

## 知覚的解釈を促すノートツール

中村美恵子<sup>†</sup> 宮下芳明<sup>†</sup>

本稿では図形や色などの視覚的な情報が知覚的に解釈されることを利用し、後から見返した時でも印象に残る講義ノートを作成できるノートツールの開発を行った。講義ノートはノート PC でとることが多いため、操作が煩雑となるポインティングデバイスを使用することなく、キーボード操作のみでテキストの入力とその装飾まで行えるようにした。そのために、テキストボックスの自動生成機能と装飾コマンド実行のための「どつき実行」機能を実装した。

### A Note Tool Stimulating Perceptual Understanding

Mieko Nakamura<sup>†</sup> and Homei Miyashita<sup>†</sup>

We propose a note tool for making lecture notes comprehensive and memorable by adding visual embellishments on normal texts. To use this software we do not need to use any pointing devices and can make lecture notes quickly. This software has two features. One is a function to generate textboxes automatically. Another is a function to execute command to add visual embellishments by hitting PC.

## 1. はじめに

知覚的解釈とは、図形や色などの視覚的な情報が、内的表現に置き換えられることなく外的表現として直接知覚的に理解されることを言う。これに対して、言語などの記号的な情報が認知的プロセスを経て内的表現に置き換えられ処理されることを認知的解釈と言う[1]。知覚的な解釈と認知的な解釈では情報の解釈の仕方が違うと考えられる。

近年、チャートに装飾を施した chart junk についての議論が活発であり、Scott[2]らは普通のグラフと装飾的なチャートについて評価実験を行っている。増加量を示す普通のチャートに対して、モンスターの歯の大きさで増加量を示し、支出金の増加の恐ろしさを訴えるようなチャートでは、短期記憶には差がなくても長期的には後者のほうが記憶に残ることが示されている。デザイナー向けの教則本[3]には、デザイナーの大きな目的はコミュニケーションであると述べられている。デザイナーはビジュアルによる「話法」や「文法」を習得することが求められており、ビジュアルには人の心を動かす力があるとも述べられている。

文字の羅列になることの多い講義ノートでも、視覚的な変化や装飾を加えることができれば、もっと印象に残り、後から見返したときにも、その講義の内容を思い返しやすいノートにすることができると考えられる。プロのデザイナーがデザインしたポスターや紙面には及ばないにしても、ノートをとるとき言葉以外の情報を記述したノートは、見返した時の効果も大きいと考えられる。本稿ではまずこの仮説のもと予備実験を行った。その結果、視覚的な変化を加えたノートのほうが、記憶に残りやすいという効果が検証された。

本稿ではさらに、簡単に視覚的な変化を加えることのできるノートツールの開発を行った。提案システムでは講義中でも使いやすいように、ポインティングデバイスを使うことなくキー入力のみで、テキストの入力と装飾両方が行えるようにしている。その為に、テキストボックスの自動生成機能と装飾コマンド実行のための「どつき実行」機能を実装した。「どつき実行」とは、コマンドを実行する際に PC を叩く（どつく）ことで実行する機能である。

## 2. 予備実験

テキストエディタなどでとったノートは文字が整然と並んでいるため、後で見返したときに、どこがポイントだったのか、どこが自分や講演者が強調したいところだったのかなどが、思い出せないことが多い。手描きのノートであれば、内容のまとめ

<sup>†</sup> 明治大学 理工学研究科 新領域創造専攻 デジタルコンテンツ系  
Program in Digital Contents Studies, Program in Frontier Science and Innovation, Graduate School of Science and Technology, Meiji University



ことがわかった。講義の内容も講義者独特の関連付けや言葉の使い回しがあり、既習の知識で回答することは困難であった。その結果、確認テストでの得点はどちらのグループもあまり高くなかった。確認テストの最後の自由記述欄に書かれた感想では、どちらのノートについてもわかりづらいというような否定的な意見が多かった。また、ノート B で色づけされたところが、設問には直接関係しておらず、B グループの被験者はかえって戸惑いがあるようにも思われた。

しかし、確認テストの採点結果では、グループ B の得点が、グループ A の得点よりも、平均点で 3 点高く、その結果 t 検定 5% の有意差が示された。これは、被験者たちのテストを受けた後の感想とは全く異なる結果であり、一見雑然としているように見えても、意味のまとまりごとにテキストブロックがわかれ、それらが自由な配置で書かれたノートを見た時の学習効果の優位性を示すものだと思う。

