

## New Control System for the KEK-Linac

N. Kamikubota, K. Furukawa, H. Akimoto\*, K. Nakahara and I. Abe

National Laboratory for High Energy Physics (KEK)

\* Mitsubishi Electric System and Services Engineering Co. Ltd.

### Abstract

New control system for the KEK-Linac has been developed. Unix-based workstations and VME-bus computers are introduced. They are inter-connected with an Ethernet, which is used as a high-speed data-exchange network. New system will start the operation after October 1993.

## KEK-Linac の新制御システム

### 1 はじめに

#### 1.1 新制御系の目的

KEK-Linac の旧制御システムは、minicomputer (Mitsubishi MELCOM 70/30)8 台と約 300 の  $\mu$ -processor (CAMAC で minicomputer に接続) からなる分散処理系であり、minicomputer 間は専用の光ファイバーネットワーク (Loop-1) で相互接続されている (陽電子部は minicomputer の代わりに SBC) [1, 2]。1978 年以来このシステムで運転を続けてきたが、最近では

- 計算機能力が新しい要求に対応できない
- minicomputer の保守が打ち切られる

などの問題が生じている。そこで、minicomputers の担当部分 (制御メッセージの分配) を新しい制御系で replace することが検討されてきた [3]。

ここで考えている新しい制御システムの目的は、

- 制御機器を適度に速いレスポンス (10ms 程度) で制御/監視出来る手段を与える
- エラーが発生した場合、その原因がわかる

ことである。KEK-Linac では、運転操作のほとんどをパソコンネットワーク (Fujitsu FM-R etc., NetWare) と BASIC 言語を利用した表示システムで行っている [4] が、ここで言う制御システムとは [4] の様な上位システムを構築するための「道具」を提供する。

#### 1.2 新制御系設計の方針

新制御系で置き換えられる部分は CAMAC より上層 (CAMAC 自身も含む) であり、CAMAC より下層の置き換えは今回は考えていない。

新制御系を設計するに当たり以下の方針をとった。

ハード・ソフトとも業界標準を採用 加速器およびその制御系は、通常 10 年以上継続して使用される。このため特定の計算機に依存する設計では、その親会社の方針の変化の影響を受け好ましくない。

将来の拡張に対応可能なこと KEK-Linac は運転用と同時に研究用の加速器でもあり、日々どこかで改良が行われている。従って制御系も、機器の拡張・変更に対して即応できることが望まれる。

十分高速な通信ネットワークを持つこと KEK-Linac は物理的に広い範囲 (500m) に機器が分散しているので、この距離スケールで十分高速な通信ネットワークが必要である。

低価格・小マンパワー KEK-Linac の予算規模・制御グループの実働人数からの制限である。この点からも、業界標準の採用は望ましい。

これらの点を考慮して設計した新制御システムは、Unix-workstation とフロントエンドの VME 計算機を Ethernet (TCP/IP protocol) で接続した形態になっている (図 1)。新旧の対応を表 1 に示す。

### 2 ハードウェア

現在、KEK-Linac 制御ネットワーク (Ethernet with TCP/IP protocol) に接続されている計算機群が図 1 に示されている。

**workstations** 新制御系で使用されている Unix workstation は、現在 peach (DECstation 5000, 運転用), lime (disk-less DECstation 5000, 画面表示用), maple (DECstation 5000, プログラム開発用), などがある。特に peach は、disk-server (NFS で大容量ハードディスクを他計算機に解放)、制御メッセージ distributor、など、新制御システム上で重要な役目を担っている。

**front-end computer (VME)** 各セクタに1台ずつ、VME-bus 計算機 (Force CPU-40) が設置されている。実際の制御機器は、何らかの形でこれら VME 計算機に接続される。

**touch panel drivers** 制御室のタッチパネルドライバとして、PC9801DS (MS-DOS v3.3) が直接制御ネットワークに接続されている [5]。

**klystron monitor display** 全クライストロンの状態 (インターロック、スタンバイ、等) をモニターするソフトが pfumx で動作しており、パソコン pfpc (PC9801RX) で表示させている。pfumx と pfpc 間は、常時制御ネットワーク経由で情報を交換している。

