



# IMI技術仕様書

## 構造化項目名記法

バージョン 1.0

2018年3月23日

## 1 概要

構造化項目名の記法(以下、構造化項目名記法)は、IMI語彙などの階層構造をもつデータ表現の位置を文字列によって表現するための仕様です。1つ以上の構造化項目名記法を利用することで、階層構造をもつデータを表形式のデータとして表現することができるようになりますため、階層構造をもつデータの表計算ソフトウェアによる編集や、表計算ソフトウェアなどで作成されたデータから階層化されたデータへの効率的な変換ができるようになります。

## 2 構造化項目名記法の基本

構造化項目名は、クラス用語 > プロパティ用語 > プロパティ用語 > . . . のように「>」で区切って一つのクラス用語と任意の数のプロパティ用語を並べたものです。

非常に簡単な例を次に示します。

例 1:

施設型 > 種別

この例では、「施設型」というクラス用語のプロパティである「種別」を使用することを意味しています。ここで、「種別」プロパティの値型は単純な文字列(xsd:string)です(<http://imi.go.jp/ns/core/2#種別>を参照)。この仕様書では、このようなデータの構造を表現する場合に、図1のように円、矢印、長方形を用いた図を用います。ここで、円はクラス用語を、矢印はプロパティ用語を、長方形は単純な値を表します。上の例を図に表したもののが、図1になります。

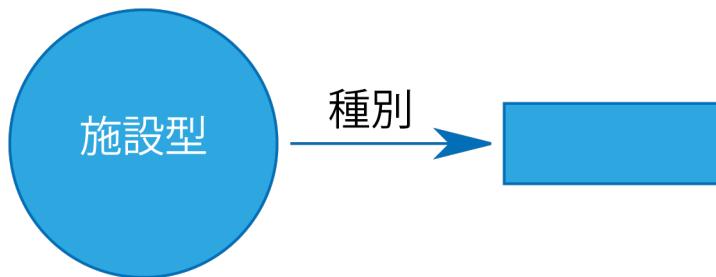


図1: 値型が単純型のプロパティ

参考: クラス用語の「型」について

IMIコア語彙バージョン2では、「人型」のように、すべてのクラス用語の識別子には最後に「型」が付けられています。これは用語を一意に識別するためのもので機械処理を行うために必要です。そのため構造化項目名記法の仕様では、「人型」のように識別子を正確に書くことを求めています。

しかし、単に意味や構造の説明を行うような場合には、人 > 氏名 > 姓のように「型」が省略されることもあり、ツールによっては省略された場合でも「型」を補うなどの対応を行う場合もあります。

しかし、プロパティの値型はいつも単純な値とは限りません。例えば、例2の「地理座標」プロパティの値型は、クラス用語である「座標型」です。

例 2:

施設型 > 地理座標

プロパティの値型がクラス用語である場合は、図 2 のように、プロパティの先はクラス用語になります。しかし、この場合でも構造化項目名には、そのクラス用語を記述する必要はありません。

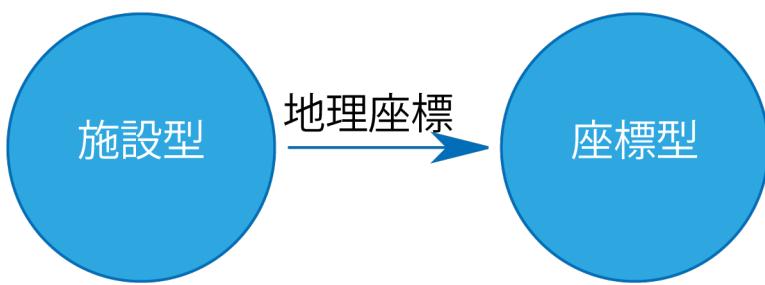


図 2: 値型がクラス用語のプロパティ

このようにプロパティの値型がクラス用語である場合、「地理座標」プロパティの値型である「座標型」にもプロパティがあります。構造化項目名では、「座標型」のプロパティを記述する場合に、例 3 のように「施設型> 地理座標」の後に続けて表記します。

例 3:

施設型 > 地理座標 > 緯度

例 3 を図で表すと、図 3 のようになります。

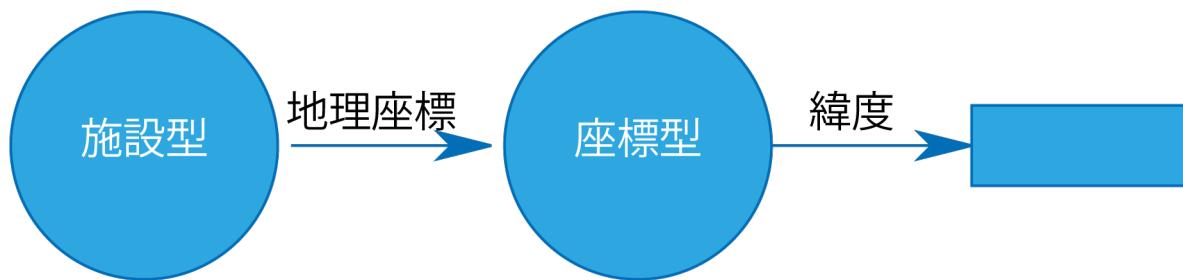


図 3: 階層化したプロパティ

同様に、「人型」には「氏名」プロパティがあり、その値型である「氏名型」には「姓」というプロパティがあります。この「姓」というプロパティは例 4 のように構造化項目名で表現され、図で表すと図 4 のようになります。

例 4:

