

4M-2 自由な形式の表表現仕様とそれからのソースコード生成

奥野 浩 智

旭化成工業(株)経営計画管理部

松本 秀 樹

旭化成情報システム(株)オープンソフト事業部

浅井 宏 則

1. はじめに

ソフトウェアシステムの開発において、世の中に提供されている開発方法・手順は、基本設計⇒機能設計⇒詳細設計、又は、システム分析・計画⇒システム設計⇒プログラム設計を行って、⇒プログラミング⇒テストを行うものが多い。しかし、このような工程で作られるドキュメントは、発注者が確認すべき外部仕様書なのか又はシステム開発者側で用いる内部仕様書なのかがはっきりせず且つ解り難いものになり勝ちである。この問題に対して、視覚的プログラミング環境の発展がある。その視覚的言語の1つとして、行列又は表の形式を用いるものがある。代表的なものとしては、JIS に制定されている決定表[1]がある。しかしながら、これらには大きな欠点がある。それは、表の形式が固定的であることである。通常ある問題の論理・計算処理を考え、整理する場合、人は、問題に適した思考の流れと形式で整理する。この整理した結果に固定形式の既存ツールを用いようとすると、頭脳にある論理・計算処理の構造を、再構成することが必要となる。この論理の再構成の作業は面倒であり、記述もしづらいわけである。当然、論理の流れが不自然になるので、それにしたがって、記述したものが利用者（発注者）には解り難くなる。そこで、問題に適した思考の流れと形式で整理された論理・計算処理の仕様を、外部仕様書としてそれに一番適した自由な形式の表に記述し、そしてそれから、そのままプログラムモジュールが生成出来ることが望ましい。筆者らは、これを実現する方法を考案し[2]開発したので報告する。

TableSpec: Free-format Specification Table and Source Code Generation
 Hirotoimo Okuno, Hideki Matsumoto, Hironori Asai
 Asahi Chemical Industry Co. Ltd.
 1-1-2 Yuraku, Chiyoda, Tokyo 100, Japan

2. 表の構造と表現力

表を用いる論理の表現は、日常よく使われている。この表の構造を整理してみると、基本として、JISの決定表の用語でいう、何らかの場合分けを示す条件記述と、それに対応する動作記述とがある。しかしこれだけでは、表の表現には不十分である。この他に、入力、出力に関わる記述等、表の論理構造を明確にする各種の要素がある。これを整理すると表1のようになる。

手続きのアルゴリズムの制御構造を表現する規定に、JIS X 0128[3]があり、これに基づいたプログラム構成要素とその記述法は、各社から、視覚的プログラミング環境の木構造図CASEとして提供され、良く使われている。各社の木構造図CASEの図表現の互換を図るため、そのツール間のデータ交換言語DXLが検討され、実現されている[4]。上述の各CASEツールは、JIS X 0128にプラスαの機能を付加しており、このプラスαも吸収しているDXLの要素を、自由な形式の表で無理なく表現出

表1 表の構成要素 (Element of Table) □ が基本構成要素

1	表題
2	条件記述領域 <ul style="list-style-type: none"> 縦インデックス欄 <ul style="list-style-type: none"> 縦インデックス名 縦インデックス条件値 横インデックス欄 <ul style="list-style-type: none"> 横インデックス名 横インデックス条件値
3	出力名
4	動作記述領域 <ul style="list-style-type: none"> 条件付オペレーション領域 無条件オペレーション領域 <ul style="list-style-type: none"> 前処理オペレーション領域 後処理オペレーション領域
5	外部参照名
6	注釈項

来るのであれば、これは、固定様式の表表現の欠点を解消して、木構造図表現の利点を生かした、手続き的アルゴリズムの制御構造を記述する視覚的プログラム言語となるわけである。両者を対応させて検討すると、DXLの構成要素は、表の構成要素により表現されることが解る。

3. 自由な形式の表からのソースコード生成法

さて、この発注者、利用者に解りやすい表で表現された外部仕様書から、人手による再記述を経ることなく、プログラムが生成出来れば、有用であるが、任意の形式の表表現の仕様書を、システムがそのまま理解し、そこから直接自動生成することは、現在では非常に難しい。しかし、この自由な形式の表の各要素部に、人が識別記号を付加することにより、表の形式についての規約を、表の作成後に規定することが出来る。この考えかたに基づいたツールがTableSpecである。このTableSpecの使用手順を説明して、概略機能を説明する。

(1) Software Engineer(SE)は、各種分析手法を適用しながら、外部仕様書をワードプロセッサ(WP)を用いて記述する。この外部仕様書の中に、論理・計算処理仕様を、自由な形式の表に表現して記述する。外部仕様書と表に記述する業務用語は、TableSpecの持つ用語辞書から引用する。

(2) TableSpecで、このWPで作成された外部仕様書をコンピュータに読み込んで、CRT上に表示する。

(3) 表示された表に対して、SEは、マウスとメニュー操作により、表の構造を指定するために、表の各要素に識別記号を付加する。表の各要素と付加する識別記号の種類は、表1の「表の構成要素」に対応している。

(4) SEが、識別記号の付加が意図した通りかを確認して指示を与えると、TableSpecは、表の構造の論理チェックをし、上記の識別記号と表の内容を解析して、ソースコード生成に必要な内部管理テーブルを作成し、それに基づいて指定された言語のソースコードを、サブルーチン又はコピーテ

キストの形で生成する。

4. おわりに

このツールの適用対象は、表に表現出来る論理・計算処理仕様であれば、通常の事務アプリケーションだけでなく、技術計算アプリケーション、又基本ソフトウェア類にも適用出来る。そして、これを適用することによって得られるメリットは以下の様なものである。

(1) 任意の形式の表により、解りやすい外部仕様書が記述出来るので、従来は難しかったシステムの発注者・利用者が外部仕様書を理解し承認する事が、容易に出来る。

(2) この外部仕様書からソースコードが自動生成されるので、従来は困難であった外部仕様書とプログラムの完全一致が、保証される。

(3) これにより、外部仕様書を利用部門で容易に管理できるので、今後拡大する各部門での分散して行われる開発・運用が、安心して任せられる。

本ツールは、表表現された論理・計算処理仕様を、サブプログラムまたはコピーライブラリとして生成するものであり、これをプログラムとして動作させるには、必要なI/O部と制御部を持ったプログラム構造にはめ込む必要がある。しかし、本ツールは、いわゆる既存文化の環境でも、既存プログラム中の業務ロジックの部分に適用して上記の効果を上げることが出来る。さらに統合CASEと連携して更に有用なものとして運用することが可能となる。

参 考 文 献

- 1) JIS X 0125-1986(1991確認) 決定表.
- 2) Okuno, H: Information Processing Apparatus and Method Therfor: PCT/JP93/00095.
- 3) JIS X 0128-1986(1993確認) プログラム構成要素及びその表記法
- 4) 長野他: 木構造図用CASEツール間のデータ交換言語: DXL, 情報処理, Vol. 35, No. 4, PP 341 ~349(Apr. 1994)