翻訳IMEとInput Method抽象化レイヤの開発

― 言語の壁を取り払い，あらゆる言語圏の人々に快適なデジタルライフを ―

1．背景

1.1．翻訳IME

近年，ディープラーニング技術の発展により自然言語処理技術も進化を遂げている．特に機械翻訳タスクにおいては，従来の辞書ベースの翻訳と比べて，より自然な翻訳を達成している．

一般的な翻訳サービスは，Webアプリケーションの形でエンドユーザに提供されていることが多い．これらの多くは，Webアプリケーション内のテキストボックスにサポートしている言語Xの文字列を入力すると，別のサポートしている言語Yに翻訳した文字列を出力するという挙動をとる．こうしたサービスは，ブラウザのある環境であればどこでも高品質な機械翻訳を利用できるため，近年急速に普及が進んでいる．

しかしながら，こうしたサービスはユーザビリティに課題を抱えている．ユーザは文字列を翻訳する際に，ブラウザを立ち上げ，翻訳サービスのWebサイトにアクセスし，テキストボックスにフォーカスをあて，文字列を入力するという一連の操作ステップを踏む必要がある．特にテキストチャットなど短い文章を頻繁に翻訳するといったケースでは，1文字あたりの操作ステップに多くの時間を要してしまう．

1.2．Input Method抽象化レイヤ

一般的なコンピュータに付属するキーボードは，各キーにアルファベット・数字・いくつかの記号が割り当てられている．したがって，直接漢字かな交じり文を入力することは出来ない．そこでかな漢字変換と呼ばれる方法で漢字かな交じり文を入力する方法が一般的となっている．

かな漢字変換機能を備えたシステムは，一般的にFig. 1のように複数のコンポーネントによって構成されている．ユーザが発生させたキー入力イベントは，Input Method（IM）を経由してInput Method Engine（IME）に渡される．

そしてIMEによって変換された文字列がIMを経由して，アプリケーションに渡されることになる．こうしてかな漢字変換が実現されている．

IMは，OSごとにいくつか存在する．LinuxにはiBusやFcitx，macOSにはInput Method Kit（IMKit），WindowsにはText Service Framework（TSF）などがそれぞれ存在し，これらは共通したプログラミングインタフェースを持っていない．
そのため，複数のプラットフォームに対応したIMEを実装したい開発者は，各IM向けにコードを実装する必要性に迫られる．

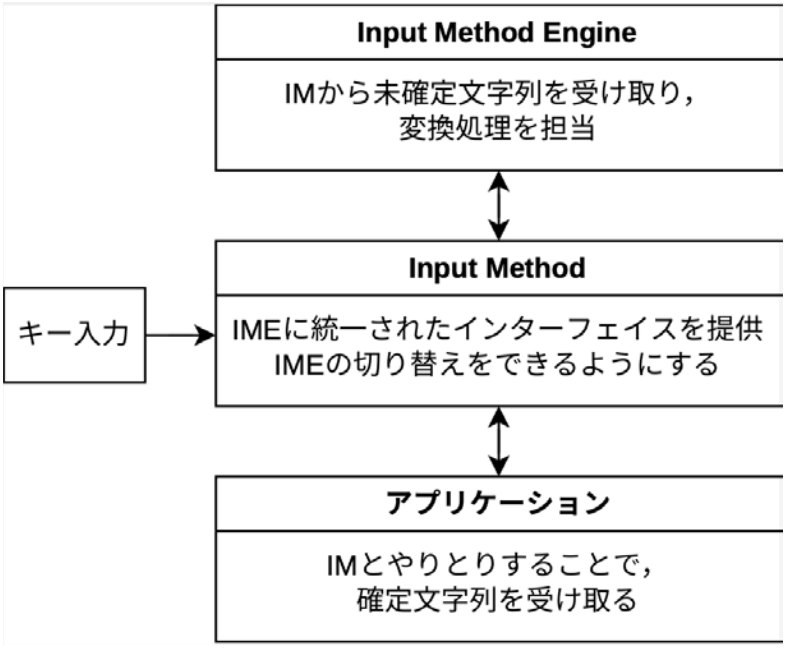

Fig. 1 日本語入力環境が整ったシステムの構成図

2. 目的

1.1節で説明した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UIで提供する．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．

また1.2節で説明した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の上にインタフェースを共通化した抽象化レイヤを開発する．

3. 開発の内容

3.1. 翻訳IMEの開発

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」（Fig. 2）を実装した．実装に使用したプログラミング言語はNimである．かな漢字変換にはlibkkcを，翻訳にはDeepL APIを用いた．

Fig. 2 Konjacのロゴ

3.2. Input Method抽象化レイヤの開発

プラットフォームによってIMのインタ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」（Fig. 3）を設計・実装した．また，IMEとIMPlaneがやりとりをするためのプロトコルとして，IMProtocolを開発した．

Fig. 3 IMPlaneのロゴ

3.2.1. IMProtocolの策定

本節では，新たに策定したInput Methodと IMEの間の通信プロトコルである IMProtocolについて説明する．

IMProtocolの大まかなデザインを決めるにあたり，IMとIMEの役割を明確に分離させることにした．IMには，キーボードからの入力の受け付け・アプリケーションへの文字列の受け渡し・変換候補ウィンドウなどの画面描画・ステート管理などの役割をもたせることとした．IMEは，文字列の変換処理のみを担うこととした．

IMProtocolによる通信を行う際の伝送路は，TCP Socketを用いることとした．これによりネットワーク透過性が生まれ，IMとIMEが異なるマシンに存在していても機能するようになっている．また，TCP Socketを用いたプログラミングは多くのプログラマに親しまれており，開発における学習コストを低く抑えることが期待できる．

