拡張可能な C++ ソースコード・ブラウザ — ユーザインタフェース —

7Q−7

大平剛†, 三ッ井 欽一†, Shahram Javey‡
†日本アイ・ビー・エム(株)東京基礎研究所　‡IBM Canada, Toronto Laboratory

1 はじめに

本稿では、ブラウザのグラフィカルユーザインタフェース(GUI)に関する実装についてのべる。特に、ウィンドウの構成、各サブウィンドゥの機能、サブウィンドウ間の連係などのGUIそのものの設計とユーザインタフェース記述言語UIDLによる拡張性について。

2 開発方針

GUIの設計では，メニュー，ウィンドウレイアウト，ダイアログボックス等が変更の対象になる事が多い．我々はGUIの設計を効率よく行うために，変更が容易になるように，これらのウィンドウオブジェクトを制御するための，インタープリタ言語UIDLを導入した．ただし全ての制御をインタープリタで行うのではなく，柔軟性を求められる部分，例えばウィンドウレイアウト，メニューの追加/変更，コールバック処理，ダイアログボックスとの対話などはUIDLで記述し，実行効率が求められる部分，例えばデータベースへの問合わせ結果の表示（リスト，グラフ）等はC++で記述している．

3 設計

3.1 ユーザインタフェース

3.1.1 ウィンドウの構成

メインウィンドウは図1のようにメニューバー，複数のサブウィンドウをtilingすることができる領域，メッセージとフォーカスを表示するための領域を持つ．

各サブウィンドウは2つのコントロールバーと表示領域を持つ．最初のコントロールバーは各サブウィンドウに共通の搬入ボタン，自動搬入設定ボタン，フォーカス表示領域と，フォーカス・オプションメニューがある．次のコントロールバーは，問い合わせを作成するためのオプションメニューを持ち，サブウィンドウのタイプによって異なる．また各タイプに特有の機能を実行するためのボタンを持つ．

我々が2または4個のサブウィンドウによるtilingを考えているのは、複数のウィンドウを同時に参照しながらブラウザを利用することが多いと考えられるからである。もちろん、overlappingをして任意の個数のウィンドウを開くこともできる。overlappingのみを用意することも考えられるが、この場合、一般にウィンドウを操作する手間が多くなり、複数のウィンドウを利用するのが典型的であるならtilingのほうが有利である。

3.1.2 サブウィンドゥの機能

サブウィンドウには次の3つのタイプがあり，それぞれ次のような機能を持つ．

- リスト: 問合わせの結果をリスト表示する．
- グラフ: 問合わせの結果をグラフ表示する．指定したノード以降のサブグラフを拡張/省略表示する．グラフ全体を拡大/縮小表示する，ノードの色指定，ラインの色/スタイル指定する．
- テキスト: 問合わせの結果をテキスト表示し，目的の場所にカーソルを移動する．任意の部分文字列をフォーカスにする，またその文字列が何であるかをデータベースに問合わせる．エディタとしての編集機能を持つ．

サブウィンドウの問合せの作成の操作は、そのタイプに関係なく一般性を持っている。問合せは、特定のプログラムオブジェクト（ファイル，クラス，関数，変数など）をパラメタとして指定できるものと、そのようなパラメタを持たないものに分けられる。例えば、前者の例は、全てのクラス。後者は、あるクラスのメンバ関数。いずれのタイプのサブウィンドウでも、特定のオブジェクトを表示している矩形領域を選択することができる。例えば、グラフではノード、テキストでは任意の部分文字列がこれに相当する。選択されたオブジェクトは、現在のフォーカスとなり、後の問合せのパラメタとして使われる。

各サブウィンドウは、問合せを作成するためのメニュー群(オプションメニュー、ポップアップメニュー)を持っている。メニュー選択の組合せにより[2]で説明しているPrologにより記述されたデータベースの問合せ要求が生成され、実行される。各サブウィンドウに付属する搬入ボタンを押すと、現在のフォーカスがそのサブウィンドウのパラメタになる。

このように一連の問合せ動作は、サブウィンドウのタイプや場所に独立で共通なのでユーザにとって理解しやすい。

オプションメニューの組合せはフォーカスの有無やフォーカスのタイプによって，自動的に切り替わる．ポッ

Extensible C++ Source Code Browser — User Interface
Tsuyoshi Ohira†, Kin'ichi Mitsui†, and Shahram Javey‡
†IBM Research, Tokyo Research Laboratory
‡IBM Canada, Toronto Laboratory

プアップメニューはサブウィンドウ上での選択の有無や，選択されたオブジェクトのタイプによって自動的に切り替わる．これらの機能により，サブウィンドウは搬入されたフォーカスのタイプに合わせて，ブラウズ機能を動的に変化させる．

