

ユーザインタフェースの設計へのアプローチ

An Approach to Design of Human Computer Interface

溝口 文雄*
Fumio MIZOGUCHI

* 東京理科大学理工学部経営工学科
Dept. of Industrial Administration, Science University of Tokyo.

1987年3月6日 受理

Keywords: user interface, expert system, mixed initiative, exploratory drive, object-oriented, PSI machine.

1. はじめに

新しい計算機環境や利用方式が開発されるに伴って“使いやすい”、“ユーザフレンドリな”といった意味のインタフェース (Interface) が注目されている。もともと、インタフェースというのは異なる装置間の結合のために使われた言葉である。今でも通信用のインタフェースは RS-232C を用いて、コンピュータと接続するというのはごく普通に使われている表現である。

ところが、ユーザインタフェースとなると、ハードウェアとしてのインタフェースのイメージは消えて、“使いごこち”、“ユーザになじみ深い”といったソフトウェア中心の内容になることが多い。すなわち、使う側に立って、ソフトウェアの開発をどのように高度化していくかということであろう。ただし、高度化というのは、高い生産性と高度の利用形態を意味している。こうした考え方の背景は、基本的にはこれからはソフトウェアが鍵であるという認識に立っている。つまり、ハードウェアの高度化に対応して、どのようにしてそうしたハードウェアを使いこなしてソフトウェアの開発を高度化していくかという視点が、インタフェースの議論には必要である⁽¹⁾。

本稿は、ソフトウェアの開発にとっての高度化のうちでも、高度の利用形態、すなわち、人工知能の応用におけるインタフェースの設計を中心に述べていく。もちろん、応用システムの開発には、ソフトウェアの生産性の問題を含んでいるので、厳密には、生産性と高度の利用形態の両者は分離することはできないこと

を断っておく。また、インタフェースの問題は、一般論から始めるのではなく、具体的な例から考えていくことが必要である。つまり、話の進め方はボトムアップ的に例を示し、次に、それを整理する形でインタフェースの設計のための支援環境は何かを説明し、そして、一般的にインタフェースの持つ機能について検討する。最後に、インタフェースの設計の今後の動向や進め方について触れてみたい。

なお、筆者らの立場はこの数年来、人工知能の応用システム、特に、医療診断や故障診断のエキスパートシステムを開発しているところにある。したがって、本稿も結果的に応用システムのインタフェースに焦点をあてたものとなっている⁽²⁾。

2. ユーザインタフェースについてのいくつかの事例

現在までに開発された人工知能の応用システムのうちで、筆者が“さすがに良いインタフェースを持ったシステム”だと思うのは ONCOCIN⁽³⁾ (Stanford Univ.), STEAMER⁽⁴⁾ (Naval Personnel Research and Development Center) および Dipmeter Advisor⁽⁵⁾ (Schlumberger-Doll Research) である。もちろん、そのほか Xerox PARC の LOOPS⁽⁶⁾ のデモなども、すぐれたインタフェースを持っていることはわかるが、前者のシステムと比較すると、実用性という点ではやや見劣りがする。ONCOCIN は、Shortliffe, E. H. が作成したデモ用のビデオを、STEAMER および Dipmeter Advisor は、実際のデモンストレー

ションを、見たときの筆者の感想である。

ONCOCIN は 1979 年以來、開発が続けられている医療治療の支援システムである。特徴は、診断が決定した後に、抗ガン剤をどのように投与していくかを医師にアドバイスするところにある。通常、抗ガン剤は副作用も多く、また、時間的要因(進行状況)、あるいは、どのようなテストをするかなど、個々の患者のプロトコルを医師自身が全部記憶することができない。そのために患者の記録を管理し、次の治療に必要な情報を維持していくのが ONCOCIN の主な役割である。したがって、もっとも重要なことは、患者の記録と治療方法(計画)を医師自身が操作し、利用しやすい支援システムにすることである。そのために、ONCOCIN のインタフェースは oncology のクリニックで使用されているデータシートのイメージをそのままシミュレートするという方式を採用している。つまり、医師が日常使っているカルテや、患者のデータシートのイメージが、LISP マシンのビットマップディスプレイ上で見え、ほとんどの操作がマウスで行うことができるというものである。こうしたインタフェースの技術は、いわゆるスプレッドシート方式を採用したものである。データ処理についてもうひとつの利点は、図 1 に示すような身体イメージを使ってガ

ンの転移部の情報を加え、さらに、その部分を抽出し拡大することができる点である。このように ONCOCIN のインタフェースは、日常診療のデータシートイメージが使えて、さらに、人工知能の手法を用いた推論プロセスを表示する機能などが統合されている点に大きな特徴を持っている。外側だけを見ていると、“人工知能的”でないという批判もあるようであるが、使えるインタフェースを持ったシステムは、大げさな人工知能の手法を明示的(explicit)に使っていくのではなく、むしろ、そうした手法が隠された機能として組み込まれている点に注目する必要がある。

STEAMER は蒸気エンジン系の教育システムで、1981 年に BBN (Bolt Beranek & Newman) の Stevens, A. らによって報告されている。このシステムの目的は、知識ベースシステムの技術が低コストの訓練システムの設計に役立つかを開発を通して評価することにあった。対象領域としては蒸気推進エンジン系であり、どのような因果関係で制御系が作動するかを生徒に理解させることが訓練の目標であった。蒸気推進エンジン系というのは、船のもっとも重要な要素であり、空間的にも船の1/3を占めている。そして、複雑なパイプラインで各々の要素(約 3000 個)が結合している。系の制御には 16~20 名のオペレータが必