人型 > 氏名 > 姓

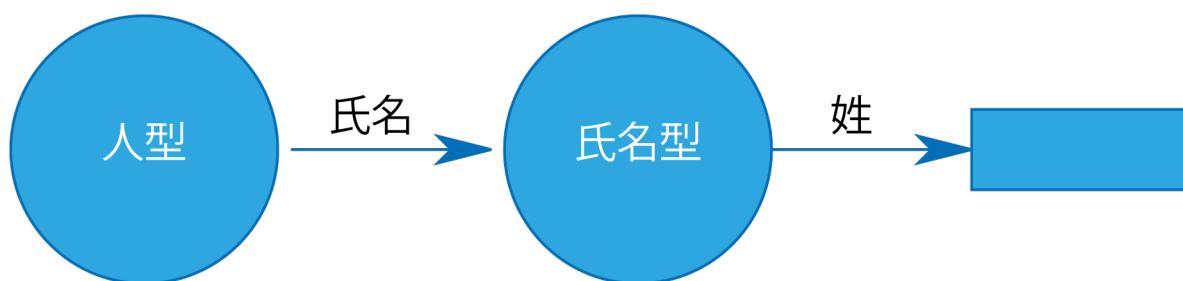


図 4: 階層化したプロパティ (人型の場合)

構造化項目名記法は、構造化項目名を単独で使用するほか、複数の構造化項目名を用いて、データの概念的な構造を表現することができます。

例えば、次の例は、図 5 のような構造を表します。

例 5:

人型>氏名>姓  
人型>氏名>名

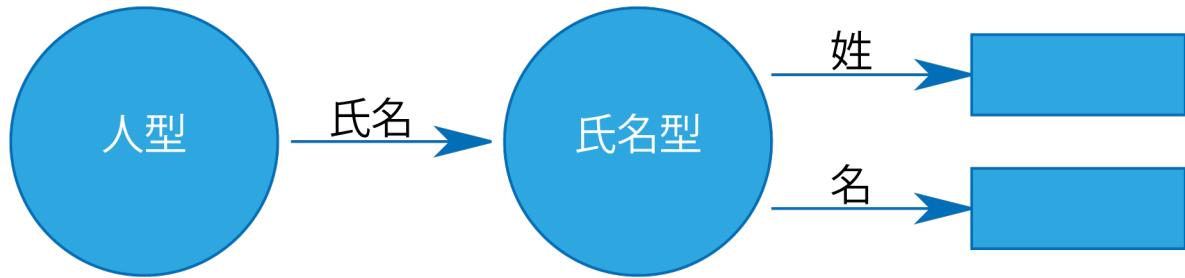


図 5: 構造化項目名による概念的構造の表現

また、次の例のように、複数の構造化項目名を CSV など表形式データのヘッダーラベルとして使用することもできます。

例 6:

人型>氏名>姓, 人型>氏名>名

### 3 グループ名の利用

構造化項目名記法では、直接記述することはありませんが、人型>氏名のように記述した場合には、図 4 のように、氏名プロパティの先に氏名プロパティの値型である氏名型クラスの構造をもつデータがあることを意味します。このように、氏名型クラスの構造をもつデータのことをクラスのインスタンスと呼びます。図 4 のように、氏名型クラスのインスタンスが一つの場合には、人型>氏名>姓のように後ろに続くプロパティは全て唯一の氏名型クラスのインスタンスのプロパティとなります。しかし、人型は氏名プロパティを複数もつことができますので、氏名型クラスのインスタンスは一つとは限りません。このような際には、複数の氏名型クラスのインスタンスを区別する必要があります。構造化項目名記法ではクラスのインスタンスを区別するために、それぞれのクラスに名前を付けることができ、その名前をグループ名と呼びます。グループ名には、任意の文字列を指定することができ、各用語の後ろに「[」と「]」に囲んで記述します。グループ名を使用することで、同じプロパティを複数使う場合に、プロパティの先にあるインスタンスを区別することができます。同じグループ名をもつ用語は同一のインスタンスを表します。

グループ名は構造化項目名上でインスタンスを区別するものであり、構造化項目名をもとに生成される構造化データの実際のインスタンスには反映されません。

グループ名を使用した例を次に示します。

例 7:

人型>氏名[本名]>姓  
人型>氏名[本名]>名  
人型>氏名[芸名]>姓  
人型>氏名[芸名]>名

この例は、図 6 のような階層構造を表しています。ここで、上側の「氏名」は氏名 [本名] を表しています。実際のデータ構造にはグループ名は反映されません。例 5 ではグループ名を指定していませんでしたが、その場合、共通部分は同じインスタンスを表します。例えば、例 5 では、「人型>氏名」が共通ですので、図 4 のように人及び氏名は、同一のインスタンスを表します。

