS(Simple Knowledge Organization System)、再利用のためのライセンスを記述する Creative Commons などがある。これらの語彙は、ダブリンコアや各領域の語彙と組合せることで、より豊かなメタデータの表現を可能にする。

以下にこれらの語彙で定義されている主要なプロパティ、クラスを示す。

### 5.3.2 ダブリンコア

4.1.3 で取り上げたダブリンコアの語彙には、異なる領域を通じて共通に使える語彙として、シンプルで基本的(コア)なものに限定した「ダブリンコアメタデータ要素」と、基本要素を精緻化し、詳細なメタデータ記述が可能な拡張要素を含めた「ダブリンコア・タームズ」がある。

(a) **ダブリンコアメタデータ要素(シンプル DC)**      ダブリンコアメタデータ要素(DCMES, Dublin Core Metadata Element Set)は、できるだけ幅広い領域のリソースを記述できる「コア」なメタデータ記述語彙として定義された。要素はシンプルで基本的なものに限定しており、必要ならこの共通項を基盤にして拡張ができる柔軟性を提供している。

DCMES には表 5.1 の 15 のプロパティが含まれる。

DCMES は、次の名前空間で定義されている。

<http://purl.org/dc/elements/1.1/>

DCMES を次のダブリンコア・タームズと区別するために、シンプル DC と呼ぶ。

◆ 表 5.1 :DCMES のプロパティ

プロパティ	定義
title	リソースに与えられた名前
description	リソースに関する説明
date	リソースのライフサイクル中の出来事に関連する日時もしくは期間
creator	リソースの作成 * に主たる責任を持つ実体
contributor	リソースへの協力、貢献に責任を持つ実体
publisher	リソースを利用可能にすることに責任を持つ実体
type	リソースの性質もしくはジャンル
format	ファイル形式、物理メディア、リソースのサイズなど
language	リソースの言語
identifier	ある文脈における、リソースへの曖昧さのない参照
rights	リソースに適用される権利に関する情報
relation	関連するリソース
source	リソースの派生元リソース
subject	リソースのトピック
coverage	リソースの空間的あるいは時間的トピック、あるいは適用対象、リソースが有効となる地域など

(b) **ダブリンコア・タームズ** DCMES は、シンプルな「コア」であることを目指して 15 のプロパティを定義するにとどめ、より精緻な記述のためには、利用者がプロパティ拡張できるようにした。しかし「作成日」「更新日」といった頻繁に用いる記述を利用者ごとに拡張しては、「意味の相互運用性」という重要な目的が果たせない。そこで、ダブリンコア自身が基本要素を精緻化し、詳細なメタデータ記述が可能な拡張要素を導入し、DCMES の 15 要素とあわせて DCMI Metadata Terms を定義した。

表 5.2、表 5.3 にこのダブリンコア・タームズから主要なものを挙げる。

ダブリンコア・タームズは、次の名前空間で定義されている。

<http://purl.org/dc/terms/>

◆ 表 5.2: タイトル、著者などに関するプロパティ

プロパティ	定義
title	リソースに与えられた名前
alternative	代替となるタイトル
description	リソースに関する説明
tableOfContents	リソースのサブ単位のリスト(目次)
abstract	リソースの要約
date	リソースのライフサイクル中の出来事に関連する日時もしくは期間
created	リソースが作成された日
valid	リソースが有効となる期日もしくは期間
available	リソースが利用可能となる期日もしくは期間
issued	リソースの公式発行日
modified	リソースが変更された日
creator	リソースの作成に主たる責任を持つ実体
contributor	リソースへの協力、貢献に責任を持つ実体
publisher	リソースを利用可能にすることに責任を持つ実体
language	リソースの言語
identifier	ある文脈における、リソースへの曖昧さのない参照
rights	リソースに適用される権利に関する情報

◆ 表 5.3: フォーマットや関連を記述するプロパティ

プロパティ	定義
type	リソースの性質もしくはジャンル
format	ファイル形式、物理メディア、リソースのサイズなど
extent	リソースのサイズもしくは長さ
medium	リソースの素材もしくは媒体
relation	関連するリソース
isVersionOf	主語リソースは目的語リソースのバージョン、版、翻案
hasVersion	目的語リソースは主語リソースのバージョン、版、翻案
isPartOf	主語リソースは目的語リソースの物理的もしくは論理的の一部
hasPart	目的語リソースは主語リソースの物理的もしくは論理的の一部
isReferencedBy	主語リソースは目的語リソースから参照、引用、あるいはポイントされている
references	目的語リソースは主語リソースから参照、引用、あるいはポイントされている
source	リソースの派生元リソース
subject	リソースのトピック
coverage	リソースの空間的あるいは時間的トピック、あるいは適用対象、リソースが有効となる地域など
spatial	リソースの空間的な特徴
temporal	リソースの時間的な特徴

### 5.3.3 FOAF

メタデータには著者をはじめ人物に関する記述がしばしば登場する。人とその活動に関する情報の記述には、FOAF(Friend of a friend)語彙がよく用いられている。FOAF 語彙は、次の名前空間で定義される。

<http://xmlns.com/foaf/0.1/>

FOAF は人物だけでなく、人物に関連する幅広いリソースを記述することを念頭に置いているため、人物だけではなく、その制作物である文書、写真、あるいは所属する組織などを表現するためのクラス(表 5.4)が用意されている。

◆ 表 5.4:FOAF が定義するクラス

クラス	説明
Agent	人間、グループ、ソフトウェアなど、「ある行為をする能力のある人(もの)」を総称するクラス
Person	人物を表すクラス。Agent のサブクラス
Group	Agent の集まりであるグループを表すクラス。Agent のサブクラス
Organization	会社、協会など、社会的な Agent をあらわすクラス。Agent のサブクラス
Document	文書を表すクラス
Image	画像を表すクラス。Document のサブクラス

人(もしくは Agent)を主語として、そのプロフィールを表現するプロパティには、表 5.5 のものが含まれる。

FOAF には、人(Agent)ばかりでなく、人が関心を持つ一般的なリソースなどを記述するためのプロパティも用意されている。主要なものを表 5.6 に挙げる。

◆ 表 5.5: Agent のプロフィールを表現するプロパティ

プロパティ	定義
name	名前(人に限らず使える)
surname	姓
givenname	名
nick	ニックネーム
gender	性別
title	敬称(Mr, Ms, Dr など)
mbox	特定の所有者に結びつけられたメールボックスで、家族などと共有していないもの
homepage	ホームページ
weblog	ウェブログ
openid	OpenID
knows	知っている人
interest	関心を持っていることに関するページ
topic_interest	関心を持っているトピック
publications	主語人物の出版、発表物リストのページ
made	主語エージェントが作ったもの
schoolHomepage	母校のホームページ
workplaceHomepage	勤務先のホームページ
currentProject	現在手がけているプロジェクト(のページ)
pastProject	以前手がけたプロジェクト(のページ)

◆ 表 5.6: 人が関心を持つ一般的なリソースなどを記述するためのプロパティ

プロパティ	定義
depiction	主語を描いたもの。写真やイラストなど
depicts	主語画像が描いている内容(depiction の逆)
maker	リソースの作者(made の逆)
based_near	目的語の近くにいる(ある)ことを示す
page	主語リソースに関するページ、文書
topic	主語ページ、文書のトピック(page の逆)
primaryTopic	主語ページの中心トピック。topic のサブプロパティ、かつ関数型プロパティ
isPrimaryTopicOf	primaryTopic の逆の関係を表す IFP
member	目的語は主語グループのメンバーである

### 5.3.4 SKOS

シソーラス、タクソミー、分類表や件名標目表などの知識組織化体系をウェブを通じて共有、リンク付けするための語彙として、SKOS (Simple Knowledge Organization System) が W3C から 2009 年に勧告されている。SKOS では分類や件名を概念 (Concept) クラスで表現し、それらの属性や関係を体系的に記述する。

SKOS の主要なプロパティとしては、概念に名前を与えるもの (表 5.7)、概念の関連を表現するもの (表 5.8)、概念記述の注を与えるもの (表 5.9) が定義されている。

◆ 表 5.7:SKOS のラベルプロパティ

プロパティ	定義
prefLabel	優先ラベル(標目)
altLabel	代替ラベル(同義語)
hiddenLabel	非表示ラベル

◆ 表 5.8 SKOS の概念関連付けプロパティ

プロパティ	定義
broader	広義語
narrower	狭義語
related	関連語
broadMatch	別のシソーラスにおける広義語
narrowMatch	別のシソーラスにおける狭義語
relatedMatch	別のシソーラスにおける関連語
closeMatch	別のシソーラスにおける類似語
exactMatch	別のシソーラスにおける同義語
inScheme	概念が属するスキーム(シソーラス)
hasTopConcept	シソーラスの最上位概念
topConceptOf	最上位概念として属するシソーラス

SKOS は次の名前空間で定義される。

<http://www.w3.org/2004/02/skos/core#>

また、読みを持たせるなど構造的なラベルを表現するために、SKOS 拡張ラベル語彙も SKOS 勧告で定義されている。これは拡張ラベルを Label クラスとして表現し、表 5.10 のプロパティで関連付けるものである。

SKOS 拡張ラベルは次の名前空間で定義される。

<http://www.w3.org/2008/05/skos-xl#>

◆ 表 5.9: SKOS の注釈プロパティ

プロパティ	定義
notation	表記法
note	注
editorialNote	編集注
historyNote	編集履歴
changeNote	変更注
scopeNote	スコープノート(適用範囲)
example	例
definition	定義

◆ 表 5.10: SKOS 拡張ラベルのプロパティ

プロパティ	定義
literalForm	ラベルの参照形
prefLabel	優先拡張ラベル
altLabel	代替拡張ラベル
hiddenLabel	非表示拡張ラベル
labelRelation	拡張ラベル間のリンク

## 6. メタデータ記述規則を定義するための標準方式

メタデータを適切に作成するためには、それぞれの組織において記述のための規則を定める必要がある。この記述規則の定義方法について、現状と標準化動向を確認し、形式的定義が可能な定義言語と、簡易な表で記述規則を定義できる簡易表現を提案する。

### 6.1 メタデータ記述規則定義の現状

メタデータの記述規則がどのような形で定義されているかについて、主要なコンテンツ提供組織の例をもとに現状を確認する。

#### 6.1.1 国立国会図書館書誌記述スキーム

国立国会図書館は、自ら大規模な書誌メタデータを作成するとともに、全国の図書館に書誌記述の標準を示す役割も果たしている。2001 年 3 月に「国立国会図書館メタデータ記述要素」を、2007 年 5 月にその改訂版「国立国会図書館ダブリンコアメタデータ記述要素」を定めてきたが、2010 年 6 月にアプリケーション・プロファイル(「6.2.1.リンクするデータとは」参照)の考え方に基づく「国立国会図書館ダブリンコアメタデータ記述」(DC-NDL)を定めて公開した<sup>1</sup>。

DC-NDL は、ダブリンコアの語彙をベースに、国立国会図書館独自の記述要素を定義して追加し、さらに記述上の制約規則、説明、例などを示したものである。基本的な考え方と概要を示した文書に加え、独自語彙の意味定義「第一部 NDL Metadata Terms」、語彙の用法についての規定「第二部 Application Profile」、RDF による語彙の定義「第三部 RDF スキーマ」および付録(語彙の一覧)によって構成されている。

このうち、第三部で提供される RDF スキーマ<sup>2</sup>は機械可読形式でそのまま利用可能である。例として、ここで定義されるプロパティのひとつ Transcription についてのスキーマ記述を次に示す。

```
<rdf:Description rdf:about="http://ndl.go.jp/dcndl/terms/transcription">
  <rdfs:label>Transcription</rdfs:label>
  <rdfs:comment>読みまたは翻字形</rdfs:comment>
  <dcterms:description>
    Title, Creator 等の値とセットで表現できる場合に使用する。
  </dcterms:description>
  <rdfs:isDefinedBy rdf:resource="http://ndl.go.jp/dcndl/terms/" />
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
  <dcterms:created>2010-06-21</dcterms:created>
</rdf:Description>
```

<sup>1</sup>書誌データ作成ツール：メタデータ基準 国立国会図書館-National Diet Library, <http://www.ndl.go.jp/jp/library/data/meta.html>

<sup>2</sup> ndl-terms, <http://www.ndl.go.jp/jp/library/data/meta/2010/06/ndl-terms.rdf>

第二部のアプリケーション・プロファイルは表 6.1 に示す要素を記述項目(プロパティ)ごとに表形式で説明する PDF 文書となっている。

複雑な表による PDF 形式で、基本的には人間の書誌記述者が参照するための資料であり、機械処理は困難な状態になっている。

◆ 表 6.1: 国立国会図書館アプリケーション・プロファイルの記述要素

要素	記述内容
プロパティ URI	当該プロパティを表す、参照先となる URI を示す。
QName	接頭辞と語彙の組み合わせによって、プロパティ URI の短縮形を表す。(例: dcndl:transcription)
定義の発生源	プロパティの定義元を URI で示す。
語彙のタイプ	語彙のタイプを記す。
表示名	利用者の理解を助けるために与える短い表示名。
使用法	Application Profile における使用法を記す。元の定義については、Dublin Core Metadata Terms、及び NDLMetadata Terms をそれぞれ参照のこと。
補足説明	「使用法」の詳細・補足説明等を記す。
語彙符号化スキームの使用	当該語彙に使用する語彙符号化スキームを示す。語彙符号化スキームは、必要に応じて使用する。各語彙における語彙符号化スキームの採否、出現順序、繰返しについては、Application Profile では制約を設けない。
値の記述形式	プロパティの値の記述形式を示す。URI による記述、任意の文字列による記述、構文符号化スキームによる記述、入れ子による記述がそれぞれ可能であるかどうか記す。入れ子による記述とは、RDF 形式を用い、構造化グラフとして表現することを指す。また、記述形式に制約がある場合はここに示す。
表現例	RDF/XML 形式による表現例を示す。
入力レベル	DC-NDL における記述の入力レベルの目安を、「必須」、「あれば必須」、「推奨」、「選択」の 4 段階で示す。

### 6.1.2 国立国会図書館デジタルアーカイブシステム

国立国会図書館の電子図書館サービスのひとつ「NDL デジタルアーカイブシステム」(NDL-DA)は、デジタルコンテンツを長期的に蓄積・保存し、将来にわたり利用可能とすることを目的として開発されている。NDL-DA では、OAIS 参照モデル(「5.2.4.管理のためのメタデータ」参照)に準拠し、多種多様なコンテンツをメタデータと関連付けた情報パッケージの単位で保存す

る。

メタデータには、表 6.2 に示すように、書誌情報に該当する記述メタデータのほか、長期保存に必要なとされる技術、権利、保存、管理の各メタデータ及び情報パッケージ自身のメタデータが含まれる。

◆ 表 6.2:デジタルアーカイブシステム・メタデータスキーマの構成

メタデータの構成	記述する情報	使用するスキーマ
情報パッケージのメタデータ	情報パッケージ自身を記述するメタデータ。	METS version 1.6
記述メタデータ	目録に相当する書誌情報。	MODS version 3.2
技術メタデータ	コンテンツを再生するために必要な技術情報。	PREMIS+独自要素
権利メタデータ	コンテンツの利用について規定する情報。	PREMIS+独自要素
保存メタデータ	コンテンツの作成履歴、受け入れた日付等の情報。	PREMIS+独自要素
管理メタデータ	インターネット情報の収集に関わる情報や、メタデータの登録・更新情報。	独自設計

メタデータの記述には METS、MODS、PRISM といった標準に加え、国会図書館が独自に追加定義したスキーマが用いられる。これらはそれぞれ XML スキーマが提供される。これに加えて、「NDL デジタルアーカイブシステム・メタデータ・スキーマガイドライン」<sup>3</sup>において、NDL-DA メタデータを構成する要素と属性の構造や制限が、要素ごとに次の内容の表で示される。

- 定義:NDL-DA メタデータにおける要素の定義を示す
- 値の記述形式:要素の値の記述形式を XML スキーマのデータ型で示す
- 下位要素:使用可能な下位要素を示す
- 属性:使用可能な属性を示す
- 入力レベル:「必須」「あれば必須」「任意」「使用しない」「コンテンツ種別により異なる」の5段階で示す。繰り返しの可否についても示す
- 使用方法:NDL-DA メタデータにおける要素の使用方法等の原則を示す

さらに、属性値の記述形式(入力レベル、使用方法等)および要素の記述例が示される。

<sup>3</sup> NDL デジタルアーカイブシステム・メタデータスキーマについて 国立国会図書館-National Diet Library, <http://www.ndl.go.jp/jp/standards/da/>

### 6.1.3 東京国立博物館

2005 年に、東京国立博物館の“博物館情報処理に関する調査研究プロジェクトチーム”により、CRM などの博物館分野における国際標準を踏まえた「ミュージアム資料情報構造化モデル」が発表された（「5.1.2. 博物館分野のメタデータ記述規則」参照）。

モデルを定義する文書<sup>4</sup>では、34 の属性それぞれについて、出現回数、定義とコメント、簡易例を示し、さらにその属性を細分化する詳細要素について、要素名、出現回数、定義とコメント、型を列挙した表を示している。また型（各属性の詳細要素の値となるデータの記述の枠組み）に関して「数値、識別子」「日付、期間、時期」「行為者、個人、グループ」「文書」「地域、遺跡」の 5 つの区分と、これらを構造化するための詳細要素を示している。

モデルの利用に際しては「組織や分野によって、属性や詳細要素を拡張することができるが、モデルに基づく情報共有のためには、このモデルで示されている構造に変換することが求められる」と、相互運用性の指針が記されている。

(a) スキーマ このモデルに基づいた情報共有のためのスキーマとしては、「ミュージアム資料情報 RDF ボキャブラリ (案)」<sup>5</sup>と「同 RDF 表現 (案)」<sup>6</sup>が示されている。

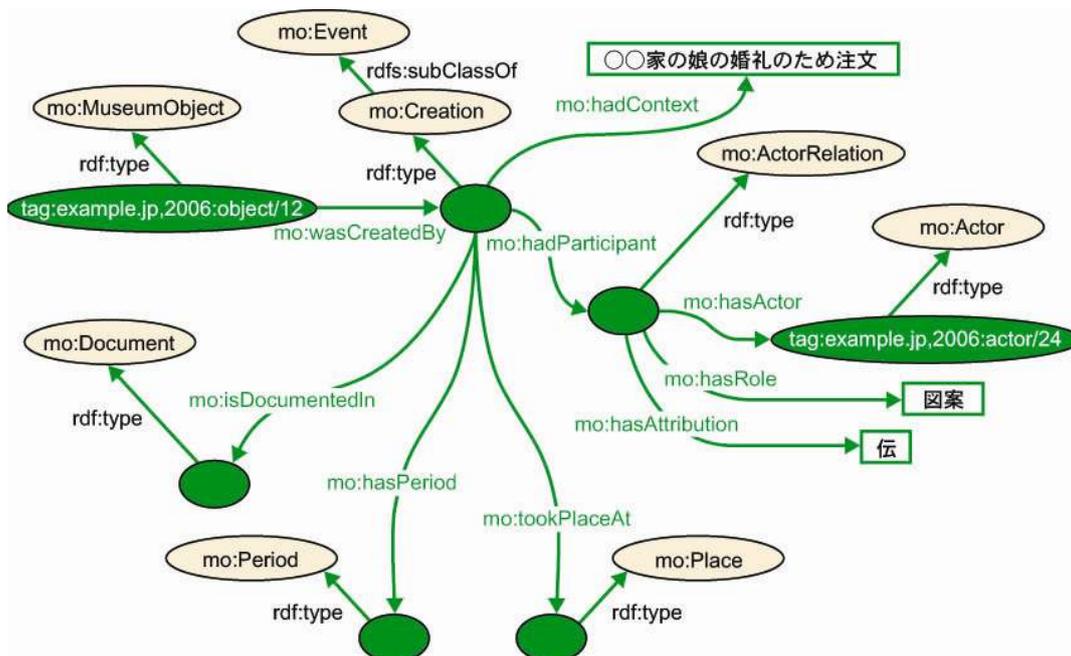
「RDF ボキャブラリ」では、MuseumObject、Actor、Event など 39 のクラスと、詳細要素レベルに対応する 102 のプロパティが定義されている。「RDF 表現」では、この語彙を用いてモデルの属性、型を表現する方法が、RDF グラフの図とともに説明されている。

たとえば、クラス mo:Creation、mo:ActorRelation とプロパティ mo:wasCreatedBy および mo:hadParticipant を用いて「制作」「役割」を表現する RDF グラフは、図 6.1 のように示されている。

<sup>4</sup> <http://webarchives.tnm.jp/docs/informatics/smmoi/>

<sup>5</sup> <http://webarchives.tnm.jp/docs/informatics/smmoi-rdf/>

<sup>6</sup> <http://webarchives.tnm.jp/docs/informatics/smmoi-rdf-expression/>



◆ 図 6.1:ミュージアム資料情報構造化モデルの「制作」「役割」の RDF 表現

### 6.1.4 国立美術館

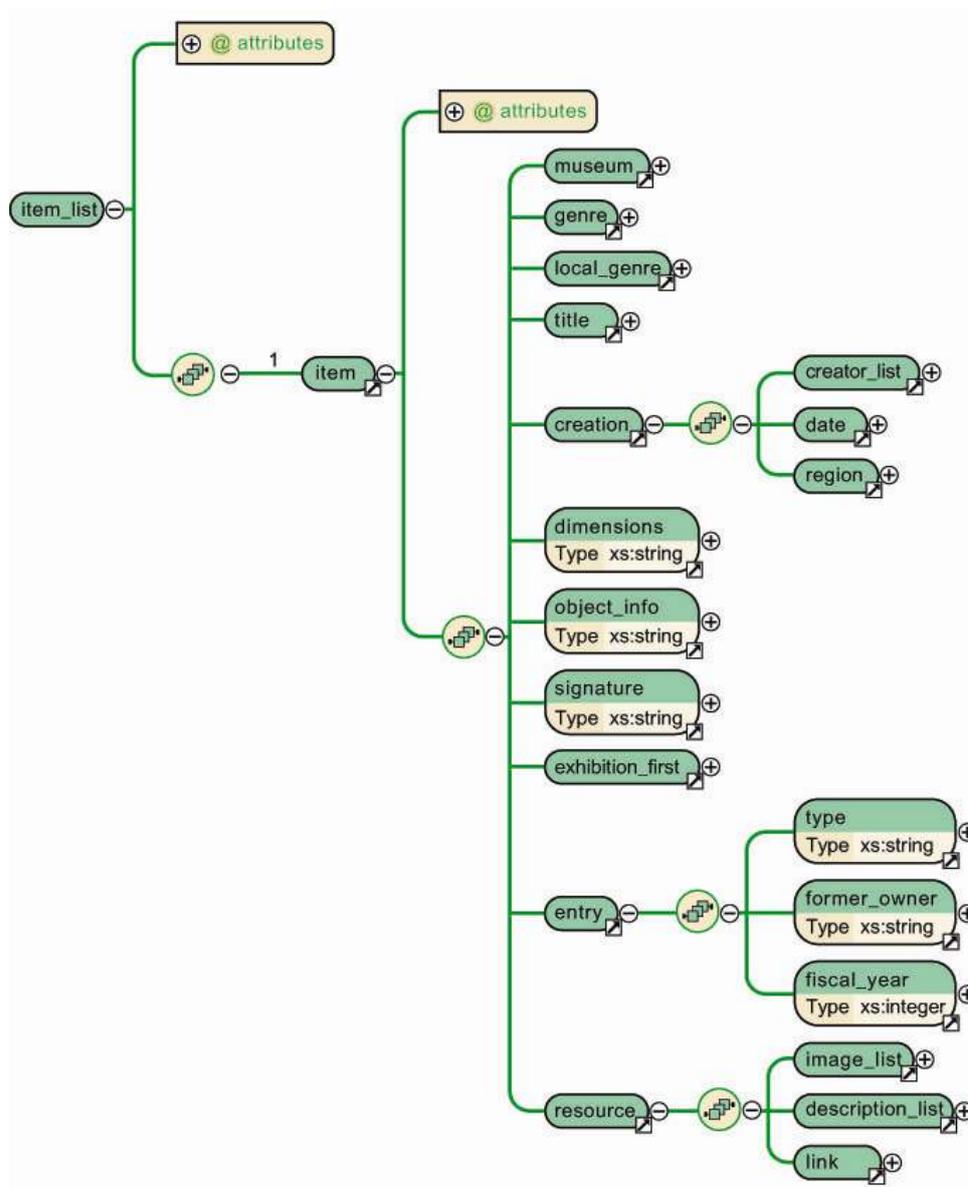
国立美術館は、東京国立近代美術館、京都国立近代美術館、国立西洋美術館、国立国際美術館の 4 館を統合した所蔵作品総合目録検索システム<sup>7</sup>を公開して成果を挙げているが、データの記述規則も XML スキーマとして整備が進みつつある。

国立美術館のスキーマは、図 6.2 に示すように、収蔵品を (item) として捉え、所蔵館 (museum)、タイトル (title)、制作 (creation)、サイズ形状 (dimensions)、受入れ (entry) といった要素で記述する。制作、受入れといったイベントは構造化して記述される。

記述項目の出現回数や値に関する制約は、XML スキーマの機能を用いて表現されている。このスキーマに合わせた XSLT を用意すれば異なるフォーマットのメタデータへの変換も可能である。XSLT を利用した文化遺産オンラインへの作品情報登録自動連携実証実験も報告されている<sup>8</sup>。

<sup>7</sup> 独立行政法人国立美術館・所蔵作品検索, <http://search.artmuseums.go.jp/>

<sup>8</sup> 水谷長志, 室屋泰三, 丸川雄三, 「独立行政法人国立美術館における情報〈連携〉の試みー美術館情報資源の利活用試案ならびに他関連機構との連携について」, 東京国立近代美術館『研究紀要』12号

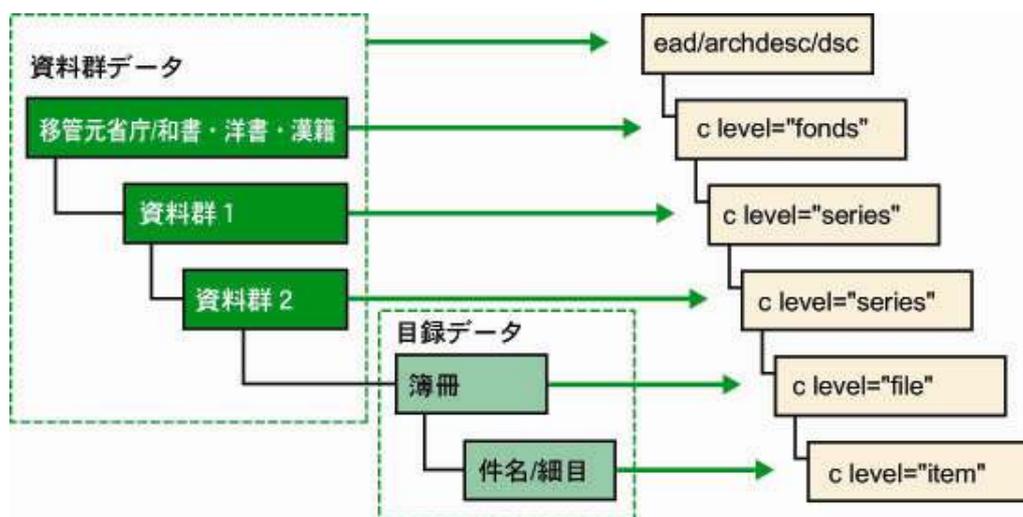


◆ 図 6.2: 国立西洋美術館の XML スキーマ(部分)

### 6.1.5 国立公文書館

国立公文書館ではEAD(「5.1.3.文書館分野のメタデータ記述規則」参照)に準拠した「国立公文書館 EAD 定義」<sup>9</sup>を定めて資料のメタデータを記述し、デジタルアーカイブで検索機能を提供している。

公文書の移管元である各省庁(内閣文庫の古典の場合は和書、洋書、漢籍)を最上位のフォンズとし、図 6.3 に示すように資料群をシリーズとして階層的に記述する。



◆ 図 6.3: 国立公文書館 EAD における資料群の階層

仕様書では、EAD の各記述要素について、資料種別、対応項目、記述例を表形式で説明し、繰り返しや値の制約については表外の文章で補足する形をとっている。XML スキーマは公開されていない。

なお 2009 年 3 月付けの「デジタルアーカイブシステム標準仕様書」<sup>10</sup>には、横断検索のためのダブリンコアの利用、表計算シートや RDB からの EAD/XML への移行についての考えがまとめられている。

### 6.1.6 学術機関リポジトリの相互運用メタデータ指針

国立情報学研究所は、機関リポジトリの相互運用性確保のためにメタデータ・フォーマット「junii2」<sup>11</sup>を策定し、公開している。

フォーマットの定義は表の形をとり、56 の記述要素それぞれについて、出現回数制約、書式、オプション属性などが示される。この定義では、junii2 の要素をダブリンコア(oai\_dc)に簡略化(ダ

<sup>9</sup> 国立公文書館 EAD 定義, [http://www.digital.archives.go.jp/support/pdf/naj\\_ead107.pdf](http://www.digital.archives.go.jp/support/pdf/naj_ead107.pdf)

<sup>10</sup> da\_100118, [http://www.archives.go.jp/law/pdf/da\\_100118.pdf](http://www.archives.go.jp/law/pdf/da_100118.pdf)

<sup>11</sup> 学術機関リポジトリ構築連携支援事業 | ドキュメント | システム情報 | メタデータ・フォーマット junii2, <http://www.nii.ac.jp/irp/archive/system/junii2.html>

ム・ダウン)する場合の対応プロパティも記載されている。主要な要素の定義の例を表 6.3 に示す。

XML スキーマも提供されているが、ここではダブリンコアへのダム・ダウン対応は示されていない。

さらに、junii2 メタデータ・フォーマットの各データ要素について説明と使用例を掲載したガイドラインも提供される<sup>12</sup>。これは、国立国会図書館のアプリケーション・プロファイルと同様、要素ごとに定義、制約、注意点などを表にして示すものである。例として表 6.4 に title 要素の説明を挙げる。

◆ 表 6.3:junii2 フォーマット定義の一部

項目	element	内容	必須	繰返し	最小	最大	書式	oai_dc	属性
タイトル	title		必須	×	1	1		title	lang
日付	date	コンテンツの作成日付			0	unbounded	YYYY-MM-DD, YYYY-MM, YYYY	date	
作成者	creator		あれば必須		0	unbounded		creator	lang

<sup>12</sup> junii2guide\_ver1.0, [http://www.nii.ac.jp/irp/archive/system/pdf/junii2guide\\_ver1.0.pdf](http://www.nii.ac.jp/irp/archive/system/pdf/junii2guide_ver1.0.pdf)

◆ 表 6.4:junii2 ガイドラインの例

項目	説明
要素名	title
使用レベル	必須
繰り返し	1
スキーム	—
oai_dc	title
説明	コンテンツを表すタイトルを記入する。論文における論文名である。コンテンツ本文と同じ言語を用いることを基本とする。
注意点	<ul style="list-style-type: none"> <li>・要素を繰り返すことは出来ない。</li> <li>・別言語のタイトルやタイトルのヨミは「2.alternative」を使用する。</li> <li>・掲載誌名は「27.jtitle」を使用する。</li> </ul>
推奨例	<p>&lt;title&gt;情報爆発時代の研究基盤構想：データマイニング、次世代サーチなどの研究&lt;/title&gt;</p> <p>コンテンツが日本語論文の場合。</p> <p>&lt;title&gt;Linking service to open access repositories&lt;/title&gt;</p> <p>コンテンツが英語論文の場合。</p>
非推奨例	<p>&lt;title&gt;富士山東麓湧水群における地下水環境と細菌の分布 = Distribution of bacteria in subsurface environment in east foot of Mt. Fuji&lt;/title&gt;</p> <p>別言語のタイトルは「2.alternative」を使用する。</p> <p>&lt;title&gt;NII の概要 (日本病院会雑誌)&lt;/title&gt;</p> <p>掲載誌名は「27.jtitle」を使用する。</p> <p>&lt;title&gt;レポート作成におけるコピー防止策&lt;/title&gt;</p> <p>&lt;title&gt;コピーを超えるライティング授業デザイン&lt;/title&gt;</p> <p>要素を繰り返すことは出来ない。</p>

## 6.2 メタデータ・スキーマ定義方法の標準化動向

メタデータ・スキーマの定義方法は、領域、組織ごとにさまざまである。また多く記述規則は人間が読んで理解することを前提としており、コンピュータによるメタデータの検証に用いたり、他組織の記述規則を再利用することが難しい。

メタデータを共有し、相互運用性を高めるためには、メタデータ・スキーマ定義のための標準方法が必要となる。この標準化の動向を概観する。

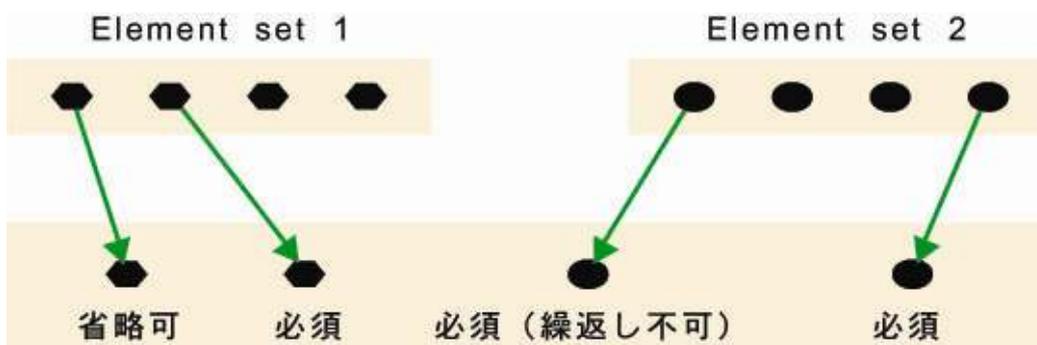
### 6.2.1 アプリケーション・プロファイル

メタデータを記述するためのスキーマは

1. メタデータ語彙
2. 構造的な制約の定義
3. コンピュータ・ネットワーク上での具象的表現形式

で成り立つが、従来のメタデータの場合、自組織で記述するメタデータのニーズに合わせ、これら(特に 1 と 2)が一体的に定義されてきた。しかし実際には、他組織でも同様のメタデータ語彙を定義していたり、領域を超えた相互運用を図るためには可能な範囲で標準的な語彙を利用することが望ましい。一方、他組織で定義されたスキーマを利用する場合、語彙はそのままに、制約定義を部分的に変更したいこともある。

語彙定義と構造制約の定義を分離することで、相互運用性を高めることが可能になる。複数の語彙定義の要素集合から、実際の応用(アプリケーション)ごとに必要なものを選び、構造制約を定義するという考え方を「アプリケーション・プロファイル」(AP)と呼ぶ。(図 6.4)



◆ 図 6.4: 複数の要素セットから応用ごとに要素を選び、構造規則を定義する。

### 6.2.2 DCMI シンガポール・フレームワーク

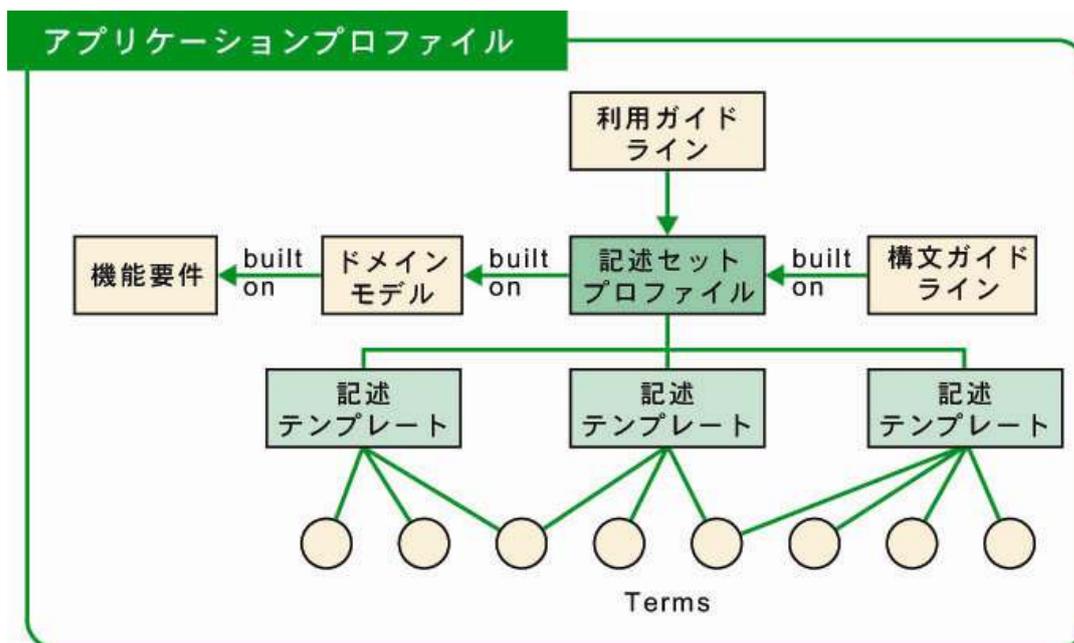
2007 年にシンガポールで開催された DCMI (Dublin Core Metadata Initiative) の国際会議において、このアプリケーション・プロファイルを定義するための枠組みが検討され、「シンガポール・フレームワーク」として文書化された<sup>13</sup>。このフレームワークは、次の 5 つの要素から構成される。(図 6.5)

- 利用ガイドライン(省略可): AP 全般の応用方法や AP に含まれる属性の提供方法等を示す
- 機能要件(必須): AP によって定義されるシステムやサービスにおいて実現が必要とされる機能を定義
- ドメインモデル(必須): AP によって定義されるシステムやサービスに含まれる基本的な実

<sup>13</sup> The Singapore Framework for Dublin Core Application Profiles,  
<http://dublincore.org/documents/singapore-framework/>

体とその実体間の関係を定義する

- 記述セットプロファイル(必須): AP によって定義されるシステムやサービスにおいて、メタデータの実体(レコード)の構造制約を定義し、妥当性検証を可能にする
- コード化構文ガイドライン(省略可): AP で示されるメタデータ・スキーマを、特定の応用システムやサービスとして実現するためのコード化方法やそのガイドラインを定義する



◆ 図 6.5:DCMI アプリケーション・プロファイル

このフレームワークでは、従来の「語彙から選んだ記述要素に、構造制約定義を加える」というアプリケーション・プロファイルを一歩進め、機能要件の定義、サービスの対象とする領域に含まれる実体の定義を行なって、システムやサービスを実現する上で必要な情報をまとめるものになっている。どの記述要素を用いてそこにどんな構造制約を加えるかという基本部分は、「記述セットプロファイル」において定義することになる。

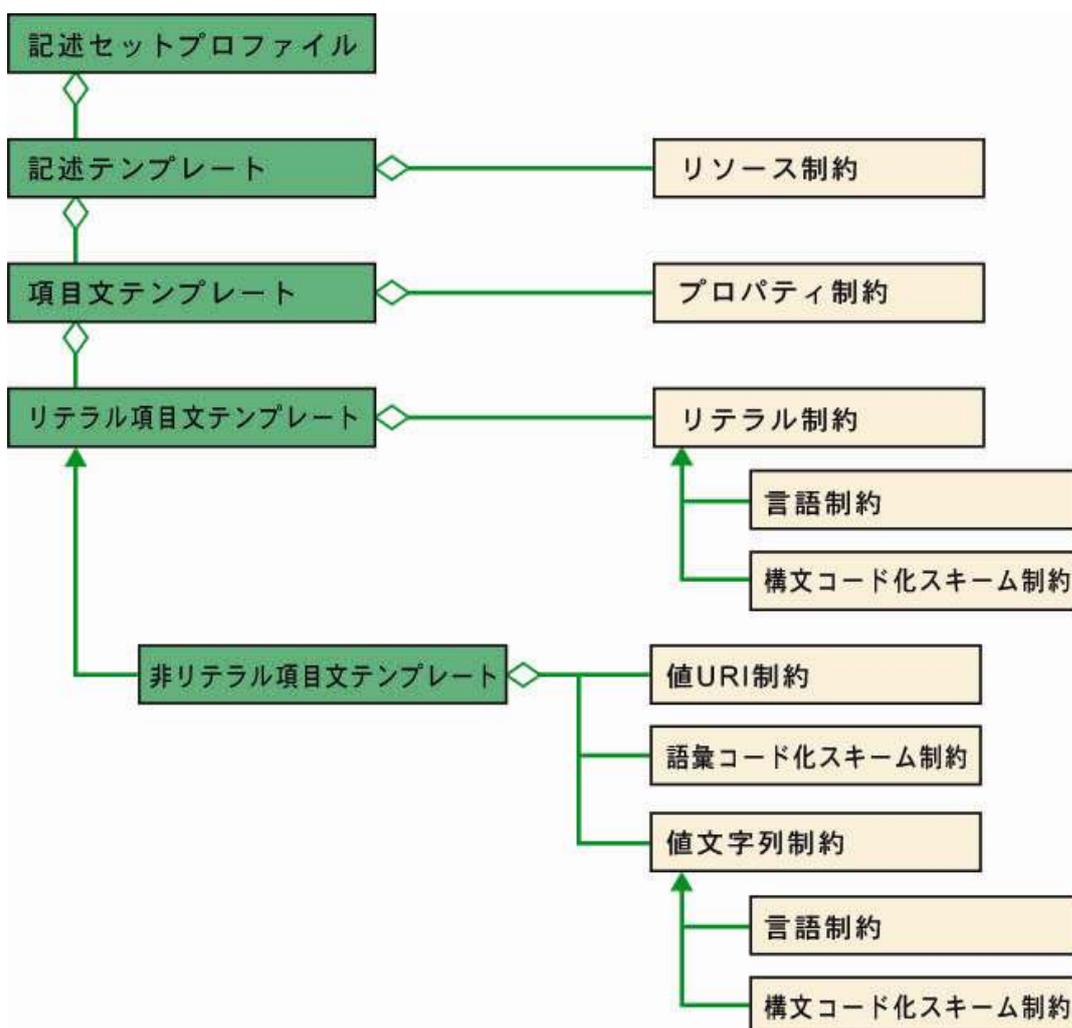
### 6.2.3 DCMI 記述セットプロファイル

シンガポール・フレームワークにおいて、メタデータの記述方法を機械可読な形で定義するために提示されたのが記述セットプロファイル(DSP, Description Set Profile)<sup>14</sup>である。あるリソースを記述するレコード(Description)は、1つの属性を記述する Statement(RDFにおける文=トリプル)の組み合わせで構成される。記述セットは、メタデータをこの記述レコードの集合として捉えるモデル

<sup>14</sup> Description Set Profiles A constraint language for Dublin Core Application Profiles, <http://dublincore.org/documents/dc-dsp/>

で、DCMI の抽象モデル<sup>15</sup>の基本をなす概念のひとつである。(図 6.6)

- **記述テンプレート**(Description Template) : ひとつのリソースを記述するための項目文の集合を示し、対象リソースに関する制約を表現する
- **項目文テンプレート**(Statement Template) : ひとつの記述(項目文)に適用される制約を表現する



◆ 図 6.6:DCMI の記述セットプロファイルの構成

記述セットプロファイルは任意の数の記述テンプレートで構成される。記述テンプレートは、対象リソースに関する制約と、任意の数の項目文記述テンプレートで構成される。

<sup>15</sup> DCMI Abstract Model, <http://dublincore.org/documents/2007/06/04/abstract-model/>

(a) **項目文テンプレート** 項目文記述テンプレートは、項目記述に用いるプロパティに関する制約と、プロパティの値に関する制約を持つ。項目文記述テンプレートには、「リテラル項目文記述テンプレート」と「非項目文記述テンプレート」の 2 種類がある。

- リテラル項目文記述テンプレートの値制約(リテラル制約)は、リテラル値に用いる言語の制約と、値の構文解釈(構文コード化スキーム)の制約がある
- 非リテラル項目文記述テンプレートの値制約には、次の 3 つがある
  - 値として URI を持つことができるかどうかの制約
  - 値としてとることができる語彙(語彙コード化スキーム)の制約
  - 非リテラル値実体を表す(代表する)値文字列の制約(リテラル制約と同じ)

非リテラル項目文における値文字列とは、たとえば作品の“タイトル”をリテラル値ではなく“タイトル文字列”と“読み”からなる構造化ラベルと定義した場合、“タイトル”項目文の値は非リテラル値(構造化ラベル)となり、その値文字列として“タイトル文字列”を持つ、という形で適用されるもので、DCMI の抽象構文の考え方にしただったモデルである。

あるレコード(実体のメタデータ)の記述制約は記述テンプレートによって表現され、その記述テンプレート内にはレコード記述に用いる項目に対応する複数の項目文テンプレートが含まれる。項目の値が別の実体(たとえば著者項目の値が人物)である場合、さらにその実体を記述するための記述テンプレートを入れ子的に定めることができる。

(b) **記述例** この DSP を XML あるいは RDF として記述する構文は、2008 年に草案が出されたまま未完成の状態になっている。この草案に示される XML 構文を用いて、国立国会図書館アプリケーション・プロファイルの「タイトル」項目文定義

- プロパティとして <http://purl.org/dc/terms/title> を用いる
- 語彙符号化スキームは指定しない
- URI による記述は不可
- 任意の文字列による記述は可
- 構文符号化スキームによる記述は不可
- 入れ子による記述は可
- 入力レベルは必須

を表現すると、およそ次のようになるだろう。

```

<DescriptionTemplate ID="DCNDL-AP">

  <StatementTemplate ID="Title" minOccurs="1" type="nonliteral">
    <Property>http://purl.org/dc/terms/title</Property>
    <NonliteralConstraint>
      <ValueURIOccurrence>disallowed</ValueURIOccurrence>
      <VocabularyEncodingSchemeOccurrence>
        disallowed
      </VocabularyEncodingSchemeOccurrence>
      <ValueStringConstraint minOccurs="1"/>
    </NonLiteralConstraint>
  </StatementTemplate>
  ...
</DescriptionTemplate>

```

#### 6.2.4 OWL を用いた制約記述

OWL のクラス定義では、プロパティの値に関する制約を記述することができる(「4.3.2.OWL」参照)。これは、プロパティの持つ値、あるいは出現回数によってクラスを表現するものであり、DSP の項目文記述テンプレートが表現する制約と同様の機能を持つと考えることができる。

たとえば《「タイトル」はダブリンコアの title プロパティを用い、入力レベルは必須》という制約記述は、次のように表現できるだろう。

```

<#Title>
  a owl:Restriction ;
  owl:onProperty <http://purl.org/dc/terms/title> ;
  owl:minCardinality 1 .

