

## オブジェクト指向自然日本語記述言語 OONJ の設計とその記述例

島山 正行†

**要約:** 本論文では非情報系の科学技術諸分野の専門家(以降, ドメインユーザ (DU) と呼ぶ) が扱う対象世界をオブジェクト指向(以降, OO と略)に基づいてモデリングし, それを自然日本語をベースとして記述する分析段階の記述言語 OONJ を設計/提案する. そのため, OONJ で使える要素種類及びその内部構成のリスト, 記述構造化のための規則(構文規則), 意味解釈規則を設計した. 更に多様/多重な構造化とその記述を許す様に規則を拡張すると共に, しばしば現れる定型的な文の集まりに対して構造化を施す複合文様式も設計した. これ等広義の文法は以前開発した OO 構造化フレーム OOSF の上に実現した. 以上のような広義の文法を持つ OONJ を提案し, 幾つかの記述例を作って議論・検討した. その結果 OONJ は新たな OO 分析記述言語として必要な記述力を持つと共に, DU が実用的に使える有用な記述言語であることが分かった. 今後更に記述例を増やして OONJ の評価と普及, 及び言語の拡張を行う計画である.

## A Design of Object-oriented Natural Japanese Description Language OONJ and Its Description Examples

MASAYUKI HATAKEYAMA†

**Abstract:** In the present paper, we have designed and propose a new Object-Oriented (hereafter OO) analysis Description Language OONJ based on the Natural Japanese. The OONJ has been developed for the Domain Users (DUs) to analyze and describe their target worlds in their expert domains. The DU is defined as the expert of a certain professional domain except the computer science/technology. To attain the aim, we have actually designed and constituted the followings; the element kind list and their internal element constitutions list for each element to show the usable vocabulary in the OONJ, and its syntax and semantics. The special rules for the multiply or diversely defined structures and their representation methods have also been developed for a variety of the target worlds of the DU's. These extended rules have been developed based on the OO Structured Frame OOSF. We have described various description examples to estimate, discuss, and revise the OONJ. As the results, we have confirmed that the OONJ provide the enough description power as a newly provided OO analysis description language, and has shown to be the valuable and useful description language for the actual various usages of the DUs. As for the conclusion, the OONJ has successfully been developed for the OO-structured descriptions of the DUs.

## 1. はじめに

我々の研究グループではここ何年かに亘ってオブジェクト指向(以下OOと略す)パラダイムに基づき自然日本語を基盤としたOO分析モデリングと記述を行い, それを順次OOプログラミング言語記述に半自動で変換し, プログラムを生成しようという研究を進めてきた<sup>1)~5)</sup>. この計画の一環として本研究においてはOO自然日本語記述言語 OONJ の設計を提案し, その記述力や記述容易性等を検証するために OONJ を用いた記述例を提示して考察を行う.

本研究の対象ユーザは広く科学技術計算/分析/設計に関わる分野を専門としている科学者や技術者である. 以降, 彼等のことを纏めてドメインユーザ(以降, DU と略す)と呼ぶ. ただし情報科学や情報工学の分野の専門家はDUからは除外される. この様なユーザを対象とした理由は, 彼等は人数的にも量的にも情報分野のユーザよりは遙かに多くのプログラムを書いていると推定されるにもかかわらず, プログラム作成支援の環境が殆ど存在しないからであり, 潜在するこのマーケットを顕在化させることが出来れば非常に大きい需要を持っているという経験的な確信があったからである.

さて最近の論文において我々はOOに基づいた構造化記述を実現するため, OO構造化フレームシステム

† 茨城大学工学部情報工学科

Department of Computer &amp; Information Sciences, Faculty of Engineering, Ibaraki University

OOSF<sup>7)</sup>を提案した。OOSFでは自然日本語(以降、NJと略)記述に対して、そのNJ記述本体を単文単位に離散化して単位文として十分な構造化表現を実現する方法を提案し、その方法の合理性と有効性を幾つかの実験や記述例等を通して立証した。

そこで本論文ではこの形式的なOO構造化の枠組みを利用し、OOSFを用いたNJ記述の構造化と以前の論文<sup>4)</sup>で提案したDUのためのOOモデリング方法とを前提にした上で、DUのためのNJを基本としたOO構造化記述を実現する総合的な諸条件を満たし、幾つかの点でNJを越える記述力を持たせた分析記述言語OONJとして提案する。

## 2. OO分析記述言語OONJの設計

### 2.1 記述言語系OOJの全体設計計画

現在我々の研究グループにおいて開発されている記述言語系OOJとは以下の構成を持つ。(1)分析記述言語OONJ(本論文)

(2)OONJトランスレータ

(3)実行記述言語OEDJ<sup>6)</sup>

(4)OEDJトランスレータ<sup>6)</sup>

(5)上記の四つを内部要素として統合された記述言語系OOJ

(6)分析からプログラム生成までの上記五つの言語(系)をサポートするGUI一貫支援環境

OONJを用いて分析記述されたモデリング後の対象世界記述は(6)の支援環境を通してDUから収集された情報(データ型、数式定義等)を併せてOONJトランスレータに掛けられ、OEDJ記述に変換される。OEDJ記述は更に支援環境から必要なデータ(例えば、変数名、配列定義、関数名、制御情報)を受け取った上で、OEDJトランスレータに掛けられてOOプログラミング言語(現状はC++)記述を生成する、という手順で計画されている。

