

解 説



1. ユーザインタフェース管理システムの基礎

1.1 ユーザインタフェース管理システム
の基本概念および対話の方式†

守 屋 慎 次††

1. はじめに

ユーザインタフェースは、機械（電子情報機器）の機能部分と人との間にあって、人と仕事と機械の相互間の不整合を最小化し、生産性と満足度と心地よさを最大化することを目的としている[MOR89]。すなわち人にとってユーザインタフェースは目的でなく手段である。手段に目を奪われ目的の達成を手間どるようでは本末が転倒する。手段は人の一部となって人と融けあい、人が透明と感じるのが理想である。

そのようなユーザインタフェースを、必要なときに、できれば安価に、しかも迅速につくれないものだろうか。これの解となることが期待され、重要性を増しつつあるのがユーザインタフェース管理システム（以後 UIMS, User Interface Management System）である。UIMS は応用プログラムにおけるグラフィカルユーザインタフェース部分（GUI）を効率よく構築するための実行ライブラリとその構築を支援するツールである。

UIMS について最も影響力のあった文献は [PFA 85] であろう。この本は 1985 年に出版されたが、その内容は 1983 年 11 月 1 日から 3 日にかけてドイツの Seeheim で開催された UIMS に関するワークショップでの論文と報告書である（224 頁，15 論文，4 報告，27 著者）。この本のまえがきでは UIMS を次のように定義している。

User Interface Management Systems are the mediators between the user and the application programs.

つまり UIMS は、利用者と応用プログラムの間に位置し、その役割は mediator である、という。このように初期における UIMS は、かなりのあいまいさを含んだままで設計しなければならない

システムであった。筆者の実感は、入力仕様（利用者とインタフェース）と出力仕様（応用プログラムとのインタフェース）のない、しかもその機能がほとんど明確でないプログラムをつくる問題であった。したがって、入力仕様と出力仕様、そして UIMS そのものの機能をどう定めるかが、UIMS 研究の課題であった。そしてこれは、UIMS 研究の今後の課題でもあり続けると思う。

ユーザインタフェースが利用者と応用プログラムとの間に位置するという考え方は、すなわち応用プログラムからユーザインタフェース（入出力部分）を分離する、ということの意味した。この考え方は、当時はかなりの影響力をもって急速に世界へ広がったように想像する。ユーザインタフェースという言葉さえ一般的でなかった当時は、応用プログラムのなかにその応用プログラム専用入出力部分を組み込むことにあまり疑問を感じない、という時代であったと思う。筆者は寡聞にして日本で UIMS が当時どのようにみられていたのかよく知らない。UIMS というテーマで企画された日本における学会活動は筆者の知る限り本特集が最初である。欧米では ACM の SIGCHI や SIGGRAPH を中心として何度も UIMS に関するワークショップやパネル討論やセッションがもたれ、UIMS の考え方そのものにたいする激しい応酬もあった。たとえば UIMS 研究の草分けの一人 W. Buxton (Toronto 大学) が IFIP '86 で行った UIMS に関する招待講演において、質問に立ったある高名な学者が Buxton の仕事を「no future」と決めつけた状況を思い出す。

本稿の目的は、上記の Seeheim でのワークショップの成果を紹介しながら、UIMS の基本概念を述べ、UIMS の今後について考えることである。本稿では、2. で UIMS の基本概念について述べ、3. では UIMS の今後を対話方式の今後という視点から述べる。

† Fundamental Concepts of User Interface Management Systems, and Interaction Styles by MORIYA, Shinji (Department of Electrical Communication Engineering, Tokyo Denki University).

†† 東京電機大学工学部電気通信工学科

2. UIMS の基本概念

2.1 UIMS の概念が生まれたころ

(1) こんな時期

文献 [PFA 85] が検討され出版された 1985 年前後はどんな時期だったろうか。UIMS が注目された背景や UIMS の考え方を理解するうえで、この時期のことをふりかえてみる価値がありそうに思う。

この時期を象徴的に表現する言葉はあるであろうか。一言で表現するという危険をあえてしてみよう。この時期は、コンピュータの要素技術が揃いつつあって、パーソナライゼーションとインテグレーションの大潮流が個別に始まりつつあった時期のように筆者は感ずる。メインフレーム時代におけるコンピュータ分野の主要問題が解決されつつあって、一見すると技術的な停滞のようにも感じたためであろうか、新しい何か（たとえば人工知能）に過大な期待が寄せられた時期でもあった。

この時期は Lisa 2, Macintosh, IBM PC が、また MS-DOS 2.0, Lotus 1-2-3, WordPerfect がでた直後であった。Star, UNIX, TEX はすでにあった。RISC と CISC 論争も盛んであった。現在これらはいずれも著名であるが、当時において今日の姿を予想していた人は希であったと思う。さて、その一方でその後に忘れられていくものが当時は健在であった。たとえば、CP/M, SNOBOL 4, Apple II, TRS-80, そして Ada。

1984 年 1 月の Byte 誌は The year of the 32-bit microprocessors. という特集を組み、同 7 月の Comm. of the ACM は Computer graphics comes of age. という論文をのせている。ユーザインタフェース関係では、Comm. of the ACM の 1984 年 4 月号が Humanization of Computer Interface 特集を、IEEE Computer の 1984 年 11 月号が Icon Driven Systems 特集を、同誌 1985 年 8 月号が Visual Programming 特集を、それぞれ組んでいる。

