

# ユーザモデルによる インタフェース評価

北島宗雄

通産省 工業技術院  
生命工学工業技術研究所

茨城県つくば市東 1 ~ 1

電話 : 0298(54)6731  
E-mail : kitajima@nibh.go.jp

ヒューマン・コンピュータ・インタラクション (HCI) はコンピュータシステムとユーザの界面で起こる相互作用を研究対象としている。そのひとつのアプローチ方法にユーザモデルを利用したものがある。ユーザモデルでは、ユーザが機器と相互作用する過程がモデル化され、それを解析することにより、使いやすさなどインタフェース評価に係わる指標が導きだされる。本講演の Part I では、ユーザモデルのインタフェース設計における役割と、現在までに HCI 分野で提案されてきたユーザモデルを紹介し、それらのモデルによって解析することができる人間の認知的な活動の側面について説明する。続いて、Part II では、最近急速に普及してきているグラフィカルユーザインタフェース (Windows、Macintosh など) に焦点を当てて、それを利用するユーザの認知プロセスモデルの開発に関する最近の研究の成果について非常にホットな話題を提供する。

## Part I

# HCIにおけるユーザ・モデルとモデルによる認知的側面の解析<sup>1</sup>

### あらまし

Part I では、人間と機器の相互作用場面を解析するために考案されてきた方法のひとつとして、ユーザモデルによる方法を紹介し、それらのモデルによりユーザの認知的活動のどのような側面を解析できるかを説明する。認知的活動として、ここでは、長期記憶、短期記憶、知覚の3つのレベルを考える。また、モデルとしては、ユーザが機器と相互作用する過程をトップダウン的にモデル化するタスク解析的モデル、ボトムアップ的に捉える状況依存的なモデル、そして、ユーザの概念構造に注目したモデルを取り上げる。

各モデルは全ての認知的側面を解析できるわけではなく、それぞれ得意とする領域がある。本講演では各モデルでどのような認知的側面（長期記憶、短期記憶、知覚）を解析できるかについて説明し、また、インタフェースの設計段階に応じたモデルの使い分けの考え方を説明する。

### まえがき

コンピュータをはじめとする情報機器の進歩により機器の性能はますます向上し、実行できる機能のバリエーションも増大の一途をたどっている。一方、それらの機器を使う側も多様化する傾向にあり、ユーザ層も機器の細部については全く素人で関心のないユーザからマニア的なユーザまで多岐にわたっている。

このように、機器、ユーザ双方の次元が多様化する中であって、機器の性能評価を、従来行われていたような機能的な側面、何ができるか、からだけではなく、使いやすさ、操作のしやすさ、わかりやすさなど、どのようにできるか、といったユーザの認知的な側面からも行うことの必要性が認識されてきた (Draper & Norman, 1986; Norman, 1988)。本講演の主題である「ユーザモデルによるインタフェース評価」もそのためのひとつのアプローチ法である。

さて、機器の性能を使い勝手などの認知的側面から評価するためには、まず、ユーザが機器を使用して作業を遂行している過程、ユーザ・機器相互作用過程、に含ま

れるユーザの認知的活動を解析できるような認知モデルを構成することが必要である。そして、そのモデルに基づいて長期記憶、短期記憶、知覚などの負荷量を算出し総合的な認知的評価（使い勝手、覚えやすさ、エラー頻度など）と結び付けることが必要である。

認知的な評価を行うためのアプローチには実験的手法と理論的なモデル解析の手法の2つがある (Olson, 1987)。実験的手法においては、インタラクションに含まれる認知的活動を客観的に計測しデータを収集する。そのデータは、理論によって検査されるべきデータであることもあるし、また、理論が説明すべき新しい行動データである場合もある。これにより、新しいシステムや、現存システムの効率を具体的に評価することができる。現在、インタフェース評価のために広く行われているいわゆる Usability テストはこの範疇にはいる。

一方、解析的手法では、相互作用をしているユーザ、機器、双方の処理過程を適切にモデル化することにより、想定される相互作用のさまざまな側面についての予測、評価を行う。相互作用の局面においては、ユーザは機器の呈示する情報に対して何らかの認知的な処理を行い機器に働きかける。一方、機器はユーザの応答に応じて内部処理を行い、その結果をユーザに視覚的、聴覚的に呈示する。解析的手法においてはこれらをモデル化するので、特別にデータを集めることなくデザインを評価できる。このことは、多様化に対処するには非常に重要なポイントであり、実際、デザインのサイクルが2回以上になる場合には解析的手法の方がコスト的に有利になると言われている (Butler, Bennett, Polson & Karat, 1989)。しかしながら、現状ではまだモデルに基づいた Usability 評価が現場で実際に行われるまでには至っていない。

より使い勝手の良い機器を設計するには、機器の最終仕様を設計の初期段階から検討できることが必須であり、その点において、解析的手法の必要性はますます大きくなっていくものと思われる。ここでは、現在提案されているいくつかの解析的手法を紹介し、認知活動のどのような側面の評価・予測に力があるかを説明し、さらに、インタフェース設計過程のなかに組み込むための一つの考え方を説明する。

### ユーザの認知特性

まず、本講演で取り上げる認知的側面について説明する。相互作用過程に関連するユーザの認知的側面として、知覚、短期記憶、長期記憶がある。それらの定性的な特質をまとめると次のようになる (Card, Moran & Newell,

<sup>1</sup>本原稿のPart I は、「機器使用過程表現の認知的解析」テレビジョン学会技術報告、Vol. 14, No.15, pp.7~14, 1990、をもとに若干の修正を加えたものである。

1983)。なお、以下の説明では、機器とのインタフェース場面に係わるユーザの認知特性で、その得意とするところ(+)と苦手とするところ(-)をまとめている。よいインタフェースは、+面をうまく利用し、-面が表面化しないように設計されたインタフェースである。

### 知覚

- + : 入力を並列処理することにより、視覚情報を高速に処理できる。また、情報源間の調整をすることができる(注意)。
- + : 曖昧情報を含むボタンを知覚できる。また、エラーに強い。
- : 詳細な情報に含まれる小さな差を知覚することが難しい。

### 短期記憶

- : 最近経験したことを必要となるまで一時的に保持したり、長期記憶から引き出された情報を保持する(例えば「実行中のプランのどこに自分が位置しているかを覚えていること」など)。短期記憶容量には限界があり、時間の経過、後続事象の干渉により、項目は容易に失われる。

### 長期記憶

- + : 似ているが全く同じではない過去の経験を利用して、概念、規則について推論をすることができる。経験から、規則の帰納推論、時間・空間における規則性の抽象化を行なえる。
- + : 共通パターンを効率よく貯蔵できる。頻繁に使われる行動系列はチャンクとして貯蔵される。
- + : 関連する情報を素早く検索できる。
- : 長期記憶内の情報はエラーに対して強いが故に、必ずしも常に正しいとは限らない。類似した項目を混同するし、また、細かな違いを識別し、学習することは難しい。
- : 長期記憶に情報を高速に格納することが難しい。
- : 繰り返し使用に対して忍耐強くない。

### 相互作用過程を表現するモデル

ユーザが機器を操作する過程にはさまざまな心理的活動を伴うが、その過程は、

- ゴール形成 (Goal Formation)
- 意図形成 (Intention Formation)
- 行動列挙 (Action Specification)
- 行動実行 (Action Execution)
- システム状態知覚 (System State Perception)
- システム状態解釈 (System State Interpretation)
- 結果の評価 (Outcome Evaluation)

の7段階に分けて捉えられる (Norman, 1986)。

機器操作過程は、これらを系列的につなぎ合わせたものである(図 A1)。では、それらの間の遷移・制御はどのようになされているのであろうか。その考え方として2通りのものがある。

その1つは、ユーザは、階層的に表現されたゴール構造の中を個々のサブ・ゴールを達成できる方法を選択しながらトップ・ゴールを目指して系統的に移動する(Newell & Simon, 1972)とするものであり、これは、ユーザの行動をプランに基づいた問題解決の行動として捉える見方である。このアプローチはさらに

- 最終的に生成されるユーザの行動の具体的な記述を目指すもの
- 大局的なゴール構造に着目するもの

に分けられる。それぞれのアプローチに含まれるユーザモデルについては「タスク解析的アプローチ」「ユーザとシステムの概念比較」の章で説明する。

もう1つの見方は、ユーザの行動を、先のアプローチのように「(予め計画された)プランに基づいている」と考えるのではなく、相互作用に関連する事象の極めて状況依存的な局所的解釈に基づいている(Suchman, 1987)と考えるものである。これは、ユーザの行動をよく観察してみると全く(ゴールを合理的に達成しようとするという意味で)論理的ではなく、場当たりに見えるという結果に基づいている。ユーザの行動を理解するには、実際に行動が行われた状況をぬきにしては考えることができないという立場である。最近では、Situating Cognition (Lave, 1988) という分野でこのような考え方に基づいた研究が進められている。

タスク解析的アプローチ

形式文法  
(Reisner, 1981)

タスク指向の最も代表的なアプローチは、機器が受け入れることができる入力を文法によって規定する方法である。ここで用いられる形式文法 (formal grammar) は、コマンド入力を行うのに必要な要素や、それらの要素を具体的に構成する際にとられる手続きを「書き換え規則」によって表現する。例えば

エディタを用いてテキストのあるブロックをコマンド Dn によって消去する

という行動は、次のように記述される。

```
use Dn      identify first line + enter Dn command
           + press ENTER
identify first line  get first line on screen + move
           cursor to first line
get first line on screen  use "Locate" strategy |
           use scroll strategy
"Locate" strategy  move cursor to command input
           field + type "locate" command + press ENTER
move cursor to command input field  use cursor
           keys | press PFCURSOR | null
type "locate" command  type "locate" keyword +
           type line number
type "locate" keyword  L + O + C | L | L + O + C -
           + A + T + E
type line number      type number
```

このように、形式文法によると表現が非常にコンパクトになり、また、要素間の関係の階層性を容易に見い出すことができる。

コマンド言語文法  
(Command Language Grammar ~ CLG ~)  
(Moran, 1981)

CLG (Command Language Grammar) 表現は、先の形式文法による表現の延長に位置しそれに若干の心理的な構造を取り入れた表現方法である。CLG は、ユーザがシステムに対して抱く心理的な概念モデルが、概念、コミュニケーション、物理の3つのコンポーネントから構成さ

れると仮定している。これらのコンポーネントはさらに6つのレベルに分解され、概念コンポーネントはタスクレベル(そのシステムに関連したタスク領域)と意味レベル(システムによって表現される概念)に、コミュニケーションコンポーネントは統語レベル(コマンド・引数の構造)と相互作用レベル(会話構造)に、また、物理コンポーネントは空間配置レベル(入出力装置の配置、グラフィックスレイアウト)とデバイスレベル(他の物理的特徴)、に分けられる。

これらの各々のレベルは、そのレベルで作用を受ける対象であるオブジェクト、まとまったタスクを遂行するための一連のオペレーションや行為を記述するメソッド、メソッドを表現するための基本的要素であるオペレーションという要素によって表現され、要素のいくつかは他のレベルの要素と関係付けられている。各レベルを形成する要素を表に示す。