3.1.3 サブウィンドウ間の連係

各サブウィンドウは搬入ボタンを押す事でフォーカスを設定するが，ブラウズの対象と表現方法が固定している場合，いちいち搬入ボタンを押す事が煩わしくなる場合がある．これに対処するため，選択されたオブジェクトを自動的に他のサブウィンドウに搬入する機能を持つ．各サブウィンドウはサブウィンドウ間の関係を表現するための４つボタンを持ち，これを設定する事で自動的な搬入を実現する．この機能により，独立していたサブウィンドウ間に，主従関係を持たせる事ができる．

3.2 ユーザインタフェース記述言語

ユーザ自身がブラウザのGUIを容易に変更できるようにする。例えば、新しい問合せパターンを考案した場合、対応するメニューを加え、メニューの選択に対して適当な問合せを実際にデータベースに行ない、結果をサブウィンドウに表示する、という記述をユーザが行なう。この記述は実行時にシステムに読み込まれ解釈実行される。従って、もとのシステムを変更することなしにGUIの挙動を変えることができる。

よく言われるように、GUIはオブジェクト指向に基づく設計が適している。我々のブラウザでも、サブウィンドゥ、メニュー、ダイアログといった基本部品はC++のオブジェクトとして実装されている。記述言語では、このC++オブジェクトを実際に生成し、それらのオブジェクトの必要な機能をC++の外から呼び出すことを可能にする。言いかえると、ユーザは、基本機能を持った部品としてのオブジェクトを記述言語で組み合わせることによりGUIを構築する。

4 評価（考察・議論）

各サブウィンドウがユーザの興味によって，動的にブラウズ機能を変化させられる事は，新しくブラウズ・ウィンドウをオープン手間を省略する事ができ便利だと思われる．その半面，同じウィンドウを使い回すので，以前の状態に戻したい時に，やり直しが必要になる．ヒストリ機能が求められる．

問合せメニューの追加・変更、あるいはshellの別コマンドを呼び出すなど、GUIの基本的な枠組を崩さない程度のカスタマイズならユーザにとって十分容易に可能であると考えられる。

しかし、一般に、たとえばサブウィンドウの数を変えたりボタンの位置をかえるなどのレイアウトに関して高い柔軟性を持たせた場合には、比較的多くの記述を必要とするので容易とは言いがたい。我々の記述言語は、任意のウィンドウレイアウトが可能であるほど完全ではない。

この記述言語は、我々にとっては、コンパイル・リンクの回数をへらし、開発のサイクルを速める上で有効であった。

参考文献

[1] 三ッ井欽一, 久世和資, Shahram Javey: 拡張可能なC++ソースコードブラウザ - 基本設計, 第45回情報処理全国大会, 7Q-05, 1992.

[2] 中村宏明, 安田和, 三ッ井欽一, Shahram Javey: 拡張可能なC++ソースコードブラウザ - プログラムデータベース, 第45回情報処理全国大会, 7Q-06, 1992.

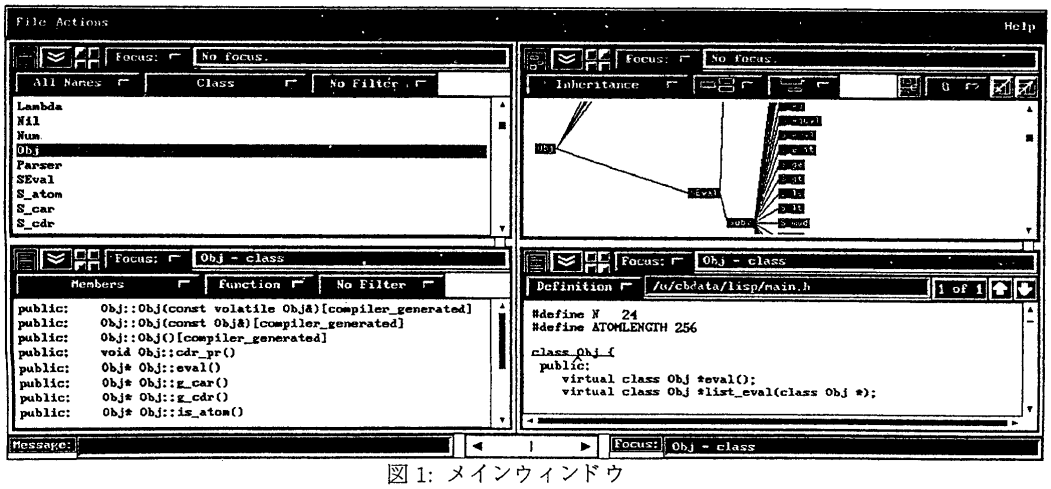

図 1: メインウィンドウ