

オブジェクト指向記述日本語 OODJ とその記述環境

畠山 正行[†] 加藤木 和夫^{††} 石井 義之^{†††}

多様な科学/工学分野のドメインユーザの低いプログラム開発力を向上させるため、ユーザの母国語である自然日本語をベースとしてオブジェクト指向の構造化記述ができるオブジェクト指向記述日本語 OODJ を分析段階の記述言語として考案・開発した。OODJ は自然日本語に対し強い構造化記述性を持つオブジェクト指向の枠組みを記述モデルと構文規則の形で導入・設計された。オブジェクト指向記述環境もあわせて設計・実装して OODJ に含めた。評価用に記述例を作成するとともに、数人のドメインユーザによる記述実験を行った。その結果、自然言語の持つ強い記述力の上にオブジェクト指向の構造化記述法を確立したことで記述力の高さに対する顕著な向上が認められ、設計の狙いは実現した。そのほか、プログラミング言語フリー、自然言語の範疇であることからくる記述の容易性・高い了解性、日本語一貫記述言語系の一環であることのメリットが認められ、かつ重大な欠点は見出せなかった。他の言語(主として UML)との比較・評価も行われ改良案も提案された。以上から OODJ がドメインユーザにとって十分有用な分析記述言語であることが結論された。

Object-Oriented Description Japanese OODJ and Its Description Environment

MASAYUKI HATAKEYAMA,[†] KAZUO KATOUGI^{††}
and YOSHIYUKI ISHII^{†††}

To improve the system (program) development ability of the domain users in various scientific/engineering fields, we have developed a new description language called the OODJ (Object-Oriented Description Japanese) based on the native mother language (Japanese) in the target analysis processes. We have introduced and designed a new set of the description model and the syntax based on the Object-oriented paradigm as a description framework with the aid of the Object-oriented description support environment. Some typical description examples have been made by some domain users and evaluated. As the results, we have confirmed that the OODJ have realized the Object-orientedly structured descriptions, and that the description abilities of the OODJ become higher than the original Japanese. The description support environment have been confirmed to be rather useful for the domain users. The comparisons with other languages are performed and discussed mainly with the UML. We can finally conclude that the OODJ is one of the most useful and valid description language for the domain users.

1. はじめに

プログラムを自身でよく作り・しばしば改訂しながら使うユーザに、「ソフトウェア専門家」と「ドメインユーザ」⁽¹⁾と呼ばれる代表的な二群のユーザがある。ソフトウェア専門家は、複数のあるいは新規なプログラミング言語(頻出するので以降、PL と略す)を修得して使うことは意識(心理)の面でも知識の質と量の面でも問題なく実際によく使う。

かたやドメインユーザは自身の専門(流体や構造物の解析やシミュレーション, 画像分析, 建築設計等)には深い知識と興味を持つが, 計算機システムは単に結果のデータ(数値)を出す「道具」にすぎず, 必要以上に「道具」に興味を持たない。むしろ最小限のエディタとコンパイラおよび可視化ツール程度を動かせるだけ, 対象ドメイン(専門分野)に対する分析記述やプログラムの「設計」経験はない, というドメインユーザも少なくない。このようなドメインユーザは, 個人あるいは2~3人で数千行程度の「比較的小規模な」プログラムを作ることが多い。

[†] 茨城大学工学部情報工学科
Department of Computer & Information Sciences, Faculty of Engineering, Ibaraki University

^{††} 日立プロセスコンピュータ・エンジニアリング株式会社
Information Technology Center, Hitachi Process Computer Engineering, Inc.

^{†††} 日本電気株式会社
NEC Corporation

ソフトウェアの設計・製作を自分の専門の職業とする人。本論文では, 各専門分野(たとえば科学技術計算分野)で Fortran 等を用いて計算機を利用するエンドユーザの一種。新規のプログラミング言語の習得は強く回避し, 既存プログラムや蓄積ライブラリの現駆動環境(UNIX)上での利用を望む。

しかし、ドメインユーザが作るプログラムは、膨大な種類の分野に多種多様な利用法で使われており、たとえばスーパーコンピュータ利用プログラムのほぼほとんどを占めるといったことでも分かるように、作成プログラムの総量は膨大な量にのぼり、また、彼らの計算結果のもたらす成果は科学技術の発展に大きな影響を及ぼす。

その影響力に比して、彼ら自身のプログラム(ソフトウェア)の開発力は一般に低い。また現状計算環境の継続利用の要求、特に新規 PL 習得を回避したいという希望の強さゆえか、ドメインユーザの要求の大きさにもかかわらず、開発力の低いドメインユーザ向けのプログラム開発環境は今まではほとんどなきに等しい^{1)~3)}。

そこで、ドメインユーザの「利用言語」と「記述力」の点の同時の解決策となるアイデアを見出した。それはドメインユーザを含め全員共通の母国語である自然日本語の上に新規な記述言語(詳細定義は 2.2 節参照)を再構成することである^{1),3)}。それが PL フリーであればドメインユーザの最も重要な要求である「新規 PL 習得の回避」を満たし、分析力やその結果の記述力をそして最終的にはプログラム開発力を向上させることができよう、というものである。

そのような狙いで前論文¹⁾では自然日本語、オブジェクト指向記述日本語 OODJ²⁾、日本語設計記述言語 JSMDL、の複数種の自然日本語の記述言語のシリーズを提案・設計・実装した。しかし分析段階の日本語記述言語である OODJ の構成や機能は不十分であった。

そこで、本論文では OODJ を前論文の OOJ^{1)~4)}と同じ基本方針ながらその持つべき特性の分析を改めて行って設計方針を定め(2章)、オブジェクト指向の記述モデルを構成し(3章)、その記述モデルを表現できる構文規則(4章)の新規な付加、意味規則(5章)と記述支援環境(6章)を加えた記述言語 OODJ を、設計・構築し直した。7章で再設計した OODJ の記述例を示し、8章で評価と考察、9章で結論を述べる。

2. OODJ の設計方針

2.1 オブジェクト指向日本語一貫プログラミング環境とその問題点

図 1 は筆者らの開発したオブジェクト指向日本語一貫プログラミング環境 JOMON^{1),2)}を表す。本プロ

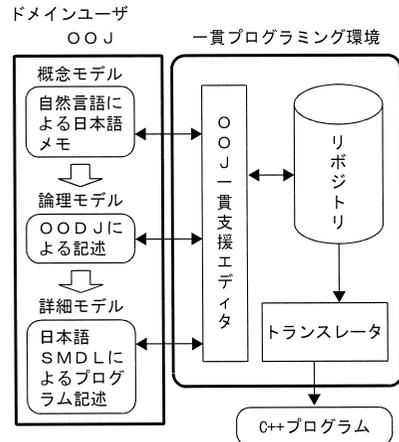


図 1 オブジェクト指向日本語一貫プログラミング環境
Fig. 1 Object-oriented, integrally consistent Japanese programming environment.

グラミング環境の特徴の 1 つは、各モデリング段階の記述が互いに相似であり相互に変換可能な 3 種類の日本語系言語(自然日本語、旧 OOJ, JSMDL)を用いて行われ、複数の段階を経ながら順次相互に変換される方式である点である。このオブジェクト指向日本語一貫プログラミング環境はいくつかの記述例も示してその有効性、実用性がすでに検証された。しかし旧 OOJ に関しては以下の問題点が残された。

- (1) 少数の構文以外はユーザ自身の記述力に頼り、特に対象世界の構造に関する記述法が不足。
- (2) 自然日本語記述からの変換記述法がほとんどない。
- (3) 日本語メモエディタだけでは記述支援環境が不足。

本論文では旧 OOJ を改めて OODJ として設計し直して再構築する。

2.2 自然言語ベースの記述言語

一般に自然言語は他の言語たとえば PL と比べると記述力が強くかつ豊かであるといわれている。このことは確かに事実である一方、その強い記述力をどのような対象世界でもいつの場合でも誰でも十分に発揮・駆使できるとは限らないこともまた事実である。そのような場合の対策としては、自然言語の文法 や用語・記号・記述スタイルに対して新規なものや改良あるいは簡潔化されたものを付加することにより、記述の「枠組み」を明示的に拡張あるいは改良して、その強い記述力を容易に発揮しやすくする方法がしばしば

PL の文法・用語・記述形式を持たないことを意味する。
本論文の OODJ は前論文¹⁾では OOJ と称した言語であるが、都合により以降では OODJ と名称変更する。また以降では、OOJ は JOMON で用いる記述言語のシリーズの総称とする。

本論文では構文規則(4章)と意味規則(5章)の集約された規則体系を文法と呼ぶものと定義し、以下で用いる。

表 1 自然言語ベースの記述言語

Table 1 Description language based on natural language.