なお、OONJトランスレータの段階以降において必要で、かつ対象世界からモデリング可能である要素、値、記述は、OONJ記述と併せて支援環境を通してDUに依って新たにモデリングされて入力されるか、OONJ記述から変換されて以降の利用のために準備される。この設計仕様も含めた分析記述言語はOONJ+と呼ぶことにする。

### 2.2 OONJの設計目標

DUがOOモデリング/記述すべき想定対象世界は非常に多様である。またDUがOONJを利用する用途

として、自身の対象世界を詳細に分析記述をしておき、OOPL記述に変換するのではなく自身の理解のため、共同研究等のための他の人への自身の世界についての体系的説明や議論の叩き台とするため、その他多様な目的に使うのも一つの重要な用途である。

従って、分析記述言語としては多様な対象世界に対応して可能な限り詳細な記述を実現する設計仕様にしておく必要がある。それに沿った具体的な目標項目を表1に挙げる。表中の(1)では多様な構造化記述を実現できるような仕様への拡張、(2)では任意に詳細な対象世界の記述が出来るような仕様への拡張、そして(3)では定型的な記述(文の集まり)の構造化の実現、(5)では元となった対象世界とその記述とが可能な限り高い相似性(相同性)の実現、その他を目標とした。

表1 OONJ設計の目標

Table 1. Constraint Rules for NJ Descriptions

- |  |
|--|
| (1) 同一内容の多様(複数)表現、意味や別名/別記述等の多義多重多様な記述を許す記述言語。 |
| (2) 全対象世界の階層記述化とその記述法                          |
| (3) 定型パターン化された文の集団の記述構造化。                      |
| (4) NJベースのOO記述言語という枠組みの堅持                      |
| (5) 対象原世界とその記述との高い対応関係の実現。                     |
| (6) 計算機世界への持ち込みを将来的に狙える準備。                     |
| (7) 僅かな制約以外はOO記述の自然言語性の維持                      |

### 2.3 OONJの設計方針

本記述言語OONJの設計に当たって、詳細な記述の実現という方針以外に最も重要視した点は、モデリングした結果の対象世界を可能な限り正確に表現できる記述言語とする、という点である。それは、DUが科学技術計算/解析/設計/シミュレーション分野の人達である故にOONJ記述を対象世界の出来る限り精密な相同な写像記述にする必要があるからである。この設計方針の下での広義の人工的記述言語の設計に必要な項目は以下の様である

(一)OONJで記述可能な語彙(ボキャブラリ)を定めること。

(二)モデリングした要素(相互関係の要素も含む)の構造化記述の方法を汎用的に規定すること。

(三)既に出来ている記述が対象世界のどの要素とどの様な関係にあるかを明確に解釈できる方法を汎用的に規定すること。

まず、第(一)項目に対しては、OOモデリングに使われる要素種類を分類/定義し、表形式で設計するこ

とにより実現した。第(二)項目については対象世界の構造を正確に記述するための構造化規則(通常は構文規則というべきもの)を設計する事で実現した。第(三)項目については記述に対する意味解釈規則を定めることで、対象世界のどの要素にどの様に関係しているのか、即ち意味を明らかに規定する事で実現した。

つぎに第(二)項と第(三)項は逆方向の対応関係にあり、第(二)項で規定する規則は対象世界の記述のための必要条件であり、第(三)項で規定する規則はOONJ記述が対象世界を正確に表現しているか否かの十分条件であると考えられることも可能である。つまりモデリングされた対象世界とOONJ記述とは理想的には等価でなければならない。あるいは両方向の対応関係を含むこの様な構造化(即ち、相互関係を与えること)は相同の関係にある構造化であることが理想である、とも言える。

対象世界とOONJ記述規則とのその様な関係を表したのが、図1である。図において、まず対象世界から構成要素定義規則(第(一)項の要素種類表)を使ってモデリング(着目、抽出、名前付け)<sup>4)</sup>し、これ等の要素を構造化規則(構文規則)を用いて構造化記述し、その記述が対象世界をどの様に正確に表現しているかを意味解釈構造化規則(意味規則)で確認する。十分に正確に表現されていれば等価な表現あるいは相同構造化と評価できることになる。ただし、この様に評価すれば相同構造化と言えるかという点の判断はDU自身には確信できても客観的な証明は困難である。

OONJの設計にはこれ等三つの記述規則がセットで必要である。設計及び利用の便宜上、これ等を纏めてOONJ記述三法(要素規則、構造化記述規則、意味解釈規則)と呼ぶことにする。

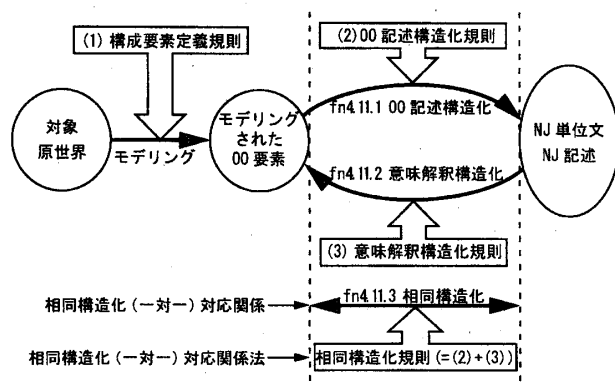


図1 OO構造化の実現とOO記述三法

Fig. 1 A Realization of OO-Structured Description and Its Three OO Description Law

