# 人間の思考に合った計算機環境

馬場　康彦（NTT 電気通信研究所）

## １．　まえがき

一般の人が計算機を利用する場合には、使い易さが重視される．使い易さの要因には、キーボード配列、文字の大きさ、画面の輝度、応答速度、エラー・リカバリ、会話性、ドキュメンテーション等多種多様なものがあるが、一般の人と計算機の専門家との違いが計算機の仕組み、構成、操作法に関する知識の差にあると考えれば、一般の人に使い易いものとは、計算機側の知識から可能な限り独立された人間の思考に合っているものと言うことができる．自然言語による計算機との会話はその典型である．

何が思考に合っているかを知るためには、まず人間の思考過程そのものを解明する必要があり、心理学、脳生理学からの研究[1]やプログラミングにおける思考過程モデルの研究[2]も行われている．本論では、思考および計算機で取り扱う問題として試行錯誤の過程を捉え、さらに思考を構成要素としたマンマシン会話モデルを分析することにより、人間の思考に合ったマンマシンインタフェースとは何かを考察し、事例として言語変換用ワークステーションの機能を紹介する．

## ２．　望まれる機能と特性

マンマシンインタフェースの良否は大きく機能と特性から決まる．望まれる機能、特性は、それぞれ取り扱う問題およびマンマシン会話モデルから導くことができる．ここでは、取り扱う問題として意思決定、プログラム開発等の試行錯誤の過程を考える．それらは、ディスプレイ装置や高機能ワークステーションをインタフェースとして、人間と計算機が会話を行いながら問題解決を図るもので、人間の思考を支援することに重点を置いたシステムである．マンマシン会話モデルは、取り扱う問題によらないもので人間と計算機の対話を一般的に描いたものであり、そこからは一般的に望まれる計算機との会話特性が導かれる．

### ２．１　試行錯誤の特徴と望まれる機能

試行錯誤の例を図1 に示す．意思決定にしろプログラム開発にしても失敗を重ねながら、だんだんと目標に適応させて行くものであり、共通の特徴を持つ．試行錯誤の特徴とそこから導かれる望まれる機能を表1 に示す．示された機能のうち、各種のツール、知識処理等は取り扱う問題により異なるものになるが、他は共通の概念で

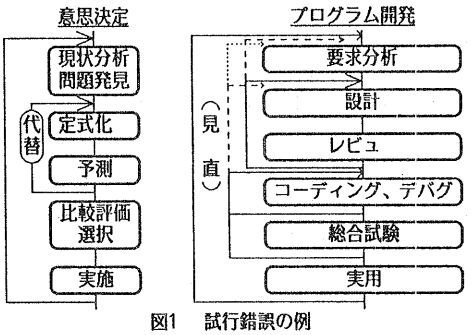


図1　試行錯誤の例

あり操作機能として実現される．

表1　試行錯誤の特徴と望まれる機能

| 特　徴 | 機　能 | |
|---|---|---|
| 多技法 | 各種のツール | |
| 多情報 | 自動参照、関連付け、ディレクトリ、変換 | |
| 多重的、複合的 | メニュ／機能キー、並列処理 | |
| 多更新、一時的 | 編集／加工、清書、メモ | 履歴／変分 |
| 再帰的 | 中断／復帰、後戻り、分岐 | |
| 不完全 | 自動解釈、誤り回復、検証（擬似、実行） | |
| 学習的 | 手引き、案内、促進、例示 | |
| 経験的、発見的 | 知識処理、記憶 | |

## ２．２　マンマシン会話モデルと望まれる特性

マンマシン会話モデルを図2 に描く．一般にマンマシンインタフェースという言葉は、人間と機械の接点（図中の鎖線）を指し、機械側の入出力に対応した操作、認識等の人間活動の表層部分を検討の対象とするが、人間の思考に合ったインタフェースを実現しようとするのであれば、思考、記憶等の深層部分と表層・深層間の仲介を司どる言語活動についても十分、意識する必要がある．これは、利用者インタフェースを設計する際に、プログラムと利用者との関係を互いに呼び合うコルーチンと見做す考え方[3]に一致する．

