九州大学学術情報リポジトリ
Kyushu University Institutional Repository

画面指向型プログラミング支援環境の構築

宇津宮，孝一
九州大学大学院総合理工学研究科情報システム学専攻

落合，和正
日本アイ・ビー・エム株式会社

畠山，貴次
九州大学工学研究科情報工学専攻

https://doi.org/10.15017/17587

出版情報：九州大学大学院総合理工学報告. 5 (2), pp.197-203, 1983-12-01. 九州大学大学院総合理工学研究科
バージョン：
権利関係：

画面指向型プログラミング支援環境の構築

宇津宮孝一*・落合和正**・畠山貴次***

（昭和58年9月30日　受理）

An Implementation of A Screen-oriented Programming Environment

Kouichi UTSUMIYA, Kazumasa OCHIAI
and Takatugu HATAKEYAMA

Programming environments have a considerable effect on the productivity and quality of programs by computer users. A screen-oriented programming support system, HUMANICS-0, has been developed to study a better human-computer interface and improve such enviroments. This system has been implemented on a typical TSS system with IBM 3270-compatible full-screen display terminals and the function of invoking TSS commands from user programs. Such new facilities as a virtual screen manager for a three-level screen hierarchy, both command-driven and menu-driven user interfaces, and the multi-editing of program texts are shown to be pretty effective in an interactive programming environment. The design and implementation presented here is also adaptable to the development of personal computing environments expected to be widespread in the near future.

1. はじめに

我々は，先に，教育の現場で利用することを目的に，画面指向型のプログラミング実行演示システムを実現した[1]．このシステムは，

(1) 画面が計算機における非常に重要なヒューマン・インターフェースの1つである．

(2) 画面指向のプログラミングを支援するソフトウェアの作成は，使用するディスプレイ装置の多様性のため，意外と困難である．

(3) 計算機との対話を改善するデータの収集や分析が今後重要である．

ということを示唆した．

最近の計算環境は，計算力と記憶量は中央側に，人との対話や手近かな情報処理は端末側へという傾向が顕在化し，個人用計算環境へと急速に移行している．個人用計算環境の構築に当っては，次のように研究すべき課題が多い[2]．

(1) 使い易いヒューマン・インターフェース

(2) 多様な環境に適したソフトウェアの作成法

(3) 計算環境内の構成要素の統合化

(4) 従来の計算環境の評価

本論文では，これらを実際的な立場から研究するために，特に大学の研究者の計算環境を対象にして実現を図った，画面指向型プログラミング支援環境 HUMANICS-0 (HUMAN-oriented Interactive Command processing System-0) について述べる．

2. HUMANICS-0 の設計方針

2.1 設計方針

会話型システムにおいて，人間と計算機の間に作成したソフトウェア・インターフェースにより，使い易さ等利用者の使用環境の向上を図るために考慮した HUMANICS-0 の設計方針は次のとおりである[3]．

(1) 利用者適応性の重視　経験の程度や使用頻度に応じた使用環境の設定，利用情報の検索．

(2) 分り易さの追求　概念モデルの簡単化，作業

* 情報システム学専攻
** 工学研究科情報工学専攻（現在日本アイ・ビー・エム(株)）
*** 工学研究科情報工学専攻修士課程

と画面の対応，コマンドのグループ化．

(3)　作業の円滑化　作業間の移行が容易，一連の作業の記述，端末の操作性の向上．

(4)　合理的な入力　画面上のデータの再利用，検索結果の活用．

(5)　装置の仮想化　画面の装置独立性の強化．

(6)　評価データの収集　利用の客観的評価，改善情報の収集・分析．

2.2　設計と試作

HUMANICS の第 0 版は，種々の利用者階層による使用が期待できることを見込んで，九州大学大型計算機センター FACOM OS IV/F4 上に実現した．前節の方針に基づき，下記の環境と機能を第 0 版の段階では考えた．