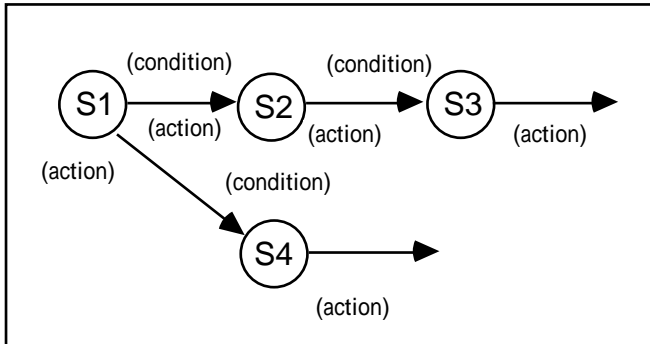
レベル	要素
タスク	概念、ゴール、手続き、メソッド (手続きとゴールを結ぶ)
意味	システム概念、システムオペレーション (SO)、ユーザオペレーション (UO)、手続き (SO + UO)、メソッド
統語	コマンド、引数、引数記述式、コマンドコンテキスト、引数コンテキスト、状態変数、表示領域、表示アイコン、手続き、メソッド
相互作用	構成要素、入力デバイス、基本アクション (ユーザ & システム)、会話制御規則、時間順序規則、ユーザアクション規則、システムアクション規則、手続き、メソッド

ここでは述べないが、形式文法による表現方法には、他に、ユーザのタスクと、実際に機器に対してなされるユーザのアクションの間のマッピングを形式文法により表現するモデル、Task-Action Grammar (Green & Payne, 1986) がある。インタフェースの一貫性を分析するのに特に適している。

一般化遷移ネットワーク  
(Generalized Transition Networks ~ GTN ~ )  
(Kieras & Polson, 1983)

GTN (Generalized Transition Networks) は、機器の振る舞いを表現するために遷移ネットワークの考えを発展させて作り出されたものであり、機器の側から見たユーザのモデルという部類に入る。GTN は、通常のネットワー

クと同様に、システムの状態を示すノードと状態間の可能な遷移を表現するアークから構成される。アークは、条件、作用、そして、次にとれる状態から成る。GTN の



基本的な書き方は、

である。GTN はネストを許している点に特色がある。ネストはシステムの状態 ( 図中 S 記した部分 )、条件 ( condition )、作用 ( action ) のどの部分に現われてもよく、これにより、通常のネットワークでは表すことが困難であったシステムの階層構造を自然な形で表現することができる ( 図 A2 )。

GOMS モデル  
(Card, Moran & Newell, 1983)

GOMSモデルは、ユーザのタスク遂行過程を、

- (1) ゴール (Goal)、
- (2) オペレータ (Operator)、
- (3) メソッド (Method)、
- (4) 選択規則 (Selection rules)、

の4つの要素によってモデル化し、特に操作過程に再現性があるエキスパート・ユーザの行動のパフォーマンス ( 実行時間、オペレータ系列など ) を予測することを目的としている。オペレータは最低レベルの運動やメンタル活動を、メソッドはプリミティブ・オペレータが結合してできるチャンク化された活動を、また、選択規則はある状況におけるメソッドの選択を支配する規則を表している。対象がエキスパートなので、実際に収集された行動データを解析することによりかなりの精度でこれらの要素を確定できる。

ユーザの行動は概略次のようにモデル化される。タスクが与えられると、ユーザは最高レベルのゴールを構成する。次に、それはサブ・ゴールに分解される。それらは、メソッドの選択を起動し、オペレータを活性化する。サブ・ゴールは個々のオペレータ、または、一連のオペレータ ( メソッド ) によって達成されるが、それにより環境の変化が引き起こされ、それがトリガーとなって次のサイクルが起動される。

以下は原稿の訂正指示を MANUSCRIPT から読み取って修正作業をする過程の GOMS モデルによる記述例である。表中、get-next-page, get-next-task, verify-edit はオペレータ、Use-S-Command などはメソッドである。

---

**GOAL:EDIT-MANUSCRIPT**

- **GOAL:EDIT-UNIT-TASK**  
repeat until no more unit tasks
- • **GOAL:ACQUIRE-UNIT-TASK**  
• • • get-next-page  
if at end of manuscript page
- • • get-next-task
- • **GOAL:EXECUTE-UNIT-TASK**  
• • • **GOAL:LOCATE-LINE**  
• • • • [ select:  
Use-QS-Method  
Use-LF-Method ]
- • • **GOAL:MODIFY-TEXT**  
• • • • [ select:  
Use-S-Command  
Use-M-Command ]
- • • • verify-edit

---

プロダクション・システム  
(Cognitive Complexity Theory ~CCT ~ )  
(Kieras & Polson, 1985)

ユーザの学習時間、実行時間とエラーを予測する為のモデルとして、CCT (Cognitive Complexity Theory) が、Kieras & Polson (1985) によって提案されている。これは、いわば、GOMS モデルのプロダクションシステム版であり、GOMS モデルでは対象とされていなかった初心者ユーザの学習を扱えるようにしたモデルである。ユーザは個々のプロダクションを学習し機器操作ができるようになる。

ユーザのタスク遂行過程は IF-THEN 形式のプロダクションによるプログラムによってモデル化されており、そのプログラムを実行することによりユーザと全く同じ振る

まいを再現でき、また、タスクを遂行する過程で要求される認知オペレーションをシミュレートできる。このプロダクションには、ユーザの“how to use a system”に関する知識が記述されるが、その一般的な形は下のようになる（具体例は図 A3 参照）。

---

(label

**IF** (AND  
 (TEST-GOAL *description of goal*)  
 (TEST-NOTE *description of note*)  
 (TEST-SCREEN *description of information on screen*)  
 (TEST-MSS *description of editing task*)  
 (NOT (*condition*)))

**THEN** (  
 (ADD-GOAL *description of goal*)  
 (DELETE-GOAL *description of goal*)  
 (ADD-NOTE *description of note*)  
 (DELETE-NOTE *description of note*)  
 (DO-KEYSTROKE *depress single key or control*)  
 (DO-TYPE-IN *enter string stored in working memory*)  
 (LOOK-MSS *find string in manuscript*)))

---

### ユーザとシステム の概念比較

前述した方法では、システムやユーザの処理過程をモデル化の切り口としていたが、それよりも上位の概念、それを産み出している概念構造、に焦点を当てて、相互作用過程においてユーザが抱いている概念（メンタル表象）と、それに対応するシステムの表象の対応関係を解析することにより、システムの使い勝手、学習容易性についての評価・予測を行うことができる。ここでは、そういったアプローチのいくつかを紹介する。

#### タスク-デバイス・マッピング (Kieras & Polson, 1985)

システムに埋め込まれている概念構造をゴール構造グラフによって表現し、ユーザの視点から構成されるものと比較する。システムのゴール構造グラフは前述の GTN 表現から直接導きだされる。また、ユーザのゴール構造は、CCT におけるプロダクション・システムを動かして生成されるゴール部分だけを抜き出すことによって、階層的に表現される。このマッピングにより、どこでエラーが起こり易そうかということを簡単に知ることができる（

図 A4）。

#### ETIT (External-Task Internal-Task) 解析 (Moran, 1983)

ユーザが考える操作対象とそれに関連したアクションを、システムのものと比較する。一例として、ライン・エディタを考えてみる。システムにとっての操作対象は「行中の文字列」であり、それに操作可能なアクションは、inserting、cutting、pasting、breaking、merging、そして、replacing である。一方、ユーザの操作対象はもっと変化に富み、「文字」「単語」「文節」「段落」などであり、また、アクションには adding、removing、changing、transposing、moving がある。

タスクをコンピュータで実行するときには、ユーザは、まず、タスクを自分自身の言葉によって表現し、さらにそれを計算機が理解できる表現に変換する必要がある。Moran は、変換を行なうために知らなければならない規則を詳細に検討した。この変換を行なうのに必要な規則数により、初期学習の負荷を評価できる（図 A5）。

#### アナロジーによる学習 (Douglas & Moran, 1983)

新たに機器の操作を学ぶときにはアナロジーを利用すると学習が促進されることが多い。しかし、マッピングが完全になることは稀で、必ず例外があり、そのことが学習や、学習後のパフォーマンスに悪影響を及ぼすことがある。Douglas & Moran は、タイプライターから Emacs へのアナロジーによる学習を、各システムを表現する「ゴール・アクション」のペアを基に解析し、

- ゴールが同様なのに異なったアクションが要求される場合、
- アクションが同様なのに異なったゴールが満たされる場合、

にエラーが頻発することを明らかにした（図 A6）。

### 状況依存的なアプローチ

#### Situated Action (Suchman, 1987)

Suchman は、エキスパート・ヘルプ・システムを備えたかなり複雑なコピー装置をユーザが初めて使用するときに遭遇する様々な問題点を分析するための方法として、

初心者が2人一組で相談しながら操作する様子をビデオ・テープに収録しそれを以下に示すようなごく単純な枠組みで解析する方法を提案している。

機器側

機器が認知できないユーザの行動  
機器が認知できるユーザの行動

ユーザ側

ユーザが認知できる機器の状態  
ユーザには知りえない機器の設計理念

ここでは、ユーザの行動を、**機器が認知できる行動**(例：ある操作ボタンを押した)と、**機器には知ることができない行動**(例：問題解決のためになされる相談)に分け、また、機器の状態を、**ユーザが認知できる機器の状態**と、ユーザには知ることができない**設計理念**とに分けている。この表現方法によれば、ユーザと機器が相互作用をしているまさにその瞬間の様子を、それぞれが互いに観測できる側面によって非常にうまく表すことができる(図 A7)。

Suchman は、実際に、前述のトップ・ダウン的アプローチでは解析が及ばない点をこの枠組みによれば解析できることを示している。録画されたテープを解析することにより、相互作用を円滑に進めるうえで非常に重要な要素、ユーザとシステムがそれぞれ相手の状態をどの程度理解できるかということ、がコミュニケーションの展開に及ぼす影響を浮き彫りにしている。そこでは、インテリジェント・システムが暗黙のうちに仮定しているユーザ像と、実際のユーザの振るまいが異なっていることが、問題の根源として明らかにされている。システムは、ユーザが極めて堅固なプラン構造に基づいて行動をしている(記述的説明)という想定の上にインストラクションやヘルプの内容を決定する。ところが、ユーザの相互作用の理解の仕方はそれとは全く異なっていて、相互作用に関連のある事象を相互作用が起こった時点における状況という文脈の中で解釈する、という局所的な事象の解釈(解釈的説明)に基づいているのであり、従って、相互作用は一見したところ非常に詳細なプランに基づいているように見えるけれども、実際はトップレベルのプランにはほとんど影響されていないというのである。そして、このギャップがコミュニケーションにおける問題を引き起こしているのである。

このように、このモデルでは相互作用が起こっているまさにその瞬間に分析の目を向けることにより、タスク

解析的アプローチでは見落とされていた側面に光をあてることに成功している。

認知単位表現

(Kitajima, 1989)

あまり複雑でない装置を使用している状況を解析する場合には、Suchman の提案している枠組みで十分であるが、より複雑な装置、あるいはシステム(例えば、ワード・プロセッサ)の使用状況を解析するには表現力に欠ける。Kitajima はより記述が詳細であり、かつ、**ユーザ・機器相互作用の局所性**を取り入れた相互作用過程の表現方法を提案している。

