

CONTROL SYSTEM ARCHITECTURE OF KEK 2.5GEV LINAC

Kazuro FURUKAWA, Norihiko KAMIKUBOTA, Isamu ABE and Kazuo NAKAHARA

KEK, National Laboratory for High Energy Physics

1-1 Oho, Tsukuba, Ibaraki, 305

ABSTRACT

The control system of KEK 2.5GeV linac is being rejuvenated. The main part of the hardware will be replaced and the architecture of its software is upgraded as well. System software has been developed to achieve new control scheme as well as old one. Since we invested in operator's console system which uses old control messages, we will continue support it. The new control scheme is based on the object oriented design of accelerator controls. It will enable the flexible and robust linac controls.

KEK 2.5GeV Linac のコントロールシステムの構成について

1. はじめに

現在 2.5GeV 電子線形加速器では制御系の更新作業を行っており、ハードウェアの主要部分の置き換えを行うと共に、ソフトウェアの構成も変更する。これによって、アプリケーションプログラムからはこれまで同様のアクセスが可能であるのと同時に、オブジェクト指向のデザインに基づく、より使いやすいサービスの提供を受けることが可能になる。そして、加速器の各コンポーネントに対して正確で理解しやすい制御を実現することができるよう努めている。ここではそのような考えによって実現する制御系全体の構成について考えてみたい。

2. オブジェクト指向

新しい制御体系では、サーバ計算機としての Unix Workstation とハードウェアに対するフロントエンドとしての VME system を TCP/IP プロトコルのネットワークで接続し、またオペレータコンソールとしては NetWare プロトコルで結ばれた Personal Computer が利用される。そこで目指しているものは拡張性があり、柔軟に新しい問題に対処しうる制御系である。限られた人員、予算と将来の拡張のために国際標準と業界標準をできるだけ採用したシステムとなっている [1, 2, 3]。

この制御系に対するアクセスの手順はできるだ

け対象指向(オブジェクト指向)になるように注意しており、作成したアプリケーションソフトウェアの生産性が上がるようにしている。

また、加速器を我々が扱うときに、制御系が一般的なインターフェースとなるよう、機能を用意している。これによって、加速器オペレーションと共に加速器の技術的な側面や科学的な側面についても、同じインターフェースで情報のやり取りが可能になる。

オブジェクト指向デザイン 内部ソフトウェアは別にして、アプリケーションプログラムとのインターフェースに、できるだけオブジェクト指向を持たせるべきことは疑う余地のないことであると思われる。このことは言葉は違っていても繰り返し言われてきた [4]。しかし、これまでは様々な制限から、一般的には、個々のアプリケーションプログラムが直接ハードウェアに依存した情報にアクセスし、また、それらの特性データベースもアプリケーション内部に持っていた。このためハードウェアの種類が増えたりやデータベースが変更になった際のアプリケーションソフトウェアの維持に労力が必要で、また、既存のプログラムと同様のプログラムを開発する際にも最初からプログラムを書き直すというようなことが起こり、生産性を高めるためには別の努力が必要であった。

現在、ある加速器装置をそのコントローラのハードウェア情報としてではなく、その装置自体の

オブジェクトとして、アプリケーションソフトウェアから扱えるような環境を整えており、個々の装置の特性を知らなくても装置にアクセスすることができる。また、グループ化した装置や、ビーム特性、さらには履歴、フィードバックシステムというようなサブシステムもオブジェクトとしてとらえて利用できるようにしていく予定である。

これらによって、品質が高く維持や開発が容易なソフトウェアを作ることが可能になる。

オブジェクト指向プログラミング オブジェクト指向のソフトウェアの構築を直接支援するプログラミング環境が最近利用可能になってきており、それらを使えば、継承や多相性、動的バインディングといった機能も利用して、堅実でより再利用性の高いソフトウェアを開発することができる。

