

UML Action Language を用いた意味構造の記述について

On descriptions of semantic structures using the UML Action Language

山田隆弘^{*1}
Takahiro Yamada

^{*1} 宇宙航空研究開発機構宇宙科学研究所

Japan Aerospace Exploration Agency/Institute of Space and Astronautical Science

In an earlier paper, the author proposed a method of describing the structure of semantic structures based on the object-oriented modeling technique, and showed how the semantic structures of language expressions can be described with this method. That paper did not present a concrete notation for describing semantic structures, but a standard notation or language is necessary to enable automatic processing of semantic structures. This paper proposes a standard method of describing the semantic structures of language expressions based on a language called the UML Action Language, which was designed to describe operations for software objects.

1. はじめに

言語表現の意味を厳密に記述するための研究としては、語彙の意味を記述するための語彙概念構造(Lexical Conceptual Structure) [Jackendoff 1990] [Kearns 2011]という方法が知られている。この方式の特徴は、語彙の意味を基本的な意味要素に分解して記述することであり、語彙(特に動詞)の意味を精密に記述する方法として注目を集めている[竹内 2007]。

筆者は、語彙概念構造をさらに精密に記述するために、オブジェクト指向モデリング[Coad 1997] [Blaha 2005]の方法に基づいて言語表現の意味構造を記述する方法を提案した [山田 2014]。この方法では、世の中の事物や事象を言語表現とは独立にオブジェクトとしてモデル化し、そのモデルの要素を参照して言語表現の意味を記述する。この方式を用いれば、言語表現とは独立に規定されるオブジェクトを拠り所とすることによって意味構造の精密な記述が可能になるものと思われる。

ところで、先の論文[山田 2014]においては、オブジェクト指向に基づいて意味構造を記述する方法を示したが、意味構造を記述するための具体的な記法は示さなかった。しかし、言語表現の意味構造を機械的に処理できるようにするためには、意味構造を記述するための標準的な記法(あるいは言語)が必要になる。本論文は、オブジェクトに対する操作を記述するための言語であるUML Action Language [Mellor 2002] [OMG 2013]を用いて意味構造を記述するための標準記法の提案を行う。

本論文では、まずオブジェクト指向モデリングの手法を簡単に紹介し、先の論文[山田 2014]で提案したオブジェクト指向に基づく意味構造の記述方法を説明する。次に、UML Action Language を意味構造の記述に適用する方法について論じ、UML Action Language を用いた意味構造の記述例を示す。最後に、UML Action Language を用いる方法の効用について議論を行う。

2. オブジェクト指向モデリング

2.1 オブジェクト指向モデリングの原理

ここで提案する方法では言語表現の意味構造をオブジェクト指向モデリング[Coad 1997] [Blaha 2005]の手法に基づいて規

定する。オブジェクト指向モデリングに関する用語は、オブジェクト指向モデリングの標準言語である Unified Modeling Language (UML) [Rumbaugh 2004]のものを用いるものとする。

ここでは、言語表現によって意味される世界に存在する物を抽象化したものをオブジェクトと呼び、意味される世界は一群のオブジェクトにより構成されるものとする。

各々のオブジェクトは、一つあるいは複数のアトリビュートを有する。各々のアトリビュートは値を有しており、アトリビュートの値は、そのオブジェクトの状態や性質を表す。各々のオブジェクトは、一つあるいは複数のオペレーションを有してもよい。各々のオペレーションは、そのオブジェクトが実行する行為を表す。オペレーションを起動するときには、一つあるいは複数のパラメータを指定してもよい。二つ(あついは三つ以上)のオブジェクトは、一つあるいは複数の関係(リンクと呼ばれる)で結びつけてもよい。ここでは、リンクの両端には名前を付けるものとする。リンクがアトリビュートやオペレーションを持ってもよい。

同じアトリビュート、オペレーション、リンクを共有するオブジェクトの集合をクラスと呼ぶ。あるクラスのオブジェクトと別の(あるいは同じ)クラスのオブジェクトを結びつける同じ名前を持つリンクの集合をアソシエーションと呼ぶ。

