

## 部 人間工学応用の方法論・技法と支援技術

## 2 認知モデルとシミュレーション

本章では、知覚・認知・運動を含む人間の行動過程の予測を行うことを目的として研究が進められている認知モデリング (cognitive modeling) について説明する。ここで、予測の対象とされるのは、実行時間、操作学習時間、エラーなどである。これらに関する予測は、認知モデリングによって開発される工学モデル (engineering models) に基づいて、人間が製品を利用する過程をシミュレートすることによって行われる。

以下の節では、まず、工学モデルの考え方について述べ、続いて、既にさまざまなインタフェース設計に利用されている工学モデル「GOMS モデル」について説明を行う。また、GOMS モデルでは扱われない非ルーチ的なタスク、問題解決を含むようなタスクをユーザが遂行する過程の工学モデルである探査学習 (learning by exploration) のモデルについても簡単に説明する。

## 2.1 工学モデル

工学モデルは、知覚・認知・運動過程の近似モデルであり、以下の特徴を持つ (John & Kieras, 1996a)。

1. 近似の程度を予測の目的に応じて適当に設定することができる。
2. 心理学の教育を受けていないデザイナーでも簡単に使える。
3. 幅広いタスクを対象とすることができる。

工学モデルに基づいた行動予測は次のような状況において有効である。まず、製品開発の初期の段階において人間の行動の定量的な予測を行うことにより、プロトタイプを作成を待たずに製品の評価を行うことができる。また、新製品を導入しようとしている場合に、仮にそのような場合を想定して人間がそれを利用する過程のシミュレーションを行って旧製品との比較検討を行うことができる。

工学モデルを利用する際には次の点に注意する必要がある。まず、工学モデルはあくまでも近似モデルであるので、工学モデルを完全にユーザテストに代わるものと考えべきではないということである。むしろ、ユーザテストの量を減らすためのものと考えるのが適当である。また、タスクのなかには、もともと工学モデルを構成するこ

と自体が適当でないものもあるということである。創造的なタスクや発見を含むタスクなどがそのようなタスクの例である。これらのタスクには、推論・発見などの高次の思考過程が含まれ、そこから生じる行動を定量的に予測することは現状では不可能である。

## 2.2 GOMS (ゴムズ) モデル

GOMS モデルは、ゴール指向のルーチ的なタスクを対象とした工学モデルであり「GOMS 基本概念」(2.2.1)に基づいて構成されるモデル群の総称である。GOMS モデルは Card, Moran, & Newell (1983)が彼らの著書、The Psychology of Human-Computer Interaction、においてその考え方を示して以来、ヒューマンコンピュータインタラクションにおける理論的な基盤を提供する考え方のひとつとして支持されてきている。

現在、KLM、GOMS、NGOMSL、CPM-GOMS の4種類のGOMSタスク分析・モデリング技法が開発され、実用的に利用されている。それぞれの技法では、タスクの遂行を制御する認知アーキテクチャ (cognitive architecture) として異なったものが仮定されているので、GOMS 技法の利用者 (タスク分析を行い予測を行う人) はタスクのタイプや予測の目的に適したものを選択する必要がある。本節では、各GOMSモデリング技法について例を用いながら概略を説明する。詳細については、各GOMSモデリング技法に関する代表的な文献を図1に示しているのので、そちらを参照してほしい。

## 2.2.1 一般 GOMS 概念

Card, Moran, & Newell (1983)のGOMSモデルの提案以来、多くのGOMSモデルが構成されてきた。これらのGOMSモデルに共通する考え方を John & Kieras(1996a)は一般GOMS概念 (the general GOMS concept) とよび、次のように定義している。

タスクの遂行の仕方に関する知識を、ゴール、オペレータ、メソッド、選択規則によって分析することは有効である

以下に、一般GOMS概念の各要素について説明する。

## a. ゴール (Goals)

ゴールは「ユーザが達成したいこと」を意味する。これは通常用いられている意味と同様である。例えば「論文を共同で執筆する」というタスクを考

えてみよう。この場合には、最高位のゴールは「論文を書くこと」である。このゴールは、さらにいくつかのサブゴールに分解される。

1. 最新のドラフトを共著者に送る
2. 参考書目をフォーマットする
3. (コメントに従って)文書を編集する

ここで、上位のゴールが達成されるためには、それに属する全てのサブゴールが達成されることが必要である。

サブゴールへの分解は分析の目的に応じて適当なレベルまで繰り返される。例えば、コマンドレベルや打鍵レベルの分析を行う場合には、サブゴール「文書を編集する」はさらに以下のように分解される。

1. テキストを移動する
2. 単語を削除する
3. 単語を挿入する

#### b. オペレータ (Operators)

オペレータは、ゴールを達成するために実行される知覚、認知、運動、またはこれらの複合アクションであり、以下に示す特徴を有する。

1. オペレータの実行によりユーザの心的状態や、外部環境の状態などに何らかの変化が生じる
2. オペレータを特徴づけるパラメタ(実行時間など)は、ユーザやシステムが現在の状態に至った経緯(オペレータの履歴)の影響を受けない。

例えば、「単語を削除する」というサブゴールは以下のオペレータにより達成される。

1. マウスを移動する
2. マウスボタンをクリックする
3. シフトキーを押しながらマウスボタンをクリックする
4. 削除キーを押す

#### c. メソッド (Methods)

メソッドは、ゴールを達成するために実行されるサブゴールとオペレータの系列である。これらのオペレータ系列は、十分に学習されたものであり、従って、その実行は自動的である。ゴールが階層構造を持つ場合には、メソッドもそれに対応した階層構造をなす。

