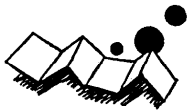


解説



PL/I の国際規格と標準化動向†

花田 收悦††

1. ま え が き

PL/I の国際規格 (ISO*6160) が昨年 (1979 年) 制定された。PL/I という名称を付した最初の言語仕様は作成されたのが 1965 年であるからちょうど 15 年目に当る。本稿では、これを機会に PL/I の国際規格の概略、および現在継続中の標準化活動等について紹介する。

2 章では PL/I の国際規格が制定されるまでの歴史をふりかえる。3 章でその内容の概略について述べるとともに、国際規格と現在最も広く利用されていると想定される言語仕様との主要な相違点を明らかにする。4 章では国際規格制定に至るまでに討議された幾つかの内容について、標準化機関に提出したコメントの概略を紹介し、あわせてその採否状況を報告する。

5 章では PL/I の国際規格は巨大すぎるという意見などがあるため、国際規格から汎用のサブセット仕様を切り出す作業を進めているが、その考え方および仕様の概要を国際規格と対比して示す。

一方、6 章では国際規格の拡充機能のひとつとして検討中のリアルタイム機能について、7 章ではそれ以外の現在進行中の標準化活動項目について言及する。

2. PL/I の歴史と標準化の経緯¹⁾

1960 年代初期には計算機用の汎用高級言語として、FORTRAN, COBOL, ALGOL 等が普及していたが、計算機のハードウェア、オペレーティングシステム等の急速な発展や、アプリケーションプログラムの複雑化、広汎化を背景として、より一層の高機能を有する汎用言語開発への要望が強まってきた。

PL/I は、これらの要望を受けて、SHARE**と IBM との協力で開発された言語である。

PL/I コンパイラは、1966 年に IBM により第 1 版が開発されたが言語仕様すべてを満たすものではな

く、1968 年の第 4 版においてほとんどの機能を含むコンパイラが完成した。

一方、PL/I の標準化は組織作りから着手され、1965 年に ECMA***に PL/I の標準化を行う委員会が作られた。1968 年に IBM から ECMA と ANSI****へ提供された資料が「BASIS/1」と名付けられ、標準仕様案の第 1 版となった。また、米国においても翌年には「Ad Hoc Subcommittee on PL/I」が設立されたが、1970 年に至って標準化作業は ECMA, ANSI の共同プロジェクトとして進められることとなった。その後各国の協力の下に言語仕様の討議を経て 1976 年に ANSI 規格が制定され、ひきつづき 1979 年に国際規格が制定された (表-1)。

さらに、国際規格の検討と並行して 1976 年から、ANSI の X3J13 と X3J14 の委員会において、それぞれ汎用サブセット (サブセット G)、リアルタイム用拡張仕様案 (ドキュメント T) の検討が開始された。サブセット G は 1979 年に ISO Draft Proposal 6522 となり、ドキュメント T は第 3 版が作成されて ANSI を中心にさらに検討が進められている。

3. 国際規格²⁾

PL/I は、科学技術計算、事務処理、その他種々のアプリケーションへの適用を目的として開発された汎用プログラミング言語である。その言語仕様は、FORTRAN, COBOL, ALGOL 等の有用な仕様をできるだけとり込み、さらに新たな機能も追加するという意欲的なものであり、以下の特徴を持つ。

- (1) プログラムの記述カラム位置が任意である。
- (2) 予約語がない。
- (3) 豊富なデータの種類 (文字列、ビット列、種類の算術データ、制御用データ*****等)、豊富なデータ構造の種類 (スカラ、配列、構造体等) を持つ。

* International Organization for Standardization.

** 科学技術計算を中心とする IBM のユーザ団体。

*** European Computer Manufacturers Associations.

**** American National Standard Institute.

***** ラベルデータ、入出力データなどのようにプログラムの実行制御に影響を与えるデータ。

† PL/I International Standard and the Trend of its Standardization Activity by Shuetu HANADA (Yokosuka Electrical Communication Laboratory, N. T. T.).

†† 日本電信電話公社横須賀電気通信研究所

表-1 PL/I の歴史と標準化の経緯

年	言 語	コンパイラ	標準化活動
1963	• IBMとSHARE とで新言語開発 委員会(ALDC) を設立		
1964	• ALDC が新言 語 NPL を提案		
1965	• NPL の改訂版 を PL/I と改名 • PL/I 言語仕様 第1版を作成 (IBM)		• ECMA に TC10/ PL/I 委員会設立
1966		• IBM PL/I コンパ イラ第1版開発	• USASI に X3.4. 2C "Ad Hoc Sub- committee on PL /I" 設立
1968		• IBM PL/I コンパ イラ第4版開発 • 電電公社 PL/I コ ンパイラ開発 (HITAC 5020)	• ECMA TC/10 TGI で PL/I の 標準化を行うこと が決定 • PL/I 標準仕様案 第1版(BASIS/1) の提案
1970		• 富士通、日電 PL/I コンパイラ開発 (FACOM230-50, NEAC 2200)	• PL/I 標準化作業 を ANSI, ECMA 共同プロジェクト とすることが決定
1974			• PL/I 標準仕様案 第12版(BASIS/1 -12) の提案
1976			• BASIS/1-12 をも とに ANSI 規格 (ANSIX 3.53-19 76) 制定 • サブセット G、ド キュメント T の検 討開始
1979			• 国際規格 (ISO 61 60 Programming Languages-PL/I) の制定 • サブセット G 第8 版が DP 6522 とな る • ドキュメント T 第 3版の提案