(2) ユーザインタフェース部分の大きさ

この時期、応用プログラムの総量のうちでユーザインタフェース部分（入出力部分）は何%ぐらいを占めていたのだろう。表-1 に 1987 年時点における 4 つの調査結果をのせた [MOR89]。割合は 30~80% の間にある。各調査における割合の最大値だけを見ると、年代が進むに従って、おおむね

表-1 対話型システムのうちユーザインタフェースが占める割合

調 査 者	調査年	対話型システムのうちユーザインタフェースが占める割合(%)	調査の対象と方法
Foley	1987	50~80	
守 屋	1987	35~60	テキストエディタ, CAD システムなど調査者が開発した 5 つのシステム
Bobrow, Mittel, Stefik	1986	30~50	エキスパートシステム
Sutton と Sprague	1978	59	22個のビジネスシステム

増加傾向にある。増加の傾向と割合の大きさは、この問題に対して広範囲にわたるかなり積極的な対応の必要性を示していた。

この理由のひとつとして通常のプログラム言語の不備をあげることができる。対話処理部分は通常、たったふたつの入出力機能、すなわち read 文と write 文（および format 文）だけで書かれていた。少々極論すれば、一括処理時間の名残りでだけ対話処理部が応用プログラムごとに書かれていたといえよう。すなわちユーザインタフェース部分の大きさの主原因は、ユーザインタフェースに対する利用者側の要求と、ユーザインタフェースを記述する言語やプログラムライブラリおよびコンピュータアーキテクチャとの間の、意味上の溝 (semantic gap) の大きさと深さであった。溝を埋める必然性が関係者の間で認識されつつあった時期といえる。

(3) ユーザインタフェースまわりのツール

この時期に発表されていたソフトウェアツールは、古くは UNIX ライブラリ中の Lex, Yacc, 当時のものでは GKS の論理入出力装置 (locator, pick, choice, valuator, string, stroke) 用のパッケージ [FOL84, HOP83], メニュー作成器 [SCH85], ウィンド管理システム [HOP86], TEX [KNU79] などのフォーマッタ, 対話技法ライブラリ [COU85], そして Exxon Office Systems 社の Ergonomic Database [GIL83] などがあった。これらのうち Macintosh や Star 以外の機械にたいして一般に出まわっていたのは Lex, Yacc 程度であった。なお、Macintosh の応用プログラム開発者用のツール MacApp が発表されたのは 1986 年 8 月のことである [SCH86]。

2.2 ユーザインタフェース管理システム

(1) 応用プログラムからの入出力部分の分離 [MOR86a, b]

入出力部の特性は、上で述べたような占める量の大きさだけではない。当時、ユーザインタフェースに関心をもつ人は、どの対話型システムの入出力部も比較的類似した構造をもつことを経験的に知っていたと思う。たとえば 2.1(3) で述べたように、コマンドの構文解析、メニュー管理、ウィンドウへの出力など、明らかにどの対話型システムにも出てきそうな類似の概念がみえていた。

したがって入出力部を対話型システムから分離し、その構造を分析してモジュール化することにより、有用な共用ツールを作成できる可能性が非常に大きいと考えるのは、自然であろう。

同様な考え方によって、応用プログラムからその当時まですでに分離されてきた共用ツールの例として、古くは入出力管理システムやデータベース管理システム、その当時では図形処理システム（国際標準案の段階であった GKS, PHIGS など [HOP83]）やウィンドウ管理システムなどをあげることができる。応用プログラムにおけるこのような分離傾向は計算機技術における「大きな潮流」のひとつと考えていた人が多かったように思う。

以上のような動機により、応用プログラムから入出力部分を分離した構成を図-1 に示す。

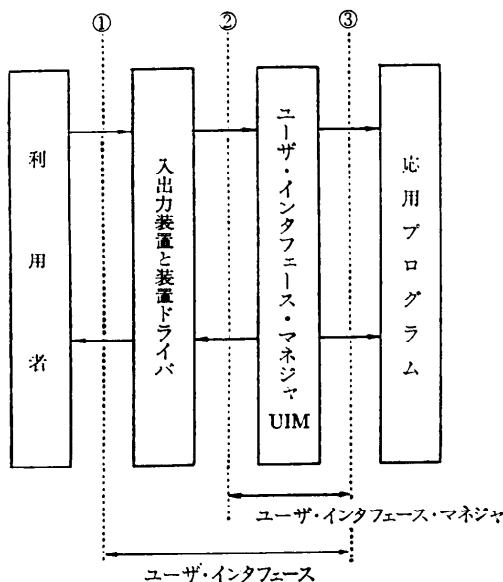


図-1 ユーザインタフェースマネージャ

(2) ユーザインタフェースとユーザインタフェース・マネージャ [MOR86a, b]