例えば「単語を削除する」というサブゴールを達成するためのメソッドは、以下のようなオペレータ系列によって表される。

#### M1: マーク-削除メソッド

1. マウスを(単語の最初に)移動する
2. マウスボタンをクリックする
3. マウスを(単語の最後に)移動する
4. シフトキーを押しながらマウスボタンをクリックする
5. 削除キーを押す

#### d. 選択規則 (Selection Rules)

選択規則は、あるゴールを達成するためのメソッドが複数ある場合に、その中の一つを選択するために適用される知識である。例えば「単語を削除する」というサブゴールは、上記のマーク-削除メソッドの他に、以下のメソッドによっても達成できる。

#### M2: 削除キーメソッド

1. マウスを(単語の最後に)移動する
2. マウスボタンをクリックする
3. 削除キーを(単語長分)押す

このとき、M1、M2のどちらを選択するかを決める規則は、例えば、次のように表現される。

#### S1: 「単語を削除する」に関する選択規則

単語の長さが5文字以上ならばマーク-削除メソッドを、それ以外の場合は削除キーメソッドを用いる

この例の場合、単語長が8文字なのでマーク-削除メソッドが選択される。ただし、選択規則は個々のユーザに固有なものであるため、適用の判断基準となるパラメタの値(この場合は単語長)はユーザ毎に異なっても構わない。しかし、ひとりのユーザの内部ではタスクの状況に拠らない一定の基準となっていることが必要である。

#### e. ゴールとオペレータの関係

a)においては「単語を削除する」をサブゴールの例として説明した。しかしながら、分析の目的によってはこれをオペレータとして取り扱う場合もある。この場合には「単語を削除する」はマウス移動などのより詳細なレベルにまでは分解されない。このようにGOMSモデルによる分析に

においては、ある行動をゴールとして扱うか、オペレータとして扱うかは、分析者の選択する分析のレベルに依存して相対的に決められる。ある記述をゴールとして取り扱いたいならば、それを達成するためのメソッドをより下位のレベルのオペレータを用いて定義しなければならない。オペレータとして取り扱いたいならば、それ以上分解する必要はない。

しかしながら、全く任意に分析のレベルを設定できるという訳ではない。オペレータとしては、以下の2つの条件を満足しているものを採用することが必要である。

1. オペレータの実行時間は履歴に依存しないこと
2. オペレータの実行時間の評価を既存のデータを用いて行うことができること

これらの制約条件により、GOMS モデルによる分析は、通常、コマンドレベル、あるいは打鍵レベルで行われている。

### 2.2.2 GOMS モデルの形式

GOMS モデルを記述する方法として、プログラム形式と系列形式がある(John & Kieras, 1996b)。

#### a. プログラム形式

プログラム形式の GOMS モデルにおいては、メソッドに任意のタスクパラメータを持たせることができる。具体的にタスクが与えられたときに、その値をタスクの状況に合わせて設定することによってタスクを誤りなく実行させることができる。例えば、マーク-削除メソッドは、タスクパラメータとして、削除されるべき単語の開始位置と終了位置を持つ。それらの値をタスクに応じて設定することによって具体的に与えられるタスクを遂行させることができる。

個々のメソッドには一定の表現が与えられる。しかしながら、個々のタスクを遂行するときのメソッドの実行順序は、与えられたタスクに依存して決まってくる。従って、プログラム形式のモデルにおいては、どのようなタスクの状況のときにどのメソッドを適用するかということを明示的に条件付き分岐によって表現することが必要となる。このようにしてモデルを定義することにより、想定し得る全てのタスクを実行させられる。

プログラム形式のモデルの長所は、分析者が全ての手続き的知識を把握できる点にある。また、多

くのタスクの具体例を解析する必要がある場合には、生成的な特徴を利用して解析したいタスクを即座に生成させられる。

一方、プログラム形式のモデルの短所は、あるタスクを遂行するときに行われるオペレータ系列を効率よく得ることができない点である。その唯一の方法は、モデルをタスクパラメータを用いて実際に実行してどのメソッドが使われるかを記録することである。もう一つの短所は、完全かつ正確なプログラム形式の GOMS モデルを定義するためには多くの時間が必要とされることである。以下 2.2.4, 2.2.5 に、プログラム形式の GOMS モデルの例を述べる。

#### b. 系列形式

系列形式の GOMS モデルにおいては、タスクは分析者が定めたメソッドの系列によって実行される。ここで、メソッドは一定のオペレータ系列によって表現されている。従って、任意のタスクが実行されるのではなく、あらかじめ想定されたタスクが決められたオペレータ系列によって実行される。

系列形式のモデルの長所は、オペレータ系列の把握が容易であること、また、表現の簡便さにある。分析者はプログラムのように細部にわたってすべての可能なタスクに対して手続き的知識を明示的に定義する必要がない。

系列形式のモデルの短所は、メソッドの階層構造を表現できない点である。メソッドの階層構造が実行時間に影響しているような場合に、それを知ることができない。また、普通は、オペレータ系列を枚挙するのは容易であるが、具体的なタスクの例の数が増えるに従ってそれに要する時間が増大する。以下、2.2.3, 2.2.6 に系列形式の GOMS モデルの例を述べる。

### 2.2.3 KLM (The Keystroke-Level Model, 打鍵レベルモデル)

KLM は、GOMS 技法の中でも最も単純な系列形式のモデルである(Card, Moran, & Newell, 1980, 1983)。タスク実行時間の予測は、タスクを遂行するためのオペレータ系列を明らかにして、個々の実行時間を合計して行われる。ここで、オペレータは打鍵レベルのもののみが用いられる。