```

開世界仮説(ウェブ上には未知の情報が多数あるので、手元の情報だけで定義が完結するとは限らない)に基づくOWLにおいては、これは明示されない値を推論する(メタデータにタイトルが記述されていなければ、それはどこか別の場所で記述されている)手段を提供するものであって、本来は検証のための制約ではない。しかしレジストリなどのシステムでの応用としては、前提を閉世界(必要な情報はレジストリに揃っている)と置き換え、プロパティ値の検証に利用することもできる。

#### 6.3 メタデータ記述規則定義言語の提案

メタデータ記述規則の構造的制約の定義はXMLスキーマでも可能であるが、メタデータをRDFで表現する場合、モデルをXMLスキーマで定義することはできない。一方、RDFスキーマでは語

彙とモデルの定義はできるが、制約記述を十分表現できない。DCMI 記述セットプロファイル (DCMI-DSP)には、RDF による定義も提案されているが、2008 年の草案のままであり、十分な機能を果たすものには至っていない。

現状ではメタデータ記述規則の構造的制約を定義する言語が十分標準化されているとはいえないので、本プロジェクトでは、DCMI-DSP を視野に入れつつ、新しい定義言語を定め、提案する。

### 6.3.1 メタデータ記述規則定義言語の要件と考え方

メタデータ記述規則定義言語は、さまざまな組織の記述規則がそれぞれの方法において表現している制約条件などの定義を、標準的で機械可読な形で表現する必要がある。本プロジェクトで提案する定義言語は、次の要件を満たすものとする。

- ひとつのメタデータ・レコード (DCMI-DSP における記述セット) を記述する規則を表現できる。レコード記述規則 (DCMI-DSP における記述テンプレート) は、レコードを構成する項目の記述規則 (項目記述規則、DCMI-DSP における項目文テンプレート) と、レコード全体としての規則から成る
- 項目記述規則は、
  - 項目の記述に用いるプロパティと、その出現回数制約 (入力レベル)、および値の制約を表現できる
  - 項目の値として構造化した記述 (たとえば「作者」項目の値が「氏名」「連絡先」などの入れ子項目を持つ) を表現できる
- 定義した記述規則に基づくデータが、制約条件を満たしているかどうか機械的に検証できる
- 記述規則の作成日などのメタデータを表現でき、バージョン管理ができる
- 他のシステムとの相互運用性が高く、汎用ツールが利用できるフォーマットを用いる

これらの要件を満たすために、記述規則の定義は OWL を用いて表現する。レコード記述規則を OWL のクラス定義、項目記述規則をクラスのプロパティ制約として表現することで、汎用の RDF、OWL ツールを利用可能とする。また記述規則全体を OWL のオントロジーとして表現することで、メタデータの記述やバージョン管理を可能とする。

記述規則定義言語は次の名前空間で定義するものとし、ここでは接頭辞 dsp: を用いて表現する。

<http://purl.org/metainfo/terms/dsp#>

DCMI の DSP (DCMI-DSP) に対して、この記述規則定義言語を OWL-DSP と呼ぶことにする。

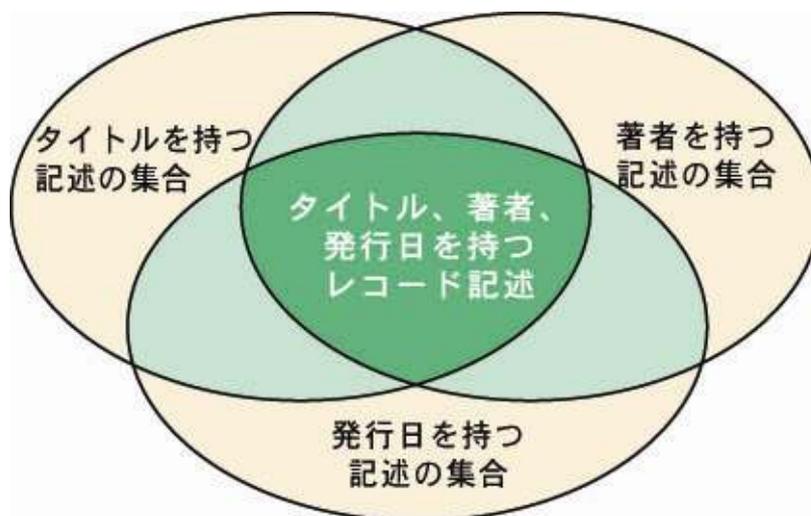
### 6.3.2 OWL クラス定義によるレコード記述規則の表現

レコード記述規則は、ひとつのレコードとして記述されるリソースについて

- リソースを記述するために用いる項目記述規則
- リソースの属するクラス

を表現する。また、項目の値を構造化記述する場合の規則もレコード記述規則として表現するために、

- リソースを空白ノードとして表現できるかどうか
- も表現可能とする。
- レコード記述規則は OWL のクラスとして表現し、項目記述規則を表現するクラスと `rdfs:subClassOf` で結びつける(図 6.7)のように、各項目記述規則クラスの制約を満たす記述の共通部分がレコード記述規則となる)。



- ◆ 図 6.7:タイトル、著者、発行日を持つレコードは、それぞれの項目を持つ記述の集合の共通部分として表現される

リソースの属するクラス、空白ノードの可否は、それぞれプロパティ `dsp:resourceClass`、`dsp:valueURIOccurrence` を用いて表現する。また、レコード記述規則は DCMI-DSP の記述テンプレートに対応することを示すために、`owl:Class` のサブクラスとして `dsp:DescriptionTemplate` を定義し、これを用いて規則を表現する。

タイトル、著者、発行日項目を持ち、`foaf:Document` クラスとして記述し、URI が必須であるレコードの記述規則は、OWL-DSP で次のように表現できる。

```
<#レコード> a dsp:DescriptionTemplate ;
  dsp:valueURIOccurrence "mandatory" ;
  dsp:resourceClass foaf:Document ;
  rdfs:subClassOf <#タイトル>, <#著者>, <#発行日> .
```

(ここで<#タイトル>、<#著者>、<#発行日>はそれぞれの項目記述規則の URI を示す)

### 6.3.3 プロパティ制約による項目記述規則の表現

項目記述規則は、ひとつの項目について

- 適用するプロパティ
- プロパティの出現回数(入力レベル)の制約
- プロパティの値の制約

を定義する。これは OWL クラスのプロパティ制約(owl:Restriction)で表現できるが、OWL1 では値制約と回数制約を同時に表現できないため、OWL2 の制約方法を用いる。

項目記述規則は DCMI-DSP の項目文記述テンプレートに対応することを示すために、owl:Restriction のサブクラスとして dsp:StatementTemplate を定義し、これを用いて規則を表現する。

適用プロパティは owl:onProperty の値として表現する。

出現回数は owl:qualifiedCardinality (最小、最大ならばそれぞれ owl:minQualifiedCardinality、owl:maxQualifiedCardinality) 値として表現する。実際の記述規則には「推奨」「あれば必須」など、数値としての制約が表現できない“入力レベル”が示されることがしばしばあるので、これを規則に組み込むために dsp:cardinalityNote を導入し、リテラル値としてその“入力レベル”をそのまま表現する。

値制約は、項目値がリテラルである場合は owl:onDataRange を、非リテラルである場合は owl:onClass を用いて表現する。

(a) **リテラル制約** DCMI-DSP では、リテラル値の制約として言語の制約と、値の構文解釈(構文コード化スキーム)の制約を示す。OWL-DSP では、前者を dsp:langTagOccurrence で示す。後者は、構文コード化スキームを RDF のデータ型として扱い、owl:onDataRange の値として示す。たとえば「発行日」を

- プロパティとして dcterms:issued を用いる
- 入力レベルは「あれば必須、最大 1 回」
- 値の構文は XML スキーマの date 型

という制約によって表現する場合、OWL-DSP は次のようになる。

```
<#発行日> a dsp:StatementTemplate ;
  rdfs:label "発行日" ;
  owl:onProperty dcterms:issued ;
  dsp:cardinalityNote "あれば必須" ;
  owl:maxQualifiedCardinality 1 ;
  owl:onDataRange xsd:date ;
  rdfs:comment "文書の発行日".
```

(b) **非リテラル制約** DC-DSP での非リテラル制約は、

- 値として URI を持つことができるかどうかの制約
- 値としてとることができる語彙（語彙コード化スキーム）の制約
- 非リテラル値実体を表す（代表する）値文字列の制約（リテラル制約と同じ）

の 3 つの制約を表現する。DC-DSP の XML 構文では、この制約を `NonLiteralConstraint` 要素内に直接記述することも、`descriptionTemplateRef` 属性を用いて別途定義する記述テンプレートを参照することもできる。

OWL-DSP においては、非リテラル値の制約は、値として用いるレコード記述規則を定義することと同等になる（DCMI-DSP での `descriptionTemplateRef` に相当）。URI を持つことができるかどうかは `dsp:valueURI` Occurrence、値文字列はレコード記述規則のリテラル制約として記述できる。

語彙コード化スキームは、レコード記述規則の `dsp:resourceClass` に相当するが、実際には語彙コード化スキームはクラスではなく、NDLSH のようなシソーラス語彙が指定されることが多い。そこで OWL-DSP では、`dsp:inScheme` プロパティを導入して、非リテラル値の語彙コード化スキーム URI を指定できるようにする。

非リテラル制約は `owl:onClass` の値として表現する。この値は、別途定義するレコード記述規則への参照であっても、既存のクラス URI であっても、匿名レコード記述制約として直接記述してもよい。

「タイトル」の制約定義が

- プロパティとして `dcterms:title` を用いる
- 入力レベルは「必須、1 回のみ」
- 値は構造化し、タイトル文字列と読みを持たせる

であるとき、OWL-DSP では次のように表現できる。

```
<#タイトル> a dsp:StatementTemplate ;
  rdfs:label "タイトル" ;
  owl:onProperty dct:terms:title ;
  owl:qualifiedCardinality 1 ;
  owl:onClass <#構造化タイトル> ;
  rdfs:comment "文書の表題" .
```

構造化したタイトルは、

- タイトル文字列を `xl:literalForm` で必ず 1 回記述する
- 読み文字列を `dcndl:transcription` で最大 1 回記述する

と定義されているとすれば、次のレコード記述制約として表現できる。

```
<#構造化タイトル> a dsp:DescriptionTemplate ;
  rdfs:subClassOf [
    owl:onProperty xl:literalForm ;
    owl:qualifiedCardinality 1 ;
    owl:onDataRange rdfs:Lialal
  ], [
    owl:onProperty dcndl:transcription ;
    owl:maxQualifiedCardinality 1 ;
    owl:onDataRange rdfs:Lialal ;
  ] .
```

構造化タイトルの項目記述規則も `dsp:StatementTemplate` だが、別途定義せずに、上記のように匿名制約クラスとして表現できる。

### 6.3.4 管理情報の表現

記述規則は OWL のオントロジー (`owl:Ontology`) として表現し、OWL のオントロジー管理語彙 (「4.3.2.OWL」参照) やダブリンコア語彙を用いて、バージョン、作成日などのメタデータを記述する。

記述規則 (スキーマ) は一度作成したら不変とは限らず、必要に応じて語彙を追加したり、定義の細部を修正する場合がある。更新内容によっては、古いバージョンの規則に準拠して書かれたメタデータが、新バージョンでは正しく解釈できなくなる可能性もある。メタデータを長期にわたって利用可能にするため、適切な情報の提供が必要である。

最新バージョンと直前のバージョンは、`owl:priorVersion` を用いて関連付けておく。スキーマの名前空間 URI が `http://example.org/terms/ex#` である場合、2011-02-14 版の最新バージョンには、

2010-08-31 版の旧バージョンとの関係を次のように記述する。

```
<http://example.org/terms/ex#>
  a owl:Ontology ;
  owl:versionIRI <http://example.org/terms/ex-20110214#> ;
  owl:priorVersion <http://example.org/terms/ex-20100831#> ;
  owl:versionInfo "version 1.3, 2011-02-14" ;
  dcterms:created "2011-02-14" .
```

## 6.4 メタデータ記述規則の簡易表現

### 6.4.1 記述規則と OWL-DSP

メタデータ記述規則を OWL-DSP によって表現することで、コンピュータによる規則の参照、利用が可能になる。しかし一般的な記述規則は表形式で定義されていることが多く、また OWL による記述は専門知識がないと難しい場合があるので、表形式 (TAB 区切りテキスト) の記述規則を OWL-DSP に変換できることが望ましい。

そこで、表から OWL-DSP への変換を確実にこなうために、規則記述表の書式を定め、この書式に従った記述規則をそのままレジストリに登録できるようにする。ここではこの書式を「簡易 DSP」と呼び、その書き方を定める。

### 6.4.2 データ項目と記述規則

多くの場合、メタデータは表の形で表現できる。たとえば簡単な書籍データとして、表 6.5 のような例が考えられる。

◆ 表 6.5: データ記述の例

書誌 ID	タイトル	著者	発行日	主題
b01234	音楽美学	ハンス・リック	2000-10-20	音楽
b01235	LOD 入門	団太郎	005-01-07	セマンティック・ウェブ
b01236	今晚のおかず	山本花子	2010-08-10	料理

このような書籍データを作成するための「記述規則」は、書誌 ID、タイトル、著者、発行日、主題という項目ごとに、値のタイプや必須／任意といった「制約」を表 6.6 のように示すことによって構成されるだろう。

◆ 表 6.6: データ記述規則の例

項目名	値タイプ	出現制約
書誌 ID	固有 ID	必ず 1 回
タイトル	文字列	必ず 1 回
著者	文字列	あれば必須、複数可
発行日	日付	あれば必須、複数不可
主題	NDLSH	任意、複数可

簡易 DSP はこうした表形式の記述規則における「値タイプ」「出現制約」などの記述方法を統一して、機械的な処理を可能にするものである。また、この規則を用いてデータを RDF によるメタデータに変換できるようにするため、各項目に対応する RDF のプロパティも記述するようにする。

#### 6.4.3 簡易 DSP の記述例

簡易 DSP は、表 6.7 の要素を用いて各項目の記述規則を定義する。

◆ 表 6.7: 簡易 DSP の要素

要素	役割
項目規則名	「タイトル」などの項目名
プロパティ	dc:title など、項目に対応する RDF プロパティ
最小出現回数	任意であれば 0、必須であれば 1 など
最大出現回数	繰り返し不可であれば 1、制約なしならば など
値タイプ	文字列、構造型(入れ子に相当)、参照値 (URI) など
値制約	文字列のデータ型 (xsd:date など) や値の語彙 (NDLSH など)
コメント	記述に関する補足説明

前項 6.4.2 の記述規則を、簡易 DSP を用いて表現すると表 6.8 のようになる。

◆ 表 6.8:簡易 DSP 記述例

項目規則名	プロパティ	最小回数	最大回数	値タイプ	値制約	コメント
書誌 ID		1	1	ID		文書の ID
タイトル	dc:title	1	1	文字列		文書の表題
著者	dc:creator	あれば必須	-	文字列		文書の作者
発行日	dc:issued	あれば必須	1	文字列		文書の発行日
主題	dc:subject	0	-	参照値	ndlsh:	文書の主題

ここで、プロパティの記述に用いている接頭辞は、(a)に定める標準名前空間であれば、そのまま用いることができる。

主題の値を国立国会図書館件名標目に限定するために、ndlsh:という接頭辞を用いて示しているが、これは標準名前空間としては定義されていない。そこで、表 6.9 のように未定義接頭辞を名前空間 URI に結びつける宣言を、簡易 DSP の先頭で行なうことができる。

◆ 表 6.9:簡易 DSP の名前空間宣言の例

接頭辞	名前空間 URI
ndlsh	<a href="http://id.ndl.go.jp/auth/ndlsh/">http://id.ndl.go.jp/auth/ndlsh/</a>

簡易 DSP ファイル内で、名前空間宣言と記述規則のブロックを区別するために、それぞれの前に[@NS]、[MAIN]という識別子を置く。

#### 6.4.4 OWL-DSP への変換

簡易 DSP の「プロパティ」欄の定義を利用することで、規則に基づいて記述されたデータを RDF (OWL-DSP) に変換することができる。たとえばデータ例 1 行目の書籍は、次のような RDF となる。

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix ndlsh: <http://id.ndl.go.jp/auth/ndlsh/>.

<b01234>
  dc:title "音楽美学";
  dc:creator "ハンス・リック";
  dc:date "2000-10-20"^^xsd:date;
  dc:subject ndlsh:音楽.
```

ここで、この書籍リソースは相対 URI で示されているが、メタデータを公開するためには絶対 URI とする方が望ましい。また、書籍リソースに foaf:Document などの型を与えたい場合も多い。

値タイプが ID である項目規則 (ID 型項目規則) は、RDF 変換のために、これらの情報を記述しておくことができる。ID 型項目規則においては、

- プロパティ欄に、レコードの型 (クラス) を示すことができる
- 値制約欄に、レコード ID と組み合わせる名前空間接頭辞を示すことができる

という特別な記法を適用する。名前空間宣言に

ndllbooks	http://iss.ndl.go.jp/books/
-----------	-----------------------------

を加え、ID 型項目規則を

項目規則名	プロパティ	最小回数	最大回数	値タイプ	値制約	説明
書誌 ID	foaf:Document	1	1	ID	ndllbooks:	文書の ID

とすることで、RDF は次のように変換される。

簡易 DSP を OWL-DSP に変換し、メタデータ・スキーマ・レジストリに取り込むための対応付けは、10.5 で定義する。また、これらの要素を反映した簡易 DSP の記述例は Appendix4.2.5 に示す。

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix foaf: <http://xmlns.com/foaf/0.1>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix ndlsh: <http://id.ndl.go.jp/auth/ndlsh/>.
@prefix ndllbooks: <http://iss.ndl.go.jp/books/>.
```

```
ndllbooks:b01234
```

```
  a foaf:Document;
  dc:title "音楽美学";
  dc:creator "ハンス・リック";
  dc:date "2000-10-20"^^xsd:date;
  dc:subject ndlsh:音楽.
```

## 7. メタデータ情報共有のためのガイドラインの策定

メタデータの共有と円滑な流通のためには、記述の標準が必要である。しかし標準を定めただけでは、実際のスキーマ設計やメタデータ作成、さらにその利用において、その標準をどう適用してよいのか不明だったり、迷ったりする 경우가少なくない。

そこで本プロジェクトでは、標準に基づいてメタデータおよびスキーマを作成、利用するための参考、指針となるガイドラインを作成する。尚、作成したガイドラインは、別添の「メタデータ情報基盤構築事業 報告書Appendix」にA3として添付している。

### 7.1 ガイドライン作成の準備と検討

ガイドラインの基本方針の立案と、本体の執筆のために、事例調査、骨子案の作成と討議を行ない、改訂を繰り返しながら作成した。

#### 7.1.1 ガイドラインの事例検討

現在「メタデータ作成のガイドライン」と称するものは比較的多く見られる。しかしその大半は 6.1 で検討したような「メタデータ記述規則」であり、個々の記述項目に関する規則や推奨記述は詳しく説明されているが、メタデータ作成一般に関する指針を示しているものはほとんどない。基本的にはそれぞれの組織におけるメタデータ作成指針であって、他組織が応用できるガイドラインは少数である。

ガイドラインの事例を検討した中から、本プロジェクトでのガイドライン作成の参考となるものを以下に示す。

(a) **junii2 ガイドライン** 国内におけるメタデータ交換のためのガイドラインとしては、6.1.6 でとりあげた国立情報学研究所の「junii2 ガイドライン」が挙げられる。

これは、ガイドラインの位置づけとして「junii2 メタデータ・フォーマットの項目説明と使用例を示したもの」と述べられているとおり、junii2 スキーマの記述規則に他ならない。しかしその背景として、

これまでは、各機関リポジトリのメタデータから junii2 用のメタデータを出力するにあたり、junii2 側の記述方法が詳述されていなかったため、結果として各機関リポジトリからハーベストしたデータの記述が統一されておらず、JAIRO、CiNii 等、NII のサービス基盤を効果的に使えない可能性があります。

と分析していることから、共有のための指針策定において参考になるものといえる。

具体的には、メタデータの共有、連携のためには、形式的スキーマだけでなく、推奨使用例、非推奨使用例などの指針が必要であることが分かる。また

「junii2 ガイドライン」の creator 項目に準拠していただくことで、CiNii 著者検索において機関リポジトリ由来のデータが正しく検索できるようになることが期待されますと指針に準拠することによって得られるメリットを示すのも、利用者にガイドラインにしたがってもらうためには重要な点といえる。

**(b) CrossRef のガイドライン** 電子論文の引用リンクによる相互参照システムを提供する CrossRef は、メタデータの質を高め、サービスを向上させるためのガイドとして、シンプルな 9 つの指針を示している<sup>1</sup>。

1. 記事のタイトルを登録する
2. 著者全員の名前を(第 1 著者だけでなく)登録する
3. 掲載誌、書籍、カンファレンスとのタイトルを確認する
4. 参照リンク発行者は、自身の引用も含め、「すべての」参照を登録する
5. URL はグーグルで検索できるものを登録する
6. 補助的素材(コンポーネント)の DOI を登録する
7. デジタル化されたすべてのバックナンバーのメタデータと DOI を登録する
8. タイトルリスト(CrossRef が提供するタイトル検索)をチェックする
9. 登録に失敗した記事のリストをチェックする

これはメタデータ記述方法のガイドラインというよりも、メタデータの質を高めるためのチェックリストのようなものだが、さまざまなソースのメタデータを共有・再利用するために、それぞれの品質を向上させることは重要である。メタデータの作成におけるガイドラインとして参考にできる。

**(c) 画像メタデータの扱いに関するガイドライン** デジタル画像は、通常ファイル内にメタデータを格納することができるが、デジタル写真の Exif/TIFF、アドビシステムズ社による XMP、国際新聞電気通信評議会が制定した IPTC-IIM などの複数のメタデータ規格があり、各規格間の互換性は十分保たれていない。そこで、画像データのメタデータの互換性に取り組むコンソーシアム Metadata Working Group(MWG)が 2006 年に作られ、Guidelines For Handling Image Metadata が 2008 年に発行された<sup>2</sup>。

ガイドラインではまず画像の作成者、変更者、利用者というアクターを定義し、それぞれがメタデータの処理においてどのような役割を担うかを明確化した。さらに複数のフォーマットおよびその組み合わせにおいてメタデータをどう処理すべきかを示したうえで、主要な共通メタデータ項目について、各フォーマットでどのように表現されているか、どのように扱うべきか、どのような制約

<sup>1</sup> Metadata Guidelines, [http://www.crossref.org/02publishers/metadata\\_guidelines.html](http://www.crossref.org/02publishers/metadata_guidelines.html)

<sup>2</sup> Metadata Working Group (MWG), <http://www.metadataworkinggroup.org/>

があるかといった指針を示している。取り上げられている項目は次の 11 である。

1. Keywords
2. Description
3. Date/Time
4. Orientation
5. Rating
6. Copyright
7. Creator
8. Location
9. Image
10. Hierarchical
11. Collections

これはメタデータ作成ではなく利用のためのガイドラインであるが、異なる形式のメタデータ間どのように共通点を生かし、共有していくかという視点は本ガイドラインとも共通する。またアクターを示すことで、指針の対象が明確になるという点も参考になる。

各指針は分かりやすく整理され、図表やコード例も含めて丁寧に説明されている。

### 7.1.2 作成の経過

(a) **骨子案の作成** ガイドライン WG において 12 月初旬から骨子案の作成に取り掛かり、メタデータの設計、作成、利用、運用のライフサイクル 4 段階(「7.2.1.ガイドラインの目的と役割」参照)について、指針案を検討。計 17 の指針からなる骨子案を作成した。

(b) **検討会での意見聴取** 1 月 31 日に実施した検討会において、ガイドラインの基本方針と骨子案を説明し、討議を行なった。

実際にメタデータを提供する立場から、

- メタデータ作成のためにスキーマを定義する必要があるが、メタデータの例を見ながらの手探りになるので、指針があるとありがたい
- 位置情報も標準が欲しい。住所の書き方はどうか
- 副題の書き方、日本語と英語で併記されているような事例など、どのように書くのかをまとめてほしい
- 図書館的な印象。美術館であれば主題、分類といった言葉の印象が異なるので、そういった事も念頭に置いてほしい
- 共通語彙間の連携を意識して欲しい

- 分かりやすくする工夫が必要

などの意見を受けた。

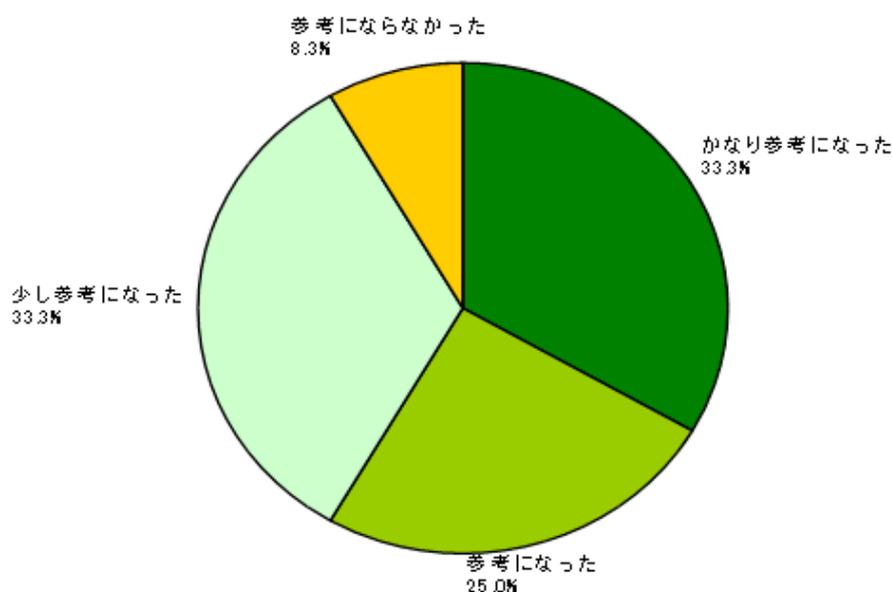
(c) **ガイドライン原案の作成と研究会での意見聴取** 検討会での意見をふまえてメタデータ設計・作成の現場でのヒアリングを行ない、ガイドライン原案を作成した。原案では指針を 19 に拡大した。

原案を 2 月 21 日の研究会に提示し、意見を受けた。

- 必須指針と推奨指針を分けるなどのレベル設定が必要ではないか
- 技術的な内容は読者によっては難しすぎるので、分離したほうがよいのではないか(一般指針と技術指針の分離方針は、この研究会後に導入した)

などの意見が出され、継続してメーリングリストで意見を受けることとした。

(d) **メタデータ・スキーマ・レジストリ実証実験** レジストリの実証実験(「13.サービス(レジストリ)の実証」参照)において、実証者にガイドラインのドラフトを配布し、スキーマ登録、編集実証の際の参考にしてもらった。簡易記述規則作成の参考になった度合いを質問したところ、92%から何らかの参考になったとの回答を得た(図 7.1 図 1.2)。



◆ 図 7.1: 実証アンケート:記述規則作成にあたって、ガイドラインは参考になりましたでしょうか？

また、レジストリを利用するにあたって、スキーマ登録も専門家が行なうとは限らないこと、なぜ

RDF による記述が望ましいかなどの普及啓発活動が重要といった意見が寄せられた。

これを受け、ガイドラインの第 1 章に「メタデータとは何か」の説明を加えるとともに、第 4 章に RDF の解説をおき、RDF を用いた記述によりどのようなメリットがあるかの説明を加えることとした。

(e) **ガイドラインのとりまとめ**討議、意見交換、メーリングリストでのフィードバック、実証実験を受け、原案の内容を基本指針と技術詳細に分割した。またガイドラインを理解するためには、メタデータとは何か、メタデータ共有はなぜ重要かといった基本的な背景説明を加える必要があることが分かったため、冒頭のメタデータについての概説を補強した。

これらの改定を加えた最終案を、3 月 23 日の検討会に提出し、最終的な意見交換、調整を行なった。

構成の改良は評価されたが、ガイドラインの技術詳細説明において、細部に至って解説しているセクションと、概要にとどまっているセクションの差があることが指摘された。これを受け、ガイドラインの構成説明において、技術詳細は「関連するメタデータ技術を網羅するものではなく、疑問が生じやすい点、個別の記述規則などであまりカバーされていない点を中心に解説する」旨を追記した。

## 7.2 ガイドラインの基本方針

プロジェクトの目的と事例の検討結果を踏まえ、ガイドラインの基本方針を策定した。方針および構成は、検討会、研究会での議論を受けて修正を加えたが、ここでは 3 月 23 日の検討会での最終案を報告する。

### 7.2.1 ガイドラインの目的と役割

このガイドラインは、デジタルコンテンツに関わる様々な機関(図書館、公文書館、美術館、博物館、サービス事業者、出版社等)におけるメタデータの相互利活用のために広く利用されるものでなければならない。

メタデータとのかかわりは、コンテンツを提供する立場か、消費する立場か、また領域における標準化を推進する立場か、標準に従ってデータを作成する立場かなどによってさまざまである。そこで本ガイドラインは、メタデータ作成のライフサイクルを次の 4 つで考え、それぞれに対応する指針を検討する。

- メタデータ・スキーマの設計
- メタデータの作成・公開
- メタデータの利用
- メタデータ・スキーマの運用・管理

全体を通して、相互運用性を確保するために標準的な語彙、記述手段を用いたスキーマとメタ

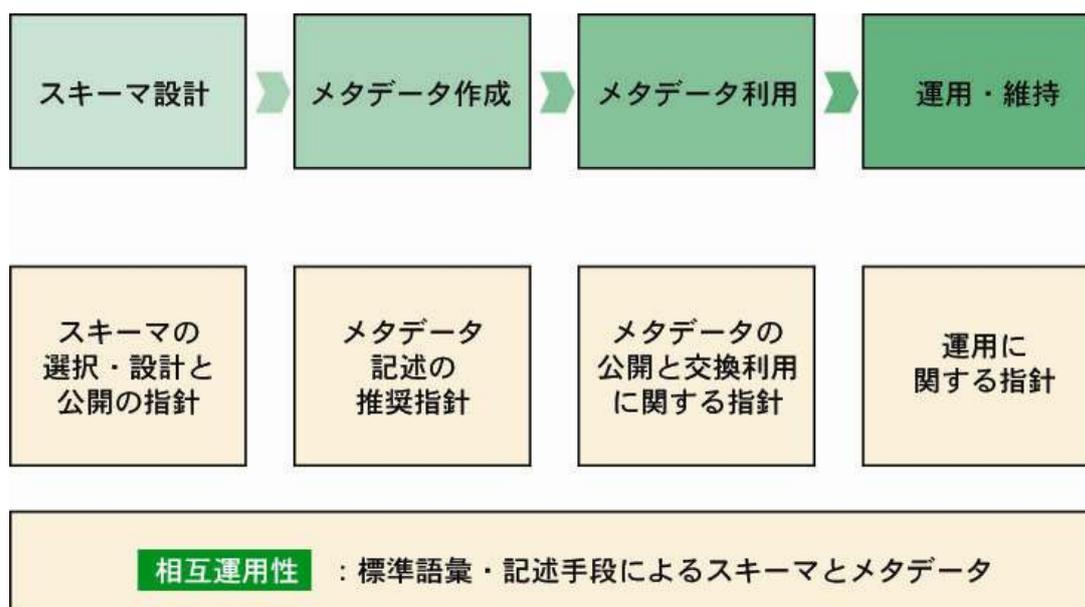
データとすることを基本とし、ステージごとの指針を検討する(図 7.2)。

このガイドラインは、次の目的を達することができるように構成し、作成する。

- コンテンツ提供者が、そのメタデータを公開する際に、共有・再利用性の高い記述をするためのよりどころとなる。
- 利用者がコンテンツを探索・選択する際に、試行錯誤ではなく標準的な手順でメタデータを利用し、効率的な発見・活用が可能となる。
- 分野の違いを超え、ネットワーク上でメタデータを幅広く共有できるようにする。

### 7.2.2 このガイドラインの構成

本ガイドラインは、メタデータの提供者、利用者双方を対象に、メタデータの設計、作成から利用、運用管理まで、メタデータの相互運用性、長期利用可能性を高めるための指針を示す。対象とする範囲が広く、一般的な原則から具体的な記述のための技術上の注意点までを扱うため、次のように構成する。



◆ 図 7.2: メタデータのライフサイクルとガイドライン

(a) 章の構成 全般的な説明から技術的な詳細に段階的に章を構成する。

1. **はじめに**: メタデータとその共有についての背景説明を最初におき、ガイドラインの基本的な考え方を述べた上で、指針を一覧する。また、本ガイドラインで用いる用語を簡単に説明する
2. **ガイドライン**: メタデータ共有のための指針を説明する。技術的な詳細には踏み込まず、メ

タデータ一般を扱う上での推奨事項を解説する

3. **ガイドラインの技術詳細**:メタデータを実際に設計、作成したり利用する際に必要になる技術的細部について、推奨事項を説明する。関連するメタデータ技術を網羅するものではなく、疑問が生じやすい点、個別の記述規則などであまりカバーされていない点を中心に解説する。項目番号はガイドラインと対応している。
  1. **RDF について**:本ガイドラインで推奨するデータモデル RDF (Resource Description Framework)について、その概要と記述上の留意点を解説する
  2. **メタデータ語彙**:メタデータの記述によく用いられる語彙について簡単に紹介する
  3. **メタデータ・スキーマ言語**:記述規則を定義するために本ガイドラインで推奨する定義言語および簡易記述方法についての詳細を説明する

(b) **ガイドラインの内容** 指針を枠で囲んで提示し、背景や狙いを述べた上で、指針の細目（推奨するポイント）を箇条書きで示す。細目の前に、対象がメタデータの提供者（設計者）なのか利用者なのかを明示する。

指針の基本内容を示す文

**優先度**:A~C

指針の背景や狙いなどを、1~2 段落程度で説明した文章。……

メタデータ提供者 | 利用者は:

1. 指針の細目
2. もうひとつの細目
3. ………

必要に応じて、細目を設けてより詳しく説明する。具体的な記述例や技術解説は、同じ指針番号に対応する技術詳細において説明する。

(c) **優先順位** 本ガイドラインをメタデータの設計、作成、利用に取り入れるにあたって、一度に実施することが難しい場合は、段階的に推奨時効を採用していくことも考えられる。その際の参考となるように、各指針に A、B、C の優先順位を示し、取り組みのプランを立てやすくした。

(※優先度 C は重要度が低いということではなく、A、B の指針を実践してからのほうが効率がよい、というほどの位置付けである)

## 7.3 基本ガイドライン

ガイドラインの中核である第 2 章(技術詳細を除く、メタデータ一般を扱う上での推奨事項の指

針)を掲載し、報告する。

### 7.3.1 スキーマの選択・設計と公開の指針

交換のためのメタデータ設計について、記述語彙の選択・設計、記述規則の定義、シンプルなメタデータ・フォーマットへの変換に関する指針を示す。

#### (a) スキーマの選択と設計

スキーマを相互運用可能な形で選択・設計する。

##### 優先度:A

メタデータを設計する際に、当面扱う資料や組織内での使用のみを念頭に記述規則を決めてしまうと、外部とのメタデータの交換や共有が難しくなる恐れがある。将来メタデータを他組織と交換したり、広く公開して流通させる可能性を考えて、できるだけ標準的な語彙を選択し、相互運用可能な形で記述規則を設計する。

##### メタデータ設計者は:

1. 同じ領域、もしくは近隣領域で用いられている相互運用可能な既存スキーマがあれば、できる限りそれを採用する
2. スキーマを新たに定義する場合は、相互運用可能な形を考える
3. 記述規則に用いる語彙は、可能な限り広く普及した標準語彙を用いる
4. 領域固有の項目(プロパティ)を定義する場合は、共有可能な標準語彙とのマッピングを留意する

#### (b) 相互運用性を尊重した記述規則設計

新たなスキーマを設計する場合、使用語彙や参照記述規則の定義を尊重し、メタデータを相互運用できるよう設計する。

##### 優先度:A

既存スキーマを利用してメタデータを設計する際に、語彙の定義や記述規則の制約が部分的にニーズに合わない場合があり得る。このとき、記述規則で元語彙の定義と矛盾する規則を定めると、同じ語彙を用いている他のメタデータとの相互運用が困難になってしまう。非互換を避けるため、使用語彙や参照記述規則の定義は尊重する。

**メタデータ設計者は：**

1. 記述規則で既存語彙のタームを用いる場合は、タームの定義を尊重する
2. 既存語彙の定義がニーズに合わない場合は、独自語彙を定義して、標準(既存)タームと関連付ける

**(c) 領域固有の知識なしでも理解可能なスキーマ設計**

独自スキーマを設計する場合も、特定領域の知識なしに理解し、交換可能なフォーマットに変換するための情報・規則を用意する。

**優先度：B**

扱うコンテンツの特性や組織で利用するための要件によって、メタデータ記述に用いる語彙、構造はさまざまに異なる。これらの異なるフォーマットのメタデータを相互利用可能な設計が必要である。

詳細なデータを表現するためには、領域固有の専門用語による語彙が必要な場合もある。メタデータを広く共有・交換するためには、専門外であっても最小限の共通理解を可能にするための情報も提示されていることが望ましい。

**メタデータ設計者は：**

1. 未知の領域の語彙を利用者(人間)が理解するための説明を提供する
2. コンピュータ処理のために、何らかのレベルで構造を単純化した共通項を得られるようにする

**メタデータ利用者は：**

1. メタデータの理解や異なるデータの組み合わせのために、スキーマで提供される情報を利用する

**読んで理解できる説明の付与** 未知の領域のメタデータを利用するためには、まず人間の利用者がそのメタデータの記述語彙を理解する必要がある。語彙、記述規則の定義には、機械可読情報に加えて、表示用のラベルと理解のための説明(コメント)を付与する。

領域外からの利用も考慮し、公開スキーマには、非専門家であっても基本的な意味を理解できるような補助説明、もしくは解説への参照を加えることが望ましい。

※この指針の狙いは、特定領域でしか扱わない特殊な専門情報を非専門家に説明するというのではなく、専門的な用語ではあるが、一般的な言葉に言い換えも可能、あるいは一般的な用語が細分化・特殊化されたものであるとき、一般用語と関連付けて説明するということである。たとえば音楽データで「フォアシュピラー」(Vorspieler)というプロパティを用いるときに、それは「次席

奏者」の意味である、あるいは member のサブプロパティである、などの補助情報を提供すれば、領域外でもその意味を大まかに把握し、データを利用することが可能になる。専門知識がないと利用不能な情報については、“領域特有プロパティ”などのマークを与えておくが良い。

**構造を単純化した共通項** 異なるフォーマットのメタデータをコンピュータで利用する場合、処理のために、何らかのレベルで構造を単純化した共通項が必要となる。これは、固有のプロパティをより汎用的なプロパティに置き換えたり、構造化された値を単純な文字列値に変換することを含む（この単純化をダム・ダウンと呼ぶ）。

#### (d) 標準方法によるスキーマの定義

スキーマ定義を、コンピュータ処理可能な標準方法でも表現し、公開する。

#### 優先度:C

組織において一貫したメタデータを作成するためには、どのような記述項目でレコードを表現するか、それぞれの項目にはどのプロパティを用い、その値をどう記述するかを明示する記述規則が必要となる。

これらの記述規則はフォーマットがまちまちであり、規約として表現される内容も異なっているため、コンピュータによるメタデータの検証に用いたり、他組織の記述規則を再利用することが難しい。記述規則の表現方法も標準化することが、相互運用性の向上につながる。

#### メタデータ設計者は:

1. 人が読むための説明的記述規則だけでなく、コンピュータ処理、相互参照可能な標準形式の記述規則を用意する
2. 標準形式記述規則では、《項目規則名》、《プロパティ名》、《最小出現回数》、《最大出現回数》、《値タイプ》、《値制約》、《コメント(説明)》を示す
3. 標準記述規則を、コンピュータ処理しやすい形で公開する

#### メタデータ利用者は:

1. 語彙定義、記述規則定義を参照し、活用する手順を組み込む

記述規則をコンピュータにも利用できるようにするためには、項目ごとにセクションを分け、説明文と表を交互に提供する形よりも、すべての項目が一覧できる表の形が望ましい。たとえば、

◆ 表 7.1: 項目ごとに表を用いる記述規則

規則	規則内容
項目規則名	タイトル
プロパティ名	dc:title
最小出現回数	1
最大出現回数	1
値タイプ	文字列
値制約	なし
コメント	コンテンツを表すタイトル

という表を説明文とともに項目ごとに繰り返すよりも、次の一覧表のほうがコンピュータ処理には適している。

◆ 表 7.2: すべての項目が一覧できる規則表

項目名	プロパティ	最小回数	最大回数	値タイプ	値制約	コメント
書誌 ID		1	1	ID		文書の ID
タイトル	dc:title	1	1	文字列		コンテンツを表すタイトル
著者	dc:creator	あれば必須	-	文字列		作者を一人ずつ記述
発行日	dc:issued	あれば必須	1	文字列		コンテンツの発行日

人が参照するための記述規則に加え、別途定める「記述規則定義言語」(DSP 言語)を用いた RDF 形式、もしくは「簡易 DSP」の書式による一覧の形の規則表を、コンピュータ可読な形で用意する。表 7.2 は簡易 DSP を用いた記述の一例である。

### 7.3.2 メタデータ記述の推奨指針

メタデータを共有・交換目的で公開する場合の推奨記述方法の指針を示す。個別の目録、データベースにおける記述は必ずしもこの指針に沿う必要はないが、公開に際しては指針に対応するよう変換できることが望ましい。

#### (a) リソースの識別子

リソースにグローバルな識別子 (URI) を与える。

**優先度:A**

組織内のデータベースにおける目録(メタデータ)であっても、一般にレコードの ID(主キー)フィールドを持つ。メタデータを公開して共有可能とするためには、この ID が他組織の ID と重複することなく、また WWW のようなグローバルな環境においてもどのレコードを示すのか確実に認識できるようにする必要がある。

メタデータ提供者は:

1. グローバルな識別子として、URI (Uniform Resource Identifier) を用いる
2. メタデータ・レコードは、この URI を主語として記述する

組織内データベースの ID フィールドは、必ずしも URI とする必要はなく、メタデータの名前空間 URI を定め、ID と名前空間 URI との組み合わせで URI に一意に変換して公開できればよい。

メタデータ利用者は:

1. 複数の情報源から収集したメタデータのレコードを、URI を用いて識別、集約する

**(b) 標準ラベルの付与**

(コンピュータ処理可能な識別子に加えて) 人間に理解可能なラベルを標準的な方法で与える。

**優先度:A**

メタデータは、コンピュータによる処理、連動、再利用に加え、画面表示のためにも用いられる。このとき、コンテンツが何であるかをすばやく確実に把握できるような情報が必要である。

メタデータ提供者は:

1. 著作物のタイトルなど、コンテンツ作者が付与した名前(オーセンティック名)があればその名前を与える
2. オーセンティック名がない場合でもコンテンツ提供者が付与した名前を、代表ラベルとしてメタデータ・レコードに加える。
3. 関連ラベル(サブタイトルなど)は、原則として、区切り文字で連結するか構造化して、本タイトルと一体化して扱う。代替ラベル(別称など)は、本タイトルとは別の項目として記述する

名前は可能ならばユニークな(一組のメタデータ内で重複しない)ものを与える。複数コンテンツのオーセンティック名が重複する場合は、補助的なプロパティとの組み合わせで識別できるようにすることが望ましい。

**メタデータ利用者は：**

1. 技術詳細に示す推奨記述法に準じてラベルのプロパティを処理する。

**(c) 作者の標準記述**

標準的で再利用可能な形で、コンテンツの作者を記述する。

**優先度：B**

作者の記述にはさまざまな方法が用いられ、それぞれの合理性があるが、異なるメタデータ間での相互運用性を高めるために、できるだけ推奨記述方法を採用する。

**メタデータ提供者は：**

1. メタデータ利用者が作者の姓名を区別できるようにする
2. 共著・共作、連名の作者は、原則として全員の名前を、順序を保持できるように記述する
3. 作者を名前のみで示す方法(リテラル記述)と、構造化して連絡先などの属性も含めて示す方法(実体記述)を適切に使い分ける

(リテラルと実体記述の使い分け、および役割の記述については技術詳細を参照)

**メタデータ利用者は：**

1. 提供者によって作者の記述法が異なる場合があることを踏まえ、技術詳細の推奨事項に準じて作者情報を適切に処理する

**姓名の区切り** 作者の名前は、姓名を区別できるように記述する。

特に日本語の場合、姓名が直接つながれた名前から姓と名を分離することは難しいので、可能な限り姓と名を区切った形で名前を記述する。区切り文字には、スペース文字もしくはカンマを用いる。外国名のカタカナ表記については中点「・」を用いる。

- スペース文字で区切る場合、言語によって姓名の順が逆になることがあるので、言語タグを用いるなど、記述規則のコメントで解釈方法を明示する
  - 西欧系外国名の原綴表記、日本人名のローマ字表記は、「名姓」の順序でスペース区切りするか、「姓、名」とカンマを用いて姓を前に出すかたちで、姓名を識別できるようにする
- 同姓同名を区別するために所属、生没年などの補助情報を加える場合、補助情報は括弧に入れるなどして姓名と機械的に区別できるようにし、記述規則のコメントで解釈方法を明示する

※人名の記述に関する詳細な規則提示はこのガイドラインの範囲を超えるので、国立国会図書館の「個人名・団体名標目の選択・形式基準」<sup>3</sup>などを参照。

**共著、連名の記述** 共著、連名の場合のリテラル著者ラベルは、ひとつの著者プロパティの値として、資料に記載された順序を尊重して、一貫した区切り文字で著者を区切って記す。区切り文字の約束は、記述規則で明記する。

実体型の著者記述が必要な場合は、リテラル型著者記述と併記する。

#### (d) 日時・位置情報

曖昧さのない標準形式で日時、位置情報を付与する。

#### 優先度:B

メタデータの記述において、リソースを時空間に位置づける日時情報、位置情報はしばしば重要な役割を果たす。これらのプロパティおよび値が標準化されていれば、たとえば異なる情報源からのデータを組み合わせてカレンダーや地図上に表示するといったサービスで有効に利用することができる。

#### メタデータ提供者は:

1. 日時記述には曖昧さの無い標準的なフォーマットを用いる
2. 位置情報は、緯度経度であれば世界測地系の百分率表記を、住所の構造化には標準と互換性のある記述法を用いる

#### メタデータ利用者は:

1. ガイドラインに示されている日時・位置情報のさまざまなフォーマットについて、対応できるような処理手順を組み込むか、処理できない場合を想定した対応手順を組み込む。
2. プロパティ値の精度、粒度にさまざまな程度があることを踏まえてメタデータを処理する。

**日時のフォーマット** 日時プロパティの値は、原則として 2011-02-10T12:30:00 の形(XMLスキーマの日時形式)もしくは 2011-02-10 の形(XMLスキーマの日付形式)とし、可能ならば項目規則の値制約としてどちらかの形かを明示する。

個別データベースにおける日時プロパティ値はこれらに限定する必要はないが、公開する場合は上記のいずれかに変換することを原則とする。特に、スラッシュ区切りの日付("02/10/11"の形)は、地域によって意味が異なる<sup>4</sup>ので、公開メタデータの日時プロパティ値での使用は避ける。

<sup>3</sup> [http://www.ndl.go.jp/jp/library/data/personalname\\_standard.html](http://www.ndl.go.jp/jp/library/data/personalname_standard.html)

<sup>4</sup> "02/10/11" は、日本では2002年10月11日、米では2011年2月10日、欧州では2011年10月2日

**位置情報の値** 位置情報を緯度経度で与える場合、世界測地系(WGS84)の百分率値で記述する。住所を構造化して記述する場合は、VCARD 語彙に対応付けできる形で記述する。

#### (e) キーワード

可能ならばキーワードを統制語彙で付与する。

#### 優先度:B

リソースの発見を容易にするために、メタデータにはしばしばキーワード(タグ、主題件名、分類など)が含まれるが、多くの場合これらのキーワードは任意の単語が用いられ、検索漏れやノイズを生じてしまう。キーワードを広く共有可能な形での記述が求められる。

#### メタデータ提供者は:

1. 可能ならば統制語彙からキーワードを選ぶ
2. どの語彙から選んだキーワードであるかを(できれば URI で)示す

#### メタデータ利用者は:

1. キーワードを単純文字列として処理するだけでなく、(URI による識別を含む)統制語を有効に利用するための手順を用意する。

#### (f) 読みについて

ラベルに読みを与える場合は、言語タグを用いて区別するか、ラベルを構造化して記述する。

#### 優先度:C

日本語では多くの場合に読みが必須要件となるが、欧米語圏ではラベルを単純にリテラルとして捉えないため、読みに関する国際的な標準は確立されていない。このガイドラインでは、読みの与え方として 2 通りの方法を示す。

#### メタデータ提供者は:

1. 1 つのリソースについて 1 つしかない項目(たとえば本名)の場合、ラベルと読みのために同じプロパティを 2 回使い、読みを言語タグで区別して記述してよい
2. 1 つのリソースで複数の値を持つ可能性がある項目(たとえば別名)の場合や、出現回数制約の検証を行ないたい場合は、ラベルを構造化して正規形と読みを記述する

と解釈される。

**メタデータ利用者は：**

1. ラベルの値が構造化されて読みを持っていたり、同じプロパティが反復されて言語タグで区別される可能性があることを考慮する。
2. 読みが提供される場合、読みを利用した並べ替えや検索ができるようにする。

**(g) リテラル値のデータ型と言語タグ**

リテラル値のデータ型、言語タグは、目的が明確な場合に限って用い、スキーマで使用を宣言して一貫した形で与える。

**優先度:C**

プログラムでメタデータのリテラル値を扱う際に、(1)その値がテキスト文字列なのか、数値なのか、年なのかといった「型」が分かるほうが処理がスムーズな場合がある。また(2)、音声読み上げや自動翻訳のためなど、何語(日本語、英語、仏語など)の文字列なのかが分かるほうが都合がよい場合もある。

メタデータのリテラル値には、(1)のために「データ型」を、(2)のために「言語タグ」を持たせることができる。一方、データ型、言語タグを持つものと持たないものは、リテラル値が同じであっても、アプリケーションは異なる値として処理するので、使用には注意を要する。

**メタデータ提供者は：**

1. データ型は、数値、日付など、アプリケーションで単純文字列とは異なる処理を行なうための情報が必要である場合に付与する
2. 言語タグは、複数言語によるデータが混在するケースなどの明示的区別が必要である場合に、一貫して付与する

**メタデータ利用者は：**

1. リテラル値がデータ型や言語タグを持つ場合があることを理解する
2. リテラル値の比較、特に異なる情報源からのリテラル値を比較・検索する場合は、値をプレーンテキストにキャストした上で行なう

**7.3.3 メタデータの公開と交換・利用に関する指針**

作成したメタデータを公開したり、公開されているメタデータを利用したり組み合わせたりする際に、相互運用性を高めるための指針を示す。

**RDF によるメタデータの公開**

メタデータの公開には、標準的なデータ表現方法として RDF を用いる。

**優先度:A**

メタデータの交換、共有は、関係する組織やサービスが多くなると複雑さを増す。さまざまな組織、サービス間で柔軟にメタデータをやり取りするために、公開データの記述に RDF(「4.2.データモデルの標準化:RDF」参照)を用いる。

**メタデータ提供者は:**

1. メタデータの公開、共有、交換に RDF を用いる
2. 内部的な作成、利用、維持管理には目的に応じたシステムと表現方法を用い、公開示には RDF に変換する

**メタデータ利用者は:**

1. RDF を利用して異なる情報源のメタデータを集約する
2. RDF ではないデータを RDF に変換するゲートウェイサービスを利用する

異なる組織・サービス間でのデータ交換のためには、それぞれのデータ項目や構造のマッピングが必要になる。この交換が複数組織・サービス間になると、必要なマッピングは N 対 N に幾何級数的に増加する。ここで、データ交換の中間項となるフォーマットがあれば、変換マッピングは各組織と中間項の間だけ(N 対 1)でよい。

RDF は、

- 多様なデータを柔軟に表現できる
- 対象の識別、データ値に URI を用いるため、複数の情報源からのデータを集約したり関連付けられる
- プロパティ(項目名)を標準語彙に関連付けて定義することで、異なるプロパティを同一の標準語彙に集約して横断検索などを可能にする

といった特徴を持つ。これを利用し、RDF を各組織・サービスのデータ交換の中間項と位置づけることができる(「4.2.4.RDF パス」参照)

**(b) スキーマを利用したメタデータ共有・活用**

メタデータを正しく理解・利用するためにスキーマを参照し、必要に応じてプロパティの整合調整を行なう。

**優先度:B**

多様なメタデータを組み合わせて利用する場合、領域特有の関係を表すデータが含まれる可

能性がある。このとき、語彙定義もしくは記述規則において定義されている上位プロパティが既知の語彙のものであれば、およその意味を把握して利用することができる。

メタデータ提供者は:

1. メタデータがどのスキーマに準拠しているかの情報を加えて提供する
2. スキーマには、プロパティ単純化(ダム・ダウン)のための情報を用意する

メタデータ利用者は:

1. 既知でない語彙のタームが含まれる場合は、語彙定義を参照し、記述されているデータの意味を確認する
2. データを組み合わせるためにプロパティを揃える(整合調整)必要があれば、上位プロパティ関係を利用して単純化(ダム・ダウン)を行なう

**(c) 変換の際のデータ粒度とダム・ダウン**

データを公開用などに変換する場合は、情報が失われないように構造と粒度を保ち、利用者がダム・ダウンする。主要プロパティはあらかじめ単純化値を提供する。

**優先度:B**

個々のデータベースで詳細に記述されているメタデータを公開用 RDF に変換する際に、汎用標準プロパティに合わせるために、複数のメタデータ項目が結合されてしまうケースがある。いったん結合され単純化されてしまったメタデータを再度詳細化することは非常に難しいので、情報が欠落してしまい、再利用に支障をきたす場合がある。

公開のための変換時に、無理な連結や単純化をせず、利用者が変換できるようにする。また、基本的な属性については単純化した値をオリジナルと併せて提供する。

メタデータ提供者は:

1. データを公開用に変換する際には、元データの情報が失われないように、構造と粒度を保っておく
2. 基本的なプロパティについて、ダム・ダウンに相当するデータをあらかじめ加えておく。

元データの構造が冗長である場合は、情報が失われない範囲で、公開時に単純化してもよい。スキーマで十分な情報が提供されていれば、アプリケーションはそれを用いてダム・ダウンを行なうことができる。しかし、すべてのアプリケーションがスキーマを参照するとは限らず、またデータ構造によってはダム・ダウンにはコストがかかる場合もある。

**タイトル(標準ラベル)、作者、日時情報**に関しては、あらかじめシンプル DC 相当 (rdfs:label など、

7.3.2 での推奨プロパティを含む)に変換したプロパティも併せて提供することを推奨する。

メタデータ利用者は:

1. メタデータの集約のために、まず標準プロパティによる単純値が提供されているかどうかを確認する
2. 構造化されたデータを、情報が欠落しないように適切に保存した上で、必要な情報を単純化して利用する

### 7.3.4 運用に関する指針

公開するメタデータやスキーマを長期にわたって利用可能にするための、運用に関する指針を示す。

#### (a) スキーマのバージョン管理

スキーマの管理データを明示し、バージョン管理を行なう。

**優先度:A**

スキーマは一度作成したら不変とは限らず、必要に応じて語彙を追加したり、定義の細部を修正する場合がある。スキーマの更新内容によっては、古いバージョンのスキーマに準拠して書かれたメタデータが、新バージョンでは正しく解釈できなくなる可能性もある。メタデータを長期にわたって利用可能にするため、適切な情報の提供が必要である。

メタデータ提供者は:

1. スキーマには、作成日、作成者、バージョンなどのメタデータ(管理情報)を加える
2. 旧バージョンのスキーマも参照できる形で保存する

メタデータ利用者は:

1. データセットを最初に利用する時に、データの準拠スキーマとそのバージョンを確認する

#### (b) メタデータ自身の管理情報の付与

メタデータには作者、作成日時、準拠スキーマなどの管理データを付与する。

**優先度:A**

メタデータを適切に利用し、長期の相互運用性を確保するためには、メタデータ自身がいつ誰によって作成されたかなどの「メタ・メタデータ」も重要になる。

**メタデータ提供者は:**

1. メタデータのラベル(タイトル)、作者、作成日時、準拠スキーマなどの管理データを付与する
2. この管理データは、原則としてメタデータを提供するファイル単位で付与する

**メタデータ利用者は:**

1. メタデータ取得時に管理情報でデータの鮮度、信頼度などを確認する
2. さらに、次項の由来情報のために、メタ・メタデータも保存しておく

**(c) 由来情報の保持**

データを集約して格納する場合、由来情報とあわせて管理する。

**優先度:B**

複数の情報源のデータは、いったん集約してしまうと情報源を区別できなくなってしまう。また同じ情報源であっても、収集のタイミングによってデータが異なる可能性があり、どの時点での情報であるかが重要になることがある。

**メタデータ利用者は:**

1. データを集約、保管する場合は、情報源ごとにデータセットを管理できるようにする
2. データセットごとに、取得元、収集日時、(b)のメタ・メタデータなどの情報を記録する

**(d) レジストリへの登録**

スキーマを公開レジストリに登録し、利用者の発見を助けるとともに、最新版、旧版を確認できるようにする。

**優先度:B**

メタデータを設計・作成するために、またメタデータを利用する際にその規則を確認するために、スキーマを検索したい場合がある。またあるスキーマについて、過去にさかのぼってその定義を確認したい場合もある。こうした検索・発見を容易にするために、公開レジストリを活用する。

**メタデータ提供者は:**

1. 公開するスキーマは、可能な限り公開レジストリに登録する
2. レジストリには、最新版に加え、旧版のスキーマも確認できるように登録する

メタデータを新たに設計・作成する場合に、既存スキーマを参考にできれば開発期間が短縮でき相互運用性も高まるが、適当なスキーマを見つけるのは難しいことが多い。また、メタデータを利用する際に、準拠スキーマが明示されていなくても、ジャンルや使用プロパティからスキーマを検索できれば、規則を確認できる可能性がある。

メタデータ利用者は:

1. スキーマ情報の確認にレジストリを利用する

**(e) メタデータの検証**

メタデータを作成・公開する場合、スキーマの記述規則と矛盾がないか検証する。

**優先度:C**

メタデータを利用する場合、そのデータがスキーマに正しく準拠していないと、アプリケーションでの処理がうまくいかなくなってしまう。

メタデータ提供者は:

1. メタデータを公開する際は、準拠スキーマを明示するとともに、そのスキーマに対して妥当であるかの検証を事前に行なっておく

## 8. 専門家による調査・検討

### 8.1 検討会での報告と議論

2010 年 11 月から 2011 年 3 月までに計 5 回の検討会を開催し、プロジェクトの進捗報告を行なうとともに、専門家を交えての意見交換を行なった。専門家から出された主要な論点・意見を整理して報告する。

#### 8.1.1 メタデータの設計とスキーマの現状について

(a) **プロジェクトの問題認識** メタデータのレジストリを作成していく場合には、形式的に RDF で作られている物を集めてデータベース化するのは困難ではない。いろいろなところでアンフォーマルなエクセルの表のようなもので管理されているものも多い。運用でもたとえば日付等でも形式が異なっている。こういった部分も含めできるだけ多くの人が相互運用できることが必要である。

出発点は出席機関がお持ちのデータ(公開できる範囲)を参考にし、どういう記述がなされており、どういう形にすれば無理なくレジストリに登録可能かという方法論から詰めていきたい。

(b) **メタデータの設計の考え方** メタデータの構造化につき、詳しくすればするほど深くなってしまい実用的でなくなってしまう。表現の美しさを追求すると、流通で難しい面が出てくる。議論のみ深くなってしまう可能性もあるため、目標地点「旗印」が必要ではないか。

LODAC プロジェクトでは(16.2.5(b)LODAC Musenm 参照)、メタデータデザインはできるだけシンプルにしている。元データに沿って厳密にデザインすると、非常に複雑なデータとなり使いにくくなってしまふ。

メタデータをシンプルにデザインしようとする、エレメントを多数設けなければならなくなる、という問題に対しては、状況に応じて、提供するデータを変えるようにしている。

#### 8.1.2 RDF とウェブ上でのメタデータについて

(a) **Linked Data** 現在提供されている Linked Data の質は良くなく、ゴミがほとんどだと言われるが、とにかく RDF として出すことが大事である。

セマンティック・ウェブと Linked Data は新しい言語を設計することである。自然言語をウェブの単純なプロトコルの上に載せる際にどうするのか、という問題がある。ウェブの言語はカタログの言語とは違う。

(b) **書誌の RDF 化** 典拠データを RDF 化することの意義は理解できるが、書誌データを RDF 化することの意義や、実践例はあるか、という問いに対して、「スウェーデン国立図書館が、書誌データを RDF 形式で提供している。全世界の目録がリンクし合うことができるという意義がある」という考えが示された。

