ヒューマンインタフェース 41−20
（1992. 3. 3）

認知機構の汎用シミュレータによるキー配列習得のモデル化

下郡信宏、木村泉
東京工業大学 理学部

AndersonのACT*理論に基づく認知モデルと、Rumelhartらの熟練者のキーボード操作モデルを融合させた仕組みを用いてキー配列を習得したばかりの初心者から熟練者までを一貫して対象とすることのできるキーボード操作モデルを作成した。また、キーボード初心者3名及びQWERTY配列熟練者2名に新しいキーボード配列を習得させる実験を行なった。実験結果を解析し、打鍵間隔の推移、キー配列間の干渉、英文入力と日本文ローマ字入力の干渉の違いなどの特徴を観察した。作成したキーボード操作モデルによってこれらの特徴を解釈し、簡単なシミュレーションを別稿の大野らのシミュレータを用いて行ない、モデルによりこれらの特徴が再現可能であることを示した。

Modeling novice keyboard acquisition by an ACT* simulator

Nobuhiro SHIMOGORI and Izumi KIMURA

Tokyo Institute of Technology, Department of Information Science
2-12-1 Ookayama, Meguro-ku, Tokyo 152 JAPAN

A novice keyboard acquisition model which can deal with the typing process ranging from beginners who has just started blind typing, to a skilled typist is described. This model is based on a conjunction of Anderson's ACT* theory: a human cognitive model, and Rumelhart's ATS system: a model of a skilled typist. Using five subjects (three completely novice typist and two expert QWERTY typist), an experiment to learn a new keyboard arrangement was run. Changes of keystroke interval, interference between two keyboard arangements, and difference of interference between English and Japanese text entry were observed. We explain these characteristics by our model. A simple simulation on a simulator described in a companion paper by Ohno et al. was also done to show these characterristics are reproducable.

1.まえがき

人がキーボードを操作する様子を理解することは、利用者インタフェースを考える際に重要であるが、まだ、充分には解明されていない。そのために、キー配列の善し悪しなどを論ずると、どうしても議論が主観に偏りがちである。そこで、人がキーボードを操作する際に観察される打鍵間隔やエラーなどの様々な特徴を理解するためにキーボード操作のモデル化を行なった。

キーボード操作をモデル化した例としてはRumelhartら[1]のATSシステムがあるが、これは熟練者を対象としたものであり、習得途上のタイピストは対象としていない。そこで、このATSシステムと人の認知モデルである、Anderson[2]のACT*理論を融合した仕組みを用いることにより、キー配列を習得したばかりの初心者から熟練者までを一貫して対象とすることのできるモデルを作成した。

さらに、被験者5名にキー配列を習得させる実験を行ない、観察されたいくつかの特徴をこのモデルにより解釈した。

2.キー配列習得実験

まず、被験者がこれまでに使ったことのないキー配列を習得する実験を行なった。

被験者: 被験者はキーボード初心者3名(被験者A,B,C)、QWERTY配列熟練者2名(被験者D,E)。被験者A,Bには謝礼を支払った。他はボランティアである。キーボード初心者はキーボードを一本指でいじったことがあったり、ゲームをするためにテンキーを使ったりした、という程度の使用経験はあるが、それ以上の経験はない。QWERTY配列熟練者は両被験者とも計算機系学科の学生で、4年間以上、QWERTY配列のキーボードを日常的にブラインドタッチで使用している。ただし、両被験者とも、QWERTY配列以外のキー配列の使用経験は全くない。

作業: SKY配列[3]に基づき、多少の改良を加えたキー配列[4](ここではSKY!配列と呼ぶ)で日本文を入力する。SKY!配列を図1に示す。

はじめに、基本練習として増田法[5]に準じたキー配列習得テキストを使って、キーの大体の位置を覚える。この練習に要した時間は各被験者により異なるが、2〜4時間で、全ての被験者はこの基本練習を終えた。基本練習を終えた段階で、被験者はポツポツとではあるがブラインドタッチができるようになった。