<p>特定の記述方式(たとえば,オブジェクト指向記述方式)や特定対象の記述のため,自然言語に対して新規な文法・記述形式・記述スタイル・用法等の「枠組み」を導入した言語を記述言語と呼ぶ.用語・語彙は特別の理由がない限り原語のままであり,あくまでも自然言語として理解可能な言語であり,PLではない.基になった自然言語とは相似で順逆両方向の等価な変換が可能であることが条件.</p>
--

採用される.そのような例としては「表」形式,本や論文の章・節・小節・段落構成,短歌・俳句・連歌,等があげられよう.本論文ではこの方法による効果を,「高い記述力,あるいは,記述性の高さ」の実現と呼ぶ.付録の用語定義「記述力」も参照.

このような言語拡張方針は実は,自然言語をベースとした記述言語の構成方針にほぼ対応している.そこで改めて表1の定義のような記述言語を構成し,より高い記述力を実現する.

ただしこの新規な拡張あるいは改良された枠組みは,それに対する共通認識を持つ人に対してのみ,より高い記述力を与える.つまりこの枠組みの共通認識の範囲が言語の汎用性・特殊性を決定する.特定ドメインや特定ユーザにのみ理解される枠組み(構文,意味規則)を与えることは記述性の高さを獲得すると同時に,理解できる人を少なくし汎用性も失わせるというトレードオフにつねに留意すべきである.

2.3 オブジェクト指向記述日本語の設計方針

対象世界を自然日本語で記述(描写,話)した結果¹⁾を見ると,通常の記述では“振舞い”に関する記述が多く,“構造”(付録:用語定義参照)に関する記述が「少ない」という特徴があげられる.「少ない」というのは,最終的にプログラムに変換するには自然日本語記述の情報・データだけでは不足である,という意味である.もちろん自然言語の記述力が弱くて不足なのではなく,対象世界に関する共通概念・共通認識のある部分が省略され,記述されていないからである.

したがって,プログラム記述に変換可能な情報量を持った記述にするには,構造記述を強化する方法が必須である.そのためには,強い構造記述性(付録:用語定義参照)の枠組み形式と記述法を持つオブジェクト指向パラダイム^{5),6)}を自然言語記述の仕様の一部代替として明示的に導入するのがよい.したがってドメインユーザはオブジェクト指向を「概念として理解」していることが要求される.オブジェクト指向PLは

知らないか使えなくてもかまわない.これに基づき具体的な新規言語構成の設計方針を考案した.

- (1) 構造記述部分には自然言語に由来する制約はないので,オブジェクト指向に沿って十分な構造記述力を持つ文法・記述形式・枠組みを新規に考案・構成した.ただし記述スタイル,キーワード,説明文等は可能な限り自然言語類似として特殊な形式や語彙は可能な限り避ける.
- (2) 振舞い記述部分にはオブジェクト指向に由来する制約はないので,論理的な一意性・構造性を持たせた.自然日本語並みの読み・書き・解釈・理解の容易性をそのまま維持する.
- (3) 語彙・用語・記号は原語である自然言語のものを極力用いる.
- (4) PLフリーの要求(PL風文法の排除)を実現する.
- (5) 記述すべき要素の提示・ガイド機能,記述の形式的なチェックを行う記述環境を構築する.

以上,全体の記述の枠組みはオブジェクト指向の構造化された記述形式を導入する.この特徴的な形式を本論文ではオブジェクト指向構造化記述形式と呼び,4章でオブジェクト指向記述日本語文法の一部である構造化構文規則の中で具体化する.

オブジェクト指向パラダイムにおいて最も特徴的な記述形式は構造記述部分であり,振舞い記述部分ではない.そこで本論文においては構造記述部分だけに絞ってオブジェクト指向の特徴を持った構文規則を,新規なオブジェクト指向記述文法として導入・設計した.

2.4 定式化記述とOODJ記述

分析記述については現状でも多くのドメインユーザはたとえば論文で「定式化」「モデル化」「物理モデル」「Target Description」「Formulation」「解析手法」「基礎方程式」等という節を設け,各ドメインの用語と特殊な記述規約を持つ論理的な自然日本語や記号・数式を用いて分析結果を記述していることが多い.これらを「定式化記述」と総称する.

この定式化記述を2.2節の定義に照らせば,原初的なながら記述言語に沿った記述に近い場合もかなり存在する.しかし,対象世界の構造部分の記述が十分ではない場合が多く(2.3節参照),意識的に・形式的に記述言語として体系化した構文や意味規則を定めている例もほとんどなく,不十分な記述が多い.

OODJはこのようなドメインユーザの現状の定式化記述の記述力を飛躍的に改善するべく,オブジェクト指向構造化記述形式を明示的に具体化した構文規則の枠組みその他を導入したものである.

表2 記述モデル構成(その1:シング名と属性)

Table 2 Constitution of description model (part1: thing name and its attributes).

1. シング名
2. 属性
2.1 単独属性 記述構成:
(属性名, 属性値, 作用振舞い名, 属性種, 制約)
2.2 相互関係属性
2.2.1 相互関連属性
記述構成:
1) 汎化・特化:[上位(汎化)シング, 下位(特化)シングリスト]
2) 集約:(上位シング, 下位シングリスト)
3) 集合:(所属シング名リスト)
4) 一般関連:
(相互関連属性名, 相互関連相手シング名リスト, 制約)
2.2.2 相互作用属性
記述構成:
(相互作用相手シング名, 相互作用抽象名, 相互関係シング名リスト, 制約)

表3 記述モデルの構成(その2:振舞い)

Table 3 Constitution of description model (part2: behavior).

3. 振舞い
3.1 内部振舞い
記述構成:
[内部振舞い名, 振舞い記述 (OODJ 単文, 複数可), 制約]の群
3.2 相互作用振舞い
記述構成:
[相互作用抽象名, 振舞い記述(OODJ 単文, 複数可), 相互関係相手シング名, 制約]の群

3. OODJ のオブジェクト 指向記述モデル

オブジェクト指向に基づくシングモデルの構成要素の一般形を表2, 表3のように提案する。記述モデルは「対象世界の何を記述すべきか」の記述要素を指定する。すなわち, 対象世界から抽出・モデル化・記述すべき「記述要素, およびその相互関係(付録:用語定義参照)」をすべて列挙するためのモデルである。構成要素の記述は括弧で括ってセットにしたスロット記述形式, データ型でいえばレコード型を採用した。

分析記述段階のうち, 上流の自然言語で記述する段階でのモデル化単位を“もの”と呼び, 下流で記述言語で記述するモデル化単位を“シング”と呼ぶことにする。

表2, 3に指定された必要なすべての要素を抽出・記述した結果を, 本論文では「完備な記述」(付録:用語定義参照)という用語で特に呼ぶ。

3.1 シングと属性

表2において, シングとはモデル化単位名称である。属性(2.)のうち, 他のシングと直接に相互関係のない単独属性(2.1)の記述では, 何らかの作用・振舞いから参照あるいは更新されなければ意味がないので, アクセスすべき振舞い記述と必ずセットで構成される。

相互関連の記述(2.2.1)や相互作用の記述(2.2.2)を, 本記述モデルでは属性として扱うこととした。相互関連属性を表現するにはUML^{7)~9)}等のダイアグラム記述言語を補助的に用いてもよい(8.6.4項参照)。特に, 初期の相互関係属性の抽出には役に立つ。

相互作用属性(2.2.2)とは, シング間の相互作用と相互関連(構造)とのかかわりを記述するものである。相互作用の起動・駆動順序やその相互関連の実体は表3においてその抽象名でラベル付けされた(3.2)の振舞い記述文に記述され, (2.2.2)内の相互作用属性記述内にその相互作用抽象名を再定義して記述する。相互作用属性の記述は, 振舞いを起動・駆動させる要素と手順の集約された記述であり, 振舞い機構の記述に相当する。

3.2 振舞いの記述

表3において内部振舞い(3.1)とは単独振舞いともし, 単独シング内のみで起きる振舞いで, シング外との相互作用は起こさないものを指す。

相互作用にともなう振舞い(3.2)は, 複数の内部振舞い記述(3.1)と相互関係属性記述(2.2)も用いて記述される。複雑な振舞いの場合には, さらにそれらを構造化(たとえば抽象階層構造化)した構成にし, 粒度の細かい複数の振舞い記述文の構造化された順序列とその制御手順の記述で詳細に記述する。そのような(3.2)の振舞い記述を相互作用起動手順記述と呼ぶ。ただし階層構造の記述構成は形式的には相似な構成になるので表3では省略した。

本記述モデルはオブジェクト指向構造化記述形式を自然言語に導入することを狙いにして構築した。そのため自然言語記述で詳しく書かれる振舞いや相互作用については, 本論文の主題・狙いを絞る必要性, 現状の自然言語での記述力, および紙幅を考慮して表3では特別な記述モデルを構成しなかった。したがって, たとえばOMTやUML記法でいう動的モデルや機能モデルに対応するものは本論文においては割愛した。

4. OODJ 構造化構文規則の設計

OODJ 構文規則は2章の設計方針と3章の記述モデルを基に新たに設計した。OODJ 構文規則とは自然言語記述の概念モデル記述に対して, シングごとに

表 4 シング記述構文規則
Table 4 Thing description syntax.

```

1. < OODJ 記述 > ::= { < シング定義 > }
< シング定義 > ::= # シング : " < シング名 >
    < 説明 > [ < 属性 > ] [ < 振舞い > ]
< シング名 > ::= < 自然日本語名詞 >
< 説明 > ::= "% 説明 : " < 日本語単文 >
    { < 日本語単文 > }
< 日本語単文 > ::= < 自然日本語単文 > " . "

```

属性, 相互関係属性, 振舞いの手順記述の定義を拡張 BNF 記法 で記述したものである. 表 4, 表 5, 表 6, 表 7 にそれらを示す.