### 8.1.3 読みに関する問題とダブリンコアの定義

(a) **dcterms:title の値域** dcterms:title の値域が rdfs:Literal に制約されてしまったが、日本語には読みがあり、さまざまな title は読みとセットで表す必要があるため、データをリテラルではなく入れ子の構造にせざるを得ないので、困っているとの意見が出された。

**DCMI の見解** 第 2 回検討会に招聘した DCMI のトム・ベイカー氏から、考え方が説明された。

標準化する側としては、各分野の声に耳を傾け過ぎていたら決められない。日本で NDL と NII とが協力して読みの表現について実績をつくってしまい、日本におけるデファクトスタンダードの形にしてしまえば問題ない。その方が日本のユーザにとっても便利だろう。語彙の値域をリテラルにするか、リソースにするか、それとも指定しないかは悩ましい問題。

RDF ではリテラルかリソースかを厳密に分けておくことが重要。値域ごとに語彙を分けるという方法もある。構造化する場合は、SKOS の拡張ラベルの事例が参考になるのでは。(skosxl:prefLabel、skosxl:altLabel の値域はリソースが指定されているが、リテラルを持ってないかという議論があったらしい)

### 8.1.4 ガイドラインについて

(a) **記述規則、ガイドラインの必要性** データベースを開発していると、様々なメタデータ・スキーマの事例を参考に作成しようとして、見えそうな項目だと思ってしまうものが、実際は使いにくい状況になることもある。どのように書くのかがわからないからだ。

記述規則を定めるのか、ベストプラクティスガイドとするか議論の余地はあると思うが、そういったものが欲しい。

(b) **対象読者と読みやすさ、関連資料について** メタデータの専門家が活用するのであればいいが、実際は少し離れた人間が活用する事が多くなるとされる。その場合、領域の知識がまったくわからない状況になることもある。

実際に苦勞する事があるので、関連仕様の翻訳も行なって欲しい。全文翻訳でも構わないし、企業が翻訳しているものを集めてくるのも構わないので検討して欲しい。

電子出版に関しても意識している事業であり、4000 の出版社を含めた関係者にも理解されるものでなければならない。

ガイドラインの扱うボリュームが膨大なものになりそうなので、読みやすくするために、ガイドラインのメタデータ(目次を説明的にするなど)を検討する必要がある。

## 8.2 研究会での報告と議論

メタデータに関する大きなプレーヤーは国立国会図書館や国立公文書館等のいわゆる MLA の機関となる。他方、電子出版や電子流通に関しては民間企業の影響が大きい。研究会においては、これら関係者の知見についての意見交換が行なわれた。

### 8.2.1 問題提起

(a) **読みについて** メタデータにおける日本語の「読み」取り扱いについて、国際的に標準がない。そもそも国際会議において、読みとは何かということを説明して理解してもらうことから困難である。このプロジェクトのような場で標準を提起してもらいたい。できれば、英文の説明を用意してもらい、国際標準として提案したい。

(b) **スキーマ情報について** スキーマに関する情報を集めようとしても、大変。(Google で検索したら大変な事になる)。専門家でない人間が、目録規則を読む事は非常に困難。ましてや、英語を読まなくてはいけない状況であればなお困難となる。ベストプラクティスガイドのようなものが最低限必要なのかもしれない。

各省庁で文化政策を議論している事もあるが、情報の専門家や資料整理の専門家が参加していない事もある。そうした潜在化していないユーザをフォローしていく必要があるのではないか。

(c) **メタデータ情報共有システムの課題について** 実際に使う人は誰で、どの様な使い方をするのか、ほんとうに使えるのか、を考える必要がある。図書館、博物館、公文書館の方や、そのシステムを請け負うベンダーの方や関連事業者等の利用は想定しているが、一般の方の利用は今のところ考えていない。

誰もが使うわけではないので、実際に、スペックを書く人、サービスを使っていく人が使えるかが重要。この仕組みを使って相互運用性を高めていくには、2～3 年かかかって実現させるものと思われる。

### 8.2.2 研究会およびレジストリの継続的運用について

レジストリを格納するデータをどのように増やしていくのか、メンテナンスしていくのかを考えると、筑波大学地的コミュニティ基盤研究センターが単体でやるのではなく、大きな組織を形成して運用を行えないかと考えている。

「こういったコミュニティは海外にはある。日本でもできないか、私的研究会の方が集まりやすいのか、最終的に『公』に提案する必要が求められる機会があるだろう」といった意見に対し、「筑波大学知的セ単体で支えるのではなく、複数の機関で支えあう組織を作るために、最初に意見をまとめていきたい」という考えが示された。

(a) **継続的運用** 今回のシステムや蓄積されたデータを維持、管理、成長させたいが、年度が変わると状況も変化していく。今は国立大学をベースとしたノンプロフィット組織を考えているが、長期的なことを考えると基盤の設置には不安が残る。

運営費用をどこから、どのように確保するかによって、どういう組織にするか違ってくると思う。博物館分野を考えると、文化庁ともなんらかの関わりがあるべきではないか。

### 第3部 新しいサービスの創出に向けた開発と実証

#### 9. メタデータの共有とスキーマ・レジストリ

インターネット上で、われわれは、いろいろな資料を探し、アクセスし、評価し、利用し、そして場合によっては取引をする。この過程のさまざまな場面で、用途に応じたメタデータが用いられる。1990年代半ばからのインターネットの利用の広がりとともに、メタデータの重要性が認識されるようになってきた。

メタデータは「データに関するデータ(data about data)」あるいは「データに関する構造化されたデータ(data about structured data)」と定義される。すなわち、「なんらかの対象に関する記述」はすべてメタデータである。この定義はシンプルであるが故に、様々なものがメタデータの範疇に含まれることになる。目録や索引、抄録は典型的なメタデータである。また、対象を識別するための名前、映画やテレビ番組の rating、書評、コンテンツの利用条件に関する記述などもメタデータである。また、図書、論文、絵画、写真、地図、ビデオ、ソフトウェアやデータベースといった物理的あるいは具体的な資料に限らず、資料の集まり(コレクション)、人間や組織、サービス、事象(イベント)といったものもメタデータの記述対象となり得る。

メタデータを記述する目的に関しても、記述対象となる情報資源が何者であるのかを示す属性の記述、情報資源の発見・検索、情報資源の維持管理あるいは保存、情報資源の提供や取引など様々である。さらにインターネット上では、様々な目的を持った複数のコミュニティの視点から作られるメタデータを組み合わせて利用すること、メタデータの相互利用性を高めることが求められるため、それに適したデータモデルと記述形式が必要とされている。メタデータの記述項目を定義したメタデータ・スキーマに関する情報の共有と流通を支援するために必要な課題と、そのための基盤となるメタデータ・レジストリについて述べる。

#### 9.1 メタデータ共有の課題

これまで、様々なコミュニティ(例えば、図書館、博物館、美術館、公文書館、学会など)が、それぞれの目的の元にメタデータ・スキーマを開発している。以下はメタデータ・スキーマの例である。

- Dublin Core は、ネットワーク上の情報資源発見のためのメタデータ・スキーマとして開発されたもので、分野にまたがるメタデータの相互運用性を実現することを目的とした語彙を決めている<sup>1</sup>
- IEEE LOM(Learning Object Metadata)は教育学習資源の記述を目的としており、教育・学習の視点からの記述要素を含んでいる<sup>2</sup>
- EAD(Encoded Archival Description)の経験を元に開発された METS(Metadata Encoding and Transmission Standard)は、デジタル仕様の記述、保存、管理を目的として

<sup>1</sup> Dublin Core Metadata Terms, <http://purl.org/dc/terms/>

<sup>2</sup> Learning Object Metadata (LOM), <http://ltsc.ieee.org/wg12/>

いる<sup>3</sup>

- MODS(Metadata Object Description Schema)は、MARC を基礎として XML による書誌データ記述を目的としている<sup>4</sup>
- OAIS(Open Archival Information System)の参照モデルを基礎として提案されたデジタルコンテンツ保存のためのメタデータ参照モデルを与えている<sup>5</sup>

近年のセマンティック・ウェブ<sup>6</sup>の普及により、メタデータ・スキーマの種類は膨大な数になっている。2010 年にインディアナ大学が行ったメタデータ・スキーマに関する調査<sup>7</sup>では、図書館や博物館などにおいて利用される 105 種類のメタデータ・スキーマについて、それぞれの特徴や関係性を 4 つの軸(対象資料のタイプ、使用する機関、役割、メタデータのタイプ)の視点から分類し可視化を行っている(図 9.1)。また、メタデータ・スキーマに関する情報を蓄積した Schemapedia<sup>8</sup>には、2011 年 2 月現在で 277 種類のメタデータ・スキーマが登録されている(図 9.2)。このように様々な目的を持ったコミュニティが作成したメタデータ・スキーマは増加の一途をたどり、現在も増え続けている。

多種多様なメタデータが利用されている状況の中でメタデータを共有し情報資源の発見や相互利用性を高めるためには、以下に示した原則に従ったメタデータ記述が求められる。

- メタデータ記述の語彙やフォーマットに、広く標準として普及しているものを用いる、もしくは新たに定義したものを標準的手法で公開する
- 日本語の読み、日付、地理情報など、メタデータの記述項目の値(プロパティ値)は標準方法に則って記述する
- 将来に渡って長期的にメタデータを解釈し情報資源を利用できるようにするため、バージョン管理を含むメタデータ維持管理の情報を加える

---

<sup>3</sup> Metadata Encoding and Transmission Standard, <http://www.loc.gov/standards/mets/>

<sup>4</sup> Metadata Object Description Schema, <http://www.loc.gov/standards/mods/>

<sup>5</sup> <http://public.ccsds.org/publications/archive/650x0b1.pdf>

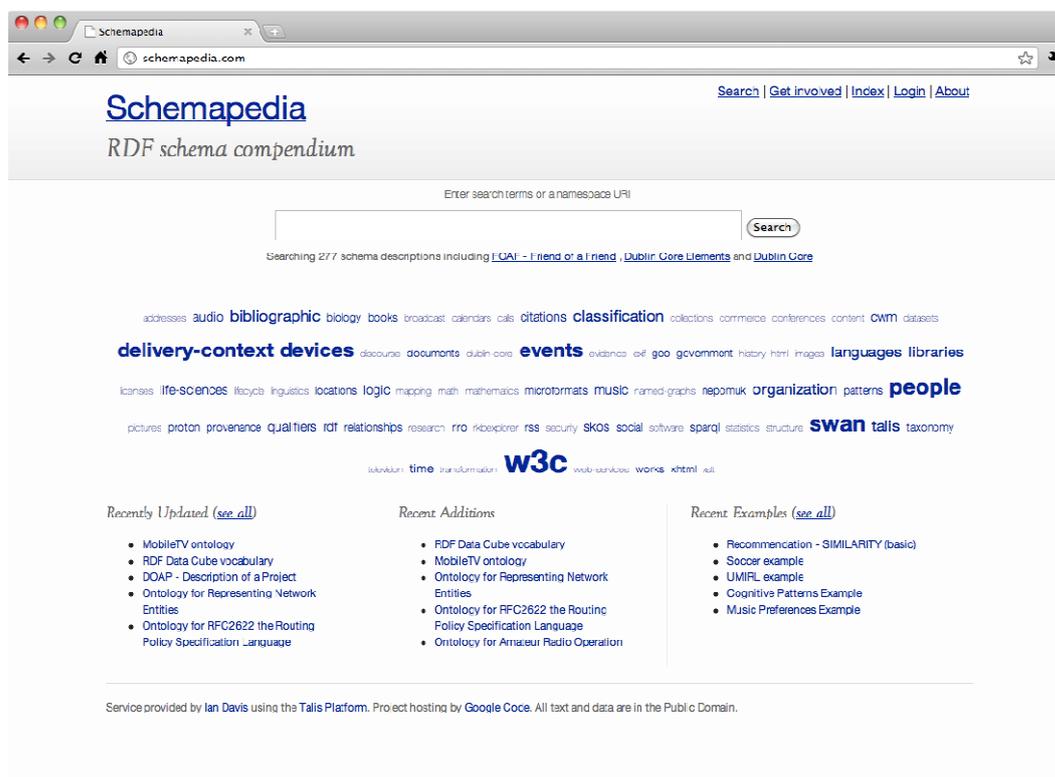
<sup>6</sup> W3C Semantic Web Activity, <http://www.w3.org/2001/sw/>

<sup>7</sup> Seeing Standards: A Visualization of the Metadata Universe, <http://www.dlib.indiana.edu/~jenlrile/metadatamap/>

<sup>8</sup> Schemapedia, <http://schemapedia.com/>



◆ 図 9.1: Seeing Standards: A Visualization of the Metadata Universe (インディアナ大学によるメタデータスキーマの分類と可視化)

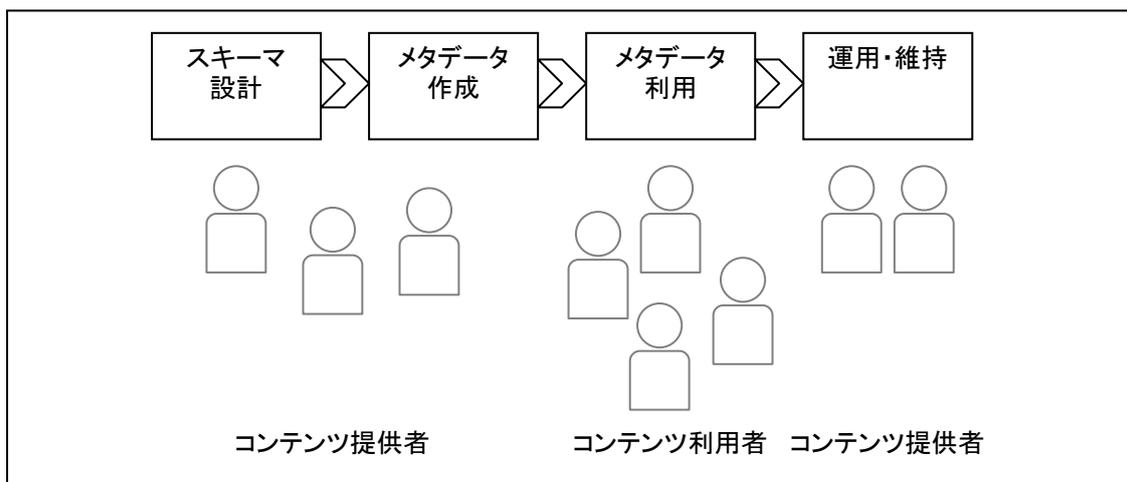


◆ 図 9.2: Schemapedia: RDF schema compendium

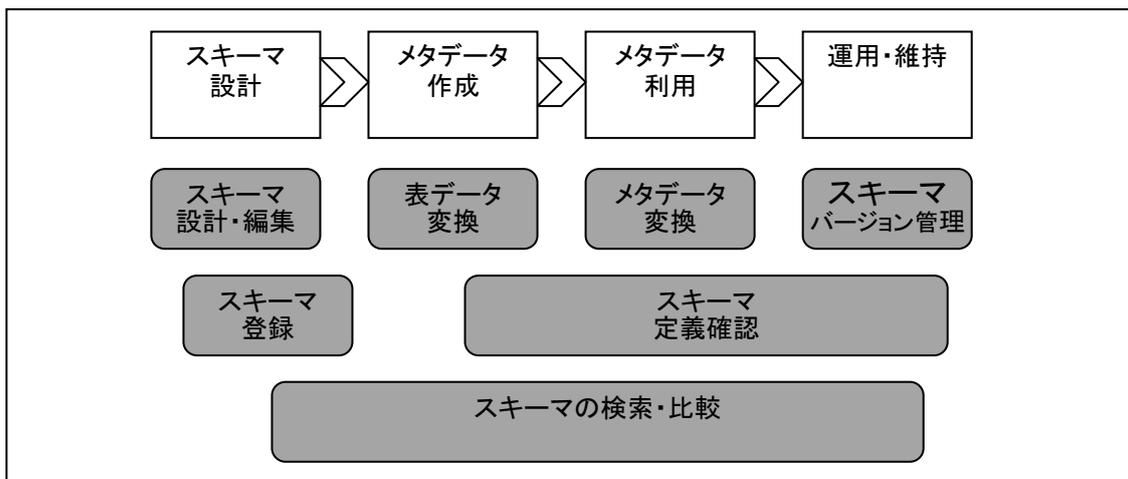
## 9.2 レジストリの役割

ネットワーク上のコンテンツの利用には、出版から配信、閲覧の過程において多様な形式のメタデータが利用されている。メタデータ・スキーマ・レジストリは、メタデータの相互運用性や流通性の高度化を目的として、メタデータ・スキーマに関する情報の蓄積、管理、公開を支援するためのシステムである。メタデータ・スキーマ・レジストリをハブとして多様な情報資源やサービスに係わるメタデータ・スキーマの参照を容易にし、メタデータ・スキーマの相互参照や再利用が進むことでメタデータの相互運用性や流通性の向上が期待される。

図 9.3 はメタデータのライフサイクルを示している。メタデータは、まずコンテンツ提供者によって、目的に合わせたメタデータ語彙の設計と選択、そして記述規則の設計が行われる。そしてコンテンツ提供者によって記述規則に従ったメタデータが作成される。作成されたメタデータは、コンテンツ利用者によって共有・活用され、情報資源の発見、組織化、新しいサービスの創出に利用される。一旦作成されたメタデータ・スキーマは必要に応じてコンテンツ提供者が語彙の追加、定義の修正、バージョン管理等を行い、運用・維持される。図 9.4 はメタデータのライフサイクルに対応したメタデータ・スキーマ・レジストリの機能を示している。以下本節では、コンテンツ提供者とコンテンツ利用者それぞれの視点から見たメタデータ・スキーマ・レジストリの役割と機能について述べる。



◆ 図 9.3:メタデータのライフサイクル



◆ 図 9.4:メタデータライフサイクルに対応したレジストリの機能

### 9.2.1 コンテンツ提供者＝メタデータ記述者にとってのレジストリ

メタデータのライフサイクルの中で、コンテンツ提供者とメタデータ記述者はメタデータ・スキーマの設計、メタデータの作成、そしてメタデータ・スキーマの運用と維持を行う。7 の「メタデータ情報共有のためのガイドライン」で述べたように、それぞれのフェーズの要件は以下の通りである。

#### スキーマ設計:

- スキーマを相互運用可能な形で選択・設計する
- 異なるメタデータの記述規則を相互に理解・再利用可能とする
- スキーマは機械的な検証にも活用するため、標準的な方法で定義する
- 既存スキーマを利用して新たなメタデータ・スキーマを設計する際には、相互運用できるように元のスキーマとの関連を付与する

#### メタデータ作成:

- メタデータを機械的に利用しやすい標準的な形式で作成する

#### 運用・維持:

- スキーマのバージョン管理を行う
- コンテンツ提供者、メタデータ作成者、メタデータ利用者の理解を支援するために、スキーマの最新版、旧版を確認できるようにする
- 作成したメタデータが規則に従っているか検証する

そこで、メタデータ・スキーマ・レジストリは、コンテンツ提供者が相互利用性と再利用性の高い

メタデータの作成を支援するために、各フェーズに対して以下の機能を提供する。

スキーマ設計:

- メタデータ・スキーマ(メタデータ語彙、記述規則)の設計・編集機能を提供する
- 設計したメタデータ・スキーマを登録する
- 標準的方法でメタデータ・スキーマを定義する
- 既存のスキーマを参考にし、スキーマ作成のコストや手間を削減するために、組織やスキーマの目的に類似したメタデータ・スキーマの発見を支援する
- 相互運用や変換のために、関連する語との対応付けを支援する。特に、ダブリンコアとの対応を付け、単純化を行うためのダム・ダウン用のマッピングを支援する

メタデータ作成:

- メタデータの公開には、標準的なデータ表現方法として RDF が用いられるが、コンテンツ作成の現場では、表形式で編集されることも珍しくない。そこでレジストリは、CSV 形式や TSV 形式で作成されたメタデータを、RDF 形式に変換するメタデータ作成支援機能を提供する

運用・維持:

- 必要に応じてスキーマには語彙を追加や定義の修正が行われることがあるため、変更を記録し、バージョンを管理する機能を提供する
- メタデータを長期に渡って理解して利用するために、最新版だけではなく過去のバージョンのスキーマを参照するための機能を提供する
- 記述したメタデータが、スキーマに従っているかどうかの検証機能を提供する

## 9.2.2 コンテンツ利用者にとってのレジストリ

メタデータのライフサイクルの中で、コンテンツ利用者は公開されているメタデータを活用して情報資源の発見や組織化を行い、関連する情報資源へのリンク付けや新たなサービスを創出する。そのためには、メタデータの記述に用いられている語彙と記述規則を理解して活用するためにスキーマを参照することが求められる。このフェーズにおいてレジストリは以下の機能を提供する。

- スキーマ(語彙定義、記述規則)の参照機能
- スキーマの検索機能: 人間が語彙の役割を理解するために付与されているラベルや定義の説明を文字列で検索する機能に加えて、語と語の関係(上位語、下位語、関連語)を利用して検索する機能
- スキーマを RDF や Topic Maps など機械的に利用できる標準的な形式で出力する機能

- 複数のメタデータを組み合わせて利用するために、上位プロパティとの関連を利用してダム・ダウンを行い、語彙を整合調整する機能

### 9.3 メタデータ・レジストリの事例

本プロジェクトで構築したメタデータ・スキーマ・レジストリ Meta Bridge の他にもメタデータ・スキーマを蓄積し、メタデータ・スキーマに関する情報の発見と流通を目的としたレジストリの構築が行われている。本節では、その事例として、代表的な2つのメタデータ・スキーマ・レジストリ(DCMI Metadata Registry と Open Metadata Registry)を紹介する。また、それぞれのレジストリと本事業のレジストリとの違いについて述べる。

#### 9.3.1 DCMI Metadata Registry

DCMI(Dublin Core Metadata Initiative)<sup>9</sup>とは、ダブリンコアに関連する標準化と維持管理を行う団体である。DCMI Metadata Registry は DCMI が策定したメタデータ語彙に関する情報を蓄積・提供するための公式なサービスで、DCMI の Working Group のひとつである DC Registry Working Group(現在は DCMI Registry Task Group)が開発したシステムである<sup>10</sup>。以下、DCMI Metadata Registry 開発の経緯、蓄積されているスキーマ、機能、本事業で構築した Meta Bridge との違いに分けて述べる。

##### (1) 開発の経緯

DCMI Metadata Registry (単に DC Registry とも呼ばれる)の開発は、1998 年に遡る。第 7 回ダブリンコアワークショップ(1998 年にフランクフルトで開催)において、現在の DCMI CIO (Chief Information Officer) である Tom Baker 博士を中心にスキーマ・レジストリに関する議論が行われた。その議論の中で、メタデータの流通性を高め、長年に渡ってメタデータを理解し利用するために、メタデータ・スキーマに関する情報を蓄積し提供する基盤(スキーマ・レジストリ)の重要性が認識された。そして、1999 年にオタワで開催された第 8 回ダブリンコアワークショップにおいて、DCMI が標準化を行ったメタデータ語彙に関する情報を蓄積し提供するためのスキーマ・レジストリの開発と議論を行うためのワーキンググループ(DCMI Registry Working Group<sup>11</sup>)が設置された。その後、2000 年にプロトタイプが開発され<sup>12</sup>、2003 年に DCMI Registry が正式に公開された。2004 年には、ソフトウェアからレジストリに蓄積された情報を検索するための SOAP および REST を利用した

<sup>9</sup> Dublin Core Metadata Initiative, <http://dublincore.org/>

<sup>10</sup> DCMI Metadata Registry, <http://purl.org/dcregistry/>

<sup>11</sup> DCMI Registry Task Group,

[http://wiki.metadataregistry.org/DCMI\\_Registry\\_Community#DCMI\\_Registry\\_Community\\_Projects](http://wiki.metadataregistry.org/DCMI_Registry_Community#DCMI_Registry_Community_Projects)

<sup>12</sup> ULIS Open Metadata Registry, <http://avalon.slis.tsukuba.ac.jp/registry/>

API が公開され、2008 年には W3C が定めた RDF を検索するための標準である SPARQL<sup>13</sup>を使ったメタデータ語彙の検索機能が追加された。

DCMI Metadata Registry は 2007 年まで OCLC Research<sup>14</sup>において運用されていたが、現在は移管され、筑波大学知的コミュニティ基盤研究センター<sup>15</sup>によって運用と保守が行われている。

## (2) 蓄積されているメタデータ・スキーマ

DCMI では DCMI Metadata Terms<sup>16</sup>の中で、以下の 4 種類の語彙を定義している。

- プロパティ –メタデータ記述項目の属性(例:title, creator, publisher 等、合計 71 種類)
- クラス –メタデータの記述対象である情報資源の型(例:Agent,MediaType 等、合計 35 種類)
- Vocabulary Encoding Scheme –メタデータに記述する値が利用する統制語彙の型(例:DDC, LCSH, MESH 等、合計 9 種類)
- Syntax Encoding Scheme –メタデータに記述する値の形式(例:ISO3166(国コード), W3CDTF(日付)等、合計 11 種類)
- DCMI Type Vocabulary –情報資源の type を記述する際に利用することができる統制語彙(Event, Image, Sound, Text 等、合計 12 種類)

例えば、情報資源の作成者を記述するために用いる creator プロパティは表 9.1 のように定義されている。語彙の定義は RDF スキーマ<sup>17</sup>を用いて記述されている。また、ボランティアによって一部の語の Label や Definition は 22 言語に翻訳されたものが提供されている。図 9-3 は、creator プロパティの定義を RDF スキーマで記述した例である。

---

<sup>13</sup> SPARQL Query Language for RDF, <http://www.w3.org/TR/rdf-sparql-query/>

<sup>14</sup> OCLC Research, <http://www.oclc.org/research/>

<sup>15</sup> 筑波大学知的コミュニティ基盤研究センター, <http://www.kc.tsukuba.ac.jp/>

<sup>16</sup> Dublin Core Metadata Terms, <http://purl.org/dc/terms/>

<sup>17</sup> RDF Vocabulary Description Language 1.0: RDF Schema, <http://www.w3.org/TR/rdf-schema/>

◆ 表 9.1:DCMI Metadata Terms での Creator の定義

語の名称 : creator	
URI:	http://purl.org/dc/terms/creator
Label:	Creator
Definition:	An entity primarily for making the resource.
Comment:	Examples of a Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.
Type of Term:	プロパティ
Refines:	http://purl.org/dc/elements/1.1/creator
Refines:	http://purl.org/dc/terms/contributor
Has Range:	http://purl.org/dc/terms/Agent
Version:	http://dublincore.org/usage/terms/history/#creatorT-001

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
  <rdf:Description rdf:about="http://purl.org/dc/terms/creator">
    <dcterms:issued>2008-01-14</dcterms:issued>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/terms/contributor"/>
    <rdfs:label xml:lang="en-US">Creator</rdfs:label>
    <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/elements/1.1/creator"/>
    <dcterms:modified>2008-01-14</dcterms:modified>
    <rdfs:comment xml:lang="en-US">An entity primarily responsible for making the resource.
  </rdfs:comment>
    <rdfs:range rdf:resource="http://purl.org/dc/terms/Agent"/>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/dc/terms/" />
    <dcterms:hasVersion rdf:resource="http://dublincore.org/usage/terms/history/#creatorT-001"/>
    <dcterms:description xml:lang="en-US">Examples of a Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.</dcterms:description>
  </rdf:Description>
</rdf:RDF>

```

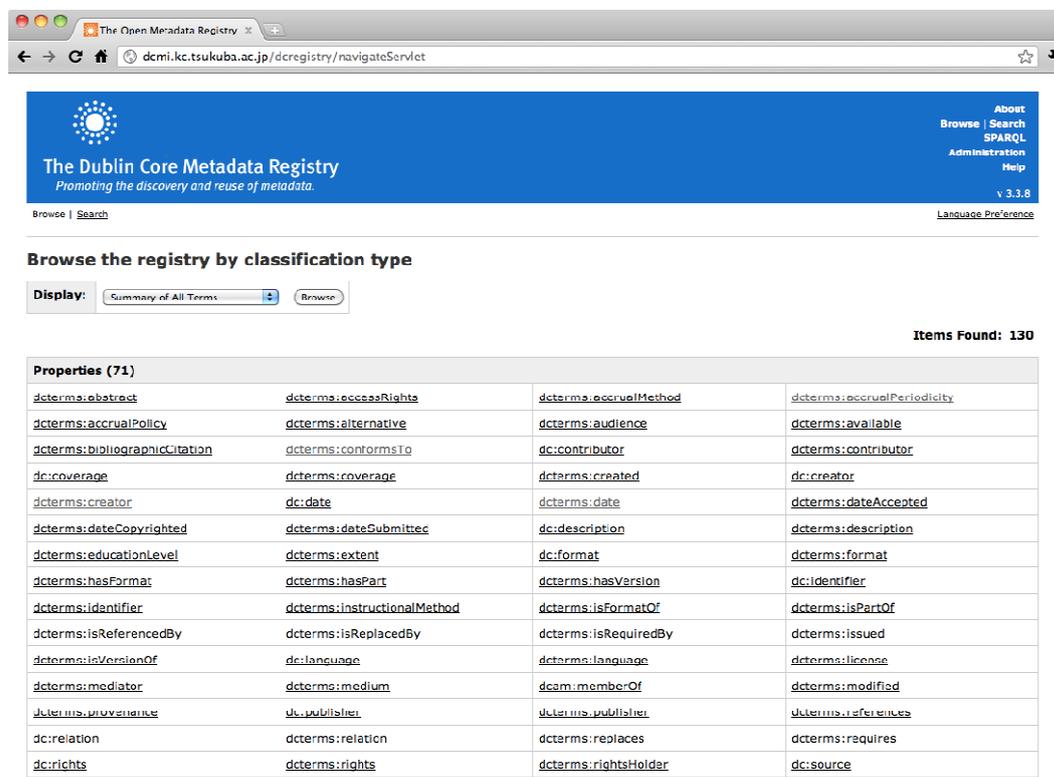
◆ 図 9.5:RDF スキーマでの語の定義例:Creator プロパティ

## (3) 機能

DCMI Metadata Registry は人や計算機が DCMI で定義したメタデータ語彙を検索し、参照できるようにするために以下の機能を備えている。

- ・ 語彙の一覧表示機能

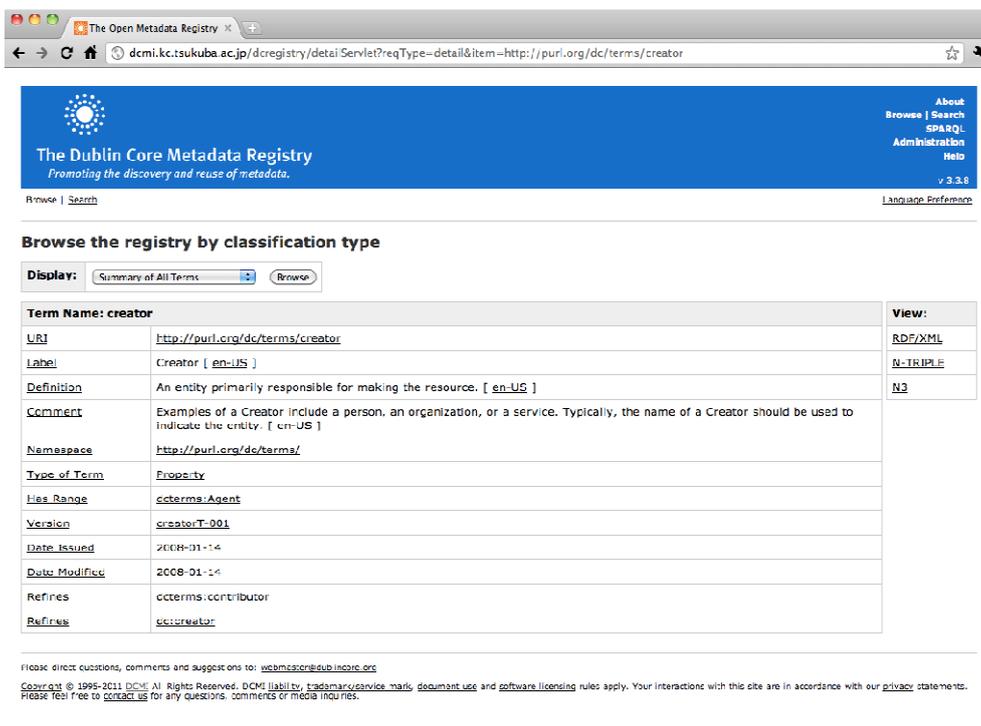
登録されている語彙を、プロパティ、クラス、Vocabulary Encoding Scheme, Syntax Encoding Scheme に分けて一覧表示する(図 9.6)。それぞれの語を選択することで詳細な定義が表示される。



◆ 図 9.6:メタデータ語彙の一覧表示

- ・ 詳細な定義表示機能

それぞれの語の詳細な定義を表示する(図 9.8)。関連するプロパティ(super/sub プロパティ)やクラス(super/sub クラス)があれば、それら語へのリンクを提示する。また、語のラベルや役割の説明が翻訳されている場合は、図 9.7 のように翻訳を並べて表示する機能も持つ。



◆ 図 9.7: 語の詳細な定義の閲覧

Term Name: creator	
URI	<a href="http://purl.org/dc/elements/1.1/creator">http://purl.org/dc/elements/1.1/creator</a>
Label	الكاتب او المبدع الدليل [ ar-SA ]
Label	Autor o Creador [ es-ES ]
Label	Creator [ en-US ]
Label	Kaihanga [ mi-NZ ]
Label	Twórcza [ pl-PL ]
Label	créateur [ fr-FR ]
Label	Tekijä [ fi-FI ]
Label	作成者 [ ja-JP ]
Label	เจ้าของงาน [ th-TH ]

◆ 図 9.8: 複数の言語での表示

- ・ 検索機能

入力されたキーワードにマッチする語を検索する機能を持つ(図 9.9)。しかしながら DCMI Metadata Registry は、ラベルや役割説明の定義を使った部分一致による単純な検索機能しか持っておらず、プロパティやクラスの関連を利用した検索はできない。

#### Search the registry for terms and other metadata

<b>Search for:</b>	<input type="text" value="title"/>
<b>Case sensitive?</b>	<input checked="" type="radio"/> No <input type="radio"/> Yes
<b>Display results that match:</b>	<input checked="" type="radio"/> Exact phrase <input type="radio"/> All terms <input type="radio"/> Any term
<input type="button" value="Submit"/> <input type="button" value="Reset"/>	

<a href="http://purl.org/dc/elements/1.1/title">http://purl.org/dc/elements/1.1/title</a>	
Label	Title

<a href="http://purl.org/dc/terms/alternative">http://purl.org/dc/terms/alternative</a>	
Label	Alternative Title

<a href="http://purl.org/dc/terms/title">http://purl.org/dc/terms/title</a>	
Label	Title

◆ 図 9.9:"title" で検索した結果の表示

- ・ ソフトウェア API

レジストりに登録された語彙の情報を計算機から利用するためのインターフェースを備えている。利用可能な API は、listItemSummary (登録されている語の一覧を取り出す)、listSupportedLanguages (語彙を記述している言語の一覧を取り出す)、listItemDetail (指定した語の詳細定義を取り出す)、getUpdates (指定した日時以降に定義が更新された語の一覧を取り出す)、listElements (シンプル・ダブリンコアで定義しているプロパティの一覧を取り出す)、listElementRefinements (詳細化したプロパティの一覧を取り出す)、listVocabularyTerms (統制語彙の一覧を取り出す)、listEncodingSchemes (エンコーディング方式を表す語の一覧を取り出す)である。なお、全ての結果は RDF/XML 形式で返される。

- ・ 多言語インターフェース

ダブリンコアに関する議論は英語を中心に行われているが、その利用は世界的な広がりを見せている。英語以外のユーザの利便性を考慮して、DCMI Metadata Registry では、ユーザインターフェースの言語を切り替えることができる。図 9.10 は日本語に、図 9.11 は韓国語にユーザインターフェースを切り替えた例である。

Term Name: creator	
URI	<a href="http://purl.org/dc/terms/creator">http://purl.org/dc/terms/creator</a>
ラベル ( label )	Creator [ en-US ]
定義	An entity primarily responsible fo
内容記述	Examples of a Creator include a p indicate the entity. [ en-US ]
定義元(is defined by)	<a href="http://purl.org/dc/terms/">http://purl.org/dc/terms/</a>
RDFタイプ	Property
値域 ( range )	dcterms:Agent
版として持つ ( 関係 )	creatorT-001
発行された ( 日付 )	2008-01-14
修正・更新された ( 日付 )	2008-01-14
Refines	dcterms:contributor
Refines	dc:creator

◆ 図 9.10:ユーザインターフェースを日本語に切り替えた例

Term Name: creator	
URI	<a href="http://purl.org/dc/terms/creator">http://purl.org/dc/terms/creator</a>
레이블	Creator [ en-US ]
정의	An entity primarily responsible fo
설명	Examples of a Creator include a p indicate the entity. [ en-US ]
정의된	<a href="http://purl.org/dc/terms/">http://purl.org/dc/terms/</a>
RDF 타입	Property
범위	dcterms:Agent
버전을 가진	creatorT-001
발행된	2008-01-14
수정된	2008-01-14
Refines	dcterms:contributor
Refines	dc:creator

◆ 図 9.11:ユーザインターフェースを韓国語に切り替えた例

#### (4) Meta Bridge との違い

DCMI Metadata Registry は、DCMI が標準化を行ったメタデータ・スキーマを対象としているが、Meta Bridge では組織を限定せずに様々なコミュニティが作成したメタデータ・スキーマを対象としている。また、DCMI Metadata Registry に蓄積されているのはプロパティやクラスといったメタデータの記述に用いる語彙のみであるが、Meta Bridge では、メタデータ語彙に加えて、メタデータ項目を記述する際に決められる制約等を定義した項目記述規則(アプリケーション・プロファイル)を蓄

積の対象としている。DCMI Metadata Registry は語彙の参照機能のみしか持っておらず、メタデータインスタンスを変換する機能を持っていない。また、DCMI Metadata Registry は、メタデータ語彙のバージョン管理機能を持たない。

◆ 9.2:DCMI Metadata Registry と Meta Bridge の機能比較

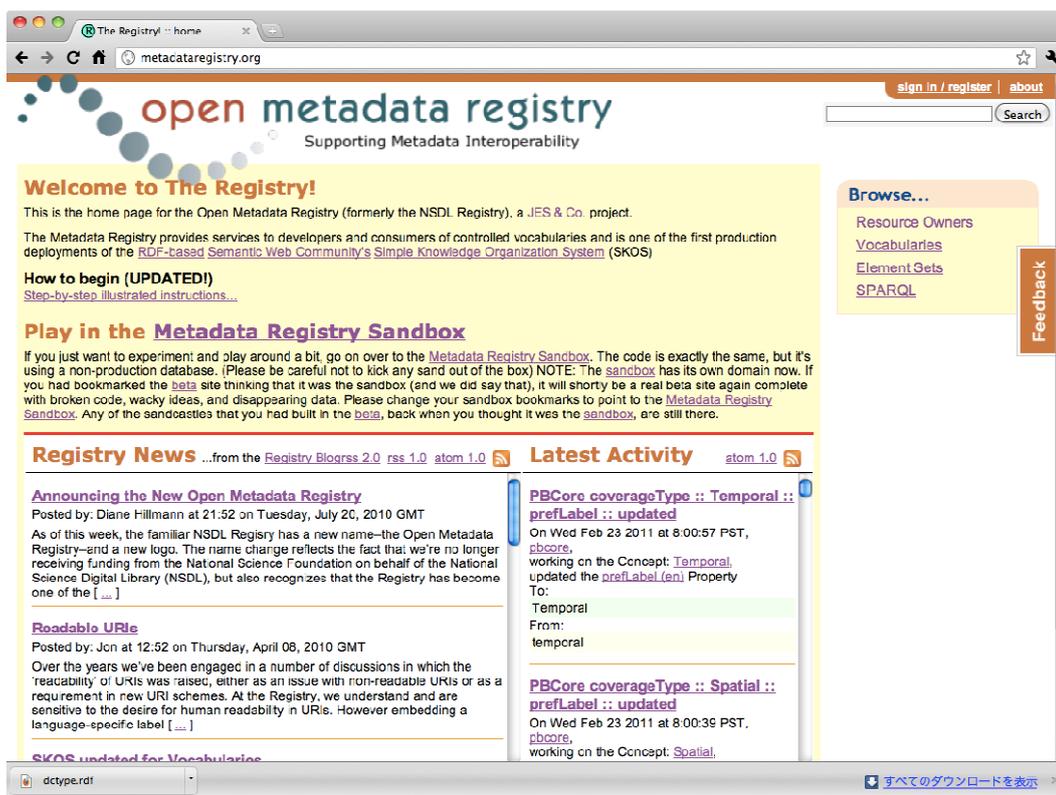
	語彙	項目記述規則	メタデータ変換	多言語 UI	スキーマの追加	API	バージョン管理
DCMI Registry	○	×	×	○	△	○	×
Meta Bridge	○	○	○	×	○	○	○

### 9.3.2 Open metadata registry

Open Metadata Registry <sup>18</sup> は、コーネル大学の Diane Hillmann やワシントン大学の Stuart Sutton 等を中心として、2005 年に開発が始まったメタデータ・スキーマ・レジストリである(図 9.12)。当初は米国立科学財団(NSF)の米国科学デジタル図書館イニシアチブ(NSDL)支援のもとに NSDL Registry という名前で開発が進められていたが、現在は JES & Co. <sup>19</sup> のプロジェクトとして進められている。これまで NSDL Registry と呼んでいたが、2010 年夏にその名称を Open Metadata Registry に変更した。なお、Hillman や Sutton は、DCMI の活動に初期の頃から関わっており、Open Metadata Registry は、DCMI Metadata Registry での経験や議論を基にして、様々なコミュニティがメタデータ語彙を作成・登録し、ネットワークを通じて提供するサービスの構築を目的としている。以下、Open Metadata Registry の目的、機能、Meta Bridge との違いについて述べる。

<sup>18</sup> Open Metadata Registry, <http://www.metadataregistry.org/>

<sup>19</sup> ES & Co., <http://www.jesandco.org/>



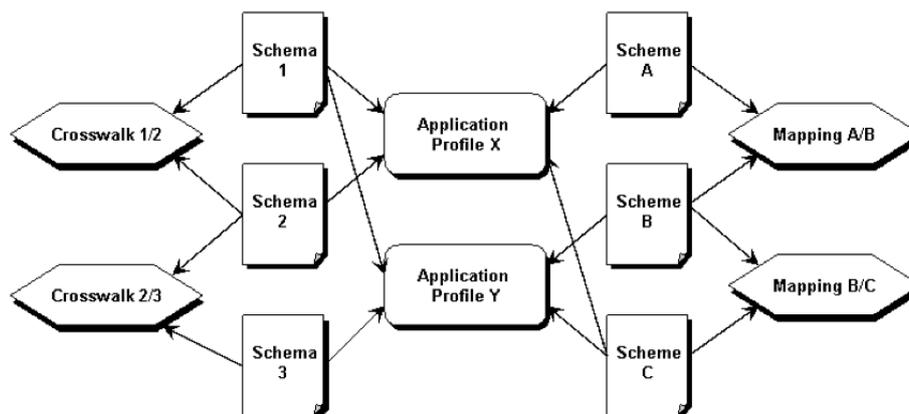
◆ 図 9.12: Open Metadata Registry

## (1) 目的

Open Metadata Registry は、メタデータの発見、再利用、標準化、相互利用性の向上をキーワードに、(A)メタデータ語彙の識別、(B)ソフトウェアを利用したメタデータ・スキーマの定義支援、(C)メタデータ・スキーマの提供の3点を目標に掲げている<sup>20</sup>。Open Metadata Registry では、メタデータ・スキーマを構成する以下の5種類の要素の蓄積と提供を目標としている(図 9.13)。

- Schema(属性語彙) – title, creator, publisher 等、情報資源の特徴を表したメタデータ記述のためのプロパティの定義
- Scheme(属性値語彙) – 統制語彙等、メタデータ項目に記述する値の定義
- Application Profile – メタデータ項目を記述する際のプロパティの組み合わせ方や記述する値に関する制約
- Crosswalk – 異なる Schema(属性語彙)に属するプロパティ間の関連の定義
- Mapping – 異なる Scheme(属性値語彙)に属する値の間の関連の定義

<sup>20</sup> D. Hillmann, S Sutton, et al., A Metadata Registry from Vocabularies Up: The NSDL Registry Project. DC2006, <http://ecommons.library.cornell.edu/handle/1813/9373>



◆ 図 9.13:Open Metadata Registry で提供する要素間の関連

## (2) 機能

現在公開されている Open Metadata Registry では、以下の機能を提供している。

- メタデータ・スキーマ(プロパティ、クラス、統制語彙、アプリケーション・プロファイル)の閲覧(図 9.14、図 9.15)
- メタデータ・スキーマの作成(図 9.16)
- 更新履歴とバージョン情報の閲覧(図 9.17)
- SPARQL を利用した検索と様々な形式での出力  
検索には SPARQL のインターフェースを備え、出力形式には RDF/XML, や Turtle 形式に、JSON, XML, HTML Table, プレーンテキスト等を選択することができる(図 9.18)。
- ユーザ登録  
Open Metadata Registry を利用するためのユーザ登録機能。メタデータ・スキーマの閲覧や検索機能は誰でも利用できるが、作成機能を利用するためにはユーザ登録が必要である。

Element Sets				
Label	Owner	Last Updated	Updated by	Actions
SKOS Community Extensions	SKOS Community	2011-01-18	aliman	
academics	Dapme Passah	2010-11-18	dapme	
SAAP Vocabulary	Matt Boris	2010-11-07	mjboris	
FRBRer open model	Gordon Dunsire	2010-10-04	Gordon Dunsire	
FRBR Entities for RDA	Metadata Management Associates	2010-09-17	DianeH	
FRSAD model	Gordon Dunsire	2010-07-24	Gordon Dunsire	
ISBD elements	Gordon Dunsire	2010-06-10	Gordon Dunsire	
Publication metadata	Max Planck Digital Library	2010-05-21	natasab	
FRBRer model	Gordon Dunsire	2010-03-23	Gordon Dunsire	
FRAD model	Gordon Dunsire	2010-02-15	Gordon Dunsire	
Application Profile for Ceramic Resources	Chandana Patra	2010-02-02	chandana	
RDA Group 1 Elements	Metadata Management Associates	2009-11-14	DianeH	
RDA Relationships for Works, Expressions, Manifestations, Items	Metadata Management Associates	2009-11-08	DianeH	
RDA Relationships for Concepts, Events, Objects, Places	Metadata Management Associates	2009-10-30	DianeH	
RDA Relationships for Persons, Corporate Bodies, Families	Metadata Management Associates	2009-10-12	DianeH	
Application Profile: Ceramic Repository	Chandana Patra	2009-08-20	chandana	
Title	Chandana Patra	2009-08-20	chandana	
Metadata Schema for Ceramic Resources	Chandana Patra	2009-08-13	chandana	
RDA Group 3 Elements	Metadata Management Associates	2009-06-23	DianeH	
RDA Group 2 Elements	Metadata Management Associates	2009-06-17	DianeH	

30 results

◆ 図 9.14:メタデータ・スキーマの一覧表示

Element Sets: Show detail for FRBR Entities for RDA			
Detail	Elements	History	Maintainers
<b>Detail</b>			
Owner:	Metadata Management Associates		
Label:	FRBR Entities for RDA		
<b>Namespace</b>			
Name:	FRBRentitiesRDA		
URI:	http://RDVocab.info/uri/schema/FRBRentitiesRDA (RDF)		
<b>Documentation</b>			
URL:			
Note:	Registration of FRBR entities for use in RDA.		
Tags:	Libraries		
<b>Defaults</b>			
Status:	New-Proposed		
Language:	English		
<b>Users +</b>			
<a href="#">List</a>   <a href="#">Get RDF</a>			

◆ 図 9.15:メタデータ・スキーマの詳細表示

<b>Detail</b>	
Owner:	<input type="text" value="Metadata Management Associates"/> <small>The individual or organization that registered the element set</small>
Label:	<input type="text" value="Project Properties"/> <small>A human-readable descriptive label for the element set. This will be displayed in lists of element sets.</small>
<b>Namespace</b>	
Name:	<input type="text" value="ProjProp"/> <small>A unique token representing this element set. This will be used as the prefix identifying this URI in RDF or a qname in XML.</small>
URI:	<input type="text" value="http://metadataregistry.org/uri/schema/ProjProp"/> <small>The URI of this element set. By default this will be a concatenation of the Registry default domain and the element set Name.</small>
<b>Documentation</b>	
URL:	<input type="text" value="http://example.com/myVocab"/> <small>A non-Registry URL that contains more information about this element set</small>
Note:	<input type="text" value="This vocabulary was developed in 1966 as part of an early project describing rodents."/> <small>A note about this element set</small>
Tags:	<input type="text" value="Mammalian zoology"/> <small>Tags identifying communities of practice that this element set addresses</small>
<b>Defaults</b>	
Status:	<input type="text" value="New-Proposed"/> <small>Default status for the properties of this element set</small>
Language:	<input type="text" value="English"/> <small>Default language for the properties of this element set</small>

◆ 図 9.16:メタデータ・スキーマの作成画面

**Element Sets: Show detail for FRBR Entities for RDA**

Detail Elements History Maintainers

Changed at	Property URI	Property	Statement	Action	User
2009-08-04 10:34	.../schema/FRBRentitiesRDA/Object	Object	uri	updated...	DianeH
2009-08-04 10:34	.../schema/FRBRentitiesRDA/Object	Object	subClassOf	updated...	DianeH
2009-08-04 10:33	.../schema/FRBRentitiesRDA/Event	Event	uri	updated...	DianeH
2009-08-04 10:33	.../schema/FRBRentitiesRDA/Event	Event	subClassOf	updated...	DianeH
2009-08-04 10:32	.../schema/FRBRentitiesRDA/Place	Place	uri	updated...	DianeH
2009-08-04 10:32	.../schema/FRBRentitiesRDA/Place	Place	subClassOf	updated...	DianeH
2009-08-04 10:31	.../FRBRentitiesRDA/Concept	Concept	uri	updated...	DianeH
2009-08-04 10:31	.../FRBRentitiesRDA/Concept	Concept	subClassOf	updated...	DianeH
2009-08-04 10:29	.../FRBRentitiesRDA/Subject	Subject	uri	updated...	DianeH
2009-08-04 10:28	.../schema/FRBRentitiesRDA/Family	Family	uri	updated...	DianeH
2009-08-04 10:28	.../schema/FRBRentitiesRDA/Family	Family	subClassOf	updated...	DianeH
2009-08-04 10:27	.../FRBRentitiesRDA/CorporateBody	Corporate Body	uri	updated...	DianeH
2009-08-04 10:27	.../FRBRentitiesRDA/CorporateBody	Corporate Body	subClassOf	updated...	DianeH
2009-08-04 10:25	.../schema/FRBRentitiesRDA/Person	Person	uri	updated...	DianeH
2009-08-04 10:25	.../schema/FRBRentitiesRDA/Person	Person	subClassOf	updated...	DianeH
2009-08-04 10:24	.../schema/FRBRentitiesRDA/Agent	Agent	uri	updated...	DianeH
2009-08-04 10:23	.../schema/FRBRentitiesRDA/Item	Item	uri	updated...	DianeH
2009-08-04 10:22	.../FRBRentitiesRDA/Manifestation	Manifestation	uri	updated...	DianeH
2009-08-04 10:18	.../FRBRentitiesRDA/Expression	Expression	uri	updated...	DianeH
2009-08-04 10:17	.../schema/FRBRentitiesRDA/Work	Work	uri	updated...	DianeH

120 results

rss2 rss1 atom

◆ 図 9.17:変更履歴の表示(どの利用者が、どの語に対して、いつ、どのような操作を行ったののかが表示される)

◆ 図 9.18:Open Metadata Registry の SPARQL 検索インターフェース

(3) Meta Bridge との違い

Open Metadata Registry と Meta Bridge は、メタデータ共有の支援を目的として、様々なコンテンツ提供者が作成したメタデータ・スキーマを作成・登録し、その情報を提供するという観点では類似した機能が多い。しかしながら、Application Profile の記述形式に関しては大きく異なる。Meta Bridge では DCMI が提唱する DSP を採用しているが、Open Metadata Registry では独自に決めた形式を基にしている(図 9.19)。また、Open Metadata Registry は語彙の作成と提供に特化しており、メタデータを変換するための機能は持っていない。

```

<!-- Element Set: Dryad Level One application profile -->
<rdf:Description rdf:about="http://metadataregistry.org/uri/schema/dryad-L1-AP">
  <dc:title xml:lang="en">Dryad Level One application profile</dc:title>
  <foaf:homepage rdf:resource="https://www.nescent.org/wg_digitaldata/Level_One_Application_Profile"/>
</rdf:Description>

<!-- Properties -->

<!--Property: available-->
<rdf:Description rdf:about="http://purl.org/dc/terms/available">
  <rdfs:isDefinedBy rdf:resource="http://metadataregistry.org/uri/schema/dryad-L1-AP" />
  <reg:status rdf:resource="http://metadataregistry.org/uri/RegStatus/1001" />
  <reg:name xml:lang="en">Available</reg:name>
  <rdfs:label xml:lang="en">available</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
</rdf:Description>

<!--Property: foaf:name-->
<rdf:Description rdf:about="http://xmlns.com/foaf/0.1/name">
  <rdfs:isDefinedBy rdf:resource="http://metadataregistry.org/uri/schema/dryad-L1-AP" />
  <reg:status rdf:resource="http://metadataregistry.org/uri/RegStatus/1001" />
  <reg:name xml:lang="en">FOAF Name</reg:name>
  <rdfs:label xml:lang="en">foaf:name</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
</rdf:Description>
    
