

# オブジェクト指向日本語分析記述環境 OOJ の設計と開発

石井義之\*、 畠山正行†、 加藤木和夫‡

## 概要

ドメインユーザ用のプログラミング環境として JOMON が開発されているが、オブジェクト指向日本語 (OOJ) を用いての論理モデリング段階の記述の際には、JOMON にはその構文法や、記述方法、記述支援環境等が充分には整っていなかった。そこで、OOJ 構文、四段階に区分した段階的詳細化記述法、OOJ 記述支援環境、等を構築・整備した。その結果、これらを組み込んだ記述環境あるいは記述言語はオブジェクト指向日本語一貫プログラミング環境の論理モデリング段階までの分析記述環境として一応十分な機能と性質を持ったものと結論された。

## Object-Oriented Japanese Analysis and Description Environment OOJ

Yoshiyuki Ishii\*, Masayuki Hatakeyama† and Kazuo Katougi‡

## Abstract

A consistently integrated Japanese programming environment, JOMON, had already been developed. The JOMON contains three insufficient points. These are the lack of the syntax system of the OOJ description language, the methodology to describe the logical model of the target world, and finally the OOJ description support environment. We have developed the OOJ syntax system, the OOJ analysis and description methodology and its support environment to recover these insufficient points of JOMON. As the result, we have almost fully satisfied these requirements. We have established the Object-oriented Japanese analysis and description environment OOJ.

## 1 はじめに

計算機をよく使うユーザの中に、ドメインユーザ<sup>1</sup>と呼ばれる一群のユーザがある。彼らはかなりの数を占めるが、ソフトウェア開発力が低い故に、また開発するソフトウェアが最終目的物ではない故に、専門家であるソフトウェア開発者以上にプログラミング環境に対するニーズは高い。しかし、「新しい」計算環境 (プログラミング環境) に対する非積極姿勢、特に主用言語 (多くは Fortran) に対する現状維持意識の強さのためか、

従来ドメインユーザ向けのプログラミング環境は殆ど無きに等しかった。

しかしこの様なドメインユーザの無理とも思える要求を実現させる道がある。それは分析の当初から、ドメインユーザ全員共通のもう一つの主用言語、しかも生来の主用言語、である日本語 (或いはその類似言語) を用いるという方針で一貫してモデル化と設計を行う、という考え方とその実現方式である。我々は既に、モデリング方法については一貫モデリング過程を実現する方法論を開発済みである。従って、モデリング方法とその表現言語の両面から一貫したプログラミング過程が実現可能である。オブジェクト指向パラダイムを基盤概念として、概念モデルの記述からプログラム記述までを相似な日本語表現を用いて連続的に変換していく方法をオブジェクト指向日本語一貫プログラミングと呼ぶことにする。この考えを具体化したのが図 1 である。

\*茨城大学大学院理工学研究科情報工学専攻

Ibaraki University

†茨城大学工学部情報工学科

Ibaraki University

‡日立プロセスコンピュータエンジニアリング (株)

Hitachi Process Computer Engineering Co. Ltd.