図 2は Card, Moran, & Newell が 1980 年に発表した KLM に含まれてる 6 種類のオペレータとその

実行時間を示している。

KLM 技法では、図 3 に示した経験則に従って、' 心的準備 ' オペレータを他の物理的なオペレータの間に挿入してメソッドを定義する。

#### a. 認知過程に関する仮定

KLM では、人間の情報処理モデルとして系列ステージモデルが仮定されている。即ち、人間のアクティビティが順々に系列的に実行されてタスクが完了するというものである。また、人間の情報処理アクティビティは全てオペレータに含まれていると仮定される。このことにより、KLM を適用できるタスクに制約が課せられることになる。KLM で近似することができるタスクは、同時進行するアクティビティがなく、割り込みがなく、いくつかのゴールが交互に実行されないようなものに限定される。ひとりのユーザによって遂行されるコンピュータタスクの多くはこれらの制約を満足するので KLM で近似することができる。

#### b. 例

マーク-削除メソッドの実行時間の予測例を図 4 に示す。なお、オペレータ実行時間は図 2 から引用したが、マウスボタンクリックに要する時間は 20 秒としている。

### 2.2.4 GOMS

ここでは、Card, Moran, & Newell (1983) の第 5 章に示されている形式の GOMS モデルを説明する。GOMS モデルにおいては、ゴールは完全な階層構造を持ち、メソッドはサブメソッドと条件を含むプログラム形式によって表現される。GOMS モデルを構成すれば、具体的なタスクの状況を与えることにより、オペレータ系列と実行時間を予測することができる。

#### a. 認知過程に関する仮定

人間の情報処理モデルとしては、KLM と同様に系列ステージモデルが仮定されている。また、プログラム形式で表現するための仮説として 2.2.7 において説明するモデルヒューマンプロセッサの 10 項目ある動作原理のうち、合理性原理 (原理 8) と問題空間原理 (原理 9) が採用されている。これらはどちらも人間の問題解決の理論に由来する。問題空間原理では、ユーザのアクティビティは、初期状態をゴール状態へと変換するためにアクション (オペレータ) 系列を適用するアクテ

ィビティとして特徴づけられる。経験を積むことによりゴールを達成するためのオペレータ系列をその度に導き出すことは必要なくなり、系列 (メソッド) を思い出して実行することができるようになる。また、合理性原理は、人間は ' タスク環境の構造 (システムデザイン) ' と ' (人間の) 情報処理能力とその限界 ' が与えられたとき、最も効率的なメソッドを使う、とするものである。即ち、人間はゴールを達成するためにメソッドを実行するようになり、効率的であろうとするが故に、どのメソッドを実行するかはシステムデザインによって概ね決定される。従って、GOMS モデルを、タスクとシステムデザインに基づいて構成することにより、実行時間、オペレータ系列の予測を行うことができる。

#### b. 例

GOMS モデルによる「文書を編集する」というタスクのマーク-削除メソッドの実行時間の予測例を図 5 に示す。

### 2.2.5 NGOMSL (Natural GOMS Language ; エヌゴムズル)

NGOMSL は構造化された自然言語による GOMS モデル構成技法である (Kieras, 1988)。NGOMSL モデルはプログラム形式で表現され、オペレータ系列、実行時間、メソッド学習時間に関する予測を提供する。

#### a. 認知過程に関する仮定

NGOMSL 技法では、メソッドはプロダクションルールによって表現される。プロダクションルールの記述や発火は、認知複雑性理論 (CCT, Cognitive Complexity Theory, Bovair, Kieras, & Polson, 1990 ; Kieras & Polson, 1985) と呼ばれる認知アーキテクチャに基づいて行われる。CCT も単純な系列ステージモデルのひとつである。プロダクションルールは、作業記憶内の項目との照合が取れた場合に発火し、作業記憶の内容の更新といった内的オペレータや、マウスボタンをクリックするといった物理的オペレータを実行する。また、プロダクションルールは一定間隔で発火する (~0.1 秒)。

NGOMSL においては、手続き的知識であるメソッドの学習はメソッドに含まれるプロダクションルールを学習することによってなされる。また、一度学習されたプロダクションルールは新しいメソッドの学習にトランスファーする。即ち、新

しいメソッドの学習は、それに含まれるプロダクションルールの中でまだ学習の済んでいないものだけを対象に行われる。CCT は実行時間、学習時間、手続き的知識のトランスファーに関してよい予測を与えることが示されている (Bovair, Kieras, & Polson, 1988)。

NGOMSL 技法においては、オペレータとして、ユーザにとって既知であり学習の必要がないようなものを用いることが必要である。従って、ユーザが新規のオペレータを使わなければならないような新しいインタフェースを用いたタスク実行の予測はできない。また、NGOMSL は CCT の仮定している単純な系列ステージモデルに基づいているので、階層的で系列的なメソッド以外には適用できない。

### b. 例

NGOMSL モデルによる「文書を編集する」というタスクのマーク-削除メソッドの実行時間の予測例を図 6 に示す。各 NGOMSL の記述の実行時間は 0.1 秒としてある。また、編集結果の確認時間を 1.2 秒、位置の決定を 1.2 秒としている。

NGOMSL では学習時間の予測を以下のような式に基づいて行う。

$$\begin{aligned} \text{学習時間} = & \\ & 17 \text{ 秒} \times \text{NGOMSL 記述数} + \\ & 6 \text{ 秒} \times \text{LTM チャンク数} \end{aligned}$$

ここで、第 1 項は NGOMSL メソッド (手続き的知識) の学習時間であり、NGOMSL メソッドに含まれる記述数 (ステップ数) に比例する。第 2 項は長期記憶に記憶すべき宣言的知識 (長期記憶チャンク; LTM chunk) の学習時間である。長期記憶チャンクの数え方はまだ確定したものがあるわけではないが、その目安として、例えば、コマンド“CUT”に対しては 3 個の長期記憶チャンクを記憶することが必要になる。