```

◆ 図 9.19:Open Metadata Registry での Application Profile の定義例

◆ 9.3:Open Metadata Registry と Meta Bridge の機能比較

	語彙	項目記述規則	メタデータ変換	多言語UI	スキーマの追加	AP I	バージョン管理
Open Metadata Registry	○	○ (独自)	×	×	○	○	○
Meta Bridge	○	○ (DSP)	○	×	○	○	○

## 9.4 メタデータ・スキーマ・レジストリの課題と方向性

インターネットにおいて DNS (Domain Name System) がホスト名と IP アドレスを管理して名前解決のために必要不可欠な基盤であるように、メタデータ・スキーマ・レジストリは今後のセマンティック・ウェブの世界において、人や計算機がメタデータを理解して利用するために必要な基盤的サービスであると考えられる。そのために、今後レジストリは以下に示した課題に取り組む必要がある。

### レジストリ間相互運用性の向上

ひとつのレジストリにメタデータ・スキーマを集約して蓄積することが理想的ではあるが現実的には難しい。そこで、レジストリ間でメタデータ・スキーマとメタデータ・スキーマに関する情報を問い合せて流通させる仕組みが求められる。9.3 で述べたように、本プロジェクトの Meta Bridge 以外にも、これまでに DCMI Metadata Registry や Open Metadata Registry といったメタデータ・スキーマ・レジストリが開発されているが、レジストリ間の連携機能は考慮されていない。今後は、レジストリ間相互運用性の向上を目的として、レジストリ間でメタデータ・スキーマに関する情報を検索し、メタデータ・スキーマを流通させるために必要な要件の整理と実現が求められる。

### 高度なメタデータ・スキーマの設計支援

メタデータの流通性向上の観点から、新たなメタデータ・スキーマを開発しようとするコンテンツ提供者には、相互運用性の高いメタデータ・スキーマの設計が求められる。レジストリは、コンテンツ提供者の既存メタデータ・スキーマの調査、目的に合った語彙や記述規則の発見と再利用、そしてメタデータ・スキーマの設計を支援する。しかしながら、でき上がった語彙定義や記述規則が、

- ・ 十分なメタデータ記述能力を備えているか
- ・ 相互運用可能であるか
- ・ メタデータ変換を正しく行えるか

といった評価はしていない。今後は、高度なメタデータ・スキーマ設計支援を目指し、7 で述べたガイドラインを基礎とするメタデータ・スキーマの相互運用性の高さを評価する尺度の開発が求められる。

## 10. レジストリの機能要件

本プロジェクトで開発したレジストリの機能要件の、考え方とユースケース、および要件概要を示す。なお、準備段階で定義した機能要件のうち、開発スケジュールなどの関係でプロジェクト期間内には実装しなかったものについては、要件を示した上で、その旨注記する。

### 10.1 考え方とユースケース

本プロジェクトの提案概要における「新しいサービスの概要」をふまえ、

- 基盤構築時の記述規則収集・登録に関する要件
- 基盤を利用してコンテンツ提供者がメタデータを付与したり、新たなスキーマを構築するための要件

の 3 つの視点でユースケースを考え、メタデータ・スキーマ・レジストリに求められる機能を定義する。

#### 10.1.1 既存の記述規則を標準的な形で表現・公開する

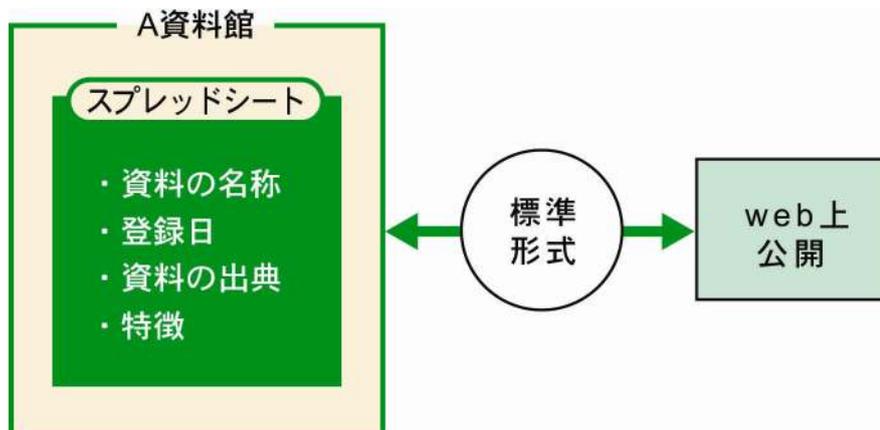
(a) ユースケースシナリオ A 資料館は、収集した資料を管理するために、簡易データベースソフトを用いている。データベースには、資料の名称、登録日、資料の出典(ソース)、特徴などを、受け入れ担当者が登録する。複数の担当者がデータを登録するので、登録内容がバラバラにならないよう規則を定め、担当者間で共有している。

データ記述の規則は、項目ごとにどのような値を書くかといった説明が表にまとめられ、スプレッドシートで管理されている。担当者が読めば理解できるが、これを用いてコンピュータがデータの整合性チェックを行なうことはできない。

A 資料館はこの収集資料の情報をウェブで公開しようと考えている。人が検索・閲覧するためのウェブページに加え、API などの形でもメタデータを利用できるようにしたい。そのためには、独自データベースの内容を標準的なメタデータ形式に変換するとともに、それがどのような規則に基づいて記述されているかも標準的な方法で公開する必要がある。とはいえ、担当者はメタデータ標準に関する専門知識を持っているわけではないので、簡単に標準形式データを作成する方法を望んでいる。

#### (b) レジストリに求められる機能

- メタデータの記述規則を登録し、公開する機能
- 一定の方法で書かれた記述規則を、専門知識がなくても登録できる機能

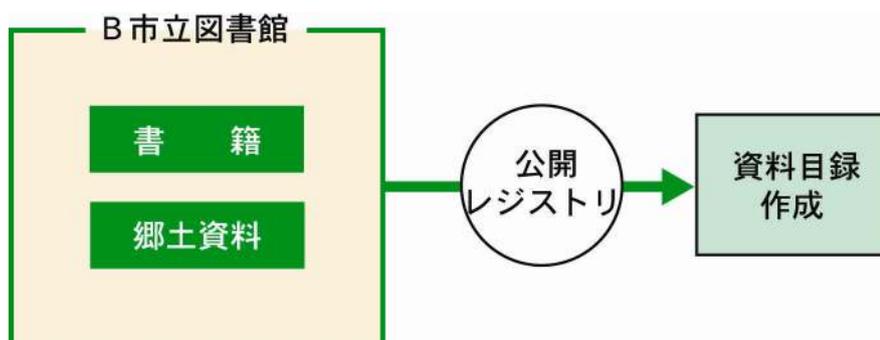


◆ 図 10.1: 資料館内部のデータベースに登録していた情報をウェブで公開する

### 10.1.2 メタデータを記述するために利用可能な記述規則を探す

(a) ユースケースシナリオ B 市立図書館は、通常の書籍に加え、郷土資料の収集・公開も始めた。この資料の目録を作るためには、書誌情報とは異なる、博物館系の記述規則を用いるほうが適切と思われる。しかし調べたところ標準仕様は複数存在するようであり、しかもその体系は複雑で、どんな項目を用いると将来的な共有のために有益なのか判断が難しい。

大きな博物館の記述規則、あるいは同様な規模で郷土資料を収集している他の公共図書館が採用している記述規則が公開されていれば、それを借用して資料目録を作成することができる。しかし、記述規則が公開されている例は少なく、あったとしてもどこを探せばその公開規則を見つけられるのかということからして分からない。代表的な記述規則が公開レジストリに登録され、検索できれば、目録設計の負荷を大きく軽減できる。



◆ 図 10.2: 図書館で書籍以外に郷土資料も収集・公開するようになり、異なる種類の目録記述規則が必要になる

#### (b) レジストリに求められる機能

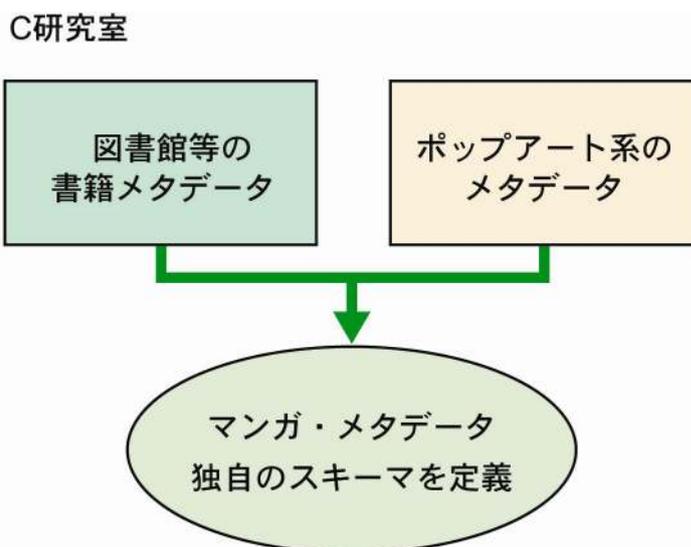
- 登録されている記述規則を検索、閲覧できる機能
- 目的別検索、類似規則比較など、適切な記述規則の選択を支援する機能

### 10.1.3 公開されているスキーマを参考に独自のスキーマを定義する

(a) ユースケースシナリオ C 研究室では、マンガを体系的に研究するために、マンガ・メタデータを作ろうと考えている。マンガは一般的には書籍として流通するので、基本メタデータは書籍の書誌情報で記述できるが、それ以外にマンガの内容を表すメタデータも必要となる。

この場合、マンガ記述メタデータを一から作ったのでは、開発効率が悪い上に、相互運用性を確保できない恐れがある。また、書誌情報の目録規則は標準化されているが、実際に図書館で利用されるのはそのサブセットであり、どういった項目が必要・十分なのかは、専門家でないとうりにくい。

図書館で実際に用いられている記述規則（目録規則から実際に使用するものを選択した実用規則）が公開されていれば、それを出発点にして、マンガの内容を表すメタデータを加えたカスタム記述規則を相互運用性の高い形で定義できる。マンガに近い分野（ポップアートなど）のメタデータ記述規則を検索できれば、これらを組み合わせることで新しい規則を定義することもできる。



◆ 図 10.3: マンガ・メタデータを設計するために、既存の記述規則を調べて参考にする

#### (b) レジストリに求められる機能

- 参考となる記述規則を検索、参照する機能
- 既存の記述規則をベースに、新たな規則を追加して独自規則を定義する機能
- 既存の記述規則から特定の項目記述規則を選び、組み合わせたり制約を変更して新たな記述規則を定義する機能

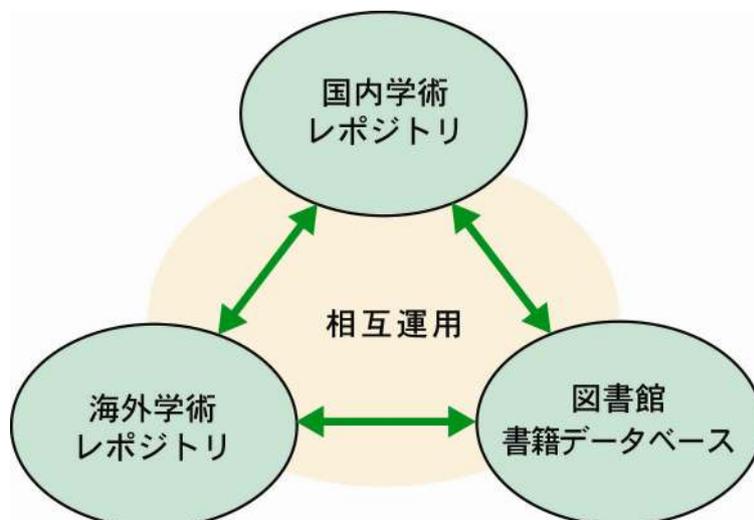
### 10.1.4 異なるメタデータ形式間での相互運用

(a) ユースケースシナリオ D 学術レポジトリは、国内の学術論文を収集してデータベースに登録し、検索利用できる公開サービスを提供してきた。キーワードや著者名を指定して検索する API も提供している。

論文情報をより広く活用できるよう、論文のメタデータを海外の学術レポジトリや、図書館の書誌データベースとも相互運用可能な形で提供したい。しかし、海外の論文メタデータには「読み」という概念がなかったり、書誌のメタデータでは「著者の順序」という概念がなかったりと、それぞれのメタデータに必要な記述フォーマットが微妙に異なっており、単にメタデータを公開するだけでは、利用しにくい。

公開用メタデータはシンプル DC のような汎用フォーマットに単純化する方法もあるが、オリジナルメタデータの詳細な情報が失われてしまう。オリジナルメタデータとシンプル DC 形式の両方を公開する方法もあるが、もっと他の形式で利用したいというニーズもあるかもしれない。

オリジナルメタデータと汎用フォーマットのマッピング規則を登録するレジストリがあれば、利用者が必要に応じた変換を行ない、より有効な形で利用できるのではないか。また、メタデータ共有のための標準モデル(読みの表し方など)や指針があれば、公開データをその形式に合わせることで、相互運用性が高まるのではないか。



◆ 図 10.4: 学術レポジトリの論文メタデータを、内外のデータベースと相互運用するために、異なる形式のメタデータを標準モデルに変換して交換する

#### (b) レジストリに求められる機能

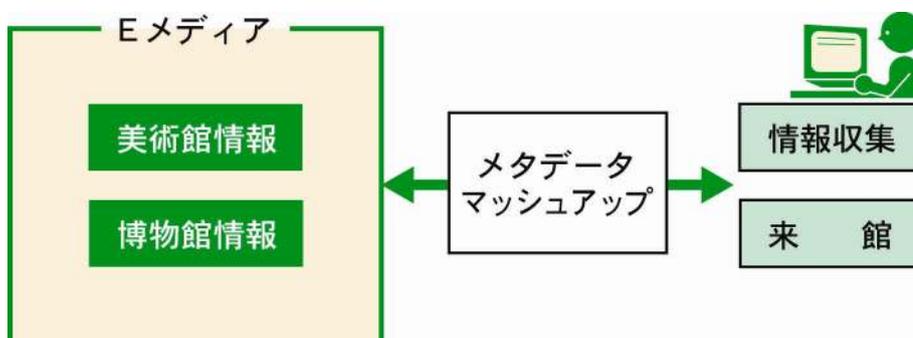
- 個別の記述規則を汎用語彙・フォーマットにマッピングする機能
- 記述規則定義に際し、相互運用性を高める方法を提案する機能

### 10.1.5 複数のメタデータをマッシュアップする

(a) ユースケースシナリオ E メディアは、各地の美術館、博物館の情報を集めて、利用者が目的別に検索できるサービスを提供している。現状では、各美術館、博物館のウェブサイトの情報ページをクロールし、それぞれ異なる記述方法の HTML から独自のアルゴリズムで展覧情報を抽出し、データベースに蓄積している。

HTML の情報ページは記述方法がまちまちだけでなく、ユーザがブラウザで閲覧することを前提にしているため、日付の年が省略されるなど、データ抽出には問題が多い。また、常設展示や館蔵品の情報は部分的にしか公開されておらず、データも不十分であるため、「あるテーマについて調査するために、利用できる資料がどの館に収蔵されているか」といった検索機能を提供することも困難である。

美術館、博物館が、収蔵品目録をメタデータとして公開すれば、情報の収集は効率的で確実になる。公開メタデータの記述方法が館ごとに異なっても、汎用的なメタデータ形式へのマッピングが提供されていれば、E メディアのサービスのためには十分に役に立つ。さらに、企画展の「イベント情報」メタデータが標準的な形で公開されれば、ユーザにお勧め展覧会を推奨するなどのサービスも可能となり、美術館、博物館の来館者増にも寄与できる。



◆ 図 10.5:各地の美術館、博物館の情報を集めて、利用者が目的別に検索できるサービスのために、それぞれのメタデータを収集する

#### (b) レジストリに求められる機能

- 第三者サービスの提供者が、記述規則についての専門的な知識なしに、異なるメタデータの意味を確認できる機能
- 登録記述規則を、API などの形で、コンピュータ・プログラムが利用できる機能
- 登録記述規則に基づくメタデータを、シンプルな汎用形式に変換する機能

## 10.2 機能要件:スキーマ登録に関する要件

レジストリに既存スキーマを登録し、維持更新するための要件を定義する。

### 10.2.1 RDF グラフとして表現されるスキーマを格納、管理できる

さまざまな分野のメタデータを交換する情報基盤とするため、レジストリに収録するスキーマは RDF(「4.2.データモデルの標準化:RDF」参照)で表現し、処理する。

#### (a) スキーマの RDF トリプルをデータベースに格納する 適切な RDF 処理ライブラリとデータベ

ース結合 API を用いて、スキーマを表現した RDF のトリプルをデータベースに格納し、照会、更新などの操作を行なう。

(b) **スキーマごとに RDF グラフを区別し、指定グラフのみを処理できる** 各スキーマのグラフを URI で名前付けする(グラフ URI)。各トリプルについて、それぞれが属するグラフ URI も保持(トリプル+グラフ URI=四つ組みなど)し、指定するグラフのみを対象に操作できるようにする。データセットなどの概念でもよい。

(c) **スキーマ RDF グラフのメタデータを管理できる** 各グラフ(スキーマ)について、名称、出所、登録日時などのメタデータを保持する。このメタデータも RDF で表現する。

### 10.2.2 RDF 形式の語彙定義を確認しながら登録できる

一つの名前空間で構成される語彙をレジストリに登録する。登録する語彙の定義はあらかじめ RDF で記述するものとする。

- RDF/XML および Turtle 形式を読み込んで RDF トリプルを取得できる
- 入力はローカルファイル選択のほか、URI 指定によるウェブからの読み込み、テキストエリアへの貼り付けに対応する

さらに維持管理のために、語彙に関するメタデータを登録する。

トピックマップ(4.4 参照)に関しては、XML TopicMaps (XTM) のファイルを読み込み、RDF に変換して処理できることとする。

### 10.2.3 表などの半形式的定義をもつ記述規則を対話的に登録できる

記述規則は、メタデータ記述規則定義言語(「6.3.メタデータ記述規則定義言語の提案」参照)として表現し、処理・格納する。記述規則のほとんどは表の形で提供されていると考えられるので、ここでは表からの対話的登録の要件を定める。

(a) **表から基本的な項目記述規則を生成する** 記述項目一覧の形で表がある場合、項目ごとに項目記述規則(StatementTemplate)を生成する。

(b) **自動生成した項目記述規則を対話的に修正できる** 生成した項目記述規則を項目名ごとに画面表示し、表から読み取った定義をユーザが修正できるようにする。

(c) **レコード記述規則の生成と登録** 項目記述規則の修正が完了したら、名前空間を対話的に決定し、レコード記述規則(DescriptionTemplate)としてデータベースに登録する(表ではなく RDF

形式の OWL-DSP として提供される記述規則があれば、ファイルを読み込んでこのステップから登録を開始する)。

維持・管理のために、スキーマ名称、登録者、登録日時などのメタデータも合わせて記述・登録する。スキーマの分類(分野)の登録についても検討する。

※RDF として登録するとトリプルの順序が失われてしまうので、たとえば:statementOrder のような形で並び順を保持しておく。

(d) **独自語彙の登録** の項目記述規則修正時に「独自語彙定義」マークをつけたプロパティを、対話形式に登録する。

(e) **シンプル DC へのマッピングの登録** 可能であれば、シンプル DC(「5.3.2.ダブリンコア」参照)へのマッピングを、記述規則登録者が作成する。これにより、シンプル DC への「公式マッピング」が提供されることになる。

#### 10.2.4 スキーマファイルを解析して変換の前処理を行なう

表形式スキーマからの DSP 生成の前処理として、各種フォーマットのスキーマファイルを DSP 化可能な表データに変換する。

(※本要件は、プロジェクト期間内には実装しない)

(a) **エクセルのワークシートの読み取り** エクセルで作成された表を直接読み取ることができる。

(b) **PDF ファイルの読み取り** 表が PDF ファイルになっている場合、ページ区切りにおけるノンブルや脚注、表のセル内改行の扱いなど、抽出テキストをそのまま表として扱えないケースが多いので、フィルタリングを行なう。

(c) **一覧表形式ではないスキーマ定義** 一覧表ではなく、項目ごとに表を作成しているスキーマの場合、DSP 化可能な一覧データに変換しておく必要がある。自動変換が難しい場合も想定されるので、何らかの対話的な変換方法を用意する。

#### 10.2.5 登録スキーマの更新ができる

登録済みスキーマの更新、バージョン管理、部分修正ができるようにする。

(a) **スキーマ全体の更新** スキーマ全体を更新する場合は、新規登録と同じ手順でデータを取り込み、新しいバージョンとして登録する。登録に際しては、メタデータは旧バージョンのものをデフォルトで示し、必要部分を変更登録できるようにする。

(b) **新旧バージョンの保存** 旧バージョンは削除せず、遡って参照できるようにする。

- 最新スキーマと保存スキーマの 2 種のテーブルを用意し、スキーマ登録時に両方のテーブルに登録する
- 最新スキーマテーブルには、スキーマのグラフ URI を用いて登録し、通常の検索などは全てこのテーブル、グラフ URI に対して実施する
- 保存テーブルでは、スキーマのグラフ URI に登録年月日を加えたグラフバージョン URI を用いて登録する
- スキーマ更新時には、(1)最新テーブルのグラフは新しいものに置き換え、(2)保存テーブルでは旧グラフはそのまま残して、新しいグラフをグラフバージョン URI を用いて追加登録する

たとえば国会図書館 DC-NDL スキーマのグラフ URI が <http://ndl.go.jp/dcndl/terms/> であるとき、2011-01-15 に登録したスキーマのグラフバージョン URI は <http://ndl.go.jp/dcndl/terms/?20110115> のような形にする。

(c) **スキーマの部分修正** スキーマの最新グラフは、誤りの訂正などのために修正を施すことができるようにする。ただしこの修正は、誤字訂正、登録ミスなど、基本的にスキーマの意味論を変えない範囲に限る。値制約の変更など意味論が変わる変更を伴う場合は、ひとつのターム、項目文テンプレートの修正であっても、スキーマ全体の更新として扱い、バージョンを変える。

### 10.3 機能要件:コンテンツ・メタデータ提供者のレジストリ利用に関する要件

基盤を利用してコンテンツ提供者がメタデータを付与したり、新たなスキーマを構築するための要件を定義する。

#### 10.3.1 登録されている記述規則、語彙定義の一覧、詳細を表示できる

画面のメニュー／リンクをたどることで、登録されている記述規則、語彙定義をブラウザできるようにする。

(a) **記述規則の一覧表示** 登録されている記述テンプレートを、登録組織別、または分類別に一覧して表示できる。一覧の表示にはラベルと説明を用いる。記述テンプレートを選択(クリック)すると、その詳細説明を表示する。

(b) **コード記述規則の詳細表示** レコード記述規則(記述テンプレート)のメタデータ(タイトル、作成者ほか)と、そこで記述する項目文テンプレートを表示する。

(c) **項目記述規則の詳細表示** 項目記述規則(項目文テンプレート)の制約規則を、表もしくは定義型リストとして表示する。

(d) **語彙定義の一覧表示** 登録されている語彙定義の RDFS を一覧して表示できる。一覧の表示にはラベルとコメント(の先頭部分)を用いる。語彙名を選択(クリック)すると、その詳細説明を表示する。

(e) **語彙定義の詳細表示** 語彙のメタデータ(タイトル、作成者ほか)と、語彙に含まれるターム(クラス、プロパティ)を表示する。

(f) **ターム定義の詳細表示**

- 各タームの URI、ラベル、および定義 RDF トリプルのプロパティ=値ペアを、表もしくは定義型リストとして表示する
- 上位プロパティ、上位クラス、定義域、値域の値となるタームは、登録されていればクリックすることでそれぞれの詳細を表示する
- プロパティの場合、このターム(プロパティ)を用いている項目文テンプレートをリスト表示する。これもクリックで詳細表示する

### 10.3.2 登録されている記述規則、語彙定義を検索し、該当するものを表示できる

利用法に応じてレジストリを検索し、必要な記述規則、語彙定義を発見できるようにする。

(a) **自由キーワードによる記述規則、語彙定義検索**

- 自由キーワードを用いて、記述規則、語彙定義のラベル、説明などのメタデータ項目をまたいだ検索を行なう
- 検索結果を記述テンプレート、項目文テンプレート、語彙定義、ターム定義に分けて一覧表示する。

(b) **語彙による検索** 語彙を指定して、その語彙を利用する記述テンプレートを一覧表示する。検索には記述テンプレートの生成と登録時に用意した、記述テンプレートの使用語彙リストを用いる。

(c) **プロパティによる項目記述規則の検索** プロパティを指定し、そのプロパティを用いる項目記述規則(項目文テンプレート)を検索、一覧表示する。

- プロパティは、登録語彙ごとにメニューなどで選択できるようにする

- 語彙が分からずプロパティ名のみが念頭にある場合 (creator など) は、自由入力として、ラベルを対象にまずプロパティを検索し、そのプロパティを用いて項目文テンプレートを検索する
- プロパティが下位プロパティを持つ場合 (たとえば dc:creator の下位プロパティとして ex:composer が独自に登録されている場合)、これらも含めて検索対象にするオプションを用意する

### 10.3.3 記述規則、語彙定義を複数フォーマットで取得できる

ウェブブラウザ画面での閲覧に加え、記述規則、語彙定義の RDF グラフそのものを取得、利用できるようにする。提供フォーマットは複数用意する。

#### (a) 複数形式の RDF 記述

- デフォルトで RDF/XML、ほかに Turtle を選択できるようにする
- ダウンロードに用いる URI は、コンピュータ間い合わせ用 REST の URI と同じになるようにする

#### (b) 定義 RDF を TopicMaps に変換して取得できる

### 10.3.4 独自記述規則の作成を支援できる

登録済み記述規則では不十分あるいはオーバースペックである場合、独自の記述規則を作成できる。登録済み記述規則をベースに、対話的に操作ができる。

(a) 独自の記述規則を対話的に定義・登録できる ウィザード的に手順を追いながら、独自の記述規則を定義・登録する機能を提供する。

(b) 登録済み項目記述規則を選んで、独自レコード記述規則に組み込むことができる 新しいレコード記述規則を作るときに、登録されている項目記述規則を部品として組み合わせることができる。

- ブラウズあるいは検索結果の規則一覧で、必要な項目文テンプレートを複数選択できる。
- 項目文テンプレートの詳細を確認してからの選択もできるよう、詳細画面でも同様に選択できる。

選択が終了したら、「記述テンプレートの生成と登録」と同様の画面でメタデータを入力する。このとき、項目文テンプレートの削除や追加などの変更を可能にする。

(c) 登録済み項目記述規則を編集して新しい規則を作成し、独自レコード記述規則に組み込むこと

**ができる** 検索しても適切な項目記述規則が見当たらない場合、既存の項目記述規則をベースにして制約記述を編集し、新たな項目記述規則を作成できるようにする。

- 編集の画面は、「自動生成した項目記述規則を対話的に修正」のステップに準じる
- コピー元となった項目記述規則との関係をメタデータに加える
- 作成した項目記述規則を登録し、独自レコード記述規則に組み込むことができる

#### (d) 新規の項目規則を定義・登録できる

- 空のテンプレートとなるダミーの項目文テンプレートを用意しておき、このテンプレートを編集することで、新規の項目記述規則を作成できるようにする
- 未登録プロパティを用いたい場合は、RDFS/OWL の形で記述されている語彙定義であれば、「RDF 形式の語彙定義を確認しながら登録」の手順に従いレジストリに追加登録できる。まずこの登録を行なって、登録済みプロパティとして利用可能にした上で、項目文テンプレートを編集する

### 10.3.5 複数記述規則間の関連を調べ、分かりやすく示すことができる

記述規則をブラウザ、検索しただけではどれが適切なものか分からない場合などに、複数の記述規則の類似関係を調べ、選択判断の補助とすることができる。

(※この要件は、プロジェクト期間内には実装しない)

(a) 複数の記述テンプレートを表形式で列挙し、共通プロパティ項目規則をハイライトできる  
複数の記述テンプレートの中で、同じプロパティを用いる項目文テンプレート(共通プロパティ項目規則)がどれであることを示す。

- 基準となる記述テンプレートを一つ選び、比較記述規則を選択すると、各規則に含まれる項目文テンプレートを縦に並べた表として表示できる
- 基準規則の項目文テンプレートに対して、比較記述規則の項目文テンプレートの中で同じプロパティを用いるものを調べ、ハイライト表示する(可能であれば線で結ぶなどする。あるいは同じ行に並ぶようにする)
- 同様に、上位・下位関係にあるプロパティを用いるものをハイライト表示する

(b) 共通プロパティ項目規則を表形式で複数列挙し、共通する制約規則をハイライトできる  
記述に用いるプロパティに注目し、同じプロパティを用いる項目文テンプレート(共通プロパティ項目規則)の間で、制約規則がどの程度共通か、あるいは異なるかを示す。

「プロパティによる項目文テンプレート検索」の結果表示のオプションとして、項目文テンプレートを横に、制約規則を縦に並べた表形式の表示を行い、共通する制約規則があればハイライト表示する。

(※本要件は、全体開発スケジュールにおいてゆとりがある場合に実施する)

(c) **プロパティ類似度による記述規則検索ができる** 選択した記述テンプレートに対し、登録されている他記述テンプレートそれぞれにおいて《共通プロパティを用いる項目文テンプレートの数》を調べ、その数が多い(一致度が高い)順に一覧表示する。オプションで、結果から選択した記述テンプレート間の関係を表を用いて表示する。

(※本要件は、全体開発スケジュールにおいてゆとりがある場合に実施する)

### 10.3.6 選択した記述規則に基づくメタデータ記述の支援できる

記述規則を利用したメタデータ活用のため、メタデータの検証、規則に基づくメタデータの作成・変換を支援する。

#### (a) メタデータが記述規則どおりに書かれているかどうか検証できる

- 入力フィールド、もしくはファイルから読み込んだメタデータを、選択した記述規則に対して検証する
- 規則に反する部分があれば、入力メタデータのどの箇所が、どの規則に合致しないかを分かりやすく示す
- 構文レベルではなく、RDF グラフに対する検証を行なう(一般に同じグラフを異なる構文で記述できる)

#### (b) 記述規則に基づいて、表形式のデータを RDF に変換できる

- 表データの 1 行目が項目名となっており
- 記述規則の項目文テンプレート名 (URI 参照のローカル名) が表の項目名と対応するとき
- 記述テンプレートで ID 欄となっている項の値目から各レコードの URI を生成し
- 項目文テンプレートのプロパティ、制約規則に基づいて、各項目のデータを RDF トリプルに変換する

(c) **記述規則に基づいて、データ入力の HTML フォームテンプレートを用意できる** 利用者は、このフォームの HTML ソースをコピーして、データベース用の入力フォームを容易に作成できる。レジストリでこのフォームにデータを入力して送信した場合は、前項の手順で RDF に変換した結果を表示する。

(※この要件は、プロジェクト期間内には実装しない)

#### (d) 記述規則に基づいて、データ入力のエクセル入力ウイザードを用意できる

(※この要件は、プロジェクト期間内には実装しない)

### 10.3.7 コンテンツの長期保存・利用のためのメタデータ記述標準

コンテンツの長期保存・利用のために、メタデータも長期にわたって適切に利用できるよう、記述規則を提供する。

(a) **記述規則のバージョンを管理し、指定時期の規則を参照できる** バージョン情報と登録日によって、過去の記述規則を参照することができる。

## 10.4 機能要件：サービス提供者の高度なメタデータ利用に関する要件

ウェブサービスなどの提供者が、コンテンツ提供者のメタデータを利用して、マッシュアップなど高度なサービスを展開するための要件を定義する。

### 10.4.1 レジストリを検索し、語彙、記述規則を調べることができる

コンテンツ提供者のメタデータをサービス提供者(メタデータ/コンテンツ利用者)が理解できるよう、メタデータに用いられている語彙や記述規則を検索して調べることができる。

(a) **キーワードで記述規則、語彙の一覧を検索できる** コンテンツ・メタデータ提供者のキーワード検索<sup>(a)</sup>を参照。

(b) **URI を指定して語彙、タームを検索できる** 語彙定義の RDF をグラフ URI を指定して取得するのは別に、URI から定義詳細を表示できるようにする。

(c) **メタデータから、使用語彙一覧と類似記述規則一覧を得ることができる** メタデータインスタンス(RDF)を入力して、使用されている語彙の一覧と、条件の近い記述規則の一覧を得ることができる。

- 名前空間宣言から使用している語彙を調べて、登録されているものを一覧表示。未登録語彙の場合は、名前空間 URI へのリンクを表示
- 使用語彙で記述規則を絞り込み、用いられているプロパティの種類が一致もしくは近いものを表示する

(※この要件は、プロジェクト期間内には実装しない)

### 10.4.2 メタデータ変換の支援

コンテンツ提供者のメタデータをサービス提供者(メタデータ/コンテンツ利用者)がより利用しやすい形で扱えるよう、変換を支援する。

(a) **メタデータを、JSON フォーマットに変換する** 登録記述規則によるメタデータ(インスタンス RDF)を JSON 形式に変換し、ウェブアプリケーションから利用しやすくする。

(※この要件は、プロジェクト期間内には実装しない)

**(b) メタデータを、TopicMaps に変換する**

**(c) メタデータを、シンプル DC による記述に変換する** マッシュアップなどの利用を容易にするために、異なるスキーマによるメタデータの共通化・標準化として、シンプル DC へのマッピングを提供する。

「公式マッピング」もしくは第三者によるマッピングが登録されていれば、そのマッピング ID を、メタデータ(作成者、日付など)とともに利用者に知らせる。

※記述規則は、たとえば title プロパティ値が構造化されて読みも含む場合、dc:title へは本来のタイトル(rdf:value 値など)のみを取り込み、読みの値は別プロパティに振り分けるなどの最適化に利用する。

### 10.4.3 コンピュータによる問い合わせインターフェース

サービス提供者は、人手によるレジストリ利用の他に、API 経由でプログラムからレジストリを利用し、自動化サービスに結びつけることができる。

**(a) 記述規則、語彙定義の RDF を、グラフ URI を指定して取得できる**

- {レジストリ URI}?graph={グラフ URI} の形で問い合わせ、取得できる
  - パラメータとしてフォーマットを指定できる (format=xml,turtle,json)。デフォルトは XML
- (※フォーマット指定は、プロジェクト期間内には実装しない)

**(b) キーワード検索、プロパティ指定検索を、REST API で実行できる**

- 結果は記述規則、語彙定義のグループに分けた上で、スキーマのラベルと前項のグラフ取得 URI をセットにする
- オプションで結果フォーマットを指定できる (XML もしくは JSON、デフォルトは XML)

**(c) 登録マッピングを用いて、URI で指定したメタデータを変換できる** ウェブアプリケーションは、変換アルゴリズムを自身のプログラムに組み込む代わりに、レジストリのサービスをゲートウェイとしてメタデータを変換・利用することも可能にする。

- dumb down マッピングとしてあらかじめ登録したシンプル DC へのマッピングと、変換対象メタデータの URI を、{レジストリ URI}?map={マッピング ID}&source={対象メタデータ URI} の形で指定する
- レジストリは、対象メタデータ URI から取得したデータを登録マッピングのアルゴリズムで変換し、シンプル DC としてのメタデータを返す
- オプションで結果フォーマットを指定できる (XML もしくは JSON、デフォルトは XML)

## 10.5 簡易 DSP から OWL-DSP への変換

6.4 で定義した簡易 DSP による記述規則をレジストリに取り込むために、OWL による DSP 定義言語 (OWL-DSP) に変換する対応関係を定義する。

### 10.5.1 名前空間と管理情報

(a) **名前空間宣言** 名前空間宣言ブロックから、名前空間接頭辞宣言を生成する。標準接頭辞マッピングはそのまま用いてよいが、名前空間ブロックで標準接頭辞に異なる URI がマッピングされているときは、その値で上書きする。

DSP 定義言語を用いるため、表 4.1 の接頭辞と名前空間の宣言を加えておく。

◆ 表 10.1: 追加標準名前空間

接頭辞	名前空間 URI	役割
dsp:	http://purl.org/metainfo/terms/dsp#	DSP 定義
reg:	http://purl.org/metainfo/terms/registry#	メタデータや簡易 DSP の情報をレジストリで保持

(b) **記述規則オントロジーのメタデータ** 簡易記述から OWL を生成する際に、次のメタデータを対話的に取得する。ここで、全レコード記述規則を合わせた記述規則全体を、個々の規則と区別するために記述規則オントロジーと呼ぶ。

- 記述規則オントロジーのラベル (reg:title) : 規則ファイル名から拡張子を除いた名前をデフォルトでラベルとして示す
- 記述規則オントロジーの名前空間 URI : 名前空間宣言に @base がある場合はデフォルトでこの URI を示す。@base がない場合は、上記ラベルの値を利用してシステムがデフォルトを示す
- 作者 (reg:creator) : ログインして利用する前提で、システムがデフォルトを示す
- 作成日時 (reg:created) : 登録操作日をデフォルトで示す
- バージョン (reg:version) : (デフォルトの表示は検討)
- コメント (reg:comment) : オプション

作者、作成日とは別に、登録ユーザ (reg:owner)、登録日時 (reg:registered) も記録する

### 10.5.2 レコード記述規則の表現

- 前項において対話的に取得したメタデータを用い、記述規則オントロジー自体を owl:Ontology として表現する
- 各レコード記述規則 ID から、主語 URI<#記述規則 ID>を生成し、dsp:DescriptionTemplate

のインスタンスとする。主規則ブロック ID が省略された場合は<#MAIN>を主語として用いる

- ID 型項目規則がある場合、dsp:valueURIOccurrence プロパティ値を"mandatory"とする。またレジストリとしては reg:idField プロパティ値を ID 型項目規則のラベルとし、ID 列を記録しておく
- ID 型項目規則でクラスが与えられている場合、dsp:resourceClass として表現する
- ID 型項目規則で値制約として名前空間が与えられている場合、レジストリを用いた表データ→RDF 変換用に、reg:resourceNsURI として保持しておく

```
<> a owl:Ontology;
  rdfs:label "国立国会図書館書誌記述";
  reg:created "2011-01-15";
  reg:creator ex:aRegisteredUserId;
  reg:version "第 1.1 版".

<#MAIN> a dsp:DescriptionTemplate;
  dsp:valueURIOccurrence "mandatory";
  dsp:resourceClass foaf:Document;
  reg:idField "書誌 ID";
  reg:resourceNsURI ndlbooks:;
  reg:statementOrder "書誌 ID,タイトル,...";
```

※簡易 DSP での項目規則の出現順序は、項目規則のラベルを出現順にカンマ区切りでつなぎ、reg:statementOrder としてレジストリ用に保存しておく。

(a) dsp:valueURIOccurrence と dsp:resourceClass 簡易 DSP では、空白ノードとなるレコード記述規則にクラスを指定することはできない(クラスの指定は ID 型記述規則で行なう)が、これは DSP 定義ならば、dsp:valueURIOccurrence が "disallowed" で、dsp:resourceClass を持つ dsp:DescriptionTemplate として定義できる。

```
<#構造化タイトル> a dsp:DescriptionTemplate;
  dsp:valueURIOccurrence "disallowed";
  dsp:resourceClass foaf:Agent;
  rdfs:subClassOf ...
```

URI あり、空白ノードのいずれも可の場合は、dsp:valueURIOccurrence を"optional"とする(あるいはこのプロパティを省略する)。

### 10.5.3 項項目記述規則からクラス制約の生成

項目記述規則を `dsp:StatementTemplate` とし、`<#MAIN>`をそのサブクラスとすることで制約を与える。

- 項目記述規則をレコード規則 ID-項目規則 ID として生成する URI で表し、`dsp:StatementTemplate` クラスのインスタンスとする。これを`<#MAIN>`と `rdfs:subClassOf` で結びつける。ただし ID 型項目規則はプロパティの制約ではないので、`dsp:StatementTemplate` を生成しない
- 項目記述規則名を `rdfs:label` として保持する
- 記述規則のプロパティを、`owl:onProperty` で結びつける
- 出現回数制約を、`owl:minQualifiedCardinality`、`owl:maxQualifiedCardinality` で表現する。最小回数 0、最大回数-(制約なし)は記述しなくてよい。最大値、最小値が同一の場合は `owl:qualifiedCardinality` を用いる。ただし最小回数が数字でない場合は、その値をそのまま `dsp:cardinalityNote` のリテラル値とし、`owl:minQualifiedCardinality` は用いない
- 値制約がリテラル値の場合は `owl:onDataRange`、実体値の場合は `owl:onClass` で表現する。
- コメントを `rdfs:comment` として表現する

```
<#MAIN> a dsp:DescriptionTemplate;  
  dsp:valueURIOccurrence "mandatory" ;  
  reg:idField "書誌 ID" ;  
  rdfs:subClassOf <#MAIN-タイトル>, ... .
```

```
<#MAIN-タイトル> a dsp:StatementTemplate ;  
  rdfs:label "タイトル" ;  
  owl:onProperty dct:terms:title ;  
  owl:qualifiedCardinality 1 ;  
  owl:onClass <#構造化タイトル> ;  
  rdfs:comment "文書の表題" .
```

(a) **リテラル値制約の OWL 表現** 値制約がリテラル値の場合は `owl:onDataRange` の目的語として制約を書く。

◆ 表 10.2: リテラル値制約と OWL の対応

値制約	owl:onDataRange
	rdfs:Literal
xsd:decimal	xsd:decimal
"バナナ" "りんご" "みかん"	[owl:oneOf("バナナ" "りんご" "みかん")]

(b) **実体値制約の OWL 表現** 実体値(構造化、参照値)の場合は owl:onClass の目的語として制約を書く。特定の語彙(名前空間)の場合、その語彙を表す匿名クラスを owl:onClass の目的語とする。特定の名前空間の語の集合からなる匿名クラスは dsp:inScheme で示す。

たとえばプロパティが ex:acceptableCard、出現回数が 1/-、値タイプが参照値で値制約が card:VISA card:AMEX の場合、

◆ 表 10.3: 構造化タイプ値制約と OWL の対応

値制約	owl:onClass
#構造化タイトル	<#構造化タイトル>
foaf:Agent	foaf:Agent
foaf:Person rda:Family	[owl:unionOf(foaf:Person rda:Family)]

◆ 表 10.4: 参照値タイプ値制約と OWL の対応

値制約	owl:onClass
ndlsh:	[dsp:inScheme ndlsh:]
ndlsh: bsh:	[owl:unionOf([dsp:inScheme ndlsh:] [dsp:inScheme bsh:])]
card:VISA card:AMEX	[owl:oneOf(card:VISA card:AMEX)]

```
[a dsp:StatementTemplate ;
  owl:onProperty ex:acceptableCard ;
  owl:minQualifiedCardinality 1 ;
  owl:onClass [
    owl:oneOf (card:VISA card:AMEX)
  ]
]
```

という OWL クラス制約を生成する。

## 11. レジストリの設計

前章に記載されたメタデータ・スキーマ・レジストリの機能要件に従い、本システムのレジストリ設計を報告する。

### 11.1 開発言語・ライブラリの選定

システム開発を行うプログラミング言語、およびメタデータ・スキーマをシステムで扱うために必要なライブラリを以下の通り選定する。

#### 11.1.1 開発言語: Java

Web アプリケーションの開発では、様々な言語を使用する事が出来るが、『Java』を選定する。選定の理由については、以下の3つを挙げる。

- プラットフォームに依存しない言語
- 習得している開発者が多いため、開発リソースを確保しやすく、また、要員コストを低く押さえる事ができる
- 後述する RDF を扱うためのライブラリ(Jena/Ontopia)を利用出来る

ライセンスはGNU General Public License

参照URL:<http://www.oracle.com/technology/global/jp/tech/java/index.html>

#### 11.1.2 データベース : PostgreSQL

メタデータ・スキーマを保存する手段として、データベースを使用する。そのデータベースとして『PostgreSQL』を選定する。選定の理由については、以下の3つを挙げる。

- オープンソースライセンスであるため、導入時のコストを低くすることができる
- ライブラリ『Jena』が正式にサポートしているデータベースソフトウェアである
- 多くのシステムで使用実績がある

ライセンスはThe PostgreSQL License

参照URL:<http://www.postgresql.org/>

#### 11.1.3 ライブラリ、フレームワーク

(1) Jena

RDFを扱うためのJavaライブラリとして、『Jena』を選定する。

Jena は、セマンティックWebアプリケーション開発のためのフレームワークである。要件として求められているほぼ全ての機能が、このJenaではサポートされている。

ライセンスはBSD License

参照URL:<http://sourceforge.net/projects/jena/>

(2) Ontopia

トピックマップを扱うためのJavaライブラリとして、『Ontopia』を選定する。

Ontopiaは、トピックマップのためのリーディングアプリケーション開発プラットフォームである。

Ontopiaを使用する事によって、RDFからの変換など、要件として求められている機能を実現する事ができる。

ライセンスはApache2.0ライセンス

参照URL:<http://code.google.com/p/ontopia/>

## 11.2 基本設計

システムの機能要件に従い、システム構成、機能の検討結果を報告する。

### 11.2.1 機能一覧

#### (1) 開発対象範囲

機能	画面・機能名称
語彙	語彙定義登録
	語彙定義登録エラー
	語彙定義登録確認
	語彙定義完了
	ターム詳細表示
	バージョン一覧
	登録済語彙一覧
	語彙定義検索一覧
	語彙定義参照
	ターム詳細表示
	RDF/XML 出力
	Turtle 出力
	TopicMaps 出力
	キーワード検索(語彙定義)
	キーワード検索結果
記述規則	記述規則ファイル取込
	記述規則エラー
	記述規則編集
	記述制約追加編集
	記述制約検索
	記述規則(構造化)追加編集
	参照設定
	バージョン一覧
	未登録語彙登録
	記述規則完了
	記述規則一覧
	ターム検索
	記述規則一覧

	記述規則名前空間設定
	記述規則検索一覧
	記述規則参照 (RDF 出力は除く)
	記述規則参照 (構造化)
	記述制約参照
	記述制約検索
	RDF/XML 出力
	Turtle 出力
	TopicMaps 出力
	dumb down マッピング
	キーワード検索 (記述規則)
	キーワード検索結果
メタデータ インスタンス 変換	メタデータRDF生成 (CSV 取込)
	メタデータ変換
	dumb down
コンピュー タによる問 い合わせ (API 提供)	語彙検索
	記述規則検索
	メタデータインスタンス変換

## (2) 開発対象範囲外

※技術ワーキンググループでの検討の結果期間の制約で実装を見送った機能

機能	画面・機能名称
語彙	語彙定義登録 (TopicMaps の取り込み)
	JSON 形式での出力
記述規則	不定期フォーマットから記述規則を作成
	JSON 形式での出力
	記述規則を HTML の画面として表現
	記述規則に従ったマクロチェックを行う Excel 出力
	複数の記述規則間の関連を分かりやすく示す
	ダム・ダウンマッピングを利用者が修正できる
	ダム・ダウンマッピングに ID を付与し再利用できる
ダム・ダウンマッピングアルゴリズムを JavaScript や PHP で提供する	

## 11.2.2 アプリケーション構成図

(1) Apache

httpアクセスを受け取るフロントアプリケーション

(2) Tomcat

JavaをWeb上で動かすためのアプリケーション

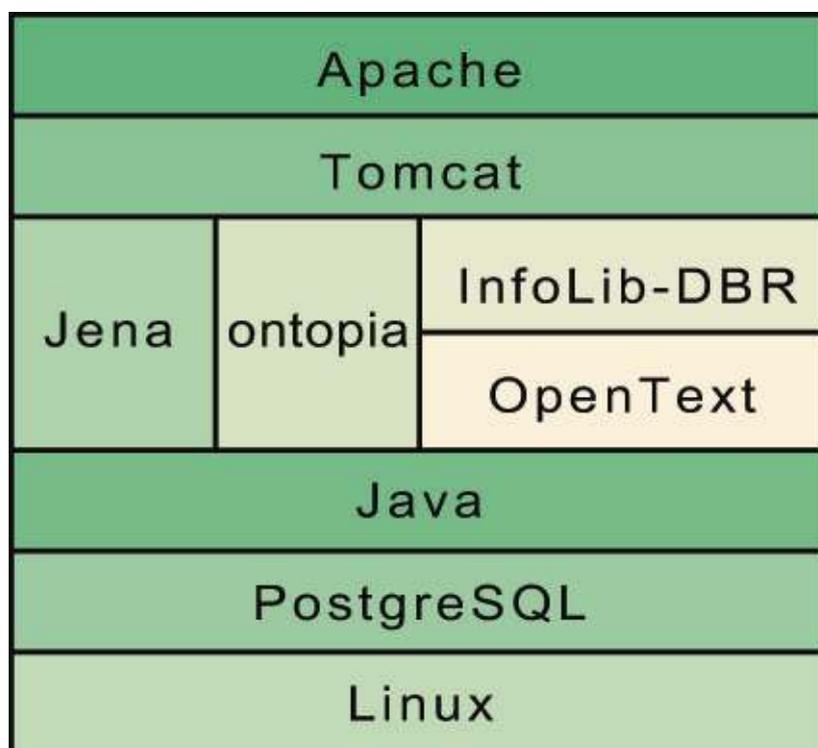
(3) InfoLib-DBR

グループ、ユーザの管理としてインフォコム(株)の製品を提供

(4) OpenText

将来、メタデータインスタンスの横断検索を行うことを目的としている。

(インフォコム社製のパッケージソフトウェアを使用)



## 11.3 ユーザインターフェース設計

基本設計で示した機能のうち、画面操作や入出力されるファイル等のユーザインターフェースに関する機能を以下に報告する。

### 11.3.1 語彙定義登録機能

#### (1) 概要

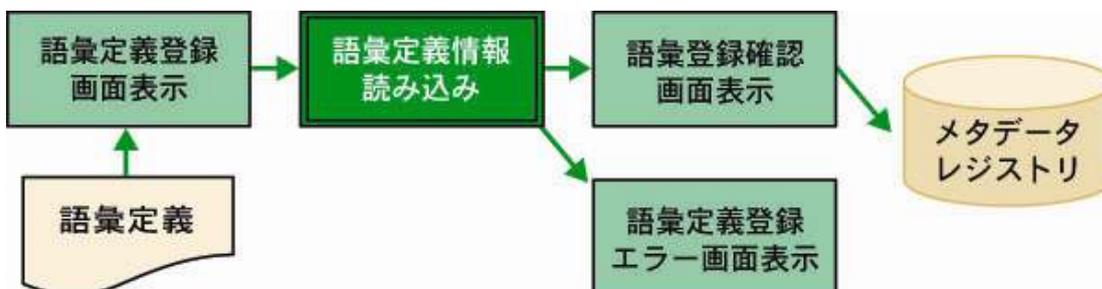
本機能は、各機関で管理している語彙定義をユーザが容易に本システムに登録し、確認できるユーザインターフェースを提供する。

語彙定義については、本システムでの新規作成は行わない。語彙定義を作成するユーザは、ある程度の知識を保有することを前提とし、本システムに登録する語彙定義は、ユーザがあらかじめRDF/XML、Trurtle形式で記述したものとす。

入力方法は、ローカルファイルのアップロード指定、URL指定によるウェブからの読み込み、テキストエリアへの貼り付けに対応し、自由度を持たせる設計とした。

読み込むファイルの文字コードについても、UTF-8を基本としているが、他の文字コードでも変換して読み込みが行えることを検討し、読み込むことが可能(文字化けが発生していないことを確認)であれば、登録することができることとした。

#### (2) 処理概要



- ① 登録する語彙定義をユーザがあらかじめ用意する。(RDF/XML、Trurtle形式で記述)
- ② 語彙定義登録画面にて、ユーザの環境にあった入力方法で取込を行えるように「ローカルファイル取込」、「URL指定によるウェブ取込」、「テキストエリア取込」に対応。

## ◆語彙定義登録

③ 語彙定義情報を読み込み。

④ エラーが発生した場合は、語彙定義登録エラー画面へ遷移する。

入力ファイルのエラーが発生した場合は、どの部分でどのようなエラーが発生したかをユーザに分かるように示す。

## ◆語彙定義登録エラー

## ◆読み込みファイル