オブジェクト世界は、クラス定義とオブジェクト定義の2段階に分けて定義される。クラス定義は、クラス、アトリビュート、オペレーション、アソシエーションを定義したものであり、これらの定義はオブジェクト定義で使用される。オブジェクト定義は、オブジェクト、アトリビュート値、リンクを定義する。

2.2 オブジェクト指向モデリングの例

簡単なオブジェクト指向モデリングの例を UML の記法に従って図1に示す。

図1の上の部分には、クラス、アトリビュート、オペレーション、アソシエーションを定義したものである。このモデルには、人間と組織という二つのクラスが存在する。生年月日と住所は、人間クラスのオブジェクトが有するアトリビュートである。見ると食べるは、人間クラスのオブジェクトが有するオペレーションである。また、アソシエーションが二つ定義されていて、これらは、二つの人間クラスのオブジェクトの間には夫-妻というリンクを張ることができること、および、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は従業員-雇用主というリンクを張ることができることを示している。

連絡先: 山田隆弘, 宇宙航空研究開発機構宇宙科学研究所,
相模原市中央区由野台 3-1-1, tyamada@isas.jaxa.jp

図1の下部分は、オブジェクト、アトリビュート値、リンクを定義したものである。この世界には四つのオブジェクトが存在する。太郎と花子というオブジェクトは人間クラスに属し、夕日新聞と駅前大学というオブジェクトは組織クラスに属す。これらのオブジェクトのアトリビュート値も示されている。これらのオブジェクトのオペレーションは、クラスで規定されているものと同じであるので、オブジェクト定義においては省略されている。また、太郎と花子の関係がリンクとして示されている。太郎と花子の雇用主もそれぞれリンクを用いて示されている。

なお、この図の中の R1 と R2 は UML Action Language で使用されるアソシエーションとリンクの番号である。

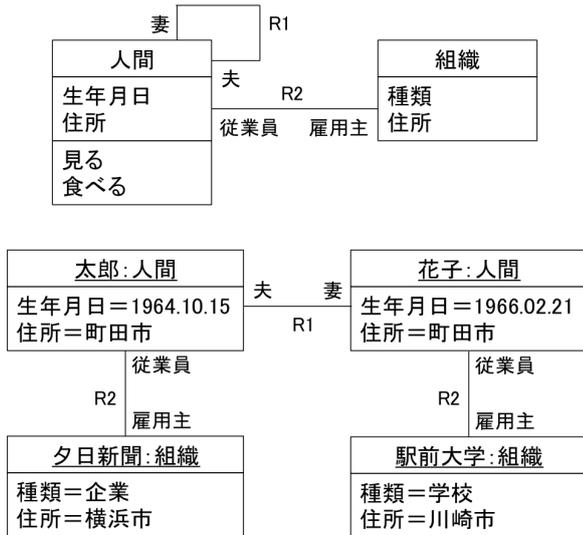


図1 簡単なオブジェクト指向モデリングの例

3. オブジェクト指向意味構造記述

本節では、前節で定義されたオブジェクト指向モデリングに基づいて言語表現の意味構造を記述する方法を示す。

ここでは、自然言語の一つの文が表現する意味内容を文情報として定義する。文情報は、状況と複合体の二つに分類される。状況は単文(一つの述語を含む文)に、複合体は複文(複数の述語を含む文)にそれぞれ対応する。複合体は、複数の状況を何らかの規則で結合したものである。

状況(単文)の意味構造は、以下の三つに分けて規定される。

- A) 基本構造
- B) 述語構造
- C) 付加構造

3.1 状況(単文)の基本構造

基本構造は状況に関係しているオブジェクトに関する情報を表すものである。状況の基本構造は、状態と事象とに分類される。状態は「オブジェクト世界がどうなっているか」を表すものであり、前節で示したオブジェクト定義の内容そのものである。事象は「オブジェクト世界で何が起きるか」を表すものである。

状態の基本構造は、以下の三つのいずれかとして記述される。

- 1) オブジェクトの存在
- 2) オブジェクトのアトリビュート値
- 3) オブジェクト間のリンクの存在

事象の基本構造は、以下の四つのいずれかとして記述される。

- 4) オブジェクトの生成あるいは消滅
- 5) オブジェクトのアトリビュート値の変更
- 6) オブジェクト間のリンクの生成、消滅、変更
- 7) オブジェクトのオペレーションの実行

