

Operator Interface Using Tk Widget

Kazuro Furukawa, Norihiko Kamikubota and Shirou Kusano*

Photon Factory, National Laboratory for High Energy Physics (KEK)

*Mitsubishi Electric System & Service Engineering Co. Ltd.

Abstract

As accelerators got larger complex, it has become difficult to operate and understand whole accelerator. Thus it's important to make effort to reduce information and to build easily configurable user-interface for operators. We tested a building tool for graphical-user-interface called Tk in our Kek Linac control system. Tk was very efficient to develop user-interface for accelerator operation. Its feature and performance are described.

Tkウィジェットを利用したオペレータインターフェース

1、はじめに

近年、加速器は大型化に進みそれにともない制御点数も多くなってきた。そのため運転も複雑となり、オペレータが加速器全体の状況を把握できるようにさまざまな努力が重ねられてきた。そのなかで、多種多様の情報のうち重要なものを選択的にオペレータが表示できるようなオペレータインターフェースが必要となっている。現在、KEK-LinacではオペレータインターフェースはBasic言語、C言語等のさまざまな言語で構築されている。しかし、現在の環境では効率的なインターフェースの構築が、困難となってきている。そこでTk-Widget¹⁾と呼ばれる簡易なX-Windowのユーザインターフェース構築のためのシステムをテストしてみた。オペレータインターフェースを比較的容易に作成することができるので、テストシステムなどには有効であると思われる。その概要と他の方法との比較について以下に報告する。

2、Tk Widget

1)、Tkとは

Tkは、X-Windowのユーザインターフェース構築のためのtoolkitで、他のX11 toolkitと同様な目的を持っていてTclと呼ばれるインタプリタ言語がベースとなっている。Tkでは、Motif²⁾ (今後標準となるであろうX-Windowのtoolkit) とほぼ同様の機能をもつWidgetを備えている。Widgetとはユーザインターフェースを構築する際の基本的な部品であり、Tcl言語で記述することで複雑な部品を作ることもできる。そして1つの部品を多くのプログラムで再利用することも可能で、また既存のWidgetを組み合わせて新しいWidgetを作ることもできる。TKのWidgetは動的にバインドや色等の特性を変化させることが容易でユーザインターフェースが構築しやすい。

2)、Tcl

Tcl (Tool command language) は、簡単な構造化プログラム言語でプログラミングの間違いが起りにくく、しかもインタプリタ言語であるので変更が容易であるのが特徴である。(Tclは、変数、数式、リスト、文字やif、forのような制御構造、等を提供している。) TclはUnixのShellコマンドに似ているが、より文字列処理に向くように設計されている。TclのLibraryは、Cプログラムにリンクされていてその中にある処理機構はTclのコマンドをいくつかの方法で実行する。それらは、標準入力またはファイルから一行ずつ読み取る方法、TkをX-Window上でTkのButtonをやキーにバインドしてXのイベントが起こったときにTclコマンドが実行されるようにする方法、また次で述べる'Send'で送られて来たコマンドの実行も行う方法などである。これらの方法は平行動作が可能でリアルタイムなアプリケーションを簡単に構築できる。

3) Send

Tkの大きな利点は、プログラム間の通信の方法が優れていることである。現在、通常のX-Windowアプリケーションでは、'selection'や'cut buffer'のようにユーザの手が介在させなくてはアプリケーション間の情報が転送ができない。

そこでTkは、'Send'と呼ばれるTclコマンドで非力な通信の問題を解決している。'Send'を実行すると、Tkは対象となるプログラムを探し出し、そのプログラムにコマンドをおくる。対象のプログラムは、コマンドを実行すると送ったプログラムに実行の結果を返す。(ただし、これは同じディスプレイ上のTk-basedのプログラム間で可能となる。)このように'Send'は、remote-procedure-callのようなもので、'selection'より一般的で強力な通信の方法である。これはTclが、インタプリタ言語であるためにこの機構が比較的簡単かつ強力に実現されている。例えばTk-basedのMagnet Controllerを、ステータス表示と入力装置のプログラムを別々に構築することができる。入力装置が、現在動作しているマグネットの電流値の設定を変えた時に変更内容をステータス表示に送り、ステータス表示は設定後の値を返すということが可能となる。'Send'により、強力な方法で同時にプログラムを動かすことができ、その一つのプログラム内に全て機能押し込む必要性がなくなった。'Send'はTkプログラムの開発時のデバッグにも有用である。このため、多くのtoolを開発することが可能でそのtoolは独立に開発と保守ができて、また違った組み合わせの使い方ができるようになる。

4) 他のToolkitとの比較

他のtoolkit (Motif) と比較すると、Motifでユーザインターフェースを構築する場合は、Motif全体の理解が必要でプログラムもC言語で組まなくてはならない。Motifにはユーザインターフェースの詳細記述のためにUILという補助言語が提供されているが、コンパイラ言語なのでコンパイルする手間がかかりすぐに実行はできない。Tkでバインドする場合、簡単な手続きでできるがMotifは簡単ではない。Motifには'Send'にあたるコマンドを提供されていないので、このために一つのプログラムに全ての機能を登録しなくてはならないのでプログラムが大きくなり維持が困難になる。UILはTkよりいろいろな部品の細かい指定ができるが言語が複雑でifやforのような制御構造をもっていない。

