

解 説



ワークステーション ユーザインタフェースの構成技術†

上 林 憲 行††

1. はじめに

従来の端末機器を単に言い換えた意味でワークステーションという言葉が使用され世の中に氾濫しているが、本稿では、実際に使用するユーザの仕事環境、仕事の方法論（スタイル）およびその機能をシミュレーションし、その業務を効果的にサポートする機能を備えた本格的なワークステーションについて論題を絞る。ワークステーションの定義であるが、なににワークステーションといった場合、『その分野のプロフェッショナルが、通常馴れ親しんでいる仕事環境や仕事のスタイルを、そのまま、または、より洗練された形でシミュレーションして、提供する』ものを意味する。つまり、コンピュータの提供するインタフェースを単に提示するターミナルはワークステーションとは言い難く、その業務に適したユーザインタフェースを提供してくれるのがワークステーションの最低の要件だと考えられる。

ここでは、こうした本格的なワークステーションのユーザインタフェースモデルとその実現技術について概観する。

2. ワークステーションユーザインタフェースモデル

最初に、ワークステーションに実現すべきユーザインタフェースモデル (UIM: User Interface Model) の基本的な考え方について言及する。

UIM は従来のヒューマン (マン) マシンインタフェース (HMI: Human Machine Interface) とは本質的に異なる役割を担うものである。HMI は人間とコンピュータとの相互コミュニケーションにおいて人

間の五感に相当する外界からの情報収集または外界への情報伝達の機能・器官の役割をシミュレートするものである。もちろん、HMI を充実して情報収集や情報伝達手段・能力を豊富にすることは、表現能力の豊かなコミュニケーションを実現するために不可欠な要素である。

しかし、ここで重要なことは HMI がバラエティに富んでいても、真のコミュニケーションは成立しないことである。実は人間は大脳中枢によって目 (耳) から入ってきた視覚 (聴覚) 情報を、意味のある情報として認識している。大脳中枢には言語中枢が存在し、そうした情報は言語中枢のモデルの助けによって言葉として解釈され、初めてコミュニケーションが成立するのである。人間とコンピュータとのコミュニケーションでも両者に共通のコミュニケーションモデルを前提にしなければフレンドリなコミュニケーションは行えない。言葉を換えると、情報のセマンティックスを交換するためには、両者を介在する“言語”の存在が必要である。この場合、“言語”は、2.1 節で述べる条件を満足するのであれば自然言語であっても人工言語であっても良い。

UIM をさらに解析するとコミュニケーションモデル (COM: Communication Model) とユーザ環境モデル (UEM: User Environment Model) の 2 つのモデルから構成されると考えると理解しやすい。

COM はコミュニケーション (意志の伝達) におけるセマンティックスを解釈する中枢に相当し、意志の伝達に責任を負う。

UEM はユーザの作業環境・空間を電子的にかつ効果的にシミュレートする役割を果たし、コンピュータの機能インタフェースとユーザ作業空間とのギャップを最小化する責任を負う。また COM における意志の伝達・理解をスムーズに行うアシストや方法論を提供するものである。

† Key Technology of Designing Workstation User Interface by Noriyuki KAMIBAYASHI (Software Technology Center, Fuji Xerox Co., Ltd.).

†† 富士ゼロックス(株)

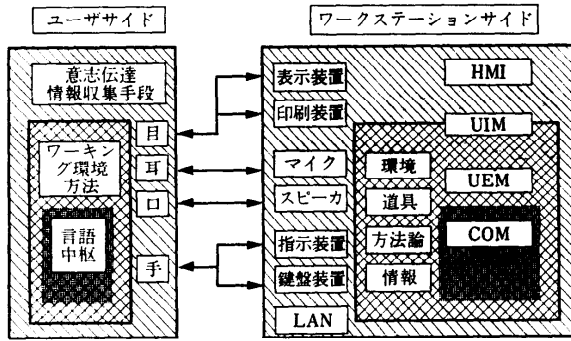


図-1 ワークステーションにおけるユーザインタフェースモデルの位置付け

COM: Communication Model
 UEM: User Environment Model
 UIM: User Interface Model
 HMI: Human-Machine Interface Model