一般に「ユーザ」と「機器」が相互作用している状況を記述しようとする場合、そのどちらを主体として表現しても構わない。つまり、「ユーザ」も「機器」もそれぞれ他方の振る舞いを観測・表現する言葉を持っているから、どちらの表現言語を用いても両者の間で生じている相互作用過程を表現することができる。Kitajima は、この観測的な側面を表現するために、その記述要素として認知心理学の分野で宣言的知識を表すのに利用される**認知単位**(cognitive units; Anderson, 1983)を導入した。

ユーザ・機器相互作用の文脈においては、認知単位は一方の観測主体が他方を観測したときにその状態を表現するための概念と、その概念に付随して参照される概念のさまざまな側面である属性の組、

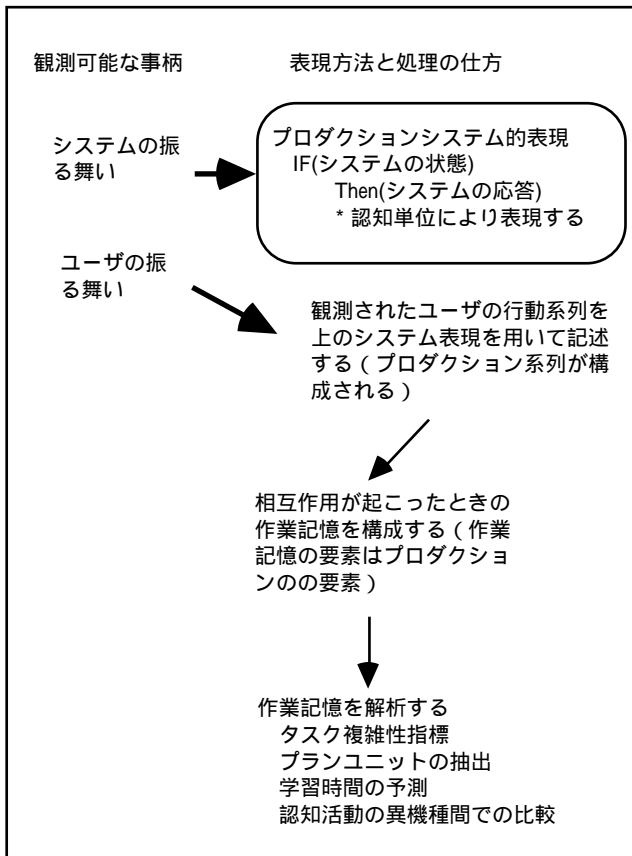
認知単位 (概念 属性)

として定義され、個々の観測主体(ユーザとシステム)に固有な相手の行動を表現するための記述要素である。

ユーザ・機器相互作用過程におけるユーザの認知的活動を解析するという目的にとっては、明らかに、ユーザを観測主体とした認知単位(ユーザがシステムの状態や反応を表現するときに使う表現言語)を選択するのが適当である。ユーザの認知単位は、具体的には、次のようなものである。一般的な編集作業を例に取れば、概念としては、編集集中のテキスト、カーソル、システム・メッセージなどが挙げられる。属性としては、カーソルを例にとれば、

属性	値
色	{緑、赤、青、・・・}
行	{1,2,3,・・・}
列	{1,2,3,・・・}
位置	(<カーソル 行><カーソル 列>)

等であり、カーソルという概念に関して参照できる側面を枚挙したものである(図 A8)。



上の図はユーザと機器の相互作用過程における様々な認知的活動を、認知単位を基本記述要素とする表現方法によって解析するための方法を示している。図の左側には客観的に観測される事柄が示されている。具体的には、システム側においては ユーザの何らかの作用に対する反応としての機器の動作(システムの振る舞い)、また、ユーザ側においては 一連の操作過程(ユーザの振る舞い)である。この部分は Suchman の解析の枠組みと同様の要素である。図の右側にはこれらがどのように表現され、解析されるかが示されている。

まず、機器の個々の振る舞いはプロダクション・システム的に、

If (システムの状態) Then (応答)

のように表現される。そして、システム全体の振る舞いは、全てのユーザの作用に対するシステムの応答を網羅することによって表現される。ここで、ルール記述はユーザの認知単位を用いてなされる。このようにして認知単位表現されたシステムは認知単位のレベルにおいて実際のシステムと全く同じ振る舞いをする。

さて、実際のシステムにおけるユーザの操作過程をこのシステムに入力することにより、条件部が適合し実行されるシステム・ルールの系列(トレース)を得ることができる。トレースの各々の要素は認知単位で表現されたシステム・ルールの条件部、実行部であり、各入力時点においてユーザが認知している事柄の断片を表している。次に、これらトレースの要素をいくつかのヒューリスティックな基準を用いて整理・再構成することによって、ユーザが各入力時点に抱いていたと想定される要素の集合からなる作業記憶が生成される。作業記憶の要素はシステム・ルールの条件部と実行部である。

この作業記憶を様々な観点から解析することにより認知的活動の諸側面を明らかにすることができる。例えば、作業記憶内要素数を指標として解析することにより作業の複雑さや学習に関する知見が得られ、また、作業記憶内要素の因果関係に着目することによりプランについて、また、異なった機器を使って同種の作業をしたときの作業記憶を比較することにより知識転移を解析できる。

### 認知的評価

この章では、今までに説明してきた手法が、長期記憶、短期記憶、知覚に及ぼす機器の影響を解析・予測するという観点からどの程度有効であるかを整理してみる。

### 長期記憶

長期記憶の負荷は学習の容易さに特に関係する。学習には、全く新たに操作を学習する場合もあるし、既存知識を援用しながら新しい機器の操作を学習する場合もあるが、いずれの場合も、何が学習されるのか、そして、その学習がどの程度困難なのかを解析することが重要である。各モデルが長期記憶に対する負荷をどのように扱



えるかを以下に考察する。

- 形式文法：長期記憶を直接的には表現していないが、学習時間を書き換え規則数によって、また、学習すべき要素数を終端記号数によって表現できる可能性がある。
- CCT (Polson, Bovair, & Kieras, 1987)、認知単位表現、GOMS：長期記憶はプロダクション (GOMS においてはメソッド) を単位として形成されるから、その数が増えるほど学習時間がかかる。
- GTN、CLG：長期記憶負荷を直接的には表していないが、長期記憶に影響する 1 つの側面を挙げることができる。いずれも、各時点におけるディスプレイの状態を表現できるので、次のアクションとして適切なものは何かを指示しているプロンプトがどのような状態の時にユーザに与えられるか、また、どのような状態の時に同じプロンプトが出されるかを評価できる。プロンプトが似ているほど、また、次に取るべき行動についての情報が少ないほど、ユーザは長期記憶からの想起に頼らなければならなくなる。
- タスク-デバイス・マッピング、ETIT、アナロジーによる学習：いずれも、新たな知識を獲得し長期記憶を形成するときに、既有知識 (タスク構造、External-Task、アナロジーによってトランスファされる知識) がどのように影響するかを解析できる。既有知識が有効利用されるほど長期記憶の負荷は少ない。

### 短期記憶

人間の短期記憶の容量には限界がある。有限な短期記憶に過度な負荷がかかると、エラーの増加、処理効率の低下 (処理時間の増加) をもたらし、使い勝手の悪さの原因となる。モデルが、短期記憶に対する負荷を考慮するためには、モデルの記述が相互作用の過程を動的に、また、ある程度以上の細かさで捉えていることが必要である。その点で、Situating Action、タスク-デバイス・マッピング、ETIT、アナロジーによる学習は除外される。

- 形式文法、CLG：センテンス長が短期記憶負荷量に関連すると考えられるが、そのセンテンスの各部分がどのような過程でいつ生成されたかがわからないと、短期記憶内要素 (この場合にはコマンド系列をどこまで行なったかという情報) 数を知ることができず、その関係は間接的で

ある。

- GOMS：短期記憶はゴール・スタックとして表され、タスク実行時における作業記憶負荷が変化する様子を見ることができる。また、個々のオペレータを実行する時間を評価することによりタスク実行時間を予測できる。
- CCT (Polson, 1987)：作業記憶を陽に備えている。タスク実行時間をプロダクション・システムのサイクル数、及び、作業記憶に付加されるプロダクション数によって予測できる。
- 認知単位表現：作業記憶を陽に備えている。タスクを遂行する過程において作業記憶内の要素が変化する過程を詳細に検討することにより作業記憶内要素数で表される作業記憶負荷を増大させている原因を同定することができる。

### 知覚

知覚、特に視覚による提示情報の認識は、相互作用過程における機器と人間との最も外側の接点を形成する。そこにおける負荷は、情報をユーザが取り込むときにユーザが求めている情報が適切な方法で提示されているか、ということに関連している。欲しい情報が、適切な場所に、適切なタイミングで、適切な表示方法で与えられなければ、使い勝手の悪い機器となってしまう。

CLGとGTNのみが知覚レベルまでの記述を可能としているのでこの過程における負荷を表現できる可能性がある。他の表現はこのレベルまで記述していないので知覚については無力である。CLGもGTNも各状態においてスクリーン上に表示される情報を知ることができるので、その情報とユーザが必要とする情報との適合性、ユーザが探しているものとターゲットの呈示方法 (スペーシング、フォント、色、位置) の関係、という観点から解析することができる。

### おわりに

以上、相互作用過程を表現するモデルを認知的解析力という観点から検討してきた。その結果、いずれのモデルも、相互作用過程において重要と考えられる認知的側面、長期記憶、短期記憶、知覚、の一部しか解析することができず、万能なものはまだ存在していない。しかしながら、現状のままであっても、これらの特質を正確に理解しながら、うまく合体、もしくは、併用することに

よって、実際の設計に役立てることは可能である。

今までにはなかったような新しい概念構造をベースにしたシステムを作り出すときには、相互作用に含まれる概念を大まかにつかむだけで十分であろう。そのような場合には、詳細なシミュレーションを行う前に概念構造がユーザに受け入れられるかを調べることが必要であり、その方法としては、ゴール構造グラフ、タスク-デバイス・マッピングなどの粗いモデルが適している。また、これらにより、サブ・ゴールの深さ（短期記憶負荷）をざっと見積ることもできる。

概念構造がある程度固定された後は、具体的なシミュレーションを行って認知的評価を行うことが必要になる。シミュレーションを行えるモデルは種々あるが、それぞれ力点が違っている。実行時間を予測する上ではGOMSやその工学モデルであるKeystroke-levelモデル(Card, Moran & Newell, 1980)、CCTが適しているし、学習時間の予測にはCCT、認知単位表現が、また、相互作用のまさにその瞬間の様子を解析するには、Suchmanの枠組み、認知単位表現が向いている。

長期記憶、短期記憶の解析においては表現方法に共通点が見られるが、知覚レベルでは様相が異なっており、そこを扱えるのは、GTNとCLGだけである。それは、長期記憶や短期記憶を解析できる枠組みがすでに認識、意味付け、選択がなされた情報の高いレベルの上に構築されている一方で、GTNやCLGが情報を提示するシステムの側の立場で、ユーザの意思によって選択される以前の低いレベルの情報を全て平等に扱っているからである。

