

簡潔なコマンド表現を採用した文字列編集プログラム“HQED”

“HQED” a Text Editor with Simple Command Expressions

最近、TSSの利用拡大とともに、使いやすく高性能なTSS処理の実現が望まれている。特に、TSSによる作業量の多くを占める原始プログラムなどの文字列データの編集機能に対する要求は、日増しに増大の傾向にある。

日立製作所はこのような要求にこたえるために、VOS 3 “HQED”を開発した。“HQED”には次の特長がある。

- (1) 簡潔なコマンド形式を採用している。
- (2) 特殊文字(記号)を用いて文字列を簡略に表現できる。
- (3) 使用者の習熟度に応じて、機能を選択できる仕組みがある。
- (4) 編集処理の高速化を図り、多端末での同時利用を可能としている。

本論文では、これらの特長と構造について解説する。

青島健一* Ken'ichi Aoshima

旭 寛治* Hiroharu Asahi

浜 正章** Masaaki Hama

1 緒 言

TSS(Time Sharing System: タイムシェアリングシステム)の急速な拡大に伴い、その利用目的も多様化の道をたどっている。その顕著な例として、従来原始プログラムの編集が主目的であったテキスト編集作業も、文書や入力データなどの一般の文字列データの編集へと利用が拡大されている。例えば、大学や研究所を中心として論文などをTSSで作成する場合に、テキストエディタが利用される例が挙げられる。

このため、従来のような原始プログラムの編集を中心とした機能しかもたないテキストエディタでは不十分であり、より強力な文字列編集機能をもったものが要求されてくる。

一方このようなTSS業務の質の変化とともに、量的にも大きな変化が現われている。すなわち、TSSの規模は急速に拡大する傾向にあり、1システムで200~500端末を支援する要求が出てくる状況にある。

“HQED”(“Hitachi Quick Editor”)は、以上のような要求にこたえとともに、既存のVOS 3 (Virtual Storage Operating System 3) TSSとの整合を考慮して開発したテキストエディタである。

“HQED”は昭和55年3月に最初に市場に出され、版を重ね出荷を継続している。以下、“HQED”の特長と構造について紹介する。

2 “HQED”の特長

VOS 3 TSSのテキストエディタとしては、既に提供されている標準のEDITコマンドや、ビデオ端末を用いるDESP (Display Editor for Structured Programming)がある。

これらに加えて開発した“HQED”は、特に文字列による編集機能を重視したテキストエディタで、簡潔な表現をもつコマンド群により、原始プログラムのほか一般の英文テキスト(論文・手紙・事務用文書)を編集するのにも適している。文字列データに対する検索機能も、他のテキストエディタに比べて豊富である。また、行編集形のテキストエディタであるため、ビデオ端末だけでなくタイプライタ端末でもすべての

機能を使用できる。

“HQED”は、特に次のような条件をもつユーザーに導入されると有効である。

- (1) 英小文字を含むテキストを扱う。

“HQED”は、英小文字も大文字と同様に扱えるので、論文や文書、手紙などを編集するユーザーに適したエディタである。

- (2) 多数の端末をかかえている。

“HQED”は高能率な多重処理を実現するための専用の制御方式を採用しており、比較的少ないシステムの負荷で効率よく会話的な編集が行なえる。特に、公衆回線で接続された多数のタイプライタ端末をもつユーザーには、行編集形のテキストエディタである“HQED”を利用する効果は大きい。

- (3) 習熟度の異なる端末ユーザーがいる。

大学のように、“TSS”に慣れているユーザーと初心者とが混在している場合に、習熟度に応じて機能の切替えができる“HQED”が適している。

- (4) 文字列データを組み合わせて検索したい。

“HQED”では、文字列表現機能を利用して、例えば「ABCで始まりXYZで終わる文字列」とか、「Dog又はdogを含む文字列」を探せ、というような簡単な検索ができる。

以下に、“HQED”のこれらの特長について解説する。

2.1 簡潔なコマンド形式

TSSの使いやすさを決める重要な要素の一つに、操作を指定するコマンドの表現形式がある。本節では、“HQED”のコマンドの表現形式の特徴と、とりわけ行編集形のテキストエディタのコマンドを要素として、重要な行アドレス(編集点の位置や範囲を表わす情報)部の表現形式の特徴について解説する。

