

Vim エディタ上でのキー入力解析による 入力短縮化方法の学習を支援するツールの構築

杉本 元気[†] 坂本 一憲[†] 鷲崎 弘宜[†] 深澤 良彰[†]

早稲田大学 基幹理工学部 情報理工学科[‡]

1. はじめに

1.1. コーディング短縮化の重要性

ソフトウェア開発において、コードの記述や修正は避けて通れない開発工程である。それゆえ、コーディング工程を短縮することは、開発工程全体の短縮に繋がる。

コーディングを効率化するための一つの手段として、使用するテキストエディタの操作に習熟するということが挙げられる。より少ないタイプ数でコーディングが可能となれば、コードの記述も修正もより短時間で行うことが可能となる。

1.2. エディタの学習方法

エディタの使用法を上達させるには、大きく分けて 2 つの方法が考えられる。一つは、チュートリアルやヘルプ等を利用して新しい操作を学ぶ方法である。もう一つは、自分の操作の非効率性に気づき、改善するという方法である。

一つ目の手法は初心者には有効であるが、ある程度習熟したユーザには効果的でない。なぜならば、チュートリアルやヘルプに含まれる情報は一般的なものであり、個々のユーザに特化した情報は得られないからである。

二つ目の手法は、Vim の製作者である Bram が提唱する方法である [1]。自分が普段行う操作の中から非効率な部分を発見するため、この方法はある程度習熟したユーザに適している。

2. 現状の問題点

前述のように、ある程度習熟したユーザがさらにエディタの使用法を上達させるには、自分の操作の中から非効率な部分を見つけ出す必要がある。しかし、非効率を発見することに集中しては実際のコーディングへの集中が困難となる。

そこで、ユーザが意識せずとも非効率な操作を発見できるような仕組みが求められる。しかし、現在そのような仕組みは存在しない。これゆえ、テキストエディタにある程度習熟した後はさらなる上達が困難であるというのが現状である。

3. 関連研究

ユーザのテキストエディタへのキー入力を利用する研究として、キー入力を始めとした様々なログデータを新しいテキストエディタの設計に利用するという研究が存在する [2]。しかしこの研究には、ユーザの学習支援という側面はない。

本研究と類似した目的を持って作成されたと思われるソフトウェアに、Huffshell [3] というものがある。このソフトウェアは、ユーザが実行したシェルコマンドの履歴を元に、ハフマン符号を用いてキー入力が最小となるようなエイリアスを提案するものである。

4. 提案手法

本研究では、Vim テキストエディタ [4] を対象に、ユーザのキー入力を保存、解析してユーザにとって有用な情報を抽出することを試みる。

4.1. 設計

本研究で作成したツールの構成を以下に示す。ユーザは本ツールを介して Vim を起動することで、キー入力のログが適切に保存される。そしてユーザは任意のタイミングでツールの解析機能を実行することで、保存されたログの解析、情報提示が行われる。

ログの生成には、Vim に備わっている W オプションを利用した。このオプションは、Vim へのキー入力を全てファイルに書き出すというものである。この機能を利用することによって制限が生じるが、これは後述する。

A Construction of a Text Editor Learning Supporting Tool for Better Manipulation by Mining User Input on Vim Editor

[†] Genki Sugimoto, Kazunori Sakamoto, Hironori Washizaki, Yoshiaki Fukazawa

[‡] Waseda University

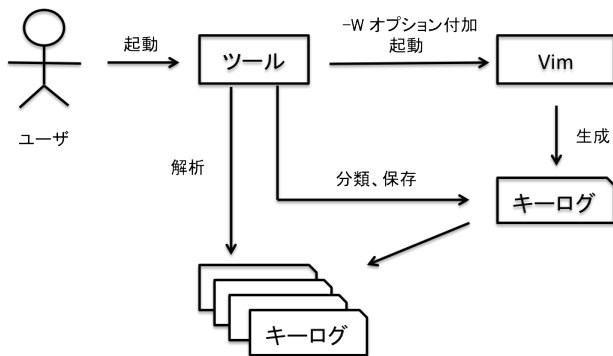


図1 ツールの構成

4.2. 解析

本論文では、キー入力のログの中から繰り返し出現するパターンを抽出した。繰り返し行われている操作を省略または短縮化することで、全体として大きな効率化が見込めるためである。