当時、ユーザインタフェースという言葉は定義なしにあいまいに使われることが多かった。ここでは当時の定義例 [MOR86a, b] を示してみよう。

ユーザインタフェースの範囲の最小と最大はそれぞれ下記のとおりである。

(1) 最小：図-1 の破線①上の情報

(2) 最大：図-1 の破線①および③上の情報と、①③間のすべての機能単位

ユーザインタフェースという言葉が表す意味との整合性と合理性という理由で、[MOR 86 a, b] では上記(2)の最大範囲をユーザインタフェースと呼んでいる。さらに、図-1 の②③間に位置する機能単位（プログラム）をユーザインタフェース・マネージャ (UIM: User Interface Manager) と呼び、ユーザインタフェースは次の三つからなるとしている。

(1) 入出力装置（と装置ドライバ）

(2) UIM

(3) 次に述べる機能単位間の接続方式

I 利用者と入出力装置 II 入出力装置と UIM
III UIM と応用プログラム

上記IIの接続方式すなわち図-1 の破線②の一例として、当時、国際標準案 CGI (Computer Graphics Interface) が検討されていた。上記Iの接続方式すなわち破線①上の情報は、コマンド、自然言語、音声、画像、アイコンなどである。

上記IIIの接続方式すなわち図-1 の破線③はまだ研究が始まったばかりであった [PFA85]。これは現在においても（おそらく将来においても）重要な研究課題であり、当時は、本稿の主題であるユーザインタフェース管理システムという名のもとで研究が盛んになり始めたころであった。当時、API (Application Programming Interface) という言葉は使われていなかったか、少なくとも一般的ではなかった。

(3) UIMS

当時においても応用プログラムは多種多様で、その数は増大の一途であった。それぞれの応用プログラムに対する UIM が実現しやすくなれば、その効果は非常に大きい。入出力装置も多種多様であるが、それぞれに対する UIM が実現しやすくなればその効果も大きい。利用者も多様で、それ

それぞれの個性や特性をもった利用者に対する UIM が実現しやすくなれば、その効果はやはり大きい。

多種多様な応用プログラムや入出力装置、そして個々の利用者に応じた UIM を構成することを究極の目的としたシステムとして UIMS が考えられた。すなわち UIMS に対して、応用プログラム、入出力装置、そして利用者に関する仕様や条件を入力してやると、仕様や条件に合う UIM が(半)自動的に構成されるというものである。

(4) UIMS の利点

UIMS の利点として次のものが考えられていた [HAY85, PFA85]。

① インタフェース構築の省力化

インタフェースをプログラミング言語で直接書くよりは、UIMS のインタフェース定義法によって仕様や条件を与えるほうが簡単であろう。

② より高性能、より高級なインタフェース

上記①で省力された余力をもって、より高度なインタフェース機能、エラー対策、利用者への学習機能などが提供できよう。

③ 人間工学や認知科学の成果をインタフェース設計に反映

応用プログラムの設計にはその分野の専門家が専念でき、インタフェースの設計には人間工学や認知科学を含めた専門家が参加できよう。

④ 汎用性のあるインタフェース

複数の応用プログラムに利用可能なインタフェースが存在することになる。したがって利用者による学習の労力が軽減されよう。

⑤ 単一の応用にも複数のインタフェースを用意可能

インタフェースの定義を応用プログラムとは独立に行うことができれば、ひとつの応用プログラムに複数個のインタフェースを用意することが可能となろう。これによりひとつの応用プログラムを異なる入出力装置上で使ったり、利用者の知識や熟練度に応じたインタフェースを用意することも可能になる。

上記のいくつかはすでに現実のものになっている。また上記の多くはいまでも対話型システムがめざす方向を示していると思う。

2.3 UIMS の構成法

(1) モデル

当時考えられていた UIMS まわりの情報の流

れを図-2 に示した。UIM 設計者は定められた言語や表を用いて UIM 仕様を記述する場合が多かった [CAR85, FOL86, SCH85]。仕様は UIMS 内のトランスレータで変換され、その結果が UIM となる。UIM と応用プログラムは別々のプログラムとして作られるので、両者にまたがる共用情報(橋渡しの情報、たとえば UIM から応用プログラムへ、またその逆方向に渡される情報の構造など)を UIMS と応用プログラムへ与えてやることもある [MON86, SCH85]。

初期のころにおける UIMS の構造やモデルは [COU 85, MON 86, PFA 85, SHA 84, SHA 86] などにみられる。

(2) 応用プログラムの構成

UIM は応用プログラムから分離された機能を含んでいる。そして UIM と応用プログラムは接続され、情報の授受を矛盾なく円滑に遂行しなければならない。したがって UIM の構成法は応用プログラムの構造と独立に考えることはできない。しかし、一方では UIM を応用プログラムからできるかぎり独立した機能単位として構成したいという要求もある。その場合は応用プログラムの作成法になんらかの規格を設ける必要が生ずる。

UIM と応用プログラムのインタフェース(図-1 の破線③)は UIMS 研究の最大の問題のひとつであった。これはまた同時に、応用プログラムの作成法やハードウェアに影響するという点から、ソフトウェア工学や計算機アーキテクチャにおけ

