

簡単な操作で高度な組版を行える文書作成システム -WYSIWYG エディタと組版言語の融合-

1 背景

日常的に文書を作成する人々（特に理系の学生や研究者など）を対象に行ったインタビュー調査では、実に 79% の人が既存の文書作成ソフトウェアに何らかの不満を抱えており、「多少の学習コストがかかっても、より良いソフトがあれば乗り換えを検討する」という声が多く聞かれた。

現在主流となっている文書作成システムには、大きく分けて2つの代表的なアプローチがあり、それぞれに対極的な強みと弱みが存在している。

WYSIWYG エディタの代表「Microsoft Word」の課題

見たままの完成形を直接操作できるため直感的で、専門知識がなくても使い始められるというメリットがある。一方で、数式入力に手間がかかる上、美しい配置計算は行われない点や、理系文書に求められる専門的な拡張性が弱いという課題がある。さらに、行分割や図の配置といった厳密な組版ルールへの対応に妥協が見られる点も弱点である。

組版言語の代表「LaTeX」の課題

厳密なルールに基づいた非常に美しい組版と強力な数式表現が可能であり、専門分野のパッケージも豊富である。しかし、ソースコードをコンパイルするまで完成形が確認できないため直感的な操作が不可能であり、さらに環境構築の難易度の高さや LaTeX 言語の習得が必要な点が、初学者にとって極めて高い参入障壁となっている。

2 目的

本プロジェクトの目的は、これら既存ソフトウェアの課題を解決し、Wordの「直感的な操作性・はじめやすさ」と、LaTeXの「高度な組版処理・専門的な表現力」という、従来はトレードオフの関係にあった強みを両立（良いところ取り）させた、全く新しい文書作成システム『Monk』を開発することである。

「完成形を見ながら、そして直接触りながら、高度な専門表現も自在に記述できる」という新しい文書作成体験を提供し、誰もがストレスなく高品質な文書を執筆できることを目指している。

3 開発の内容

3.1 プロダクトの概観

開発した Monk は、画面左側に CUI、右側に GUI が同時に表示される構成となっている。この概観を図 1 に示す。

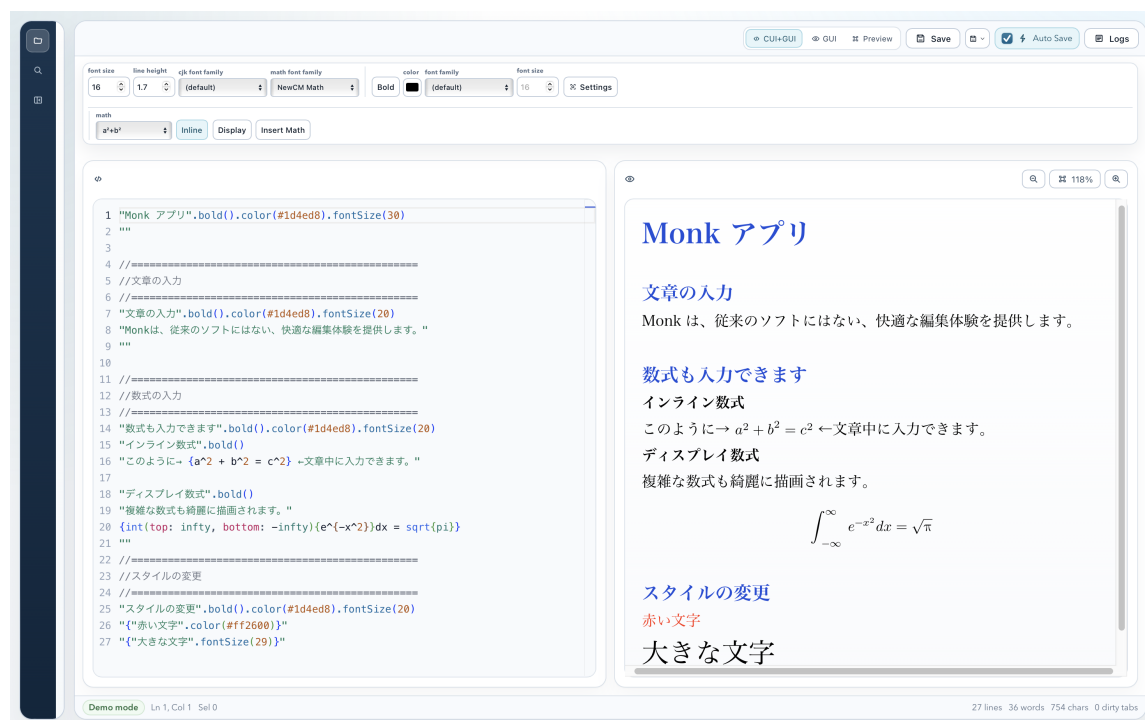


図 1 Monk アプリケーションの概観

CUI

ソースコードは独自で設計した新言語である、Monk 言語で記述する。CUI からはコードエディタのように記述でき、加えられた変更は即座に右画面の GUI に反映される。

