

作文行動を測定・分析するための マトリックス型テキスト編集モデルの設計

山口 琢¹ 高橋 慈子² 小林 龍生³ 大場 みち子⁴

概要: 作文行動を測定して分析するために、測定データの前提となるテキスト編集操作のモデルを設計した。新旧のさまざまなエディターやワードプロセッサが実装することを想定した、ログデータのモデルである。作文指導や業務改善で活用することを想定して、ワークシートや業務帳票を抽象化し、編集対象の単位に見出しをつけられることが特徴である。見出しは多次元にできるので、このモデルをマトリックス型と呼ぶ。このモデルに基づいてテキスト編集操作を記録するリファレンス実装を開発し、測定実験を行った。メモリ所要量などから他システムでの実装可能性を評価した。また、自明だが書き手の意図や編集操作の種類に関するような傾向をデータから読み取れるか試みた。最後に、従来の文書データのモデルや人間側の認知心理的なモデルとの関係を検討した。

キーワード: テキスト編集, 作文, 文章産出, モデリング, 測定, 分析

Designing Matrix Type Writing Model: Metrics for Writing Activity Analytics

TAKU YAMAGUCHI¹ SHIGEKO TAKAHASHI² TATSUO KOBAYASHI³ MICHIKO OBA⁴

Abstract: We have designed a matrix type information model for writing. When text editing systems sense and log their users' operation according to that model, it becomes possible to monitor and analyze the writing processes in a large scale and in a long term. The system will help to improve the teaching methods and reflection. Texts edited in the Matrix mode can be linked to those in the Page mode mutually, so that users can edit texts in trial and error manner. The Matrix Model is designed to run on cross-platform. The data of the model is stored in HTML format. We made a proof-of-concept that the Model can be implemented on the Open Web Platform and the editing operation logs tells us useful information.

Keywords: text editing, writing, text production, modeling, measurement, analytics

1. はじめに

近年、学習分析では分析対象の細粒度化が進み、成績といった行動の「結果」に加えて、ページめくり操作といっ

た「行動」を、分析の対象に含みつつある。これは、ICTデバイスや、デジタルコンテンツの普及によるところが大きい。作文・読書・プログラミングといったさまざまな学習活動を、ICTを用いて行えるようになってきた。このため、UIの操作などさまざまなログを、幅広く安価に得ることが現実的になってきたからである。

そのような研究では、アプリケーションのUIを工夫することによって、ユーザの思考内容をアプリケーションの操作(行動)から推定できると考える。そして、大きく次の4点において工夫して、研究を進めることになるだろう:

¹ フリー

Independent Researcher

² 株式会社ハーティネス

Heartiness Co., Ltd.

³ 有限会社スコレックス

Scholex Co., Ltd.

⁴ 公立はこだて未来大学システム情報科学部

Faculty of Systems Information Science, Future University
Hakodate

○行動を制約することで、様々な思考を促して記録・分析するための、さまざまな編集操作 UI

◎知的行動を記録するデータ

○アプリケーション操作を記録したデータの分析手法

○分析結果を指導や学習に応用する方法. 分析結果を提示する UI の工夫など.

このうち、本稿では「◎知的行動を記録するデータ」に着目し、題材を作文とする. すなわち「作文行動を記録するデータ」が本稿の対象である. さらに、設計(モデル化)して、何らかの実装によって最初のデータを得られるまでの「最初の一歩」を報告対象とする. 記録データ研究は、この最初の小さな1サイクルが困難だからである.

「作文行動を記録するデータ」の困難は、データの蓄積には手間と時間がかかるにもかかわらず、分析手法がこれから開発されること、そして、その分析手法によってどのような指導・学習が可能となるのかが未知数なことである. 先行研究で述べるように、作文行動を分析する研究の歴史は古いものの依然として未解明な部分が多い. このため、例えば「コピペによる作文の検出」といった目的ありきのアプローチよりも、作文の認知心理学的な解明といったオープンエンドなアプローチが求められる. この傾向は、現在の学習分析全般にいえるだろう. なんの役に立つのか分からないが、分析して役立つ目的で、データ構造を決めて記録を始めなくてはならない.

この困難は、近年のデータ分析において顕著である. 上の4点は別々の主体によって行われることがある. そして、4つの主体は、密接に関係しながらも別々の目的のために運営されている. 例えば、SNS への投稿を分析して災害対策に役立てようとする場合、投稿を記録している SNS 運営者、投稿データを分析する研究者やデータ分析事業者、そして分析結果を災害対策に役立てようとする自治体は別の主体である. データを予め与えられた変更できないものとして研究の外に置けば、この問題は顕在化しづらい. しかし、他者の有益な分析や指導・学習に結びつけようとして、自分のアプリケーションや記録データを改善しようとするれば、この困難に取り組まなくてはならない. データの抽象化(モデル化)による汎用性の確保や、書かれる文章の機密性(読書であれば、読まれるコンテンツの機密性や著作権)にも配慮しなくてはならない.

本論文では、まず、作文行動分析の先行研究としてキーストローク分析を取り上げて、その特長と課題を指摘する. つぎに、本研究のアプローチとして反復型開発の目標と、評価基準としての確認項目を示す. その目標を踏まえて、マトリックス型テキスト編集モデルを示す. また、目標を達成するための設計方針を示し、結果としてリファレンス実装(プロトタイプ・システム)により提案モデルを評価する. リファレンス実装と評価結果を踏まえて、キーストローク分析やその他の関連研究との違い・位置づけを考察

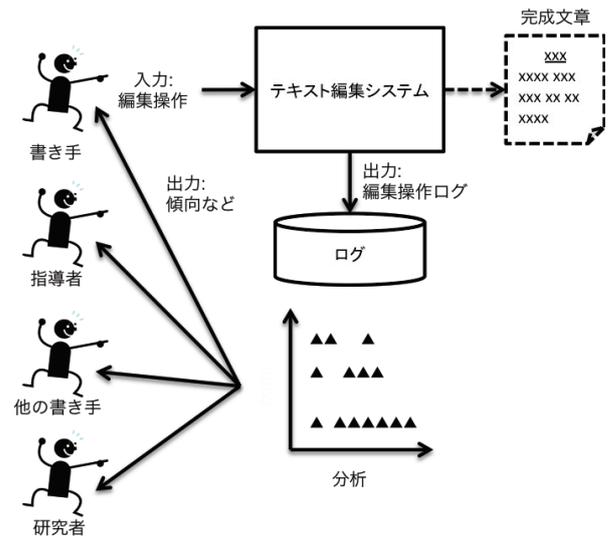


図1 Mechanical analysis of writing activity

Fig. 1 図1 機械的な作文過程分析の概念

する.

