UPGRADE OF GUI FOR LINAC CONTROL

Tadahiro OONUMA and Yoshinobu SHIBASAKI

Laboratory of Nuclear Science, Tohoku University

ABSTRACT

We are now upgrading GUI (Graphical User Interface) of the control system at Tohoku Linac. This system uses Personal Computer (DECpc466D2LP-66MHz) and Visual Basic which makes coding GUI easy and simple. The first results of this system are presented.

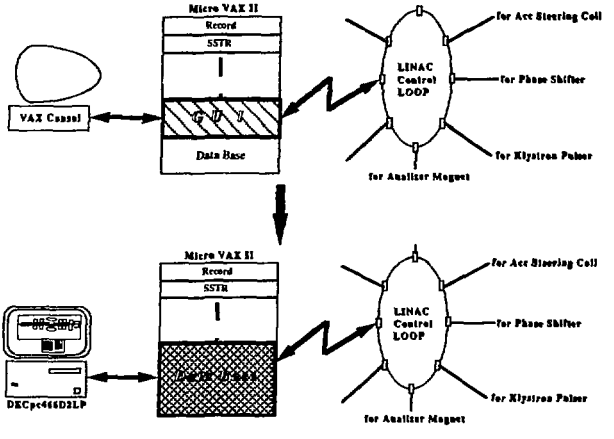
加速器制御用GUIの更新

1．はじめに

東北大学300MeVリニアックは、電子リニアックとパルスビームストレッチャーの制御の充実を図るため、制御系の改造を行なってきた。その間幾度となくマン・マシン・インターフェースの構築、改造を重ねてきた。これまでのマン・マシン・インターフェース部分の構築方法として、VAXを使ったVMS上でのC言語によるプログラミングが主であった。しかし近年、パーソナルコンピュータ（以下PCとする）の性能の向上、発展がめざましく、ワークステーションに代わる存在になってきた。またそれに伴いGUIの開発環境の向上により、以前の言語依存の開発環境からグラフィカルなプログラミング環境に変わりつつある。今回の加速器制御用GUIプログラムの開発は、従来のリニアなプログラミングスタイルに変わり、Windowsプログラミング・モデルによる開発をすすめているので、その概要と計画、テスト結果について以下に報告する。

2．更新の必要性　（第1図参照）

東北大学300MeVリニアックの制御用GUIプログラムは、これまでVAX-C言語を使いプログラミングしてきた。そのために様々な面で問題が生じてきた。1つはプログラム開発の困難さと作業性の悪さである。結局C言語で1行づつコーディングをし、またGUI用のツール等が我々のVAXには用意されていない状況のため、丸を描く、線を引くにしても1つ1つ座標設定をして、その作業の積み重ねで全体を描画するという方法をとってきた。またこのようなプログラミング方法では、後々に問題がでてくる可能性がある。それは、メンテナンスの困難さとプログラム拡張の難しさにある。1行1行コーディン



第1図　ハードウェア構成図

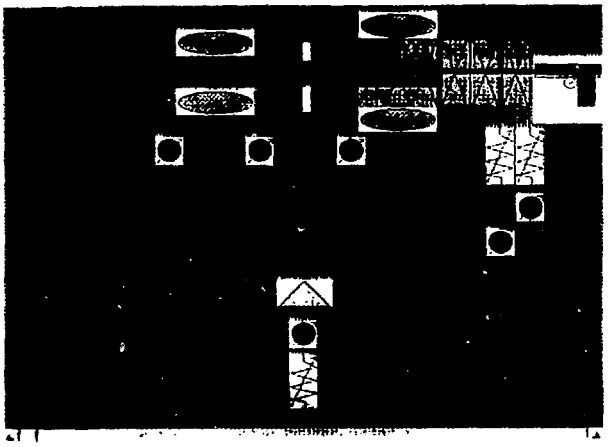
グされているリニアなプログラムは、作った者にしか理解できない面がある。またプログラムの変更、追加作業を行う場合に、現在使用している部分を壊さないように組み替える作業も困難を極める。そこで今回の新システムでは、このような問題点をクリアしながら、更に使い勝手の良いシステムにしていかなければならない。そこで我々はVAX上の現在の環境でのプログラミングはすでに限界であると判断し、今後はVAXにGUI部分を頼らず、最近性能も向上し、かつ安価に入手可能なPCに着目し、VAXの加速器制御用データベース・プログラムにPCのGUIプログラムを接続する方法を採用することにした。また作業を進めていく上でPCを使うことで有利な点が幾つかあることが解かり、それが現在の問題点を解決する手段であるということも解かった。その1つにGUI構築ツールが整っていることが言える。それによって複雑なプログラミング作業を簡素化し、後のメンテナンスを容易にすることができる。

