

解説

3. 標準化の状況



3.4 プログラム構成要素及びその表記法†

長野宏宣†† 東基衛†††

1. はじめに

ここでは、JIS規格として制定中の「プログラムの構成要素及びその表記法^{1),2)}」について、日本からの新しい図記法の提案に始まるISOにおける制定の経過、規格の概要、結果としてJIS規格の「参考*」とした図記法などについて解説する。

1960年代末期のGOTO文論争に始まる構造化プログラミング(SP: Structured Programming)は、大規模ソフトウェアの品質保証、生産性の向上をめざすソフトウェア工学の幕開けであった。ソフトウェアの規模が年々増大し、その品質保証の難しさと開発期間の延伸に悩まされていたソフト開発担当者は、SPに始まる設計法の議論を通じてプログラムを作るとき、分かりやすさにも注意しなければならないことに気づき、現場の作業を変えた。

ベテランのプログラマは、SPの考え方の中に自分の主張の一端を発見し、さらにその延長で何を考え何をなすべきかのヒントを与えられた。さまざまな現場においてSPの具体的な適用実験と方法論の体系化が試行された。その結果は初心者や、汚いプログラムを書く同僚にプログラミングの方法の修正を迫るとき、かなり強力な説得の論拠となり、論争はあったもののソフトウェアの生産現場に広く受け入れられて現在に至っている。

C. A. R. ホーアがE. W. ダイクストラのノートについて「すぐれたプログラマがこれまで意識せずに用いている方法の考え方を明快に説明している³⁾」と述べているとおりである。

CPU能力向上やメモリ価格の低下などによるハー

ドウェアの余力を、高級言語の利用やモジュール化にともなうオーバーヘッドに振り向ける余裕も出てきた時期と一致したこともあって、SPの考え方は提唱者の考えたままではないにせよ、実用レベルに消化されて産業界に定着したといってもよいだろう。

この規格は、SPなどの方法論においてプログラムを構造化する考え方の基本となったプログラムの構成要素(Program Constructs, 以下、構成要素と略することがある)及びその組み立て方などについて規定している。

2. 規格化の経過

2.1 日本からの提案と規格化の対象

SC7国際会議では、1976年に流れ図規格の改正が欧州各国から提案された(この経過については、本特集2.1参照)。しかし当時の国内では、構造化プログラミングの試行とこれを実現するための図式表現⁴⁾がいくつか提案され、流れ図は広く使用されてはいたものの、技術的に過去のものになりかけていた。

このような状況から日本のSC7専門委員会としては、流れ図の改正よりもSPに関連する新しい図記法の規格を作るよう提案することとし、1978年の第7回SC7国際会議(ストックホルム)に、プログラムの制御構造を表現する新しい図式表現の例として、以下の文書を提出した。

- ① Structured Flowchart (NSチャート)、
- ② Compact Chart (その後HCP: Hierarchical and Compact description chartと改訂⁵⁾)、
- ③ Structured Programming Diagram (SPD⁶⁾)。

これらは、それぞれ国内の企業内で使用されていたものについて、規格の体裁を整えたものである。

1979年の投票により、SP関連の規格化の提案が新作業項目(New Work Item)として承認され、まずSC7/WG3(Program Design)において、図記法の規格化の前提となる「SPを意識したプログラムの構成

† Program Constructs and Conventions for their Representation by Hironobu NAGANO (NTT Software Engineering Center) and Motoei AZUMA (Waseda University).

†† 日本電信電話(株)ソフトウェア開発センター

††† 早稲田大学理工学部

* JISの「参考」は規格に含まれない。

要素」を規格化することとなった。これが本解説の対象とする規格である。

新しい図記法については、構成要素の規格化の後に、それを表現する方法として WG 1 (Symbols, Charts and Diagrams) で議論することとなった。実際には、WG 3 の方向がみえた時点から、WG 1 において並行して審議が進められた。

1981 年第 9 回 SC 7 国際会議 (ベルリン) には、英国から

④ Design Structure Diagram (DSD⁷⁾),

ドイツ/オランダから①と同様な Program Structure Diagram (PSD=NS チャート⁸⁾) が提案された。これらは、いずれも現在では提案国の国内規格であり、当時提案国内での規格化が進められていたものである。

