



2025 年度 未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当 PM

曾川 景介 (newmo 株式会社 CTO)

2. クリエータ氏名

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)

3. 委託金支払額

2,880,000 円

4. テーマ名

簡単な操作で高度な組版を行える文書作成システム

5. 関連 Web サイト

なし

6. テーマ概要

本プロジェクトでは、直感的な操作性と高度な組版機能を兼ね備えた、新しい文書作成システム「Monk」を 3 人チームで開発した。日常的に文書を作成する人々 (特に理系学生や専門職) が抱える既存の文書作成システムへの不満から出発し、チームが実施したインタビュー調査では 79% の人が既存のソフトに何らかの不满を抱えていた。

現在主流の文書作成システムである Microsoft Word と LaTeX は、それぞれ対極的な強みと弱点を持っている。Word は直感的な操作が可能だが高度な組版処理には限界があり、LaTeX は厳密なルールに基づく美しい組版が可能だが環境構築や操作の敷居が高い。本プロジェクトでは、これら両者の弱点を克服し、Word の「直感的な操作性」と LaTeX の「高度な組版処理」を両立させることを目的として、Monk を開発した。

Monk は画面左側に CUI (独自マークアップ言語 Monk 言語によるソースコード)、右側に GUI (組版結果の完成形) を同時に表示する構成を取る。CUI での変更は即座に GUI に反映され、逆に GUI での編集も瞬時に CUI に反映され

る双方向の連動を実現した。バックエンドには Rust、フロントエンドには React (TypeScript) を用い、Tauri で両者を統合している。

7. 採択理由

本提案は、直感的な GUI 操作による組版と高度な CUI ベースの組版をリアルタイムに連携させることで、数式や図表を用いた美しい文書を初心者でも容易に作成できる文書作成システムの実現を目指すものである。独自言語により厳密な構文・動的な修飾・リアルタイムプレビューを可能にすることによって、LaTeX のような高い表現力と Microsoft Word のような使いやすさを両立し、既存のソフトウェアが抱える学習コスト、組版の自由度、環境構築の難しさといった課題の解決を目指している。さらにこの独自言語は、画像や数式、表、動画、3D モデル、コードの実行までを文書に埋め込めるようにすることを目指しており、現代的な表現を充足させた新しいマークアップ言語となることも期待できる。Rust 的なパッケージ機構による拡張性や AI 活用を前提とした設計思想、既に言語仕様やプロトタイプを用意している点を評価し採択とした。

8. 開発目標

本プロジェクトの開発目標は、以下の 3 つの柱を実現することであった。

第一に、GUI からの操作を前提とした新しいマークアップ言語の設計と言語処理系の実装である。既存の組版言語である LaTeX 等は GUI からの操作を想定して設計されておらず、動的な要素を組み込むことも考慮されていない。そこで、マークアップ言語とプログラミング言語の両方の特徴を持つ静的型付け言語を設計し、字句解析器から型検査器、中間表現生成器に至るまでの言語処理系を構築することを目指した。

第二に、日本語と欧文の標準的な組版規則を基にした独自の組版エンジンの構築である。TeX の box/glue/penalty モデルを採用した行分割アルゴリズム、OpenType MATH テーブルに基づく高精度な数式配置、JLReq や JIS X 4051 に準拠した日本語組版規則の実装など、美しい文書を生成するための技術的基盤の構築を目指した。

第三に、CUI と GUI がリアルタイムに連動する WYSIWYG エディタの実現である。ユーザが最終的な見た目を確認しながら直接操作できる環境を提供するため、グラフィック要素木 (GCT) による構造管理と、ユーザ操作の体系化・状態遷移による安全な操作管理を設計・実装することを目指した。

9. 進捗概要

本プロジェクトでは、3 人のクリエイターがそれぞれマークアップ言語の処理系 (石田)、組版エンジン (中村)、WYSIWYG エディタ (丸山) を分担して開発を

進めた。

マークアップ言語の処理系については、言語設計から字句解析器、構文解析器、名前解決、型検査器、中間表現生成器に至るまでの全工程を、Rust を使ってフルクラッチで実装した。Hindley-Milner 型推論アルゴリズムをベースにした型推論器や Union-Find による型制約管理など、高度な言語処理系の機能が実現された。

組版エンジンについては、入力テキストを文書解析、ノード生成、行分割、行組立、ブロック統合、出力 (SVG / HTML5 Canvas / PDF) の 6 段階で処理するパイプラインが構築された。Python で PoC を作成した後に Rust へ移植するアプローチを取り、Knuth-Plass アルゴリズムに基づく行分割や、OpenType MATH に基づく数式配置、JLReq 準拠の日本語組版規則など、高い技術的完成度に達した。

WYSIWYG エディタについては、GCT (グラフィック要素木) による構造管理と、ユーザ操作を InputGesture として体系化し、ステートマシンによる状態遷移で安全に制御する仕組みが React (TypeScript) で設計・実装した。

一方で、言語処理系と組版エンジンの完成に時間を要したため、フロントエンドとの連携はプロトタイプ段階にとどまり、成果報告までのプロダクトへの完全な統合には至らなかった。また、当初計画にあった Language Server の開発やテキストエディタの予測変換機能なども期間内には実装できなかった。

10. プロジェクト評価

本プロジェクトは、Microsoft Word と LaTeX という対極的な文書作成システムの強みを統合するという野心的な目標に対して、3 人チームで独自言語の設計、組版エンジンの構築、WYSIWYG エディタの実装という 3 つの高度な技術課題に同時に取り組んだ挑戦的なプロジェクトであった。

各コンポーネントの技術的な完成度は高い。独自マークアップ言語 Monk は、GUI からの操作を前提として設計された静的型付け言語であり、System F ベースの型システムや Hindley-Milner 型推論など、言語処理系としての設計水準が高い。組版エンジンは、TeX の box/glue/penalty モデルの採用、OpenType MATH テーブルに基づく数式配置、JLReq に準拠した日本語組版規則の実装など、既存の組版システムの知見を取り入れつつ独自の設計を行っている。WYSIWYG エディタについても、ユーザ操作の InputGesture への体系化やステートマシンによる状態管理など、拡張性と安全性を両立する設計がなされている。

チームプロジェクトとしての進め方については、3 人がそれぞれの専門性を活かして分担する体制が機能していた。中盤以降、厳密な仕様のすり合わせの重要性を認識し、チーム内のコミュニケーションを改善していった過程も評価できる。

11. 今後の課題

最も重要な課題は、言語処理系・組版エンジン・エディタ間の連携を完成させ、一つのプロダクトとして統合することである。現在はプロトタイプ段階にとどまっており、各コンポーネント間のインタフェースを設計し、安定した動作を実現する必要がある。当初から掲げている動く組版という目標もまだ道半ばである。

また、文書作成ソフトウェアとしての基本機能（画像の挿入、表の作成など）の基本パッケージの整備や、外部パッケージの開発方法の策定も必要である。ユーザインタフェースの設計・開発についても、初学者も対象としたユーザテストに基づく改善を繰り返す必要がある。組版エンジンの技術的限界としては、合字とハイフネーションの競合、LaTeX 数式との完全互換の困難さ、フロート（図表の自動配置）の未実装などが挙げられる。

長期的には、アプリケーションとして公開し、従来の「紙面に印刷された静的な文書」という固定観念を取り払い、「デバイス上で閲覧する静的・動的な要素を含んだ文書」という新しい文書形式の標準規格となることを目指して欲しい。