4.3. 解析結果

筆者の一ヶ月に渡る Vim 使用履歴の解析結果を以下に示す。上部のグラフがパターンと出現回数、下部の表がグラフのラベルと実際のパターンとの対応関係を示す。

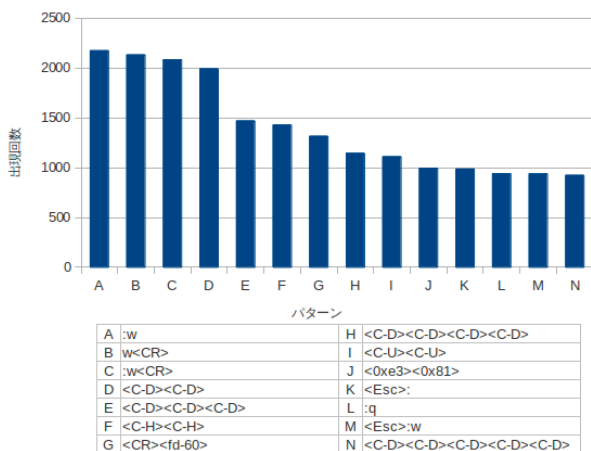


図2 解析結果

4.4. 評価

解析結果を見ると、「:w」とそれに類するパターンが最も多く出現していることがわかる。この入力、ファイルを保存するためのコマンド入力である。これは、筆者のファイルに変更を加えるとすぐに保存をするという癖によるものである。こうして可視化されることでこの操作が不必要なものであると気づくことができる。この入力パターンは、自動でファイルが保存される仕組みを導入するか、Vim のスワップファイル機能を活用することで省略することが可能である。

このように、キー入力のログを解析し、情報を提示することによってユーザがより効率的な操作を学習することを支援することができる。

5. 制限

4.1 で述べたように、本ツールではキー入力ログの取得に、Vim に備わっている W オプションを使用した。しかしこの方法では、キーが入力されたタイミングを検知することが不可能である。そのため、連続して行われた一連の入力と、断続的に行われた入力の区別がつかない。

これを解決するためには、Vim の起動中に動的にキー入力を検知する必要がある。しかし、現在の Vim の機能ではこれを実現することは難しい。しかし、テキストエディタの使用方法を解析する上で時間というのは非常に重要な情報である。今後は如何に時間要素を取得するかを考えたい。

本ツールではユーザへの情報提供のみを行っているため、非効率的操作の改善方法はユーザが考えなくてはならない。ユーザへ可能な改善方法を提案するための手法として、他のユーザが似たような状況をどのように改善したかを記録、閲覧できるコミュニティサイトの建設が考えられる。似たような操作履歴を持つユーザを検索し、有用な提示するといったことも行いたい。

6. まとめ

本論文では、ユーザのテキストエディタへのキー入力ログを解析することで、ユーザがテキストエディタをより効率的に使用できるようになるために有用な情報を提供できることを示した。テキストエディタを効率的に使用できるようになることはコーディング速度の向上に繋がり、それはソフトウェア開発の効率化に繋がる。

今後は、このツールを発展させてソフトウェア開発の効率化に貢献することを目標としたい。

参考文献

[1] Bram Moolenaar, “Seven habits of effective text editing 2.0”, 2007
 [2] Good Michael, “The Use of Logging Data in the Design of a New Text Editor”, ACM SIGCHI Bulletin 16.4 (1985) : 93-97
 [3] paulmars, Huffshell, <https://github.com/paulmars/huffshell>
 [4] Bram Moolenaar, Vim, <http://www.vim.org/>