さらに 1982 年第 10 回 SC 7 国際会議 (パリ) には、

⑤ Problem Analysis Diagram (PAD⁹⁾),

を国内で使用されている図記法例として追加した。

最後にフランスから、1985 年に、

⑥ Logical Conception of Program (LCP¹⁰⁾) が追加された。

2.2 ISO 規格化の経過

WG 3 におけるプログラム構成要素の規格化内容については、ダイクストラの母国であるオランダが SP の定義の規格化を主張し、他の国は SP という言葉が人や国などによって必ずしも同じ意味では使用されていないとの認識から避けたい意向であり、オランダ/ドイツとイギリス/日本/フランスとの間で長年にわたって対立を続けた。このために作業項目 (Work Item) 名や規格の名称が以下に示すように 2 転 3 転した。

1978 年: Basic Constructs of Program Flow

1980 年: Constructs for Structured Programming and Conventions for their use

1982 年: Constructs for Programs and Conventions for their use

1983 年: Program Constructs and Conventions for their Representation (現在の規格名称)

オランダの代表はほとんどが大学の教授であり主として教育的見地からの規定を望んだのに対して、日本やイギリスは企業の現場で使用する立場であり、実用性を重視したことが、議論のもとであった。

先にも述べたように、オランダは SP の提唱者ダイクストラの母国であり、SP はオランダのものだとい

う強烈な意識があり、SP そのものの規格化と NS チャートの採用を強硬に主張した。

2.3 規格の内容の審議

規格内容の審議は、SC 7/WG 3 の国際会議の場で長時間にわたる議論となった。主要な議論の対象は以下のとおりであった。

- ① プログラム構成要素 (構造) の概念について、
- ② いわゆる制御構造の 3 要素の名称の選定、
- ③ 割り込み (イギリス)、無限ループと脱出 (日本) を含めるかどうか、
- ④ 部分集合の規定について、
- ⑤ Structured Programming (オランダ) という言葉を定義するかどうか。

名称については、以下の章で紹介する。そのほかの事項では日本の主張で、無限ループからの脱出及び異常処理や例外処理などで必要となる“打ち切り (Termination)”を構成要素ではないが、規格に含めることで妥協が成立した。部分集合の規定は、オランダの考えた教育のための純粋な要素の組み合わせをねらっている。しかし、どのような部分集合も許すとしてしまったことから、この部分は規格としてはやや曖昧な規定になっている。

3. ISO 規格の概要

3.1 構成概要

規格は 7 章からなる。各章は、序文 (Introduction) 適用範囲 (Scope)、プログラム構成要素の定義 (Definition of Program Constructs)、構成要素の結合方法 (How Constructs may be combined)、構成要素の仕様 (Specification of Constructs)、打ち切り (Termination)、部分集合の定義 (Definition of Subsets)、及び付属書 (ANNEX) である。英文の原案は、目次を含めて本体が 4 頁、付属書が 9 頁の簡潔な規格である。

3.2 適用範囲

この規格は、手続的アルゴリズムの制御構造を表現する方法について、以下を規定している。

- ① プログラム構成要素の性質、
- ② プログラムを構成する方法、
- ③ 要素のもつ構造についての仕様、
- ④ 定義された要素の集合からさまざまな部分集合を定義する方法。

3.3 規定内容と意義

- (1) 構成要素の概念

この規格は、プログラムが、一般に有限個の種類
のプログラム構成要素を組み合わせることによって表現
できることを前提としている。これは、制御の流れの
組み合わせに着目した SP の考え方を、手続き部を含
めた構成要素という単位を用いて再構成したもので
ある。

この規格では、一つのプログラム構成要素が一つの
手続き部と一つの制御部とからなると定義している。

制御部は、指示といくつかの条件とからなり、指示
の性質と条件の値に従って、手続き部の実行を制御す
る。指示も条件もない制御部を暗黙の制御部といい、
対応する手続き部を無条件に 1 回だけ実行する。

手続き部は一つ以上の演算か空であってもよい。

(2) 整構造プログラムの構成

プログラムの構成法として、ある構成要素の一つの
手続き部を、一つの完全な構成要素によって置き換え
ることを規定している。この方法によって構成したプ
ログラムは、整構造をもつという。

(3) 用語の統一

これまで、自然発生的に企業で使用されてきた構造
の名称には、技術的な整理の観点からみると、必ずし
もふさわしくない用語も見受けられる。

社内規格への反映については、JIS 制定作業の一環
として、各企業の作業標準を担当する方へのアンケート
を実施した。その回答では、本規格の制定にあわせて
社内の名称も変更するという意見が大勢を占めている。
ばらばらであった国内の用語について、この規格
の制定を契機に、以下のように統一される見とおしが
得られたことは、今後のソフトウェア産業の発展にと
っても朗報である。