しかしながら、GTNやCLGの枠組みで解析が行なえるのは、スクリーン上に表示されている内容という側面に関してのみであり、それだけでは、まだ不十分である。この種の解析では、スクリーン上で探索される情報の特徴と、その情報がどのように提示されるかを対応付けて解析することが必要である。つまり、ユーザは自分が探している単語やシンボルを心に抱いており、それがどこにあるかについて何らかの考えを持っている。一方、情報がどのようなフォント、色、強調、プリンキングで提示されるかは、背景からの識別容易性に影響を及ぼす。また、情報の種類によって提示される場所が異なれば、前もって次に提示される情報の場所を予測することができる。どれくらい簡単にスクリーン上から欲しい情報を捜し出せるかは、使い勝手を左右する重要な要因の一つであり、詳細に評価する必要がある。

相互作用過程を統一的に表現できる枠組みは、よりよいインタフェースを構築する際に必須であるが、それを実現するためには、知覚レベルを扱える表現と、それより高いレベルを扱える表現の融合が必要である。次のPartで取り上げるグラフィカルインタフェースでは視覚情報が非常に多く用いられ、より一層包括的なモデルがそのインタフェース評価には必要となる。

## 文献

- ANDERSON, J. R. (1983). *The Architecture of Cognition*, Cambridge, MA: Harvard University Press.
- BUTLER, K., BENNETT, J., POLSON, P., and KARAT, J. (1989). Report on the Workshop on Analytical Models; Predicting the Complexity of Human-Computer Interaction. *SIGCHI Bulletin*. vol. 20, no. 4. pp. 63-79. New York: ACM.
- CARD, S. K., MORAN, T. P., and NEWELL, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.
- CARD, S. K., MORAN, T. P., and NEWELL, A. (1980). The Keystroke-Level Model for User Performance Time with Interactive Systems. *Communications of the ACM*. vol. 23, no. 7. pp. 396-410.
- DOUGLAS, S. A. and MORAN, T. P. (1983). Learning Text Editor Semantics by Analogy. in *Proceedings of the CHI '83 Conference: Human factors in computing systems*. pp. 207-211. New York: ACM.
- DRAPER, S. W. and D. A. NORMAN. (1986). *USER CENTERED SYSTEM DESIGN: New Perspectives on Human-Computer Interaction*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.
- GREEN, T. R. G. and PAYNE, S. J. (1986). Task-action grammars: A model of the mental representation of task languages. *Human-Computer Interaction*. 2. pp.93 - 133.
- KIERAS, D. and POLSON, P. G. (1983). A generalized transition network representation of interactive systems. in *Proceedings of the CHI '83 Conference: Human factors in computing systems*. pp. 103-106. New York: ACM.
- KIERAS, D. and POLSON, P. G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*. vol. 22. pp. 365-394.
- KITAJIMA, M. (1989). A formal representation system for the human-computer interaction process. *International Journal of Man-Machine Studies*. vol. 30, no. 6. pp. 669-696.
- LAVE, J. (1988). *Cognition in Practice*. Cambridge

University Press.

- MORAN, T. P. (1983). Getting into a System: External-Internal Task Mapping Analysis. in *Proceedings of the CHI '83 Conference: Human factors in computing systems*. pp. 45-49. New York: ACM.
- MORAN, T. P. (1981). The command language grammar: a representation for the user interface of interactive computer systems. *International Journal of Man-Machine Studies*. vol. 15, pp. 3-50.
- NEWELL, A. and SIMON, H. A. (1972) *Human Problem Solving*. Englewood Cliffs, New Jersey: Prentice-Hall.
- NORMAN, D. A. (1988). *THE PSYCHOLOGY OF EVERYDAY THINGS*. New York: Basic Books.
- NORMAN, D. A. (1986). Cognitive Engineering. in *USER CENTERED SYSTEM DESIGN: New Perspectives on Human-Computer Interaction*, Eds. D. A. Norman & S. W. Draper. pp. 31-61. Hillsdale, New Jersey: Lawrence Erlbaum Associates.
- OLSON, J. R. (1987). Cognitive Analysis of People's Use of Software. in *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, ed. J. M. Carroll. pp. 260-293. Massachusetts: MIT Press.
- POLSON, P. G., BOVAIR, S., and KIERAS, D. (1987). Transfer Between Text Editors. in *Proceedings of the CHI + GI '87 Conference: Human factors in computing systems and graphics interface*. pp. 27-32. New York: ACM.
- POLSON, P. G. (1987). A Quantitative Theory of Human-Computer Interaction. in *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, ed. J. M. Carroll. pp. 184-235. Massachusetts: MIT Press.
- REISNER, P. (1981). Formal Grammar and Human Factors Design of an Interactive Graphic Systems. *IEEE Transactions On Software Engineering*. vol. SE-7, no. 2. pp. 229-240.
- SUCHMAN, L. A. (1987). *PLANS AND SITUATED ACTIONS: The problems of human machine communication*. Cambridge University Press.

図 A1 – Norman のアクションサイクルモデル (Norman, 1986)

☒ A2 – Generalized Transition Network (GTN) (Kieras and Polson, 1983, 1985)

☒ A3 – Cognitive Complexity Theory (CCT) (Kieras and Polson, 1985)

☒ A4 – Task-Device Mapping, GTN and CCT (Kieras and Polson, 1985)

☒ A5 – External Task – Internal Task Analysis (Moran, 1983)



図 A6 – Learning By Analogy (Douglas and Moran, 1983)

☒ A7 – Transcription from Suchman's Experiment

☒ A8 – Examples of Cognitive Units

## Part II

# ディスプレイ・ベースのHCIの 認知モデル<sup>2</sup>

### あらまし

本 Part では、Windows や Macintosh などに代表されるようなグラフィカルインタフェースに関する非常にホットな研究について紹介する。

グラフィカルインタフェースの大きな特徴は、ユーザが操作を選択し実行する過程がディスプレイに表示される視覚情報にガイドされながら進行することである。従来のコマンド入力を中心としたコマンド・ベースのインタラクションでは予め周到に計画された操作系列が実行されるのと比較すると、非常に柔軟で、また、記憶に負荷をかけない、優れたインタラクションの方式である。本 Part では、このようなディスプレイ・ベースのインタラクションの特質を理論的に理解するためのモデルについて説明する。

また、モデルに基づいてシミュレーション実験を行うことによりグラフィカルインタフェースを使うユーザの認知活動のさまざまな側面を理解することができるが、ここでは、その中からエラーをとりあげ、エキスパート・ユーザが犯すエラーについて、その発生する理由の説明をモデルに基づいて行う。さらに、一旦犯されたエラーからどのように正常操作に戻るかについてモデルに基づいて考察できることを示す。

### はじめに

#### グラフィカル・インタフェース

Windows や Macintosh に代表されるようなグラフィカルインタフェースが盛んに利用されるようになった背景には、ハード、ソフト、両面の進歩がある。ハード面では、精彩な画像を表示できるビットマップ・ディスプレイ、画像を高速に表示できる CPU の開発・普及がその根底にある。

従来のインタフェースがコマンド入力を主体としたものであり、シンタックスに関する知識を持っていることが必要であったのに対し、グラフィカル・インタフェースでは、コンピュータの画面上に表示される対象（アイコン、メニューなど）、それ自身、または、それらの間

<sup>2</sup>本原稿のPart II は、「ディスプレイ・ベースの HCI の認知モデル」情報処理学会研究グループ報告、93-TCG-1-9, 1993、をもとに若干の修正を加えたものである。

の関係から、その機能（何ができるのか）や結果（何が起こるのか）が容易に連想されることを利用して、認知的な負荷の低減や、誤操作の防止が図られている。

グラフィカル・インタフェースに用いられている基本的概念は WIMP と呼ばれている。これは、Windows、Icons、Menus、Pointing device の最初の文字を結合したものである。この概念の上に、デスクトップ・メタファを用いて、机の上のような環境をディスプレイの上に実現して、その上で、アイコンやウィンドウの中に表示された対象をマウスでポイントし、作業を進める直接操作方式（Hutchins, Hollan & Norman, 1986）は、このようなグラフィカル・インタフェースの好例である。

人間は、非常の多くの情報が凝集されている視覚情報の中から、状況に応じて最も必要とされる情報に注意を向け、それを利用して次に行なうべき操作の選択を行なっている。グラフィカル・インタフェースは、人間のこの優れた特性を積極的に利用しようという試みである。

*Communication of ACM* で最近組まれたグラフィカル・インタフェースの特集 (Mandelkern *et al.*, 1993) に報告されているように、今後もますますハードウェアが進歩し、現在よりも多様な仕方で情報が表示できるようになることは確実である。本講演では、その発展をささえる基盤研究として、ユーザモデルの開発について説明し、例えば、

- なぜ、グラフィカル・インタフェースが使いやすいのか、
- どのようにしてユーザの認知的な負荷が低減されているのか、
- どのようなエラーが、どのようなときに発生しやすいのか、

といった点についての回答のヒントを求めたいと思う。

#### ディスプレイ・ベースの問題解決

グラフィカル・インタフェースを人間がどのように使っているか、それを考えるための一つの視点は、我々が日常生活において機器を使うときに、それらの見え方が次に行なうべき操作の決定にどのように関与するのか、を考えることによって与えられる。ここで見え方は単に何が見えたか、といった知覚レベルよりも、むしろ、さらに処理が進んだレベル、見えたものどうしの関連も意味づけられたレベルをさす。機器のボタンやスイッチの物理的な構造が次に行なえる操作を想起させたり、操作の

範囲を限定し、エラーの発生を未然に、しかも自然に防いでいることは、身の回りの機器を考えるとよく分かるであろう。

Larkin (1989) はコーヒーメーカーでコーヒーを沸かすことや、代数の問題をノートの上で解くといった問題解決の場面を ディスプレイ・ベースの問題解決 (Display-Based Problem Solving) とよび、そのプロセスをプロダクション・システムを用いて分析している。彼女は以下のように、ディスプレイ・ベースの問題解決を特徴付けている。

- 1) プロセスは簡単である
- 2) 殆どエラーを犯さない
- 3) 割り込みによる影響を受けない
- 4) ステップは種々の順序で実行される
- 5) プロセスの変更は簡単である
- 6) タスクをスムーズに簡単に行なうには学習が必要である

これらの特徴は、そのままグラフィカル・インタフェースを用いてタスクを遂行するプロセスにもあてはまるように思われる。本講演では、この考え方に基づいて研究を進めている、ユーザの認知プロセスモデルについて述べる。このモデルは、Larkin がプロダクション・システムを用いて行なっている分析よりも詳細なレベルでの分析を行なうためのモデルであり、ユーザが、グラフィカル・インタフェースを備えた装置を利用するとき、どのような知識をどのように使って次に行なうべき操作を選択し実行するのか、その実行プロセスを認知のレベルでモデル化する。

本講演では、まず、モデルで扱おうとしているタスクとその遂行プロセスを例を挙げて示し、グラフィカル・インタフェース使用プロセスが、ディスプレイ・ベースの問題解決の諸特徴を持っていることを示す。次に、モデルの提示を行なうが、最初に、ユーザとコンピュータの相互作用プロセスの基本的な視点を与えている Norman の行動理論を説明し、そのグラフィカル・インタフェースの認知プロセスモデルへの適用を説明する。

最後に、本モデルを用いて、エキスパート・ユーザが犯すエラーの発生メカニズムについて考察する。本モデルの大きな特徴は、特にエラーを発生させるための特別な誤った知識を仮定しなくても、一見、偶発的に犯されるエラーをシミュレートできることである。また、従来の認知モデルではエラーを発生させるための特別な知識を想定する必要があったが本モデルでは一切そのような

