

マシンモデルを利用した加速器制御系の構築

DEVELOPMENT OF ACCELERATOR DEVICE CONTROL SYSTEM USING THE MACHINE MODEL

川瀬雅人^{#, A)}, 高橋博樹^{B)}

Masato Kawase^{#, A)}, Hiroki Takahashi^{B)}

^{A)} Japan Atomic Energy Agency (JAEA), Tokai, Ibaraki, JAPAN / J-PARC Center

^{B)} Japan Atomic Energy Agency (JAEA), Rokkasho, Aomori, JAPAN

Abstract

The accelerators construct different type of devices. And because constructing control processes unique for each device increases the number of types of the processes, such control systems lack maintainability, scalability and flexibility. Therefore it is necessary to develop the accelerator device control system with a common architecture. To overcome these problems, the common control system using the model called “Machine Model” has been developed for the needs of constructing a unified control system. It is realized by extracting statuses common to each device, careful modeling and defining the abstract class. The device status is assigned to one of the six kinds of abstract statuses. Those statuses are all defined inside device controllers. It also controls procedure-type processes for device status transitions. It allows us to unify control systems of different kinds of devices. In addition, because this method hides the procedural control of each device from upper application levels, it is possible to reduce the correction time of the application when it’s modified or upgraded.

This report is about development of the accelerator device control system using the machine model.

1. はじめに

J-PARC 加速器線形加速器施設 (LINAC) および 3GeV シンクロトロン加速器施設 (RCS) では、多種多様な加速器構成機器で構成されている。各々の加速器施設は、運転、建設、調整や改造が平行に進められており、これら、全ての工程に対応できるような柔軟性が加速器構成機器制御系開発には求められている。加速器構成機器には上位制御系との連携及び機器自身の制御を行うために機器制御コントローラが実装されている。将来を見据えると、機器制御コントローラの換装も想定される為、本コントローラ換装後の制御系変更が容易に行えることなども含め、メンテナンス性や拡張性を有した制御システムが必要となる。

2. 従来の制御系開発方法

従来の加速器構成機器制御系の開発開始は、各機器の製作が終了した後に開始された。そして、機器動作等を把握するための仕様確認、遠隔制御すべきパラメータの抽出等 (制御系仕様の検討) が行われた。それに基づき各々の加速器構成機器に特化した制御系の設計、コーディング、試験が行われる。この工程を制御系構築が完了するまで続けることになる。LINAC 及び RCS は、

- 電磁石系電源
- ビームポジションモニタ、ロスモニタ
- 真空機器
- 高周波システム

などの加速器構成機器で構成されており、従来の制御系開発方法においては、これら 1 つ 1 つの機器に対して、上記の工程を繰り返し行い、各々の遠隔制御系を完成させることになる。このように、従来の方法では各加速器構成機器もしくは機器の種類毎に制御系の開発を行わなければならない、開発作業に投入する人員や時間は膨大になる。このため、J-PARC 初期ビーム運転開始時期においては、人員および開発時間増となることが懸念されていた。

3. 従来の開発方法における問題点

上述の通り、従来の制御系開発方法においては、機器制御系開発の着手時期が後回しになってしまう。加速器構成機器制御系開発工程中において、何らかの問題 (トラブル) により工程が遅れた場合、その分、加速器ビーム運転スケジュール等に影響を与えることとなる。J-PARC 加速器施設建設中には多くの機器製作を行っていたため、当然、それらの制御系開発も同時に行うこととなっていた。制御系開発を進める人員は決して多くはなく、1人が複数の機器制御系を担当することも少なくなかった。また、制御開発作業には効率性が求められ、全体のスケジュールに合わせられるよう (影響を与えないよう) 綿密な作業計画を立て、開発作業を遂行しなければならなかった。さらには、開発工程中に何かしらの問題が生じた場合でも、迅速な対応が求められた。ただし、従来の開発方法においては、機器制御開発を行う上での統一的な方針がなく、機器動作が異なる等の理由から、独自の制御プロセスが構築されることとなった。その多様さは加速器構成機器の台数

[#] masato.kawase@j-parc.jp

分や種類の数量に相当するため、機器制御系のメンテナンス性欠如という新たな問題も生み出してしまふこととなった。

そして、機器制御系のメンテナンスや機能拡張においては、各機器それぞれに対応した作業を行うことが要求される。異なる構成機器において共通化できる作業が少ないため、作業効率は悪く、作業量は膨大なものとなる。作業時間は必要以上に必要となり、制御系の改良等を含めたメンテナンス性は悪いものとなることが想定されていた。

4. マシンモデルの導入

従来の開発方法における問題点（統一性、メンテナンス性、拡張性、柔軟性及び効率性の欠如）を考慮し、“マシンモデル^[1]”と呼ぶ機器状態定義方法を加速器構成機器制御系開発に取り入れることとした。この方法は、各加速器構成機器が持つ固有の状態に対してマシンモデルを定義することにより、機器状態の共通化を計る。このマシンモデルを利用し、加速器構成機器の種類毎にあった制御プロセスを統一化させ、従来の開発法での問題点として挙げた、メンテナンス性、拡張性、柔軟性及び効率性の欠如を排除することが可能となる。