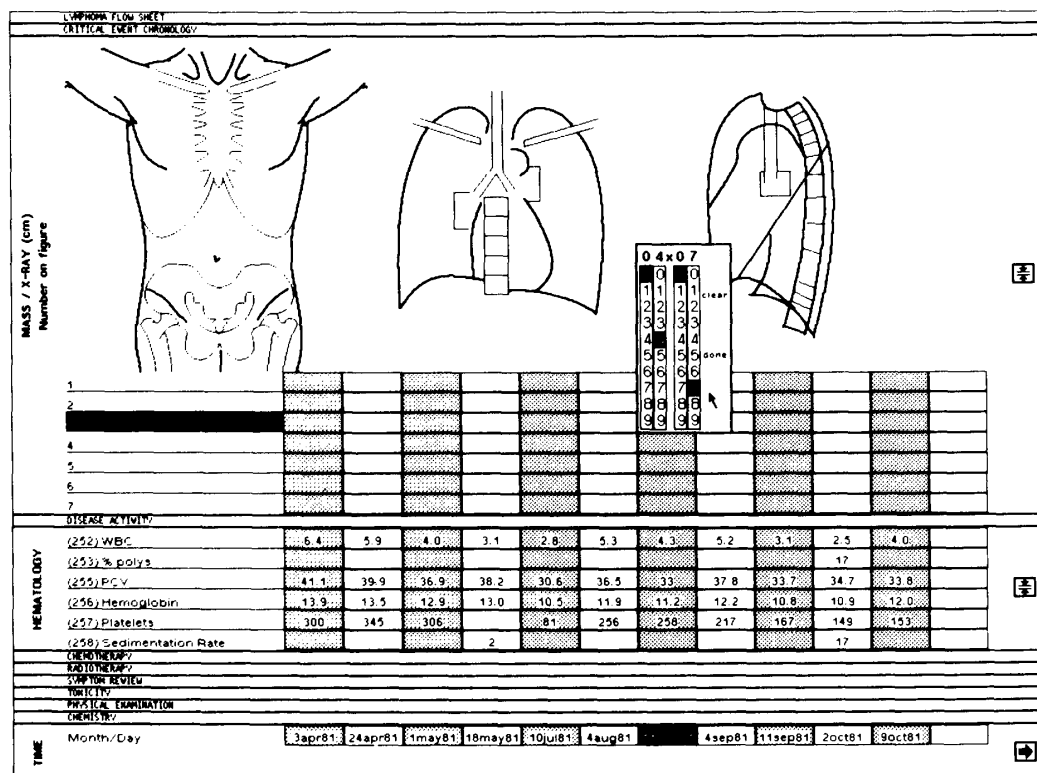


図 1 ONCOCINにおけるフローシートの出力例(マウスによりポップアップメニューの選択を行っているところ)

(出典: A Publication of the Knowledge Systems Laboratory, Fig. 8 of page 38, 1985)

