

文字入力システムの再構築

徳永拓之
亀田大輔
表雅仁

平成 15 年 10 月 3 日

1 動機

ここ数年における Gnome, KDE などのデスクトップ環境の進化は目覚ましいものがあり、英語などの言語で使うには、既に実務に耐えられるだけの品質を備えつつある。

しかし、FreeSoftware における日本語入力環境は、ここ 10 年近く進歩の歩みが止まっており、残されている問題はまだまだ多い。これらの問題点を解決すること無しに、日本における FreeSoftware で構成されたデスクトップ環境の普及はありえないと言っても過言ではない。

2 問題点

この章では、現在のフリーな日本語入力環境が抱えている問題点について述べる。

2.1 複雑かつ X 依存な XIM

XIM は現在最もよく用いられているインプットメソッドのためのフレームワークであるが、いくつかの問題点がある。

まず、X Input Method という名前が示す通り、X Window System に大きく依存している。次に、XIM は複雑であり、理解し、メンテナンスを行うための負担が大きい。負担が大き過ぎるせいかわからないが、もっともよく使われている XIM Server である kinput2 はほとんどメンテナンスされていない。

2.2 Qtにおける入力機構

デファクトスタンダードな GUI ツールキットは Gtk+,Qt のふたつに収束してしまった感があるが、Gtk+がインプットメソッドのために新しく immodule というシンプルかつ必要十分な機能を備えた機構を実装したのに比べ、Qt には今のところ貧弱かつ微妙に Buggy な XIM のサポートしか存在しない。(もっとも、微妙に Buggy であるのは Gtk+の immodule も同じではあるが。)

2.3 GUIフロントエンド

まともに使える状態の辞書ツールなどの GUI フロントエンドが存在しないというのは、初心者にとって大きな問題である。全ての人間がコマンドラインを使いこなすようになるわけではない事を考えると、辞書を編集するツール、設定を変更するツールなどは必要である。

2.4 セキュリティ

現在多く使われている canna や FreeWnn では、ユーザ辞書は他人からまる見えである。これは変換サーバが独自の権限で動作するという設計からすると自然な事ではあるが、セキュリティの観点からみてあまり望ましくない。

新しい変換エンジンである anthy や PRIME ではセキュリティの問題が考慮され、変換エンジンはユーザ権限で動作するが、これらの新しい Software はあまりメジャーであるとは言いがたい。

3 問題解決のために

我々は前章の問題に対処するために、いくつかのソフトウェアを改善し、いくつかのソフトウェアを新規に開発することにした。具体的には、多言語入力ライブラリ uim を改善し、Qt に Gtk+における immodule に相当する機能を実装し、新規に辞書ツールなどの GUI フロントエンドを開発することにした。

4 開発内容

4.1 多言語入力ライブラリ uim

uim とは、anthy project の田畑氏らが開発を始めた多言語入力ライブラリである。まず最初に、uim の開発動機を挙げる。

- InputMethod の GUI 依存部分と非依存を分離する事で、ツールキットやウィンドウシステムに依存しないインプットメソッドのためのフレームワークを用意したい
- アプリケーションとインプットメソッドの間に中間層を一つ挟むことで、組み合わせ数の爆発を防ぎ、開発者の労力を軽減したい

同じような目的を持つプロジェクトとして、iiimf(internet/intranet input-method framework) というプロジェクトが存在する。uim の開発開始当初、既に iiimf が存在したが、わざわざ anthy project の田畑氏が uim の開発を始めた理由として、

- iiimf は、日本語や中国語以外の単純な input method を使う人々に対して無用な負担をかけてしまう。彼らに負担をかける事で、めんどくさいから日本語や中国語のサポートはやめよう、という話になっては困る。
- サーバー/クライアント方式であり、プライバシーやセキュリティ面があまり考慮されていない

などが挙げられている。実際に iiimf の仕様書を眺めながら iiimf の LanguageEngine としてかな漢字変換を実装しようと考えてみると、いくつかの問題が見つかる。iiimf では変換サーバが独自の権限で動作しており、LanguageEngine も複数のユーザからのイベントを自力で処理しなければならない。

かな漢字変換においては各ユーザ固有のユーザ辞書、現在の変換状態などを保存しておく必要があるが、iiimf ではサーバ内で各ユーザの情報を分離するのが困難である。また、ユーザ辞書をユーザの権限で保存する事が難しい。

これらはネットワークを越えて変換サーバと通信する事を念頭に置いている事によるトレードオフであるので、一概に否定できるものではないが、現行の iiimf の実装では、プライバシーに対する配慮が足りないのは事実である。iiimf はこれらの部分を隠蔽する層を用意する必要があると思われる。

これに比べ、uim は基本的に共有ライブラリであるので、権限管理については OS の管理機能をそのまま使うことができる。変換エンジンはユーザ権限で動作するので、ユーザ辞書のユーザ権限での保存に関して特に気にする必要はない。この点においては uim の方が優れていると言える。

