

Popie : フローメニューに基づく日本語入力手法

佐藤 大介[†], 志築 文太郎[†]
三浦 元喜^{††} 田中 二郎[†]

FlowMenu はメニュー操作や文字入力を行うインタフェースであるが、日本語入力のような候補選択をとまなう入力について考慮されていない。そこで FlowMenu を日本語入力ができるように拡張したうえで、日本語入力システム Popie として実装を行い、その評価を行った。Popie は子音入力方式や予測・補完入力を用いることで、入力速度の向上を図っている。さらに、従来の子音入力方式では候補数が多くなりすぎた場合に、選択に時間がかかるという問題を、母音を必要に応じて決定する母音選択を用いることで解決した。実験の結果、子音入力と母音選択による本システムの入力方式は、ローマ字や子音入力方式よりも入力に必要な操作数が少ないことが分かり、また母音選択導入によって入力操作が効率的になることが分かった。ユーザ実験では平均の入力速度が 25 文字/分、最大で 31 文字/分であった。

Popie: A Japanese Input Method Utilizing FlowMenu

DAISUKE SATO,[†] BUNTAROU SHIZUKI,[†] MOTOKI MIURA^{††}
and JIRO TANAKA[†]

FlowMenu is an interface which performs menu operation and a character input. However, the input accompanied by candidate selection like a Japanese input has not been considered. Therefore, we have extended FlowMenu to make Japanese input possible, and developed Japanese input system called "Popie." Popie attempts the improvement of the input speed by using the consonant input method and the prediction/complement input. Furthermore, the problem of taking time to select a target word in case of too much ambiguity is solved by selecting corresponding vowel for consonants. Our results of experiments show that our input method, using the both of the consonants input and the vowel selection, reduces the number of operations, comparing with both the Roman character input method and consonant input method. The average input speed is 25 characters per minute (cpm), and the maximum input speed is 31 cpm.

1. はじめに

コンピュータを操作するために、マウス・キーボードではなくペン型のデバイスを用いる、いわゆるペン・コンピュータの数が増えている。最近では、プラズマテレビにタッチパネルを組み合わせた大画面のペン・コンピュータも登場し、ペン・コンピュータを操作するためのインタフェースもそれに合わせて改良されて

きている。

ペンでコンピュータを操作する場合、マウスで操作する場合に比べて画面が大きくなるほど手の移動量が増加し、マウスのクリックに相当するタップ操作を正確に行うことが難しい。そのため、ペンの移動量の増加を防ぐために、画面上に散在するメニューなどのインタフェースを、ペンが画面に触れた点の周囲に集約した Pie Menu¹⁾ や Marking Menu²⁾, FlowMenu³⁾ などのメニューが開発されてきた。これらは円形のメニューであり、各メニュー項目までの距離が均等になっている。また、ペンを動かす方向によってメニュー項目を特定できるため、選択速度が速くなるという利点を持っている⁴⁾。

Pie Menu はペンのタップによってメニューを選択するのに対して、Marking Menu や FlowMenu はペンのストロークを用いてメニューを選択する。このた

[†] 筑波大学システム情報工学研究科コンピュータサイエンス専攻
Department of Computer Science, Graduate School of
Systems and Information Engineering, University of
Tsukuba

^{††} 北陸先端科学技術大学院大学知識科学研究科
School of Knowledge Science, Japan Advanced Institute of
Science and Technology
現在、日本アイ・ピー・エム東京基礎研究所
Presently with Tokyo Research Laboratory, IBM Japan

めペンタップによる操作ミスを軽減し、より素早い操作を可能にしている⁵⁾。FlowMenuはMarking Menuの発展であり、1つのストロークで複数のメニューを選択でき、文字入力もメニューと同じ操作体系で行うことができる。これらのことから、フローメニュー（FlowMenuの操作体系を持つメニューをこれ以降フローメニューと呼ぶ）はペン・コンピュータに対して有効なインタフェースであると考えられる。

しかし従来のフローメニューでは、アルファベットや記号などの文字入力しか考慮されていないため、日本語入力において重要な役割を持つ候補選択のような操作ができない。また、様々な日本語入力手法のうちフローメニューに適した入力手法が何かというような知見はない。そこでフローメニューを拡張することで候補選択を可能にし、それに適した日本語入力手法を考察することで、日本語入力が可能なフローメニューを実現し、日本語環境においてもフローメニューを活用できるようにすることが本研究の目的である。

以下2章ではフローメニューについて説明し、3章ではフローメニューの拡張と、日本語入力手法について述べる。4, 5章ではフローメニューにおける日本語入力システムの実装であるPopieについて、そのインタフェースと実装の詳細について述べ、6章ではその評価について述べる。

2. フローメニュー

フローメニューはペンが画面に触れた位置を中心として表示され、中心部とその周囲の領域を8等分した合わせて9つの領域からなる図1のようなインタフェースである。中心の領域をレストエリア、その周囲を8等分にしたそれぞれの領域をオクタントと呼ぶ。ここで周囲の領域は必ずしも8等分である必要はないが、8等分から12等分に増やすとエラー率が上昇するため²⁾、8等分以下で利用されることが多く、本研究においても8等分を採用した。

通常フローメニューでは、レストエリアからオクタント、オクタントからレストエリアにペンが移動する際に、移動先もしくは移動元のオクタントに対応するコマンドが実行される。ペンは初期状態でレストエリアにあり、8つのオクタントに対応するメインメニューの項目8個が選択可能である。いずれかのオクタントにペンを移動するとサブメニューが表示される。続いて、ペンをいずれかのオクタントからレストエリアに戻すことで、そのオクタントに対応するサブメニューの項目が選択される。たとえば図1において「C→N→NE→C」のような順でペンを動かす場合、

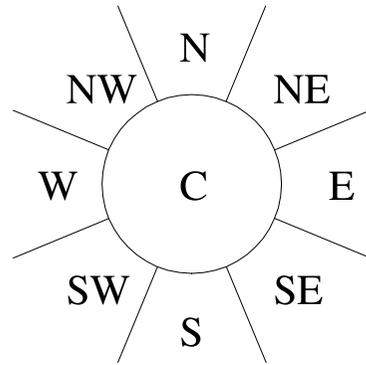


図1 フローメニューの外観
Fig. 1 The appearance of FlowMenu.