<sup>1</sup>ドメインユーザとは、ソフトウェア開発やプログラミング言語以外の専門家であって、コンピュータを道具として使う一群のユーザを指すものとする。

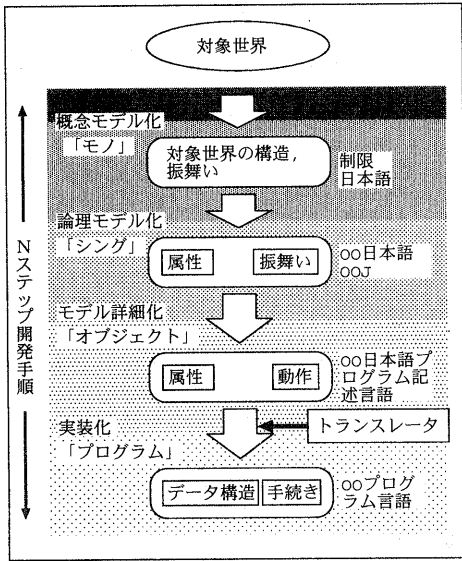


図 1: 日本語一貫プログラミングの方法論

現在これに対応できる記述支援環境として JOMON が開発されている [1, 2]. JOMON は, Japanese Object Modeling Notrs の頭文字から命名されたシステムで, オブジェクト指向に基き一貫した日本語系記述を用いたプログラミング方式およびこれを基礎にした個人用プログラミング環境である. 図 2 にその概略を示す [2, 1]. 図 2 は一貫プログラミング過程とその支援環境を表す図である. 三段階の開発ステップは互いにリポジトリを通じて情報の共有をする. ドメインユーザは OODJ の相互に変換可能な三種類の日本語系記述言語 (自然言語, OOO, JSMDL) を, 用いることによって順次オブジェクト記述を完成させ, 最終記述ソースコードはトランスレータにより図 2 の様に C++ プログラムに変換される [1, 2].

## 2 JOMON の不十分な点

JOMON では確かに図 2 の様に日本語系記述の一貫性は一応ある水準にまで達した実現が為されたと言えよう. しかし, いくつかの不十分な

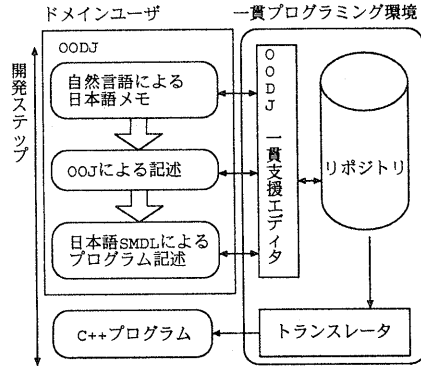


図 2: 日本語一貫プログラミングとその支援環境

点が論理モデリング段階に残された.

まず第一点として, JSMDL においては構文がある程度詳細に規定されているが, 論理モデリング段階に当たるオブジェクト指向日本語 (OOJ) については, 多少の構文らしきもの (メッセージ記号や, 制御構文など) が定義されたに過ぎず, 基本的には自然言語記述と同様に自由記述であり, 論理モデリング段階の記述として必要な構造記述などは必ずしも望めず, ユーザの記述力に依存してしまっている.

次に第二点として, 日本語一貫記述方式の複数の各ステップにおける記述内容を規定することで段階的に詳細な記述を重複無く仕上げて行く方法を探っているが, 論理モデリング段階の OOJ においては殆ど無いに等しく, その記述方法が不明確である.

第三点として, 記述支援 GUI (エディタ) 環境も JSMDL に対しては整備されているが, 概念モデル段階 (自然言語) と論理モデリング段階 (OOJ) に対しては, 自由記述が出来る日本語メモエディタが実現されているに過ぎない.

このような不十分さのため, 現状の OOJ 記述では, どこまでの記述でプログラム記述変換可能であるかとか, 概念モデルと矛盾などがないか否か, などについてはユーザが本来持っている記述力と知識や判断力に頼っており, それ以外では試行錯誤をもとにした経験に基づいて判断するしかない.

そこで、本研究では論理モデル化段階における記述言語の構文を新たに規定し、JSMDLでも行われてきた段階的詳細化手順を開発する。最後に記述支援環境(エディタ群)を構築する。

### 3 オブジェクト指向日本語 OOJ

オブジェクト指向日本語 OOJ とは、自然言語の語彙と意味体系を持つという側面と、オブジェクト指向パラダイムに基づく記述に要求される様な明示的に区別されたモデル化単位で、カプセル化、情報隠蔽、相互作用記述形式を記述できるという側面の両側面を併せ持つ記述言語である。この様な記述言語の土台となる言語は、表現力が強くかつ豊かな表現が出来る自然言語が最も適している。OOJ の満たすべき条件は、

1. 表現メディアは日本語の表現形式や語彙・用法、意味体系・知識体系の一部を借用。OOJ の記述文字は漢字・仮名を主用する。
2. 記述順序・構造において、オブジェクト指向の概念体系を記述でき、オブジェクト指向パラダイムを用いてモデル化した構造を記述する事が可能な言語である。必然的に対象世界の構造に対する記述力が強い「構造化記述用日本語」であると共に、オブジェクト指向の特徴的な制約、即ちカプセル化、情報隠蔽、相互作用メッセージ、クラス(階層)、等が上記の表現メディアで明示的に記述可能でなければならない。
3. 対象世界の自然言語記述は、その記述のみでは多くの場合、コンピュータシミュレーションシステムにした場合には情報が不足する。即ち、対象世界の「モノ」単位での構造記述は、複数の記述・編集段階で補う必要がある。
4. OOJ は自然言語日本語と JSMDL との中間に位置する記述言語であるので、自然言語としての性格持ち、プログラミング言語記述にも変換可能な性質が必要である。プログラム言語に近すぎる構文規定や記述法をとると、ドメインユーザにとって使いに