この予備実験では、ノートをとった時の記憶がノートを見返すときにどれくらいよみがえるかを検証するには至っていない。しかし、テキストを空間的に配することで記憶に残りやすいノートになるという仮説を裏付けるひとつの根拠になると考えられる。

この予備実験で作成したノート B のようなノートは、パワーポイントやイラストレータなどのソフトを使って作成することができる。しかし、それらのソフトウェアは講義を聞きながら使用するには使いにくく、実際にあまり使われていない。そこで、講義を受けながらノート B のようなノートが簡単に作成できるノートツールを開発した。

### 3. デザインコンセプト

今回作成するのは、印象的な講義ノートをノート PC でとるためのノートツールである。つまり、今回は予備実験で作成したノート B のようなノートが講義を受けながら素早く作成できることを目的としている。一般的なワープロソフトやドロー系ソフトを使って、このようなノートを作成することは可能だが、それらのツールは概して高機能であり、そのために操作の選択肢が多く、装飾のための操作に時間がかかってしまう。講義ノートツールに最も必要な機能は素早くテキスト入力が行えることであり、テキストの装飾を行うことで思考の流れを妨げてしまえば本末転倒となる。講義ノートをとるのにテキストエディタを使うひとが多いのもそこにひとつの理由があると思われる。そこで、テキスト入力に対しては余分な操作を必要とせず、かつ自由な位置にテキストを配置・装飾出来るように次の機能を実装することにした。

- テキストボックス自動配置
- コマンドの「どつき実行」

テキストを自由な位置に配置するためにはテキストボックスの機能が必要であるが、テキストを素早く入力できるようにするために、テキストボックスの位置や大きさの指定、入力終了を示す操作を不必要にした。提案システムでは、常にテキストボックスが表示されており、ひとつのテキストボックスの入力を終了したときは、自動的に次のテキストボックスが生成される。その位置のコントロールは、ショートカットキーで指定する。

また、講義中の教室でのノート PC の使用状況を考えた場合、マウスは使いづらく、素早く操作を行うためにも、マウスなどのポインティングデバイスの使用は極力少なくしたい。そこで、提案システムではポインティングデバイスなしで操作できるようなインタフェースを実装した。それが、「どつき実行」である。「どつき実行」とはコマンドを実行するためのトリガとして、PC を叩く（どつく）操作をさす。

テキストの装飾は、その機能を限定したうえで、コマンドで実行するようにした。コマンドは通常のテキストと同じようにテキストボックスに入力する。入力した内容が、コマンドであるということを PC に教えるために「どつき実行」を行う。実際にはマイクからの音を拾うことにより、叩かれたことを検知する。

装飾の種類は、講義ノートに必要なと思われるもの限定し、次のようなものにした。

- ・ フォントサイズの変更
- ・ テキストブロックの回転（斜め書き）
- ・ テキストブロックを囲む楕円
- ・ 矢印やカッコによるテキストブロックの関連付け
- ・ ラインマーカ機能

提案システムで目指すのは清書用のワープロソフトではない。機能を限定しその中で印象的なノートを作成することは可能であり、その方が講義中という状況での使用に適していると考えた。

### 4. システム概要

前章で述べたコンセプトのもと、本稿ではシステムの実装を行った。実装には Adobe AIR で行った。

#### 4.1 基本機能

提案システムの機能は、すべて、ポインティングデバイスを使用することなくキーボードのみで操作できるようにしている。そのため、いくつかのショートカットとコマンドを覚える必要があるが、それらは、F1 キーを押すことにより、表示することができる。また、コマンド入力の際、テキストボックスの番号を指定する必要があるが、この番号も F1 キーにより表示・非表示を切り替えることができる。

コマンドの入力についてはプログラミング言語 HMMMML[4]でもスペルミス許容しているように、提案システムでも出来る限りのスペルミスや大文字小文字の混在などを許容するようにしている。コマンドが実行されたときのみ、入力したコマンド命令が削除され、入力ミスで実行されなかったコマンドはそのまま表示されるので、ユーザはどこが悪かったか気づくことができる。

一般的に使われているショートカットキーは提案システムでも同じように使用することができる。

#### 4.2 テキストボックス自動生成

一般のソフトでテキストボックスを使用するには、概ね次のような操作が必要となる。

- (1) テキストボックス機能を選択 (テキストボックスアイコンをクリック)
- (2) 文字を書きたい位置にテキストボックスをクリックまたはドラッグ
- (3) 文字を入力
- (4) 他の場所をクリックするなどしてテキストボックス機能を終了