(4) 省略時属性の適用, 異なった属性データ間の属性変換が行われる。

(5) 割り込み機能を持ち, エラーや例外条件が生じたときの処理が可能である。

(6) 単純な入出力(流れ入出力)から複雑な入出力(レコード入出力)まで, 種々の入出力機能を持つ。

(7) 記憶域の動的な割付け, 解放が行える。

(8) 算術データ, ビット列データ, 文字列データ処理用の種々の組み込み関数を持つ。

3.1 言語仕様の概要

(1) 文字セット

市販の PL/I では, 2種類の文字セット(例えば, 60文字セットと48文字セットの2種類)を用意して, 特殊なキーを持たない小型の端末でも使用できるよう配慮しているものがあるが, 国際規格としてはそれらの配慮はなされておらず, 表-2のように57+ α (α は

表-2 文字セット一覧

分 類	文 字 セ ャ ッ ト			
英 字	A, ~, Z (26個)			
数 字	0, ~, 9 (10個)			
特殊文字	名前	字体	名前	字体
	プラス	+	コンマ	,
	マイナス	-	セミコロン	;
	アスタリスク	*	コロン	:
	斜線	/	空白	
	大なり記号	>	引用符	'
	小なり記号	<	左括弧	(
	等号	=	右括弧	)
	否定記号	~	下線	_
	論理積記号	&	ドル記号	\$
	論理和記号		パーセント記号	%
	ピリオド	.		
	(21個)			
言語用文字セット 以外の文字	処理系により定義される			

表-3 文の種類と一覧

種 類	文
代入文	代入
制御文	CALL, DO, END, GOTO, IF, STOP, RETURN
宣言文	DECLARE, DEFAULT
例外制御文	ON, REVERT, SIGNAL
入出力文	ファイル準備文
	データ指定文
	データ転送文
プログラム構造文	BEGIN, DO, END, PROCEDURE, ENTRY
記憶域割付け文	ALLOCATE, FREE
その他	空

表-4 属 性 一 覧

分 類	属 性
データ属性	処理用データの属性
	プログラム制御用データの属性
入口属性	ENTRY, RETURNS, OPTIONS
ファイル属性	FILE, INPUT, OUTPUT, STREAM, RECORD SEQUENTIAL, DIRECT, KEYED, UPDATE PRINT, ENVIRONMENT
有効範囲属性	INTERNAL, EXTERNAL
記憶域の型属性	AUTOMATIC, STATIC, BASED, CONTROLLED, DEFINED, PARAMETER
整列属性	ALIGNED, UNALIGNED
その他	VARIABLE, CONSTANT, BUILTIN, CONDITION, STRUCTURE, MEMBER, GENERIC, POSITION, INITIAL, 次元, LIKE

表-5 組込み関数一覧

分 類	組 込 み 関 数
列処理組込み関数	• AFTER (sa, ca): 列 ca と同じ部分列が sa の中にあるか調べ、あれば、その部分列以降の sa を返す • SUBSTR (sa, st, le): 文字列 sa のうち, st 番目から長さ le 分の列を返す • EVERY (X): X のすべてのビットが 1 であれば '1'B, そうでなければ '0'B を返す • その他 17 種
算術組込み関数	• ABS (X): X の絶対値を返す • BINARY (X [,p[,q]]): X を精度 ([p[,q]]) の BINARY に変換する • CEIL (X): X 以上の値で最小の整数値を返す • その他 19 種
数学組込み関数	• ACOS (X): $\cos^{-1}(X)$ を返す • ATANH (X): $\tanh^{-1}(X)$ を返す • ERF (X): $\frac{2}{\sqrt{\pi}} \int_0^X e^{-t^2} dt$ を返す • その他 18 種
配列処理組込み関数	• DIMENSION (X, n): 配列 X の n 次元目の要素を返す • DOT (X, Y [,p[,q]]): X と Y の内積を精度 ([p[,q]]) で返す • HBOUND (X, n): X の n 番目の次元の上限値を返す • その他 3 種
条件組込み関数	• ONCHAR ([]): CONVERSION 条件を起こした文字を返す • ONCODE ([]): オンコード値を返す • ONFILE ([]): RECORD 条件などを起こしたファイル名を返す • その他 4 種
基底付き記憶域組込み関数	• ADDR (X): X のアドレスをポインタ値として返す • EMPTY ([]): 空のエリア値を返す • NULL ([]): 空のポインタ値を返す • その他 2 種
その他	• DATE ([]): 年・月・日を文字列として返す • LINENO (f): ファイル f の LINE 値を返す • TIME ([]): 時・分・秒を文字列として返す • その他 3 種

(注) (1) [] は省略可能ということを意味する。

(2) 精度指定で, p は全桁数, q は小数点以下の桁数を示す。

処理系により定義される) 個の文字セットを持つ 1 種類が制定されている。

(2) 文

文は PL/I プログラムの中で制御上の最小の単位であり、表-3 に示す 31 種類がある。

(3) 属 性

属性は、プログラム中のデータやファイル等の特性を定義するものであり、表-4 に示す 56 種類がある。

(4) 組込み関数

数値処理や列処理等に便利な多くの組込み関数を用意されており、プログラムから利用できるようになっている。規格として制定されている組込み関数は表-5 に示す 87 種である。

(5) 擬 変 数

擬変数とは、変数が値を受け取るために現われる文脈で用いられる組込み関数のことである。表-6 に示す 8 種類がある。

(6) 条 件

表-6 擬変数一覧

分 類	擬 変 数
列処理擬変数	• STRING (X): X の要素を連結した列として文字列が代入される • SUBSTR (sa, st, le): sa の st 番目から長さ le の部分列へ文字列が代入される • UNSPEC (X): X の内部表現としてビット列が代入される
算術擬変数	• IMAG (X): X の虚数部へ値が代入される • REAL (X): X の実数部へ値が代入される
条件擬変数	• ONCHAR ([]): 代入された値が、CONVERSION 条件を発生した文字の代りに再評価される • ONSOURCE ([]): 代入された値が、CONVERSION 条件を発生した変数の代りに再評価される
その他	• PAGENO (f): ファイル f ヘページ番号が代入される