(1)　TSS 環境で使用する．

(2)　使用端末は，画面入出力可能な IBM 3270 互換の文字表示端末とする．

(3)　利用者の使用経験に適用できるよう，コマンド入力及びメニュー選択両方式を併用する．

(4)　画面は多重窓を採用し，各窓ごとに独立した作業が行えるようにする．ただし，画面管理の複雑さを避けるため，窓の重ね合せは行わない[4]．

(5)　画面の有効利用を図るため，窓の動的拡大・縮小を許す．

(6)　利用法に関する情報を統一的に管理し，必要に応じて検索し，その結果を次の作業に活用する．

(7)　コマンドは機能別にグループ化し，必要なものが容易に取り出せ，その関係が利用者に理解し易いように階層化する．

(8)　端末の機能キー（PF キー）の活用や画面表示情報の再利用等を通じて操作性を向上させる．

(9)　作業の概念モデルを理解し易くし，モードなど内部処理の都合で導入された考えは極力排除する．

(10)　利用の客観的評価とインターフェース改善のために必要な情報の収集・分析機能を実現する．

(11)　他システムへの移植性を高くする．

3.　HUMANICS-0 の画面管理

3.1　画面の概念

画面には，(1) 表示と (2) 記憶の 2 つの見方がある．

(1) の見方は，対象の可視表現としてであり，人が見たい部分を，見易い形でしかも速く，理想的には常時見える状態で表示することが重要である．一般には，一度に全部を表示するのは困難なので，全体を少しずつ見て行くための窓移動（Windowing）や対象移動（Scrolling）の考えが導入された．

一方，(2) の見方は，対象の記憶表現が主たる問題である．すなわち，対象の論理記憶空間や物理記憶空間の構成法や写像に関する問題である．このとき，表示するということは，記憶内容を表示記憶に移すことに置き換えられる．

従って，記憶の立場から画面というものをとらえれば，従来，仮想計算機の OS で実現されている記憶管理の考え方を，画面管理に持ち込むことができる．その結果，ディスプレイ装置の仮想化と他システムへの移植が比較的容易になる．

3.2　画面の階層構成

画面は，**Fig. 1** に示すように，論理画面，仮想画面及び物理画面の 3 つの画面から階層構成される．

(1)　論理画面　それぞれの処理の対象となる論理的にまとまった情報の集合，例えば，テキスト処理のためのテキスト・ファイルなど．

(2)　仮想画面　それぞれの論理画面を見る窓（論理画面ウインドウ）を一まとめにした画面，装置仮想化のための画面．

(3)　物理画面　実際の表示画面，仮想画面を見る窓でもある．

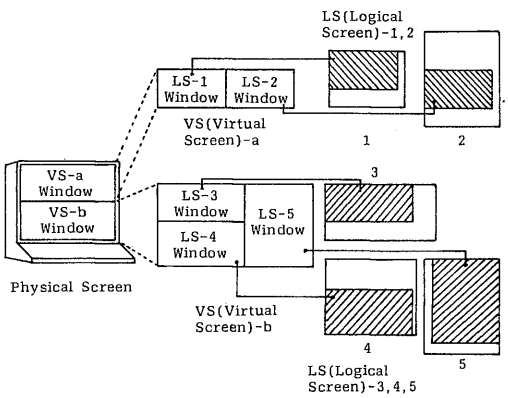

Fig. 1. Hierarchical screen structure.