1. テキストを CUT するという概念( $C_1$ )
2. “CUT”というコマンド( $C_2$ )
3.  $C_1$  と  $C_2$  が結合しているという事実

の 3 項目である。

### 2.2.6 CPM-GOMS

CPM は、認知-知覚-運動 (Cognitive-Perceptual-Motor) レベルの解析、並びに、限界経路法 (Critical-Path Method) の両方を意味する。オペレ

ータとしては、単純な知覚、認知、運動レベルのものが用いられる。他の GOMS モデルと異なり、オペレータが並列に実行される場合も取り扱える。CPM-GOMS ではスケジューリングチャート (パートチャート, PERT chart) を用いて、オペレータとそれら相互の関連が表現される。

#### a. 認知過程に関する仮定

CPM-GOMS では、認知アーキテクチャとしてモデルヒューマンプロセッサ (MHP; 2.2.7 参照) が仮定されている。タスク分析は MHP の 3 種類のプロセッサ (知覚プロセッサ, 認知プロセッサ, 運動プロセッサ) によって実行されるオペレータのレベルで行われる。MHP では、これらのプロセッサは独立に動作するので、その上で動くオペレータの並列実行が可能になる。一方、オペレータの表現のレベルは、他の GOMS モデルに比べて格段に詳細なものとなる。これは、オペレータが MHP のプロセッサのサイクル時間 (図 7 参照) のオーダーで実行されるので、オペレータをそのレベルで表現しなければならないからである。

CPM-GOMS では、並列に実行されるオペレータの同期を、全体のタスク実行時間が最短になるように取る。従って、完全に最適化された専門家ユーザのパフォーマンスがモデル化されていることになる。

#### b. 例

CPM-GOMS モデルを構成するためには、まず最初に、通常の GOMS モデルを構成する。そしてさらに分析を進め、MHP レベルのオペレータにまで展開する。

図 8 は、タスクを MHP オペレータ系列によって表現するために用いられるテンプレートである。タスクの範囲が限定されれば、考慮しなければならないオペレータの種類が決まってくる。また、オペレータの実行順序も典型的なものがいくつかある。これらのテンプレートには、典型的なオペレータの実行パターンが表現されているので、タスクの状況に応じてパラメータ値を選択してタスク分析を行うことができる。

テンプレートを用いて系列的にオペレータを結合した後は、並列処理が可能な個所を探して、オペレータ系列の最適化を行う。最長の経路を与えるオペレータ系列が限界経路 (critical path) となり、この系路上のオペレータ実行時間を合計すれ

ばタスク実行時間が得られる。

図 6 にマーク-削除メソッドの分析結果を示す。限界経路がマウス操作を含む経路となっている。これは、マウスの移動に時間がかかるためである。このとき、同時に実行される他のオペレータ、眼球運動オペレータ（眼球運動を実行する）、視覚オペレータ（情報を知覚する）、認知オペレータ（情報を確認する）、左手運動オペレータは、限界経路上のオペレータ（カーソル移動を実行する）の終了を待たされることになる。

### 2.2.7 モデルヒューマンプロセッサ

本節では、Card, Moran, & Newell (1983) の GOMS モデル、および、CPM-GOMS モデルに関連の深いモデルヒューマンプロセッサについて説明する。モデルヒューマンプロセッサは単純な人間の並列情報処理のモデルであり、人間をいくつかのプロセッサ（知覚プロセッサ、認知プロセッサ、運動プロセッサ）と記憶システム（長期記憶、作業記憶）によってモデル化する。知覚プロセッサは、感覚情報（聴覚、視覚）を取得・認識し、その結果を作業記憶に置く。認知プロセッサは、その情報に対して作用し、運動プロセッサに物理的なアクションを実行することを命令する。各々のプロセッサはその内部では固有のサイクル時間で系列的に処理を行うが、他のプロセッサとは並列に動作する。図 7 に各プロセッサの平均サイクル時間とその範囲を示す。

以下に、Card, Moran, & Newell (1983) に述べられているモデルヒューマンプロセッサの動作原理、10 項目を示す。なお、Card, Moran, Newell (1983) の GOMS モデルは動作原理 8, 9 に基礎を置いた系列処理モデルである。また、CPM-GOMS モデルは MHP に直接に基づいた並列処理モデルである。

#### 原理 0 ・ 認知プロセッサの認識-動作サイクル (Recognize-Act Cycle of Cognitive Processor)

認知プロセッサの各サイクルにおいては、作業記憶の内容に照合する長期記憶内のアクションが実行される。アクションの実行により作業記憶の内容は更新される。

#### 原理 1 ・ 知覚プロセッササイクル時間の変動原理 (Variable Perceptual Processor Rate Principle)

知覚プロセッサのサイクル時間<sub>p</sub> は刺激の強度に反比例する。

#### 原理 2 ・ 個別符号化原理 (Encoding Specificity Principle)

長期記憶に記憶される内容は知覚された内容（何が知覚されたか）によって決定される。また、記憶されたものをアクセスする場合にどのような検索手がかりが有効となるかは記憶されている内容によって決定される。

#### 原理 3 ・ 識別原理 (Discrimination Principle)

記憶検索の難易度は、検索の手がかりに応じて決定される候補に依存する。

#### 原理 4 ・ 認知プロセッササイクル時間の変動原理 (Variable Cognitive Processor Rate Principle)

知覚プロセッサのサイクル時間<sub>c</sub> は、タスク要求または情報負荷の増加に伴ってより多くの努力が誘導されるとき、より短くなる。また、練習により減少する。