- ① 基本要素 (imperative construct)
- ② 順次要素 (serial construct)
- ③ 並列要素 (parallel construct)
- ④ 繰返し要素 (iterative construct)
 - 前判定繰返し (pre-tested iteration)
 - 後判定繰返し (post-tested iteration)
 - 継続繰返し (continuous iteration)
- ⑤ 選択要素 (selective choice construct)
 - 単岐選択 (monadic selective)
 - 双岐選択 (dyadic selective)
 - 多岐選択 (multiple exclusive selective)
 - 多重岐選択 (multiple inclusive selective)

これらそれぞれの手続き部と制御部の規定内容につ
いては、自明のものが多く(表-1)ので、ここでは基

本要素と多重岐選択について紹介するに留める。

基本要素は、一つの手続き部と暗黙の制御部 ((1)
参照) からなり、ソフトウェアシステム全体から、こ
れを構成するサブシステム、モジュール、ルーチンな
どと呼ばれるプログラムの単位や、極端には 1 行の
ソースコードまでを意味する。

多重岐選択は、多岐選択が通常の CASE 文に相当
するのに対して、条件付きの並列処理ともいえる構成
要素である。この構成要素を直接記述するプログラミ
ング言語の制御文は、筆者らの知る範囲ではみあた
らない。この構成要素は、ドイツの主張で導入された
が、現実の問題を記述するには便利な場面もあるよ
うだ。

(4) 図記法の現状認識

付属書については、WG 1 において審議されたが、
複数の図記法を統一するための適切な評価基準がな
く、国内で使用されていた図記法を欧州各国の記法と
併記する形となり、標準化は今後の課題とした。しか
し、その後も国内各社では似て非なる図記法が多数提
案され、統一化とは逆方向の状況である。

日本の場合、他の分野についても他人の成果を利用
することには、大変大きな障壁があり、自社内に図記
法のなかったメーカーでは、わざわざ独自のものを作っ
ている状況である。

その中であって、あえて既存の記法を採用している
大手メーカーも存在しており、今後の流れとなることを
願う。

情報処理に関連する方々に、図記法についての問題
を提起し、今後の大きな課題として認識していただ
ければ、この規格の意義も大きい。

4. JIS 制定の経過

従来から、SC 7 に関連する ISO 規格の制定につ
いては、国際規格の作成段階で日本の意見を反映し、ISO
制定の見込みがついた時点で JIS 原案を作成する手順
を採用している。この規格についても同様に、ISO 規
格は、日本の委員が種々の議論に参加し整理してまと
めたものを、事務局 (カナダ) が清書した。

したがって、JIS 原案作成にあたっては、ISO 規格
案をベースに JIS 規格の体裁を整える作業が主体とな
り、主たる議論は用語の選定と、構造の図記法を示し
た ANNEX の扱いについてであった。

4.1 用語の選定

現在国内の大手メーカーで使用されている、プログラ

表-1 プログラム構成要素 (JIS X 0121 による表現)

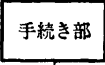
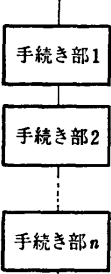
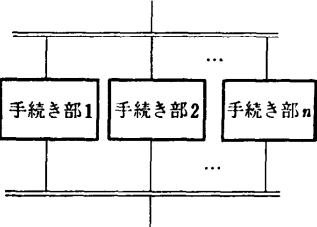



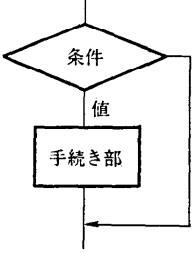
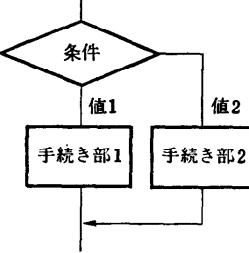
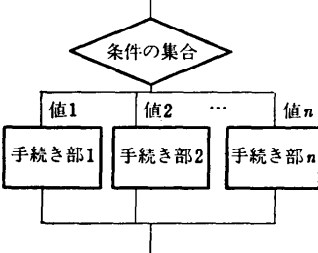
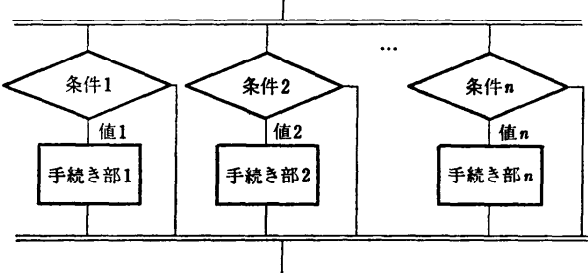
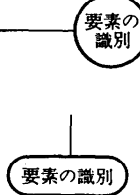
基本要素	順次要素	並列要素
		