```

22 :titleAndStatementOfResponsibilityArea↓
23   a owl:ObjectProperty ;↓
24   rdfs:label "titleAndStatementOfResponsibilityArea";↓
25   rdfs:comment "1.1 タイトルと責任表示に関する事項";↓
26   rdfs:domain :BibliographicDescription ;↓
27   rdfs:range :TitleAndStatementOfResponsibilityArea ↓
28 ↓
29 :titleProper↓
30   a owl:DatatypeProperty ;↓
31   rdfs:label "titleProper";↓
32   rdfs:comment "1.1.1 本タイトル";↓
33   rdfs:domain :TitleAndStatementOfResponsibilityArea .↓

```

- 29行目でエラー。⇒27行目のピリオド「.」がないことが分かる

- ⑤ エラーが発生しない場合は、語彙定義登録確認画面へ遷移する。  
読み込んだ語彙定義は、ユーザに分かりやすい形で画面表示する。  
語彙定義自体の説明を画面上部にメタデータとして、表示し、タームは、クラス、プロパティ、その他に分類して表示。  
クラス、プロパティ、その他のタームのリンクをクリックすることで、ターム詳細画面にタームの詳しい内容を表示。

◆語彙定義編集

The screenshot shows the 'META BRIDGE' interface for editing a vocabulary definition. The main content is titled '語彙定義参照' (Vocabulary Definition Reference). It features several sections:

- メタデータ (Metadata):** A table with fields like '名前空間 (URI)', 'バージョン', 'タイトル', 'コメント', '作成者', '作成日', and 'その他'. A yellow callout box labeled 'メタデータ' is overlaid on the right side of this section.
- クラス (Class):** A table with columns 'ローカル名', 'ラベル', '上位クラス', and 'コメント'. It lists 'Schema' and 'User'. A yellow callout box labeled 'クラス' is overlaid on the right side.
- プロパティ (Property):** A table with columns 'ローカル名', 'ラベル', '上位プロパティ', '定義域', and '値域'. It lists various properties like 'derivedFrom', 'owner', 'creator', etc. A yellow callout box labeled 'プロパティ' is overlaid on the right side.
- その他 (Other):** A table with columns 'ローカル名', 'ラベル', and 'タイプ'. A yellow callout box labeled 'その他' is overlaid on the right side.

- ⑥ 登録ボタンで本システムのレジストリに登録。

### 11.3.2 記述規則登録機能

#### (1) 概要

本機能は、各機関のメタデータの記述規則を本システムに登録し、確認できるユーザインターフェースを提供する。

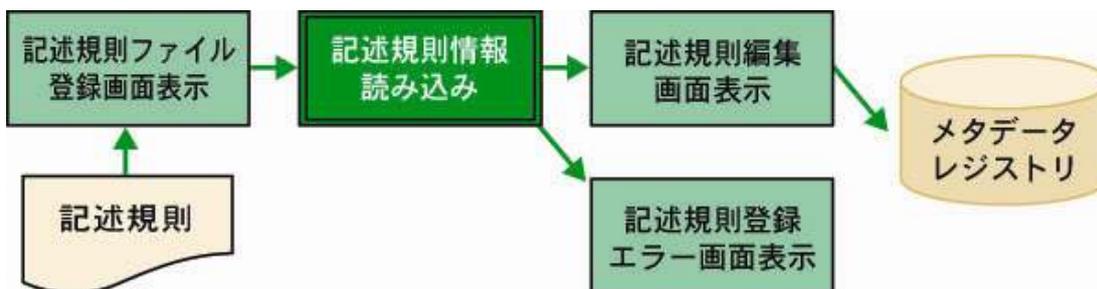
本システムへのレジストリに登録する記述規則は、簡易DSP(タブ区切テキスト形式)やRDF/XML、Trurtle形式で記述されたものとした。

記述規則に関しては、ユーザが記述しやすいようにタブ区切テキスト形式に対応し、メタデータの記述の表形式の項目一覧がある場合に容易に入力しやすいようにした。

入力方法は、ローカルファイルのアップロード指定、テキストエリアへの貼り付けに対応し、自由度を持たせる設計とした。

語彙定義の登録と同様に読み込むファイルの文字コードについても、UTF-8を基本としているが、他の文字コードでも変換して読み込みが行えることを検討し、読み込むことが可能(文字化けが発生していないことを確認)であれば、登録することができることとした。

#### (2) 処理概要



- ① 簡易DSP(タブ区切テキスト形式)やRDF/XML、Trurtle形式で記述された項目規則をユーザが用意する。
- ② 記述規則ファイル登録画面にて、ユーザの環境にあった入力方法で取込を行えるように「ローカルファイル取込」、「テキストエリア取込」に対応。

### ◆記述規則ファイル登録

- ③ 記述規則情報を読み込み。
- ④ エラーが発生した場合は、記述規則登録エラー画面へ遷移する。  
 入力ファイルのエラーが発生した場合は、どの部分でどのようなエラーが発生したかをユーザが分かるように示す。

### ◆記述規則登録エラー(タブ区切テキスト形式)

### ◆読み込みファイル

8	[MAIN] ↓
9	項目規則名>プロパティ> 最小> 最大> 値タイプ> 値制約> 説明 ↓
10	書誌ID> foaf:Document> 1> 1> ID> rdf:books:> 文書のID ↓
11	タイトル> dcterms:title> 1> a> 構造化> #構造化タイトル> 文書の表題 ↓
12	著者> dcterms:creator> あれば必須 -> 構造化> foaf:Agent> 文書の作者 ↓
13	発行日> dcterms:issued> あれば必須 1> 文字列> xsd:date> 文書の発行日 ↓

- 11行目でエラー ⇒ 最大出現回数「a」なので、不正であることが分かる

- ⑤ エラーが発生しない場合は、記述規則編集画面へ遷移する。  
読み込んだ記述規則は、ユーザに分かりやすい形で画面表示する。  
記述規則自体の説明として、どのような機関のどのような分野の記述規則か画面上部に表示し、メタデータの項目一覧を項目順に表示。  
各項目がどのような使用プロパティを利用し、値制約があるのかが一覧で表示される。また、構造化されている場合は、記述規則(構造化)項目として、別の一覧として、表示。  
リンクをクリックすることで、記述制約の詳細やタームの詳細を表示。

◆記述規則編集画面

The screenshot shows the 'META BRIDGE' metadata editing interface. The main content area is titled '記述規則編集' (Description Rule Editing). It includes several sections:

- Output Buttons:** RDF/XML出力, Turtle出力, TopicMaps出力, 登録, 戻る.
- Metadata Section (メタデータ):**
  - 名前空間 (URI): http://www.infocom.co.jp/test7
  - バージョン: (empty)
  - バージョン情報 (reg:version): 第 1.0 版
  - タイトル \* (reg:title): DSP記述規則サンプルver01
  - コメント (reg:comment): DSP取り込みのサンプルです。
  - 作成者 (reg:creator): インフォコム
  - 作成日 (reg:created): 2011-02-16
- Description Rules Table (記述規則):**

ラベル	プロパティ/クラス	値域	記述内容	出現	出現	出現
書誌ID	foaf:Document			1	1	nd books:
タイトル	dcterms:title	rdfs:Literal	文書の表題	1	1	<#構造化タイトル>
著者	dcterms:creator	dcterms:Agent	文書の作者	0	1	foaf:Agent
発行日	dcterms:issued	rdfs:Literal	文書の発行日	1	1	dcterms:W3CDTF
主題	dcterms:subject		文書の主題	0	-	nd shi: bsh:
- Structured Description Rules (記述規則 (構造化) 項目):**

ID	タイトル	コメント
<a href="#">#構造化タイトル</a>		

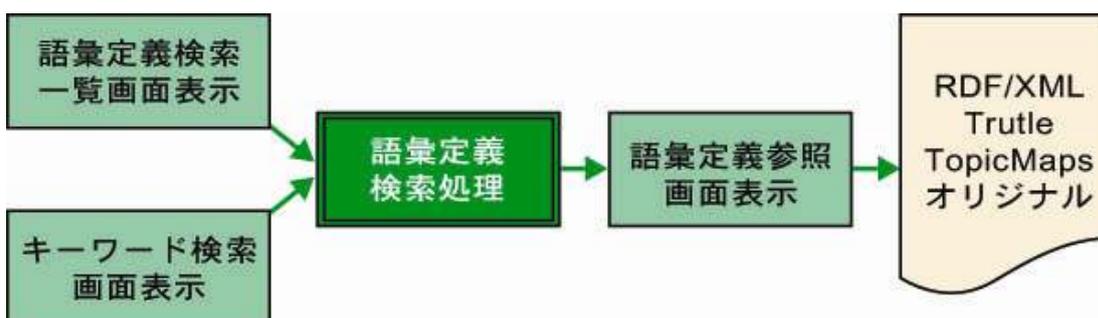
- ⑥ 登録ボタンで本システムのレジストリに登録。

### 11.3.3 語彙定義検索およびファイル出力機能

#### (1) 概要

本機能は、レジストりに登録されている語彙定義を登録組織やキーワードなどの検索条件を指定し、RDF/XML、Trurtle、TopicMaps形式で出力できるユーザインターフェースを提供する。ファイル登録したものについては、オリジナルのファイルとして、出力することができるように対応。

#### (2) 処理概要



#### 〈語彙定義検索一覧〉

- ① 語彙定義検索一覧画面にて、ユーザが目的の語彙を検索しやすいように登録組織、プロパティやクラスを含むパターンで語彙定義を検索できるように対応。  
入力した内容で、レジストリの検索処理を実行し、検索結果を画面に表示する。

#### ◆ 語彙定義検索一覧画面

選択	語彙定義名	コメント	URI	語彙定義 利用記述規則数	作成日
<input type="checkbox"/>	<a href="#">Metadata terms related to the DCMI Abstract Model</a>		<a href="http://purl.org/dc/dcam/">http://purl.org/dc/dcam/</a>	4	2011-02-15
<input type="checkbox"/>	<a href="#">DCMI Type Vocabulary</a>		<a href="http://purl.org/dc/dcmitype/">http://purl.org/dc/dcmitype/</a>	0	2011-02-15
<input type="checkbox"/>	<a href="#">Dublin Core Metadata Element Set, Version 1.1</a>		<a href="http://purl.org/dc/elements/1.1/">http://purl.org/dc/elements/1.1/</a>	18	2011-02-14
<input type="checkbox"/>	<a href="#">DCMI Metadata Terms in the /terms/ namespace</a>		<a href="http://purl.org/dc/terms/">http://purl.org/dc/terms/</a>	19	2011-02-14
<input type="checkbox"/>	<a href="#">The RDF Vocabulary (RDF)</a>		<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a>	19	2011-02-14

- ② 語彙定義名をクリックすることで、該当する語彙定義参照画面を表示する。  
画面上部の各出力ボタンをクリックすることで、レジストりに登録されている語彙定義を複数フ

フォーマットで出力する。

ユーザの語彙定義の利用環境に応じて、RDF/XML、Turtle、TopicMaps形式の複数フォーマットに対応。

#### ◆語彙定義参照画面



#### 〈キーワード検索〉

- ① キーワード検索画面にて、語彙定義名や語彙定義の説明をキーワードで検索できるように対応。

入力したキーワードで、語彙定義レジストリのメタデータを部分一致で取得する検索処理を実行し、キーワード検索結果を画面に表示する。

#### ◆キーワード検索画面



#### ◆キーワード検索結果画面



- ② 語彙定義名をクリックすることで、〈語彙定義検索一覧〉時と同様に該当する語彙定義参照画面を表示し、語彙定義を複数フォーマットで出力する。

### 11.3.4 記述規則検索およびファイル出力機能

#### (1) 概要

本機能は、レジストりに登録されているメタデータの記述規則を登録組織や利用語彙定義、キーワードなどの検索条件を指定し、RDF/XML、Trurtle、TopicMaps形式で出力できるユーザーインターフェースを提供する。ファイル登録したものについては、オリジナルのファイルとして、出力することができるように対応。

#### (2) 処理概要



#### 〈記述規則検索一覧〉

- ① 記述規則検索一覧画面にて、ユーザが目的の記述規則を検索しやすいように出力したい組織、利用している語彙定義から検索できるように対応。

入力した内容で、レジストリの検索処理を実行し、検索結果を画面に表示する。

#### ◆記述規則検索一覧画面

記述規則名	記述内容	登録組織	作成日
<a href="#">Application Profile</a>	国立国会図書館記述規則	国立国会図書館	2011-02-16

- ② 記述規則名をクリックすることで、該当する記述規則参照画面を表示する。

画面上部の各出力ボタンをクリックすることで、レジストりに登録されている記述規則を複数フォーマットで出力する。

ユーザの記述規則の利用環境に応じて、RDF/XML、Turtle、TopicMaps形式の複数フォーマットに対応。

#### ◆記述規則参照画面



〈キーワード検索〉

- ① キーワード検索画面にて、記述規則名や記述規則の説明をキーワードで検索できるように対応。

入力したキーワードで、記述規則レジストリのメタデータを部分一致で取得する検索処理を実行し、キーワード検索結果を画面に表示する。

#### ◆キーワード検索画面

#### ◆キーワード検索結果画面

記述規則名	記述内容	登録組織	作成日
<a href="#">ミュージアム資料情報構造化モデルのRDF</a>		東京国立博物館	2011-02-17

- ② 記述規則名をクリックすることで、〈記述規則検索一覧〉時と同様に該当する記述規則参照画面を表示し、記述規則を複数フォーマットで出力する。

### 11.3.5 メタデータインスタンスの変換機能

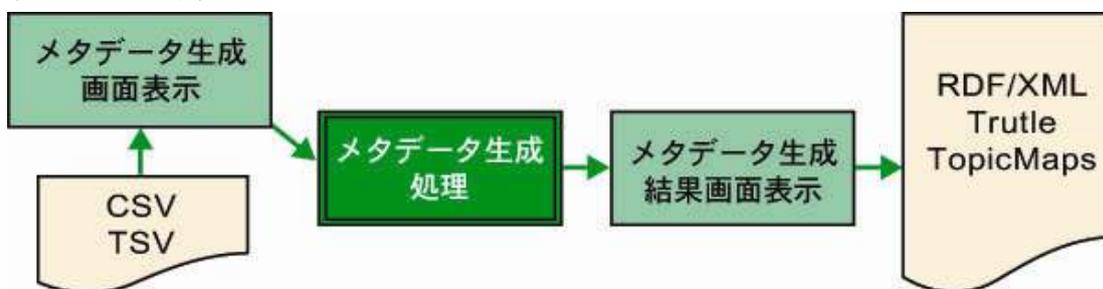
#### (1) 概要

本機能は、レジストリに登録されているメタデータの記述規則から、CSV、TSV形式で記述されたメタデータインスタンスをRDF/XML、Trurtle、TopicMaps形式で変換して出力する機能を提供する。

また、RDF/XML、Trurtle形式のメタデータインスタンスをダム・ダウン(DCマッピング)したRDF/XML、Trurtle、TopicMaps形式で出力する機能を提供する。

#### (2) 処理概要

〈メタデータ生成〉



- ① 記述規則のメタデータインスタンスを生成したいCSVファイルまたはTSVファイルをユーザが用意する。
- ② 生成したい記述規則のメタデータ生成画面にて、ユーザの環境にあった入力方法で取込を行えるように「ファイルから生成」、「テキストから生成」に対応。

## ◆メタデータ生成画面

記述規則	
名前空間URI	http://webarchives.tnm.jp/docs/informatics/smaoi-rdf-expression
バージョン	http://webarchives.tnm.jp/docs/informatics/smaoi-rdf-expression?20110217
バージョン情報 (reg:version)	1.0
タイトル (reg:title)	ミュージアム資料情報構造化モデルのRDF
コメント (reg:comment)	
作成者 (reg:creator)	東京国立博物館
作成日 (reg:created)	2011-02-17

- ③ メタデータインスタンスファイルを読み込み、記述規則用のXMLスキーマにより、メタデータインスタンスが記述規則に基づいて正しく記述されているかのチェックを行う。エラーが発生した場合は、メタデータ生成結果にどのような部分でエラーが発生したかをユーザに分かるように示す。

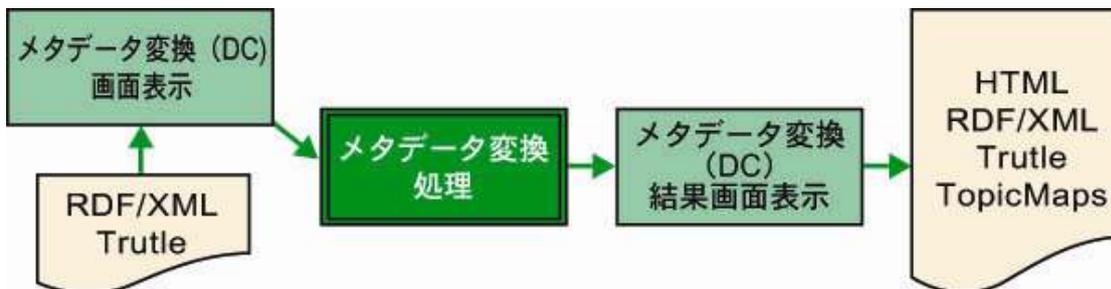
メタデータインスタンスの入力ファイルが正しく入力されていた場合、指定された出力形式でファイルを出し、ダウンロードボタンでダウンロードできる。

## ◆メタデータ生成結果画面

確認結果

1行目: 書誌ID:b01234: XMLスキーマエラー内容を表示します。  
3行目: 書誌ID:b01234: XMLスキーマエラー内容を表示します。

## 〈メタデータ変換(DC)〉



- ① 記述規則のメタデータインスタンスをDC変換したいRDF/XMLまたはTrurtle形式のファイルをユーザが用意する。
- ② 生成したい記述規則のメタデータ生成画面にて、ユーザの環境にあった入力方法で取込を行えるように「ファイルから変換」、「URLからの変換」、「テキストから変換」に対応。既にRDFファイルのメタデータインスタンスを公開しているユーザは、URL指定により、メタデータインスタンスをDC変換できるように対応した。DC変換により、各機関でのメタデータインスタンスを標準的に扱える形になる。

## ◆メタデータ変換(DC)画面

ようこそ、インフォコム さん ログアウト

**META BRIDGE**

メタデータ変換(DC)

戻る

記述規則

名前空間URI	http://webarchives.tnm.jp/docs/informatics/smoj-rdf-expression
バージョン	http://webarchives.tnm.jp/docs/informatics/smoj-rdf-expression?20110217
バージョン情報 (reg:version)	1.3
タイトル (reg:title)	ミュージアム資料情報構造化モデルのRDF
コメント (reg:comment)	
作成者 (reg:creator)	東京国立博物館
作成日 (reg:created)	2011-02-17

メタデータ変換(DC)

入力形式  RDF/XML形式  Turtle形式

出力形式  HTML形式(DC)  RDF/XML形式(DC)  Turtle形式(DC)  TopicMaps形式(DC)

メタデータ DC変換実行

参照...

ファイルから変換

URLから変換

テキストから変換

メタデータ変換

- ③ メタデータインスタンスのRDF/XMLまたは、Turtle形式ファイルを読み込み、記述規則用のX

MLスキーマにより、メタデータインスタンスが記述規則に基づいて正しく記述されているかのチェックを行う。エラーが発生した場合は、メタデータ変換結果にどのような部分でエラーが発生したかをユーザに分かるように示す。

メタデータインスタンスの入力ファイルが正しく入力されていた場合、指定された出力形式でDC変換を行い、変換したファイル出力し、ダウンロードボタンでダウンロードできる。

#### ◆メタデータ変換結果(DC)画面



## 11.4 API 設計

メタデータ・スキーマ・レジストリが保有する情報に対して、外部のシステムが利用できるAPIに関する検討内容を以下に報告する。

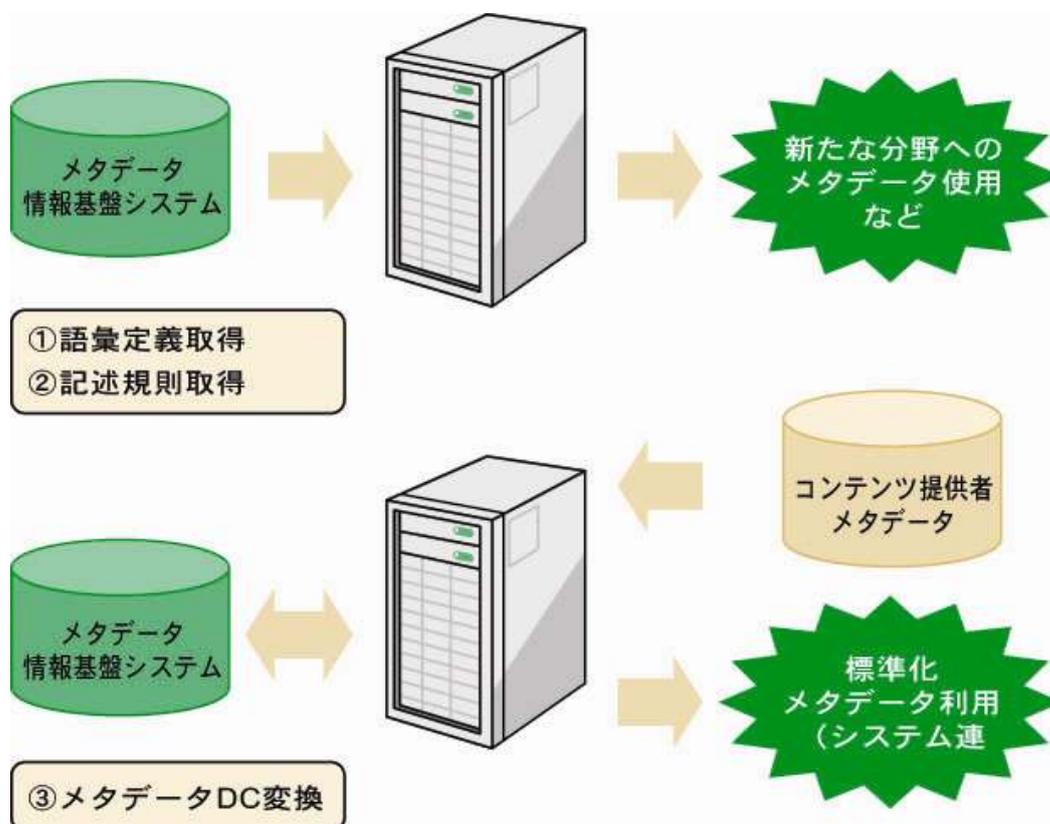
### (1) 概要

メタデータ・スキーマ・レジストリでは、システムが保有する語彙や記述規則情報に対して、外部プログラムからAPI経由でレジストリを利用し、自動化サービスに結びつけることがシステムを検討する。

本システムでは、以下のAPIの機能を提供する。

- ① 語彙定義のRDFを、グラフURIを指定することで取得可能なAPIを提供する。
- ② 記述規則のRDFを、グラフURIを指定することで取得可能なAPIを提供する。
- ③ 記述規則のグラフURIと変換対象のメタデータURIを指定することであらかじめ登録されたシングルDCとのマッピングで変換した形のメタデータを返却可能なAPIを提供する。

### ◆概要図



## (2) API 詳細

## ① 語彙定義取得API

語彙定義取得レジストリURIに対して、指定されたパラメータを加えることで、レジストリから、該当する語彙定義RDFを取得する。バージョンが指定された場合は、該当するバージョンの語彙定義RDFを取得する。指定されたパラメータのフォーマット形式のファイルで、語彙定義RDFを呼び出し元へ出力する。

エラーが発生した場合は、ステータスコードを返却し、エラーXMLを出力する。

指定方法は、

{語彙定義取得レジストリURI}?graph={グラフURI}&format={フォーマット}&version={日付}

## ◆ 語彙定義取得APIパラメータ

パラメータ名	説明
graph	必須 取得する語彙定義グラフ URI を指定
format	任意 (指定なしの場合は、RDF/XML) format=xml.....RDF/XML format=turtle.....Turtle format=topic.....TopicMaps format が ない 場合 .....RDF/XML
version	任意 (指定なしの場合は、最新の語彙定義) format=turtle.....Turtle version=20101011 version が ない 場合

## ◆ エラー時

内容	説明
ステータスコード	404.....Not Found. 指定されたリソースが見つからない場合に返却。 500.....その他のエラー。内部的な問題によってデータを返すことができない場合に返されます。
エラーXML	<?xml version="1.0" encoding="utf-8"?> <Error> <Message>error message</Message> </Error>

## ②記述規則取得API

記述規則取得レジストリURIに対して、指定されたパラメータを加えることで、レジストリから、該当する記述規則RDFを取得する。バージョンが指定された場合は、該当するバージョンの記述規則RDFを取得する。指定されたパラメータのフォーマット形式のファイルで、記述規則RDFを呼び出し元へ出力する。

エラーが発生した場合は、ステータスコードを返却し、エラーXMLを出力する。

指定方法は、

{記述規則取得レジストリ URI}?graph={グラフ URI}&format={フォーマット}&version={日付}

## ◆記述規則取得APIパラメータ

パラメータ名	説明
graph	必須 取得する記述規則グラフURIを指定
format	任意(指定なしの場合は、RDF/XML) format=xml.....RDF/XML format=turtle.....Turtle format=topic.....TopicMaps formatがない場合.....RDF/XML
version	任意(指定なしの場合は、最新の記述規則) version=20101011 versionがない場合

## ◆エラー時

内容	説明
ステータスコード	404.....Not Found. 指定されたリソースが見つからない場合に返却。 500.....その他のエラー。内部的な問題によってデータを返すことができない場合に返されます。
エラーXML	<?xml version="1.0" encoding="utf-8"?> <Error> <Message>error message</Message> </Error>

## ③メタデータDC変換API

メタデータDC変換レジストリURIに対して、記述規則のグラフURIと変換対象のメタデータURIをパラメータ指定することで、レジストリを利用し、該当記述規則のシンプルDCのマッピングで、対象メタデータをDC変換する。

変換対象メタデータは、指定された記述規則用のXMLスキーマによって、整合性のチェックを行う。整合性でエラーになった場合、エラーXMLにエラーメッセージを返却する。エラーが存在しない場合は、記述規則のDCとの関連マッピングから、変換対象メタデータをDCにマッピングする。

指定方法は、

{メタデータDC変換レジストリURI}?graph={グラフURI}&source={対象メタデータURI}

## ◆メタデータDC変換API

パラメータ名	説明
graph	必須 記述規則レジストリURIを指定
source	必須 変換対象のメタデータURI(RDF/XML) 例) source=http://XXX.XXX.jp/XXX/source/00010.rdf

## ◆エラー時

内容	説明
ステータスコード	404・・・Not Found. 指定されたリソースが見つからない場合に返却。 500・・・その他のエラー。内部的な問題によってデータを返すことができない場合に返されます。
エラーXML	<?xml version="1.0" encoding="utf-8"?> <Error> <Message>error message</Message> </Error>

## ◆変換対象メタデータ(変換前)

```
例)
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:ndlbooks="http://xxx.xxx.jp/books/"
  xmlns:ndlsh="http:// id.ndl.go.jp /auth/ndlsh/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<foaf:Document rdf:about="http://xxx.xxxx.jp/books/b01234">
  <dcterms:creator>〇〇出版</dcterms:creator>
  <dcterms:issued>2000-10-20</dcterms:issued>
  <dcterms:subject rdf:resource="http://id.ndl.go.jp/auth/ndlsh/科学"/>
  <dcterms:title>地球の神秘</dcterms:title>
</foaf:Document>
</rdf:RDF>
```

## ◆変換対象メタデータ(変換後)

```
例)
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:dc="http://purl.org/dc/terms/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:ndlbooks="http://xxx.xxx.jp/books/"
  xmlns:ndlsh="http:// id.ndl.go.jp /auth/ndlsh/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<foaf:Document rdf:about="http://xxx.xxxx.jp/books/b01234">
  <dc:creator>〇〇出版</dc:creator>
  <dc:date>2000-10-20</dc:date>
  <dc:subject rdf:resource="http://id.ndl.go.jp/auth/ndlsh/科学"/>
  <dc:title>地球の神秘</dc:title>
</foaf:Document>
</rdf:RDF>
```

## 12. レジストリの開発作業

### (1) 開発対象

レジストリの開発対象は、以下の2つを対象とした。

- 語彙定義(クラス、プロパティなど)
- 記述規則(使用プロパティ、値制約などの記述規則)

レジストリについての実証実験のため、あらかじめ利用するデータの準備、登録を行う必要があり、既に語彙定義・記述規則でのデータ管理が行われている国立国会図書館やAGRSなどの語彙定義・記述規則を利用した。

本開発を進めるに当たり、データ登録機能に関する開発を第1フェーズ、第1フェーズにて登録したデータの検索・出力機能に関する開発を第2フェーズ、データ変換機能に関する開発を第3フェーズとして行った。

フェーズリリース方式については、確定箇所から開発を行うことで、要求仕様と差異が発生することを防ぎ、またリリースを行う主機能単位での動作を確認いただくことで、システムに対する解釈の食い違いを早期に発見する目的で導入を行った。

また、本システムは、レジストリに登録できるユーザを管理する必要があり、公開サイトでもあるため、ユーザ管理機能があるInfoLibを利用している。

### (2) スケジュール

2010年11月～2011年3月の間、以下の開発を行った。

- ① 1フェーズの開発 … 2/15リリース  
語彙定義・記述規則についてファイルによる登録、更新
- ② 第2フェーズの開発 … 2/25リリース  
語彙定義・記述規則のキーワード検索、及びRDF/XML、Turtle、TopicMaps形式の出力。記述規則の画面による追加、更新。
- ③ 第3フェーズの開発 … 3/15から徐々にリリース  
メタデータインスタンスの変換

### ◆スケジュール表

タスク	作業開始日	作業終了日	ステータス	11月28日	12月12日	12月26日	1月9日	1月23日	2月6日	2月20日	3月5日	3月19日
基本設計	11月29日	12月10日	完了	■								
詳細設計	12月13日	12月28日	完了		■	■	■					
PHASE1												
開発	12月20日	1月31日	完了		■	■	■	■				
テスト	2月1日	2月13日	-					■				
リリース準備	2月14日		-									
リリース	2月15日		-							■		
PHASE2												
開発	1月24日	2月14日	20%					■				
テスト	2月15日	2月23日	-							■		
リリース準備	2月24日		-									
リリース	2月25日		-							■		
PHASE3												
開発	2月15日	2月28日	-							■		
テスト	3月1日	3月9日	-								■	
リリース準備	3月10日		-									
リリース	3月11日		-									■

凡例: ■ 予定 ■ 実績

### ◆フェーズ別リリース機能一覧

	PHASE1(2月15日)	PHASE2(2月25日)
実施実験範囲	<b>語彙</b> <ul style="list-style-type: none"> <li>■ファイルによる登録、更新                             <ul style="list-style-type: none"> <li>・RDF形式ファイルもしくはURLからの取り込み</li> </ul> </li> <li>■バージョン管理                             <ul style="list-style-type: none"> <li>・過去バージョンの参照</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>■キーワード検索                             <ul style="list-style-type: none"> <li>・プロパティやメタデータなどによる検索</li> </ul> </li> <li>■RDF、TopicMaps出力                             <ul style="list-style-type: none"> <li>・別形式のRDFやTopicMapsへ変換して出力</li> </ul> </li> </ul>
	<b>記述規則</b> <ul style="list-style-type: none"> <li>■ファイルによる登録、更新                             <ul style="list-style-type: none"> <li>・本プロジェクトで検討された形式のファイル、RDF形式ファイルの取り込み</li> </ul> </li> <li>■バージョン管理                             <ul style="list-style-type: none"> <li>・過去バージョンの参照</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>■画面による追加、更新                             <ul style="list-style-type: none"> <li>・簡易記述規則の作成支援</li> </ul> </li> <li>■キーワード検索                             <ul style="list-style-type: none"> <li>・プロパティやメタデータなどによる検索</li> </ul> </li> <li>■RDF、TopicMaps出力                             <ul style="list-style-type: none"> <li>・別形式のRDFやTopicMapsへ変換して出力</li> </ul> </li> </ul>
実施実験対象外	PHASE3(3月11日)	
	<b>将来拡張予定機能</b> <ul style="list-style-type: none"> <li>■メタデータインスタンスの変換                             <ul style="list-style-type: none"> <li>・CSVファイルを取り込み、RDF形式ファイル、TopicMapsで出力</li> <li>・語彙を統一化するためのdumb down変換出力</li> </ul> </li> <li>■メタデータインスタンスの変換                             <ul style="list-style-type: none"> <li>・メタデータインスタンス変換のチェック強化</li> <li>・RDF形式ファイルからTopicMapsへ変換</li> </ul> </li> <li>■コンピュータによる問い合わせ(API提供)                             <ul style="list-style-type: none"> <li>・語彙検索</li> <li>・記述規則検索</li> </ul> </li> </ul>	

#### 12.1 第1フェーズの開発

第1フェーズでは語彙定義・記述規則についてファイルによる登録、更新を行い、次に挙げる項

目を開発対象とした。

(1) 語彙定義

画面名称	概要
語彙定義ファイル登録	RDF/XML、Turtle形式の語彙定義ファイルを本システムにアップロードし、登録、更新する。
語彙定義登録エラー	語彙定義ファイル登録でアップロードしたデータで、エラーが発生した場合にエラーメッセージを表示する。
語彙定義登録確認	本システムで利用する語彙定義を登録、更新前に確認する。
語彙定義完了	語彙定義ファイルの登録、更新時に登録完了のメッセージを表示する。
ターム詳細表示	指定したURIのターム詳細を表示する。
バージョン一覧	名前空間(URI)のバージョン一覧を表示する。
登録済語彙一覧	本システムに登録されている語彙定義を一覧で表示する。
登録済語彙一覧	本システムに登録されている語彙定義を一覧で表示する。
語彙定義参照	本システムで利用する語彙定義を参照表示する。

(2) 記述規則

画面名称	概要
記述規則ファイル登録	タブ区切テキスト形式、RDF/XML、Turtle形式の記述規則ファイルを本システムにアップロードし、登録、更新する。
記述規則エラー	記述規則ファイル登録でアップロードしたデータで、エラーが発生した場合にエラーメッセージを表示する。
記述規則編集	本システムで利用するメタデータの記述規則を登録、更新する。
バージョン一覧	名前空間(URI)のバージョン一覧を表示する。
記述規則完了	記述規則ファイルの登録、更新時に登録完了のメッセージを表示する。
記述規則一覧	登録済みの記述規則を一覧で表示する。
記述規則参照	当システムで利用する記述規則を参照表示する。
記述規則参照(構造化)	記述規則の構造化項目の記述制約(Statement Template)を参照する。
記述制約参照	記述制約(Statement Template)の詳細を参照表示する。

## 12.2 第2フェーズの開発

第2フェーズでは、語彙定義・記述規則のキーワード検索、及びRDF/XML、Turtle、TopicMaps形式の出力と記述規則について画面による追加、更新を行い、次に挙げる項目を開発対象とした。

### (1) 語彙

画面名称	概要
登録済語彙一覧	第1フェーズにて作成した当機能へ削除機能を追加する。
語彙定義検索一覧	検索条件のクラス/プロパティを利用している語彙定義の一覧を表示する。
ターム詳細表示	指定されたURIのターム詳細を表示する。
RDF/XML出力、Turtle出力、TopicMaps出力	語彙定義を各ファイル形式のファイルにて出力する。
キーワード検索	登録されている語彙定義から「キーワード」に該当するラベル、コメント、記述内容のテキスト項目を検索する。
キーワード検索結果	語彙定義のキーワード検索の検索結果を表示する。

### (2) 記述規則

画面名称	概要
記述規則編集	第1フェーズで作成した当機能にメタデータ以外の記述規則を登録、更新機能を追加する。
記述制約追加編集	記述規則の項目の制約を追加編集する。
記述制約検索	プロパティ名から登録されている記述規則を検索し、該当する記述制約を一覧で表示する。
記述規則(構造化)追加編集	記述規則の項目の制約を追加編集する。
記述制約検索	プロパティ名から登録されている記述規則を検索し、該当する記述制約を一覧で表示する。
記述規則(構造化)追加編集	記述規則の構造化項目の制約を追加編集する。
参照設定	記述規則の参照設定を行う。
未登録語彙登録	記述規則で「未登録プロパティ」を設定した場合に語彙定義の新規登録、更新を行う。
ターム検索	語彙定義に登録されているタームを検索し、検索条件に該当する一覧を表示する。

記述規則一覧	第1フェーズで作成した当機能に削除、新規作成、流用機能を追加する。
記述規則名前空間設定	記述規則URIを入力し、設定する。
記述規則検索一覧	登録されている記述規則を一覧で表示する。

画面名称	概要
記述制約検索	プロパティ名で登録されている記述規則を検索し、該当する記述制約を一覧で表示する。
RDF/XML出力、Turtle出力、TopicMaps出力	記述規則を各ファイル形式のファイルにて出力する。

### 12.3 第3フェーズの開発

第3フェーズではメタデータインスタンスの変換を行い、次に挙げる項目を開発対象とした。

#### (1)メタデータインスタンス変換

画面名称	概要
メタデータRDF生成	ファイルアップロードしたCSV形式のメタデータインスタンスから、メタデータ生成形式で選択された形式で、メタデータインスタンスを出力する。
メタデータ変換	該当の記述規則の入力形式のメタデータインスタンス(RDF/XML,Turtle)をシンプルDC形式のメタデータインスタンス(RDF/XML,TurtleTopicMaps)へ変換し出力する。

## 12.4 開発上の課題と解決

### (1) 語彙や記述規則の収集

インターネットを用いて既に公開されている語彙、記述規則について調査を行った。

図書館、博物館など、様々な機関でいろいろな形でメタデータを公開しているが、実データのみ  
のケース、深い階層に資料があるケース、資料として纏まっていないケースなどがあり、本システム  
で想定した形の語彙、記述規則の適応可否について、判断を行う必要があった。また、自団体の  
情報として、自サイトにて公開を行うに留まっており、積極的な情報資源の相互運用のためには、  
情報を共有化する必要性を感じた。

### (2) Jena、RDF/OWLの知識向上

Jena は、セマンティックWebアプリケーション開発のためのフレームワークであり、今回のシステム  
を構築にあたり、絶対不可欠な知識であった。また、RDF/OWLについても、仕様を実現させる  
ための知識として絶対不可欠であり、本システムの構築にあたり最大の課題であった。

Jenaの資料は、インターネット上の情報も数少なく、英語サイトを翻訳したり、サンプルプログラム  
を作成したりするなどをして、理解に努めた。

- ・ モデル構造の理解(リソース、プロパティ、リテラル、ステートメントなど)
- ・ 推論による検索(SPARQL)など

また、RDF/OWLについても、インターネット上の情報も数少ない専門書を  
参考資料として、用語集などを作り知識向上に努めた。

### (3) 技術WG定例会の実施

本システムの構築にあたり、設計上の課題や疑問点など、毎週、技術WG定例会を実施し、有  
識者を交えて検討を行った。

開発期間が短期であるのと、難しい点が多いので課題も多かったが、毎週行うことで、有識者  
により回答を得ることができた。また、仕様の再確認も行うことができた。以下にいくつかの検討課  
題を示す。

#### (4)簡易DSPの取込

本システムにおいて、記述規則に関しては、ユーザが記述しやすいように簡易DSP(タブ区切テキスト形式)に対応する設計とした。簡易DSPは、今回のシステムに合わせて、作成した記述方法であるため、取込むためにいくつかの課題があり、検討を重ねた。

##### ① フォーマット形式

CSV形式、TSV形式といった一般的なフォーマットでないため、独自に読み込む形とした。そのため、Javaでパッケージ化されたCSV形式、TSV形式の読み込みモジュール等を利用することができなくなった。

ただ、今回、開発当初は工数が発生したが、独自で読み込む形にしていたため、読み込みルールの仕様変更等に臨機応変に対応できた。

#### ◆簡易DSP記述例

```
[@NS]
dcndl http://ndl.go.jp/dcndl/terms/
ndlsh http://id.ndl.go.jp/auth/ndlsh/
bsh http://id.ndl.go.jp/auth/bsh/
ndlbooks http://iss.ndl.go.jp/books/
@base http://ndl.go.jp/dcndl/dsp/biblio

[MAIN]
書誌 ID foaf:Document 1 1 ID ndlbiblio: 文書の ID
タイトル dcterms:title 1 1 構造化 #構造化タイトル 文書の表題
著者 dcterms:creator 0 1 構造化 foaf:Agent 文書の作者
発行日 dcterms:issued 1 1 文字列 dcterms:W3CDTF 文書の発行日
主題 dcterms:subject 0 - 参照値 ndlsh: bsh: 文書の主題

[構造化タイトル]
リテラル値 xl:literalForm 1 1 文字列 タイトル自身
読み dcndl:transcription 0 1 文字列 タイトルの読み
```

例)

- ・ 簡易DSPでネームスペースもなく、構造化もない場合は、[MAIN]記述がなくても、取り込める。
- ・ [@NS]がない場合は、デフォルトで標準語彙を付加する

##### ② 独自語彙の定義

プロパティに接頭辞がない場合、簡易DSPファイル取込後、記述規則編集画面にて、独自語

彙定義として、登録できることとした。

本仕様を取込むために、簡易DSPからRDFの形にする際、接頭辞がないプロパティは、未登録語彙のURIを仮設定し、Jenaで読み込み記述規則編集画面に表示するようにした。そこで、未登録語彙がある場合は、新たな語彙として、登録できる仕組みを提供することで対応した。

#### ◆簡易DSP記述例