望まれる特性を導出する評価基準としては、疲れない、いらいらしない等の生理的あるいは心理的なものもあるが、ここではそれらを含めた総合的なものとして、問題解決に要する時間を最小にすることを基準として整理する．時間の短縮は①人間・機械間の会話回数の削減②人間側の表層、言語、深層における各活動（工程）の削除③各工程の高速化により図られる．マンマシン会話モデルにおける人間側の活動を評価項目として整理した望まれる特性を表2 に示す．

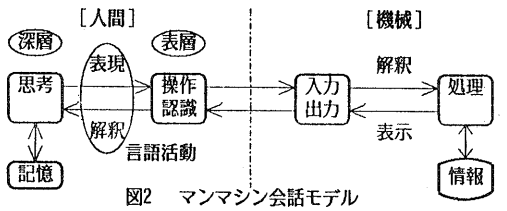

図2　マンマシン会話モデル

表2　評価項目と望まれる特性

| 項目＼効果* | 会話回数削減 | 工程削除 | 工程高速化 |
|---|---|---|---|
| 表層 操作 | 確実 | 感応的 | 直接的、直覚的、少量、連動的、 |
| 表層 認識 | 確認容易 | 感応的 | 誘導的（選択／埋込／例示） |
| 言語 表現 | 完全 | 感応的 | 統一的、形式的、図式／表形式** |
| 言語 解釈 | 無曖昧 | 感応的 | 具体的、対象指向、再利用可能 |
| 深層 学習 | 少概念、一貫 | 自動的 | 体系的 |
| 深層 判断 | | 自動的 | 定量的 |
| 深層 予測 | | 自動的 | 連続的 |
| 深層 創造 | | 自動的 | 再利用／編集可能 |
| 深層 推論 | 無矛盾 | 自動的 | 論理的、抽象的*** |
| 深層 連想 | 連続的 | 自動的 | 網羅的 |
| 深層 分類 | 無重複、直交 | 自動的 | |
| 深層 記憶 | | 自動的 | 意味的、階層的 |

*問題解決に要する時間を最小にする
**入出力相互間
***内部状態／構造の隠蔽、モードレス等

会話回数は、操作誤りや重複の抑制により削減される．工程の削除は、自動化そのものであり、また人間の思考に即反応、実行してくれるような超未来的なものも考えられる．工程の高速化は多様な概念から実現され、統一／連動的なUNIX、連動／選択的なInterlisp-D 、図表／例示を用いたQBE、論理指向のPROLOG、対象指向のSTARやSMALLTALK 等が実例として挙げられる．[4)～9)]

## ２．３　支援環境のイメージ

表1,2 に示された機能、特性は、例えば、試行錯誤に必要な旧来の机、紙／資料、鉛筆に代えて、それぞれディスプレイ画面、多重窓、マウスに対応させると、高機能ワークステーションの設計評価要因となる．試行錯誤を支援する計算機環境のイメージを図3 に描く．節2.2 で人間の深層部分を重視した考え方と同様に、イメージ中に処理系を設けたことにより、各種のツール、情報が連動、一体化していること、またそれを可能とするために情報がツール間に共通な形式であることが処理系に望まれる特性として抽出される．これは、多くの技法と情報を多重、複合的に用いる試行錯誤においては特に重要である．