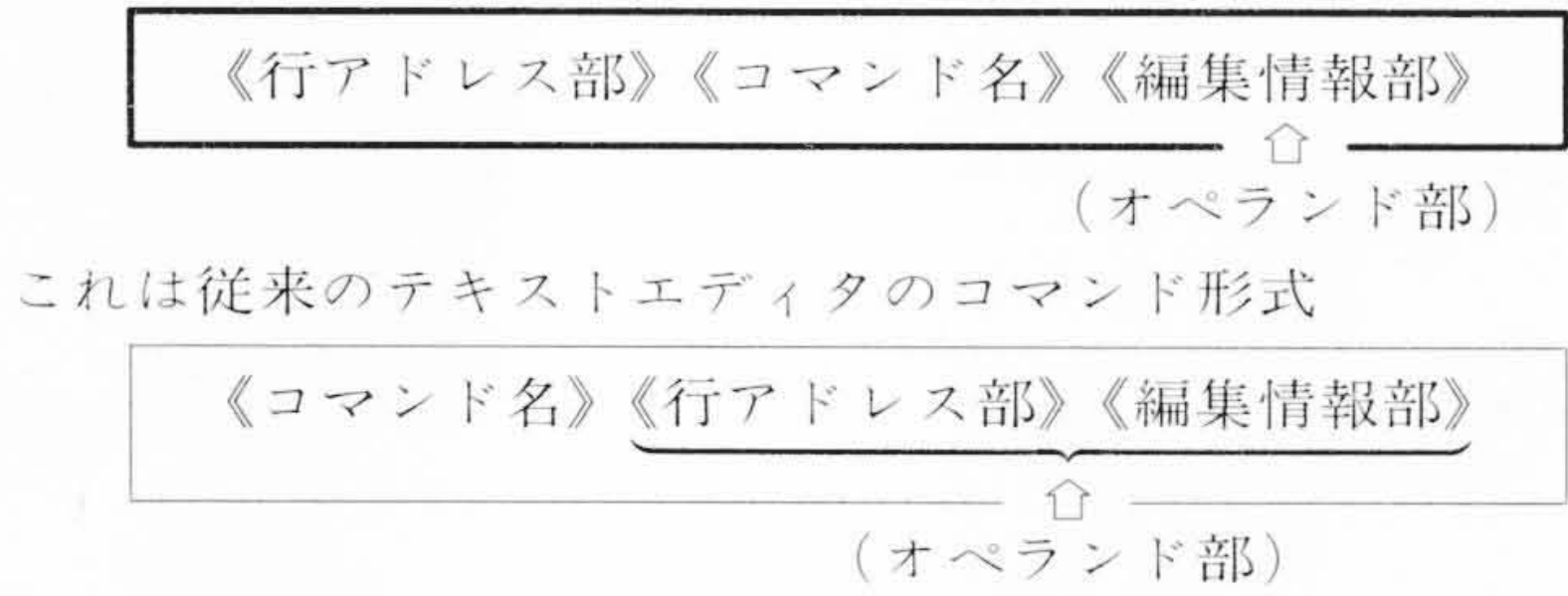
- (1) コマンドの形式

使いやすいコマンド表現を実現するための要素として、次のものを挙げることができる。

- (a) TSS端末でのキータッチ数

* 日立製作所ソフトウェア工場 ** 日立ソフトウェアエンジニアリング株式会社

- (b) コマンド及び操作手順の覚えやすさ
- “HQED”では、(a)を最少にするために次の方策を実施している。
- (i) コマンド名は、1～2文字とする。
 - (ii) 一度指定した文字列を記憶し、同じものを用いるときには、投入を省略可能とする。
 - (iii) テキストの表示機能のように、よく使用する操作を極力簡単な操作(送信キーだけを押す。)にする。
 - (iv) 複数のコマンドを一度に続けて指定可能とする。
- また(b)の方策の一つとして、コマンドの形式を次のようにした。



- と異なっており、次に述べるような特長がある。
- (v) 編集点を指定する行アドレス情報と編集内容の詳細情報とを分離することによって、オペランド部の指定規則が簡潔になる。
 - (vi) 編集の手順(思考手順)とコマンドの表現手順が一致するので操作が行ないやすい(図1)。
 - (vii) コマンドの省略指定が容易となる。例えば、行アドレ