```
...  
[MAIN]  
書誌 ID foaf:Document 1 1 ID ndlbiblio: 文書の ID  
タイトル title 1 1 構造化 rdfs:label 文書の表題  
著者 creator 0 1 構造化 foaf:Agent 文書の作者  
発行日 issued 1 1 文字列 dcterms:W3CDTF 文書の発行日  
主題 subject 0 - 参照値 ndlsh:bsh: 文書の主題  
...
```

#### (5) 文字コード問題

本システムにおいてはサーバOSとしてLinuxを採用しているが、利用者の主要なクライアントOSはWindowsであることを想定している。異なるプラットフォーム間の文字データ交換においては「文字化け」が問題となる。

#### 【発生した課題】

本システムの開発に当たっては高水準のユーザビリティを目標としており、データ登録においても一括登録機能を提供することを検討した。

国立公文書館の公開データなどの事例では、語彙、記述規則は表形式で管理されており、そのデータ作成には表計算ソフトを使用していることが予想される。Windows OSではMicrosoft Excel(以下「Excel」)が圧倒的なシェアを誇っていることから、Excelで編集が行えることを要件とした。

そこで問題になるのが本システムにアップロードするデータフォーマットである。

#### (A) Excelブック形式(.xlsx)

課題：旧バージョンのExcelユーザが利用できない。

#### (B) Excel 97-2003 ブック形式(.xls)

課題：多くのExcelユーザが利用できるが、アプリケーション依存、バージョン問題が残る。

#### (C) OpenDocument形式(.ods)

課題：まだ一般的ではなく、また本システム側での解析処理の実装が容易でない。

#### (D) テキスト形式(CSV、TSV)

課題：ExcelではShiftJISエンコーディングのテキストのみ出力でき、UTF-8をはじめとするUnicode系エンコーディングに対応していない。

サーバ側(Linux+Java)でShiftJISコードを読み込むと「文字化け」が発生し得る。

ユーザ側で別途UTF-8変換するにはある程度の技術的知識が必要であり、ユーザビリティが犠牲になる。

#### 【背景】

「文字化け」とは、コンピュータ上で取り扱われる文字データが異なる環境で別の字形(グリフ)で表示される現象を指す。異なるプラットフォーム間のデータ交換で発生する問題に限らず、同じコンピュータ内でのソフトウェア/フォントの相違により発生する現象も含まれる。この「文字化け」に代表される文字コード問題は、近年では「文字符号化方式」と「文字集合」の概念で議論されている。

「文字化け」は二つの問題に分けることができる。

#### ・字形の相違

「文字集合」の相違。さらに文字集合規格に起因する相違と、活字／フォントのデザインの相違に分けることができる。

文字の異同の識別は主観的な事柄であり、全ての人が別の文字として認識するものもあれば、個別の事情によりごく限られた人が問題にするレベルの相違もある。例えば、ASCII系文字集合の5C(16進)はASCIIでは「¥」(バックスラッシュ)JIS(X 0201)では「¥」(円記号)であり、いかなるフォントを用いても同一字形と見なされることはないと思われる。他方、「吉」と「𠄎」は同じ文字と見なされることがあるが、氏名を扱う戸籍業務では別の文字として扱われることがある。「未」と「末」は日本人にとっては別の文字だが、外国人には区別が難しいであろう。「勻」(フォント:HG正楷書体-PRO)と「勻」(フォント:HG明朝B)は同じコードの文字であるが、運筆的には異なっている。

#### ・文字データ(文字コード)の相違

「文字符号化方式」(エンコーディング)の相違と言い換えることができる。

エンコーディングが異なっても1対1のコード変換が可能ならば解決できる問題である。問題となるのが、JIS、Unicodeといった異なった規格間のコード変換に非可逆なパターンがあり、「文字コード化け」が発生するケースである。

Linux(およびJava)とWindows間の「文字化け」は、Windowsで採用されている文字集合「マイクロソフト標準キャラクターセット」に起因する。この独自文字集合の歴史はMS-DOSの日本語化における符号化方式としてShiftJISが採用されたことから始まる。同OSの複数のOEM版では文字集合が独自定義され、Windows3.1日本語版でそれらを統合した文字集合が定義された。この文字集合の相違が考慮されていないシステムでは、特に拡張文字について文字化けが発生する。

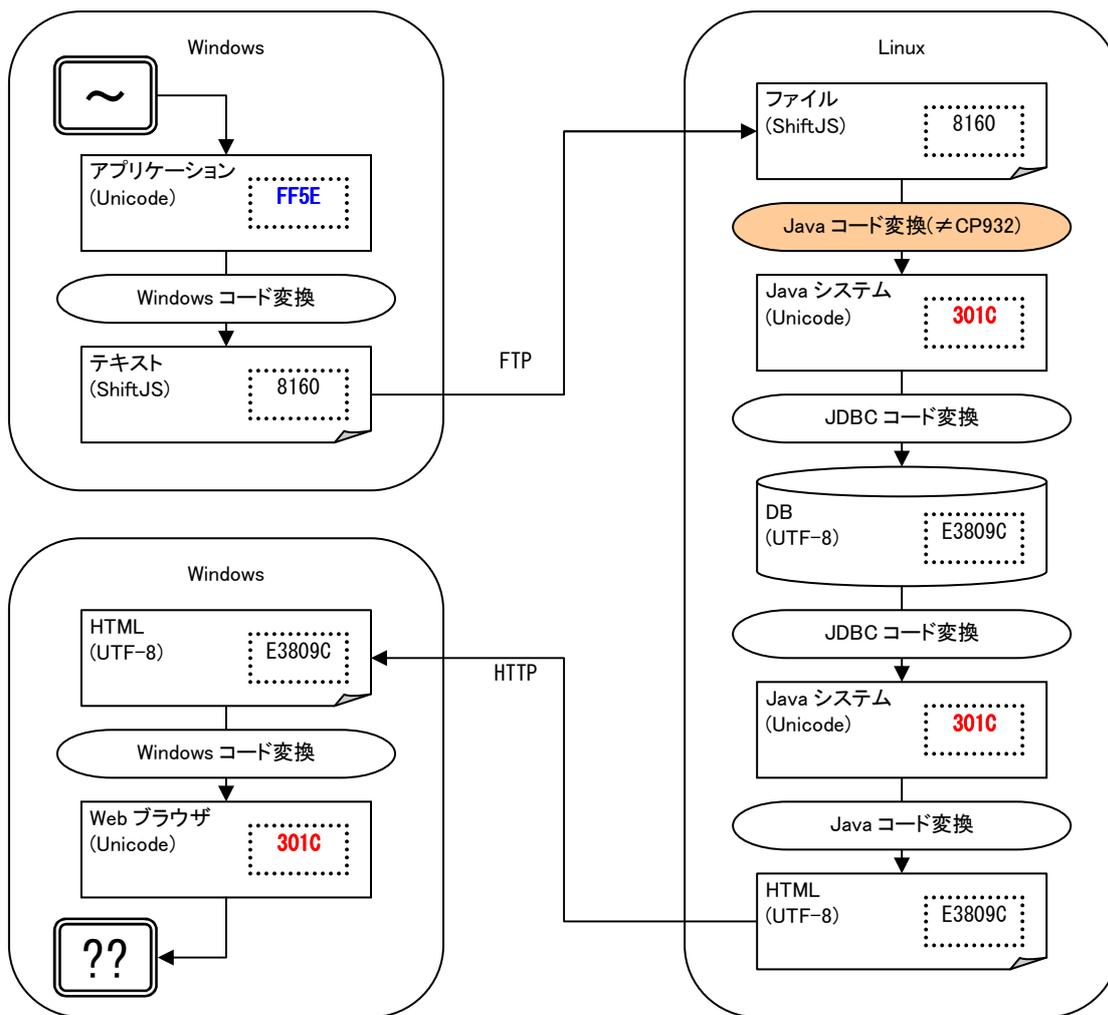
更にエンコーディングについても問題を抱えている。その後WindowsNTにおけるOS内部コードのUnicode化でShiftJISとUnicode間のコード変換テーブルが実装されたが、その対応付けには一部不具合があった。発覚したのは実装後数年を経た後であり、互換性維持のため「仕様」として今日まで残っている。特に有名なのが「~」(全角チルダ)であり、文字化けの試験で多く使用されている。

Windows、Linux、Javaはいずれも内部エンコーディングはUnicode系であり、データ交換時の符号化方式としてUTF-8など—コード変換方式としてコード値の演算を行うもの—を採用する限り、文字列データ交換によるデータ損失は発生しない。非Unicode系のエンコーディング—コード変換を変換テーブルによって行うもの—でデータを交換した場合に、それぞれで異なった変換テーブルを用いてUnicode化することにより「文字コード化け」が発生する。後発のLinuxおよびJavaは相互運用性を持たせるために「コードページ932」としてWindows環境のマッピングテーブルを再現しているが完全ではない。またLinux+Java環境にはその柔軟性故にエンコーディングの設定箇所が数多く設けられており、データ交換の際、どの箇所でのどのエンコーディングを選択するかを十分に

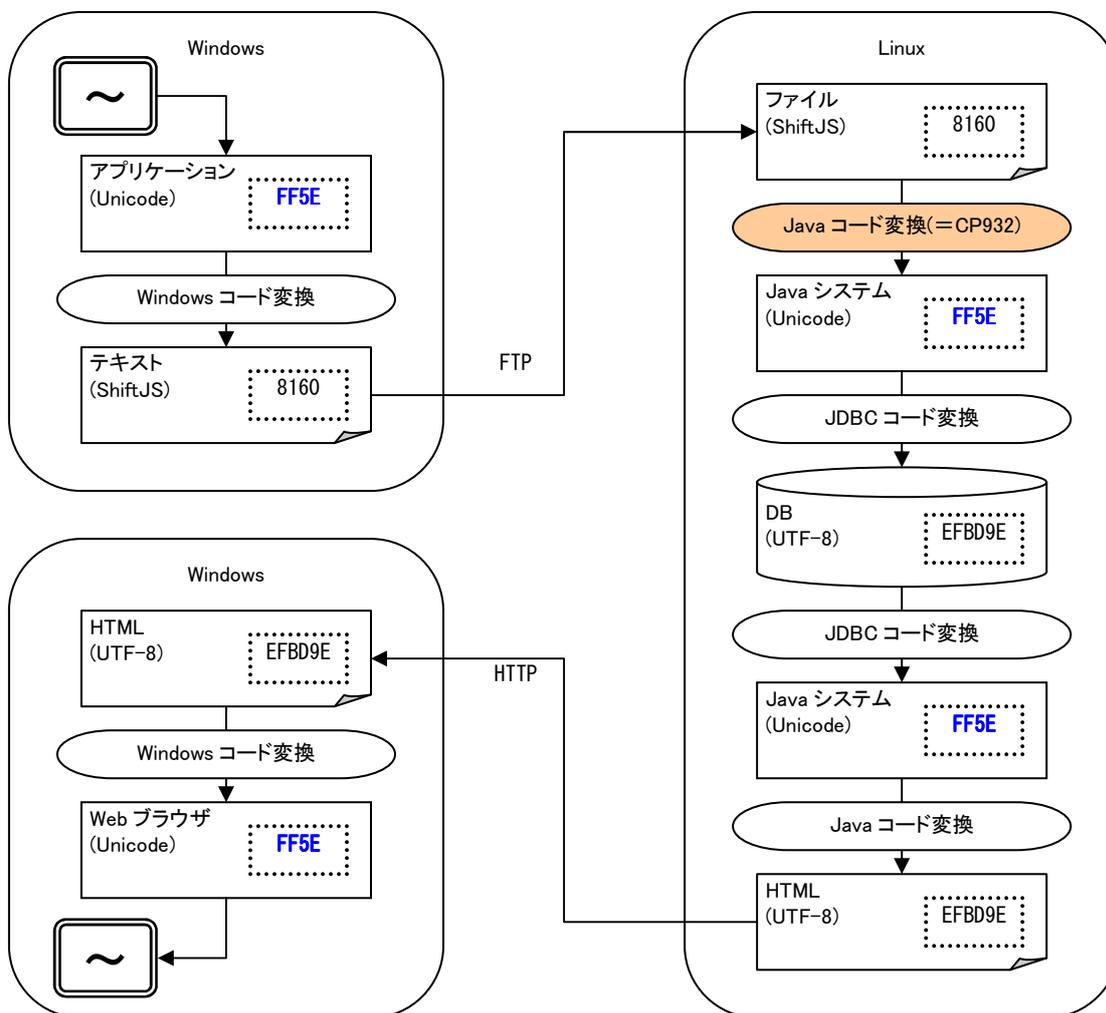
検討しなければならない。

本システムで扱うのはメタデータであり、細かい字形の相違については問題にならないが、クライアント・サーバ間の文字データ交換でビットデータが変わることは重大な問題となるため、その点に留意して仕様を検討した。

(図) 文字化けの発生例



(図)文字化けへの対応後



## 【解決方法】

ユーザに2つの手段を提供した。

## ◆テキスト形式ファイルのアップロード機能

テキスト形式ファイルに対するエンコーディング自動判定機能を付加した。

UTF-8と判定できた場合は正常に読み込まれる。

それ以外のエンコーディング(ShiftJIS/EUC-JP)は厳密には識別できないが、EUC-JPが使用されるケースが少なくなってきたことからShiftJIS(コードページ932)と仮定して読み込む。この場合はユーザに文字コード化けの発生に関する注意喚起メッセージを表示し確認を求める。

## ◆テキストエリアコントロールによるPOST機能

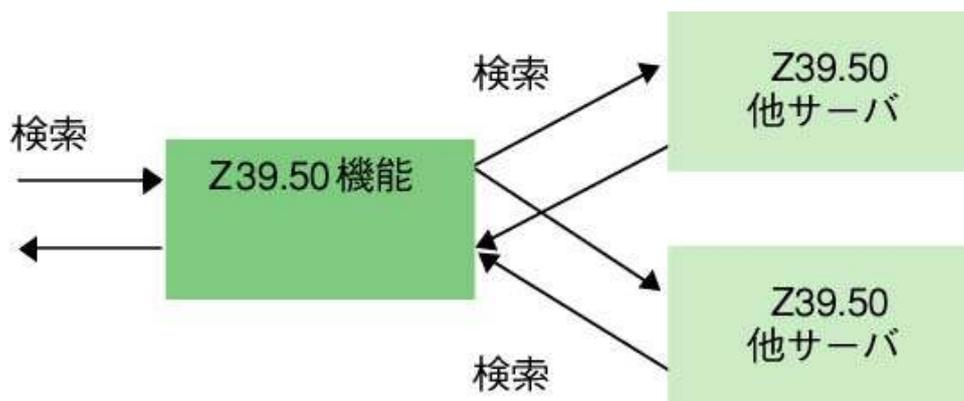
UTF-8エンコーディングのHTMLページにテキストエリアコントロールを設けた。

このページからHTTP - POST方式でサーバに送信した場合、文字コード化けは発生しない。

## 12.5 今後の展望

### (1) 他システムとの連携

本システムと他システムとの相互運用性を目的として、各種プロトコルZ39.50,SRU,SRWでの連携なども検討している。利用者の検索要求により、複数のZ39.50サーバを検索し、検索結果を統合して返却し、利用者は複数のデータベースを一つのデータベースとして検索可能である。



### (2) 実証実験結果のシステム反映。

本システムを実証実験により、様々な機関のユーザに利用していただくことにより、得られた結果をもとに、更なる情報利用を可能とするための必要な機能やユーザインターフェース、APIの提供等を行っていく。

## 13. サービス(レジストリ)の実証

### 13.1 実証実験の立案

#### 13.1.1 実証実験の方針

すでに述べてきた通り、メタデータを取り巻く最大の課題は、記述規則の標準化等が各分野単位で行われており、複数分野間での連携や相互運用の環境が整っていないことが挙げられる。記述規則や記述形式に関する情報が十分に公開されていないこと、標準規格に基づく開発という意識が浸透していないこと、そして公開形式も文書形式であることが多いためコンピュータ自身が直接的に利用するには向いていないこと等が課題であるが、メタデータ規則の共有化を進めるには、まずメタデータ規則のネット上での公開が必要となる。

特に、メタデータの規則を機械処理可能な形式で表し、蓄積・提供することは、インターネット上での共有性・再利用性の向上を促す上では非常に重要と言える。そこで本事業では、WWW の標準規格や Dublin Core Metadata Initiative(DCMI)のメタデータ相互運用性向上のためのフレームワークを基盤として、メタデータスキーマレジストリを核とするメタデータ情報基盤の構築を推進し、メタデータ、ひいてはコンテンツの流通性向上に資することを目的としている。

#### 13.1.2 実証実験の内容と実証者の選定

本プロジェクトの実証においては、スキーマ・レジストリに求められる「登録」「検索・作成」「変換・再利用」の3つの観点から実験を行なう。登録および検索・作成はコンテンツ提供者の立場からの実証、変換・再利用はコンテンツ利用者の視点による実証という位置づけとなる。

なお、コンテンツの提供側と利用側、各々の実証者は以下の通りに選定した。

コンテンツ提供者	関係者	筑波大学 知的コミュニティ基盤研究センター
コンテンツ提供者	関係者	国立国会図書館
コンテンツ提供者	関係者	東京国立博物館
コンテンツ提供者	関係者	東京国立近代美術館
コンテンツ提供者	関係者	京都大学 地域研究総合情報センター
コンテンツ提供者	関係者	個人※
コンテンツ提供者	一般モニター	岡山県立図書館
コンテンツ提供者	一般モニター	大日本印刷株式会社
コンテンツ提供者	一般モニター	凸版印刷株式会社
コンテンツ利用者	関係者	合資会社ゼノン
コンテンツ利用者	関係者	株式会社ナレッジ・シナジー
コンテンツ利用者	関係者	ロジカルウェブ株式会社
コンテンツ利用者	関係者	株式会社ジオ・プレーン

コンテンツ利用者	関係者	インフォコム株式会社
コンテンツ利用者	関係者	株式会社インフォコム西日本
コンテンツ利用者	一般モニター	大日本印刷株式会社
コンテンツ利用者	一般モニター	凸版印刷株式会社
コンテンツ利用者	一般モニター	農林水産省農林水産研究情報センター

※敬称略。

※一部の機関へはメタデータ情報基盤研究会を通じて実証実験を依頼し協力を受けた。

#### 実証回答機関及び人数

コンテンツ提供者	関係者	5機関	11名
		個人	1名
	モニター	3機関	3名
コンテンツ利用者	関係者	6機関	8名
	モニター	3機関	3名

※事業関係者に、個人という立場で協力頂いたケースもあった。

### 13.1.3 評価基準の設定

実証者には、以下のような評価基準を設定し、実証内容として依頼した。なお、以降、メタデータ・スキーマ・レジストリの名称を、「META BRIDGE」とする。

#### (1) 実証 1: スキーマの登録に関する実証

既存の記述規則を標準的な形で表現・公開する際の操作性等についての評価を依頼する。

##### <実証者に依頼する作業説明概要>

META BRIDGE では、RDF で記述された語彙定義、記述規則定義を登録できるほか、一定の書式で書かれた表形式の簡易記述規則を登録する機能も提供する。そこで、次の(あるいは別紙の)書式とサンプルを参考に、「簡易記述規則の作成」「簡易記述規則の登録」「記述規則メタデータの登録」「未登録語彙の登録」についての評価を依頼する。

##### <実証者に依頼する評価内容>

- ア. 表形式の簡易記述規則の作成はスムーズにできたか
- イ. 簡易記述による登録手順は分かりやすかったか
- ウ. 簡易記述規則登録の一環としての未登録語彙の登録は、分かりやすかったか
- エ. 記述規則登録についての改良すべき点はあるか

#### (2) 実証 2: スキーマの検索・作成に関する実証

### ① メタデータを記述するために利用可能な記述規則を探す機能についての評価依頼

#### <実証者に依頼する作業説明概要>

META BRIDGE では、登録済みのスキーマ(語彙、記述規則)を、キーワードや使用プロパティによって検索し、目的に応じたものを探す手段を提供する。そこで、次の手順を参考に、「語彙の検索」「記述規則の検索」「検索結果の詳細確認」を実施し、その評価を依頼する。

#### <実証者に依頼する評価内容>

- ア. キーワードからの検索で目的のスキーマを見つけることができたか
- イ. プロパティからの検索で目的の記述規則を見つけることができたか
- ウ. 検索結果の表示、スキーマの詳細表示は分かりやすかったか
- エ. 他の検索手段としてはどのようなものが求められるか

### ② 公開されているスキーマを参考に、独自のスキーマを定義する上での操作についての評価依頼

#### <実証者に依頼する作業説明概要>

META BRIDGE は、新たな記述規則を作成・登録することができる。この時、新たな規則を最初から定義することも、雛形とする登録済みの規則をコピーして編集することも可能である。そこで、次の手順を参考に、「記述規則の新規作成・登録」「雛形とする登録済み規則の検索」「登録済み規則をコピー・編集しての新規規則作成・登録」を実施し、その評価を依頼する。

#### <実証者に依頼する評価内容>

- ア. 記述規則の新規作成はスムーズにできたか
- イ. 登録済み規則のコピー・編集はスムーズにできたか
- ウ. 規則の作成・登録に、他にどのような機能が求められるか

### ③ 異なるメタデータ形式間での相互運用についての評価依頼

実証対象とする場合は、語彙の登録時点での DC マッピング、もしくは登録済み語彙の DC マッピングの編集についての評価を依頼する。

### (3) 実証 3: メタデータの変換と再利用に関する実証

複数のメタデータをマッシュアップする上での操作についての評価依頼

#### <実証者に依頼する作業説明概要>

META BRIDGE は、登録済みのスキーマ(語彙、記述規則)のプロパティを、シンプル・ダブル・コア語彙(シンプル DC)に対応付けるためのマッピングを用意している。これを用いることで、登録済みスキーマに従って記述されたメタデータをシンプル DC によるメタデータに変換することができる。そこで、異なるスキーマのメタデータをシンプル DC に変換したらマッシュアップ(統合)が可能になることを念頭に、次の手順を参考にして、「サンプルのメタデータ(元データ)をシンプル DC に変換(変換データ)」「元データと変換データの比較」「複数の元データを変換して統合」を実施し、その評価を依頼する。

#### <実証者に依頼する評価内容>

- ア. 元データ→変換データの結果は、予想と比べてどうだったか
- イ. 複数の元データを変換、統合した結果は、異なる情報源の組み合わせ利用が可能になるものだったか
- ウ. マッピングはシンプル DC で十分か、他にマッピングを加えるとすればどのような語彙が望ましいか
- エ. MetaBridge を利用したメタデータの活用法として、どのようなものが考えられるか

## 13.2 実証実験 1:スキーマの登録

### (1)実証実験の依頼

実証実験を開始するにあたっては、平成 23 年2月 21 日に行われた第4回検討会がキックオフとなった。会議に参加した評価者に対し、席上でメタデータ情報基盤構築事業技術ワーキンググループよりシステムの概要を説明し、実証協力を依頼した。

事務局からは、上記の骨子や依頼内容をまとめた依頼書を各機関個別に送付し、実証実験と評価を正式に要請している。各段階での作業内容の概略を記した実証作業シートに加え、作業手順書や画面の操作説明書などの関連資料を用意し、登録制 Web サイトにアップロードした。これによって、実証実験の作業から関連資料の閲覧まで、すべてをインターネット上で完結できる環境が整った。

スキーマの登録機能部分(フェーズ1)のシステム構築は予定通りに進行し、2月 15 日にリリースした。なお、第4回検討会を挟んで 25 日には検索・作成機能(フェーズ2)をリリースし、実証実験の実作業へと移行した。

なお、実証実験1～3は同時に依頼を行っているが、作業期間は上記開発フェーズに応じて2回に分けて設定した。実証実験1および2は、フェーズ2のリリース当日である2月 25 日から3月3日とし、当該機能部分がリリースに向けて準備中の段階にあった実証実験3は3月 11 日～3月17日と設定した。

タスク	作業開始日	作業終了日	ステータス	11月28日	12月12日	12月26日	1月9日	1月23日	2月6日	2月20日	3月5日	3月19日
基本設計	11月29日	12月10日	完了	■								
詳細設計	12月13日	12月28日	完了		■	■	■					
PHASE1												
開発	12月20日	1月31日	完了		■	■	■					
テスト	2月1日	2月13日	完了					■				
リリース準備	2月14日		完了									
リリース	2月15日		完了									
PHASE2												
開発	1月24日	2月14日	完了					■				
テスト	2月15日	2月23日	90%						■			
リリース準備	2月24日		-									
リリース	2月25日		-									
PHASE3												
開発	2月15日	2月28日	60%						■			
テスト	3月1日	3月9日	-									
リリース準備	3月10日		-									
リリース	3月11日		-									

## (2) 作業の具体的内容

自機関内で管理していると想定した記述規則サンプルとシステムの操作説明書を元に、簡易記述規則の作成と登録、記述規則メタデータの登録および未登録語彙の登録・公開という一連の作業フローの実施をお願いした。

各作業における具体的な内容の指示については、操作画面のキャプチャ画像を貼付した上で詳細な指示を記した実施手順書を用意した。作業手順の概略は以下の通りである。

### <記述規則のシステムへの取り込み>

- ① レジストリ管理メニュー内で「記述規則ファイル登録」を選択する。
- ② システム画面内の記述規則 URI と Excel 内の@base の一致を確認する。
- ③ Excel ワークシート内に作成した記述規則(サンプル)をテキストエディタに貼り込む。
- ④ 文字コードを UTF-8 に設定し、テキストファイルとして保存する。
- ⑤ 保存したテキストファイルをシステム上へ取り込む。

なお、Excel 内のテキストデータを直接システム上に取り込む方法も併せて紹介している。

### <未登録語彙の確認と追加>

- ⑥ 取り込んだ記述規則のメタデータがシステムに登録されているかどうかを確認する。
- ⑦ システム未登録だった場合は接頭辞や関連付けなどを定義してメタデータを追加登録す

る。

- ⑧ 登録されたメタデータとその定義の内容を確認する。

なお、タブ区切りのテキストデータではなく、RDF 形式のファイルで取り込む機能についても簡単な説明を補足した。

### 13.3 実証実験 2: スキーマの検索・作成

#### (1) 実証実験の依頼

スキーマの検索・作成についての実証実験は、前述の登録機能部分の実証と並行する形で進めた。フェーズ2がリリースされた2月 25 日より2つの実証作業を同時に開始し、同じ3月3日を作業・評価完了の期日とした。よって、作業開始までの経緯は前述の通りである。

#### (2) 作業の具体的内容

実証実験2は、スキーマの検索および作成機能を検証するものである。登録済みの語彙や記述規則をキーワード或使用プロパティによって検索する作業、新規もしくは既存データの編集によって新たな記述規則を作成・登録する作業となるため、実証作業の進行の円滑化を考慮し、別途手順書を用意せず、実証作業シート内に操作説明書の該当ページを指定する形を採った。主な実証の作業内容は以下の通りである。

#### <スキーマの検索>

- ① 特定の語彙について、すでにレジストリに登録され、公開済みとなっているものの中から登録機関やクラス/プロパティなどの条件を使用して検索し、語彙の定義ほかの詳細情報を参照する。
- ② 特定の記述規則について、すでにレジストリに登録され、公開済みとなっているものの中から登録機関やクラス/プロパティなどの条件を使用して検索し、語彙の定義ほかの詳細情報を参照する。
- ③ レジストリ登録されている語彙ならびに記述規則を任意のフォーマット(RDF/XML、Turtle、TopicsMaps、及び登録時のフォーマット)でダウンロードする。

#### <スキーマの作成>

- ④ 記述規則を新規に登録する際に、すでにレジストリに登録され、公開済みとなっているものを雛形として流用するために検索する(ここでは、既存の規則を流用するものと想定するために、あらかじめサンプルを登録する)。
- ⑤ 登録済みの記述規則をコピーし、編集することで新しい記述規則を作成し、それをレジストリに登録する。

### 13.4 実証実験 3:スキーマの変換

#### (1)実証実験の依頼

スキーマの変換についての実証実験は、前述の登録機能、検索・作成機能の実証とは別のスケジュールで実施した。該当機能(フェーズ3)のリリースは予定通り3月 11 日であったが、同日発生した東日本大震災の影響で、作業及び進捗管理に若干の混乱を来すこととなった。そのため、本実証実験は作業期間を3月 23 日まで延長することに決定した。

#### (2)作業の具体的内容

実証実験で3は、メタデータインスタンスの変換と再利用に関する作業をお願いした。先行する2つの作業に比べて内容が複雑であるため、より具体的な手順書を用意した。実証作業は、実証実験2で利用した記述規則(サンプルデータ)をすでにレジストリに管理されているものと想定し、CSV 形式、及び TSV 形式のメタデータインスタンスを RDF 形式や TopicsMaps 形式に変換出力することに主眼を置いた。主な実証の作業内容は以下の通りである。

##### <CSV ファイルを RDF 形式ファイルに変換出力する>

- ① レジストリ管理メニューから、既存あるいは新規の記述規則を選択する。
- ② 記述規則の内容を参照・確認し、メタデータインスタンスの入力形式・出力形式を指定する。
- ③メタデータ生成にあたって、既存の Excel のワークシート上のテキストデータを使用してメタデータインスタンスを指定する。
- ④ RDF 形式のメタデータを生成・ダウンロードし、ファイルの内容を確認する。

##### <参照した記述規則から DC 変換(シンプル・ダブリン・コア語彙)を行う>

- ⑤ メタデータ変換(DC)画面を開き、メタデータインスタンスの入力形式・出力形式を指定する。

メタデータ生成にあたって、④で作成した RDF 形式のファイルをアップロードしてメタデータインスタンスを指定する。

メタデータ変換(DC)を生成・ダウンロードし、ファイルの内容を確認する。

##### <システムの利活用に関する実証>

上記作業に併せて、コンテンツ供給者という想定での実証をお願いした。電子書籍や所蔵資料などのコンテンツの供給、既存コンテンツを活用あるいは変形した新サービスの実施などを行う立場としての視点から META BRIDGE 全体の機能を自由に操作し、実証実験への評価とともに感想をもらうよう依頼した。

### 13.5 実験結果のまとめ

参加各機関の協力によって、実証実験は概ね円滑に進行したが、フェーズ3のリリース日の3月11日に発生した東日本大震災によって若干の作業の遅延が生じた。また、当初の期間設定が余裕を持たせたものとは言えないため、各機関には負担をかけることとなった点については反省材料と言える。

実証実験の作業を終えた各機関による事後アンケートへの回答からは、有意義な情報が多数寄せられた。レジストリの使用感に留まらず、現状課題の指摘や今後のガイドラインづくりへのヒントとなり得る意見なども目立ち、メタデータ記述規則の環境整備全体に対する現場の関心の高さが浮き彫りとなる回答を得ることができた。

そこで、以下では寄せられた回答の集計と分析を行い、その結果を考察しながらメタデータ情報基盤システムの今後の展望について述べる。

なお、実証のサンプルデータ作成にあたって、岡山県立記録資料館の協力を頂いた。本実証は、こうした関係者の協力があって実現した事を付記するとともに、謝意を表したい。

## 第4部 今後の展望

### 14. メタデータ情報基盤システムのサービス展開

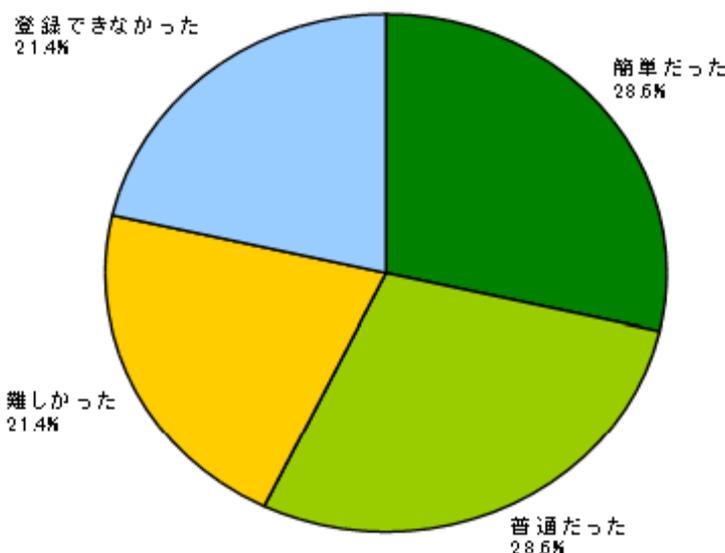
スキーマ・レジストリ実証実験の結果寄せられた評価を整理し、それらを踏まえたメタデータ情報基盤システムの今後の課題と展望を報告する。

#### 14.1 実証実験 1(スキーマの登録)の評価と課題

スキーマの登録機能に関しては、簡易記述規則の工夫、記述規則から語彙登録、メタデータ登録といった流れに関して評価を得た。一方で、内容が専門的で難しく、インターフェースや説明書の改善の必要が指摘されている。

##### 14.1.1 簡易記述規則の登録

簡単、普通を合わせて 57%と、過半数は無理なく登録できるという回答が得られたが、難しかったが21%あり、専門知識なしでも記述規則を登録できるようにするための方法としてはまだ改善の余地があることがわかった(図 14.1)。



◆ 図 14.1: 実証アンケート:簡易記述規則の登録について伺います。登録は簡単でしたでしょうか？

「簡易記述規則を作成し登録、という一連の作業は、専門知識を有していない者でも比較的容易に行なえるように感じました」という声に示されるように、シンプルな表形式による登録は、方向性としては受け入れられるものだと考えられる。

一方で、「値タイプにどんな値を入れれば良いのか、値制約としてどのような制約を与えれば良

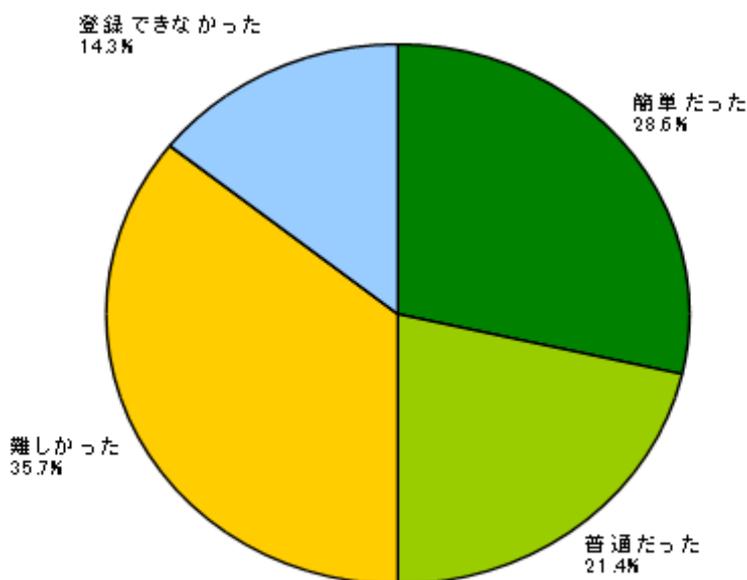
いのか等、不明な点が多い」との指摘を受けた。これは簡易記述規則に関する説明資料や例を十分に用意することで、ある程度は対応できるものと考えられる。

ただし、「Excel で維持管理している記述規則を META BRIDGE 登録する場合…TSV の形式に手作業で変換したうえで登録しなければならないので、メタデータに通じた人でないと作業が難しいと思いました」「エクセルファイルをベースに簡易 DSP を作成する機関が多いと思いますので、そのまま取り込める機能があると便利だと思います」などの声に示されるように、やはり既存のファイルをそのまま処理できる機能は重要と考えられる。

これは、機能要件「10.2.4.スキーマファイルを解析して変換の前処理を行なう」に相当するが、実装としては非常に難易度が高く、プロジェクト期間でのレジストリ組み込みを見送った機能である。

#### 14.1.2 記述規則メタデータの登録

簡単、普通を合わせて 50%、難しいが 36%であったが、簡易記述規則の登録が問題なくできた場合はメタデータも問題なく登録でき、前段階で躓いた場合は難しく感じられるという結果となった（図 14.2）。

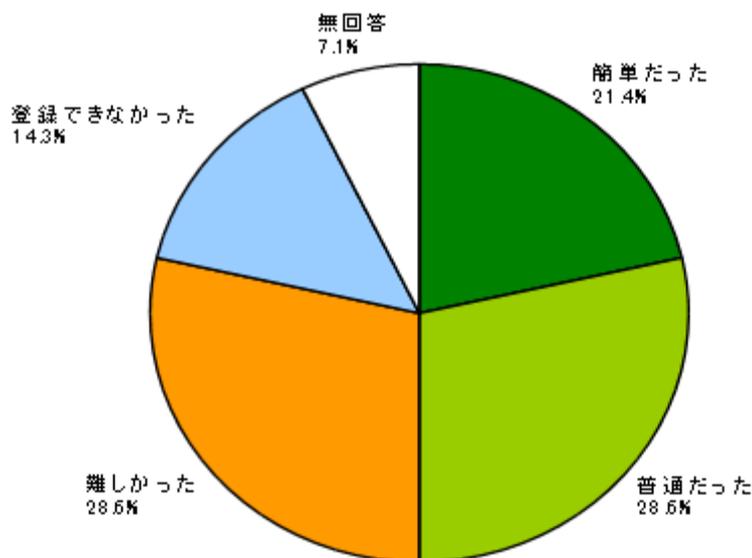


◆ 図 14.2: 実証アンケート:「記述規則メタデータの登録」(画面による登録)について伺います。登録は簡単でしたでしょうか？

記述規則にメタデータを与えることは特に問題なく理解されているが、手順書の記述、エラーが生じた場合のインターフェースなどが不十分で、分かりにくさの原因となっている面があった。

### 14.1.3 未登録語彙の登録・公開

簡単、普通を合わせて 50%、難しいが 29%であった。簡易記述規則の登録が問題なくできた場合はメタデータも問題なく登録でき、前段階で躓いた場合は難しく感じられるという結果となった(図 14.3)。



◆ 図 14.3: 実証アンケート:「未登録語彙の登録・公開」について伺います。登録・公開の操作は簡単でしたでしょうか？

「記述規則メタデータと未登録語彙の登録が一連の流れでできるのは便利だと思います」など、記述規則に加えて語彙を登録することについてはおおむね理解が得られている。

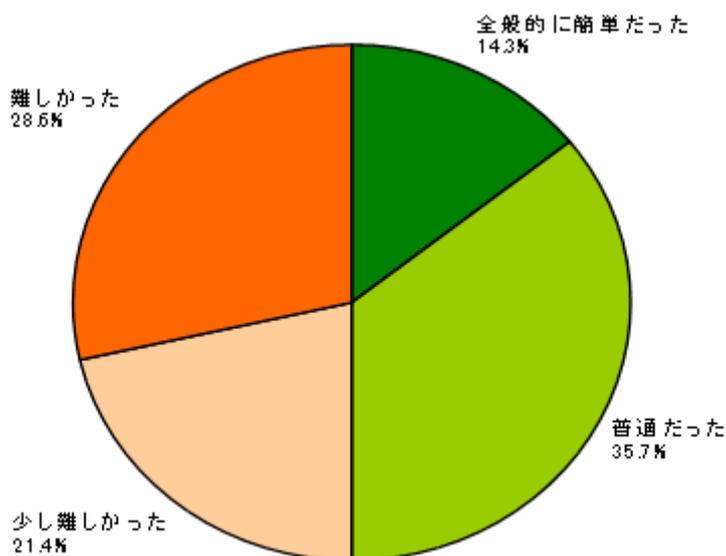
登録語彙は RDF スキーマで記述されていることを前提としており、この点のハードルの高さが指摘された。これは、既存語彙を記述規則で利用することを念頭に置いたものであったが、やはり語彙登録を求めることは負担が大きいと見られる。レジストリの登録語彙が増えれば「未登録語彙」を登録する必要は減少するので、この問題はある程度解消されると考えられる。

既存語彙ではなく、独自の語彙を用いた記述規則については、レジストリの登録画面で語彙を対話的に登録できる。この違いについては、より分かりやすい説明が必要である。

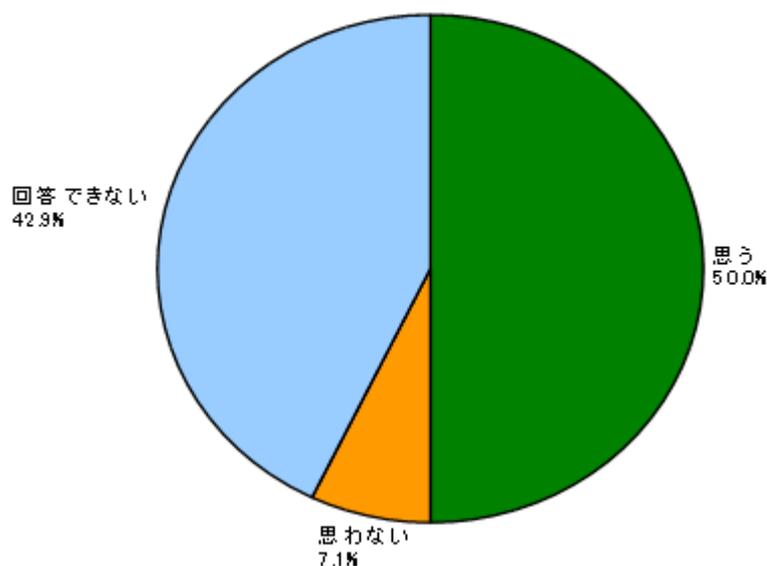
### 14.1.4 登録機能全般についての評価

全体的な感想としては、簡単、普通を合わせて 50%、少し難しいが 21%、難しいが 29%であった。

また登録の意向については、回答できない(スキーマを保有していない)を除くと、大半が登録したいと思うという回答であった(図 14.4、図 14.5)。



◆ 図 14.4: 実証アンケート:スキーマを登録した一般的な感想について伺います。実際にサンプルを登録した感想について教えてください。



◆ 図 14.5: 実証アンケート:保有するメタデータ・スキーマを登録してみたいと思いましたか？

開発期間の制約により、プロジェクトで開発したレジストリの登録機能は、メタデータ専門家でな

いと利用が難しいものにとどまっている。しかし登録意向の高さに示されるように、適切なガイド、インターフェース、および既存ファイル読み取り機能などが整えば、レジストリにスキーマを登録し、共有してもらうことは十分可能と考えられる。

## 14.2 実証実験 2(スキーマの検索・作成)の評価と課題

スキーマの検索・作成機能については、特に既存スキーマから新規規則を作成できる点について評価された。また詳細画面において多様な形式でスキーマが提供されることなども評価された。一方で、登録と同様、インターフェースや説明書を丁寧に変更する必要があるが指摘されている。

### 14.2.1 14.2.1 語彙の検索

語彙の検索については、「異種のメタデータから同一内容の要素を通覧でき、比較などが可能で便利ではないか」「専門的な知識がある人で、さらに、目的の語彙が明確であれば検索するのに役に立つと思います」と、機能の役割は一定の評価を得た。

使い勝手に関しては、

- 語彙の登録数が増えたときに、検索性が低下しないか
- 曖昧検索ができるとさらに使い勝手が良くなる
- 未知の語彙を発見することが非常に難しいと感じた
- 語彙のカテゴライズなどの検索支援が必要だと感じた
- クラス／プロパティでの検索時に、URI による検索や接頭辞を用いた検索もサポートされるとよいと思う

など具体的な改善点が挙げられた。

### 14.2.2 記述規則の検索

記述規則の検索については、「同じ語彙を利用している記述規則を通覧、比較でき作成の参考となる」「ある語彙定義を用いている記述規則を容易に調べることができ、新しく記述規則を作成する際の参考としても用いることができる」と、機能は評価された。

使い勝手については、

- 記述規則の用途が、検索一覧画面ですぐに分かるとよい
- 記述規則が多くなると検索しにくくなると思われる
- 用途やカテゴリごと(図書館メタデータ用・博物館メタデータ用など)に記述規則が一覧できるとより便利

といった点が挙げられた。

特に記述規則は語彙よりも用途が細分化されるものであるため、「記述規則のメタデータを丁寧  
に記述することを各機関に求めたほうが良いと感じました」「すでに登録されている記述規則の  
うち、内容説明のないものがありますが、ここは入力必須にした方がよいと思います」など、用途  
が分かるような情報が必要であるとの指摘があり、重要な改善点と考えられる。

### 14.2.3 語彙・記述規則の詳細情報

詳細情報は、「検索したメタデータ・スキーマをRDF形式で出力できる点に感動しました」「上位  
プロパティ、定義域、値域に指定したものの詳細情報が簡便に見れて良いと思います」「いろい  
ろな形式で取り出せるのは、非常に便利だと感じました」など、全体的に高い評価を得た。

改善点としては、

- 複数の規則をマッピングした形で比較して見られるようにできると良い
- 出現回数などの定義の意味などをわかりやすく表示する機能があるとさらにわかりやすくなると感じました
- ある程度構造化が行われた記述規則は、現在のインターフェースでは構造を理解しにくい。そのため、構造をグラフ表現等で視覚化したインターフェースが有用だと思う

などの意見が挙げられた。

### 14.2.4 登録済み規則を雛形としての新規規則作成

登録済みの規則を元にして新規規則を編集・作成する機能については、「既存の規則を流用す  
ることで、登録者の負担が減ってよいと思います」「似たようなメタデータを登録する際には、非常  
に効率的に行えると感じました」など、おおむね高い評価を得た。

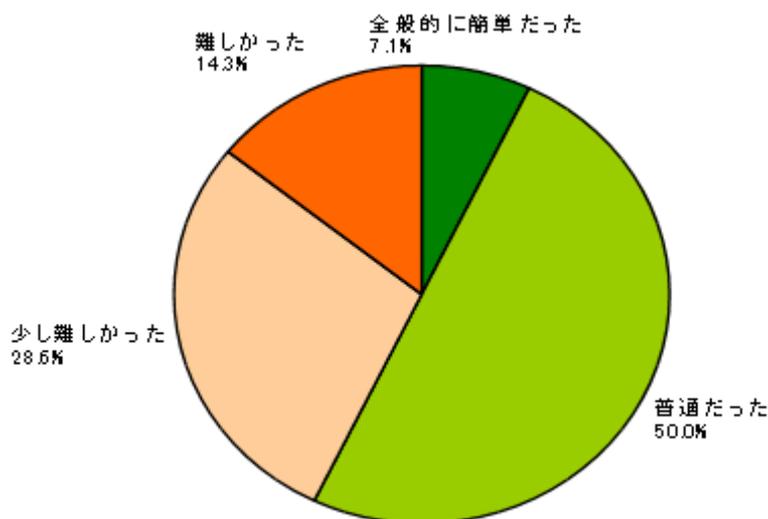
使い勝手に関しては、

- 使う語彙はそのままに、出現回数制約の変更をどうやるのかがわかりませんでした
- クラスや値制約のリンクをクリックするとタームの詳細の方に遷移するので、遷移先が異なることを明示した方が親切だと思います
- 順序の入れ替えなどの各操作が若干煩雑なように感じました
- Web インターフェースで編集するよりもローカルファイルとして編集した方が速いこともあるので、記述規則は登録方法に関わらず簡易記述規則を出力できるとよい

など、細部について改良の必要性が指摘された。

### 14.2.5 検索・作成機能全般についての評価

全体的な感想としては、簡単、普通を合わせて 57%、少し難しいが 29%、難しいが 14%であった。語彙の登録に比べると難易度は低くなっているが、まだ改善の余地がある(図 14.6)。



- ◆ 図 14.6: 実証アンケート:スキーマを登録した全般的な感想について伺います。実際にサンプルを登録した感想について教えてください。

機能全体としては、「登録したスキーマをすぐにRDF やトピックマップの形式で提供できること、バージョンの管理が簡便に行えることは魅力的であると思います」など、ほぼレジストリの狙いに沿ったものとして評価されている。

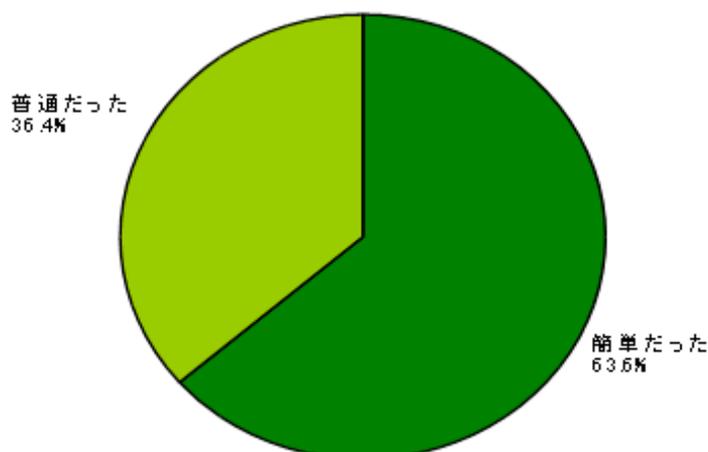
一方で「利用者操作説明書が説明不足であったり、操作が直感的に分かりづらいところがあるなど、ユーザへの配慮に欠けるところが散見されるので、多くの方が利用しやすいように改良していただければと思います」と指摘されているように、実際の利用に提供するためには、時間をかけて丁寧な改良が必要だといえる。

## 14.3 実証実験 3(メタデータの変換と再利用)の評価と課題

メタデータ変換やダム・ダウンの機能は高い評価を得た。実証した機能自体は使いやすいとの評価を得たが、積極的に活用されるためには、もう一段の使いやすさが求められる。また、API による機能提供の必要性も確認された。

### 14.3.1 CSV 形式データの RDF への変換

レジストりに登録された記述規則を用いて、CSV 形式のデータを RDF に変換する機能は、64%が簡単、36%が普通と回答し、難しい、できないという答えはなかった(図 14.7)。



◆ 図 14.7: 実証アンケート: RDF 形式ファイルに出力した感想をお伺いします。出力は簡単でしたでしょうか？

シンプルな規則に基づく表形式のデータでメタデータを管理している組織は少なくないと思われるので、この変換が容易にできることは意義があるといえる。

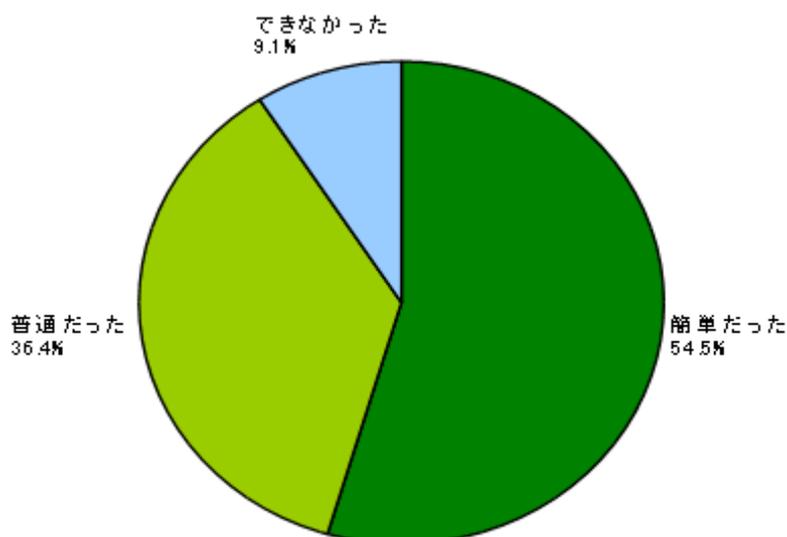
使用感としては、

- 手順書に沿って行うのは簡単でしたが、実際に自機関でこの機能を使おうとする際に、果たしてスムーズに使えるか、若干不安を感じました
- 出力結果が表示されるまでのクリック数が多く、煩雑に感じた
- 有向グラフ構造が可視化されていると、RDF 体系の理解が促進されると思われるなど、ユーザインターフェースに関する改善意見が挙げられた。また、
- ファイルの出力方法として、REST のインターフェースがあるとより便利に使えると思う

という意見が挙げられた。本レジストリは、「コンピュータによる問い合わせインターフェース (10.4.3)」を機能要件として開発しているが、CSV 形式データの RDF への変換については API を用意しておらず、今後の改良点といえる。

### 14.3.2 トピックマップとしての出力

変換したメタデータをトピックマップとして出力する機能は、55%が簡単、36%が普通と回答した、9%ができなかったという回答であった(図 14.8)。



◆ 図 14.8: 実証アンケート:TopicMaps へ出力した感想をお伺いします。出力は簡単でしたでしょうか？

機能については、

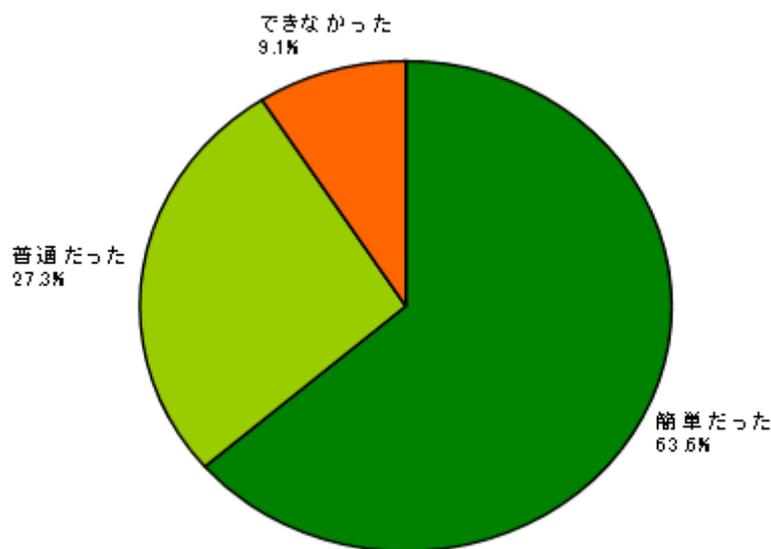
- 現場で、TopicMaps へ出力する機会がどれほどあるかは何とも言えませんが、TopicMaps の仕様を知らなくても変換できるという点は便利だと思いました
- 出力されたトピックマップは、RDF とトピックマップのモデルの違いにより完璧にはできない

が、これをもとに完成させるのに十分役立つ

という意見があった。また「一般的に具体的事例がやや少なく、エンドユーザにとって出力の意味があるのか検討が必要と思われる」という指摘もあり、今後の開発における優先順位を考える必要がある。

### 14.3.3 シンプル DC への変換

記述規則のマッピングを利用して、メタデータインスタンスをシンプル DC に変換(ダム・ダウン)する機能については、64%が簡単、27%が普通と回答したが、9%ができなかったという回答であった(図 14.9)。



◆ 図 14.9: 実証アンケート: シンプル DC への変換を行ってみた感想をお伺いします。変換は簡単でしたでしょうか？

使用感については CSV の RDF 変換とほぼ同様の意見が寄せられたが、機能については「シンプル DC へ簡単にダム・ダウンできるのはとても良いと思いました」など、高い評価を得た。また、ダム・ダウンについては「各規則がどの程度あいまいになるのかが示される仕組みがあっても良いかもしれない」という意見もあった。

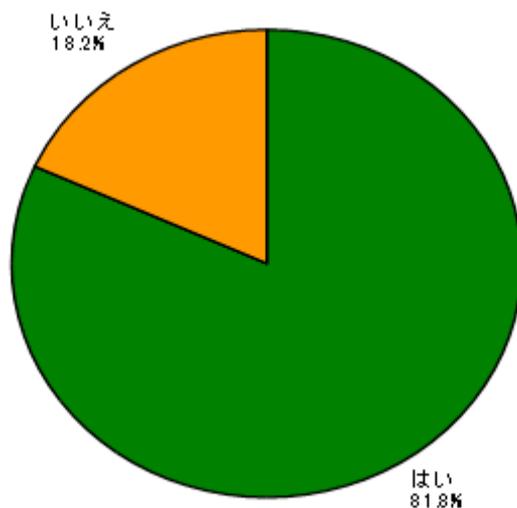
#### 14.4 全体的な評価と今後の課題

レジストリの機能全体は一定の評価を受けており、適切な運用と機能改良により、今後広く利用される可能性は示された。特に異なる領域のスキーマ検索やバージョン管理など、従来どこでも提供されなかった機能については、大きな期待が寄せられている。

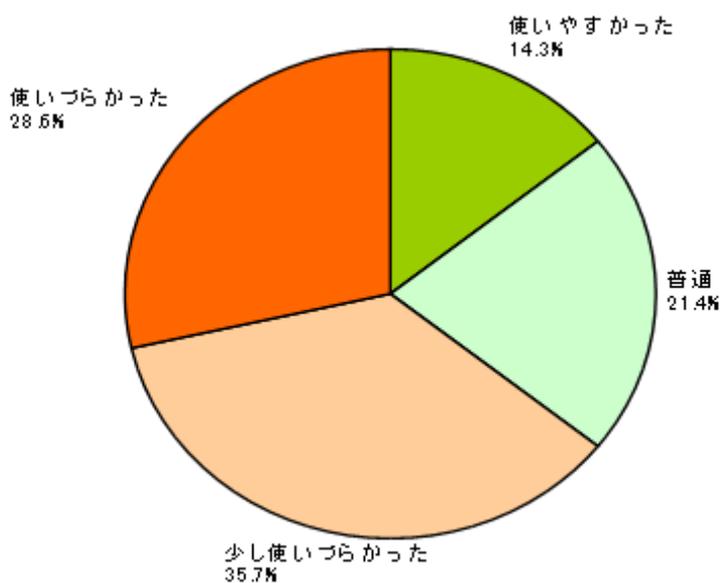
一方で、プロジェクト期間の制約の中で開発したレジストリには、使い勝手や説明書の面で不十分な点が残されていることも明らかになった。

##### 14.4.1 システム全体に関する評価

サービスの利用意向としては、82%から使ってみたいという回答を得たが、使いやすさについては少し使いづらい、使いづらいを合わせて 64%という結果となった(図 14.10、図 14.11)。



◆ 図 14.10: 実証アンケート:実際にサービスに利用してみようと思いましたか？



◆ 図 14.11: 実証アンケート:システムの使い勝手はいかがでしたでしょうか？

利用したいという回答のコメントとしては

- 必要としているメタデータを検索できる事と、容易に RDF 形式などの出力ができるため
- 自分が作成したメタデータが公開され他社に利用されることで、メタデータ設計のためのフィードバックが得られるようになると考えられる
- 図書館や博物館などで使用されるメタデータは有用なものが多く、それらのメタデータ・ス

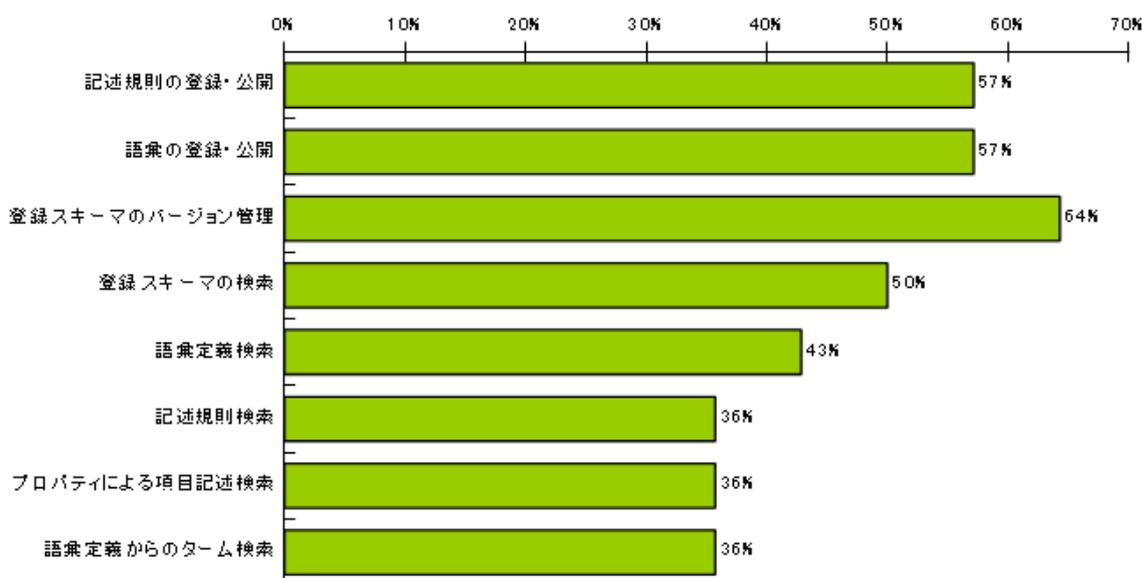
キーマを網羅的に知る方法として優れている

- メタデータ語彙作成などを十分な知識なく手探りでやってきたが、このようなツールがあれば語彙の作成も容易になるほか、特に他の語彙を流用するなどして利用できる意義は大きいと考える
- 様々なスキーマを集積する基盤として有効性がある
- 他の登録済みスキーマを参照/流用できる点で有効性がある

などの積極的な意見が得られた。

#### 14.4.2 関心のある機能

実証実験 1 および 2 において、実証項目以外も含めて関心のある機能を答えてもらったところ、図 14.12 のような結果となった。



◆ 図 14.12: 実証アンケート:登録機能も含めて、関心のある機能がありましたら、教えてください。

回答者の立場がメタデータ提供者ということもあるが、規則・語彙の登録および公開と、バージョン管理について高い関心が寄せられている。

またスキーマの検索に関しては、

- 多数のスキーマや語彙が単に「登録してある」というだけでは、どれを選択すればいいのか利用者側で判断しづらい。検索をする中で、ユーザが求めるメタデータ・スキーマや語彙がどれであるのか分かるように、選択基準をシステム側で提示できるとよいのではないだろうか

という意見があった。要件として挙げた「10.3.5.複数記述規則間の関連を調べ、分かりやすく示すことができる」機能は、やはり開発上の制約で実装できなかったが、こうした補助情報をどのように提供するかも、レジストリの今後の課題である。

#### 14.4.3 今後の課題

各実験の評価でも述べたとおり、機能については期待が高くまた一定の評価も得ているが、限られた開発リソースでは、十分に使いやすいシステムを作ることができなかった。

総合的な意見としては、次のようなものがあげられる。

- 大変面白いシステムであると感じました。メタデータ語彙の作成について、十分な知識がなくても既存の語彙の「いいとこ取り」をして作成できる、という点は大きく評価できると思います。
- 職場でも何人かに見ていただきましたが、用語が難しい、との意見を頂きました。用語集付きのヘルプがあれば解決できるかと思います。
- 全体的にメタデータ、あるいはメタデータシステムに精通した者にとっては有効なシステムだと思われるが、中小規模の MLA 利用対象者にはシステムに明るくない者も含まれると思われる。そのような利用者に対して、専門用語や RDF の仕組みを理解せずとも直感的に操作できる UI が必要と思われる。
- 特に博物館領域においてはスキーマが著しく異なるため、今後のデータ流用や再利用が困難な場合も出現すると思われる。そのような場合に本システムが緩やかに相互利用を促す仕組み作りが必要と思われる。

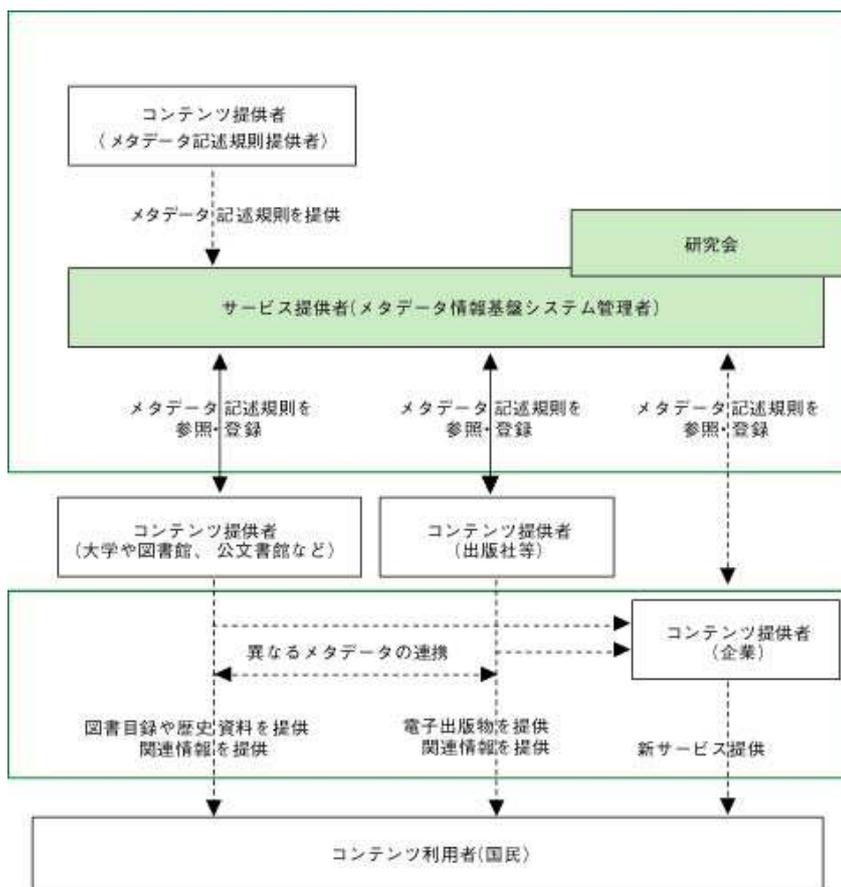
実験協力者の期待にこたえるためにも、継続した開発でシステムの改良を進めることが重要である。

また重要な指摘として、「スキーマや運用を変える度にいちいち訂正作業を半永久的に行っていかなければならず、非常に手間が大きい」という運用の負担の問題が挙げられた。これは、レジストリ機能の改良でカバーできる面もあるが、このレジストリを通じたメタデータ情報共有の重要性・意義が理解されないことには、こうしたシステムの利用は促進されないことをも意味している。「そもそも『なぜ RDF によるデータの公開が重要なのか』等、普及啓発活動が重要になる」という指摘のとおり、開発と普及啓発活動のバランスをとり、しっかりした基盤によって今後の活動を維持推進していくことが不可欠である。

## 15. メタデータ記述規則を収集、提供、維持管理するための体制の整備

### 15.1 レジストリの維持運用

言うまでもないことであるが、レジストリの運用は長期間の維持管理を前提としたものでなければならぬ。本プロジェクト自体の期間は平成 22 年度に限られるものであるが、プロジェクト期間終了後も構築したシステムの運用を継続していく計画である。



◆ 図 15.1: プロジェクト実施後のメタデータとレジストリの位置づけ

#### 15.1.1 メタデータ・スキーマの登録と共有

メタデータ・スキーマの利用性を高めるには、産官学民の区別なくメタデータ・スキーマを登録し、利用できるようにすることが望まれる。その一方、これまで産官学民のメタデータ・スキーマに関する情報共有は積極的には行われてこなかったため、今後いろいろな新しい課題がでてくることを想定して情報共有を進めていかねばならない。

たとえば、官学のほうにも機関や領域によって、メタデータ・スキーマ開発と維持管理に関する経験と実績は多様である。加えて、従来は、図書館や博物館、美術館、公文書館といった領域ごとに情報共有が進められてきたのに対し、今後は領域にまたがった情報共有を進めることが必須

である。学術領域で作られるメタデータ・スキーマは多様であり、かつ学術領域の専門家は必ずしもメタデータの専門家ではないといった問題もある。

一方、視点を世界に転じると、欧米の国立図書館や公文書館等による Web 上でのメタデータ・スキーマ共有のための取組、MLA 機関によるデジタルアーカイブ開発の協調的取組等が目につく。こうした海外の取組と我国の取り国の間での連携を容易にするためにもメタデータ・スキーマの相互運用性を高めるための基盤を構築することは有用である。

### 15.1.2 維持運用のための検討課題

レジストリの維持運用には、今後いくつかの視点からの更なる検討が必要である。

#### (a) 方針面での検討課題

- 公的機関と大学等の非営利機関、民間が作るメタデータ・スキーマの収集、蓄積、提供に関する組織整備
- 長期的視点に基づくレジストリサービス提供に関する要求要件の明確化

#### (b) 技術面での検討課題

- 本プロジェクトの実証評価等を基礎にしたメタデータ・スキーマの収集、蓄積、提供をより効率化するための技術開発
- 他レジストリとの相互連携の仕組みの開発
- 登録されたメタデータ・スキーマの長期利用のための技術開発

#### (c) 運用面での検討課題

- メタデータ・スキーマの収集、蓄積、提供をより効率化するためのレジストリ運営組織のあり方
- 産官学民の間で協調しやすいレジストリ運営組織のあり方

### 15.1.3 レジストリの継続的开发

(a) 開発の継続 第 14 章で述べたとおり、実証実験においては、レジストリには高い期待が寄せられた反面、多くの課題も指摘された。これらの課題の多くは、本プロジェクトにおいて要件としてあげながらも、開発期間、予算の関係から実装できなかった、以下の機能に含まれているものである。

- スキーマファイルを解析して変換の前処理を行なう(10.2.4)
- 複数記述規則間の関連を調べ、分かりやすく示すことができる(10.3.5)
- 記述規則に基づいて、データ入力の HTML フォームテンプレートを用意できる(10.3.6)
- 記述規則に基づいて、データ入力のエクセル入力ウィザードを用意できる(10.3.6)

- メタデータから、使用語彙一覧と類似記述規則一覧を得ることができる(10.4.1)
- メタデータを、JSON フォーマットに変換する(10.4.2)
- 記述規則、語彙定義の RDF を、グラフ URI を指定して取得できる(10.4.3)

これら以外に実証実験で指摘された改善点も含め、今後の継続的開発で解決すべき課題となっている。

さらに、DCMI(Dublin Core Metadata Initiative)のレジストリ・タスクグループ(DCMI-RTG)が示している「レジストリの要件(案)」<sup>1</sup>の動向も見据えて、国際的水準で必要とされる機能が十分に備わっているかについても、確認と改良を続けることが望まれる。

**(b) レジストリの国際化** 本プロジェクトのレジストリは、機能としては日本語に限らず多言語のメタデータ・スキーマを登録・検索利用することができる。しかし利用者インターフェースについては、開発期間の関係もあり、日本語のメニューしか備えていない。

本プロジェクトは、DCMI-RTG においても”Metadata Information Infrastructure Project”として認知されており、国際的に見ても先進的な試みのレジストリとなっている。利用者インターフェース、利用説明の英語版を提供し、国際的な利用と連携を進めることが期待される。

**(c) レジストリ間連携** 本レジストリに蓄積されるメタデータ・スキーマは、レジストリが提供する機能を通して広く利用されるべきであるとともに、他の同様なレジストリとも連携して利用可能となることが望ましい。レジストリ間の連携は国際的な課題でもある。

DCMI-RTG は、レジストリ間相互運用について2011年度の活動プランを示している。プランでは、レジストリ間相互運用の基本要件の例として

- レジストリ間での相互自動更新
- メタデータ・スキーマ、語彙のハーベスティング

を挙げ、これらの要件の明確化とともに、相互運用機能のテスト実装を 2011 年度前半にも実施したいとしている<sup>2</sup>。テストの対象としては本レジストリも挙げられており、対応が望まれる。