## 2.4 要素規則の設計

OONJで使える用語(語句)や記述の基本はNJであり、僅かな制約を付けられているだけである。したがってOOモデリング要素はOOSF<sup>5)</sup>で定義したOOモデリング要素種類表、要素構成表を拡張する設計を行う事で実現した。その要素種類リストを表2に示す。

## 2.5 構造化記述規則(構文規則)の設計

OONJの文法の一部である構文規則(本論文ではより実体に近い名称として構造化記述規則と呼ぶ)の大半は既にOOSF<sup>5)</sup>に定義されている。この規則の設計原理は前小節に示されたOOモデリング要素間の相互関係を規定することであり、それによりモデリングされた対象世界の構造化記述を実現することにある。つまりモデリングされた対象世界が持つ構造を可能な限り正確に表現できる汎用的な記述形式を規則という形に設計することになる。

現状のOOSFの構造化記述規則<sup>5)</sup>もその方針に沿って設計された。OOSFではその名前に示すように、対象世界の知識の記述に用いられて来たMinskyのフレーム理論<sup>7)</sup>を基礎として、フレーム以外にスロットやサブスロットの概念も導入し、それらの間の相互関係を付与することでOOに基づく構造化の方法を導入したものである。しかしOO構造化の方法は必ずしも充分ではなかった。例えば、サブスロットとスロットに関わるそれぞれの重要な一部分だけの規定に留まっている等の問題を残していた。そこで以下の様な点を含めて再検討し、拡張や改訂を行った。

(1) 次節で述べる多重・多様な構造化や、特徴ある構造を持つ文章の形式的構造化等。

(2) 全対象世界の必要な(集約、汎化)階層構造を表現できる形式への拡張。

この方針の下での設計結果を表3、表4に示した。

## 2.6 意味解釈規則の設計

意味解釈規則(本論文では意味解釈構造化規則と呼ぶ)とは、既に為された記述に対してどの様にその意味(即ち対応関係)を解釈すべきかを規定する構造化規則(つまり、記述と対象世界の対応する要素の両者の間の相互関係を付与する規則、を指す)である。つまり、記述と対象世界のどの要素(複数)がどの様に関係しているのかを意味という形で表現した規則である。

この意味解釈規則は「構造化」規則とあるように特定の名称/記述/値と対象世界との対応関係を定義する規則に他ならない。したがって、意味解釈規則の大半は記述構造化規則のちょうど逆方向の規則である筈で

ある。そこで、記述構造化規則の内、ちょうど逆方向に対応関係を解釈でき、そうすることで意味規則をも表現している記述構造化規則については変換対応記号である“: =”を変更して“: =:”とすることで左辺から右辺への記述構造化規則を表すと共に、右辺から左辺への意味構造化規則も兼用で表す様にした。これら以外の逆方向の対応関係を示す意味規則記述は別表で定めた。ただし、紙幅の制約のため掲載は省略した。

**2.7 多重構造化, 多様構造化, 複合文様式の設計**  
三つの新たな構造化記述様式をより詳細なあるいはより複雑な構造を容易に表現するために導入した。

**多様構造(化)**とは同一要素に対して「同一/同等内容の別表現」である複数の名称/記述/値を与え得(取り得), それを表現することを許す構造(化)とその表現方法を規定する。例えば、薬を薬品と言い換えるとか、「私は彼の父親である。」と「彼は私の息子である。」, 息子の立場から「彼は私の父親である。」という同一内容の別表現の例、式で言えば  $a \neq 0$ ,  $b \neq 0$  において  $a/b=2.0$  と  $b/a=0.5$ , 等である。

**多重構造(化)**とは同一要素に対して複数(多重性, 多義性を持たせた)の名称/記述/値)を取り得, それを表現することを許す構造(化)を指す。例えば, 同一人物が, 会社経営者, 誰かの父親, ある別の人物の友人, というような複数の立場を持つことを許し, かつその表現方法を規定する構造(化)である。これらの構造化の記述例は紙幅の都合上割愛した。

**複合文様式**とは, 数個から十数個程度の複数の単文から構成されている文の集まりであり, その文の集まりが特有な構造を持っている様式を指す。この特有な構造を明示形式に表現することで, より構造(化)を促進しよう試みた。例えば条件分岐文, 振舞い記述文の繰り返しや, 関係代名詞(副詞)節を持つ複文を複数の単文に変換したのものがある。

### 3. 記述例とその検討/考察

DSMC 数値風洞<sup>8)</sup>の記述例の一部を図 2, 図 3 に示した。現時点においては, 多種多様な対象世界についての記述例を書いては増やしている状況で, 総合的な評価をだす段階には至っていない。

### 4. 結論, 今後の課題, 展望

**結論: 充分詳細な記述力の実現**

本論文では記述言語 OONJ としての記述力を総合

的に評価する段階には至っていないが, OOSF の蓄積等も含めれば, 現状において必要な記述方法は OONJ の提案中に含まれていると推定している。

#### 今後の課題

**記述力と記述性の更なる向上:** 例えば複雑な数式, シミュレーションアルゴリズム, 対象世界全体の動きの追跡, 時間や空間の定義, 等を組み込む必要がある。今後記述例の対象世界を拡げて行く過程において現状には無い新たなあるいは拡張した仕様の必要性は常にあり得ると考えられる。