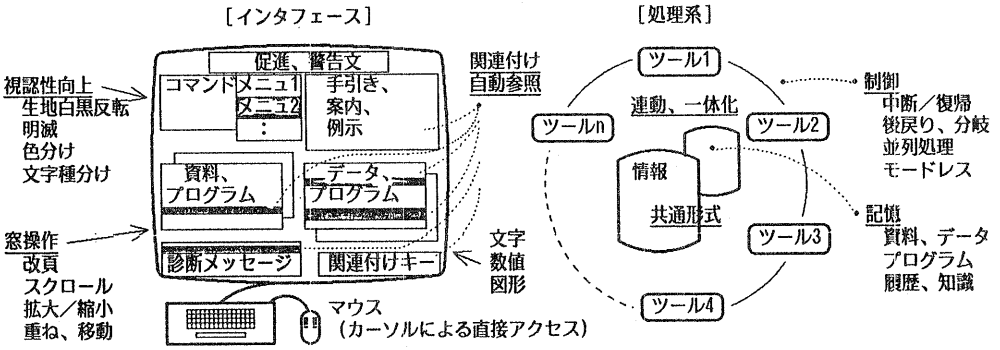


図3　支援環境のイメージ

## ３．　言語変換用ワークステーションの機能と方式

章2 では、人間の思考に合った計算機環境について一般的考察を行い、設計評価要因を抽出した．ここでは、それらの要因を意識しながら作成した言語変換用ワークステーションの機能と方式について述べる．変換は、システム記述用言語SYSL* で書かれたプログラムを機種共通言語Ada 記述に書き直すものである．機械変換における異種言語間の自動変換率は70～90%程度であり、本ワークステーションは残りの手直し作業を支援するものである．

### ３．１　システム構成

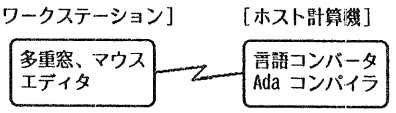


図4　システム構成

システム構成を図4 に示す．ホスト計算機上には、言語コンパータ、Ada コンパイラ、リンケージエディタ等があり、それらはワークステーションから起動される．ワークステーション上には、多重窓とマウスの操作を中心とした利用者インタフェース機能の全てとエディタがある．ワークステーションはXEROX 1100

* NTT 汎用大型計算機DIPSの主要言語であり、PL/Iサブセット仕様に機械語記述機能等を補ったもの．

SIP を用いており、言語変換支援機能のプログラムはInterlisp-D 約3ks で記述されている．ホスト計算機は、NTT DIPSである．

## ３．２　機能

処理の流れとインタフェースを図5 に示す．大きな処理単位は、メニュ操作により選択される．ワークステーションは、機械変換の結果、手直し用のマニュアル等を多重窓に表示する．利用者は、マウスの操作によってそれらの情報を参照しながら手直しの方法を考え、機械変換されたソース・プログラムをエディタを使って修正する．それをホスト計算機に送り、コンパイル、実行を行う．エラーがあれば、再度、手直しを行い、それを繰り返す．

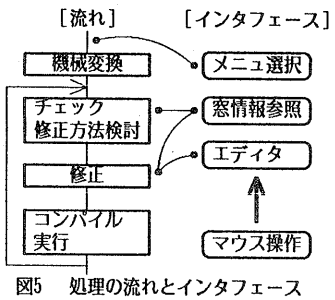

図5　処理の流れとインタフェース

以下、窓の種類と操作、情報の参照方法、エディタの操作について述べる．

### ３．２．１　窓の種類と基本操作

機械変換（言語コンバータ）終了時およびエディタ操作時の窓の種類、使用目的を図6, 表3 に示す．メニュ、コマンド、促進／警告文窓は、本ワークステーション走行中には常駐であり、各処理に共通のものである．SYSL プログラム、Ada プログラム、診断メッセージ、変換メッセージ窓は機械変換終了時に表示され、手直し方法の検討やエディタの操作時に参照される．これらの窓について、図7 に示されたInterlisp-D で提供された削除、移動、拡大／縮小、スクロール等の操作が可能であるが、本ワークステーションではそれ

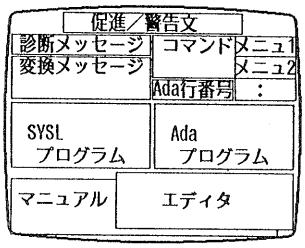

図6　窓の種類

表3　窓の種類と使用目的

| 種　類 | 使用目的 |
|---|---|
| メニュ | 機械変換、エディタ、コンパイル、実行の選択 |
| コマンド | ファイル名等の入力 |
| 促進／警告文 | メニュ選択、コマンド入力の促進、待時間表示 |
| SYSLプログラム | 機械変換前のソース・プログラムを表示 |
| Ada プログラム | 機械変換後のソース・プログラムを表示 |
| 診断メッセージ | SYSL言語仕様違反の表示 |
| 変換メッセージ | 機械変換状況の表示 |
| Ada 行番号 | Ada プログラムの離散* 状況を表示 |
| マニュアル | Ada 言語仕様、例題、手直し方法の表示 |
| エディタ | Ada ソース・プログラムの修正 |

