

3.やってみましたIMI対応・素朴な疑問の 共有と対策のアイデア共有

2018年12月21日
「IMI事例共有・勉強会」



3.やってみましたIMI対応・素朴な疑問の共有と対策 のアイデア共有



- ▶ 1.コア語彙の検索、選択はどうしたらよいのかわからない
- ▶ 2.ドメイン語彙、応用語彙の作成方法がわからない
- ▶ 3.クラス内の構造化されたプロパティの選び方がわからない
- ▶ 4.コア語彙のプロパティだけ使うといけないのか
- ▶ 5.データモデル設計時の項目名の付け方がわからない
- ▶ 6.データの範囲(指標的)は、語彙としてどう表現すればよいのかわからない
- ▶ 7.コード、コードリストの使い方がわからない
- ▶ 8.構造化されたXMLデータセットがXSD Sequence Errorとなる
- ▶ 9.固定値制約、グループ指定のデータモデルの意味がわからない
- ▶ 10.現実の日付けデータに合わない場合の語彙表現がわからない
- ▶ 11.既に構造化されたデータセットをどのようにマッピングしたら良いのかわからない
- ▶ 12.座標系のデータセットはどのように語彙設計したらよいのかわからない



- ▶ 1.コア語彙の検索、選択はどうしたらよいのかわからぬ

コア語彙の検索、選択はどうしたら良いのか分からぬ

クラス用語

事柄(事物や出来事)に関する概念です。
性質や属性を表すプロパティ用語と結びついて
使っています。

プロパティ用語

事柄の性質や事柄と事柄の関係を指示する概念
を表すための用語です。

※コア語彙は、この関連で定義し提供しています。

人型
...ic:氏名
...ic:生年月日

日付型
...ic:年
...ic:月
...ic:日

生年月日

型: ic:日付型

説明: 生年月日を記述するためのプロパティ用語
定義域: 人型

日

型: xsd:integer

説明: 日付や日時などの日の値を記述するためのプロパティ用語
定義域: 日時型 日付型

探し方その1

基本は、コア語彙のHTML一覧から探す・・

クラス用語: 62
プロパティ用語: 121



情報連携に不可欠な基本情報やツールを提供するサイトです

English

共通語彙基盤 コア語彙2.4.1

To the extent possible under law, Ministry of Economy, Trade and Industry and Information-technology Promotion Agency, Japan has waived all copyright and related or neighboring rights to Infrastructure for Multilayer Interoperability (IMI) Core Vocabulary 2.4.1. This work is published from: Japan .

赤い取り消し線が引かれた用語およびプロパティは将来の版では使用できなくなるため、使用を推奨しません。

=>プロパティ用語一覧

クラス用語一覧

人型	名称型	氏名型	住所型
連絡先型	電話番号型	組織型	業務組織型
法人型	人命数型	構成員型	組織関連型
場所型	座標型	ID型	ID体系型
製品型	製品個品型	地物型	土地型
施設型	施設間連型	建物型	駐車場型
設備型	イベント型	活動型	閑与型
測定値型	数量型	容量型	面積型
重量型	長さ型	時間型	金額型
価格型	実体型	状況型	日時型
日付型	期間型	期間スケジュール型	イベントスケジュール型
定期スケジュール型	詳細スケジュール型	詳細スケジュール規則型	コード型
単位コード型	コードリスト型	アクセス型	アクセス区間型
概念型	事物型	参照型	記述型
サービス型	文書型	対象型	制約型
コード制約型	範囲制約型	期間制約型	

プロパティ用語一覧

体系	ID	応援參照系	シリアル番号
製造者ID	地理識別子	版元著者ID	アクセス区間
アクセス	関連イベント	料金	価格
金額	資本金	購入価格	代表者
座標データ書式	数量単位	カレンダー	国コード
市町村コード	住所コード	都道府県コード	コード種別
量種別	果数	府県国コード	国種コード
種別コード	性別コード	出生国コード	組織種別
施設	関連施設	管理者	入数
収容人数	期間	利用可能時間	設備
時間	設置者	組織	関連組織
代理人	建物	駐車場	FAX番号
携帯電話番号	電話番号	有効開始日時	有効終了日時
開始日時	終了日時	從業員数	消費期限
製造日	死亡年月日	端末ID	生年月日
構成員	名称	延べ面積	建築面積
敷地面積	連絡先	地理座標	氏名
担当者名	開港港	開港	所有者
住所	本籍	重量	場所
出生地	活動状況	開港場所	地物
始点	終点	設置地点	測地高度
高さ	長さ	幅	最高の高さ
第1生日	開催日	開始日	事業年度開始日
終了日	使用期限	眞味期限	現置日
長さ単位	面積単位	時間単位	通貨コード
単位表記	容量単位	重量単位	単位
参照	URI	Webサイト	画像
標準型日時	自動認識ID	地図	標準型日付
秒	数値	地下階数	地上階数
年	月	時	収容台数
人數	日	分	標準表示式
色	Eメールアドレス	緯度	概要
力ナ表記	開催期日	外部利用可否	区
国	キーワード	規制情報	原産国
号	形態	経度	座標データ
参加方法	国籍	属性	事業種目
市町村	時間帯	種別	種別
詳細	始点備考	統合備考	姓
姓力ナ表記	状態	使用方法	性別
姓名力ナ表記	製造者	姓名	設置位置
説明	姓名ローマ字表記	姓名ローマ字表記	測定名稱
組織種別位置	送付先	測定単位	開作者
担当者役職	対応言語	対象者	通称
都道府県	駐車場種別	丁目	名力ナ表記
名ローマ字表記	名	内線番号	バリアフリー
番地	年号	バージョン	備考
表記	番地補足	版元番号	ミドルネーム
ミドルネームカナ表記	ビル番号	ビル名	役割コード
郵便番号	ミドルネームローマ字表記	役割	利用者
ローマ字表記	要約	略称アルファベット	構造
主要用途	路線	ロット番号	通貨
発行者	出生国	町名	開始時間
終了時間	部屋番号	方書	関連コード
日時	上位コード	下位コード	活動
提供者	関連人員	面積	作成者
寄与者	参照先	表題	記形式
記録形式コード	日付	資源タイプ	権利管理
利用対象	出処	言語	制約コード
制約対象	利用方法	制約	下限値
境界を含まない下限値	上限値	境界を含まない上限値	体重
メタデータ	年齢	身長	
	記述		

探し方その2

コア語彙のスキーマ(RDF/TTL)をRDFデータベースへgraphへ投入して人型を検索してみる

PREFIX ic: <http://imi.go.jp/ns/core/rdf#>

SELECT * WHERE {

?s ?p ?o

FILTER (REGEX(?s, "http://imi.go.jp/ns/core/rdf#") && REGEX(?o, "人型"))

}

The screenshot shows the Virtuoso SPARQL Execution interface. The query window contains the following SPARQL code:

```
PREFIX ic: <http://imi.go.jp/ns/core/rdf#>
SELECT * WHERE [
?s ?p ?o
FILTER (REGEX(?s, "http://imi.go.jp/ns/core/rdf#") && REGEX(?o, "人型"))
]
```

The results table displays the following data:

s	p	o
http://imi.go.jp/ns/core/rdf#人型	http://www.w3.org/2000/01/rdf-schema#label	"人型"@ja
http://imi.go.jp/ns/core/rdf#法人型	http://www.w3.org/2000/01/rdf-schema#label	"法人型"@ja
http://imi.go.jp/ns/core/rdf#構成員	http://www.w3.org/2000/01/rdf-schema#range	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#住所	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#体重	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#出生国	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#出生国コード	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#出生地	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#国籍	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#国籍コード	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#年齢	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#性別	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#生別コード	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#本籍	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#死亡年月日	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#氏名	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#生年月日	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#身長	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#連絡先	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#人型
http://imi.go.jp/ns/core/rdf#組織種別	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#法人型
http://imi.go.jp/ns/core/rdf#組織種別位置	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#法人型
http://imi.go.jp/ns/core/rdf#資本金	http://schema.org/domainIncludes	http://imi.go.jp/ns/core/rdf#法人型

探し方その3

PostgreSQLデータベースで、RDF検索して見る(テーブル、ファンクション)

-- テーブル定義

```
DROP TABLE IF EXISTS "コア語彙" CASCADE;
CREATE TABLE "コア語彙" ("主語" TEXT, "述語" TEXT, "目的語" TEXT);
```

-- ファンクション定義

```
DROP FUNCTION IF EXISTS "コア語彙検索"(text, integer);
CREATE FUNCTION "コア語彙検索"(text, integer) RETURNS
TABLE("主語" TEXT, "述語" TEXT, "目的語" TEXT, "パス" TEXT[], "階層" TEXT ) AS $$%
WITH RECURSIVE "一次結果"("主語", "述語", "目的語", "パス") AS
(SELECT "主語", "述語", "目的語", ARRAY["主語"] FROM "コア語彙" WHERE "主語" = $1
UNION ALL
SELECT コア語彙."主語", コア語彙."述語", コア語彙."目的語", result."パス" || コア語彙."主語"
FROM "一次結果" AS result,"コア語彙" AS コア語彙 WHERE result."目的語" = コア語彙."主語"
AND コア語彙."目的語" != ALL(result."パス"))
SELECT "主語", "述語", "目的語", "パス",ARRAY_UPPER("パス", 1)::TEXT AS "階層" FROM "一次結果"
WHERE ARRAY_UPPER("パス", 1) <= $2;
$$ LANGUAGE SQL;
```