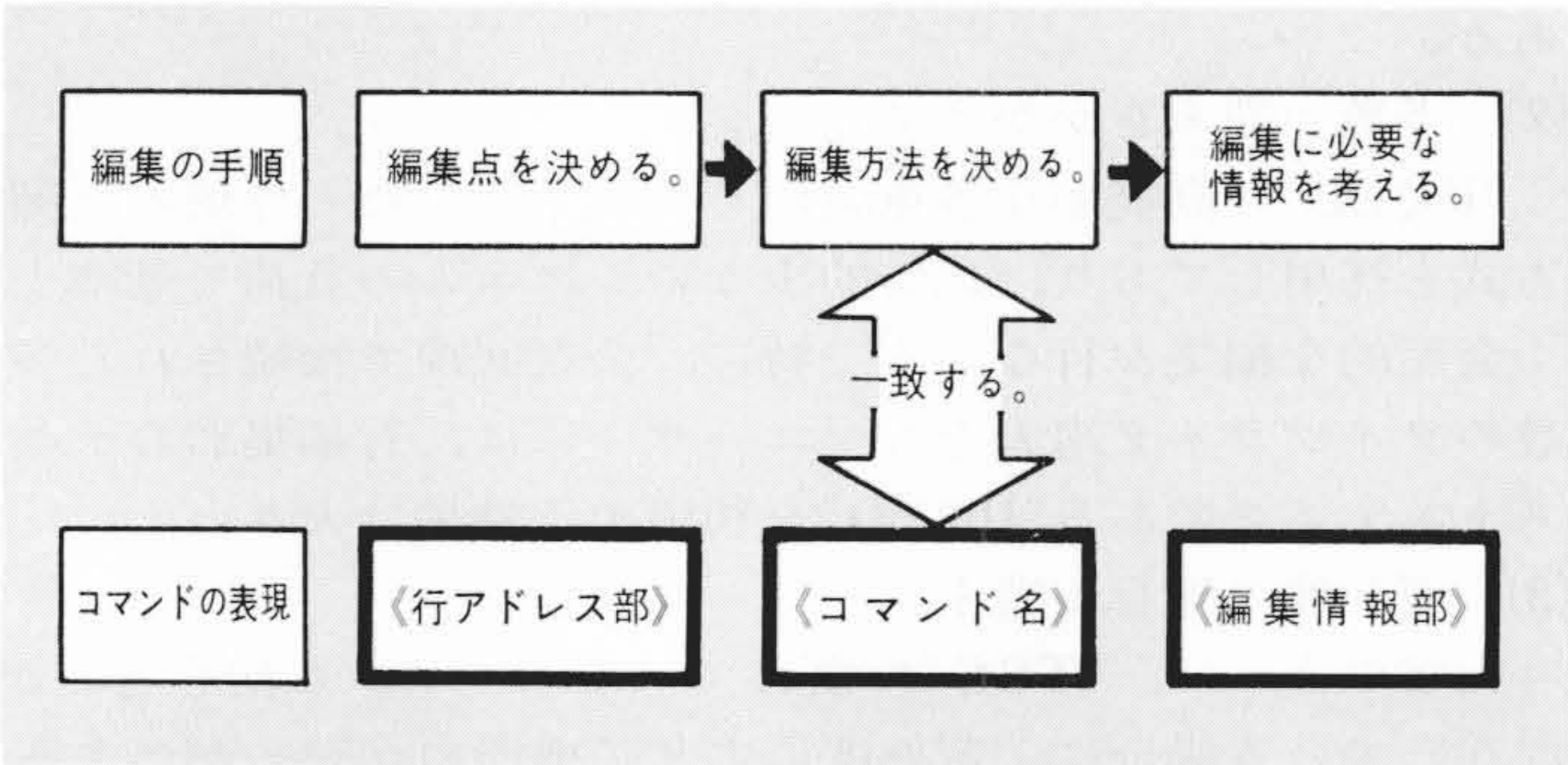


図1 編集手順とコマンド表現との一致性 思考過程に沿ってコマンドの投入ができる。

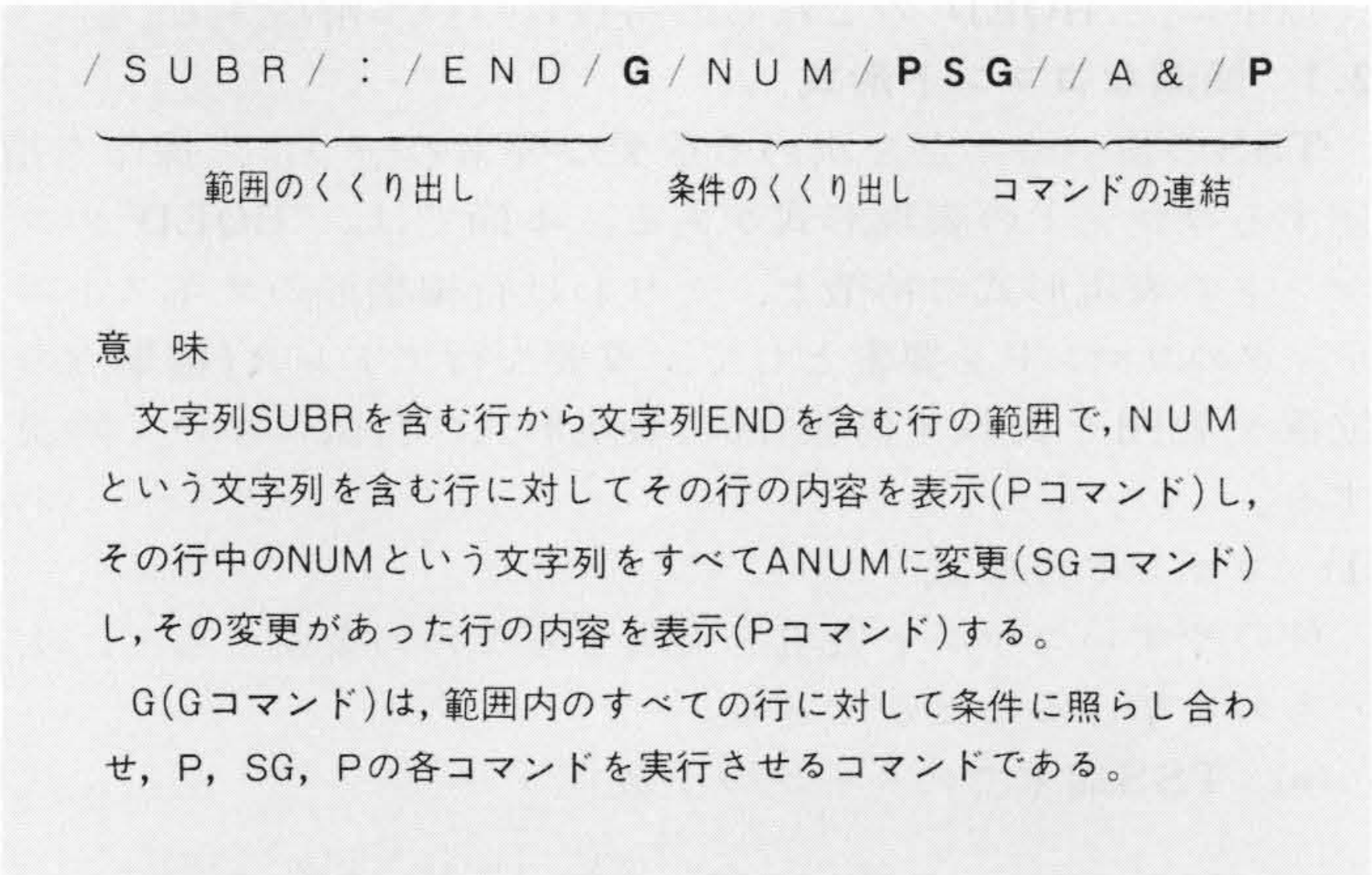


図2 コマンドの共通情報くり出しの例 “HQED”では、編集の範囲と条件をくり出すことができる。

- スだけを指定すれば、表示又は検索表示機能を行なうなど使用頻度の高い機能の操作を簡潔化することができる。
- (viii) 一度に複数の編集操作を行なう場合に、共通の指定情報をくり出すことができる(図2)。

(2) 行アドレスの指定方法

編集点を指定する行アドレスの表現要素として、従来からある行番号のほかに文字列による方法、特殊文字による方法及び相対行数による方法を用意している。なお、これらの要素は互いに組み合わせて使用することもできる。表1に表現要素と代表的な組み合わせを示す。

なお同表中の「正規表現」とは、文字列の表現方法の一つであり、次の2.2節で解説する。

2.2 簡略な文字列の表現

“HQED”では、文字列を表現する方法として正規表現を利用している。“HQED”で言う正規表現とは、有限長の文字列を表わす表現方法であり、例えば、「Aで始まりBで終わる文字列」などの表現ができる。“HQED”のコマンドで正規表現を用いる場合には、その前後に/又は^記号を付けている。また、ある条件を満たす文字列の集合を表わす場合に、特殊文字を用いている。この特殊文字のことを正規表現特殊文字と呼ぶ。表2に代表的な正規表現特殊文字の一覧を示す。

