

Linked Data における述語の意味を規定する方法

A Method for Specifying the Meaning of Predicates in Linked Data

山田隆弘¹

Takahiro Yamada¹

¹宇宙航空研究開発機構 宇宙科学研究所

¹Japan Aerospace Exploration Agency / Institute of Space and Astronautical Science

Abstract: There are many research activities concerning Linked Data, but Linked Data are not actively utilized for many purposes. One of the reasons is that it is not easy to express various kinds of information as Linked Data. Another reason is a lack of standard vocabularies for many domains. It is also desirable to specify the meaning of vocabularies in a formal way so that the meaning of Linked Data can be processed automatically. This paper proposes a method for specifying the meaning of vocabularies (in particular, predicates) used in Linked Data by using object-oriented modeling technologies and a formal specification language. With this method, standard vocabularies, with a formal specification of their meaning, can be generated for many domains, and the meaning of Linked Data generated with these vocabularies can be processed automatically to some extent.

1 はじめに

機械処理可能なデータを Linked Data [1]として構築するための試みが盛んに行われている。しかし、現状では Linked Data が様々な目的に盛んに利用されているとは言い難い。例えば、Linked Data の代表例である DBpedia の日本語版である日本語 DBpedia の利用状況を分析した資料[2]によると、74%の利用者は日本語 DBpedia を1日だけ利用し、それ以降は全く利用していない。これは、ほとんどの利用者は日本語 DBpedia を試しに使ってみただけであることを示している。

Linked Data の利用が盛んに行われていない理由にはいろいろなものがあるだろうが、その理由の一つとして、Linked Data によって表されている情報の質と量が利用者の期待に達していないことがあると思われる。Linked Data によって表されている情報の質を高めるためには、より多くの種類の情報を Linked Data として表現できるような技術の開発が必要である。また、Linked Data として表すことのできる情報の種類が増えれば、必然的に Linked Data として表される情報の量も増えていくと思われる。

Linked Data の利用が盛んに行われていないもう一つの理由は、Linked Data で使用される語彙が必ずしも統一されていないことであると考えられる。Linked Data は「主語+述語+目的語」として構成されるトリプルの集合であるが、そこで使用される語

彙が多くの Linked Data で統一されていない限り、多くの Linked Data を結合して検索や処理を行うことはできない。多くの Linked Data で標準的な語彙（あるいは、標準的な語彙に基づいて定義された語彙）が用いられれば、多くの Linked Data を結合して利用することが可能になり、Linked Data の有用性は著しく向上すると思われる。

Linked Data 用の標準語彙としては、Dublin Core [3]、FOAF [4]、schema.org [5]等が存在し、Linked Data で実際に活用されているが、様々な分野の語彙を網羅しているとは言い難い。Linked Data の利用を盛んにするためには、さらに多くの語彙を標準化し、多くの Linked Data で共通の語彙が使えるようにする必要がある。

Linked Data で使用する語彙を定める時に重要なことは、単に語彙を定めるだけでなく、語彙の意味を可能な限り形式的に規定することである。語彙の形式的な意味規定が存在すれば、そのような語彙を用いて作成された Linked Data は、意味に関する処理がある程度は機械的に行えるようになる。

本論文では、言語学の意味論で使用されている語彙概念構造[6][7]の手法を用いて Linked Data で使用される語彙（特に述語）の意味を規定する方法を提案する。また、形式的な言語を用いて意味を規定する方法も提案する。このように規定された語彙を標準語彙として使用すれば、多くの Linked Data を結合して使用できるようになる。さらに、語彙の形式的

な意味規定を参照することによって Linked Data の意味を機械的に処理することもある程度は可能になる。

2 Linked Data における語彙の意味

2.1 語彙の意味

「語彙の意味とは何であるか？」というのは、非常に難しい問題であるが、ここでは、Linked Data における語彙の意味を次のように考えることにする。まず、該当分野においてドメインモデル（あるいはドメインオントロジー）が既に確立されているものとし、そのドメインモデルで使用されている語彙の意味は、その分野においては自明であり、形式的に定義する必要はないものとする。そして、Linked Data における語彙の意味は、そのドメインモデルで使用されている概念を使用して規定されるものとする。

