

# ユーザインタフェースの時相図と それによる対話の分析・検査法

守屋慎次, 中谷吉久

東京電機大学工学部電気通信工学科

本研究では, 初めに①利用者対話型システムの動きを可視化する方法を提案する。可視化に用いられる図をユーザインタフェース(U I)の時相図と呼ぶ。時相図は, 横軸に時間を刻み縦軸に操作対象を配した2次元空間上に, “フェッチと実行の木”を組み立てることによって描かれる。次に②時相図を用いて取消機能に関する3種類の一貫性を検査する方法を導く。

本研究の意義として, ①時相図が, 利用者自身と利用者が用いる文具や入出力機器までを含めた対話の模型として利用可能なこと, ②時相図を用いた分析により, 3種類の一貫性を自動検査する方法が初めて得られたこと, が挙げられる。

THE TIME-PHASE CHART AND THE METHODS FOR CHECKING  
THREE KINDS OF CONSISTENCY FOR USER INTERFACES

Shinji MORIYA and Yoshihisa NAKAYA

Department of Electrical Communication Engineering  
Tokyo Denki University  
2-2 Kanda-Nishiki-cho, Chiyoda-ku, Tokyo 101

The time-phase chart for user interfaces is proposed, giving how to draw it. The chart is drawn on the two-dimensional space where the horizontal axis represents a sequence of time when the user and the system take actions, and the vertical axis constitutes a hierarchy of objects in user interfaces. The chart traces of the fetch-execution cycles caused by a sequence of actions taken by the user and the system. The paper shows that firstly, the chart realizes a visual modeling of human-computer interaction, and secondly, it gives a way of checking three kinds of consistency in user interfaces.

## 1. はじめに

本研究では、まず①利用者対話型システムの動きを時間の流れの上で体系的に可視化する方法を提案する。可視化に用いられる図を、ユーザインタフェース(以後UIと略記)の時相図と呼ぶ。次いで、②時相図を用いて、UIの一貫性を検査する方法を導く。

対話の流れを記述する方法としては、誰もが用いる画面単位の遷移図、階層的な流れ図に似た図を用いる方式<sup>(9)</sup>、状態遷移図を用いる方式<sup>(10)(11)</sup>など<sup>(1)(3)(4)(5)(13)(14)</sup>がある。しかし、これらのいずれにも次のような問題がある。即ち、①利用者の動作や利用者が操作する用具・器具が適切に記述できない、②利用者に対するシステムの特性(例えば一貫性)が表現できない、③順序は表現できても時間の流れが明確に示されない、以上である。本論文の時相図は、対話の記述手段のうちで、①②③の表現を可能にした最初のものである。

UIを検査(あるいはデバッグ)することの必要性や重要性が多くの人によって語られている<sup>(3)(12)(13)(14)</sup>。特にShneiderman<sup>(3)</sup>は、メニュー項目とコマンド構造の一貫性について、実験結果に基づいた比較的具体的な提言を行っている。しかし、UIの検査を少しでも体系的に扱う試みはまだない。本論文の時相図はその最初の試みと考えている。

時相図は、横軸に時間を刻み、縦軸に操作対象を配した2次元空間に描かれた図である。第2章では、対話型システムの使い易さと一貫性について述べる。また市販の対話型システムにおける一貫性の欠如について具体例を挙げ、それが時間の進行・遡及に関係している点を指摘する。第3章では、対話における操作対象を分類し、その結果を用いて時相図の描き方と意味について述べる。第4章では、市販ワープロの取消機能を時相図を用いて分析し、三種類の一貫性の検査法を導く。

## 2. 使い易さと一貫性

本章の目的は、現在の対話型システムが抱えている問題点の一つとして一貫性の欠如を具体的に示し、その解決のための一つの方向を示すことである。

対話型システムの使い易さを代弁するあい昧な言葉として利用者との友好性(user friendliness)がある。文献(1)によると、友好性の要因は次の5つに集約される。①対話の柔軟さ(flexibility)、②透明度(transparency)、③利用の容易さ(ease of use)、④習熟の容易さ(ease of learning)、⑤信頼性(reliability)。ここでは、透明度をとりあげてみる。文献(2)によると、システムが透明であるとは、利用者が目的とする動きだけが見え、他の動きは利用者には見えないこと、と