選択されるメインメニューの項目はNの位置、サブメニューの項目はNEの位置である。

ここでは、メニューは2階層であるが理論的には無限階の階層を構成することが可能である。ただし多くの場合、操作性の点から2階層で利用されている。

3. 日本語入力のためのフローメニューの拡張

フローメニューはメニュー項目の位置を覚えることで素早い入力が可能になる点と、連続した操作を連続したストロークで実行できる点が大きな特徴である。そのため、日本語入力をフローメニューで行えば、キーの入力を素早く行うことができる点と、連続したストロークでキー入力と候補選択の操作をシームレスにできることが期待される。ただし、シームレスな操作を実現するためには注意してメニューを設計する必要がある。このことを考慮したうえで、日本語入力をフローメニューで行うために以下の2点について検討した。

- (1) フローメニューで考慮されていない候補選択をどのように実現するか。
- (2) 様々な日本語入力手法のうち、フローメニューに適した手法はどれか。

以下に、上記に関する検討とそれに基づいて行ったフローメニューの拡張について述べる。

3.1 候補選択に関する検討と拡張

一般的な日本語入力ではキーの入力に変換がともなっており、変換の際、入力に対して複数の候補が提示されるため、候補を選択するという操作を行う。

フローメニューにおいては、あるメニューの位置が第1位の候補で隣が第2位の候補というように、候補の順位によって候補のフローメニューにおける位置を決めることができる。しかし、候補は入力によって異なり同じ入力でも学習によって候補の順位は動的に変

化するため、目的の候補の順位がユーザには分からない。よってフローメニューの利点である素早い操作は期待できないし、メニューが2階層であれば候補は最高でも64通りしか提示できない。

そこでフローメニューの拡張を行い、ある位置のメニューを選択するとその位置に対応する候補（対応候補）が選択されるようにし、その対応候補はリストとして表示し、スクロールによって自由に変えられるようにする。ここで、候補を対応させるオクタントは1つ以上で連続したオクタントを利用することが望ましく、1つのオクタントに2つ以上の候補を割り当てることは表示との対応関係が分かりにくいいため、1つのオクタントに1つの候補を割り当てるのが良い。

そして、キー入力と候補選択の操作が完了したときに必ず中心のレストエリアにペンが戻るように操作を設計することで、キー入力と候補選択の操作をシームレスに統合することができる。この際、ペンを使った操作ではユーザの腕によって隠れてしまう部分ができることを考慮し、腕に隠れてしまう部分には変化のないキー入力などの操作を配置し、動的に変化する対応候補のリストはつねにユーザから見えるように配置するべきである。

3.2 日本語入力手法の検討

日本語入力には様々な方式があるが、ここではフローメニューで日本語入力することを前提として、候補選択を必要とする入力方式について考える。

表1にかな入力方式、ローマ字入力方式、子音入力方式、母音入力方式について入力に使うキーの組合せ数と入力の曖昧さについて示した。

フローメニューは2階層で64個のメニュー項目を設定できるが、サブメニューの項目のうちメインメニューの項目の位置から離れている項目は、選択にかかる時間が長くなりエラーも増えるという実験結果がある⁶⁾。そのため、かな入力方式のようにキーが多い入力方式では、入力速度やエラー率の点で不利であるため、かな入力方式は適さない。

ローマ字入力方式と子音・母音入力方式を比較する

ローマ字の子音のみの入力を基本とする方式。ただし「あ行」には子音がないので便宜的に「A」で代用し、子音として扱う。たとえば「さとう」を入力したい場合は「STA」を入力する。「Y」も言語学上は母音であるが、子音として扱う。また「ん」や「ー」はわ行の文字とし、濁点や半濁点の付く文字には、基本の音の子音のみを用いるか、特別なキーを割り当てる。たとえば「がぎくげご」を「K」もしくは「K*」で表現する。

ローマ字の母音のみの入力を基本とする方式。たとえば「さとう」を入力したい場合は「AOU」を入力する。「ん」や「ー」は子音入力と同様に代用のキーに割り当てられる。

表1 各入力方式の入力キーの組合せの数と入力の曖昧さ
Table 1 The number of input key combinations and ambiguity of input on each input method.

入力方式	キー	組合せ	曖昧さ	
かな入力	50 個	かなの数	多 ↓	少 ↓
ローマ字入力	26 個	かなの数		
子音入力	10 個	子音の数	少 ↑	多 ↑
母音入力	5 個	母音の数		

と、日本語では子音と母音をほぼ交互に入力するので、子音・母音入力方式ではほぼ半分の操作で入力が行える。また、ローマ字入力方式、子音入力方式、母音入力方式の順に覚えるべきキーの数が少なくなる。しかし、子音・母音入力方式では変換の曖昧さが多くなるので、これらを比べるには、変換の曖昧さが入力操作をどれだけ増加させるかという点を考慮する必要がある。

携帯電話向けのシステムで子音入力方式を採用している Touch Me Key⁷⁾ では、提示する候補の並びを工夫することで、変換の曖昧さの問題をある程度解決している。Touch Me Key と携帯電話で一般的に使われる他の入力方式との比較では、入力に必要な操作数を削減する効果が示されている。

一方、母音入力方式は、子音入力方式と入力にかかる操作数は同じであるが、曖昧さがより大きくなるだけである。

以上のことをふまえて、フローメニューにおける日本語入力システムには子音入力方式を採用した。我々の子音入力方式では「AKSTNHMYRW」の10個のキーを用い、子音のない「あ行」は便宜的に子音を「A」とし、「ん」や「ー」は「W」に含めた。濁点や半濁点、拗音については基本の音の子音、たとえば「ば」や「ぱ」は「H」、「ゃ」は「Y」のように扱っている。また、本論文ではこれ以降「TNK」などのキーの入力列を単に子音列と呼ぶ。

3.2.1 従来の子音入力方式の問題点

Touch Me Key では、子音入力方式において変換候補の数が増大するという問題に対して、隠れマルコフモデルと PPM (Prediction by Partial Match) を用いた言語モデルにより、ある文脈においてよく使われると推測できるものを候補の上位に出すことで解決している。しかし、文脈が切り替わったり、文脈が利用できない状況において、候補の数が極端に多くなってしまい選択に時間がかかることになる。

実際に母音入力方式のシステムを試作したが、用意した10万単語を超える辞書では候補の数が多くなりすぎてしまい、入力するのに余計に時間がかかってしまった。

この問題は母音選択の機能を導入することで解決できる。母音選択とは、子音列の先頭から適宜母音を決定する機能で、「TNK」、「たNK」、「たなK」のように子音列から仮名を決定する操作である。

ここで、子音を N 個入力したうち先頭から M 個の母音を確定することを考えると、各 N, M に対して約 5^{N-M} 個の仮名の組合せが存在しそれぞれに候補が複数存在する。この候補数の平均値を表 2 に示した。「TNK」と入力した場合は $N = 3, M = 0$ に相当し、「たNK」と入力した場合は $N = 3, M = 1$ に相当する。この表から分かるように、先頭の 1 文字だけでも母音を決定することで候補の数を大幅に絞り込むことが可能である。

子音を入力してさらに母音を選択するという操作では、ユーザは入力文字列を 2 回考えることになり効果的な操作ではないように思われる。しかし母音選択は候補を絞り込むことが元々の機能であり、子音入力のようなキー入力とは使い方が異なる。候補数が多くなった場合には選択操作を行うよりも効率が良くなると考えられる。

3.2.2 母音選択のための拡張

子音入力、候補選択、および母音選択のすべてでシームレスな操作を可能にするには、フローメニュー

表 2 実装システムの辞書において、子音 N 個のうち先頭から M 個に対応する母音を確定したときの候補数の平均
Table 2 The number of candidates in our system's dictionary when the user inputs N consonants and selects M vowels from the beginning of the consonant sequence.