表-7 条 件 一 覧

分 類	条 件
計算型条件	CONVERSION, FIXEDOVERFLOW, OVERFLOW, SIZE, STRINGRANGE, STRINGSIZE, SUBSCRIPTRANGE, UNDERFLOW, ZERODIVIDE
入出力条件	ENDFILE, ENDPAGE, KEY, NAME, RECORD, TRANSMIT, UNDEFINEDFILE
プログラマが名付けた条件	CONDITION (条件名)
その他	AREA, ERROR, FINISH, STORAGE

プログラムの実行中に、例外条件により割込みが発生することがある。PL/I では、表-7 に示す 21 の例外条件を定義できるようになっており、これらが発生すれば自動的に検出され、そのときの割込みに応じた任意の処理をプログラム中に ON 文を用いて指定することができる。

(7) 省 略 形

使用頻度の高い一部のキーワードと組込み関数は、省略形でも指定できるようになっており記述性を高めている (例えば、ALLOCATE を ALLOC, BINARY を BIN と省略できる)。

以上が国際規格の骨子であるが、その内容は各国からのコメント等を受けて制定されたにもかかわらず、結果として現在最も広く利用されていると想定される PL/I 仕様 (例えば IBM が 1970 年に発表した、PL/I Language Specifications GY 33-6003-02; 以下、LS 70 と略す) と相当に異なる部分があり、互換に関する問題点がクローズアップされている。

LS 70 との主要な相違点について表-8 に要約する。

4. 国際規格制定における主要な討議事項

PL/I 国際規格制定に至るまでの討議内容は他国とも大略同様の観点からなされたので、言語仕様案に対する日本からのコメントとその採否状況等を中心に主

表-8 LS 70 仕様との主な相違点

機 能	LS 70 仕 様	国際標準仕様
翻訳時機能	翻訳時変数, 翻訳時式, 翻訳時手続き, 翻訳時組込み関数 (SUBSTR, LENGTH, INDEX), 翻訳時文 (%ACTIVATE, %DEACTIVATE, 翻訳時代入, %DECLARE, %DO, %GOTO, %IF, %INCLUDE, 翻訳時空文)	翻訳時文 (%INCLUDE)
マルチタスク機能	EXCLUSIVE 属性, CALL 文の TASK, EVENT, PRIORITY 機能, DELAY 文, EXIT 文, WAIT 文, UNLOCK 文, EVENT 付き入出力文, EVENT 付き DISPLAY 文, COMPLETION 組込み関数と変数, PRIORITY 組込み関数と変数, STATUS 組込み関数と変数	(削 除)
非同期入出力機能	入出力文の EVENT オプション, WAIT 文	(削 除)
最適化機能	BUFFERED/UNBUFFERED, CONNECTED, ORDER/REORDER, REDUCIBLE/IRREDUCIBLE, SECONDARY	LOCAL 属性
集合体の入口名によるサブルーチンの一括呼出し	有	無
ビット列定数表現	2進表現	2進, 4進, 8進, 16進表現
BASED 属性を持った変数の属性とローデータ修飾された変数の属性	両方の属性が一致していなければならない	両方の属性が一致していなくても良い場合がある
DO 文の REPEAT オプション	無	有
DEFAULT 文の構文	DEFAULT ALL [省略時属性指定] DEFAULT 適用条件指定 [省略時属性指定, 適用条件指定 [省略時属性指定]]...; 適用条件指定は以下のいずれかである。 RANGE (名標[, 名標]...), RANGE (英字 (英字: 英字) [, 英字]...), RANGE (*), DESCRIPTORS, CONSTANTS, (適用条件指定 [省略時属性指定]) 省略時属性指定は, 以下のいずれかである。 属性の並び (VALUE (値指定)), VALUE (値指定) 値指定は, 以下のいずれか, またはコンマで区切って並べたものである。(精度[, 位取り関数]), CHARACTER (長さ), BIT (長さ), AREA (大きさ)	DEFAULT NONE; DEFAULT SYSTEM; DEFAULT (適用条件指定) ERROR; DEFAULT (適用条件指定) 省略時属性指定; 適用条件指定は以下のいずれか, またはそれらを, " ", "&" および括弧で組合せた論理式である。 属性キーワード, RANGE (名標), RANGE (英字: 英字), RANGE (*) 省略時属性指定は属性の並び, または属性の並びをコンマで区切って並べたもの
組込み関数	(国際標準仕様でない組込み関数) ALL, ANY, COMPLETION, COUNT, DATA FIELD, ONCOUNT, ONIDENT, POLY, PRIORITY, REPEAT, STATUS	(LS 70 仕様でない組込み関数) AFTER, BEFORE, COLLATE, COPY, DECAT, DOT, EVERY, ONFIELD, PAGEND, REVERSE, SOME, SUBTRACT, VALID
ピクチャ	繰返し回数	ゼロである繰返しがかかるべき文字が無視される
	文字列ピクチャ	繰返し回数
	浮動小数点数ピクチャ	繰返し回数
	指数指定ピクチャ	繰返し回数
	小数部のみからなる数値ピクチャ	繰返し回数
	英貨ピクチャ文字	繰返し回数
パラメータの記憶域属性の指定	繰返し回数	繰返し回数
	繰返し回数	繰返し回数
	繰返し回数	繰返し回数
	繰返し回数	繰返し回数
	繰返し回数	繰返し回数
	繰返し回数	繰返し回数
DEFAULT 文の指定	どのブロックに指定しても良い	最外ブロックのみ指定可能である
INITIAL 属性	CALL オプションを指定することができる	CALL オプションは指定できない
ベース, スケール, モードが欠けているときに補われる属性	DECIMAL, FLOAT, REAL	BINARY, FIXED, REAL
集合体式の評価	集合体演算は, オペランドのスカラ要素間の演算に展開される	集合体全体を集合体値として定義し, 集合体値に対して演算を行う
暗黙宣言に対するシステム省略時属性	先頭の文字が I~N の変数に対しては FIXED, BINARY を与え, その他に対しては FLOAT, DECIMAL を与える	すべてに対して FIXED, BINARY を与える
言語用文字セット	60文字セット, 48文字セットの2種類	57文字セット (LS 70 の60文字セットと比べ, \$, @, ? が無い)