しかし、残念ながら、現在のリアルタイムシステムまで含めた互換性を保証できるオブジェクト指向プログラミングの環境は存在しない。また、我々のシステムの場合分散環境が重要であるが、ネットワークを経由した処理についてはそれらの環境は未熟である。現在はこのようなものが利用できるようになるのを待っている段階である。

オブジェクトの実現 上に述べた理由でソフトウェア技術からのオブジェクト指向に対する支援をいまのところ受けることができないので、アプリケーションプログラムと制御システムのインターフェースの定義だけはオブジェクト指向になったものの、その機構自体は既存の技術の利用となっている。

オブジェクトはできるだけ取扱いが理解しやすいように構成し、アクティブなものについては、有限状態機械としての定義をはっきりさせて利用できるようにしている(図1)。

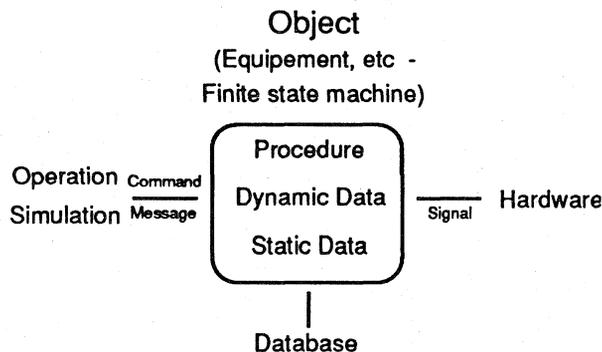


図1 オブジェクトの構成

例えば、クライストロンの操作には高周波の

出力を開始するコマンドがあるが、実際には複数の情報と複数の操作で達成されるものである。これらの手続きと特性情報をまとめてオブジェクトとして扱っているということである。

実際のインターフェースとしては、C言語のファンクション、ネットワーク経由のメッセージ、通常の Shell Command が利用でき、これらについては、計算機の種類にはほとんど依存しない利用が可能である [1, 3]。現在さらに Tcl というインタプリタ言語も利用できるよう準備が進められており、すでにいくつかのアプリケーションプログラムも Tcl で動作している [5]。また、既存の技術の延長として各計算機上から共有メモリや共有ディスクファイルの上にある情報も利用可能になる。

オブジェクトとして加速器の制御系をデザインすることは、制御系を通して見ることでできる加速器を仮想的に再定義することにも相当する。その加速器ではその技術的な側面や科学的な側面、そして運転に必要な側面についても見せてくれるようなものであってほしい。

技術的側面 加速器を維持、開発のために技術的に扱う場合にも制御系を利用することができる。例えば、装置に対するなんらかの特性の設定値と実現値の間の較正は重要なことであるが、それらの測定の援助や、特性値の履歴、平均値、標準偏差などの統計情報も役立つものと思われる。

また、オブジェクトの標準的なインターフェースでは隠されている情報や細かい操作を、今後装置担当者に部分的に見せるようにする必要があると思われる。少なくともオブジェクト内で手を加えていないハードウェアの情報は現在も提供することが可能になっている。

科学的側面 加速器を研究対象としてみた場合にも、例えば、我々がマシンスタディと呼んでいる加速器実験の際に情報収集の媒体として充分利用できると思われる。また、その結果がオペレーションに役立つものであれば、制御系の中に組み込んで常時利用できるようにしたい。ビームモニタ等の整備と協力して準備を進めれば、科学的情報がまだまだ眠っていて、興味ある研究対象を掘り起こせるのではないかと考えられる。

オペレーション そして、最も重要であるのが運転の支援である。加速器に対する効率の良いユーザインターフェースをオペレータに提供できる

かどうかが問題になる。また、ほとんどの担当者が問題にするのもこの点である。これらについても様々な努力が払われている [5, 6]。将来は各オブジェクトにユーザインターフェースの定義を含めるような設計にしたいが、それらを可能にするためのハードウェアとソフトウェアの確立はまだ図られていない。現在は、ユーザインターフェースを行うプログラムから各オブジェクトに対してメッセージを送ることにより制御が行われる。