している。

システムの動きを不透明にしている巨大要因は、構文と計算機の知識<sup>(3)</sup>の存在である。従って、可能な限り構文や計算機の知識を減らすことが透明度を高める上で重要である。しかし、現状は、構文も計算機の知識もなしでシステムを使うことは不可能である。従って、それらを学習し、記憶保持の必要性を減らすことが重要となる。これは、習熟容易性を高めることにも通ずる。

透明度と習熟の容易さの両者に最も影響を及ぼす設計要因の一つは一貫性(consistency)と考えられる。以下に一貫性の分類と一貫性に欠ける事例を挙げ、何を解決すべきかを検討する資料としたい。

対話型システム設計における4段階<sup>(4)</sup>の一貫性として次が考えられる。

(1) 構想の一貫性 (一貫性に欠ける例) 取消可能な項目の種類が一貫性に欠ける。例えば、範囲設定の際、始点は取消できるが終点はできない等。

(2) 意味の一貫性 (一貫性に欠ける例) undo実行後のカーソル位置がコマンドによって違う。例えば、複写コマンドのundo実行後カーソル位置はコピー先に、移動コマンドのundo実行後は始点にある。また、前述の始点取消の際、カーソル位置は変わらない等。

(3) 構文の一貫性 (一貫性に欠ける例) コマンドによって操作と操作対象の指定順序が違う。例えば、削除コマンドでは「削除」の指定は先頭であるが、罫線等の削除では「削除」の指定は最後である等。

(4) 語いの一貫性 (一貫性に欠ける例) 操作対象によって取消キーが違う。例えば、コマンドの取消はctrl+Uで、始点取消はBS(backspace)キーである。また、確定の取消はctrl+↓であり、文節の取消はESC(escape)キーである等。

前掲(1)(2)(4)の「一貫性に欠ける例」には共通の概念が存在する。いずれも時間の進行・遡及に関係している点である。進行・遡及时的要素(始点やカーソル位置)の扱い方や、進行・遡及を起動する方法に、一貫性が保たれていないことを指摘している。このような指摘は本論文が最初と筆者らは考えている。次章では主として時間の進行・遡及を体系的に扱うための道具について考えてゆく。

## 3. 時相図

本章では時相図の描き方とその意味を述べる。時相図は、横軸に時間を刻み、縦軸に操作対象を配した2次元空間に描かれた図である。目標は、UIの構成要素の変動を時間の流れの上で図式的かつ体系的に扱うことである。

### 3.1 操作対象の種類とその階層

ここでは縦軸の操作対象を説明する。操作対象には次の三種がある。

①人間による操作対象・・・人間が“直接”操作する対象。(例1)人間の手,指,頭。(例2)鍵盤,マウス。(例3)文書,操作手引書,紙,鉛筆。

②画面上の操作対象・・・画面上に殆ど常時表示されている操作対象で,①の操作対象を通じて操作されるもの。(例1)章,節,段,句,語,字。

(例2)漢字,平・片仮名,英数字。

(例3)ウィンドウ,メニュー,コマンド。

③保存されている操作対象・・・システムに保存される操作対象。(例)ファイル,バッファ,利用者やシステムが特定したカーソル位置(フラグなど)。

これらの操作対象のうち,特に②・③には,できる限り計算機の知識<sup>(3)</sup>(例えば,バッファ,ファイル,コード体系)を持ち込まないことが大切である。その対話型システムが目的としている仕事の知識<sup>(3)</sup>,換言すればその仕事分野における,概念の明かな言葉で操

作対象を構成するのが望ましい。これにより操作対象の透明度が高まり,学習と記憶保持が容易になると考えられる。

### 3.2 時相空間

対話型システムでは,「利用者による入力とシステムからの応答」の繰返しにより仕事が進められる。この入力・応答の列,すなわち時間の経過を横軸に,操作対象の種類とその階層を縦軸にとり,これをUIの時相空間と呼ぶ(図1)。