3.3　画面管理

(1)　物理画面管理　物理画面と仮想画面との写像を管理

(2)　仮想画面管理　仮想画面と論理画面との写像を管理

による管理が必要である．

我々は，(1) に対しては，既存の画面管理用パッケージ IPF (Interactive Programming Facility) を利用し，(2) に対しては，IPF 上に VSM (Virtual Screen Manager) を開発した[5]．HUMANICS-0 は，VSM, IPF を利用した1つの FORTRAN 応用プログラムである．

4. HUMANICS-0 の構成

4.1 構成

HUMANICS-0 全体の構成と環境を **Fig. 2** に示す．HUMANICS-0 は，TSS の端末入力を管理する TMP (Terminal Monitor Program) とコマンドの実行を行うコマンド・プロセッサの間に挿入された1つのプロセスである．

一般的には，端末からの使い勝手は，TMP とコマンド・プロセッサのインターフェースの良し悪しに依存する．これを改善する1つの方法は，上記二者の間に仮想的インターフェースを提供する仲介者を置くことである．この手法を用いれば，基盤となる計算機上に，使い易いヒューマン・インターフェースを有する仮想計算機を実現することが可能になる．

HUMANICS-0 は，後述する機能を実現するため，主として次の4つの部分から構成される．

(1) 画面管理　論理，仮想，物理各画面の管理
(2) コマンド処理系　コマンドの処理
(3) 編集系　テキストの編集処理
(4) データセット管理　テキスト，利用情報，ログ情報の管理

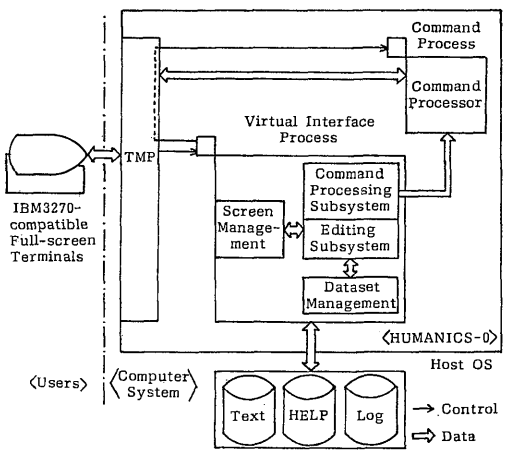

Fig. 2. System organization of HUMANICS-0.

4.2 画面構成

利用者との接点になる HUMANICS-0 の標準画面の構成例を **Fig. 3** に示す．

標準画面は，大別すると下記の5つの論理画面から成り，それぞれ，物理画面上の対応するウインドウにその表示がなされる．

(1) コマンド入力画面
最新の入力コマンドを格納する画面である．
(2) コマンド履歴画面
入力コマンドの履歴を保存する画面である．
(3) テキスト編集画面 (TEXTS)
テキスト編集に関係する画面で，次の4画面の集合
(3-1) TEXTS サブコマンド入力/データセット名画面
(3-2) 窓移動幅/桁位置/メッセージ画面
(3-3) 行番号画面
(3-4) テキスト画面
(4) コマンド・メニュー画面 (MENUS)
コマンドに関する情報を検索するためのメニュー画面で，次の2画面の集まり
(4-1) MENUS サブコマンド入力画面
(4-2) メニュー表示画面
(5) 機能キー (PF キー) 画面 (PFS)
PF キーに関する画面で，次の2画面から成る．
(5-1) PFS サブコマンド入力画面
(5-2) PF キー機能表示画面

5. HUMANICS-0 の機能

プログラミング支援環境の中で特に重要と思われる (1) 利用者との接点である画面の処理，(2) コマンド処理，及び (3) テキスト編集処理に対する HUMANICS-0 の各機能について述べる．

5.1 画面処理

5.1.1 ウインドウの割り付けと論理画面のアクセス

HUMANICS-0 では，操作の煩雑さを避けるため，論理画面の初期ウインドウは，**Fig. 3** のように仮想画面へ静的に割り付けられる．

各論理画面へのアクセスは，対応するウインドウを通してそれぞれ独立に行う．論理画面の未表示部分へアクセスしたい場合には，その部分へウインドウを移動すればよい．ウインドウの移動は，カーソルをその