ユーザが目的とするのは(3)の文字を入力する操作だけであり、そのために、(1)(2)(4)のような操作が必要となる。それが素早いテキスト入力の妨げになるとともに、ユーザの思考の流れを中断することにもなる。

そこで提案システムではテキストボックスを自動生成し、適当な位置に自動的に配置するようにした。提案システムでのテキスト入力の流れは次のようになっている。

- (1) 提案システムを起動したとき、または新規ファイルを選択したときなど、テキストボックスはすでに生成されている。
- (2) テキスト入力を始めれば、その位置に自動で表示される。テキストボックスのサイズはテキストの文字数と行数により自動的に決定される。
- (3) テキストボックス内へのテキスト入力が終わったときは **Alt** キー+矢印キーを押すことで、そのテキストボックスへの入力を終了する。
- (4) ひとつのテキストボックスへの入力が終了した時点で、次のテキストボックスが自動生成、自動配置される。

ここで、ユーザが操作しなければならないのは、テキスト入力と、入力が終わった際に **Alt** キー+矢印キーを押すことだけである。ユーザはテキストエディタと同じようにキー入力をするだけで、テキストの入力を行うことができる。**Enter** キーを押せば、テキストボックス内で改行できるので、箇条書きなどで複数行をまとめて書くこともできる。あるまとまりのあるテキストを入力し終わり、新しいテキストボックスを生成するときは、**Alt**+矢印キーで新しいテキストボックスが自動で作成される。この時、矢印キーの向きで新しいテキストボックスの位置をコントロールすることができる。新しいテキストボックスの配置の例を図2に示す。

テキストボックスの間隔は、直前のテキストボックスの幅とテキストの高さにより

自動的に計算される。

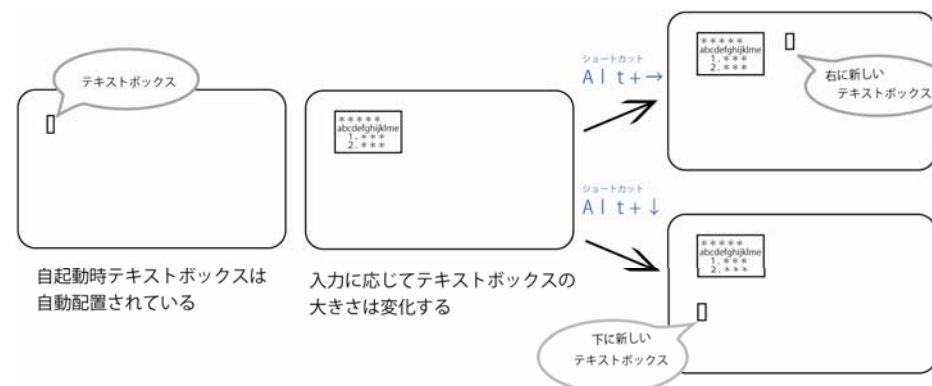


図2 テキストボックス自動配置

#### 4.3 コマンドの「どつき実行」

テキストの装飾はコマンドの「どつき実行」により行う。画面上でコマンドを入力する位置は通常のテキストボックスであり、ユーザは、普通のテキストを入力するイメージでコマンドを入力することができる。入力されたコマンドを実行するトリガとして、提案システムでは、「どつく」インタフェースを採用した。ノートPCを叩くと、その音をトリガとして、コマンドを実行する。本稿ではこの操作を「どつき実行」と呼ぶことにする。ユーザは装飾をおこないたいテキストボックスの番号とコマンドを入力し、「どつき実行」を行う。この時指定するテキストボックスの番号は、テキストボックスが生成されたときに順番に振られる番号であり、F1キーにより表示非表示を切り替えることができる。

