



6. P L / I †

花田 収悦††

1. はじめに

プログラミング言語 PL/I の開発は 1963 年に IBM 社とそのユーザ団体である SHARE* によって始められた。当時、計算機用高級言語としては、FORTRAN, ALGOL, COBOL が普及していたが、計算機のハードウェアやオペレーティングシステムなどの急速な技術進歩によってもたらされた新たな機能を容易に利用でき、しかも科学技術計算、経営事務計算などを含む広範囲の複雑なアプリケーション・プログラムにも利用できる高級言語の開発に対する要望が強くなってきた。このような背景から PL/I は、FORTRAN, ALGOL, COBOL などの、その当時に存在していた言語のほとんどの機能を包含するとともにいくつかの新たな機能を導入した。

初めは NPL** という名前で 1963 年に提案されたが、後に PL/I と改名され 1965 年に IBM 社によって言語仕様の第 1 版が作成された。そのコンパイラは 1966 年に同じく IBM 社が第 1 版を開発したが言語仕様の全てを満たすものではなく、ひきつづき改版を重ねて 1968 年の第 4 版においてほぼ全機能を含むコンパイラとして完成した¹⁾。

PL/I の標準化は 1965 年から ECMA と ANSI を中心に作業が開始され、10 年以上の検討を重ねて、1979 年に PL/I 国際規格 ISO 6160 が制定された^{2), 12), 17), 18)}。

しかし、現在利用されている処理系の大部分はまだ標準規格に準拠しておらず、仕様面からは ISO と IBM 系の 2 系列が存在するとともに、最適化処理などに独自の工夫がなされており多様化している。日本において利用されている代表的な処理系の特徴として以下のものがあげられる(表-1)。

† Programming Language One by Shuetsu HANATA (Yokosuka Electrical Communication Laboratory of NTT).

†† 日本電信電話公社横須賀電気通信研究所

* 科学技術計算を中心とする IBM マシンのユーザ団体

** New Programming Language

表-1 日本における代表的な処理系の特徴

処理系の種別	仕様注1)	特 徴
A	LS 70 ISO注2)	① 強力な最適化処理 ② カナメッセージ ③ 文の出現頻度などの統計情報機能
B	ISO	① 翻訳時機能 ② DECIMAL FLDAT の内部 10 進表現 ③ EBCDIC 系と ASCII 系の指定
C	LS 70	① 漢字データ ② DECIMAL FLDAT の内部 10 進表現 ③ EBCDIC 系と ASCII 系の指定
D	LS 70	① 漢字データ ② データベースアクセス機能 単体デバッグなどのチェックアウトコンパイラ
E	LS 70	① データベースアクセス機能 ② オンライン通信機能 ③ 漢字データ
F	LS 70	① 翻訳時機能 ② マルチタスク機能 ③ 非同期入出力機能
G	ISO	① データベースアクセス機能 ② リアルタイム機能 ③ 文字列操作機能の強化

注1) ISO 規格に準拠した仕様を ISO とし、IBM 社が 1970 年に発表した PL/I^{††} Language Specifications GY 33-6003-02 に準拠した仕様を LS 70 とする。

注2) 両仕様に準拠している。

(1) ISO 規格に準拠した仕様よりも、IBM 社が 1970 年に発表した PL/I Language Specification GY 33-6003-02 に準拠した仕様を持つ処理系が多い。

(2) 強力な最適化機能とデバッグ機能の充実がはかられている。

(3) 漢字データの処理機能やカナメッセージなどの日本語固有の処理を追加しているものが多い。

(4) データベースアクセス機能、オンライン通信機能等の拡張機能を導入しているものがある。

一方、1976 年からは PL/I 仕様のサブセット化と拡張に関する 2 つの作業が、それぞれ ANSI の X3 J13 と X3 J14 の委員会において開始された。サブセット仕様はサブセット G (general purpose の意) と呼ばれ 1979 年に ISO Draft Proposal 6522 となった⁹⁾。