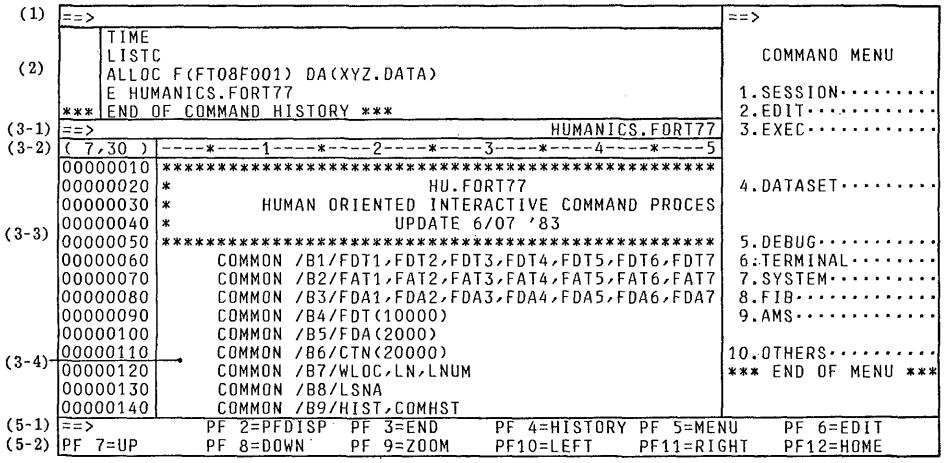

Fig. 3. Standard screen with multiple windows.

ウインドウ内へもって行った後，UP，DOWN，LEFT，RIGHT の各移動用 PF キーを押せば，表示される画面部分は，それぞれ上下左右に移動する．

5.1.2 ウインドウの動的再設定

物理画面の大きさには制限があるため，必要な情報をすべて同時に表示することは不可能である．このため，次の2つの機能を提供した．

(1) ウインドウの拡大・縮小

カーソルをウインドウの任意の位置に移動させ，ZOOM PF キーを押下すれば，そのウインドウは拡大され，再度押すと縮小される (**Fig. 4**).

(2) ウインドウの消滅・再生

PFDISP, MENU, EDIT, HISTORY の各 PF キーをトグルスイッチのように ON, OFF することにより，消滅・再生を容易に行える．

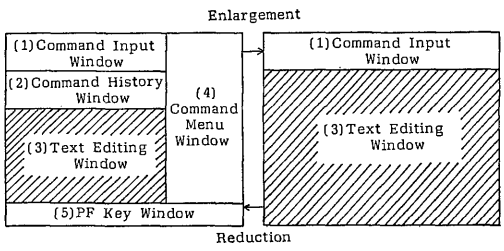

Fig. 4. Enlargement and reduction of windows.

5.2 コマンド処理

5.2.1 コマンドと入力欄

HUMANICS-0 が扱うコマンドは，TSS コマンドと HUMANICS コマンドの2種類である．TSS コマンドは，**Fig. 3** のコマンド入力及びコマンド履歴表示ウインドウから入力する．HUMANICS コマンドの形式は，

〔HU〕 (/TEXTS/ / /MENUS/ / /PFS/) (テキストサブコマンド / メニューサブコマンド / 機能キーサブコマンド)
　　↑
　　コマンド名（論理画面名）

である．これらのコマンドは，各論理画面に付随するサブコマンド入力欄から入力する場合には，サブコマンドを指定するだけでよい．TSS コマンドと同じ方法で入力する場合は，/コマンド名/サブコマンド名の全部を指定しなければならない．