2. 先行研究

作文活動を機械的に記録・分析する手法は以前から存在する (Fig. 図1). 例えば、文字入力を記録して時間の経過に沿って分析するキーストローク分析は次の特長を持つ:

(1) 機械的な分析: 例えば、熟達した書き手と未熟な書き手では、入力の間隔 (pause) の現れ方に違いがあることを見いだした.

(2) 広範な適用範囲: クロスプラットフォームに適用でき、個人のさまざまな情報端末に適用できる. 言い換えると、書き手にとってのアクセシビリティが高い.

(3) 中立性: さまざまな作文・作文法・作文指導法、およびテキスト編集システムやワードプロセッサに中立的に適用でき、これらの比較に用いることができる.

(4) 記録の網羅性: 作文の最初から最後までカバーし作文の試行錯誤に対応している.

しかし、キーストローク分析など従来の手法では、書き手が「何を書こうとして、これらキー入力をしたのか」が分からないため、発話プロトコルなどと組み合わせて分析・活用されるようになってきた [1][3]. このように組み合わせた手法は、人間(被験者と研究者)の作業が必要となり、SNS に対するような機械的な大規模分析に応用できない. データを取りさえすれば所望の情報を得られるわけではなく、かといって、どのようなデータを取ればよいか分からないというのが、作文行動分析の状況であった.

先行研究を踏まえると、作文行動を記録するデータの課題は、キーストローク分析の特長(2)(3)(4)を引き継ぎつつ、書き手が何を書くべきか (what-to-say) 取捨選択している状況を検知できる(1)機械的な分析を実現できるデータであることである.

表 1 Goal and the evaluation criteria of the first iteration

目標	確認項目		
	[1] 使える	[2] 仮説触発	[3] 実装可能
[1] 機械的分析		○	
[2] 適用範囲	○		○
[3] 中立性		○	
[4] 記録の網羅性	○		○

なお、プログラミングのキーストローク分析、文章記述モデル、操作記録の網羅性と発想支援ツールの実装、アクセシビリティなどシステムの適用範囲、作文(文章産出)の認知心理学的モデルは、先行・関連研究ではあるが、本稿の評価 5.3 節のあと 6 章で取り上げて、本研究の位置づけを考察する。

3. アプローチ

われわれは、後述するマトリックス型テキスト編集モデルを出発点に、テクノロジー・プッシュのアプローチをとり、反復型の手法でモデルを開発する [14]。得られる分析結果は取得できるデータに依存するが、われわれがデータ収集のために提案するモデルは、過去に類型のないまったく新しいものだからである。まずモデルに基づいてデータを収集し、分析してみても役に立つ仮説を見つけて、次の実験で仮説検証や新たな分析を試みる一方でモデルも調整する。このような場合にソフトウェア開発では、反復型の手法 (iterative and incremental development) を採用する。われわれも反復型の手法でモデルを開発することにした。その最初の反復 (iteration) と、次以降の反復について説明する。

3.1 最初の反復：問題抽出と目標

2 章で述べた先行研究の特長と課題を踏まえて、最初の反復の目標は、次の 4 点を満たす新しい機械的な作文過程分析の仕組みを設計して、リファレンス実装によって、これらの実現可能性を概念実証 (proof of concept) することとした：

[目標 1] 機械的な what-to-say 分析

[目標 2] 広範な適用範囲：クロスプラットフォーム、あるいは書き手にとってのアクセシビリティ。さらに機密性の高い文章を含むユースケースの広さ。

[目標 3] 中立性：さまざまな作文・作文法・作文指導法に中立である。

[目標 4] 記録の網羅性：作文の最初から最後までカバーして、あとで構成を練り直すといった作文の試行錯誤に対応する。

3.2 最初の反復：設計・開発と評価

[目標 1~4] は、後で述べる 4 章のモデルを 5.1 節の設計

方針で実装して、次の 3 点を確認して、それらと提案モデルの原理に基づいて判断する。目標と確認項目の対応関係を表に示す (Table 図 1)。

[確認 1] 実際に使えること。ここで「使える」とは、アプリケーションを熟知した研究者本人 1 人が、数件の実際の文章 – 例えば研究会の予稿や論文 – の作成にあれば十分である。問題にしているのは抽象的な編集操作モデルであり、具体的な編集 UI の使い勝手 – 例えば段落を追加するのに要する操作の回数など – ではない。

[確認 2] 仮説の触発：編集操作ログが、what-to-say に関する何らかの仮説をインスパイアすること。

[確認 3] 実装可能：アプリケーションとして許容可能な性能で実現できること。

最初の反復では、これら [確認 1~3] で確認した内容を材料として、次のように [目標 1~4] の達成を判断する。個々の確認項目について「5.3 評価」で目標達成の観点で検討し、「7.1 まとめ」で目標達成の判断を述べる。これらの判断において、[確認 1~3] は、[目標 1~4] を満たす新しい機械的な作文過程分析の実現可能性の証拠となる

[目標 1] 機械的 what-to-say 分析は、最初の反復で [確認 2] のような仮説が触発されれば、将来の実現可能性が示される。

[目標 2] 広範な適用範囲は、「5.1 設計方針」に基づくシステムを実装できることが [確認 3] によって確認され、その実装が [確認 1] によって実際に使えることが確認されれば、将来の実現可能性が示される。

[目標 3] 中立性については、[確認 2] の仮説の内容を検討することによって、実現可能性を判断する。

[目標 4] 記録の網羅性は、「5.1 設計方針」に基づくシステムを実装できることが [確認 3] によって確認され、その実装が [確認 1] によって実際に使えることが確認されれば、将来の実現可能性が示される。