次に、「実力テスト」と呼ぶ実験を行なう。実力テストでは、被験者は与えられた特異でない日本文を漢字変換せずに、ひらがなの読みだけを入力する。実験開始前に5分間のウォームアップセッションを行ない、これに続いて、20分間の本番の実験を行なう。

その上で、「10時間練習」と呼ぶ練習を行なう。10時間練習では、被験者は、ふだん読み書きしている文章に近い、特異でない日本文の本を一冊選び、この本を、漢字変換を行ないながら、清書タイプする。ただし、用字の詳細や送りがなは被験者の好みに合わせてよいものとした。この作業は被験者の都合のよい時に、都合のよい時間だけ行なう。これを作業時間の累計が10時間を越えるまで行なう。実力テストと10時間練習を、10時間練習の総時間が50時間に達するまで繰り返す

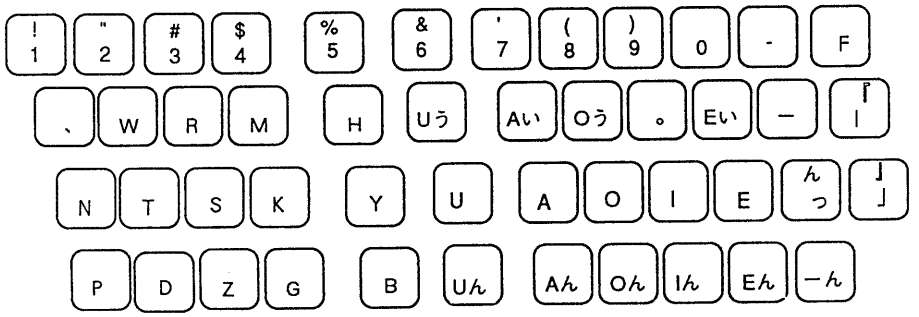


図1:SKY!配列

練習が50時間分終了した後の実力テストで被験者が入力した文字数は、平均で1秒あたり1.6文字(ひらがな)であった。被験者間に差はさほどなかった。10時間練習の最後のセッションでの入力文字数は、平均で1秒あたり0.86文字(漢字かなまじり)であるが、各被験者により0.75～1.00文字/秒とバラつきは大きかった。

QWERTY配列熟練者はこれ以外に、実力テストのついでに実力テストと同じ形式で、時々QWERTY配列でのローマ字入力や英文入力を行なった。QWERTY配列熟練者には、実験期間中は極力QWERTY配列を使用しないよう注意した。

環境: 実力テストはPC-9801RXとワープロソフト「新松」を用いた。10時間練習は大部分を被験者の自宅で行なったため用いたPC-9801シリーズの機種は被験者により異なっている。ただし、全機種PC-9801VM以降のものを用いて、ワープロソフトの辞書はRAMディスク、またはハードディスク上に置いた。

実験、および練習中のすべての打鍵は打鍵収集ツールBMKS[6]により記録した。また、BMKSにはキー配列を変更する機能が備わっているため、これを用いてSKY!配列を実現した。

実力テストは個室で行ない、実験中は被験者だけにした。実験中は被験者の手の動きをビデオカメラで記録した。

椅子、机、キーボードやディスプレイの位置などの環境は被験者の作業しやすい状態に調整してもらった。

3.キーボード操作モデル

キーボード操作モデルは、人の認知モデルであるACT*理論と熟練者のキーボード操作モデルであるATSシステムを融合させることにより作成した。ACT*理論だけでは人の手の特性などが充分に扱えない。一方、ATSシステムだけでは人の学習などの仕組みが扱えない。深沢[7]、村本[8]は、これら2つのモデルを融合することにより、初心者から熟練者までを扱えるモデルの作成を提案した。深沢はATSシステムの認知的役割を果たしているスキーマシステムの一部をACT*シミュレータと置き換えられる形の手のシステムを計算機上に実現し、村本はACT*シミュレータの原型を作成したが、これら二つのシステムは接続されていなかった。大野ら[9]がACT*シミュレータを改良し、これにともない両システムの接続が可能となった。キーボード操作モデルはこのACT*シミュレータの動作を記述するプロダクションにより作られている。なお、モデルの設計に際しては、ATSシステムは熟練者のキーボード操作モデルとしては完成度の高いモデルであるため、プロダクションを充分に学習させた後にはATSシステムの枠組内に収束するように注意した。