以下では、ドメインモデルはソフトウェアの分野で使用されているオブジェクト指向モデリング[8]の概念に基づいて規定されるものとし、ドメインモデルに基づいて語彙の意味を規定する方法としては、語彙概念構造[6][7]をオブジェクト指向化したもの[9]を使用する。

2.2 簡単な Linked Data の例

簡単な例を用いて説明しよう。以下では、Linked Data は RDF [10]で記述され、turtle [11]によって表記されるものとする。また、ex という名前空間は、この例のために使用する仮のものである。(1)等の番号は、便宜的に付与したトリプルの番号であり、トリプルの中には含まれない。

ここでは、以下の Linked Data を考える。

ex:太郎 rdf:type ex:人間 . (1)

ex:花子 rdf:type ex:人間 . (2)

ex:夕日新聞 rdf:type ex:組織 . (3)

ex:太郎 ex:生年月日 "1964-10-15" . (4)

ex:太郎 ex:住所 "町田市" . (5)

ex:太郎 ex:妻 ex:花子 . (6)

ex:太郎 ex:雇用主 ex:夕日新聞 . (7)

(1)のトリプルは、「太郎は人間というタイプである」ことを表している。これをオブジェクト指向語彙概念構造[9]を用いて表現すると「太郎オブジェクトは人間クラスのオブジェクトである」となる。(2)と(3)も同様に表現できる。

(4)のトリプルは、「太郎の生年月日は 1964-10-15 である」ことを表す。これをオブジェクト指向語彙概念構造で表現すると「太郎オブジェクトの生年月日アトリビュートの値は 1964-10-15 である」となる。(5)も同様である。

(6)は、「太郎の妻は花子である」ことを表すが、オブジェクト指向語彙概念構造では「太郎オブジェクトの妻は花子オブジェクトである」となる。(7)も同様である。

(4)～(7)で使われている述語の意味を形式的に規定する方法については、3節で論ずる。

2.3 語彙とドメインモデルとの対応

上の Linked Data で使用されている語彙は、リテラル（例えば、1964-10-15）と既存の規格で規定されている語彙（例えば、rdf:type）を除けば、すべて図 1 のドメインモデルに現れる概念（すなわち、クラス、アトリビュート、アソシエーション）に対応している。

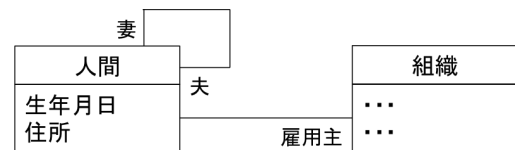


図 1：人間と組織に関するドメインモデルの例

図 1 はオブジェクト指向モデル化言語である Unified Modeling Language (UML) [12]のクラス図の記法を用いて描いたものであり、以下のことを表している。

- ・人間と組織という二つのクラスが存在する。
- ・人間クラスのオブジェクトは生年月日と住所という二つのアトリビュートを有する。
- ・人間クラスの二つのオブジェクトの間に夫-妻という関係（アソシエーション）を設定できる。
- ・人間クラスのオブジェクトと組織クラスのオブジェクトとの間に従業員-雇用主という関係（アソシエーション）を設定できる。

また、図 1 は、記述論理[13]の用語を用いると TBox に相当している。

上で述べた Linked Data の語彙とドメインモデルの対応のさせ方を規則としてまとめると以下のようになる。

〔語彙制定規則 1〕トリプルに現れる主語あるいは述語は、(リテラルを除いて)ドメインモデルのクラスあるいはインスタンスに対応すべきである。

〔語彙制定規則 2〕静的な状態を表すトリプルの述語は、(既存の規格で規定されているものを除い

て) ドメインモデルに現れるアトリビュートあるいはアソシエーションに対応すべきである。

ところで、上の規則は自明のものであり、多くの語彙の制定において既に使用されているものである。しかし、このような規則を語彙制定のための規則として陽に規定することは重要であると思われる。

