Operator Interface Using Tk Widget (II)

Kazuro Furukawa, Norihiko Kamikubota and Shirou Kusano*
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. It isimportant to make effort to reduce information and to build easily configurable user-interface for operators. We have tested a building tool for graphical-user-interface called Tk in our Kek Linac control system. Tk is very efficient to develop user-interface for accelerator operation. The possible usage of Tk in our control system is described as well asits feature and performance.

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（2）

1、はじめに

最近、加速器の大型化と精密化が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することが困難なってきている。そのためオペレータが、多種多様の情報を選択的に表示できるオペレータインターフェースが必要となってきた。これまで、KEK-Linac のオペレータインターフェースは Basic 言語, C 言語で構築されてきた。現在更に、効率的なオペレータインターフェースの構築が検討されている［1］。そこで、我々は昨年のリニアック研究会で X - Window 上のユーザインターフェースを容易に構築することができる Tk / Tcl と呼ばれるシステムの報告をした［2］。それをもとに、制御したい機器に対して現実的な名前と単位で制御できるようにTcl コマンドを改良したシステムの構築を試みた。

2、Tcl (Tool Command Language) と Tk

Tk［3］は、ワークステーション等で標準的に用いられているX - Window 上でのユーザインターフェース構築のためのシステムで、Tcl はそれのベースとなる言語とライブラリを指している。Tcl 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こりにくく、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。Unix のShell に似ているが、より文字処理に向くように設計されていて、全てのコマンド、コマンドの引き数、コマンドの結果、そして変数の値は全て文字列となっている。Tcl のライブラリには、Tcl 言語を解釈するためのの Parser 、組み込みコマンド用のルーチン、及びアプリケーション用にコマンドを追加して Tcl を拡張するためのプロシージャが含まれている。Tcl ライブラリは、コマンドを受取り、解析し組み込んだコマンドであれば直接実行し、そうでなければ間接的に Shell を通して実行する（図1）。Tcl を使うことにより、アプリケーションを自分で用意する必要がなくなり、Button やキーバインドに対応するコマンドをプログラムに記述することが、簡単に実現できる。Motif［4］（現在、X-Window 上の標準的インターフェース）で Tk / Tcl と同じようなことを実現するには、Motif やもしくは X-Window の全体の知識を必要とするが、Tk/Tcl のプログラムでは簡単に記述ができ、Motif のようにさまざまの手続きを必要としない、またX-window の知識がなくてもプログラムの記述は可能である。しかも、インタプリタ言語であるのでどこで Error が発生したかが容易に把握でき、その修正もすぐにできる。

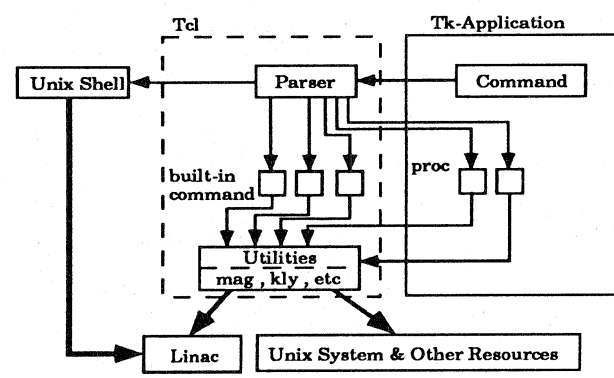

図 1

3、Tclコマンドを利用した加速器制御とその例

Tclコマンドで加速器を制御するにはいくつかの方法がある。Tclのライブラリの中にC言語等で作ったコマンドを組み込む方法（内部コマンド：Built-in Command）、外部からUnix Shellを通してコマンドを実行する方法(外部コマンド）、Tcl自体でライブラリを作成し記述する方法とがある（図1）。その中で今回は、ライブラリの中にC言語で作成した加速器制御のコマンドを組み込んだシステムを試みた。現在KEK - Linacでは、低レベルでの制御機器へのコマンドはバイナリで行われている。しかし、バイナリでのコマンドではアプリケーションプログラマにとっても非常にわかりづらいものであるし、ユーザインターフェースを構築するのも困難となってしまう。そこで組み込むコマンドは、ユーザインタフェースの構築が容易にできるように制御する機器に対して名前と単位、量を送るだけで制御できるユーティリティを作成した（例1）。今回作成したコマンドは、C言語で書いており"Tcl - Interp"型で定義されたライブラリ関数で解釈する（例2）。これらの関数では、Tclとライブラリとの間では文字列で受け渡し、制御用低レベル関数との間ではバイナリで受け渡す。今後は、Tclライブラリと制御用高レベル関数の間では文字列で受け渡しを行い、どのような物にもアクセスが可能となる。（図2）