「エラー知識」は不要である。知識ではなく認知プロセス自身にエラー発生の原因がある。この点に焦点を合わせたシミュレーション実験の結果についても触れる。

### タスクと実行プロセスの例

デスクトップメタファーを用いたアプリケーション・ソフトウェア上で実行されるタスクの遂行過程は、本モデルで理解しようとしているインタラクションの代表例である。以下では、Macintosh のグラフ作成ツールのひとつである Cricket Graph をグラフィカル・インタフェースの例としてとりあげ、仮想的なユーザのグラフ作成作業の遂行過程をシミュレートする。いかにディスプレイが操作選択に重要な役割を果たしているかの感触を得てもらいたい。

#### タスクの例

ユーザにタスク、

*印刷されたグラフ(図1)を指定されたファイルのデータを用いて再構成する*

が与えられたものとする。コンピュータ化されたタスクは、通常、ユニット・タスクに分解できる(Card, Moran & Newell, 1983) が、この場合も、次のようなサブ・タスクに分解可能である。

- サブ・タスク1: “Serial Position” をX 軸、“Observed” をY 軸として折れ線グラフを描く。
- サブ・タスク2: グラフのタイトルを編集する。
- サブ・タスク...: ●●●

ユーザは、タスクを開始するに当たって、グラフを作成するのに必要なデータが、ドキュメント・ファイル “Example Data” に納められていると教示されている。以上が、タスクが開始された時の状況の記述である。

ユーザがまず最初に行なうことは、“Example Data” アイコンをダブル・クリック(マウスのボタンを2回連続して短時間のうちに押すこと)し、ドキュメント・ファイルを開くことである。これによりデータを納めたスプレッド・シートが表示される。また、同時に、Cricket Graph の作業環境に入ることができる。図2(a)は、この最初の操作によって表示される画面を示している。

この状態を最初の状態として、サブ・タスク1は、例えば、次のような操作の系列により遂行される。

ステップ1: マウスカursorを Graph に移動する

ステップ2: マウスボタンを押し続ける

図2 (b) に示すプルダウン・メニューが表示される

ステップ3: マウスカursorを Line に移動する。

ステップ4: マウスボタンを放す。

図2 (c) に示す変数選択のためのダイアログ・ボックスが表示される

ステップ5: マウスカursorを X 軸選択リストの項目、**Serial Position** に移動する

ステップ6: マウスボタンを1回クリックする

ステップ7: マウスカursorを Y 軸選択リストの項目、**Observed** に移動する。

ステップ8: マウスボタンを1回クリックする

ステップ9: マウスカursorを **New Plot** ボタンに移動する

ステップ10: マウスボタンを1回クリックする

図2 (d) に示す標準の折れ線グラフが表示される

ステップ1からステップ10を実行することにより、最初のサブ・タスクは達成されることになるが、目標とするグラフ(図1)との違いはまだ大きく、それを縮めるために、サブ・タスク2を達成しようとする。

サブ・タスク2は、ステップ10によって得られる図2(d)に示した画面を初期画面として、例えば、次のように遂行される。

ステップ11: マウスカursorを **Data from "Example Data"** (グラフのタイトル) に移動する

ステップ12: マウスボタンをダブル・クリックする

図2 (e) に示すグラフ・タイトル編集用のウィンドウが表示される

このようにして、順次サブ・タスクが遂行され、最終的に図1のゴールが達成される。

ここでは、2つのサブ・タスクの実行のようすを、画面の変化と共にその過程を追跡してきた。重要な点は、ユーザが何らかの操作を行なうと、ディスプレイに表示されている内容が大きく変化する ことである。例えば、プルダウン・メニュー(図2(b))、変数選択のダイアログ・

ボックス(図2(c))、標準の折れ線グラフ(図2(d))、グラフ・タイトル編集用のウィンドウ(図2(e))は、いずれも、ユーザの操作にシステムが応えてスクリーン上に表示したものであり、それ以前の表示内容が大幅に変更され、次の操作を選択するときの文脈を与えていることが分かる。

ここで述べたシミュレーションは、グラフィカル・インタフェースを用いたタスクの遂行過程の一例に過ぎない。また、そこに含まれる認知的なプロセスに深く踏み込んだ分析をしているわけでもない。しかしながら、ディスプレイに表示されている情報が次の操作の決定に大きく影響し、それにより、操作選択の容易さ、エラーの発生の防止、割り込みの影響の受けにくさ、等、Larkinの主張しているディスプレイ・ベースの問題解決の特徴を再確認することができる。

### アクション・サイクル

人間が一般の機器をどのように使うのか、その心理プロセスの見方のひとつに、Norman (1988, pp. 46 - 47) のアクション・サイクルがある。図3にその概略を示す。

Normanの考え方は非常に単純である。人間は、何かを成し遂げたいとき、まず、達成されるべき状態、即ち**ゴール**を持って機器に臨み、そして、それを達成するために**世界**に何らかの**働き掛け(実行)**を行なう。また、その結果は、**ゴール**が達成されたかどうかという観点から**評価**される。そして、この評価に基づき、次に行なうべき操作が決定され、実行される。この4要素を、人間が目的をもって機器を扱う場合に考えなければならない基本要素と考えるのである。

Normanの考え方は、ゴール志向の行動を考えるための一般的な枠組みを提供している。コンピュータ化されたタスクは、高度にゴール志向的であり、また、世界(この場合はインタフェース・デバイス)に対する操作、そして、その結果の評価がサイクリックになされる。本講演で提案する心理プロセスモデルは、Normanのアクション・サイクルの考え方を踏襲し、インタラクションが高度にディスプレイ・ベースに行なわれる場合に発展させたものである。

図4にディスプレイ・ベースのHCIの認知モデルの概略を示す。モデルは大きく分けて、

- ・ハードウェア、
- ・ゴール、
- ・実行のステージ、
- ・評価のステージ、

の4要素から構成される。これは、Normanのアクションサイクル・モデルと同じである。

図4の下部には世界を構成するハードウェアが示されている。そこには、視覚情報を表示するディスプレイ、また、操作が行なわれる装置である、マウスやキーボードが含まれる。他の3要素(ゴール、実行のステージ、評価のステージ)は、いずれも心理的な内容を含んだ要素であり、それらを定義することによってモデルの記述が与えられることになる。

### タスク・ゴールとデバイス・ゴール

図4の上部には2種類のゴールが示されている。Normanのモデルでは明確には区別されていなかったが、機器操作の心理モデルを考えると、以下に定義するように、ゴールを2種類に分けて捉えるのが重要である。この仮定は、*Yoked State Space Hypothesis* として提案され、その妥当性の検証が行なわれている (Payne, Squibb, & Howes, 1990)。

1つのゴールは、タスク・ゴールである。これは、タスクをサブ・タスクに展開したときに構成されるタスク・スペースの要素であり、その表現はタスクに直接関連したことばで与えられる。例えば、先に示した、サブ・タスク1、サブ・タスク2はタスク・ゴールの例である。いずれも、タスクのことば、つまり、図1に示した再構成すべきグラフの要素や要素の関係を表す言葉で表現したものになっている。

もう1つのゴールは、デバイス・ゴールである。これは達成したいデバイスの状態の心的表現である。ディスプレイ・ベースのHCIにおいてデバイス・ゴールによって表現されるものは、ディスプレイに何が、どのように表示されているか、に関する心的な表現である。図2に示したタスク遂行過程の途中に現われるディスプレイ状態のそれぞれに対して与えられる心的表現が、デバイス・ゴールに対応する。例えば、図2(b)では、メニュー・バーに表示されている項目、それ自身、また、その状態(ハイライト、グレー、点滅、等)がディスプレイ表現の要素となる。デバイス・ゴールを相互に関連づけることによって、デバイス・スペースが構成される。

タスク・スペースとデバイス・スペースに関する知識は長期記憶に保持されていると仮定するが、それらがどのように活性化され、また利用できるようになるか、その過程について、前節で示したシミュレーションを用いて説明する。

まず、タスク・ゴールとそのサブ・タスクへの分解は、タスクがユーザに与えられた時に、タスクに関連したタスク・ゴールを長期記憶から引き出すことによってなされる。ここでは、標準的な(ディスプレイ・ベースでない)問題解決のプロセスが働いていると考えられる。このようにして得られたのがサブ・タスク1と2である。そして、ディスプレイの状態を観察し、その心的表現を連合のキーとして、タスク・ゴールに関連したデバイス・ゴールが長期記憶から引き出される。例えば、サブ・ゴール1が設定されたとき、図2(b)、(c)、(d)を表現したデバイス・ゴールが長期記憶から引き出される。どのタスク・ゴールにどのデバイス・ゴールを結び付けるのか、ということは、操作の経験・学習を通じて獲得されるものであり、この知識も、長期記憶に格納されていると仮定する。このようにして、サブ・タスクの心的表現 ディスプレイの心的表現 デバイス・ゴールの活性化、が行なわれる。

タスク・ゴール、デバイス・ゴール、ディスプレイ上への情報呈示、の一組が既に与えられているものとして、ディスプレイ・ベースのHCIの認知モデル(アクション・サイクル)の説明を、図4に従って、以下に述べる。まず、評価のステージ(図の右側)について説明し、次に、実行のステージ(図の左側)について述べる。なお、モデルに用いられているパラメータや仮定の詳細については、Kitajima and Polson (1992)を参照されたい。

### 評価のステージ

評価のステージは、ディスプレイ表現の形成、ゴール達成の検査、ディスプレイ表現の精緻化の3つのプロセスから構成される。

#### ディスプレイ表現の形成

ディスプレイに表示されているものは、物理的には、数百万のピクセルの濃淡パターンである。しかしながら、知覚レベルでは、近接するピクセルに表示されている情報を関連付けることにより、表示されているパターンから、何が、また、どのように表示されているのか、という情報が抽出される。

知覚レベルの表現には、対象が何であるか、という対象のアイデンティティに関する表現と、対象の見え方、アピラランスに関する表現が含まれる。例えば、本モデルでは、図 2 (a) のメニューの 1 つの項目である **Graph** のアイデンティティに関する表現として、次のものを与えている。

- ・ **Graph** はメニューの項目である
- ・ **Graph** は操作可能な対象である
- ・ ...

また、アピラランスに関する表現は、次のように与えられている。

- ・ **Graph** はポイントされていない
- ・ **Graph** はハイライト表示されていない
- ・ ...