ここまでを、本稿の範囲とする。

3.3 2 回目以降の反復：実践・評価・改善

その後は、指導・学習に適用することでデータが得られ、分析が行われ、指導・学習の見地からマトリックス型テキスト編集モデルを評価できる。小規模でも、そのような結果を得られれば、研究をすすめるやくなるだろう。実践・評価の結果は、モデルの設計に反映して改善していく。段階的に実践の範囲を広げていく。

4. マトリックス型テキスト編集モデル

提案モデルの着想はワークシートに由来する。ワークシートは典型的には、特別に記入欄を用意・レイアウトしたメモ用紙や原稿用紙などの形で提供される。作文のガイドあるいは指導ツールとしてよく用いられる [12][13]。これらは、FLOWER and HAYES の認知心理モデルで言え

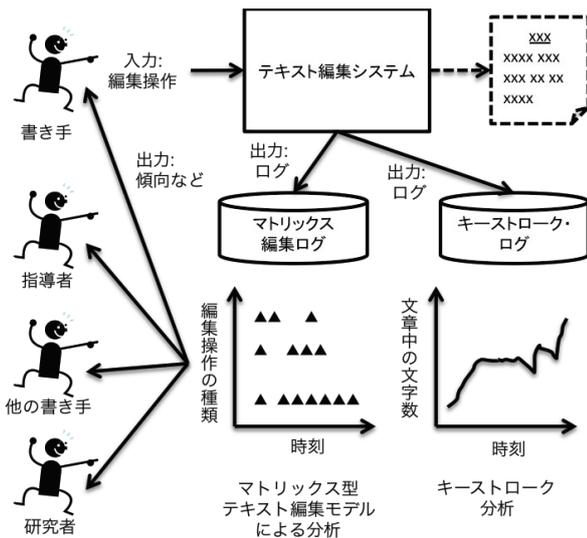


図 2 Inputs and outputs of the components which constitute the editing log analysis

Fig. 2 図 6 編集操作分析の概念: 入力と出力

ば、「訴えたいこと」など内容的なプランニング (planning) を支援するものである [11]. これらは、企業や官公庁で使われている帳票・フォームの、より抽象的で長文対応のものと考えられる。ワークシートが視覚的に紙などのメディアで提供されなくても、「この部分には、このような内容を書くべし」と規定することは広く行われている。論文の IMRAD[2] などがその例である。その規定を踏まえて、「その部分を書くには、このようなことを考慮すべし」という執筆ガイドが広く存在する。

ワークシートの記入欄に書き込まれたテキストは、「その記入欄で求められる内容を、しかるべきことが考慮されて、入力された」とものと解釈できる。個々の指導方法・作文方法や固有のワークシートのデータ構造と操作をモデル化することで、この原理で [目標 1] 機械的な what-to-say 分析を達成できると考えられる。

われわれは、テキスト編集システムにマトリクス型モデルを導入して、編集操作を時間の経過に沿って分析することを提案している [16]. Fig. 図 2 に編集操作ログ分析システムの構成を示す。システムへの入力は書き手の編集操作、システムの出力は編集操作ログあるいはそれを分析した作文の傾向などである。

4.1 マトリクス型テキスト編集モデルの原理

マトリクス型テキスト編集モデル(または、ここでは単にモデルと言う場合もある)は、what-to-say に着目して、書く活動をモデル化する。「ここに、読み手の知識を考慮して、『根拠』を記入しなさい」といった、「見出し付き記入欄」を介して作文する書き方を抽象化したものである。そして、その記入欄へのキー入力、その見出しが示す内容を入力したものを見なす。これによって、従来のキース

氏名 姓 名

図 3 Example of a name part of a worksheet for Japanese writers

Fig. 3 図 8 記入欄の例

氏名 姓 名

図 4 Example of an input written in Japanese

Fig. 4 図 9 記入欄への入力例

時刻	操作内容
13:15:00.001	見出しが「姓」の欄に「琢」と入力
13:15:05.002	見出しが「名」の欄に「やまだ」と入力
13:15:05.002	見出しが「名」の欄を削除
13:15:15.003	見出しが「名」の欄に「やまぐち」と入力
13:15:16.004	見出しが「名」の欄を削除
13:15:17.005	見出しが「名」の欄に「山口」と入力

図 5 Editing operation log

Fig. 5 図 10 編集操作ログの概念

トローク分析の課題を解決する。

Fig. 図 3 に、ごく簡単な例をあげる。あるテキスト編集システムの UI に「氏名欄」がある。「姓」という見出しのついた欄には氏名の姓を、「名」欄には名を入力するように求めていると考えられる。

Fig. 図 4 で、姓が「山口」名が「琢」である書き手が「姓」欄に「琢」、「名」欄に「山口」と書いた。

モデルでは、記入欄の画面配置や見出しの内容を捨象することで、[目標 3] さまざまな作文・作文法・作文指導法に対する中立性を確保する。氏名の「名」とは何か?といった概念をモデルに含めない。それらは、モデルを利用する作文法や指導者などの側に属する。モデルは、各欄に人間にとっては無意味な id をつけるのみである。Fig. 図 4 の入力が終わった時点までの編集操作ログは Fig. 図 5 のようになる。

モデルは、入力欄の見出しや入力されたテキストを記録しないこととする。これは、【目標 2】適用範囲で、機密性の高い文章にも適用できるようにするためである。Fig. 図 6 のように、入力欄の見出しや入力されたテキストを記録せず、分析者に隠されている(図中で「●」で表示)とする。その一方で、見出しや入力欄を id で認識・区別できるものとする。図中「id=c1」などがこれにあたる。また、見出しと入力欄を種類で区別できるものとする。図中「ax」が見出し、「c1」が入力欄を示す。その場合の編集操作ログは Fig. 図 7 のようになる。この場合でも、2つの記入欄は id によって識別されていて、id=c2 の記入欄の方が入力に手間がかかっているという情報を得ることができる。それが何かは分からなくても「何を書くべきか取捨選択している」様子を検知できる。これは従来のキーストローク分析ではできなかったことである。

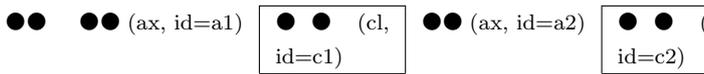


図 6 Labels and input texts are hidden to the researcher. "●●" stands for hidden text, "ax" for label, "cl" for cell.