IMProtocolはステートレスなプロトコルとして設計した．これによりIME開発者はステート管理の複雑性を考慮する必要がなくなり，変換ロジックの実装に集中することができる．また将来のバージョンアップに備えて，なるべく後方互換性を保つことのできるようなデータフォーマットにした．IMProtocolのリクエストは，以下のフィールドから構成される．

- **Header**: メタデータを格納する．IMPlaneがフォーカスしているアプリケーションのユニークなIDなどを含める．
- **Method**: リクエストの種別を指定する．
  - Auth: 認証認可のリクエストをIMEに通知する．
  - Input: 文字列の入力をIMEに通知する．
  - Convert: 変換リクエストをIMEに通知する．
  - Selected: 変換候補の選択結果をIMEに通知する．
- **Value**: Methodに応じたデータを格納する．

レスポンスは，以下のフィールドから構成される．

- **Header**: メタデータを格納する．
- **Status**: レスポンスの成否を表現する．
- **Method**: レスポンスの種別を指定する．リクエストのMethodに関わらず，IMEは好きなレスポンスを返すことができる．

  - AuthResult: 認証認可の結果をIMに通知する．Authリクエストに対しては，AuthResult以外のレスポンスを返してはならない．
  - SetPreedit: IMに対してプリエディット文字列を通知する．SetPreeditを受け取ったIMは，プリエディット文字列をアプリケーション画面に描画する．
  - Converted: IMに対して変換候補を通知する．Convertedを受け取ったIMPlaneは，変換候補リストを画面に描画し，ユーザへ候補選択を促す．
  - Done: IMに対してアプリケーションに渡す確定文字列を通知する．Doneを受け取ったIMは，プリエディット文字列をクリアし，確定文字列をアプリケーションへ渡す．また，ステートの初期化を行う．
- **Value**: Methodに応じたデータを格納する．

3.2.2. IMPlaneの開発

Input Method抽象化レイヤを設計し，macOS向けに実装した．実装に用いた言語はSwiftである．

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．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．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．

IMPlaneは複数のIMEを接続可能とした．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．

IMPlaneはTCP Socketを用いてIMEと通信しており，特にインターネット上にIMEを設置した場合に遅延が無視できない．この問題に対し，投機的実行を実装することで対処するなどの工夫も行った．

3.3. その他開発に取り組んだこと

本プロジェクト期間中，Konjacに用いる機械翻訳モデルを自前で用意するため，以下の開発にも取り組んだ．現在も開発途中であり，完成を目指して引き続き取り組んでいく．

- Transformerベースの翻訳モデルの開発
- 日本語文・英語文のペアを収集するWebクローラの開発
- Transformerベースのかな漢字変換モデルの開発
- かな漢字混じり文・ひらがな文のペアを収集するWebクローラの開発

また，IMPlaneの他プラットフォームへの移植にも取り組んだ．現在WindowsおよびLinux（Wayland）向けの実装に取り組んでおり，今後も引き続き開発を進めていく予定である．

4. 従来の技術との相違

今回開発したKonjacと既存の翻訳サービスを比較し，その優位性を以下に記す．

・DeepL・Google翻訳

ディープラーニング技術を用いており，自然な翻訳を得意としている．現在はウェブサイト・デスクトップアプリケーション・Chrome拡張機能などの形態でサービス提供を行っている．DeepLのChrome拡張機能は，あらゆるWebサイトのテキストフォームにおいて，入力した日本語をその場で英語に翻訳することができる，その点Konjacと同等のことが行えるが，Webアプリケーションにしか対応できないことが難点である．KonjacのようにIMEの形態であれば，あらゆるデスクトップアプリケーションにおいて使用することができる．

・Baidu IME for エキサイト翻訳

Baidu IMEにエキサイト翻訳ボタンを搭載したもの．エキサイト翻訳ボタンをクリックするとポップアップが立ち上がり，その場で手軽に翻訳機能を利用することができる．翻訳機能はIMEに統合されているわけではなく，使用感はあくまでデスクトップアプリケーションに文字列を入力して翻訳文を得る方法と変わらない．また，エキサイト翻訳自体が辞書ベースであり，ディープラーニングを活用した翻訳サービスと比較して自然な翻訳は望めない．

・Gboard

Googleが提供するモバイル向け多言語IME．翻訳機能が搭載されており，入力中の文字列をその場で他言語に翻訳することができる．モバイルにしか対応しておらず，デスクトップ環境では使用できない．

また，今回開発したIMPlaneについては同様の製品を確認できなかった．

5. 期待される効果

Konjacにより，オンラインコミュニケーションにおける言語の壁を取り払うことができ，世界中の人々がよりスムーズに繋がることができるようになると期待している．

またIMPlaneとIMProtocolにより，最低限の知識で誰でもIME開発者になれる未来が到来すると期待している．近い将来，無数のIMEが生まれ，すべての言語圏の人々が快適なデジタルライフを送ることができるようになると考えている．

6. 普及の見通し

Konjacは，今後サブスクリプションサービスとしてエンドユーザに提供することを検討している．またIMPlaneは，ドキュメントを整備した上でオープンソースソフトウェアとして公開することを検討している．

7. クリエータ名（所属）

竹村 太希（慶應義塾大学 環境情報学部 環境情報学科）

関連URL

- https://implane.kekeho.net
  本プロジェクトのホームページである．IMPlaneおよびIMProtocolについて紹介している．
- https://github.com/implane
  本プロジェクトのGitHub Organizationである．