

[F18p17]

## Paradigm Shift on Accelerator Control System by Active X

Isamu Abe, and Masakatsu Mutoh\*

High Energy Accelerator Research Organization (KEK)

1-1 Oho, Tsukuba, Ibaraki 305-0801, Japan

\*Laboratory of Nuclear Science, Tohoku University

### Abstract

The KEK 8GeV Linac have been running PC system at both the operators' console and the device front end using Windows NT clients and server system. The PC system was developed by VB and its OCX, and a trial of upgrade from OCX to ActiveX has been made on the Windows NT system recently. In this paper, Component ware Oriented Accelerator Control Kernel (COACK) based on ActiveX is proposed and discussed.

### ActiveX による加速器制御パラダイムシフト

#### 「概要」

加速器制御では、小型加速器ほどUNIX等のワークステーションからスケールダウンしたPCシステムの導入が進んでいる。同時に、ハード及びOSの著しい進化のみならず、加速器制御ソフトについても静かにパラダイムシフトが起こっており、その新しい展開を見る事が出来る。ヨーロッパの研究所でとり組まれているACOP<sup>[1]</sup>に加え、ここでは、「COACK」を提案し、その概要を述べると共に、COACKのフェーズ2におけるコンポーネントウェア・ベースの加速器制御について述べる。

#### 1. 従来の加速器制御方式

加速器制御は歴史的に見て、手続き型の言語によって構造化手法等で構築された時代から、言語を離れツールによって加速器を制御しようと言う時代に移行して来た。ツール以前の加速器制御ソフトの開発は、大規模システム構築や生産コストの点で問題があった。従ってツール指向は当然の方向であった。加速器制御ツールソフトが市場に回り、又は、共同開発によって専用ツールの構築が行なわれる様になった。小型計測器を計算機で制御するような小規模のもの (LabVIEW, VEE, etc.) から、大型加速器の制御ソフト・ツール (EPICS, Vista, etc.) まで様々なものが存在した。しかしながら、昨今、更に新しいスタイルの制御システムが議論される土壌が出来て来た。ツールは、その提供される機能・運用の中においては便利で強力であるが、他のツールとの合い乗り入れの点では難点がありほぼ不可能で、持ち合わせている機能以外の要求に対しては、迅速な対応は困難且つ結構な手間を要する事が一般的である。この難点を解決すべき方法が、本レポートで述べる新しいスタイルの制御システムである。

#### 2. 次世代加速器制御方式

従来の幾つかの加速器制御ツールは、それぞれ異なった仕組みと機能を持つ。ツール間に共通点、共有出来るものは殆どない。新しい形の制御スタイルでは、図1の横軸における各ツールは、縦軸にある汎用化コンポーネントを共有し、ツールと言う固定した形態をとらずに、必要なオブジェクト (コンポーネントウェア) を組み合わせる事で柔軟なツールを構築し、汎用化コンポーネントを組み合わせる事で出来あがる制御ソフトである。現実の加速器制御を分析すれば、オブジェクトレベルで、他の加速器制御と共有可能な点が多い事は、実は容易に理解出来る。最近、各種のコンポーネントウェアが、かなり市場から入手する事が可能になってきており、不足な分についても共同開発を可能にするコンテナが実用的になってきた。これらの状況が、この方式を議論する事になった理由である。このコンポーネントウェアを載せる事ができる基本的な器を加速器制御ソフト・カーネル (COACK : Component ware Oriented Accelerator Control Kernel) と呼ぶ事にする。