5. マシンモデルを使用した制御開発方法

マシンモデルは、加速器構成機器が持つ様々な状態や動作を基にした、機器の状態を抽象化された共通の状態として定義された抽象化クラスに相当する。定義決めを行う際には、各機器に共通する情報を抽出し、それらの情報を基にして適切なモデリングを行う。このマシンモデルを導入した結果、加速器構成機器制御における基本的な制御プロセスは、機器の現在の状態監視プロセス、機器の状態遷移プロセス、設定値操作プロセス等の約3通りとシンプルなものとなる。これらのプロセスは、上位制御系の特殊性のあるソフトウェアで管理するものではなく、機器がどの抽象化された状態であるのかの判断、機器側の状態遷移処理などは、全て機器側制御系にのみ組み込むこととする。この様にして、各機器固有の手続き型制御を上位制御系アプリケーションではなく機器側制御系でのみ処理インターフェースとすることにより、機器改造や拡張時における上位制御系アプリケーション修正作業の大幅な低減を可能とする。

5.1 マシンモデル定義

マシンモデルの抽象化された共通の状態は6種（DOWN, STOP, STANDBY, RUN, FAULT, EMERGENCY）に集約する。

これら6種の状態はTable 1のように定義する。“DOWN”は、機器への通電が無い状態のため、抽象化状態の中で一番安全な状態である。“STOP”は、機器のブレーカー（主電源等）がONされている状態である。“STANDBY”は、余熱処理等が終了し負荷への出力がいつでも行える状態である。“RUN”は負

荷への出力を開始している状態である。“FAULT”は、比較的軽度な異常を検知した場合であり、遠隔リセットを可としている。“EMERGENCY”は重度の異常を検知し、現場での原因追求を行うまでリセットできない状態としている。そのため、遠隔リセットを不可とする。マシンモデルでの機器状態遷移サイクルをFigure 1に示す。

Table 1: The Machine Model Definition

状態	定義
DOWN	ブレーカー等 OFF
STOP	ブレーカー等 ON
STANDBY	出力許可状態
RUN	出力中
FAULT	軽故障検知
EMERGENCY	重故障検知

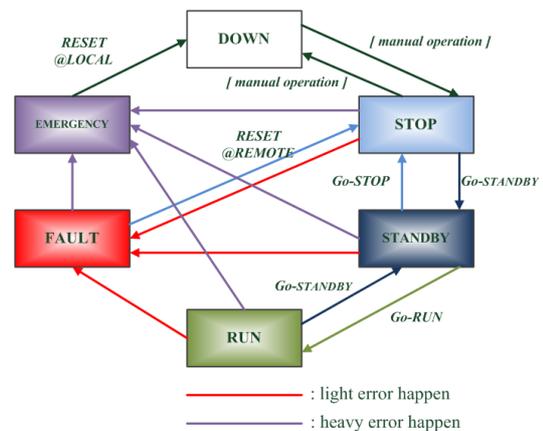


Figure 1: The machine model status transition cycle.

マシンモデルは、機器状態を抽象化した共通の状態として定義された抽象化クラスだが、本クラスは、“機器の現在の状態”、“機器の遷移先の状態”、“機器の入出力信号”の3つの属性で構成する。機器状態は、先に述べた6種の状態に集約するが、機器担当者は、担当機器の詳細な情報を確認する必要があるため、これらの情報は、すべて入出力信号に集約する。基本的な入出力信号の種類は、インターロック、ステータス、オペレーション、モニター、パラメータ設定の6種であり、アナログ値や異常事象の検知等が、本属性で確認できるようにしている。

5.2 マシンモデルを使用した機器状態遷移

マシンモデルにおける機器状態遷移処理は、従来制御系開発方法においては機器毎に必要であったが、本モデルを組み込むことで、複数必要であった手続きを状態遷移コマンドと呼ぶイベント型コマンドのみで処理することが可能となる。本コマンドにより、現場操作のような細かな手順を踏むことなく機器を起動させることができ、決められたプロセスに沿って機器状態を遷移させることが可能となる。本コマンドは、Figure 1に示した状態遷移サイクルに特化したコマンドであり、“Go-RUN”、“Go-STANDBY”、“Go-STOP”、“RESET”の4種から構成される。この

コマンドを構築したことにより、機器依存型の手続きを排除し、すべての機器に対応した共通性を上位制御系に持たせることが可能となる。状態遷移コマンドとコマンドによる状態遷移先を Table 2 に示す。