くくなり、日本語的な解釈や理解がしにくくなり、使われなくなる。自然言語に偏ると、プログラム言語への変換が困難になる。

以上の条件を考慮して、OOJ 分析記述環境を設計した。

## 4 OOJ 構文

### 4.1 構文総則

OOJ の各文は"%<キーワード>:"で始まり、次の"%<キーワード>"の前までの記述がキーワードの内容となる。例えば,"%シング:"という文は次に"%記述","%属性:"や"%振舞い:"などのキーワードが現れるまでがシング名となる。また,"%シング:"が現れるまでがそのシングの定義となる。キーワードは必ず%"で始まる。"%シング:"や"%属性:"のように名前や説明がそのあとに続く場合は":"を書きそのあとに記述する。

1. 構文に現れる<>に囲まれた部分は非終端記号である。
2. " " で囲まれた部分、<>で囲まれていない構文記号以外の部分は、終端記号である。
3. 構文 | は「又は」の意味である。
4. ... は繰り返しを表す。( )... は( )内の繰り返しを意味する。

OOJ 構文全体の詳細は付録 Aを参照されたい。また、記述サンプルについては長くなるので石井の論文 [3] を参照されたい。

### 4.2 シング定義

<OOJ 記述>: OOJ 記述は一つ以上のシングの定義から構成される。

<シング定義>: シングの説明記述と、0 個以上の属性セクション、振舞いセクションから構成される。

<説明記述>: 日本語の単文の接続で行う。

<日本語単文>: 緩やかな制約を持つ単文の自然言語日本語で記述し、"."で終わる。

### 4.3 属性セクション

〈属性セクション〉：属性セクションは一般属性セクション又は関係属性セクションによって構成される。

〈一般属性セクション〉：一般属性は、属性名、属性の説明記述、属性に対する制限・制約事項の記述と属性の種別から構成される。

〈制限・制約記述〉：一般属性属性、内部振舞い、相互作用はとりうる範囲、起動条件などの制約を記述する。

〈属性種〉：汎用属性と、汎化または集約の上下位のシングと、その他のシングからの作用制約を定義する。

〈汎用属性〉：他のシングと共有したい属性について指定する。

〈関係属性セクション〉：関係には汎化、集約、一般関連の3種類がある。

〈汎化〉：汎化はその関係名と説明記述、汎化上位及び下位にあるシングのリストを記述する。

〈集約〉：集約はその関係名と説明記述、集約上位及び下位にあるシングのリストを記述する。

〈一般関連〉：汎化、集約以外の関連は全てここに記述。

### 4.4 振舞いセクション

〈振舞いセクション〉：振舞いは、他のシングと関係を持つか持たないかによって内部振舞いと相互作用に分類される。

〈内部振舞いセクション〉：内部振舞いは、内部振舞い名とその説明記述、振舞いの概略を示すアウトライン記述と、振舞い起動や動作中に受ける制限や制約に関する記述を行う。

〈アウトライン記述〉：振る舞いのアウトライン記述はアウトライン文の繰り返しで記述する。

〈アウトライン文〉：アウトライン文は、振る舞いの内容を記述する日本語単文と、条件分岐を行うための選択文と処理を繰り返す繰り返し文で構成される。

〈選択文〉：選択文は、条件文を評価して振る舞いの内容を選択する。

〈条件記述〉：条件は日本語単文によって記述する。

〈繰り返し文〉：“LOOP” 〈条件記述〉 “DO”

〈相互作用〉：紙幅の関係で割愛。

## 5 OOJの記述区分と手順

オブジェクト指向パラダイムを用いて分析や記述をする手法は既に数多く存在し、利用されている [5, 6]。しかし本論文では、実用性を重視した手順設定を独自の考えで構成した。本研究では四段階のステップに区分した。