この例からもわかるように、ディスプレイ表現は知覚レベルの表現なので、対象に対してどのような操作が可能なのか、といった、操作決定には不可欠な情報が何も含まれていないのが特徴である。本モデルでは、このような情報は、対象の意味に関する情報であり、ユーザの知識が係わってはじめて顕在化するものであると仮定している。

#### ゴール達成の検査

このように構成されたディスプレイ表現は、まず、それがデバイス・ゴールを満たしているかどうかについてチェックされる。なお、先に述べたようにデバイス・ゴールは、ディスプレイに表示される対象により表現されているので、デバイス・ゴールはディスプレイ表現と照合ができるのである。

もし、現在のデバイス・ゴールが満たされているならば、さらに、現在のタスク・ゴールに関連づけられているデバイス・ゴールが全て満足されたかどうかテストされ、未達成のデバイス・ゴールがあれば、それを新たなデバイス・ゴールとして、次のプロセス(ディスプレイ表現の精緻化)に進む。もし、すべてのデバイス・ゴールが達成されたならば、それらに関連したタスク・ゴールも達成されたことになるので、新たなタスク・ゴールの長期記憶からの検索・設定、連合されているデバイス・ゴールの検索・設定を行ない、新たな、タスク・ゴール、デバイス・ゴールの組み合わせのもと、次のプロセスへと進む。

#### ディスプレイ表現の精緻化

このプロセスは、表面的で、貧弱なディスプレイ表現を、長期記憶に蓄えられている知識を利用して深めるプロセスである。知覚レベルで与えられているディスプレイ表現は、そこに表現されている概念を手掛かりとして長期記憶に格納されている関連した情報を検索することにより精緻化される。この段階を経て始めて、ディスプレイの意味的な理解を導き出すことが可能となる。

長期記憶には、ディスプレイに表示される情報に関するさまざまな知識が納められている。**Graph** メニュー項目について例をいくつか挙げると、

- ・ **Graph** メニュー項目はプルダウンできる
- ・ **Graph** メニュー項目はテキストではない
- ・ **Graph** メニュー項目は折れ線グラフ・メニュー項目と連合している

・ ...

などがある。

知識の検索プロセスは、認知心理学における文章理解プロセスのモデル (Kintsch, 1988) における記憶検索の方法 (Raaijmaker, J. G. & Shiffrin, 1981) に基づいている。すなわち、ディスプレイ表現に現われる個々の命題(述語といくつかの引数によって定義される)と、(1) 長期記憶に蓄えられている命題、(2) ディスプレイを表現する命題、及び、(3) ゴールを表現する命題、の間の関連の強さは、それらの命題間で共有される引数の数に比例して大きくなり、連合が成立する確率はこの関連の強さに比例する、と考える。

ディスプレイ、または、ゴールを表現する個々の命題が、長期記憶に格納されている命題と連合を成立させたとき、その知識は有効になる。検索された知識のみが、操作対象の選択や、操作の選択のプロセスで利用可能である。長期記憶の検索の試みは、個々のディスプレイ表現、ゴールを用いて、数回にわたって行なわれるが、検索される知識の内容は、命題間に与えられた関連の強さの具体的な割当方によって大きく影響される。もし、ディスプレイ表現における命題が、特定の命題と非常に強く関連づけられている場合には、何回検索を行なっても、同じ命題との間の連合が成立する確率が高いため、結果として、狭い範囲の知識しか検索されないことになる。一方、特に強い関連性を持った命題がない場合には、多くの知識が幅広く検索される確率が高くなる。また、ディスプレイに表示されている対象は全て検索に用いられる。従って、精緻化のプロセスによって長期記憶から検索さ



れる知識の具体的な内容は、ディスプレイに表示されている対象に敏感に影響されることになる。

このように、本モデルでは、検索される知識の内容は、

- (1) ディスプレイされている対象の内容、
- (2) 関連の強さを反映して与えられる連合が成立する確率、
- (3) 長期記憶を検索する回数、

に応じて変化することになる。また、ゴールと精緻化されたディスプレイ表現は、操作を決定し、実行する際に重要な役割を担う文脈を規定する。文脈の詳細がディスプレイが与えられた後に、知識検索プロセスを経て形成されることは、本モデルの重要な特徴であり、エキスパートの犯すエラーについての新しい理解を与えることになる。この点については、後でまた触れる。

### 実行のステージ

何を実行すべきかの決定・実行は、これまでに集められた情報、すなわち、タスク・ゴール、デバイス・ゴール、そして、精緻化されたディスプレイ表現、を用いて、操作対象の選択、認知操作の選択、認知操作の実行、のプロセスを経て、なされる。図5は実行のステージを模式的に示している。図4の左上部から下方への実行のステージの流れは、図5の左から右への流れに対応する。

### 操作対象の選択

図2の、どの画面状態を見ても、ディスプレイには数多くの操作可能な対象が存在していることが分かる。本モデルでは、まず、どの対象が最も操作を行なうのに適切であるかを、既に得られた情報によって定義される文脈に照らして評価し、それに基づいて操作対象を選択する。

図5では、サブ・タスク2(グラフ・タイトルのエディット・タスク)の最初のステップ(正しい操作はマウスカーソルをグラフ・タイトルに移動する)が決定・実行されるときの流れが示されている。この場合には、恐らく、操作対象として、

Edit メニュー項目  
 Formats メニュー項目  
 Data from "Example Data" (グラフタイトル)

が選ばれるであろう。いずれも、グラフ・タイトル、エディットといった現在のゴールに密接に関連したディスプレイ対象である。

### 認知操作の選択

モデルでは、ユーザは操作に関する汎用的な知識(汎用的な認知操作知識)を長期記憶に保有し、ディスプレイの具体的な状況に合わせてそれを特殊化して用いる(特殊化された認知操作知識)と仮定している。ユーザが知識として持っているのは、操作の物理レベルでの表現、例えば、カーソルを移動する、マウスボタンを押す、等、ではなく、認知レベルの表現である。認知レベルの表現は、操作前後のディスプレイの状態で規定されていると仮定されている。ここで取り上げているタスクの場合、本モデルでは、物理的な操作として、

- (1) マウスカーソルを移動する、
- (2) マウスボタンを1回クリックする、
- (3) マウスボタンを2回クリックする、
- (4) マウスボタンを押し続ける、
- (5) マウスボタンを放す、
- (6) タイプする、

を考えているが、モデルでは、ユーザは、そのそれぞれに対して、ディスプレイの状態の操作前後の変化によって定義される複数の認知表現を長期記憶に操作知識として持っているとは仮定している。

現在のモデルでは、右端に示した6種類の物理操作は、18種類の汎用的な認知操作知識によって表現されている。図5で、●●●●は個々の認知操作に対応する。例えば、マウスカーソルを移動するという物理操作(図中、最上段)は、認知的には6種類のバリエーションによって表現されている。ここでは、マウスカーソルを、テキスト領域に移動する場合、メニューの項目に移動する場合、プルダウンメニューの項目に移動する場合、等を、物理的には同一でも、認知的には異なった操作として区別している。

汎用的な認知操作では、操作は任意のディスプレイ対象に対するものとして表現されている。この表現と、先のプロセスで選ばれた操作対象候補を組み合わせることにより、汎用的な認知操作を実行可能な、特殊化された認知操作の形式とすることができ、実際に物理的に実行される操作は、この特殊化された認知操作に基づいてい

る。

認知操作の選択のプロセスでは、特殊化された認知操作の中から、そのときの文脈に照らしてもっとも適当であり、また、実行可能な操作を選択することになる。なお、認知操作の表現には、それを実行するときに満たされていなければならない条件が含まれているが、精緻化されたディスプレイ表現によって規定される現在の文脈がそれを満たせば、その認知操作は実行可能となる。

操作対象の選択と認知操作の選択において、その時の文脈において最も適当なものを選ぶ、と述べたが、具体的には、ネットワークモデルで用いられる活性伝播機構を用いている。精緻化されたディスプレイ表現を、各命題をノードとするネットワークにより表現し、ノード間のリンクをいくつかのパラメタで決定する。そしてディスプレイ表現の命題と、ゴールを表現する命題を、活性伝播の源として、長期記憶から検索された命題を表すノードや、認知操作を表すノードを活性化させる。ネットワークの活性パタンが平衡状態に達したら、操作対象を表すノード、また、認知操作を表すノードのなかで活性レベルの高いものをそれぞれ選択する。

#### 認知操作の実行

前段によって選択された特殊化された認知操作は、対応する物理操作にマップされ、実際に実行される。

図中、太い線で結ばれたものは、操作対象 **Edit** と、物理操作 **マウスカーソルを移動する** に対応する汎用的な認知操作が結合して形成される特殊化された認知操作 **...のディスプレイの状態の時に、マウスカーソルを Edit に移動する** の関係を示したものである。

本モデルでは、長期記憶に蓄えられている操作知識として汎用なものを仮定している。実際に実行可能な形式は、ディスプレイが与えられて始めて形成される文脈を考慮して選択される操作対象との組み合わせプロセスを経て構成される、特殊化された操作知識である。この操作知識は一時的に保持され、利用される。このことは、ユーザは、特殊化された恒久的な知識を持つ必要がないことを意味しており、記憶の負荷がどのようにして軽減されるかについての説明を与えている。

## エラー

### エキスパートの犯すエラーのモデル

本モデルの特徴は、エキスパートのパフォーマンスを 非常に低いレベルの操作を生成するアクション・サイクルの繰り返し によって特徴付けている点である。標準的なエキスパートのモデル (Anderson, 1987; Bovair, Kieras & Polson, 1990; Newell, 1990) では、これとは対照的に、エキスパートの行動を 一連の操作系列に関する詳細な正しい知識の選択 によって特徴付けている。

この基本的な考え方の違いにより、エラーに対する説明も大きく異なってくる。例えば、ACT\* (Anderson, 1983, 1987) やSOAR (Newell, 1990) のような理論によれば、学習の初期の段階では行動は本質的に問題解決行動であり、学習の積み重ねにより正しい行動系列がプロダクション・ルールの形式で長期記憶に蓄積されるとしている。エキスパートは、初心者が習熟した状態として位置づけられ、そのパフォーマンスの特徴は、長期記憶に完備された正しい規則と、それが確実に検索されるということである。従って、エキスパートのエラーの説明は、正しい規則が偶然に適用されないようにする仕組みを認知機構に仮定し、また、そのときに適用される誤った規則をエキスパートの知識に組み込む ことによってなされる (Brown & VanLehn, 1980)。

一方、本モデルによれば、エキスパート・ユーザが典型的なタスクにおいて非常に多くのエラーを犯すことの非常によい説明を、全く別の観点から与えることができる。まず、エキスパート・ユーザであるから、タスク・ゴール、デバイス・ゴール、汎用的な認知操作、長期記憶に蓄えられた知識、全てが、正しいと仮定できる。本モデルでは、知識が全て正しくても、精緻化のプロセスで得られるディスプレイ表現が、不完全、不正確な場合に、エラーが発生することが予測される。

ユーザがおかれている状況により、精緻化の程度は深くもなり、浅くもなる。例えば、タイムプレッシャーが非常に厳しいとき、また、注意レベルが低いときには、そのレベルは浅くなることが予想される。精緻化プロセスは本質的に記憶検索プロセスであるから、その時間は検索のための記憶走査の回数に比例する。従って、深い精緻化には時間を要することになるが、また、そうすることにより、重要な情報が見逃されない確率も増加することになる。深い精緻化により、タスクに関連した情報が幅広く検索され、正しい操作対象の選択、正しい操作の選択の確度が上がることになる。逆に、精緻化に時間

をかけなければ、エラーの確率が增大することになる。精緻化の深さの程度は 速さと正確さをどのようにトレードオフするか という判断の結果であると考えられる。

精緻化プロセスに起因するエラーは、その深さばかりでなく、ディスプレイに表示されている情報にも依存する。操作を実行するのに欠くことのできない重要な知識のよい検索手掛かりを与えないようなディスプレイが与えられた場合には、たとえ、正しい知識を持っていたとしても、エラー発生率の確率が增大する。

