

1. 担当 PM

竹迫 良範（株式会社リクルートテクノロジーズ 執行役員）

2. クリエータ氏名

福田 優太郎（電気通信大学 大学院）

3. 委託金支払額

2,304,000 円

4. テーマ名

OS に依存しないキーマッピングのカスタマイズシステム

5. 関連 Web サイト

<https://violets.io/>

6. テーマ概要

本プロジェクト“violets”では、キーマッピングの構造に即したキーマッピング記述言語および、この言語から Windows、macOS 上の既存のキーマッピングソフトウェアの設定ファイルを生成するトランスパイラを開発した。このソフトウェアにより、個人がそれぞれ独自にカスタマイズしたキーマッピング設定の共有が OS を横断して可能になった。

7. 採択理由

古いワークステーションのコンピュータではキーボードと端末が一緒であることが普通であり、DEC VT100、SUN Type5 など名機のキーボードが評判を集めた。そんな中で単体のキーボード Happy Hacking Keyboard、Kinesis、Ergodox、BAROCCO などが登場し、コンピュータを乗り換えても自分の愛着のあるキーボードを使い続けるユーザが増えてきた。優れたプログラマの間では、キー入力を効率化するためにキーマップをカスタマイズすることは一般的である。たとえば Vim ユーザではなく Emacs ユーザの間では、キーマップを編集し、Ctrl キーと Caps Lock の位置を入れ替えたりする。さらにキー入力の効

率を求めるユーザは QWERTY 配列ではなく、Dvorak 配列や、親指シフトの NICOLA 配列などを使うこともある。最近では、自作キーボードが普通に販売されるようになり、Helix、Iris、Corne、Mint60 などのキットを使って自分独自の専用キーボードを作ってみるエンジニア文化も浸透してきている。その中で、既存のキーマッピングツールは OS 依存であることが多く、自分でカスタマイズした設定を複数環境にインポートエクスポートすることが難しいといった問題があった。本提案では、様々なキーマップのカスタマイズやワンショットモディファイアの設定を各種 OS でポータブルにして、乗り換えの手軽さと複数ユーザ間での設定共有を加速させることが期待できる。本プロジェクトを進めることによって日本一キーボードカスタマイズ文化に詳しくなって、異文化交流を加速し、プログラマの世界平和に貢献して欲しいと考え採択した。

8. 開発目標

本プロジェクトでは、単一のキーマッピング定義ファイルから、複数の環境をターゲットにキーマッピングツールの設定ファイルを生成するトランスパイラを開発することとし、以下の機能を実現できることを開発目標とした。

- 統一的な記法であらゆるキーマッピングをカバーすることができるキーマッピング記述言語
- キーマッピングヘビーユーザの恩恵をライトユーザも受けられるようなモジュール性の高さ
- ユーザのユースケースに合わせた柔軟なキーマッピング変更
- ユーザが自分の違和感に基づいて編集できるカスタマイズ性の高さ

9. 進捗概要

本プロジェクトで開発した violets の全体概要を図 1 に示す。

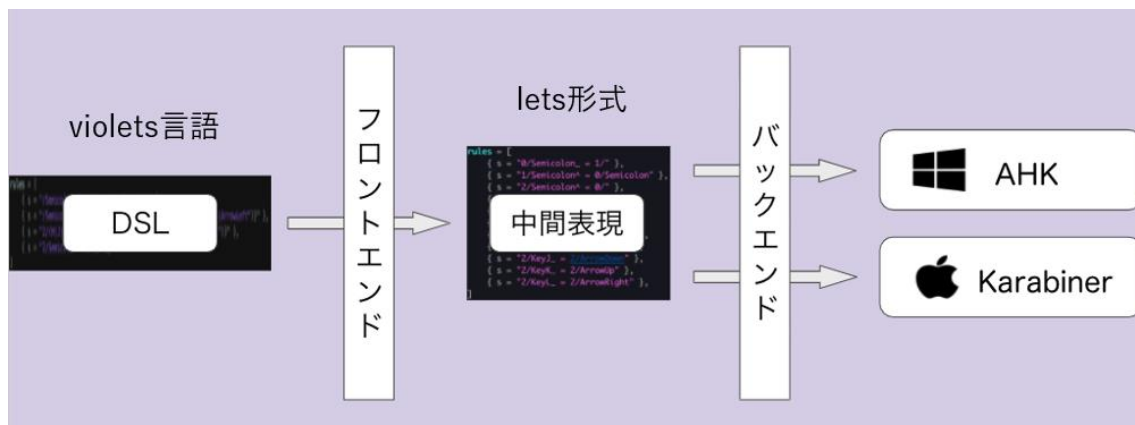


図 1 : 本プロジェクトで開発した violets の全体概要

violets 言語の仕様は TOML をベースに独自に拡張したものである。キーマッピングが設定された violets.toml の例を以下に示す。

```
name = "lcolon"
edition = 202002
rules = [
  { s = "0/Semicolon_ = 1/" },
  { s = "1/Semicolon^ = 0/Semicolon" },
  { s = "2/Semicolon^ = 0/" },
  { s = "1/KeyL_ = 2/ArrowRight" },
  { s = "2/KeyL_ = 2/ArrowRight" },
]
```

状態遷移 0 番から始まり、セミコロンのキーが押下 (`_`) されたら何も出力せずに状態遷移 1 番に移行し、そのままセミコロンのキーが引上 (`^`) されれば、セミコロンを出力する。しかし、セミコロンが押下された状態で `L` キーを押すと、右矢印キーを出力し、カーソルを右に移動させる副作用の動作を実行し、状態遷移 2 番に移行する。さらに `L` キーを何度も押したらカーソルを右に移動させる動作を繰り返す。キーを離れた動作が記述されていない場合は副作用を解消して状態遷移 0 番に戻る。

本プロジェクトではキーマッピングを副作用つきオートマトンでモデル化し、OS に依存にしない中間表現として `lets` 形式を定義したことで大きな成果を出すことができた。`lets` 形式では、キーの押下、キーの引上それぞれのキーイベント動作に対して状態遷移図を記述することができ、状態を遷移しながら最終的にどのようなキーを出力するかといった副作用を与えることができる。その結果として、既存のキーマッピングツールに依存しない統一的な記法でキーマッピングを定義することが可能になり、バックエンドのトランスパイラを別途開発することで Windows 向けには AHK (AutoHotKey)、macOS 向けには Karabiner (Karabiner-Elements) に対応することができた。

10. プロジェクト評価

プロジェクトとしては紆余曲折があったが、最終的にキーカスタマイズ設定を OS に依存しない中間表現として副作用付きの有限オートマトン `lets` 形式で定式化できることを証明した。これは状態遷移図を生で記述するプリミティブな形式のため、ユーザが記述しやすい `violets` 言語を別で開発した。バックエンドの開発、フロントエンドの開発について量が多過ぎて一部間に合わなかったものがあるが、それらは今後の課題とし、既存のキーカスタマイズ設定すべてを定式化できることを証明したことを大きく評価したい。

11. 今後の課題

現在の violets 言語の仕様では、グルーピング表記とモジュール表記の導入によって状態遷移を書き下すための記述量は減っているが、新しく正規表現の量指定子の導入などして、ユーザが直接記述しなければならない状態遷移の負担を軽減できるユーザフレンドリーな記法の拡張が必要である。また、現在入力しているアプリケーションの対象によってキーマッピング設定の切り替えができるように、ブラウザで閲覧している URL 毎のキーマッピングの切り替えにも対応できると良い。Web サイトの充実も今後の課題である。