文字数 N	確定文字数 M					
	0	1	2	3	4	5
1	194.10	23.40				
2	129.42	20.05	4.99			
3	34.63	7.26	3.04	1.95		
4	8.06	3.10	2.19	1.60	1.46	
5	2.31	1.61	1.47	1.37	1.27	1.20

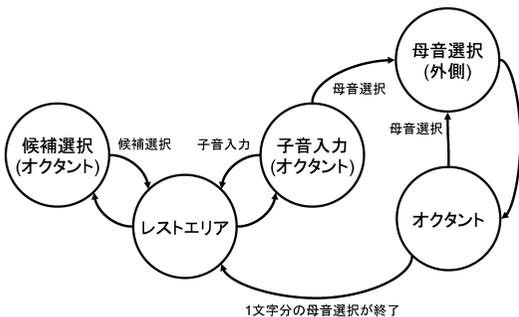


図 2 入力の状態遷移図

Fig. 2 State transition diagram of input.

のさらなる拡張が必要である。

フローメニューは 2 章で述べたように、中心とその周囲を 8 等分にした 9 つの領域から構成されている。我々は母音選択を行うために、オクタントの領域を限定してドーナツ状にし、オクタントよりも外側の領域を追加した 10 個の領域で操作を行うようにフローメニューを拡張した。

図 2 に入力の状態遷移図を示す。遷移を示す矢印のラベルは状態が遷移するときにかかる入力を示し、ラベルがない遷移は何も入力がないことを意味する。

状態はペンが含まれる領域によって変わる。レストエリアから、子音入力のオクタント、もしくは候補選択のオクタントに移動し、レストエリアに戻ると、それぞれの入力が発生する。子音入力のオクタントから外側に移動すると、その時点で母音選択が発生し、外側、オクタントと行き来することで、同じ子音の濁点、半濁点などの選択が順次行える。この状態からレストエリアに戻るときは、別の状態を経由してレストエリアに戻るため入力は発生しない。また、ここで 1 文字分の母音選択が終了し、次の母音選択では次の子音に対する母音選択が行われる。

4. 日本語入力システム Popie

Popie の主な特徴は

- (1) すべての操作をフローメニュー上で実行でき、その特徴を生かして素早い入力ができる、
- (2) 入力には基本的に子音 (10 個のキー) を用いる、
- (3) 入力途中の子音列から補完される単語や、入力された単語から予測された単語を選択できる、

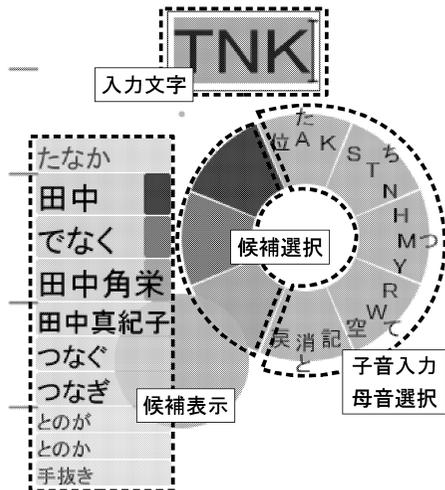


図 3 Popie のインタフェース

Fig. 3 The appearance of Popie.

