

THE INTRODUCTION OF THE OBJECT ORIENTED PROGRAMING TECHNIQUE INTO CONTROL PROGRAM AT TOHOKU LINAC

Masakatsu MUTOH

Laboratory of Nuclear Science, Tohoku University
1-2-1 Mikamine, Taihaku-ku, Sendai-shi, 982, Japan

ABSTRACT

The computers and their related technologies have been making rapid progress, which makes influences on an accelerator control field widely. The trend seems to expand not only in hardware technologies, but also in software technologies, and an object oriented programing technique is one of them. In order to improve productivity, flexibility and easy maintenance, an introduction of the object oriented programing technique into the control program at Tohoku Linac is considered. This paper is described about our discussions.

東北大理ニアックの制御プログラムへのオブジェクト指向プログラミング技術の 導入について

1.はじめに

最近のコンピュータ及びその関連技術の急速な進歩は、加速器制御の分野にも大きな影響を与えている。それらのいくつかを列挙すると、

- (1)大きなデータベースの構築が容易になり、個々の機器に依存した専用のプログラムから、データベースを参照しながら制御するプログラムへと簡素化が可能となった、
- (2)ネットワークの利用により、大きな加速器に於いての分散処理システムの構築が容易になった、
- (3)ワークステーションを制御卓に用いることにより、これまで主に使われてきたタッチパネルやノブに比べ、マルチウインドやマウスの採用により、マンマシーン・インターフェースが一層充実した、
- (4)CAMACに対してVMEのようなインテリジェント化のできるインターフェースが広く普及した、
などである。

一方、制御プログラム技術に於いても、加速器がしだいに大型かつ複雑になるにつれて、新しいプログラミング手法や言語が積極的に取り入れられてきた。例えば制御プログラムを機能別の独立したモジュールに分け、それらを組み合わせてシステムを構成しようとする構造化プログラミングがある。プログラムのモジュール化あるいは部品化は、プログラムの生産性、保守性を高めるために有効であることから、構造化プログラミングは広く普及した。しかし、プログラムが長く複雑になると構造化プログラミングの特徴でもあるモジュールの独立性を確保することが困難になり、開発効率は低下してくる。そこで最近このモジュール内のデータとそのデータの操作を一体化し、そのことによってモジュールの独立性を確保し、プログラムの開発や拡張を容易にしようとしたオブジェクト指向プログラミング（以下OOPと略す）がいろいろな分野で利用されるようになってきた。

OOPとは、例えば加速器を構成している各機器を階層構造的に管理し、それぞれの機器をオブジェクトとしてとらえ、機器のデータとその機器を制御するためのアルゴリズムを一体化して問題を解決しようとする考え方である。

我々はこうした技術は、今後ますます複雑化する加速器の制御に於いて有効であると考え、東北大理ニアッ

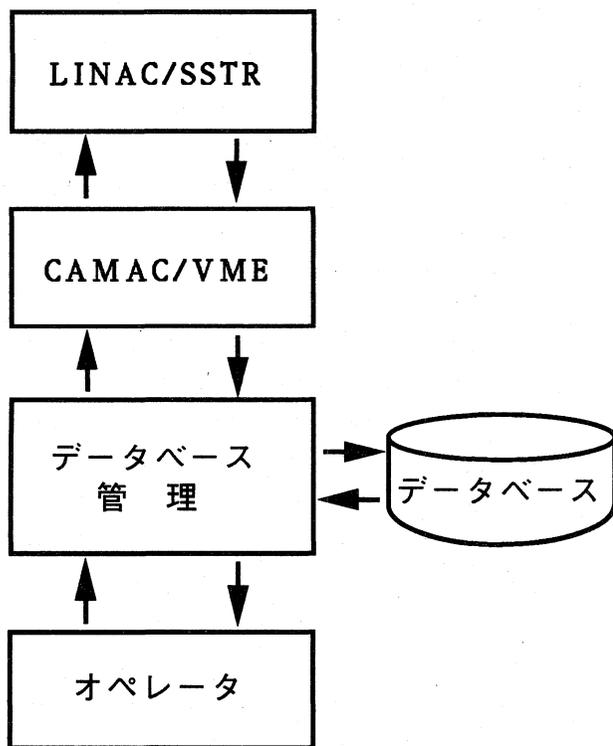
クの制御プログラムへの導入について考えてみた。

2.現在の制御プログラム[1]

核理研のリニアックとパルスビーム・ストレッチャ(SSTR)の制御にはマイクロVAXが使われ、そのプログラムの主なブロックダイアグラムを第1図に示す。加速器を構成し制御の対象となっている機器のデータを集約したデータベースがこのシステムを中心に置かれ、制御に必要なプロセスはすべてこのデータベースとコミュニケーションしながら加速器を制御し、データベースを経由せずに直接機器を操作することはできないようになっている。例えば、オペレータが、あるマグネットの磁場を変えようとする場合、コンソール上の操作内容がデータベースに通知され、データベースではそのマグネットの電源をCAMACまたはVMEを使ってコントロールするために必要な操作手順を組立て、CAMAC/VMEドライバに送っている。各機器の値や運転状態に関するデータは、CAMAC/VMEドライバで読みとられ、一定時間毎または必要に応じてデータベースへ通知される。それを受け取ったデータベースでは、変化があった機器のデータを更新すると同時に、データをコンソール上に表示させるプロセスに伝送する。このようなデータベースを制御の中心に据えたことにより、加速器の機器構成や制御手順に変更があっても、このデータベースの内容を専用のツールを使って修正するだけで済み、制御プログラムそのものに変更を加えることはない。このデータベースと制御プロセスはコミュニケーション・プロトコルに従っており、データベース内のデータは他のプロセスから直接アクセスすることはできないようになっている。