要な項目について要約する。

日本におけるコメントは、主として ISO/TC 97/SC 5 (プログラミング言語) の国内委員会である情報処理学会 SC 5 専門委員会傘下の組織である PL/I 作業グループ (以下 PL/I WG と略記) において検討し、ECMA/ANSI 共同プロジェクトへ提出した。これらのコメントは他国のコメントに基づく改訂内容に対する検討結果も含めたものである。

その内容は、機能と定義に関する事項に大別される。機能上の問題では機能拡張の要求が、定義上の問題では、矛盾した定義の指摘が最も多かった。そのうち機能に関する主要な討議事項の概略について述べる。

(1) 機能拡張に関する事項

(ア) 採用された内容

① POSITION 属性の指定で "式" が省略された場合は "(1)" が省略されたものとみなす。

② DATA 入力の場合、入力データや入力先変数に BASED 属性を持った変数の使用も許す。

(イ) 否決された内容

① ENTRY 属性の記述子への "*" の指定

② DEFINED 属性指定時の参照変数として使用できる変数属性の追加

③ ENV 属性、OPTIONS 属性から暗黙に引き出される属性の追加

④ 指数指定ピクチャでの浮動符号フィールドの使用

⑤ GENERIC 属性を持った変数のアークギュメントとしての使用

⑥ BOOL 組込み関数における集合体アークギュメントの指定

⑦ OFFSET 変数や、遠隔書式項目における関数参照

⑧ ALLOCATE 文の SET オプションへの OFF SET 変数の指定

⑨ TRUNC, FLOOR, CEIL 組込み関数への複素数のアークギュメントの使用

⑩ UNSPEC 組込み関数での集合体アークギュメントの使用

⑪ STRING 擬変数のアークギュメントで指定できる属性の追加

(2) 機能変更に関する事項

(ア) 採用された内容

① ビットストリングの n 進出力で、出力データ長

が n の倍数にならないときは、データの前の方に 0 ビットを付けて倍数になるように合せているが、うしろの方に付ける。

(イ) 否決された内容

① DEFINED 指定の構文則の厳密化

② 入力書式制御における COLUMN 指定の位置の処理系定義化

③ 2 項演算の掛算で、結果が FIXED の場合の精度の変更

(3) 機能削減に関する事項

(ア) 否決された内容

① FILE 属性が文脈宣言されたときの CONSTANT 属性の付加の中止

② DEFAULT 文の中に現われる名標への暗黙宣言、および文脈宣言の適用の中止

③ SUBSTR 組込み関数で、文字列長を指定するアークギュメントへの集合体使用の禁止

5. 汎用サブセット言語仕様³⁾

汎用サブセット (ISO DP 6522) は科学、商業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作られたものである。

この目的から、使用頻度、インプリメントの難易度、有用性等を考慮して、フルセットの仕様のうちのいくつかの機能について削減、制限等がはかられた。

これらの機能削除および制限等は ANSI において以下のような判断根拠でなされている³⁾。

(1) データタイプが豊富であると、コードパターンの増加や実行時ルーチンの増加をまねくことから、使用頻度の少ないデータタイプを削減し、約半分に減らす。

(2) 式で配列が扱えることは、科学技術計算等で非常に有用であるが、サブセットの機能として含めるにはインプリメントコストがかかりすぎる。また、構造体や集合体の混在等については配列の場合に比べて使用頻度は少ないと考える。

(3) 書式指定の中には定数しか書けないように制限する。また、PUT DATA 文と GET DATA 文は実行時に必要なコードの数を減らすために削除する。これらの制限によりコンパイラが簡単となり、インプリメントコストが削減される。