表-1 ユーザインタフェースモデルのアプローチ
 - (J)Star の場合 -

簡単	VS.	困難
具体的表現		抽象的表現
視覚化情報		文字列情報
複製する		創造する
選択する		言葉をうめる
識別する		想起する
編集する		プログラムする
対話の処理		一括処理

そして HMI はコミュニケーションにおける意志伝達の効率化とその手段を多様化し意志疎通の円滑化を図る責任を負うものとする。図-1 に HMI, UIM, UEM そして COM の役割の関係を示す。

このような見方で、PWB (Programmer's Work Bench) として評判の高い UNIX のユーザ環境を考えるとコマンドインタフェースのパイプ等の機能は COM に相当すると考えることも可能である。またユーザ環境への適応機能としてはシェルが対応すると考えられる。また、HMI は CRT ディスプレイキーボードということになる。

ワークステーションユーザインタフェースの設計アプローチの一つの具体例を表-1 に示す。

2.1 コミュニケーション (インタラクション) モデル

ここでは、コミュニケーションモデルについて議論する。

現在の大部分のシステムではコミュニケーションモデルや認識モデルが欠落したまま、個々の流儀で機能

本位にしかも安易にアクションコマンドが追加されているのが現状である。

現実に自然言語でコミュニケーションすることは、自然言語に内在されている膨大な“あいまい性”の障壁 (論理的完結性の不備) がある。また、コミュニケーションを成立させるために不可欠な両者に共通で暗黙的な知識の土台の欠落がある。そのため、自然言語でコミュニケーションすることは現在の技術レベルでは困難であるとされている。そのため、人工言語によるコミュニケーションモデルを確立することが、現実のアプローチとして選択される。

こうしたインタラクション言語が備えるべき要件として考えられるべきものとしては次のようなものがある。

- 1) 簡潔性, 2) 一貫性・統一性, 3) 表現・操作能力,
- 4) 認識・理解能力, 5) 論理性

例えば、IBM 社の OA 戦略の先兵となっている QBE (Query By Example)⁴⁾ およびその発展形である OBE (Office Business Procedure By Example) もユーザとコンピュータ・ソフトウェアのコミュニケーションギャップを埋める目的で開発されたものである。QBE は関係データベースの理論的基盤に立脚しており、その基礎概念は平易であるにも拘わらず論理深度の深い表現が可能である点などで評価されている。

さらに、Xerox PARC でこの十年間地道な研究が続けられてきた Smalltalk³⁾ も以上述べたコミュニケーション言語の要請を満たすものである。Smalltalk が前提とするコミュニケーション (コンピュータシオン) モデルは Communicating Objects モデルとして知られているものである。これはわれわれの環境、知識、意志をモデル化し表現するのに、まず個々のリソースを object (対象) としてとらえコミュニケーション (意志伝達) は objects 間の message として定義される。この Smalltalk-80 の世界では、すべてのリソース (ユーザ、ソフトウェア) 間の相互の意志伝達 (インタラクション) は、あるリソース (object) が message を別のリソース (object) に送付するという方法で統一的に行うことができる。このような UIM (COM) はオブジェクト指向 UIM⁷⁾ と呼ばれる。これは従来のアクション指向のコマンドインタフェースとは異なり、次のような優れた点があるとされている。

- ユーザのオペレーションの習得性

- ユーザの日常の事象の認識方法との類似性

- 新たなシーンにおける操作 (対話) の類推・応用能力

- 表現方法の一貫性
- ドメインの設定能力
- ソフトウェア設計の容易性

2.2 ユーザ環境モデル (UEM)

以下, (J)Star^{11), 21), 22)} を題材に取り, OAワークステーションにおけるユーザ環境モデルの基本的な考え方とアプローチについて述べる.

1) 電子ディスクトップコンセプト

電子ディスクトップの考え方はオフィスにおける第一の仕事環境ファシリティである『机』をそっくりそのまま電子的にシミュレーションするというものである. つまり『机』の周辺にある情報整理用ツール, 各種書類, テキストや図表を書くためのエンジニアリングツール, 電子メール等の基本的なファシリティを高解像度ビットマップディスプレイ上にユーザに親しみのある具体的なイメージ (視覚 ICON: 3.1 節で後述) として実現したものと考えられる. ユーザは『机上』の書類やツールの配置まで自分の仕事のスタイルにあわせることが可能となり, なんの違和感もなく従来の仕事環境がスムーズに『ワークステーション』に移行できる.