横軸には,人間とシステムの行動及びその時間経過が刻まれる。時間は連続値もしくは離散値である。

縦軸には,行動によって生じた操作対象の変動が記される。

各操作対象毎に横方向の時間軸が存在する。これら時間軸の内容・性質等が互いに異なるため,ある時点における縦軸断面には,異質な時間軸の「相」が写し出される。時相の名の所以である。

### 3.3 フェッチと実行

対話型システムでは入力と応答の繰返しによって仕事が進められる。換言すれば,次の2段階の繰返しによって仕事が進められる。

- ①フェッチサイクル・・・操作対象を用意する段階。
- ②実行サイクル・・・操作がなされる段階。

①,②共に一般的には時間の経過を伴う。

フェッチサイクルの例として次が考えられる。

- (イ)文書上の修正要求を人間が目で見取る,
- (ロ)マウスやカーソルを移動して目的点に到る,
- (ハ)文字をいくつか入力してオペランドを形成する,
- (ニ)キーボード上で指が目的とするキーに達する。

実行サイクルの例としては,(イ)置換キーを押す,(ロ)マウスのボタンを押す,などがある。

『フェッチと実行』という考え方はプログラム内蔵方式の基幹を成す概念であり,本稿における『フェッチと実行』の,その発想源もそこにある。

本稿の『フェッチと実行』とよい整合を見せる類いの概念として,Card<sup>(5)</sup>らのKeystroke-Level Modelがある。あるunit-taskを遂行するのに要する時間Tunit-taskは,そのunit-taskを遂行する方法を得るための所要時間Tacquireと,その方法を実行する上での所要時間Texecuteとの和から成る,というモデルである。即ち,

$T_{unit-task} = T_{acquire} + T_{execute}$   
と表現される<sup>(5)</sup>。acquireとexecuteをそれぞれフェッチと実行に読み替えれば,その形も質もよく類似していると言えよう。Keystroke-Level Modelと時相空間との関係が次の3.4節例3で述べられる。

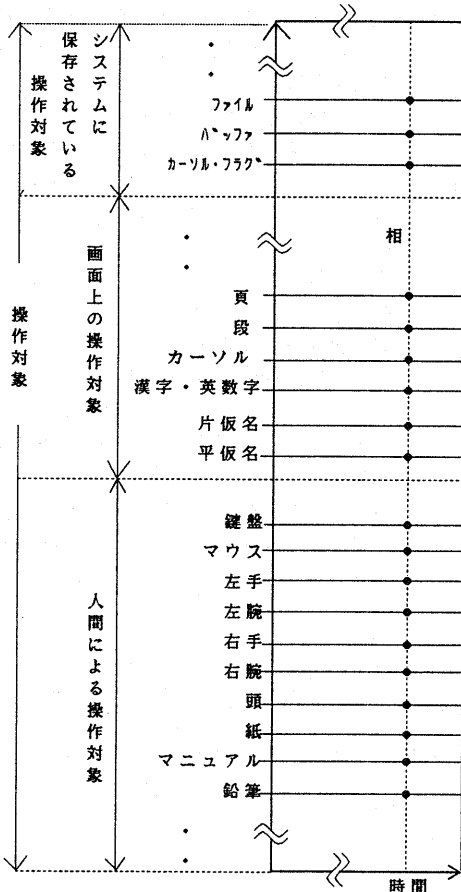


図1 ユーザインタフェースの時相空間

### 3.4 時相図

ここではいくつかの例を掲げながら、フェッチと実行のサイクルを時相空間上に位置づける方法を述べる。空間上に描かれた図をUIの時相図または時相流れ図と呼ぶ。

#### (例1) 鍵盤入力の時相図

最初の例として鍵盤からの入力文字に対する時相図を示す。図2(a)(b)(c)はいずれもローマ字“kirei”を入力して平仮名“きれい”へ自動変換する段階を示している。(a)はその詳細図、(b)(c)はこの順に、より簡略な図となっている。打鍵間隔は任意として描かれている。