しかし、当然ながら uim にも課題点はある。

- ドキュメントがたりない
- 仕様が固まっていない

これらは非常に大きな問題であるが、開発を続けていけば解決できる問題である。

ネットワーク越しに通信しないというのも、場合によっては課題点として挙げられるかもしれない。複数マシンでの辞書の共有などに関しては、変換サーバを1つ用意してそこでユーザ辞書を共有するモデルが自然である。そこで、Cannaサーバとのネットワーク越しの通信機能の実装なども予定している。

4.1.1 普及のための戦略

uimのようなインプットメソッド用フレームワークは普及しなければあまり価値がない。対応するアプリケーション、変換エンジンが増える事で価値が増大するからだ。

そこで、普及させるための戦略を考える。ここでの普及対象は主に海外のアプリケーション開発者である。インプットメソッドに対する知識や関心は薄いアプリケーションを開発している、というような人々の間でまず普及させる事が一番重要である。

この層に普及する、つまり対応するアプリケーションが増えれば、結果として多くの人がuimを使うようになるのは自明である。そして、普通のユーザが増える事で、開発者に対してuimに対応させるためのプレッシャーがかかるようになる。

- 対応するためのコストを下げる事

キーボードから直接すべての文字を入力できてしまうような言語を使っている開発者にインプットメソッドの重要性を実感してもらう事は不可能だ。(そもそもそのような開発者にとってインプットメソッドは実際に重要ではない。)

そこで、インプットメソッドの重要性を説くことも大事であるが、対応のためのコストを下げる努力も重要である。

そのために、入力に必須な機能とそうでない機能を分ける事、実装のためのドキュメントの充実などを考えている。

- インプットメソッドがあると便利だと実感させる事

前項と矛盾するが、キーボードから直接すべての文字を入力できてしまうような言語を使っている人間も便利に使えるソフトウェアがインプットメソッドの仕組みを利用して開発可能である。予測入力とスペルチェックの二種類である。予測入力は英語などの言語においても、入力速度向上のために有効な方法である。既にタッチタイプを修得してしまった人間に対しては入力速度の向上にはつながらないかもしれないが、初心者にとってはかなりの速度向上に繋がる。また、これまでスペルチェッカはワードプロセッサなどでしか使えなかったが、インプット

メソッドの仕組みを利用すれば、テキストを入力する全ての場面でスペルチェック機能が利用できるようになる。

- インストールが簡単である事

ちょっと試してみようかな、と思った時にすぐに試せる程に簡単に動作するソフトウェアであった方が、確実に普及に弾みがつけられる。現在の iimf は、まだこの段階には達していない。

4.2 immodule for Qt

現在の Qt の実装では、インプットメソッドとして XIM しかサポートしていない。しかも XIM に関連する処理が複数のクラスに散らばってしまっている。

そこで、新しく QInputContext というクラスを作り、インプットメソッドの処理をひとつのクラスに局所化すると同時に、Gtk+ の immodule と同様の機構を実装し、XIM モジュールや uim モジュール、iimf モジュールといったモジュールを実装する事で様々なインプットメソッドに対応する柔軟性を確保する。また、Qt の Plugin 機構を利用することで、インプットメソッドを動的に変更可能にする。

開発成果は Qt4 に取り入れられる事を目標としている。

4.3 辞書ツール栖 (sumika)

現在の日本語入力環境には、単語登録や編集、設定などをインターフェースが欠けている。辞書ツール sumika は辞書の管理、単語登録、設定といった基本的機能はもちろんの事、コミュニティなどでのユーザ辞書の共有、日本語入力システム間の登録単語の同期を可能にするツールとすることを目標としている。また、内部にも辞書データを持つことにより、辞書の同期、共有を用意に行うことができるようにする予定である。

GUI インターフェースは Gtk+, Qt の両方への対応を予定している。また、コンソールベースのツールも作成する予定である。

5 開発状況

このプロジェクトは IPA(情報処理振興事業協会) の平成 15 年度未踏ソフトウェア創造事業の支援を受けており、原稿執筆段階ではプロジェクトは始まったばかりである。

uim に関しては支援を受ける前から開発が開始されていたので、現在のところ日本語の変換エンジンとして anthy, PRIME が利用できるようになって

いる。また、SKK,t-code のサブセットも実装されており、動作している。対応しているアプリケーションは今のところ、Gtk+2.x での immodule に対応しているもののみである。また、ドキュメントもほとんどない。

これからは API の再設計を行い、ドキュメントを整えてから再実装にとりかかる予定である。

immodule for Qt は現在設計を行っている段階である。

sumika に関しては現在 SourceForge.jp にて初期バージョンが公開されている。現段階では Anthy の個人辞書の読みこみ、単語登録などができる状態になっている。

6 参考文献, URL

- 田畑氏による iimf の疑問点の説明
<http://anthy.dyndns.info/?IIIMF>
- uim 公式サイト
<http://anthy.sourceforge.jp/>
- 辞書ツール栖公式サイト
<http://sumika.sourceforge.jp/>
- Gtk+公式サイト
<http://www.gtk.org/>
- Qt 公式サイト
<http://www.trolltech.com/products/qt/>