PostgreSQLデータベースで、RDFデータ(class,property,set)

```
INSERT INTO コア語彙 values('ic:人型','class','ic:実体型');  
INSERT INTO コア語彙 values('ic:名称型','class','ic:事物型');  
INSERT INTO コア語彙 values('ic:氏名型','class','ic:事物型');  
INSERT INTO コア語彙 values('ic:住所型','class','ic:事物型');  
INSERT INTO コア語彙 values('ic:連絡先型','class','ic:事物型');  
....
```

```
INSERT INTO コア語彙 values('ic:始点','property','ic:地物型');  
INSERT INTO コア語彙 values('ic:終点','property','ic:地物型');  
INSERT INTO コア語彙 values('ic:設置地点','property','ic:地物型');  
INSERT INTO コア語彙 values('ic:測地高度','property','ic:長さ型');  
INSERT INTO コア語彙 values('ic:高さ','property','ic:長さ型');  
....
```

```
INSERT INTO コア語彙 VALUES('ic:人型','set','ic:氏名');  
INSERT INTO コア語彙 VALUES('ic:人型','set','ic:性別');  
INSERT INTO コア語彙 VALUES('ic:人型','set','ic:性別コード');  
INSERT INTO コア語彙 VALUES('ic:人型','set','ic:生年月日');  
....
```

SELECT "コア語彙検索"("主語",4) FROM "コア語彙" WHERE "主語" = 'ic:氏名型' GROUP BY "コア語彙検索"("主語",4);

コア語彙検索

```
(ic:ID,property,ic:ID型,"{ic:氏名型,ic:事物型,ic:ID}",3)
(ic:ID型,class,ic:概念型,"{ic:氏名型,ic:事物型,ic:ID,ic:ID型}",4)
(ic:ID型,set,ic:参照,"{ic:氏名型,ic:事物型,ic:ID,ic:ID型}",4)
(ic:ID型,set,ic:識別値,"{ic:氏名型,ic:事物型,ic:ID,ic:ID型}",4)
(ic:ID型,set,ic:体系,"{ic:氏名型,ic:事物型,ic:ID,ic:ID型}",4)
(ic:ミドルネーム,property,xsd:string,"{ic:氏名型,ic:ミドルネーム}",2)
(ic:ミドルネームカナ表記,property,xsd:string,"{ic:氏名型,ic:ミドルネームカナ表記}",2)
(ic:ミドルネームローマ字表記,property,xsd:string,"{ic:氏名型,ic:ミドルネームローマ字表記}",2)
(ic:メタデータ,property,ic:文書型,"{ic:氏名型,ic:事物型,ic:概念型,ic:メタデータ}",4)
(ic:画像,property,xsd:anyURI,"{ic:氏名型,ic:事物型,ic:画像}",3)
(ic:概念型,set,ic:メタデータ,"{ic:氏名型,ic:事物型,ic:概念型}",3)
(ic:概念型,set,ic:種別,"{ic:氏名型,ic:事物型,ic:概念型}",3)
(ic:概念型,set,ic:種別コード,"{ic:氏名型,ic:事物型,ic:概念型}",3)
(ic:記述,property,ic:記述型,"{ic:氏名型,ic:事物型,ic:記述}",3)
(ic:記述型,class,ic:概念型,"{ic:氏名型,ic:事物型,ic:記述型}",4)
(ic:記述型,set,ic:説明,"{ic:氏名型,ic:事物型,ic:記述,ic:記述型}",4)
(ic:参照,property,ic:参照型,"{ic:氏名型,ic:事物型,ic:参照}",3)
(ic:参照型,class,ic:概念型,"{ic:氏名型,ic:事物型,ic:参照,ic:参照型}",4)
(ic:参照型,set,ic:参照先,"{ic:氏名型,ic:事物型,ic:参照,ic:参照型}",4)
(ic:氏名型,class,ic:事物型,ic:氏名型),1
(ic:氏名型,set,ic:ミドルネーム,{ic:氏名型},1)
(ic:氏名型,set,ic:ミドルネームカナ表記,{ic:氏名型},1)
(ic:氏名型,set,ic:ミドルネームローマ字表記,{ic:氏名型},1)
(ic:氏名型,set,ic:姓,{ic:氏名型},1)
(ic:氏名型,set,ic:姓カナ表記,{ic:氏名型},1)
(ic:氏名型,set,ic:姓名,{ic:氏名型},1)
(ic:氏名型,set,ic:姓名カナ表記,{ic:氏名型},1)
(ic:氏名型,set,ic:姓名ローマ字表記,{ic:氏名型},1)
(ic:氏名型,set,ic:名,{ic:氏名型},1)
(ic:氏名型,set,ic:名カナ表記,{ic:氏名型},1)
(ic:氏名型,set,ic:名ローマ字表記,{ic:氏名型},1)
(ic:事物型,class,ic:概念型,"{ic:氏名型,ic:事物型}",2)
(ic:事物型,set,ic:ID,"{ic:氏名型,ic:事物型}",2)
(ic:事物型,set,ic:画像,"{ic:氏名型,ic:事物型}",2)
(ic:事物型,set,ic:記述,"{ic:氏名型,ic:事物型}",2)
(ic:事物型,set,ic:参照,"{ic:氏名型,ic:事物型}",2)
(ic:事物型,set,ic:説明,"{ic:氏名型,ic:事物型}",2)
(ic:事物型,set,ic:表記,"{ic:氏名型,ic:事物型}",2)
(ic:種別,property,xsd:string,"{ic:氏名型,ic:事物型,ic:概念型,ic:種別}",4)
(ic:種別コード,property,ic:コード型,"{ic:氏名型,ic:事物型,ic:概念型,ic:種別コード}",4)
(ic:性,property,xsd:string,"{ic:氏名型,ic:性}",2)
(ic:姓カナ表記,property,xsd:string,"{ic:氏名型,ic:姓カナ表記}",2)
(ic:姓ローマ字表記,property,xsd:string,"{ic:氏名型,ic:姓ローマ字表記}",2)
(ic:姓名,property,xsd:string,"{ic:氏名型,ic:姓名}",2)
(ic:姓名カナ表記,property,xsd:string,"{ic:氏名型,ic:姓名カナ表記}",2)
(ic:姓名ローマ字表記,property,xsd:string,"{ic:氏名型,ic:姓名ローマ字表記}",2)
(ic:説明,property,xsd:string,"{ic:氏名型,ic:事物型,ic:説明}",3)
(ic:表記,property,xsd:string,"{ic:氏名型,ic:事物型,ic:表記}",3)
(ic:名,property,xsd:string,"{ic:氏名型,ic:名}",2)
(ic:名カナ表記,property,xsd:string,"{ic:氏名型,ic:名カナ表記}",2)
(ic:名ローマ字表記,property,xsd:string,"{ic:氏名型,ic:名ローマ字表記}",2)
(52 行)
```

(ic:種別コード,property,ic:コード型,"{ic:氏名型,ic:事物型,ic:概念型,ic:種別コード}",4)