4.1 シング記述構文規則

表 4 にあるように, < シング定義 > は, < シング名 >, < シング自体の < 説明 >, < 属性記述 >, < 振舞い記述 > から構成される. < 説明 > は自然言語の概念モデル記述をそのまま残した場合と新規に記述を加える場合がある. < 説明 > は個々の記述要素の後方の任意の位置に記述できる. そのため以降では特に必要な以外は構文定義から形式上省略する. < 日本語単文 > は自然日本語単文で " . " で終わる. 構文規則数を減らすため, 自然日本語で記述されることが自明な < ~ 名 >, < 自然日本語単文 > 等は終端記号を用いた構文規則の記述を省略してある.

表 4~ 表 7 から分かるように, OODJ では (狭義の) オブジェクト指向で用いられるクラスとインスタンスの区別をしていない. 抽象モデルであるクラス相当の記述を禁止してはいないが, PL ではない OODJ では特に支障はないので, 簡潔さを保持するためインスタンスに対応する記述構文しか定義しないオブジェクトベースモデルを用いた. クラスとインスタンスの区別等は必要に応じて設計段階の記述言語である JSMDL 以降で定義される¹⁾.

4.2 属性記述構文規則

表 5 の属性記述構文規則において, < 属性種 > とはその属性への作用 (アクセス) の可不可を示す. 固有とは当該シングからのみの作用 (アクセス) を許し, 特定公開は特定のシング群からのみに作用を許し, 公開はすべてのシングからの作用を許し, < 特殊作用制約 > は文字どおり特殊な作用制約を規定する. (2.1) の < 単独属性 > とは属性名とその属性が関連する振舞い, 相互作用時の制約記述から構成される. < 制約 > には属性の定義域, 起動条件等を記述する.

構文記号として, 非終端記号の < >, 終端記号の " ", 定義・書き換えの ::=, 繰返しの { }, 省略可能な [], 選択の |, 等の通常の記号のほかに, シング, 属性, 振舞いを各々示す #, %, \$, ユーザのコメントの " C : " を用いる.

表 5 属性記述構文規則 (その 1 : 単独属性)
Table 5 Attribute description syntax (part 1 : simple attribute).

```

2. < 属性 > ::=
    { < 単独属性 > } { < 相互関係属性 > }
2.1 < 単独属性 > ::=
"% 単独属性 : "
    < 属性名 > [ "( = " < 属性値 > ")" ]
    < 作用振舞い名 > < 属性種 > [ < 制約 > ]
< 属性種 > ::= ( "% 固有 " |
"% 特定公開 ( " < 特定シングリスト > " ) " |
"% 公開 " |
"% 特殊作用制約 " )
< 作用振舞い名 > ::=
"% 作用振舞い名 : " < 内部振舞い名 >
< 制約 > ::= "% 制約 : " < 日本語単文 >
    { < 日本語単文 > }

```

表 6 属性記述構文規則 (その 2 : 相互関係属性)
Table 6 Attribute description syntax (part 2 : mutual relationship attributes).

```

2.2 < 相互関係属性 > ::=
    { < 相互関連属性 > } { < 相互作用属性 > }
2.2.1 < 相互関連属性 > ::= "% 相互関連属性 : "
( < 汎化 > | < 集約 > | < 集合 > | < 一般関連 > )
< 汎化 > ::=
[ "% 汎化 ( 上位 ) シング名 : " < 上位シング名 > ]
[ "% 特化 ( 下位 ) シング名 : "
    { < 下位シング名リスト > } ]
< 集約 > ::= "% 集約 : "
[ "% 上位シング名 : " < 上位シング名 > ]
[ "% 下位シング名 : " { < 下位シング名リスト > } ]
< 集合 > ::= "% 集合 : " { < 構成シング名リスト > }
< 一般関連 > ::=
"% 一般関連 : " < 相互関連属性名 >
"% 関連先 : " { < 相互関連相手シング名リスト > }
2.2.2 < 相互作用属性 > ::= "% 相互作用属性 : "
< 相互作用相手先シング名 >
< 相互作用 ( 抽象 ) 名 >
< 相互関係シング名リスト >
[ < 特別関係記述 > ] [ < 制約 > ]
< 相互作用相手先シング名 > ::=
"% 相互作用相手先シング名 : " < シング名 >
< 相互作用抽象名 > ::=
"% 相互作用名 : " < 相互作用名 >
< 相互関係シング名リスト > ::=
"% 相互関係シング名リスト : " { < シング名 > }
< 特別関係記述 > ::=
"% 特別関係記述 : " { < OODJ 単文 > }

```

本論文では相互関係属性 (2.2) にあるように, 静的な相互関連だけでなく動的な相互作用振舞いについても, これに関係するシングとの間の諸関係を明示的に示すために属性として記述した. これにより振舞い (相互作用) とそれに関係するシング等との相互関連を明示的に記述できるようになった.

表 6 の属性記述構文規則において, (2.2.1) < 相互

関連属性>には<汎化>,<集約>,<集合>,<一般関連>の4種類がある.その記述はいずれも,相互関連名と関連シング名のセットで記述される.<一般関連>とは特定の関連属性以外を指す.

(2.2.2) <相互作用属性>では,すでに3.2節で述べたように,相互作用抽象名と相互作用の相手先シング名との関連を記述することで,構造部分と振舞いと相互関係を記述する.相互作用の記述は相互作用(抽象)名のみをここに記述し,実際の手順およびその詳細記述は(3.2)で行う.特別関係記述とは特別な相互作用やそれともなう振舞いの際に用いる記述で,形式は規定していない.

4.3 振舞い記述構文規則

表7の振舞い記述構文規則において,(3.)<振舞い記述>にはシング間の相互作用のない<内部振舞い>とシング間の相互作用を記述する<相互作用振舞い>記述がある.

(3.1) <内部振舞い>の<振舞い記述>は,その記述法を意味規則(5章)で規定される日本語の単文である<OODJ単文>から成る.<OODJ単文>で記述される<振舞い記述文>の記述内容を規定する構文としては,条件分岐を行うための<選択文>、処理を繰り返す<繰返し文>、条件によって多方向に分岐す

る<多方向分岐文>、振舞い起動の判断基準等に用いる<条件>がある.

(3.2) <相互作用振舞い>はシング間の相互作用振舞いの詳細を属性の参照も含めて記述する.煩雑でもあり似た構成でもあるため構文では省略したが,記述モデルの(3.2)と同様その必要に応じて,たとえば抽象記述と詳細記述の2階層以上の,集約(抽象)階層構造で記述することもある.

5. OODJ変換意味規則, OODJ意味規則

OODJは自然言語記述からの変換や,対象世界から新規に抽出・補充されてOODJ記述に仕立てる関係上,自然言語記述からOODJ記述への変換規則を定めた「OODJ変換意味規則」と,変換後の最終記述形が満たすべき「OODJ意味規則」との2つを定める.

5.1 OODJ変換意味規則

OODJ変換意味規則とは,表8に示したように自然言語記述をOODJに記述し終わるまでの変換規則を定めたものである.これは論理的で一意性を持つ基本文型の「平述文の単文」の接続に変換する過程での変換規則と記述補充規則を示しただけであり,記述を明快にはするが,自然日本語の記述力に対する障害をまったく作りだしていない点が特長である.