**gateways** 別の計算機システムとの Gateway として、複数の計算機が導入されている。

- pfumx は、旧制御システムとの Gateway である
- maple は、KEK 所内ネットワークとの Gateway でもある
- peach は、TRISTAN control との Gateway でもある
- peach は、NetWare 表示システムとの Gateway でもある

表 1: 新旧制御システムの対応表

component	旧 system	新 system
console (server)	MEL70/30 RealtimeOS	workstation Unix
network (response)	Loop-1 (5Mbps)	Ethernet (10Mbps)
front-end	MEL70/30 [SBC for e+] CAMAC	Force CPU40 68040,25MHz VME-bus

- pfua60 は、放射光大型計算機 (Fujitsu FACOM) との Gateway である

この様に、多種多様な計算機機種が入り乱れているのが KEK-Linac 制御系の著しい特長になっている。この様な状態は管理上好ましいものではないが、歴史的には旧制御システムの能力不足を補うために次々に導入されてきたものであり不可避であった。

なお、新制御システムは制御用に整備された Unix-workstation ならどの機種からも利用できる。また、VAX/VMS やパソコン (DOS) でも、TCP/IP socket を装備していればほとんどの機種で制御システム installation が可能である。今後も年々接続される計算機が増加して行くと考えられる。

### 3 ソフトウェア

#### 3.1 ソフトウェア構造

##### 3.1.1 software model

新制御システムのソフトウェアは、図 2 に示す簡略化された model に基づいて設計された。

手順 1 operator は、制御しようとする特定の object (名前 (name) で区別される) に対し、コマンド (com) を送る。必要ならばその com に付随した設定値 (value) も併せて送られるが、この場合は object の property の value が変更される。

手順 2 a 送られた com が指定した object に正常に作用したならば、return-code として 0 が operator に返される。com によっては、object の property の value も帰って来る。

手順 2 b エラーが起こればそのエラーに対応した負値の return-code が返される。この場合、operator は return-code を調べて何のエラーが発生したか知ることが出来る。

結局 operator は、制御システムに対してデータ組 (com, name, value) を与え、その回答 (return-code, value) を得る、という手順を繰り返して実際の加速器を制御する。

全ての制御をこの model に従って設計すれば、非常に単純化された通信ソフトウェア採用が可能になり、保守の点で有利である。一方この model では、複数の property・object の同時一括変更が 1 メッセージでは出来ない (たとえ Hardware で support されていても) などの制限も生まれる。

### 3.1.2 table と log (database)

新制御システムでは、情報データベースとして parameter table、運転記録データベースとして log file を利用している (図3)。

object の名前 (name) が実際にはどの VME のどのモジュール、何チャンネルであるかといった対応付けは、parameter table で管理されている。parameter table は通常の text file で、editor によって修正・追加している。制御システムは、operator の制御対象 (name) に関するハードウェア情報をこの table から得る。

制御システムが実際の機器制御に伴って出力する情報 (制御記録、エラー、など) は、log に残される。情報記録のレベル (エラー関連のみ情報を残すか、全てのアクセス記録を取るか、等) は制御機器に応じて設定を変えている。

parameter table, log file とも、NFS を利用して制御システム全体で同じファイルを参照している。これらのデータベースは自力開発のごく簡素なものである。将来は、機能や保守性の観点から市販の製品 (SYBASE, ORACLE など) を導入したい。