*同一SYSL文がAda プログラム中の複数箇所へ変換出力されたもの

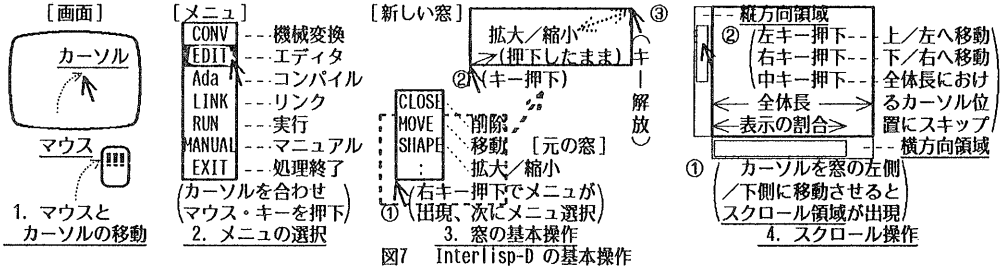

図7　Interlisp-D の基本操作

らの自動化と各種の窓を画面中に適当に収めたことにより、メニュ選択、横方向のスクロール操作を除いてあまり必要としない．

### ３．２．２　情報の参照方法

多重窓に表示された各種の情報を参照する際、窓ごとにスクロール操作を行うような方法では時間もかかり、また情報間の関係を調べるのに労力を要する．本ワークステーションはその自動化の試みに重点を置いたものであり、簡単なマウス操作によって、互いの窓から関連する窓の情報を参照することができる．これにより、利用者は、情報を対応付ける煩わしい作業から解放され、手直し変換方法の検討に集中できる．以下、機械変換の結果とマニュアルの参照方法を説明する．

#### (1) 機械変換結果の参照

機械変換の結果を参照する方法を図8に示す．カーソルを画面上で移動させるだけで、互いに関連する部分の生地が白黒反転され、対応関係を容易に把握できる．その時マウス・キーを押下すれば、関連する部分が窓の先頭位置から表示される．これにより、窓内に表示されていない部分へのスクロールが容易に行われる．また関連する部分が同一窓の複数箇所に分散されている場合には、行番号を選択してマウス・キーを押下すれば、指定の箇所にスクロールされる．