まず第一段階では概念モデル記述を、OOJ 論理モデル記述段階へ受け入れるためのいわば「仕分け」作業ステップである。従って、OOJ 構文の要素各々に対応する記述或いは単語を抽出して、構文の各項目に振り分ける作業である。まず、対象記述からシングを識別・抽出する。次に識別された各シングの属性・振舞い、シングとシングとの関係を識別する。この段階を対象識別ステップと呼ぶ。

第二段階では、対象識別段階で記述されなかった対象世界からの新規な諸要素の直接抽出、記述の分類・振り分けと追加、それに伴う属性や振る舞いや相互関係の記述の追加・補完・訂正を行う。この段階では各シングに関して振舞いの詳細の記述、相互作用に関する詳細な記述、また、属性や関係などに制約があればそれらを全て記述する。このステップで目的記述に必要な要素は原則として全て揃えられる。この段階を構造化記述ステップと呼ぶ。

第三段階では、前段階までに識別された全てのシング間の相互関係を改めて再点検・再検討を行い、その正当性・合理性の検証を行う。この段階を相互関係規定ステップと呼ぶ。

第四段階では、モデル記述全体についてのレビューを行う。具体的には、全体の相互関係が正しいものかを再チェックし、シングをもっと抽象化できないかを再検討し、使われないシングの削除等を行う。この段階をレビューステップと呼ぶ。

以上から分かるように、4つのステップは重複しており、前段階に見落としたものも後の段階で

見いだしたり、最後の第4段階でチェックする機能を付けて分析及びその結果としての記述の完成度を高める方式にした。

## 6 OOJ 記述支援 GUI エディタ

前章までの議論、特に OOJ 構文、四段階の記述区分段階を考慮しつつ、JOMON の日本語メモエディタと JSMDL エディタとの連携を保った連結型の OOJ 一貫記述支援エディタを設計した。OOJ エディタは、四つの各段階（ステップ）のそれぞれにおいてシング定義、属性、振る舞い、関係の作業を要素ごとに編集しやすいように設計し、一貫支援エディタとして内部に組み込んだ。図3に連続的に変化する記述過程を支援するエディタの構成を示す。

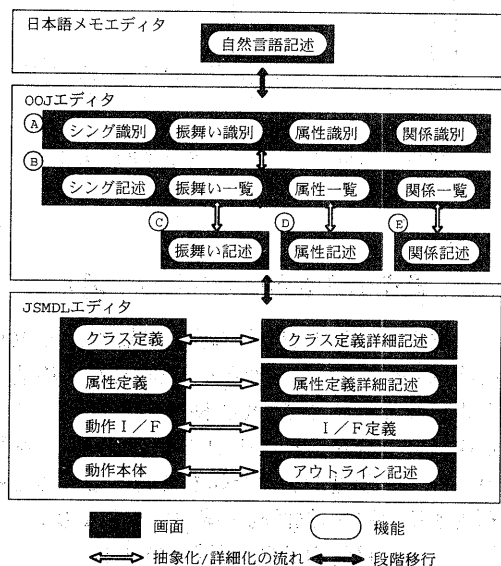


図3: 日本語一貫支援エディタの機能構成図

### 6.1 OOJ エディタの画面構成とその役割

OOJ エディタは論理モデル化段階を支援するエディタである。日本語メモエディタを使って記述された対象記述から、第4節で説明した論理モデルを OOJ を用いて記述することを支援する。

- シング・属性・振舞い・関係識別 (図3の(A))

ユーザは対象記述の日本語文からシングや属性、振舞い、関係の識別をし、この対象識別段階の処理を行うこのエディタに登録する。OOJ 記述の最初のこの段階では、概念モデル記述の中から、論理モデルを構築する四つの要素（シング、属性、振る舞い、相互関係）の抽出と振り分けを行なうことに集中する。

- シングの記述 (図3の(B))

構造記述化段階において、識別されるべき四つの要素ごとに記述を行なう。追加されるべき要素を見いだしてそれに対する記述も加える。シングとその属性・振舞いについては図3(B)にてシング単位で行なう。関係については、二つのシングの間関係を関係元、関係先を視覚的に理解できるビュー図3(D)にて記述を行なう。

