

simpleOONJ:ドメインユーザでも使いこなせるオブジェクト指向技術の提唱

島 山 正 行†

要約: 非情報系の科学技術諸分野の応用ドメインの専門家ユーザ (以降ドメインユーザ, DU) がオブジェクト指向 (OO と略) で分析し, 自然日本語 (NJ と略) で記述する記述言語として OONJ が使われている。しかし DU から OONJ の記述力の強さや詳細さ, NJ 記述のやさしさに対する高い評価と共に, 記述規則の数が多くて使い難いと評価された。そこで離散 (化) した単位とそれらの構造化だけを基礎にし DU にも使いこなせる簡潔な OO 技術, すなわち simpleOO 技術を開発した。それを基礎に簡潔に OONJ 記述を実現するためにその記述規則を四段階に分けた simpleOONJ (oonj0, 1, 2, 3 と呼ぶ) に再構成設計し, 記述エディタも開発した。

記述実験や記述例の検討の結果, oonj1 では狙い通り DU に充分に簡潔に書けること, 記述エディタで更に容易に使えることが分かり, 有効性が検証された。oonj2 でもサブスロットの複数 NJ 単位の構造化記述法が確立し, 記述エディタの有効性も検証された。以上から simpleOONJ と記述エディタは DU にとって有用な技術であること, simpleOO 技術が妥当であることが立証できた。

simpleOONJ : A proposal of simple object-oriented technique for Domain Users

MASAYUKI HATAKEYAMA†

Abstract: We have already proposed a Natural Japanese(NJ)-based, Object-oriented(OO) description language OONJ for the Domain Users (DUs) who are the experts of a certain professional domain except the computer domain. The OONJ description users have highly evaluated the powerful/detailed description power, and the simpleness of the NJ descriptions. They have also pointed out that the description rules of OONJ are too many and complex. To solve this problem, we have newly developed the "simpleOO" technology, and then designed a new description language "simpleOONJ" with the four stage sub language series oonj0, 1, 2, 3 with their description editors. Some description experiments/examples have shown that the oonj1 is simple enough for DUs, and that the description editor has provided far simple environments than that without it. The oonj2 with its description editor has shown its availabilities of the structurization functions among the NJ plural statements in the sub slots. These results have shown that the simpleOONJ are far simpler ones than the original OONJ. As for the conclusion, the simpleOONJ based on the simpleOO technology have successfully been developed with just the primary stage description editors for the sake of DUs.

1. はじめに

著者は既出の論文^{1),2)}において自然日本語 (NJ) 記述を用いてオブジェクト指向 (以降, OO と略) に基づいてモデリングし記述する方法として OOSF (Object-oriented Structured Frame) を提案した。またこの OOSF を核とする OO 分析記述言語 OONJ (Object Oriented Natural Japanese)^{3),4)} を設計/開発した。OONJ は NJ を用いた OO 構造化記述の方法として次第に広がりを見せつつある。

強い記述力と記述の容易性の trade off

しかし記述例が増えるにつれ表現法/記述力の不足

が出てきた。そこで利用想定ユーザであるドメインユーザ (以降, DU)^{1),2)} の要求に沿って記述規則を増やして記述力強化を図り, 記述例を作って検証した結果, 確かに狙い通りに記述力の“強さ”や“詳細さ”は充分良好に実現した⁴⁾。

しかしその反面の問題として記述規則が多く, 表現形式も煩雑で使い難くなった⁴⁾。これらは OONJ の設計/開発の当初から予測されていた。そこで OONJ では, まずは専門家としての DU が要求する「記述力/表現力」の実現だけを狙いとした。それと対照的に本論文では最小限の負担で容易に OO 記述が出来ることを狙いとし, その実現を試みる。