氏名型

概念型 ☈ 事物型 ☈ 氏名型

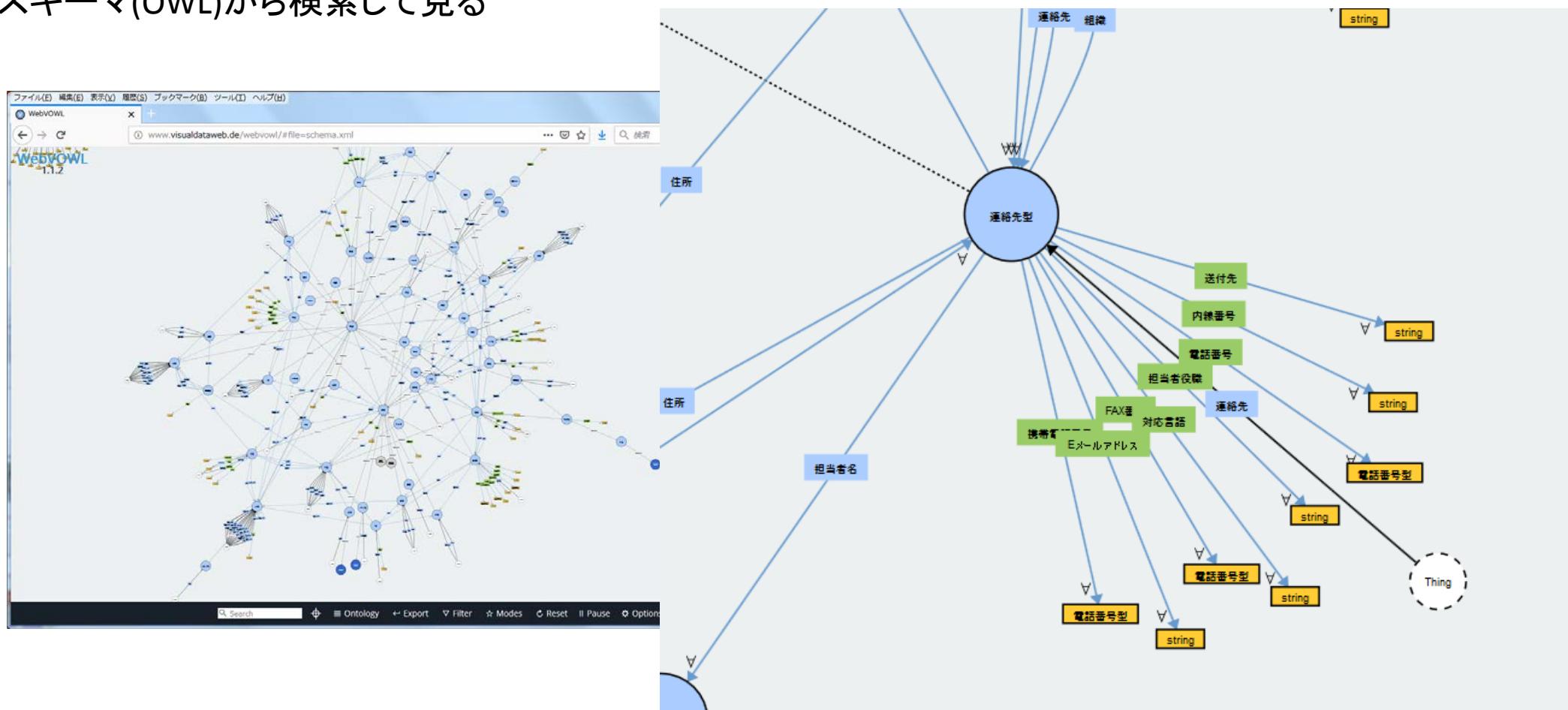
人の氏名を表現するためのクラス用語

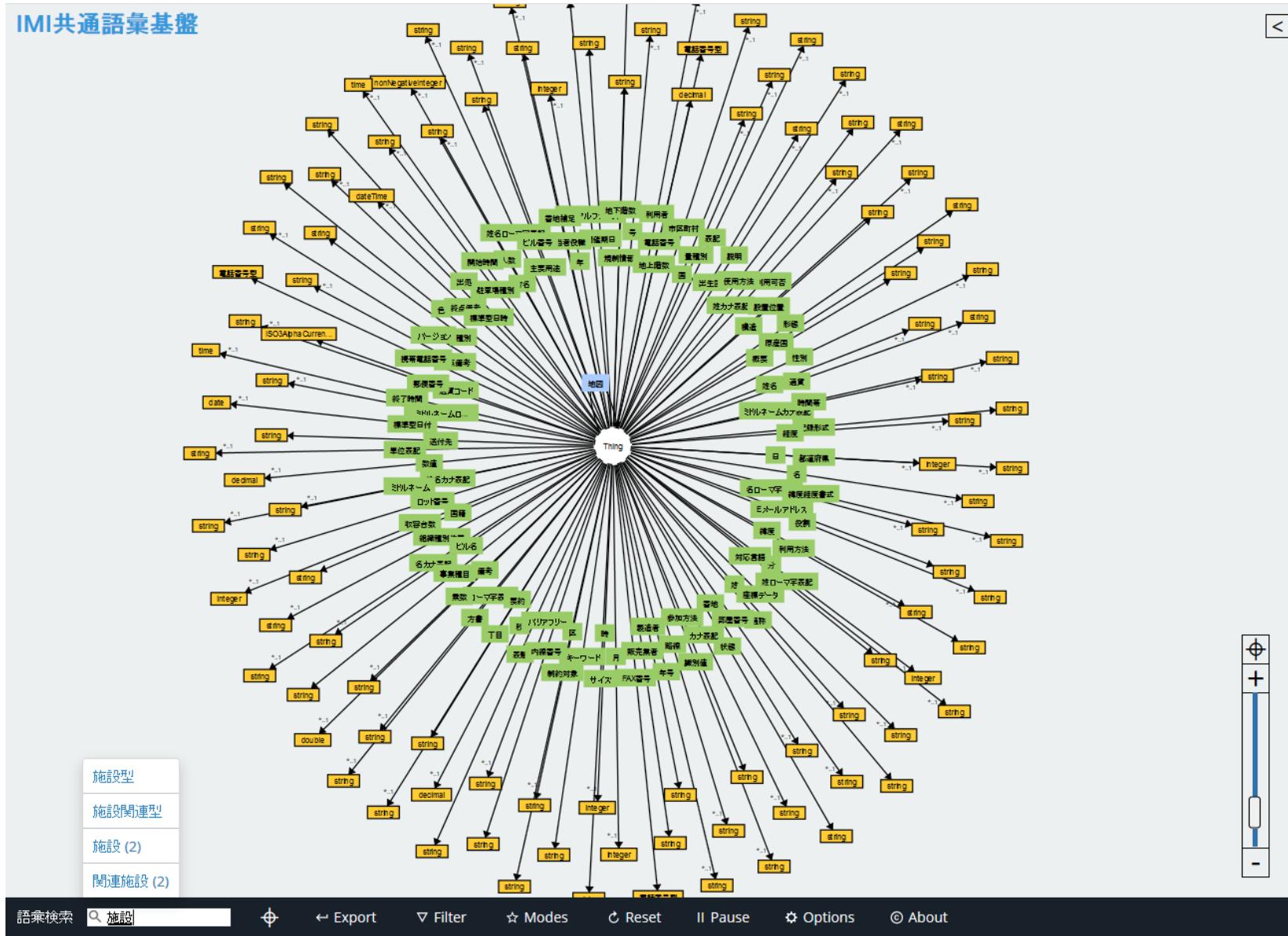
氏名型 のプロパティ			
識別子	値型	回数	説明
ic:種別	xsd:string	0..n	種別の表記を記述するためのプロパティ用語
ic:種別コード	ic:コード型	0..n	種別コードを記述するためのプロパティ用語
ic:メタデータ	ic:文書型	0..n	データのメタデータを記述するためのプロパティ用語
ic:ID	ic:ID型	0..n	事物に割り振られたID (識別子)を記述するためのプロパティ用語

探し方その4

<http://www.visualdataweb.de/webowl/#file=schema.xml>

スキーマ(OWL)から検索して見る



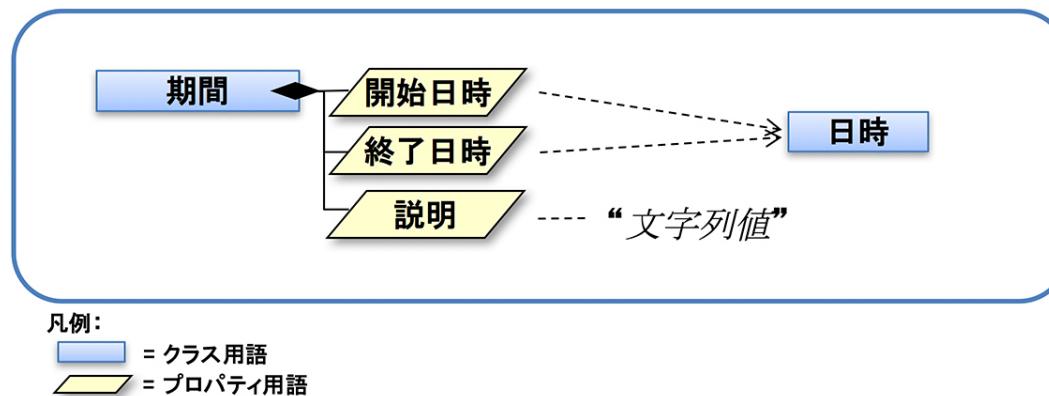


- ▶ 2.ドメイン語彙、応用語彙の作成方法がわからぬ

ドメイン語彙、応用語彙の作成方法

実際に語彙を作り上げるためには、クラス用語とプロパティ用語およびそれらの関連性を定義する必要があります。

プロパティ用語は、クラス用語の性質・属性を表すものとして、クラス用語を何らかの値や、他のクラス用語と結び付けるために用いられます。たとえば、【期間】というクラス用語は、【開始日時】、【終了日時】、【説明】というプロパティ用語を組み合わせることができます。このような用語の集合が「語彙」です。



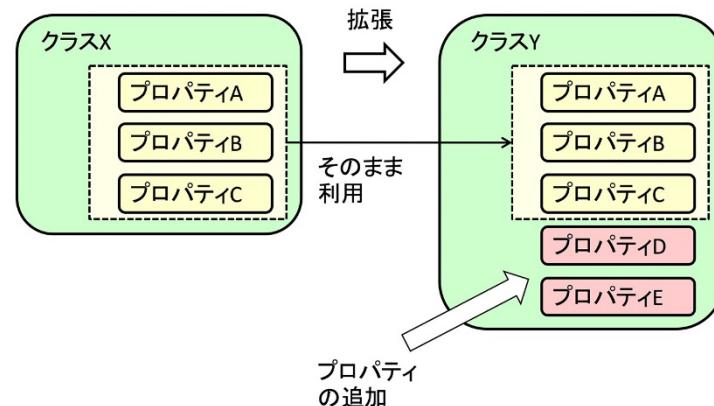
用語には、それ自体の意味と共に、クラス用語とプロパティ用語の関係のような構造があります。また、テキストデータの値の書式制限(文字数の上限など)もあります。従って、用語の集合である「語彙」も、用語に関わるそれら様々な情報を伴ったデータの集合として捉えられます。

継承による新規クラス用語の作成

語彙のクラス用語をベースに、プロパティ用語の追加、回数の変更などを行って新たなクラス用語を定義することをクラス用語の「継承」と呼びます。

継承には「拡張」と「制限」の2種類がありますが、共通語彙基盤の核となる「コア語彙」では、プロパティ用語を追加してクラス用語を派生させていく「拡張」の手法を使っています。