- 属性・振舞い・関係の記述 (図3の(C),(D),(E))

シングに関して理論的に可能な汎化関係にあるすべてのシングを識別する作業は図3の(D)にて行なう。新たな汎化関係を識別した場合は、その関係についての記述を行ない、その記述を図3(A)へ持っていき、新しいシングの識別を行なう。

- モデル記述全体のレビュー

ユーザが各記述を再検討し、見直す。シング記述と関係記述を見るので図3(B),(D)を利用してレビューする。

## 7 OOJ エディタの利用法

OOJ エディタ部分を拡張した JOMON は、基本 OS として Windows98/NT4.0、GUI フレームワークとしては MFC4.2、開発環境は VisualC++6.0 を利用して実装している。

本章では取り込み画面例を参照しながら、段階的詳細化記述をどのように行うことが可能かを説明しかつ、実装システムの評価も行う。

## 7.1 対象識別ウィンドウ： 図4

概念モデルの記述は画面右上の部分に表示される。ここから各要素を選択し、その下の4つのリストボックスにドロップすることにより行なう。特定のシングの属性や振舞いを対象識別する場合は4つのリストボックスの中の左上のリストでシングを指定してから、属性や振舞いのリストボックスにドロップする。

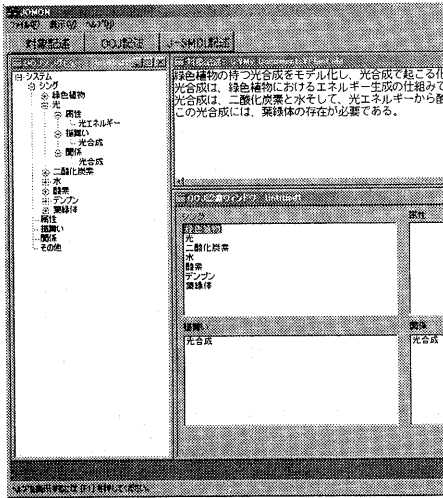


図4: OOJ エディタ (対象識別ウィンドウ)

## 7.2 構造記述化ウィンドウ： 図5

構造記述化段階のシングの記述、シングの属性、振舞い、関係、制限・制約についての記述・編集を各画面で行なう。新しく追加した記述については図4のエディタで未記述のシングなどを新規に識別することができる。上部のコンボボックスに識別済みのシングが登録されている。そこで記述対象のシングを選択すると、そのシングの情報が下部に表示される。

シングの属性、振舞い、関係は、下部のタブコントロール内にリスト表示されるので、リスト内の要素をダブルクリックすることにより属性、振舞い、関係の記述、制限・制約などを記述するダイアログを開くことができる。

属性、振舞い、関係記述、制限・制約の各ダイアログでは、アクセスを許可する相手のシングを記述する項目がある。この項目を規定・点検するのが、相互関係規定ステップにおける作業になる。関係記述ウィンドウでは、関係の追加、削除、変更が可能である。

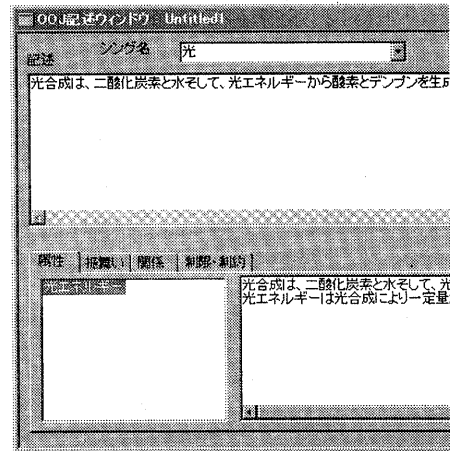


図5: OOJ エディタ (構造記述化段階)

## 7.3 構造化日本語ウィンドウ： 図6

上記ウィンドウ・ダイアログにて記述したOOJ記述は、OOJ最終記述形として図6の様に構造化日本語ウィンドウにOOJ構文(第3節)に従った構文で出力される。モデル全体のレビューを行うレビュー段階で記述全体を確認するためにも利用する。図中の例は緑色植物の光合成のシミュレーション目的のOOJ記述である[4]。

