

# レポート作成システム

7X-5

岡 浩

(日本電気(株))

## 1. はじめに

最近の金融機関を中心としたソフトウェア開発の費用は、莫大なものがあるが、この内かなりの部分は、帳表編集作業が占める。複雑多岐な銀行帳表編集プログラムの自動生成と、その為の帳表記述専用言語を作成したので、その概要について報告する。

## 2. 設計の指針

この言語の設計の指針は、次の様なものであった。

(1) COBOL(等)で作成する場合に較べ3倍以上の生産性。(2) 現存銀行帳表を調査の上その80%以上が記述可能。(3) 帳表のレイアウトを思い浮かべるのと同じ思考で簡潔に記述できる。(4) 日本語帳表を基本とする。

## 3. 言語

言語の骨格となる、帳表の組立て方法としては、概略以下に示す構成単位を設定し、1本の分類軸(時間軸)を持つ入力データが、これらに対応づけられるものとした。(構成単位上大きいものから示す。)

### (1) ページ及びユニット

時系列上逐次的に並べられるページ単位で、ユニットは、ページ内に入出力関係でレイアウト単位を持つものである。

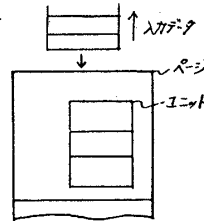


図1. ページとユニット.

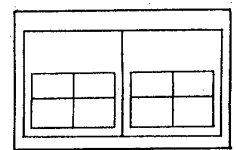


図2. ユニットの構成単位.

### (2) ウィンド

時系列上並列に並べられる部分を、入力データとの対応でまとめたもので、改行や改ページ動作は、ウィンド毎に独立に行われる。(例えば、入力データ中に入金情報と出金情報が混在し、各々を別々の帳表に印字するかのような効果を持たせながら1本の帳表とする等。)

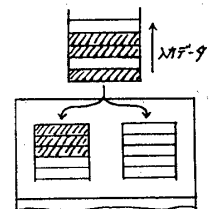


図3. ウィンド

### (3) グループ

入力データの制御グループに対応するレイアウト単位で、 $n$ 種  $m$ 個のブロックから構成される。グループは、複数のページに渡るものから、単一ブロック逐階層構造を持つ。

### (4) ブロック

改行(及び改欄)の単位で、大きくは固定ブロックと可動ブロック(相対改行、改欄を含むもの)に分類し、固定ブロックは、特定のページ、ユニット等の印字時刻をとらえて、ページやユニット内の絶対位置に印字されるもの、可動ブロックは、繰返し印字データとして前印字位置から相対移動するものとした。

### (5) フィールド

Report generation System  
Hiroshi Oka  
NEC Corporation

1ブロックの中には、任意個数のフィールドを任意の位置にレイアウトすることができ、フィールドは、一般的には入力データの1フィールドに対応し、帳表を構成する最小単位である。フィールドは、縦横任意の大きさを持つとした帳表構成単位のレイアウト指定方法としては、行、欄及びその組み合わせとした。

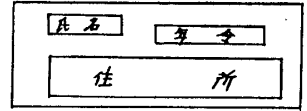


図4. プログラムの例.

例. 3行目14欄へ印字. 2行改行し3欄右へずらして印字. 等.

文字列データのレイアウトについては、任意の形の印字範囲を指定しておくことにより、自動的に範囲にあてはめて

印字が行われるものとした。また、文字列の長さが変化するに伴い、印字範囲も動的に変化(拡張, 縮小)可能とした。



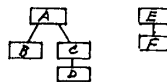
図5. 印字範囲への動的な例.

その他、印字条件、導出データ、関数等を加え、非手続き型の言語としてまとめた。図6に記述例を示す。

4. 入力データ

当言語では、入力データから帳表レイアウトへの対応づけが、ほとんど非手続き的に行われる為、入力データの持つ論理的構造(階層関係, 包含関係等)が重要な意味を持つ。次のリスト形式で表現される木構造を扱えるとした。

例. A(B,C(D)), E(F)



5. コンパイラ

コンパイラは、帳表生成論理を内包するCOBOLプログラムジェネレータとした。コンパイラの特徴及び作成上の留意点は次の様なものである。

- (1)巨大なオブジェクト母体と、それからの生成時の最適化方法。
- (2)ソースとオブジェクトの隔絶と、その間の中間言語変換方式。
- (3)構文解析については、遷移図からの自動生成とした。

6. 評価等

当言語の作成は、従来のコンピュータ帳表が、装置(プリンタ)の制約により、行単位の打下し形に終始し、必ずしも人間的な親しみよい形とはなっていなかったものを、ページプリンタの普及とも相応する形で、ページ単位の自由なレイアウトの編集を行うプログラムを簡易に作成可能とすることを狙ったものであったが、実業務への適用を通じて、概次の様な効果が得られている。

- (1)プログラムの大きさ(行数)は、COBOL比1/10から1/40前後。
- (2)調査範囲内の銀行帳表については、すべて記述できた。
- (3)性能については、計算値、実測値ともCOBOLでの作成値(推定)の数10%増の範囲内にある。

また、使用者や関係者各位からは、良い評価と好意を持って迎えられている。

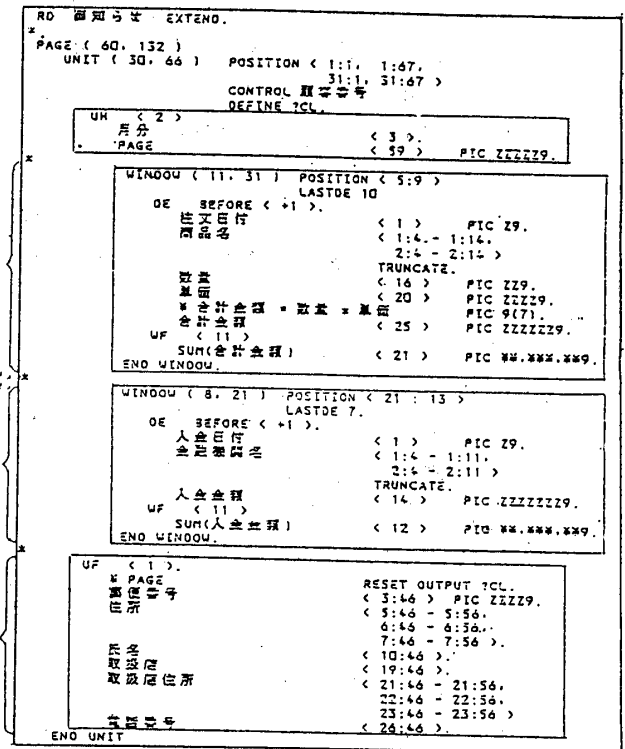


図6. 記述例