

1L-5

# 柔軟な対話処理を実現する 知的マンーマシンインターフェースの一構築法

高野 智 海老原義彦 池田克夫  
(筑波大学) (京都大学)

ない場合でも、より詳しい解説が可能となる。

## 1.はじめに

現在、情報化社会と言われる中で、人間と、コンピュータを中心とした情報処理システムとのインターフェースは、完全であるとは言えない。今後、社会はより急速に高度情報化社会へと移行していくであろう。このような状況のもとで、我々は今までとは異なった、新しい情報化社会を構築する必要がある。つまり、人間とコンピュータとが協調できる社会を作らねばならない。

コンピュータの対話処理は、人間とコンピュータとのコミュニケーションである。しかし、現在のコンピュータには融通性がなく、人間がコンピュータに歩み寄って初めて人間とコンピュータとのコミュニケーションが成り立っているのが現状である。我々は、シェルに知識と知識処理機能を付加し、人間とコンピュータのコミュニケーションを人間同士の会話レベルにまで近づけるための研究を行っている。本稿では、知的コミュニケーションに必要な知識表現を考え、知識の獲得と利用方法、または問題点について述べる。

## 2.コンピュータに持たせる知識

人間とコンピュータとのコミュニケーションを、人間同士のコミュニケーションに近づけるためにマンーマシン・インターフェースに要求されることは、次の事柄である。

(1) 柔軟な解釈

(2) 個人用の仕立

人間は、コミュニケーションの過程で多くの知識を利用していると考えられるが、まず第一段階として、本稿では次の2つの知識の獲得と利用について考えた。

(1) タイプミスのパターンとユーザのくせ

(2) ユーザの作業内容の状態遷移

### 2.1 タイプミスのパターンとユーザのくせ

対話型処理においてコマンドのタイプミスが生じた場合、コンピュータは”コマンドが見つかりません”と出力するだけで何の解釈もしてくれない。ところが、そのほとんどは人間が見るとその意図するところは十分理解できるものである。そこで、タイプミスが生じたとき、コンピュータにも何らかの解釈をさせるために、タイプミスに関する知識を持たせる。まず一般的に起こりやすいタイプミスをコンピュータに教えておく。それをもとに対話処理の過程で、ユーザの頻出するタイプミスのパターンの知識を得る。

### 2.2 ユーザの作業内容の遷移

人間同士の会話では、話の筋というものが重要になる。意味の不明瞭な事柄も話の筋から理解できることもある。話の筋と同様にして、コンピュータの対話処理にも流れがある。例えば通常よく見られる作業は、ファイル作成、コンパイル、実行とデバッグの繰り返しである。シェルがこの対話処理の流れを把握できれば、ユーザが次に入力するものの予測が可能となり、タイプミスのパターンだけではユーザの意図したものを特定でき

Intellectual Man - machine Interface

with Flexible Interactive Processing Facilities

Satoshi TAKANO<sup>1</sup>, Yoshihiko EBISHIARA<sup>1</sup>, Katsuo IKEDA<sup>2</sup>  
1. University of Tsukuba      2. Kyoto University

## 3.個々の機能の実装方法

ここでは、前述した知識の獲得と利用を実現するための機能について述べる。我々は、UNIXのシェルに知識処理機能を付加している。入力されるものは、コマンドとファイル名などの各種パラメータである。パラメータの誤りは、多くの場合シェルが検出するのではなく入力コマンドによって検出されるため、現段階では、以下に示す機能の適用は入力コマンドに限られる。

### 3.1 タイプミスのパターンとユーザのくせ

タイプミスには、大きく分けて次の4つのパターンが考えられる。

(1) 脱落

(2) 挿入

(3) 転置

(4) 置換

大分類	小分類
脱落	脱落
挿入	鍵盤配列の隣接文字 二重打ち 反復打ち
転置	両手 左手 右手
置換	鍵盤配列の隣接文字 シフト チャタリングを含むもの

我々は、対話処理におけるタイプミスのデータを採り、それを分析してさらに細かく分類した。その結果が表1である。

入力コマンドに誤りがあった場合、まずユーザのサーチパスを参照し認識可能なコマンドのリストを作る。そして、表1に分類されているタイプミスを想定して、入力コマンドとコマンドリスト中のコマンドとを比較して、ユーザが意図したもの候補を検索し出す。そして、タイプミスのパターンと共に一時格納する。候補が複数現れた場合、そのユーザのタイプミスのパターンの確率をもとに、少数に限定してディスプレイに出力する。次に誤りのないコマンドが入力されれば、誤った1つ前の入力コマンドと比較して、そのタイプミスのパターンをカウントして、新たに確率を計算する。