Fig. 6 図 10-1 入力欄の見出しや入力されたテキストが分析者に隠されている場合

時刻	操作内容
13:15:00.001	id=c1 (ax: id=a1) の欄 (cl) に入力
13:15:05.002	id=c2 (ax: id=a2) の欄に入力
13:15:05.002	id=c2 (ax: id=a2) の欄を削除
13:15:15.003	id=c2 (ax: id=a2) の欄に入力
13:15:16.004	id=c2 (ax: id=a2) の欄を削除
13:15:17.005	id=c2 (ax: id=a2) の欄に入力

図 7 Editing operation log; "axes" indicates corresponding labels

Fig. 7 図 11 編集操作ログの概念

	科目 (ax, id=a1)	性格 (ax, id=a2)
志望動機 (ax, id=b1)	xxx yyy (cl, id=3)	(cl, id=5)
希望職種 (ax, id=b2)	(cl, id=6)	aaa bbb (cl, id=4)

図 8 Matrix structure of text input

Fig. 8 図 12 入力欄のマトリックス構造 (w=0.3 に)

時刻	操作内容
13:17:00.001	id=c3 (ax: id=a1, id=b1) の欄に入力

図 9 Editing action of text input into a cell of a matrix

Fig. 9 マトリックスのセルへの入力を記録したログ

4.2 複数次元マトリックス

記入欄に見出しや考慮点が多数ある場合がある。例えば、就職活動のエントリーシートの場合、「希望職種」欄に記入するにあたって「力を注いだ科目」や「性格」と関連づけよ、といった場合である。このため、「見出し付き記入欄」は見出しや考慮点を変数とする多変数関数であり、複数次元マトリックスとなる (Fig. 図 8)。物理的なワークシートでは、この抽象的なマトリックスを表として可視化することが多い。

Fig. 図 8 の入力欄 id=c3 に対する入力操作ログは Fig. 図 9 のようになる。id=c3 に対する入力が、id=a1 の見出しと id=b1 の見出しに関係していることを記録している。われわれの設計対象は、Fig. 図 8 のように図示される入力

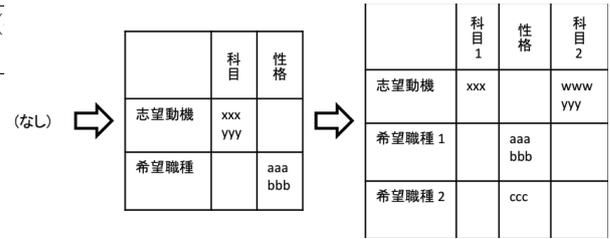


図 10 Edit operation of adding areas (cells) to write

Fig. 10 図 13 記入欄を追加する編集

欄に対する入力が Fig. 図 9 のように記録されるというモデルであって、これらの操作を行う UI ではない。例えば、そのような UI が表形式である必要はなく、入力欄の前に「得意科目を考慮して、志望動機を書きなさい」という指示が文章中に添えられているだけでもよい。設計者が、その欄への入力を Fig. 図 9 のようにログ出力するようにシステムを開発すればよい。

ここまでは既存の記入欄に従う作文の例を示したが、作文の指導では何を書くべきか (what-to-say) を自ら考えるよう指導されることもある。その場合は、抽象的には「記入欄の追加」が行われる (Fig. 図 10)。

このように、抽象化したマトリックス操作を記録して時間の経過に沿って分析することで、書き手の熟達度や指導法の効果判定の目安になるデータを提供する。

4.3 マトリックス型以外の編集操作

書く行為のすべてがマトリックス型モデルに該当するわけではない。[目標 4] 記録の網羅性を実現するためには、システム上に従来型の編集操作も実装して、マトリックス型モデルの操作と連携させる必要がある。

Fig. 図 11 で、文章を提出するときの形態は、右側のページ型モデルで見たものであるとする。左側のマトリックス型モデルでは、「xxx」、「yyy」、「aaa」、「bbb」しか入力していない。ただし、左右の「xxx」は、書き手にとって同じ「xxx」でなければならない。システムは、マトリックス型モデル側の「xxx」を、ページ型モデル側の「xxx」と関連づけて管理する。どちらかのモデルで「xxx」というテキストを編集すれば、瞬時にもう一方の側の「xxx」も、同じテキストに変更する。従来のワード・プロセッサで、アウトライン・モードでテキストを編集したとき、それがページ・モードにも反映されるのと同じである。

5. 実装と評価

5.1 設計方針

前記、マトリックス型テキスト編集モデルの原理に加えて、本研究の [目標 2] 広範な適用範囲と [目標 4] 記録の網羅性を実現するために、リファレンス実装の設計方針を、次のように設定した。

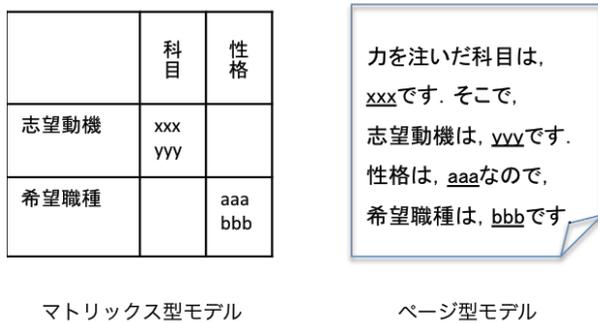


図 11 Composing text by matrix

Fig. 11 図 14 マトリックスによるテキスト内容の整理

5.1.1 Open Web Platform (OWP) の採用

幅広い書き手，指導者，研究者にとってシステムをアクセシブルにするために OWP 技術を採用する。EDUPUB で採用されている電子書籍フォーマット EPUB も「アクセシビリティのための特殊なものを作るのではなく，普通のものアクセシブルにする」という理念で設計されている [15].