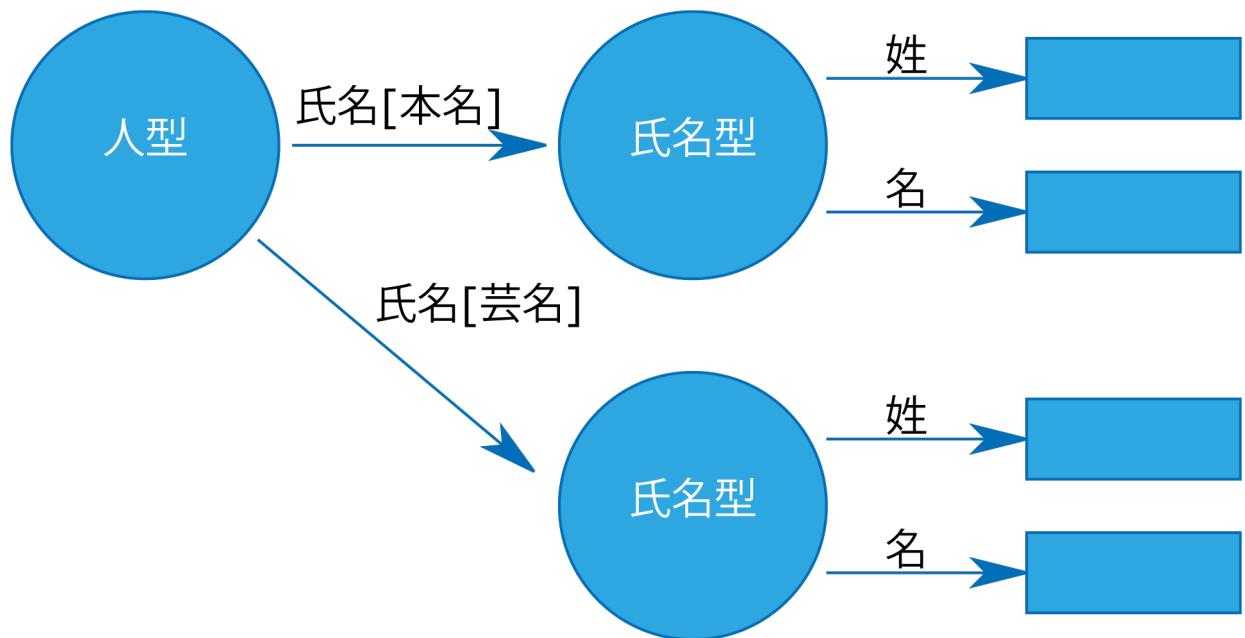


図 6: グループ名

また、グループ名は同じプロパティを一つだけ使用する場合にも付けることができるので、次の例のように単にラベルとして使用することもできます。

例 8:

```

施設型 > 名称 [学校名] > 表記
施設型 > 種別 [分類]
  
```

既存の表形式データに構造化項目名記法を適用する場合などに、既存の項目ラベルをグループ名として使用するなどの使い方が考えられます。

## 4 主キーの設定

データを作成する場合には、データに含まれる事物を正しく参照できるようにするためにデータ中の各事物に一意のキーを与える必要があります。例えば、施設の一覧であれば各施設を一意に特定するためのキーが必要になります。表形式データの行番号を用いることは推奨されません。20行目記載された施設は、途中の行が削除されてしまうと19行目になってしまいます。このように、行番号と各施設の対応は変わってしまうからです。各施設に、将来にわたって変わることがない番号や文字列を与えるのが理想的です。そのような番号や文字列を与えることが難しい場合は、施設の名前や電話番号など、できるだけ変化しないものをキーとして代用します。構造化項目名記法では、先頭に「\*」を付けることでその項目を主キーとすることを指定することができます。

次の例は、施設の名称の表記を主キーとして設定しています。

例 9:

```

*施設型 > 名称 > 表記
施設型 > 名称 > カナ表記
施設型 > 種別
  
```

上記の例の場合、将来にわたって施設の名前が変わらないことが前提になっていることに注意してください。プロパティの値ではなく、主キーとして別途決めた値を利用する場合には、次の例のようにプロパティを指定せずに主キー

を設定することができます。

例 10:

```
*施設型
施設型>名称>表記
施設型>名称>カナ表記
施設型>種別
...
```

## 5 URL 変換

項目の内容を値としてデータに含めるのではなく、インターネット上にあるリソースを参照したい場合があります。例えば、施設の関連施設のデータがすでにインターネット上にある場合、その関連施設の名称、電話番号や住所をデータの中に含めるよりも、その施設のデータを参照した方が効率的です。このように、その項目の内容を URL として扱う必要がある場合があります。構造化項目名記法では、最後を>で終了することで、内容を URL として扱うことを表します。

次に例を示します。

例 11:

```
*施設型
施設型>名称>表記
施設型>関連施設>
```

## 6 名前空間の指定

構造化項目名記法では、用語の名前空間は解釈や変換を行うツールなどによって解決されることを前提としています。そのため、多くの場合、名前空間の指定を行う必要はありません。しかし、異なるドメインに同じ名前の用語があり（共通語彙基盤では、このような重複は、可能な限り避けることが求められています）、どちらも使用したい場合など特殊な場合には、次の例のように、名前空間プレフィックスを指定して区別することができます。構造化項目名記法では、名前空間を指定する際には、用語の前に名前空間プレフィックスおよび ":" を記述します。ただし、構造化項目名記法では、プレフィックスと実際の名前空間の対応を記述する文法は規定されていません。プレフィックスから名前空間への変換はツール等が行うことが前提となります。

次に例を示します。

例 12:

```
固有施設型>A県:施設分類
固有施設型>B市:施設分類
```

## 7 マッピングによるデータ制約と組込み制約

構造化項目名記法を用いて CSV 等の表形式データの列のラベル（データ項目の名称）と IMI 語彙との対応を示すことによって、列の意味を明確にする（意味づけを行う）ことができます。このように、CSV 等の表形式データの列のラベルに IMI 語彙を対応付けることを「マッピング」と呼びます。

マッピングを行うことにより、CSV から構造化されたデータの作成を行うこともできます。

共通語彙基盤では、JSON を用いてマッピングを記述する方法が決められています。マッピングの詳細については、「マッピングファイル仕様」を参照してください。

例えば、次のような CSV の各ヘッダーラベルに対して適切な構造化項目名を指定することで、CSV から構造化されたデータを作成することができます。

例 13:

```
電話番号, FAX, 当日連絡先
000-000-0000, 111-111-1111, 222-222-2222
```

例 13 の CSV にマッピングファイル仕様にしたがって、イベント型にマッピングを行う場合は、例 14 のようになります。

例 14:

```
{
  "mapping": {
    "電話番号": "イベント型 > 連絡先[担当課] > 電話番号",
    "FAX": "イベント型 > 連絡先[担当課] > FAX番号",
    "当日連絡先": "イベント型 > 連絡先[当日連絡先] > 電話番号"
  }
}
```

