

プロセス間通信を用いたデータ入力支援環境

7Q-6

岡本 一弘

日立ソフトウェアエンジニアリング（株）

1. はじめに

GUI環境が一般化する中で、データ入力環境についても様々な改善がなされている。しかし、従来の入力支援はアプリケーションに組み込みのものが主であり、アプリケーションプログラムに独立に使用されているのは日本語入力のためのかな漢字変換のような限られた例しかない。

汎用的なGUI環境を提供しようという試みとしては、登内らによる、GUI部品をさらに細かい機能部品に分け、カスタマイズを可能にする方法がある[1]。これは、GUIが提供すべき機能構造のアプリケーションへの依存性に着目した方法であるが、結局は各アプリケーションについて個別にGUIを構築する必要がある。

この問題はGUI環境が提供すべき機能がアプリケーション、特にそのアプリケーションが処理するデータの性質や内容に依存していることに起因している。つまり、各アプリケーションに対する入力支援の実現は其中で扱うデータの性質や内容をあらかじめ知らないと難しい。

ここではプロセス間通信機能を用いてアプリケーションの扱うデータに関する情報を取得し、それを用いてアプリケーションに独立に入力支援を行う方式を提案する。さらに、この方式を実現する場合の問題点について検討する。

2. 入力支援機能

ここで提案する入力支援の方式は複数の入力支援機能を汎用的に利用可能な形で提供することを考えているが、当面の主な利用分野としてスプレッドシートや定型的なデータ入力システムを想定している。このようなデータ入力支援方式を検討するにあたって、入力支援機能としては以下のものを取り上げることにした。それぞれについて機能の実現に必要な情報の内容とともに述べる。

(1) メニュー選択入力

その時点で入力可能な値の候補をメニューの形で表示し、そこから選択することで入力できるようにする機能である。実現には該当欄に入力可能な値のリストが必要となる。

(2) 補完

文字列入力の際に、途中まで入力された文字列について、その文字列で始まる値の候補を表示して選択させたり、特定できる場合にはその候補と置き換えたりする機能である。例えば、曜日の欄に「M」と入力してこの機能呼び出せば「Monday」と置き換えられる、といった形で利用できる。実現には該当欄に入力可能な値のリストが必要となる。

(3) デフォルト値設定

データの入力を受け付ける前に、頻繁に設定する値をあらかじめ入力フィールドに入れておく機能である。実現には入力項目に対するデフォルト値が必要となる。

(4) 入力文字の制限

文字列入力の際に、入力されるべき文字列以外の文字については入力できないようにする（無視するか、警告を発生する）機能である。例えば、生年月日の月の欄には1から12までの整数しか入力できないようにする、といった場合がこれにあたる。実現には入力項目のデータ型が必要となる。

ここで取り上げた「入力可能な値のリスト」、「デフォルト値」、「データ型」などの情報はアプリケーションプログラムではなく、アプリケーションの扱うデータに関するものである。したがって、これらの入力支援機能を実現するためにはアプリケーションの扱うデータに関する情報を得る必要がある。

3. 入力支援方式の概要

今回提案する入力支援方式では図1のように入力支援機能を前処理プログラムとして実現する。前処理プログラムは入力・表示装置からの入力を受け付け、処理した結果をアプリケーションプログラムへの入力として渡す。アプリケーションプログラムでは従来の入力に対するのと同様に入出力処理を行う。

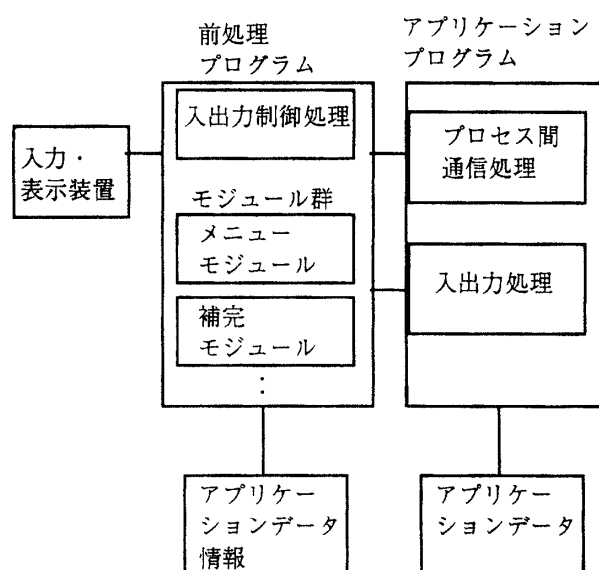


図1 入力支援システムの構成

Data Input Support Environment Using Inter-process Communication

Kazuhiro Okamoto

Hitachi Software Engineering Co.,Ltd.