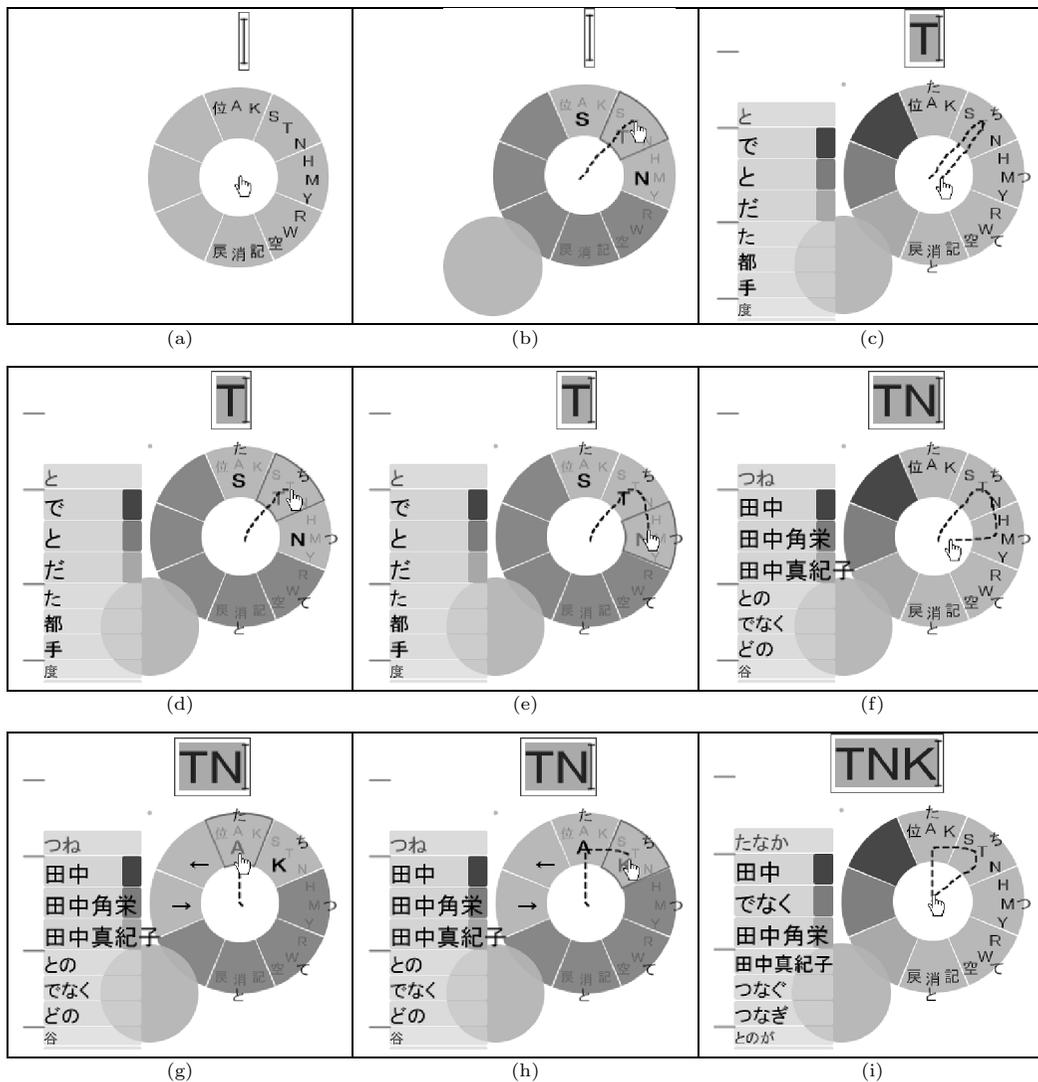


図 4 Popie において、子音列「TNK」を入力する例
 Fig. 4 Example of inputting “TNK” with Popie.

(4) 候補が多すぎる場合など、適宜子音に対応する母音を選択することで絞込みを行える、の4点である。(3)において「補完」とは入力された子音列が先頭にマッチする単語を提示することであり、たとえば「TN」の入力で「田中」などを提示することである。また、「予測」とは入力された単語の直後に続く単語を提示することで、たとえば「田中」を入力直後に「真紀子」などを候補として提示することである。これらの機能はいくつかの入力手法で入力速度向上の効果を得られており^{7),8)}、本手法にも適用可能である。

4.1 Popie のインターフェイス

Popie のインターフェイスを図 3 に示す。この図は、「田中」と入力するために「TNK」と子音を入力した

状態を示している。Popie のインターフェイスは、ユーザが操作を行う部分と、候補や入力文字の表示部分に分けられる。

8つのオクタントのうち上、下、右側3つのオクタント(図1における記号がN, NE, E, SE, Sの5つ)では主に子音入力を行い、その外側部分で母音選択の操作を行う。子音入力に使う10個の子音キーはかな順に配置し、ユーザが覚えやすいようにしている。また、「位」「空」「記」「消」「戻」と表示された部分ではそれぞれ、カーソル位置の変更、空白の挿入、記号の挿入、1文字消す、アンドウ・リドゥなどの操作を行う。

残りの3つのオクタントでは候補の選択操作を行う。

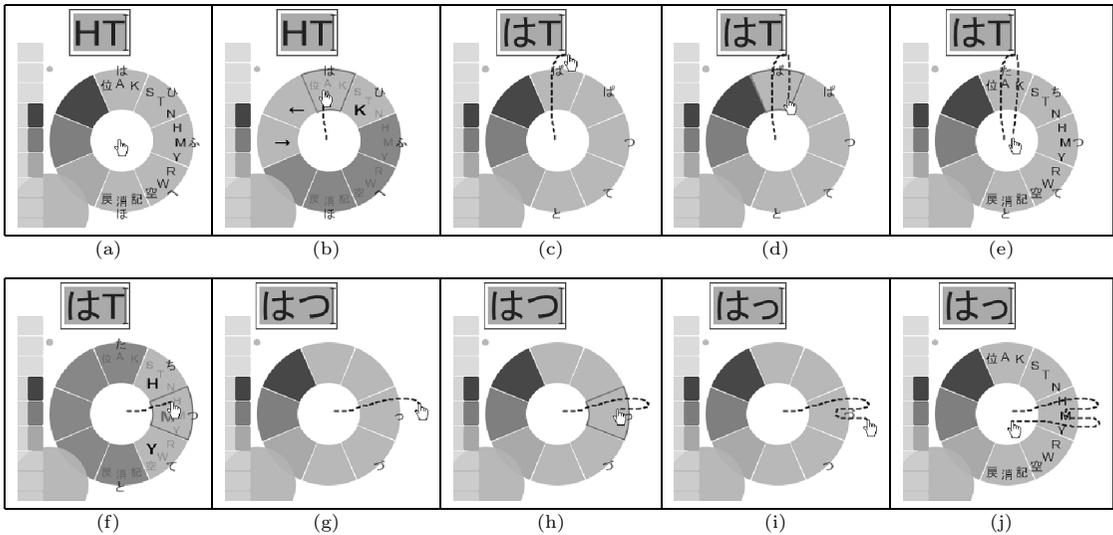


図5 Popieにおいて、母音を選択する例
Fig. 5 Example of selecting vowels with Popie.

上部に表示されている「TNK」の部分にはユーザが入力した文字を表示する．左側のリストはユーザの入力に対してシステムが提示した候補を表示している．リストの「田中」の位置に表示される候補が左上のオクタントに対応し、順に「でなく」が左のオクタント、「田中角栄」が左下のオクタントに対応する．リストの横に表示されている円は候補の量を示しており、候補数によって円の大きさが変化する．

このインタフェースは右利きのユーザを想定しており、左利きのユーザには左右が逆転したインタフェースを提供する．これは、ユーザの手で動的に変化する候補を隠さないようにするためである．逆に子音入力や母音選択を行う部分は腕に隠れてしまうが、子音キーなどの位置は固定されており、位置を覚えることで問題は解消される．

4.2 子音の入力

図4にPopieを使って子音列「TNK」を入力する操作を示す．図中の点線は動作の説明のためにペンの軌跡を示したものである．まず初期状態から(a)、「STN」とラベルの付いた右上側のオクタントにペンを移動し、「S」と「T」と「N」それぞれが別れた状態のサブメニューを表示させ(b)．そのままレストエリアにペンを戻すことで「T」を入力する(c)．同様に「STN」のオクタントにペンを移動してサブメニューを表示させ(d)．続けてサブメニューの右下側のオクタント「N」の位置にペンを移動し(e)．レストエリアにペンを戻すことで「N」を入力する(f)．さらに「位AK」とラベルの付いた上側のオクタントにペンを移動してサブ

メニューを表示し(g)．「K」のオクタントにペンを移動した後(h)．レストエリアに戻すことで「K」を入力する(i)．

図中では1文字入力するごとに線が途切れているが、ここで示した3つの子音を入力する操作は1つのストロークで実行されている．各子音はそれぞれ対応するストロークによって入力され、ユーザは子音キーの位置を覚えることで、子音を素早く入力することができる．