(a)図には、鍵盤から入力された字が一時表示される時間軸が設けられている。時間軸FEはフェッチと実行の段階を示す一種の操作対象である。(a)図において、FE<sub>0</sub>の時間軸は「鍵盤文字の一時表示」軸と対になっている。FE<sub>0</sub>は入力された字を取り出し(フェッチし)、それを画面の適切な位置にシステムが表示する(実行する)という段階を示す。フェッチ(f)と実行(e)は入力文字毎に繰り返される。「網掛け」部はfとeが同一

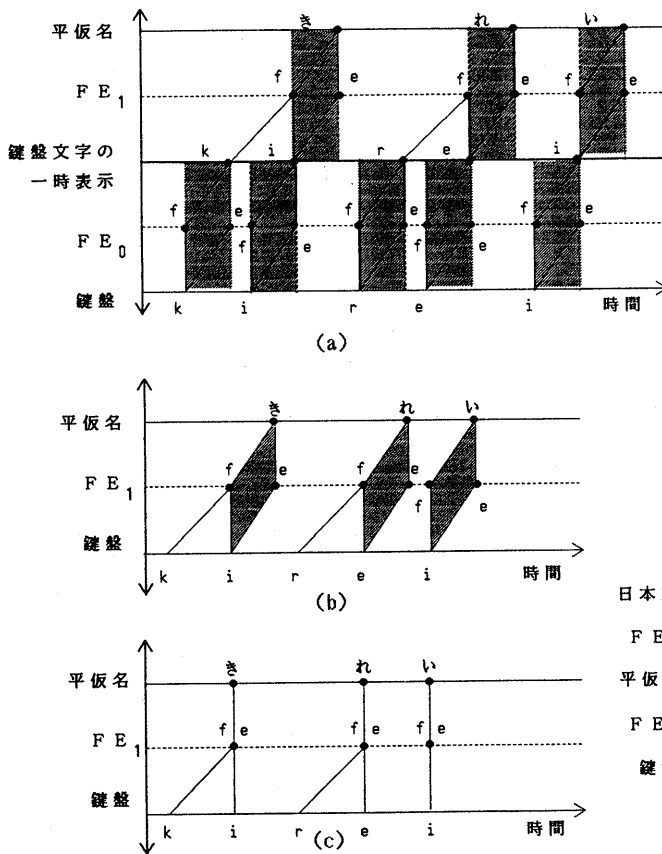


図2 鍵盤入力の時相図

キーによって生起することを示す。FE<sub>1</sub>の時間軸は、「平仮名」軸と対になっており、ローマ字を平仮名に自動変換する段階を示す。ローマ字は母音が出現する度に平仮名へ変換されると仮定している。

(b)図は、(a)図におけるFE<sub>0</sub>と『鍵盤文字の一時表示』の時間軸、及び網掛けの一部を略したものである。(c)図は、(b)図の網掛け部を省略したものである。以後は(b)または(c)図の記法を用いる。

#### (例2) かな漢字変換の時相図

図3を参照されたい。

#### (例3) 人間の操作対象と時相図

- CardらのKeystroke-Level Model<sup>(5)</sup> -

図4は、3.3節で述べた

Tunit-task = T acquire + T execute

におけるT executeの、より詳細な時相図を示している。

図4の例はCard<sup>(5)</sup>からとったもので、文書編集の一場面を示している。すなわち、画面上の英単語1語をマウスで指定し、それを鍵盤から入力された語で置換する例である。詳細な手順及び各段階における所要推定時間は次の通りである<sup>(5)</sup>。

- ①手を伸ばしマウスを掴む。 H[mouse]
- ②マウスを移動して目的とする語を指す。 P[word]
- ③マウスのyellowボタンを押し、その語を選ぶ。 K[yellow]
- ④手を鍵盤へ戻す。 H[keyboard]
- ⑤置換コマンド(R)を打つ。 MK[R]
- ⑥置換する語(5文字)を打つ。 5K[word]
- ⑦ESCを打ち置換を終了する。 MK[ESC]