5.1.2 標準データ形式の採用と記録の網羅性

記録の網羅性を高めるためには，従来型モデルの文章処理システムと相互運用できる必要がある。そこで，データ形式に OWP の標準データ形式を採用する。具体的には HTML や CSS を採用することで，HTML 対応のアウトライン編集システムや，CSS 対応のフォーマット・ツールをメモリ上でシームレスに利用できる。例えば，マトリックス型モデルで編集した論文を CSS を使って 2 段組で PDF 化することができる。

5.2 リファレンス実装

マトリックス型テキスト編集モデルのリファレンス実装は Fig. 図 12 のような画面構成になっている。左側の表が表示されている領域がマトリックス型モデルで編集するマトリックスビュー領域である。右側は，従来型モデルに対応するプレビュー領域である。プレビュー領域の文章が，作文のアウトプットとしての文章であり，印刷や Web ブラウザーでの表示など，読者が読むときの見え方に対応している。Fig. 図 11 の右側のページ型モデルを実装したものである。マトリックスの中で，枠線がハイライトされている部分が，現在，選択されているテキスト，右側のプレビューで白黒反転表示されている部分が，アウトプット中でそれに対応するテキストである。このように，アウトプットの文章に採用されているマトリックス側のテキストは，末尾に「⇒」マークが付いている。「⇒」マークが付いていないテキストは，アウトプットの文章に採用されていない。

リファレンス実装の文章は HTML で記述されている。画面右のアウトプットはもちろん，画面左のマトリックス

や，画面でハイライトされている両者間の関係も HTML で記述されている。このため，加工や変換することなく，スマートフォンやタブレット PC でも表示することができる。このような，OWP 技術によるシステム構成の詳細は本稿では触れない。

5.3 評価

5.3.1 [確認 1] 実際に使える

リファレンス実装を使って，1 年間にわたって本稿を含めて 11 の論文・予稿を執筆して投稿した。これをもって，提案モデルは確認項目の [確認 1] を達成したと考えられる。先行研究 [1] のシステムは認知心理学実験のための実験室システムである。[確認 1] は実利用を確認しているので，先行研究 [1] と比べて，「データとして記録する技術」としての適用可能性で少なくとも劣っていない。

5.3.2 [確認 2] 仮説の触発

Fig. 図 13 と Fig. 図 14 は，リファレンス実装を使った編集操作ログを編集操作の種類で分類して散布図にしたものである。1 つの点が，1 つの編集操作に該当する。2 つの散布図は，それぞれ別の論文・予稿に対応する。中段に分類された操作群は，Fig. 図 10 で示した，マトリックスに対する編集操作である。Fig. 図 13 から，この書き手は，編集の最後の頃には，マトリックス操作を行っていないことが分かる。これは，もっともなことだろう，メ切り間際になって，論点を変えろといった大胆な見直しはなかなか行わないからである。Fig. 図 14 では，マトリックス操作がもっと早い段階で終わっている。

リファレンス実装が出力するログ Fig. 図 13 は，例えば，次のような仮説をインスパイアする。「論文やシンポジウム予稿の執筆では，what-to-say にかかわる編集操作は，投稿前の一定時間以降には行われない。この一定時間は文章の量に依存するが，内容には依存しない。」われわれはこの仮説が正しいと言っているのではない。このような仮説を抱かせるポテンシャルが，Fig. 図 13 にあると考える。他方で，この仮説は有意義である。メ切り日時と，規定のページ数・文字数やそれまでの書き手の傾向などの条件によって，「メ切りに間に合わないの，もう文章の構成は変えない方がよい。」といった自動アドバイスの実現可能性を示唆するからである。これが実現すれば，書く活動のビッグデータから知識を引き出したことになる。

これら読み取りと仮説において，マトリックスの見出しを知らなくてもよいことから，[目標 3] の中立性を達成する見込みがあると考えられる。

5.3.3 [確認 3] 実装可能

本論文が 8 ページになった段階のものについて，メモリー所要量を測定した。比較のために，テキスト抽出したものをワード・プロセッサに流し込んで利用した。使用したのは iMac，プロセッサ Intel Core i5，搭載メモリー

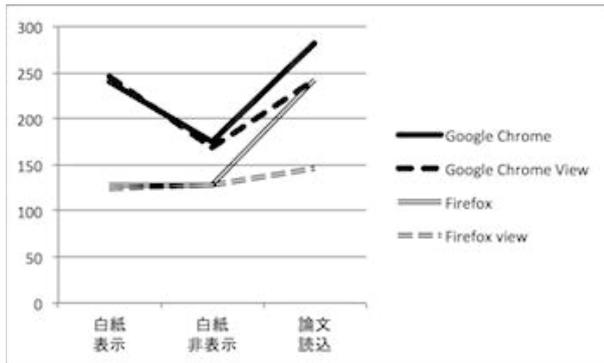


図 16 Memory requirements for viewing an article

Fig. 16 図 18 論文をブラウザで表示するだけの場合のメモリー所要量

い。しかし、Fig. 図 16 によると、そもそもブラウザで HTML 文書を表示するときのメモリー所要量が大きく、それに比べて提案モデルで編集するための増分はさほど大きくない。そこで、マトリックス型テキスト編集モデルがメモリー所要量の観点で実現困難とは言えないと判断できる。

メモリー所要量の評価にあたっては、文章量を想定する必要がある。論文 1 ページは 24 字 x 44 行 x 2 段 \approx 2 千文字あり、それだけで 400 字詰め原稿用紙 5 枚分ある。従来の指導における文章量を踏まえつつも、このシステムがもたらす新たな指導の可能性も考慮して、文章量の想定サイズは今後、具体化しつつ変動もしていくだろう。

6. 考察

本章では、提案モデルとキーストローク分析を含む関連研究・取組み・実装との違いや、位置づけを考察する。「文章 (text)・文書 (document, record)」「作文 (writing)・文章産出 (text production)」「モデル」といったキーワードで、認知心理学や ICT システムなどの分野で、幅広く研究や実践が行われてきた。本稿のテーマは、これらとの連想を喚起するので、ここで関連を検討しておく。

6.1 キーストローク分析

6.1.1 作文の場合

作文におけるキーストローク記録・分析とは、テキスト・エディターなどでの書き手のキー操作を記録して分析するものである。記録されるのは、キー操作やマウス操作とそれらを行った時刻、追加・削除された文字などである [1][3]。LEIJTEN and VAN WAES の作文に関する認知心理学研究によれば、優れた書き手と初学者とでは中断 (pause) のパターンが異なることが、キーストロークの分析から分かる [1]。また、キーストローク分析は適用範囲が広いことが