#### 15.1.4 維持運用の組織と計画

以上のような視点からの継続的検討・開発、および持続的なサービスの実現を行うための組織の立ち上げ、あるいは既存機関による持続的サービスの実現に関する方向性を明らかにしていくことが必要である。

<sup>1</sup> Common requirements – Metadata-Registry, [http://wiki.metadataregistry.org/Common\\_requirements](http://wiki.metadataregistry.org/Common_requirements)

<sup>2</sup> DCMI Registry Community – Metadata-Registry, [http://wiki.metadataregistry.org/DCMI\\_Registry\\_Community](http://wiki.metadataregistry.org/DCMI_Registry_Community)

(a) **筑波大学におけるレジストリの運用** 筑波大学知的コミュニティ基盤研究センターにサーバを設置し、メタデータ・スキーマ・レジストリを引き継ぐ。

当面は大学において継続的運用を行なうが、メタデータの長期的な活用の観点からは、組織変更などの可能性のある大学よりも、より安定的な組織による運用に移行できることが望ましい。

(b) **法人化** 長期安定的な運用を行なう組織としては、NPO 法人等の設立も検討する必要がある。大学を中心に投資する NPO 法人が考えやすいが、安定的運用のための資金を確保するためには、より広い基盤に立った法人設立も検討しなければならない。

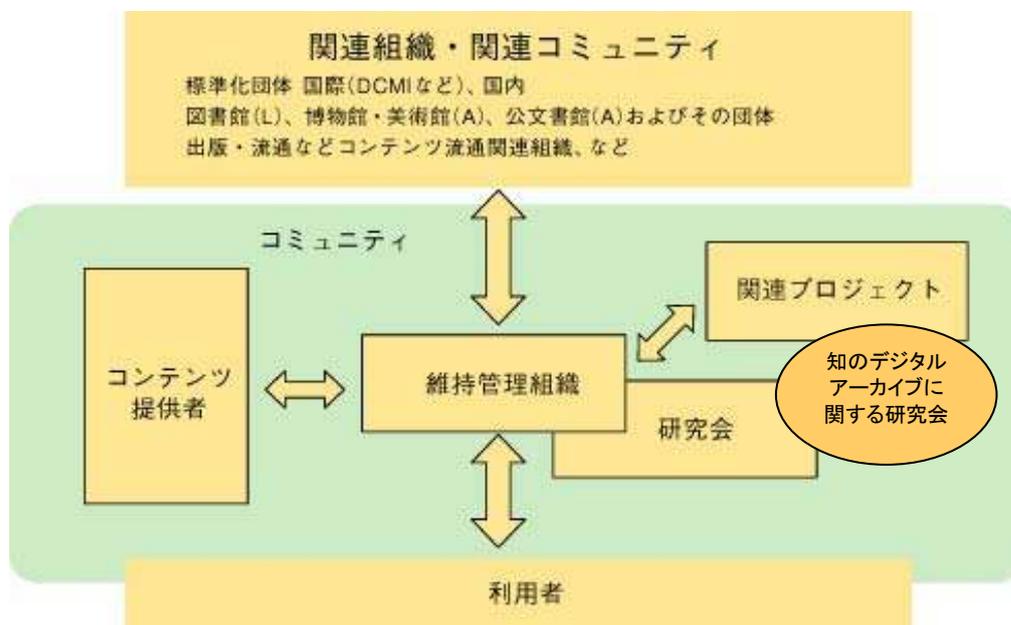
また、継続的な検討・開発のためにも、資金面の裏付けを持つ組織における維持・運用が望まれる。

(c) **研究会の継続** 本プロジェクトを機に筑波大学知的コミュニティ基盤研究センターにおいて発足したメタデータ情報基盤研究会は、事業終了後も継続する計画である。今後も、より発展した形で、産官学民の有識者が連携して、メタデータ共有のための基盤について意見交換する場とすることが望まれる。

研究会がレジストリの運営を行なうわけではないが、こうした議論の場から、レジストリに対する助言・支援を受けることは重要である。

## 15.2 記述規則定義言語とガイドラインの周知・普及

本プロジェクトで開発したメタデータ・スキーマの記述形式(記述規則定義言語)、またメタデータ記述規則記述の普及は重要な課題である。関係機関との調整や利用者からのフィードバックを受けた改良、また国際的な標準化も念頭に置いた体制とコミュニティの形成が求められる。(図 15.2)。



◆ 図 15.2: コミュニティによって記述規則定義言語・ガイドラインを維持・発展させる

### 15.2.1 記述規則定義言語を用いたスキーマの共有

メタデータ記述規則をネット上で共有することは、これから新しいデジタルコンテンツやアーカイブのサービスを始めようとするすべての機関・組織にとって有用である。たとえば、図書館の場合であれば、新しいサービスをデザインする図書館員が記述規則を利用すると思えねばならない。

(a) **記述規則定義言語の公開** 記述規則定義言語(OWL-DSP)は、ウェブオントロジー記述の標準である OWL を用いて表現されている(Appendix 4.1 参照)。この OWL による定義自身は、恒久 URI サービス<sup>3</sup>を用いて以下に公開する予定である。

<http://purl.org/metainfo/terms/dsp>

また本報告でも示したとおり、簡易な表形式記述のための規則(簡易 DSP)を定め、その解説も用意しており、同様に公開する予定である(Appendix 4.2)。

さらに記述規則定義言語は、スキーマ・レジストリにおいても、語彙定義のひとつとして登録・公開される。

<sup>3</sup> 組織変更などによって URI が変更されてもリソースへのアクセスを確保するために、登録した URI へのリクエストを、実際にリソースがおかれているサーバに転送するサービス。PURLs (Persistent Uniform Resource Locators) が代表的である

(b) **メンテナンス** プロジェクトの成果としての定義言語(OWL-DSP)は第 1 版ということになる。こうした記述言語は、実際のメタデータ・スキーマ記述への適用からのフィードバックを受けて改良されるべきものである。

レジストリの維持管理組織において、OWL-DSP の改良とレジストリ登録の更新を行ない、メンテナンスしていく必要がある。

(c) **記述情報と一般向けの解説** 本プロジェクトの提案は WWW コンソーシアムの標準をベースにしており、特に記述規則定義言語(OWL-DSP)に関しては、技術的知識がないと理解が困難な部分もある。その一方、ネットワーク上での相互運用性を考えた場合、こうした技術的部分は必要不可欠の内容でもある。

こうしたことを総合すると、技術的な知識無しに理解できる部分とそうでない部分の明確化、利用者の立場・役割に応じて理解すべき内容の違いを明確化すること、そして、利用者の立場や役割に応じた内容の違いを反映した適切な学習資料を開発することなどが求められる。また、セミナー等の学習の場を設けることが望まれる。

OWL-DSP の理解には、実例を見ることがもっとも近道である場合もあり、レジストリからの定義言語 OWL の出力を利用することも周知させて行きたい。

(d) **国際化** 66.2 で示したように、メタデータ・スキーマ定義方法の標準化に関しては、DCMI のシンガポール・フレームワークによって理論的基盤が固まっているものの、定義言語(DCMI-DSP)に関しては 2008 年に草案が出されたままで進展を見ていない。

OWL-DSP は、シンガポール・フレームワークおよび DCMI-DSP を踏まえ、かつセマンティック・ウェブの技術を組み込んで設計されており、メタデータ記述規則定義言語の国際標準となり得るものである。仕様の英語版を作成して DCMI に働きかけるなどして、日本発の国際標準として推進することが望まれる。

### 15.2.2 ガイドラインの周知と運用

ガイドラインについては、本事業において有識者、関係者からのコメントの収集を行った。そこにおいても回答者が持つ知識や経験の違いによって理解の容易さに関する回答は様々であった。こうした読者の背景知識や立場による違いはこれからも変化するとは思えない。

メタデータ記述規則の共有やメタデータの相互運用に関する啓蒙活動や研修活動については、レジストリの維持運用とともに、長期の視点の下に進めるべき今後の課題であると思われる。

(a) **パブリックコメントによる周知と継続的開発** プロジェクト終了時点で本ガイドラインの第 1 版を公開することになるが、記述規則定義言語と同様、ガイドラインも利用者のフィードバックを受けてよりよいものに改良していくべきものである。

具体的には、総務省ホームページにおいてパブリックコメントを受けるといった方法が考えられる。

パブリックコメントを募ることにより、ガイドラインを広く告知し、関心を持って内容を吟味してもらえると期待できる。

**(b) 関連団体、プロジェクトへの提示と意見収集** 本事業と平行して進められた他の「新 ICT 利活用サービス創出支援事業」プロジェクトの中には、何らかのレベルでメタデータに関わるプロジェクトも存在する。これらプロジェクトの実証・開発成果等も参考に、本ガイドラインに修正すべき点はないか等について、引き続き検証する。また、反対に、本ガイドラインも含め本プロジェクトの実証・開発成果等を、他の関連プロジェクトへフィードバックすることも有益であると考え。これらの方等を通じ、各プロジェクト間の関連な意見交換や成果共有化を図っていくことが重要である。

また、メタデータの提供者である雑誌協会、図書館協会などの動きとも連動し、ガイドラインへのフィードバックを求めるとともに、それぞれのメタデータ規則をより相互運用性の高いものにするためにガイドラインを活用していくことも重要である。

上記以外にも、デジタルコンテンツの発信、流通に関わるプロジェクトが国の指導の下に行われており、そうしたプロジェクトとの間でも本プロジェクトで開発したガイドライン等、メタデータ・スキーマ共有に関わる情報の共有の機会を設けることが望まれる。

**(c) 継続的開発の主体** ガイドラインの継続的開発の主体としては、レジストリの維持管理組織(筑波大学)がまず考えられる。継続する研究会からも、ガイドラインに関する助言を引き続き受けるべきである。

また、総務省が開催している「知のデジタルアーカイブに関する研究会」における議論もメタデータとは関係が深く、当該分野に深い知見を持つ学識経験者も多く参加している。こちらでより多角的な視点からガイドラインをメンテナンスしていくことも考えられる。

**(d) メタデータ共有のためのコミュニティ** メタデータの共有を実現するためには、ガイドラインが文書として存在するだけでなく、それを実際に適用し、議論していくコミュニティが不可欠である。ガイドラインの継続的な開発やメンテナンスも、コミュニティが関与し、意見が反映されることが望ましい。

既存のコミュニティとしては、たとえば「情報知識学会」「日本図書館情報学会」「情報科学技術協会」のようなメタデータを扱う学協会も考えられる。

また、メタデータの共有を考えるための新たな枠組み、コミュニティが必要という意見もあり、産官学民の壁を越えた多くの関係者が集う形での新しい組織体を形成することも考える必要がある。

## 16. メタデータ共有がもたらす可能性

本プロジェクトの目的は、メタデータの利用率や相互運用性の向上を図り、コンテンツの流通性向上に資することである。これが実現することにより、どのような可能性が開けるのかを検討する。

### 16.1 メタデータの応用と領域を超えたサービス

WWWでの検索におけるメタデータの利用、ウェブサービスにおけるマッシュアップ、MLA領域での取り組みから、メタデータの応用と、領域を超えた情報・コンテンツ流通の可能性を展望する。

#### 16.1.1 百科事典から WWW へ

領域を超えたさまざまな情報をすばやく得たいという考えは、中国や日本の「類書」やプリニウスの『博物誌』にさかのぼり、18 世紀末の『百科全書』を出発点とする今日の百科事典に連なっている。1960 年代末に出版された「The Whole Earth Catalog」や 1992 年の「The Whole Internet User's Guide & Catalog」も、この流れで捉えることができるだろう。

これらは、目録的なメタデータとは異なり、情報をまとめた記述がその中心となっている。しかし情報本体へのアクセスが容易であれば、目録メタデータとアクセス手段の組み合わせにより、百科事典に近い役割を実現することも可能になる。

WWW の発達と普及は、まさにこの情報へのアクセスに革命的な変化をもたらした。あらゆる領域の情報が、居ながらにして利用可能となったのである。これを百科事典として活用するために必要なのは、情報の中から有用なものを発見するための目録メタデータということになる。

初期の WWW では、まず人手による「ディレクトリサービス」としてこのメタデータが提供された。WWW 発祥の地である CERN の The WWW Virtual Library に始まり、米 Yahoo! の登場によって広く利用されるようになったディレクトリサービスは、WWW 上のさまざまな情報リソースを人手によって収集し、カテゴリ別に分類したものである(図 16.1)。



◆ 図 16.1: 初期の Yahoo!ディレクトリと WWW Virtual Library

リストにはウェブページのタイトルが用いられ、簡単な紹介文や人気度などの評価マークが加えられる場合もあった。メタデータとしてはごく限られたものだが、ディレクトリに掲載される時点で人手による情報のフィルタリグが行なわれており、その点でも事典による情報探索に近いものだと見えるだろう。

### 16.1.2 検索サービスとメタデータ

WWW 上の情報リソースが急速に増加し、数十億ページの規模になると、人手によるディレクトリサービスでは十分なアクセス機能の提供が困難となり、ウェブ上のあらゆる情報をデータベースに収集した全文検索機能を提供するサービスが登場する。ここでは、機械的に生成されたメタデータが中心的な役割を果たすことになるが、ここに人手によるメタデータを加味する動きも出始めている。

(a) **全文検索とメタデータ** 膨大な情報に対する全文検索では、検索結果が多過ぎて必要なものが見つけられないという事態が生じる。機械的な収集であるため、ヒットした情報の質も保証されない。そこで、情報リソースの更新日などのメタデータを利用して並べ替えを行ない、利用者の選択を助ける方法が採られた。また、ウェブページの一部や、ページサイズ、情報タイプ (HTML、ワード文書など) なども、補助的なメタデータの役割を果たした (図 16.2)。



◆ 図 16.2: 全文検索の結果には、タイトル、テキストの一部のほかに、日付やサイズなどの補助メタデータも示された

(b) **検索結果の評価メタデータ** 大量の検索結果から適切なものを見出すための指標メタデータとして、さまざまなものが用いられてきた。検索語の出現頻度、語のユニークさも勘案した TF-IDF、出現位置による重み付けなどが挙げられる。

Google の PageRank に代表されるリンクポピュラリティも、計算による評価メタデータの一つである。これは WWW のハイパーリンク構造を利用し、多くのページからリンクされているページほど高く評価するというものである。単純なリンク数ではなく、リンク元ページの評価 (リンク数) を加味することにより、相互リンクなどの意図的なリンク増に左右されないランク付けを可能とした。

(c) **作者メタデータの組み合わせ** HTML の meta 要素を用いた作者によるキーワードは、書誌

の標目（「5.1.1.図書館分野のメタデータ記述規則参照」のような第三者によるメタデータとは異なり、意図的に検索結果を操作しようとするものが含まれるなど信頼度が低いことから、うまく活用できなかった。そのため全文検索ではリンクポピュラリティのような機械的メタデータが主流を占めてきた。

しかし、意味を正しく伝えるメタデータは、今のところ人の手によって作るしかない。より高度な検索のために、作者メタデータが改めて見直されてきている。RDFa などを用いたセマンティック・マーク付け（「16.3.セマンティック・マーク付けとレジストリ」参照）により、ウェブページの商品情報や評価情報を読み取り、検索結果に反映させようというものだ（図 16.3）。

**In Search of Beethoven (2009) - IMDb** 🔍  
 ★★★★★ Rating: 7.4/10 - from 132 users  
 Directed by **Phil Grabsky**. Starring **Juliet Stevenson, Emanuel Ax, Jonathan Biss**. The first truly comprehensive feature length cinema documentary ever made about **Beethoven**. With over 60 live performances.  
[www.imdb.com/title/tt1308123/](http://www.imdb.com/title/tt1308123/) - Cached - Similar

**In Search of Beethoven Movie Reviews, Pictures - Rotten Tomatoes** 🔍  
 ★★★★★ Rating: 92% - 39 reviews  
 Directed by **Phil Grabsky**. Starring **Juliet Stevenson**. With a wealth of insightful talking heads and deft musical performances, **In Search of Beethoven** is a thorough, thoughtful examination of the great ...  
[www.rottentomatoes.com/.../in\\_search\\_of\\_beethoven/](http://www.rottentomatoes.com/.../in_search_of_beethoven/) - Cached - Similar - Add to iGoogle

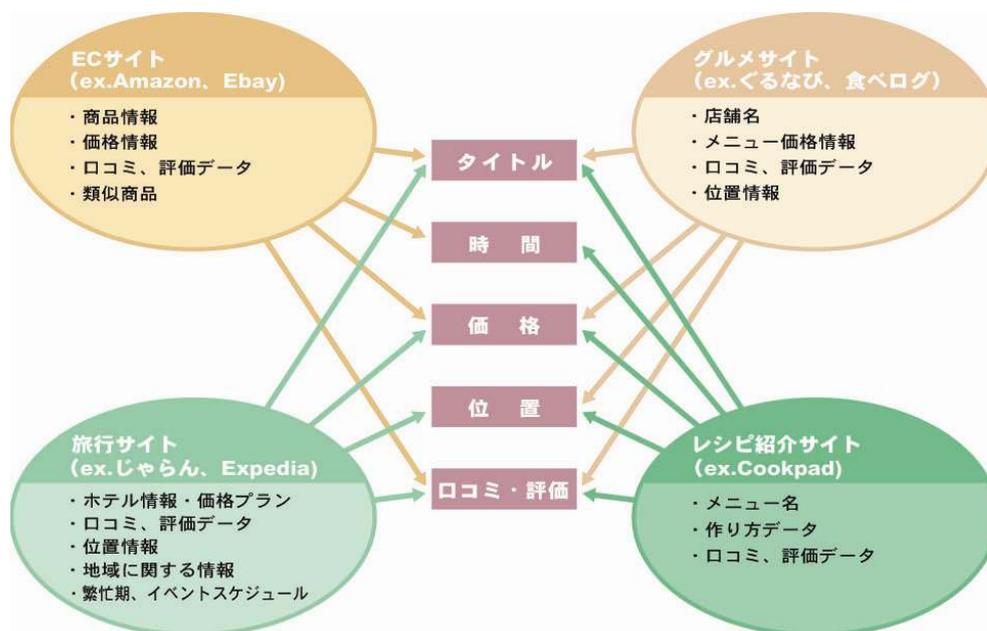
レビューや監督、俳優などのメタデータを含む検索結果

◆ 図 16.3: 検索結果に、サイトから得た評価情報などのメタデータが組み合わされる

検索エンジンが統計的手法で生成する機械的メタデータとこうした作者メタデータの組み合わせは、メタデータ利用の新しい可能性を示している。検索サービスにおけるセマンティック・マーク付けの利用については、16.3.2 検索エンジンとセマンティック・マーク付けで詳述する。

### 16.1.3 ウェブサービスを結ぶメタデータ

オンラインショップの商品情報、グルメサイトの店舗情報、イベントサイトのカレンダー情報など、ウェブサービスはメタデータの塊といえる。これらの多くは API という形で外部にも提供され、複数サイトのメタデータを組み合わせたマッシュアップと呼ばれるサービスも盛んだ（図 16.4）。



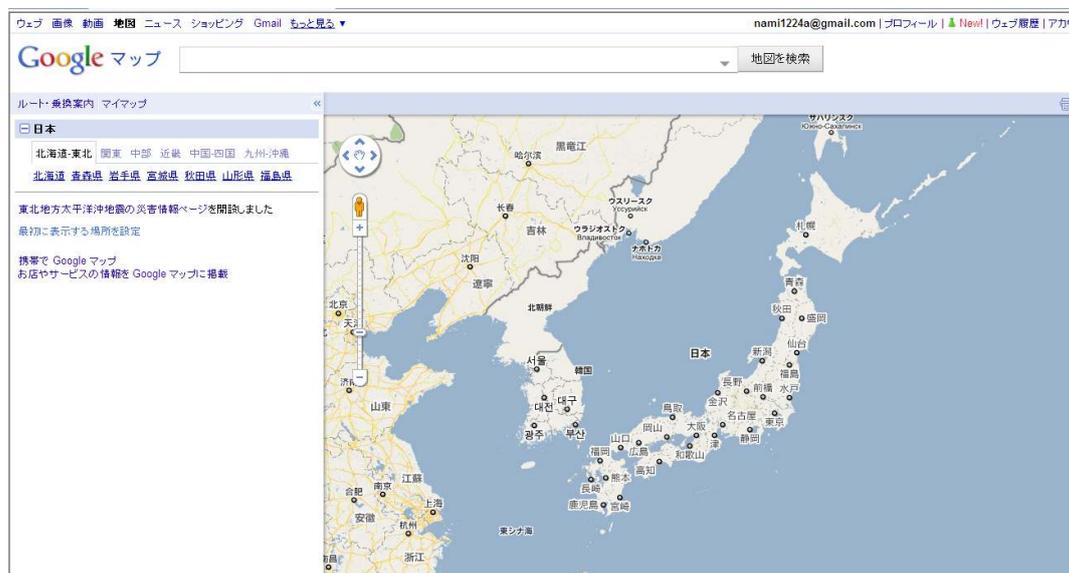
◆ 図 16.4 複数のウェブサービスのメタデータを組み合わせて利用する

こうしたメタデータは、API のパラメータも提供される情報もサイトごとにまちまちで、マッシュアップのためにはそれぞれのサービスに個別に対応するしかない。とはいえ、いくつかのメタデータ項目は多くのサービス間で共通しており、現状でもそれらを用いた組み合わせの工夫が行なわれている。これらのメタデータが共通化されれば、組み合わせ利用の可能性は大きく広がるだろう。

(a) **位置データの利用** 店舗の所在地、イベントの開催場所、写真の撮影場所データなど、多くのメタデータは何らかの形で位置情報を含んでいる。位置情報の記述方法はさまざま存在するが、データそのものは GPS 利用の普及で世界測地系による百分率表記<sup>1</sup>が一般的となり、記述プロパティも GoogleMaps が採用している KML や GeoRSS、Geo タグの利用が広まりつつある。

これを利用すると、あるイベントの開催地を地図上に表示し、その付近のお勧めレストランや近辺で撮影された写真をマッシュアップするといった応用が可能になる。イベント情報を多面的に伝え、集客向上につなげることができるだろう。

<sup>1</sup>東経 139 度 46 分 12.45 秒とする代わりに 139.770125 と記述する。



◆ 図 16.5: GoogleMaps 上でのマッシュアップ

また、異なるサービス提供者が提供するレストランのメタデータが標準的方法で提供されれば、それらを一つの地図上に一括表示させることも可能となり利用者の利便性向上を図ることができる。さらに、データのリンクによりレストランのメニューの情報から、料理のレシピサイトへのナビゲートや食材の直売サイトへのナビゲート等のサービス展開も実現可能である。

(b) 時間データの利用 Google カレンダーのようなクラウド型カレンダーが普及し、スケジュールを公開したりグループ間で共有できるようになってきている。また、ソーシャルイベントサイトの Eventcast<sup>2</sup>のように、新商品発売日、TV 番組の放映スケジュール、展覧会など、さまざまなイベントスケジュールをユーザが登録し、共有するサービスもある(図 16.6)。

これらのイベント情報の多くは、iCalendar 形式<sup>3</sup>での入出力が可能となっている。これを利用して、興味あるイベント情報を自分のスケジュール表に取り込んだり、複数のイベントをマッシュアップしてひとつのカレンダーに表示するといった組み合わせが実現している。

さらに、位置データと並んで時間データは情報リソースを現実世界に結びつける役割を担う。領域を超えた情報活用のための、鍵となるデータと考えることができる。

<sup>2</sup> eventcast - ソーシャルイベントサイト, <http://clip.eventcast.jp/>

<sup>3</sup> rfc2445, <http://www.ietf.org/rfc/rfc2445.txt>

The screenshot shows the Eventcast website interface. At the top, there is a navigation bar with the Eventcast logo and a user greeting "こんにちは! ゲストさん". Below the navigation bar is a search bar with a "キーワードで探す" button and a "検索 オプション" button. The main content area is titled "「まもなく開催のイベント」" and lists three events:

- Event 1:** [今日]2011/03/20(日) 19:00 - 民放FMで同時生放送の特別ライブ、今年はSuperflyが登場. URL: k-tai.impress.co.jp. Date: 2011年01月24日 02時56分. 2 users. Includes links for [Clip], [Y!], [G], [RTM], and [iCal]. Category: ラジオ 音楽 KDDI.
- Event 2:** [今日]2011/03/20(日) 10:00 - 15:00 HARU COMIC CITY 16. URL: www.akaboo.jp. Date: 2011年02月18日 15時18分. Includes links for [Clip], [Y!], [G], [RTM], and [iCal]. Category: イベント.
- Event 3:** [今日]2011/03/20(日) - 21(月) 「河鍋曉斎記念美術館」公認! 神田サオリ&横井紅炎&宇田川優 (Vigoo) スペシャルコラボレーションイベント! URL: soul-event.jugem.jp. Date: 2011年03月05日 23時34分. Includes links for [Clip], [Y!], [G], [RTM], and [iCal]. Category: 音楽 クラブ ダンス フェスティバル ライブイベント.

On the right side, there is a calendar for March 2011, showing the current date as 2011年03月20日(日). Below the calendar is a section titled "eventcastって?" with a brief description of the service and a "もっと詳しく" link. At the bottom right, there is a "無料登録/ログイン" button.

◆ 図 16.6: ユーザがイベントスケジュールを登録して共有できる Eventcast

(c) **商品情報と価格データ** すべての商品は基本的に価格を持つ。価格データは、さまざまな EC サイトのメタデータをつなぐ共通項のひとつとなる。

価格データを利用するサービスの代表例としては、価格.com(家電一般)やカーセンサー(中古車)などの価格比較サイトが挙げられる(図 16.7)。

各 EC サイトの価格データは、共通の方法で提供されているわけではない。価格.com においては、当初は運営者が手作業で集めた情報を掲載していた。現在は、各店舗の担当者が価格.com に情報を登録する仕組みをとっているが、いずれにしても効率的ではなく、再利用も難しい。

商品データを RDF として記述する GoodRelations 語彙<sup>4</sup>の利用や、セマンティック・マーク付けによる価格データの明示が行なわれれば、こうした情報を自動収集できるようになる。商品データを利用した独自のサービス開発も容易になり、さらにセールスの情報を位置データ、時間データと組み合わせる応用も考えられるだろう。

(d) **ユーザ評価と口コミ** 多くの EC サイトでは、利用者が感想や評価を書き込む機能を備え、他の利用者が商品を選択するための情報として活用されている。ユーザ評価を集約するレビュー

<sup>4</sup> GoodRelations The Web Ontology for E-Commerce, <http://www.heppnetz.de/projects/goodrelations/>

サイトも数多い。またウェブログ、あるいはツイッターなどのマイクロブログによって個人が直接発信する情報の中にも、商品やコンテンツの感想・評価が無数に存在する。

こうした評価、あるいは口コミ情報は、現状では個々のサイトごとにしか利用できない。しかし、RDF Review 語彙<sup>5</sup>、あるいは hReview マイクロフォーマット<sup>6</sup>を用いることで、こうした評価をセマンティック・マーク付けとして記述したり、共有することが可能になる。

The screenshot shows a product page for 'Music Synthesizer S90 XS'. The main price is ¥230,000 (tax included). It lists 6 stores and shows a price comparison table with two entries, both at ¥230,000. The first entry is from 'ムラウチ' in Tokyo, and the second is from 'オフプライス楽器' in Chiba. The page also features a user review section with a 53rd ranking and a 'レビュー募集中!' (Review collection!) banner.

順位	価格 (差額)	送料	在庫・発送の目安	店頭	ショップ/評価 支払方法	地域	登録日/コメント	ショップ 情報
1位	¥230,000 (最安)	無料	~5営業日		ムラウチ 代揃	東京	2010/11/28 11:00:29 クレジットカードOK! 年に一度の「大決算セール」開催中!	詳細を見る
1位	¥230,000 (最安)	無料	有		オフプライス楽器 代揃	千葉	2011/03/18 15:35:47 送料無料!	詳細を見る

◆ 図 16.7: 商品スペックと価格比較を提供する

評価・口コミ情報を適切に利用するためには、いくつかの視点が必要になる。

- 評価の対象: 何を評価しているのかが分からなければ役に立たないので、評価される側のコンテンツや商品の標準メタデータとセットになることが不可欠である。
- 評価者: 位置・時間データや商品データと異なり、評価者個人の主観に基づくものであるため、誰が評価しているかによって情報の信頼度や価値が異なってくる。評価者を識別し、その評価者を「判断」するための情報が必要になる。
- 評価時期: 新製品の発売直後と1年後、あるいは対向商品の発売前後では、評価の基準が異なってくる可能性がある。そのため、評価情報には日時データが不可欠である。

<sup>5</sup> hReview 0\_3 · Microformats Wiki, <http://microformats.org/wiki/hreview>

<sup>6</sup> WebID - W3C Wiki, <http://www.w3.org/wiki/WebID>

特に評価者の識別と「判断」のための情報は、今後のさまざまなメタデータ利用において重要になってくる。EC サイトや口コミサイトでは、その評価者が他にどんなレビューを書いているかが「判断」の材料となるが、評価者をそのウェブログやツイッターアカウントと結びつけることができれば、より多角的な「判断」が可能となるだろう。

こうした情報は、FOAF 語彙(「5.3.3.FOAS」参照)でも記述できる。またウェブ上での個人を識別する標準方法として、WebID<sup>7</sup>が提案されている。

#### 16.1.4 MLA メタデータの広がり

博物館・図書館・文書館(MLA)領域では、5.1 で示したように領域ごとにメタデータ記述の国際的な標準化が進められているが、これを超えてさらに広くメタデータを共有するための試みも始まっている。

(a) **ヨーロッパナ** ヨーロッパナ(Europeana)<sup>8</sup>は、欧州委員会が 2008 年 11 月に公開した電子図書館ポータルサイトである。欧州各国の図書館、美術館が保有する一千万を越す文化遺産の情報を集約し、横断的に検索できるようにする。ヨーロッパナ自身が作品を保有するのではなく、作品のメタデータを蓄積して、それぞれの資源を保有する図書館、美術館(の目録ページ)へのリンクを提供する(図 16.8)。

---

<sup>7</sup> WebID - W3C Wiki, <http://www.w3.org/wiki/WebID>

<sup>8</sup> Europeana - Homepage, <http://www.europeana.eu/portal/>

◆ 図 16.8: 欧州各国の一千万を越す文化遺産の情報を検索できる Europeana

ヨーロッパでは、ひとつのキーワードからテキスト資料、画像資料、ビデオ資料、音声資料を一括検索する。さらにこの結果は、作品の年度、国など異なるファセットによって絞り込むことができる。資料を単独で見るのとは異なる切り口、文脈で提示することによって、新たな発見が導かれる。

さらに、アムステルダム自由大学が提供するヨーロッパの実験版セマンティック検索エンジンでは、作品のメタデータを RDF によって記述している。これを利用して、作品からその作品が作られた土地へ、さらにその土地で生まれた別の作者の作品へ、あるいはその土地を描いた作品へと、メタデータの関連をたどってさまざまな情報資源を引き出していくことができる。

(b) 図書館とウェブ 図書館領域では、ウェブを経由した検索機能の提供や館相互のデータ交換に早くから取り組んできた。さらに FRBR(「5.1.1.図書館分野のメタデータ記述規則」参照)を中心とする次世代メタデータモデルは、図書館以外の領域からも注目されている。

図書館のデータをよりひろくウェブで流通可能にするために、2010 年に W3C において Library Linked Data Incubator Group が組織された<sup>9</sup>。これはメタデータを RDF を用いてモデル化し、リンク

<sup>9</sup> W3C Library Linked Data Incubator Group, <http://www.w3.org/2005/Incubator/llid/>

するデータ(「16.2.リンクするデータ」参照)として流通できるようにしようという活動で、2011 年 5 月に最終報告が出される予定になっている。

また、標準化組織による活動だけでなく、図書館の現場や開発者もデータ共有の方法を積極的に探っている。2003 年の秋にアメリカの図書館関係のプログラマ、システム技術者を中心としたコミュニティとして発足した Code4Lib<sup>10</sup>は、データ共有のための研究発表やコード提供を活発に行っており、2010 年には Code4Lib JAPAN<sup>11</sup>も発足した。

さらに、2010 年にスタートした図書館横断検索サービスの「カーリル」<sup>12</sup>は、領域外から図書館データの活用を試みた例として注目に値する。全国 5000 館以上の図書館から蔵書、貸し出し状況を検索してくれるだけでなく、API を公開し、ユーザがそれぞれの“レシピ”を作ったり、それを公開して共有するなどの斬新な試みが展開されている。メタデータが標準化され、領域外からも利用可能になると、従来とは異なる切り口のサービスが生み出され得ることを示している(図 16.9)。



◆ 図 16.9: 全国の図書館の横断検索や API を提供するカーリル

<sup>10</sup> code4lib coders for libraries, libraries for coders, <http://code4lib.org/>

<sup>11</sup> <http://www.code4lib.jp>

<sup>12</sup> カーリル 日本最大の図書館蔵書検索サイト, <http://calil.jp/>

## 16.2 リンクするデータ

メタデータを RDF で記述・共有することで、セマンティック・ウェブの技術を用いたデータの再利用やコンテンツ流通性の向上が望める。特に、データ相互が有機的につながって大きなウェブを構成する「リンクするデータ」への期待が高まっている。

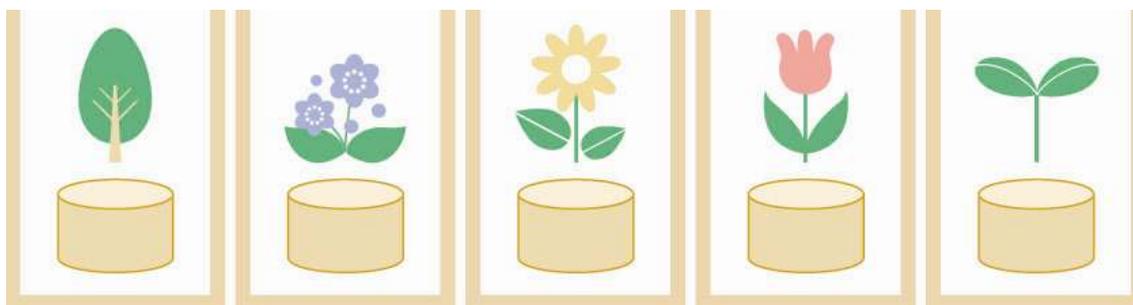
### 16.2.1 リンクするデータとは

WWW が発展してきたのは、さまざまな情報を結びつけるハイパーリンクの力によるところが大きい。情報が一元的に管理されるのではなく、ユーザがそれぞれ作成した文書において、シンプルな方法で容易に別の文書とのリンクを設けることができる。文書のつながりはどんどん広がり網の目が密になって「**文書のウェブ**」を織り成す。読者はリンクをたどることで、関連する情報だけでなく、思わぬ未知の世界との出会い(セレンディピティ)が得られるようになり、ウェブの力はネットワーク効果によってどんどん高まってゆく。

セマンティック・ウェブは、データを RDF で記述して公開するものである。RDF はデータの名前付けに URI を用い、データを併合することができる。RDF のデータが文書とハイパーリンクと同じようにつながりを広げていけば、「**データのウェブ**」ができあがり、データもネットワーク効果で価値を高めることができるだろう。

(a) **リンクしないデータ** RDF は URI を用いた記述が特徴だが、その URI は名前付け(識別)が第一義的な目的であって、必ずしもリンクを意図したものではなかった。人間や会社を識別する URI をたどってもその人やオフィスに出会えるわけではない(「4.2.3.URI とリソース」参照)。

RDF の記述に用いる URI が参照解決可能であっても、そのデータを ZIP ファイルなどで圧縮して公開しているのでは、アプリケーションがリンクを次々にたどっていくことはできない。また会員制でパスワードを要求する「閉じたウェブ」では、その内部でいくらデータがつながっても「データのウェブ」とはならず、リンクしないデータのままである(図 16.10)。



◆ 図 16.10: 閉じたシステムのデータは、その中でしか生かせない

領域をまたいで広がりつつあるウェブサービスのメタデータも、個々の API でそれぞれ異なった形態の情報が提供され、それを取得して利用できるだけでは、さらにデータがつながる「ウェブ」に発展しないのである。

(b) **リンクするデータの 4 原則** データのウェブを実現するためには、URI を識別だけでなくリンクとしても用いることが重要になる。WWW の創始者でもあるティム・バーナーズ＝リーは、2006 年に「Linked Data」という文書を発表した<sup>13</sup>。そこでは、データが文書と同じようにリンクしていくために必要な 4 つの原則が述べられている。

1. ものごとの名前付けに URI を用いる
2. その名前を参照できるように、URI には http:スキームを用いる
3. その URI を参照したら、有用な情報が標準的な方法で得られるようにする
4. 他の URI で名前付けされたものごとへのリンクを含め、新たな情報を発見できるようにする

文書のウェブにおいては、ウェブページの URL が http:スキームで、HTML という標準的な方法で提供され、そこに他のページへのリンクが含まれる、ということは当然のように実現されている。しかし従来のデータと比較したとき、文書のウェブが成功した要因は、そしてウェブ上のデータに欠けていたものは、この原則であった。



◆ 図 16.11:リンクするデータが新たな価値を生む

このシンプルで強力な原則を受け、セマンティック・ウェブを「データのウェブ」「リンクするデータ」として実現させようという動きが広がり始める。

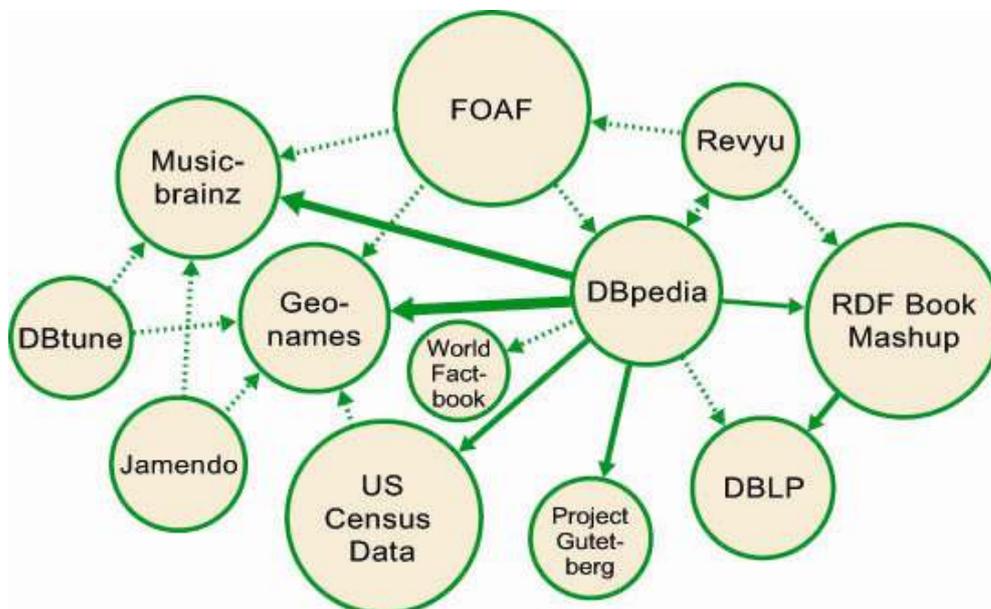
<sup>13</sup> Linked Data - Design Issues, <http://www.w3.org/DesignIssues/LinkedData.html>

## 16.2.2 Linking Open Data プロジェクト

ソフトウェアのオープン・ソースと同様に、データについても著作権や特許に縛られない誰でも自由に使えるものを提供しようという「オープン・データ」の考え方がある。さまざまな科学データの共有を目指す Science Commons から写真共有サイトに自由ライセンスで公開されている画像に至るまで、ウェブ上ではさまざまなオープン・データが提供されている。

これらが「リンクするデータ」となれば、アプリケーションによる共有・再利用の道が大きく開ける。ウェブ上のオープン・データを RDF で記述して結びつけようと、「リンクするオープン・データ(LOD, Linking Open Data)」活動が 2007 年の春から始まった<sup>14</sup>。

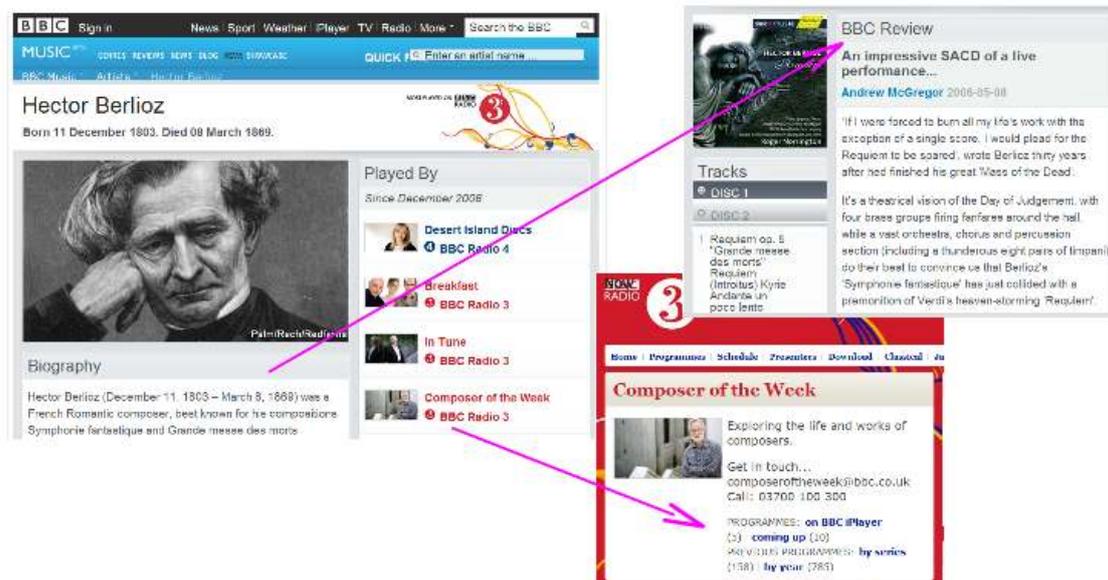
発足当時は図 16.12 のような範囲であった LOD のネットワークは、急速な勢いで広がりを見せ、2010 年 9 月には図 16.13 のような密なつながりに成長している。



◆ 図 16.12:2007 年 5 月の LOD ネットワーク

<sup>14</sup> SweoIG-TaskForces-CommunityProjects-LinkingOpenData - W3C Wiki, <http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>





◆ 図 16.14: BBC の作曲家ページは CD 評ページや番組ページにリンクする

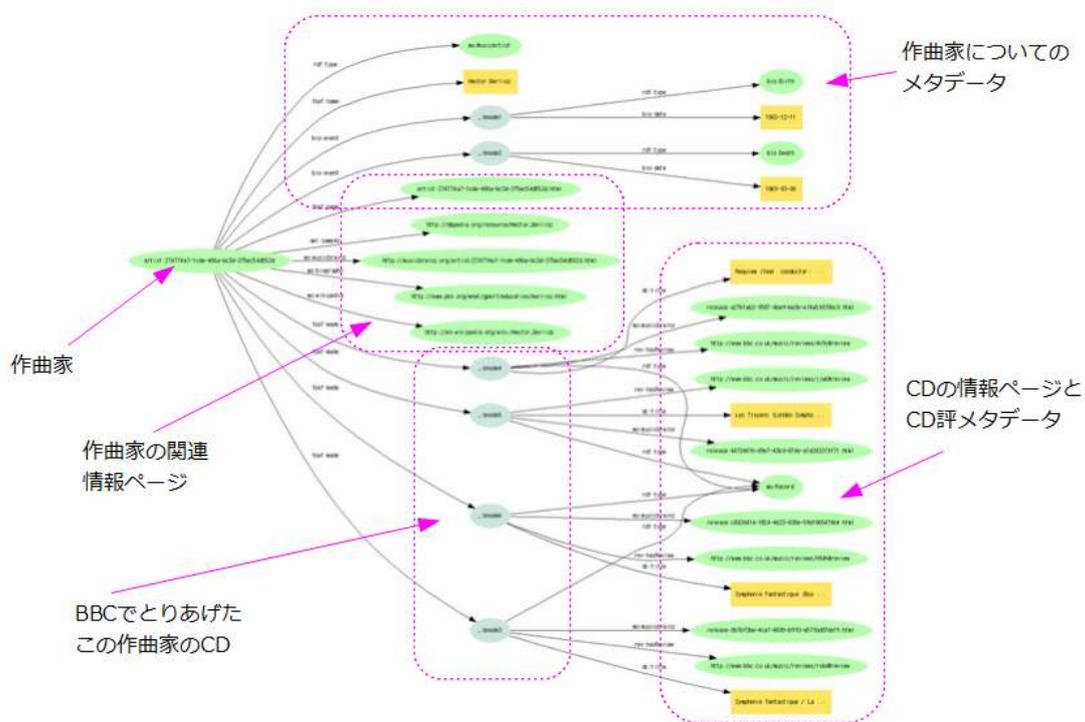
一方、同じ URI に RDF アプリケーションがアクセスすると、サーバからは HTML ではなく、RDF のデータが送り返される。この RDF からは、図 16.15 のようなグラフが得られる。作曲家を表すリソースから、作曲家自身のメタデータだけでなく、BBC がとりあげた作曲家の CD を表すリソース、その CD のメタデータや CD 評のメタデータなど、関連するリソースにリンクがつながっている様子が分かる。

(a) BBC 音楽データと MusicBrainz BBC の音楽ページは、URI に MusicBrainz ID (「5.2.2.音楽データ」参照) を利用している。たとえばベルリオーズには MBID として 274774a7-1cde-486a-bc3d-375ec54d552d が割り当てられているので、BBC のこの作曲家のページは次の URI を持つ。

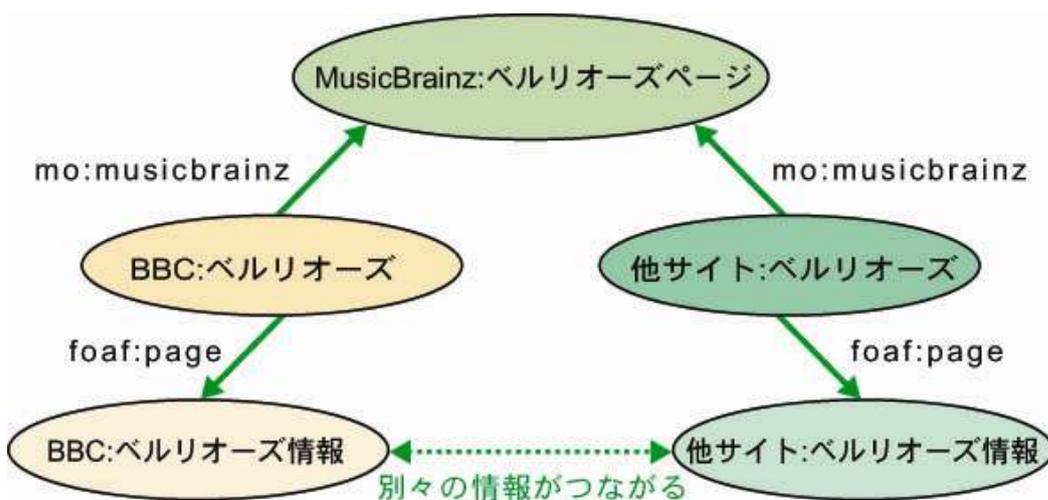
<http://www.bbc.co.uk/music/artists/274774a7-1cde-486a-bc3d-375ec54d552d>

さらに RDF においては、Music Ontology 語彙<sup>15</sup>を用いて、作曲家リソースを MusicBrainz のベルリオーズのページに結び付けており、また取り上げた CD は MusicBrainz のそれぞれのアルバムページに結び付けている。BBC 以外のウェブサイトでも、同じベルリオーズに関する記事ページが同様に MusicBrainz に結び付けられれば、独立して作られたベルリオーズに関する情報が結びつくことになる。「リンクするデータ」が実現するわけである(図 16.16)。

<sup>15</sup> Music Ontology Specification, <http://musicontology.com/>



◆ 図 16.15:BBC の作曲家 RDF は関連するリソースの記述にリンクする



◆ 図 16.16:リンクするデータによって異なる情報がつながる

### 16.2.4 リンクするデータのハブ

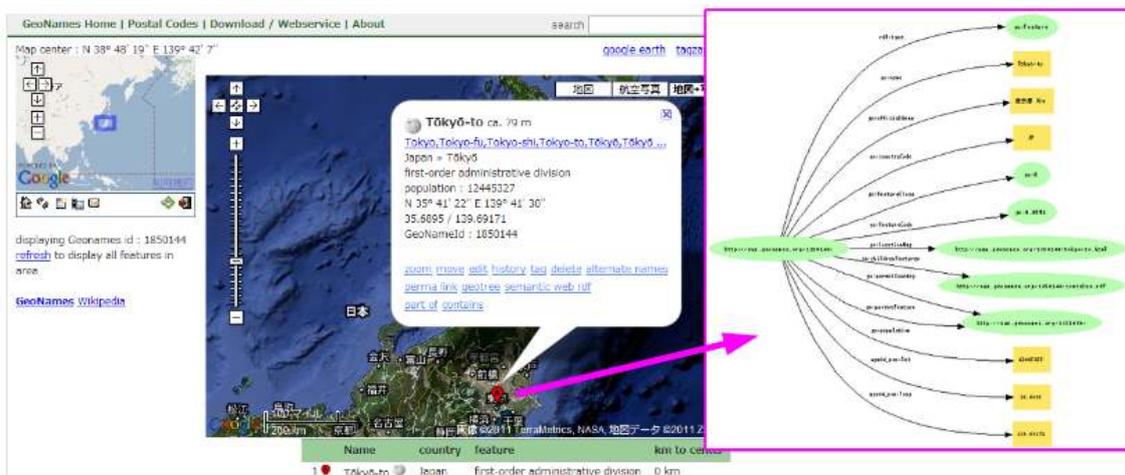
音楽データの場合、MusicBrainz が対象を識別する ID を提供し、データを互いに結びつけるハブとして機能している。では音楽以外のデータはどのような識別子(名前)を用いて「リンクするデータ」を形成することができるだろうか。

(a) GeoNames 地名であれば、GeoNames が同様の識別子を提供している。たとえば東京都ならば「1850144」という GeoNames ID が与えられ、この ID を用いてウェブページにアクセスすると地図とともに東京の人口や緯度経度などの情報が表示される。またこのページに対応する RDF が用意されており、これらの情報が RDF メタデータとして得られると同時に、近隣や広域の GeoNames ID 情報へのリンクも示される(図 16.17)。

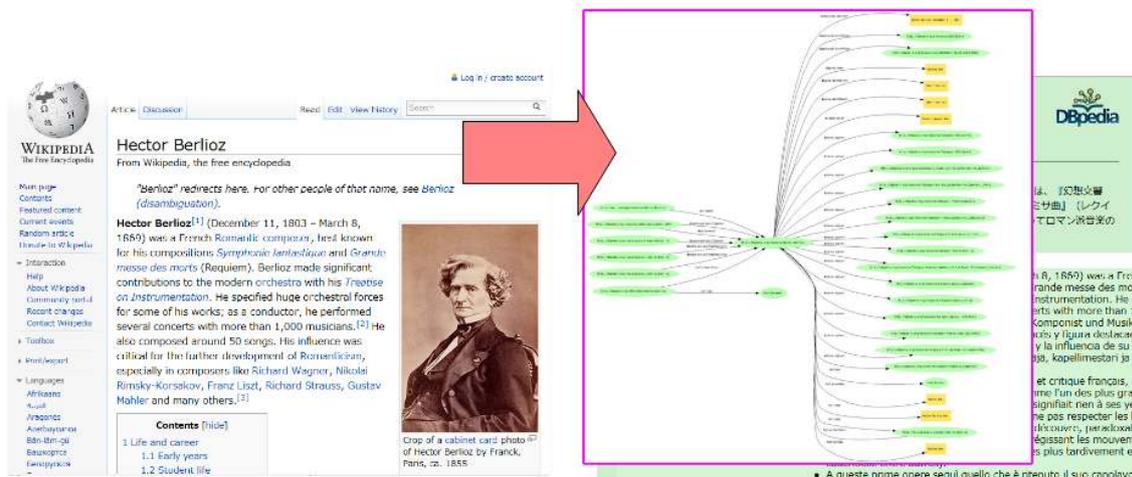
(b) DBpedia 音楽情報、地名などの分野ごとのハブではなく、幅広いテーマについて参照できる識別子(名前)があれば、リンクするデータの核となってさまざまなデータを結びつけることが可能になる。

ウェブ上で幅広いテーマを扱っているサイトといえば、Wikipedia が挙げられる。Wikipedia には 2011 年 3 月現在で 350 万以上の項目があり、これらが一定の規則で名前付けられた URI を持っている。この URI を識別に用いれば、一般的な情報のかなりの部分を共通の名前で表現し、リンクすることが可能になる。

ただ Wikipedia の情報は HTML であり、RDF の形にはなっていない。RDF アプリケーションは、Wikipedia を介して複数の情報を結びつけることはできても、そこからメタデータを取り出したり、先にリンクをたどっていくことができない。そこで、Wikipedia の情報ボックスのデータを RDF に変換し、Wikipedia 名に対応する形で公開したのが DBpedia である(図 16.18)。DBpedia のデータは、Wikipedia のカテゴリなどを RDF で表現して、関連する項目にリンクする。さらに他の LOD データセットとの関連付けも持っており、さまざまな形でデータが繋がっていく。



◆ 図 16.17: GeoNames の東京都ページと RDF グラフ

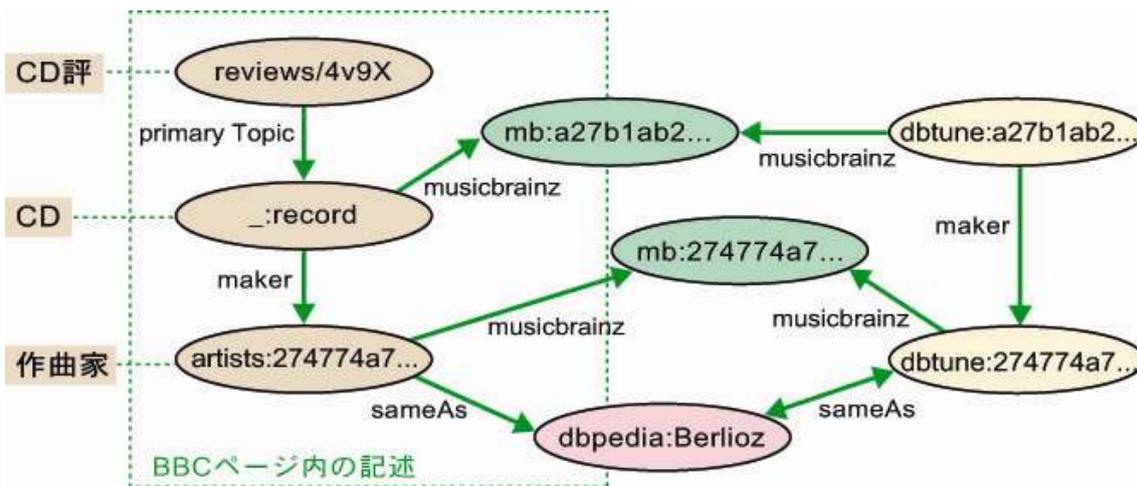


◆ 図 16.18: Wikipedia から RDF を抽出する DBpedia

この DBpedia の URI を用いてメタデータを記述すれば、

- 同じことがらについて、さまざまなところで記述される情報を、同じ名前で識別し、集約できる。さらに、
- 多くの情報が DBpedia 経由で結びつき、それぞれの情報が「リンクするデータ」に組み込まれる

ことなる。16.2.3 で述べた BBC の RDF は、この DBpedia へのリンクも含んでいる(図 16.19)。BBC のさまざまなジャンルの膨大な番組情報が、DBpedia を介してリンクするデータの輪に参加している。



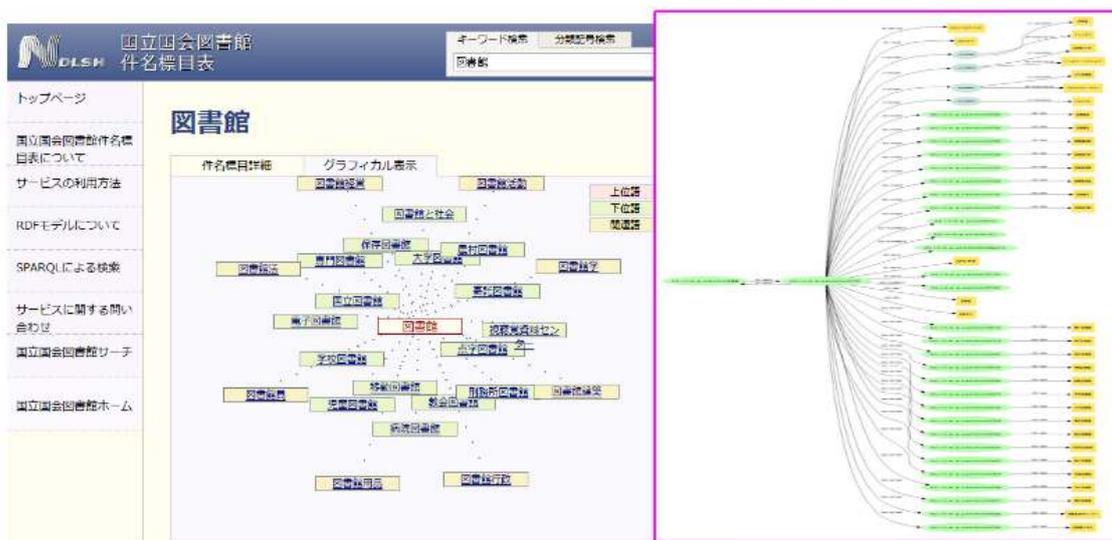
◆ 図 16.19: BBC の RDF は、サイト内の情報、MusicBrainz、DBpedia とつながる「リンクするデータ」

### 16.2.5 日本におけるリンクするデータ

LOD のつながりの中で、日本語によるデータは出遅れの感があったが、2010 年頃から徐々にデータをリンクさせる動きが出始めている。

(a) WebNDLSH 2010 年に国立国会図書館が主題件名標目表を RDF 化し(WebNDLSH)、リンクするデータとして公開した<sup>16</sup>。件名標目表は目録の内容を利用者が探すための手がかりとなる「標目」(「5.1.1.図書館分野のメタデータ記述規則」参照)として主題を体系化した統制語彙で、国立国会図書館では約 10 万の件名をシソーラス化している。

WebNDLSH はこの件名標目表を SKOS 語彙(「5.3.4.SKOS」参照)を用いて RDF 化し、上位語、下位語、関連語などのシソーラスのリンクを形成するとともに、米議会図書館が公開する LCSH の RDF 版にも関連付け、LOD の輪に加わった。ウェブブラウザで閲覧すると、標目の詳細データ表示や語の関連のグラフィカル表示が提供され(図 16.20 左)、RDF アプリケーションでアクセスするとその RDF グラフ(図 16.20 右)が得られる仕組みになっている。



◆ 図 16.20: 国立国会図書館の WebNDLSH では、語の関連をグラフィカル表示するほか、RDF データも得られる

(b) LODAC Museum 情報・システム研究機構新領域融合センタープロジェクトによって《学術リソースのためのオープンソーシャルセマンティック Web 基盤の構築》を目的として 2010 年に開始された LODAC(Linked Open Data for Academia)では、日本のミュージアム領域の多様な情報を情報を「リンクするデータ」として流通させる LODAC Museum<sup>17</sup>に取り組んでいる。

このプロジェクトでは、日本全国のミュージアム 14 館から収集したデータと、国指定文化財データベース、文化遺産オンラインなどの情報を、日本美術シソーラスと照らし合わせて統合する。そ

<sup>16</sup> NDLSH(国立国会図書館件名標目表), <http://id.ndl.go.jp/auth/ndlsh/>

<sup>17</sup> LODAC Museum, <http://lod.ac/>



(b) **記述規則の探索とガイドライン** ただし、登録スキーマを利用してデータを処理するためには、そのデータがどの記述規則に基づいているのかが分からなければならない。アプリケーションでの処理の前に、人間がレジストリの機能を用いて適用記述規則を検索することはできるが、リンクをたどって未知のデータを取り込んでいくためには、アプリケーション自らが記述規則を確認するための標準方法が必要である。ガイドラインにおいて「スキーマを利用したメタデータ共有・活用」として示した指針に基づいてメタデータを公開することで、記述規則の発見が可能となり、リンクするデータの処理が進むことになる。

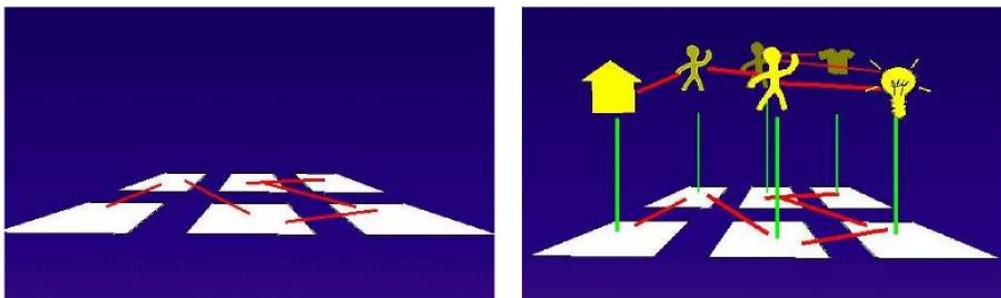
### 16.3 セマンティック・マーク付けとレジストリ

HTML で記述されたウェブ・コンテンツをリンクするデータに組み込み、新しいコンテンツ共有の可能性を開く方法である「セマンティック・マーク付け」の展望と、そこでレジストリが果たす役割を考察する。

#### 16.3.1 セマンティック・マーク付け

RDF で記述されるメタデータも増えつつあるとはいえ、ウェブのコンテンツの大半は HTML 形式で提供されるウェブページである。企業カタログなど、データが重要なコンテンツでもユーザ向けには HTML のページしか提供されない場合が多い。

1994 年に開かれた初の WWW 国際会議でバーナーズリーは、図 16.22 を示しながら、ハイパーテキストのウェブは人間にとってはエキサイティングだがコンピュータにとっては平坦で意味を持たないもの(左)であり、文書中の情報をコンピュータで読める形式にしてリンクの意味を表現できるようにすれば(右)プログラムがそれを有効に利用できることを述べていた。



◆ 図 16.22: コンピュータに理解可能なリンクの表現を持ったウェブ

HTML ページとは別に RDF ファイルを提供するには余分な手間がかかるし、メンテナンスも難しい。そこで、HTML に含まれているデータ部分に特別なマーク付けを施し、それを直接 RDF としても取り出せるようにする記述方法、**セマンティック・マーク付け**が注目されるようになる。

(a) **Microformat** HTML には要素の役割を細分化するための class 属性が用意されている。一般にはページ作者や作成組織が任意の語をこの属性に割り当て、スタイルシートによるデザイン用

のページ部分識別子(セレクト)として用いられているが、この class 属性の名前付けと解釈をコミュニティで共有し、HTML 文書内に埋め込まれたデータを明示しようとするのがマイクロフォーマット (Microformat) である<sup>18</sup>。

たとえば簡単なレシピを紹介するページにおいて、その材料や手順を hRecipe と呼ぶ class 属性値のグループによって示し、ツールで抽出可能にする。