2.3 ユーザーの習熟度に応じた機能分け

“HQED”は使用者の習熟度が増せば増すほど、各機能の組み合わせにより高度な編集操作が行なえるような機能体系になっている。例えば、文字列編集について言えば、正規表現の組み合わせによって極めて高度な文字列検索が可能となる。

しかし、初心者にはこれらの高度な機能は、かえって導入を妨げる要素となる場合がある。また、メッセージの詳しきや出力量に関する要求も、初心者と習熟者とは異なっている。

そこで“HQED”では、次に述べるような機能分けをしている。

(1) 機能レベルに応じた動作モードの設定

初心者用の基本モードと習熟者用の拡張モードとを設ける。

(2) 出力するメッセージの詳しきの調節

簡略形式メッセージと詳細形式メッセージとを設け、あらかじめ選択することを可能とする。また、簡略形式メッセー

表1 行アドレスの表現方法一覧 行アドレスの表現方法として行番号、特殊文字、文字列及び相対行数が利用できる。

項番	行アドレスの表現方法	行アドレス表現の意味
1	行番号 (符号なし10進数)	先頭行を1として、配列順に1ずつ増加した値。 例 10……10番目の行
2	.	カレント行(現在編集行)
3	¥	最終行
4	/正規表現/ —正規表現—	正規表現に適合する文字列を含む行。カレント行から下方向に探して最初に見つかった行を示す。—を付けたときは上方向へ探す。
5	^正規表現^ —^正規表現—	正規表現に適合する単語*を含む行。カレント行から下方向に探して最初に見つかった行を示す。—を付けたときは上方向へ探す。
6	行アドレス+10進数 行アドレス-10進数	上記行アドレスに、符号付き10進数で前後相対値を示す。

注：* 単語とは、空白や(), *, +, —などの記号で区切られた文字列を言う。

表2 正規表現特殊文字一覧 正規表現特殊文字を用いることによって、長い文字列を短く表現することができる。

項番	特殊文字	機能概要
1	"	行の先頭を意味する。 例 "ABC……行の先頭がABCで始まる。
2	¥	行の最後を意味する。 例 XYZ¥……行の最後がXYZで終わる。
3	?	0文字以上の任意の文字を意味する。 例 A?B……Aで始まりBで終わる文字列でAB, AACBなど。
4	.	1文字の任意な文字を示す。 例 A.B……A X B, A+Bなど。
5	*	直前の文字の0文字以上の繰り返しを意味する。 例 AB* C……AC, ABC, ABBCなど。
6	[]	指定された文字群のどれか1文字を示す。 例 [ABC]……A, B, Cの文字を示す。
7	%	正規表現特殊文字を、単なる文字として表現することを意味する。 例 A%.B……A.Bを示す。
8	空正規表現	直前に使用した正規表現を引き継ぐことを意味する。 例 /ABC/S//XYZ/……ABCを含む行を見つけ、XYZに変える。 //はABCを示す。

ジが出力されたときにその意味が分からない場合に、?を入力すると詳細形式メッセージが出力され、更に?を入力するとメッセージの解説情報が表示される。

(3) メッセージ出力量の調節

メッセージの重大度によって出力量を調節するメッセージフィルタを用意している。

これらの選択は端末ユーザーごとに行なえ、“HQED”に記憶させておくことにより、毎回これらの情報を指定する手数を省くことができる。

2.4 高速な編集処理の実現

“HQED”は、高速なテキスト編集処理とともに、高性能な多重処理を実現するための専用の制御機能をもっている。

編集処理では、文字列の検索処理の高速化や作業領域の利用の局所化、モジュール間接続処理の簡略化及び専用制御機能との連携動作による作業用データセットへのアクセスの高速化などの処理方式を採用している。