5.2 OODJ意味規則

OODJ意味規則は表9に示したように,多くは振

表7 振舞い記述構文規則
Table 7 Behavior description syntax.

```

3. <振舞い> ::=
  { <内部振舞い> | <相互作用> }
3.1 <内部振舞い> ::= " $ 内部振舞い : "
  <内部振舞い名> <振舞い記述>
<振舞い記述> ::=
  " $ 振舞い記述 : " { <振舞い記述文> }
<振舞い記述文> ::=
  ( <OODJ単文> | <選択文> |
  <繰返し文> | <多方向分岐文> )
<OODJ単文> ::=
  <OODJ文法に従う日本語単文> . "
<選択文> ::= " IF " <条件>
  " THEN " { <振舞い記述文> }
  " ELSE " { <振舞い記述文> }
  " ENDIF "
<繰返し文> ::= " LOOP " <条件>
  " DO " { <振舞い記述文> }
  " ENDOOP "
<多方向分岐文> ::= " 多方向分岐 "
  <条件> { " ケース : " { <振舞い
  記述文> } }
<条件> ::= <OODJ単文>
  { (" AND " | " OR " ) <OODJ単文> }
3.2 <相互作用振舞い> ::= " $ 相互作用 : "
  <相互作用(抽象)名> <振舞い記述>
  [ <相互関係相手シング名> ] [ <制約> ]
(非終端記号を用いた階層構造構文規則は省略)

```

表8 OODJ変換意味規則
Table 8 OODJ transform semantics.

- (1) 記述を文単位で抽出し,単文はそのまま,複文は単文の列(接続)に変換する.
- (2) 変換前の複文全体を抽象振舞い(記述)名で置換し,接続化された単文をまとめて抽象振舞い名に指定する,または,変換された接続単文を構造化して捉え,相互作用起動手順記述へと変換して抽象振舞い名に指定してもよい.
- (3) 意味が同じなら各文の順序は必要に応じて入れ替えてよい.
- (4) 論理的な意味の矛盾は最終記述形までに取り除く.記述は論理的で一意的な解釈がなされる単文に変換しなければならない.
- (5) 叙情文は内容を変更・削除・補充し,同等内容の叙事文の列へ変換する.
- (6) 倒置文,強調文は意味を同等に保つ基本文型の列に変換する.
- (7) 否定文は,平述文または分岐選択文に変換するか制約記述にする.
- (8) 疑問文は平述文と分岐選択文の列に変換する.例:これは鉛筆か? > 鉛筆に名前を聞くという相互作用記述文に変換.
- (9) 命令文は相互作用を行う平述文に直す.
- (10) 変換意味規則を満たした単文は,OODJ構文規則の適用へ引き継ぐ.

表 9 OODJ 意味規則
Table 9 OODJ semantics.

- (1) 各語彙の持つ意味自体は特に定義しない限り日本語の原義と同一である。
- (2) 動詞の活用法, 名詞の単複, 時制の表現はそのまま残す。
- (3) 平述文の基本文型は S+V, S+O+V, S+C+V, S+O+C+V, S+O+O+V である。
- (4) 完成した OODJ 文は上記の基本文型を持つ平述文の単文とする。
- (5) その他の基本文型として, 選択文, 繰返し文, 多方向分岐文を持つ。
- (7) 記述は論理的で一意的な解釈がなされる単文でなくてはならない。
- (8) OODJ 記述は上記の単文およびその接続で構成される。
- (9) “高い・速い, 接近する” 等の曖昧あるいは抽象的な数量表現も許す。

舞いを記述した自然日本語単文に対する規則を定める。表 9 に示した事項以外については自然日本語の文法をそのまま用いる。形は新規な意味規則ではあるが、整理された論理的な自然日本語ともいべき内容であり、特別な制約を課してはならず、記述において新規な支障は生じていない。

6. OODJ の記述環境の設計

自然日本語の通常の記述では、実際に対象世界に存在しても当初からは明示的 (Explicit) には記述化がされにくい要素 (特に対象世界の構造部分の記述) や、2.3 節でも述べたように既存の共通概念・共通認識であると考えられて暗黙の (Implicit) 了解事項として半無意識に記述から外れる要素、等もある。一方、ドメインユーザの方も分析段階の記述において各専門の世界の把握・識別能力は高いが、偏ったり理解しにくかったりする用語や記述を使うことも多い。そこで OODJ の一部として、記述すべき要素の提示、入力・訂正・編集、相互関係のチェック・表示、形式的文法チェック、記述結果の一覧表示等の機能を持つ GUI 記述環境をオブジェクト指向に基づいて設計・実装した。

6.1 OODJ 記述手順

まず記述の処理手順・順序を 4 つのステップに区分して再構成・設計・実装した^{3),4)}。

(1) 対象識別・仕分けステップ

図 2 にあるように、自然言語を用いた概念モデル記述に対して、まずシングを識別・抽出し、その属性、振舞い、シング間の相互関係を識別・抽出・記述・改訂をする。未抽出シングを探索・抽出してエディタ画面から新規に定義を行う。

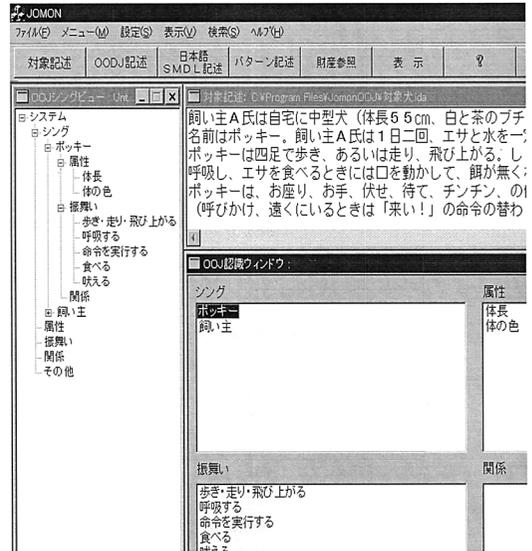


図 2 対象識別・仕分けステップの記述環境画面
Fig. 2 Target identification and classification step.

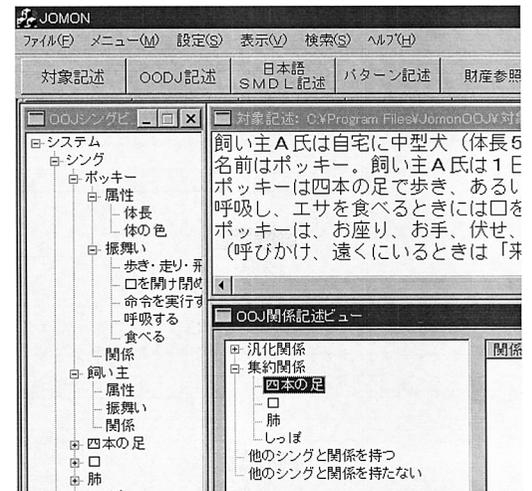


図 3 構造記述 (相互関係記述) ステップ画面
Fig. 3 Structured, mutual relationship description step.

(2) 構造記述・相互関係記述ステップ

識別・仕分け記述されたすべての構造記述要素である属性の詳細記述や相互関係属性の記述の追加・補充・訂正を行う。原則として構造や相互関係に関するすべての要素を揃える。図 3 にあるように集約や汎化の相互関連、相互作用等の相互関係を複数画面に記入しておけば、多少のユーザ自身の修正で後述の図 5 のような最終記述に至る。OODJ および本記述環境は特に自然言語風の記述ながら構造記述の多様さや詳細さとその表示に特徴を持たせてある。

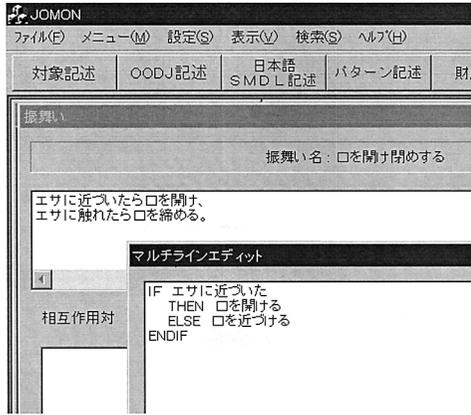


図4 振舞い記述ステップ画面
Fig.4 Behavior description step.

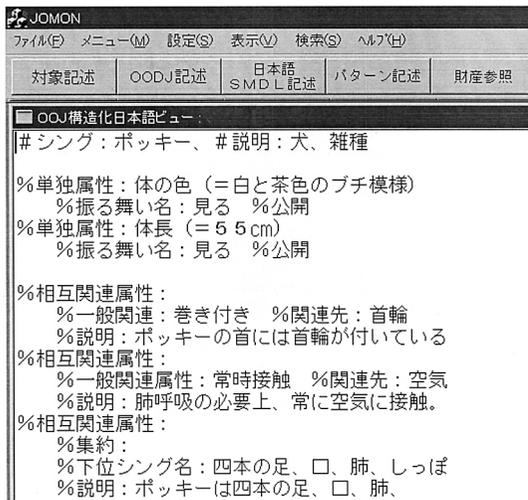


図5 分析モデルの最終記述ステップ画面
Fig.5 Final description step of analysis model.