```
<div class="hrecipe">
  <span class="item">
    <h1 class="fn">枝豆</h1>
  </span>
  
  <span class="summary">簡単にできるおつまみの定番。</span>
  材料:
  <span class="ingredient">
    <span class="name">枝豆</span>:
    <span class="amount">1 パック分</span>
  </span>
  <span class="ingredient">
    <span class="name">塩</span>:
    <span class="amount">大さじ 2 杯</span>
  </span>

  作り方:
  <span class="instructions">
    1. 枝豆ははさみでさやを枝から切りおとし、塩もみして、うぶ毛を取る...
    2. 鍋に 1 リットルほどのお湯をわかし、塩がついた枝豆を茹でる...
    ...
  </span>
</div>
```

マイクロフォーマットは、イベントをマーク付けする hCalendar、連絡先を表す hCard など、よく使われるデータに関するフォーマットが用意されている。簡単で直感的に利用できるが、単語による class 属性値では名前の衝突が生じて意味が不明瞭になる恐れがあること、任意の拡張ができないことなどの限界もある。

(b) Microdata 既存の属性を流用するのではなく、HTML の新バージョン HTML5 の策定の一

<sup>18</sup> Microformats, <http://microformats.org/>

環として、データマーク付けのための新たな属性を導入する仕様が Microdata である。<sup>19</sup>。

itemscope 属性を持つ要素内をひとつの「アイテム」として扱い、その型を itemtype 属性値で宣言する。アイテムの持つプロパティは itemprop 属性によって示される。プロパティの意味はアイテムの型によって定まるので、同音異義語によって解釈が曖昧になることはない。

同じレシピを Microdata で記述すると、次のようになる。

```
<div itemscope itemtype="http://data-vocabulary.org/Recipe">
  <h1 itemprop="name">枝豆</h1>
  
  <span itemprop="summary">簡単にできるおつまみの定番。</span>
  材料:
  <span itemprop="ingredient" itemscope
    itemtype="http://data-vocabulary.org/RecipeIngredient">
    <span itemprop="name">枝豆</span>:
    <span itemprop="amount">1 パック分</span>
  </span>
  <span itemprop="ingredient" itemscope
    itemtype="http://data-vocabulary.org/RecipeIngredient">
    <span itemprop="name">塩</span>:
    <span itemprop="amount">大さじ 2 杯</span>
  </span>

  作り方:
  <div itemprop="instructions">
    1. 枝豆ははさみでさやを枝から切りおとし、塩もみして、うぶ毛を取る...
    2. 鍋に 1 リットルほどのお湯をわかし、塩がついた枝豆を茹でる...
    ...
  </div>
</div>
```

Microdata のマーク付けから、Microformat に変換したり、RDF グラフを抽出する方法も仕様で定義されている。

(c) **RDFa** HTML の属性を用いて文書中に RDF を直接埋め込むフォーマットが RDFa (a は attribute=属性) である<sup>20</sup>。使用する語彙名前空間 URI を接頭辞にマッピングした上で、プロパティ

<sup>19</sup> HTML Microdata, <http://www.w3.org/TR/microdata/>

<sup>20</sup> RDFa in XHTML Syntax and Processing, <http://www.w3.org/TR/rdfa-syntax/>

を XML の短縮名と似た CURIE (Compact URI)<sup>21</sup>で示す。

- プロパティは、リテラル目的語の場合は property 属性に、非リテラルの場合は rel 属性に記述する
- 目的語は、リテラルの場合は要素内容のテキスト、もしくは content 属性値となる。非リテラルの場合は、href 属性(もしくは src 属性、resource 属性)に値を記述する
- rel 属性を持つ要素内にさらに RDFa の属性を持つ子孫要素がある場合、RDF グラフを連結した入れ子の記述ができる
- リソースが型を持つときは typeof 属性で指定する

同じレシピを RDFa で記述すると、次のようになる。

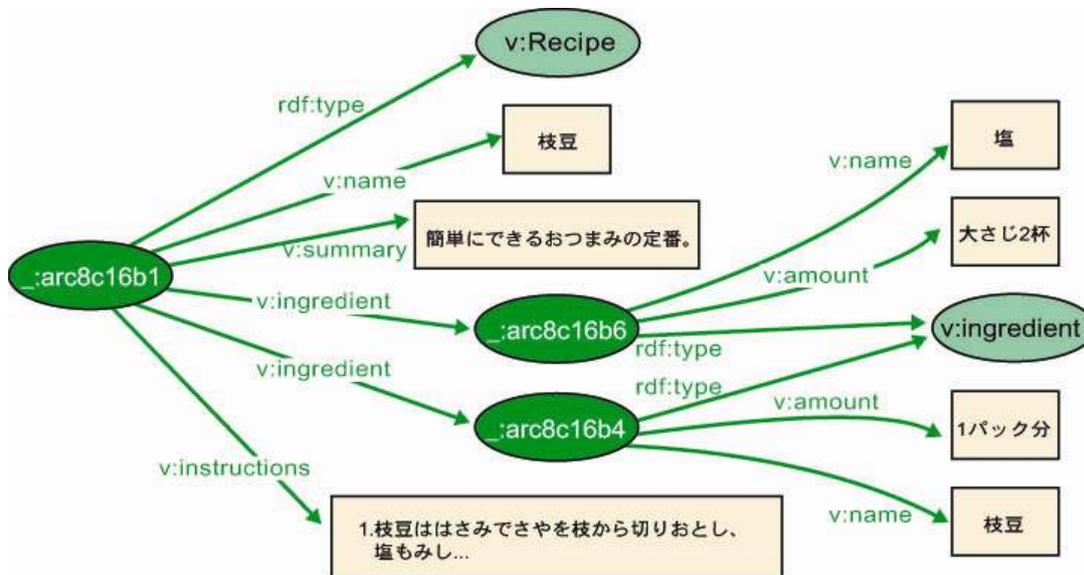
```
<div xmlns:v="http://rdf.data-vocabulary.org/#" typeof="v:Recipe">
  <h1 property="v:name">枝豆</h1>
  
  <span property="v:summary">簡単にできるおつまみの定番。</span>
  材料:
  <span rel="v:ingredient">
    <span typeof="v:Ingredient">
      <span property="v:name">枝豆</span>:
      <span property="v:amount">1 パック分</span>
    </span>
  </span>
  <span rel="v:ingredient">
    <span typeof="v:Ingredient">
      <span property="v:name">塩</span>:
      <span property="v:amount">大さじ 2 杯</span>
    </span>
  </span>

  作り方:
  <div property="v:instructions">
    1. 枝豆ははさみでさやを枝から切りおとし、塩もみして、うぶ毛を取る...
    2. 鍋に 1 リットルほどのお湯をわかし、塩がついた枝豆を茹でる...
    ...
  </div>
</div>
```

CURIE を用いることで複数の語彙を混在させることができ、入れ子の規則により複雑なグラフ

<sup>21</sup> CURIE Syntax 1\_0, <http://www.w3.org/TR/curie/>

を比較的簡単に記述することができる。上の RDFa 記述例からは、図 16.23 の RDF グラフが得られる。



◆ 図 16.23: レシピ RDFa から取り出したグラフ

### 16.3.2 検索エンジンとセマンティック・マーク付け

検索エンジンは、16.1.2 でも触れたとおり、より精度の高い検索結果を提供するためにこのセマンティック・マーク付けを積極的に利用し始めている。Google は、リッチスニペットと呼ぶ付加価値情報を検索結果に追加し、セマンティック・マーク付けから得られたレビューの評価などを表示している。また前節の例に挙げたレシピのマーク付けを利用してレシピ検索専用の「Google Recipe」を展開し、材料、調理時間、カロリーなどの条件による絞込みを提供している(図 16.24)。

こうした詳細な情報や検索条件は、従来の HTML から正確に抽出することは困難である。通常の HTML に含まれるデータを、人間の利用者向けに表示するだけでなく、そこからアプリケーションがメタデータを抽出して処理できるようにすることで、情報の活用の範囲が大きく広がっている。

The screenshot shows a Google search for '枝豆' (Edamame). The search bar contains '枝豆' and the search button says '検索'. Below the search bar, it indicates '約 11,000 件 (0.12 秒)' and a '検索オプション' link. The left sidebar shows navigation options like 'すべて', '画像', '動画', 'ニュース', 'ショッピング', 'レシピ', and 'もっと見る'. Below that, there are filters for 'ウェブ全体から検索' and '日本語のページを検索', along with a list of ingredients (材料) with checkboxes for '有' (Yes) and '無' (No). The main content area displays search results for recipes, including '枝豆の美味しい茹で方!!', '叩き枝豆と夏野菜のパスタ', and '枝豆腐'.

◆ 図 16.24: セマンティック・マーク付けのデータを利用した Google Recipe

### 16.3.3 セマンティック・マーク付けとスキーマ

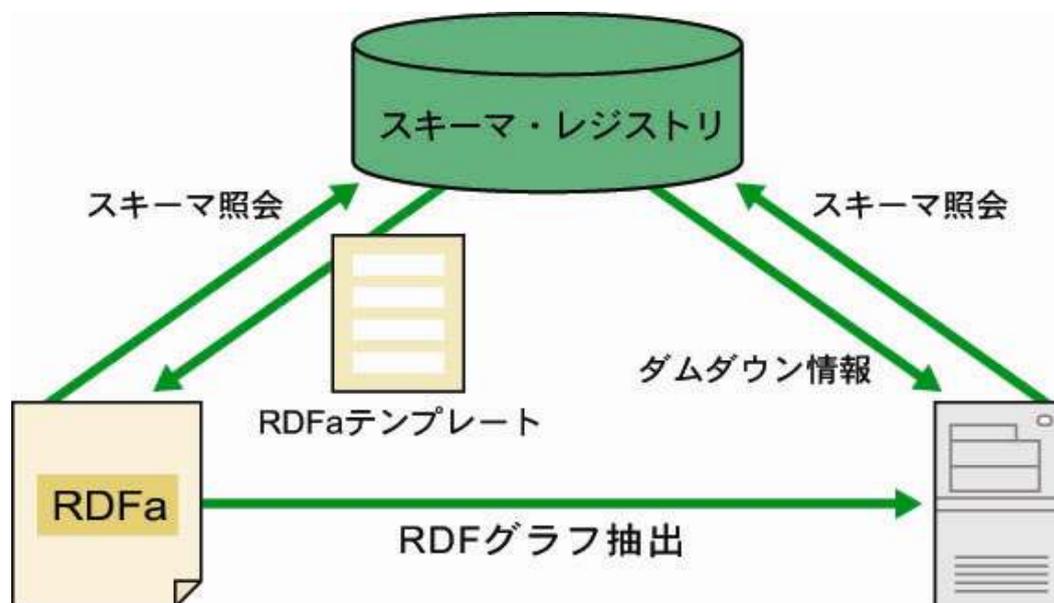
セマンティック・マーク付けでは、HTML の中に(メタ)データを記述する。文書中のデータをマーク付けするために、メタデータの専門家でないページ作者が、どのような語彙を使うべきなのかをきちんと理解して RDFa などを記述するのは容易ではない。また、HTML ページの情報は、料理レシピや商品スペックなど、表やリストを用いて書けるものばかりではなく、段落中の平文にデータが分散している場合もある。適切なデータ構造に対応したマーク付けは、十分な知識なしには難しい。

セマンティック・マーク付けで、HTML 文書からメタデータを得る仕組みが用意され、データを共有する可能性は大きく広がった。しかし、非専門家も含めた HTML 作者によるマーク付けでメタデータを共有し再利用するためには、従来以上に多くの課題を解決する必要がある。

(a) レジストリとセマンティック・マーク付けテンプレート ウェブログなどのツールには、情報を一定の構造で容易に記述できるようにする「テンプレート」の機能が用意されている。フォームの形で示される入力欄に必要項目を埋めると、規則に従って構造化された HTML が出力され、ページの中に組み込まれる仕組みだ。

メタデータ・スキーマ・レジストリに登録された記述規則を利用すれば、必要なデータを入力し、そこから記述規則に準拠した RDFa マーク付けを生成するためのテンプレートを作成するサービスが提供できるだろう。API を用いて動的にレジストリに問い合わせ、必要な箇所に RDFa を埋め込むようなブラウザのプラグイン・拡張機能を作ることでもできる(図 16.25)。

ウェブ上には、有用な情報を含む膨大な HTML ページが存在している。ここからメタデータを抽出するために、従来はスクリーン・スクレイピング（「4.1.1. 公開メタデータとマーク付け」参照）などの手法が用いられてきた。



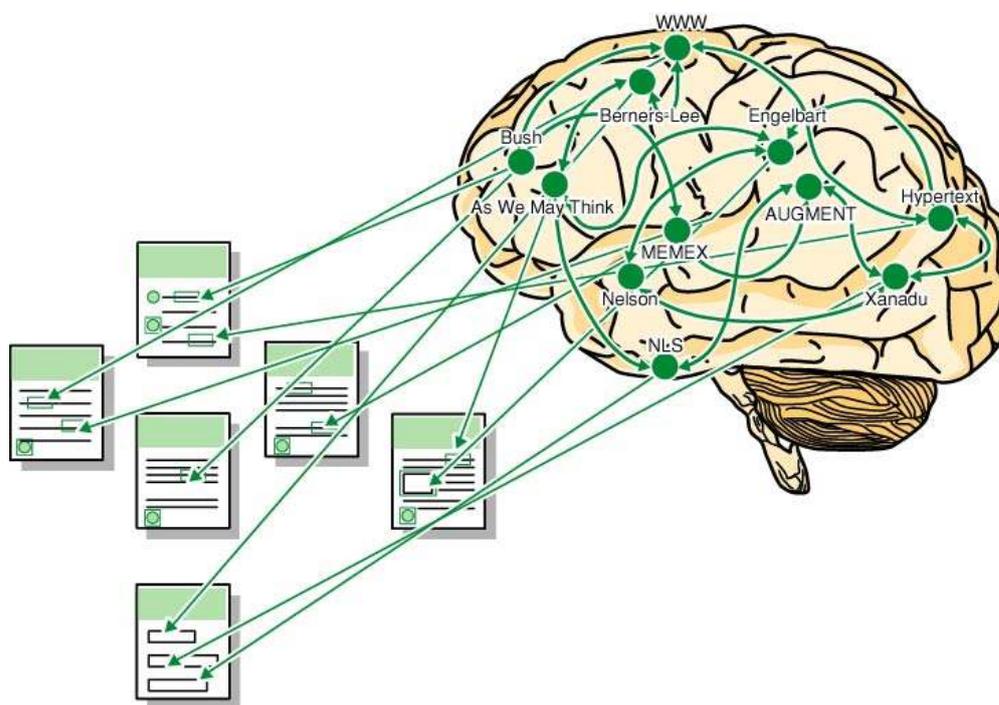
◆ 図 16.25: スキーマ・レジストリを利用して、RDFa の生成をサポートしたり、抽出したメタデータをダム・ダウンできる

セマンティック・マーク付けによるウェブページが増加すれば、活用できるメタデータは飛躍的に増大する。そのときに、またメタデータの共有を最初からやり直すことにならないよう、標準スキーマを分かりやすい方法で提供していくことが重要である。

## 16.4 トピックマップがもたらす可能性

### 16.4.1 可能性を広げるためのシステム構造

トピックマップは、主題を体系化、構造化し、主題に関係する情報リソースへのリンクをもつ。主題中心に情報の体系化、知識化をするため、Subject-centric computing (主題中心のコンピュータ処理) と呼ばれている。文書中心の考え方 (Document centric) からのパラダイムシフトである。Subject-centric computing のイメージを図 16.26 に示す。



◆ 図 16.26: Subject-centric computing

出典: Steve Pepper, As We Really May Think,

<http://www.knowledge-synergy.com/news/atoms2007-presentations-en.html>

正確な主題識別、および、主題間の構造に基づいて、主題を結合点として情報を有機的に連結可能にする。現在認識している範囲で、Subject-centric computing を実現するために必要な機能要素を以下に示す。

(a) PSI サーバ/クリアリングハウス 思考、検索の対象となりうるすべての主題(概念)に、IRI を割当て、PSI(Published Subject Identifier) として公開することが必要になる。

PSI サーバ は、各種 PSI を登録し、Web 上で、誰からも参照可能にしているサーバ であり、以下のものがよく知られている。

## subj3ct.com

<https://subj3ct.com/>

NetworkedPlanet 社によって、立ち上げられ、運営されているサーバである(図 16.27)。自身について、「Subj3ct は、セマンティック Web のための、主題識別子の登録及び情報交換所サービスである。」と述べている。Subj3ct は、登録された主題識別子と関係する情報リソースを見つけるための Web サービスを提供している。Subj3ct には、2011 年 2 月 15 日時点で、48 の提供サイトから 17,259,386 の主題が登録されている。

例えば、subj3ct.com を "Motomu Naito" で検索すると

"http://psi.ontopedia.net/Motomu\_Naito", "http://www.topicmapslab.de/people/Motomo\_Naito" などが表示される。これらの IRI は、"内藤求" という主題(この場合、実在の人物が主題である)を識別するための公開主題識別子(PSI: Published Subject Identifier)である。トピックマップの作成者が、その識別子に付随する記述を読んで、自分の概念と最も合致する PSI を選び、トピックに割り当てる。

subj3ct.com の設立、運営者である Graham Moore と Khalil Ahmed は、PSI サーバの主要な機能として以下の 3 点を挙げている。

- 同じ主題を示している Subject Identifier の宣言
- 主題に関する情報リソースの発見
- 信頼性の評価と出所の明示

HOME ABOUT BLOG COMMUNITY CONTACT [ Sign In ] or [ Register ]

subj3ct<sup>beta</sup>  
NetworkedPlanet

Search for subjects... Search

### The Subjct Centric Web

Subj3ct is an infrastructure technology for Web 3.0 applications. These are applications that are organised around **subjects** and semantics rather than documents and links. Subj3ct provides the technology and services to enable Web 3.0 applications to define and exchange subject definitions.

Feeds: 48  
Subjects: 17265672

Random Subjects

- Tuili
- Ussassai
- ....09125,13.63068;crs=wgs84
- Montefortino
- La Maddalena

### Register

While most aspects of Subj3ct don't require you to sign in, creating an account will improve the trust that other users have in the identifiers you register. Having an account also means you can add identifiers directly into your personal Subj3ct namespace.

[Create a New Account](#)

### Contribute

Subj3ct allows users to register identifier feeds. Each feed can contain as many identifiers, for as many different subjects as you want. The feed registration process is simple to use and feeds are easy to build.

Read our [Quick-Start Guide](#) for more information.

### Search

### Develop

◆ 図 16.27: subj3ct.com (subj3ct-com1a.png)

## ONTOPIEDIA

<http://psi.ontopedia.net/>

ONTOPIEDIA 社 によって開設され、運営されているサーバである(図 16.28)。例えば、人、イベント、場所、発表、組織、規格など、多くの PSI と PSD を管理していて、その管理対象を日々拡張している。このサーバを利用して、主題と PSI を検索することができる。このサーバは、自身について、「公開主題識別子を定義、表示できるサーバ」と述べている。登録されている PSI の数は公表されていない。

IRI は、"http://psi.ontopedia.net/XXXXXXXX" の形になっている。例えば、"内藤 求 (Motomu Naito)"の PSI は、"http://psi.ontopedia.net/Motomu\_Naito" である。

今後は、日本においても複数の PSI サーバが立ち上がり、国外の PSI も含めて、登録内容の交換、更新など、連携可能になっていくものと思われる。

(b) トピックマップ Web アプリケーション 個別または複数の領域、テーマを扱い、出所が明確で、安定した品質をもつ情報リソースを扱うトピックマップ Web アプリケーションが、ますます増加していくものと思われる。それらは、例えば、辞書、事典や件名標目、学習支援、コンテンツ、各種 Web ポータル、行政サービス、知的財産、製品、製造などに関わる情報を管理し提供するア

アプリケーションである。

User: anonymous [login](#)

# ONTOPEdia

Home
Project | PSI changes: RSS 2.0, Atom

Find concept, resource or assertion

*Currently we have a small number of subject identifiers in the system. Our main goal is to investigate various approaches related to building and maintaining a useful PSI Server*

Subject stream
Concepts
Web resources
Assertions

### Subject stream

tibbr	2011-01-24
Introduction of tibbr by Tibco	2011-01-24
The Game Has Changed (track in TRON:Legacy soundtrack by Daft Punk)	2010-12-18
Overture (track in TRON:Legacy soundtrack by Daft Punk)	2010-12-18
Finale (track in TRON:Legacy soundtrack by Daft Punk)	2010-12-18
Adagio for TRON (track in TRON:Legacy soundtrack by Daft Punk)	2010-12-14
Patrick Durusau	2010-12-11
Singularity University	2010-12-09
SDShare	2010-11-30
Apple's "Back to the Mac" event	2010-10-20

« Previous 1 2 3 4 5 6 7 8 9 ... 117 118 Next »

◆ 図 16.28: ONTOPEdia

これらのアプリケーションの中には、主題についてのクライアントからの要求に応じて、自分が保有するその主題についての情報を提供するような Web サービス機能をもつものも表れていて、今後増えていくと思われる。

(c) トピックマップ共有/交換プロトコル トピックマップ Web サービス、トピックマップの共有/交換のためのプロトコルである。これにより、分散して存在しているトピックマップの間で、トピックマップフラグメントの問合せ、交換等が可能になる。

- TMRAP (Topic Maps Remote Access Protocol)

TMRAP は、Ontopia に含まれている Web サービスインターフェースで、HTTP と SOAP に対応している。メソッドとして、以下のものがあり、Web アプリケーション間でトピックマップフラグメントを交換することができる。

- get-topic
- get-topic-page
- get-tolog
- add-fragment
- delete-topic

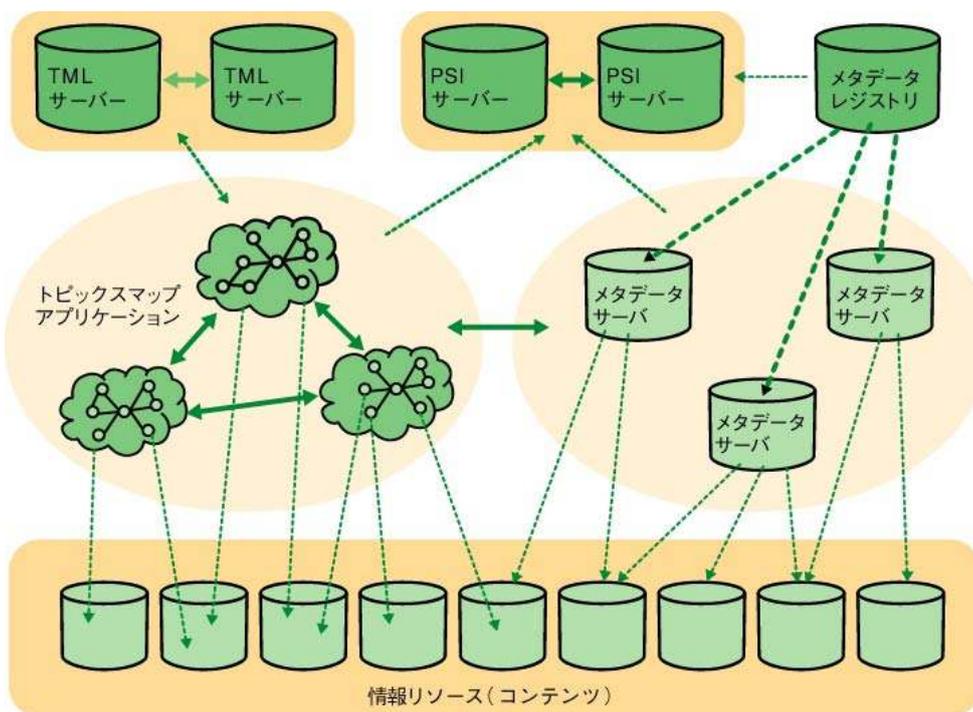
- SDShare: Protocol for the Syndication of Semantic Descriptions

ISO/IEC SC34 WG3 標準化が始まろうとしているセマンティック記述の交換のための基本的な配信プロトコルである。このプロトコルは、Atom 配信フォーマットとトピックマップデータモデル (TMDM) に準拠し、XTM 1.0、XTM 2.0 及び RDF/XML で表現されたセマンティック記述を用いる。

(d) TML サーバ: Topic Maps Link Server (トピックマップ連携サーバ) どこに、どのようなトピックマップ Web アプリケーション、Web サービス、メタデータサービスが存在しているか管理する。他の TML サーバとも連携し、所在情報の交換、更新をする。

クライアント(トピックマップ Web アプリケーション)と、サーバ(トピックマップ Web アプリケーション)の間を仲介して、主題に基づくトピックマップ Web アプリケーションの連携を可能にする。

(e) 全体像 これまでに述べてきた機能要素を組み合わせた場合の全体のシステム構造を図 16.29 に示す。



◆ 図 16.29: 「識別された主題のネットワーク」

ここでは、これを「識別された主題のネットワーク(Identified Subjects Network)」または、「トピックマップネットワーク」と呼ぶこととする。「識別された主題のネットワーク」においては、多様なトピックマップ Web アプリケーションやメタデータ Web アプリケーションが TML サーバを通じて連携し、

利用者の要求に合致する適切な情報リソースやサービスを提供する。

情報リソースに対して、管理体系を含め、様々な体系、視点からトピックマップを作成することができる。トピックマップの中のトピック(主題)には、PSI サーバを通して PSI が付与され、正確な主題の識別が可能になっていて、主題に関係する情報リソースとも結び付けられている。各 Web アプリケーションは、主題に基づき、自分たちが保有する情報リソースを維持、管理し、同一トピック(主題)について他の Web アプリケーションと情報を交換し、情報リソースを共有できる。

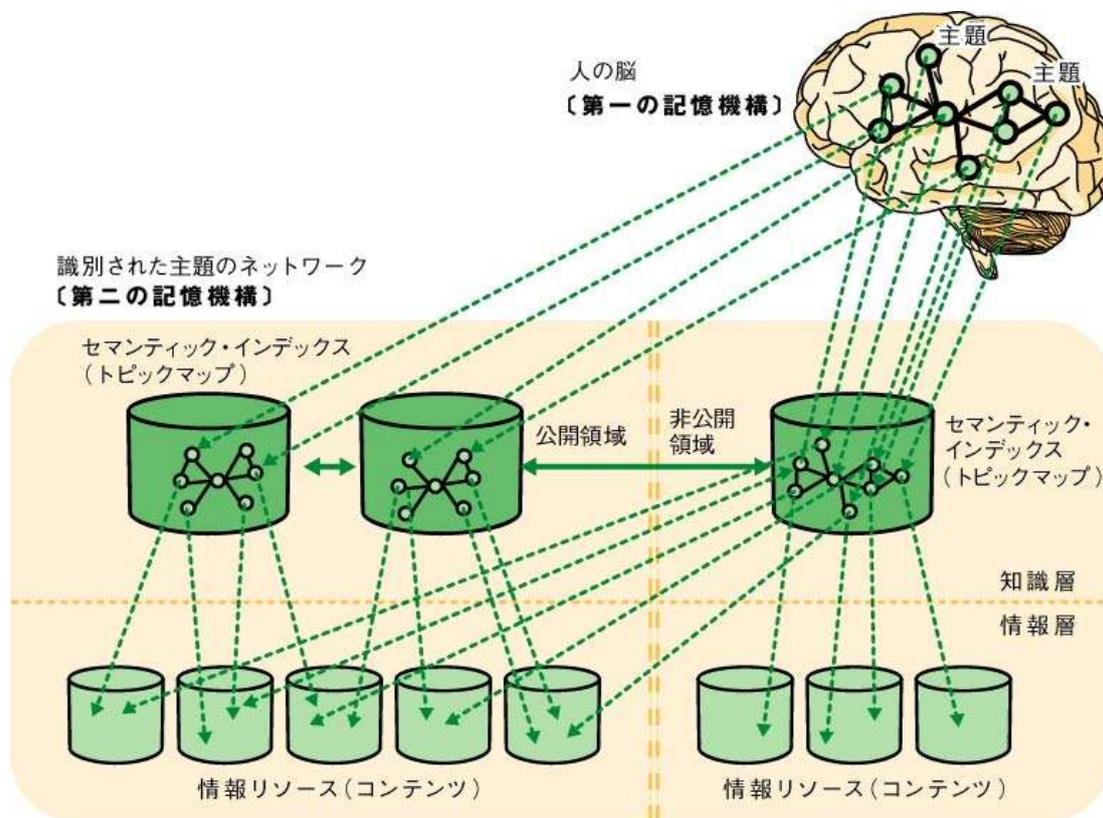
利用者は、興味のある主題とそれに関連する主題の間を自由にナビゲートでき、ネットワーク上に分散して存在している主題についての情報、情報リソースをワンストップ・ショッピングできるようになっている。

#### 16.4.2 可能性の切り口

(a) **セマンティック・インデクシング** トピックマップは、意味に基づいて主題の識別が可能であり、主題間の関係にそって、主題間をナビゲート可能にする。そして、主題に関係する情報リソースへのリンクをもつ。あたかもサイバースペース上の意味的なインデックスのような役割をする。それをセマンティック・インデックスと呼ぶことにする。

個人、集団/組織の概念体系、特性、好み、要求を加味して、いろいろな領域、文脈に合わせ、同一の情報リソース/知識に対しても複数視点からセマンティック・インデックス作成することができる。

(b) **第二の記憶機構** 前述したシステム構造、「識別された主題のネットワーク」、および、セマンティック・インデックスの実現により、人間は、自身の記憶能力の他に、「識別された主題のネットワーク」を、もう一つの記憶機構として利用できるようになる。人がもつ概念および概念間の関係を整理、体系化し、この記憶機構に写像することにより、コンピュータ・ネットワークと直観的、有機的に連携することができるようになる。これを第二の記憶機構と呼ぶこととする。図 16.30 に第二の記憶機構のイメージを示す。



◆ 図 16.30: 第二の記憶機構 (double-storage.png)

「識別された主題のネットワーク」は、大きくは、非公開領域と公開領域に分ける必要があると考えられる。非公開領域は、第二の記憶機構として個人または組織に帰属する知識/情報リソースを格納、管理、再利用し、反芻的に成熟させるための領域である。一方、公開領域には、現在のインターネットのように広く万人によって発信され、利用可能な知識/情報リソースが存在する。公開領域はさらに、特定多数の人に利用が限定された領域と、不特定多数の人が利用可能な領域に分けて考える必要がある。

「識別された主題のネットワーク」は、人間にとっての第二の記憶機構として、現在のコンピュータ・ネットワークに比べて、より有機的、相乗的に利用可能になると考える。個人、集団/組織の知識を永続的に蓄積、共有し、知識活動を活性化し、知識の熟成度を高めて、精神的な豊かさをもたらし、相互理解を促進する可能性を持っていると思われる。そこでは、知的活動の対象としてのコンテンツの重要性がますます高まるものと思われる。

また、直観力、理解力、創造力など人間の脳が持つ特長を活かしつつ、小容量で揮発しやすく、誤りやすいなどの短所を、正確、高速、大容量のコンピュータパワーで補完することができる。以下の表 表 16.1 に、人間とコンピュータが持つ能力の比較を示す。

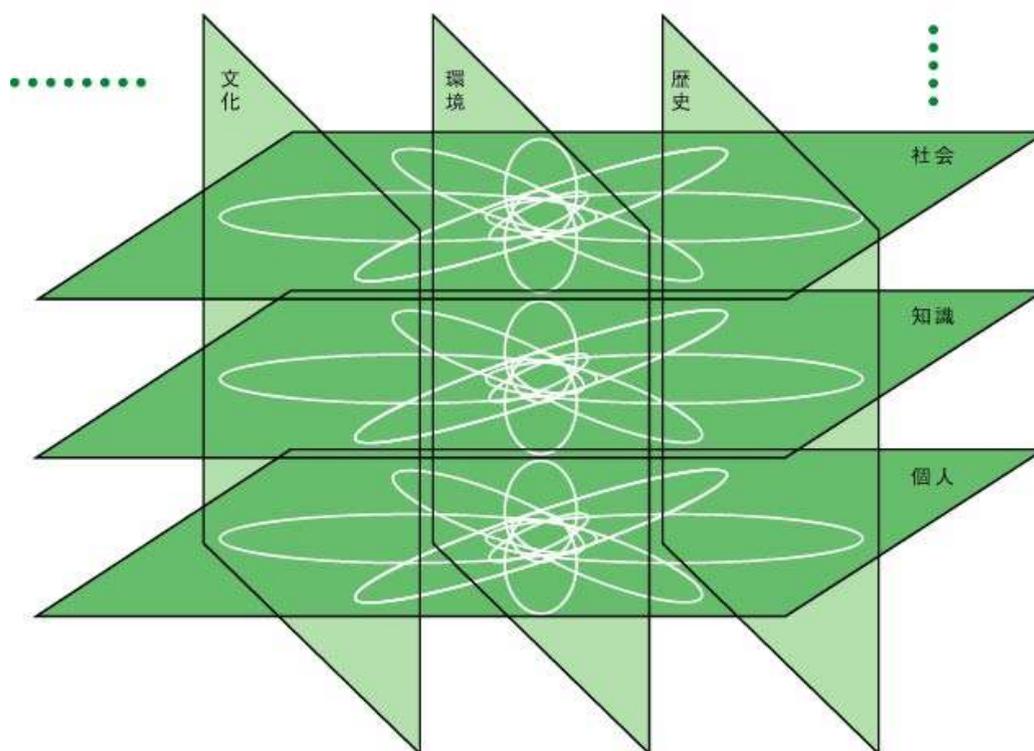
◆ 表 16.1:人間とコンピュータの特徴比較

項目	人間	コンピュータ
思考の特徴	帰納的、発見的(inductive) 全体的意味の把握	演繹的(deductive) アルゴリズム(解法手順)に従った処理
演算能力	低速、誤りやすい 連続作業による能率の低下	高速、正確 能率は常に一定
記憶能力	小容量、誤りやすい 連想	大容量、正確 位置による呼出し
その他	単純な反復は苦手 動機付けによる能力の向上	単純な反復が得意 アルゴリズムの改良による能率の向上

(c) **多次元・多層的な知識空間** ネットワーク上に存在する情報/知識空間は、多次元、多層的であり、ほとんど無限の広がりを持っている。現実世界を反映して非常に複雑な様相を呈している(図 16.31)。

現在の検索システムは、混沌として分類されていない情報/知識空間をまるごと相手にしている。まったくの異分野から予想外の発見をする可能性がある半面、「必要なときに、ピンポイントに領域、文脈を絞り、必要な情報/知識のみを的確に見つけ出すことが困難」という欠点を持っている。

利用者にとって、特定の状況において必要な情報/知識空間は、ネットワーク上の無制限の情報/知識空間から、ごく狭い範囲を切り出したものと考えられる。切り出す範囲は、対象とする領域、文脈、アスペクト、視点など、様々な条件、タイミング、場所、さらに、作成者、所有者、管理者、利用者、消費者など個人が立脚する状況に依存する。限定された範囲のみを対象とすることで、情報/知識の見つけやすさ、利用しやすさ、更新のしやすさ等が格段に向上する。



◆ 図 16.31: 多次元・多層的な知識空間のイメージ (multi-dimension.png)

したがって、情報/知識空間を分割、分類し、識別可能し、必要に応じて選択可能にすることが非常に重要な課題になる。さらに、Published Subject Identifier のような個々の空間に存在する主題を、異なる空間にまたがって同定可能にする仕組みの実現が不可欠になる。

人とネットワークから構成されるシナジー知識空間を、作成者と利用者の視点から見た場合、例えば、以下のように分類することができる。

1. 作成者が個人で、利用者が作成者と同一である場合
2. 作成者が個人で、利用者が特定多数の人(集団/組織)である場合
3. 作成者が個人で、利用者が不特定多数の人である場合
4. 作成者が集団/組織であり、利用者が個人である場合
5. 作成者が集団/組織であり、利用者が特定多数の人(集団/組織)である場合
6. 作成者が集団/組織であり、利用者が不特定多数の人である場合

上記のそれぞれに、自然史・歴史・民族・文化・美術・科学・技術・交通・海事・航空・軍事・平和などの領域、そして、小学生用・中学生用・高校生用・大学生用・研究者用などのレベルが交差し細分化される。

### 16.4.3 利用イメージ

トピックマップを用いることにより、「識別された主題のネットワーク」と人の相互作用を円滑にし、個人、組織に帰属する情報/知識を、個人、組織の思考体系、分類体系に合わせて組織化することが可能になると考えられる。情報/知識の入力、蓄積、取り出し、再利用が容易になり、人の思考活動を活発にし、効率、創造力を高め、知識の熟成を支援するような環境を実現できると考える。

(a) **個人の知的活動支援** 個人に帰属する情報/知識には、文字通りプライベートなもの、所属する集団/組織の他の構成要員との間で共有されるものがある。それらは、外部に公開されるものとされないものに分けることができる。

個人に帰属する情報/知識は、具体的には、個人経歴にはじまり、医療、教育、生活、思想、哲学、資産、趣味、専門知識、研究日誌、論文、プレゼンテーション資料、デジタル写真などなど、実に様々なものが考えられる。それらを主題別に分類、整理体系化するセマンティック・インデクシングを構築する。そして、セマンティック・インデクシングを通して、いろいろな視点からの情報/知識へのアクセスや更新を可能にする。

例えば、プレゼンテーション資料について考えれば、過去に行ったプレゼンテーションの資料を主題ベースに分類、体系化し、セマンティック・インデクシングを介してアクセス、更新可能にすることにより、新規プレゼンテーションの構成を容易にし、過去の反省をもとに改善、進化させることが容易になる。このようにして、個人の知識資産を、質、量ともに充実させていくことができると考える。

(b) **集団/組織の活動支援** 集団/組織に帰属する情報/知識は、集団/組織に依存して千差万別であり、個人にもものより、はるかに多様と考えられる。また、外部に公開されるものとされないものに分けることができる。それらについても、主題別に分類、整理体系化するセマンティック・インデクシングを構築する。そして、セマンティック・インデクシングを通して、いろいろな視点からアクセス、更新、共有、交換にする。

例えば、組織の研究活動においても、研究対象をテーマ別、主題別に情報/知識を整理体系化し、回を重ねながら、それらを洗練、成熟させ、全体としての構造に磨きをかけていくような運用が可能になると考えられる。

(c) **最新事例に見る利用イメージ** トピックマップ(「4.4.その他のデータモデル「トピックマップ」参照)における適用事例から、以下の利用イメージが考えられる。

- 1.教育機関による学習トピックの体系化と学習コンテンツの提供
- 2.図書館、研究機関などが作成、維持するシソーラスに基づくデジタルコンテンツの体系化と提供

3. 特定分野に特化した適用事例、発表事例および関連資料の提供
4. 個人所有のデジタルコンテンツの体系化、公開
5. 美術館、博物館などが所有するデジタルコンテンツの体系化、公開、販売
6. 個人が学習を通して獲得した概念の体系化、維持の支援
7. 学習カリキュラムの学習目標を達成するために有効な放送番組の提供
8. 行政のサービス向上のための行政サービスポータル運営

#### 16.4.4 ビジネス機会

(a) **コンテンツ保有者** まず、コンテンツ保有者、管理者は、自らが保有、管理しているコンテンツに対して、デジタルコンテンツはそのものを、そうでないものはデジタル化してネットワーク上に置き、利用可能にする必要がある。それらに対してメタデータを付与し、セマンティック・インデックスを作成し、様々な視点からアクセス可能するとともに、その存在を広く発信していく必要がある。そうすることにより、コンテンツにアクセス、利用されるケースが増大していくものと考えられる。デジタルコンテンツの Web 上での販売もちろん、魅力的なコンテンツを増やしていく努力も欠かせない。魅力的なデジタルコンテンツについては、Web 上での販売も検討の余地がある。コンテンツの粒度についても、より細かい粒度での提供も検討していく必要がある。

(b) **コンテンツの流通業者** 利用者のニーズに合わせて、様々なテーマでセマンティック・インデックスを作成し、複数のコンテンツ保有者が提供しているコンテンツを、横断的に提供していくことができるようになる。

コンテンツへの客観的な評価情報を付与したり、いろいろな視点からコンテンツ間の関係を見えるようにすることなどにより、提供サービスに付加価値を付けることができる。魅力的で有益なコンテンツを増やし、付加価値を付けることにより、流通する量を増大させ、販売量を増やすことができると考えられる。

(c) **コンテンツ製作者** 音楽、映画、コミック、ゲーム、アプリケーションソフトウェア、ニュースなど、魅力的なコンテンツを供給すれば、コンテンツの価値を高め、流通、販売量の増加に繋げることができると考えられる。

(d) **ソフトウェア開発者** 「識別された主題のネットワーク」環境構築、メタデータ、トピックマップ作成のニーズが高まり、その中において主題中心のコンピュータ利用を実現するためのツール、アプリケーションの需要が増えると考えられる。

(e) **行政サービス提供者** 行政サービスについての情報発信が有効的に行われ、利用者への周知が徹底し、行政サービスの質、量ともに向上されることができる。それにともない、利用者の満足度も向上し、市民活動を活発にすることができると考えられる。

(f) **企業** 組織が保有する知識、ノウハウを体系化し、共有可能にする。関連する情報リソースへのアクセスも容易にする。それにより、組織活動を活発にし、競争力を高めるとともに、次世代への継承を容易にする。