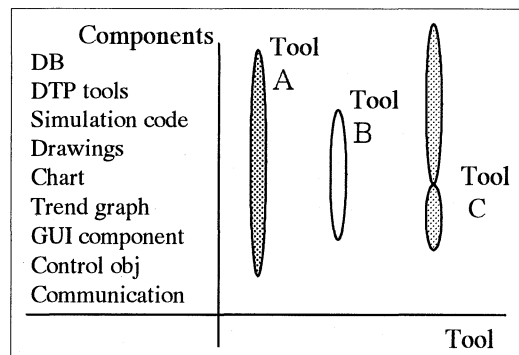


図1 ツールとコンポーネント

### 3. コンポーネントウェア

従来、VBX、OCXはVB(Visual Basic)上でコンポーネントウェアを可能にするツールとして市場に出回っていた。また、ヨーロッパでは、手作りで加速器特有のコンポーネントウェアを構築し共有しようとのアイデアも現われ、幾つかの研究所間でACOP (Accelerator Component Oriented Programming) プロジェクトとして進められた。最近では、ネットワークを越えて自由にダウンロード出来、又、組み込み運用が可能なActiveXが、Windows系上で可能になり、この仕組みを導入する事で加速器制御も大きく概念を変える事が可能になったと言える。ActiveXの形態(図2)として以下の項目がマイクロソフトより準備されている。1) ActiveXコントロール、2) コードコンポーネント(exe, DLL)、3) ActiveXドキュメント等の環境が既にVBレベルでもver.5から可能になって来た。Javaより便利な点も幾つか明らかになった。勿論、弱点も同様である。これらによって、コンポーネントで市場に無いものはActiveXによって構築でき汎用化が可能になった。市場にある物は、それらを採用し、無いもののみ加速器特有なコンポーネントとして、独自に開発し、他のユーザーも利用可能なようにActiveXを汎用化し、サーバーにおく事で、他の加速器でもネットワークを介してコンポーネントウェアを自分の所にも取り込み、加速器制御ソフトを独自にカスタマイズ出来る様になる。最近では、LabVIEWの様なツールでさえActiveXの導入でコンポーネント化されたツールと見る事が出来る様になった。

IE	HTML MS-word, Excel	ActiveX control, ActiveMovie VisualBasic
Application	ActiveX document	ActiveX script

図2 ActiveX形態

### 4. タスク分析

ActiveXコンポーネントウェアによって加速器制御構築が何処まで可能であるかは、ドメインにおける必要なタスク分析をもって議論・結論する事が可能である。つまり必要なコンポーネントが自由に手に入り、必要条件を満たせば、加速器制御ツールより柔軟なツールとなる事は明白である。従来の加速器制御ツールは、汎用化されていないコンポーネントから成り立っている点で対照的であった。ActiveXによって加速器制御ツールの構築が可能かどうか探る為に、まず加速器制御におけるレイヤー分類でそのタスクについて分析を行なうことを試みた。加速器制御の一般系として、次の3層構造が常識的である。

- 1) デバイスレイヤー処理系
- 2) データベース、中間処理・各種統計処理<sup>[2]</sup>
- 3) ヒューマン・インターフェース系<sup>[3]</sup>

これら3層の詳細を更に分析してみると、

- a) デバイス・ローカル制御、ローカル・フィードバック、ローカルインテリジェンシー機能、上位層間通信
- b) 運転データのデータベース、デバイス・テーブル、各種データベース、ロギング、データ処理系、統計処理系、コード運転
- c) 操作系(直接操作系、間接操作系)、一次二次表示系、トレンド、グラフ表示系一般情報(Web)、図面、DTP系、写真・画像系

等に分類され、b, c) 層については、多くの場合市場のコンポーネントウェアを採用できるケースが多い。a) 層については、加速器固有のものが多く、既に開発済みのものをコンポーネントウェアとして汎用化するか、新規開発が必要である。

### 5. 加速器制御形態分析

前述のタスク分析の中で、中間レイヤーにおける処理は、必ずしも中間にある必要はなく、CPUのゆとりのあるステーションに分散しても問題はない。むしろ多くがコンポーネント化され得るとして、そのコンポーネントを分離するとすれば、

- 1) 上位層(ヒューマンインターフェース部)、
- 2) 下位層(デバイスレイヤー)の2層であり、この間には、はっきりしたタスクの目的と機能の大きな差がある事から、分離出来る事は、

明白である。しかしながら、上位層のヒューマンインターフェース層では、PCの問題点として抱えている頻繁なアップグレードの問題があり、これに対する解決策としてアプリケーションサーバーによる更新の自動化は必要不可欠である。これは、特に上位層に顕著に起きる現象である。加速器が実験装置である以上、時代に即した便利なユーザー環境を提供する事は不可欠である。従って個別アップグレード方式は、台数が多いシステムでは保守の点で難点があり、このアプリケーションサーバー方式こそ、現在採用すべき方向である。一方、下位層のデバイスレイヤーにおいては、一旦建設されると10年なりそう頻繁に変更される事はない。

#### a) 中間層のタスク分散

タスクが特定のCPUに常駐して、通信により処理を遂行していた過去のシステムでは、中間層の位置づけは避け難いものであった。しかしながら、本件で提案しているシステムでは、分散可能コンポーネントウェアをアプリケーションサーバーからとってくる方式なので、中間層の位置づけ

は不要になる。一方、他の加速器からも見えるアプリ・サーバー（図3）が必要になる。

分散オブジェクトの為のコンテナが運用でき、タスクが分散的に配置可能になった現在、中間の処理層は最適分散を行なった方がCPUの利用率、システムの簡略化、保守性の向上の点で有利となってきた。