表1 タイプミスのパターン

### 3.2 ユーザの作業内容の遷移

ここでは、ユーザが次に入力すると考えられるコマンドを予測するための機能について述べる。現在、次の2つについて開発、実装中である。

(1) コマンドのLRUスタックの導入

(2) 遷移確率の導入

次にそれぞれについて説明する。

#### [コマンドのLRUスタックの導入]

入力コマンドのデータを見てみると、会話処理には、一度入力されたコマンドは近い将来再び入力されるという局所参照性があると考えられる。またユーザは、特定の作業用のディレクトリを作り、その中で作業する場合が多い。その場合、作業の中心はそのディレクトリ内で行われ、必然的に他のディレクトリに移ったときと、作業の内容も異なったものとなる。そこで、ディレクトリごとにコマンド履歴を作り、ディレクトリを移動するたびに、移動前のディレクトリ内でのコマンド履歴をファイルに格納

し、移動先のディレクトリの前回のコマンド履歴を取り出してくれる。そして、新しいディレクトリのコマンド履歴からLRUスタックを積み直し、その後の入力コマンドは、ディレクトリのコマンド履歴に入れられ、同時にLRUスタックにも積まれる。

実際にLRUスタックをシェルに組み込み、データを採った結果、5つ前までのコマンドを再度入力する確率は60~70%，3つ前までのコマンドについては50~60%となった。

### [遷移確率の導入]

コマンドには、互いに関連する組がいくつもある。例えば、プロセスを消去する“kill”というコマンドを入力する前には、プロセス番号を調べる“ps”というコマンドが入力される可能性が高い。そのような場合により効的なコマンド予測のために、コマンド遷移確率の導入を考えた。対話処理の過程で、コマンドごとに、次に入力されたコマンドと遷移確率を知識として格納する。この知識を利用することにより、次に入力される可能性が高いコマンドが分かる。

ここで問題となるのは、データの量である。UNIXのコマンドは標準的なものだけで200以上もある。よって、すべてのコマンドについて、個々のコマンドへの遷移確率を計算し格納しようとすると、少なくとも4万個の記憶領域が必要となる。しかもそのデータの多くは0であることが予想される。これは非効率的であり、実用的ではない。この問題の対処方法として次のことを考えた。

#### (1) コマンドのグループ化

#### (2) よく使用するコマンドに限定

- コマンドのグループ化：**遷移する状態数を減らすために、コマンドのグループ化を考えた。全コマンドを50のグループに分け、グループからグループへの遷移確率を計算する。また、個々のグループ内でコマンドの生起確率を計算し、グループ内の優先順位も決定できるようにする。グループ化の方針は、頻出コマンド同士は同一グループには入れないことと、同様な効果をもたらすものは同一グループに入れることである。これは、なるべく同一グループ内で生起確率に偏りが見られるようにすることと、グループの遷移確率のばらつきをなくすためである。ここで問題となるのは、新たに追加されたコマンドの処理である。

- よく使用するコマンドに限定：**コマンドの使用状況のデータを採ってみると、生起確率1%以上のものはせいぜい20個程度で、かなりの偏りが見られる。よって、ユーザーの使用頻度の高いもの50個に限定して遷移確率を計算する。ここでの問題点は、あまり使われないが関連性の高い2つのコマンドについては無視されることである。

現在、上述の2つのアルゴリズムの実装はほぼ完了している。これからデータを採って最適化を図り、問題点を取り除く必要がある。

## 4. システム全体の実装

本研究のシステム環境は、米国シーケント社のシンメトリ・システムで、OSは4.3BSDのシステムVへの拡張版であるDYNIX (R) V3.0.12を採用している。本システムは8つのCPUを搭載しており、物理的な並列化が可能である。