4.3 母音の選択

母音選択は、入力された子音に対して先頭から順番に母音を指定する操作であり、拗音や濁音、半濁音などの選択もまとめて行う．図5に子音列「HT」を「はっ」という文字列に確定する操作を示す．

まず、「HT」が入力された状態で(a)．上側のオクタント上部に表示された「は」を横切るようにペンをオクタントの外側に出し「は」を選択する(b,c)．その後ペンをレストエリアに戻すと「は」が確定する(d,e)．次に「っ」を確定するため最初に「っ」を選択する(f,g)．その後オクタントの外側からオクタントの内側までペンを戻し(h)．もう一度外側にペンを移動させることで「っ」を選択する(i)．最後にレストエリアにペンを戻すと「っ」が確定する(j)．

母音はオクタント上側から順にというお順に配置されており、オクタントの外側の境界上に、選択される仮名が表示されている．

4.4 候補の選択

変換の候補は、ユーザの子音入力や母音選択に対し

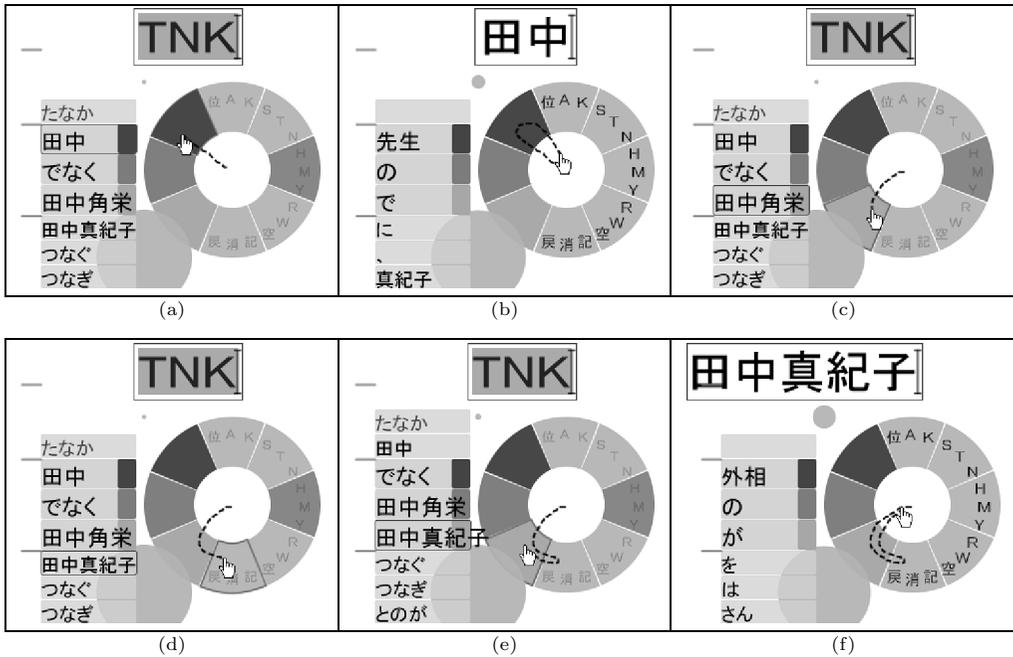


図 6 Popie において候補を選択する例
 Fig. 6 Example of selecting candidate with Popie.

て、フローメニュー左側に縦長のリストに表示される。候補の選択は基本的に左側 3 つのオクタントを使う。図 6 (a), (b) に示すように、「田中」は対応する左上のオクタントにペンを移動し、そのままレストエリアにペンを戻すことで選択する。上位 3 候補はスクロールすることなく選択が可能である。

目的の単語が上位 3 候補にない場合は、母音選択により絞り込むか、候補をスクロールすることで候補を選択する。図 6 (c) ~ (f) では、候補を 1 つスクロールして「田中真紀子」を選択をしている例である。文献 9) ではスクロールに関する詳しい考察を行っている。

5. Popie の実装

5.1 システム構成

実装に使用した言語は Java で、Java™2 SDK, Standard Edition バージョン 1.4.2 を使用した。FlowMenu³⁾ はソースコードが公開されていないので、フローメニューの基本部分から実装し、Popie インタフェースはその派生として実現した。プログラムは全体で約 8,000 行である。Popie の入力エンジンはインタフェースと切り離して実装してあるため、異なるインタフェースを持つシステムから容易に利用可能である。

Popie のプログラムは以下の URL で公開している。
<http://www.iplab.cs.tsukuba.ac.jp/popie/>

表 3 辞書データの形式
 Table 3 Format of dictionary.

辞書	データの形式
単語辞書	スコア, 読み, 単語
単語間関係辞書	スコア, 直前の単語, 単語, 読み
ユーザ辞書	スコア, 読み, 単語 { , 直前の単語 }

5.2 辞書データ

辞書データは変化のない静的な辞書が 2 種類とユーザ辞書からなる。静的な辞書は単語の出現回数からなる単語辞書と、単語が続けて出現した回数からなる単語間関係辞書であり、ユーザ辞書は単語と単語間関係の両方のデータを含み、ユーザの入力によって更新される。それぞれの辞書のデータ形式は表 3 のとおりである。

静的な辞書データは CD-毎日新聞 2001 年度版¹⁰⁾ のテキストデータを、形態素解析システム茶釜¹¹⁾ を用いて単語ごとに切り分け、それぞれの単語の出現回数と、2 つの単語が続けて出現する回数を計測し作成した。テキストデータは約 28 万単語からなり、のべ約 3,600 万単語で構成されていたが、そのうち 1 回しか出現しない単語を削除し、約 11 万単語の出現回数をスコアとする単語辞書を作成した。しかし、新聞のテキストだけでは単漢字などの語彙が少ないため、SKK¹²⁾ の辞書を用いて単語を補った。補った単語のスコアは 1 とした。また単語間関係辞書も同様に、あ

る2単語が連続して出現する回数が50回より少ない組合せを削除し、約10万組のデータを作成した。この場合もスコアは出現回数としている。

ユーザ辞書の単語は使用回数が多いほどスコアが高くなるようにし、使われないとスコアが徐々に低くなるようにしている。スコアが0になったユーザ辞書の単語は、静的辞書にエントリがある場合削除される。

5.3 入力エンジンの実装

入力エンジンは、子音列の入力と母音選択に対してマッチする単語を辞書から検索し、辞書の持つスコアをもとにソートした結果を出力する。辞書が大きいため、最初の数文字を使ったハッシュによって得られる単語リストから検索を行う。検索にはPOBox⁸⁾の検索アルゴリズムを利用した。