Apple 社の Lisa¹⁰⁾ では, オフィスファシリティとして, 6つのツール (Lisa Calc, Lisa Graph, Lisa Draw, Lisa Project, Lisa List, Lisa Write) を標準的に用意している.

Star のユーザ環境モデルを構成するファシリティを図-2 に示す (WYSIWYG については後述).

また, こうしたモデルを基礎に, ユーザ環境の設定能力の自在性 (User Tailorability: ユーザ環境の仕立て上げ能力) もユーザ環境モデルの重要な要素である. Lisa, Star, Smalltalk-80 などでは, こうした機能は標準的に用意されている.

3. ユーザインタフェースの構成要素技術

ここでは, ワークステーションにおけるユーザインタフェースの構成技術について言及する.

3.1 視覚コミュニケーション技術

従来のシステムでは, マシンとの対話においては, 基本的には, 同時に一つのインタラクションストリー

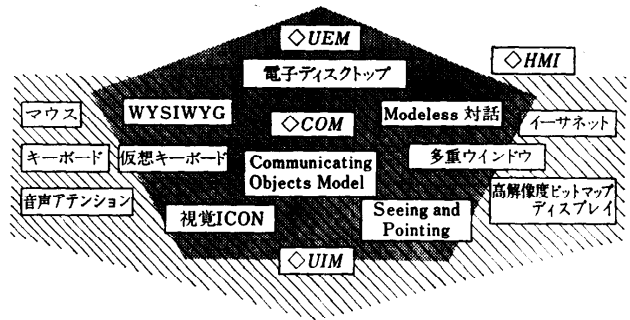


図-2 ユーザインタフェースモデルの具体例—(J)Star—

ムしか設定出来ないのが普通であった. ワークステーションを本当の意味での『仕事机』として機能させるためには, マシンとの間に, 同時に複数のインタラクションストリームが設定出来なくてはならないし, またより視覚的なインタラクション機能の充実を図り, スムーズな情報流通を促進させることが, ますます必要になってくる. こうした視覚コミュニケーション技術を実現する手段としては, 以下の技術に着目したい.

1) ビットマップディスプレイ技術

このビットマップディスプレイ技術は, マルチウィンドウ, 字体のヴァリエーション, グラフィックス, イメージなどの多彩な表現を可能にする要素技術である. このビットマップを効率的にサポートするには BITBILT 機能が重要であると認識されている. この BITBILT 関数はスクリーン上のあるポジションのパターンを, ある指定されたロケーションにコピーする機能を基本とする. ただし, この間, さまざまなコンビネーションルールが作用する. ビットマップディスプレイをサポートするコストは多大であるが, 視覚的なユーザインタフェース (例えば ICON 等) を提供することによる恩恵は, それを上回るものがある.

2) マルチウィンドウ技術

通常われわれは『ワークステーション』にいくつかの書類 (レポート用紙) や道具を置き, それらを交互に参照しながら仕事をして行く. これと同じ状況を実現するのがこのマルチウィンドウの概念である. この意味で, マルチウィンドウは, ユーザとマシンの間に複数の情報チャネルを設定でき, インタラクションの効率を飛躍的に改善する.

一口にマルチウィンドウといっても, 表層的には, スクリーン上に, 単に, 複数のウィンドウを表現した

表-2 マルチウィンドウの機能と能力の比較

システム/比較項目	Lisa	JStar	Interlisp-D	Smalltalk-80
マルチウィンドウ機能	○	○	○	○
オーバーラップ機能	○	X	○	○
ウィンドウの数	リソースの許す限り	最大6個	リソースの許す限り	リソースの許す限り
ウィンドウのサイズ	可変	固定(12種類), 可変も可	任意	任意
補助ウィンドウ機能	○ (ポップアップ & スプリッド window)	○ (プロパティ window)	○ (ポップアップ window)	○ (ポップアップ window)
基本ウィンドウの種類	6種類 (Lisa Calc, Lisa Graph, Lisa Draw, Lisa Project, Lisa List, Lisa Write)	7種類(ドキュメント, フォルダ, ドロウ, メールシート, レコードファイル, 3270ウィンドウ, TTY ウィンドウ)	3種類 (top type-script window, prompt window, general window)	5種類 (system transcript, workspace, project, system-browser, system-workspace)
基本ウィンドウスクロール機能	上下-左右スクロール (固定ピッチ, ページめくり, 任意ページめくり)	上下-左右スクロール (固定ピッチ, ページめくり, 任意ページめくり)	上下スクロール (任意ピッチ, 任意ページめくり)	上下スクロール (任意ピッチ, 任意ページめくり)
ウィンドウの activity	activated window	nothing	window with process activity	activated window
基本ウィンドウオペレーション	?	表示オプション (3種類) 表示位置	clear close bury move shape repaint paint snap shrink	under move frame collapse close
スクリーンリフレッシュング	?	自動リフレッシュング	自動リフレッシュング	activated リフレッシュング