<p>前判定繰返し</p> 	<p>後判定繰返し</p> 	<p>継続繰返し</p> 
<p>単岐選択</p> 	<p>双岐選択</p> 	<p>多岐選択</p> 
<p>多重岐選択</p>		<p>打切り</p>
		

表-2 プログラム構成要素の名称と国内各社で使用中の名称

プログラム構成要素名称 (ISO 案)	JIS 原案	HCP の名称	SPD の名称	PAD の名称	YAC II の名称	TFF の名称	文献 3) の用語
imperative construct	基本要素	基本形処理	基本構造	処理要素, 処理	処理要素, 処理	処理	action
serial construct	順次要素	連接構造	順次構造	連接	順次構造体	順次 (展開) 処理	concatenation
parallel construct	並列要素	並列処理	並列構造	並列処理	並列処理構造体	並列処理	...
iterative construct	繰返し要素	繰返し処理	繰返し構造	反復	反復構造体	反復 (展開) 処理	repetition
pre-tested iteration	前判定繰返し	(前判定ループ)	前検査繰返し	WHILE 形, DO 形	前判定反復構造体	前判定反復	while-do
post-tested iteration	後判定繰返し	(後判定ループ)	後検査繰返し	UNTIL 形	後判定反復構造体	後判定反復	repeat-until
continuous iteration	継続繰返し	(無限ループ)	無限繰返し	無限ループ	無条件反復構造体
selective choice construct	選択要素	振分け処理	選択構造	選択	選択構造体	分枝 (展開) 処理	selection
monadic selective	単枝選択	(二分岐の特殊形)	単分岐選択	IF THEN 形	単一選択構造体	条件付き処理	if-do
dyadic selective	双枝選択	二分岐振分け処理	二分岐選択	IF THEN ELSE 形	二者択一型選択構造体	(多重分岐の特殊形)	if-then-else
multiple exclusive selective	多枝選択	多分岐振分け処理	多分岐選択	CASE 形	単一選択ケース構造体	多重分岐処理	case-of
multiple inclusive selective	多重枝選択	条件付き並列処理	多分岐多重選択	...	複数選択ケース構造体
termination	打ち切り	終了記号	終了	終了	...
規格にない構成要素	...	チェック チェック付き処理 チェックアウト 再試行 手続き呼出し マクロ呼出し モジュール呼出し	...	再定義: 一つの処理 要素の詳細を別のと ころで定義する。	反復構造体の拡張/組 み合わせ 反復構造体の途中脱出 反復構造体の途中反復 分岐制御 例外制御 モジュール	条件判断 GOTO ラベル 反復抜け出し 手続き呼出し マクロ呼出し 例外抜け出し	...

ム構成要素の名称を JIS 規格と対比して表-2 に示す。原案作成委員会では、これらにとらわれず技術用語として適切な言葉を選定した。

以下に用語選定における主要な議論点を示す。

① imperative (基本)

単語の意味は、“指令”、“命令”などであり、直訳に近い用語を採用する意見もあった。

一つの演算からプログラム全体までの、あるかたまりを代表する基本的な構造を意味していることを重視して“基本”と意識した。

② monadic selective (単岐選択), dyadic selective (双岐選択), multiple exclusive selective (多岐選択), multiple inclusive selective (多重岐選択)

これらの用語については、簡潔さと全体の統一性を重視した。分かれ道を選択するときの選択の可能性の個数を単, 双, 多によって表現し, 多数の中から複数個を選択する意味で多重という言葉を選んだ。

⑥ termination (打ち切り)

この用語は, JIS 情報処理用語に従った。

4.2 ANNEX の扱い

図記法の事例として ISO 規格の ANNEX にあげられた日 (HCP, PAD, SPD), イギリス (DSD), ドイツオランダ (PSD), フランス (LCP) の 6 つの記法のうち, 国内で利用者のいない記法は JIS から落とし, かわりに国内で最近利用されているいくつかの記法を追加したいという意見が一部の委員から提出された。特に LCP は, フランスが規格の最終案がまとまった時点で参加各国の反対を押し切って強引に追加した事情もあって, JIS では削除する意見も強かった。

しかし, 一部の記法のみを採録する基準の設定が難しく, JIS 規格としては国際規格の表をそのまま採用し, その他の国内の主要な図記法については, JIS 解説で扱うこととした。さらに, 規格に含まれないことを明確にする意味で, ANNEX を JIS では“参考”とした。

5. 図記法のありかた

この規格を制定した日本からの提案の意図は, SP に基づく新しいプログラム構造を表現する図記法が, いろいろ提案され効果を上げているので, これらについて調査し標準化しよう, ということであった。