6. Popie の評価

Popie の評価として、1) 子音キーの配置の検討、2) 子音入力と母音選択による入力方式(子音 + 母音入力方式)とローマ字入力方式および子音入力方式との比較、3) ユーザによる入力実験を行った。

6.1 子音キーの配置の検討

各メニュー項目の位置における入力時間やペンの移動距離などの入力コストと各キーの使用頻度をもとに、最適な配置を総当りで計算した¹³⁾。その結果、現在のかな順の「AKSTNHMYRW」の配置に対して5%の速度向上が期待される配置「TWRSYMKHNA」が見つかった。しかし速度向上が5%と低いため、かな順の配置の覚えやすさを考慮すると、配置はかな順のままが妥当である。最適な配置とのコストの差が小さかった要因として考えられることは2点ある。1つは、キーを1つ入力するたびにペンがレストエリアに戻るため、コストが直前の入力にほとんど依存しないことである。もう1つは、かな順の配置がたまたまコストが低かったということである。最も悪い配置では、かな順よりも約20%コストが多いので、かな順が最適な配置に近い配置だったことが分かる。

6.2 子音入力と母音選択による入力方式の評価

日本語入力システム Popie の子音 + 母音入力方式を評価するため、まずローマ字入力方式と比較を行った。また、子音 + 母音入力方式と母音選択のない子音入力方式についても比較を行い、この際、予測・補完の機能の有無による比較も行った。比較はそれぞれの入力方式における操作回数によって行い、操作回数は計算機によるシミュレーションによって算出した。

6.2.1 自動入力プログラム

シミュレーションを行うため、各方式に対応した自

動入力プログラムを作成した。子音入力方式については、母音選択や補完・予測の機能のオン、オフをプログラムの実行時に与えられるようにした。入力するテキストデータは、CD-毎日新聞2001年度版¹⁰⁾の社会面の記事を用いた。入力データは未知語を含まないよう、辞書のコーパスに含まれるクローズテストのみで評価したが、日本語入力のためのモデルの評価ではなく、入力手法の間での比較であるため、実験結果への影響はないと考えられる。

シミュレーションでは子音キーの入力回数、母音キーの入力回数、候補選択の操作数を計測した。子音 + 母音入力方式では子音キーを10個と母音選択が5通りであり、ローマ字入力では、濁点や半濁点などを表現するため、全部で24個のキーを使った。

候補選択の操作数はスクロール操作を含む候補を選択するために要した操作数である。Popieの実装では上位3つの候補をスクロールせずに選択できるため、スクロール幅は3である。

Popie以外のインタフェースを仮定したシステムでは、選択可能数とスクロール幅が N の場合、候補選択の操作数は(Popieの候補選択の操作数-1) \times 3/ N +1で近似される。子音キーの入力回数、母音キーの入力回数は共通だと考えられるので、このシミュレーションは特にPopieに限ったものではない。

自動入力プログラムは単語ごとに入力を行い、これまでの経験から子音 + 母音入力方式では以下のような戦略を用いた。

- (1) 子音を1つ入力する。
- (2) 上位3つに目的の単語があれば選択し(2)に戻る。目的の単語がなく入力できる子音がある場合は(1)に戻る。
- (3) 母音を1つ入力する。
- (4) 上位3つに目的の単語があれば選択し(2)に戻る。目的の単語がなく選択できる母音がある場合は(3)に戻る。
- (5) スクロールする。
- (6) 上位3つに目的の単語があれば選択し(2)に戻る。目的の単語がなければ(5)に戻る。

ローマ字入力方式では、上記の戦略において子音と母音を同時に入力する。

6.2.2 シミュレーション1. 子音 + 母音入力方式とローマ字入力方式

子音 + 母音入力方式とローマ字入力方式それぞれのシミュレーションに、社会面の記事約3カ月分(約145万単語)を入力として用いた。このシミュレーションではどちらの入力方式についても、予測・補完機能

表 4 社会面の記事入力における単語あたりの操作数
Table 4 The number of operation per a word for inputting articles of general news.

項目	子音 + 母音入力方式	ローマ字入力方式
子音入力回数	2.054	1.650
母音入力回数	0.376	1.547
候補選択回数	1.341	1.370
操作数合計	3.772	4.567

表 5 社会面の記事入力における単語あたりの操作数
Table 5 The number of operation per a word for inputting articles of general news.

母音選択 予測・補完	あり あり	あり なし	なし あり	なし なし
子音入力回数	2.054	2.603	2.005	2.534
母音入力回数	0.376	0.378	0.000	0.000
候補選択回数	1.341	1.266	3.154	2.519
操作数合計	3.772	4.247	5.159	5.054

をありとしている。

表 4 にそれぞれの方式について、1 単語あたりの子音・母音入力回数、候補選択回数、および操作数の合計を示す。子音 + 母音入力方式では、ローマ字入力方式に比べて、子音入力回数が約 24%多いが、母音入力回数が約 75%も削減され、全体として約 18%の操作数を減らすことができています。候補選択回数はほぼ同じであった。

6.2.3 シミュレーション 2. 子音 + 母音入力方式の母音選択と予測・補完

Popie の子音 + 母音入力方式において、予測・補完と、母音選択の 2 つの機能をありとなしとした 4 通りをシミュレーションした。入力には社会面の記事約 3 カ月分 (約 145 万単語) を用いた。

表 5 にそれぞれの機能のあり、なしに対応する、1 単語あたりの子音・母音入力回数、候補選択回数、および操作数の合計を示した。両機能ともありの方式では両機能ともなしの方式に比べ、操作数が約 25%削減されている。また、予測・補完の機能のみありの方式や、母音選択の機能のみありの方式に比べてもそれぞれ、操作数が約 27%、約 11%削減されている。

予測・補完の機能のみありの方式では、両方なしの方式に比べて操作数が増えている。これは、予測・補完の機能により候補の数が増大し、スクロール操作が余計に増えてしまったことが原因であると考えられる。一方母音選択は、濁点などを含め 5 つ以上ある母音の可能性を 1 つに絞り込むことができるので、スクロール操作よりも有効な手段であることが分かる。また、母音選択と予測・補完の機能の組合せの相性が良く、

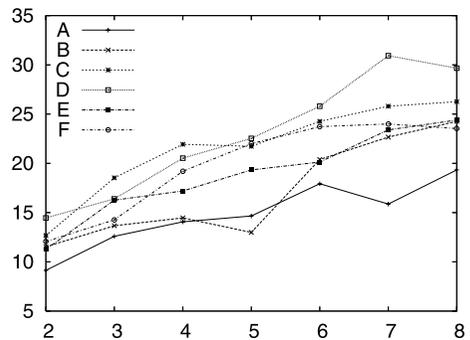


図 7 入力実験の結果：各セッションでの 1 分間の入力文字数
Fig. 7 Result of input experiment: input speed of characters per minute at each sessions.