(3) 振舞い記述ステップ

図4にあるように、シング間の相互関係を基にその振舞いや相互作用の起動・駆動順序の整合性や合理性をチェックする。各振舞い記述が参照する属性との対応とその参照処理も再確認する。振舞いの詳細記述を行う。この段階で単文チェック機構を通して、基本文型の単文であるか否かの判定を行う。

(4) レビューステップ

図5にあるように、JSMDL^{1),2)}移行への最終記述の完成のため、モデル記述全体についての相互関係等を再検討・探索・重複チェックを行う。図5の完成記述を7章の表11~14に示す。

6.2 記述エディタの設計と実装

すべての実作業はドメインユーザ自身の手動記述・

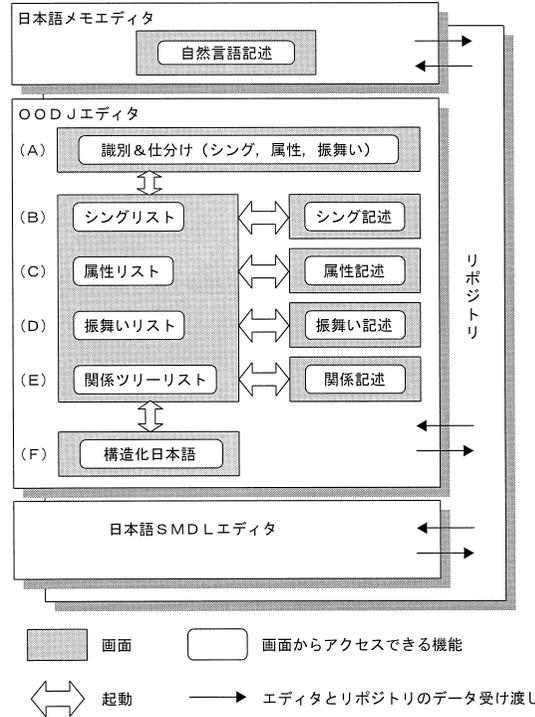


図6 OODJ一貫支援エディタの構成
Fig.6 Constitutions of OODJ integrated support editor.

変換で実現するので、記述エディタではオブジェクト指向記述日本語文法の枠組みを分類・提示して入力を要求・誘導・ガイドする。前節の作業手順の提示およびガイダンス機能を実装し、各手順間での自在な移動と以下の機能とを備えた複数の記述エディタ画面から成る OODJ 一貫支援エディタを設計・実装した^{3),4)}。

- (1) 振り分け・仕分けのためのドラッグ&ドロップ (あるいはコピー&ペースト) 機能。
- (2) シング単位および記述要素の追加・削除・訂正機能。
- (3) 属性や振舞いの関係先候補の表示機能や画面間移動機能。
- (4) 振舞い記述の形式的な文法チェック機能。
- (5) レビュー・確認用の一覧表機能と、相互関係チェック用のツリー表示機能。
- (6) シング、属性、振舞い間の参照・更新の関係や相互関連の表示・チェック機能。

これからも分かるように、相互関係の記述支援とその表示に特徴がある。

実装された記述エディタはシング定義、属性、振舞い、関係属性の記述の作業ごとに個別ウィンドウを設けた。図6に連続的に変化する記述過程を支援するエディタの全体構成、画面の種類と画面がガイドする機

能を示す．図 6 中の (A) は識別&仕分けステップ用の機能と画面(図 2)を示し，(B)～(E)は各要素単位に記述，追加，削除のできる機能と画面(図 3, 4)である．(F)は(A)から(E)までに個別に入力された記述データを OODJ 記述として統合して出力する機能を持たせた画面(図 5)である．

7. OODJ の記述例とその評価

OODJ 記述例としては特定ドメインの記述例を示すべきであるが，厳密性・無矛盾性を追求して記述量が大きくなり，基礎知識や用語の定義が必要になったりするので，本論文では一般的で誰でも分かる飼犬の散歩のシミュレーションを取り上げる．表 10 に自然日本語を用いた概念モデル記述を示し，表 11，表 12，表 13，表 14 にその OODJ への変換および追加記述の結果の例を示す．表 10 から表 11・12・13・14 への分析や変換の途上，多くの文や単語(用語)の追加・変換・変更作業が前述の記述環境を用いて行われた．

8. 評価と考察

8.1 記述実験と記述例の評価

記述実験は流体の数値解析を自身のドメイン(専門分野)とする 3 人の大学院学生，および情報工学専攻の大学院学生数人を対象にして実施した．彼らはオブジェクト指向の一応の知識は持っている．まずオブジェクト指向の概念を 3 章の記述モデルを使って再確認し，7 章の例を説明し，並行して 4 章の構文の説明を行った．説明時間は約 1～2 時間．その後で各自対象世界を選んで記述してもらった．

記述実験の対象世界としては，植物の光合成・成長，熱帯魚の水槽飼育，将棋ゲーム，カードゲーム UNO，株式市場，CD 情報管理システム，自動販売機，ファストフードの製造販売システム，劉備と玄徳の会戦の場面，等であった．流体解析専攻の学生による流れの記述実験も試みられたが，記述量が大規模になり，ドメイン特有の事象に対する表記法の未整備を実験時間内では解消できなかったため，一部の記述にとどまった．彼らの行った記述実験のコメントや意見・感想は本章で必要に応じて引用する．

分析や設計に経験が少ない流体分野の被験ドメインユーザは，分析段階で早々と属性と振舞いの参照・更新のかかり(相互作用)を明示的に記述する等の要求に経験上初めてぶつかったことで，記述実験中に戸惑いの声が生じた．このような記述は本質的かついづれは必要となる記述であること，OODJ では方針として記述可能なすべての要素の記述をあらかじめ分析

表 10 概念モデル記述例

Table 10 Concept model description example.

```

飼い主 A 氏は中型犬を飼っている．それを見ると体長は 55 cm，体色は茶色と白のプチで名前はポッキーである．飼い主 A 氏は 1 日 2 回エサと水を一定量ずつ与える．ポッキーは 4 足で歩き，あるいは走り，飛び上がる．しっぽを振り，肺で呼吸し，エサを食べるときには口を動かして，エサがなくなるまで食べる．
ポッキーは，お座り，お手，伏せ，待て，チンチン，その他「ポッキー！」(呼びかけ，遠くにいるときは「来い！」の命令の代わり)「ほーい」(投げたボールを取ってこい)ができる．ポッキーは首輪を付けている．室内では放し飼いで散歩に出るときにはひもを付ける．

```

表 11 OODJ シング記述例

Table 11 OODJ thing description example.

```

#シング：ポッキー %説明：犬，雑種
%単独属性：体色( = 白と茶色のプチ )
%振舞い名：見る %公開
%単独属性：体長( = 55 cm )
%振舞い名：見る %公開
( 表 12, 13, 14 に続く )

#シング：飼い主
%説明：ポッキーを飼い，命令を出す．
( 以下は省略 )

```

表 12 OODJ 相互関連属性記述例

Table 12 OODJ mutually relational attribute description example.

```

%相互関連属性：
%一般関連：巻き付き %関連先：首輪
%説明：ポッキーの首には首輪が付いている
%相互関連属性：
%一般関連属性：常時接触 %関連先：空気
%説明：肺呼吸の必要上，つねに空気に接触．
%相互関連属性：
%集約：
%下位シング名：4 本の足，口，肺，しっぽ
%説明：ポッキーは 4 本の足，口，肺，しっぽから成る．
( 以下は省略 )

```

段階で求めていること，を説明して納得が得られた．しかし情報工学専攻の大学院生も含め，相互関係属性と相互作用のモデル化と記述には不慣れによる戸惑いが見られた．

前章での概念モデル記述例と OODJ 記述例とを比べると，記述の正確さと完備の程度は OODJ の方が当然優れており，記述環境の提示とガイドの支援を受けることで必要記述要素を「ほぼ記述している(= 記述の完備性が比較的高い)」といえよう．しかも OODJ 記述は自然言語記述と同等の分かりやすさを持つため，

表 13 OODJ 相互作用属性記述例
Table 13 OODJ mutually interactive attribute description example.

%相互作用属性:
%相手シング名: 地面
%相互作用名: 歩き・走り・飛び上がる.
%関連シング名: 4本の足
%説明: ボツキーは4本の足を使って 歩き, 走り, 飛び上がる.
%制約: 3日間エサが貰えないと筋肉は 動かなくなる.
%相互作用属性:
%相手先シング名: 空気
%相互作用名: 呼吸する
%関連シング名: 肺
%制約: 肺呼吸
%相互作用属性:
%相手先シング名: 餌
%相互作用名: エサを食べる
%関連シング名: 口
%説明: エサに近づいたら口を開け, エサに触れたら口を締める. (判断機能やセンシング機能は省略. エサがある限り無限に食べ続ける.)
%制約: なし.
%相互作用属性:
%相手先シング名: 飼い主
%相互作用名: 命令を実行する
%関連シング名: 4本の足 (以下は省略)

表 14 OODJ 相互作用記述例
Table 14 OODJ mutual interaction description example.