ものを含めて議論されることが多いが、マルチウィンドウ技術の焦点はそのウィンドウの持つ属性およびアクティビティによって評価すべきものである。表-2は代表的なワークステーションにおけるマルチウィンドウ技術についてまとめたものである^{6),9),10)}。

3) 視覚 ICON

視覚 ICON は、従来の文字列によるコンピュータとの対話ではなく、ユーザが利用できるリソース (オブジェクト) を絵解きとして表示するものである。『イメージの提供する情報量は数千の文字に相当する』と一般に評価されている。また、こうした視覚 ICON によるインタラクションは、ユーザに親近感を与え、張り詰めた緊張感を癒すのに効果がある。

3.2 選択, 指示コミュニケーション技術

スムーズなインタラクションを実現するには、スクリーン上に表現された任意の情報を選択しシステムに

的確な指示を行う技術が必要となる。選択, 指示コミュニケーション技術は、スクリーン上の任意のロケーションにいかにかスムーズにカーソルを移動させることが可能であるかがその評価基準である。こうした技術としては、ライトペン, トラックボール, マウス (機械式, 光学式), タッチパネル, カーソル移動キー等がある。選択, 指示コミュニケーション技術の評価基準としては、精度, スムーズ性, 任意性, 疲労感, 信頼性, 付加オペレーションとの関連等が挙げられる。こうしたポインティングデバイスの特性の簡単な比較を表-3に示す。

またこうした技術の評価する法則としては、Fitt's law が有名である。Fitt's law は以下の式で評価される。 T は、移動スムーズ性を、 S はポイントする対象の面積を、 D は現在のロケーションからポイントする対象のロケーションまでの距離を、 k は常数である。

表-3 ポインティングデバイスの機能と能力の比較

システム/比較項目	カーソルキー	ライトペン	タッチパネル	トラックボール	マウス
ポインティングの精度	X	○ (精度を合わせるのがたいへん)	X	◎	◎
移動のスムーズ性	X (デジタル感覚)	△	△	○ (アナログ感覚)	◎ (アナログ感覚)
疲労感	△ (時間がかかる)	X (スクリーンまで手を伸ばす事が必要)	△ (スクリーンまで手を伸ばす事が必要)	△ (肘を挙げて操作することが必要)	◎ (肘, または手を机に置いて操作が必要)
信頼性	◎	△	△	△	△
付加オペレーションとの関連	なし	なし	なし	なし	重要

表-4 ポインティングデバイスのボタン機能の比較

システム/比較項目	Lisa	JStar	Interlisp-D	Smalltalk-80
ボタンの数	1	2	3	3
ボタンの機能の割り当て	① オブジェクトセレクション (ポップアップメニューオープン) ボタン ボタンプッシュキーピング ロール	① オブジェクトセレクションボタン ② セレクトッドオブジェクトアジャストボタン	① セレクションボタン ② プロパティポップアップメニューオープン & 選択ボタン ③ ウィンドウメニューオープン & 選択ボタン	① セレクションボタン ② プロパティポップアップメニューオープン & 選択ボタン ③ ウィンドウメニューオープン & 選択ボタン
クリック (テキスト, グラフィックス)	ウィンドウのオープン/クローズ (2回クリック)	セマンティックテキストクリック (4段階)	セマンティッククリック機能なし (ただし, Dedit, Tedit ではあり)	セマンティックテキストクリック (2段階)

