

1. 担当 PM

竹迫 良範（株式会社リクルート データプロダクトユニット ユニット長）

2. クリエータ氏名

竹村 太希（慶應義塾大学 環境情報学部環境情報学科）

3. 委託金支払額

2,736,000 円

4. テーマ名

翻訳 IME と Input Method 抽象化レイヤの開発

5. 関連 Web サイト

- ホームページ : <https://implane.kekeho.net>
- GitHub Organization : <https://github.com/implane>

6. テーマ概要

日本語を入力すると英語（あるいはその他の言語）に翻訳してアプリケーションに入力する翻訳 IME を開発した。高品質な機械翻訳サービスを IME というインターフェースで提供することで、ユーザはどんなアプリケーションでも煩わしさを感じることなく自然に外国語を入力できるようになった。同時に、各プラットフォームにそれぞれ存在する Input Method のインターフェースを共通化する抽象化レイヤと通信プロトコルを開発した。これにより、アプリケーション開発者が変換ロジックの実装のみにフォーカスして、簡単にマルチプラットフォーム対応の独自 IME を実装できるようになった。

7. 採択理由

既存の翻訳アプリは、Web ブラウザや文書作成ソフト毎に独自のプラグインを持っていたり、翻訳のための独自の UI 入出力画面を持っていたりするなど、主に文章作成の段階で使われることを想定している。たとえば、オンラインゲームで日本語話者と English speaker とでリアルタイムでチャットをするときに、既存の翻訳アプリを使おうと思っても画面越しで翻訳文章のコピーアンドペー

ストの作業が発生してしまいコミュニケーション効率が良くない。

提案者の開発した独自に抽象化した IME レイヤで日英翻訳のような変換が実行されるようになれば、オンラインゲームの他に、Slack や Discord などのチャットツールなどでも、IME で日本語文章を入力すると英語に自動翻訳されて気軽にコミュニケーションが取れるようになる。日本語入力 IME の実装は OS 毎に実装依存の部分が多いが、Input Method 抽象化する通信プロトコルも開発する計画も提案の中にあるので、今後、自由な発想で気軽に IME 開発ができるような未来が実現することを期待して採択した。

8. 開発目標

従来の Web 翻訳サービスにおけるユーザビリティの課題を解決するため、機械翻訳を IME というユーザインターフェースで提供することを開発目標とした。これにより、日本語を入力してからアプリケーションに英文を渡すまでの時間を大幅に短縮することができる。また IM のプログラミングインタフェースを持っていない問題に対処するため、各種 IM の上にインターフェースを共通化した抽象化レイヤを開発することも目標とした。

9. 進捗概要

ひらがなを入力すると英語に翻訳してアプリケーションに入力する IME 「Konjac」を実装した。実装に使用したプログラミング言語は Nim で、かな漢字変換には libkkc を、翻訳には DeepL API を用いた。プラットフォームによって IM のインターフェースが異なるという問題に対処し、開発者が容易にマルチプラットフォーム対応の IME を実装できるようにするため、IM 抽象化レイヤ 「IMPlane」を設計・実装した。また、IME と IMPlane がやりとりをするための通信プロトコルとして、IMProtocol を開発した。IM と IMPlane と IMProtocol と IME の関係を図 1 に示す。

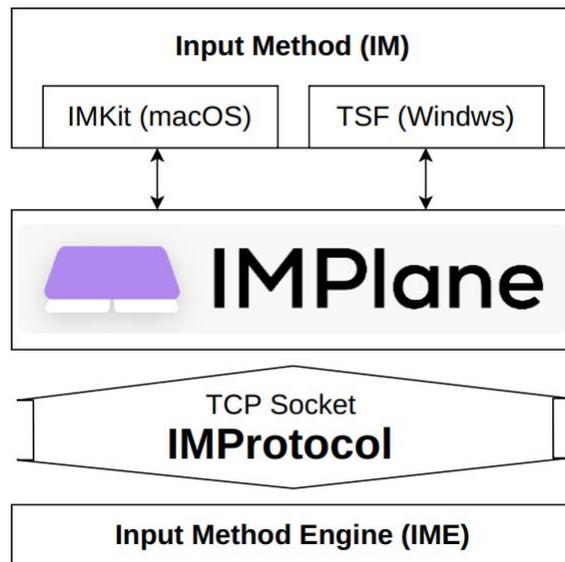


図 1 : IM と IMPlane と IMProtocol と IME の関係

IMPlane はデーモンとして起動し、OS の IM に接続して IME として振る舞う。その後、IMPlane と IMProtocol に対応した各 IME 間で TCP コネクションを確立する。日本語 IME の通信フローを図 2 に示す。ユーザからのキー入力が発生したタイミングで、IM が IMPlane に Input リクエストを送信し入力を通知する。ユーザが変換キーを押すと Convert リクエストを送信する。ユーザが変換候補を選択した際には Selected リクエストを送信する。また、IME から SetPreedit レスポンスを受け取ると、アプリケーションウィンドウ上にプリエディット文字列を描画する。SetCandidates レスポンスを受け取ると、変換候補を描画し、ユーザに選択を促す。Done レスポンスを受け取ると、確定文字列をアプリケーションへ渡し、状態をリセットする。