**記述環境の開発:** DU の記述力の向上のためには記述を支援する環境, 豊富な記述例が今後は是非必要である。既にその開発は進みつつある。

#### 展望:

DU の要求としては計算機の世界で実行できる記述に変換出来なければ OONJ の価値は半減する。そこで OEDJ<sup>6)</sup> を経てシミュレーションプログラムを(半)自動生成するシステムを開発中である。

### 参 考 文 献

- 1) 畠山正行, 加藤木和夫, 石井義之, オブジェクト指向記述日本語 OODJ とその記述環境, 情報処理学会論文誌, **41-9**, pp.2567-2581, (2000).
- 2) 畠山正行, 加藤木和夫, 上田賀一, オブジェクト指向自然日本語 OONJ の設計と評価, 第 130 回ソフトウェア工学研究会研究報告, 情報処理学会, **01-SE-130**, pp.153-160, (2001).
- 3) 加藤木和夫, 畠山正行, 上田賀一, 「オブジェクト指向プログラム設計記述言語 OOPD とその記述環境」, 情報処理学会論文誌, 第 43 巻, 第 5 号, pp.1401-1415, May, 2002.
- 4) 畠山正行, オブジェクト指向分析モデリングの明示形式化・詳細化・手順化, シミュレーション学会誌, **21-4**, pp.295-309, Dec., (2003).
- 5) 畠山正行, オブジェクト指向分析自然日本語構造化フレーム OOSF の設計と表現技法, シミュレーション学会誌, **22-4**, 2004 年 12 月号掲載予定.
- 6) 加藤木和夫, 畠山正行, 竹井健太郎, オブジェクト指向実行記述言語 OEDJ とそのトランスレータ, 第 145 回ソフトウェア工学研究会研究報告, 情報処理学会, 2004 年 8 月.
- 7) 長尾真, 知識と推論(岩波科学 ソフトウェア科学 14), 第 6 章, 岩波書店, (1988).
- 8) M. Hatakeyama, I. Kaneko, H. Uehara, "DSMC Analyses for Highly Complicated and Interactive Flow Based on the Object-Based Mechanism", The 19th International Symposium on Rarefied Gas Dynamics, **2**, 1175/1181, Oxford University Press (1994).

表 2 OO 分析モデリング/記述の要素種類リスト

\*

<b>fn04 個別の離散モデリング単位 (DMU)</b>	<b>fn3.4 変身振舞い</b>
<b>fn1 DMU(オブジェクト) 名</b>	fn3.4.1 発生/消滅 / fn3.4.2 変形/復元
fn1.0 要素種類間交差 DMU (注1)	fn3.4.3 コピー / fn3.4.4 破壊/復元
fn1.1 モノ/概念 DMU 名 / fn1.2 属性 DMU 名	fn3.4.5 分裂/融合 / fn3.4.6 名称や数の変化
fn1.3 振舞い DMU 名 / fn1.4 相互関係 DMU 名	fn3.4.7 変異・遷移 / fn3.4.8 その他諸々の変身
fn1.5 対象世界共通要素 DMU 名	<b>fn3.5 複合 (抽象) 構造化振舞い (CSB と略)</b>
<b>fn2 属性</b>	fn3.5.1 状態表現 CSB / fn3.5.2 遷移表現 CSB
fn2.0 要素種類間交差属性 (注1)	fn3.5.3 初期化 CSB / fn3.5.4 変身 CSB
fn2.1 振舞い参照属性 / fn2.2 汎化/特化属性	fn3.5.5 相互関連属性活性化 CSB
fn2.2 汎化/特化属性	<b>fn4 相互関係</b>
fn2.3 集約/被集約属性 / fn2.4 一般関連属性	fn4.0 要素種類間交差相互関係 (注1)
fn2.4 一般関連属性	fn4.1 相互作用伝達 (mp)
fn2.5 相互作用 (伝達付置) 属性	fn4.2 汎化/特化 / fn4.3 集約/被集約
fn2.6 制約属性	fn4.4 グループ化 (カプセル化を含む.)
fn2.7 アクセス属性 (情報隠蔽を含む)	fn4.5 再帰展開/縮約 / fn4.6 離散化/修復再構成
fn2.7.1 DMU アクセス属性	fn4.7 詳細化/簡易化
fn2.7.2 属性アクセス属性	fn4.8 時間的/空間的相互関係
fn2.7.3 振舞いアクセス属性 (振舞い発起条件)	fn4.8.1 連接 / fn4.8.2 反復 / fn4.8.3 分岐
fn2.7.4 相互関連アクセス属性	fn4.8.4 逐次 / fn4.8.5 並列/並行
fn2.8 所有 (所属)/所用属性	fn4.8.6 分散 / fn4.8.7 接着/固着
fn2.8.1 共通属性 / fn2.8.2 共用属性	fn4.8.8 その他の時間的/空間的相互関係
fn2.8.3 共有/共属属性	fn4.9 所有 (所属)/所用
fn2.9 複合 (抽象) 構造化属性 (CSA と略)	fn4.9.1 共通/特有 / fn4.9.2 共用/専用
fn2.9.1 状態 CSA / fn2.9.2 遷移 CSA	fn4.9.3 共有・共属/専有・専属
fn2.9.3 初期化 CSA / fn2.9.4 変身後 CSA	fn4.10 変身前後関係
fn2.9.5 相互作用伝達付置 CSA	fn4.10.3 破壊/復元 / fn4.10.4 分裂/融合
<b>fn3 振舞い</b>	fn4.10.5 名称や数の変化 / fn4.10.6 変異・遷移
fn3.1 内部振舞い	fn4.11 その他諸々の相互関係
fn3.1.1 一般内部振舞い / fn3.1.2 内部変身	<b>fn05 対象世界共通要素</b>
fn3.1.3 振舞い開始 / fn3.1.4 振舞い停止	fn5.1 離散時間定義
fn3.1.5 制約属性有効化 (活性化) 実行振舞い	fn5.1.1 離散時間単位 / fn5.1.2 離散時刻
fn3.1.6 相互関連有効化 (活性化) 実行振舞い	fn5.2 離散空間定義
fn3.2 相互作用伝達 (message passing, mp と略)	fn5.2.1 離散空間単位 / fn5.2.2 離散空間
fn3.2.1 mp 発信 (直接目的で fn1.1 モノを伝達付置)	fn5.3 初期状況記述
fn3.2.2 mp 発信 (情報のみを伝達付置)	fn5.3.1 時間/空間初期化記述
fn3.2.3 mp 受信 (直接目的で fn1.1 モノを伝達付置)	fn5.3.2 属性初期化記述
fn3.2.4 mp 受信 (情報のみを伝達付置)	fn5.4 対象世界駆動シナリオ
fn3.3 相互作用振舞い (=fn3.1 + fn3.2)	fn5.5 対象世界共有要素 / fn5.6 注釈
fn3.3.1 内部振舞い + mp 発信	
fn3.3.2 mp 受信 + 内部振舞い	