### 3.1.3 制御メッセージの分配

operator から制御システムに向けて送られる制御メッセージ (com, name, value の組) は、実際には2段階の process で処理される (図3)。operator からの制御メッセージは、まず第1の process (message distributor) に送られる。第1の process は parameter table を参照して制御メッセージを目的の VME の第2の process に転送するための郵便局であり、メッセージの転送以外何もしない。第2の process はそれぞれの VME で実際に制御を行う daemon process である。第1 process と同じ parameter table を参照して、制御に必要なハードウェア情報を得ている。daemon process は VME 毎に独立に存在するが、NFS ディスクから同じ実行イメージをロードしている。図3では片方向の矢印しか示していないが、daemon process が用意した reply メッセージ (return-code, value の組) は同じ経路で operator まで帰される。また、第1・第2 process が出力した制御情報 (エラー情報など) は、同じ log に書き込まれる。

この様に2段階に分割したのは、小数の複雑な process の組合せよりも多数の単純な process で制御システムを組んだ方が保守上有利と判断したためである。実際、message distributor は全ての制御要素で同じものが使用されている (parameter table と log だけが異なる)。制御メッセージが message dis-

tributor を経由するための overhead は 1ms 程度であり、応答速度の点で問題にはならない。

### 3.1.4 メッセージ分配上の諸問題

**socket の利用** KEK-Linac では前述したように多種多様な計算機が既に導入されていたため、process 間通信には異機種間でも通信可能な TCP/IP socket 関数を採用した [6, 7]。また、socket 関連の parameter を与えれば daemon の通信部分の code を半自動的に生成するテンプレートが unix 用に開発され、利用されている [8]。

**service name 利用による非干渉性** KEK-Linac では、制御の対照となる機器が多種類ある (次節参照) が、これまでに述べた message distributor, daemon, parameter table, log はそれぞれの機器毎に独立に存在しており、異なる機器間ではお互いに干渉しない。この仕掛は、socket 通信で機器毎に異なる service name を割り当てることで実現している。

**複数の接続要求の処理** message distributor のレベルでは、同時に複数の制御 process から接続要求が到来することが有り得る。socket は基本的に 1:1 の通信なので、socket 関数だけではこの問題の解決は出来ない。現在は、message distributor が制御メッセージ毎に socket 接続を open/close することで複数の process を見かけ上同時に処理している。operator と VME の daemon process 間でのメッセージ交換時間は概ね 40ms (往復) であるが、このうち open/close 処理に 26ms かかっている。ところで、unix の一般的な通信 daemon (telnet, ftp, etc.) では、接続要求がある度に子 process を作り、子 daemon に接続要求を出した process の相手をさせている。親 process はそれぞれ自分専用の子 daemon を確保できるため、open/close 処理による時間の遅れの問題は無い。現在 message distributor をこの形態に変更することを検討している。

**security** message distributor は、socket の接続 (open) を行う際に、接続要求を出した host 名を登録 table の hostname と照合している。登録 table に無い host からの接続要求は拒否され、その host 名は log に記録される。また、host によって特定のコマンド ("GET" など read-only コマンド) のみ受け付けるような設定が出来る。

## 3.2 制御可能な機器

新制御システムでは、制御の対象となる機器を2層に分類して考えている。

第1層(下層)は、VME module と対応する層である(表2)。これらの要素は一部を除いてごく一般的なもの、KEK-Linac の制御以外にも広く利用可能と考えられる。

第1層の制御要素に対するコマンド(com)は、使用している module に特有の hardware 仕様には出来る限り依存しないように設計されている。これは、将来保守上の理由などから module の機種変更が必要になっても、これらの制御要素を利用する上位プログラムに影響が及ばない様にするためである。

第2層(上層)は、第1層の制御要素を利用して制御する機器である(表3)。第2層の制御要素は第1層の機器に比べてより上位の概念で、KEK-Linac の加速器の構成要素に対応している。

機器操作上、第1層と第2層は全く同じである(同じ model に基づく)。

表 2: 制御可能な機器 (第1層)