この例では、図 7 のような対応付けが行われます。

共通語彙基盤では、この例のように 2 種類の連絡先がある場合には、連絡先型の種別プロパティによって、それぞれの用途を表すようになっています。このような共通語彙基盤で期待されるデータ構造を表現するために、構造化項目名記法では、組込み制約と呼ばれる機能が用意されています。

組込み制約には、固定値制約（指定されたプロパティに固定の値を設定する）と型制約（プロパティの値として実際に使用される型を指定する）の 2 種類があります。

## 7.1 固定値制約

固定値制約は、グループ名によって選択されたインスタンスのプロパティに固定の値を設定するために使用されます。

固定値制約では、値を入力したプロパティの項目名、「=」、指定する値（二重引用符で囲む）の順に記述し、その全体を「{」「}」で囲みます。例えば、連絡先型の種別プロパティに「担当」という値を設定する場合には、次の例のように記述します。

例 15:

```
施設型 > 連絡先[担当部署] {種別="担当"}
```

次の例は、固定値制約を用いたマッピングの例です。

例 16:

```
{
  "mapping": {
    "電話番号": "イベント型 > 連絡先[担当課]{種別='担当'} > 電話番号",
    "FAX": "イベント型 > 連絡先[担当課]{種別='担当'} > FAX番号",
    "当日連絡先": "イベント型 > 連絡先[当日連絡先]{種別='現地'} > 電話番号"
  }
}
```