モデルの作成に際しては以下の仮定をおいた。

1)基礎練習を終了した時点からのモデルとする
2)入力する文字列は階層化されて取り込まれる
3)ブラインドタッチを行ない、
　手はホームポジションを中心に動かす
4)人は打鍵中には打鍵以外のことは考えない
5)エラーの修正は扱わない

これより、実験で観察されたいくつかの現象、および木村[4]により報告されているキー配列の干渉という現象を説明しながら、モデルの構造とこれらの現象をモデルではどの様にとらえるかを説明する。

4.観察された事象とモデルによる解釈

4.1. 打鍵間隔の推移

被験者Aの実力テストでの打鍵間隔の一部を図2に示す。基本練習終了直後の打鍵はぽつぽつと行なわれ、一部、ひらがな内の子音と母音の打鍵間隔が短くなり始めているのが分かる。練習10時間終了後の打鍵では、ひらがな内の子音と母音間の打鍵間隔とひらがな間(母音から次のひらがなの子音まで)の打鍵間隔は明らかに異なる。次の練習30時間終了後の打鍵では、いくつかのひらがながかたまりで入力されはじめているのが分かる。

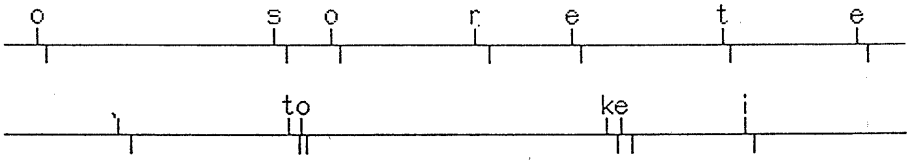
モデルではこれを以下のように説明する。

まず、入力すべき文は階層化された時系列という形で取り込まれる。「雪子は. . . 」という文を入力する場合を考えると、これは、

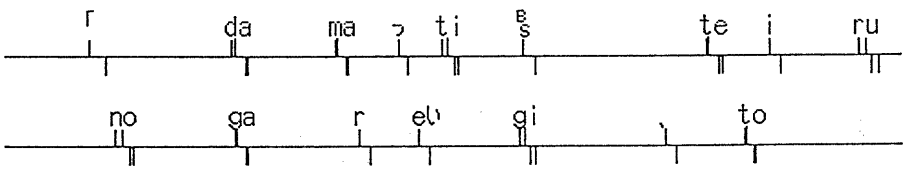
{{{{ゆ,き},{こ}},は},{. . . }}

という形になる。はじめのプロダクションはこの時系列型の文字列から再帰的に要素を取り出し、要素がひらがなになるまで繰り返す。その後、ひらがなをローマ字に分解し、ローマ字列から英字を一字抜き出し、これを打鍵することを次々とゴ

基本練習終了直後



10時間練習後



30時間練習後

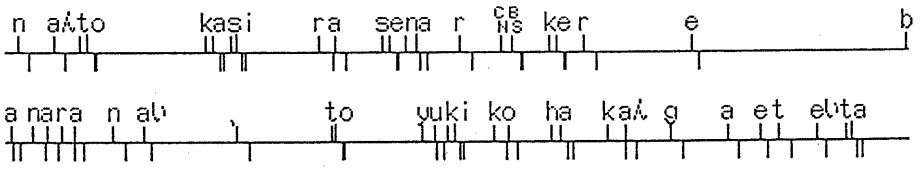


図2:被験者Aの「実力テスト」の様子(一つの軸は10秒)

ールに設定してゆく。

プロダクションの発火の流れは以下の様になる。箇条書きにした各要素がそれぞれプロダクション・インスタンスに対応する。

・文を入力する({{{{ゆ,き},{こ}},は},{. . . }})
・かたまりを入力する({{{ゆ,き},{こ}})
・かたまりを入力する({ゆ,き})
・ひらがなを入力する(ゆ)
・ローマ字を入力する({y,u})
・英字を打鍵する(y)