Table 2: The Machine Status Transition Command

状態遷移コマンド	遷移先
Go-STOP	[DOWN => STOP]
	or
	[STANDBY => STOP] [RUN => STANDBY]
Go-STANDBY	or
	[STOP => STANDBY]
	[STANDBY => RUN]
Go-RUN	[STANDBY => RUN]
RESET	[FAULT => STOP]

5.3 マシンモデルと上位ソフトウェアとの連携

上位制御系は、EPICS (Experimental Physics and Industrial Control System) をベースに構築している。加速器構成機器の遠隔制御する際には、EPICS 経由で行われるが、加速器構成機器の機器制御コントローラに特化した EPICS デバイスドライバ (以下、EPICS ドライバ) が遠隔からの制御をサポートしている。J-PARC LINAC 及び RCS においては、機器制御コントローラとして横河電機製 PLC FA-M3 を採用している加速器構成機器が多いことから、本項では、この PLC に対応した EPICS ドライバとマシンモデルの連携について説明する。

EPICS ドライバの通信仕様は、データレジスタにのみアクセス可能な仕様としている。その他のアドレスには、ドライバレベルの制限によりアクセス不可としている。そのため、EPICS 経由での機器制御を実現するためには、機器制御に必要な情報は全てデータレジスタへ集約することが必須である。データレジスタのアドレス群は、定められたフォーマットで構成される。基本フォーマットは、“構成情報領域”、“制御領域”、“パラメータ領域”、“モニター領域”、“インターロック領域”となっている。各領域は、加速器構成機器毎に必要なアドレス数が変わることから可変型になっている。そのオフセット (必要アドレス数) は、“構成情報領域”に記載する。

上記のフォーマットは EPICS ドライバ仕様に基づいて構成されたフォーマットであり、このフォーマットに合致している機器であれば、本 EPICS ドライバを使用して共通性及び整合性を有した上位制御系からの機器制御を行うことができる。共通化されたデータレジスタ構造をもった PLC であれば、同じ手法での遠隔制御が可能となる。また、通信仕様が共通化されているので、情報授受系の異常があった場合、担当者による異常の原因道程が容易にでき、メンテナンス性も向上する。

加速器構成機器状態遷移処理、パラメータ設定処理は、すべて PLC ラダーシーケンスプログラム内部で行われる。状態遷移イベントやパラメータ設定イベントは、EPICS 経由で PLC に送信され、コマンド

(情報) を受信後、そのコマンドに適した処理を PLC 内部で開始する。そのため、処理条件が変更 (追加または削除) になった場合においては、PLC 内部の変更作業のみが必要であり、上位制御系アプリケーションを含めた外部のソフトウェアの修正作業は不要である。状態遷移やパラメータ設定に必要な詳細な情報は、すべて PLC 内部に隠蔽されているため、上位制御系アプリケーションはその変更等を意識する必要はない。将来的にデータレジスタへの集約情報が増減した場合においても、構成情報領域の修正のみで上位制御系との情報授受は確立される。

6. 従来の方法とマシンモデルとの対比

従来の方法においては、機器製作後、仕様確認 (システム設計) を行い、その後、コーディング、単体試験、総合試験という流れで制御系開発作業が進められていた。また、機器操作手順を含めた機器側の変更があった場合には、それに適した上位制御系アプリケーションの変更も必要であり、改良作業のための膨大な時間が必要となった。

マシンモデルを導入した現在の方法においても、開発工程の流れは基本的には変わらない。一方、マシンモデルを導入する場合は、すべての機器制御の整合性、統一性を持たせるためのモデリング作業が重要となることから、この作業に十分な時間をかける必要がある。機器特有の情報は、すべて PLC 内部に隠蔽されることから、機器側での機能 (状態) が追加された場合、モデリングの再検討後、その修正対応等は PLC ラダープログラム内部で行うことになる。そのため、上位制御系アプリケーションの修正等が必要となる場合においても、最小限に抑えることが可能となる。

7. まとめ

加速器構成機器制御系開発にマシンモデルを導入した。各機器から共通する状態を抽出し、抽象的な共通の状態を構築するとともに、機器状態遷移処理を行う際のコマンドを統一した。各機器固有の動作は機器制御 PLC ラダーシーケンスプログラムで行うことにより、上位制御系アプリケーションからの状態遷移操作は、状態遷移コマンドのみに限定することができ、操作体系の統一化を実現した。また、PLC との通信は、データレジスタのみに限定し、その仕様によってデータレジスタを構築することにより、共通化した EPICS ドライバを使用した制御が可能となった。以上から、マシンモデル導入により、メンテナンス性、柔軟性および統一性を有した機器制御系を構築でき、この制御系開発方法は、将来的な機器改良にも柔軟に対応できる制御系開発モデルを構築できることを明らかにした。

参考文献

- [1] H.Yoshikawa, et al., “Control of SPring-8 Injector Linac”, Proceedings of the 20th Linear Accelerator Meeting in Japan, Osaka, September 6-8, 1995.