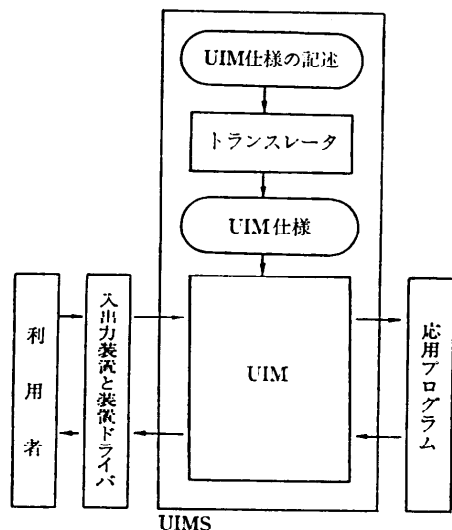


図-2 ユーザインタフェース管理システムの構成

る重要な問題のひとつでもあった。

応用プログラムの主要な構成法として、次のふたつが考えられていた^[PFA85]。

(1) 内部制御 (internal control) 方式

(2) 外部制御 (external control) 方式

このふたつの違いは、制御権が応用プログラムと UIM のどちらにあるという点にある。内部制御は応用プログラム内に、外部制御は応用プログラム外 (UIM や利用者) にある。

内部制御方式は、UIM がパッケージ化 (単位仕事にライブラリ化) されていて、応用プログラムから必要に応じて UIM パッケージが呼び出される方式。外部制御方式はその逆で、応用プログラムがパッケージ化されていて、UIM から必要に応じて応用プログラムのパッケージが呼び出される方式である。

内部制御方式は、当時 (現在も) 市販されていた対話型システムの延長線上にあった。利用者は応用プログラムからの入力要求に応じて対話を進行する。この意味で内部制御はより機械主導の方式といえる。

外部制御は、次の意味でより利用者主導の方式といえる^[TAK85]。第一に、利用者は応用プログラムの入力要求に単にこたえるだけでなく、入力要求がなくても応用プログラムに指令できる。第二に、モードなしの操作が実現可能である。モードという一種の免許証なしに、応用プログラム中のパッケージを直接呼び出して運転できる。このほかの内部・外部制御方式の比較については [TAK85] がくわしい。

なお現在の UIMS における制御方式の動向については 3.3(3) で述べる。

(3) 初期のころに開発された UIMS

ここでは 1986 年ころまでに開発された UIMS について概要を述べる。UIMS のプロトタイプとしては次のものがあった。

大学などの研究で作成されたものとして、Toronto 大学の MENULAY^[BUX83]、Brigham Young 大学の GRINS^[OLS85]、AT & T の Squeak^[CAR85]、Alberta 大学の UIMS^[GRE85] などがあった。我が国では山梨大学^[IMA85]、東京大学^[YAM85]、東京電機大学^[MON86] などがあった。

市販中のものでは TRW 社の Flair^[WON82]、Apollo 社の ADM^[SCH85] である。[FOL86] の調

査によると、このほかに Precision Visual 社の ENTER/ACT、Unicad 社の UIMS、Boeing Computer Service 社の Tiger/Ticcl、Rubel Software 社の Blox、Cadlinc 社の UIMS があったという。

これらのうち現在にまで残って競争しているシステムはどれであろうか。筆者らのシステムは競争から脱落した。この類のシステム、すなわち、初期の時代には概念そのものが不確かで、しかも、主流のひとつにまで成長していくシステムを構築していくには、すぐれた構想と問題意識、幅広い機器やシステムへの対応、たゆみない積み上げと改良、そして何よりも不屈の信念がそのリードに求められるように思う。

1987 年、筆者にとっては衝撃的な研究発表があった。それは SIGCHI '87 において Toronto 大学の B. A. Myers (現在、Carnegie-Mellon 大学) が発表した Peridot システム^[MYE87] である。当時ほとんどの UIMS は、有限オートマトンや状態遷移表を用いて UIM の仕様を記述していた^[NEW68, WAS85]。この記述をすること自身が重荷であることを関係者は熟知しており、筆者は (おそらく他の UIMS 研究者も) いずれはこの記述をメニューとマウスで肩代りできるようにしなければと考えながらも、そこまで到達できずにいた時期であった。Peridot ではすでにこれを達成していたのである。発表の最後に Myers が語ったのは、Peridot を用いて Macintosh のユーザインタフェースが構築可能ということであった。彼の研究はその後も発展をつづけ、現在は Garnet システム^[MYE91] として結実し多くの人に影響をあたえている。

3. UIMS の今後、対話方式の今後

本章では UIMS の今後について考えてみたい。マクロな視点から、できるかぎりではあるがおおきな動向をみてみたい。そのために、まず対話方式のおおきな動向をみてみよう。UIMS はユーザインタフェースを作成するためのツールであるから、マクロな視点からは、対話方式の動向におおきく左右されると考えてよい。

3.1 と 3.2 では UIMS の話題からすこし離れ、UIMS の将来をマクロにとらえるための準備として、対話の方式についてのべる。3.3 で UIMS の今後について考える。

3.1 対話に関する基本概念

(1) 場^[MON89]

対話における場は、人が住む「実世界」とコンピュータ内の「仮想世界」との境界上にあり、一方の世界から他方の世界に向けて情報が変換される場所である。対話における場には入力と出力の場がある。入力場の例としてキーボード面がある。手や指の動きが符号の列に変換される。出力場の例としてディスプレイ画面がある。符号の列が光に変換される。

(2) 指示と操作、直接と間接^[MOR91]