3．新システムの基本理念

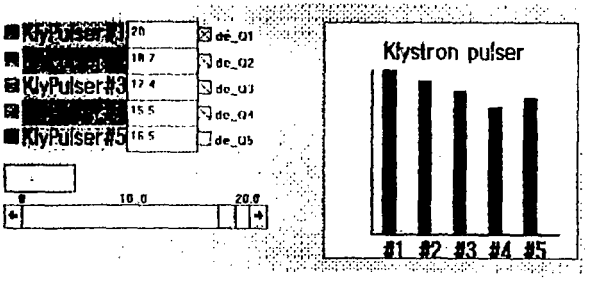
新しいGUIプログラムを構築するにあたり我々が確認した事項は、1つは直感的に、かつ視覚的にユーザーが判断できる制御用画面を構築することである。今までの制御用画面は、我々のVAX環境からいっていたしかたないのではあるが、マン・マシン・インターフェースという部分では多少貧弱であった。それというのもVAXにGUI用ツールが用意されていなかったため、極め細かに作画することができず、結局文字でのみしかユーザーに伝える手段がなかった。今回PCを使うことで機動性もよく、GUI構築用ソフトウェアもいろいろ用意されているので直感的、視覚的な制御画面を容易に構築することができる。2つ目にGUI画面をユーザーが各自、自由にカスタマイズできるよう作るということである。これはWindowsが持っている機能にDDE(Dynamic Data Exchange)およびOLE(Object Linking and Embedding)という複数のアプリケーションを結び付け、自由に切り貼りできる機能を利用することである。この機能を利用して各ユーザーが、自分なりの使い勝手の良い制御画面に変更、更新することができる。これらの機能を利用する考え方の応用で、表計算ソフトウェアを使い各機器のデータを管理することも考えた。このやり方として表計算ソフトウェアとGUIプログラムをあらかじめリンクし、機器設定値に変化があった場合に、自動的にそのデータを表計算ソフトウェアに入力する。また全ての機器を一括設定する場合等、プリセット時にはこの表計算ソフトウェアのテーブルを利用し機器を設定できるようプログラミングする。このように今回我々が構築するGUIプログラムの基本として、市販のソフトウェアを組み込み、リンクすることによってプログラムを簡素化し、標準化することと、作成者以外の者がメンテナンスを行なっても、解かりやすいプログラムを構築するようということを目指し開発に取り組んだ。

4．問題点とその対策　（第2図、第3図、リスト1、リスト2参照）

従来のリニアなプログラミングスタイルを捨て、グラフィカルなプログラミングを構築するために、我々が選択したGUI開発用ソフトウェアに今回はVisual Basicを採用した。このソフトウェアは本格的なWindowsアプリケーションを、短時間で簡単に作ることを可能にする新しい開発環境である。またこのソフトウェアはGUI環境とBasic言語を組み合わせたものであり、標準的なリニアなプログラミングスタイルではなく、イベント駆動型のWindowsプログラミング・モデルを採用している点で、我々はこのソフトウェアを選択し、開発にとりかかった。開発作業の中で種々の問題点があったので、その対策方法と合わせて紹介する。1つは細部の描画が困難であったことである。これはVisual Basicのみの単一のソフトウェアでは表現できない細かな描画を作りたい場合である。これには他のソフトウェアを最大限利用することで解決した。例えばVisual Basicのオプションを使っても表現できない絵にはBorland C++のWorkshop機能を使いIcon、Bitmap絵をリンクして表現した。またBasic言語という欠点から速度の問題もある。表現する絵の数が多くなればなるほどプログラムのロード時間を費やすことになる。これは今後同じVisual系GUI開発用ソフトウェアであるVisual C++に移行する計画によってこの問題点はクリアしていきたい。



第２図　Visual Basicで作成したフォーム１（加速管A部周辺構成表示）



第３図　Visual Basicで作成したフォーム２（KlystronPulser制御用表示）

```
Begin CommandButton Set
   Caption         =   "Set"
   FontBold        =   -1   'True
   FontItalic      =   0    'False
   FontName        =   "System"
   FontSize        =   16.2
   FontStrikethru  =   0    'False
   FontUnderline   =   0    'False
   Height          =   1080
   Left            =   2100
   TabIndex        =   13
   Top             =   1800
   Width           =   1092
End
```

リスト１　オブジェクト・プロパティ・リスト

```
' Click event procedure for number keys (0-9).
' Appends new number to the number in the display.
'
Sub Number_Click (Index As Integer)
    If LastInput <> "NUMS" Then
        ReadOut = "."
        DecimalFlag = False
    End If
    If DecimalFlag Then
        ReadOut = ReadOut + Number(Index).Caption
```

リスト２　Visual Basicプログラムリスト

５．まとめ

今回の報告は、開発途中のテストプログラムについてのみの報告にとどまった。開発作業に携わって感じたことは、グラフィック画面を作る効率、プログラミングする効率などの作業効率が大変すぐれている点であり、それが作業時間の短縮につながり、また後々のメンテナンス、拡張、変更の容易さにつながると感じた。過去の経験で、我々のVAXにより表現する絵を作るのに１箇月費やして作られた画面が、今回のテストでは、同じ画面を作る作業に、PC＋Visual Basicを使うことで１週間程度でほぼ完成した。このことからいってもWindows＋Visual系GUI開発ソフトウェアによるマン・マシン・インターフェース・プログラムの構築はおおいに有効であるということが言える。またこれらの構築環境は、加速器制御用のGUIプログラムにも応用できると考える。