#### 原理 5 ・ Fitts の法則 (Fitts's Law)

距離  $D$  だけ離れたところにある大きさ  $S$  の標的に手を移動するのに要する時間  $T_{pos}$  は、

$$T_{pos} = I_M \log_2(D/S + 5)$$

で与えられる。ここで、

$$I_M = 100 [70 \sim 120] \text{ msec/bit}$$

である。

#### 原理 6 ・ 練習の指数法則 (Power Law of Practice)

あるタスクを  $n$  回試行したとき、その実行時間  $T_n$  は次の指数法則によって与えられる。

$$T_n = T_1 n^{-\alpha}$$

ここで、 $\alpha = 0.4 [0.2 \sim 0.6]$  である。

#### 原理 7 ・ 不確定性原理 (Uncertainty Principle)

決定時間  $T$  は判断や決定に関わる不確定性ととも増加する。

$$T = I_c H$$

ここで、 $H$  は情報論的エントロピー

$$I_c = 150 [0 \sim 157] \text{ msec/bit}$$

である。 $n$  通りの選択があり、等確率である場合には、

$$H = \log_2(n+1)$$

である(Hick の法則)。異なった確率  $p_i$  を持つ場

合には、

$$H = -\sum p_i \log_2(1/p_i + 1)$$

となる。

#### 原理 8 ・ 合理性原理 (Rationality Principle)

タスクの構造，タスク実行者に対する情報入力，および，知識の範囲と情報処理能力の限界が与えられたとき，その実行者は，合理的なアクションによってゴールを達成するように行動する。

ゴール + タスク + オペレータ + 入力 + 知識  
+ 処理限界 行動

#### 原理 9 ・ 問題空間原理 (Problem Space Principle)

人間が問題を解決するために行う合理的なアクティビティは次の項目によって記述される。(1)知識の状態，(2)状態を更新するためのオペレータ，(3)オペレータを適用する際の制約条件，(4)次に適用すべきオペレータを決定するための制御知識。

### 2.3 探査学習の理論

本節では，探査学習の理論 (Learning by Exploration, Polson & Lewis, 1990) について説明する。探査学習の理論が取り扱うのは，GOMSモデルとは異なったヒューマンコンピュータインタラクションの側面，即ち，初心者ユーザが初めて遭遇する場面で問題解決をしながらゴールを達成するために操作選択を行うという場面である。例えば，使ったことがない機器を操作せざるをえない場合，見たことがない券売機で切符を購入しなければならない場合がこれに相当する。このようなとき，ユーザは，望む結果が得られるような操作手順が見つかるまでさまざまな方法を試すことになる。

#### 2.3.1 探査学習の理論に基づくユーザの認知モデル

探査学習の理論では，このようなユーザの行動を以下の4つの認知的なアクティビティによってモデル化する：

0. ゴール設定：ユーザはまずタスクに関する大まかな記述を行う。ここでは，何を達成したいのかが表現される。
1. 探査：ユーザはタスク遂行に有効なアクションを発見するためにシステムのインタフ

ェースを探索する。

2. 選択：ユーザはタスクを遂行できると考えられるアクションを選択する。選択の基準としてしばしば用いられるのは，ゴールの記述 (G) と，インタフェースにおけるアクションの記述 (対象 X に操作 Y を行う) の適合度に関する評価である。適合度の評価は，例えば，G と X もしくは Y のラベルの類似度などで測られる。
3. 評価：ユーザはタスクが進展しているかどうかを評価する。評価は，システムの応答を理解して，実行したアクションが正しい選択であったかどうかを判断することによって行われる。また，次に行おうとしているアクションの手がかりが得られるかどうかによっても行われる。

ユーザはタスクを遂行するためにこれらのステップを系列的に実行するものとして，ユーザ行動のモデル化が行われる。ユーザはまず始めに自分自身でゴールを設定し，それに続いて，探査，選択，評価のステップをゴールが達成されるまで繰り返す。

ここで，探査学習の理論によるモデル化の例として，ATM (Automatic Teller Machine；現金自動取扱装置) を利用する銀行の取引客の行動を考えてみよう。客は ATM から指示「金額を千円の倍数で入力してください」に従って，「20」を入力する。ここで**ゴール**は2万円を引き出すことである。これは ATM が受理する正しい操作ではないのでこれが繰り返される，という状況である。正しい入力値は20000である。

この状況を探査学習の理論に基づいてモデル化すると以下ようになる。まず，ATM から指示が与えられたとき，客はパネルを探索して，現金引出ボタンを選択する。その結果として，金額を尋ねられる。客はこれをゴールへの進展として評価する。次のステップとして客はシステムの要求「千円の倍数で」が，引き出したい金額を1,000で割ってその結果を入力することを意味していると解釈する。これにより次のアクションとして「20」を入力することを選択する。そして，同じ質問が繰り返される。利用客はこれを進展していないことを意味すると評価する。そして再び「20」を試みる。

### 2.3.2 認知ウォークスルー

デザイナー自身が探査学習の理論に基づいて簡単にユーザインタフェースデザインの分析を行えるように開発されたデザイン分析手順に認知ウォークスルーがある(Polson, Lewis, Tieman, & Wharton, 1992; Wharton, Rieman, Lewis, & Polson, 1994)。認知ウォークスルーではデザイナーは次の質問に答えることによってインタフェースデザインの不具合の発見を行える。

まず、最初にユーザのゴールを明確にする。

質問0) ユーザは何を達成したいのだろうか

そして、インタラクションの各段階で実行される探査、選択、評価の各ステップのシミュレーションを以下の質問1~3に答えることによって行う。

質問1) 正しい操作がユーザに明確に分かるようになっていようだろうか。

質問2) 正しい操作とユーザがしたい作業の記述が一致するであろうか。

質問3) ユーザは選択した操作に対するシステムの応答を正しく解釈できるであろうか。つまり、正しい選択をしたかどうか分かるであろうか。

認知ウォークスルーを行うと、以上の質問に対して答が No になるとき、対応する部分にデザイン上の問題点があるという解析結果が得られる。このような問題点の解決案を次のデザインに反映させることによってデザインの改良が行われる。