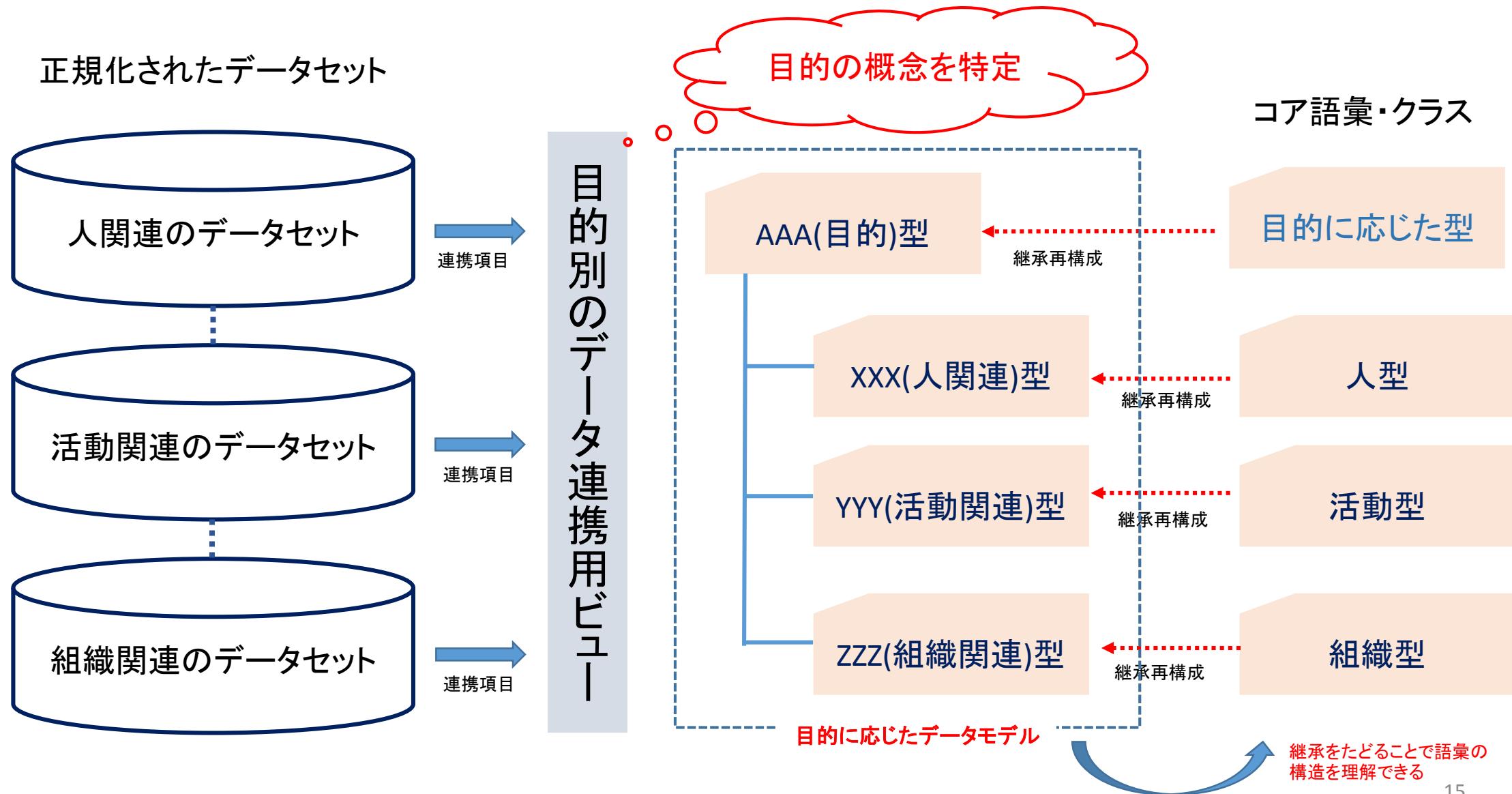
このような継承の考え方を用い、「コア語彙」を継承して「ドメイン語彙」や個別の応用で用いるデータモデルに必要なクラス用語とプロパティ用語を新規に作成することができます。



制約事項：

新規のクラス用語は、コア語彙をベースとした継承によって定義すること（少なくともコア語彙の全ての型のベースとなる基本型としての「ic:事物型」は継承する。なお、直接の継承元がコア語彙のクラス用語ではなくても、継承関係を辿った中にコア語彙のクラス用語（例えばic:事物型）が現れればよい）

複数のルートがあるデータセットはどのようにクラスを考えれば良いのか



- ▶ 3. クラス内の構造化されたプロパティの選び方がわからない

クラス内の構造化されたプロパティの選び方が分からぬ

選択したクラスのプロパティは、すべて利用するのではなく目的にあったプロパティを選択して使用します。

日付型

概念型  日付型

日付を表現するためのクラス用語

日付型 のプロパティ		
識別子	値型	回数
ic:種別	xsd:string	0..n
ic:種別コード	ic:コード型	0..n
ic:メタデータ	ic:文書型	0..n
ic:標準型日付	xsd:date	0..1
ic:年号	xsd:string	0..1
ic:年	xsd:integer	0..1
ic:月	xsd:integer	0..1
ic:日	xsd:integer	0..1
ic:時間帯	xsd:string	0..1
ic:カレンダー	ic:コード型	0..1
ic:表記	xsd:string	0..1

A	B	C	D	E	F	G
No	施設名	臨時休業日(外部)	補足日	臨時休業日(内部)	営業日	
1	文京グリーンコートセンター	2018-12-21	2018-12-21	2018/12/21	休祭日以外	
2						
3						
4						
5						

プロパティの値型は意識して使う必要があります。
※xsd:dateは YYYY-MM-DD で規定されています

クラス内の構造化されたプロパティの選び方が分からぬ

選択したクラスのプロパティの回数指定のあるものを設定する場合には、グループ指定でユニークにして使います。

日付型		
概念型  日付型		
日付を表現するためのクラス用語		
日付型 のプロパティ		
識別子	値型	回数
ic:種別	xsd:string	0..n
ic:種別コード	ic:コード型	0..n
ic:メタデータ	ic:文書型	0..n
ic:標準型日付	xsd:date	0..1
ic:年号	xsd:string	0..1
ic:年	xsd:integer	0..1
ic:月	xsd:integer	0..1
ic:日	xsd:integer	0..1
ic:時間帯	xsd:string	0..1
ic:カレンダー	ic:コード型	0..1
ic:表記	xsd:string	0..1



Excel screenshot showing the '日付型' data in a spreadsheet:

No	施設名	臨時休業日(外部)	臨時休業日(内部)	営業日	補足日
1	文京グリーンコートセンター	2018-12-21		2018/12/21 休祭日以外	2018-12-21

use ic:日付型>ic:標準型日付[臨時休業日(外部)];
use ic:日付型>ic:標準型日付[補足日];

- ▶ 4.コア語彙のプロパティだけ使うといけないのか
- ▶ 5.データモデル設計時の項目名の付け方がわからない

コア語彙のプロパティだけ使うといけないのか

コア語彙のクラスが用途に合わないし…
項目名の標準化にプロパティ用語だけ使用したい…



情報連携に不可欠な基本情報やツールを提供するサイトです

English

共通語彙基盤 コア語彙2.4.1

To the extent possible under law, Ministry of Economy, Trade and Industry and Information-technology Promotion Agency, Japan has waived all copyright and related or neighboring rights to Infrastructure for Multilayer Interoperability (IMI) Core Vocabulary 2.4.1. This work is published from: Japan .

赤い取り消し線が引かれた用語およびプロパティは将来の版では使用できなくなるため、使用を推奨しません。

=>プロパティ用語一覧

クラス用語一覧

人型	名称型	氏名型	住所型
連絡先型	電話番号型	組織型	業務組織型
法人型	人數型	構成員型	組織連型
場所型	座標型	ID型	ID体系型
製品型	製品個品型	地物型	土地型
施設型	施設関連型	建物型	駐車場型
設備型	イベント型	活動型	閥与型
測定値型	数量型	容量型	面積型
重量型	長さ型	時間型	金額型
価格型	実体型	状況型	日時型
日付型	期間型	期間スケジュール型	イベントスケジュール型
定期スケジュール型	詳細スケジュール型	詳細スケジュール規則型	コード型
単位コード型	コードリスト型	アクセス型	アクセス区間型
概念型	事物型	参照型	記述型
サービス型	文書型	対象型	制約型
コード制約型	範囲制約型	期間制約型	

プロパティ用語一覧

体系	ID	応接券照系	シリアル番号
製造者ID	地理識別子	版元番号ID	アクセス区間
アクセス	関連イベント	料金	価格
金額	資本金	購入価格	代表者
座標データ書式	数量単位	カレンダー	国コード
市町村コード	住所コード	都道府県コード	コード種別
量種別	果数	原産国コード	国種コード
種別コード	性別コード	出生国コード	組織種別
施設	関連施設	管理者	入数
収容人数	期間	利用可能時間	設備
時間	設置者	組織	関連組織
代理人	建物	駐車場	FAX番号
携帯電話番号	電話番号	有効開始日時	有効終了日時
開始日時	終了日時	往來券	消費期限
製造日	死亡年月日	端末日	生年月日
構成員	名称	延べ面積	建築面積
敷地面積	連絡先	地理座標	氏名
担当者名	開港場	開港	所有者
住所	本籍	重量	場所
出生地	活動状況	開港場所	地物
始点	終点	設備地点	測地高さ
高さ	長さ	幅	最高の高さ
第1生日	開催日	開始日	事業年度開始日
終了日	使用期限	意味期限	現置日
設立日	面積単位	時間単位	通貨コード
長さ単位	容量単位	重量単位	単位
単位表記	URI	Webサイト	画像
参照	自動認識ID	地図	標準型日付
標準型日時	数値	地下階数	地上階数
秒	月	時	収容台数
年	日	分	標準表示
人數	Eメールアドレス	緯度	緯度経度表示
色	開催期日	外部利用可否	概要
力ナ表記	キーワード	規制情報	区
国	形態	経度	原産国
号	国籍	属性	座標データ
参加方法	時間帯	識別値	事例項目
市町村	始点備考	統合備考	種別
詳細	状態	使用方法	姓
姓力ナ表記	製造者	姓名	性別
姓名ローマ字表記	姓名ローマ字表記	姓ローマ字表記	設置位置
説明	送付先	測定単位	測定名称
組織種別位置	対応言語	対象券	開港者
担当者役職	駐車場種別	丁目	通称
都道府県	名	内線番号	名力ナ表記
名ローマ字表記	年号	バージョン	バリアフリー
番地	面積補足	版元番号	備考
表記	ビル番号	ミドルネーム	ミドルネーム
ミドルネームカナ表記	ビル名	役割	役割コード
郵便番号	郵便番号	略称アルファベット	利用者
ローマ字表記	郵便番号	ロット番号	構造
主要用途	出生国	町名	通貨
発行者	部屋番号	方書	開始時間
終了時間	上位コード	下位コード	関連コード
日時	関連人員	面積	活動
提供者	参考先	表題	作成者
寄与者	日付	資源タイプ	記述形式
記録形式コード	出処	言語	権利管理
利用対象	利用方法	制約	制約コード
制約対象	上限値	境界を含まない上限値	下限値
境界を含まない下限値	年齢	身長	体重
メタデータ	記述		