#### (2) マニュアルの参照

マニュアルの参照方法を図9 に示す．メニュ選択と変換メッセージの選択により、変換項目に対応したAda 言語仕様の説明や記述例が得られる．

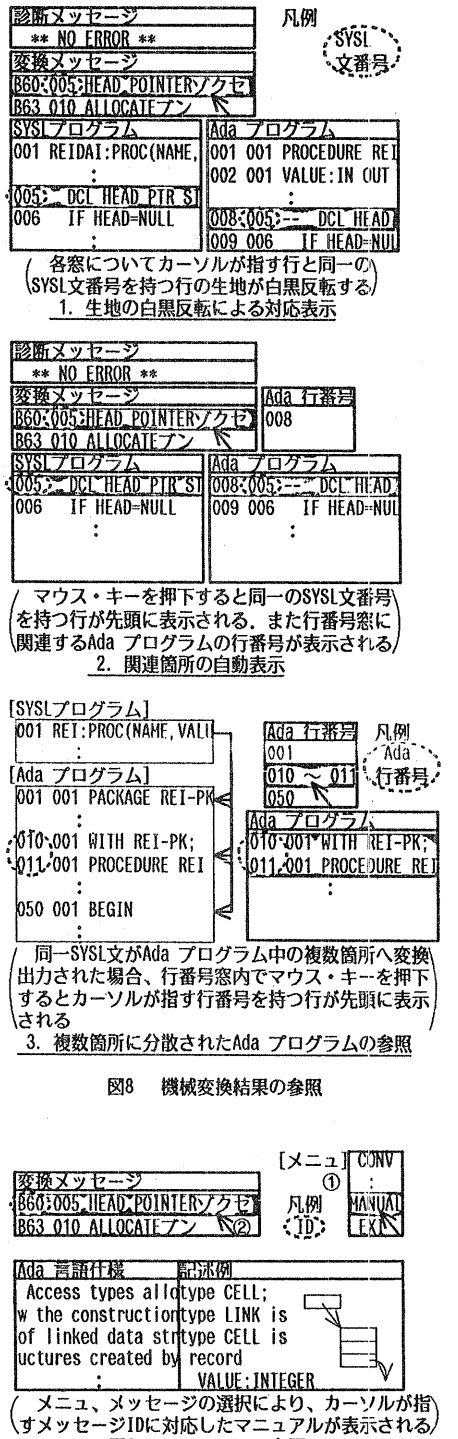


図8　機械変換結果の参照

図9　マニュアルの参照

### ３．２．３　エディタの操作

本ワークステーションのエディタは、Interlisp-D のテキスト・エディタ TEDIT をそのまま組み込み、機能拡張を図ったものである．文字（列）の挿入、削除、コピー、移動等の基本操作は、図10に示すInterlisp-D TEDITの機能をそのまま用いる．拡張した機能は、テンプレートの挿入操作と外部窓からのプログラムのコピー操作である．以下拡張機能を説明する．―

なおエディタ処理終了時には、Ada プログラム窓の内容も更新される．これにより利用者は最新の状態を参照した手直しができる．

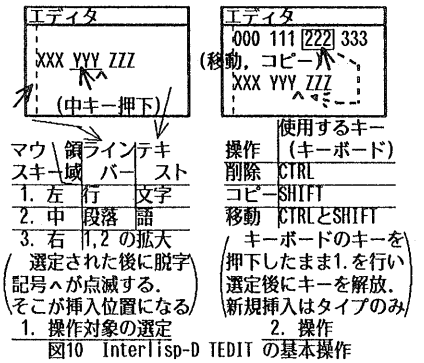

図10　Interlisp-D TEDIT の基本操作

#### (1) テンプレートの挿入

テンプレートの挿入操作を図11に示す．編集は、変換メッセージの選択により抽出されたテンプレートがAda プログラムの指定された箇所に挿入され、その文字列を基本操作によって修正して行くものである．

この方法は、機械変換後の手直し作業の大部分が、変換メッセージごとにパターン化されたものになることに一致させたものであり、利用者は変換方法の詳細な検討に集中できる．

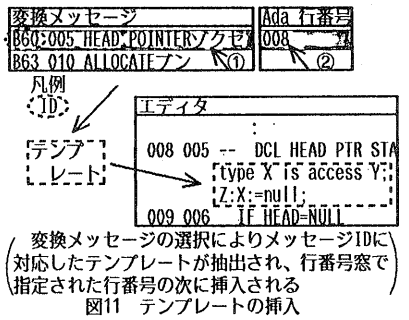

図11　テンプレートの挿入

#### (2) 外部窓からのプログラム・コピー

外部窓からのプログラム・コピー操作を図12に示す．挿入は、Ada プログラム窓の任意の行をエディタ窓のAda プログラムの任意の箇所にコピーするものである．この方法は、プログラム中に同様の変換項目が複数ある場合に、既に変換済みのものをコピーさせるもので、エディタ窓だけでそれを行う場合の頻繁なスクロール操作を避けるものである．