OO は本来は簡潔な技術

本研究を立ち上げたもう一つの動機でもあり発想の原点でもあるのは, 本来的に OO 技術は, 例えば

† 茨城大学工学部情報工学科

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

表 1 simpleOONJ の構築戦略

* Table 1 Constitution strategy for simpleOONJ

<p>(I) OO 技術を新たな視点で根底から再構築</p> <p>(1) NJ ベースの OONJ の言語仕様自体が DU-oriented な simpleOO 技術の一部.</p> <p>(2) OO 構築原理: 離散化, 構造化, 明示形式化</p> <p>(3) オブジェクトベース (OB) に基づくモデリング/記述パラダイムの全面採用</p> <p>(4) OOSF 構造化記述モデルの再構成が核心部</p> <p>(5) OB 一貫相似性過程の方法論が全体背景に.</p>
<p>(II) OONJ を組み直し/階層構造化/再構成</p> <p>(1) 言語仕様全てを階層構造化→ oonj1/2/3/4 (設計や実装段階も含めて同様に階層構造化する予定. 一貫相似性過程の実現にも貢献)</p> <p>(2) oonj0 も NJ 記述から OO 記述への最初の移行段階の記述法として構築する (3.3 節).</p>
<p>(III) 記述環境ベースの DU 利用 GUI の構築</p> <p>(1) 記述環境に (I)(II) の全ての項目を組み込む.</p> <p>(2) 直接目標は「記述規則学習フリー」である.</p>
<p>(IV) DU の OONJ 学習過程にも特別な工夫</p> <p>(1) oonj0~4 に対応した DU 向けの習得方法が必要. その1つは段階的 (階層的) 習得方法.</p>

複雑な事象を抽象化して扱ったりすることで簡潔に扱うことを得意とする技術であることから分かるように, DU の思考様式にも適合した簡潔に扱える技術である⁵⁾というのが著者の考えである. ある程度の練度のユーザであれば, 自在に使えて多くのメリットを享受できる筈の技術である.

現在の OO ソフトウェア工学との対比

しかし現代の複雑化した OO 技術, 特に OO ソフトウェア工学 (以降, OOSE) の技術はソフトウェア開発の専門家向けにその手法やシステムが開発されてきたために DU の利用が難しい状況にある. そこで本来の簡潔な特性を保持した OO 技術を再構築し, 非常に幅広い分野の莫大な数の DU が OO 技術を使える様にしたと考えていたのである. 以上の二つが simpleOO 技術⁵⁾の開発動機である.

simpleOO 技術の提案

本論文では具体技術として, 複雑化した OO 分析記述言語 OONJ の理解性と記述性を大きく向上させ, DU にとってより簡潔で良好な記述性を実現するために, 四つのサブ記述言語からなる記述言語系 simpleOONJ⁵⁾を設計/実装した. 表 1 にはその simpleOONJ の構築戦略を示す. 更にはその簡潔さを更に徹底させ, 記述容易性を向上させ, DU の負担を軽くさせるため, 専

表 2 simpleOONJ の四つのサブ記述言語の構成

* Table 2 Constitutions of 4 sub-description language *

oonj0: まずは純粹 NJ から原始的 OO 記述へ

- (1) まず最初に“とにかく”記述する際 OO 風書き分け法. DU 自身の常用エディタを利用.
- (2) オブジェクト名を決め個々別々に記述する (フレームの概念). 振舞いを個々別々に分離し各オブジェクトに書く. (スロットの概念)
- (3) オブジェクト間の汎化と集約の相互関連を見出して NJ で記述する. mp はまだ不要.

oonj1: 必須最小限構成の記述言語

- (1) 離散的記述単位であるフレームと, その内部の離散要素としてのスロットを記述する.
- (2) 構造化の仕組みである相互関係を記述
- (3) 上記だけで OO 構築原理 (1) と (2) や 2.6 節の条件の (A) と (B) を満たしている.
- (4) NJ 記述は何らの制約無しで良いとした.
- (5) 付置属性は記述不要とした.
- (6) 相互関連図^{1),2)} と相互作用図¹⁾ を書く.