6-81 Onoe-cho, Naka-ku, Yokohama, Kanagawa 231, Japan

入力支援機能の実現に必要なアプリケーションの扱うデータに関する情報は前処理プログラムとアプリケーションプログラム間でのプロセス間通信によって取得する。前処理プログラムには各入力支援機能を実現するモジュール群を用意し、ユーザの設定にしたがって起動される。前処理プログラムのモジュールを追加・変更することによって入力支援機能の追加・変更が行える。したがって、2. で挙げた4種類以外の入力支援機能もモジュールを用意すれば利用できる。また、プロセス間通信に関するインタフェースを用意できるアプリケーションプログラムであれば、この前処理プログラムは共通に利用可能である。

4. 入力支援機能の実現

ここで入力支援機能の実現方法を検討するために例題を用いて考える。図2はスプレッドシート処理を行うアプリケーションでの名簿作成に入力支援機能を適用した場合の画面イメージを表した図である。

The figure shows a screenshot of a spreadsheet application. At the top, there is a menu bar with 'ファイル', '編集', '移動', 'ウインドウ', and 'ヘルプ'. Below the menu is a table titled '名簿' (Name Register) with three columns: '氏名' (Name), '出身学科' (Department of Origin), and '電話番号' (Phone Number). The table contains five rows of data:

氏名	出身学科	電話番号
山田光一	情報工学科	03-7313-0254
横井和男	機械工学科	03-6413-2025
加藤景子	機械工学科	03-8311-2907
森康弘	電気工学科	045-013-6890
田中英明	—	

Below the table is a dialog box titled '補完入力' (Input Completion). It has a label '出身学科' (Department of Origin) and a text input field containing '情報'. Below the input field is a list box with two items: '1 情報工学科' and '2 情報科学科'. At the bottom of the dialog are three buttons: '補完' (Complete), '確定' (OK), and '取消' (Cancel).

図2 画面イメージ

ここでは、ユーザが指定したカラム「出身学科」への入力に補完を適用した場合を示している。アプリケーション側ではマクロ機能を用いた処理としてプロセス間通信に対応した処理が記述してあるものとしている。プロセス間通信によるアプリケーションプログラムからのデータ取得は前処理プログラムの起動時およびアプリケーションプログラムでの新しいデータ（スプレッドシート）の読み込み時に行う。補完処理に必要な「入力可能な値のリスト」はこのプロセス間通信によって取得し、これを利用して処理を実現する。

5. 実現上の問題点の検討

実現上の問題点としては以下のようなものが考えられ

る。ここでは、それぞれの対策も合わせて検討する。

(1) プロセス間通信のオーバーヘッド

入力支援機能の種類およびアプリケーションの扱うデータのサイズによっては入力データの属性に関する情報の通信量がかかなり多くなり、そのためのオーバーヘッドが問題となる可能性がある。対策としては前処理プログラムの側でアプリケーションの扱うデータに関する情報の保持を行う方法や入力データの履歴取得で補う方法などが考えられる。

(2) プロセス間通信のインタフェースプログラムの準備

アプリケーションプログラムから処理するデータに関する情報をプロセス間通信によって前処理プログラムに送る処理のインタフェースプログラムは各アプリケーションごとに準備しなければならない。対策としては入力支援機能の種類と内容に合わせてできる限り簡単なインタフェースで通信できるようにするしかない。

(3) アプリケーションデータと関連情報の整合性

(1) で述べた前処理プログラムでの各アプリケーションの扱うデータに関する情報の保持は元となるデータとの不整合を招くおそれがある。つまり、前処理プログラムを利用しないでデータ更新が行われた場合に前処理プログラムの持つアプリケーションの扱うデータに関する情報とデータ自身との間に内容の食い違いが生じ、前処理プログラムの不具合や利用効果の減少につながるというものである。対策としては不整合の検出と検出された際のプロセス間通信によるアプリケーションの扱うデータに関する情報の更新を行う方法が考えられる。

(4) 入力操作手順の整合性

アプリケーションプログラムの用意した機能と前処理プログラムが用意した入力支援機能の整合性がどれだけとれるかが問題である。特にトータルでの処理手順が使いやすいものになるかどうかの検討が重要である。例えば、画面のレイアウト、操作手順等についてプロトタイプなどによる検証を行うことが考えられる。また、操作手順の細かい制御を行いたい場合にはより緊密なプロセス間通信が必要になるかもしれない。

6. おわりに

プロセス間通信を用いてアプリケーションの扱うデータに関する情報を取得し、それを用いて入力支援を行う方式を提案し、その実現上の問題点を検討した。プロセス間通信のインタフェース等を具体化し、オーバーヘッドや操作性などの問題点についての評価と改善を行うことが今後の課題である。

<参考文献>

- [1] 登内敏夫、中島震：組み合わせオブジェクトに基づくGUIライブラリ、コンピュータソフトウェア Vol.12, No.3 (1995)