(4) DEFAULT 文、ENTRY 文、LOCATE 文

表-9 サブセットGとフルセットとの仕様の相違

分 類	概 要
プログラムの構造	削除 ①DEFAULT 文, ENTRY 文 ②条件接頭語 ③BEGIN 文の OPTIONS オプション
	構文上の制約 ①1つの文にラベル接頭語を複数付けてはならない。また、DCL 文にはラベル接頭語を付けてはならない ②ラベル接頭語に2つ以上の添字を付けてはならない。また、PROC 文, FORMAT 文のラベル接頭語は添字をもってはならない ③END 文の多重クロスの禁止 ④PROC 文の RETURNS オプション, RETURNS 属性, STATIC 変数の宣言で指定する列長は定数でなくてはならない ⑤STATIC 変数の上下限指定は定数, または符号付き定数でなくてはならない ⑥パラメータの宣言では列長指定は定数かアステリスクでなくてはならず, 上下限対はアステリスクか定数が符号付き定数でなくてはならない ⑦算術定数に P または F の文字を付けてはならない ⑧コメントの中に / * のペア文字を書いてはならない
	意味上の制約 全関数はスカラー値を返さなければならない
プログラムの制御	削除 ①デバッグ条件 (CONVERSION, SIZE, STRINGRANGE, STRINGSIZE, SUBSCRIPTRANGE) ②プログラムの名付けた条件 ③I/O 条件 (NAME, RECORD, TRANSMIT) ④その他の条件 (AREA, FINISH, STORAGE) ⑤ON 文の SNAP と SYSTEM オプション
	構文上の制約 ①THEN 節, ELSE 節にラベルを付けてはならない ②DO 文で V, V による多重指定を禁止する ③手続参照, 関数参照では2つ以上のアーギュメント対をもってはならない
	意味上の制約 ①DO 文の制御変数は整数かポインタ変数でなくてはならない。TO, BY オプションの値は整数でなくてはならない ②IF 文や WHILE 節で指定した式は長さ1のビット列値をもたなければならない ③ON 文, REVERT 文では1つの条件名のみ指定できる ④非連続記憶域を持った配列はアーギュメントとしてはならない ⑤UNDERFLOW 条件に対応する ON ユニットから制込点へ制御を戻してはならない ⑥ON 単位の中に RETURN 文を書いてはならない
記憶域制御	削除 ①BY NAME 代入 ②ALLOCATE 文の IN エルオプション
	構文上の制約 ①代入文で代入先変数は1つでなくてはならない ②ALLOCATE, FREE 文では1つの変数しか指定できない。ALLOCATE 文では SET オプションを必ず付けなければならない
	意味上の制約 列 S を列 T に代入する場合, 列 S が列 T の前にあり, かつ列 S と列 T とが重なっていることはない
入出力	削除 ①LOCATE 文 ②OPEN 文の TAB オプション, GET 文の COPY オプション, GET 文 PUT 品の DATA オプション, READ 文の IGNORE オプション, CLOSE 文の ENVIRONMENT オプション ③C書式項目
	構文上の制約 ①OPEN, CLOSE 文では1つのファイルしか指定してはならない ②GET, PUT 文では1つの I/O リスト, 1つ以下の書式リストしか指定してはならない ③書式指定の中に式や変数参照を含んではならない ④E, F 書式項目では3番目の値を指定してはならない ⑤レコード入出力の INTO, FROM オプションではビット整列の変数を参照してはならない ⑥REWRITE 文には FROM オプションが必要 ⑦ファイル定数はスカラーでなくてはならない
	意味上の制約 入出力文中の式の中で, 現在のファイルへの入出力動作をとまらうような手続きを呼んでいる場合, その手続から呼び出し元へ制御を戻してはならない
属性・ピクチャ指定	削除 ①属性 (AREA, COMPLEX, CONDITION, CONTROLLED, FORMAT, LIKE, LOCAL, OFFSET, POSITION) ②ピクチャ文字 (A, E, I, K, R, T, X, Y) ③BIT VARYING ④SUB 定義 ⑤DECLARE 文の REFER オプション
	構文上の制約 ①ピクチャは固定小数定数値ピクチャのサブセット版とする ②以下の属性は存在するがキーワードではなくなる (CONSTANT, DIMENSION, MEMBER, NONVARYING, PRECISION, PARAMETER, REAL, STRUCTURE, UNALIGNED) ③ALIGNED 属性は列変数にのみ指定できる ④INITIAL 属性では, 列定数, 算術定数, NULL 組み関数のみが初期値として指定できる。INITIAL 属性での反復回数も定数でなくてはならない。INITIAL 属性の指定は STATIC 変数にのみできる。 ⑤負の桁移動子を指定してはならない。0 でない桁移動子は FIXED DECIMAL 変数にのみ指定できる
	意味上の制約 ①プログラムで使われる名前のデータのタイプは DCL 文か, ラベル接頭語か, 組み関数かで明示して宣言されなくてはならない。RETURN 属性の返す値, PROC 文の RETURNS オプションの返す値と同様に ENTRY 属性のパラメータにもデータタイプを与えなければならない ②DEFINED 属性で指定された変数は, DEFINED 変数と, データタイプ, 限界が一致していなければならないか, または列重ね定義でなくてはならない
組み込み関数・変数	削除 ①組み込み関数 (ADD, AFTER, ALLOCATION, BEFORE, COMPLEX, CONIG, DECAT, DOT, EMPTY, ERF, ERF, EVERY, HIGH, IMAG, LOW, MULTIPLY, OFFSET, ONCHAR, ONFIELD, ONLOC, ONSOURCE, POINTER, PRECISION, PROD, REAL, REVERSE, SOME, SUBTRACT, SUM) ②関数 (IMAG, ONCHAR, ONSOURCE, REAL)
	構文上の制約 ①MAX, MIN 組み込み関数はちょうど2つのアーギュメントをもっていなければならない ②関数は代入文の左辺にしか使用できない
	意味上の制約 ①組み込み関数の多くは, コンパイル時に評価できるようアーギュメントに制約がつけられている ②組み込み関数は集合体値を返してはならない

分 類	概 要
構文上の制約	手続き、関数参照ではアーギュメントリストは1つでなくてはならない
意味上の制約	①すべての演算は結果がスカラとなるものでなくてはならない ②すべての式は、負の桁移動子を持った固定小数点10進数を生じてはならないし、桁移動子を持った固定小数点2進数を生じてはならない ③(割算)には固定小数点2進数を用いてはならない ④添字や境界値は整数でなくてはならない ⑤算術演算子はオペランドとして算術またはピクチャしか使えない ビット列演算子はオペランドとしてビット列しか使えない 文字列演算子と文字列用組込み関数はオペランドとして文字列しか使えない 比較演算子は2つとも算術データか、または互いに同じタイプのデータでなくてはならない ⑥配列代入としては、左辺が接続記憶域を持つ配列で右辺がスカラの場合のみ許される ⑦配列代入では、互いに接続記憶域を持ち、その他のタイプと次元が一致しなければならない