拡張仕様はリアルタイム機能の拡張をはかった仕様案 (ドキュメント T) が提案され、現在第 4 版まで改版されている^{4),5)}。

2. 特 徴

PL/I の特徴としては以下の 3 点に要約できよう。

- (1) 言語仕様が豊富である。
- (2) 汎用のサブセット仕様を持つ。
- (3) 仕様の規定が厳密である。

2.1 言語仕様の豊富さ¹²⁾

PL/I は科学技術計算や経営事務計算のいずれの分野も包含する広汎なアプリケーションプログラムを記述できることを最大の特徴とする。このため、他言語に比較しデータの種別や、入出力機能、演算機能などにおいて豊富である。反面、仕様が大きすぎて使いにくいという批判が当初から存在していた。しかし、最近では FORTRAN, COBOL においても機能拡充が行われ、機能的には PL/I に匹敵するものになりつつある。

表-2 サブセット G における削除または制約機能とその理由

項番	削除または制約機能	ANSI の判断根拠
1	データタイプの数を約半分にする	コードパターンや実行時ルーチンの数を削減するため。
2	・配列に対し式は使えない ・構造体や集合体の混在は許さない	サブセットの機能としてインプリメントコストがかかりすぎる。また混在データの使用頻度は低い。
3	・書式指定の中には定数しか書いてはいけない ・PUT DATA, GET DATA 文の削除	・実行時に必要なコード数を減らす。 ・コンパイラが簡単になり、インプリメントコストが削減できる。
4	DEFAULT 文, ENTRY 文, LOCATE 文の削除	コンパイラや言語仕様が複雑になるのに比べ使用頻度は少ない。
5	使用変数はすべて DCL 文で宣言するかラベル後頭語として明示宣言しなければならない	プログラムエラーを減らし、読解性を良くする。
6	データタイプの変換は取り易いものについて禁止する	プログラムエラーが減少する。
7	条件についていくつかの制限を設ける	インプリメントを簡単にする。
8	条件接頭語を禁止する	多くのハードウェア上で効率良く実現する。実行時エラーの選択的検出はコンパイラオプションやデバッグシステムなどで行われるべき。
9	DO 文を単純化する	エラーをおかしくにくくする。
10	CONTROLLED 属性を削除する	BASED 属性と機能的に重複する面が多い。
11	REFER オプションを削除する	インプリメントコストに比べ使用頻度が少ない。
12	DEFINED 属性は単純列オペラレ定義のみ許す	POSITION, ISUB, * 添字はユーザにとってもインプリメントにとっても複雑になる。

PL/I で新たに導入された概念としては、

- (1) 流れ入出力, レコード入出力を分離し, ファイル操作まわりの概念を明確化したこと,
- (2) STATIC, AUTOMATIC, CONTROLLED および BASED のクラスを設けて記憶域の操作機能を導入したこと,
- (3) 翻訳時に原始プログラムの出力形式を指定したり, 原始プログラムの一部をテキストファイルから入力するといった翻訳時機能を導入したこと, などがある。

2.2 汎用のサブセット仕様^{9),13)}

汎用サブセット (ISO DP 6522) は科学技術計算, 事務処理, およびシステムプログラミングの各分野に適用でき, 大部分のミニコンピュータと一部のマイクロコンピュータも含めた小規模の計算機においても効率良く実現できることを目的としている。