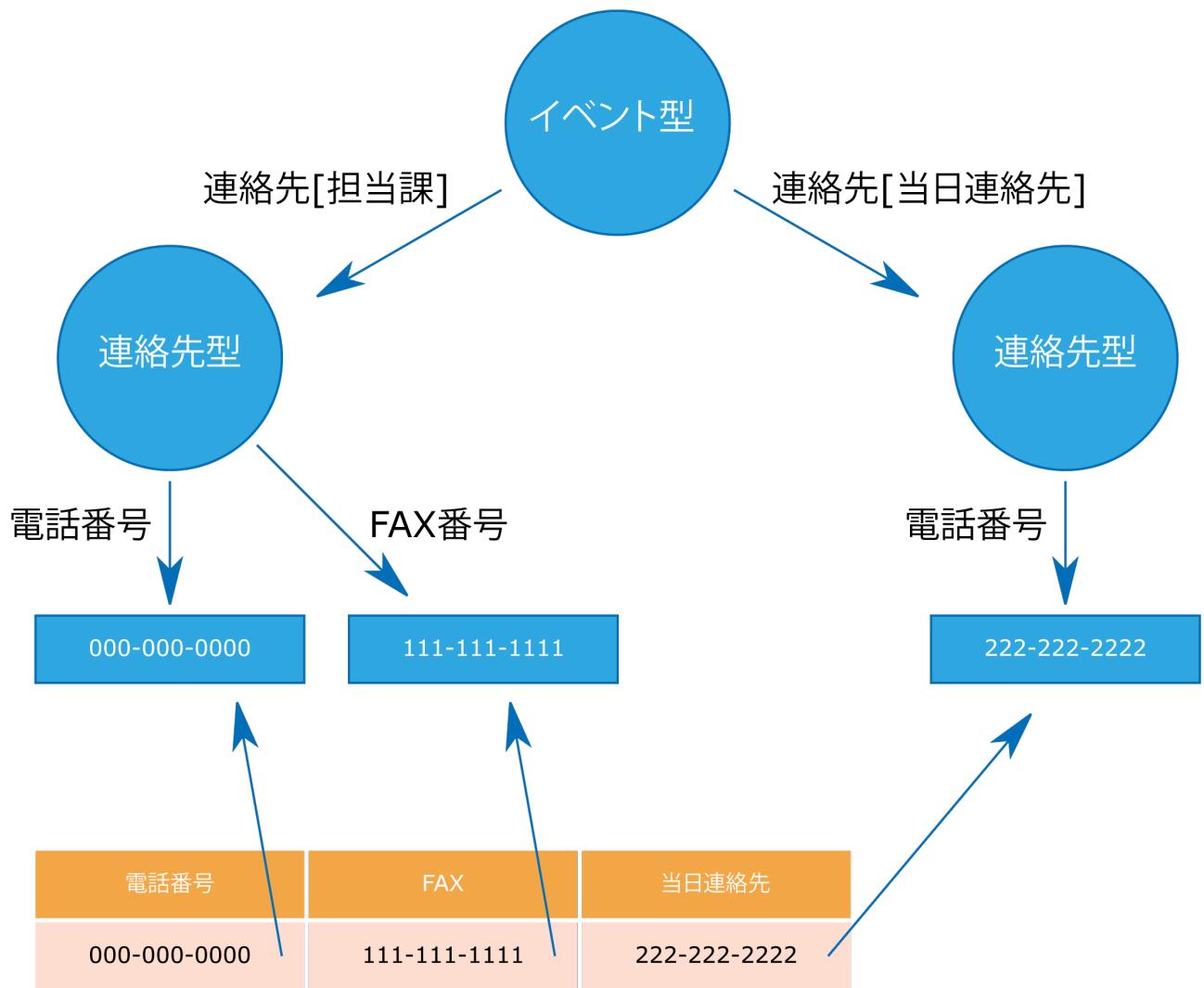


図 7: 基本のマッピング

この例では、図 8 のような構造を表します。

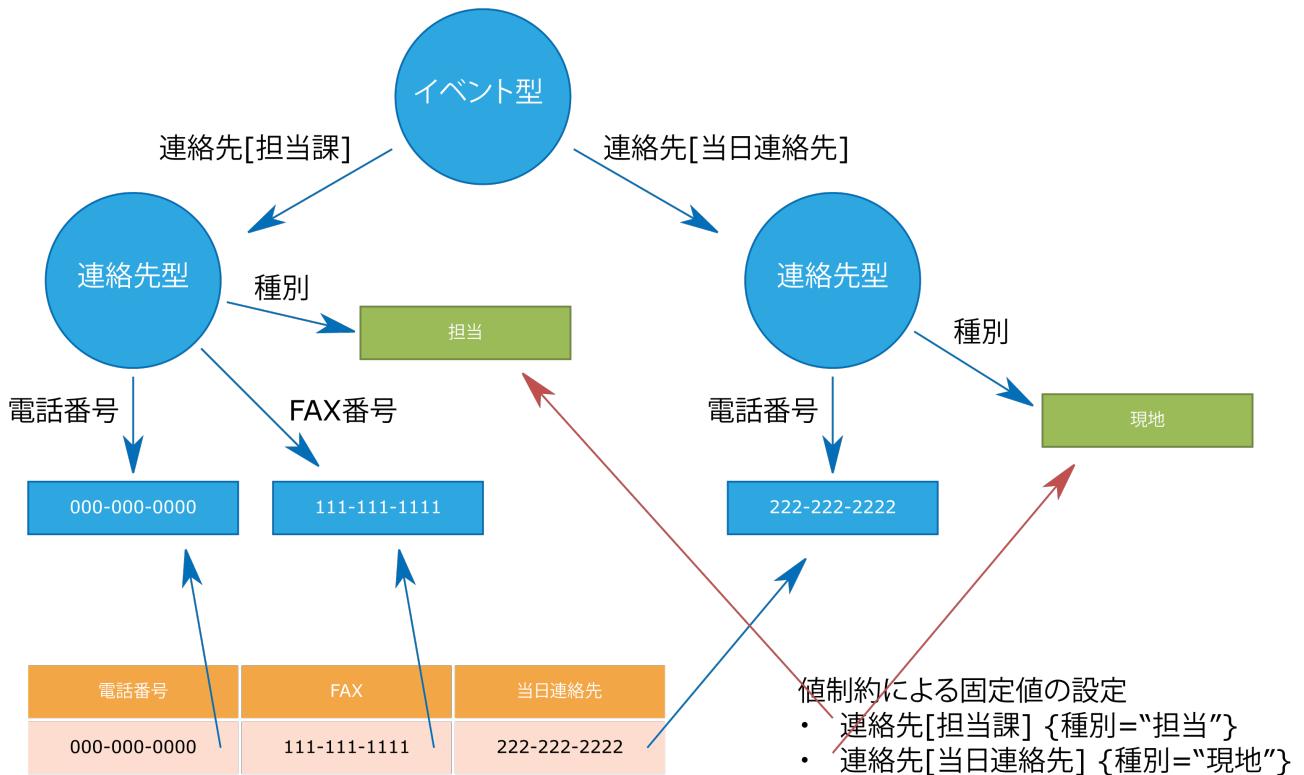
## 7.2 型制約

共通語彙基盤では、プロパティの値に、値型として指定されている型だけでなく、その型を継承するすべての型を指定することができます。型制約は、グループ名によって選択されたインスタンスのプロパティの値として実際に使用する型を指定するために使用されます。

型制約では、「@」に続けて型の項目名を記述し、その全体を「{」「}」で囲みます。例えば、イベント型の期間プロパティの値型は、期間スケジュール型となっていますが、実際のデータでは、期間スケジュール型を継承するイベントスケジュールを使用したい場合には、次の例のように記述します。

例 17:

イベント型 > 期間 {@イベントスケジュール型} > 開催日 > 表記



## 8 解釈

この章では、構造化項目名記法にしたがって記述された一つ又は複数の文字列を構造化されたデータに変換する際の解釈について説明します。

### 8.1 表記法

構造化項目名記法の文法は、字句規則及び構文規則により定義される。字句規則及び構文規則を合わせて「文法規則」と呼ぶ。字句規則は、言語の最小字句要素である字句を構成する文字の組み合わせを定義する。構文規則は、言語を構成する字句の組み合わせを定義する。

対象となる文字の並びが複数の規則に一致する場合は、最も長く一致する規則を選択する。

字句規則及び構文規則に用いる構文要素には、構造化項目名内にそのまま記述される終端記号、字句規則にしたがった文字列として記述される非終端記号、及び、文字の3種類がある。

終端記号は二重引用符 (") で囲んで表記し、非終端記号は <> で囲んで表記する。

例えば、次の例において、<> 言語指定句 > 及び <> 識別子 > は非終端記号で、"@" は終端記号である。

例 18:

```
<> 言語指定句 > := "@" <> 識別子 >
```

文字は、Unicode 文字表記 (8.1.8) 又は一重引用符 (') で囲んで表記する。

字句規則の定義では、“::=” の左辺に定義する非終端記号を、右辺に左辺の非終端記号の定義となる非終端記号及び文字の列を配置する。

構文規則の定義では、“:=” の左辺に定義する非終端記号を、右辺にその定義となる非終端記号及び終端記号の列を配置する。

構文規則における右辺の字句と字句との間の《空白文字》は構文規則の表記上は省略する。例えば、上記の例は、次の例の構文規則と同じ意味となり、構造化項目名記法では、“>”と《整数リテラル》の間に任意の数の《空白文字》を置くことができる。