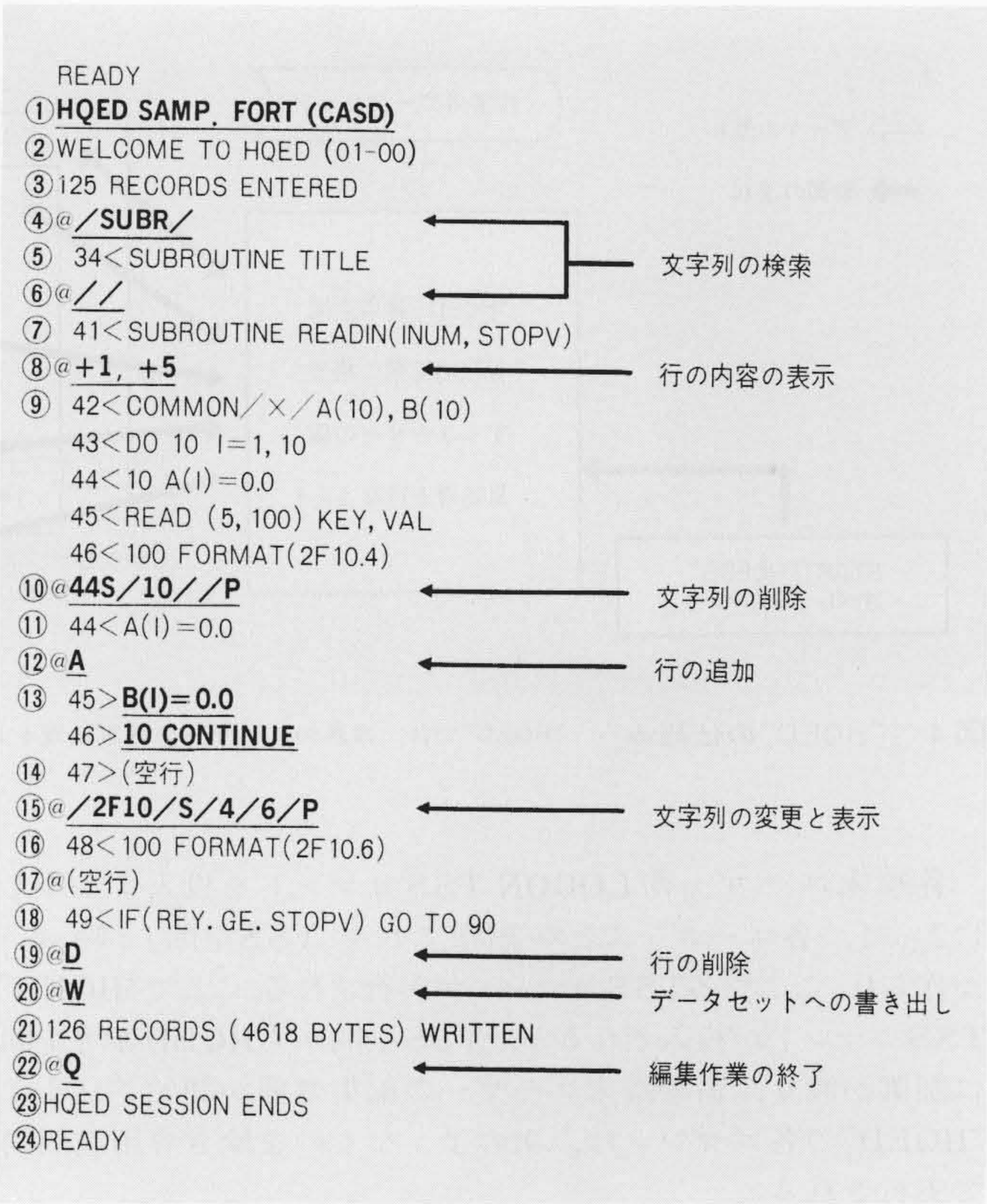
また専用制御機能では、主記憶装置やCPU(Central Processing Unit: 中央処理装置)のスケジューリングの最適化や作業用データセットのための高速アクセス法の提供などの高速化手法を採用している。

3 “HQED”による編集操作

“HQED”のコマンドの主なものには次に述べるようなものがある。

- A: 指定行の直後にテキストを挿入する。
- D: 指定範囲の行を削除する。
- P: 指定範囲の行を表示する。
- S: 文字列の置換又は削除を行なう。
- W: データセットへ出力する。
- Q: “HQED”による編集を終了する。

次に“HQED”を用いてFORTRAN原始プログラムを編集する例を図3に示す。



注: アンダーライン部が入力情報を示す。

図3 “HQED”による編集操作の例 “HQED”により文字列の検索, 行の表示, 文字列の削除, 行の追加, 文字列の変更, 行の削除及びデータセットへの書き出しの操作を行なう例を示す。