### 2.4 その他の認知モデル

本節では認知モデルの中でも工学モデルとして実際のデザイン開発に利用されてきているものを説明した。しかしながら、GOMS や認知ウォークスルーが基礎を置いている認知アーキテクチャは単純なものであり、そこでは人間と機器とのインタラクションに関わる認知プロセスのごく一面しか取り扱われていない。現在、認知モデリングの分野では、問題解決や学習メカニズム (ACT-R, Anderson, 1997; Howes & Young, 1997), 知覚・認知・運動 (EPIC, Kieras & Meyer, 1997), 理解過程 (LICAI, Kitajima & Polson, 1997) などを対象とした認知アーキテクチャの研究が進められ、将来、GOMS や認知ウォークスルーよりも包括的で精度の高くより有効な工学モデルが開発されることが期待されている。

### 2.5 文献

- Anderson et al: ACT-R: A theory of higher-level cognition. *Human-Computer Interaction, Special Issue on Cognitive Architectures in HCI*, 12: 1997.
- Bovair, S., Kieras, D.E., & Polson, P.G.: The acquisition and performance of text editing skill: A cognitive complexity analysis. *Human-Computer Interaction*, 5: 1-48, 1990.
- Card, S., Moran, T., & Newell, A.: The keystroke-level model for user performance time with interactive systems. *Communication of the ACM*, 23(7): 396-410, 1980.
- Card, S.K., Moran, T., & Newell, A.: *The psychology of human-computer interaction*, Hillsdale, NJ: Lawrence Erlbaum Associates, 1983.
- Cluah, M.C., John, B.E., & Pane, J.: Analyzing graphic and textual layouts with GOMS: Results of a preliminary analysis. In *Conference Companion of CHI'94 conference on human factors in computing systems*, 323-324, New York:ACM., 1994.
- Ekenon, J. & Palmiter, S.L.: Designing help using a GOMS model: An information retrieval evaluation. *Human Factors*, 33(2):185-204, 1991.
- Gong, R. & Kieras, D.: A validation of the GOMS model methodology in the development of a specialized, commercial software application. In *Proceedings of the CHI'94 conference on human factors in computing systems*, 351-357, New York: ACM, 1994.
- Gray, W.D., John, B.E., & Atwood, M.E.: Project Ernestine: A validation of GOMS for prediction and explanation of real-world task performance. *Human-Computer Interaction*, 8(3): 237-309, 1993.
- Haunold, P. & Kuhn, W.: A keystroke level analysis of a graphics application: Manual map digitizing. In *Proceedings of the CHI'94 conference on human factors in computing systems*, 337-343, New York: ACM, 1994.
- Howes, A. & Young, R.M.: Learning consistent, interactive, and meaningful task-action mappings: A computational model. *Cognitive Science*, 20: 301-356, 1996.



- 
- John, B.E. & Kieras, D.E.: Using GOMS for user interface design and evaluation: Which technique? *ACM Transactions on Human-Computer Interaction*, 3(4):287-319, 1996a.
- John, B.E. & Kieras, D.E.: The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Human-Computer Interaction*, 3(4):320-351, 1996b.
- John, B.E.: Why GOMS? *Interactions*, 2(4):80-89, 1995.
- John, B.E., Vera, A.H., & Newell, A.: Toward real-time GOMS: A model of expert behavior in a highly interactive task. *Behavior and Information Technology*, 13(4):255-267, 1994.
- Kieras, D.E. & Meyer, D.E.: An overview of the EPIC architecture. *Human-Computer Interaction, Special Issue on Cognitive Architectures in HCI*, 12: 1997.
- Kieras, D.E.: Towards a practical GOMS model methodology for user interface design. *The handbook of human-computer interaction* M. Helander, Ed.), 135-157, Amsterdam:North-Holland, 1988.
- Kitajima, M. & Polson, P.G.: A comprehension-based model of exploration. *Human-Computer Interaction, Special Issue on Cognitive Architectures in HCI*, 12: 1997.
- Lane, D.M., Napier, H.A., Batsell, R.R., & Naman, J.L.: Predicting the skilled use of hierarchical menus with the keystroke-level model. *Human-Computer Interaction*, 8(2):185-192, 1993.
- Lerch, F.J., Mantel, M.M., & Olson, J.R.: Translating ideas into actions: Cognitive analysis of errors in spreadsheet formulas. In *Proceedings of the CHI'89 conference on human factors in computing systems*, 121-126, New York: ACM, 1989
- Polson, P.G., Lewis, C., Rieman, J., & Wharton, C.: Cognitive walkthroughs: A method for theory-based evaluation of user interfaces, *International Journal of Man-Machine Studies*, 36:741-773, 1992.
- Wharton, C., Rieman, J., Lewis, C., & Polson, P.: The cognitive walkthrough method: a practitioner's guide. *Usability Inspection Methods* (J. Nielsen and R. Mack, eds.), New York:John Wiley, 1994.
-

図1 – GOMS モデルの応用事例 (John,1995 より)