例題として、ペン入力による対話をとりあげてみよう。ペン入力において入・出力と編集に用いる道具は、終始、スタイラスとよばれるペン1本である。この1本のペンを、たとえばメニュー選択用に、筆記用に、消しゴム用に、見えゴム（消えた部分の復元）用に、そして指示棒用に使いわけられる。利用者はスタイラスペンを手にしてペン先で対象を指示し（すなわちペンで直接指示し）、対象（筆跡など）の上でペンを押下して動かすことによって対象を操作する（すなわちペンで直接操作する）。このように、直接指示・直接操作の方式をとっている。

上の手順を、通常の対話型システムでは次のように行っている。利用者はマウスを手にしてカーソルによって対象を指示し（すなわちマウスで間接指示し）、マウスのボタンを押して対象（文字列など）を操作する（すなわちボタンで直接操作する）。このように、間接指示・直接操作の方式をとっている。

(3) 相互参照場、場の融合^[MON89]

たとえば画面エディタにおいて、キー入力した文字をバックスペースキーで削除する場合や、文章中のカーソル位置へ文字列を挿入する場合を考えてみよう。利用者は、ウィンド上に視覚化された文字列に対して削除や挿入などを（キーボードやマウス経由で）直接的に操作できる。このときカーソルの位置は、机上のキーボードやマウスをもちいて間接的に指示することになる。すなわちこれは、間接指示・直接操作の場合である。

この方式における循環情報の環を調べてよう。利用者側からみると「大脳中枢→手→キーボード（またはマウス）→画面上の情報とその変化→目→大脳中枢」という環に感ずるであろう。画面エディ

タ側からみると「画面上の情報とその変化→画面エディタによるデータ構造の操作→画面上の情報とその変化」という環になる。

利用者側とコンピュータ側の二つの環が、かみ合う二つの歯車のように同期して回転する。回転の動力となるのは利用者である。歯車がかみ合う所は「画面上の情報とその変化」の場であり、ここを相互参照場と呼ぶことにする。相互参照の名は、利用者が入力した指令やデータをコンピュータが参照し、コンピュータが出力した情報を人が見たり聞いたりして参照する、という双方向の流れの合流点を意味する。対話の流れにおける交差点であり、要衝といってもよい。inter-referential I/O という呼び名を用いて相互参照のことを説明したのは [DRA 86] である。

相互参照場は、利用者とコンピュータの双方による、操作の場が融合したものと考えることができる。しかし、利用者による指示の場は机上にあり、コンピュータによる指示（すなわちカーソル）の場は画面上にあって、指示の場は非融合の状態にある。

(4) 人の入力場と機械の出力場との融合^[MOR91]

● 人の入力場を機械の出力場へ融合する

この例として人工現実感がある。たとえば頭部搭載型のディスプレイを2眼で立体視している人の場合、人の視界（という場）と機械が生成する立体的な場とは「等しい」といえる。この場合、人の視界と機械が生成する場とは「融合」しているといえる。これは、人が機械へ大きく歩み寄っている例といえる。

● 機械の出力場を人の入力場へ融合する

この例としてペンコンピュータとタッチスクリーンがある。ペン入力の利用者がスタイラスペンを用いて筆記する場合は、紙の上でペンを用いて筆記する場と「ほぼ等しい」といえる。また、ペンコンピュータの利用者がその文書画面上にみる筆跡は、紙の上にみる筆跡と「ほぼ等しい」といえる。この意味で、ペンコンピュータでは入力場と出力場の両方とも、機械の出力場を人の入力場へ融合する方式となっている。タッチスクリーンも同様である。

これらの例から分かるように直接指示が可能となるためには、人の運動器（指や手や目や口）と

ほぼ「一体」となって働く入力装置（たとえばタッチパネル、スタイラスペン、視線検出器、データグローブ）によって直接、仮想世界の対象を指し示すことができ、それを仮想世界が検知するようにはしておかねばならない。

これは、入力する位置（入力場）と出力する位置（出力場）が（ほぼ）一致していることを必要としている。言いかえると、入力場と出力場の、部分もしくは全体がほぼ融合している必要がある。

上記のことをすこし厳密にいうと次のようになる。コンピュータが生成表示している対象を利用者が直接的に指し示すことができ、しかもコンピュータがそれを受理できるとき、すなわち直接指示できるとき、物理的（すなわちハードウェアによって）と論理的（すなわちソフトウェアによって）の両者において入・出力場が融合している。またこの逆が成立することも明らかである。したがって、直接指示が可能となるための必要十分条件は、物理的と論理的の両者において入・出力場が融合することである。

入・出力場が融合したシステムの例として、先に述べたように、ペンコンピュータ、人工現実感、タッチスクリーン、および音声認識・合成システムがある。すなわち、これらの4システムは直接指示・操作という対話方式の下に同類である。

3.2 これまでに開発されてきた対話の

方式^[MOR 90]

3.1 で述べたことを対話方式の発展経過という視点からまとめてみよう。ただしここでは、これま

でに開発されてきた方式だけに焦点をあて、将来の対話方式については3.3で述べることにする。

人間とコンピュータとの対話を型どってきた、これまでににおける代表的な方式は、一括処理（バッチ処理）、コマンド言語、間接指示・直接操作、直接指示・直接操作の4つである。表-2は、これら四方式と入出力場の融合との関係をあらわしている。