本モデルでは、エラーという言葉で、標準的なモデルが主張する 誤った操作知識としてのエラー とは異なり、不完全な文脈においては最も適当な操作として選ばれた操作が、完全な文脈に基づく操作とは異なっている現象をさしている。そして、正しい操作を産み出すのと全く同じ知識と認知プロセスが、ディスプレイの理解の深さに依存して、このようなエラーをも引き起こす。エラーを、ディスプレイの誤解に基づく“正しい操作選択” と考えている。

### シミュレーション実験

図6は例で述べた Cricket Graph タスクを用いて、知識検索の回数を様々に変化させてシミュレーションを行なった実験の結果を示している。知識検索の回数として、4、8、12、16、20、24 を与え、それぞれについて10回のシミュレーションを行ない、その結果を、知識検索回数の小さいグループ(4、8、12)と大きいグループ(16、20、24)の2つにわけて、平均正操作選択率を操作のステップを横軸にとって表したものである。

なお、この実験は、エラーの様相を考察することを目的として行なっているため、長期記憶に格納されている知識、タスク・ゴール、デバイス・ゴール、認知操作の表現は予め用意したものをを用いている。また、完全に自立的にシミュレーションが進むわけではなく、重要な節目で正しいディスプレイの状態に戻され(ステップ1、3、4、5、7、9、11、12の前)、これらのディスプレイに対する表現として、アイデンティティ、アピランスが適当に与えられている。また、操作対象の選択には活性伝播機構を用いず、予め用意した操作対象の組を与えることでそのプロセスをシミュレートしている。ディスプレイ表現の精緻化プロセス(知識検索過程)と、操作選択プロセスは、計算機プログラムにより自動化され、シミュレートされている。

図から、知識検索回数が少ないグループで、正操作選

択率が顕著に小さくなる箇所がいくつかあることがわかる。以下に、シミュレーション結果から読み取れるいくつかの特徴を挙げる。

第2、4ステップでは知識検索の回数によらず正しい操作が選ばれている。また、これは、選ばれているメニューの項目の正誤に依存していない。メニューの項目が選ばれている状態に対するディスプレイ表現が、それに続く正しい操作の選択のための十分な文脈を与えているためである。

第3ステップでは、プルダウンメニューの項目にマウスカーソルを移動する操作を選択することが、知識検索の回数が少ない場合には難しくなっていることがわかる。その理由は、本モデルにおけるプルダウンメニュー選択の認知操作を可能とするための条件を満たすために、長期記憶に格納されている比較的多くの知識が必要とされるためである。

第11ステップでは、グラフのタイトルにマウスカーソルを移動することが難しい操作になっていることがわかる。それは、このディスプレイでは、正しい対象であるグラフのタイトルのほかに、メニューの項目である Edit もタスク・ゴールに関連しているために、そこへマウスカーソルを移動する操作も高い活性レベルを獲得するためである。

第12ステップでは、正しい画面が与えられているにも関わらず、ダブルクリックを選択することが難しくなっている。これは、ダブルクリック操作の条件、

グラフタイトルが編集用のダイアログボックスにリンクしている

の検索が難しいことが原因である。実際、この命題は、グラフ・タイトルを表現する命題により検索されなければならないが、その確率が、グラフ・タイトルがゴールと強い関連を持つことにより、小さくなってしまっていることが、シミュレーション実験の結果を分析することによってわかっている。

### おわりに

本講演で述べたディスプレイ・ベースのHCIの認知プロセスモデルは、Normanのアクションサイクルモデルを基にして、ゴール、ディスプレイに表示される情報、長期記憶に蓄えられている知識、それらを用いて次に操作を行なうべき対象の選択と操作自身の選択がなされる

プロセスをモデル化したものである。モデルはいくつかのパラメタによってその特性（操作選択の速度、エラー率、等）をコントロールでき、適当なパラメタの値を用いることによりエキスパートのパフォーマンスをシミュレートできることが確認されている（Kitajima and Polson, 1992）。また、誤操作が行なわれる理由を、ディスプレイの状態や、正しい操作が実行されるための条件の点から考察できる。

本モデルを用いることにより、ゴール、ディスプレイ、長期記憶に蓄えられている知識が与えられ、仮想的なユーザの操作選択への構えが与えられたとき、エラー、実行速度、認知負荷など、さまざまなパフォーマンスの予測を行なうことが可能となる。また、逆に、エラーを減少させるようなディスプレイの設計を試すことが可能となる。操作説明書についても、その記載内容が長期記憶に蓄えられる要素となることを考えると、設計方法について、さまざまな代替案を試し、評価することが可能である。モデルに基づくパフォーマンスの解析により、ここに挙げたような点について実際の示唆を得ることが期待される。

グラフィカル・インタフェースをユーザがどのように使っているか、また、どのように学習していくのかについての認知レベルの研究は重要であるにも係わらず、インタフェース設計の指針を与えるほどには研究が進んでいない。研究の成果は、日常生活に利用される機器のインタフェースをよりユーザフレンドリイにするといった生活に密着した面ばかりでなく、産業面においても、例えば、大規模なプラント（原子力発電所など）や航空機のcockpitのインタフェース設計といった分野で緊急性、必要性が非常に高い。このようなニーズ応えるため、一層の研究が望まれている。

## 文献

- ANDERSON, J. R. (1983). *The Architecture of Cognition*. Cambridge, MA: Harvard University Press.
- ANDERSON, J. R. (1987). Skill acquisition: Compilation of weak-method solutions. *Psychological Review*, **94**, 192-211.
- BOVAIR, A. S., KIERAS, D. E. & POLSON, P. G. (1990). The acquisition and performance of text-editing skill: a cognitive complexity analysis. *Human Computer Interaction*, **5**, 1, 1-48.
- BROWN, J. S. & VANLEHN, K. (1980). Repair theory: A generative theory of bugs in procedural skills. *Cognitive Science*, **4**, 379-426.
- CARD, S. K., MORAN, T. P. & NEWELL, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- HUTCHINS, E. L., HOLLAN, J. D., & NORMAN, D. A. (1986). Direct manipulation interfaces. In Norman, D. A. & Draper, S. W., Eds. *User Centered System Design*. 87-124. Hillsdale, NJ: Lawrence Erlbaum Associates.
- KINTSCH, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, **95**, 163-182.
- KITAJIMA, M. & POLSON, P. G. (1992). A computational model of skilled use of graphical user interfaces. In *Proceedings of the CHI'92 conference on human factors in computing systems*. 241-249. New York: ACM.
- LARKIN, J. H. (1989). Display-based problem solving. In Klahr, D. & Kotovsky, K., Eds. *Complex Information Processing: The Impact of Herbert A. Simon*. 319-342. Hillsdale, New Jersey: Lawrence Erlbaum Assoc.
- MANDELKERN, D. et al. (1993). Graphical User Interfaces: The Next Generation. *Communication of ACM*. **36**, 36 - 109.
- NEWELL, A. (1990). *Unified theory of cognition*. Cambridge, MA: Harvard University Press.
- NORMAN, D. A. (1988). *The Psychology of Everyday Things*. New York: Basic Books.
- PAYNE, S. J., SQUIBB, H. R., & HOWES, A. (1990). The nature of device models: The yoked state hypothesis and some experiments with text editors. *Human-Computer Interaction*, **5**, 4, 415-444.
- RAAIJMAKER, J. G. & SHIFFRIN, R. M. (1981). Search of associative memory. *Psychological Review*, **88**, 93-134.