以上の方法により，モードをなくし，コマンドの有効範囲を明確にすることができる．

5.2.2 コマンド履歴の保存と利用

コマンド入力を中心とする利用者の作業の大部分は，幾つかのコマンドの繰り返し使用又はその変形が多い．

これを容易にし，更に自分の作業を客観的に眺められるようにするには，使用コマンドの履歴を保存することは効果が大きい．そこで，次の機能を実現した．

(1) コマンド履歴の収集

コマンド履歴論理画面上で，コマンドに入力時刻及び終了コードを付けて履歴として保存する．

(2) コマンド履歴中のコマンド（群）の再利用

ウインドウを適当に動かしながら，入力したいコマンドを走査する．**Fig. 5** (**a**) に示すように，そのコ

マンド（群）に対して必要があれば変更して 送信 キーを押せば指定したコマンド（群）が実行される．

(3) コマンド履歴中のコマンド群の逐次的実行

一連の作業を繰り返し行う場合，コマンド群を一括して実行する機能あるいは順序を指定して実行する機能が有効である（**Fig. 5 (b)**）．

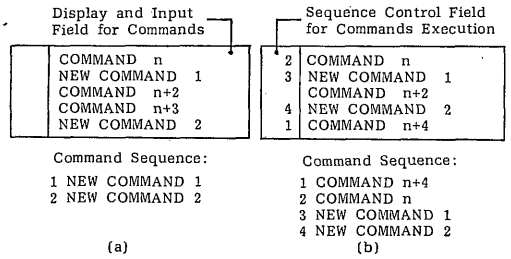

Fig. 5. Commands execution on command history screen.

5.2.3 コマンド検索

一般に TSS 会話システムにおいては，基本コマンドだけでも数十以上ある上に，機能拡張に伴ってコマンドの数は増加の一途をたどる．ある程度そのシステムに慣れ親しんだ利用者でさえ，しばらく使用しないと分らなくなってしまい勝ちである．

従来のコマンド及びメニュー方式に関して下記の問題点がある．コマンドに対しては，

(1) 統一的管理ができる構成ではない．

(2) コマンド名を忘れると，問合せが難しい．

(3) 問合せ結果を，コマンドの実行に直接活用できない．

一方，メニューに対しては，

(4) その性質上，利用者の経験に応じて適用させるということが難しい．

従って，HUMANICS-0 では，(1) コマンドの階層化，(2) メニューの併用によるコマンド検索の多様化，(3) 検索結果の次入力への活用などの機能を追加することにより，プログラミング支援環境の改善を図った．

(1) コマンドの階層構成

例えば，**Fig. 6** に示すように，TSS 基本コマンドを幾つかのグループに分類し，サブシステム—コマンド—サブコマンドという階層的体系としてまとめた[6]．利用者は，この階層に沿ってメニュー画面上で，必要なコマンドに到達することになる．

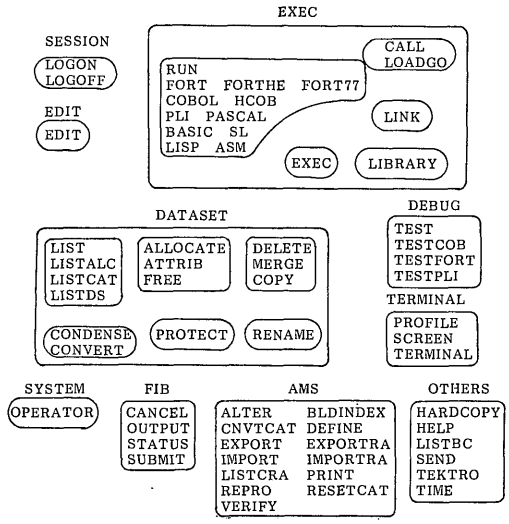

Fig. 6. Grouping of TSS basic commands.

(2) コマンド検索の多様化

Fig. 7 に示すように，利用者が必要とするコマンドの詳細を，(1) サブシステム名，(2) コマンド名，(3) キーワードの3方向から検索することができる．これらの検索は，下記の2種類の方法によって行える．すなわち，

メニュー画面をたどる場合は，

〔/MENUS/〕MENU { n / 文字列 } 〔F〕〔S〕〔E〕

キーワードによる場合は，

〔/MENUS/〕KEY C1 { AND / OR } C2 … { AND / OR } CN

と指定する．ここで，Ci＝キーワード又は NOT キーワードである．

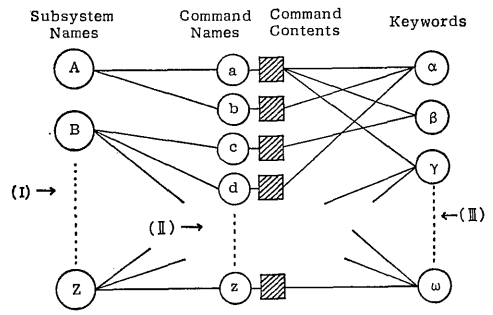

Fig. 7. Command retrieval from three directions.