文献	GOMS 技法	タスク領域	予測内容
Card, Moran & Newell (1983).	KLM	テキストエディタ	実行時間
Lane, Napier, Batsell & Naman (1993).	KLM	スプレッドシート	実行時間
Haunold & Kuhn (1994).	KLM	地図のデジタル化	実行時間
Card, Moran & Newell (1983).	GOMS	テキストエディタ, オペレーティングシステム, VLSI CAD	オペレータ系列, 実行時間
Lerch, Mantel & Olson (1989).	GOMS	スプレッドシート	作業記憶エラー
Ekenon & Palmiter (1991).	GOMS	HyperCard™	学習時間
John, Vera & Newell (1994).	GOMS	ビデオゲーム	オペレータ系列
Kieras (1988).	NGOMSL	テキストエディタ	学習時間, 実行時間, 作業記憶エラー
Bovair, Kieras & Polson (1990).	NGOMSL	テキストエディタ	学習時間, 実行時間
Gong & Kieras (1994).	NGOMSL	人間工学 CAD	オペレータ系列, 実行時間, 学習時間
Gray, John & Atwood (1993).	CPM-GOMS	電話交換手のワークステーション	実行時間
Cluah, John & Pane (1994).	CPM-GOMS	航空路スケジュール	実行時間

図2 - 打鍵レベルのオペレータの内容と実行時間

Card, Moran, &amp; Newell (1983) 図 8.1.打鍵レベルモデルのオペレータより

オペレータ	内容	時間(秒)
K	キーまたはボタンを押す シフトキーを押すこと、コントロールキーを押すことは、ひとつのK操作とみなされる。時間はユーザのタイプ技能によって変化する。以下に典型的な値を示す。	
	最も優れたタイピスト(135wpm)	.08
	優れたタイピスト(90wpm)	.12
	平均的なタイピスト(55wpm)	.20
	平均的な非職業的タイピスト(40wpm)	.28
	ランダムに文字をタイプする	.50
	複雑なコードをタイプする	.75
	キーボードに不慣れなタイピスト	1.20
P	ディスプレイ上の目標をマウスでポイントする ポイントする時間は目標までの距離、目標の大きさに応じて Fitts の法則に従って0.8 から1.5 秒の範囲で変化する。1.10 秒はその平均値である。このオペレータには、ポイントの後に通常行われるボタンを押す操作(0.2 秒)は含まれていない。マウスによるポインティング時間はジョイスティックなどの他のアナログポインティングデバイスに対してもよい評価を与える(第7章を参照)。	1.10
H	キーボードや他のデバイス上の定位置に手を移動する	.40
$D(n_D, l_D)$	全長 $l_D$ cmの折れ線を $n_D$ 本の線分によって描く このオペレータには強い制約条件がある。描画は全ての線分が一辺.56 cmの格子で・・・ようなシステムの上でマウスで行われることを想定している。ユーザは異なった描画技能を持っているので、時間は平均値である。	$.9n_D + .16l_D$
M	心的準備	1.35
$R(t)$	システム応答 コマンド毎に異なった応答時間が必要になる。応答時間はユーザが待たされる場合にのみ考慮される。	$t$

図3 – 心的準備オペレータの挿入規則  
Card, Moran, & Newell (1983) 図 8.2 より

メソッドのエンコーディングは、まず物理的な操作、システムの応答から始める。まず、規則0を使ってMを仮に挿入する。そして、個々のMに対して規則1から規則4を適用し削除すべきかどうかを確かめる。

規則0	すべてのKの前にMを挿入する(ただし、引数である場合は除く)。すべてのコマンド選択のためのPの前にMを挿入する(ただし、引数選択は除く)。
規則1	Mに続くオペレータが、Mの直前のオペレータから完全に予期できるものである場合にはそのMを削除する(例えば、PMK PK)。
規則2	MKの繰り返し系列が認知単位に属する(例えば、コマンド名)場合には、最初のMだけを残し、その他のMは削除する。
規則3	あるKが冗長なターミネーター(例えば、コマンド引数に続くコマンドターミネーター)である場合は、その前のMを削除する。
規則4	あるKが一定の文字列(例えば、コマンド名)を終了させるものである場合は、その直前のMを削除する。しかし、Kが一定でない文字列(例えば、引数文字列)を終了させる場合はその直前のMを保持する。

図4 – KLMによるマーク-削除メソッドの実行時間の予測

内容	オペレータ	時間(秒)
心的準備	M	1.35
単語の先頭にカーソルを移動する	P	1.10
(規則1により削除)	<del>M</del>	
マウスボタンをクリックする	K	0.20
(規則1により削除)	<del>M</del>	
単語の末尾にカーソルを移動する	P	1.10
(規則1により削除)	<del>M</del>	
シフトキーを押しながらマウスボタンをクリックする	K	0.28
	K	0.20
心的準備	M	1.35
削除キーを押す	K	0.28
全予測時間		5.86

図5 – GOMS モデルによる「文書を編集する」タスクの分析

ゴール：文書を編集する

- ・ゴール：単位タスクを編集する・・・単位タスクがなくなるまで繰り返す
- ・ゴール：単位タスクを得る・・・タスクを思い出せない場合
- ・ゴール：ページをめくる・・・文書ページの最後だった場合
- ・ゴール：文書から単位タスクを得る
- ・ゴール：単位タスクを実行する・・・単位タスクが見つかった場合
- ・ゴール：テキスト変更
  - ・ [ 選択：
    - ・ゴール：テキストを移動する
    - ・ゴール：単語を削除する
    - ・ゴール：語を挿入する ]
  - ・編集結果を確認する(1.35)

ゴール：単語を削除する

- ・ [ 選択：
  - ・ゴール：マーク-削除メソッドを使う
  - ・ゴール：削除キーメソッドを使う ]

ゴール：マーク-削除メソッドを使う

- ・カーソルを先頭に移動する(1.10)
- ・マウスボタンをクリックする(0.20)
- ・カーソルを末尾に移動する(1.10)
- ・シフトキーを押しながらマウスボタンをクリックする(0.48)
- ・ハイライトされていることを確認する(1.35)
- ・削除キーを押す(0.28)