この制御プログラムは、1987年に制御用コンピュータをそれまで使ってきた国産のミニコンからマイクロVAXへ置き換えた際に、全面的に作り直された。このとき拡張性、保守のしやすさなどを考慮して、データベースの中で、データと制御手順を一体化して管理する今の方式を採用したが、それは結果的に無意識のうちにオブジェクト指向の枠組みに沿った考え方をしていたことになる。もちろんここでは、データと制御手順を

一体化していると言ってもデータベース管理プログラムの中でのことで、オブジェクト、クラスの階層化やクラス間の継承関係といったものではなく、単に構造体を活用しただけのものである。



第1図 制御プログラムのブロック図

3.OOPへの展開

加速器を構成し、制御対象となる機器をオブジェクトととらえようとする、まず加速器はどんな機器で構成されているか、それらの機器の動作はどうなっているか、また機器相互の関係はどうなっているか調べる必要がある。その中から何をオブジェクトにするかを決めるためにオブジェクト・モデルを作る。第2図にモデリングの例を示す(実際このようになるかは検討を要するが)。こうしたモデルをもとに次のような手順で作業を進める。

(1)何をオブジェクトにするか。

つまり制御の対象となるものは何かを決める。

(2)オブジェクトの表現をどうするか。

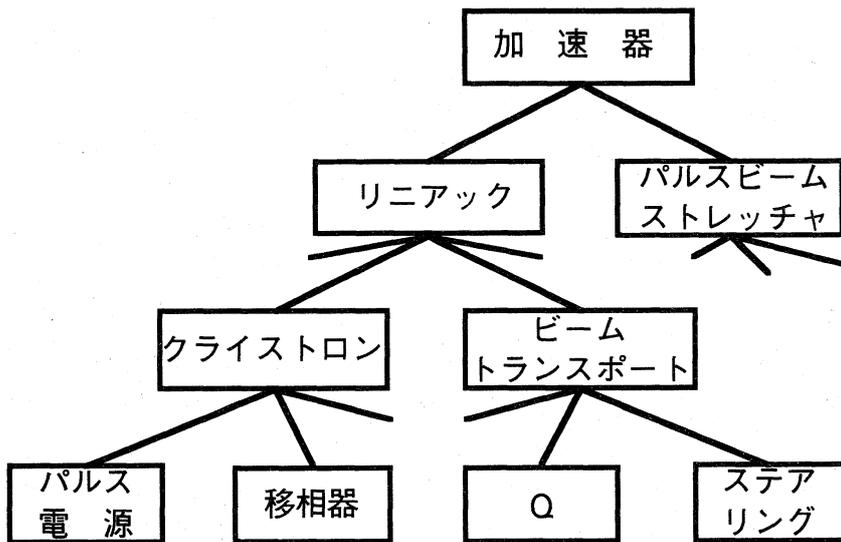
オブジェクトの性質、振る舞いを調べて、それらを制御するための方法をどうするかを決める。

(3)各オブジェクト間の関連をどうするか。

共通情報を持つオブジェクトをクラスという。このクラスの持っている機能によりクラスの階層化をおこない効率化をはかる。そして、上位クラスから

下位クラスへの性質の継承をどうするかなどを決める。

以上のような作業を何度も繰り返しながら修正を加え、最適なオブジェクトを決定することになる。こうした作業がOOP化のための重要なところで、システムができあがった後で、機器の変更や制御方法が変わっても該当するオブジェクトの最小限の変更だけで済むようにしないと、かえって効率の悪いシステムになってしまう。幸い東北大リニアックでは、先に述べたように制御対象となる全ての機器はデータベース化されており、これをもとに上記(1),(2),(3)の作業は順調に進められると考えている。また、制御卓のマン・マシン・インターフェースのOOP化についての作業はすでにおこなわれており[2]、制御システム全体のOOP化を積極的に押し進めていきたいと考えている。現在OOPのためのプログラミング言語としては、C++を使うことで検討している。



第2図 オブジェクト・モデリング

4.最後に

我々はOOP化を進めるための準備の段階である。生産性の高いシステムを作り上げるには、加速器制御に関する実用的なクラスライブラリをどれだけ揃えることができるかが大きな要因である。そのためには同じ加速器をもっている大学、研究所の関係者が、共通に使えるライブラリの開発を共同で進めるべきだと考える。今回、種々議論して頂いた高エネルギー研究所の阿部勇氏、加速器の制御に使っているマイクロVAXにOOP用のC言語を準備した核理研の柴崎義信氏に感謝いたします。

参考文献

- [1]M.Mutoh et al., "Improvement of the control system for linac and pulse beam stretcher at Tohoku University," Proceedings of the 7th Symposium on Accelerator Science and Technorogy, Osaka, December 1989,pp.240-242.
- [2]I.Abe et al., "The PF LINAC Console system by object oriented programing," This conference.