テキストボックス装飾の意味とコマンドについて説明する。

##### 4.3.1 矢印

矢印はテキストボックス間の関係を示すために使うことができる。テキストボックスの番号と<> (不等号記号) によって指定する。矢印の描画イメージを図3に示す。

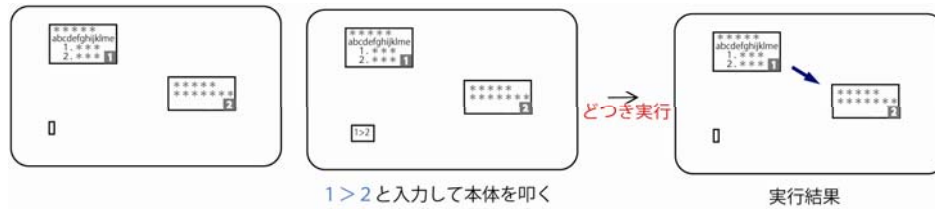


図3 矢印の描画

図4のように不等号の書き方により矢印の形を変えることもできる。不等号の向きが矢印のむきになるようになっており、<>のように指定すると線の両端に矢印が付く。

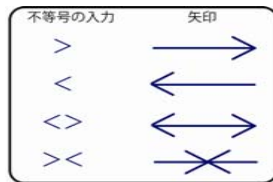


図4 矢印の形

#### 4.3.2 括弧

複数のテキストボックスの前（または後ろ）に{(または)}を表示することによりテキストボックス間のまとまりを表示することができる。括弧の位置をボックス番号の前に置くか、後ろに置くかで、描画される括弧の位置も変化する。括弧の描画イメージを図5に示す。

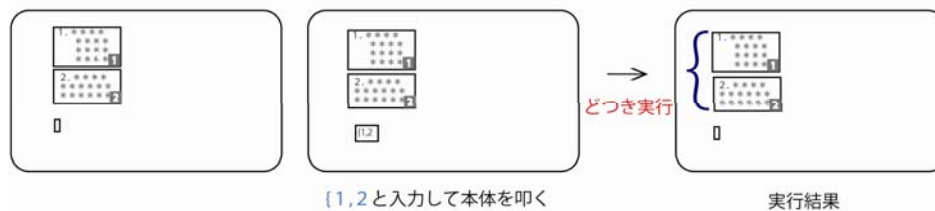


図5 括弧の描画

#### 4.3.3 囲み

囲みの機能は、テキストボックスを大きく楕円で囲む機能である。指定するテキストボックスの数により、ひとつのテキストボックスに楕円をつけることもできれば、図6のように複数のテキストボックスをまとめるように楕円をつけることもできる。

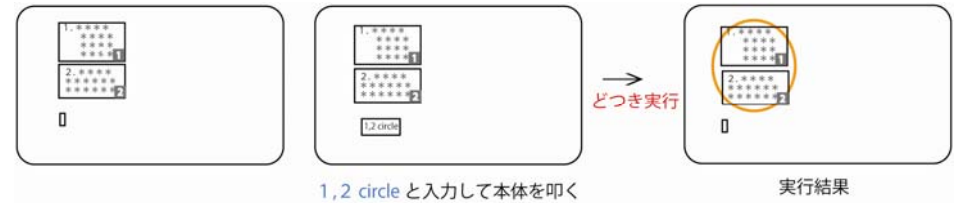


図6 囲みの描画

#### 4.3.4 フォントサイズ変更

テキストが主体のページでは、文字のジャンプ率が視線誘導のポイントになり、まず一番大きな文字を読み、次にその次に大きな文字を読むといわれている。手描きのノートでも見出しや、強調したい内容は大きな文字で書くことも多い。フォントサイズの変更は次のコマンドで行う。

n, fontup 「どつき実行」 フォントサイズを大きくする  
n, fontdown 「どつき実行」 フォントサイズを小さくする  
n: テキストボックス番号

#### 4.3.5 テキストボックスの傾き

講義中にふと思いついたアイデアや、本筋から外れるようなエピソードなどは、ノートの端に斜めに走り書きすることが多い。そこで、提案システムでもテキストを斜めに書く機能を実装した。

n, rotate,m 「どつき実行」 フォントサイズを大きくする  
n: テキストボックス番号  
m: 角度

この機能はあまりに多用すると、無秩序で見づらいノートになってしまうことも考えられるが、ある程度無秩序にすることで、注意を引いたり緊張感を高めたりすることができる。予備実験で作成したノート B でも多くのテキストボックスを斜めに配置しており、効果は確かめられている。

#### 4.3.6 ラインマーカ

入力中のテキストの一部分を強調するにはラインマーカ機能を使用する。ラインマーカは他の装飾機能と異なり、ショートカットキーで操作する。

Shift + 矢印キー	テキストの選択
Alt + l(エル)キー	ラインマーカ機能

## 5. 関連研究

辰川らはテキストの入力にキーボード、図形の入力にペンを用いるノートエディタについて研究している[5]。テキストや図形を素早く入力する方法について試みられており、提案システムで参考にする点も多いが、講義ノートをとるような環境でノート PC への入力にペンを用いるという点に疑問が残る。他にも講義ノートに限定した研究は多いが、ノート PC だけでなく、教室という環境のもとで、教室全体をシステムとするものが多い。駒形らは講義のリアルタイム画像を表示しながら、ペンをつかってノートをとるシステムを提案している[6]。