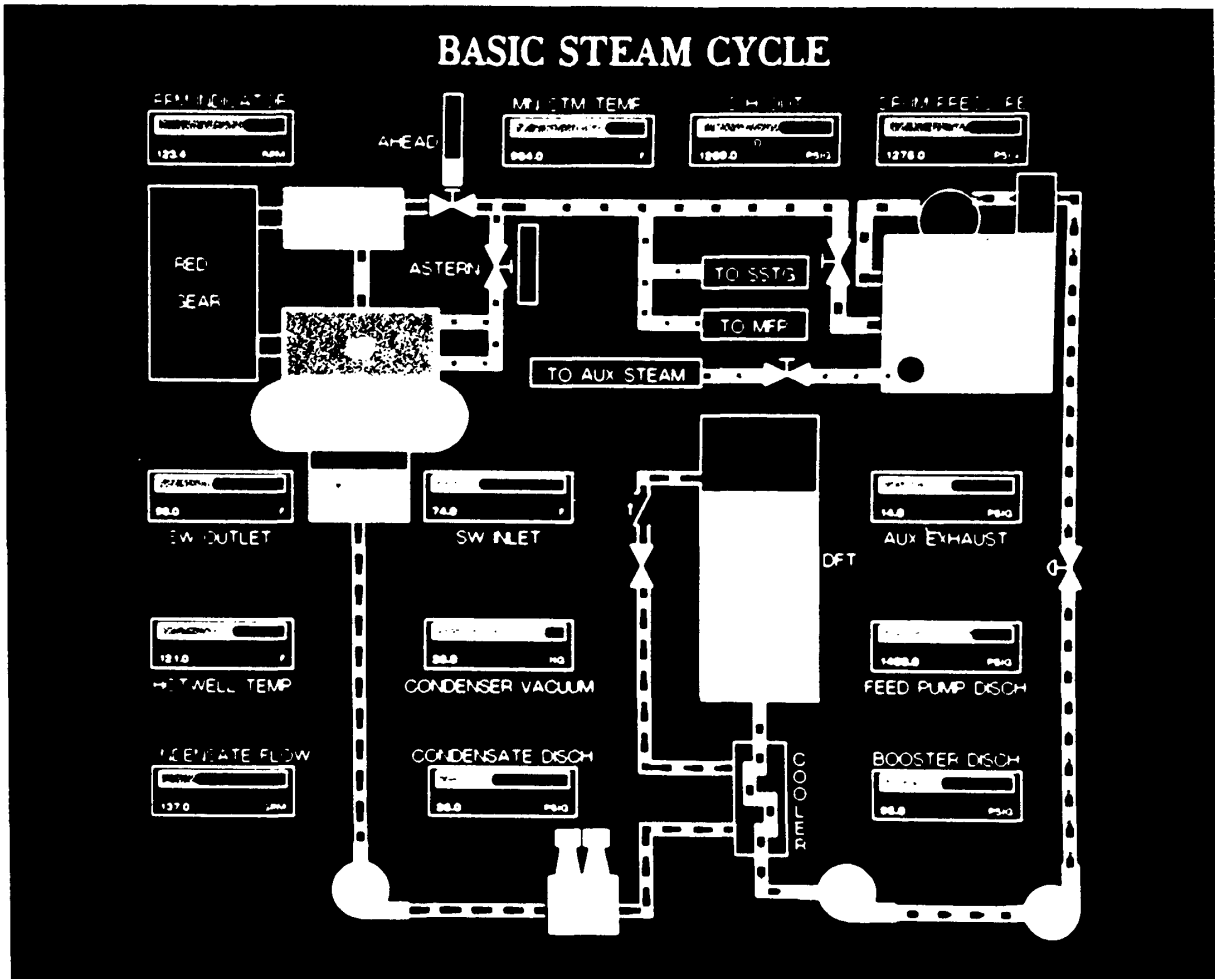


図2 STEAMERにおけるエンジン系のサイクル
(出典: The AI Magazine, Fig. 2 of page 17, 1984)

要であり、エンジン系のゲージを読み取るのに多くの経験を必要とするという対象領域である。

こうした内容を扱うために、もっとも重要であったのが、動的なシミュレーションの結果を表示するグラフィックインタフェース部である。蒸気の循環サイクルの様子を示したグラフィックスを示したのが図2である。実際のデモンストレーションでは、カラーグラフィックスを用いて、さらに、蒸気が循環するに伴って各々のパイプラインに蒸気の流れがあり、ゲージのレベルや数値が動的に変化するものであった。そして、蒸気プラント系をさまざまな階層から操作して、各要素をマルチウインドウシステムからみることができる。つまり、グラフィックインタフェースを通して、次のような3点をSTEAMERは特徴にしている。すなわち、蒸気エンジン系の動的シミュレーションの様子を知ることができる点、また、カーグラフィックスを使って、特定の要素が動作しているかを見れる点(たとえば、グリーンで動作、レッドで停止)、さらに、パイプを流れる蒸気のスピードの違いをみて要素の因果性

をチェックできる点である。

そして、実際のプラントでは考えられないような操作も、STEAMERではシミュレーションとして可能である。また、そうした操作によってプラントがどのような状態になるかを生徒に予測(envision)させることで、訓練の習熟度をSTEAMERはみることができた。STEAMERプロジェクトの初期では定性的推論で知られるForbus, K.が参加しており、また、システムの記述言語もSteeleとSussmanの制約指向言語CONLANが使われていたようである。したがって、グラフィックインタフェースだけでなく、内部の技術としても非常に進んだ内容を持っていたのがSTEAMERである。

Dipmeter Advisorについては、地下資源のデータベースを効果的に表示でき、さらに、現場のエンジニアが使えるかどうかというきびしい評価から開発されたインタフェースを持っている。このシステムもONCOCINと同じように、現場のエンジニアが日常的に使っているDipmeterという測定装置から得ら

れるデータシートのイメージをグラフィクスに取り入れている。また、推論部も重要であるが、パターン抽出部（地層のデータの分類）の処理の結果を、インタフェースに取り入れている点も大きな特徴となっている（この点については、本特集の森の論文が詳しい）。

3. ユーザインタフェース設計の支援環境

前節で取り上げた ONCOCIN, STEAMER および Dipmeter Advisor が持っているユーザインタフェースは、開発の履歴から考えてもユーザの立場を十分に考察した結果とみることができる。すなわち、ONCOCIN では、Stanford 大学での MYCIN の経験に基づいて、実用的な医療支援のユーザインタフェースとしてスプレッド方式を採用している。また、STEAMER を開発した BBN 社は、知的教育システムの研究ですでにいくつかの先行的な研究、たとえば SCHOLAR, SOPHIE で知られており、認知科学的なアプローチで知的教育システムの設計に実績を持っている。さらに、Dipmeter Advisor を開発した Schlumberger-Doll 研究所の Smith, R. らも、フレーム表現言語の

UNITS (その後 STROBE の開発) のインプリメンテーションとして知られている。特に、Schlumberger-Doll では、Dipmeter Advisor のグラフィクス部分の設計者である Schoen, E. という有能なプログラマがいた。このように、すぐれたユーザインタフェースの設計のためには、数多くの経験（再設計のサイクル）、コードの蓄積およびシステムのインプリメンタが必要である。

さらに、ユーザインタフェースの設計をプログラム環境やグラフィクスの支援環境でみると、上記のシステムには固有の支援システムが存在している。ONCOCIN では、スプレッドシート方式を採用しているが、治療方針を決定するためのグラフィック言語である OPAL が使われている。OPAL は、ウィンドウシステム上に表れるメニューボックスを選択して、治療開始→治療方法1→治療方法2……というグラフを作成する言語である。グラフが完成すると、このグラフはインタプリタを用いて ONCOCIN の推論用知識や治療計画に変換されていく。また、個々の患者への治療プロトコルに応じて会話的に治療計画のネットワークを作成し、そのグラフにどのようなルールを使って診断した