特長である [目標 2]。

しかし、書き手がそのように編集する意図を、分析から知ることにはできないとも述べられている。そこで、LEIJTEN and VAN WAES はキーストロークの記録とプロトコル分析などを組み合わせて分析することを提案している [1]。また、館野らの、正規課程外におけるアカデミック・ライティング指導を想定した実験では、キーストロークの再現を見ながらチューターが学習者と会話することで成果をあげている [3]。このような方法は、より深い理解にたどり着くかもしれないが、大規模に実施することは困難であると 2 章で指摘した。

6.1.2 プログラミングの場合

作文と同じくキーで文字を入力してテキスト・ファイルを作り上げるプログラミングでも、キーストロークを記録・分析する研究が行われている [4][5]。

プログラミングのキーストローク分析では、プログラムの書き手、すなわちプログラマーがシステムに対して行ったさまざまな命令を検出している。これによって、テスト、不具合の原因究明、修正といったレベルで、プログラマーの行為を判別できている。このような記録・分析は、人が介在しない機械的なものなので、大規模に行うことも可能である。

6.1.3 作文とプログラミングの違い

作文とプログラミングで検出できる行為レベルの違いは、テキスト・エディターや開発環境といったツールが持つ機能レベルの違いで説明できる。プログラミングでは、文法チェック、コンパイル、デバッグ、モジュール依存関係の追跡などの、論文の試読レベルのようなことをシステムに行わせる命令が存在する。その命令の実行を記録に残せる。

そこで、プログラミング環境のようなレベルの機能を作文環境 (テキスト編集システム) に導入すれば、作文のキーストローク分析でも、より高いレベルで書き手の行為を記録し分析できると考えられる。マトリックス型テキスト編集操作は、そのようなレベルの機能に該当するものである。

6.2 文章記述モデルと編集操作記録

今回、新しいテキスト編集モデルを導入したが、従来のテキスト編集システムの文章記述型モデルを検討する。プログラミングにおけるキーストローク分析のように、作文でも、既存のワードプロセッサで編集操作を記録すれば、より高いレベルの分析が可能かもしれないからである。

この問いに答えるために、従来のワード・プロセッサやテキスト・エディターが採用しているモデルを、ODA (Open Document Architecture, 開放型文書体系、または Office Document Architecture, 事務文書体系) に照らして考察する。

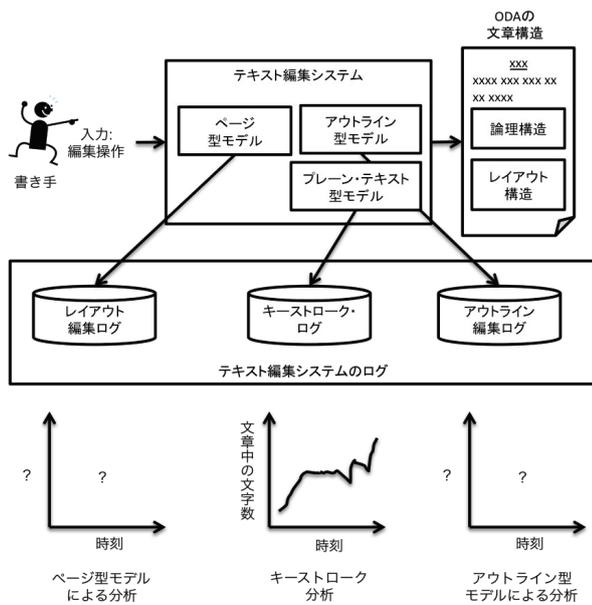


図 17 ODA's document structure, text editing model of a computer system and the editing log analysis

Fig. 17 図 2 ODA による文書の構造, システムの編集モデルおよび編集操作ログ分析

6.2.1 ODA : 標準モデル

ODA は, さまざまな文章編集システム間で文章を交換または長期保存するための, 文書ファイル・フォーマットの国際規格である [6][7]. ISO (International Organization for Standardization) と ITU-T (International Telecommunication Union - Telecommunication Standardization Sector) において同じ内容が規格化された. ITU-T の規格は無償で入手できる. 筆者らは, 1988 年に ODA を用いた文書交換の実証実験に参加し [8][9], また ODA に基づく文書処理システムを開発した [10].

ODA による文章構造のモデルは ITU-T 「T.412 Document structures」に記述されている. ODA によれば, 文書はレイアウト構造と論理構造を持つ. ここでは, 本稿の用語に合わせて, ODA の「document(文書)」を「文章」, 「architecture」を「モデル」と読み替える. また, キーストローク分析を説明するために, プレーン・テキスト型モデルをあわせて考察する (Fig. 図 17).

6.2.2 ページ型モデル

ページ型モデルでは, 印刷したときなどのページレイアウトに基づいて, 文章を記述し編集する. これによると, 文章は用紙, ページ, ヘッダー/フッター, (ページ構成する長方形の領域としての) 段落などで構成される. 文章を作成・編集するとは, これらページなどを作成・編集することである. コンピュータソフトウェアによる代表的な実装は PDF である.

このモデルに基づく編集操作ログを分析すると, 例えば, 「しめ切り間際になって, 2 段組のレイアウトに段通しの図

を配置する作業で苦しんでいる」ということが判る.

しかし, Fig. 図 8 や Fig. 図 9 で示したような, 「得意科目や性格と, 希望職種のマッチングを試行錯誤した」のかどうかを, ページ型モデルの編集操作ログから検出するのは困難である. すなわち, ページ型モデル的な「表レイアウトの編集操作」ではなく, 「プランニング, マッチングを目的としたワークシート操作」をモデル化する必要がある.

6.2.3 アウトライン型モデル

アウトライン型モデルでは, 章節項のツリー構造に基づいて文章を記述する. 文章は論理的な階層構造 (hierarchy of logical objects) で記述される. 代表的な利用例では, 文章を章・節・項・(論理的な構成要素としての) 段落などから構成する. 文章を作成・編集するとは, 段落を作成・編集したり, 章を入れ換えたりすることである. コンピュータソフトウェアによる代表的な実装はワード・プロセッサのアウトラインビューである. ここで, 論理的階層構造を書き手に示すグラフィックスや画面配置は問わない. ワード・プロセッサによる「目次」的なグラフィックスもあれば, アイデア・プロセッサによる放射状のグラフィックスも, アウトライン型モデルの実装である.