図1に示したオブジェクトモデルの例を使っていくつかの言語表現の基本構造を記述した例を表1に示す。ここでは、「」内の日本語表現に対応する意味構造記述(ただし基本構造のみ)を<>内に示す。

表1 状況の基本構造の記述例

分類	「日本語表現」→ ＜基本構造記述＞
オブジェクトの存在	「太郎は人間である」→ ＜太郎は人間クラスのオブジェクトである＞
アトリビュート値	「太郎は町田市に住んでいる」→ ＜太郎オブジェクトの住所アトリビュートの値は町田市である＞
リンクの存在	「太郎は夕日新聞に勤務している」→ ＜太郎オブジェクトの雇用主は夕日新聞オブジェクトである＞
オブジェクトの生成・消滅	「花子が生まれる」→ ＜人間クラスの花子オブジェクトが生成される＞
アトリビュート値の変更	「太郎が鎌倉市に引っ越す」→ ＜太郎オブジェクトの住所アトリビュートの値が鎌倉市に変更される＞
リンクの生成・消滅・変更	「太郎が隔日新聞に就職する」→ ＜太郎オブジェクトの雇用主が隔日新聞オブジェクトになる＞
オペレーションの実行	「太郎がラーメンを食べる」→ ＜太郎オブジェクトが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する＞

3.2 状況(単文)の述語構造と付加構造

述語構造は、状況を表す述語に関する補足情報を表すものであり、以下のものを指定する。

- ア) 時制(過去、現在、未来)
- イ) 相(進行中、完了した、習慣的に行う)
- ウ) 法性(～である、～だろう、～すべき等)
- エ) 外部行為者(事象を引き起こす外部オブジェクト)

付加構造は状況全体に対する補足情報を表すものであり、以下のようなものを必要に応じて指定する。

- a) 時点(いつ、いつから、いつまで)
- b) 場所(どこで、どこへ、どこから、どこを經由して)
- c) 程度(どのくらい)
- d) 方法(何を使って)
- e) 理由(何だから)
- f) 目的(何のために)
- g) 条件(何の場合は)
- h) 交換条件(何を得て)

3.3 複合体(複文)の意味構造

代表的な複合体の意味構造の記述方式をその意味と例とともに表2に示す。この表に現れる状況1、状況2等は、3.1 節に示した状況のどれかに対応する。この表に示されているように、複合体の意味構造は、複合体に含まれる状況間の意味関係に基づいて規定される。

表2 代表的な複合体の意味構造

意味構造記述	複合体の意味（上段）と例（下段）
<状況1 FROM 状況2>	状況2は状況1の理由 「太郎は、勉強した(状況2)ので試験に合格した(状況1)」
<状況1 FOR 状況2>	状況2は状況1の目的 「太郎は花子を喜ばす(状況2)ために花を買った(状況1)」
<状況1 IF 状況2>	状況2が成立すれば状況1が成立する 「明日晴れて(状況2)いれば、太郎は山に登る(状況1)」

4. UML Action Language とは？

UML Action Language とは、UML [Rumbaugh 2004]を用いて記述したソフトウェアの構造や振る舞いをプログラムに変換するための道具として考案されたものであり[Mellor 2002]、UMLで図式的に表現されたソフトウェアの構造や振る舞いを厳密に記述するための言語である。UML Action Language はオブジェクトに対する操作(すなわち、オブジェクトの作成、リンクの作成、アトリビュートへの値の代入など)の記法を定めたものであり、この点でオブジェクト指向に基づいた意味構造を表す記法として最適である。

UML Action Language が JAVA や C++等の個々のプログラム言語と異なっているのは、前者は UML で規定されている要素(オブジェクトやリンク)に対する操作を UML の概念のみに従って(すなわち、個々のプログラム言語の文法には依存せずに)記述できることである。従って、UML Action Language は個々のプログラム言語よりも抽象度が高く、UML Action Language の記述を個々のプログラム言語のプログラムに変換することが可能である。

UML Action Language としては過去にはいくつかのものが提案されたが、現在では Alf (Action Language for Foundational UML)という言語が OMG (Object Management Group, UML 等の規格を制定している業界団体)によって制定されている[OMG 2013]。以下では、UML Action Language として Alf を使用することにする。