● 一括処理方式

利用者は、まず紙カード上（または紙テープ上）に、オフラインの紙カード穿孔機を用いてプログラムとジョブ制御言語を穿孔した。次に、その紙カードの束を紙カード読み取り機の上に載せ、コンピュータに読み込ませた。結果はラインプリンタに出力された。対話に必要な循環情報の環が物理的に形成されていなかった。対話の相互参照場は存在しなかった（厳密に言えばコンソールのランプとボタンが相互参照場）。

● コマンド言語による対話方式

人間とコンピュータはキーボードとディスプレイ（初期のものは電動タイプライタ）を経由して物理的に循環情報の環を形成した。この考え方は現代の対話型システムの原型になったという意味で画期的といってよい。しかし、キーボード入力されるコマンドの表示（または電動タイプライタ用紙上への印字）とコンピュータ出力の表示（または印字）とが、画面上（または用紙上）で流れてスクロールされていた（例、MS-DOS、ラインエディタ）。すなわち、コンピュータ出力画面上（または用紙上）の情報を利用者が直接的に操作

表-2 対話の方式と入出力場の融合

装置分類		人間から機械へ		機械から人間へ	
対話方式	人による指令 入力装置	指令表示 装置	指令操作装置	指令表示装置	指令操作 装置
一括処理方式	カード穿孔機 紙テープ穿孔機	カード 紙テープ	カード読み取り機 紙テープ読み取り機	コンソール ラインプリンタ タイプライタ	メモリ
コマンド言語による 対話方式	キーボード	ディスプレイ			
		画面表示の最下行		画面上の、入力されたコマ ンドの下	
間接指示・直接操作 する対話方式	キーボードまたは はマウス	・ディスプレイ ・画面上のオブジェクト(ウィンドメニューなど)			
直接指示・直接 操作する対 話方式	実 空間	例・入出力一体型の装置とペン ・画面上のオブジェクト			
	仮想 空間	例・頭部搭載型ディスプレイとデータグローブ ・立体視空間オブジェクト			

することはできず、したがって論理的には相互参照場はまだ形成されていない。

- 間接指示・直接操作方式

人間とコンピュータは、物理的にも論理的にも、操作に関しては相互参照場をもち直接操作が可能となった。しかし、マウスなどの指示具は間接的に操作された。すなわち、指示における入力と出力の物理的な位置が異なっていた。

- 直接指示・直接操作方式

入力と出力それぞれの、物理的および論理的な場が融合し、そこが相互参照場となった。したがって直接指示が可能となった。

3.3 将来の UIMS, 将来の対話方式

本節では、対話方式の発展の視点から UIMS の将来について考えてみよう。そのために将来の対話方式についても考えてみる。

(1) UIMS と相互参照場

UIMS の研究が盛んになり始めたのは、間接指示・直接操作方式の対話型システムが盛んに作り出されたところである。いいかえると、間接指示・直接操作における相互参照場（すなわち UIM）をソフトウェアによって実現するうえで、すなわち論理的な相互参照場を実現するうえで、2.1(2)で述べたように膨大なコーディングが必要になり始めたところである。2.2で述べた UIM は相互参照場を形成するためのプログラムであった。このことから、UIMS に関する今後のおおきな方向づけは相互参照場の視点からながめるのがよいと筆者は考えている。3.1 と 3.2 で UIMS の話題から離れて、相互参照場などに紙数を使った理由はここにある。

したがって UIMS の今後の課題として、直接指示方式における論理的な相互参照場に対応していくことがあげられる。直接指示すなわち入・出力場が融合したシステムの例として、3.1(4)で述べたように、ペンコンピュータ、人工現実感、タッチスクリーン、および音声認識・合成システムなどがある。

(2) 意味的な処理もする相互参照場

3.2 でのべた 4 つの対話方式は、おおまかには相互参照場の有無によって特徴づけられ、さらに物理的、論理的というキーワードによって細分されている。このような考え方をもちいて、次に来る（来つつある）ものに推定してみよう。

3.1 で述べたように相互参照の名は、利用者が入力した指令やデータをコンピュータが参照し、コンピュータが出力した情報を人が見たり聞いたりして参照する、ということを示している。ここで重要なのは「参照」である。

筆者は、物理的参照、論理的参照の次に、意味的 (semantic) 参照がすでに必要になっていると考える。意味的な相互参照場における仕事の例として次がある。ワープロ画面上で英字を入力した時点においてスペルチェックや構文チェックをしてくれる、VLSI 配線の CAD 画面上で配線した時点において配線規則の検査をしてくれる、などである [MAR92]。

したがって今後の UIMS として、主として間接指示と直接指示において、意味的な相互参照場を構成する問題が考えられる。意味的な処理のアーキテクチャを整え、意味的な処理ライブラリを積み上げ、それらを使って相互参照場 (UIM) を組み立てる方式を確立する必要がある。

(3) UIMS と応用プログラムは共同で制御

意味的な処理は、その多くは応用プログラムの仕事である。したがって、意味的な処理をとまなう相互参照場 (UIM) と応用プログラムとは、たがいに合流（融合）していくことを意味する。UIM は、ひとたびは応用プログラムから分離され、ふたたび融合していくことを意味する。このことを [MAR 92] は次のように述べている。「今後のユーザインタフェースは、人とコンピュータのコミュニケーションの初期段階にたちもどる。つまり応用プログラムとユーザインタフェースの境界は曖昧になるだろう。」「したがって、UIMS と応用プログラムは共同で対話の制御を行うことになる。」