ちなみに、(1)~(7)のトリプルはインスタンスモデル(記述論理の用語では ABox)として UML のオブジェクト図の記法を用いて図 2 のよう表せられる。

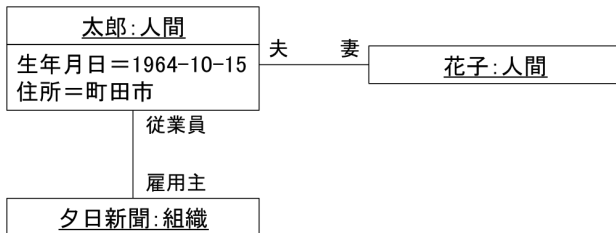


図 2 : (1)~(7)に対応するインスタンスモデル

3 事象を表す述語の意味の規定

前節では静的な状態を表す述語を対象としたが、本節では動的な事象を表す述語の意味の規定について検討する。

3.1 物の生成

例えば、「太郎は 1964 年 10 月 15 日に鎌倉市で生まれた」を Linked Data として表現することを考える。[14]で推奨されている Blank Node を使う方法で Linked Data を作ると、例えば以下ようになる。

ex:太郎 ex:生まれる
 [ex:生年月日 “1964-10-15”]
 [ex:住所 “町田市”] . (8)

上記の「生年月日」と「住所」は、図 1 のドメインモデルに現れるアトリビュートであるが、「生まれる」の意味はどのように規定され得るだろうか。オブジェクト指向語彙概念構造[9]を用いると、「太郎は 1964 年 10 月 15 日に鎌倉市で生まれた」は、「人間クラスの太郎オブジェクトが生成され、生年月日アトリビュートの値は 1964 年 10 月 15 日であり、住所アトリビュートの値は鎌倉市である」と表現される。これより「(人間が) 生まれる」の意味を抽出すると「人間クラスのオブジェクトが生成される」となる。つまり、「生まれる」という事象を表す述語の意味も図 1 のドメインモデルに従ってオブジェクトの生成として規定できることになる。

「生まれる」の意味を形式的に規定する方法については、4 節で論ずる。

3.2 関係の生成

次に、「太郎は 2015 年 4 月 1 日に夕日新聞に就職した」を考えよう。これを Linked Data として表すと、例えば以下ようになる。

ex:太郎 ex:就職する
 [ex:雇用主 “夕日新聞”]
 [ex:日付 “2015-04-01”] . (9)

「ex:日付」は、ここで初めて現れた述語であるが、出来事が発生した日付を表す述語であると定義することにする。このようなメタデータに相当する述語は、個々のドメインモデルよりも上位のレベルで規定すべきであるが、これについては、将来の課題とする。

述語「就職する」の意味はどのように規定され得るだろうか。オブジェクト指向語彙概念構造[9]を用いると、「太郎は 2015 年 4 月 1 日に夕日新聞に就職した」は、「2015 年 4 月 1 日に太郎オブジェクトの雇用主は夕日新聞オブジェクトになった」、あるいは、「太郎オブジェクトと夕日新聞オブジェクトとの間に 2015 年 4 月 1 日に従業員-雇用主のリンクが生成された」と表現される。ここで、リンクとはオブジェクト間の関係であり、クラス間の関係であるアソシエーションをインスタンス化したものである。これより「就職する」の意味を抽出すると「人間クラスのオブジェクトと組織クラスのオブジェクトの間に従業員-雇用主のリンクが生成される」となる。つまり、「就職する」という事象を表す述語の意味も図 1 のドメインモデルに従ってオブジェクト間のリンクの生成として規定できることになる。

3.3 物および関係の変化

ここでは、「太郎は 2013 年 7 月 8 日に鎌倉市に引っ越した」を考える。これを Linked Data として表すと、以下ようになる。

ex:太郎 ex:引っ越す
 [ex:住所 “鎌倉市”]
 [ex:日付 “2013-07-08”] . (10)