予測時間：5.86 秒

図6 – NGOMSL モデルによる「文書を編集する」タスクの分析

NGOMSL による記述	外部オペレータ時間
ゴール [ 単語を削除する ] のためのメソッド	
ステップ 1. ゴール [ テキストをハイライトする ] を達成する	
ステップ 2. 削除キーを押す	0.20
ステップ 3. ゴールの達成により戻る	
ゴール [ テキストをハイライトする ] のためのメソッド	
ステップ 1. テキストの先頭の位置を決定する	1.20
ステップ 2. カーソルをテキストの先頭に移動する	1.10
ステップ 3. マウスボタンをクリックする	0.20
ステップ 4. テキストの末尾の位置を決定する(既知)	0.00
ステップ 5. カーソルをテキストの末尾に移動する	1.10
ステップ 6. シフトキーを押しながらマウスボタンをクリックする	0.48
ステップ 7. 正しくテキストがハイライトされていることを確認する	1.20
ステップ 8. ゴールの達成により戻る	
予測時間：5.48+0.1*13=6.78 秒	

図7 - 情報処理サイクル時間 (Card, Moran, & Newell, 1983, 第2章)

サブシステム	平均サイクル時間	範囲(ミリ秒)
知覚(視覚, 聴覚)	100	50-200
認知	70	25-170
運動	70	30-100

図8 - CPM-GOMS モデルを構成するためのテンプレート。 CPM-GOMS モデルを構成するためのテンプレート (John & Gray, 1995 より)。“画面を読む”というゴールを達成するためのテンプレート。上は、眼球運動を伴う場合。下はその必要がない場合。

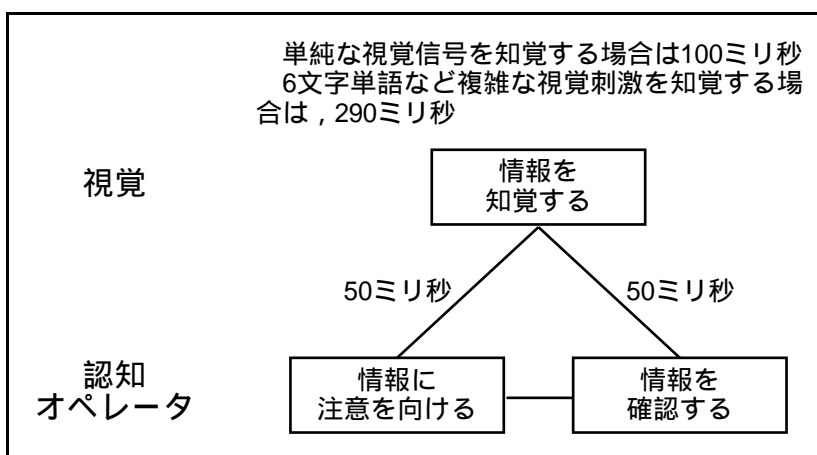
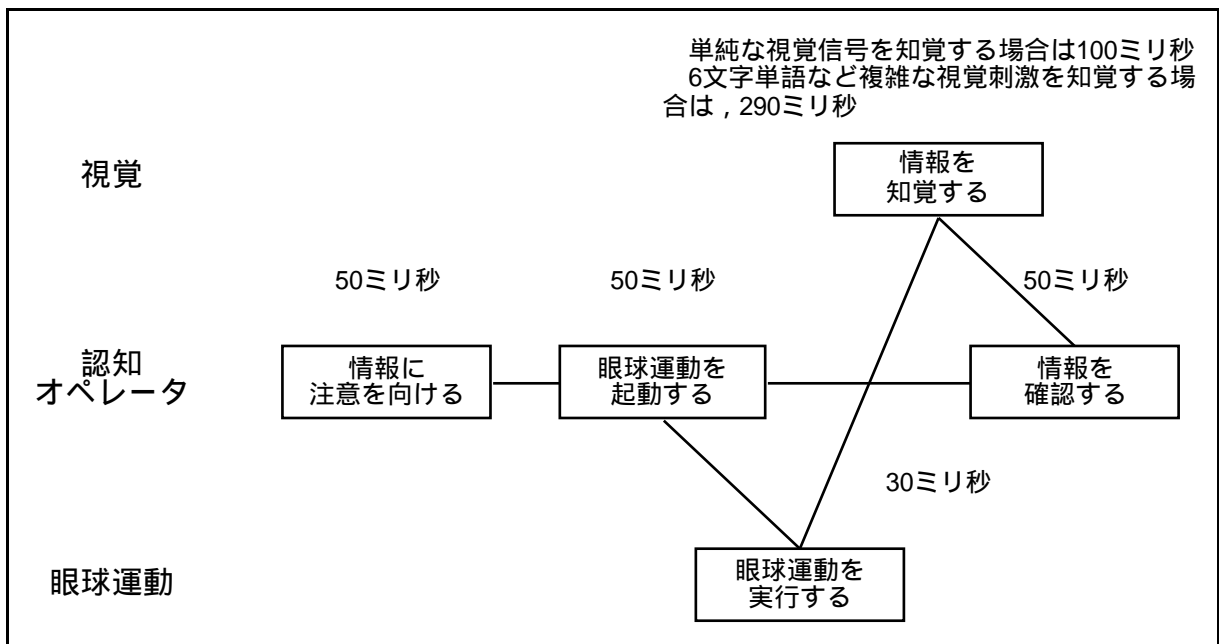


図9 - CPM-GOMS モデルによる マーク-削除メソッド の分析。テキスト削除メソッドの CPM-GOMS モデルの一部 (テキストのハイライトまで)。オペレータの右肩には、開始時間(上段)と実行時間(下段)が示されている。限界経路とその系路上のオペレータを太い枠で示す。