制御要素	module	備考
D-out(16bit)	PVME501/03	
D-in(16bit)	PVME501/01	開発中
ADC(12bit)	MPV906	+/- 10v
DAC(12bit)	MPV904	+/- 10v
GPIB	DVME-GPIB	
loop2	開発中	
loop3	kaidu8755	

表 3: 制御可能な機器 (第2層)

制御要素	第1層要素	備考
coaxial-cable	D-out	
profile mon.	D-out	
klystron	loop2	module 待ち
magnet	loop3	
vacuum	loop3	
trigger	loop3	
slit	loop3	開発中

### 3.3 利用の形態

operator が新制御システムを利用するには、2つの形態がある。

コマンド利用 新制御システムが install された計算機では、機器を制御する OS レベルのコマンドが利

用できる。コマンドは表2、表3の機器それぞれに用意されている。簡単なテスト、速度が要求されない application ではコマンド利用が便利である。

C関数利用 コマンドと対応して、C関数が表2、表3の機器それぞれに用意されている。実際に加速器運転で使用する上位 application は、なんらかの形でC関数を call している。

さらに、表2、表3の機器それぞれで documentation の準備を進めており、特に peach などの workstation 上では online-manual の準備を初めている(一部機器は既に利用できる)。一度概念を理解してしまえば、workstation 画面でこちらを参照した方がお手軽である。

## 4 まとめ

KEK-Linac では、Unix-workstation と VME 計算機を Ethernet で接続した形の新制御システムの構築が進んでいる。1993年10月の運転から運用を開始する。

## 参考文献

- [1] K. Nakahara *et al*, Nucl. Instr. Meth., A251(1986)327
- [2] K. Nakahara *et al*, Nucl. Instr. Meth., A247(1986)153
- [3] K. Furukawa *et al*, Proc. Int'l Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'91), Tsukuba Japan, November 1991, p.89
- [4] K. Nakahara *et al*, Nucl. Instr. Meth., A293(1990)446
- [5] 上窪田紀彦、他：第17回ライナック研究会会議録、仙台、1992年9月、p.276
- [6] N. Kamikubota *et al*, Proc. Int'l Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'91), Tsukuba Japan, November 1991, p.318
- [7] 上窪田紀彦、他：第16回ライナック研究会会議録、1991年9月、p.324
- [8] 上窪田紀彦：「ネットワークサーバ用ソフトウェアモジュールの開発」、KEK 内部資料 PFINJ-MC-32、平成4年3月