マウスはこの Fitt's law の特性に優れていると評価されている。

$$T = k \frac{\log D}{S}$$

マウスの設計のトレードオフについて、若干議論する。ポインティングデバイスとして、マウスが評価されるのは、肘をついての操作が可能のために、疲労感がなく、頭で意図した操作が自然に行えること、更に重要なことは、マウスに付加されている指先でコントロール可能なボタンとのコンビネーションで軽快な操作 (タイプレス) が行えることが重要である。こうした点が、従来のポインティングデバイスと一線をなす特徴である。

その意味で、マウスの設計のキーは、マウスのボタンの数とそのボタンに対するセマンティックの割り当てである。Lisa, JStar, Smalltalk-80, Interlisp-D におけるマウスボタンのセマンティックの割り当てについての簡単な比較を表-4 に示す。

3.3 ユーザ/システムインタラクション技術

ここでは、ユーザとシステムのインタラクション技

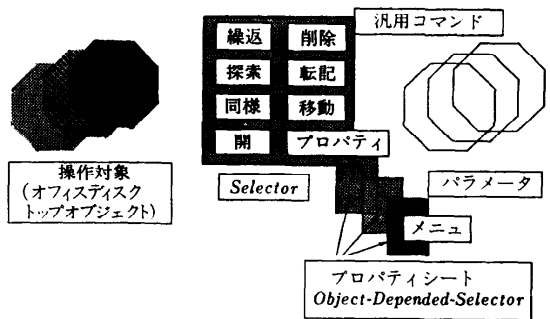


図-3 ユーザ/マシンインタラクションの基本原理の一例-(JStar)-

術について、特に Lisa や Star で共通に採用されている基本技術について述べる。

1) コミュニケーションモデル

(JStar) を例に取り、コマンドインタラクション技術つまりコミュニケーションモデル (COM) の具体的なアプローチを説明する。(JStar) コミュニケーションモデルは、前節で若干触れた Smalltalk の提示す

る Communicating Objects モデルを OA 環境に適用して具体化したものと考えられる。コマンドオペレーションの指示の手順(コミュニケーションモデル)は次の通りである(図-3参照)。

① ユーザは電子デスクトップに表示されたオブジェクト、例えばファイル、テキスト、文、図表の要素などをまずマウスで選択する。Smalltalk 風にはレシーバの指定である。

② 次いでコマンド(Smalltalk 風にはセレクタ)をキーボード上の汎用コマンドキー、または generic コマンドキーの一つであるプロパティキーによって示される、選択したオブジェクトに固有のコマンド(サブコマンド)メニューをマウスで選択する。

③ そして必要であれば、マウスでアークギュメントを指定する。

この考え方は、従来の『動詞/名詞パターン』のアクション指向の、問いかけとは異なり、『名詞/動詞パターン』を基本としたものである。

この方式が、優れているのは、強力な対話のモデルが存在するために、最初に対象を選び、次に操作の内容を与えるという原則を覚えるだけで、システムのすべての機能を、類推しながら、試行錯誤的に、習得することができることにある。

2) WYSIWYG インタラクショナル技術

WYSIWYG(“What You See Is What You Get”)の考え方は、つまり作成・編集するテキスト(字体、フォント、レイアウト)や図表の出来具合を絶えず視認して作業を進めることができ、最終的な仕上がりをディスプレイ上で認識して、絶えず仕上がり具合を目で確かめながら作業を進められる環境である。こうした作業ユーティリティを用意することにより、ユーザは無駄な作業工程を繰り返すことなく効率的に作業が進められる。

3) Seeing and Pointing インタラクショナル技術

Seeing and Pointing の概念はマウスと呼ばれるポインティング・デバイスの技術的なバックアップによって成立したものである。しかしその思想はコミュニケーションに対する深い洞察に基づいたものである。人間の思考過程において重要な役割りを果たしているのが短期記憶(Short-Term-Memory)であるとされている。スムーズなコミュニケーションを成立させるためにはこの短期記憶の役割りをコンピュータ側に代行させかつ必要最小限のタイピング(レスタイピング)

でインタラクショナルを遂行できるよう考慮したのが、『Seeing and Pointing rather than remembering and filling』の考え方である。

