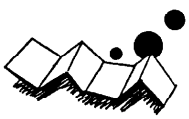
解　説



# エンドユーザのための日本語によるプログラミング†

神田泰典††　杉本正勝††　沢井　進††

## 1. はじめに

本解説ではエンドユーザがプログラミング* するのを容易にすることに焦点を合わせ，日本語によるプログラミングの必要性，米国に於ける英語によるプログラミングの研究・開発の状況，日本語によるプログラミングの方式および処理について述べる．

## 2. 日本語によるプログラミング

### 2.1 日本語によるプログラミングとは

日本語によるプログラミングを，ここでは次の二つのことを意味するものとした[1)~4)]．

（1） 人工言語としての日本語プログラミング言語の利用

従来，日本では欧米に起源をもつプログラミング言語を利用してきた．COBOL, PL/I, PASCAL 等がこれである．プログラムの表記には，英文又はローマ字文を用いてきた．また，かな文字表現もあるが一般化はしていない．

現在，日本語情報処理システムという名前で，かな漢字混じりの文章もコンピュータで処理できる汎用システムが開発されてきており，プログラムの表現をかな漢字混じりの表現にすることが容易になってきた．これによって，プログラムの読み易さ，理解のし易さを大幅に向上できることは確実である．本アプローチは従来のプログラミング言語の路線上のもので，人工

```
データ部:
ファイル節:
    ファイル記述:    売り上げ入力ファイル
                     ラベルレコードはなし,
                     データレコードは売り上げ入力領域.
                     01  売り上げ入力領域.
                         02  区分          形式  99.
                         02  得意先コード  形式  9(4).
                         02  商品コード    形式  9(10).
                         02  数量          形式  9(5).
                         02  単価          形式  9(7).
                         02  金額          形式  9(7).
    ファイル記述:    売り上げ出力ファイル
                     ラベルレコードはなし,
                     データレコードは売り上げ出力領域.
                     01  売り上げ出力領域.
                         02  区分          形式  99.
                         02  得意先コード  形式  9(4).
                         02  商品コード    形式  9(10).
                         02  商品名        形式  X(15).
                         02  数量          形式  9(5).
                         02  単価          形式  9(7).
                         02  金額          形式  9(7).
                         02  仕入単価      形式  9(7).
    ファイル記述: 元帳
                     ラベルレコードはなし,
                     データレコードは元帳領域.
                     01  元帳領域.
                         02  商品コードキー  形式  9(10).
                         02  商品名          形式  X(15).
                         02  在庫            形式  X(10).
                         02  仕入単価        形式  9(7).
                         02  仕入数          形式  9(8).
                         02  売上数量        形式  9(8).
手続き部:
    始め:    入力として売り上げ入力ファイル，出力として売り上げ
             出力ファイル，入出力として元帳を開けよ.
    読み込み:
             売り上げ入力ファイルを読め，終わりでは終わり処理へ
             行け.
             売上げ入力領域の商品コードを商品コードキーに転記せ
             よ.
             元帳を読め，無効キーの場合エラー処理へ行け.
             元帳領域を売上げ出力領域に対応をとって転記せよ.
             売上げ入力領域を売上げ出力領域に対応をとって転記せ
             よ.
             売上数量に売上げ出力領域の数量を加えよ.
             在庫から売上げ入力領域の数量を減ぜよ.
             元帳領域を再書き込みせよ.
             売上げ出力領域を書け.
             読み込みへ行け.
    エラー処理:
             '該当キーがない'，売上げ入力領域の商品コードを表
             示せよ．読み込みへ行け.
    終わり処理: '終わり' を表示せよ.
             実行停止.
             プログラムの終わり.
```

図-1 「日本語コボル」のイメージ

† Programming in Japanese for Endusers by Yasunori KANDA, Masakatsu SUGIMOTO and Susumu SAWAI (Development Division, Fujitsu Ltd.).