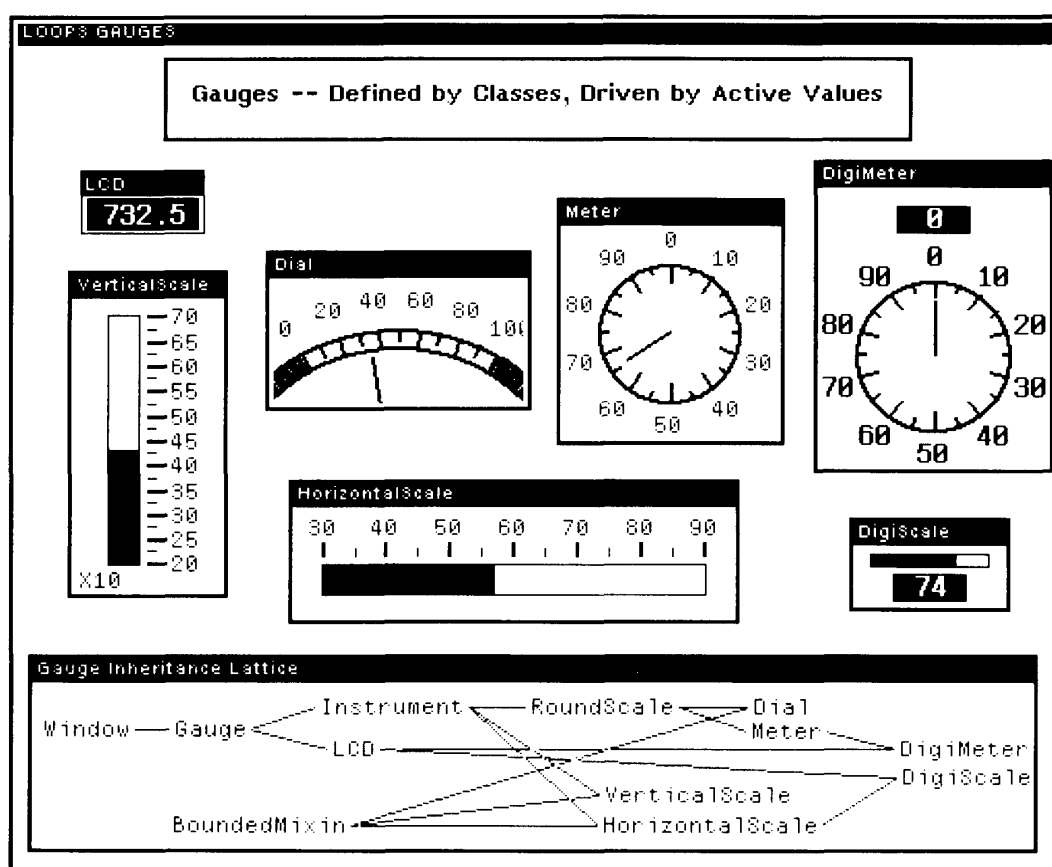


図3 Loopsにおけるゲージ類のオブジェクト
(出典：The AI Magazine, Fig. 14 of page 57, 1986)

かを付加することもできる。こうした作成プロセスは臨床医自身が、OPAL を用いて LISP マシン上で行われていく。すなわち、OPAL は、医師自身の知識（治療方針）を操作する言語のプログラム環境と言える。

ONCOCIN に対して STEAMER のインタフェース設計には、物理的装置、すなわち、機械プラントの計器類のイメージアイコンを Zetalisp の Flavors⁽⁷⁾（オブジェクト指向言語）を用いて、多数、準備している。そして、グラフィックエディタ上のメニューによりプラント全体の計器類、ポンプ、バルブ、電気系の測定器などをマウスの操作で組み立てることが可能である。ダイヤルのゲージの最大値、最小値に対するデフォルトやスケールの校正値なども、オブジェクト指向の多重継承機構によって容易に作成することができる。オブジェクト指向のこうした利点については、Stefik と Bobrow⁽⁸⁾ の用いた、図 3 におけるゲージ類のオブジェクト表現からも容易に理解できよう。

STEAMER のインタフェースのもうひとつの特徴は、会話的に検査可能なシミュレーション (Interactive Inspectable Simulations) である点である。つまり、エンジン系のシミュレーションの状態を学習者が自発的にチェックでき、内部状態を視覚的にグラフィクスにより見ることができる指導権 (Mixed Initiative)⁽⁹⁾ をユーザに許している点である。こうした点を含めたインタフェースを STEAMER のグラフィクス支援環境は持っていると言える。

もちろん、上記のグラフィクスの支援環境はビットマップのラスターキャン方式の表示装置で開発されている。そして、人工知能のプログラミング環境との結合は、Xerox の Inter-Lisp 系の場合には十分に開発されており、また、ディスプレイ用のパッケージが豊富である。このパッケージを用いて支援環境が開発されている場合が多い。たとえば、Schlumberger-Doll 研究所の知識表現言語の STROBE に対する Impulse⁽¹⁰⁾ がその代表的システムである。Dipmeter Advisor のグラフィクスは、Impulse を用いて測定データのイメージ、地下資源のデータベースが表示できるようになっている。

そのほか、最近の市販の人工知能用のツールにはグラフィクスの支援環境が準備されている。たとえば、KEE⁽¹¹⁾ における Active Image, ART における ART studio⁽¹²⁾ などもユーザインタフェース機能を拡張するための環境であり、ユーザの目的に応じて短期間でインタフェースをプログラムすることができるようになっている。