このモデルに基づく編集操作ログを分析すると, 例えば, 「メ切直前に章節項の並び順やレベルを修正している」といったことが判る.

しかし, Fig. 図 8 や Fig. 図 9 のように, いわば多重・多次元の章構成を考慮するような作文行為を検出することはできない.

6.2.4 プレーン・テキスト型モデル

プレーン・テキスト型では, 文章を, 文字や改行などの制御コードの並びとみなして編集する. 狭い意味でテキスト・エディターという場合, このモデルを採用したアプリケーションである. このモデルでは, 文章を編集するとは, 文字や改行コードなどを挿入・削除・コピー・ペーストすることである.

このモデルは汎用性が高く, 作文研究のキーストローク分析の多くは, 文章について, このプレーン・テキスト型モデルを採用している. 多くのワード・プロセッサやテキスト・エディターでも, 文章全体や一部の枠内を, このモデルで編集しているとみなすことができる. 作文のキーストローク分析がプレーン・テキスト型のモデルを採用するのは, 適用範囲の広さという点で理にかなっている.

しかし, キーストローク分析の検討で指摘した通り, このプレーン・テキスト型モデルに基づいて, 書き手の編集意図を推測するのは困難であった.

6.3 記録の網羅性と実装

ワード・プロセッサやテキスト・エディターは, 上で述べた複数のモデルを, 1 つのアプリケーション上に実装している. モデルを利用する画面を「ビュー」や「モード」

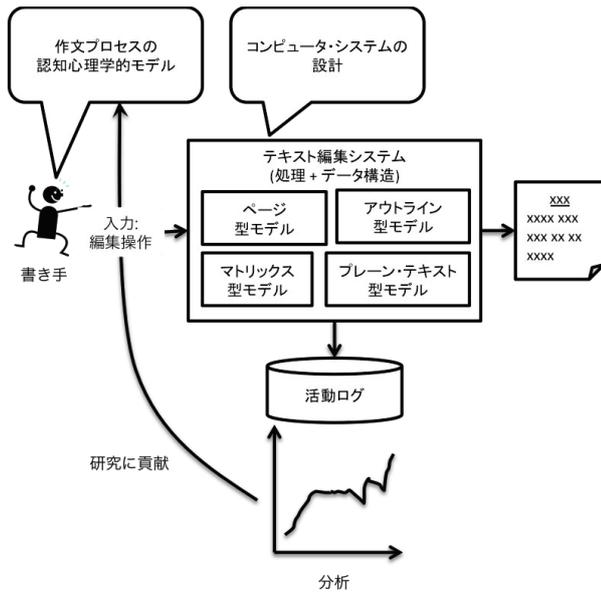


図 20 Design of a text editing system and a model of writing process in cognitive psychology

Fig. 20 図 3 認知心理学的モデルとコンピュータシステムの設計

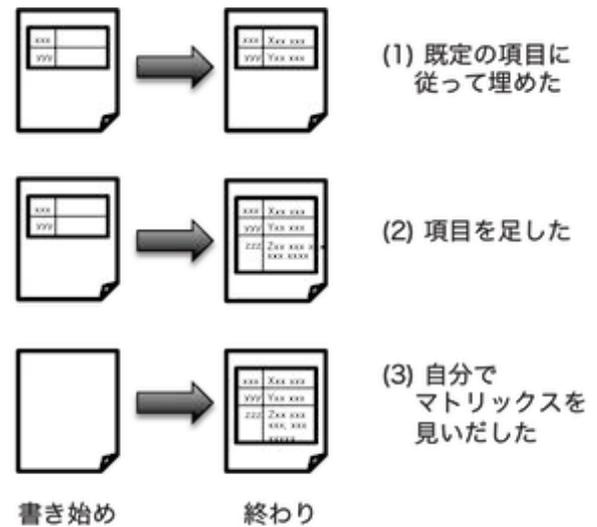


図 21 Editing patterns expected to be found by analysis

Fig. 21 図 19 編集パターン (w=0.3 に)

7. おわりに

7.1 まとめ

学習分析の細粒度化を背景に、作文行動を分析するためのマトリックス型モデルについて、先行研究を踏まえた着想・設計からリファレンス実装の開発・評価までの概念実証を報告した。モデルの設計段階の評価として、[目標 1] 機械的に作成した散布図から what-to-say に関する仮説を引き出すことができた [確認 2]。[目標 2] 適用範囲の広さについて、OWP の技術を使ってクロスプラットフォームな実装ができた。また、文章のテキストを記録しなくても有意義な分析ができる可能性を示した。OWP の技術で実装できたことは、[確認 1] リファレンス実装を使って論文・予稿を書けることと、[確認 3] アプリケーションとして許容されるメモリー所要量であることを持つて確認した。[目標 3] 作文法や指導法などに対する中立性は、マトリックスの行・列・セルの配置情報や見出しの内容をモデルに含まない設計とすることで実現した。また、見出しの内容によらずに [目標 1] の機械的な what-to-say 分析ができた。[目標 4] リファレンス実装にマトリックス型テキスト編集モデルに基づくビューと従来型ビューを両立させることができ、記録の網羅性を確保できた。二つのビューが両立していることは、[確認 1] 実際に使えることと [確認 3] 許容されるメモリー所要量であることを確認した。