GUI

本プロジェクトの最大の特徴であり、既存の組版システムと一線を画するのが、右画面の GUI (完成形) を直接操作できる点である。GUI で行った変更は、即座に左画面の CUI (ソースコード) にも自動で反映される。

構成と動作環境

デスクトップアプリケーションとして動作し、フロントエンドに React、バックエンドに Rust を用い、これらを Tauri を介して連携させている。

3.2 特徴

直感的な操作性

Wordのような「直感的操作・はじめやすさ」と、LaTeXのような「高度な組版・専門表現」という、従来は対極にあった強みを両立させている。初心者はGUI中心の操作から簡単に入門でき、慣れてきたらCUIを触って自由度の高い表現を楽しむといった使い方が可能である。

高度な組版

日本語と欧文の標準的な組版規則を基に独自の組版エンジンを実装し、数式を含む文章も美しく組版できる（図2，図3）。

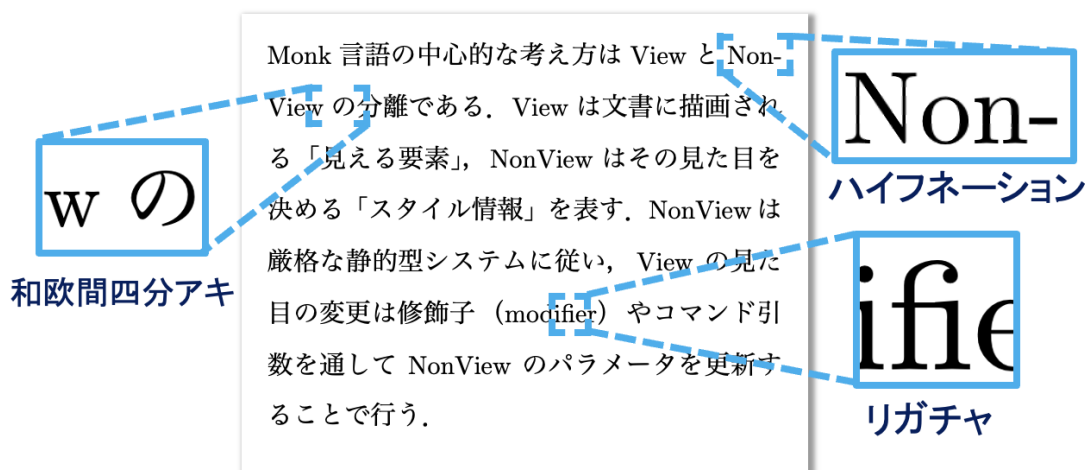


図2 厳密な日本語組版要件に対応

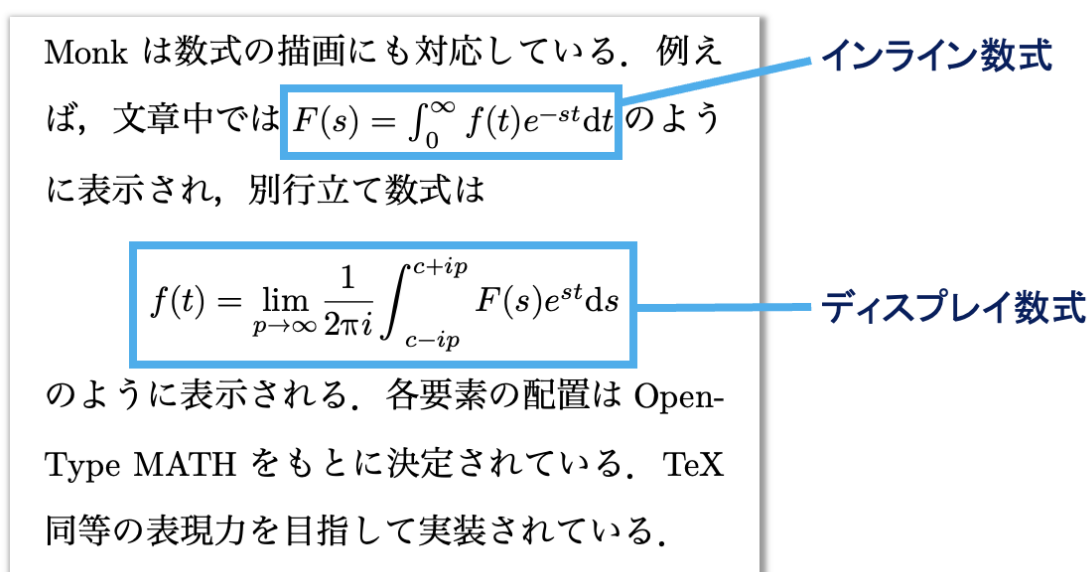


図3 文章中（インライン）、別行立て（ディスプレイ）数式も美しく組版

モダンな独自言語

初心者でも直感的に理解できる可読性の高い新言語である。また、静的型付け言語であり厳密な型検査を行うため、エラーを防ぎやすい。

Monk 言語におけるコマンドは、LaTeX のようなバックスラッシュ (\) を使用せず、シンプルな構文を採用している。

例えば、数学の総和記号（シグマ）を記述する場合、図 4 のように書く。

`sum(bottom: i=1, top: 10){ i }` → $\sum_{i=1}^{10} i$

図 4 Monk 言語の例

文字の大きさや色などのスタイル指定には、後置修飾も使用できる。対象となる要素の後ろにドット (.) で修飾子を繋げていくことで、煩雑になりがちなスタイル指定を極めて簡潔に記述できる（図 5）。

`"こんにちは"
.fontSize(54pt)
.color(Red)` → **こんにちは**

図 5 Monk 言語の後置修飾

4 従来の技術（または機能）との相違

従来のソフトウェアと比較した Monk の新規性は「双方向性をもつ CUI と GUI の融合」と「マークアップとプログラミングの性質を兼ね備えた新言語」の 2 点にある。

双方向性をもつ CUI と GUI の融合