\$相互作用: 歩く・走る・飛び上がる.
\$振舞い記述:
多方向分岐(動作の種類)
ケース(歩く): 歩くの振舞い記述
ケース(走る): 走るの振舞い記述
ケース(飛び上がる):.....
\$相互作用: 呼吸する
\$振舞い記述:
肺を膨らませて空気を吸い込む.
酸素と炭酸ガスを交換する.
肺を縮めて空気を排出する.
\$相互作用: エサを食べる
\$振舞い記述:
口を移動させ, エサに接触する.
LOOP { エサがなくなるまで }
DO 口をあけてエサを口に入れる.
口の中にエサが入ったら
口をきつく閉める.
エサをかみ砕いて飲み込む.
ENDLOOP
\$相互作用: 飼い主の命令を実行する. (以下は省略)

その理解に支障はまったく見られなかった.

OODJ 記述例とドメインユーザ自身が独自に作成した定式化記述とを比較すると, OODJ の方が構造

化記述が詳細な割に簡潔・明示的なこと(必要で)記述された要素が多いこと, 相互関係の構造記述が正確で明確なこと, 作成手順が形式的に定義・提示されているので簡潔に短時間で記述できる, 等の特長が多く見出されたことが被験者からコメントされた.

記述例から評価した OODJ の特性は, 記述しやすさ, 理解のしやすさ, 記述要素の完備の程度, 等のすべての面で OODJ は優れた特性を見出せており, かつ大きな欠点は見出せない. また OODJ はドメインユーザがソフトウェア専門家との間での議論や考察を共有できる意志疎通機能の高い言語であることも判明した. この点はソフトウェア専門家からプログラム分析・設計時にアドバイスを受ける場合等に非常に有効である.

その結果, 従来からの「定式化記述(2.4節参照)」を OODJ で記述し直す記述実験に加わったドメインユーザ自身の分析記述力がほぼ全員顕著に向上するという結果を得た. その結果各ドメインユーザは, その心理的な障壁が低くなったこともあり, 各ドメインの詳細かつ膨大な知識や概念体系をほとんど制約なしで自在にかつ直接的にオブジェクト指向に基づいて記述できる可能性が大きく広がったことは確かであり, 被験者ユーザたちもそれを高く評価した. この点は 2.4 節で述べた OODJ 構築の主たる狙いでもあり, 一定の実現を見たといってもよい.

8.2 文法, 変換文法, 記述モデルの評価

OODJ 文法

構文規則はオブジェクト指向構造化記述用に拡張した BNF 形式とした. この形式での構文提示についてはキーワードが日本語の単語であることもあって, 慣れるのに支障はほとんどなかったというのが記述実験被験者のコメントである. 実際, 例題を提示して対応を明示しながら口頭で構文規則の説明を併用すれば, (オブジェクト指向を「概念的には」理解しているドメインユーザが前提なので) 約 1~2 時間ほどで被験者のほぼ全員が十分な理解に達し, オブジェクト指向への理解も深まった. その後アドバイスを交えながら記述を試みたが, 1 日以内には例となる対象世界を十分に記述可能になった. 習得に必要な時間は新規な言語(PL 等)の習得時間に比べれば無視できる程度に短く, 違和感もなかつつやさしい. 構文に従った記述の困難さ・煩雑さ等は被験者からは聞かれなかった.

構文規則が記述モデルの記述要素を十分網羅していることの必要十分条件を確認する. 表 2~3 と表 4~7 の同番号をつけられた要素が対応する記述要素であることを示している. 各構文には対応する表中の番号の

記述要素が必ず含まれているので記述項目は十分である。それらの記述内容や対応関係にも矛盾は見出せない。したがって、構文規則の記述要素の構成は記述モデルを満たしている。特に相互関係属性(中でも相互作用属性)の記述は、後で JSMDL や PL に変換する際にこれらの明示的記述により、従来は見落とされがちな構造に関する記述が構文規則に明示的に規定された、ということが良好な評価を得た。以上から、前提とした記述モデルに対してならば必要十分の構文規則であると考えられる。

意味規則は、基本文型に沿った単文の平述文の接続にすることがその内容のほとんどであり、振舞い記述にも自然言語から見て違和感のある特殊な意味規則は規定されていない。わずかな記述制約も十分に緩い。したがって、その結果として自然言語の記述力は保持されたままで、しかも一意に規定できる単文の接続群が得られる。

OODJ 変換文法

変換構文法と変換意味規則は簡潔で一意的な日本語単文へと変換することがその趣旨であり、変換規則に問題とすべき点は見出せなかった。両者のほとんどは記述環境の設計と実装に組み込まれて反映されており、自然と利用・具現化する形になるように仕組まれてある。

記述モデル

オブジェクト指向記述モデルとしての構成については、十分な要素が用意されており、大きな不足や過剰な仕様は見出せず、記述に支障はないことがユーザの記述例から明らかにされた。記述モデルに相互関連や相互作用を属性の一部として記述するように規定した方法も問題は見出せなかった。むしろ、それらを必要要素を明示的にまとめて記述できるのはそうあるべき方法として良好であると評価された。

ただし、8.4 節の第 2 段落でも言及するように、振舞い(特に相互作用振舞い)の詳細を記述するための特別な記述モデルや文法(構文)が欲しいとのコメントもあった。振舞いに対する高い記述力を実現するにはそのようなモデルとそれに対する文法は確かに有効ではあるが、オブジェクト指向記述モデルとしての必須項目ではないと考え今後の課題とした。

8.3 記述環境の評価

記述段階を 4 つのステップに分割して各記述要素を重層的に複数回チェック・再検討できるようにした設計・実装は有効であった。また、記述すべき要素が GUI を用いて強制ガイド風に提示されているので、十分分かりやすく、見落としやミスがかなり減少した。

ドメインユーザは対象世界の把握・識別能力が高いので、まだ設計時のすべての機能を十分には実現していない現状のプロトタイプの GUI 記述環境エディタでも一応十分な記述環境として使える、というのが被験ユーザのコメントであった。形式的な文法チェックも有用であるとの感想も得た。

変換文法のかなりの部分は記述環境内に提示・ガイド・チェックという形で実装されて反映されており、記述作業に十分に役に立ったことはユーザの記述状況や記述例から結論できた。特に多様な構造記述を相互関係属性という形式で詳細かつ簡潔に記述・表示できる点に本記述環境の特徴があり、この点で他の記述環境と異なる。

以上から本記述環境が有効であることは被験者の様子とコメントおよび記述結果から結論できる。

8.4 振舞いの記述について

振舞いの記述は表 7 に定義された 4 種類の一般的な構文を除けば終端の記述は自然日本語である。振舞い記述に関しては、個別のドメインの特有の状況に適應させた構文・意味規則、あるいは記述方法が必要ではないかというコメントが記述実験の被験者からもあった。特に流体計算ドメインの学生から、流体の微分方程式の離散化過程と相互作用振舞い(たとえば差分方程式の解法を表す式)を便利に記述できる文法、あるいは何らかの記述法(枠組み)の要望が出た。

現在の OODJ は複雑なあるいは詳細な振舞いに対する記述構文を特には導入・規定していない。理由は 2.2 節でも述べたように、記述の理解性の高さと、専門的な厳密・詳細・適切な表現とがトレードオフであること、現在の OODJ の構文でも一応支障の少ない水準にあると判断したからである。したがって、複雑なあるいは詳細な振舞い、さらには各個別専門分野の振舞いに対する文法はオブジェクト指向構造記述形式(2.3 節)の共通基盤に立って、個々の分野各々で設計・実装すべきである、と筆者らは考える。ただし、少なくとも構文設計の原則やガイドライン、一部の分野に対する設計例等は提供する必要があるだろう。以上から 2 章の方針の合理性を確認した。

8.5 オブジェクト指向一貫記述日本語系の一環としての OODJ の考察

OODJ の弱点を強いて取り出せば、唯一、計算機可読ではなくそのままでは直接実行不可能という点である。しかしこの弱点も、自然日本語、OODJ、JSMDL という相似な言語のシリーズに、PL へのトランスレータを加えた記述言語の“系”^{1),2)}で見ると、

(1) モデリング全過程で自然日本語(PL フリー)

に近い記述を実現できる。

- (2) PL 記述 (現在は C++) への自動変換は JS-MDL トランスレータ^{1),2)}が支援する。

理想の狙いは (1) の実現にあり, OODJ はその一翼を担う言語である。この点を前節までに述べた特長とあわせて評価すれば, OODJ の弱点を補って余りある記述言語となるといえよう。

8.6 他の言語との比較・考察

分析段階の記述言語として OODJ と同条件で競合する研究や言語等は見出せなかったので, 比較的近い言語との比較・検討を行った。

8.6.1 仕様記述言語との比較

仕様記述には自然日本語がよく用いられるが, 曖昧さが残ることから, LOTOS¹²⁾, Z 言語¹³⁾といった仕様記述言語が開発された。これらの仕様記述言語の比較すべき特徴として以下がある。