3. 制御用アプリケーションソフトウェア

上に述べたような制御のサービスのもとで様々なアプリケーションプログラムの実現が可能になる。特に体系的に準備する必要があると考えているものを以下に挙げてみる。

監視、アラームシステム これまでの加速器の運転の経験から、加速器のパラメータの自動監視が、安定な運転のためには有効であることがわかってきている。現在、電磁石やクライストロンについては単純な情報の監視を行っているが、今後監視項目を増やして行く必要があると思われる。

これを推し進めるとプロダクションシステムを用いたエキスパートシステムの構築が必要になる。リアルタイムシステムの上にも OPS5 ベースのプロダクションシステムの移植を試みた。しかし、現在のところ、これを用いたエキスパートシステムの必要性は感じていない。われわれの場合まず簡単な条件の監視が重要で、それらの情報の解析を行ったうえで、知識ベースの構築が必要になると思われる。それらの結果によっては、将来はオブジェクトの各階層中に埋め込まれた知識ベースが必要になると思われる。

監視の結果のアラーム情報の出力については、効率良くオペレータが認識できるシステムが必要で、現在検討中である。

履歴情報 加速器の異常の検出は監視システムによって行われるが、エキスパートシステムが構築されていない場合、異常の原因の追及には加速器のパラメータの履歴情報が有効である。

現在、加速器の一部のパラメータについて1分おきに情報を蓄積しており、必要なときに履歴の読みだしが可能である。今後、この情報をどのように利用していくか考えていかなくてはならない。履歴情報を単に視覚化することのほかに、例えば、

標準偏差等の統計情報を定期的に求め、それを監視の項目に加えられるようにすることなどがある。

関連情報の取得 ひとつまたは複数のパラメータを変更しながら他のパラメータの変化の様子を観測することは通常の運転でもしばしば行われている。この情報の自動取得のシステムが SLC の運転においては大変有効であったことが示されている [7]。

われわれの加速器においても、今後加速器の増強やモニタの整備によって、読みだし可能なパラメータが増えるので、このようなシステムが適度な応答速度で動作すればオペレータにとって役立つものになると思われる。具体的には、設定パラメータ、読みだしパラメータ、設定値のステップと範囲、ステップ時間等を選んで、設定値に対して読みだし値をプロットできるようなシステムを作りたい。

視覚化 いくら良い情報が集められても、現在でも情報が多すぎるぐらいなので、オペレータがわかりやすいように情報が提供できなければ、利用されることはない。そこで、情報の視覚化が必要である。また、簡単に情報を選択できる機構も必要となる。

テストとして、X-Window 上で簡単な表示ツールを履歴情報のために作成してみた (図2)。このようなものであれば、容易に提供することができるので、オペレータや担当者とのような可視化の方法がいいのか話し合っていきたい。

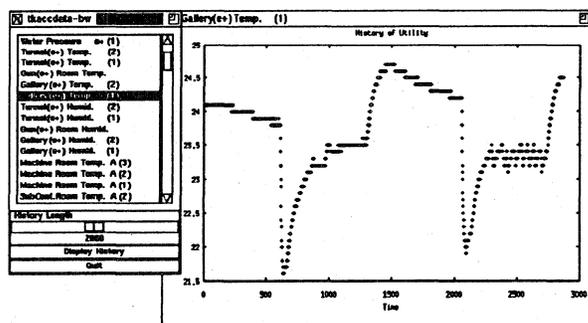


図2 情報の視覚化の例、運転停止中のクライストロンギャラリの温度履歴情報、左が情報の選択パネル、右が変動グラフ。

下流の加速器との連携 入射器は下流の加速器ひいては利用者にビームが提供できて始めて有効になる。これまでは計算機の仕様の違いなどによって、主に電話で情報交換がなされてきた。しかし、入射器と下流の加速器の制御系との間で様々な情