図1に本システムのシステム構成を示す。太い線で囲まれているものはプロセスである。プロセスは並列に実行される。並列化によりプロセス単位の機能の追加を容易にし、本来の機能である「入力・実行プロセス」の実行性能に重大な影響は与えないようになる。実線の矢印は、プロセス間での入力データの流れを表わす。そして点線の矢印は、知識データの流れを表わす。

「入力監視・候補選択プロセス」は、入力誤りを検出しユーザの意図を判断するプロセスである。このプロセスは、コマンド誤りを検出したらすぐに候補選択のプロセスを生成する。ここで、鍵盤配列の知識、LRUスタック、遷移確率からのコマンド予測の知識、個人のくせなどの知識を利用することにより正しいコマンドの候補をしぶる。入力文字列が修正されたと

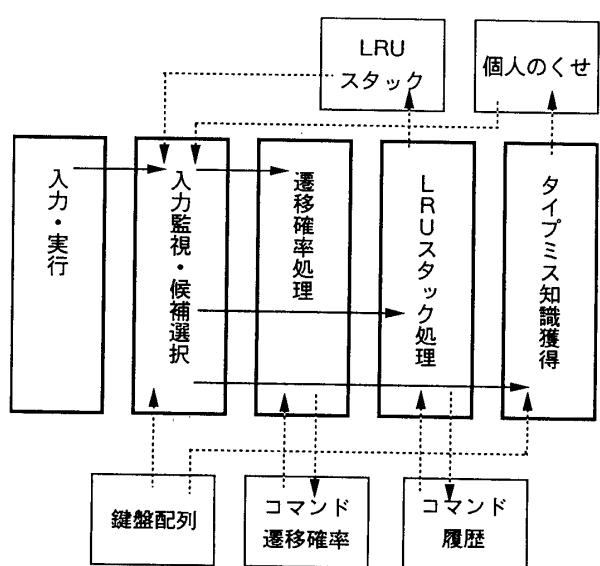


図1 本システムのシステム構成

きは、そのときの情報を知識学習プロセスに送る。また、コマンド誤りがない場合、「遷移確率処理プロセス」、「LRUスタック処理プロセス」に入力データを送る。

「遷移確率処理プロセス」、「LRUスタック処理プロセス」は、作業内容の遷移を把握するためのプロセスである。

「タイプミス知識獲得プロセス」は、ユーザのタイプミスのくせの知識を獲得するプロセスである。コマンドの誤りが起こったときに、誤り文字列を一時保持しておく。次に誤りのないコマンドが入力された場合に、両者の比較を行なって、ユーザの犯した誤りを特定し知識を獲得する。

## 5. おわりに

本研究では、人間とコンピュータとの高度なコミュニケーションを可能にする、高機能シェルの開発を行なっている。本システムの完成により、人間とコンピュータのコミュニケーションが一步人間側の立場に立ったものになることが期待できる。

本稿では、タイプミスのパターンと作業内容の状態遷移を知識として持たせた。人間は、コミュニケーションのために実際に多くの知識を利用するものと考えられる。今後、さらにコミュニケーションの基本的な機構について考察し、コミュニケーションに必要な知識を検討し、またAI的手法などの導入によって、より高度な知的マン・マシン・インターフェースの実現を目指したい。

## 謝辞

本稿の作成にあたりご協力頂いた筑波大学情報学類4年大西健司氏、熊井康子嬢に深く感謝します。

## 参考文献

- 池田克夫：“知的インターフェース”，電子通信学会誌，第69巻11号，pp. 1160–1166 (1985).
- Norman, D. A. : A Psychologist Views Human Processing : Human Errors and Other Phenomena Suggest Processing Mechanisms, IJCAI invited address , pp. 1097–1101 (1981)
- 田中栄一、小橋口高広、島村邦彦：“繰りの置換誤りの高速訂正法”，情報処理, Vol. 27No. 2, pp. 177–182 (1986).
- 高野智、海老原義彦、池田克夫：“鍵盤配列、コマンド履歴、個人のくせを知識として持つ知的マン・マシンインターフェース”，第2回人工知能学会全国大会, pp.505–508 (1988)