「引っ越す」の意味はどのように規定されるだろうか。オブジェクト指向語彙概念構造[9]を用いると、「太郎は 2013 年 7 月 8 日に鎌倉市に引っ越した」は

「太郎オブジェクトの住所アトリビュートの値は2013年7月8日に鎌倉市に変更になった」と表現される。これより「引っ越す」の意味を抽出すると「人間クラスのオブジェクトの住所アトリビュートの値が変更される」となる。つまり、「引っ越す」という事象を表す述語の意味も図1のドメインモデルに従ってアトリビュート値の変化として規定できることになる。

3.4 変化を伴わない動作

最後に、「太郎は2015年11月12日にラーメンを食べた」を考える。このためには、図1の人間クラスの定義を拡張し、図3のようにオペレーションとその引数の規定を追加する必要がある。

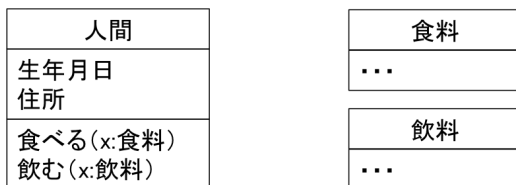


図3：人間に関するドメインモデルの改訂版

図3のモデルを用いると、「太郎は2015年11月12日にラーメンを食べた」は以下のように Linked Data として表現できる。

ex:太郎 ex:食べる
 [ex:対象 “ラーメン”]
 [ex:日付 “2015-11-12”] . (11)

「食べる」の意味はどのように規定されるだろうか。オブジェクト指向語彙概念構造[9]を用いると、「太郎は2015年11月12日にラーメンを食べた」は「太郎オブジェクトは2015年11月12日にラーメンをパラメータとして食べるオペレーションを実行した」と表現される。これより「食べる」の意味を抽出すると「人間クラスのオブジェクトが食べるオペレーションを実行する」となる。ただし、「食べる」の意味の規定としてこれが必要十分であるかどうかについては今後詳細な検討が必要であろう。

3.5 語彙制定規則のまとめ

述語の意味をドメインモデルに従って規定することの利点は、ドメイン内の動的な動きを静的なパラメータの変化として表現できることであり、これは Linked Data を用いた処理の高度化に寄与できる。

例えば、「太郎は2015年4月1日に夕日新聞に就

職した」は「2015年4月1日に太郎オブジェクトの雇用主は夕日新聞オブジェクトになった」と解釈する。この場合、太郎の職歴が全て Linked Data として表わされていて、かつ、太郎は2015年4月1日の後には職を変えていないとすると、太郎の雇用主は2015年4月1日以降の任意の時点において夕日新聞であることが上記の「就職する」の意味規定より自動的に推論できるようになる。

今までに示した述語の意味をドメインモデルに従って規定するための規則を2.3節に既出のものも含めてまとめると以下ようになる。

[語彙制定規則1] トリプルに現れる主語あるいは述語は、(リテラルを除いて)ドメインモデルのクラスあるいはインスタンスに対応すべきである。

[語彙制定規則2] 静的な状態を表すトリプルの述語は、(既存の規格で規定されているものを除いて)ドメインモデルに現れるアトリビュートあるいはアソシエーションに対応すべきである。

[語彙制定規則3] 動的な事象を表すトリプルの述語は、ドメインモデルに基づいて以下のいずれかに対応すべきである。

- ・オブジェクト(クラスのインスタンス)あるいはリンク(アソシエーションのインスタンス)の生成あるいは消滅
- ・アトリビュートの値あるいはリンク先オブジェクトの変化
- ・オペレーションの実行

4 VDM++による述語の意味の規定

4.1 述語の意味の形式的な規定

2節と3節では、述語の意味をドメインモデルに基づきながらも日本語で規定した。しかし、述語の意味を機械的に処理できるようにするためには、形式的な言語を用いて述語の意味を規定すべきである。

述語の意味を規定する言語の候補としてはいくつかのものがある。オブジェクトに対する操作を表す言語である UML Action Language [15][16]も候補の一つであるが、この言語の知名度はそれほど高くなく、この言語が実際に使用されたという例もほとんど報告されていない。

