

# SuperIME: IME によるテキスト編集機能の統合

匿名で査読を行うため著者名なし\*

**概要.** 計算機上で様々なテキストエディタが利用されているが、システムごとに編集方法や編集機能が異なっているのが不便である。本論文では、各国語入力のために OS に用意されている IME(Input Method Editor) 機能を利用することにより、あらゆるテキストエディタにおいて同じ操作によるテキスト編集を可能にする方法を提案する。我々の手法を利用すると、異なるテキストエディタ上での編集操作が共通化されるだけでなく、テキスト編集時に便利な様々な機能をあらゆるエディタで利用することが可能になる。

## 1 はじめに

計算機上でテキストを編集するために様々なテキストエディタが利用されている。文書を作成するときはワープロを利用し、メールを書くにはメールクライアントを利用し、文字端末でのプログラム開発には vim や Emacs を利用し、IDE を利用した開発では付属のエディタを利用し、ネット上でテキストを扱うにはブラウザのテキストフォームを利用するといったように、場合によって異なるエディタが利用されている。

エディタの機能や操作体系はシステムごとに異なっているのが普通である。ブラウザやワープロでテキストを 1 行消したい場合はマウスで行全体を選択してから削除キーを押せばよいが、vim では「d」キーを 2 回タイプして消すことが多く、Emacs では Ctrl-K キーがよく使われている。Emacs に慣れたユーザがワープロ上でも Ctrl-K で行を消去したいと思っても、そういう機能は用意されていないのが普通であるし、機能拡張が可能なシステムを利用している場合でも、操作体系を完全に同じにすることは難しい。あらゆるエディタの操作を統一することは難しいが、様々なエディタで共通に利用できるソフトウェア層をユーザとアプリケーションの間に置くことができれば異なるエディタの編集操作をある程度共通化できる可能性がある。

現在のパソコンには IME(Input Method Editor) と呼ばれる文字入力機構が用意されており、様々な言語のテキスト入力に利用されている。IME はエディタなどとは独立したソフトウェアであり、ユーザのすべてのキー入力を受け取って各国語に変換した結果をアプリケーションに送出する。IME はあらゆるアプリケーションで共通に利用されるので、たとえば日本語入力用の IME を利用すれば、Emacs でもブラウザでも IDE でも同じ操作で日本語を入力できる。IME は一般には各国語入力のためのみに利

用されているが、テキストの挿入/移動/削除といった編集操作も IME が受け持つようにすれば、様々なエディタ上で同じ操作で編集を行なうことが可能になる。Mac 上の様々なテキストエディタにおいて同じキー操作でテキスト編集を可能にする SuperIME システムについて述べる。

## 2 実装例

MacRuby で記述された IME である「Gyaim」に様々な編集機能を追加したものを利用した編集作業の例を示す。一般に IME はアルファベット以外の文字を入力するときだけ有効にするのが普通であるが、ここでは Gyaim を常に有効にしておくことによりあらゆるキー入力を Gyaim が取得している。

### 2.1 ブロック移動

### 2.2 文字列暗号化

### 2.3 Dynamic Macro

以上の機能に加え、文字のスワップやキャレット移動といった基本的な編集操作を Gyaim は実装しており、システムやアプリケーションによらないテキスト編集が可能である。

## 3 実装

Gyaim は MacRuby で記述された Mac 用の IME であり、ソースが 500 行程度とコンパクトであるにもかかわらず、他の IME に見られない機能を実装しており、本論文のような実験も容易である。

### 3.1 MacRuby

MacRuby は、Mac 用のアプリケーションを開発するために拡張された Ruby 実行環境であり、Mac の Objective-C ライブラリを Ruby で扱うことが出来る。Gyaim では、日本語などの入力を補助するフレームワークである InputMethodKit Framework を MacRuby から呼び出すことによって基本的な IME の機能を実装している。

Copyright is held by the author(s).

\* 匿名で査読を行うため所属名なし

しかし、ブロック移動やインデントの処理をあらゆるテキストエリアで行うためには、テキストフィールドに 入力されている全文を IME が取得する必要があり、InputMethodKit はこの機能を備えていない。Gyaim では、AppleScript などを併用することによりこの問題を解決したが、実装には課題が残っている。

## 謝辞

謝辞は、ブラインドレビューのため、投稿時には削除すること。カメラレディ時に、必要があれば追加すること。

Dynamic Macro system[10] [1] [6] [11] [3] [4] [1] [7][8][14] [2][5][12][16] [15]. [9][13].

## 参考文献

- [1] TextEditors.org. <http://texteditors.org/>.
- [2] P. H. Dietz, B. Eidelson, J. Westhues, and S. Bathiche. A practical pressure sensitive computer keyboard. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, UIST '09, pp. 55–58, 2009.
- [3] M. Doi and H. Lei. Word Processing for the Japanese Language, 2010. [http://www.ieeeeghn.org/wiki/index.php/STARS:Word\\_Processing\\_for\\_the\\_Japanese\\_Language](http://www.ieeeeghn.org/wiki/index.php/STARS:Word_Processing_for_the_Japanese_Language).
- [4] S. Doi, S.-i. Kamei, and K. Yamabana. A text input front-end processor as an information access platform. In *Proceedings of the 17th international conference on Computational linguistics - Volume 1*, COLING '98, pp. 336–340, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.
- [5] C. Harrison, D. Tan, and D. Morris. Skininput: appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 453–462, 2010.
- [6] T. Kawada and S. Amano. Japanese Word Processor. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence*, pp. 466–468, 1979.
- [7] F. C. Y. Li, R. T. Guy, K. Yatani, and K. N. Truong. The 1line keyboard: a QWERTY layout in a single line. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pp. 461–470, 2011.
- [8] I. S. MacKenzie, R. W. Soukoreff, and J. Helga. 1 thumb, 4 buttons, 20 words per minute: design and evaluation of H4-writer. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pp. 471–480, 2011.
- [9] T. Masui and S. Hashimoto. GoldFish: real-world GUI framework for Android. In *SIGGRAPH Asia 2012 Symposium on Apps*, pp. 3:1–3:1, 2012.
- [10] T. Masui and K. Nakayama. Repeat and predict – two keys to efficient text editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pp. 118–123, 1994.
- [11] K. Mori and T. Kawada. From Kana to Kanji: word processing in Japan. *IEEE Spectrum*, 27(8):46–48, August 1990.
- [12] T. Murase, A. Moteki, N. Ozawa, N. Hara, T. Nakai, and K. Fujimoto. Gesture keyboard requiring only one camera. In *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*, UIST '11 Adjunct, pp. 9–10, 2011.
- [13] J. Rekimoto. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, UIST '97, pp. 31–39, 1997.
- [14] J. Rick. Performance optimizations of virtual keyboards for stroke-based text entry on a touch-based tabletop. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, UIST '10, pp. 77–86, New York, NY, USA, 2010.
- [15] L. Tesler. A personal history of modeless text editing and cut/copy-paste. *interactions*, 19(4):70–75, July 2012.
- [16] D. Wigdor and R. Balakrishnan. TiltText: using tilt for text input to mobile phones. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, UIST '03, pp. 81–90, 2003.