Keystroke-Level modelによれば、①~⑦までの推定所要時間T executeは各推定所要時間の和で表される。即ち

$$T_{\text{execute}} = H + P + K + H + M + K + 5K + M + K$$

図4から本論文の時相図がこの関係式ともよく整合していることがわかる。

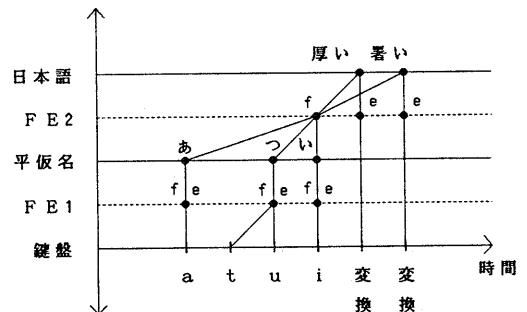


図3 かな漢字変換の時相図

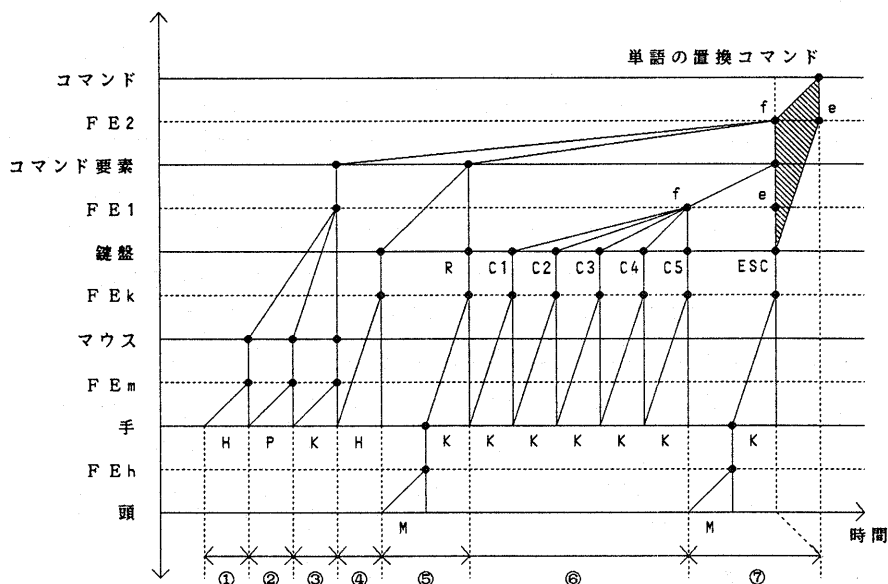


図4 Card<sup>(5)</sup>のKeystroke-Level Modelの時相図

### 3.5 その他のUI要素

時相図上で考慮すべきその他のUI要素として、①進行と遡及の機能、②情報の保存と回復の機能、③カーソル位置と階層、④属性、⑤モード、⑥メニューやウィンドウなど、多数が考えられる。紙数の都合上、②～⑥の検討は省略する。①だけを次章に述べる。

## 4. 取消機能

本章の目的は、従来の取消機能(undo/redoやbackspaceなど)の見直しと体系化を図り、利用者にとって有用な取消機能を提案することにある。

### 4.1 時相図における取消機能の表現

本節では市販のワードプロセッサ(「一太郎」)の取消機能を時相図上に表現し、一貫性という観点から考察を加える。

#### (1) 取消における三つの一貫性

図5と6を参照されたい。両図が行っている仕事は全く同じで、①“top down法の”と入力し、②入力した文字“の”から数えて右に3文字目から、3文字分を、2行下へ移動する、ということである。

図5と6上の矢円弧が取消の開始・終了位置を表現している。図5上の矢円弧は、市販システムで実現されているものだけを示す。図6上の矢円弧は、図上で可能なすべての取消方式を示している。図5と6の時間軸①の下に始・終点等の注釈が書かれている。時間軸⑥は取消時に生ずるカーソル位置の推移を示す。

図5の矢円弧のうち、イ、ロ、リ、ニは、第2章(1)(2)