- (1) 当初から厳密で形式的な仕様定義記述が必要。
- (2) 文法・記述形式・用語・記号が異質で習得の違和感が大。
- (3) PL への変換はすべてユーザ自身。

まず分析記述自体が試行錯誤中のものであるにもかかわらず, (1) のように当初から厳密な仕様に従うのはドメインユーザにとって困難でもあり, 無駄が多く, 目的に合った形式ではない。ドメインユーザは困難さを感じ, 記述する気が起きず当然避けるであろう。

次に自身の常用の PL (たとえば Fortran) に慣れた多くのドメインユーザにとっては仕様記述言語の特徴 (2) は論理数学的記述を要求するまったく異質な PL 並みあるいはそれ以上に習得の抵抗が大きい言語であることを意味する。しかも (3) でさらに PL への変換も自身でするので, ドメインユーザが実用として利用するにはまったくメリットがない。仕様記述言語をドメインユーザが分析記述用の実用言語として用いた記述例を筆者らは寡聞にして知らない。以上から現状の仕様記述言語は本研究の目的には合わない結論できる。

OODJ では, その最初の記述を作ったドメインユーザ自身が最後 (PL 記述直前の JSMDL^{1),2)}まで変換・記述を行うのであるから後の段階で正しい記述に変換される可能性は十分に高い。しかも OODJ は一意的な (曖昧性の少ない) 記述へと形式的に変換することが容易である。したがって曖昧さに起因する問題が OODJ 記述に起こる可能性は少ない。

8.6.2 日本語 PL, OOPPL との比較

実装段階の PL と分析段階の記述言語 (OODJ) とは本来は直接比較は意味が薄い。しかし OODJ の特

長を対比的に明確にする目的で比較を行った。

日本語 PL^{10),11)}のプログラムは, 表現の表面用語上は自然言語風にならべ, 通常の英語風の記述よりは比較的に分かりやすい。OOPPL (オブジェクト指向プログラミング言語) では確かにオブジェクト指向記述が明快かつ簡潔にできる。しかしいずれも PL の構文や意味規則, 実装技術・記法やテクニック, 慣用記法等の正確な使用に集中力をとられて, ドメインユーザが分析記述作業自体に集中し難いという大きな障害が存在する。この点は, 「はじめに」で述べたような新規な PL 習得の回避という点でもすでにドメインユーザの評価の差が決定的に大きく, デメリットは克服できない。

つまり新規な PL では, 表面的なメリットがあっても, 言語の根幹を成す文法が自然言語から見て特異な体系であっては, ドメインユーザが分析記述用言語として受け入れることは OODJ よりはるかに困難である, というのが結論である。

8.6.3 UML との比較

UML^{7)~9)}を分析段階のオブジェクト指向記述言語として, 筆者ら自身および被験者による UML と OODJ 両言語での記述経験から比較検討する。

両言語の共通点はオブジェクト指向という点である。UML は完成した既製品システム, OODJ はいまだ試作システムで拡張の余地が大きいという違いがある。基本的な設計から見た各々の特徴は,

- [1] OODJ はオブジェクト指向の構造化記述形式 (文法) を持つ母国語の自然言語ベースの記述言語であり, それに由来する特徴的な基本特性 (次節の表 15 参照) を持つ。
- [2] UML はダイアグラム記法を主体とし, 多様な記述形式・記述モデルを持ち, 完成度が高く詳細な言語仕様や記述環境を持つ。

個別比較の第 1 点は UML には PL 的表記法が見られる点である。すなわち, 分析段階でも現状では用語の多くが実装を視野に入れて早々と PL 特有の用語の一部を多用しており, すでに PL 記述の前駆的な影響が見られる。たとえば, 入出力, シグナル送受信, オブジェクトとその生成・消滅, 同期メッセージ, インタフェースの概念がそれである^{7),8)}。定量は困難ではあるがドメインユーザは異質な PL 風の詳細記述の強制にやっかいなものを感じ, 何よりもドメインユーザの書く気を失わせる, という点は記述実験の被験者の共通した指摘である。現状の UML に分析段階の記述言語相当のものがないのはドメインユーザには欠点としか見えない。自動生成 (スケルトン) ができない振舞

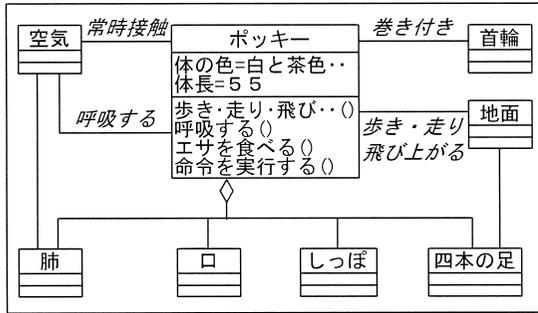


図7 UMLの表記例

Fig. 7 Description example using UML.

この記述には記述言語がぜひとも必要である。OODJにはこのデメリットはない。

第2点はUMLの持つ多様なダイアグラム表記法であり、視覚的・直感的な理解しやすさに対するドメインユーザの評価は高い。OODJにはこのメリットはない。

第3点はシステムの多様さ・完成度の高さ・普及度である。UMLの機能モデル(OMT)や動的モデル(UML)等、システムの成熟度・完成度、各種の支援記述環境、デザインパターンとのつながり、等はUMLの方がOODJよりも高度である。

基本的な設計上の特徴と個別比較の3点からみて、両言語の特徴はかなり異なり、その重点の特徴はほとんど重なっていないことが分かる。競合した結果OODJが不利なのは第3点だけである。

8.6.4 UMLとの比較検討に基づく改良

特徴に競合が少ないことから、OODJの改良案が容易に考えられる。前項の第2点に対してはすでに前論文¹⁾でも利用したように、UMLのダイアグラム記法において用語をすべて日本語としOODJの標準的な補助表記法として図7のように利用することで十分に補うことができ、有用であることが確認された。OODJは相互関係(構造)記述法が充実しており、相互関係は最初に図7のようにUMLのダイアグラム記法でまず記述し、それをOODJ記述に変換することは非常に容易かつ有用である。これでOODJが使いやすさと表記能力の点でUMLに近づいた。しかしドメインユーザにとってはUMLのPL的表記法の存在と習得コストはいぜんとして多少の障害として残る。

以上で、UMLの基本的な設計上の特徴8.6.3項の[2]を採り入れてUML並みのより高い表現力を獲得できると、記述言語を持たないUMLとの比較において[1]の特長が大きく有利さを増すと予測できる。

表15 自然言語ベース記述言語OODJの基本特性
Table 15 Fundamental characteristics of OODJ.

[1]: オブジェクト指向(構造化)記述構文則導入
[1a]: より高い構造記述力の獲得
[1b]: 理解容易な自然日本語のキーワードや説明文
[2]: 徹底した自然日本語寄りの文法・用語・記号・記述スタイル
[2a]: 記述や理解の十分な容易さ・簡潔さ
[2a-1] 強い記述力の十分な発揮
[2a-2] 分析・記述力の向上
[2b]: PLフリー
[2b-1] 新規言語習得の回避
[2b-2] ドメインユーザの歓迎実績
[3]: 日本語一貫記述言語系の一環
[4]: 有効な記述支援環境がセット

8.7 OODJの設計方針と特性の考察

既出の事実と考察を表15のように再度まとめた。基本特性[1]は自然日本語の本来の強い記述力の中に潜在埋没していた構造部分の記述力を明示的に引き出す枠組みとなり、ドメインユーザの持つ構造化の概念を実際の記述力に反映させ、純然たる向上(より高い記述力の実現)をもたらした。被験ドメインユーザとの検討結果でも、これが基本特性[2]との相乗効果で、記述力の低下や煩雑さの付加なしに記述の純向上を引き出した点が最も高く評価された。[2], [2a], [2b]の実現は、自然言語とほぼ同等の慣れ・非違和感、記述と理解の容易さ、原語の自然言語よりも高い記述力、そしてPLフリーという、優れた特性の記述言語をドメインユーザが新規に使えるようになったことを意味し、大きなメリットをもたらした。

以上の結果は自然言語ベースの記述言語という方式が、対象世界の分析記述作業において記述および理解が容易で利用価値の高いものであることを示しており、本研究で付加すべき文法の選択肢の1つとして導入したオブジェクト指向記述日本語文法もその成功例の1つとあってよいであろう。したがって2章での設計方針の合理性・目的性が検証できたと考える。