については、コンパイラや言語仕様が複雑になるのに比べそれ程の使用頻度はないと考える。

(5) プログラムのエラーを減らし読解性を良くするため、使用変数はすべて DCL 文で宣言するか、ラベル接頭語として明示宣言しなければならない。これによりコンパイラも簡単になる。

(6) データタイプの変換は誤り易いものについて禁止する。この制限によりプログラムエラーがかなり減少すると思われる。

(7) 例外処理の機能を保ち、同時にインプリメントが複雑になるのをさけるように、条件についていくつかの制限を設ける。

(8) 条件接頭語を許すと多くのハードウェア上で効率良く実現できなくなるので禁止する。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等で行われるべきであると考えられる。

(9) DO 文は覚え易く、使い易く、エラーをおかしくないように単純化する。

(10) CONTROLLED 属性は、BASED 属性と機能的に重複する面が多いことから削除する。限界値の変数指定は、他の言語機能で代用できないので残す。

(11) REFER オプションは使用頻度や有用性に比べ、インプリメントコストが高いと考えて削除する。

(12) DEFINED 属性は単純列オーバーレイ定義のみ許すこととする。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ので禁止する。

以上のように、サブセット G の仕様の作成にあたっては、使用頻度や有用性よりも、インプリメントコスト、言語仕様の簡明さをより重視しているといえる。この方針に対し、各国から種々の機能強化を要求するコメントが多く出された。その一部は採用されたが(配列へのスカラ代入、READ 文の SET オプション、GET・PUT STRING, FLOAT DECIMAL, 等)、大部分の機能強化要求は否決された。そのうちの主なものを以下に示す。

- (1) LIKE 属性の追加
- (2) MAX, MIN 組込み関数のアーギュメント数を任意とする
- (3) BY-NAME 代入の追加
- (4) HIGH, LOW 組込み関数の追加
- (5) POSITION 属性の追加
- (6) 配列式の追加
- (7) ENTRY 文の追加
- (8) ピクチャの重ね打ち符号の追加

これらの各機能は、使用頻度が高くかつ有用なものであるが、効率的なインプリメントが難しく「なるべく多くの計算機上で実現できること」という目的に照して制限されたものである。

表-9 にサブセット G とフルセット仕様の主要な相違点を示す。

6. リアルタイム機能の言語仕様 (ドキュメント T)⁴⁾

リアルタイム機能については、現在 ANSI の X3 J 1.4 小委員会で審議中であり、その言語仕様はまだ流動的である。その言語仕様をドキュメント T と呼んでいる。

ドキュメント T は、工業用も含めたリアルタイム機能の応用を目的としたものであるが、オペレーティングシステムの基本部を記述すること等を意図しているわけではない。本仕様は PL/I の国際規格の拡張仕様として位置づけられており、PL/I の国際規格の次期バージョンに組込まれる可能性もある。

このような目的に沿い、次のような機能の実現を目指している。

- (1) 多数の PL/I プログラムが共用変数やファイル等を矛盾なく、またデッドロックを起こすことなく共用利用できる機能
- (2) 他のプログラムの実行(タスク)のスケジュールを行う機能

各タスクに、アーギュメントとプライオリティを付

表-10 ドキュメント T の拡張仕様一覧

分類	名前	機能	使用例
属性	PROGRAM	タスクとして実行されるプログラムの名前を定義する	DECLARE F PROGRAM (FIXED, BIT); F: PROCEDURE PROGRAM;
	SHARED	共用オブジェクトの名前を定義する	DECLARE TABLE(10) FIXED SHARED;
	SHARED RECORDS	レコードが共用オブジェクトとして使用されるファイル名を定義する	DECLARE F FILE KEYED UPDATE SHAREDRECORDS;
文	CONNECT	共用変数と外部装置を結びつける	CONNECT BREAK TO (*INTERRUPT KEY*);
	DISCONNECT	共用変数を外部装置から切り離す	DISCONNECT BREAK;
	LOCK	共用オブジェクトをロックする	LOCK (A) READONLY (B, C) THEN A=B+C;
	SCHEDULE	タスクの実行をスケジュールする	SCHEDULE A (X, Y) PRIORITY(3) WHEN(TIME()>=START);
	WAIT	タスクを待ち状態にする	WAIT (EVENT);
	ABORT	タスクをアボートさせる	ABORT (A);
	ABANDONED	共用オブジェクトが ABANDONED の状態かどうかチェックする	IF ABANDONED (X) THEN~
組み込み関数	DELAY	$i \times 10^{**}s$ 秒の時間が経ったかどうかチェックする	DELAY (i, s);
	ADDINTERVAL	指定した時間に $i \times 10^{**}s$ 秒の時間を加える	ADDINTERVAL (t, i, s);
	DATETIME	年, 月, 日, 時, 分, 秒の値を返す	DATETIME ();
	INTERVAL	$i \times 10^{**}s$ 秒が t_1-t_2 となるような i の値を返す	INTERVAL (t ₁ , t ₂ , s);
	PRIORITY	タスクの優先度値を返す	PRIORITY (A);
	RECORD	ファイルのレコードをロックするときに用いる	LOCK (RECORD (F, K)) READONLY (G, J) THEN~
変数	PRIORITY	タスクの優先度を設定する	IF PRIORITY (SON)<5 THEN PRIORITY (SON)=10;