4. ユーザインタフェースの持つ機能

いくつかの優れたユーザインタフェースに接してみると、いくつかの点において共通した機能があることがわかる。

その第 1 は、操作性が優れていることである。たとえば、ONCOCIN ではスプレッドシートにマウスで直接、ユーザが実行したいことができる点である。また、STEAMER および Dipmeter Advisor では、ユーザの探索に応じてデータを多面的に見ることができる。STEAMER では、エンジン系の動的状態を、また、Dipmeter Advisor では地下数千フィートの地層の様子を、直接的に操作して調べることができる。つまり、インタフェースを媒介にしてユーザの操作性が増幅 (拡大) されている。

第 2 は、インタフェースを通じてユーザに情報の多様な表示形態が可能である点である。そのために、高解像度のビットマップディスプレイと、マルチウインドウシステムを採用しているインタフェースが多い。ここで情報の多様な表示形態というのは知識ベースを多角的な視点、たとえば、診断における推論のプロセス、仮説の分類を表すツリー構造、あいまいさの変化などによって、複雑なシステムの挙動をユーザに理解しやすくすることである。このことをもっとも積極的に使ったのが、Clancy, W. の GUIDON-WATCH⁽¹³⁾ である。GUIDON は、MYCIN を教育システム用にしたシステムで、図 4 のような出力画面となる。この出力では、ユーザは質問 5 でストップして、その段階で考えられる仮説の状態、また、仮説の分類木やどのような知識 (ルール) が、その仮説を支持しているかをウインドウに表示している。こうした出力は、すべて知識ベース内で働いている推論のプロセスを、どのようにすればユーザにとって理解しやすいかという試みから設計されたインタフェースと言える。

第 3 は、操作性や表示形態に奥行きがあって、ユーザに“退屈さ”を感じさせない機能を有している点である。“退屈さ”というのは、やや抽象的な言い方であるが、少なくとも多様な機能が次々と表れて、ユーザに常に新鮮さを与えることができるという意味である。そのために種々の工夫、たとえば、豊富なアイコンイメージの利用、グラフィクスの導入、および動的変化の表示 (グラフィクスによるアイコンの変化) をうまく取り入れたインタフェース機能となっている。STEAMER における蒸気の流れ、ONCOCIN における身体部位のグラフィクスなどは、こうした試みであ

The screenshot shows the GUIDON WATCH interface. At the top, there are menu items: KB Windows, Consult, Help, and a Prompt Window. The main area contains a list of questions and a differential diagnosis. Below this, there are three tables: 'Evidence for INFECTIOUS-PROCESS', 'RULE423', and 'Differential'.

Evidence for INFECTIOUS-PROCESS

FINDINGS	RULE(S)	MAXCF	MINCF
FEBRILE	RULE423	700	
WBC	RULE350	500	
PWS	RULE350	500	
BANDS	RULE350	500	

RULE423

If: The patient has a fever
Then: If you are considering any disorder, there is suggestive evidence (.7) that the underlying etiology of the patient's illness is infectious-process

Differential

HYPOTHESIS	CF	CUNCF
CHRONIC-MENINGITIS	200	
VIRAL-MENINGITIS	200	

Hypotheses With Evidence

HYPOTHESIS	CF	CUNCF
MENINGITIS	500	500
INFECTIOUS-PROCESS	---	500
CHRONIC-MENINGITIS	200	
VIRAL-MENINGITIS	200	
ACUTE-MENINGITIS	---	200

図 4 GUIDON WATCH における推論プロセスの表示例 (出典: IEEE CGUA, Fig. 3 of page 54, 1985)

り、そうした機能が暗示的 (implicit) な形で準備されている。つまり、インタフェースの多くの機能性が埋め込まれた形になっており、ユーザの要求に応じて自然な形で、こうした機能の一部がユーザに“退屈さ”を感じさせることなく現れるということである。通常、ユーザは表示された情報に対して感度が高く、また要求もきびしい。したがって、使われたアイコンのイメージや情報の表現が平板の場合には、インタフェースの設計者の意図とは別の理解になってしまうことがある。たとえば、よく慣れたイメージアイコンに比べて、新しい同じ意味のアイコンの美的センスが悪い場合には、それだけでインタフェース全体の評価は低くなってしまふという傾向がある。そうした意味で、インタフェースの機能や表示部は、内部のシステムの挙動との対応において十分な試行に基づいて設計する必要がある。そして、すぐれたインタフェースは、前にも触れたように数多くの試行の結果をもとにしていることは確かである。

インタフェースの機能を、以上のような3点にまとめる考え方もあるが、こうした3点を全体的にくるとすると、ユーザとインタフェースの指導権の混在である。つまり、インタフェースによって情報処理の制御の流れをユーザ側とコンピュータ側に分散させ、必

要に応じてどちらかに指導権をスイッチするという考え方である。その結果、たとえば、推論における探索はコンピュータが行い、また、必要なデータの収集はユーザが行うというような役割の分担ができる。さらに、コンピュータが探索を失敗したときに、ユーザ側が別の視点からの情報をコンピュータ側に与えて、会話的に問題解決をするという考え方も、指導権の混在という概念を効果的に使ったアプローチと言えよう。そして、インタフェースの多様な操作機能や表示形態が、こうした指導権のスイッチのタイミングをうまく調整しているわけである。