本プロジェクト期間中、IMPlane を macOS 上に実装した。実装に使用した言語は Swift である。また、IMPlane は複数の IME を接続可能とした。IMProtocol に準拠した IME として、ひらがなを入力すると英語に翻訳してアプリケーションに入力する IME 「Konjac」を実装した。加えて、ホスト名・ポート番号・認証キーなどの接続情報を編集する GUI エディタも実装した。

10. プロジェクト評価

macOS 向け IMPlane の実装が一通り完了した。その上で自作 IME を開発することができ、翻訳 IME だけではなく、流行の ChatGPT と IME を連携できるモックアップも開発し、自作 IME 時代の到来を予感させた。

11. 今後の課題

現在の IMPlane は macOS 向けの実装が完了しているが、まだ開発途中の Windows、Linux への移植を進めていく必要がある。また、IMPlane、IMProtocol

の利用を促進するため、ドキュメント整備に力を入れる必要がある。更に、プロトコルの詳細を知らなくても IME を開発することができるようにするため、代表的なプログラミング言語向けに SDK を作成することも検討が必要である。

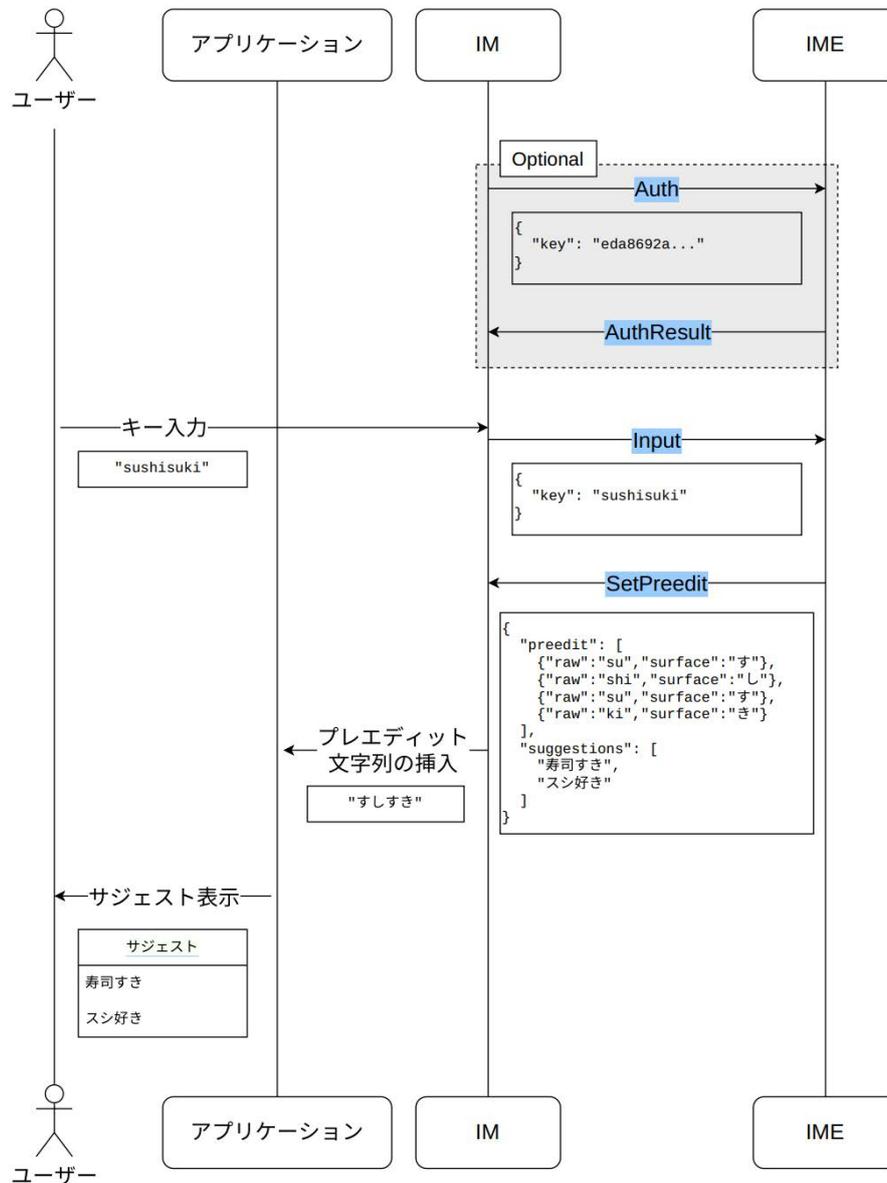


図 2：日本語 IME の通信フロー