表 2-2 全対象世界の記述階層における要素種類リスト

fn00 全対象世界 / fn01 全対象世界代表 DMU / fn02 複 DMU 集団 / fn03 複 DMU 集団代表 DMU / fn04 DMU / fn05 対象世界共通要素 DMU / fn06 複要素集団 / fn07 複要素集団代表要素 / fn08 単要素 / fn09 NJ 複単位文 / fn10 NJ 単位文

表 3 OONJ 記述構造化規則 (構造化法) (1/2)  
Table 3 OONJ Description Structured Rule (1/2)

(注) < ><sup>+</sup>n は n 個以上 (n=1 の場合は 1 は省略可), < ><sup>\*</sup> は 0 個以上を表す。<< >> は多様/多重要素で複数の要素を収容可。

(1) < fn00 全対象世界 > :=:< fn01 全対象世界代表フレーム >< fn02 複フレーム集団 > * < fn04 DMU フレーム > + < fn05 全世界共通要素フレーム >	(注) フレーム (DMU) は各々別の独立した矩形の外枠線として記述される。
(2) < fn02 複フレーム集団 >:=:< 全長階層表記線 >< fn03 複フレーム集団代表フレーム >< fn04 DMU フレーム > + <sup>2</sup>	
(3) < fn04 DMU フレーム > :=:< fn08 フレームヘッダスロット >< lf > (( < fn08 単位スロット >< lf > ) + ( < fn06 複スロット集団 >< lf > ) * )	
(4) < fn08 フレームヘッダスロット > :=:< フレーム一意特定識別子 >< sp > ( < fn1.x > < sp > < フレーム名 > ) [ < rt > < fn4.x 相互関係 > ] < lf >	(注) 短冊形枠線を付けない例外スロット
(5) < フレーム一意特定識別子 >:=:<< フレーム番号 >>	
(6) < フレーム名 >:=:<<< DMU 名 >>>	
(7) < fn01 全対象世界代表フレーム >:=: "00 fn1.1" <<< 対象世界名 >>> [ < rt > < fn4.x 相互関係 > ] < lf > ( < fn08 対象世界代表フレーム単位スロット >< lf > ) +	
(8) < fn05 全世界共通要素フレーム >:=:<< フレーム連番号 >> "fn5.x" <<< 全世界共通要素名 >>> [ < rt > < fn4.x 相互関係 > ] < lf > ( < fn08 全世界共通要素フレーム単位スロット >< lf > ) +	
(9) < fn03 複フレーム集団代表フレーム >:=: "nn fn1.x" <<< 複フレーム集団 (抽象) 名 >>> [ < rt > < fn4.x 相互関係 > ] < lf > ( < fn08 複フレーム集団代表フレーム単位スロット >< lf > ) +	
(10) < fn06 複スロット集団 >:=:< 全長階層表記線 >< fn08 複スロット集団ヘッダスロット >< lf > ( < fn08 単位スロット >< lf > ) +	
(11) < fn08 複スロット集団ヘッダスロット >:=:< 複スロット集団一意特定識別子 >< sp > ( < fnx.x > <<< 複スロット集団名 >>> ) [ < rt > < fn4.x 相互関係 > ] < lf >	
(12) < 複スロット集団一意特定識別子 >:=:< スロット番号 >< ファセット記号 >	
(13) < fn4.x 相互関係 >:=:< 相互関係種別 >< 関係相手名リスト >	(注 3)
(14) < 相互関係種別 >:=:( " > " < 相互関係名 > " > " )   ( " < " < 相互関係名 > ( " < "   " > " ) )	(注 1)
(15) < 相互関係名 >:=:<<< ( " 発信 (受信)"   " 汎化 (特化)"   " 集約 (被集約)"   " グループ化"   " 再帰展開 (縮約)"   " 離散化 (修復)"   " 詳細化"   " 接続"   " 反復"   " 分岐"   " 逐次"   " 分散"   " 並列 (並行)"   " 固定 (固着)"   " 共有 (専有)"   " 変身 (還元, 分裂, 融合, 破壊, 修理, コピー)"   " 引用"   " 説明"   " 補足"   " その他諸々の相互関係" ) + >>>	
(16) < 関係相手名リスト >:=:( < フレーム一意特定識別子 > << フレーム名 >> [ < スロット一意特定識別子 > ] ) +	
(17) < rt >:=:< 右詰め >	(注) default は左詰めである。
(18) < lf >:=:< 次の行またはスロットに移り, その左先頭へ戻ることを示す記号 >	
(19) < sp >:=:< 任意一定幅空白 >	
(20) < fn08 単位スロット >:=:<< 短冊形のスロット実線外枠線 >> (注) 各フレームのスロット長は同一に揃える。 【( < fn09 代表スロット内部記述 > ) + ( < 破線の行全幅スロット内部区切り線 > < fn09 多様/多重スロット内部記述 > ) *】	
(21) < fn08 対象世界代表フレーム単位スロット > :=:< スロット一意特定識別子 > ( " fn4.x " < 対象世界構成フレーム名 > [ < rt > < fn4.x 相互関係 > ] < lf > )	
(22) < fn08 複フレーム集団代表フレーム単位スロット > :=:< スロット一意特定識別子 > ( " fn4.x " < 複フレーム集団構成フレーム名 > [ < rt > < fn4.x 相互関係 > ] < lf > )	
(23) < fn09 代表スロット内部記述 >:=:< fn10 代表スロット総称文 >< lf >< fn09 代表スロット内サブスロット記述 > *	
(24) < fn10 代表スロット総称文 > :=:< 代表スロット一意特定識別子 >< sp >< 代表スロット総称名 >< sp ><< fn10 NJ 単位文 >> [ < rt > < fn4.x 相互関係 > ]	