また、“退屈さ”をさけるためにインタフェース側が考えなければならない点は、ユーザの持つ“探索的駆動” (exploratory drive) をどの程度まで満たすかということであろう。ユーザとしては、常に“次に何が起こるか (What is going on happen? ; WGOH と略記)”をコンピュータ側に求めていくのが普通である。とすると、インタフェースの役割は、すぐにもそのWGOHを満たすような情報をユーザに示すことによって、ユーザの“探索的駆動”を充足していくことができる。このように考えていくとインタフェースの役割は、ユーザの“探索的駆動”のためにコンピュータとの指導権をうまくスイッチさせ、そのための操作

性や表示形態をどのように設計していくかということになっていくと思われる。ただし、こうした考え方は、1つのヒューリスティックな原理であり、説明的にしか使えない。すなわち、現段階では、以上の原理に基づいて設計されている限りそれほどひどいインタフェースとなっていないという意味のヒューリスティックである。

5. ユーザインタフェースの設計事例

ここでは、筆者らが数年来開発してきた医療診断システムのインタフェースについて取り上げてみたい。設計の基本的考え方は、前節で述べたものに基づいている。計算機環境としては、第5世代プロジェクトの前期成果である PSI (逐次型推論) マシンを用いた⁽¹⁴⁾。PSI マシンは推論型言語の Prolog とほぼ同等の機能の機械語 (KLO と呼ばれる) を用いて、オブジェクト指向概念を導入した ESP で開発されたオペレーティングシステム SIMPOS を持つ。SUN や APOLLO といったワークステーションが言語 C という低レベル言語で作成された UNIX オペレーティングシステムを持つものと、PSI マシンは基本的に異なっている。すなわち、PSI マシンは高レベルの言語 ESP⁽¹⁵⁾

を用いて、オペレーティングシステムがユーザの応用までを書くことができるプログラム環境を持っている。

したがって、ESP (PSI マシンのユーザ言語) を使えば、部分的にオペレーティングシステムの組込み機能、特にデータ管理をユーザは利用することができる。つまり、ユーザはウィンドウシステムやデータ管理といったオペレーティングシステムで統合されている機能を、ユーザインタフェースの設計にまで利用することができる。また、ESP 自身はオブジェクト指向概念を導入しているので知識表現とみることができる。したがって、知識ベースが準備されていれば、それを ESP のオブジェクト表現に変換し、すぐに診断システムを開発することが可能である。筆者らの医療診断システムも、Prolog によるオブジェクト指向概念による知識表現言語の姿 (LOOKS)⁽¹⁶⁾ を用いて作成されていた。この知識ベースを ESP 用に調整して PSI マシンに移植された。

図5および図6は、PSI マシン上で作動する医療診断システム (姿/ PSI) の出力画面である。この出力は PSI マシンのディスプレイ部のハードコピーであるので、実際の出力にほぼ近いイメージと言えよう。このインタフェースは、ESP を用いてほぼ3か月で

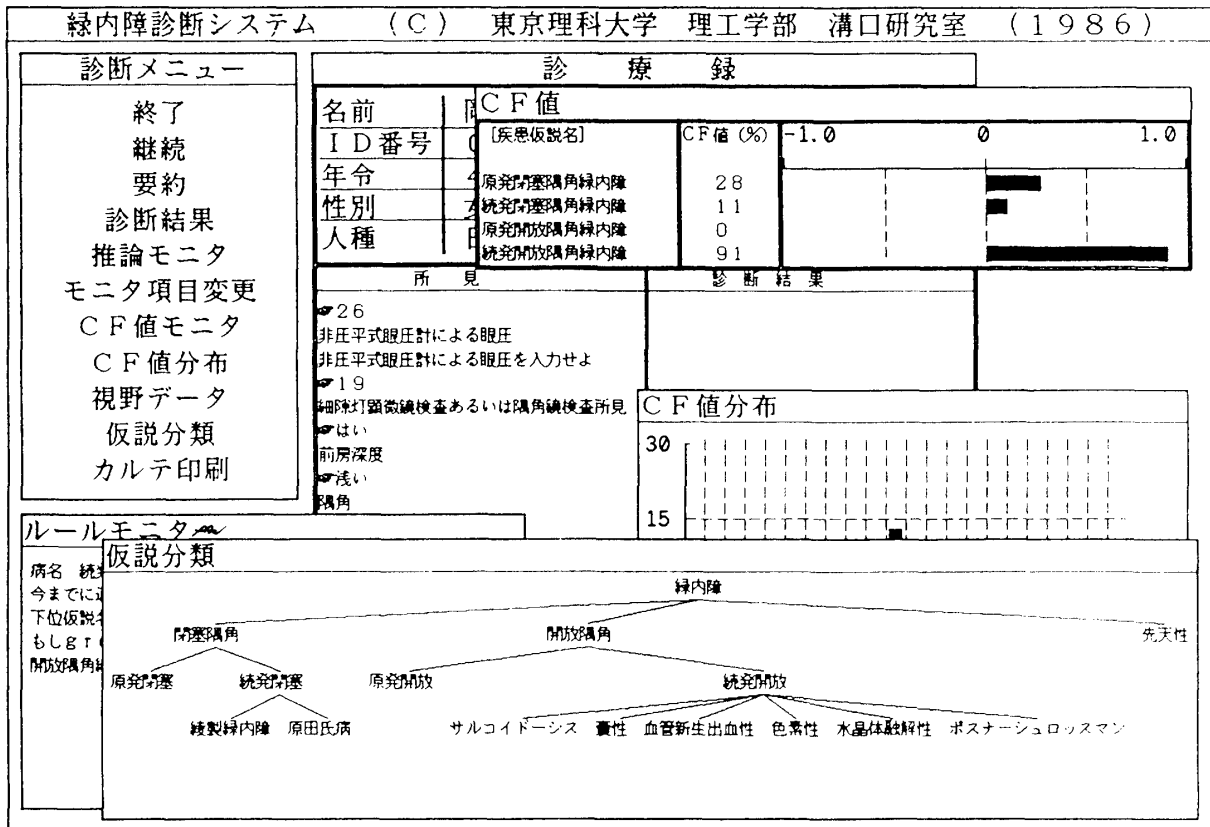


図5 姿/PSIにおける推論プロセスの出力状況 (CFの分布および仮説分類木が表示され、診断メニューが現れている)