(4) 人体と機械とが物理的に融合する

現時点における最も先進的な対話方式をあらわすキーワードは直接指示である。直接指示は、人体表面の動きを直接的にコンピュータへつたえ、コンピュータ出力を直接的に人体表面へつたえる方式、といいかえることができる。したがってこの方式では人体表面がコンピュータの入出力場といってよい。この考え方を延長してその先を推定してみよう。

直接指示よりもさらに直接的な対話となると、人体の内部（の器官など）から情報をとりだして

それをコンピュータに入力し、コンピュータからの情報を人体の内部に対して直接的に出力する、という方式が考えられる。すなわち人体の内部がコンピュータの入力場かつ出力場となる。たとえば人の網膜上に光を照射して走査することにより（ディスプレイを経由せず）像を視神経につたえる、という研究が人工現実感の分野で始まっているという〔HIR92〕。

このような方式にあえて命名すると、内接方式または人体と機械とが物理的に融合した方式とよべる。これにたいして3.2で述べた四方式は、外接方式または人体と機械とが物理的に非融合の方式とよべる。

さて、人体の内部に相互参照場をどのようにして形成するのだろうか。人体内の器官と内接する相互参照場は個人差が大きいであろうか。たとえば医療に応用されたとすれば、内接する相互参照場は緊急に形成する必要にせまられるのだろうか。そのようなとき UIMS の活躍の場が生まれて（い）るのではないだろうか。

4. おわりに

UIMS の基本概念と UIMS の今後について述べた。基本概念については、〔PFA 85〕が出版された 1985 年前後における、ユーザインタフェース作りの状況や問題点、応用プログラムからユーザインタフェースを分離すること、そして UIMS の構成法などについて述べた。

UIMS の今後については、対話方式の発展というおおきな流れのうえで UIMS を位置づけることにより、対話方式の今後という視点から UIMS の今後を考えた。これにより、UIMS の今後の対応はおおきくは次の三つ、すなわち直接指示による対話への対応、意味処理をともなう対話への対応、そして内接方式（人体と機械とが物理的に融合した方式）による対話への対応、があることを示した。なお、意味処理をともなう対話への対応のひとつとして、ひとつたびは応用プログラムから分離されてきた UIM が、ふたたび融合していくことを述べた。

一般に利用者が目にするコンピュータの姿は、徐々に、また一部ではすでに、ユーザインタフェースそのものとなりつつある。それを示す数値としては、ひとつは2.1(2)で紹介した、応用プ

ログラムの総量のうちでユーザインタフェース部分が占める割合がすでに 80% に達していることがあり、もうひとつはプロセッサの処理時間のうちでユーザインタフェース部分が占める割合で、テキサス・インスツルメンツ社のハイルマイヤ副社長によると、将来マンサイクルの 90% がユーザインタフェースに費やされるという予想である〔JMO90〕。

このように圧倒的な重みを感じさせるユーザインタフェースを、必要なときに迅速かつ安価に設計し製作するうえで、UIMS はその重要性を増していかなるをえないであろう。

本稿は主として 1985 年前後に焦点をしばって UIMS まわりを解説したが、この時期だけでも数おおくの研究があり、紙数の都合上、重要な成果のすべてを紹介できたわけではないことを最後に述べておきたい。

参 考 文 献

- [BUX 83] Buxton, W. et al.: Towards a Comprehensive User Interface Management System, Proceedings SIGGRAPH '83, published as Computer Graphics 17 (3), pp. 35-42 (1983).
- [CAR 85] Cardelli, L. et al.: Squeak: A Language for Communication with Mice, ACM SIGGRAPH '85, pp. 199-204 (1985.7).
- [COU 85] Coutaz, J.: Abstractions for User Interface Design, IEEE Computer, Graphics and Applications, pp. 21-34 (1985.9).
- [DRA 86] Draper, S.D.: Display Managers as the Basis for User-Machine Communication, User Centered System Design (Ed. Norman, D.A. and Draper, S.W.), Lawrence Erlbaum Associates, Publishers, pp. 339-360 (1986).
- [FOL 84] Foley, J.D.: The Human Factors of Computer Graphics Interaction Techniques, IEEE Computer Graphics and Applications, pp. 13-48 (1984.11).
- [FOL 86] Foley, J.D.: Managing the Design of User Computer Interfaces, ACM/SIGCHI CHI '86, Tutorial 15 (1986.4).
- [GIL 83] Gilfoil, D.W., Murray, J.T. and Praag, J.V.: The Development and Application of an Ergonomic Database Management System, Proc. Human Factors Society 27th Annual Meeting, pp. 892-895 (1983).
- [GRE 85] Green, M.: The University of Alberta User Interface Management System, ACM SIGGRAPH '85, pp. 205-213

