

カスタマイジングを容易にする UI の構築方法

2J-8

上野 和彦 山本 隆広 黒木 宏明

NTT ソフトウェア研究所

1 はじめに

現在の多くのアプリケーション (AP) のユーザインタフェース (UI) は、平均的な使用状態に合わせて作られたものであり、特殊な環境での使用はあまり考慮に入っていない。例えば、視覚障害者が使う場合は、音声によるナビゲーションや画面情報の読上げなどが必要となる。^[1]しかし、既存の AP の入出力方法を変更する場合、C などのプログラミング言語で記述されたソースプログラムを解析する必要がある。また、ソースプログラムが手に入らない場合も多い。そこで、対話型の GUI アプリケーションのエンドユーザを対象としたカスタマイズについて考案した。

本稿では、UI を構成する部品であるユーザインタフェースオブジェクト (UIO) と、AP の機能部分を利用するオペレーションとからなるメッセージフローでアプリケーションを定義する。そして、カスタマイズ時に、フローの変更によって UIO とオペレーションの結びつきを変える方法を提案する。

本方法では従来の方法と異なり、UI の変更は C ソースプログラムを必要としない。その代わりに、別に用意したフローファイルによりカスタマイズを行ない、カスタマイズ作業も容易になる。

2 UI 構築方法の概要

2.1 アプリケーションプログラムの構成

メッセージフロー

ここでは、アプリケーションプログラム A を、メッセージフロー F の集合であると定義する。

$$A = \bigcup_{i=1}^N F_i$$

メッセージフロー F は、入力用の UIO U_i と出力用の UIO U_o とその間で行う操作 (オペレーション) O よりなっていると定義する。

$$F = (U_i, O, U_o)$$

入力用 UIO U_i はユーザからのイベントを受けとってメッセージ M を送信し、出力用 UIO U_o はメッセ-

ジ M を受けとりユーザへ情報を伝える。

オペレーションはメッセージの変換を行い、ここが AP 機能の本体部分 (AP 機能部分) となる。

$$U_i = UserEvent \rightarrow M$$

$$U_o = M \rightarrow Display$$

$$O = M \rightarrow M$$

メッセージフローは、入口を U_i 、出口を U_o とする水道管のようなものであり、オペレーションは、水の組成を変えるフィルターのようなものである。

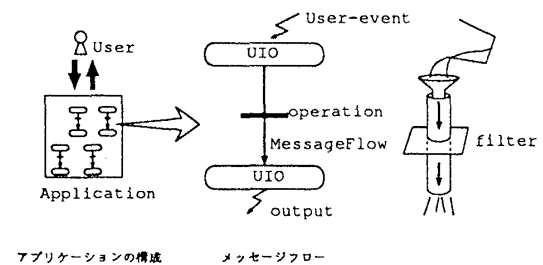


図 1: オブジェクトフロー

カスタマイズ方法

カスタマイズは、UIO とオペレーションを組み合わせるフローを加えたり変更したりすることで行う。例えば、通常ディスプレイに出力しているメッセージを音声で出力する場合は、音声出力用の UIO を追加し、新たにその UIO へ流れるフローを追加する。

2.2 従来の方法との比較

従来のウィジェットを使用したプログラムは、ボタンをマウスでクリックするなどのユーザのアクションに対して AP 機能部分であるコールバック関数を呼び出す。そして、コールバック関数で処理を行ない、ウィジェットの操作関数を使用し出力用ウィジェットに出力する。そのためカスタマイズは、コールバック関数を書き換えることになる。

ここで提案した方法は、ユーザのアクションをおこすと、UIO がメッセージを出し、そのフローにメッセージが流れます。そして、メッセージが出力用 UIO で出力されるまでの途中で、AP 機能部分にメッセージを

Method of UI for easy customizing
Kazuhiko UENO, Takahiro YAMAMOTO, Hiroaki KUROKI
NTT Software Laboratories