学習を行なっていない状態のプロダクションでは、括弧内の各要素は変数になっている。

まず、英字を打鍵するプロダクションにおいて、ACT*理論の学習機構によりプロダクションの手続き化が起こり、'y'を打鍵する専用のプロダクションが作られる。英字を打鍵するプロダクションではじめに学習が起こる理由は一般的にひらがな「ゆ」や、かたまり「ゆき」を打鍵するプロダクションの使用頻度よりも英字'y'を打鍵する頻度が高いためである。ただし、実験やシミュレーションで「ゆきこ」だけを何度も練習させた場合はこの限りではない。同様にプロダクションの合成も使用頻度の高い、下位のプロダクションから起こりやすい。したがって、まず、ローマ字を入力するゴールから、英字を打鍵するゴールへサブゴールをたてずに直接英字を打鍵するプロダクションが生成される。さらには、ローマ字を思い出さなくても、ひらがなから直接打鍵できるプロダクションが生成される。この状態が被験者Aの練習10時間終了後の状態と考えることができる。また、プロダクションの初期状態が基本練習終了直後の状態を示し、手続き化と合成が、「ゆきこ」をかたまりのまま入力するレベルまで終了した状態を練習30時間終了後の状態であると、このモデルでは解釈する。また、キーボードの熟練者とは、このようなかたまりで入力するプロダクションを多く持っている人であると考えることもできる。

4.2. キー配列間の干渉

複数のキー配列を習得するとキー配列間で干渉が起きることが報告されている。特に、2つの配列で同じ文字が近い位置に配置されている場合に干渉が起こりやすい。例えば、SKY!配列の'r'は左

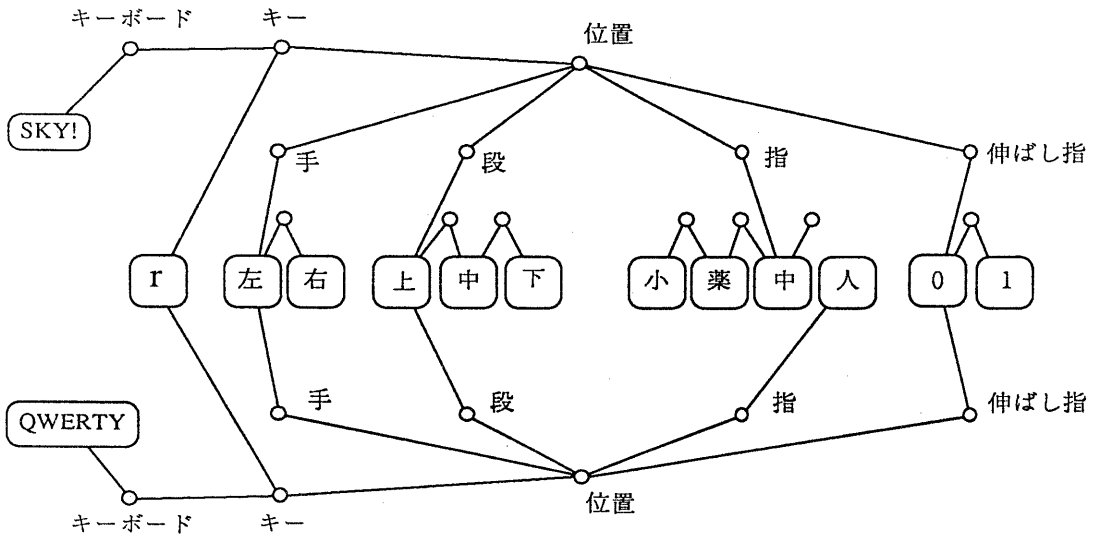


図3:キーの位置を表す知識ネットワーク

手、ホームポジションより一つ上の段の中指の位置に配置されている。一方、QWERTY配列の'r'は、隣の人指し指の位置に配置されている。モデルでは打鍵とは手の情報、段の情報、指の情報、伸ばし指の情報を手のシステムに対して送った後に打鍵命令、ホーム・ポジションに戻る命令を送ることと考えている。手、段、指の情報は、SKY!配列の'r'の場合、左手、上段、中指となる。