項目名の標準化のためプロパティ用語だけ使用したい ← しかし

原始データ設計

保守担当者名
管理者名
連絡先担当者名

コア語彙のプロパティ

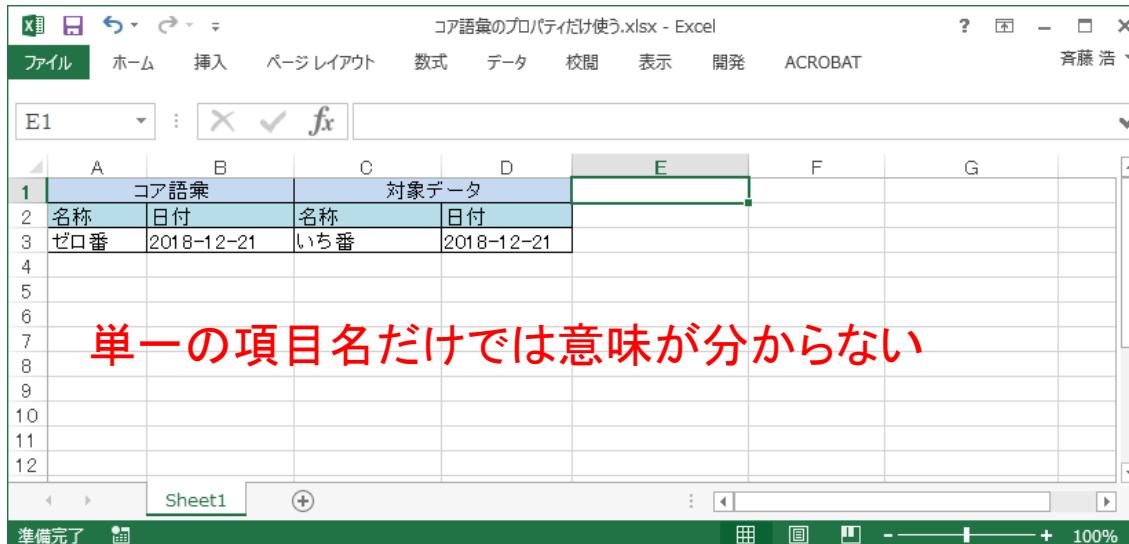
氏名型 のプロパティ
識別了
ic:種別
ic:種別コード
ic:メタデータ
ic:ID
ic:表記
ic:参照
ic:画像
ic:説明
ic:記述
ic:姓名
ic:姓名カナ表記
ic:姓名ローマ字表記
ic:姓
ic:姓カナ表記
ic:姓ローマ字表記
ic:名
ic:名カナ表記
ic:名ローマ字表記
ic:ミドルネーム
ic:ミドルネームカナ表記
ic:ミドルネームローマ字表記

プロパティ用語だけを適用した場合

ic:姓名
ic:姓名
ic:姓名

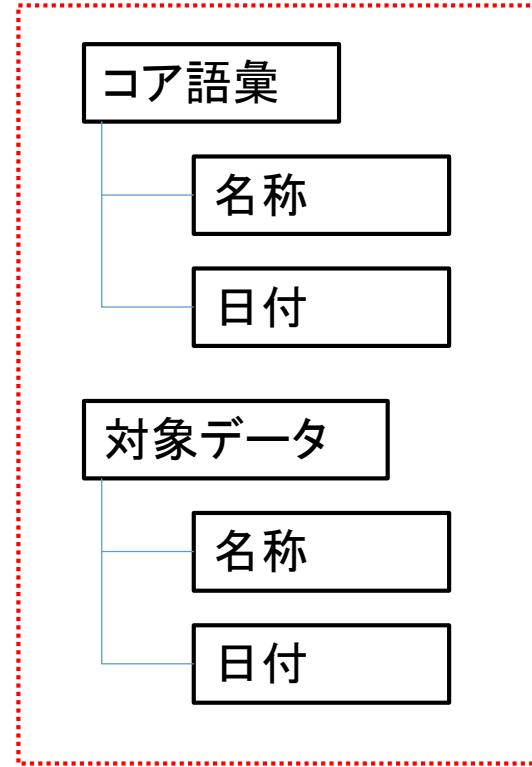
- ・項目の意味が分からない
- ・データ項目が重複してどれか分からぬ

モデル設計時の項目名の付け方

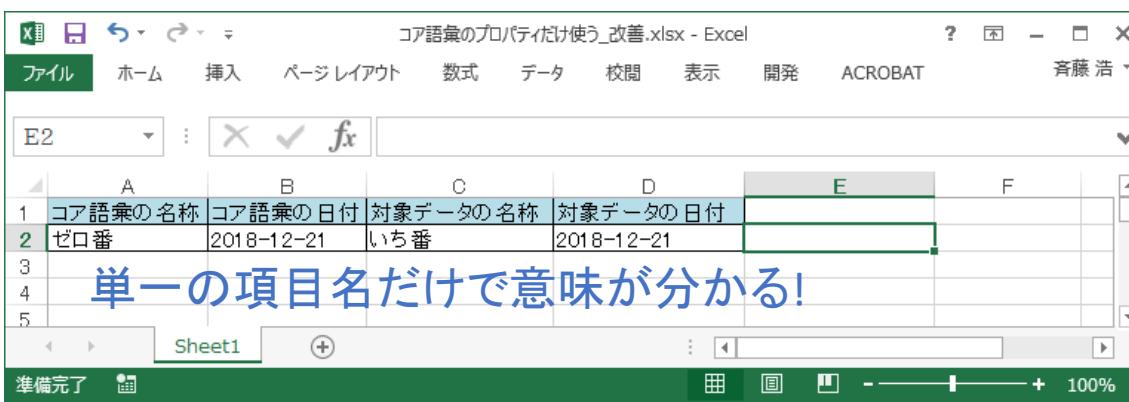


コア語彙		対象データ	
名称	日付	名称	日付
ゼロ番	2018-12-21	いち番	2018-12-21

单一の項目名だけでは意味が分からぬ



構造化されていると意味が分かるが..
表形式データでは?_?



コア語彙の名称	コア語彙の日付	対象データの名称	対象データの日付
ゼロ番	2018-12-21	いち番	2018-12-21

单一の項目名だけで意味が分かる!

- ▶ 6.データの範囲(指標的)は、語彙としてどう表現すればよいのかわからない
- ▶ 7.コード、コードリストの使い方がわからない

データの範囲(指標的)は語彙としてどう表現すれば良いのか分からぬ?

The screenshot shows a Microsoft Excel spreadsheet titled "20181219_岩田さんの課題.xlsx - Excel". The data is organized into columns A through G:

	A	B	C	D	E	F	G
1	被保険者番号	氏名	生年月日	住所	身長	体重	
2	(削除)	(削除)	45歳-47歳	文京区駒込	166	61	
3	(削除)	(削除)	45歳-47歳	文京区駒込	153	52	
4	(削除)	(削除)	42歳-44歳	文京区駒込	180以上	61	
5	(削除)	(削除)	42歳-44歳	文京区千石	166	61	
6	(削除)	(削除)	45歳-47歳	文京区駒込	166	61	
7	(削除)	(削除)	45歳-47歳	文京区池袋	153	50以下	
8	(削除)	(削除)	42歳-44歳	文京区駒込	180以上	61	
9	(削除)	(削除)	45歳-47歳	文京区駒込	166	61	
10							
11							
12							
13							
14							
15							
16							

Two specific ranges are highlighted with red rounded rectangles:

- A red box encloses the "生年月日" (Birth Date) column (Column C), with the text "K-匿名化" (K-anonymization) centered below it.
- A red box encloses the "身長" (Height) and "体重" (Weight) columns (Columns E and F), with the text "特異値措置" (Outlier handling) centered below it.

コードリスト、コードを利用する方法で考えて見る

識別値が何を表すのかを明示する個別コード定義も必要です。コード定義においてコード種別を示すことで、個々のコードとコードリストを結びつけます。コードリストと個別コードをセットで定義して、コードリスト定義からもリストに含まれるコードの全体が把握できるようにします。

コードリスト型

プロパティ	役割	値の型
ic:名称	コードリストの名称	ic:名称型
ic:発行者	コードリストを作成、発行、維持する主体	ic:実体型
ic:バージョン	定義しているコードリストのバージョン	xsd:string

コード型

プロパティ	役割	値の型
ic:コード種別	そのコードが含まれるコードリスト	ic:コードリスト型
ic:識別値	情報属性を機械処理しやすく表す符号	xsd:string
ic:表記	識別値が表す情報属性を人が理解できるように示すラベル	xsd:string
ic:上位コード	階層型コードリストにおける上位コード	ic:コード型
ic:下位コード	階層型コードリストにおける下位コード	ic:コード型
ic:関連コード	関連のあるコード	ic:コード型
ic:参照	追加情報を参照するための情報を記述	ic:参照型

範囲(指標的)情報の集合を表現するためのコードリストの場合

The diagram illustrates the transformation of data from a Microsoft Word document table to a Tableau Table View.

Left Side (Microsoft Word Document):

- A screenshot of a Microsoft Word document titled "20181219_No6_データの範…".
- The table structure is as follows:

	A	B	C	D
1	識別値	日本語表記	英語表記	
2	42	42歳-44歳		
3	43	42歳-44歳		
4	44	42歳-44歳		
5	45	45歳-47歳		
6	46	45歳-47歳		
7	47	45歳-47歳		
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				

Right Side (Tableau Table View):

- A screenshot of a Tableau Table View titled "テーブルビュー".
- The table structure is as follows:

識別値	表記	英語	下位	関連	操作
42	42歳-44歳				<button>グラフ</button> <button>編集</button> <button>削除</button>
43	45歳-47歳				<button>グラフ</button> <button>編集</button> <button>削除</button>
44	42歳-44歳				<button>グラフ</button> <button>編集</button> <button>削除</button>
45	45歳-47歳				<button>グラフ</button> <button>編集</button> <button>削除</button>
46	42歳-44歳				<button>グラフ</button> <button>編集</button> <button>削除</button>
47	45歳-47歳				<button>グラフ</button> <button>編集</button> <button>削除</button>

Bottom Buttons:

- +コードを追加
- +コード関係の自動接続
- 一つ前に戻る
- 一つ操作を進める
- 保存
- 読み込み

例えば、元データは匿名化する際に、識別と対応する表現のコードリストを持っていると仮定すると…
このようなコードリストで変換しているなどが考えられる。=>今回は、変換後のデータセットを対象とする。