以下に各操作について簡単に説明する。

- ①“HQED”コマンドにより“HQED”セッションの開始を指定する。この際、入力データセットを指定する。
- ②“HQED”セッションが開始される。
- ③入力データセットからのデータの読み込みが完了したことを示すメッセージが表示される。
- ④文字列‘SUBR’を含む行を検索し、その文字列を含む行が見つかったら表示することを指定する。
- ⑤行番号34の行が表示される。行番号は先頭の行を1として、各行に1ずつ上昇する番号を付けたものである。
- ⑥目的とする行は、サブプログラム‘READIN’であり表示された行とは異なるため、継続して文字列‘SUBR’を検索することを指定する。//は継続して文字列を検索する指定方法である。
- ⑦該当する行が表示される。
- ⑧サブプログラム‘READIN’の先頭の行を5行表示する。表示機能は行アドレスだけ指定すればよい。+1, +5は行番号41の行(これをカレント行と呼ぶ。) +1の行から+5の行まで表示することを示している。
- ⑨各行の内容が表示される。
- ⑩行番号44の行中の文字列‘10’を削除した後、その行の内容を表示することを指定する。
- ⑪文字列を削除した後の行の内容が表示される。
- ⑫行番号44の行の次にデータを追加することを指定する。
- ⑬追加するデータ行を入力する。
- ⑭データ行の終わりには空行を入力する。
- ⑮文字列‘2F10’を含む行(FORMAT文)中の文字‘4’を‘6’に変更した後、その行を表示することを指定する。
- ⑯変更後の行の内容が表示される。行番号は⑫~⑭の行の追加操作により変更されている。
- ⑰⑱サブコマンドの入力要求に対して空行を入力すると、次の行が表示される。
- ⑲⑱で表示された行を削除することを指定する。
- ⑲編集結果を、入力したデータセット上に書き出すことを指定する。
- ⑲データセットへの書き出し処理が終了したことを示すメッセージが表示される。
- ⑲“HQED”セッションを終了することを指定する。
- ⑲“HQED”セッションの終了を示すメッセージが表示される。
- ⑲TSSのREADYモードとなる。

4 “HQED”の構成

“HQED”では、各端末ユーザーの編集処理をまとめて専用の仮想空間で行なっている。これは、作業用データセットや主記憶装置などの複数のユーザー間に、共通な資源を効率よく使用することにより、特に多端末環境下での高速な多重処理を実現するために採用した方式である。この専用の仮想空間は、“HQED”を起動することによって作成される。