上述の評価・考察から、改めて母国語である自然日本語に基づいたドメインユーザ自身のドメインに対する認識の深さや蓄積知識の莫大さ・厳密さと、現状のドメイン分析記述〔定式化記述(2.4節)〕の貧弱さとの大きなギャップを見出した。このことはドメインユーザが発揮できない潜在的記述力の強さ・高さを再認識させた。その評価からドメインユーザのプログラム開発力の向上には、ドメインユーザの潜在記述

ただし、記述力の高さを向上させる記述言語の方式はほかにも多様な方向の拡張が見出されるので、この方式に基づく拡張は多様な将来性が期待できよう。

力を存分に発揮できる自然言語ベースの記述言語，しかも徹底した自然言語寄りの構文規則の導入，が最良の方針の1つであることを結論できよう。

8.8 他研究との比較

本研究と類似の狙いや目的を持つ研究は以前から多くが行われてきた^{14)~16)}。いずれも日本語記述から要求仕様記述，あるいはモジュール構造を得ようとするためのシステムや方法・方式の研究である。

これらを典型とする研究のいずれもが「オブジェクト指向パラダイムに基づき」「自然言語(風)記述を対象としている」「最終の狙いは要求仕様あるいはそれを満たすモジュール構造の作成」という共通点を持つものの，本研究とはその前提と方針・方法に大きな相違がある。OODJが他研究と異なる特徴は，

- (1) そのターゲット(使用者)をドメインユーザであることを大前提にした限定をしていること。また，日本語記述された文の分析・変換も自然言語処理等の自動化は行わず，ほとんどドメインユーザに任されていること。
- (2) 「手法・方法論の提案」というよりも，言語しかも記述言語としての日本語(記述日本語)の提案であること。
- (3) PLの要素(文法，用語，記号等)を可能な限り排除し，自然日本語のそれらを可能な限り残して活用したこと。OOPLとの対応付け等はJSMDLで初めて行う。
- (4) オブジェクト指向でモデル化した対象を「オブジェクト指向記述のできる日本語の一種で記述する」ことを目的に設計した言語であること。
- (5) ドメインユーザ自身の持つ主観的な要求の記述というよりもむしろ，ドメインユーザが専門としている対象世界の記述を客観的に相似性高く記述・実現することが狙いであること。
- (6) 実装に至る過程を複数の段階に明瞭に区分し，各段階を追って徐々に変換する設計方針を採ったこと。OODJはそのうちの1つの段階に対応する言語であること。

従来の研究のほとんどが上記の特徴の複数の点で異なる。したがって詳細な比較検討は本論文では行わなかった。

9. 結論と今後の課題

本論文の核心は言語仕様の根幹の1つである文法に対してオブジェクト指向構造化を図る「新規な構文の枠組み」を付加したことにより，オブジェクト指向構造記述力の高さの純然たる向上を実現した記述言語設

計にある。構文記述部分以外はすべて原語の自然日本語の範疇にとどめて記述力を保持した。

そのような記述言語の導入の結果，ドメインユーザが分析記述過程において十分容易なオブジェクト指向記述を可能にする記述の枠組みを自然言語の記述力を落とすことなく構成でき，ドメインユーザに歓迎されるPLフリーで実用的な分析記述言語の方法が確立された。記述例および記述実験の被験者のドメインユーザの評価，UMLとの詳細な比較からも，OODJのこの特性は支持された。

以上から結論として，このオブジェクト指向構造化記述力を強化した制限自然日本語の一種であるOODJは，ドメインユーザ用の分析記述言語としては十分に良好なもの1つであることが結論できる。

今後の課題としては，第1には動的モデル，機能モデル，UMLのダイアグラム表記法の導入による高い記述力の強化であり，第2には，複雑な階層構造の相互作用振舞いを記述する構文や意味規則，記述・起動手順の設計，第3にはPLフリーを実現するためにJSMDLとそのトランスレータの仕様の再検討と機能の向上，第4には記述環境の充実，構文/意味規則の詳細化・多様な拡張，があげられよう。

謝辞 本論文作成にあたり，原稿を閲読していただき，詳細なコメントや議論をいただいた茨城大学工学部情報工学科上田賀一講師に感謝の意を表します。

参考文献

- 1) 加藤木和夫，畠山正行：オブジェクト指向日本語一貫プログラミング環境，情報処理学会論文誌，Vol.40，No.7，pp.3016-3030 (1999)。
- 2) 加藤木和夫，畠山正行：オブジェクト指向日本語一貫プログラミング環境，情報処理学会第118回ソフトウェア工学研究会研究報告，98-SE-118，pp.15-22 (1998)。
- 3) 石井義之，畠山正行，加藤木和夫：オブジェクト指向日本語分析記述環境OOJの設計と開発，情報処理学会第122回ソフトウェア工学研究会研究報告，99-SE-122，pp.55-62 (1999)。
- 4) 石井義之：オブジェクト指向日本語分析記述環境OOJの開発，茨城大学大学院博士前期課程情報工学専攻学位論文 (Feb. 1999)。
- 5) Wegner, P.: Concepts and Paradigms of Object-oriented Programming, *key note Lecture in OOPSLA'89*. 尾内理紀夫(訳)，はやわかりオブジェクト指向，共立出版 (1992)。
- 6) Rumbaugh, J., et al.: *Object-Oriented Modeling and Design*, Prentice Hall (1991)。
- 7) Erickson, H.E. and Penker, M.: *UML Toolkit*, John Wiley and Sons (1998)。

- 8) Booch, G.: *The Unified Modeling Language User Guide*, Addison Wesley Longman (1999).
- 9) Rational Software Corporation (Ed.): *UML Description Guide* (1997).
- 10) 今城哲二: 日本語プログラム言語文献ノート, 情報処理学会秋のプログラムシンポジウム「日本のプログラミング」報告集, 1998-9-16, p.18 (1999).
- 11) 今城哲二: 日本語プログラム言語の設計, 情報処理学会冬のプログラムシンポジウム報告集, p.36 (1999).
- 12) 高橋 薫, 神長裕明: 仕様記述言語 LOTOS, カットシステム (1995).
- 13) Potter, B., et al.: *An Introduction to Formal Specification and Z*, Prentice Hall (1991).
- 14) 佐伯元司ほか: 自然言語からモジュール構造を得る手法について, 情報処理学会論文誌, Vol.31, No.11, pp.1479-1493 (1989).
- 15) 大野雅志ほか: オブジェクト指向分析支援システム CAMEO—日本語文章記述からの設計要素の自動抽出, 情報処理学会研究報告, SE99-14, pp.105-112 (1994).
- 16) 滝沢陽三, 上田賀一: 自然含語記述による要求仕様導出支援システムの提案, 情報処理学会論文誌, Vol.38, No.3, pp.626-633 (1997).

付 録

A.1 用語定義

- 相互関連: 複数要素間の静的な(時間の経過により変化しない固定的な)かかわり.
- 相互作用: 複数要素間の動的な(時間の経過に従って変化する)かかわり.
- 相互関係: 相互関連と相互作用をそのサブセットとする統合化概念. 属性の一種.
- 構造: 複数の記述要素とそれらの間の相互関連記述の組.
- 強い構造記述性: 対象世界の構造の側面に着目したモデル化記述をし, それが対象世界のモデル記述全体で特に強く支配的である, という特性.
- 完備な記述: 必要記述要素がすべて記述されていることを意味する用語. 3章まえがき脚注参照.

記述力: 本論文中では「強い」とは必ず記述可能なことを意味する「豊か」とは微細な差も書き分けることが可能なことを意味する「高い」とは対象に適合した厳密な記述が可能なことを意味する.
(平成 12 年 4 月 18 日受付)
(平成 12 年 7 月 5 日採録)



畠山 正行(正会員)

1976 年東京大学大学院博士課程(航空学専攻)修了. 工学博士. 東京都立航空高専を経て, 1990 年 10 月より, 茨城大学工学部情報工学科専任講師, 現在に至る. この間, 超音速凝縮流れ等の非連続気体流れの力学の研究に従事. 次いで, 数値シミュレーションユーザ用の計算機環境, オブジェクト指向シミュレーション, 一貫相似性モデリング過程等の研究開発に従事. 日本機械学会, 日本航空宇宙学会, 日本数値流体力学会, 日本シミュレーション学会, 日本計算工学会, ACM 等会員.



加藤木和夫(正会員)

1970 年北海道大学理学部物理学科卒業. 現在, 日立プロセスコンピュータエンジニアリング(株)勤務. 制御用プログラミング言語, ソフトウェア開発環境等の研究・開発に従事. 著書「ソフトウェア開発環境」(共著)、「PAD によるプログラム技法」(共著)ほか. 技術士(情報工学科目). 電子情報通信学会会員.



石井 義之

1974 年生. 1999 年茨城大学大学院博士前期課程情報工学専攻修了. オブジェクト指向開発環境, リポジトリ(データベース)の開発研究に従事. 現在, 日本電気(株)第一ストレージ事業部磁気テープ技術部勤務.