応用語彙でモデルを作つてみる

```
#description "ex:K-匿名化の年齢分類を表現するためのクラス用語"  
class ex:匿名化年齢分類型{@ic:コードリスト型};
```

| コードリスト型

概念型 事物型 コードリスト型

```
#description "年齢分類に関する指標を参照するためのプロパティ用語"  
property ex:匿名化年齢分類{@ic:コード型};
```

| コード型

概念型 コード型

```
set ex:匿名化年齢分類型>ex:匿名化年齢分類;
```

```

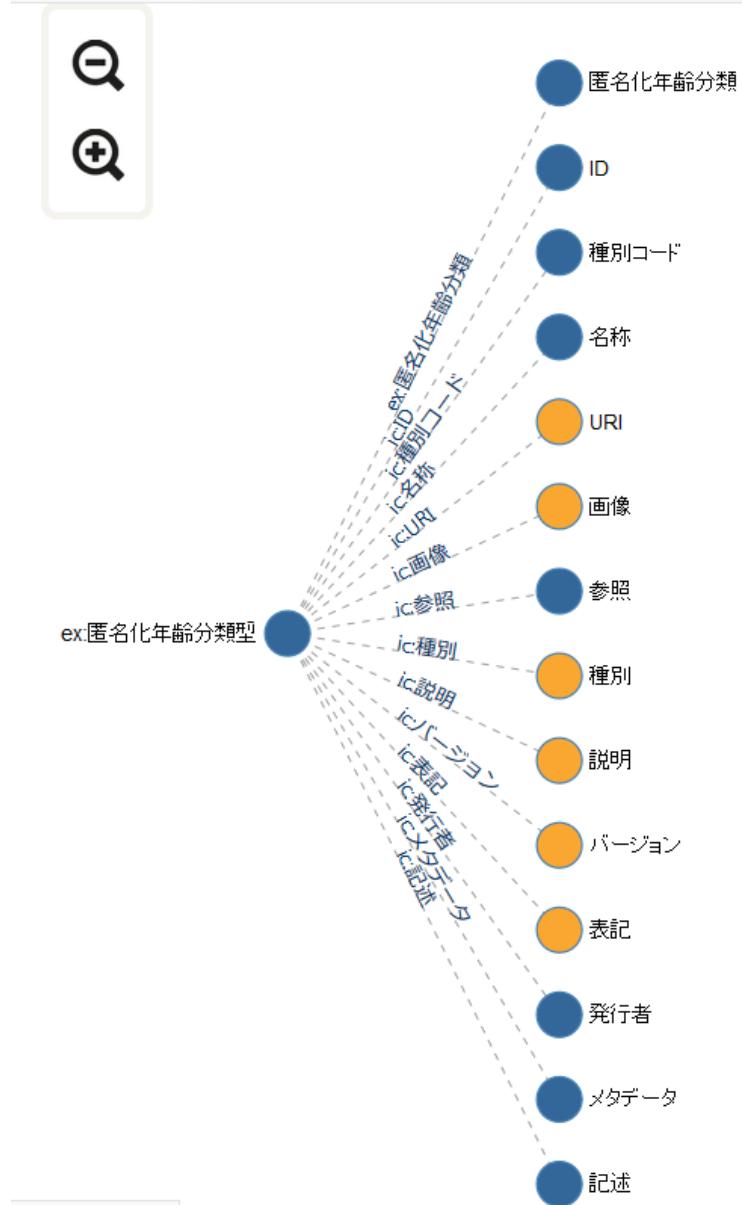
#prefix ic: "http://imi.go.jp/ns/core/2#"
#prefix ex: "http://example.org/"
vocabulary "http://example.org/" ;

#name "匿名化年齢分類型"
class ex:匿名化年齢分類型{@ic:コードリスト型};

#name "匿名化年齢分類"
property ex:匿名化年齢分類{@ic:コード型};

set ex:匿名化年齢分類型>ex:匿名化年齢分類;

```



```
#prefix ic: "http://imi.go.jp/ns/core/2#"
```

```
#prefix ex: "http://example.org/"
```

```
vocabulary "http://example.org/" ;
```

```
#name "匿名化年齢分類型"
```

```
class ex:匿名化年齢分類型{@ic:コードリスト型};
```

```
#name "匿名化年齢分類"
```

```
property ex:匿名化年齢分類{@ic:コード型};
```

```
set ex:匿名化年齢分類型>ex:匿名化年齢分類;
```

```
set ic:数量型 > ex:匿名化年齢分類;
```

年齢

識別子:

ic:年齢

型:

ic:数量型

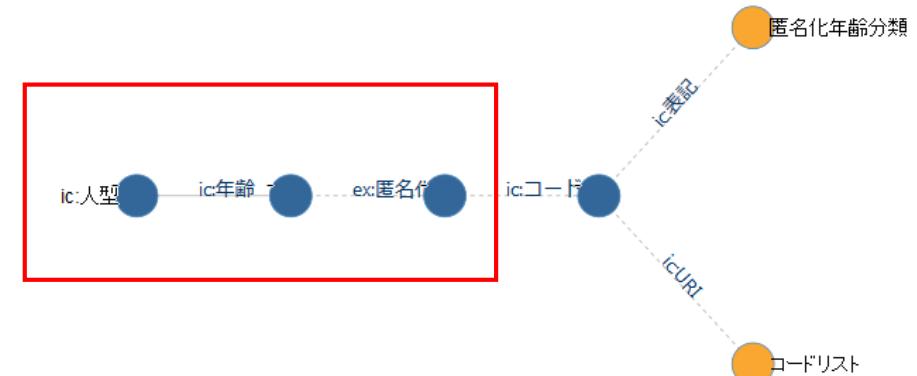
説明:

人の年齢を記述するためのプロパティ用語

定義域:

人型

年齢の匿名分類を表したいので、
数量型に結び付け



データモデル

```
use ic:人型>ic:年齢>ex:匿名化年齢分類>ic:コード種別>ic:URI("http://example/code/匿名化年齢分類");
```

```
use ic:人型>ic:年齢>ex:匿名化年齢分類>ic:コード種別>ic:表記;
```

datamodel.txt

```
#prefix ic: "http://imi.go.jp/ns/core/2#"
#prefix ex: "http://example.org/"
vocabulary "http://example.org/" ;

#name "匿名化年齢分類型"
class ex:匿名化年齢分類型{@ic:コードリスト型};

#name "匿名化年齢分類"
property ex:匿名化年齢分類{@ic:コード型};

set ex:匿名化年齢分類型>ex:匿名化年齢分類;
set ic:数量型 > ex:匿名化年齢分類;

datamodel;
use ic:人型>ic:年齢>ex:匿名化年齢分類>ic:コード種別>ic:URI{="http://example/code/匿名化年齢分類"};
use ic:人型>ic:年齢>ex:匿名化年齢分類>ic:コード種別>ic:表記;
```

mapping.json

```
{
    "mapping": {
        "コードリスト": "ic:人型>ic:年齢>ex:匿名化年齢分類>ic:コード種別>ic:URI",
        "匿名化年齢分類": "ic:人型>ic:年齢>ex:匿名化年齢分類>ic:コード種別>ic:表記"
    }
}
```

変更・追記

種別として参照先を固定値でいれて、
構造化データ上に表現してみる

mapping.json

```
{
    "mapping": {
        "匿名化年齢分類": "ic:人型>ic:年齢>ex:匿名化年齢分類>ic:コード種別 ic:URI='http://example.org/code/匿名化年齢分類'>ic:表記"
    }
}
```

この XML ファイルにはスタイル情報が関連づけられていないようです。以下にドキュメントツリーを表示します。

```
-<root>
  -<ic:人 xsi:type="ic:人型">
    -<ic:年齢>
      -<ex:匿名化年齢分類>
        -<ic:コード種別>
          <ic:URI xsi:type="xsd:anyURI">http://example.org/code/匿名化年齢分類</ic:URI>
          <ic:表記>45歳-47歳</ic:表記>
        </ic:コード種別>
      </ex:匿名化年齢分類>
    </ic:年齢>
  </ic:人>
  -<ic:人 xsi:type="ic:人型">
    -<ic:年齢>
      -<ex:匿名化年齢分類>
        -<ic:コード種別>
          <ic:URI xsi:type="xsd:anyURI">http://example.org/code/匿名化年齢分類</ic:URI>
          <ic:表記>45歳-47歳</ic:表記>
        </ic:コード種別>
      </ex:匿名化年齢分類>
    </ic:年齢>
  </ic:人>
  -<ic:人 xsi:type="ic:人型">
    -<ic:年齢>
      -<ex:匿名化年齢分類>
        -<ic:コード種別>
          <ic:URI xsi:type="xsd:anyURI">http://example.org/code/匿名化年齢分類</ic:URI>
          <ic:表記>42歳-44歳</ic:表記>
        </ic:コード種別>
      </ex:匿名化年齢分類>
    </ic:年齢>
  </ic:人>
  -<ic:人 xsi:type="ic:人型">
    -<ic:年齢>
      -<ex:匿名化年齢分類>
        -<ic:コード種別>
```

.....