(注 1) 二重引用符 "" で囲まれた語句や記号 (NJ 単語, <, >, fn, |, ||, ||| 等) は語句や記号をそのまま記すことを意味する。記号 (:=) は意味解釈構造化規則 6.2 節参照。ああああああ。

(注 2) スロット内部の単位文の構造またはその内容を代表して表記する。なお, 単一と接続とは default では省略して良い。

表 4 OONJ 記述構造化規則 (構造化法) (2/2)  
Table 4 OONJ Description Structured Rule (2/2)

(注) < > +<sup>n</sup> は n 個以上 (n=1 の場合は 1 は省略可), < > \* は 0 個以上を表す. << >> は多様/多重性要素で複数の要素を収容可.

(25) <代表スロット一意特定識別子> ::= <スロット番号>   (“<スロット番号> “-” <行番号> ”) <ファセット記号>
(26) <ファセット記号> ::= (“fn   vfn   mfn”) <ファセット番号>   “pfn” <複合文様式ファセット番号> [ “/” <文型番号> ] < sp >
(27) <代表スロット総称名> ::= <複合文様式名>   <代表スロット代表名>
(28) < fn09 代表スロット内サブスロット記述> ::= ([ “-” <行番号> ” ]   < sp > ) < sp > <ファセット記号> <インデント> <全長階層表記線> ( < fn10 NJ 単位文>   < fn09 NJ 複合文> ) [ < rt > < fn4.x 相互関係> ] < LF > < fn09 代表スロット内サブスロット記述> *
(29) < fn09 多様/多重スロット内部記述> ::= < fn10 多様/多重スロット総称文> ( < lf > <スロット全幅区切り破線> < fn09 多様/多重スロット内サブスロット記述> ) *
(30) < fn10 多様/多重スロット総称文> ::= <多様/多重スロット一意特定識別子> < sp > <多様/多重スロット総称名> < sp > ( < fn10 NJ 単位文>   < fn09 NJ 記述> ) [ < rt > < fn4.x 相互関係> ]
(31) <多様/多重スロット一意特定識別子> ::= <多様/多重スロット番号> “-” <多様/多重サブスロット番号>
(32) <多様/多重スロット総称名> ::= <複合文様式名>   <多様/多重スロット代表名>
(33) < fn09 多様/多重スロット内サブスロット記述> ::= <多様/多重スロット “行” 識別子> < sp > < sp > <インデント> <全長階層表記線> ( < fn10 NJ 単位文>   < fn09 NJ 複合文> ) [ < rt > < fn4.x 相互関係> ] < LF > < fn09 多様/多重スロット内サブスロット記述> *
(34) <多様/多重スロット “行” 識別子> ::= “-” <行番号>
(35) <多様/多重スロット番号> ::= << 2 以上の整数昇順連番数 >>
(36) 【 】 : << 短冊形のスロット外枠線の内側に収容する記述であることを示す記号 >>
(37) <複合文様式名> (注 2) ::= << (“単一”   “接続”   “反復”   “分岐”   “逐次”   “分散”   “並列”   “共有”   “総称” (注 3)   “相互作用”   “注釈”   “引用”   “説明”   “補足”   “変身”   “初期化”   “シナリオ”   “時間定義”   “空間定義”   “その他”) + >>>
(38) <スロット代表名> ::= << (“見出し   タイトル   柱書き   表題   主題   その他代表名”) >>>
(39) <スロット一意特定識別子> ::= <スロット特定記号> < sp > <ファセット記号>
(40) <スロット特定記号> ::= <<スロット番号>>
(41) <文型番号> ::= << 1   2   3   4   5 >>
(42) < NJ 記述> ::= << DU が理解できる全ての NJ 文または NJ 文の列で, 表?? に示す制約を受けないものも許す. >> (注 4)
(43) <全長階層表記線> ::= << “ ” (集約) >>   << “  ” (特化) >>   << “   ” (グループ化) >>
(44) <インデント> ::= << 各 DU 定義の任意一定幅空白 >>
(45) < LF > ::= << 改行して上の行の <全長階層表記線> の位置まで左へ戻ることを示す記号 >>
(46) < mfn1.0.1 多様/多重 DMU “名” > ::= (vfn1.x   mfn1.x) (“<代表 DMU 名> (“,” <<多様/多重 DMU 名>>)+ “”)
(47) < mfn2.0.2 多様/多重属性 > ::= (vfn2.x   mfn2.x) (“<代表属性名> (“,” <<多様/多重属性名>>)+ “”)
(48) < mfn4.0.4 多様/多重相互関係 > ::= “>>”   << (“<代表相互関係名> (“,” <<多様/多重相互関係名>>)* “”) >>   << “<相互関係先> +