†† 富士通(株)開発事業部開発技術部

* プログラミング：プログラムを設計し書きテストすること（JIS 用語）

プログラミング言語：計算機が受け入れる命令又は文（プログラム）を表現する人工言語（JIS 用語）

言語の行き方であるので，「日本語表現の人工言語」と名づけて良いものである．図-1 には COBOL のプログラム表現を英語から日本語に変えたプログラムを示す．COBOL の用語の日本語訳は JIS COBOL に従った．このプログラムを見ても，英語表現のプログラムよりも格段に読み易く，理解し易いことは明らかである．

（2） 自然言語としての日本語の利用

従来のプログラミング言語は人工言語であり，プログラムを作成するには言語ごとに定められた言語仕様書をマスターしなくてはならない．言語仕様書の厚さも数百ページとなり，一通りの勉強をするにも大変な努力が必要である．エンドユーザ（コンピュータについては専門家でない利用者）は，分厚い仕様書は歓迎しない．できるだけ少ない努力でプログラミングができることを期待している．そこで誰もが日常使用している日本語でプログラミングする技術が注目されている．

エンドユーザのプログラミングを容易にすることをねらう立場からは(2)の意味での日本語によるプログラミングの発展をめざすべきであると思う．

### 2.2 日本語によるプログラミングに注目する理由

LSI の設計／製造技術を始めとするコンピュータ技術の急速な発展の結果として，コンピュータの利用について，次の二つの変化が起こっている．

（1） プログラミング言語に精通していない利用者が増えてきた．

（2） 市場の分化が原因して，利用者の要求が多様化して複雑化してきた，

(1)については，特にビジネスコンピュータの分野ではコンピュータを導入した利用者が専門のプログラマを雇ってプログラムの開発を行う余裕がないところも多い．

汎用のソフトウェア・パッケージを利用して，ともかくコンピュータを導入した主目的の定常業務をコンピュータでこなすことができるが，利用者に固有のプログラムが必要になると高価なソフトウェアを外注しなければならない状態である．

コンピュータ・メーカおよびソフトウェア・ハウスはエンドユーザ言語を開発しエンドユーザがデータベース問い合わせをしたり，プログラミングをするのを容易にする努力をしている．日本語によるプログラミングは，正にこのエンドユーザ言語の開発の最終目標と言ってよい．

エンドユーザは日本語でプログラムが作成できるので，コンピュータをより身近に感じるであろうし，日本語で書いたプログラムを使用すれば，多くの人がその機能と処理方法を理解できるようになり，プログラムの正確な利用とか保守ができて好都合である．

### 2.3 具 体 例

米国では，英語によるプログラミングは大別して，自動プログラミングをめざした研究レベルのものと，ビジネス分野をねらったエンドユーザ向きのものがある[1),2),8)]．

前者は，人工知能の研究でありプログラムの仕様書を英語で記述すると，その仕様書通りのプログラムが自動的に生成されるという「夢のシステム」の研究である．この例には PSI, NLPQ, SAFE等がある[1),2)]．

ビジネス分野では，ごくごく小規模な英語のサブセットを定めてプログラミングするシステムが用いられている．

この例には，ADAM, DJINNI 等がある．

次に PSI, ADAM, DJINNI について説明する．

（1） P S I[9)~13)]

米国の SRI インタナショナルで研究中のプログラムの自動作成システムである．対話形式でかつ英語を用いて，作成すべきプログラムの仕様をコンピュータに与える．このシステムは，英語で与えた仕様から，LISP 言語のプログラムを自動的に作成する．対話形式で英語を用いて仕様を与える他に，高級プログラミ