## 8 考察・結論

### 8.1 オブジェクト指向日本語

オブジェクト指向日本語OOJの構文は、論理モデルの構成単位であるシング、シングを構成する振舞いと属性、そしてそれらの間の関係を明示的に記述することができた。また、それらシングや属性などの性質や制限などについては、表現

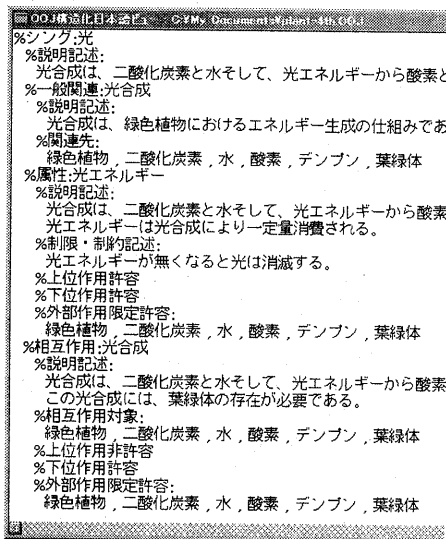


図 6: OOJ エディタ (レビュー段階)

力に優れた自然言語日本語の記述力を活用することができた。また、振舞い内部については従来からの手続き型の記述を許し、また制御構文を導入することによりアルゴリズムの記述が容易になった。

これら、OOJ の特質はオブジェクト指向に基づく論理モデルを表現するのに十分な表現力を持ち、その記述を容易にしたことは明らかである。この事は記述例の比較からも読み取れる。

ただし、本稿では相互作用に関する記述を割愛したので、情報隠蔽、メッセージパッシングで実現される相互作用に関する件は全て省略した。

## 8.2 段階的詳細化記述手順

本研究では、四つの段階分けを基に論理モデルを段階的に詳細化して記述する分析記述手順を実現した。この分析記述手順の枠組みにより具体的な手順を用いてユーザの記述を誘導し、概念モデル化段階の対象世界の記述を論理モデル化段階のオブジェクト指向パラダイムを用いた記述へと異和感無く変換することができる様になった。この事は前節の記述サンプルに依る比

較から検証された。

## 8.3 分析記述環境

実現した分析記述環境では、四段階の各段階で記述要素を充分重ねてチェックすることにより、対象記述にある概念モデルから論理モデルへの段階的にスムーズな移行とモデル間の構成要素の対応づけを実現している。

この事から本研究で開発した記述環境は論理モデリング段階の記述を OOJ を用いて行うための適切な枠組みを与え、ガイダンス(多少とも強制力のある)を行っている分析記述 GUI(エディタ)環境であると言えよう。

## 8.4 結論

オブジェクト指向日本語一貫プログラミング環境における分析・設計段階、論理モデル化段階の記述のガイドラインを提示し、記述言語の設計、コンピュータ上で支援するためのエディタの設計・実装をおこなった。記述言語である OOJ は日本語構造の中にオブジェクト指向の基礎概念をエディタという枠組みとして入れることで、用法と記述対象を制約した自然言語によるオブジェクト指向記述をおこなえるように設計した。

以上から OOJ 構文、段階的詳細化記述手順、エディタを主体とした環境を総合的に組み合わせた記述環境あるいは記述言語はオブジェクト指向日本語一貫プログラミング環境の論理モデリング段階までの記述環境として一応十分な機能と性質を持ったものと結論できよう。

ただし、例えば OOJ 文法を如何にするか、複雑な相互作用の記述を如何に記述性や可読性を保ちつつ簡潔に記述するか、本記述環境の仕様を JSMDL を用いる設計・実装支援環境で如何に拡張するか、等の問題が課題として残っている。

## 参考文献

- [1] 加藤木和夫, 畠山正行, 小林秀行, 「シミュレーション向けのオブジェクト生成支援環境の設計と実装」, オブジェクト指向'95 シンポジウム, 情報処理学会シンポジウム論文集, Vol.95, No.3, pp.205-212, 1995年6月2日.
- [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」, 情処研報, 98-SE-118(1998).
- [3] 石井義之, 「オブジェクト指向日本語分析記述環境 OOJ の設計と開発」, 平成 10 年度茨城大学大学院理工学研究科修士学位論文, 1999 年 2 月.
- [4] 三輪 知雄 監修: 「現代生物学体系 10 植物の生理・生化学」, 中山書店.
- [5] Martin Fowler, Kendall Scott, 羽生田栄一訳, 「UML モデリングのエッセンス」, アジソン・ウェスレイ・パブリッシャーズ・ジャパン, 1998 年 10 月.
- [6] Martin Fowler 著, 堀内一監訳, 「アナリシスパターン」, アジソン・ウェスレイ・パブリッシャーズ・ジャパン, 1998 年 6 月.