(注 3) <相互関係相手名リスト> : 相互関係が一对一对応でない場合には, 一般原則としては,

(1) 相互関連が一对一の場合は関わる離散単位要素間 (スロット間) の相互関連を一对一で両スロット内を書く.

(2) 一对多 (n 個) の時には一要素の側に多要素の側の要素リストを n 個分書くことを原則とする. 一对一の相互関係を n 個分記述するのも可能であり許される.

(3) 多対多 (m × n) の時には適切と判断された m 個の側の各要素にもう一方の側の要素リストを n 個分書くことを原則とする. これ以上に複雑な組み合わせの場合は別に fn1.4.x 相互関係のフレームを別途定義し, そのフレームと各 m 個のフレームとが相互関連を結ぶ形式にする. これ以外であれば個々に改めて検討する.

(注 4) NJ 文や NJ 文列はそれらが長くなる場合には他の行に合わせて文を適切に折り返す. その時, 折り返し直前と折り返し直後に行継続記号として (\*\*) を入れて示す.

**7 fn1.1 試験飛行体 X-30**      <<集約<< 1 数値風洞 [4]

1: (注釈) 試験飛行体の実体は多数の三角形固体壁の集約。 分子の衝突は直接には三角形への相互作用である。
2: fn2.3 試験飛行体は 427 枚の三角形固体壁から構成される。 >>集約>> 36 三角形固体壁 fn2.6 三角形は隙間無く接続し三次元機体形状を形成する。
3: fn2.2 各三角形は気体分子に対し、散乱反射境界壁、鏡面 反射境界壁、Nocilla モデル境界壁、のどれかである。 >>汎化>> 12 境界平面壁 [2]
4: fn2.8.3 共有 (fn1.1 セル, fn2.5(fn2.1(位置 $x$ , 速度 $V'$ )))
5: fn3.2 $Ar_i$ のデータを受ける。 <<受信<< 8 気体分子 $Ar_i$ [4]   fn2.5(fn2.1(属性 $A_3$ , 位置 $x'$ , 速度 $V'$ )) (注 1)
6: fn3.1 分子との衝突の可否と衝突する三角形の特定を行う。 >>再帰展開>> 46 分子・面衝突を行う。
7: fn3.2 分子 $Ar_i$ の衝突前データを送る。 >>発信>> 13Nocilla モデル境界壁 [4]   fn2.5(fn1.1 分子 $Ar_i$ , fn2.1(位置 $x$ , 速度 $V$ ))
8: fn3.2 分子 $Ar_i$ の衝突後データを受ける。 <<受信<< 13Nocilla モデル境界壁 [5]   fn2.5(fn1.1 分子 $Ar_i$ , fn2.1(位置 $x$ , 速度 $V'$ ))
9:fn3.2(並列 1) 衝突後データを送る。 >>発信>> 8 気体分子 $Ar_i$ [5]   fn2.5(fn1.1 分子 $Ar_i$ , fn2.1(位置 $x$ , 速度 $V'$ )) fn3.1(並列 2) 分子反射に依る自身への影響を計算する。   fn2.1(位置 $x$ , 速度 $V, V'$ )

**13 fn1.1Nocilla モデル境界壁** >>汎化>> 29 流体・固体壁 [2]

1: (注釈) 試験飛行体の三角形平面壁の特性モデルである。
2: (注釈) 衝突分子は本モデル反射則に従って反射される。
3: fn2.8.3 共有 (fn1.1 セル, fn2.5(fn2.1(位置 $x$ , 速度 $V'$ )))
4: (注釈 : fn3.3) 気体分子と衝突する。   fn3.2 衝突前のデータを受ける。 <<受信<< 7 試験飛行体 [7]   fn2.5(fn1.1 分子 $Ar_i$ , fn2.1(位置 $x$ , 速度 $V$ ))   fn3.1 分子の反射後の速度データを計算する。 >>再帰展開>> 46 分子・面衝突を行う。   fn2.1(属性 $A_{12}$ , 位置 $x$ , 速度 $V$ , 属性 $A_{13}$ )   fn2.6 散乱反射が $\alpha$ , 鏡面反射が $1.0 - \alpha$ である。
5: fn3.2 分子の反射後速度を送る。 >>発信>> 7 試験飛行体 [8]   fn2.5(fn1.1 分子 $Ar_i$ , fn2.1(位置 $x$ , 速度 $V'$ ))   fn2.6 分子の反射位置座標は入射位置座標と同一。