(4)の「一貫性に欠ける例」を図示したものである。

(1)の例はロと、(2)の例はイ、ロと、(4)の例はイ、ロ、リ、ニと、それぞれ表現されている。図5の左上には、イ、ロ、リ、ニそれぞれのキー(矢円弧の種類)が示されている。

矢円弧1つが取消の1段を示している。図5上の矢円弧のうちBS(backspace)によるものは連続多段取消が可能である。他のキーによるものは1段だけの取消である。図6上の矢円弧は、連続取消可能な段数を指定していないが、以下の議論はこの段数と無関係である。

この二つの時相図から次のことが読みとれる。仮説として3つ掲げる。

#### 仮説C1. 取消可能な要素の一貫性

取消可能な要素の一貫性を保つには、同一時間軸上のすべての要素について取消可能にする必要がある(図6参照)。

#### 仮説C2. 取消におけるカーソル位置の一貫性

取消におけるカーソル位置の一貫性を保つには、「カーソル位置」(時間軸⑥)上の矢円弧は、対応する取消の矢円弧と「同期」させる必要がある(図6参照)。ここで同期とは次を意味する。図6上の縦の破線と時間軸との交点を、取消時点と呼ぶことにする。同一の縦破線上の取消時点は同一取消時点にある。このとき、異なる時間軸上に存在する矢円弧の対のうち、取消開始時点と取消終了時点が同一ものは同期しているという。



示した。ここでは、空白は文字であると同時に単語の区切りとしての意味も持たせてある。そして、空白の入力を単語入力契機としている。図6のようなシステムでは英数字の入出力に④も必要となることが分かる。)従って英(数)字文章の取消には1つのキー(例えばbackspace)で必要十分である。さらに、要素(この場合は英数の文字)、カーソル位置、操作キーの一貫性(仮説C1, C2, C3)も保たれる。

一方、ローマ字によって日本語を入力・表示するためには、最低5本の時間軸(①~⑤)を必要とする。(図2上の「鍵盤文字の一時表示」軸を含めるとすれば6時間軸が必要である。)図6を見ると、日本語入力時に、すべての取消矢円弧が同期している時間軸の対は、①~⑤のうちでは②と③だけである(このため②上の矢円弧は省略されている)。即ち、日本語入力時には最低4時間軸(①, ②=③, ④, ⑤)の「取消方式」が存在する。当然のことながら英文流のbackspaceだけでは、日本語の取消機能としては全く不十分である。

ここで市販システムの日本語入力時における取消方式を調べてみよう。図5の①上で“hou(変換)no(確定)”と入力されている部分に注目されたい。この市販システムでは4時間軸の取消方式が考慮されている。しかし、仮説C1の「取消可能な要素の一貫性」が時間軸①, ⑤上では保たれていない。C2の「取消におけるカーソル位置の一貫性」は保たれている(図5上に示されてはいない)。仮説C3の「取消を起動する語の一貫性」は、時間軸②③④上で保たれていない。

#### 4.2 取消機能の可能性と提言

本節では、本論文が扱っている時相図上の範囲内で、可能な全ての取消機能と、その実現に向けての提言を述べる。図6から次のことが言えよう。

- (1) 5時間軸において遡及が可能である。(鍵盤文字の一時表示の取消を含めれば6時間軸となる。)
- (2) 時間軸間の推移が可能である。即ち、各時間軸における同一取消時点において時間軸間を推移(時間軸間を渡り歩き)することができる。
- (3) 取消キーの設計に一貫性を持たせることができる。即ち、時間軸とキーとを対応させ、それらのキーと「取消」キーとの併用で各取消機能を実現すればキーの設計が一貫性を保つ。特に重視すべき時間軸は⑤, ④である。
- (4) 上記を考慮した精密な取消機能の有用な範囲は入力時の先端部(最近入力された部分)だけで充分であろう。有用範囲の大きさは、文献(6)の研究から、①2~3語(文節)以内、及び②1~2コマンド以内で充分であろう。