加することができ、さらにタスクを、特別の事象の生起に対応して、生成またはスケジュールできる機能

(3) 指定した事象が発生するまで、タスクの実行を遅らせる機能

(4) 自分自身のタスクや他のタスクをアボートする機能

(5) 非同期の割り込みや、装置からのデータ転送にタスクが応答することができる機能

以上の機能は、特定の時間や、ある時間間隔ごとに割り込みを発生することのできる、タイマやクロックの機能さえ備えてあれば、これ以外の特別な実行時ルーチンや、リアルタイム用 OS 機能等がなくても実現できるよう定義されている。

具体的に、PL/I の国際規格に対して拡張されている主要な概念には以下のようなものがある (表-10)。

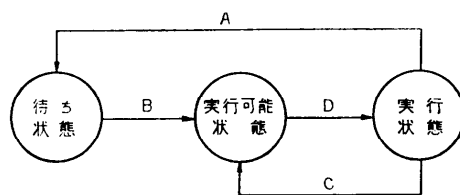
- (1) プログラム: 1つ以上の外部手続きの集まり
- (2) ジョブ: 関連したプログラムの実行
- (3) タスク: ジョブの中の1つのプログラムの実行

(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

PL/I の国際規格では、1つのプログラムの実行しか定義していなかったが、拡張案では、複数プログラムを対象とするためプログラム間にまたがるスコープが必要である。これをジョブスコープとして定義し、



A: WAIT 文は LOCK 文の実行により遷移する

B: 共用オブジェクトのアンロックまたは事象の発生により遷移する

C: タイムスライス切れ, I/O 遅延, 等処理系依存の機能により遷移する

D: CPU が使用可能になった, I/O が終了した, 等処理系依存の機能により遷移する

図-1 タスクの状態遷移

表-11 ドキュメント T における問題点

問 題 点	概 要
システムの保障	実際に稼働しているシステムがないため、仕様の有用性、機能の十分性についての保障がない
定義のあいまい性	SHARED 属性と排他的な属性の定義が不明確
	共用変数、共用ファイルがアークギュメントの場合の定義が不明確
	共用変数と DEFINED 属性の関係についての定義が不明確
	LOCK 文で同一レコードに対し、異なったロック状態を同時に指定した場合の定義がない
	CONNECT 文、DISCONNECT 文で、共用変数がそれぞれすでに結合状態、非結合状態にあった場合の定義がない
機能不足	TIMEOUT 組込み関数のアークギュメントが負の場合の定義がない
	KEYED 属性が SHAREDRECORDS 属性の指定により暗黙に引き出された方がよい
	共用変数の結合状態を知る機能がない
デッドロック	タスク識別子に数値しか使えない
	LOCK 文のクリティカルリージョン内でさらに LOCK 文や WAIT 文を使用するとデッドロックになる可能性がある

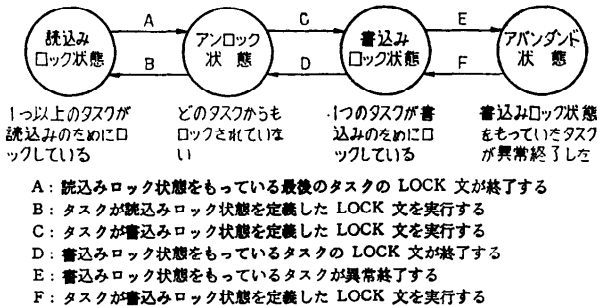


図-2 共用オブジェクトのロック状態の遷移

共用オブジェクトや、プログラム属性が宣言された入口名に与えられている。タスクは、待ち状態、実行可能状態、実行状態の3つの状態を持ち、これらの状態を遷移することによりスケジュールが行われる（図-1）。共用オブジェクトは、共用オブジェクトへのアクセス可能状態を定義するため、いくつかのロック状態がとられる（図-2）。このようなロック状態を定義するのに、LOCK 文を用いており、LOCK 文のクリティカルリージョン*内で、ロック状態のオブジェクトにアクセスできるようになっている。この場合、LOCK 文のクリティカルリージョン内でさらに LOCK 文や WAIT 文が使用されるとデッドロックに入る可能性がある。現在の仕様では、言語機能が弱くなることから、このような文構造を禁止してはならず、なるべく使用すべきでないという表現になっているが、異論のあるところであろう。

ドキュメント T は審議中であるという状況もあって、細部の点で矛盾や不備がいくつか存在するが、標準仕様として定める上での一番大きな問題は、現実に

稼働しているコンパイラがないということであると考えている。標準案として仕様を定めるためには、①その仕様で矛盾がないこと、②その仕様で十分用途がはたせること、③十分の使用（ユーザ）が見込めること、という基本的な事項に基づく評価が必要であると判断し、ANSI に対し他のいくつかの疑問点とともにコメントとして送付したところである。

これらを含めた現状のドキュメント T に関する問題点を表-11 に要約する。

7. 今後の標準化関連の活動について

約 15 年にわたる PL/I の標準化活動も 1979 年の国際規格の制定により一段落した。

しかし、他言語の例にも見られるように言語仕様はその時点における最新技術の反映と利用者の要望等によって変遷するのが常であり、今後とも標準化活動は継続する。

また、国際規格に準拠し日本では JIS 規格を制定するのが通例であるが PL/I についてはまだその計画が明確ではない。

以上の状況をふまえて、今後の標準化活動の見通しと問題点等について私見を述べる。

(1) 自然言語による PL/I 言語仕様の記述

現在の仕様は形式的記述言語を用いて書かれているため、仕様定義上であいまいさがないという利点がある反面、仕様記述が複雑となるため、保守性や読解性等において難点がある。このため、仕様を自然言語で記述すべきとの要望が強く、ANSI では既存のシステ