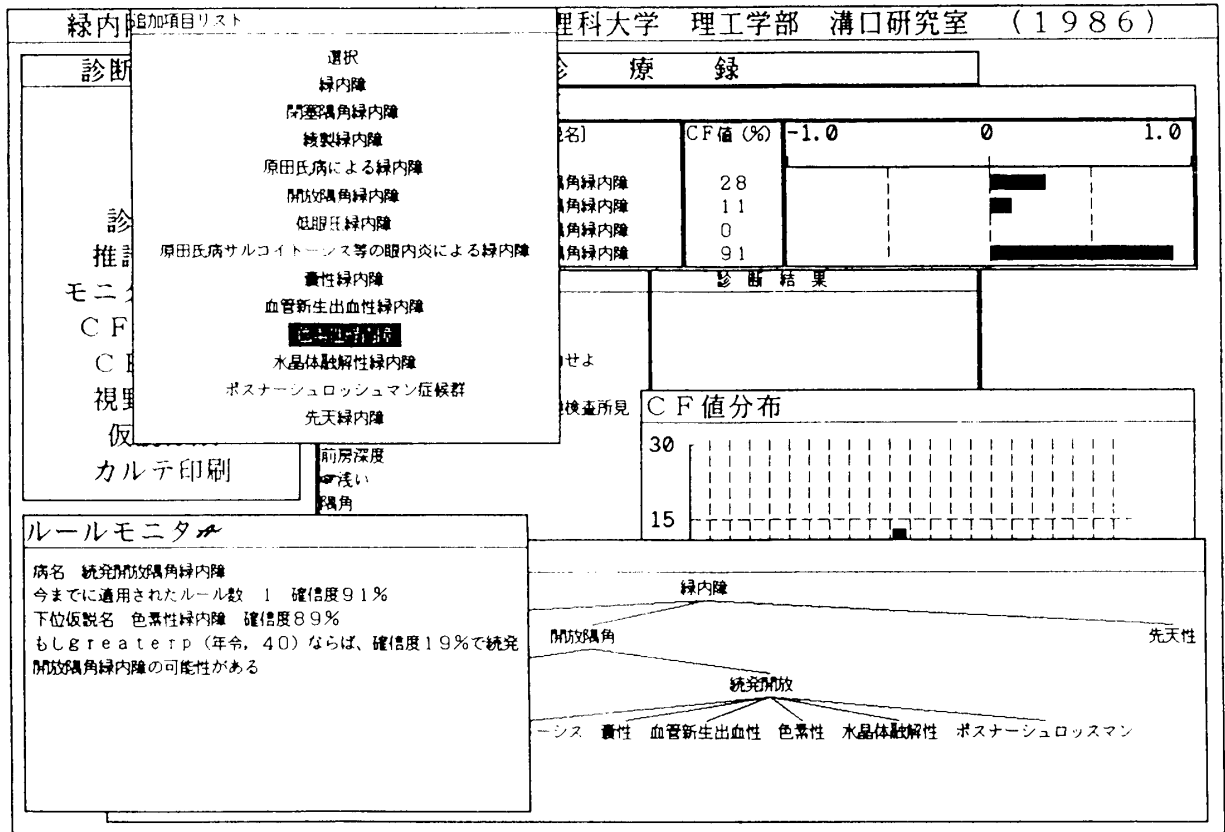


図6 姿/PSIにおけるユーザの仮説の検証項目の追加の様子を示している

完成し、昭和61年の春のビジネスショーでもデモンストレーションされた。PSI マシンの設置と使用開始が昭和60年12月上旬のことを考えると、インタフェースや推論の設計にESPが強力な言語であることがわかる。その後、推論部にDempster-Shaferの方法という理論的にも体系の整った枠組みが取り入れられている。この部分の開発に約6か月を要している。それは、Dempster-Shafer⁽¹⁷⁾の方法を効率的に計算する問題を解く必要があったためである。

姿/PSIのESPコードは、知識ベース部を除くと約1000行ほどで質問起動部250行、ルール適用部150行、Dempster-Shaferの方法による推論部200行、インタフェース部400行である。コード全体から考えると、インタフェース部が40%で、Dipmeter Advisorの42%にほぼ近いことがわかる。コードがコンパクトになっているのは、ESPとSIMPOS機能を部分的に利用しているためと思われる。

姿/PSIにおけるユーザインタフェースでもっとも重視した点は、ユーザのWGOH活動である。特に、ユーザである医師にとって必要であるのは、今自分が入力したデータが診断システムにどのような意義を与えているかを知ることである。そのために、データから求まる確信度の分布やデータを解釈するルールをマ