以上の提言により、4.1節で指摘した一貫性の欠如に対する解決の方法が示されたと考える。具体的な取消機能の決定は、全体機能の設計においてなされるべきである。上記(1)~(4)の具体的な議論については別に報告したい。

#### 4.3 時間軸の一貫性

本節では、時間軸の一貫性を定義する。

4.1節で3種類の一貫性(C1, C2, C3)が仮説として述べられた。この3つはいずれも1つの時間軸に関するものである。一方4.2節(2)で時間軸間の推移の存在が示された。そこでC1, C2, C3のような1時間軸の一貫性を複数の時間軸に拡張する必要が生ずる。次のように呼ぶことにする。

- (1) 一時軸一貫性・・・1つの時間軸について一貫性が保たれていること。
- (2) n時軸一貫性・・・n時間軸のそれぞれで同種類(例えばC1, C2, C3のいずれか)の一時軸一貫性が保たれていて、そのn時間軸間の同一取消時点において推移が可能なこと。

図5の英数字入力部分だけはC1, C2, C3のそれぞれについて5時軸一貫性が保たれている。図6は、各時間軸毎のカーソル位置が示されていないのでC2は除外して、C1, C3のそれぞれの7時軸(①~⑦)一貫性が保たれている。

#### 5. 利用分野

4章では時相図を取消機能の分析に利用し、取消機能が体系的かつ視覚的に考察可能なことを示した。またそれを通じて時軸の一貫性及びその検査法を新たに示した。

時相図の応用として次のものが考えられる。

- (1) 人間と計算機のインタラクションのモデルとして  
図1において「人間による操作対象」の範囲内は利用者のモデル(user's model)の一部を、「画面上の操作対象」と「保存されている操作対象」の範囲内は利用者モデル(user model)の一部を、それぞれ表現していると考えられる。
- (2) 対話の記述・分析・検査の手段として  
①4章のような取消機能だけでなく、他機能の分析・解析に利用可能と考えられる<sup>(7)</sup>。また岡留ら<sup>(8)</sup>やCardら<sup>(5)</sup>のモデルと併用して、UIの設計時に、所要時間や一貫性など<sup>(7)</sup>の予測が可能と考えられる。  
②多様な利用者層に対応する方法として、時間軸を活用することが考えられる。例えば、初心者には初めは少数の時間軸だけを見せ、慣れてくるに従って多数の時間軸も使用できるようにしておく、という使い方が考えられる。

## 6. おわりに

### 6. 1 まとめ

本論文で次のことを明らかにした

(1) 一貫性の分類方法として、システム設計の各段階(構想, 意味, 構文, 語い)による分類が可能であること, 及びその各段階での一貫性がシステムの透明度に大きく影響することを示した。

(2) 縦軸に操作対象をとり, 横軸に時間を刻み, 時相空間を定めた。時相空間にフェッチと実行という

“線図”を配し, 接続して組み立てることにより時相図を描く方法を述べた。これにより, 人, 文具, 入出力機器及びシステムによる仕事の時間的推移と操作対象の動的変動が体系的に可視化できること, またシステムの特性(例えば一貫性)についての分析・検査の方法が得られることを示した。

(3) 取消機能を時相空間に描く方法を示した。これにより体系的かつ視覚的に取消機能が分析可能なことを示し, 有用と思われるいくつかの提言をした。取消機能の一貫性に関する3つの仮説を述べ, 3つの一貫性の検査法を明らかにした。また時軸の一貫性を新たに提案した。本論文の方式によれば, これらの一貫性は, 人手だけでなく, 機械によっても検査可能である。

### 6. 2 検討課題

本研究にはたくさんの検討課題が残されている。ここではその主なものを述べる。

(1) ユーザインタフェースの構成要素は多種多様である。利用者, 対話型システムの種類, 行う仕事, 扱う情報の種類と内容, 入出力の機器と技法(例えば入力モード。本論文では要求入力のみを扱っている。), 操作や属性・モードの種類, など, それぞれが多様である。本論文ではその一端について述べたに過ぎない。