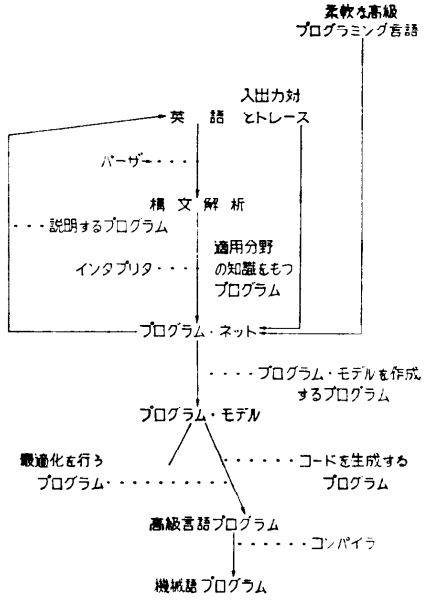



図-2 PSI の情報の流れ

PSI: What is the name of the program you want to write?
PSI: あなたの書きたいプログラムの名前は？
USER: CLASS.
利用者: CLASS
PSI: Does CLASS have a special program type?
PSI: CLASS は特殊なプログラムのタイプをもっていますか？
USER: Classification
利用者: 分類
PSI: Please describe CLASS.
PSI: CLASS を記述してください
USER: CLASS first inputs a concept. Then it repeatedly
利用者: まず初めに CLASS は概念を入力します．次に
accepts an input scene from the user, classifies it basd on
利用者から入力シーンを繰返し入力してもらいます，
whether or not it fits the concept, and outputs this
概念と一致するかどうかによってそれを分類し，利用者
classification to the user.
に この分類を出力します．
PSI: What is the structure of the concept?
PSI: 概念の構造は？
USER: A concept is a collection of relations.
利用者: 概念は関係の集合です．
Each relation is just a string.
各関係は文字列です．
PSI: What is the exit test of the loop?
PSI: ループの出口条件は？
USER: Stop when the user types "quit" instead of a scene.
利用者: シーンの代りに利用者が「やめ」とタイプするとやめる．
PSI: What is the structure of the scene?
PSI: シーンの構造は？
USER: The same as the concept.
利用者: 概念と同じ．
PSI: Please define what it means for a scene to fit a concept.
PSI: シーンが概念と一致することは何で決めますか？
USER: A scene fits a concept if all the relations in the
利用者: もし概念中のすべての関係がシーン中にあればシーンは概念
concept are in the scene.
と一致する．
PSI: What is the format of the classification output?
PSI: 分類結果の形式は？
USER: Output "fit" if the scene fit, "didn't fit" if it didn't.
利用者: もしシーンが一致するなら「一致」と出力し，もし
そうでなければ「不一致」と出力します．
PSI: Are we finished?
PSI: 終わりますか？
USER: Yes.
利用者: ハイ．

**図-3** PSI との会話の一例

ング言語で指定したり，プログラムの動きを入出力の対とトレースで与える研究も行っている．

**図-2** に示すように，PSI は対話用のプログラム，適用分野の知識をもつプログラム，プログラム・モデルを作成するプログラム等から構成されている．

**図-3** には，PSI で英語を用いて対話形式で仕様を与える具体例を示す．この例では，利用者は「CLASS」という名前の記号パターン分類のプログラムの仕様を対話形式で与えている．この仕様からプログラムが自動的に生成される．例えば，「概念の構造は？」と言うシステム側からの質問に対して，利用者が「概念は関係の集合です．各関係は文字列です．」と答えるようにして，仕様をシステムに伝える．PSI は，まずパーザを用いて構文解析をし，次に適用分野の知識をもつプログラムとインタプリタを用いてプログラム・ネットを作成する．更に，プログラム・ネットはプログラム・モデルに変換される．最後に，最適化プログラムとコード生成プログラムが，LISP 言語のプログラムを作り出す．

このシステムは人工知能の研究としてかなり高級なところをねらっている．即ち，プログラムを処理の順に手続きとして記述するのではなく，機能仕様書のように非手続き的に記述できることを目標とした自動プログラミングの研究であり，今後の成果が楽しみである．

（2） ＡＤＡＭ[14]

これは米国のロジカル・マシン社の商品であり，エンドユーザ用に開発されている．**図-4** に示すように COBOL 言語の非常に小さなサブセットのような言語である．通常のプログラミング言語でサブルーチン名とか手続き名と呼ばれる副プログラム名の定義は，ADAM では動詞の定義と呼んでいる．これは，ADAM のシステムで用意している言語仕様を英語のサブセットとして規定し，動詞を新たに定義することで，サブセットを拡張しようという設計思想からきたものであろう．但し，ADAM では動詞は定義できるが助詞，目的語等を含めた形では定義できないので，利用者が定義した動詞を頻繁に使用すると英語らしさが失われてしまう．