(3) 検索結果の活用

検索によって得られたコマンドに関する情報は，使用例，機能，構文から成る．この使用例をコマンド入力ウインドウに転送して，適当な加工を施すことによって，次のコマンド入力として活用する．

5.3 スクリーン・エディタ

スクリーン・エディタは，エディタとしての基本機能を備えていることはもちろんであるが，これらの機能が概念的にすっきりと実現されていなければならない[7]．更に，エディタの基本機能と画面，テキスト間，テキストとコマンド間の有機的つながりが重要である．HUMANICS-0 の柔軟な画面管理を利用して，従来統合化されていなかった．(1) 編集画面の動的再編成，(2) テキスト同様，コマンドの編集，(3) 複数テキストの多重編集処理（マルチ・エディティング）の編集支援機能を統合した形で実現した．これらの機能と前節で述べたコマンド処理機能により，プログラミングの重要なプロセスであるテキストの編集とその実行が円滑に遂行できる．

5.3.1 テキスト編集画面

標準画面は，**Fig. 3** に示した．画面を大きく使用したい場合は，**Fig. 4** のように EDIT PF キーを押下して拡大することができる．互いに関係ある 2 つのテキスト間で編集する場合は，**Fig. 8** の画面を用いて行う．

5.3.2 テキストの編集と表示

(1) 文字単位の編集

装置自身の編集機能を利用する．

(2) 行単位の編集

Table 1 に示した編集用行サブコマンドを，対応する編集画面の行番号フィールドに入力し，行単位の挿入，削除，複写，繰り返し，移動等を各画面あるいは複数の画面間で行う．

(3) 編集に関連するその他の作業

文字列の置き換え，行番号の付け替え，データセッ

Table 1. Editing subcommands.

Format	Function
I, In	Insert new lines
D, DD–DD	Delete lines
R, RR–RR	Repeat same lines
C, CC–CC	Copy lines
M, MM–MM	Move lines
A	Mark the line after which the moved or copied text should be inserted
B	Mark the line before which the moved or copied text should be inserted
A′	Mark the line after which the moved or copied text on another editing screen should be inserted
B′	Mark the line before which the moved or copied text on another editing screen should be inserted

```
==>
==>                                          HUMANICS.FORT77
( 7,30 ) ----*----1----*----2----*----3----*----4----*----5----*----6----*----7
00000010 ************************************************************************
00000020 *                          HU.FORT77                                   *
00000030 *        HUMAN ORIENTED INTERACTIVE COMMAND PROCESSING SYSTEM          *
00000040 *                        UPDATE 6/07 '83                               *
00000050 ************************************************************************
00000060       COMMON /B1/FDT1,FDT2,FDT3,FDT4,FDT5,FDT6,FDT7,FDT8
00000070       COMMON /B2/FAT1,FAT2,FAT3,FAT4,FAT5,FAT6,FAT7,FAT8
00000080       COMMON /B3/FDA1,FDA2,FDA3,FDA4,FDA5,FDA6,FDA7,FDA8
00000090       COMMON /B4/FDT(10000)
00000100       COMMON /B5/FDA(2000)
==>                                          VSM.FORT77
( 7,30 ) ----*----1----*----2----*----3----*----4----*----5----*----6----*----7
00000010 ************************************************************************
00000020 *     SUBROUTINE VSMDPN(PIA,VSDA,SKN,RCA)  UPDATE '83 2/17             *
00000030 ************************************************************************
00000040       SUBROUTINE VSMOPN(PIA,VSDA,SKN,RCA)
00000050       COMMON /BLK/ PI,PIB
00000060       COMMON /BLOK/ LSIA
00000070       COMMON /BLOK2/ LSDA
==>       PF 2=PFDISP  PF 3=END     PF 4=HISTORY PF 5=MENU    PF 6=EDIT
PF 7=UP   PF 8=DOWN    PF 9=ZOOM    PF10=LEFT    PF11=RIGHT   PF12=HOME
```