## A OOJ 構文

```

<OOJ 記述> ::= <シング定義>...
<シング定義> ::= "%シング" シング名
    [ <説明記述>... ] [ <属性セクション>... ]
    [ <振舞いセクション>... ]
<説明記述> ::= "%説明記述:"
    <日本語単文>...
<日本語単文> ::= 日本語文 "。"
<属性セクション> ::= <一般属性セクション>
    | <関係属性セクション>
<一般属性セクション> ::= "%属性:" 属性名
    [ <説明記述> ] [ <制限・制約記述> ]
    [ <属性種> ]
<属性種> ::= [ <汎用属性> ] [ <作用制約> ]
<汎用属性> ::= "%汎用属性"
<制限・制約記述> ::=
    "%制限・制約記述:" <説明記述>
<関係属性セクション> ::=
    <汎化> | <集約> | <一般関連>
<汎化> ::= "%汎化:" 関係名 [ <説明記述> ]
    [ "%上位:" (上位シング名)
      [ ("," 上位シング名)... ] ]
    [ "%下位:" (下位シング名)
      [ ("," 下位シング名)... ] ]
    [ "%集約:" 関係名 [ <説明記述> ]
      [ "%上位:" (上位シング名)
        [ ("," 上位シング名)... ] ]
      [ "%下位:" (下位シング名)
        [ ("," 下位シング名)... ] ] ]
    [ <一般関連> ::= "%一般関連:" 関連名
      [ <説明記述> ]
      [ "%関連先:" (関連するシング名)
        [ ("," 関連するシング名)... ] ] ]
    [ <振舞いセクション> ::=
      <内部振舞いセクション>
      | <相互作用セクション>
    <内部振舞いセクション> ::= "%内部振舞い:"
      内部振舞い名 [ <説明記述> ]
      [ <アウトライン記述> ] [ <制限・制約記述> ]
    <相互作用セクション> ::= "%相互作用:"
      相互作用名 [ <説明記述> ]
      (相互作用相手シング名)
      [ ("," 相互作用相手シング名)... ] <作用制約>
      [ <アウトライン記述> ] [ <制限・制約記述> ]
    <アウトライン記述> ::= "%アウトライン記述:"
      <アウトライン文>...
    <アウトライン文> ::= <日本語単文>
      | <選択文> | <繰り返し文>
    <選択文> ::= "IF" <条件記述> "THEN"
      <日本語単文>... [ "ELSE"
        <日本語単文>... ] "ENDIF"
    <繰り返し文> ::= "LOOP" <条件記述> "DO"
      <日本語単文>... "ENDLOOP"
    <条件記述> ::= <日本語単文> [ ("AND" | "OR")
      <日本語単文> ]... ]
    <作用制約> ::= <上位作用制約>
      <下位作用制約> <外部作用制約>
    <上位作用制約> ::= <上位作用許容>
      | <上位作用非許容> | <上位作用限定許容>
    <上位作用許容> ::= "%上位作用許容"
    <上位作用非許容> ::= "%上位作用非許容"
    <上位作用限定許容> ::= "%上位作用限定許容"
      <許容先シング名>...
    <下位作用制約> ::= <下位作用許容>
      | <下位作用非許容> | <下位作用限定許容>
    <下位作用許容> ::= "%下位作用許容"
    <下位作用非許容> ::= "%下位作用非許容"
    <下位作用限定許容> ::= "%下位作用限定許容"
      <許容先シング名>...
    <外部作用制約> ::= <外部作用許容>
      | <外部作用非許容> | <外部作用限定許容>
    <外部作用許容> ::= "%外部作用許容"
    <外部作用非許容> ::= "%外部作用非許容"
    <外部作用限定許容> ::= "%外部作用限定許容"
      <許容先シング名>...

```