ただし、第5節で述べるように、Alf だけで意味構造を完全に記述することは不可能なので、Alf で不足している部分は適当な規則を定義して補うものとする。また、Alf の構文の中で意味構造記述に不要な部分は、ここでは省略するものとする。

5. UML Action Language の意味構造記述への適用

5.1 状況(単文)の基本構造の記述

まずは、状況(単文)の意味構造の基本構造を UML Action Language で記述する方法を示す。

3.1 節で述べたように、状況の基本構造は、状態と事象とに分類される。状態は「オブジェクト世界がどうなっているか」を表すものであり、事象は「オブジェクト世界で何が起きるか」を表すものである。UML Action Language における記述の大部分は「オブジェクト世界で何が起きるか」を記述するためのものであり、UML Action Language で事象の基本構造は簡単に記述できる。

ところが、状態の記述は簡単ではない。なぜならば、UML Action Language では、オブジェクトは作られたり操作されたりするものであり、オブジェクトの静的な構造を UML Action

Language で表すことはできないからである。ただし、オブジェクトの集合であるクラスの静的な構造は UML Action Language で記述することができるので、状態の基本構造は、クラスの要素に対する宣言の形式を流用して記述することにする。

表1に現れる状況の基本構造を Alf を用いて記述した例を表3に示す。この表から明らかのように、状況の基本構造は Alf によって極めて簡潔に表されることが分かる。

表3 状況の基本構造の記述例

日本語表現	Alf での記述
太郎は人間である	object 太郎: 人間
太郎は町田市に住んでいる	太郎.住所: 町田市
太郎は夕日新聞に勤務している	太郎.雇用主: 夕日新聞
花子が生まれる	花子 = new 人間 ()
太郎が鎌倉市に引っ越す	太郎.住所 = 鎌倉市
太郎が隔日新聞に就職する	R2.createLink {従業員 => 太郎 雇用主 => 隔日新聞}
太郎がラーメンを食べる	太郎.食べる(ラーメン)

5.2 状況(単文)の述語構造と付加構造の記述

3.2 節で述べたように、状況(単文)の意味構造を規定するためには、基本構造の他に述語構造と付加構造も記述しなければならない。しかし、これらを表すための記法は UML Action Language には用意されていない。そのために、何らかの方法を考案しなければならないが、これには二つのアプローチが考えられる。

一つ目のアプローチは、述語構造と付加構造も取り込んだオブジェクトモデルを規定し、それを UML Action Language に翻訳するという方法である。例えば、「太郎は 2012 年 4 月 1 日に夕日新聞に就職した」という文を図2に示すオブジェクトモデルで表現する。このようなオブジェクトモデルを UML Action Language に翻訳することは可能であるが、この方法を用いると各々の基本構造ごとに述語構造と付加構造を記述するためのモデルを作る必要があり、モデルが煩雑になる。



図2 オブジェクトモデルとして付加構造を記述する例

二つ目のアプローチは、UML Action Language で記述された基本構造全体に述語構造と付加構造を「属性:値」の対として必要に応じて付加する方法である。この方法を用いれば、どのような基本構造に対しても統一的方法で述語構造と付加構造を記述することができるので、ここではこのアプローチを採用することにする。

例えば、「太郎は 2012 年 4 月 1 日に夕日新聞に就職した」という文の意味構造は、以下のように記述される。ところで、以下の記述は、「夕日新聞は 2012 年 4 月 1 日に太郎を雇用した」という文の意味構造記述にもなっている。

```
{R2.createLink
 {従業員 => 太郎
 雇用主 => 夕日新聞}
 {tense: past
  time: 2019.04.04}}
```

また、「太郎は明日サッポロ軒でラーメンを食べなければならない」の意味構造は、以下のように記述される。

```
{太郎.食べる(ラーメン)
 {modality: necessary
  tense: present
  time: CurrentDate+1
  place: サッポロ軒}}
```

5.3 複合体(複文)の意味構造の記述

複合体(複文)の意味構造を表すための記法も UML Action Language には用意されていない。そのために、[山田 2014] で示した複合体の意味構造記述の方法をそのまま使用することにする。この方式は、Jackendoff が提案した節修飾(Clause Modification)の意味構造記述方式[Jackendoff 1990]を拡張したものである。