ここでは、VDM++ [17][18]という言語を拡張することによって述語の意味を規定する方法を提案する。VDMはVienna Development Methodの略であり、ソフトウェアの仕様を形式的に記述するための方法としてIBMのウィーン研究所で開発されたものである。VDM++が実際のソフトウェア開発で使用された例は、幾つかのものが報告されているだけであるが、

この言語はソフトウェア開発者の間では注目が集まりつつある。VDM++を選択するもう一つの理由は、この言語はオブジェクト指向型の言語の中では記述能力が極めて高いことである。

4.2 VDM++によるドメインモデルの定義

VDM++を用いて述語の意味を規定するためには、意味規定の基となるドメインモデルを VDM++によって定義しなければならない。図 1 に示したクラスモデルの中の人間クラスを VDM++で定義すると以下ようになる。ここでは、VDM++では日本語が自由に使えるものとする。

```
class 人間
instance variables
  生年月日 : date;
  住所      : address;
  夫        : map 人間 to 人間;
  妻        : map 人間 to 人間;
  雇用主    : map 人間 to 組織
end 人間
```

上において `data` と `address` は、それぞれ年月日と住所を表すためのデータ型であるとする。

4.3 VDM++による述語の意味規定

VDM++には、Linked Data の述語の意味を規定するための機能はない。しかし、以下に示すように、VDM++の関数定義の方法を拡張することによって Linked Data の述語の意味を規定することができる。

まず、Linked Data の述語の意味を規定するために **predicates** というブロックを VDM++に追加する。ブロックとは VDM++における定義の塊であり、定義の種類ごとに設けられる。VDM++には、データ型定義ブロック、変数定義ブロック、定数定義ブロック、関数定義ブロックなどが設けられている。また、**predicates** ブロックにおいては、VDM++で用意されている種々の機能を述語の意味規定用に修正して用いるものとする。**predicates** ブロックの一般的な構成を示すと以下ようになる。

predicates

```
述語 1 (x:主語の型, y:述語の型 (,他の項) )
  主語 x と述語 y (と他の項) の間に成立する条件
述語 2 (x:主語の型, y:述語の型 (,他の項) )
  主語 x と述語 y (と他の項) の間に成立する条件
...
```

predicates ブロックでは、各々の述語に対して、述語の意味を次のように規定する。すなわち、

$$x \text{ 述語 } y \text{ .} \quad (12)$$

あるいは、

$$\begin{array}{l} x \text{ 述語} \\ \quad [\text{目的語の項名 } y] \\ \quad [\text{その他の項名 } z] \text{ .} \end{array} \quad (13)$$

というトリプルがあったときに (x と y は、それぞれドメインとレンジの規定を満たす任意の主語と目的語とし、z は他の任意の項であるとする)、述語の意味は主語 x と述語 y の間に成立する条件として規定される。また、他の項が存在する場合は、その項を意味規定で使用しても良い。

2 節と 3 節で例として使用した述語の意味を VDM++を用いて規定した例を以下に示す。

4.4 VDM++による静的な述語の意味規定

(4)~(7)のトリプルで使われている静的な述語の意味を VDM++の **predicates** ブロックを用いて示したものを以下に示す。ここで、**inv** は、トリプルが成立している時点で不変的に (時間に関係なく) 成立している条件を表すものとする。

predicates

```
生年月日 (x:人間, y:date)
  inv x.生年月日 = y;
住所 (x:人間, y:address)
  inv x.住所 = y;
妻 (x:人間, y:人間)
  inv x.妻 = y;
雇用主 (x:人間, y:組織)
  inv x.雇用主 = y
```

上記の **predicates** ブロックの意味は、2.2 節において日本語で示した述語の意味を VDM++の記法に従って表したものとなっている。

4.5 VDM++による動的な述語の意味規定

3.1 節で論じた述語「生まれる」の意味を **predicates** ブロックによって示したものを以下に示す。

predicates

生まれる (x:人間)

pre not exists p & p = x;

post exists p & p = x

ここで、**pre** は生年月日以前に成立している条件を表わし、**post** は生年月日以後で成立する条件を表すものとするが、ここでは生年月日とこれらの条件との関係は陽には表わされていない。**pre** がどの時点以前に成立する条件であり、**post** がどの時点以後に成立する条件であるかを陽に表すための記法の導入は、今後の課題である。