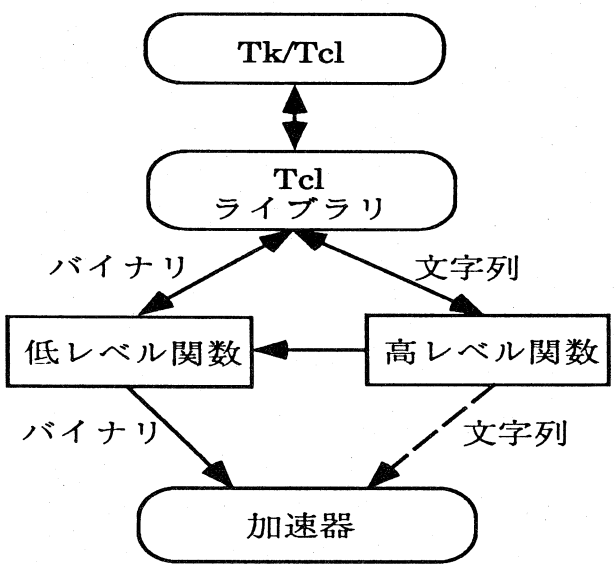

図2

（例1）コマンドの利用

vacconts <sector> <controller> <channel> st
kly <function> <klystrion name>

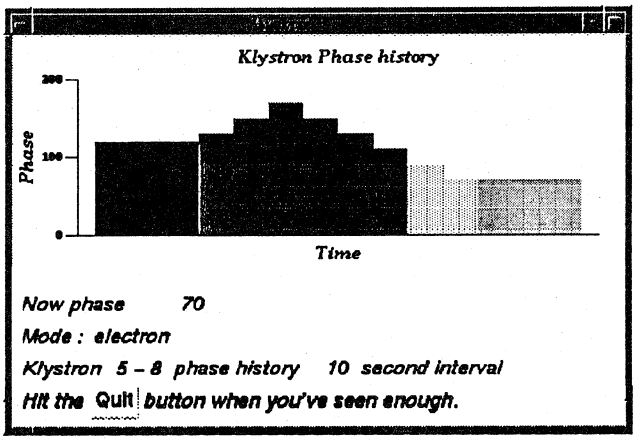

図3－1　クライストロン位相変動の履歴

各セクターのクライストロン1台の位相を30秒毎に棒グラフで表示する。過去約10分前までの位相を表示でき、更新する毎に左へシフトする。

上記（1）のコマンド使用したオペレータインタフェースの例（図3）

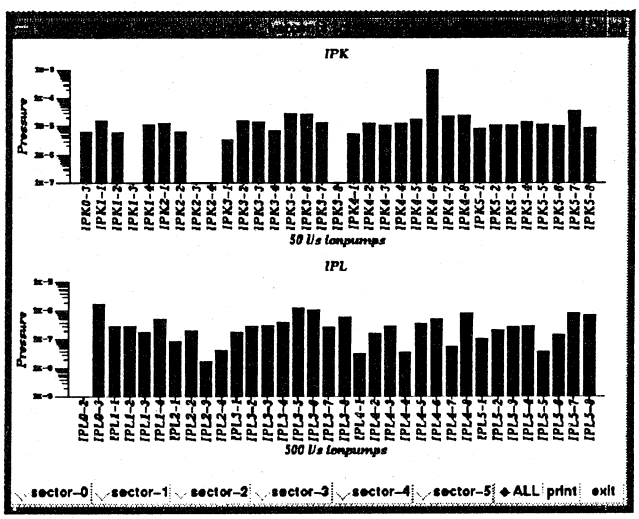

図3－2　現在のLinacの真空度表示

加速器全体及び各セクターの真空度の表示。データの更新は10分毎に行われる。過去のデータの参照は不可。＊データの校正値に多少誤差があるため表示の数値は正確ではありません。

（例２）プログラムの利用

```
int
Tcl_VaccontsCmd(dummy, interp, argc, argv)
ClientData dummy;
Tcl_Interp  *interp;
int argc;
char **argv; {
```

加速器を制御するコマンドは、Unix Sell を通しても実行可能だが多少速度的に劣る。外部コマンドと内部コマンドの処理速度の比較をしたところ例えば、単純な加速器制御コマンドを実行する場合内部コマンドで約１ msec 、外部コマンドで約１００ msec であった。インタプリタの特性を生かして開発時にはデバッグが容易な外部コマンドで実行し、最終的には内部にコマンドを組み込むのが適当である。

４、まとめ

Tk / Tcl は、X-Window のユーザインタフェースを容易に構築ができ、さらに制御機器を制御するために必要なコマンドを用意すればオペレータに解かりやすく、プログラマーにとってもユーザインターフェースの構築が容易になる。さらに現在は日本語化も進み、より見やすいユーザインターフェースの構築ができる。その他にも、インタフェースビルダー（XF）等のいろいろなサポートツールが揃っている。今年度KEK- Linac では、加速器制御のネットワークがTCP / IP ベース［５］に変わるのでTcl のさまざまな拡張も可能になる。今後は、これらの拡張機能を用いて加速器のシュミレーションプログラムやエキスパートシステム等に接続し、オペレータの援助となるユーザインターフェースの開発を行う予定である［６］。

図４　日本語化された Tk / Tcl

参考文献

[1] I.Abe, et al, this proceedings.

[2] S.Kusano, et al, Proc.17th Linear Accelerator Meeting in Japan, (1992)285

[3] J.Ousterhout, "An X11 Toolkit Based on the Tcl Language " Proc. USENIX Winter Conference,1991.

[4] OSF/ Motif Programmer's Guide Release1.1PrenticeHall,Englewood Cliffs, NJ, 1991.

[5] N.Kamikubota,et al, this proceedings.

[6] K.Furukawa, et al, this proceedings.