例 19:

```
《言語指定句》 := "@" 《空白文字》* 《識別子》
```

対象の IMI 語彙記法による文字列が複数の文法規則に合致する場合には、より長い文法規則を選択するものとする。

文法規則の表記にでは、字句の列を表現するために 8.1.1 から 8.1.8 で説明する表記法を用いる

### 8.1.1 0回以上の繰り返し

“\*”は、直前の文字、字句又はグループを 0 回以上繰り返すことを意味する。

次の例は、《識別子》が、《識別子先頭文字》の後ろに《識別子文字》が 0 個以上続く文字列であることを意味する。

例 20:

```
《識別子》 ::= 《識別子先頭文字》 《識別子文字》*
```

### 8.1.2 1回以上の繰り返し

“+”は、直前の文字、字句又はグループを 1 回以上繰り返すことを意味する。

次の例は、《クラス項目》の後に (“>”《プロパティ項目》) が 1 個以上続くことを意味する。

例 21:

```
《クラス項目》 ( ">" 《プロパティ項目》 )+
```

### 8.1.3 省略可

“?”は、直前の文字、字句又はグループが省略可能であることを意味する。

次の例の、《主キー指定子》、《言語指定句》及び《URL 変換子》は省略可能であることを意味する。

例 22:

```
《主キー指定子》? 《クラス項目》 ( ">" 《プロパティ項目》 )+ 《言語指定句》? 《URL 変換子》?
```

### 8.1.4 選択

複数の文字、字句または字句の列を “|” によって区切り “(” 及び “)” で囲んだものを選択という。選択は、“|” で区切られた複数の文字、字句又は字句の列から一つを選択することを意味する。右辺全体が選択となる場合は、“(” 及び “)” は省略する。

次の例は、« 文字列リテラル » が、« 二重引用符文字列リテラル » 又は« 一重引用符文字列リテラル » のいずれかであることを意味する。

例 23:

```
« 文字列リテラル » ::= ( « 二重引用符文字列リテラル » | « 一重引用符文字列リテラル » )
```

### 8.1.5 グループ

字句又は字句の列を “(” と “)” で囲んだものをグループという。グループは、繰り返し等の表記において、グループ全体が一つの字句のように扱われることを意味する。“(” 及び “)” で囲まれた選択（8.1.4 参照）、文字範囲選択（8.1.7 参照）、除外（8.1.6 参照）もグループと同様に扱われる。

### 8.1.6 除外

字句又は文字の列の後ろに、“-” に続けて字句、文字、又は、選択（8.1.4 参照）を記述し、“(” 及び “)” でその全体を囲んだものを除外という。除外は、“-” の右側が字句、又は、文字の場合、“-” の左側の字句又は文字の列のうち、“-” の右側の字句、又は、文字を含まないものであることを意味する。“-” の右側が選択の場合は、“-” の左側の字句又は文字の列のうち、選択に含まれる全ての文字、又は、字句を含まないものであることを意味する。選択に字句の列を指定することはできない。

次の例は、« 文字 » が 0 回以上繰り返す文字の列であるが、“\*”/“/”を含まないことを意味する。

例 24:

```
( « 文字 » * - "*/" )
```

### 8.1.7 文字範囲選択

文字と文字との間を “-” でつなぎ、“[”及び”]” でその全体を囲んだものを文字範囲選択という。文字範囲選択は、Unicode コードポイントが、“-” の左の文字の Unicode コードポイント以上で、“-” の右の文字の Unicode コードポイント以下である文字の中から一つの文字を選択することを意味する。例次の例では、« 数字 » が、'0'、'1'、'2'、'3'、'4'、'5'、'6'、'7'、'8'、及び、'9' の中の一つであることを意味する。

例 25:

```
« 数字 » ::= ['0'-'9']
```

### 8.1.8 Unicode 文字表記

“U+” に続けて 4 衔から 6 衔の 16 進数を記述したものを Unicode 文字表記という。Unicode 文字表記は、指定された 16 進数で表される値が Unicode コードポイントである文字を意味する。

例えば、U+0041 は、'A' を表す。

## 8.2 概要

構造化項目名記法では、« 構造化項目名 » の集合としてデータ全体の構造を表現する。« 構造化項目名 » は次のような構文によって表現される。

```
«構造化項目名» := «主キー指定子» «クラス項目» | «主キー指定子»? «クラス項目» ( ">" «...»
↳ プロパティ項目» )+ «言語指定句»? «URL変換子»?
```

各«構造化項目名»における«クラス項目»、及び、それに続く複数の«プロパティ項目»によって構成される列はデータ構造中のクラス用語又は値を参照する。

例えば、構造化項目名記法により表現されるデータ構造が図 9 のようなものであるとき、「人型>氏名」は氏名型のクラス用語を参照し、「人型>氏名>姓」は xsd:string 型の値を参照する。

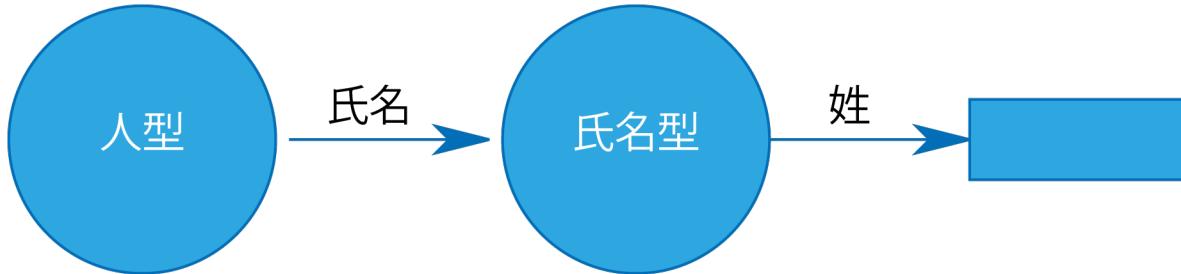


図 9: 構造化項目名による構造表現

複数の«構造化項目名»があるとき、先頭からの項目の並びが共通である部分は同一のインスタンスを参照する。例えば、図 9 の例において、「人型>氏名>姓」と「人型>氏名>名」の 2 つの«構造化項目名»の共通部分である「人型>氏名」は同一の氏名型のインスタンスを参照する。