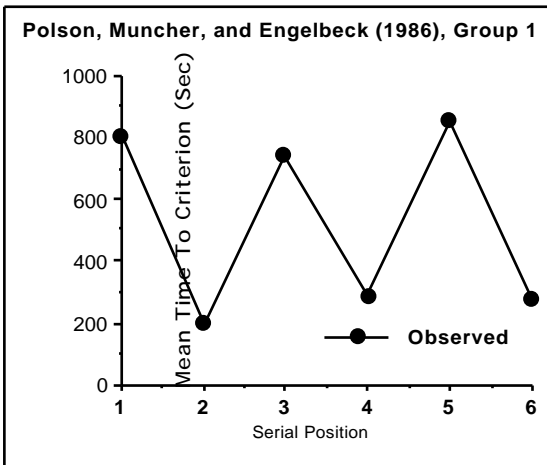


図 1 Cricket Graph タスク。ユーザのタスクはこのグラフを再現することである。

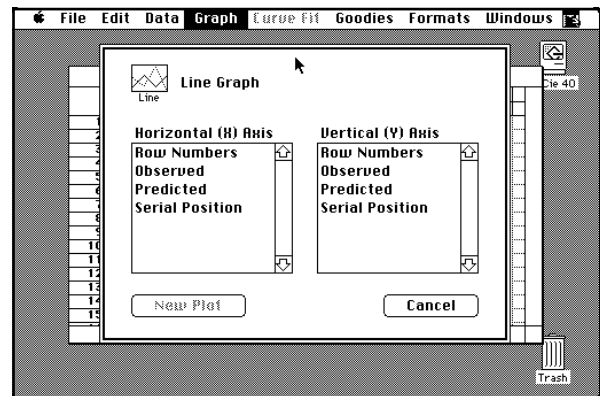


図 2 (c) 変数選択のためのダイアログ・ボックスが表示されているようす。

図 2 (a) “Example Data” をダブル・クリックしてデータの収まっているファイルが開かれたようす。

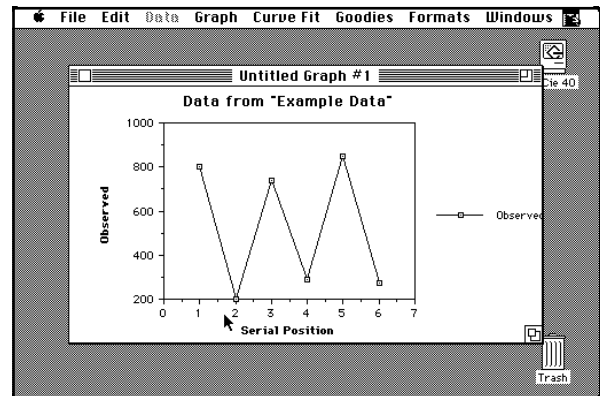


図 2 (d) 標準の折れ線グラフが表示されているようす。

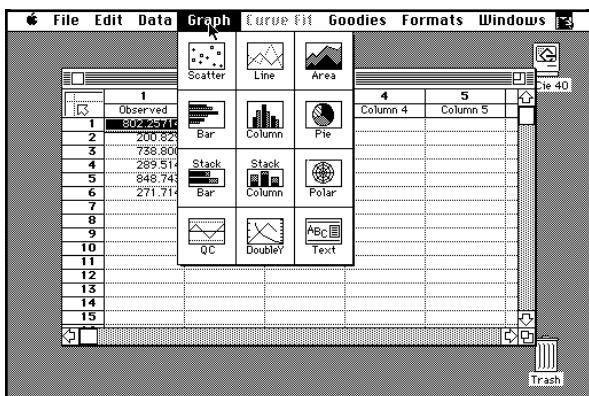


図 2 (b) Graph メニューがプルダウンされたようす。

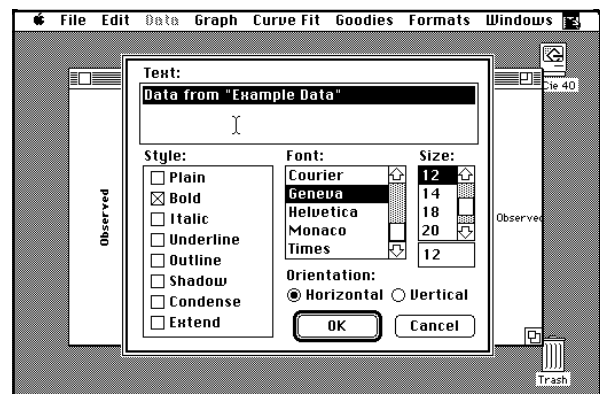


図 2 (e) グラフのタイトルを編集するためのウィンドウが表示されているようす。

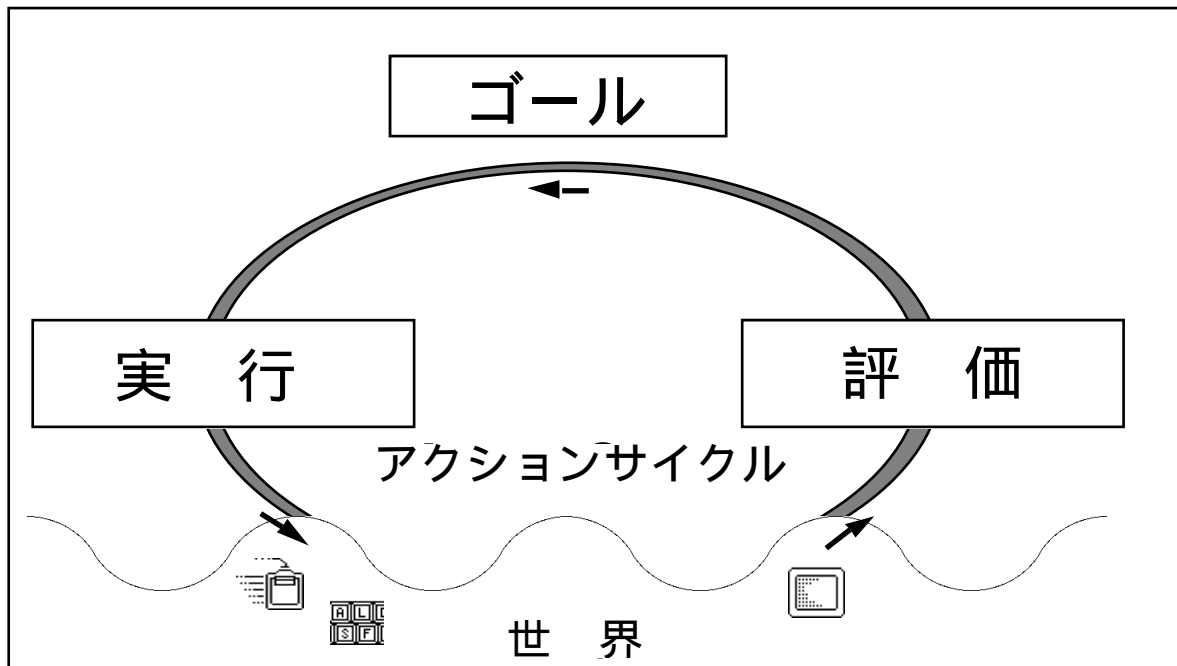


図3 Norman のアクション・サイクル・モデル. ユーザのゴールと装置 (世界) の間は 実行 と 評価 によって橋渡しされている.

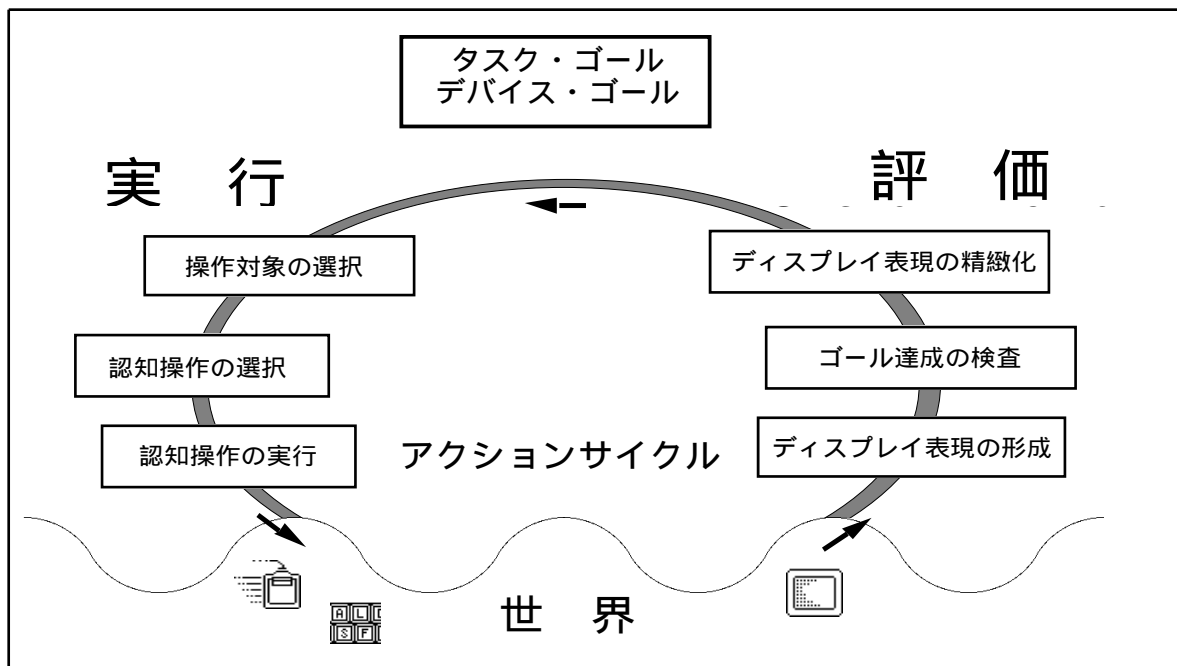


図4 ディスプレイ・ベースのHCIの認知プロセスモデル. 図3 に示した Norman のアクション・サイクル・モデルを基にして, 6つの過程でディスプレイ・ベースの HCI の認知プロセスをモデル化している.

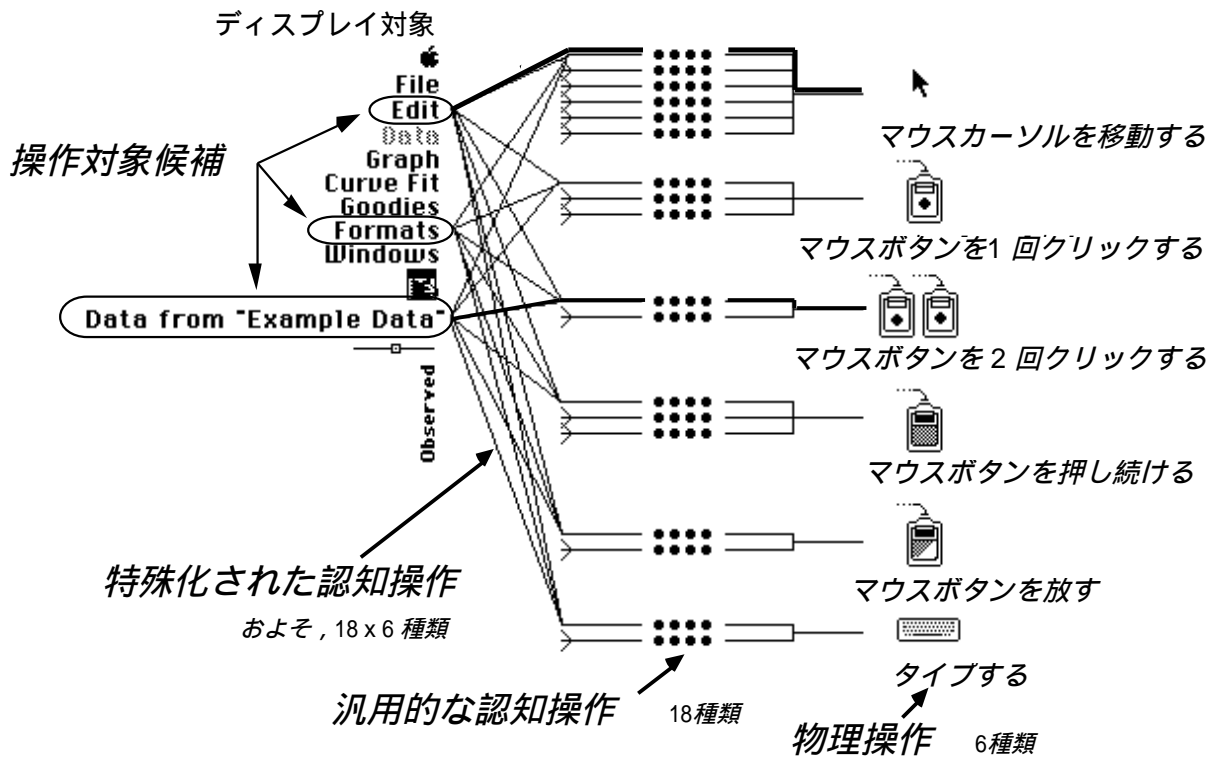


図5 実行のサイクル．左から右に操作選択がなされる．まず，操作を行なう対象がディスプレイされている対象のなかから選ばれ(左)，次に，それらの候補と，操作に対して与えられている汎用的な表現(汎用的な認知操作)とが結び付けられ，特殊化された認知操作が構成される(中)．個々の特殊化された認知操作のなかで最もその時の文脈において適切であり，しかも実行可能なものが選択され，実行される(右)．

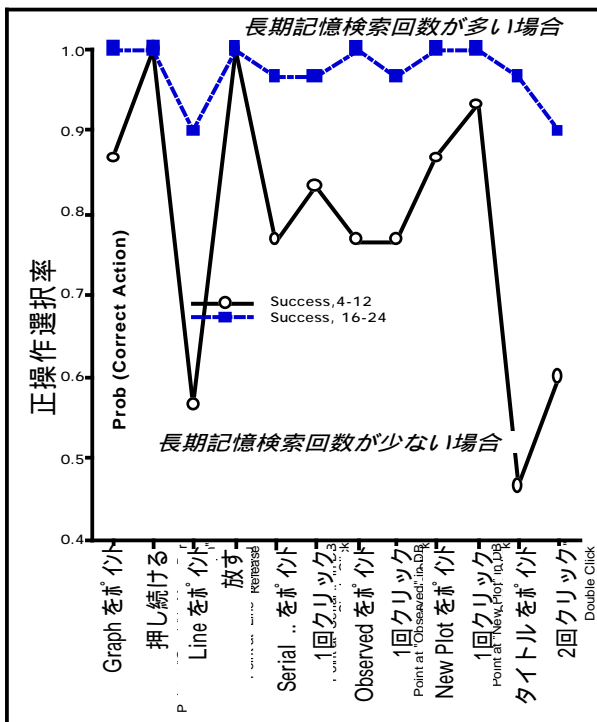


図6 Cricket Graph タスクのシミュレーション結果．長期記憶を検索する回数が少ないグループ(4、8、12)と、大きいグループ(16、20、24)にわけてプロットしてある．横軸は、ステップ、縦軸は、各記憶検索回数について10回シミュレーションを行なったときに正しい操作が選択される確率(正操作選択確率)を、それぞれのグループで平均した値である．