伸ばし指とは、ホームポジションから横に指をずらして打つキー(QWERTY配列の'h'など)のことである。伸ばし指に関しては、0、または1を情報として送る。

SKY!配列とQWERTY配列を習得した人のキーに関する宣言的知識のネットワークは図3のように表される。SKY!配列で'r'と打鍵する場合、ノード[SKY!]と[r]が活性化されている。さらに、各リンクを伝わり、手(左)、段(上)、指(中)、伸ばし指(0)が活性化され、作業記憶に入る。

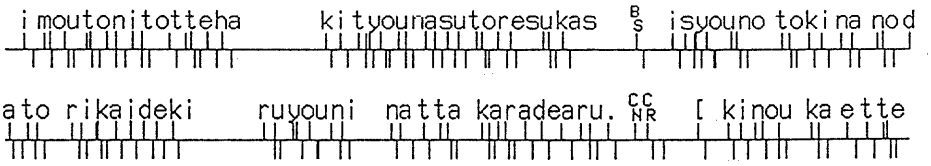
英文字を打鍵するプロダクションは以下の6つのサブゴールをたてる: 手、段、指、伸ばし指の情報を特定し、手のシステムに送るためのサブゴールと、たった今送ったキーに対する打鍵許可を送るためのサブゴール、打鍵が終了したらホームポジションへ手を戻すという命令を送るためのサブゴール。

手の情報を特定し、手のシステムに送るプロダクションでは、作業記憶内にある手の情報を取り出し、これを手のシステムに送る。ここでは、手(左)が作業記憶内にあるため、[左]を手のシステムに対して送る。段、指、伸ばし指に関しても同様に処理される。SKY!配列しか知らない場合はこれで問題がないが、QWERTY配列を知っている場合、QWERTY側のリンクを伝わって活性が伝搬する可能性がある。つまり、指(中)と指(人)が同時に作業記憶に現れる可能性がある。中指と人指し指の様にキーが隣どうしの場合は、これらのノード間にこれらの指は隣合っているという情報を与えるリンクが張ってあるため、さらに活性が伝搬しやすく、ノイズにより、活性化の度合が逆転してしまう可能性が増す。学習が余り進んでいない状態ではこの様に情報を取り出す際にエラーを引き起こす。

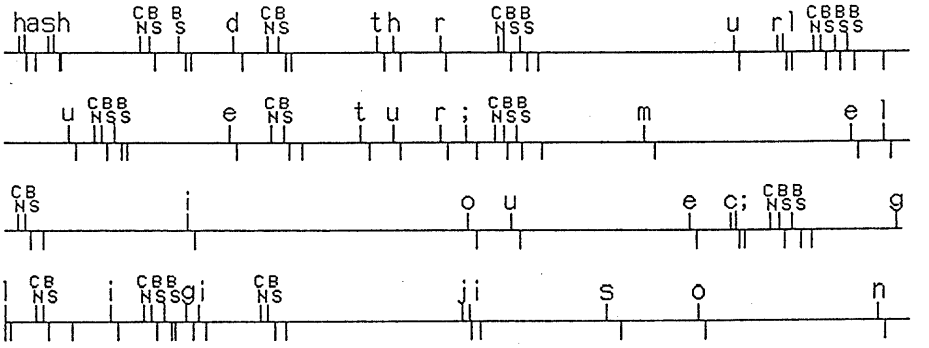
手続き化や合成が進んだ状態のプロダクションを考えると、

```
SKY!用の'r'打鍵プロダクション
  if    goal(type(r))
        and 手(左)
        and 段(上)
        and 指(中)
        and 伸ばし指(0)
  then ....
```