oonj2: サブスロットの構造化と属性記述

- (1) サブスロットの詳細化とその構造化を行う.
- (1) NJ 記述の NJ 単位文化と構造化を行う.
- (2) mp と内部振舞い文の書き分けが必要. 付置属性 (fn2.1 振舞い参照属性, fn2.2 mp 付置属性) を区別して記述する.
- (3) スロット内部での集約構造化記述法の導入
- (4) よく用いられる複合文様式の典型である反復/条件分岐の各複合文を取り入れた.

oonj3: 実行トレース可能水準の最小限記述

- (1) サブスロット記述に任意の深さと詳細さの記述を実現する集約構造化 (入れ子) を導入.
- (2) 複合文, 多様構造定義等々を多数導入する. 詳細で豊かな表現力の向上を実現する.
- (3) 式や記号も使い定量的な記述を完成させる.
- (4) 境界/初期条件の“実値”記述, シナリオ等の対象世界状況記述定義を取り入れる. これにより初めて紙上トレース実行可能になる.

oonj4 = いわゆる full 構成の OONJ

- (1) 記述規則上は OONJ とほぼ同じであるが, oonj3 に構造記述規則と複合文様式やサブスロットの詳細記法等で, 多様な表現法を実現.
- (2) 抽象階層 (汎化, 集約) の概念とその記述上の扱いの違いを体系的に導入.
- (3) 特定イベント展開, オブジェクトの変身現象, 一般適用規則の特殊な一部規則を含む.
- (4) 設計へ渡すために記述完成度の高さが必要.

(注) oonj0~oonj4 は完全上位互換任様で設計された.

用記述エディタを主体とする記述支援環境を構築した。

以上の構成戦略において、記述力は OONJ で実現し、理解や記述の容易さは simpleOONJ が受け持つ、という役割分担を設定した。つまり simpleOONJ と記述言語 OONJ は相補的な役割を担って機能する車の両輪である。

設計の結果として、表 2 に表 1 の (II) に沿って四段階に分けられた simpleOONJ のサブ記述言語 oonj0 ~4 の設計仕様を示す。表 3 には oonj1 および oonj2 の記述規則を示す。それらのサブ記述言語仕様に沿った記述支援エディタも設計/実装され、既に稼働している。oonj1 記述エディタを図 1 に示す。紙幅の制限のため、記述例の提示は割愛した。

2. simpleOONJ の実装/評価/考察

2.1 四つのサブ記述言語の実装/評価/考察

(1)oonj1 から oonj4 に至る四つのサブ記述言語は完全上位互換性を持たせ、表 2 の構成に従って段階分けを行い、それぞれに対して記述規則 (文法) を設計した。また、DU の思考様式に合わせるために、

- (1)OO モデリングした対象世界の階層構成
- (2)階層分けされた OOSF の集約階層構成
- (3)DU のイメージ中の詳細さの区分構成

の三つが十分重なるように注意深く階層分けした。更には階層構成になっても OOSF および OONJ の構成や構造を崩さない様に以下の条件を課した。

- (A) DU にとって自然な段階的拡張構成にし、重層的な理解と記述になるように完全上位互換の段階区分にすること。
- (B) 各段階毎個々に完結した OO 構造を持つこと。最後には OONJ に一致すべきこと。

以上の設計および実装 (ここでは、記述規則を定めることを指す。) の結果、四つの記述言語分けは OO パラダイムからも、DU のモデリング/記述の思考様式に適応した適切な区分設計であることが分かった。

2.2 記述環境/記述エディタの実装/評価/考察

数十名規模の記述実験、記述例の作成を行った結果、以下のような結果が得られた。

(1) 言語を 4 段階に区分して学習するので、全規則を一時に学習するよりは心理的な側面と言語の学習負荷の両面について大幅に少なく出来た。

(2)oonj1, 2 記述エディタは、記述に至る道筋が明示形式化されている故もあって学習の負担が小さく、DU に十分にやさしい記述支援機能を提供している。

(3) 記述エディタではほとんどすべての記述者が構文規則を参照することなくむしろ記述例を参照しながら記述を行った。この事実も oonj1, 2 が簡潔な記述言語の証であると言って良いであろう。