- ▶ 8.構造化されたXMLデータセットがXSD Sequence Error となる

データ変換後の構造化において、XSD Sequence Validation error がおきる問題

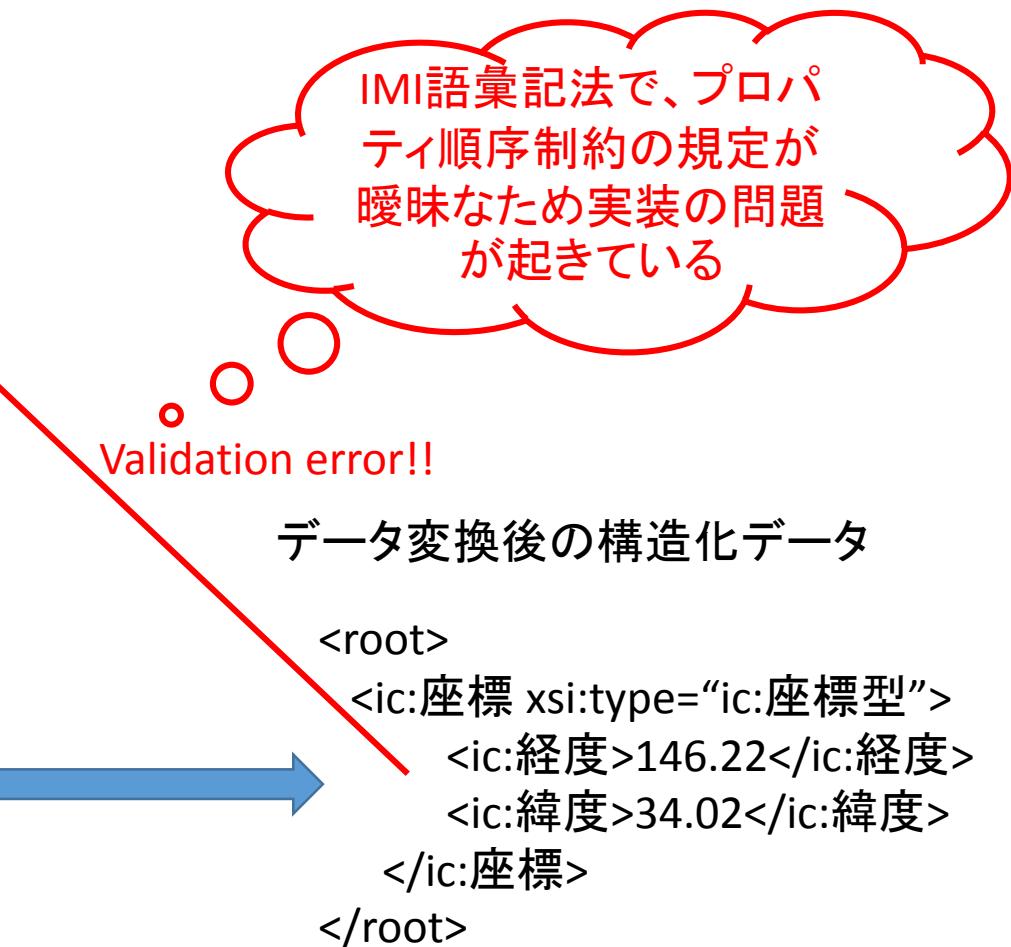
実装におけるデータ変換後の構造化問題のポイント

語彙schema

```
<xsd:group name="座標型-固有要素グループ">
  <xsd:sequence>
    <xsd:element ref="ic:座標参照系" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="ic:経度書式" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="ic:緯度" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="ic:経度" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="ic:座標データ書式" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="ic:座標データ" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="ic:測地高度" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:group>
```

IMI語彙記法で、use設定による構造順指定

```
#prefix ic: "http://imi.go.jp/ns/core/2#"
datamodel;
use ic:座標型>ic:経度;
use ic:座標型>ic:緯度;
```



- ▶ 9. 固定値制約、グループ指定のデータモデルの意味がわからない

固定値制約、グループ指定のデータモデルの意味が分からぬ

A	B	C	D	E	F
1 名称	早朝深夜_電話番号	営業時間_電話番号			
2 いち	01-1111-1111	01-2222-2222			
3 にい	02-1111-1111	02-2222-2222			
4					
5					
6					

意味の違う電話番号が2つあるが、グループ指定と固定値制約のどちらでデータモデルを表現すれば良いのでしょうか？

Modelでは、値等価制約として書きたくなるが、書かない!



項目名とデータモデル ?	
項目名	データモデル
名称	use ic:法人型>ic:名称;
営業時間_電話番号	use ic:法人型>ic:連絡先[営業時間]>ic:電話番号;
説明_営業時間	use ic:法人型>ic:連絡先[営業時間]>ic:説明;
早朝深夜_電話番号	use ic:法人型>ic:連絡先[早朝深夜]>ic:電話番号;
説明_早朝深夜	use ic:法人型>ic:連絡先[早朝深夜]>ic:説明;

項目名とデータモデル ?	
項目名	データモデル
名称	use ic:法人型>ic:名称;
営業時間_電話番号	use ic:法人型>ic:連絡先[営業時間]>ic:電話番号;
早朝深夜_電話番号	use ic:法人型>ic:連絡先[早朝深夜]>ic:電話番号;

生成されるModelとMapping

datamodel.txt

```
#prefix ic: "http://imi.go.jp/ns/core/2#"
datamodel;
use ic:法人型>ic:名称;
use ic:法人型>ic:連絡先 [営業時間]>ic:電話番号;
use ic:法人型>ic:連絡先 [早朝深夜]>ic:電話番号;
```

mapping.json

```
{
    "mapping": {
        "名称": "ic:法人型>ic:名称",
        "営業時間_電話番号": "ic:法人型>ic:連絡先 [営業時間]>ic:電話番号",
        "早朝深夜_電話番号": "ic:法人型>ic:連絡先 [早朝深夜]>ic:電話番号"
    }
}
```

手作業で、Mappingに固定値制約の値をつけて作成しなおす

```
{
    "mapping": {
        "名称": "ic:法人型>ic:名称",
        "営業時間_電話番号": "ic:法人型>ic:連絡先 [営業時間] {ic:説明='営業時間'}>ic:電話番号",
        "早朝深夜_電話番号": "ic:法人型>ic:連絡先 [早朝深夜] {ic:説明='早朝深夜'}>ic:電話番号"
    }
}
```

このDMDと手作業で作成したMappingを使用してデータ形式変換をした結果

20180801_formZZ.csv - Excel									
ファイル	ホーム	挿入	ページレイアウト	数式	データ	校閲	表示	開発	ACROBAT
C3	: X ✓ fx	02-2222-2222							
A	B	C	D	E	F	G	H	I	J
1 表記	早朝深夜_電話番号	営業時間_電話番号							
2 いち	01-1111-1111	01-2222-2222							
3 に	02-1111-1111	02-2222-2222							
4									
5									
6									
7									
8									
9									
10									
11									
12									
20180801_formZZ									
準備完了									

20180814_固定値制約



処理結果、入力項目に説明項目は無いが
マッピング先に固定値制約が反映されて
下記のようなRDF/XMLが出力される

```

-<rdf:RDF>
-<rdf:Description>
  <rdf:type rdf:resource="ic:法人型"/>
  <ic:名称 rdf:datatype="ic:名称型">いち</ic:名称>
  -<ic:連絡先 rdf:parseType="Resource">
    <ic:説明>営業時間</ic:説明>
    <ic:電話番号 rdf:datatype="ic:電話番号型">01-2222-2222</ic:電話番号>
  </ic:連絡先>
  -<ic:連絡先 rdf:parseType="Resource">
    <ic:説明>早朝深夜</ic:説明>
    <ic:電話番号 rdf:datatype="ic:電話番号型">01-1111-1111</ic:電話番号>
  </ic:連絡先>
-</rdf:Description>
-<rdf:Description>
  <rdf:type rdf:resource="ic:法人型"/>
  <ic:名称 rdf:datatype="ic:名称型">に</ic:名称>
  -<ic:連絡先 rdf:parseType="Resource">
    <ic:説明>営業時間</ic:説明>
    <ic:電話番号 rdf:datatype="ic:電話番号型">02-2222-2222</ic:電話番号>
  </ic:連絡先>
  -<ic:連絡先 rdf:parseType="Resource">
    <ic:説明>早朝深夜</ic:説明>
    <ic:電話番号 rdf:datatype="ic:電話番号型">02-1111-1111</ic:電話番号>
  </ic:連絡先>
-</rdf:Description>
</rdf:RDF>

```

- ▶ 10.現実の日付けデータに合わない場合の語彙表現がわからない

日付データと語彙の要素が合わない場合

IMIツールで処理する場合、入力データのセルの書式設定は、[文字列]となっている必要があります。

入力データ(xlsx)の項目に書式設定された場合

B	C	D	E	F	G	H
No	指定年月日(和暦1)	指定年月日(和暦2)	指定年月日(グレゴリオ1)	指定年月日(グレゴリオ2)	指定年月日(ユーザ定義1)	指定年月日(ユーザ定義2)
1	H29.5.6	平成29年5月6日	2017/12/5	2017年12月5日	2017年12月5日	2017-12-05
種類(I):	種類(I):	種類(I):	種類(I):	種類(I):	種類(I):	種類(I):
H24.3.14 平成24年3月14日	H24.3.14 平成24年3月14日	*2012/3/14 *2012年3月14日 2012年3月14日 2012年3月 3月14日 2012/3/14 2012/3/14 1:30 PM	*2012/3/14 *2012年3月14日 2012年3月14日 2012年3月 3月14日 2012/3/14 2012/3/14 1:30 PM	*2012/3/14 *2012年3月14日 2012年3月14日 2012年3月 3月14日 2012/3/14 2012/3/14 1:30 PM	yyyy"年"m"月"d"日" \$(#,##0_);[赤](\$#,##0) \$(#,##0.00_);(\$#,##0.00) \$(#,##0.00_);[赤](\$#,##0.00) [\$-411]ge.m.d [\$-411]ggge"年"m"月"d"日" yyyy/m/d yyyy"年"m"月"d"日" yyyy"年"m"月" m"月"d"日" m/d/yy d-mmm-yy	yyyy-mm-dd h'時"mm'分"ss"秒" yyyy/m/d h:mm mm:ss mm:ss.0 @ [h]:mm:ss [\$-411]ggge"年"m"月"d"日";@ [\$-411]ge.m.d;@ [\$-F800]ddd, mmmm dd, yyyy yyyy-mm-dd yyyy"年"m"月"d"日"
ロケール (国または地域)(L):	ロケール (国または地域)(L):	ロケール (国または地域)(L):	ロケール (国または地域)(L):	ロケール (国または地域)(L):	ロケール (国または地域)(L):	ロケール (国または地域)(L):
日本語	日本語	日本語	日本語	日本語	日本語	日本語
カレンダーの種類(A):	カレンダーの種類(A):	カレンダーの種類(A):	カレンダーの種類(A):	カレンダーの種類(A):	カレンダーの種類(A):	カレンダーの種類(A):
和暦	和暦	グレゴリオ暦	グレゴリオ暦	グレゴリオ暦	グレゴリオ暦	グレゴリオ暦