（3） DJINNI[15]

このシステムも「非専門家のプログラマ」用に開発されている．英語の文章でプログラムを作ることをねらったシステムである．対象とするプログラムは事務処理分野に限定している． the, of, by を含む約 400

例 題

```
VERB･･･CALLED･･･インプット・データ・プリント
1･･･ DISPLAY "インプット・データ"
2･･･ MOVE   0････TO･･･カウンタ
3･･･ INPUT      インプット・エリア
4･･･ヘッディング
5･･･ PRINT      インプット・エリア
6･･･ INCR       カウンタ
7･･･ REP
 VERB･･･ CALLED ･･･ヘッディング
1･･･ IF     カウンタ> 20
 ･･････DO･･･RETURN
2･･･MOVE 0    ･･TO ･･･    カウンタ
3･･･ PRINT "インプット・データ"
4･･･ RETURN
```

**図-4** ADAM のプログラムの例

```
Call this program "Test".
Enter data in item 1 (Stock No.).
Put data into item 2 (Quant.).
Store data into item 3 (Unit Price).
Put the product of item 2 and item 3 into
  item 4 (Extension).
Print item 1, item 2, item 3, item 4.
Done
```

図-5 DJINNI のプログラムの例

語の語彙を持っている．利用者はこれらの語彙を習得すれば，通常の英語の形式でプログラムを書くことができる．

図-5 に DJINNI のプログラムの例を示すが，the product of…and…into…のような通常の英語表現が使える．

しかし，語彙と言い回しを追加する機能が利用者に与えられていないので，利用分野と適用範囲が限定されてくる．

### 2.4 日本の現状

日本ではかな漢字混じりの日本語の文章をコンピュータで扱いにくかったことが原因で，日本語によるプログラミングはこれまでほとんど研究されていない．

COBOL 言語をかたかな表現にしたカナ COBOL も提案されたが，かなだけの文章は日常使用していないので，カナ COBOL は普及していない．

最近は日本語情報処理が盛んになり，コンピュータでかな漢字混じりの日本語の文章を取り扱い易くなったこと，人工知能の研究が進んで自然言語としての日本語をコンピュータで扱う技術が進んだこと等の理由で，日本語によるプログラミングが注目されるようになってきた[16)~18)]．

## 3. 日本語によるプログラミングの方式

### 3.1 利用者の言葉を用いたプログラミング

日本語によるプログラミングがエンドユーザに受け入れられるためには，利用者がコンピュータに利用者の言葉を登録し利用できるような機能が必要となる．

従来も COBOL, PL/I 等の高水準プログラミング言語のマクロ機能を用いて，利用者の定義したマクロに従ってソースプログラムを変更することができた．しかし，マクロ機能を用いてもプログラミング言語の構文そのものを変更することはできなかった．エンドユーザのための日本語によるプログラミングでは，利用者の言葉，即ち，利用者の言い回しとか，用語を簡単に定義して使えるようにすれば非常に使い易いものとなる．

### 3.2 利用者の言葉の登録と利用

利用者の言葉を登録するには，次の二つの方式を組み合わせて利用する方法がある．

（1） 既存の言語規則を用いて新たな語句を登録する

（2） 既存の語句を用いて新たな言語規則を登録する

この登録の過程は，幼児が日本語を習得する過程を説明する簡単なモデルにもなっている[19)]．

一方，コンピュータは利用者の語句の登録，言語規則の登録に矛盾がないかを検査する．動詞の場合は言い回しが正しい格支配をしているかを検査すること，名詞と名詞相当句の場合は語句が正しい意味分類を持つか検査する等の処理が必要となろう．

図-6 に利用者の言葉を登録し利用したプログラムの一例を示す．「終わり．」までが一つの言語規則の登録であり，この登録した利用者の言葉を「当期ファイルから当期レコードを読む．」のような形式で利用している．なおこのプログラムで〔 と 〕で囲まれる動詞，例えば〔改行する〕，〔出力する〕は定義なしに使える動詞である．