相乗効果があったことが分かった。

6.3 ユーザによる入力実験

6 人の男性に対して Popie を使った文字入力の実験¹⁴⁾の結果を図 7 に示す。1 セッションの入力を 15 分とし、1 セッション目の練習と 2 セッション目以降の本番を合わせて 8 セッションを行った。セッション中は入力に集中してもらい、入力速度を 1 分間の入力文字数で測った。文字数はかな漢字をそれぞれ 1 文字としている。ユーザには毎回異なる文章 (20 ~ 40 文字程度) を提示し、間違えた入力は訂正してもらった。

実験の結果、入力速度は平均で 25 文字/分、最大で 31 文字/分の入力速度であり、手書き文字認識による入力とほぼ同じ程度の入力速度であった。実験では毎回異なる文章を入力してもらい、ユーザが入力文を覚えてしまったり、ユーザ辞書の学習によって極端に入力速度が上がらないようにしているため、実際に利用するにはこれ以上の入力速度が期待できる。またユーザはシステムを実験中に 2 時間入力を行っただけであり、習熟によってさらに入力速度が上がると考えられる。

6.3.1 入力実験における各操作の詳細

入力実験における各操作の詳細について分析を行った¹⁵⁾。図 8 は子音入力、候補選択、母音選択、その他のそれぞれの操作にかかる時間が占める割合を分析したグラフである。候補選択にはスクロールする時間を含んでいる。その他の操作とは削除やアンドゥ・リドゥなどの操作である。最後のセッションでは子音入力 59.4%、候補選択 30.6%、母音選択 4.2%であり、母音選択が占める時間は十分に少なく、多くの場合は子音入力だけで済んでいることが分かる。

図 9 はそれぞれの操作を 1 回行うのに要した時間の平均値を分析したグラフである。最後のセッションでは、子音キーを入力するのに 0.94 秒かかり、母音

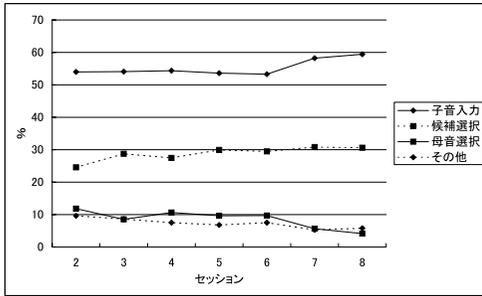


図 8 各操作が入力時間全体に占める割合

Fig. 8 Proportion of spent time for each operation.

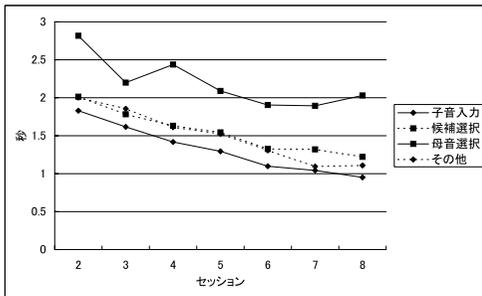


図 9 1つの操作を行うのに要した時間の平均

Fig. 9 Average of spent time for a operation.

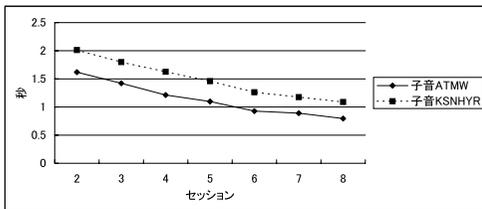


図 10 1つの子音入力操作を行うのに要した時間の平均

Fig. 10 Average of spent time for a operation of input consonant.

選択には約 2.2 倍の 2.03 秒かかっていた。Popie では中心からオクタント内側までの距離と、中心からオクタント外側までの距離が 2.3 倍であり、ペンの移動距離と入力時間の比はほぼ同じである。

図 10 は子音入力の操作を 1 回行うのに要した時間を、期待されるペンの移動距離によって分類した場合の平均値を示したグラフである。期待されるペンの移動距離は「ATMW」では半径の 2 倍であり、残りの「KSNHYR」では円弧の分だけ長い。最後のセッションにおいて「ATMW」では 0.78 秒、「KSNHYR」では 1.08 秒である。この比は 1 : 1.38 であり期待される移動距離の比 1 : 1.39 とほぼ同じである。このように Popie における操作ではペンの移動距離に比例して操作にかかる時間が増えるので、エラー率が増えない

範囲でインタフェースを小さくすることでより速く入力できると考えられる。

6.3.2 アンケート

実験後に行ったアンケートでは、Popie の良い点の記述として「慣れたら速く入力できるようになると思う」と半数のユーザが答えたが、Popie は使いやすいかという質問では「使いやすい」と「使いにくい」に分かれた。これは合計 2 時間の実験におけるインタフェースへの慣れの差だと考えられる。改善点についての記述では「子音キーが手で隠れてしまうので見えるようにしてほしい」、「候補選択で表示されている単語から直接選択したい」というような意見があった。

Popie では動的に変化する候補のリストを見えるようにするため、位置を覚えることができる子音キーを手で隠れてしまう部分に配置している。手で隠れる位置に子音キーを配置しなければよいが、手で隠れる領域には個人差もあり、フローメニュー全体を活用できない。そのためユーザが子音キーの位置を覚えるまでは、現在のインタフェースに加えて、手に隠れない位置に子音キーの配置を表示するなどの改善が考えられる。

また、実験初期に多かったエラーに、スクロールが必要な候補選択操作において、表示されている候補のリストの上にペンを移動させて選択しようとしたものがあり、このようなユーザの直感が「直接選択したい」という記述につながったと考えられる。この点、候補選択が必要なケースにおいて、Popie の候補選択と別のいくつかの候補選択の比較を行ったところ⁹⁾、候補リストに表示される単語の上でペンを画面から離す方法が最も速いことが分かった。しかし、ペンを離すことで決定する操作はフローメニューの操作体系ではないため、実際に子音入力とこの候補選択を組み合わせると使いやすいかどうかはさらに検証が必要である。

7. ま と め

本論文では、大画面ペン・コンピュータのための、日本語入力システム Popie について述べた。フローメニューの操作体系を拡張することで、フローメニュー上でキー入力と候補選択をシームレスに行うことが可能にし、さらに子音入力と母音選択による入力方式を用いて、フローメニューにおける日本語入力を実現した。