ルチウインドウに表示している点である。さらに、姿/PSIの診断結果とは別に医師が同時に持つ仮説について、姿/PSIがユーザに問い合わせ、必要ならばその仮説に関連するデータ(入力したものの中から)を再構成し、仮説の可能性をチェックしていく点も考慮されている。実際には、図5の出力の診断メニューのうちモニター項目を選択する。その結果、いわばシステムからユーザへの制御の視点を変えることに対応している。姿/PSIでは、同時に複数の仮説をみることもできるが、思いもよらない仮説については最初の段階で診断の候補から落としていくという方式を採用している。この弱点を補うために、必要ならばユーザが持つ(疑問に思っている)仮説を選択するオプションを姿/PSIは許している。図6では仮説の追加項目リストを加えて、その可能性を調べている様子を示している。この2点が、姿/PSIにおけるインタフェースの設計において留意した部分である。そのほかの考え方や基本的な設計原理は、前節で触れたとおりである。

6. おわりに

本稿では、主として人工知能の応用システムにおけるユーザインタフェースの設計について、いくつかの

事例や支援環境，さらに筆者らの経験を中心にして議論を進めてきた。現段階では，ユーザインタフェースの設計のためのガイド的な原理が，すぐれたインタフェースから少し見えてきたという段階であろう。あるいは，応用システムの広がりと共に，ようやくユーザインタフェースの重要性が認識されてくる時代に来たと言えよう。もうひとつのポイントは，筆者らが経験した PSI マシンのような新しい計算機環境の出現が，ユーザインタフェースの設計を容易にしたことである。特に PSI マシンは，日本のユーザインタフェースにとって絶対と思える日本語のサポートが十分に

されている点も，大きな長所となっていると言えよう。しかし，ユーザインタフェースの設計や試行については，まだ，それほどの蓄積や実験が存在していないのがわが国の実情であろう。やはり，ユーザインタフェースの研究は，Xerox PARC の Kay, A. ⁽¹⁸⁾ の Learning Research グループが試行していたような実験が必要であろう。

最後に，良いユーザインタフェースの例を身近でさがしている人は，ぜひマッキントッシュの娯楽用のソフトウェアである Studio Session をみていただきたい。非常に参考になると思われる。

◇ 参 考 文 献 ◇

- (1) 淵監修，古川，溝口編：インターフェイスの科学，共立出版(1987)。
- (2) 溝口文雄：推論とインターフェイス——PSI マシン上の“姿”；緑内障診断の経験から——，数理科学，No. 279，pp. 55-61 (1986)。
- (3) Shortliffe, E. H. : Medical Expert Systems Research at Stanford University, Paper presented at the 20th IBM Computer Science Symposium (1986)。
- (4) Hollan, J. D., Hutchins, E. L. and Weitzman, L. : STEAMER: An interactive inspectable simulation-based training system, The AI Magazine, Vol. 5, No. 2, pp. 15-27 (1984)。
- (5) Smith, R. G. : On the development of commercial expert systems, The AI Magazine, Vol. 5, No. 2, pp. 61-73 (1984)。
- (6) Stefik, M and Bobrow, D. G., *et al.* : Knowledge Programming in Loops: Report on an experimental course, The AI Magazine, Vol. 4, No. 3, pp. 3-13 (1983)。
- (7) Weinreb, D. and Moon, D. : LISP Machine Manual, Cambridge, MIT Artificial Intelligence Laboratory (1981)。
- (8) Stefik, M. and Bobrow, D. G. : Object-Oriented Programming: Themes and Variations, The AI Magazine, Vol. 7, No. 4, pp. 40-62 (1986)。
- (9) Carbonell, J. R. : Mixed-Initiative Man-Computer Instructional Dialogues, Tech. Rep. 1971, Bolt Beranek and Newman.
- (10) Schoen, E. and Smith, R. G. : Impulse: A display-oriented editor for Strobe, Proc. of AAAI-83, pp. 356-358 (1983)。
- (11) Fikes, R. and Kehler, T. : The role of frame-based representation in reasoning, Comm. of the ACM, Vol. 28, pp. 904-920 (1985)。
- (12) Clayton, B. : ART Programming Primer, Inference Corp. (1985)。
- (13) Richer, M. H. and Clancy, W. J. : GUIDON-WATCH: A Graphic interface for viewing a knowledge-based system, IEEE CG & A, pp. 51-64 (1985)。
- (14) 溝口文雄：PSI におけるエキスパートシステム(1)——アプローチの特徴とその経験について，情報処理学会第34回全国大会論文集(1987)。
- (15) Chikayama, T. : ESP Reference Manual, ICOT Tech. Rep. TR-044 (1984)。
- (16) 溝口，本田，片山：オブジェクト指向概念を導入した知識表現言語；姿と鏡の設計とその応用，Proc. 1st Int. Conf. on Logic Programming (JAPAN) (1984)。
- (17) 溝口，井下：PSI におけるエキスパートシステム(2)——医療診断への適用と基本的環境の開発，情報処理学会第34回全国大会論文集(1987)。
- (18) Kay, A. : Microelectronics and the personal computer, Scientific American, Vol. 273, No. 3, pp. 230-244 (1977)。

著 者 紹 介



溝口 文雄(正会員)

昭和41年東京理科大学工学部工業化学科卒業。昭和43年東京理科大学大学院修士課程修了。工学博士(東京大学)。現在，東京理科大学理工学部経営工学科教授。認知科学と知識工学，人工知能の広範囲な研究を進め，最近では，PROLOG および ESP によるエキスパートシステムの開発を行っている。対象は，VLSI 設計から医療診断まで広い領域の研究を行っている。AAAI 学会，情報処理学会，日本ソフトウェア科学会，認知科学会などの会員。