«構造化項目名»が«主キー指定子»をもつ場合、その項目の値が主キーとなる。主キーの指定には、主キー専用の項目を設ける方法とプロパティの値を主キーとして使用する方法の二つの方法が用意されている。

«構造化項目名»の構文のうち「«主キー指定子» «クラス項目」」の形式の構文では、その項目は主キー専用の値となりデータの値としては利用されず、RDF 等に変換される場合にリソースの識別子として利用される。

その他の構文では、その«構造化項目名»によって参照される値が主キーとなる。

«構造化項目名»に«言語指定句»が指定された場合、«言語指定句»の«言語名»は、JIS X 0412-1 の規定にもとづき表記された 2 つの英字からなる言語名として解釈され、その«構造化項目名»によって参照される値の言語となる。

### 8.3 クラス項目、及び、プロパティ項目

クラス項目とプロパティ項目（本節では、クラス項目とプロパティ項目を合わせて「項目」と呼ぶ）は、構文上の違いはなく、先頭に現れた項目がクラス項目、それに続く複数の項目がプロパティ項目として解釈される。

```
«クラス項目» := «名前空間プレフィックス指定子»? «識別子» «グループ指定子»? «組込み制約»...
↳ »* «プロパティ項目» := «名前空間プレフィックス指定子»? «識別子» «グループ指定子»? «組込み制約»*
```

構造化項目名記法では、名前空間は指定されない。名前空間の解決は構造化項目名記法を解釈するツール等に委ねられる。構造化項目名記法を解釈するツールは、あらかじめ適切に優先順位付けされた特定の語彙を参照する名前空間のリストをもっていることが求められる。«クラス項目»と«プロパティ項目»とで名前空間の解決の仕組みに違いはないので、ここでは、«クラス項目»を例にツールに期待される名前空間の解決方法を説明する。«クラス項目»に«名前空間プレフィックス指定子»がない場合、前述の名前空間のリストの優先順位にしたがって、名前空間に«

クラス用語名》に一致する用語が定義されているかどうかを調べていき、最初に見つかった用語を《クラス項目》として採用する。与えられたすべての名前空間によって参照される語彙に一致する用語が見つからなかった場合はエラーとなる。項目に《名前空間プレフィックス指定子》が指定されている場合には、上記の優先順位によらず、ツールにあらかじめ与えられた名前空間プレフィックス名と名前空間の対応リストにより名前空間を決定する。指定された名前空間によって参照される語彙に該当する用語が見つからなかった場合はエラーとなる。《プロパティ項目》についても同様な方法によって名前空間が解決される。

項目に《グループ指定子》が指定されている場合、《クラス項目》及びそれに続く《プロパティ項目》のすべてが同一でかつ《グループ名》が一致する場合に同一のインスタンスを表すものとして解釈される。例えば、人型>氏名 [本名]>姓、人型>氏名 [本名]>名、人型>氏名 [芸名]>姓、人型>氏名 [芸名]>名 の解釈では、図 10 のように二つに共通して現れる、人型>氏名 [本名] は同一のインスタンスとして解釈され、人型>氏名 [本名] と人型>氏名 [芸名] は異なるインスタンスとして解釈される。

同一《グループ名》をもつものが存在しなくてもエラーとはならず、その場合は解釈に影響を与えない。

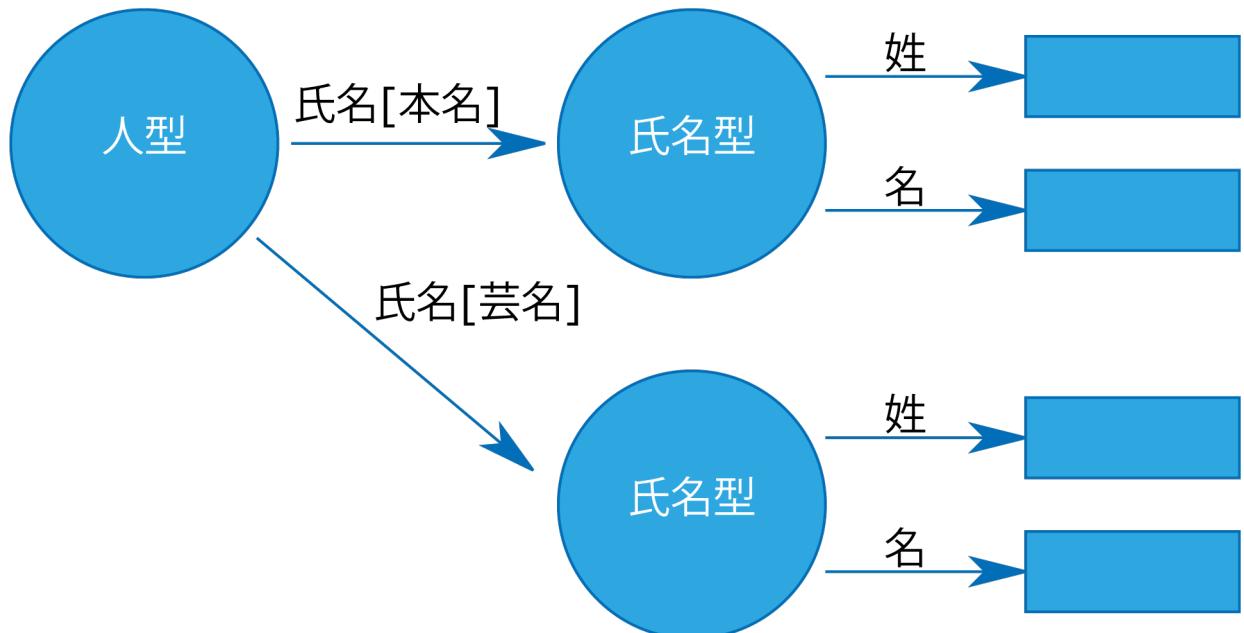


図 10: グループ名による構造表現

## 8.4 URL 変換

《構造化項目名》が「《主キー指定子》? 《クラス項目》《プロパティ列》 + 《言語指定句》? 《URL 変換子》」という形式の場合は、参照される値は値ではなく他のリソースを参照する URL として解釈される。