* LOCK 文の THEM 節

表-13 PL/I 用語の各社比較

社別	傾 向	代 表 的 な 比 較 例
A	日本語中心	基底付き, 被制御: 列: 条件: 引数, 進法, 実数法, 小数点法: 編集に従う入力
B	英語・カタカナ中心	BASED, CONTROLLED: スtring: コンデション: アーギュメント, ペース, モード, スケール: EDIT 入力
C	混 合	BASED, CONTROLLED: スtring: 条件: アーギュメント, 基数, モード, 位取り: EDIT 型入力

ムの PL/I 仕様記述例 (Multics PL/I, Control Data PL/I 等) を調査中である。このような状況から、国際規格の次期または次々期の改訂版においては自然言語で規定される可能性がある。

(2) 拡張機能の検討とフルセット仕様への組み込み方

PL/I の機能拡張についても検討中である。

リアルタイム機能についてはドキュメント T として検討中であることをすでに述べたが、その結果をフルセット仕様へ組み込む方法について言語仕様の統一性などの観点からの検討が必要になるであろう。

また、リアルタイム機能以外の拡張機能として、ECMA の TC 10 で PL/I の会話機能拡張仕様を検討しており⁹⁾、タイプライタ端末を想定した会話型 PL/I 拡張仕様案 (1976 年 6 月 ECMA/TC 10/77/9 "Outline Proposal on PL/I Extensions Supporting Interactive Data Transmission") と、ディスプレイ端末を想定した画面処理 PL/I 拡張仕様案 (1977 年 11 月 ECMA/TC 10/77/17 "PL/I Extensions Supporting Full Screen Processing") が発表されている。

(3) サブセット仕様の検討

汎用目的のサブセットについては、DP 6522 (サブセット G) がまもなく国際規格として制定される見通しである。

サブセット G の検討過程において、サブセットをひとつに限定するのは選択の自由度が低いなどの理由により、COBOL のような水準を設けるべきであるという提案もあったが具体的な活動計画は明確ではない。

(4) JIS 規格に向けての準備

PL/I の JIS 規格が懸案となっているが、COBOL や FORTRAN に比較し言語仕様が大きく多大の工数のかかることが予想される。

一般に、国際規格が制定されている場合の JIS 原案作成の作業としては国際規格の忠実な翻訳とそれに使用する日本語の用語の選定などがある。

翻訳については日本電子工業振興協会の日本語訳がすでに完成しており⁷⁾、その利用を考えるべきであろう。

表-12 JIS 規格で日本語訳の異なる例

英 語	COBOL	FORTRAN	ALGOL
RECORD	レコード	記録	—(注)
RELATIONAL OPERATOR	比較演算子	: 関係演算子	比較作用素
COMMENT (LINE)	注記(行)	注釈(行)	注釈
CONTINUATION LINE	あとの行	継続行	—
STATEMENT	命令	文	文
SENTENCE	文	—	—
ARITHMETIC OPERATOR	算術演算子	算術演算子	算術作用素
IDENTIFIER	一意名	—	名前

(注) 該当する英語が存在しない

一方、用語については PL/I WG において現状の問題点を整理している⁵⁾。そのひとつは既存の言語間で異なる用語があるがそれらについて PL/I ではどの訳語を割付けるかということであり (表-12)、他のひとつは各インプリメンタにより用語に対する基本的な考え方に差異のあることである。どれを採用するにしても決定的な理由が見当たらないようであり、しかも各インプリメンタのマニュアル等のドキュメントへの影響が大きく統一は難航しそうである (表-13)。

8. む す び

PL/I の国際規格化の概略を中心に、PL/I の歴史と標準化の経緯、汎用サブセット (サブセット G) の言語仕様のサブセッティングの考え方とその仕様の概要、および機能拡張項目として最も検討が進んでいるリアルタイム機能について述べた。問題点としては国際規格と現在広く利用されている言語仕様との相違点 (表-8)、サブセット G の問題点 (表-11)、および JIS 規格の用語統一の件 (表-12, 表-13) について指摘した。

なお、JIS 規格の作成の時期については、私個人としては ISO の次の改訂 (ANSI では 2 年間の仕様凍結の方針が表明されているので、早くとも 1982 年以降となろう) の方向が判明するまで見合わせるべきではないかと考えている。理由としては、

① 自然語による仕様の記述

② リアルタイム機能等の拡張機能の言語仕様への反映

③ 自由度のあるサブセット仕様の実現法
などのインパクトを受けて、国際規格の構成や記述法が大幅に変更になる可能性が懸念されるからである。

最後に本稿をまとめるにあたり、PL/I の各種調査報告書を PL/I WG に提供して下さった日本電子工業振興協会の PL/I 専門委員会と PL/I WG において私と一緒に PL/I の標準化活動に従事されている委員の方々に心からお礼申し上げる。

参 考 文 献

- 1) 竹下 亨: プログラム用言語 PL/I の標準化の動向, bit, Vol. 4, No. 4 (1972).
- 2) International Standard ISO 6160 Programming Languages-PL/I (1979-07-01).
- 3) American National Standard Programming Language PL/I General-Purpose Subset, BSR X 3.74, SUBSET/G Revision 8 (Mar. 1979).
- 4) X3 J1/649, Document/T (Revision 3), (May 1979).
- 5) PL/I の用語比較, 情報処理学会規格委員会 SC 5/ PL/I/WG (7月 1979).
- 6) 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 (3月 1979).
- 7) 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 (3月 1977).

(昭和 55 年 3 月 31 日受付)