日本語のプログラムをコンピュータに入力するときには，入力を簡単にするように，ディスプレイ画面上

```
プログラムの始め．
（ファイル）から（レコード）を読む：
  ファイルが終わったら，レコードの先頭に「おわり」を〔入れる〕．
  ファイルからレコードを〔読む〕．
  終わり．
（レコード）を行頭に印字する：
    〔改行する〕．
    レコードを〔出力する〕．
    終わり．
（レコード）を（ファイル）に書く：
    レコードをファイルに〔書く〕．
    レコードを行頭に印字する．
    終わり．
（当期ファイル）と（旧ファイル）から（新ファイル）を作る：
    当期ファイルから当期レコードを読む．
    旧ファイルから旧レコードを読む．
  繰り返し：
（1）当期レコードの先頭が旧レコードの先頭より大きいか，等しいなら，旧レコードを新ファイルに書き，旧ファイルから旧レコードを読む．
（2）当期レコードの先頭が旧レコードの先頭より小さいなら，当期レコードを新ファイルに書き，当期ファイルから当期レコードを読む．
（3）当期レコードの先頭が「おわり」なら，処理を終わる．
  繰返し終わり．
    終わり．
プログラムの終わり．
```

図-6 利用者の言葉の登録と利用の例

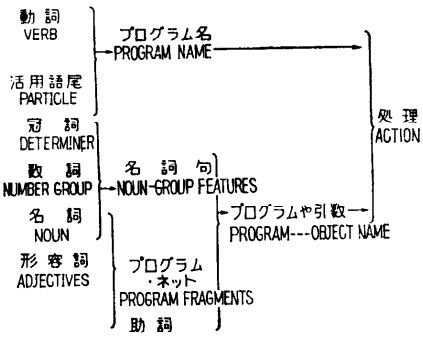



図-7 プログラムと自然言語との関係

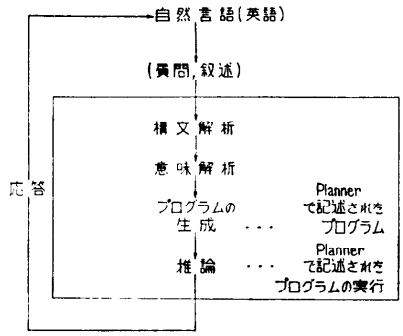



図-8 自然言語理解システムの処理の流れ (SHRDLU)

で文の種類をメニュー選択し，語句をパラメータ形式で入力するような工夫も必要となろう．

### 3.3 日本語によるプログラミングの処理法

日本語によるプログラミングを実現するためにコンピュータが行うべき処理には，人工知能の研究[20)~27)]における「言語理解システム」の処理系と同様な手法を用いることができる．

図-7 に示すように言語理解システムでは動詞は特定のプログラムを選択し，名詞がその引数となっている．

例えば，SHRDLU の「Pick up a red object which supports a pyramid.」という例では，動詞「Pick up」は引数1個を持つ関数名「PICK_UP」を指定する．このとき，名詞句「a red object…」は変換されて内部生成名「31415」になる．全体として「Pick up a red object which…a pyramid.」は (PICK_UP 31415) というリスト表現のプログラムとなる．

日本語によるプログラミングも全く同様に行うことができる．「(~)を(~)で割り(~)とする．」という言葉をコンピュータに登録しておけば，「金額の総計を出席者の総数で割り金額平均とする．」というように

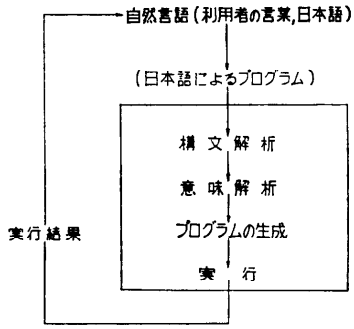



図-9 日本語によるプログラミングの処理系

プログラミングして，かつコンピュータで実行できる．

図-8 に SHRDLU に例をとって自然言語理解システムの処理の流れを示す．日本語によるプログラミングの処理系は，例えば図-9 に示すように，自然言語理解システムと同様な構成で実現できる．