《構造化項目名》が「《主キー指定子》《クラス項目》」という形式の場合は、《クラス項目》のリソース ID となる URL として解釈される。

## 8.5 固定値制約

《固定値制約》は《識別子》で指定されたプロパティ項目を対象となるインスタンスに追加する。構造化項目名記法をマッピングに用いデータ変換等を行う場合は、対象となるすべてのインスタンスの《識別子》で指定されたプロパティ項目には《文字列リテラル》で指定された値が適切に変換されて設定される。固定値制約は、一つの対象に対して複数設定できる。

## 8.6 型制約

« 型制約» が指定されたプロパティ項目の値は、« 識別子» で設定された型となる。後ろに続くプロパティ項目は、指定された型がもつプロパティの一つでなければならない。« 型制約» は一つの対象に対して 1 回だけ指定できる。

## 9 字句構造

```

« 構造化項目名 » := « 主キー指定子 » « クラス項目 »
| « 主キー指定子 »? « クラス項目 » ( ">" « プロパティ項目 » )+ « 言語指定句 »? « URL変換子 ...
↳ »? */

« クラス項目 » := « 名前空間プレフィックス指定子 »? « 識別子 » « グループ指定子 »? « 組込み制約 »*
« 名前空間プレフィックス指定子 » := « 識別子 » ":""

« グループ指定子 » := « グループ名リテラル »

« 組込み制約 » := "{" ( « 固定値制約 » | « 型制約 » ) "}"

« 固定値制約 » := « 名前空間プレフィックス指定子 »? « 識別子 » "=" « 文字列リテラル »

« 型制約 » := "@" « 名前空間プレフィックス指定子 »? « 識別子 »

« プロパティ項目 » := « 名前空間プレフィックス指定子 »? « 識別子 » « グループ指定子 »? « 組込み制 ...
↳ 約 »*

« 主キー指定子 » := "*"

« URL変換子 » := ">"

« 言語指定句 » := "@" « 識別子 »

« 文字列リテラル » := ( « 二重引用符文字列リテラル » | « 一重引用符文字列リテラル » )

« IMI項目名 » ::= « 識別子 »

« 二重引用符文字列リテラル » ::= U+0022 ( ( « 文字 » - U+0022 ) | « 二重引用符エスケープ » )* U ...
↳ +0022

« 二重引用符エスケープ » ::= U+005C U+0022 | U+005c U+005c

« 一重引用符文字列リテラル » ::= U+0027 ( ( « 文字 » - U+0027 ) | « 引用符エスケープ » )* U+0027

« 一重引用符エスケープ » ::= U+005C U+0027 | U+005c U+005c

« グループ名リテラル » ::= U+005B ( « 文字 » - U+005D )* U+005D

« 識別子 » ::= « 識別子先頭文字 » « 識別子文字 »*
« 識別子先頭文字 » ::= ['A'-'Z'] | '_' | ['a'-'z'] |
[U+00C0-U+00D6] | [U+00D8-U+00F6] | [U+00F8-U+02FF] |
[U+0370-U+037D] | [U+037F-U+1FFF] | [U+200C-U+200D] |
[U+2070-U+218F] | [U+2C00-U+2FEF] | [U+3001-U+D7FF] |
[U+F900-U+FDCE] | [U+FDFO-U+FFFD] | [U+10000-U+EFFF]

« 識別子文字 » ::= « 識別子先頭文字 » | '-' | '.' | ['0'-'9'] |
U+00B7 | [U+0300-U+036F] | [U+203F-U+2040]

« 空白文字 » ::= U+0009 | U+000A | U+000D | U+0020

```

```
« 文字 » ::= U+0009 | U+000A | U+000D | [U+0020-U+D7FF]
| [U+E000-U+FFFD] | [U+10000-U+10FFFF]
```

## この文書について

この文書は、「IMI 共通語彙基盤」の技術的な要件をとりまとめた技術仕様書のひとつです。

表題	構造化項目名記法
バージョン	1.0
公開日	2018 年 3 月 23 日
作成者	独立行政法人情報処理推進機構 (IPA) 技術本部国際標準推進センター IMI 検討部会
発行者	独立行政法人情報処理推進機構 (IPA) (法人番号 5010005007126)

## この文書のご利用にあたって

### 著作権

この文書は、IPA が著作権を持ち、CC0 1.0 全世界 (<https://creativecommons.org/publicdomain/zero/1.0/legalcode.ja>) で公開します。

### 免責事項

本書の内容を適用した結果生じたこと、また適用できなかった結果について、IPA 及び IMI 検討部会は、一切の責任を負いませんのでご了承ください。

### ご意見を募集しています

広くみなさまのご意見を募集しています。以下ご意見投稿のページに進み、ご記入ください。

<https://imi.go.jp/783/>

## この文書の改定履歴

---

- 2018 年 3 月 23 日 StructuredItemName\_V10\_20180323.pdf 発行