(注 1) 位置  $x=(x,y,z)$ , 分子速度  $V=(V_x, V_y, V_z)$

図 2 数値風洞の OOSF 記述例の一部 (1)  
Fig. 2 A Description Example:: DSMC Numerical Wind Tunnel (1)

**25 fn1.1 セル**      <<集約<< 5 風洞本体 [9]

1: (注釈) セルは $60 * 60 * 80 = 28.8$ 万個生成される。
2: fn2.3 気体分子を集約・管理する。 >>集約>> 8 気体分子 $Ar_i$
3: fn2.4 試験飛行体に接する。 >>隣接>> 7 試験飛行体 X-30
4: fn2.8.3 共有 (fn1.1 セル, fn2.5(位置 $x$ , 速度 $V'$ )))
5: fn3.1 分子を二個選び衝突ペア分子 $Ar_i$ と $Ar_j$ に指定する。   fn1.1( $Ar_i, Ar_j$ ), fn2.1( $\Delta x = \Delta y = \Delta z$ , 属性 $A_4$ ) fn3.2 衝突パラメータを送る。 >>発信>> 8( $Ar_i, Ar_j$ )[3]   fn2.5(衝突パラメータ $b_{ij}, c_{ij}$ , 属性 $A_8$ )
6: (注釈:fn3.3) 衝突後の分子を受ける。   fn3.2 衝突後のデータを受ける。 <<受信<< 8( $Ar_i, Ar_j$ )[3] と [5]   fn2.5(fn2.1(位置 $x$ , 速度 $V'$ , 所属セル番号))   fn3.1 $Ar_i$ と $Ar_j$ のセル内での新しい空間位置を変更する。   fn3.1 その新しい空間位置がセル内部ならば、その分子を記憶する。   fn3.2 その位置が隣接セルならば、その隣接セルと両分子にデータ を送る。 >>発信>> (25 セル [7], 8 気体分子 ( $Ar_i, Ar_j$ )[6])   fn2.5(fn2.1 (セル番号, 位置 $x'$ , 速度 $V'$ )))
7: fn3.2 分子が着いたら記憶を更新する。 <<受信<< (25 セル [6], 8 気体分子 $Ar_i$ [4])   fn2.5(fn2.1(セル番号, 位置 $x'$ , 速度 $V'$ )))

**8 fn1.1 気体分子 ( $Ar_i, Ar_j$ )** <<集約<< 25 セル [2]

1: (注釈) 約 288 万個生成され、各セル平均 10 個配置される。
2: fn2.8.3 共有 (fn1.1 セル番号, fn2.1(位置 $x$ , 速度 $V'$ )))
3: (注釈:fn3.3) $Ar_i$ は同一セル内の別分子 $Ar_j$ と衝突する。   fn3.2 (並列 1) データを送る。 >>発信>> 8 気体分子 $Ar_j$ [3]   fn2.5(fn2.1(位置 $x_i$ , 速度 $V_i$ , 所属セル番号))   fn3.2 (並列 2) データを受ける。 <<受信<< 8 気体分子 $Ar_j$ [3]   fn2.5(fn2.1(位置 $x_j$ , 速度 $V_j$ , 所属セル番号))   fn3.1 自身と衝突ペアとのデータから衝突後速度を計算する。 >>再帰展開>> 45 分子間衝突を行う。   fn2.1(速度 ( $V_i, V_j$ ), 衝突パラメータ ( $b_{ij}, c_{ij}$ ))   fn3.2 衝突後データを送る。 >>発信>> 25 セル [6]   fn2.5(fn2.1(位置 $x$ , 速度 $V'$ , 所属セル番号))
4: fn3.2 $Ar_i$ の衝突後の位置と速度を送る。 >>発信>> (25 セル [7], 7 試験飛行体 X-30[5])   fn2.5(fn2.1(属性 $A_3$ , 位置 $x'$ , 速度 $V'$ )))
5: fn3.2 衝突後データを受ける。 <<受信<< 7 試験飛行体 X-30[9]   fn2.5(fn1.1 分子 $Ar_i$ , fn2.1(位置 $x$ , 速度 $V'$ )) fn3.1 飛行量 $\Delta x$ と飛行後の変更位置座標 $x$ を計算する。 fn3.2 自身の新しい位置座標を送る。 >>発信>> 25 セル [6]   fn2.5(fn2.1(位置 $x$ , 速度 $V'$ , 所属セル番号))
6: fn3.2 所属セル番号を受け取る。 <<受信<< 25 セル [6]   fn2.5(fn2.1 (セル番号, 位置 $x'$ , 速度 $V'$ )))

図 3 数値風洞の OOSF 記述例の一部 (2)  
Fig. 3 A Description Example:: DSMC Numerical Wind Tunnel (2)