## 4. おわりに

日本語によるプログラミングの必要性，米国の英語によるプログラミングの研究・開発の状況，日本語によるプログラミングの方式および処理系について述べた．

コンピュータの利用者の拡大により，コンピュータの使い易さに対するニーズが増々大きくなって行くこと，他方では人工知能の研究の成果で日本語の構文／意味解析，推論，知識ベース等が実用化の領域に入ってきたことを考え合わせると，日本語によるプログラミングの将来は非常に明るいと考える．

また，日本語でプログラムが作成できることにとどまらず，プログラムの処理系はプログラムの正しさを示すための「プログラムの実行過程を説明する機能」等細かい配慮をして，エンドユーザに真に受け入れられるものとしなければならない．

## 参考文献

1) Heidorn, G. E. : Automatic Programming Through Natural Language Dialogue : A Survey, IBM Journal of Research and Development, Vol. 4, 302-313, (July, 1976).
2) Heidorn, G. E. : Automatic Programming : Prospects and Problems, Future Programming, Infotech State of the Art Report, 101-108 (1978).
3) 国井監修：ソフトウェア工学——要求仕様技術，BIT, No. 126 (8月 1978).

4) 神田：日本におけるソフトウェア，電子通信学会誌，Vol. 61, No. 9, 929-931 (1978).
5) 田中(穂)：将来のコンピュータ・システムに関する一つの話題—自然言語と計算機技術—，電子工学月報，Vol. 21, No. 41 (1979).
6) Kanda, Y. and Sugimoto, M.: SDD: Software Diagram Description, Proc. 3 rd UJCC (1978).
7) Winograde, T.: Beyond Programming Languages, CACM, Vol. 22, No. 7 ((July 1979).
8) Sammet, J. E.: Programming Languages: History and Foundamentals, Prentice-Hall (1976).
9) Green, C.: Lectures on Automatic Programming and List Programming, PIPS-R-No. 12 (Nov. 1976).
10) Green, C.: A Summary of the PSI Program Synthesis of Very High Level Programs, SIGART Newsletter, No. 64, 380-381 (August 1977).
11) Green, C. and Galeriel, R. P.: Results in Knowledge Based Program Synthesis, Proc. 6-th IJCAI, 342-344 (Aug. 1979).
12) Green, C. and Ginsparg, J.: Knowledge Based Programming, Stanford Computer Science Dept., Report -No. STAN-CS-78-695, 45-61 (1979),
13) McCune, B. P.: Synthesis of Very High Level Programs, SIGART Newsletter, No. 64, 130-139 (Aug. 1977).
14) Shaffer, R. A.: Computers That Use Plain English Permit Vast New Applications, The Wall ST. Journal, (March 27, 1978).
15) Berkeley, E. C.: Automatic Computer Programming Using Ordinary Natural Language: A Brief Report, Computers and People, 22-23 (Dec. 1978).
16) 上田，菅野，野田：高級言語 AFL とその処理システム，日経エレクトロニクス (10月 1977).
17) 神田，杉本：ソフトウェア工学に於ける日本語の役割，情報処理学会ソフトウェア工学研究会 (1月 1978).
18) 日本語情報処理シンポジウム予稿集，情報処理学会 (1978).
19) 天野：幼児の文法能力，国立国語研究所報告-58 (1977).
20) 西野：人工知能研究についての二，三の問題，情報処理，Vol. 15, No. 6 (1月 1974).
21) Winograde, T.: Understanding Natural Language, Academic Press (1972).
22) 長尾，辻井：自動書語処理プログラム，情報処理 Vol. 18, No. 1 (1977).
23) Winston, P. H.: Artificial Intelligence, Adison Wisley (1978).
24) Tanada, H., Sato, T. and Motoyoshi, F.: A Semantic Parsing System For Japanese Sentences, Proc. 3 rd UJCC, 236-240 (1978).
25) 田中(幸)：人工知能の展望，情報処理，Vol. 19, No. 10 (1978).
26) 淵：コンピュータと認知科学，情報科学，No. 193, 17-12 (6月 1979).
27) 特集 日本語情報処理，情報処理，Vol. 20, No. 10 (10月 1979).

(昭和54年10月23日受付)