- (1985. 7).
- [HAY 85] Hays, P. J. et al.: Design Alternatives for User Interface Management Systems Based on Experience with COUSIN, ACM/SIGCHI '85 Proceedings, pp. 169-175 (1985. 4).
- [HIR 92] 広瀬通孝: 広がる人工現実感の技術, 日本機械学会, 第2回デザイン・エンジニアリング・プラザ「ヒューマンインタフェース」(1992. 6).
- [HOP 83] Hopgood, F. R. A. et al.: Introduction to the Graphical Kernel System GKS, Academic Press (1983); 吉川弘之監訳, 「コンピュータ・グラフィックス基本ソフトウェア・GKS」, 啓学出版.
- [HOP 86] Hopgood, F. R. A. et al.: Methodology of Window Management, Springer-Verlag (1986).
- [IMA 85] 今宮淳美, 宮武明義: 柔軟なユーザインタフェースの作成, 情報処理学会「グラフィックスとCAD」シンポジウム(1985. 12).
- [JMO 90] ジェフ・モリス: コンピュータがコンピュータでなくなる日, 日経コンピュータ, pp. 173-176 (1990. 8. 27).
- [KNU 79] Knuth, D. E.: TEX and METAFONT, American Mathematical Society (1979).
- [MAR 92] Marcus, A. and van Dam, A.: 90年代のユーザインタフェースを展望する, 日経エレクトロニクス, No. 547, pp. 297-313 (1992. 2).
- [MON 86] 守屋慎次, 中谷吉久, 大塚誠二, 斉藤剛: 適応性のあるユーザインタフェースの構成方式について, 計測制御学会第2回ヒューマン・インタフェース・シンポジウム (1986. 10).
- [MON 89] 守屋慎次, 中谷吉久: コンピュータの対話機構と直接操作インタフェース, システム制御情報学会誌, Vol. 33, No. 11, pp. 568-575 (1989).
- [MOR 86a] 守屋慎次: 人間・機械インタフェース, 中田, 南, 平松編「情報信号処理」第6章, オーム社 (1986).
- [MOR 86b] 守屋慎次: ユーザインタフェースおよびその管理システムとその標準化, PIXEL, pp. 61-66 (1986. 12).
- [MOR 89] 守屋慎次: ユーザインタフェースの研究・開発動向—そのニーズとシーザー, 日経コンピュータ別冊「マルチメディア時代のユーザインタフェース」, pp. 21-36 (1989).
- [MOR 90] 守屋慎次: 人とコンピュータとの対話の大局的な機構, 情報処理学会グラフィックスとCAD研究会 (1990).
- [MOR 91] 守屋慎次他: ストロークエディタと直接指示・操作方式, 情報処理学会論文誌, Vol. 32, No. 8, pp. 1022-1029 (1991).
- [MYE 87] Myers, B. A.: Creating Dynamic Interaction Techniques by Deconstruction, Proceedings SIGCHI '87, pp. 271-278 (1987).
- [MYE 91] Myers, B. A. et al.: Garnet: 対話性にすぐれ, 柔軟な GUI の作成を支援するツール, 日経エレクトロニクス, No. 522, pp. 187-203 (1991. 3).
- [NEW 68] Newman, W. M.: A Graphical Technique for Numerical Input, Computing Journal, Vol. 11, pp. 63-64 (1968).
- [OLS 85] Olson, J. D. R. et al.: Input/Output Linkage in a User Interface Management System, ACM SIGGRAPH '85, pp. 191-197 (1985. 7).
- [PFA 85] Pfaff, G. E.: User Interface Management Systems, Springer-Verlag (1985).
- [SCH 85] Schulert, A. J. et al.: ADM—A Dialog Manager, Human Factors in Computing Systems, Proceedings SIGCHI '85, pp. 177-183 (1985).
- [SCH 86] Schmucker, K. J.: MacApp: An Application Framework, Byte, pp. 189-194 (Aug. 1986).
- [SHA 84] Shackel, B. (ed.): Human-Computer Interaction-INTERACT '84, North-Holland (1984).
- [SHA 86] Shaw, M.: An Input-Output Model for Interactive System, ACM/SIGCHI '86 Proceedings (1986. 4).
- [TAK 85] Takala, Y.: Communication Mediator—A Structure for UIMS, Pfaff, G. E. (ed.), User Interface Management Systems, Springer-Verlag (1985).
- [WAS 85] Wasserman, A. I. and Shewmake: The Role of Prototypes in the User Software Engineering (USE) Methodology, In Hartson, H. R. (Ed.), Advances in Human-Computer Interaction, Norwood, N. J.: Ablex Publishing (1985).
- [WON 82] Wong, P. and Reid, E.: Flair-User Interface Management Tool, Proceedings SIGGRAPH '82, published as Computer Graphics 16 (3), pp. 87-98 (1982).
- [YAM 85] 山口 泰, 木村文彦: 形状モデリングにおけるユーザインタフェース管理, 情報処理学会第31回全国大会 (1985. 9).
(平成4年8月5日受付)

守屋 慎次 (正会員)



昭和48年東京電機大学大学院博士課程修了。工学博士。現在、東京電機大学工学部電気通信工学科教授。昭和56年ニューヨーク州立大学、昭和57年イリノイ大学の各計算機科学科客員準教授。ユーザインタフェース、ペン入力システム、パターン認識、グループウェア、音声応用技法、インタラクシヨンの分析・評価・モデリング法の研究に従事。Interacting with Computers 誌の Special Editorial Board, 電子情報通信学会, 計測自動制御学会, 人間工学会, テレビジョン学会, ACM, IEEE 各会員。