従来の LaTeX は「コードを書いてコンパイル結果を待つ（一方向）」、Word は「見た目を直接編集する（GUI 完結）」であった。一方 Monk は、GUI 上で直接編集した内容が即座に CUI に反映され、逆に CUI の変更も瞬時に GUI に反映される双方向の連携により、初心者から上級者まで誰でも直感的に使用できる環境を提供する。

マークアップとプログラミングの性質を兼ね備えた新言語

組版言語の代表である LaTeX 言語は、難解なおまじないコマンドやエラーの原因特定が困難であるという課題があった。Monk は「マークアップ言語 × プログラミング言語」を目指し、文書の書きやすさと、柔軟な処理能力を融合させた

新言語を独自に設計した。さらに、静的型付けを採用して厳密な型検査を行うことで、表現の曖昧さを排除している。これにより、エラーを未然に防ぎやすだけでなく、エラーメッセージの充実も図れる。

5 期待される効果

日常的に文書を作成するユーザーの執筆における利便性の向上が期待される。

従来の LaTeX では、「コードを探す→変更する→最大数十秒のコンパイルを待つ→結果を確認する」という面倒な手順が必要であったが、Monk ではコンパイル待ち時間を極力減らし、完成形を見ながら GUI を直接操作できるため、細かい修正にかかる時間が劇的に短縮される。また、親切なエラー通知により、エラー修正に悩む時間も削減される。加えて、「LaTeX は難しそう」と敬遠して Microsoft Word などを使用していたユーザーも、Monk を使えば Word に似た直感的な操作感で美しい文書を作成できる。

なお、本文書は Monk を用いて作成されたものである。指定通りのフォーマットを正確に再現できる性能を有していることが、本文書をもって証明されている。

6 普及（または活用）の見通し

アプリケーションとして公開

Monk はアプリケーションとして公開することを想定している。ユーザーは Monk をインストールするだけで文書の作成・共有・閲覧が行える。これによって、LaTeX など専門システムで課題となっていた、環境構築・パッケージ管理など、難しい部分を取り除くことができ、「誰でも使える」形を実現できる。

理系ユーザー向けのアプリケーションとして展開

ターゲットとして、理系の学生（レポート作成や数式を多用する授業のメモ取り）や専門職（論文や技術書の作成）を設定している。LaTeX のように高度な専門表現ができ、かつ Word のように使いやすいアプリケーションとして普及を図る。

7 クリエータ名（所属）

丸山 貴楽（東京大学 工学部 計数工学科 システム情報工学コース）

石田 天歩（東京大学 工学部 計数工学科 数理情報工学コース）

中村 健（東京大学 工学部 計数工学科 システム情報工学コース）