7.2 今後の課題

今後の課題は、指導の現場への適用と評価である。Fig. 図 21 や Fig. 図 22 のように、書く活動パターンの違いを

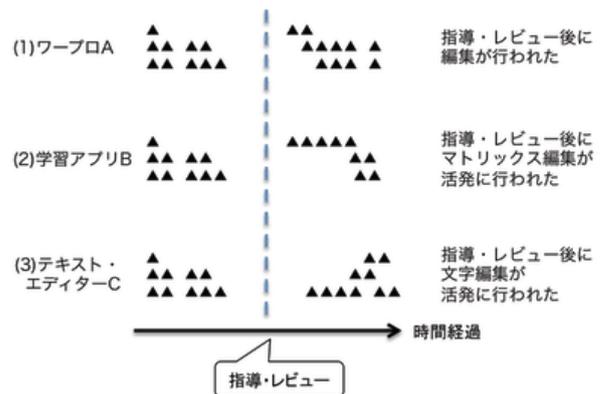


図 22 Editing patterns expected to be found after a review

Fig. 22 図 20 指導・レビュー後の編集行動の違い

モニタすることで、作文指導や文書レビューの向上に役立てることを目指している。並行してモデルの定式化や、システムの安定性向上や UI 改善にも取り組む。[目標 3] 中立性の実現については、さまざまな作文指導法への適用などによって、今後、確認していく。

本研究の目的はモデルの開発であり、モデルの良し悪しと、個別の実装の使い勝手とを切り分ける必要がある。しかし、モデル解釈の参考となるプロトタイプすなわちリファレンス実装の使い勝手は、実験に用いることがあるため継続的に改善する必要がある。

8. 謝辞

本論文は科研費 (26560124) の助成を受けたものである。
 謝辞

参考文献

- [1] LEIJTEN, M., VAN WAES, L. "Keystroke logging in writing research: Using Inputlog to analyze and visualize writing processes". *Written Communication*, 30(3): 358 - 392. (2013)
- [2] Luciana B. Sollaci, Mauricio G. Pereira: "The introduction, methods, results, and discussion (IMRAD) structure: a fifty-year survey", *J Med Libr Assoc.* 92(3) 364371 (2004-07)
- [3] Y. Tateno, T. Ohkawauchi, T. Hirano, J. Nakahara: "Development of 'Reporeco', a Visualization System for Students' Writing Processes : Focusing on a Case of an Academic Writing Course as an Extracurricular Study", *Japan Society for Educational Technology* 37(3) : 241-254 (2013-11)
館野泰一, 大川内隆朗, 平野 智紀 "レポート執筆プロセスの可視化システム「レポレコ」の開発: チューターによる正課課程外の指導場面に着目して". *日本教育工学会論文誌* 37(3) : 241-254 (2013-11)
- [4] Y. Takada, K. Torii: "A Method for Measuring Programmer Debugging Performance from Key Strokes", *The Institute of Electronics, Information and Communication Engineers (IEICE) J77-D-1(9)*, 646-655, (1994-09)
高田義広, 鳥居宏次 "プログラマのデバッグ能力をキーストロークから測定する方法". *電子情報通信学会論文誌. D-I, J77-D-1(9)* : 646-655 (1994-09)
- [5] T. Yoshimura, Y. Kamei, H. Uwano, A. Monden, K. Matsumoto: "An Analysis on How Programmers Use Breakpoints in Debugging", *The Institute of Electronics, Information and Communication Technical Report* 109(307) : 85-90 (2009-11)
吉村巧朗, 亀井靖高, 上野秀剛, 門田暁人, 松本健一: "ブレイクポイント使用履歴に基づくデバッグ行動の分析(デバッグ, モデル駆動開発)". *電子情報通信学会技術研究報告. KBSE, 知能ソフトウェア工学* 109(307) : 85-90 (2009-11)
- [6] ISO/IEC "ISO/IEC 8613-2:1995 Information technology - Open Document Architecture (ODA) and interchange format: Document structures" (1995)
- [7] ITU-T "T.412 : Information technology - Open Document Architecture (ODA) and interchange format: Document structures" (1993)
- [8] Noguchi Kenichiro, Oya Makoto: "", *Information Processing Society of Japan, Joho Shori* 31(9) pp.1235-1244 (1990-09)
野口健一郎, 大谷真: "OSIの実現とその課題 (VI) 事務文書体系 (ODA)", *情報処理学会, 情報処理* 31(9) pp.1235-1244 (1990-09)
- [9] Kobayashi Hideaki, Takahashi Osamu, Nitta Tetsuji: "Conformance test for OSI products in INTAP : ODA/ODIF", *Information Processing Society of Japan*, 779-780 (1988-09)
小林偉昭, 高橋修, 新田哲二: "INTAPにおけるOSIコンFORMANCE試験: ODA/ODIF", *情報処理学会, 全国大会講演論文集* 第37回昭和63年後期(2) pp.779-780 (1988-09)
- [10] Yamaguchi Taku, Matsudaira Hideki, Uehara Tetsuzo, Kagimasa Hideko: "A Method of Document Layout Process Based on ODA (3) : Processing Repetitive Construction", *Information Processing Society of Japan*, pp.602-603 (1990-03)
山口琢, 松平秀樹, 上原徹三, 鍵政秀子: "ODAに基づいた文書割付け処理の実現方式(3): 繰返し構造の処理方式", *情報処理学会, 全国大会講演論文集* 第40回平成2年前期(2) pp.602-603 (1990-03)
- [11] FLOWER, L., HAYES, R., "A Cognitive Process Theory of Writing", *College Composition and Communication*, 32(4): 365-387 (1981)
- [12] 山田ズーニー "考えるシート". 講談社, 東京 (2008-08)
- [13] 秋田喜代美 "読む心・書く心—文章の心理学入門 (心理学ジュニアライブラリ)". 北大路書房, 京都 (2002-11)
- [14] 清水康敬, 中山実, 向後千春 "教育工学研究の方法". ミネルヴァ書房, 東京 (2012-08)
- [15] M. Murata: "Electronic book format EPUB and Japanese typography: Mainstream people in Japan are unlikely to win international standardization battles", *Journal of Information Processing and Management* Vol. 55 No. 1 P 13-20 (2012-04)
村田真 "電子書籍フォーマット EPUB と日本語組版 日本でメインストリームにいる人間は国際標準化の舞台ではまず勝てない". *情報管理* 55(1) : 13-20 (2012-04)
- [16] T. Yamaguchi, M. Oba, O. Takahashi: "Matrix Model for Text Editing : Externalization of Writing Process", *JSiSE research report* 28(7), 101-108, (2014-03)
山口琢, 大場みち子, 高橋修 "マトリックス型テキスト編集モデルによる学習過程の可視化". *教育システム情報学会研究報告* 28(7), pp.101-108 (2014-03)