この目的を満たすべく, 使用頻度, インプリメントの難易度, 有用性などを考慮して, フルセット仕様から機能の削除, 制限などがなされている。

主要な削除機能および制限とそれらに関する ANSI の判断根拠とを表-2 に要約する。

表-3 PL/I と FORTRAN の言語仕様の記述比較例

分類	PL/I ¹⁴⁾	FORTRAN ¹⁵⁾
注) 構文の定義	(構文定義構文) <GOTO 文> = {GOTO GO TO} <参照> <参照> = [<ロケータ修飾子>] <基本参照> [<アークギュメントの組の並び>] <基本参照> = [<構造体修飾>] <名標> : (特性定義構文) <GOTO 文> = <値参照> (スカラ & LABEL) <値参照> = (<変数参照> <手続き関数参照> <組込み関数参照> <名前付き定数参照>) <データ記述> :	GO TO(S[,S]...), [,] i ここで i は整数式とする。 S は ... :
	動作: GOTO 文の実行 (go) ただし, go は <GOTO 文> である。 段階 1 vr を go の <値参照> とする。 値参照の評価 (vr) を行い (集合体値, ag を得る。ag は, <ラベル値>, lv: <実行可能単位指示>, tp <ブロック状態指示子>, ban; という形式を含まなければならない。 段階 2 ... 場合 2. 1 ... :	計算形 GOTO 文の実行により, 式 i が評価される。それについて制御の移行が起こり, 文番号の並び中の i 番目の文番号を持つ文が, 次に実行される。 : :

注) PL/I の場合, ソース文に対応する構文については構文定義構文で定義しており, さらに中間言語に対応する構文についても特性定義構文で規定している。

2.3 仕様厳密さ

PL/I は言語仕様が大きいためそれを矛盾なく正確に定義することに工夫が凝らされた。すなわち国際規格では IBM のウィーン研究所で開発した ULD* という記法をベースとして PL/I の原始プログラムレベル、中間言語レベル、オブジェクトレベルの3つの段階を形式的に定義し、それらの各レベル間の変換手続きを記述する形式を採用している (表-3)。

これにより、言語仕様は正確であいまい性のないものとなっているが、構文則が BNF 形式であること、原始プログラムレベルからオブジェクトレベルまでへの変換手続きの記述が複雑であること、などにより読解性や保守性において難点が生じた。

3. 適用分野と使用状況

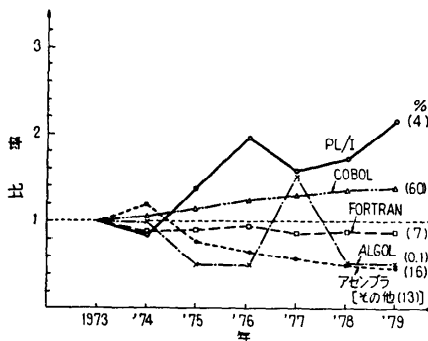
他言語と比較した使用比率を図-1 に示す。毎年増加率は大きく、また言語全体に占める使用率は COBOL, アセンブラ, FORTRAN に次いで第4位ではあるものの、その割合は約 4% である。

4. 今後の動向

約 15 年にわたる PL/I の標準化活動が 1979 年の国際規格の制定により一段落したが、他言語と同様に最新技術の反映や利用者の要望などに基づく仕様の検討が継続して行われている¹¹⁾。

4.1 リアルタイム機能^{4), 5), 13)}

リアルタイム機能の応用を目的とした拡張仕様のひとつであり、現在 ANSI の X 3 J14 委員会で審議中である。その言語仕様はドキュメント T と呼ばれ、以



注) 各言語とも 1973 年を 1 として比較。() の中は '79 年における使用比率を示す。

図-1 プログラミング言語使用比率¹¹⁾

下のような概念および定義が導入されている。

- (1) プログラム: 1つ以上の外部手続きの集まり
- (2) ジョブ: 関連したプログラムの実行
- (3) タスク: ジョブの中の1つのプログラムの実行
- (4) 共用変数: プログラムの中で宣言されていれば、それらのタスクで共用して使用することのできる変数
- (5) 共用オブジェクト: いくつかのタスクで共用して使用されるファイル、ファイル内のレコード、共用変数
- (6) ロック状態: 共用オブジェクトへのアクセス権利を定義する共用オブジェクトの状態