ユーザ実験による入力速度は平均で約 25 文字/分、最大で約 31 文字/分であり、手書き文字認識による入力とほぼ同じ程度の入力速度だが、予測入力の効果やシステムの習熟によってこれ以上の入力速度が期待で

きる。また、入力のシミュレーションによる評価から、本研究の入力方式では、母音をつねに入力し確定するローマ字入力方式と比較して操作が18%少なく、母音選択がない子音入力と比べても操作が27%少ないことが分かった。さらにユーザ実験における入力の分析結果から、子音入力が全体の約60%、候補選択が全体の約30%、母音選択が約4%の時間を占めることが分かった。これらのことから、母音選択の操作は非常に短時間で操作数を大きく削減する効果があったといえる。

今後の課題として、熟練ユーザの入力速度の調査や、初心者ユーザに対するキーの位置の提示方法の検討などがあげられる。

参 考 文 献

- 1) Hopkins, D.: The design and implementation of pie menus, *Dr. Dobb's Journal*, Vol.16, No.12, pp.16–26 (1991).
- 2) Kurtenbach, G. and Buxton, W.: The limits of expert performance using hierarchic marking menus, *CHI '93: Proc. SIGCHI conference on Human factors in computing systems*, pp.482–487, ACM Press (1993).
- 3) Guimbretière, F. and Winograd, T.: FlowMenu: Combining command, text, and data entry, *Proc. ACM User Interface Software and Technology 2000 (UIST 2000)*, pp.213–216 (2000).
- 4) Callahan, J., Hopkins, D., Weiser, M. and Shneiderman, B.: An empirical comparison of pie vs. linear menus, *CHI '88: Proc. SIGCHI conference on Human factors in computing systems*, pp.95–100, ACM Press (1988).
- 5) Ren, X. and Moriya, S.: Improving selection performance on pen-based systems: a study of pen-based interaction for selection tasks, *ACM Trans. Comput.-Hum. Interact.*, Vol.7, No.3, pp.384–416 (2000).
- 6) Guimbretière, F., Martin, A. and Winograd, T.: Measuring FlowMenu Performance, Technical report, Stanford CS (2001). CS-TR-2001-02.
- 7) 田中久美子, 犬塚祐介, 武市正人: 携帯電話における日本語入力—子音だけで日本語が入力できるか, *情報処理学会論文誌*, Vol.43, No.10, pp.3087–3096 (2001).
- 8) 増井俊之: ペンを用いた高速文章入力手法, *Proc. Workshop on Interactive Systems and software 1997 (WISS 1997)*, 日本ソフトウェア科学会, pp.51–60, 近代科学社 (1997).
- 9) 佐藤大介, 志築文太郎, 田中二郎: メニュー選択に基づく子音による日本語入力手法 Popie の候補選択インタフェースの検討, *情報処理学会研究報告 2004-HI-108*, pp.39–46 (2004).
- 10) 毎日新聞社: CD-毎日新聞 2001 年度版 (2001).
- 11) 松本裕治, 北内 啓, 山下達雄, 平野善隆, 松田寛, 高岡一馬, 浅原正幸: 日本語形態素解析システム『茶釜』version 2.2.1 使用説明書, 奈良先端科学技術大学院大学 (2000).
- 12) SKK Openlab: SKK. <http://openlab.jp/skk/>
- 13) Sato, D., Miura, M., Shizuki, B. and Tanaka, J.: Menu-selection-based Japanese Input Method with Consonants for Pen-based Computers, *Proc. 6th Asia-Pacific Conference on Computer-Human Interaction (APCHI2004)*, LNCS3101, pp.399–408 (2004).
- 14) 佐藤大介, 三浦元喜, 志築文太郎, 田中二郎: ペンによるメニュー選択に基づく日本語入力手法, *日本ソフトウェア科学会第 20 回大会論文集* (2003).
- 15) 佐藤大介, 志築文太郎, 田中二郎: ペンの周囲で操作することを可能にするインタフェース, *Proc. Workshop on Interactive Systems and software 2004 (WISS 2004)*, pp.71–76 (2004).

(平成 17 年 6 月 16 日受付)

(平成 18 年 4 月 4 日採録)



佐藤 大介 (正会員)

1980 年生。2003 年筑波大学第三学群情報学類卒業。2005 年筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻修士課程修了。現在、日本アイ・ピー・エム東京基礎研究所に勤務。ヒューマンインタフェース、アクセシビリティに興味を持つ。日本ソフトウェア科学会会員。



志築文太郎 (正会員)

1971 年生。1994 年東京工業大学理学部情報科学科卒業。2000 年東京工業大学大学院情報理工学研究科数理・計算科学専攻博士課程単位取得退学。博士(理学)。現在、筑波大学システム情報工学研究科コンピュータサイエンス専攻講師。ヒューマンインタフェースに関する研究に興味を持つ。日本ソフトウェア科学会, ACM, IEEE Computer Society, 電子情報通信学会, ヒューマンインタフェース学会各会員。



三浦 元喜 (正会員)

1974年生。1997年筑波大学第三学群情報学類卒業。2001年筑波大学大学院工学研究科博士課程修了。博士(工学)。同年筑波大学電子・情報工学系助手。2004年より北陸先端科学技術大学院大学知識科学研究科助手。現在に至る。グループウェア, ヒューマンインタフェース, 教育支援システムに興味を持つ。日本ソフトウェア科学会, ACM, 人工知能学会, ヒューマンインタフェース学会, 日本教育工学会各会員。

端科学技術大学院大学知識科学研究科助手。現在に至る。グループウェア, ヒューマンインタフェース, 教育支援システムに興味を持つ。日本ソフトウェア科学会, ACM, 人工知能学会, ヒューマンインタフェース学会, 日本教育工学会各会員。



田中 二郎 (正会員)

1975年東京大学理学部卒業。1977年東京大学大学院理学系研究科修士課程修了。1984年米国ユタ大学計算機科学科博士課程修了, Ph.D. in Computer Science。1984年から

(財)新世代コンピュータ技術開発機構で第五世代コンピュータ核言語の研究開発に従事。1993年より筑波大学に勤務。現在, システム情報工学研究科教授, システム情報工学研究科コンピュータサイエンス専攻長および電子・情報工学系長。2004年より筑波大学先端学際領域研究センター(TARAセンター)において「ユビキタス環境における情報提示・操作技術」プロジェクト研究代表者。研究分野としてはプログラミング言語やヒューマンインタフェースに興味を持つ。ACM, IEEE Computer Society, 日本ソフトウェア科学会, 電子情報通信学会, 人工知能学会, ヒューマンインタフェース学会, 計測自動制御学会各会員。CACM日本語版編集長。