次に、3.2 節で論じた述語「就職する」の意味を **predicates** ブロックによって示したものを以下に示す。

predicates

就職する (x:人間, y:組織, z:date)

post x.雇用主 = y

ここでは、**post** は z で表される時点以降で成立する条件を表すものとする。

3.3 節で論じた述語「引っ越す」の意味を **predicates** ブロックによって示したものを以下に示す。

predicates

引っ越す (x:人間, y:住所, z:date)

pre x.住所 <> y

post x.住所 = y

ここでは、**pre** は z で表される時点より前に成立している条件、**post** は z で表される時点より後で成立する条件を表すものとする。

最後に、3.4 節で論じた述語「食べる」の意味の形式的な規定であるが、これは「食べている」のか「食べ終えた」のか「習慣として食べる」のか等のアスペクト (相) の情報がなければ、厳密な意味規定とはならない。また、既存の VDM++ の構文でアスペクトを完全に規定することは難しい。「食べる」等の変化を伴わない動作を表す述語の意味を VDM++ で厳密に表現する方法を考案することは、今後の課題である。

5 他の研究との関係

ここで、本研究と関連のある他の研究と本研究との間の関係について述べる。

筆者は、様々な種類の情報を Linked Data (あるい

は Linked Information) として表現するための方法の提案を行った[19]。この方法では、自然言語の文法を抽象化した人工言語を用いて情報を表現し、それを Linked Data に変換する。この方法を用いれば、自然言語で表現できる情報は、原則的には全て Linked Data に変換することができる。さらに、[19]の方法において、文の意味を形式的に規定する方法を示し、機械的に意味処理が行えるような Linked Data を作成する方法も提案した[20]。

[20]で提案した方法では、Linked Data の意味は、ドメインモデルに基づいて規定され、共通のドメインモデルを使用すれば、自動的に語彙の共通化もなされる。しかし、この方法で Linked Data を作成するためには、あらゆる情報をドメインモデルに現れる語彙だけを用いて表現する必要があり、必ずしも人間にとって分かりやすい Linked Data とはならない。

本論文の方法は、ドメインモデルに現れない語彙をドメインモデルに基づいて定義することによって、人間にとって分かりやすい Linked Data を作成でき、しかも使われる語彙はすべてドメインモデルに準拠しているため、機械的な意味処理も可能とするものである。

McShane 等[21]は、レキシコン (特定の自然言語用の電子辞書) において、変化を伴う動詞の意味の定義を事前条件と事後条件とを用いて行うことを提案している。本論文では、変化を伴う述語の意味を VDM++ の事前条件と事後条件とを用いて規定する方法を提案したが (4.5 節)、これは McShane 等の方法に類似している。

また、上位オンロトジーである Suggested Upper Merged Ontology (SUMO) [22] においては、過程 (process) を表す名詞の意味定義において、過程の事後に成立する条件を論理的に表している。この方法では特定のモデルに準拠した意味規定は行われていないようである。今後の課題として、意味規定を特定のモデルに準拠させることによって、意味定義の作成が容易になるかどうか、および、機械的な意味処理の可能性が広がるかどうかという二つの観点からの検討を行う必要がある。

6 おわりに

本論文では、オブジェクト指向化された語彙概念構造の手法[9]と VDM++ [17] を拡張した言語とを用いて Linked Data で使用される述語の意味を厳密に規定する方法を提案した。このように規定された語彙を標準語彙として使用すれば、多くの種類の Linked Data を結合して使用できるようになる。また、語彙の形式的な意味規定を参照することによって

Linked Data の意味を機械的に処理することもある程度は可能になる。

ただし、今後に残された課題も多い。本論文では、動的な事象を表す述語を主な検討対象としたが、そもそも Linked Data において動的な事象を表すための技術が十分に確立されていないことが大きな問題である。この論文では、[14]で推奨されている Blank Node を用いて事象を Linked Data として表現する方法を使用した。事象を Linked Data として表すためのより詳細な標準方式を確立することは Linked Data における緊急の課題である。

