
 巻 頭 言

「構造的…」と「いかにして…」

和田 英 一*

構造的プログラミング（以下「構造的…」と略す。）が構造的にプログラムを作ることか、構造的なプログラムを作ることかから始めて「構造的…」とは何かに関し、ソフトウェア界は内外あげて喧々囂々、侃々諤々である。大方はこのにとなを無意識のうちに同時に意味しているように見える。新説は先般のプログラミング・シンポジウムでの某先生のそれで、「構造的…」とは Dijkstra の旗のあとを多勢でゾロゾロついてゆくもので、その旗とは Dijkstra がお金儲けのために書いた例の本**と、その本に書いてあるかれの趣味とは全く相反したプログラムの作り方だというのである。

さて例の本を通読された本誌の読者諸賢も、こんなにうまくプログラムの書ける筈がないと思われたに違いない。実際構造的に構造的なプログラムを作ることにはある程度行われているし、それでも我々は大いに苦勞しているのだから。それにしても例の本のプログラムの例題はかなり明解であり、そこに何か秘密のありそうな気がする。これもすでに世間に知れわたった話題だが、Wilkes 先生が去年の IFIP 大会で、良い教科書は構造的にできているが、最初を書くときは決して構造的にはできないといわれたそう。そこでこう考えるのだからいかにであろうか。物の構造は一旦みつかってしまえば、その構造を使って他人に説明（したり自分で納得）するのに大変役だつ。しかし構造をみつけるのはかなり大変である。例の本の例題は、問題の構造を非常によくおさえて説明しているので明解なのであり、本には書いてないがその構造をみつけるまで Dijkstra 先生は何回も試行錯誤しているであろう。教科書も執筆時間の大部分は主題のわかりよい構造をみつけるのに使われ、何度も書き直されたあと、読者の目にふれるようになるのであろう。

筆者は最近、「構造的…」はプログラムを確信したりさせたりに有効ではないかと思っている。むしろ「構造的…」でプログラムを書きあげてからどうしてそう

書いたかを反省し、問題はこんな構造をしていたかと気がつくことしばしばである。構造の発見とプログラム書きとはコンカレントでもあるようだ。もちろんすでに熟知の問題と同様の構造をもっている問題の場合は構造発見の部分は省略でき、例えばコンパイラなど何回も作れば、新仕様が追加されない限りどんどん容易になるのは当然である。またさらに新しい構造を発見する機会もあって、あとのコンパイラは大体まえのより優れているようだ。

ところで 3 年前の IFIP, TC 2 のある会合で例によって「構造的…」の議論が始ったとき、ポーランドの Turski さんが、プログラムの書き方？ Polya 先生の「いかにして問題をとくか（以下「いかにして…」と略す。）」***でそれはすべてだと怒鳴ったのを忘れない。それでその後読みかえてみたのだが、衆知のようにこの本には主として数学の問題をとくための注意がリストしてある。極端に要約すれば「いかにして…」の答は、1) 問題を理解せよ、2) 計画をたてよ、3) 計画を実行せよ、4) 結果をためせ、で、これも大いに役だつと思う。特に 1) と 2) の未知のものは何か、条件を満足させようか、似た問題を知っているか、問題をいいかえることができるかなどがゆきつもどいつ構造をみつける段階に似ているようだし、また 4) の他の問題にその結果や方法を応用することができるかは、プログラムをなぜそう書いたかを反省し、次に類似の問題に出会ったときに利用できるようにするのと同じ精神であろう。しかし考えてみると「いかにして…」も手法としては「構造的…」と同じできわめてあたりまえのことしか書いてない。これらを読んでもプログラムが左程容易にならないのは、あたりまえのことを実行するのが困難なのと、プログラミングが本質的にむづかしいからかも知れないと思う。「構造的…」と「いかにして…」をプログラミングにどう活用するかもう少し考えてみたい。

(昭和 50 年 1 月 23 日受付)

* 本会前理事 東京大学工学部計数工学科

** Dahl, Dijkstra, Hoare: Structured Programming (Academic Press)

*** Polya (柿内訳): いかにして問題をとくか (丸善)