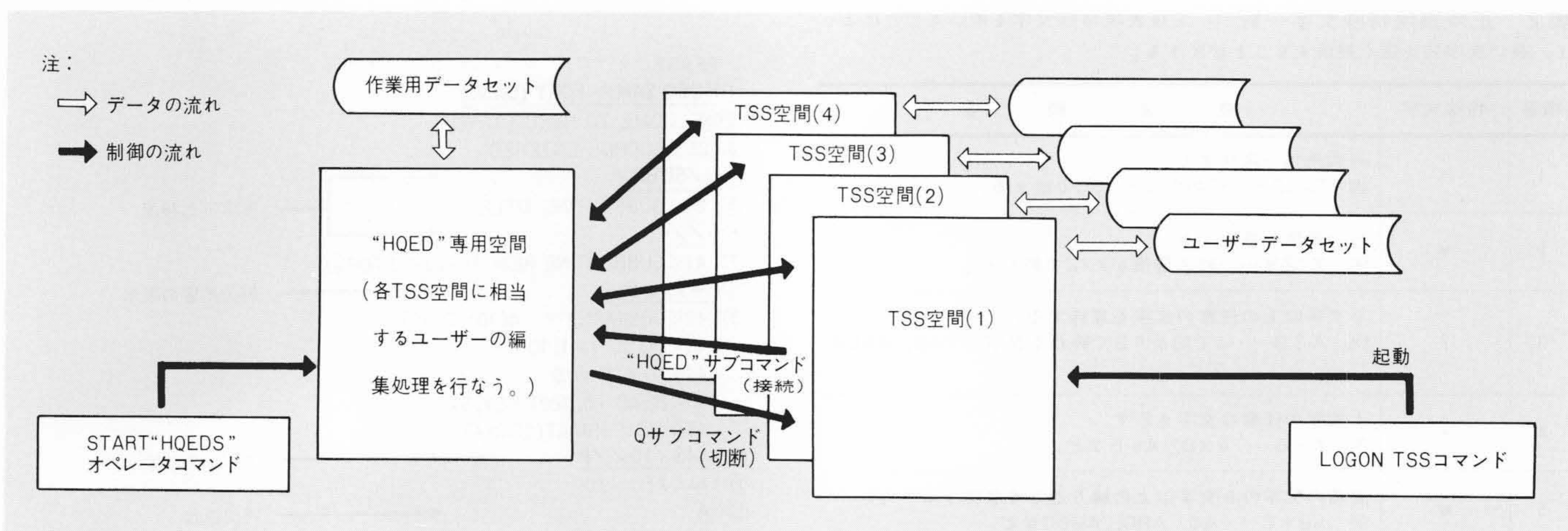


図4 “HQED”の仕組み “HQED”では、複数のユーザーの編集処理をまとめて“HQED”専用空間内で行ない、多重処理の効率化を図っている。

各端末ユーザーがLOGON TSSコマンドを投入することによって、各ユーザーごとの空間(これを「TSS空間」と呼ぶ。)が作られ、ここで各TSSコマンドが実行される。ここで“HQED”TSSコマンドが投入されると、TSS空間からHQED専用空間に制御が渡り、当該端末ユーザーの編集処理が開始される。“HQED”の各コマンドは、次のようなものを除き専用空間内で実行される。

- (a) ユーザーデータセットへの入出力コマンド
- (b) 編集集中の原始プログラムのコンパイルや実行を行なうためのコマンド

これらの関係を図示すると図4のようになる。

5 結 言

本論文では、VOS 3 “HQED”の特長と構造について概説した。“HQED”は、現在大学や研究所を中心に利用されており、端末ユーザーの数も増大してきている。

また同時に、使用するTSS端末数も既に200端末に達しつつあり、多端末環境下での高速な多重処理を実現した“HQED”の利用効果が期待できる。

今後も、更に使いやすいテキストエディタを目指し、操作性向上のための機能追加や性能向上などに努力を重ねていく考えである。

論文抄録

ソフトウェアの信頼性

日立製作所 三森定道

電気学会雑誌 100—697 (昭55-8)

ソフトウェアの高信頼化にも、障害回避と障害耐性の両方策がある。前者は誤りのないソフトウェア設計と製作、すなわちソフトウェア工学の課題である。後者は、ソフトウェア又はハードウェアの障害で生じたエラー状態(記憶装置内の情報の誤り)を、計算機システムの誤動作に結びつけないための方策である。本論文では、障害耐性の方法とソフトウェアの信頼性評価法について述べた。

エラー(状態)の検出法には、(1)多数決、(2)逆演算、(3)符号化、(4)インタフェース規約、(5)診断がある。(2)は計算結果からなんらかのアルゴリズムで入力データを逆算し、実入力データと比較するもの、(4)はモジュール間の交換情報の形式に規約を設け、その規約の合否で判定するものである。

エラー状態からの回復には、バックワードエラー回復とフォワードエラー回復とがある。前者は、正常な以前の状態への復帰であり、後者は、現在のエラー状態を利用して別の正常状態を作り出すことである。

前者の回復を実現するには、ある時点ごとに、状態データ(メモリの内容)を記録する必要がある。この記録量を減らす工夫が種々に開発されている。後者の回復法の一つは例外処理である。

エラー回復操作をプログラマが考えなくてよいソフトウェア機構が開発されてきている。前者の回復法に対するものに回復ブロックがある。プログラマは、幾つかの代替プログラムと、それらの外部変数による一つのテストプログラムだけを、ある形式に従って記述すればよい。回復ブロックを、一つのプログラム内に並置することも、入れ子構造に置くこともできる。

回復ブロックを並行プロセスに適用する場合の問題点を述べる。あるプロセス(プログラム)の実行中にエラーが検出されたとする。最後に状態データを記録した時点に状態復帰するだけでは、エラー回復はできない。他のプロセスとの情報交換により、エラー状態が他のプロセスに伝搬しているかもしれないからである。この問題を体系

的に解決するのが、回復ブロック内に設けられる「会話」という構造である。

本論文の後半では、信頼性評価の問題を述べる。1命令語当たりのバグ数 E や、デバッグ期間1箇月当たりのデバッグ率 ρ は、プログラムの複雑さに依存するが、プログラムの能力に依存する度合は小さい。 $E=6\sim 13(\times 10^{-3})$ 、 $\rho=1\sim 2(\times 10^{-3})$ 程度である。

信頼性評価モデルには、マクロモデルとミクロモデルがある。前者では、障害発生率(ハザード関数)は、残存バグ数に比例するという仮定に基づいている。障害発生間隔の実測データから、残存バグ数を推定する。後者のモデルでは、バグ数はプログラムの複雑さに関係するとする。この複雑さを規定する要素は、プログラム中のオペレータとオペランドの数である。Halsteadはこれらの量から、プログラムの難易度なる量を定義し、これとバグ数との関係式を求め、その妥当性を実証している。