報交換が行われるのは自然なことと考えられる。

昨年、入射器とトリスタン加速器の新しい計算機との間が、TCP/IPにより接続され、一部の入射器の制御パラメータがトリスタン加速器の制御系より設定読み出し可能になった。現在のところプロテクションを厳しくしていることと、トリスタン側の計算機が制御系に完全には組み入れられていないために利用は多くはないが、今後は情報交換の需要は高まるものと思われる。

シミュレーションのオンライン化 これまでは制御系内の計算機資源の問題で、シミュレーションやデザインのためのコードはほとんど大型計算機で実行されてきた。しかし、最近の Workstation の進歩とシミュレーションコードの移植の努力により、制御系内で実行することも可能になってきた。また、問題を選べば、オンラインでシミュレーション結果と加速器の情報を比較しながら運転することも可能になって来ている。

もちろん、パラメータの較正值の検討やビームモニタの整備などが不可欠であるが、さしあたって何が現実的に利用可能であるか検討する必要があると思われる。次の例は制御パラメータの設定とほぼ同様にしてシミュレーションコードの実行を可能にしたもので、Prebuncher, Buncher 部のビーム軌道の計算を行ったものである [9]。

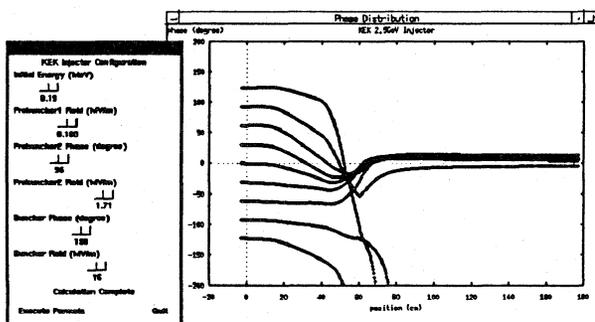


図3 入射部の軌道計算の例、左が制御パラメータの設定と同様に準備したパラメータパネル、右が計算結果

現実には入射部はオペレータの支援ができるほどにシミュレーションの精度を上げることは困難であるが、通常のビーム輸送については可能であると考えている。これは、次のフィードバックの中に組み入れる場合も含めて重要である。

フィードバック 現在のシステムではソフトウェアによるフィードバックは行われていないが、今後 B-Factory がスタートした場合には重要になる

と思われる [8]。現在のシステムで達成しうる応答速度について検討してみた結果、Unix 経由—100ms-1sec, VME 対 VME—10ms-100ms, VME 内—10msec-1ms, DSP 利用—1ms-10 μ s, という程度である。今後ビーム軌道計算のコードと共に検討していかなくてはならない課題である。

所内特に居室への情報提供 制御とは特に関係ないように思われるが、情報の有効利用という点で加速器の維持管理のためには居室への情報提供も必要であると考えている。現在、クライストロンと電磁石の監視情報などを所内計算機ネットワーク上に AppleTalk 経由で提供しているが、需要があればさらに拡大していきたい。

4. まとめ

上に述べたようなシステムの中核は今年10月より利用を開始する。もちろん、3. で述べた全てのサービスがその時点で提供されるわけではない、またそれらが本当に有効であるかどうかは、これから見極めていく必要がある。また、それらを有効に利用するためには、様々な情報の計算機への接続や実用的な単位系への較正情報の精度を上げることなども必要である。

参考文献

- [1] K.Furukawa et al., Proc. of International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'91), Tsukuba, 1992, p.89.
- [2] K.Furukawa et al., Proc. of 15th Linear Accelerator Meeting in Japan, Sapporo, 1990, p.168.
- [3] N.Kamikubota et al., this proceedings.
- [4] B.Kuiper, Nucl. Instr. and Meth. A293 (1990) 308.
- [5] K.Furukawa et al., this proceedings.
- [6] I.Abe et al., this proceedings.
- [7] L.Hendrickson et al., Proc. of International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'91), Tsukuba, 1992, p.493.
- [8] T.Himel et al., ibid., p.414.
- [9] J.Y. Choi, R.Matsuda, private communications.