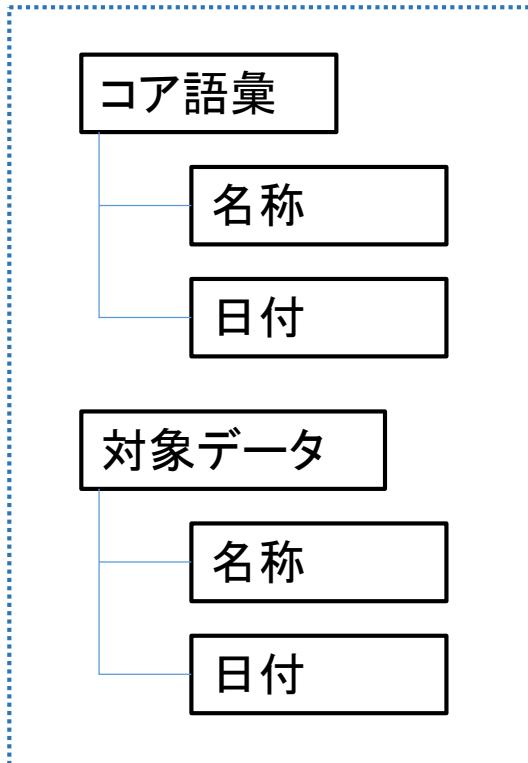
DMD作成支援「表形式データのプレビュー」(データ形式変換ツールも同様)で意図しないデータとなる場合があります。

Nº	指定年月日(和暦1)	指定年月日(和暦2)	指定年月日(グレゴリオ1)	指定年月日(グレゴリオ2)	指定年月日(ユーザ定義1)	指定年月日(ユーザ定義2)
1	g2017.5.6	ggg2017年5月6日	12/5/17	Tuesday, December 05, 2017	43074	2017-12-05

- ▶ 11.既に構造化されたデータセットをどのようにマッピングしたら良いのかわからない

構造化データのマッピング

構造化されたデータ

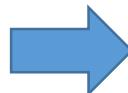


```
{  
mapping:{  
    "コア語彙>名称" : "ic:氏名型>ic:姓名",  
    "コア語彙>日付" : "ic:日付型>ic:標準型日付",  
    "対象データ>名称" : "ic:氏名型>ic:姓名",  
    "対象データ>日付" : "ic:日付型>ic:標準型日付"  
}  
}
```

このようにマッピングファイルとして定義できるが、定義の解釈と構造化データへの変換処理は実装の中で実現しなくてはなりません。

IMI共通語彙基盤ライブラリを使って、構造化データから簡易なマッピングファイルを生成することができます。

```
<rdf:Description>
<rdf:type rdf:resource="ic:施設型"/>
<ic:ID rdf:parseType="Resource">
<ic:識別値>001</ic:識別値>
</ic:ID>
<ic:名称 rdf:parseType="Resource">
<ic:表記>共通語彙基盤</ic:表記>
<ic:カナ表記>キヨウツウゴイキバン</ic:カナ表記>
</ic:名称>
<ic:住所 rdf:parseType="Resource">
<ic:表記>東京都文京区本駒込二丁目28番8号 文京グリーンコートセンター西16階</ic:表記>
<ic:方書>総合受付13階</ic:方書>
</ic:住所>
<ic:地理座標 rdf:parseType="Resource">
<ic:緯度>34.562</ic:緯度>
<ic:経度>145.26</ic:経度>
<ic:測地高度 rdf:parseType="Resource">
<ic:数値 rdf:datatype="xsd:decimal">12.5</ic:数値>
</ic:測地高度>
</ic:地理座標>
<ic:連絡先 rdf:parseType="Resource">
<ic:電話番号 rdf:datatype="ic:電話番号型">03-5978-7501</ic:電話番号>
<ic:内線番号>2613</ic:内線番号>
</ic:連絡先>
<ic:関与 rdf:parseType="Resource">
<ic:役割>設置主体</ic:役割>
<ic:関与者 rdf:parseType="Resource">
<rdf:type rdf:resource="ic:組織型"/>
<ic:ID rdf:parseType="Resource">
<ic:識別値>13105</ic:識別値>
```



IMIToolsCore.Util.jsonId2mapping

開発ツール - IMI共通語基盤ライブラリ - file:///C:/語彙基盤/20181221_12月IMI事例共有・勉強会/No11_構造化データのマ... □ X

□ インスペクター □ コンソール □ デバッガー {} スタイルエディター ⌂ パフォーマンス ⌂ メモリー » □ ...

出力を継ぎ込み 永続ログ

エラー 善告 ログ 情報 テバッガ CSS XHR 要求

Mapping_Convert.js:117:1

```
mapping: {_}
  key0: "ic:施設型>ic:ID>ic:識別値"
  key1: "ic:施設型>ic:名称>ic:表記"
  key10: "ic:施設型>ic:閲与>ic:役割"
  key11: "ic:施設型>ic:閲与>ic:閲与者{@id:組織型}>ic:ID>ic:識別値"
  key12: "ic:施設型>ic:閲与>ic:閲与者{@id:組織型}>ic:住所>ic:都道府県"
  key13: "ic:施設型>ic:閲与>ic:閲与者{@id:組織型}>ic:住所>ic:市区町村"
  key14: "ic:施設型>ic:閲与>ic:閲与者{@id:組織型}>ic:名称>ic:表記"
  key15: "ic:施設型>ic:種別コード[1]>ic:識別値"
  key16: "ic:施設型>ic:種別コード[2]>ic:識別値"
  key17: "ic:施設型>ic:種別コード[3]>ic:識別値"
  key18: "ic:施設型>ic:種別コード[4]>ic:識別値"
  key19: "ic:施設型>ic:種別コード[5]>ic:識別値"
  key2: "ic:施設型>ic:名称>ic:カナ表記"
  key20: "ic:施設型>ic:種別コード[6]>ic:識別値"
  key21: "ic:施設型>ic:種別コード[7]>ic:識別値"
  key22: "ic:施設型>ic:種別コード[8]>ic:識別値"
  key23: "ic:施設型>ic:種別コード[9]>ic:識別値"
  key24: "ic:施設型>ic:記述>ic:種別"
  key25: "ic:施設型>ic:記述>ic:説明"
  key26: "ic:施設型>ic:参照>ic:参照先{@xsd:anyURI}"
  key27: "ic:施設型>ic:備考"
  key3: "ic:施設型>ic:住所>ic:表記"
  key4: "ic:施設型>ic:住所>ic:方書"
  key5: "ic:施設型>ic:地理座標>ic:緯度"
  key6: "ic:施設型>ic:地理座標>ic:経度"
  key7: "ic:施設型>ic:地理座標>ic:測地高精度>ic:数値{@xsd:decimal}"
  key8: "ic:施設型>ic:連絡先>ic:電話番号{@id:電話番号型}"
  key9: "ic:施設型>ic:連絡先>ic:内線番号"
  ▶ <prototype>: Object { _ }
  ▶ <prototype>: Object { _ }
```

- ▶ 12. 座標系のデータセットはどのように語彙設計したらよいのかわからない

座標系のデータの語彙設計

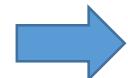
座標データは実装上で様々なフォーマットが利用されている現状である。
地点を表すデータや面で表すデータなどがあり、まだ語彙の構造として確立した状況はない…

RFC7946

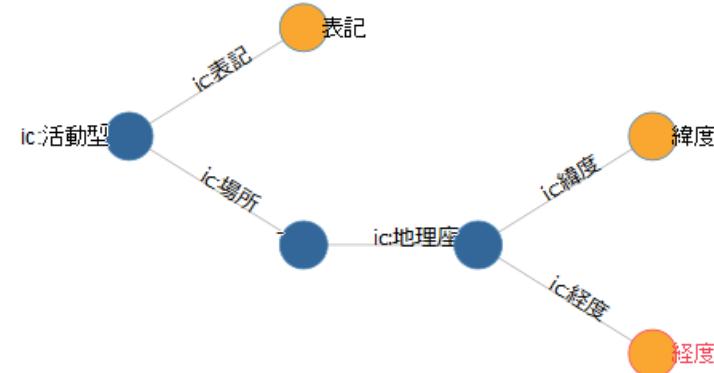
1.	Introduction	3
1.1.	Requirements Language	4
1.2.	Conventions Used in This Document	4
1.3.	Specification of GeoJSON	4
1.4.	Definitions	5
1.5.	Example	5
2.	GeoJSON Text	6
3.	GeoJSON Object	6
3.1.	Geometry Object	7
3.1.1.	Position	7
3.1.2.	Point	8
3.1.3.	MultiPoint	8
3.1.4.	LineString	8
3.1.5.	MultiLineString	8
3.1.6.	Polygon	9
3.1.7.	MultiPolygon	9
3.1.8.	GeometryCollection	9
3.1.9.	Antimeridian Cutting	9
3.1.10.	Uncertainty and Precision	10
3.2.	Feature Object	11
3.3.	FeatureCollection Object	11
4.	Coordinate Reference System	12
5.	Bounding Box	12
5.1.	The Connecting Lines	14
5.2.	The Antimeridian	14
5.3.	The Poles	14
6.	Extending GeoJSON	15
6.1.	Foreign Members	15
7.	GeoJSON Types Are Not Extensible	16
7.1.	Semantics of GeoJSON Members and Types Are Not Changeable	16
8.	Versioning	17

GeoJsonの例

```
{  
  "type": "Feature",  
  "geometry": {  
    "type": "Point",  
    "coordinates": [139.75, 35.72]  
  },  
  "properties": {  
    "name": "災害発生場所"  
  }  
}
```



データモデルに展開してみる



データモデル

```
use ic:活動型>ic:表記;  
  
use ic:活動型>ic:場所>ic:地理座標>ic:緯度;  
  
use ic:活動型>ic:場所>ic:地理座標>ic:経度;
```

このようにデータモデル化できるが、解釈と構造化データとの変換処理は実装の中で実現しなくてはなりません。