(4)oonj3 エディタは多様な機能を利用する DU による記述作業例が未だ少なく、結論は出ていない。oonj3 の仕様の基本的な機能と使い易さは備えている。

以上から、oonj1-2 エディタは DU が使いこなせる有用なツールを提供したことを立証した。

3. 結論と今後の課題

結論: OO 記述言語 OONJ の DU 向けの簡潔な記述法の開発と、DU 向けの簡潔な OO 技術の確立/提唱を狙いとして、OO 記述言語系 simpleOONJ とその記述支援環境を開発した。DU が段階的に skill up を図れるように構造記述規則を四段階に分け、oonj0, 1, 2, 3 とし、それに対応した記述エディタを開発した。

記述実験や多くの記述例等々から、simpleOONJ (oonj0, 1, 2) が DU にとっては十分に有効なツールであり、OO 分析記述を容易にする記述環境付き言語であり、記述の“簡潔さ”、記述力や記述性、DU の使い勝手などの面において優れて有用であることを、また simpleOO 技術の基礎も構築したことを立証した。今後の課題本技術の設計/実装への適用、四つのサブ言語とその記述環境の改良と統合化、更に多くの記述例の作成と記述実験 (普及) 活動が挙げられる。

参 考 文 献

- 1) 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, Vol.21, No.4, pp.295-309,(2003).
- 2) 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, Vol.22, No.4, pp.50-67(2004).
- 3) 畠山正行, 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, 第 145 回ソフトウェア工学研究会報告,2004-SE-145, pp.53-60,(2004).
- 4) 畠山正行, 松本賢人, 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, 情報処理学会第 102 回 HPC 研究会報告, 2005-HPC-102, pp.13-22, (2005).
- 5) 畠山正行, 池田武徳, 生井沢和也, 松本賢人, ドメインユーザにもやさしいオブジェクト指向自然日本語記述言語 simpleOONJ とその記述環境, 情報処理学会第 102 回 HPC 研究会報告, 2005-HPC-102, pp.23-26, (2005).

表 3 simpleOONJ (oonj1, 2) の構造記述規則
Table 3 Structured description rules for simpleOONJ (oonj1, 2)