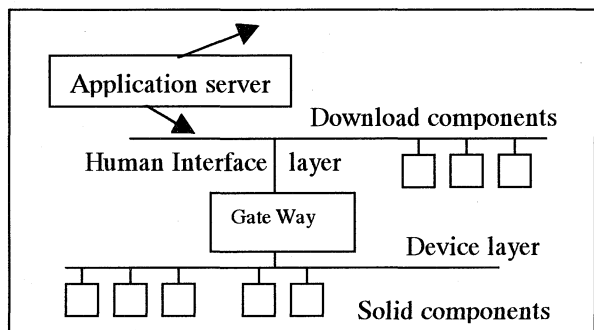


図3 レイヤー&コンポーネントサーバー

#### b) 加速器制御ソフト・カーネル

どんなデバイス系を接続していこうが、どんな表示系にアップグレードしようとも、変わらず必要である加速器制御の核になるソフトが前述の上位層と下位層に分布している。3層から2層構造にする場合、通信がN対Nで行なわれ、トラフィックの爆発が起こり得るので、変化情報のみを特定のステーションにアサインして送る等のゲートウェイ的処理ステーションは不可欠である。層間に置くゲートウェイは、トラフィック分離を行なうのみならず、幾つかの機能を持たせる事になる。この様な上位、下位層に依存せず存在し、どの加速器制御にも共通して使用し得る加速器制御カーネルソフトが必要である事は明白で、上位下位層のコンポーネントウェアと同じ形態をとる事が出来る。加速器制御カーネルソフトに分類すべくコンポーネントウェアとして

- 1) データベース・エンジン
- 2) Web server, アプリケーションサーバー
- 3) 通信処理系、コマンド記録サーバー

等が必要である。加速制御システムを集約的に管理する上でこのCOACKは不可欠である。COACKは、どの加速器でも必要であるので、コンポーネントウェアレベルで汎用化すれば、他の加速器でも共有・運用でき大変便利になる。我々は、NTに特化してCOACKを構築した。初めのバージョンでは、汎用化コンポーネントウェア化が不便な時期に製作したので、フェーズ2として部分的段階的にActiveXによる移行を進めている。

#### 6. アプリケーション・サーバー

アプリケーション・サーバーの設置は、柔軟な分散オブジェクトを管理し、ユーザーに提供する為

に、従来のファイルサーバーにも増して重要な位置づけにあり、セキュリティー等の考慮も新たに必要である。この環境下で、複数の加速器間で共有可能なオブジェクトがリアルタイムで運用される事になる。また、開発されたものも、共有を目的としてサーバーに載せて管理する事になる。新しい具体的な、共有オブジェクトのサーバーとして存在する事になる。クライアントから、ブラウザ又はVBから呼び出し、或いはオブジェクト追加によって運用される。

#### 7. アプリケーション・プッシュ (AP)

市販のAP技術も幾つか存在するが、今回は、任意のステーションに必要なソフトをリモートから自由にプッシュし起動させる仕組みも開発しアプリケーションサーバーに加えた。基本的にはローカル・アプリの起動とアプリケーション・サーバーからのアプリの起動との両方が可能である。本機能は加速器運転アプリソフトの運用上、及び、その保守上重要な機能である。これによってクライアント毎のバージョンアップ作業や運用変更を一括で行なう事が可能になり、大幅に手間を省く事が可能になった。

#### まとめ

加速器制御の一部において Windows NTでシステム構築を行なって来たが、1998年からVB上で分散可能なCOMへの移行が可能になって状況が変わった。その中で、加速器制御3層構造を変革して、加速器制御カーネルソフト(COACK)とACOPの組み合わせにより、新しい汎用化ソフト構築の可能性を見出した。PCベース制御では、確立した加速器制御ツールは未だ無かったので、次世代概念で従来のツールに代わり標準化に迫る制御ソフトが望まれていたところであった。また、ActiveXのオブジェクトをAPサーバーに置いて管理する事で、新しいスタイルのオブジェクト・分散と共有形態が加速器施設を越えて可能になる事を述べた。

#### Reference

- [1] P. Duval  
「A report on PCaPAC and status of the HERA control system」, IWCSMSA'96 (Tsukuba)
- [2] 田中、阿部  
「加速器運転支援用データベースシステム構築」, リニアック研究会 97年 (仙台)
- [3] Abe, Tanaka, etc.  
「Feedback of operators' experiences to console programs in the KEK e<sup>-</sup>/e<sup>+</sup> Linac」  
ICALEPCS'97 (北京)