また、事象を Linked Data として表す場合、4.5 節の最後に論じたようにアスペクト (相) の表現方法も規定する必要がある。さらに広範な情報を Linked Data として表現する場合は、モダリティ (法性、すなわち、～だろう、～すべき等の表現) も表現できる必要がある。筆者は、相や法性も含めて事象を表現する一般的な方法を[20]で提案したが、この方法を使いやすくするためには更なる工夫が必要であると思われる。

本論文で示した検討は、Linked Data を高機能化および厳密化するための試みの初期段階にすぎず、今後もさらに検討を行っていく予定である。

参考文献

- [1] Heath, T., and Bizer, C.: *Linked Data: Evolving the Web into a Global Data Space*, Morgan & Claypool Publishers (2011)
- [2] 濱崎雅弘, 加藤文彦: 日本語 DBpedia における SPARQL クエリログの分析, 人工知能学会研究会資料, SIG-SWO-036-08 (2015)
- [3] Dublin Core Metadata Initiative: DCMI Home, <http://dublincore.org/>
- [4] Brickley, D., Miller, L.: FOAF Vocabulary Specification 0.99, <http://xmlns.com/foaf/spec/>
- [5] Schema.org : Schema.org, <http://schema.org/>
- [6] Kearns, K.: *Semantics*, Second Edition, Palgrave Macmillan (2011)
- [7] Jackendoff, R.: *Semantic Structures*, MIT Press (1990)
- [8] Blaha, M., Rumbaugh, J.: *Object-Oriented Modeling and Design with UML*, Second Edition, Prentice Hall (2005)
- [9] 山田隆弘: 語彙概念構造のオブジェクト指向化について, 言語処理学会第 20 回年次大会, B1-1 (2014)
- [10] W3C: Resource Description Framework (RDF): Concepts and Abstract Syntax, <http://www.w3.org/TR/rdf-concepts/>
- [11] W3C: Turtle - Terse RDF Triple Language, <http://www.w3.org/TeamSubmission/turtle/>
- [12] Booch, G., Rumbaugh, J., Jacobson, I.: *The Unified Modeling Language User Guide*, Second Edition, Addison-Wesley (2005)
- [13] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., Patel-Schneider, P. F.: *The Description Logic Handbook: Theory, Implementation and Applications*, Second Edition, Cambridge University Press (2010)
- [14] W3C: Defining N-ary Relations on the Semantic Web, <http://www.w3.org/TR/swbp-n-aryRelations>
- [15] Mellor, S., Balcer, M.: *Executable UML - A Foundation for Model-Driven Architecture*, Addison-Wesley (2002)
- [16] 山田隆弘: UML Action Language を用いた意味構造の記述について, 2015 年度人工知能学会全国大会, 2F4-OS-01a-4 (2015)
- [17] Fitzgerald, J., et. al.: *Validated Designs for Object-oriented Systems*, Springer (2005)
- [18] 山田隆弘: 大規模システムに対する形式的要求記述の一方法, 第 77 回情報処理学会全国大会, 2B-02 (2015)
- [19] 山田隆弘: 文章を書くように Linked Data あるいは Linked Information を作る方法, 人工知能学会研究会資料, SIG-SWO-A1203-02 (2013)
- [20] 山田隆弘: オブジェクト指向意味構造記述を用いて Linked Information を作る方法, 人工知能学会研究会資料, SIG-SWO-A1303-04 (2014)
- [21] McShane, M., Nirenburg, S., Beale, S.: Ontology, lexicon, and fact repository as leveraged to interpret events of change. In Huang, C-R., Calzolari, N., Gangemi, A., Lenci, A., Oltramari, A., Prevot, L.: *Ontology and the Lexicon*, Cambridge University Press (2010)
- [22] Niles, I., Pease, A.: Toward a Standard Upper Ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems* (2001)