○	1 <対象世界>:=:<フレーム> (<lf> + ³ <フレーム>)+
○	2 <フレーム>:=:<フレームヘッダスロット> (<lf><スロット>)+
○	3 <フレームヘッダスロット>:=:<フレーム番号><sp>"fn1"<sp><フレーム名> [<rt><相互関係>]
○	4 <フレーム名>:=:<オブジェクト名>
○	5 <相互関係>:=:<相互関係記号><相互関係相手名>
○	6 <相互関係記号>:=:(">"<相互関係名>">") ("<"<相互関係名> ("<" ">"))
○	7 <相互関係名>:=:"mp" "汎化" "特化" "集約" "被集約" "詳細化" <<その他>>
○	8 <相互関係相手名>:=:<フレーム番号><フレーム名> ["["<スロット番号>"]"]
△	9 <スロット>:=:<短冊形のスロット実線外枠線> (<スロット総称文> <分岐/反復文> <注釈>) (<lf> (<サブスロット> <分岐/反復文> <注釈>))*
	10 <スロット総称文>:=:<スロット番号><sp><ファセット記号><sp><<NJ単位文>> [<rt><相互関係>]
	11 <サブスロット>:=:<sp><sp><ファセット記号><記述階層表現><<NJ単位文>> [<rt><相互関係>]
△	12 <ファセット記号>:=:"fn1" "fn2.1" "fn2.5" "fn3.1" "fn3.2" "fn3.3" "fn4.1" "fn4.2" "fn4.3" "fn4.7"
	13 <記述階層表現>:=:<mr> [<インデント>] <階層表記線>
	14 <階層表記線>:=:<">" "(集約)>>
	15 <mr>:=:<直上の行に<階層表記線>があればそこまで右シフト, そうでなければそのまま. move right の省略形>
	16 <インデント>:=:<下の記述階層であることを示す各 DU 定義の任意一定幅空白.default は全角一文字分>
○	17 <rt>:=:<右詰め>
○	18 <lf>:=:<次の行またはスロットに移り, その左先頭に戻ることを示す記号>
○	19 <sp>:=:<任意一定幅空白>
	20 <分岐/反復文>:=:<分岐/反復総称文><lf> (<if文> <while文>)
	21 <分岐/反復総称文>:=:<スロット番号><sp><ファセット記号><sp>["(反復)" "(分岐)"] <<NJ単位文>> [<rt><相互関係>]
	22 <while文>:=:<sp><sp><ファセット記号><記述階層表現>"while"<sp><<反復条件判定NJ文>> (<lf><"while文">反復)+ <lf><mr>"while ここまで"
	23 <"while文">反復:=:<sp><sp><ファセット記号><記述階層表現><特定条件下振舞い文> [<lf> (<サブスロット記述> <分岐反復文記述>)]
	24 <if文>:=:<sp><sp><ファセット記号><記述階層表現>"if"<sp><<分岐判定NJ文>> <lf><"if文">分岐>
	25 <"if文">分岐:=:<then 振舞い文> [<lf><else 振舞い文>]
	26 <then 振舞い文>:=:<sp><sp><ファセット記号><記述階層表現>"then"<sp> (<特定条件下振舞い文> <if文>)[<lf> (<サブスロット記述> <分岐反復文記述>)]
	27 <else 振舞い文>:=:<sp><sp><ファセット記号><記述階層表現>"else"<sp> (<特定条件下振舞い文> <if文>)[<lf> (<サブスロット記述> <分岐反復文記述>)]
	28 <特定条件下振舞い文>:=:(<<NJ振舞い文>> [<rt><相互関係>]) <分岐反復文記述>
○	29 <注釈>:=:["<スロット番号>"] ("注釈:") <<NJ記述>>

(注) ○ : oonj1 と共通の規則 △ : oonj1 との共通の規則を拡張 残りは oonj2 用規則

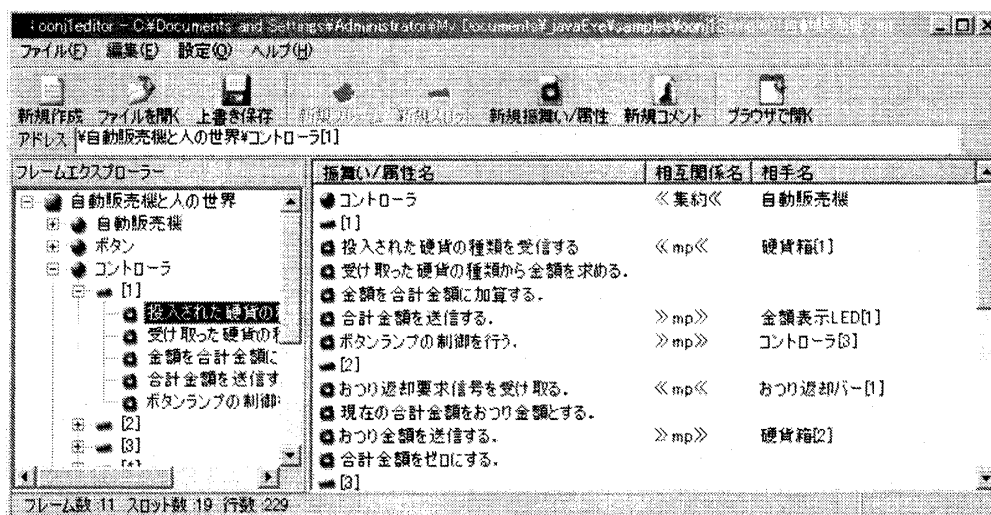


図 1 oonj1 記述エディタの主画面編集作業

Fig. 1 Editing operations at main GUI windows of oonj1 description editor