4) Modeless インタラクショナル技術

Modeless 対話(Interaction)は従来コンピュータ側の都合で設定されていたモード(Mode)の概念を取り払ったものである。モードとはキーボードからの入力コード列の意味をユニークに解釈するために必要になったもので、本来のコミュニケーションとは無関係のコンピュータ側の論理である。一つは、キーボードストリームを介さないで、例えば、マウス等によって、ポップアップメニューでコマンドを選択するか、専用のファンクションキーを用意することなどは Modeless インタラクショナル技術の基本である。こうした、Modeless インタラクショナルの考え方は、マルチウィンドウシステムでは基本的には採用されている。

3.4 日本語ユーザインタフェース技術

ワークステーションユーザインタフェースを構築する上での日本語情報処理に特有の問題について考察する。日本語情報処理を本格的にサポートするとなると巨大なデータベース(フォントと辞書)が本質的に必要となる。

1) 日本語タイピングシステム

日本で使用できるワークステーションを構築する技術は、現時点では、日本語タイピングシステムおよび日本語ドキュメントエディタに集約されると考えられる。特に、ユーザインタラクショナルの点を考慮すれば、仮名漢字変換方式のユーザインタフェースについて、本質的な見直しが必要であると考えられる。特に、本来、ワークステーションのユーザは、タイピング(コンピュータ)等に関しては素人であるために、日本語タイピングシステムに適合したユーザインタフェースの開発が望まれる。現実には、仮名漢字変換方式の効率を上げるためには、辞書の整備(登録語数とコンパクト化のトレードオフ)、簡単な学習機能(候補語の優先度の充実等の動的適応)の付加、ユーザ辞書登録機能の充実等が主たる課題である。しかし、今後の課題としては、やはり、候補キーのヒット率を上げ、単なる機械的なタイピングではないスマート(文節、複合語、文章単位の変換)な振る舞いを行うユーザインタフェースの開発が望まれる。そのためには、単に一般的な文法則だけでなく、その入力ストリームに関する背景知識およびその状況的な知識を獲得しなければ、意味的に正しい(そのユーザが所望す

る) 仮名漢字変換が行われない。単に、辞書引きを自動化したタイピングシステムから本当の意味でのスマートタイピングを目指すべきである。最近の機械翻訳システムやエキスパートシステムにおける自然言語インタフェースの成果を積極的に取り入れるべきである。

2) 日本語エディタ

コンピュータとのインタラクションにおいてエディタは文字列(テキスト, ドキュメント)を編集レイアウトするツールとして, 最も重要でかつ基本的なツールである。その意味で, 欧米では, エディタ・フォーマッタの研究が精力的に行われてきており, いくつかの優れたエディタが開発されている。また, ドキュメント編集専用のエディタも WP の爆発的な普及によって優れたものがある。WordStar, CPT8000, NBI, System 3000 等の WP 用エディタ, Bravo, Star 等のエディタ/フォーマッタ, ストリームエディタの代表例として TECO などが上げられる。また, ディスプレイベースエディタとしては, EMACS, IBM 社の XEdit 等がその代表的なものである。

しかし, 本格的に日本語でインタラクションを行う場合には(ワークステーションの使用形態からみてそのほうが自然である), そのインタラクションの基本となる日本語エディタの優劣が問われることとなる。この意味で, 欧米で開発されたエディタシステムに関してその取り扱える文字セットを拡張しただけのシステムではない, 日本語特有の言語フレームを基調としたエディタの開発が望まれる。例えば, Smalltalk-80 は, 強力なモードレスエディタが用意されているが, そのなかで, 英文のフレーズのレベルに合致した, セマンティッククリッキング(キャラクタ, ワード, センテンス, ドキュメントの単位)の機能が組み込まれている。

つまり, より良いスマートな日本語エディタの設計に関しては, 究極的には, 日本語の言語現象及びその秩序を基調としたインテリジェントなアシスト機能を組み込む必要がある。

4. おわりに

本稿では, 初めに特に OA 環境におけるワークステーションユーザインタフェースモデルの設計について, ヒューマン・マシンインタフェースモデル, ユーザ環境モデル, コミュニケーションモデルの3つの概念によって議論した。さらに, そのユーザインタフ

ェースを構成する基本技術について述べた。

現在では, ユーザインタフェースモデルの設計は, 基本的にはアートワークの領域であり, サイエンスの領域に達していない。ソフトウェアもそうであるが, 最初はすべてアートから始まりその中から昇華したエッセンスだけがサイエンスの領域に迎えられるのである。