例えば、「太郎は、勉強したので試験に合格した」という複合体の意味構造は、以下のように記述される。

```
{太郎.合格する(試験)
 {tense: past}}
FROM
{太郎.勉強する
 {tense: past}}
```

また、「太郎は花子を喜ばすために花を買った」という複合体の意味構造は、以下のように記述される。

```
{太郎.買う(花)
 {tense: past}}
FOR
{花子.気分 = 嬉しい}}
```

6. UML Action Language の意味構造記述に対する効用

本論文では言語表現の意味構造を記述するための標準言語として UML Action Language を取り上げたが、原理的には JAVA や C++等のオブジェクト指向プログラム言語を用いて意味構造を記述することも可能である。

本論文で UML Action Language を取り上げた第一の理由は、第4節で述べたように UML Action Language ではオブジェクトモデルに即した記述が可能となるからである。例えば、5.2 節で示したように、「太郎が夕日新聞に就職した」と「夕日新聞が太郎を雇用した」の意味構造は、UML Action Language では同一の表現で記述できる。これらの文は主語の選択が異なっているので語用論的な効果が異なっているが、これらの文の意味構造は主語の選択とは関係なく同一であるはずである。文の意味構造は、オブジェクトモデルを使用することによって精密に記述することができ、UML Action Language はオブジェクトモデルを忠実に記述することができる。

UML Action Language を取り上げた第二の理由は、UML Action Language の方が個々のプログラム言語よりも抽象度が高く、UML Action Language の記述を必要に応じて個々のプログラム言語に変換することが可能であるからである。

さらに、どのような言語を使うべきかという問題よりも基礎的な問題として、意味構造記述に適したオブジェクトモデルをどのように作るべきかという問題も存在する。世の中の構造を適切に表したオブジェクトモデルが存在すれば、それを利用することによって意味構造記述も簡単に行えるようになる。これは、ドメイン毎の標準オントロジーを作成することにも等しい。これは、非常に大きな課題であるが、オブジェクト指向モデリングやシステムモデリングの分野において、オブジェクトモデルをいかに体系的に作成するかという研究が進んでいるので、それらの研究成果を参考にして意味構造記述に利用できるオブジェクトモデルを作成することが可能であると思われる。

7. おわりに

言語表現の意味構造を機械的に処理できるようにするためには意味構造を記述する標準記法が必要となる。本論文では、意味構造を記述する標準記法として UML Action Language を用いる方法の提案を行った。ただし、UML Action Language だけで言語表現の意味構造完全に記述することは不可能であり、UML Action Language を意味構造記述用に拡張するための方法についての提案も行った。

今後は、ここで示した意味構造の記述方式を意味の処理(推論など)に利用するための研究とともにオブジェクトモデルをいかに体系的に作成するかという研究も行いたいと考えている。

参考文献

- [竹内 2007] 竹内孔一, 乾健太郎, 藤田篤, 竹内奈央: 語彙概念構造に基づく事態上位オントロジーの構築, 言語処理学会第 13 回年次大会, pp. 859-862 (2007).
- [山田 2014] 山田隆弘: オブジェクト指向に基づく意味構造の記述について, 2014 年度人工知能学会全国大会 (2014).
- [Blaha 2005] Blaha, M., Rumbaugh, J.: *Object-Oriented Modeling and Design with UML*, Second Edition, Prentice Hall, 2005.
- [Coad 1997] Coad, P., North, D., Mayfield, M.: *Object Models-Strategies, Patterns, and Applications*, Second Edition, Yourdon Press (1997).
- [Jackendoff 1990] Jackendoff, R.: *Semantic Structures*, MIT Press (1990).
- [Kearns 2011] Kearns, K.: *Semantics*, Second Edition, Palgrave Macmillan (2011).
- [Mellor 2002] Mellor, S., Balcer, M.: *Executable UML - A Foundation for Model-Driven Architecture*, Addison-Wesley (2002).
- [OMG 2013] Object Management Group (OMG): *Action Language for Foundational UML (Alf)*, Version 1.0.1 (2013).
- [Rumbaugh 2004] Rumbaugh, J., Jacobson, I., Booch, G.: *The Unified Modeling Language Reference Manual*, Second Edition, Addison-Wesley (2004).