(2) 時相図の記述方法と記述能力がまだ確かではない。その主な原因として, ①上記の多種多様性, ②画面上の操作対象のような多次元空間の表現が容易ではないこと, ③本論文では具体的な入力について図を描いたが, 一般形の記述を試みていないこと, ④時相図の分岐や繰返し等については言及していないこと, などがある。

### 謝辞

日頃, 御指導・御助言いただく本学 穂坂 衛教授, 平松啓二教授, 御討論いただいた本学 齊藤 剛氏と本集中研究集会の参加者諸氏, 御助力いただいた卒研生諸氏に感謝の意を表する。

### 参考文献

- (1) W. Dehning他, Lecture Notes in Computer Science 110, The Adaptation of Virtual Man-Computer Interfaces to User Requirements in Dialogs, Springer-Verlag (1980).
- (2) 守屋, 人間・機械インタフェース, 情報信号処理(中田, 南, 平松編)第6章 オーム社(1986).
- (3) B. Shneiderman, Designing the User Interface, Addison-Wesley (1986).
- (4) J. D. Foley他, Fundamentals of Interactive Computer Graphics, Addison-Wesley (1982).
- (5) S. Card他, The Psychology of Human-Computer Interaction, Lawrence Erlbaum Associates (1983).
- (6) 倉橋, 龍, 芝野, 村岡, エディタ設計のための利用者の振舞の解析, 信学誌 '87/6 vol. J70-D, No.6, pp.1248-1251.
- (7) 守屋, ユーザインタフェースの時間構造, 計測自動制御学会, 第3回ヒューマンインタフェース・シンポジウム論文集, 1987年10月.
- (8) 岡留, 小野, 山田, 日本語タッチタイプ時の情報処理の認知的側面, 計測自動制御学会, 第1回ヒューマン・インタフェースシンポジウム論文集, 1985年10月.
- (9) G. V. Kloster, A. Zellweger, Engineering the Man-Machine Interface for Air Traffic Control, IEEE Computer, Feb. 1987, pp.47-62.
- (10) D. Kieras, P. G. Polson, An approach to the formal analysis of user complexity, International Journal of Man-Machine Studies 22, (1985), pp.365-394.
- (11) R. J. K. Jacob, Using Formal Specification in the Design of a Human-Computer Interface, Proceedings of Human Factors in Computing Systems, March 1982.
- (12) S. W. Draper, D. A. Norman, Software Engineering for User Interfaces, IEEE Transactions on Software Engineering, Vol. SE-11, No.3 March 1985, pp.252-258.
- (13) J. D. Foley, Managing the Design of User Computer Interfaces, CHI'86, Human Factors in Computing Systems, Tutorial 15, April 1986.
- (14) D. Mayhew, Methods in User Interface Design: A Practical Course for Software Managers and Developers, CHI+GI 1987, Human Factors in Computing Systems and Graphics Interface, Tutorial 3, April 1987.



## 討 論

### 8. ユーザインタフェースの時相図とそれによる対話の分析・検査法

守屋（電機大）

鎌田：時相図が時間的に変化する様子をとって解析すれば、ユーザの習熟度あるいは操作性の評価ができるのではないのでしょうか。

守屋：そうですね。この図から何か知見が得られれば役に立つと思います。

川越：非常におもしろい話ですね。この時相図は分析にはたぶん有効だと思いますが、設計にはどのように使うのですか。

守屋：これはまだお話だけなのでよく分かりませんが、1つは構成要素の操作対象が構造を持っているので、それがUIMSなどの構造と対応があるかもしれない、もう1つはこの図によりユーザの動きを大きく捉えて、トップダウンの設計ができるかもしれない。

坂下：時相図を機械的にうまくサンプリングする試みはありますか。

守屋：考えてはいるのですが、まだ試みていません。

坂下：図を書くのは大変そうですね。

守屋：書くのは大変です。（笑い）

専用の用紙が用意されていれば、まあ許容範囲内だと思います。

川合：他のワープロについてはどうですか。木村泉先生が割合細かい分析をなさってましたが。

守屋：ここで示したのは一太郎の例で、IBM5550も試しました。木村先生の分析は主に入力に対してのユーザのモデルの方だと思います。