

7X-4

オフィスシステムにおけるコマンド言語サポート方式

村地 隆志 岡崎 正一 森吉 国治

(三菱電機東部コンピュータシステム) (三菱電機)

1. はじめに

オフィスシステムにおける業務処理内容は、システムの大規模化、広域化に伴って複合化、高機能化する傾向にある。また処理の対象となる情報の種類もコード化データからイメージ情報まで幅広い。このように高機能化したオフィスシステムを、エンドユーザのレベルで容易に使いこなせることはシステムが提供する主要な機能の1つになっている。

その実現方法の1つとして、従来のオフィス業務処理に、コマンドを組み合わせる強力なツールをプロシジャレベルで構築できるコマンド言語機能を融合させる方法がある。

本稿では、コマンド言語機能をオフィスシステムであるMELCOM 80システム30/40上で実現した背景とその実現方式について紹介する。

2. システムの特徴

コマンド言語機能をオフィスシステムの開発および運用上で役立てている本システムの特徴としては

- ・ ユーザサイドで手軽にプロシジャを作成、加工でき、簡単に強力なツールが作れる。
- ・ バッチ処理指向のジョブ制御コマンドと、操作性のよいTSS処理指向のコマンド言語機能を融合して使うことができる。

がある。

3. 実現方式

(1) ジョブの概念とプロセスによる管理

従来のオフィスシステムでは処理の単位をジョブあるいはジョブステップとしてとらえてきた。一方手軽にフロー制御機能を用いてプロシジャを作成しようとするれば、コマンド(プロセス)を処理の単位として考えるのが便利である。

本システムでは、図1に示すようなジョブとプロセスの対応づけを行っている。ジョブはジョブスケ

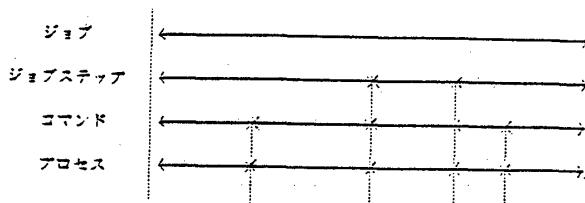


図1. ジョブとプロセスの対応

ジュラが管理し、ジョブステップはジョブステップスケジュラが管理している。一方、ユーザとのインタフェース処理を司るコマンドインタプリタはジョブステップスケジュラが行い、

ジョブステップスケジュラ = コマンドインタプリタ

と対応づけを行っている。

ジョブステップスケジュラがサポートしているコマンド言語の機能は一般的なものであり、コマンドの終了を判定して、その真偽で処理の流れを変えることを特徴としている。

たとえば、

```
IF コマンドA
THEN コマンドB
ELSE コマンドC
FI
```

では、コマンドAが正常に終了するとコマンドBを正常に終了しなければコマンドCを実行する。これは、システムのプロセス間レベルで終了コードの受け渡しを行うことにより実現している。すなわち、コマンドA(プロセスA)からの終了コードが真の値か偽の値によって処理の流れを変える。この処理の流れを図2に示す。

ジョブステップスケジュラは、コマンドを読み込むと、構文解析木と呼ぶツリー構造を形成する。構文解析木は多数の種類を持ち入力されたコマンドの組み合わせにより、それに対応したツリー構造を

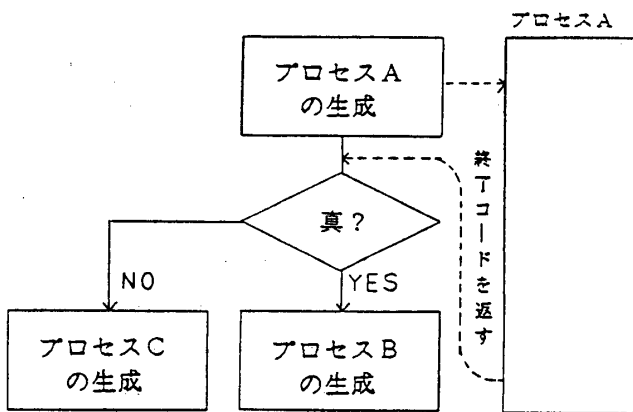


図2. コマンド言語の実現方式 (IF文)

作成する。前記例では、コマンドA (複数のコマンドから構成されて可) の構文解析木を最初に作成して処理の流れを判定する。

構文解析木はノードと呼ぶ構造体から形成される。単一のコマンドのツリーをCOMノードと呼び、ノードの最小単位となる。複数のコマンドより構成されるコマンドの場合、COMノードを下位レベルのノードとして持つツリーを形成する。

たとえば、IF文はIFノードと呼ぶ構造体に、IF、THEN、ELSEのそれぞれのCOMノードへのポインタを持つ。前記例の場合のIFノードの構造を図3に示す。

(2) ジョブ制御コマンドとコマンド言語

バッチ処理指向のジョブ制御コマンドの体系に、TSS処理指向のコマンド言語機能を導入する場合に注意しなければならないのは、フロー制御コマンドの取り扱い方である。

ジョブ制御コマンドでフロー制御機能を提供するJUMP、SKIP、TAG、BREAKなどのコマンドと、コマンド言語でフロー制御機能を提供す

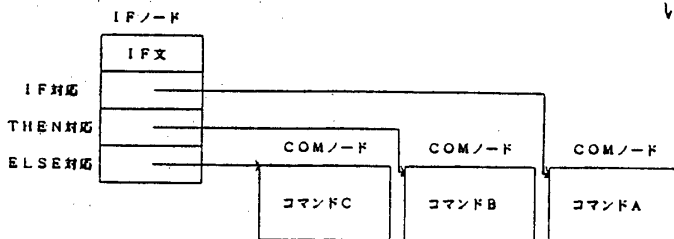


図3. IF文の構文解析木例

るIF、WHILE、FOR、CASEなどのコマンドをジョブステップスケジューラが区別して管理するために、本システムでは、2つのモードをもうけジョブ起動時にどのモードを使うかを指定する方法を採用した。

ジョブ制御コマンドによるプロシジャは、たとえば次のように作成できる。

```

JOB ID-ABC, NOCL
.
.
JUMP TAG-A, USW-CELL
.
.
TAG A
  
```

一方、構造化プログラミングが実現できるコマンド言語機能のフロー制御では次のようなプロシジャを作成できる。

```

JOB ID-XYZ, CL
IF TEST -F A, C
THEN COBOL SM-A, C
LINK OM-A, O
ELSE TYPE 'NO FILE'
FI
  
```

4. 今後の課題

本稿では、オフィスシステムにおいてTSS処理指向のコマンド言語機能を実現した方法を紹介したが、ユーザインタフェースのレベルで高機能かつ操作性のよいコマンド言語を提供することは重要な課題である。

今後の課題としては、簡易言語のレベルからコマンド言語を簡便に使用できる方式について検討していきたい。