

PFにおけるビームライン制御標準化に関する取り組み

小菅隆、濁川和幸、PF ビームライン制御標準化 WG

高エネルギー加速器研究機構物質構造科学研究所

概要

高エネルギー加速器研究機構放射光研究施設(以下 PF)では、現在 28 本(PF-2.5GeV Ring に 22 本、PF-AR に 6 本)のメインビームラインにそれぞれ設置された 70 を超える実験ステーションにおいて、シンクロトロン放射光を用いたさまざまな実験が行われている。制御系については実験内容によって様々であるが、パルスモーター等で駆動する光学系等に関しては共通の部分が多く、ネットワークを利用したシステムである STARS(Simple Transmission and Retrieval System)^{[1][2]} を用いて標準化の作業を進めている。ここでは PF におけるビームライン制御標準化の詳細及び現状について報告する。

1 はじめに

PF において放射光利用実験を行うためには、実験ステーション上流に設置されているミラーなどの集光光学系、モノクロメータと呼ばれる単色計、スリット等の光学素子の駆動を制御するためのシステムが必要である。従来これらの制御システム構築は実験ステーション担当者の裁量に任されており、多種多様な制御システムが構築される事となった。実験の分野は様々であり、これら多種多様な制御システムはそれぞれの実験分野に於いて効率的に動作するが、その一方以下のような問題が発生する事となった。

- ビームライン担当者が個々にシステムを構築する事に費やすマンパワーの浪費
- ビームライン担当者の交代に於いてたとえマニュアルが整備されていても、スムーズな引継ぎには問題が生じる可能性がある。
- 放射光を利用するユーザーの立場からみると、同じような実験を行う場合ですら、実験ステーションが異なると別の制御システムの操作方法を習得しなければならない可能性がある。

これらの問題を改善するためにはビームラインの制御系をある程度標準化することが非常に重要であるとの認識から、ビームライン制御の標準化への取り組みがスタートした。

2 標準化に対する作業

ビームライン制御の標準化に際し、標準化のためのワーキンググループ(以下 WG)が召集されることとなり、PF で使用されているシステムの現状把握、共通化可能な部分及び標準の設定についての検討が行われた。ここではこの WG で行った作業について述べる。

2.1 STARS の採用について

放射光利用実験を行うための制御システムは、ビームラインの上流から下流まで数多くのコンポーネントの状態を協調的に計測・制御しなければならない。また、これらのコンポーネントは未来永劫同じものが使用されるわけでもなく、蓄積リングやビームラインのアップグレードに伴って絶えず更新されてゆく。このような「進化し続けるビームラインの計測・制御システム」を作り上げて行くためには、はじめから全体の制御システムを 1 個のかたまりとして開発するのではなく、コンポーネントごとに分類して、個別システムを開発・統合する「分散化」の考え方が重要となる。STARS はまさに上記のような視野に立って開発された、

ネットワーク上でアプリケーション間通信を司る「交通整理のためのクライアント・サーバプログラム」であり、以下のような特徴を持っている。

- アプリケーション間通信は全て TCP/IP ソケットを利用したテキストベースのコマンド送受により行われる。
- 開発言語や OS の選択の幅が広い。
- デバッグを行う場合にも TELNET などのツールを利用することが可能。
- コア(サーバ)の部分は Perl を使って開発されており、様々なプラットフォーム上で動作可能。

本 WG では以上のことから STARS を制御システムの基盤として、PF におけるビームライン制御の標準化のための仕様の作成を行うこととした。

2.2 共通コマンドの提示

すでに STARS が導入され運用されている構造生物グループのビームラインに関しては、綿密な打合せのもと効率的なコマンドの策定が行われていた。しかし、STARS の開発グループとしては、予め STARS 自身が持つ「コマンド」、「コマンドに対するリプライ」、「イベント」のメッセージに関する規定以外に特にビームライン制御におけるコマンド等は提示していなかった。

今回、ユーザーが持ち込んだ PC をビームラインの制御システムに接続する事も考慮に入れる必要が出てきたことから、構造生物グループで作成されたコマンド群の他に、ユーザーがモノクロメータなどの光学装置を操作するためのコマンドを別途設ける事となった。

2.3 使用頻度の高い装置の調査

標準化を進める上で、本 WG では各ビームライン担当者に協力を依頼し、実際に使用しているハードウェアについてアンケート調査を行った。その結果を元に標準化の進め方や、実際に開発を行ってゆく上での問題点の洗い出しや、I/O クライアント開発の優先順位等を決定する事とした。

3 標準化の実際と開発状況

3.1 標準化されたシステムの構成

図 1 に STARS を使用したクライアント・サーバ型制御システムの構成図を示す。STARS サーバに接続されているクライアントのうち、ユーザー側のクライアントをユーザークライアント、装置側のクライアントを I/O クライアントと呼ぶ。個別コンポーネントの I/O クライアントとユーザークライアントを個別に開