今後の課題としては, ユーザインタフェースの設計は, 単にコンピュータサイエンスの分野だけでは, 解決できない問題が多いために, 学際的なチームを編成してことに当たる必要が, ますます肝要となる。例えば, 労働衛生学, 人間工学, 心理学, 認知心理学, 言語学などの成果と知識を取り入れるべきである。

最後に, ワークステーションユーザインタフェースのアイデアの源泉は, その『仕事机』の機能と使われ方を, 注意深く考察し, 現実の許容範囲でそのモデルをいかに実現するかということにつける。

ここでは, 『机』の機能と人間との作業の連係について考察し, それからコンピュータでアシストすべき機能を示唆している文献⁵⁾のさわりを紹介し, 本稿を閉じる。

『机』上整理の主要な機能は, あるものを見付け出す(Finding)と想起(Reminding)である。

『机』の重要な機能は, 単に必要となる情報を見つけたのを手助けするだけでなく, 自分が何をこれからしなくてはならないかを想起させてくれることである。

また人々が自分の『机』を活用できるかどうかは, インフォメーションをいかにしてカテゴリ分けするかということ, たいへん関連が深くなる。つまり, 流入してくるインフォメーションをいかにカテゴリ化するかといった点である。機械的な作業の煩雑さ, 例えば, ファイル, フォルダ, バイнда等のラベルを作成する面倒くささ, 分類する場合の煩雑さ, 例えば, どのようなカテゴリ分けが適切であるかを判断する。

この状況の解決策としては,

- ① 据え置き型(Defer Classification): インフォメーションを物理的な位置によって分類する手段を提供すること(例えば, バイール)
- ② 自動分類(Automatic Classification): アクセスデータ, 著者, タイトル, その他によって出来るかぎり自動的に分類を行うこと。
- ③ 多重インデックス(Multiple Categories): 同一のドキュメントに対して出来るだけ多重インデックス化を図ること。
- ④ インフォメーションを検索する場合, いくつかのキーによる方法を用意すること。

等が重要となる。

以上に述べたことは, 多分に逆説的な問題をも含んでいる。

エレクトロニクスオフィス・システムを構築する際には, これまでの『机』の役割, つまり, ルーズな分類のままのインフォメーションの置き場所であること, 視覚的に自分は次に何をなすべきかを想起させてくれることも, 軽視してはならない非常に重要なファクタであることを銘記すべきである。

と, この論文は, 指摘している。

ユーザインタフェースの設計は, 試行錯誤的な評価を絶え間なく行う必要がある。つまり使う人の立場(実際に設計する人が使って実験台になる)になって設計することにより, よりよいフィードバックが期待

できる。作る人と使う人のバリエーションを越えた、アプローチが重要である。

参考文献

- 1) Smith, D.C., Irby, C.H. and Kimball, R.B.: The Star User Interface: An Overview, AFIPS Proceedings of National Computer Conference, Vol. 51, pp. 515-528 (1982).
- 2) Smith, D.C., Harslem, E.F., Irby, C.H., Kimball, R.B. and Verplank, W.L.: Designing the Star User Interface, Byte (Apr. 1982).
- 3) Xerox PARC Learning Reserch Group: Special Issue on Smalltalk, Byte, Vol. 6, pp. 168-194 (1981).
- 4) Zloof, M.M.: Query-by-Example, AFIPS Proceedings of National Computer Conference, Vol. 44, pp. 431-438 (1975).
- 5) Malone, Thomas, W.: How Do People Organize Their Desks? Implications for the Design of Office Information Systems, Proceedings of Conference on Office Information Systems.
- 6) 上林, 伊東, 上田: オフィスプロフェッショナルワークステーション JStar と統合プログラミング環境ワークステーション 1100 SIP, 本特集号.
- 7) 上林: Object Oriented User Interface Model, IMAC '82 (1982).
- 8) 上林: ワークステーションの実際, サイエンス別冊 (ニュービジネス特集号) (1983).
- 9) Seil, B.: Power Tools for Programmers, DATAMATION 1983年2月号 (邦訳:「プログラマ待望の強力なツールが登場」, 日経コンピュータ (1984年4月18日号).
- 10) Apple's LISA.: The Seybold report on Office Systems, Vol. 6, No. 2 (1983).
(昭和58年11月4日受付)