通して様々な処理を行なう。そのため、入出力 UI を変更したり、既存機能の組み合わせといったカスタマイズでは AP 機能部分を記述したソースプログラムを必要としない。そして、個々に独立した UIO や AP 機能部分や外部プログラムを組み合わせるフローを変更するだけなので、カスタマイズが容易になる。

3 実験

3.1 実験システムの概要

SunOS, X-window 上で Motif ツールキットと UIL を使用した簡単な実装を行った。(図 2)

今回実装を行ったシステムでは、UIO は Motif ウィジェットに対応する。そして、UIO の階層構成、リソース値などを UI 記述言語 (UIL) で記述する。

メッセージフローはフロー記述ファイルに記述する。その中のスクリプトでオペレーションを記述し、メッセージを変換したり、C の関数や外部プログラムを呼び出したりする。

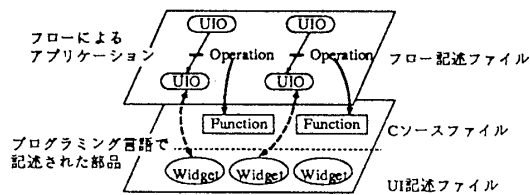


図 2: アプリケーション構成

3.2 実験例題

サンプルプログラムとして、mh コマンドを利用したメールリーダーを作成した。図 3 は、新メールを取り出す部分のフローである。

incButton が押されると、inc、grep コマンドが実行される。inc コマンドにより新たに来たメールの一覧が得られ、grep で余計な出力の部分を取り除く。それを各行ごとに分割し、それをメッセージ“add”とともに summeryList ウィジェットに送り、メールのリストを構成する。messageText ウィジェットは、メール数を知らせるメッセージに変換されたメッセージを受け、それを出力する。

3.3 カスタマイズ実験

今回の実験では、音声によるオペレーションを想定した実験を行った。ただし、音声を出力する UIO が無いため、その代わりにサブウィンドウや標準出力を出力 UIO として使用した。

行った作業は、ツールの操作ナビゲーションとメールの読み上げのための以下の項目である。

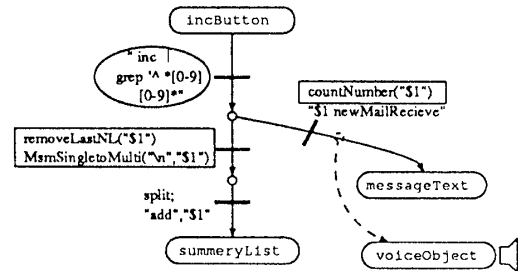


図 3: メールリーダー: 新メール取り込み

- プルダウンメニューでメニューをオープンした時のメニュー名の読み上げ
- メニュー項目選択時に選択カーソルがある項目の読み上げ
- 現在選択しているメールフォルダーの読み上げ
- 通常、メッセージ出力領域に出力されるメッセージの読み上げ (図 3 点線)
- メールの内容の読み上げ

カスタマイズ前のフローファイル約 150 行に対し、この実験で追加した行は約 20 行となった。

3.4 評価・考察

実験のようにメッセージの出力先を追加することや、すでにある機能を新しくボタンに割り振るといったことは容易になった。ただし、どのウィジェットがどんな役割をして、フローがどのように構成されているか知らなければ、困難である。また、フローを視覚化して、マウスでダイレクトに変更を行い、UI 構築ツールとうまく関係がとれれば、手間は減ると思われる。

4 まとめ

本稿では、UIO から UIO へのメッセージの流れと、その間のオペレーションで AP を構成し、その流れを変更することでカスタマイズする方法を提案した。これにより、音声出力用のウィジェットや外部プログラムなどを、実行ファイルに手を加えずに利用することが出来る。

参考文献

- [1] 栗原亨 “視覚障害者のための計算機教育”, bit vol.25, No.9, pp.22-34, 1993
- [2] McMillan, W.W. “Computing for Users with Special Needs and Models of Computer-Human Interaction”, CHI'92, May, 1992.
- [3] 兜木昭男他 “X-Window OSF/Motif プログラミング”, 日刊工業新聞社