したがって, 国内から当初提案した SPD, HCP も後で追加した PAD も, 国内の事例を示し参加各国の事例と突き合わせる事が狙いであった。

しかし, 同様な規格を制定しかけていたイギリス, ドイツ, オランダの思惑と, 主としてオランダの SP に対する思い入れの強烈さに引きずられ, 構造の規格を作成した段階ではどれか一つに絞ることはほとんど不可能であることが各国代表委員のコンセンサスとなっていた。その妥協の結果として, 各国の 6 案を構成要素の表記法の事例として ANNEX にまとめた。

その後の国内外の状況を見ると, 年々強力になるパソコンを利用した支援ツール¹¹⁾に限られた範囲での試行段階をすぎて, 商品として販売され実用に供されている。最近では, 複数種類の記法を支援するツールや, 図記法間の変換などが試みられている。一つの図記法に標準化できていけば不要な努力ともいえるが, 現状ではこれらの多様な図記法を是認した上で, 言葉の違いと同様に, 図記法間の相互理解を進めていくのが現実的かもしれない。

とはいえ, ソフトウェア産業が他の多くの製造業と同様に階層化された産業構造となっており, このまま放置してしまうと実際に物作りに従事する人はいくつもの書き方を学習させられることになり, 国家的な損失である。

当初の日本の主張でもあったように, ささまざまな記法をなんらかの公平な機関で評価するなり, 共通的なところを抽出して新しい共通記法を標準化するなどの方策によって, 実際の利用者にとって便利で役に立つものにしていく努力が望まれる。

6. おわりに

GOTO 文論争から 20 年が過ぎ, 当時第一線のプログラマとして活躍していた人も, ほとんどが現役を引退する時期であろう。その間, パソコンの急激な普及の初期段階では, ハードウェア資源の不足を機械語とテクニックで補う汎用機の初期の状況が再現して, 識者を嘆かせたりもした。しかし最近では, パソコンプログラマの間でもオブジェクト指向などが話題になるなど, 全体の技術レベルが数段向上したのは確かであろう。

この時期に, プログラムの構成要素の規格がようやくできてくるのはいかにも遅いが, これだけの合意のためにも 10 年を要したという見方もできよう。国際間や, 国内各社の思惑がさまざまに影響して難産の末の妥協による産物ではあるが, 少なくともソフトウェア工学の基本となるプログラムの制御構造の概念と名称とを規格化した意義は大きい。

国際規格の審議の段階で日本から複数の事例を提出していることが大手商業紙の記事になったこともあって、関係者の中には迷惑を被った方もいたようであったし、ことの経過を事実として残しておく意味で、整理してみた。

多数の図記法がすでに存在し、各メーカーのメンツにさえもなってしまった現在では、一つの記法への統一はなかなか難しい。本文中でも触れたように、それを乗り越える技術的な試みも出てきているのでこちらに期待を抱きたい。

最後に、常々ご指導をいただく SC 7 国内委員会の菅主査 (学習院大)、国際会議の場で協力していただいた黒田委員 (三菱電機)、松山委員/村上委員 (富士通)、及びエキスパートとして WG に参加していただいた二村氏 (日立) をはじめ、終始ご支援いただいた関係各位に感謝します。

参 考 文 献

- 1) ISO/DIS 8631-Information Processing-Program Constructs and Conventions for their Representation (1986).
- 2) 日本工業規格 (案) プログラム構成要素及びその表記法, 同規格原案作成委員会 (昭和 62 年 3 月).
- 3) Dahl, O.J., Dijkstra, E.W. and Hoare, C.A.R.: Structured Programming, Academic Press(1972) (邦訳: 野下, 川合, 武市共訳: 構造化プログラミング, サイエンス社, 1975).
- 4) 佐藤: プログラミング用ドキュメンテーション, 情報処理, Vol. 22, No. 5, pp. 383-389 (1981).
- 5) 花田: プログラム設計図法, 企画センタ (1983).
- 6) 水野, 東: 管理者のためのソフトウェア標準化, コンピュータレポート, Vol. 18, No. 8~Vol. 19, No. 1 (1978-79).
- 7) BS 6224: 1982 Guide to Design structure diagrams for use in program design and other logic applications.
- 8) DIN 66 261 Sinnbilder für Struktogramme nach Nassi-Shneiderman (Nov. 1985).
- 9) 二村他: 問題分析図によるプログラムの設計と作成, 情報処理学会全国大会論文集, pp. 269-270 (1979).
- 10) Z 67-102 Cahier des Charges des Applications de Gestion.
- 11) 原田 (編): 構造エディタ, 共立出版 (1987). (昭和 62 年 5 月 15 日受付)