山本らは「書いてまとめる」プロセスの支援のために、ユーザが表現したいと思う事柄をより「自然に」表現でき、その表現したものからユーザにとって有用な情報が「わかりやすく」フィードバックされる方法について述べている[1]。このシステムは、論文などを文章をまとめるための支援であり、本稿で目指すのは、そのもっと前段階である学習時の支援である。ノートを取った後の支援として、ノートをとった後の見返すことの大切さとわざわざ見返すことの難しさに着目し、日常生活の中で自然にノートを見返せるようにするアプリケーションを渡邊らは開発している[7]。

岩崎らはノート PC に内蔵された加速度センサを用いて打鍵圧という非言語情報を取得している[8]。提案システムの「どつき実行」でも加速度センサを用いることも考えたが、加速度センサが内蔵されていないノート PC もあることから、より使いやすマイクでの実装を行った。

## 6. 考察

ノートをとっただけでは意味がなく、後で見返してこそその価値があるはずである。

「知的生産の技術」の中で梅棹は「発見の手帳」は、単なる精神の成長の記録などではなくて、知的蓄積のための手段なのだから、それはあとで利用できなくてはならない」と手帳の効用について述べている[9]。手帳やノートを見返すことが大切だと分かっているにもかかわらず、ただ文字の羅列では書いた内容をあとから利用するという点で望ましくない。そのために見返したときに印象にのこるようなノートについての研究が重要であり、本稿では、空間的に配置されたテキストやその修飾によって、印象にのこるようなノートが作成できるのではないかと仮説を検証した。そして、そのようなノートを、実際の講義中に素早く作ることでできるノートツールの開発を行った。

講義ノートを手描きでなくノート PC でとろうとする人は、PC の操作にある程度慣れていることが予想できる。その場合でも、思い通りのノートをノート PC でとることは難しい。提案システムでは、素早くテキスト入力を行うためのテキストボックス自動生成機能と、テキスト修飾の為のコマンドを実行するための「どつき実行」を実装し、評価実験で用いたような印象に残るノートを、ポインティングデバイスの操作なしに、素早く作成することができるようになった。今後は、提案システムの詳しい評価やさらに必要となる機能の追加などを行っていきたい。

## 参考文献

- 1) 山本恭裕, 高田真吾, 中小路久美代: "Representational Talkback"の増幅による「書いてまとめる」プロセスの支援へ向けて, 人工知能学会論文誌, Vol14, No1.1, pp.82-92(1999).
- 2) Scott Bateman, Regan L. Mandryk, Carl Gutwin, Aaron Genest, David McDine, Christopher Brooks: Useful Junk? The Effects of Visual Embellishment on Comprehension and Memorability of Charts, CHI2010, pp.2573-2582, (2010).
- 3) 生田信一, 大森裕二, 亀尾敦: Design Basic Book-初めて学ぶ, デザインの法則-, ビー・エヌ・エヌ新社, p.12(2007).
- 4) 宮下芳明: プログラミングに対するモチベーションを向上させる新言語HMMMMLの開発, プログラミングシンポジウム論文集, pp.57-64(2010).
- 5) 辰川肇, Neigel WARD: キーボードとペンを併用するノートエディタ, インタラクション 2002, pp.209-216(2002).
- 6) 駒形伸子, 大平雅雄, 蔵川圭, 中小路久美代: リアルタイム講義における受講者の思考活動に着目した支援に関する研究, 情報処理学会研究報告ヒューマンインタフェース 2001, pp.35-40(2001).
- 7) 渡邊恵太, 安村通晃: ユビキタス環境における眺めるインタフェースの提案と実現, 情報処理学会論文誌, Vol49, No.6, pp.1984-1992(2008).
- 8) 岩崎健一郎, 味八木崇, 暦本純一: Expressive Typing: 本体内蔵型加速度センサによる打鍵圧センシングとその応用, WISS2008論文集, pp.91-94(2008).
- 9) 梅棹忠夫: 知的生産の技術, 岩波書店, p.30(1969).