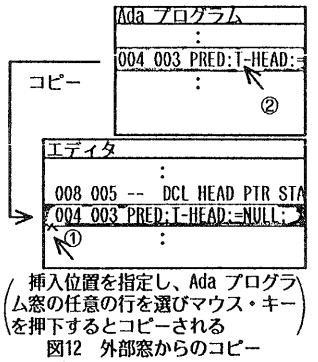

図12　外部窓からのコピー

## ３．３　処理方式

節3.2 では、情報の参照方法を中心に機能を紹介した．ここでは、各種の窓を関連付けた生地の白黒反転や自動スクロールを実現した処理方式を説明する．関連付けの方法を図13に示す．ホスト計算機からの機械変換結果の受信に際し、関連付けのための管理テーブルが作成される．管理テーブルは、SYSL文番号を指標とし、文番号に対応したSYSLプログラム、Ada プログラム、診断メッセージ、変換メッセージの各行番号を値とした配列である．窓内にカーソルが置かれた時、その行の文番号を配列の指標として各窓の行番号を検索し、その部分を白黒反転し、マウス・キーが押下されれば、そこへスクロールさせる．この方法は、各窓に共通な情報としてSYSL文番号を付与したことにより可能となったものであり、節2.3 で処理系に望まれる特性として強調したツール・情報の連動、一体化と情報の共通形式化の一例である．実際、ホスト計算機上の言語コンバータの出力であるAda プログラム・リスト中のSYSL文番号は非必須のものであるが、本ワークステーションの作成を想定して設計時に追加したものである．

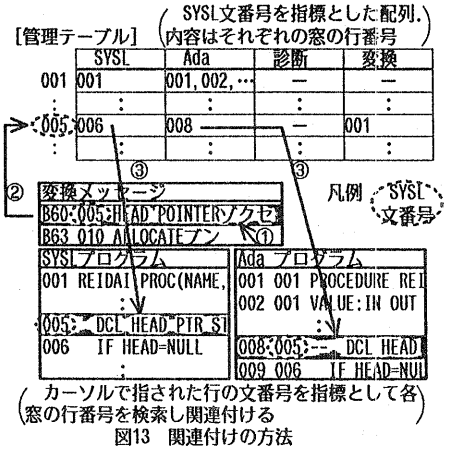

図13　関連付けの方法

## ４．　むすび

人間の思考に合った計算機環境について考察し、事例として言語変換用ワークステーションの機能を紹介した．導出した設計評価要因の内、本ワークステーションで実現した機能は、自動参照、関連付け、メニュ、編集、手引き、促進等である．また特性は、自動的、直接的、直覚的、少量、連続的、統一的、再利用可能等のものになっている．それらは主に、人間の表層と言語活動に対応したものであり、思考そのものを支援すると言うよりむしろ思考に集中させる環境を実現している．

本ワークステーションの機能の内、コンパイラ、リンケージ・エディタの起動とその処理結果の表示は現時点で未サポートであり、マニュアルやエディタ用テンプレートも展示用のものしか用意していない．また、手続き呼出し（CALL文）に使われるパラメータの定義の変換については、プログラム間を対応づけたエディタ機能の実現も望まれる．今後これらの機能を実現し、実際に使い、使ってもらうことによって問題点を抽出し、それをフィードバックして、より使い易いものにする．

謝辞　本論の展開にあたり御指導を戴いた伊吹公夫特別研究室長ならびに本ワークステーション作成の機会を下さった細谷僚一プログラム言語研究室長に感謝する．

## 参考文献


1)堀川勇壮：マン・マシン・インタフェースにおける一考察、情報処理Vol.24, No.6, pp.699-706(1983).

2)Shneiderman, B.et al.:Syntactic/Semantic Interactions in Programmer Behavior:A Model and Experimental Results, International J.Comput.Syst. Sci.Vol.8, No.3(1979).

3)Draper, S.W.and Norman, D.A.:Software Engineering for User Interfaces, IEEE Trans, Software Engng.Vol.SE-11, No.3, pp.252-258(1985).

4)Ritchie, D.M.and Thompson, K.:The UNIX Time-Sharing System, Comm.ACM, Vol. 17, No.7, pp.365-375(1974).

5)Interlisp Reference Manual, Xerox Corp.(1983).

6)Zloof, M.M.:Query-by-Example, AFIPS Proc.National Computer Conf.Vol.44, pp.431-438(1975).

7)Kowalski, R.:Predicate Logic as Programming Language, Information Processing 74, pp.569-574, North-Holland(1974).

8)Smith, D.C.et al.:The Star User Interface:An Overview, AFIPS Proc. National Computer Conf.Vol.51, pp.515-528(1982).

9)Robson, D.:Object-Oriented Software Systems, Byte, August, pp.74-86(1981).