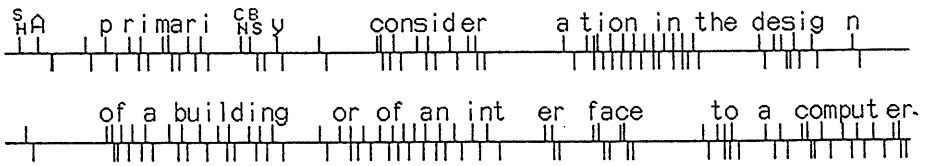
SKY!配列習得前:日本文ローマ字入力



SKY!配列習得後:日本文ローマ字入力



SKY!配列習得前:英文入力



SKY!配列習得後:英文入力

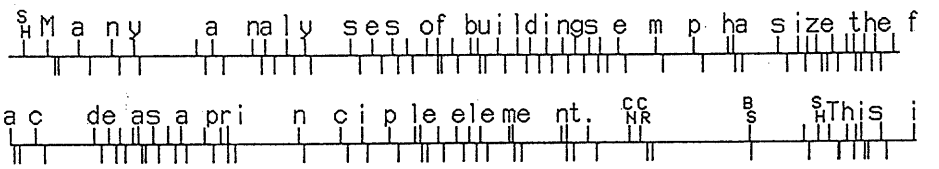


図4:被験者DがQWERTY配列で入力する様子(一つの軸は10秒)

```
  end
QWERTY用の'r'打鍵プロダクション
  if    goal(type(r))
        and 手(左)
        and 段(上)
        and 指(人)
        and 伸ばし指(0)
  then ....
  end
```

となる。唯一異なる条件が指の指定である。先に述べたのと同じ理由により、この条件の活性はノイズにより逆転しやすい状態にある。したがって、かなり学習が進んでもキー配列間の干渉が起こることが分かる。

SKY!配列とQWERTY配列の'n'のキーの位置には重複する情報がない。ここで示した'r'のプロダクションと'n'を打鍵するプロダクションを同じ条件で学習を止めたACT*シミュレータで実行し、'r'および'n'を15回ずつSKY!配列で打鍵させた。'r'を打鍵する時には4回間違えてQWERTY配列の'r'の位置で打鍵してしまったが、'n'を打鍵する時には1回だけ、QWERTY配列の'n'の位置で打鍵してしまった。ノイズなどのパラメータは、適当な値を用いたため、4:1という比率には特に意味はない。'r'の方が'n'よりも干渉が起こりやすいという結果のみに意味がある。

4.3. 日本文入力と英文入力の干渉の違い

図4は被験者DがSKY!配列習得前と習得後にQWERTY配列で、それぞれ日本文ローマ字入力と英文入力を行なっている様子である。日本文ローマ字入力はSKY!配列習得前とSKY!配列習得後では打鍵間隔に大きな差が現れていることが容易に確認される。ところが、英文入力ではSKY!配列習得前後で打鍵間隔に大きな変化は見られない。この現象は被験者Eでも観察されている。

常識的に考えると日本文を入力する場合にSKY!配列の悪影響が出た場合、英文入力でも同様の悪影響が出ても不思議ではない。なぜ、このような一見不思議な現象が起こるのか、モデルを用いて考えてみる。

被験者D,EはQWERTY配列で日本文、英文、プログラミングを4年間以上行なっていたため、どの領域でも中程度以上の熟練者であった。したがって、先に述べた、日本文や英文のかたまりを直接入力するプロダクションを相当数持っていたとする。また、ひらがなを直接、打鍵するプロダクションも持っていたと考える。

ここで、ひらがな「へ」と英単語"he"を入力することを例に考えると、SKY!配列習得前は、以下のQWERTY配列でひらがな「へ」を入力するプロダクション、および英単語"he"を入力するプロダクションを持っていた。

```
QWERTY配列で「へ」と打つプロダクション
  if goal(type_ひらがな(へ))
          ...
  then 【QWERTY配列でのキーの位置】
  end

QWERTY配列で"he"と打つプロダクション
  if goal(type_word({h,e})
          ...
  then 【QWERTY配列でのキーの位置】
  end
```

SKY!配列を習得すると、SKY!配列でひらがな「へ」を入力するプロダクションが同じゴールで生成され、しかも、被験者には実験期間中はQWERTY配列を極力使わないように指示してあるため、ACT*理論の学習機構の一つ、プロダクションの強化により、QWERTY配列用のひらがな「へ」を入力するプロダクションはSKY!配列用のプロダクションに乗っ取られてしまう。

```
SKY!配列で「へ」と打つプロダクション
  if goal(type_ひらがな(へ))
          ...
  then 【SKY!配列でのキーの位置】
  end
```