4.2 会話型機能⁶⁾

会話型アプリケーションのための拡張仕様であり 1976 年に ECMA において検討が開始されているが、各国への正式なコメント要請などは出されていない。

会話をを行うための INTERACTIVE ファイルとそのファイルへアクセスするための ASK 文とが新たに導入されている。

(1) INTERACTIVE ファイル: 同時に入出力可能なファイルで、INPUT, OUTPUT, UPDATE 属性と択一的な属性を持つ。

(2) ASK 文: INTERACTIVE ファイルへの入出力を行う文である。転送モード (入力, 出力, エコー付きアップデート, アップデート), 表示属性 (イタリック体, 下線付きなど) などの指定ができる。

4.3 画面操作機能⁷⁾⁻¹⁰⁾

ディスプレイ画面を持った端末の画面制御を目的とした拡張であり、1977 年から ECMA において検討が開始されている。仕様の詳細についてはまだ明らかではないが、画面の属性やページの概念の拡張などが提案されている。

(1) 画面の属性: PROTECTED, UNPROTECTED, LOWBLINK, HIGHBLINK, UNDERSCORED など

(2) ページの概念の拡張: 論理的なページに対し、物理的画面の大きさを WINDOW として定義し、ページをハードの大きさにとられないものとする。また、ページの中に任意の数のフィールドが定義でき、その各フィールド内でのスクロール機能などを提供する。

一方、仕様拡張以外の動向、課題としては自然言語による PL/I 言語仕様記述の計画が進められているこ

* Universal Language Definition on

とと、PL/I の JIS 規格化の計画が未だ具体化していないことなどがあげられる。

参考文献

- 1) 竹下 亨: PL/I・複合プログラミング言語, p. 410, 日本経営出版会 (1970).
- 2) International Standard ISO 6160 Programming Languages-PL/I (1979).
- 3) American National Standard Programming Language PL/I General-Purpose Subset BSR X3.74 SUBSET/G Revision 9 (Feb. 1980).
- 4) X3 J1/649, Document/T Revision 3 (May 1979).
- 5) Barnes Richard: A Working Definition of The Proposed Extensions for PL/I Real-Time Applications, SIGPLAN Notices, Vol. 14, No. 10, pp. 77-99 (Oct. 1979).
- 6) Outline Proposal on PL/I Extensions, Supporting Interactive Data Transmission, ECMA/TC 10/77/9 (Jun. 1976).
- 7) PL/I Extensions Supporting Full Screen Processing, ECMA/TC 10/77/17 (Nov. 1977).
- 8) Screen Management, ECMA/TC 10/79/7 (Jul. 1979).
- 9) List of Items Discussed by TG 4 at the Florence Meeting of March 14-16, 1978, ECMA/TC 10/78/4.
- 10) Final Disposal of the Development File, ANSI X3 J1/671 (Feb. 1980).
- 11) C-proposal, ANSI X3 J1/668 (Jan. 1980).
- 12) PL/I の標準化に関する調査, 日本電子工業振興協会, PL/I 専門委員会報告書, 52-C-327 (Mar. 1977).
- 13) PL/I の標準化に関する調査, 日本電子工業振興協会, PL/I 専門委員会報告書, 54-C-372 (Mar. 1979).
- 14) PL/I Language Specifications GY 33-6003-02, IBM (1970).
- 15) コンピュータ白書 '80, 日本情報開発協会編 (1980).
- 16) 電子計算機プログラミング言語 FORTRAN (JIS 改正原案), 情報処理学会, JIS FORTRAN 改正原案作成委員会 (1981).
- 17) 花田収悦: PL/I の国際規格と標準化動向, 情報処理, 第 21 巻第 9 号 (Sep. 1980).
- 18) 竹下 亨: プログラミング用語語 PL/I の標準化の動向, bit, Vol. 4, No. 4 (1972).

(昭和 56 年 2 月 3 日受付)