KEK-Linac  
New Control System with VME front-end

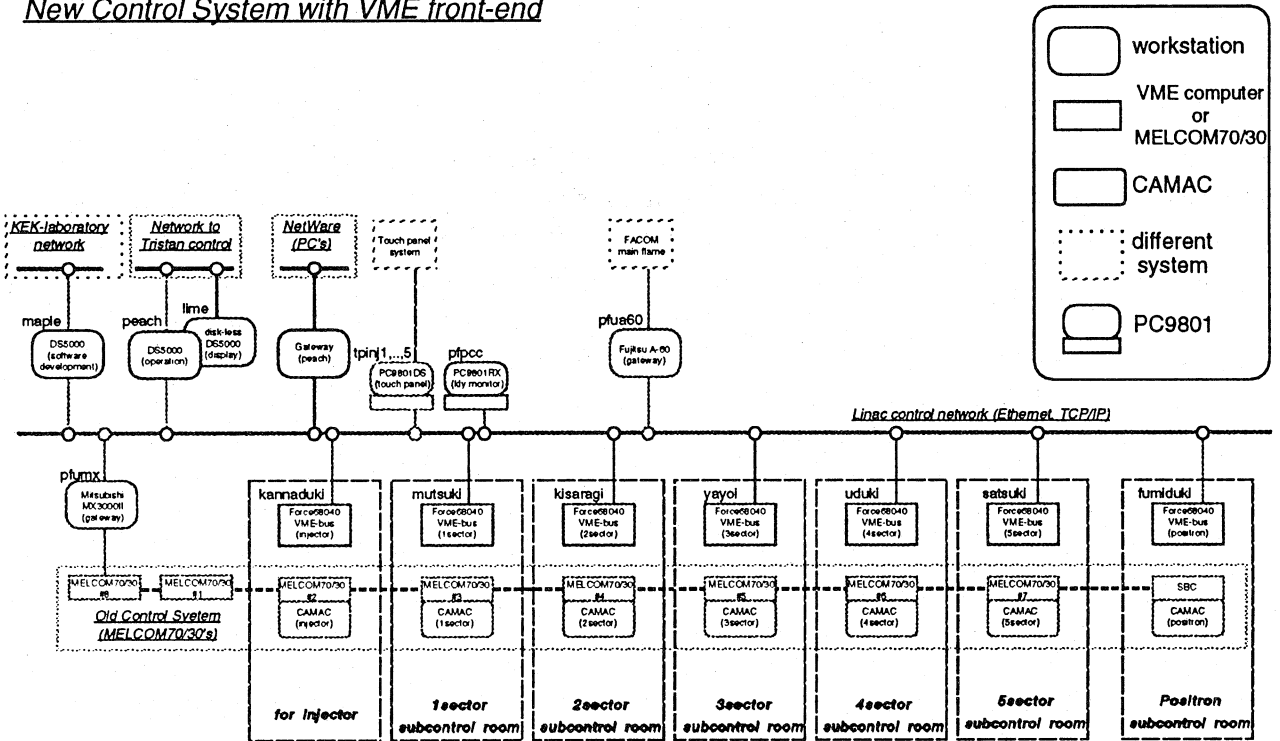


Fig.1 New control system for the KEK-Linac

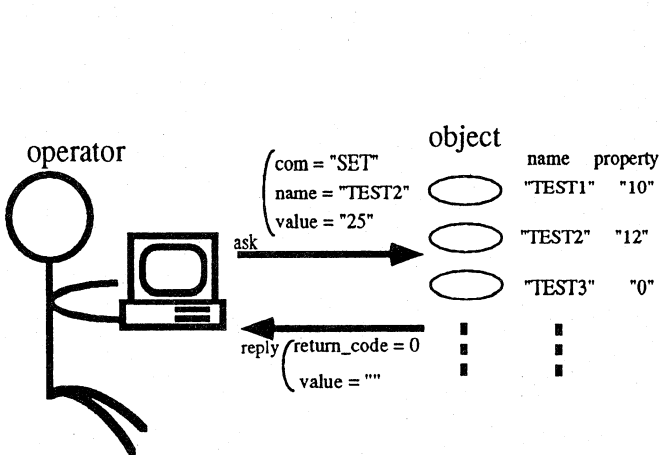


Fig.2 Software model for a control system

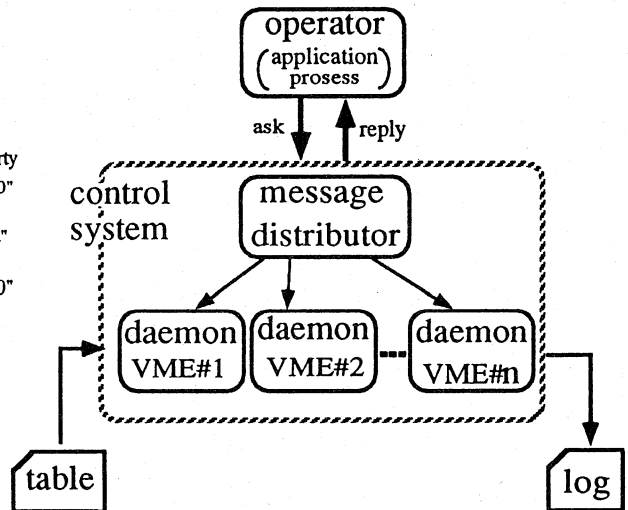


Fig.3 distribution of control messages