そこで、QWERTY配列で日本文ローマ字入力を行なおうとすると、SKY!配列用のプロダクションが強さの点で優位にあるために、発火してしまい、QWERTY配列ではまともに入力できなくなる。ところが、SKY!配列は日本文入力専用のキー配列であり、実験期間中、SKY!配列で英文を入力していないため、SKY!配列用の"he"を入力するプロダクションはできていない。また、QWERTY用の"hc"を入力するプロダクションも弱められていない。したがって、英文は従来通りに入力することができる、と解釈することが可能である。

上のプロダクションをACT*シミュレータで以下

の順序で実行した。

1)QWERTY配列用の「へ」と"he"の
　プロダクションだけのセット

2)上のものに、SKY!配列用の「へ」の
　プロダクションを加えたもの
　ただし、SKY!配列用のプロダクションの強さを
　QWERTY配列用のプロダクションの倍の強さに
　した

シミュレーションの結果、1)ではQWERTY配列で正しく「へ」および"he"と入力できた。2)では「へ」は「g;」となってしまったが、"he"は正しく入力できた。

5. 問題点

本来プロダクションには条件数に限界がある。この限界とは条件の数が多くなり過ぎるとすべての条件が同時に思い出せない(作業記憶に入り切らない)ために起こるものである。現在の学習機構のままでは、「ゆきこ」などのかたまりを直接入力するプロダクションが生成される前にこの限界に達してしまうことが予想される。そこで、ACT*シミュレータの学習機構に新たな機構を加えることを考えている。それは、部分マッチでプロダクションが発火した場合、マッチしなかった条件をプロダクションから取り除いてしまうというものだが、果たして、この機構が認知心理学的に正しい機構であるかの検証を行なわなければならない。

6. 今後の課題

今回のシミュレーションでは実際のキーボード操作モデルの一部を取り出して、各事象が定性的に説明できることを示した。

現在のところ、フルスペックでのシミュレーションも動きはじめており、「ほんじつはせいてんなり」程度の文章は打ち始めているが、本格的なシミュレーションを行なうためには、さらにシミュレータのパラメータ等の調整が必要である。今後はこれらの作業を済ませ、実際に被験者が練習した環境に即した形でのシミュレーションを行ない、実験で得たデータとシミュレーションにより得たデータとの定量的な比較が必要となる。

謝辞

本研究に関し、文部省科学研究費補助金重点領域研究(1)「高機能高品質ソフトウェアの評価法の研究」（研究代表者牛島和夫）による補助を受けた。

参考文献

1. David E. Rumelhart, Donald A. Norman: Simulating a Skilled Typist: A Study of Skilled CognitiveMotor Performance, Cognitive Science 6, 1-36, 1982.

2. John R. Anderson: The Architecture of Cognition, Harvard University Press, Cambridge, Mass., 1983.

3. 白鳥 嘉勇、小林 史彦: 日本語入力用新キー配列とその操作性評価、情報処理学会論文誌、28巻6号、pp. 658-667、1987

4. 木村 泉: QWERTYローマ字打ちとSKY配列の相互干渉、情報処理学会研究報告、ヒューマンインタフェース研究会、1991.3.7-8

5. 増田 忠: キーボードを3時間でマスターする法、日本経済新聞社、東京、1987

6. 森川 治、木村 泉、粕川 正充: パソコン用打鍵データ収集システム,情報処理学会論文誌、31巻12号、pp. 1822-1931、1990

7. 村本 栄治: プロダクションシステムによるキーボード学習のシミュレーションモデル、東京工業大学修士論文、1991

8. 深沢 安伸: 日本語文書入力における打鍵動作のシミュレーション、東京工業大学修士論文、1991

9. 大野 健彦、乗松 敏雄、木村 泉: 認知機構の汎用シミュレータ、情報処理学会研究報告、ヒューマンインタフェース研究会、1992.3(本号)