	Motif	Tk
言語	C, UIL コンパイラ	Tcl インタプリタ
Bind	難しい	容易
Size	大きい	比較的小さい

TkとMotifの簡単な比較

	Source Lines		DS3100 bytes	
	Xt/Motif	Tk	Xt/Motif	Tk
Intrinsics	24900	15100	216400	92800
Tcl		9300		61100
Geometry Manager	2100	1000	17100	7400
Buttons	6300	1000	43700	8600
Scrollbar	3000	1200	24900	8000
Listbox	6400	1600	53100	10700
Total	42700	29200	355200	188600

TkとMotif/Xtのコンパイル後のバイト数とソースコードの比較

	Basic	Tk
行数	110行	40行
Size	約6kbytes	1.5kbytes

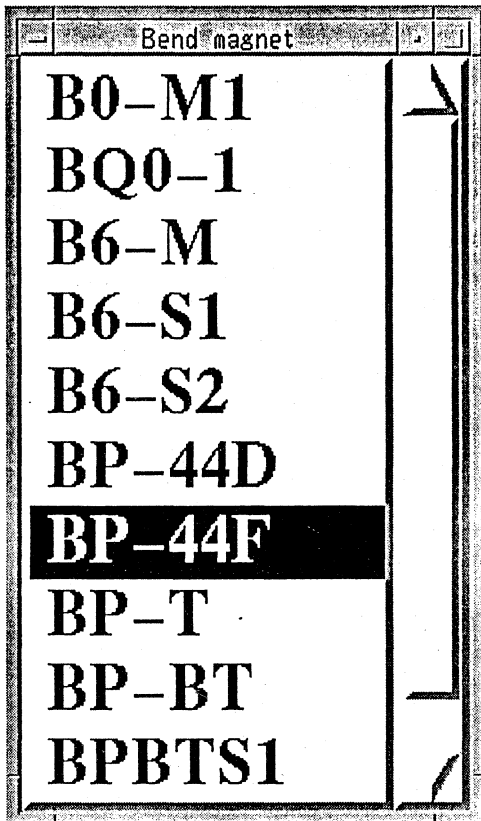
電卓プログラムのSize比較

Operation	Time
Simple Tcl command (set a 1)	41 μ S
Creat, display, 50 Button (first time)	359 mS
Creat, display, delete 50 Button	187 mS
Creat, display, delete 50 Label	169 mS
Creat, display, delete 50 Scale	189 m

簡単なコマンドの実行速度とWidgetが部品を作る速度 (DEC Station 5000/200)

3、プログラム例

ここに記述してある例はマグネット電源をONするためのメニューをTkで作成したものと、その実行のWindowである。このプログラムは2つのWidgetを組み合わせたものである。4行目、7行目はスクロールバー、リストのWidgetの構成で11行目はリスト表示の指定で12行目はキーバインディングの指定である。14行目はリストの中をダブルクリックしたときに、どのマグネットの電源をONするかを指定している。



```
#!/usr/local/bin/wish -f

set tfont "--times-bold-r--*-34*-*-*-*-*-*-*"

wm title . "Bend magnet "

scrollbar .scroll -command ".list yview" -bg White \
    -activeforeground Pink1 -relief raised -bd 4 -fg White \
    -width 30
listbox .list -yscroll ".scroll set" -relief raised -bd 4 \
    -geometry 10x10 -font $tfont -bg White \
    -selectbackground Black
pack append . .scroll (right fill) .list (left expand fill)

.list insert 0 B0-M1 BQ0-1 B6-M B6-S1 B6-S2 BP-44D BP-44F BP-T \
    BP-BT BPBTS1 BPBTS2

bind .list <Double-1> "magon"
bind . <Control-q> {destroy .}
focus .
proc magon {} {
    catch { mg sts [lindex [selection get] 0] > /dev/tty}
}
```

4、まとめ

Tkは簡単にX-Windowのユーザインターフェースを構築することができ、さらにTCP/IPなどのネットワークを使うことでさまざまな拡張が可能になる。TkはX-Windowが使用できる計算機であれば利用が可能で無償提供されている。KEK - Linacでは、ユーザインターフェースはパーソナルコンピュータ³⁾を利用したBasic言語のアプリケーションの蓄積が大きいので、さしあたってテストプログラムとして使用していくがプログラム開発環境としてはX-Windowが一般的になってきているので、将来オペレータのための多量の情報の前処理を例えばエキスパートシステムなどで行いその結果を表示するツールとして有効だと思われる。

参考文献

- 1) 、 Ousterhout, J "An X11 Toolkit Based on the Tcl Language " Proc. USENIX Winter Conference, 1991
- 2) 、 OSF / Motif Programmer's Guide Release 1.1 Prentice Hall, Englewood Cliffs, NJ, 1991
- 3) 、 M, Tanaka et.al., Proc. 16th Linear Accelerator Meeting in Japan (1991) 321