Fig. 8. Multi-editing screen.

トの保存などは，これらを処理するサブコマンドを用いて行う．

(4)　テキスト表示ウインドウの移動

移動用 PF キーによるウインドウの移動は，文字列などを対象とする移動用サブコマンドを用いて行う．

5.4　利用情報の収集

コマンド解析部で収集・分類した種々のコマンドに関する情報，すなわちコマンド名，入力時刻，終了コマンドなどをログ情報データセットに書き出す．書き出された情報は，HUMANICS-0 終了時，指定があれば編集され利用者に提示される．

6. HUMANICS-0 の評価

計算機システムにおけるヒューマン・インターフェースとして，画面が今後とも重要な位置を占め，画面機能の優劣が対話型プログラミング支援環境に重大な影響を及ぼすことは疑う余地がない．狭い画面上に，さまざまな情報を，人間に最適な形で提示することの困難さは古くて新しい問題である．このような観点から，我々が開発した HUMANICS-0 は，下記の点でその有効性が認められた．

(1)　画面の階層化の概念を導入したため，多様なディスプレイ装置を有するシステムにおいて，この種のソフトウェアの作成と移植が比較的容易になった．

(2)　コマンドとメニュー方式の有機的つながりを図ることで，利用者主導，計算機主導双方の対話形式に適用できるので，利用者層が限定されない．

(3)　プログラミングの重要なプロセスであるプログラム・テキストの編集とその実行において，編集画面の柔軟性，多重編集機能は効果的である．

7. おわりに

HUMANICS-0 は，現用の汎用計算機の TSS でも，画面入出力とプログラムからの TSS コマンド呼び出し機能が提供されていれば，この上に仮想的な利用者インターフェースを実現することにより，ある程度満足できるプログラミング支援環境を構築できることを示した．ただし，これによって，集中処理システムとしての TSS の本質的欠点が解決される訳ではない．

しかしながら，今後，普及が予想される個人用計算環境の構築に当っては，HUMANICS-0 の運用実験を通じての客観的評価データが特に有用となる．我々はこの結果を，現在，ローカルエリア・ネットワーク上に構築を進めている個人用計算環境 HUMANICS-U に反映させたいと考えている．

本研究の一部は，昭和57年度文部省科学研究補助金（一般研究(B)代表者　牛島和夫　57460211）による．

参考文献

1) 宇津宮他：TSS 環境におけるプログラム実行演示システムの作成とその検討，情報処理学会論文誌，23，1，pp. 25-34(1982).
2) B. W. Kernighan and J. R. Maskey: The Unix Programming Environment, IEEE Computer, 14, 4. pp. 12-24 (1981).
3) H. Ledgard, A. Singer and J. Whiteside: Directions in Human Factors for Interactive Systems, Springer Verlag (1979).
4) N. Meyrowitz and M. Moser: BRUIN, SIGOPS, 15, 5, pp. 180-189 (1981).
5) FACOM：会話型プログラミング・パッケージ (IPF) 使用手引書，VO2 64SP3520，富士通．
6) I. T. Hardy: The Syntax of Interactive Command Languages, SOFTWARE P & E 12, 1, pp. 67-75 (1982).
7) N. Meyrowitz and A. V. Dam: Interactive Editing Systems, ACM Comput. Surv., 14, 3 (1982).