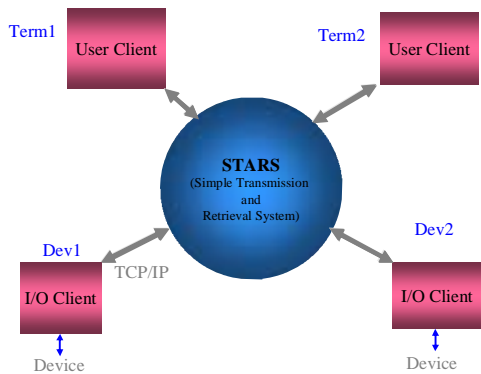


図 1. STARS を使用したシステムの構成

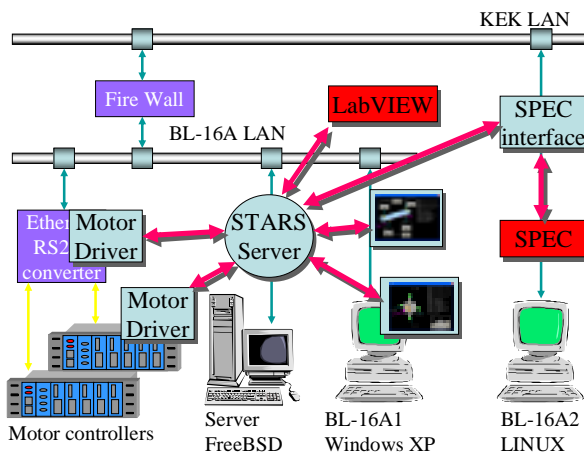


図 2. BL-16A での例

発し、それらを STARS サーバを介して有機的に結合させることで、分散化した計測・制御システムの開発が可能となる。また、図 2 は実際のビームライン BL-16A に導入した例で、ビームライン側で用意したクライアントプログラム他にユーザー側で用意したプログラム(図中、SPEC、LabVIEW)が同時に接続可能となっている。

3.2 ハードウェアの抽象化

標準化されたシステムでは、ユーザーが持ち込む PC からコマンドを送る場合など、パルスモーターなどの実際のハードウェアについて直接意識する必要の無い構造を採用している。たとえばモノクロメータを制御する場合(図 3 参照)、ユーザーは直接モーター用の STARS I/O クライアントにコマンドを送るのではなく、モノクロメータを司る STARS クライアント「Mono」に対してコマンドを送信する。次にコマンドを受け取った STARS クライアント「Mono」は送出するパルス数などを計算した後、モーター用 STARS I/O クライアントにコマンドを送出する。この事で、パルスモーターなどの構成がビームラインのアップグレードにより変更されても、ユーザー側のプログラムを変更する必要はなくなる。また、このビームラインコンポーネントを司る STARS クライアントにシミュレーション機能を追加すれば、実際のハードウェアが無い時点におけるユーザークライアントプログラムの開発が可能となる。

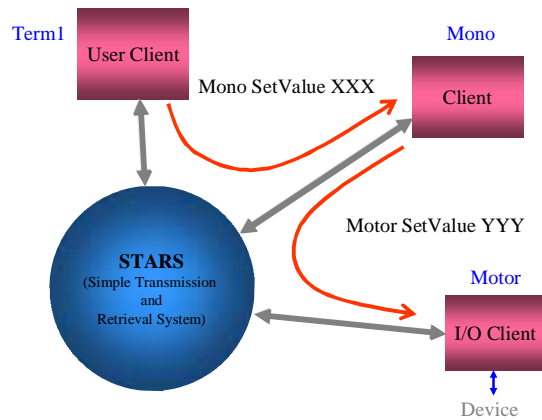


図 3. ハードウェアの抽象化

3.3 コマンドの実際

ここでは今回 WG で提示された、実際にユーザーがビームラインを制御するためのコマンドおよび受け取る事のできるイベントを示す。

基本的なコマンドは値の取得(GetValue)、値のセット(SetValue)、駆動中であるかの問い合わせ(IsBusy)、駆動の停止(Stop)からなる。コマンドはビームラインコンポーネント(実際にはビームラインコンポーネントを司る STARS クライアント)の名前を指定して"コンポーネント名 コマンド"のように送出する事となる。ただし、それぞれのビームラインコンポーネント名はピリオドを使用することによって階層化されたイメージで表現されるので、たとえば分光器のエネルギーの場合であれば"Mono.Energy"のように表現される。以下はその例である。

例: Mono.Energy GetValue (分光器のエネルギーを取得)

また、コマンドを送出すると必ず"@ "で始まるリプライメッセージが"コンポーネント > 接続時に指定したターミナル名? @コマンド? 結果"のように送られて来る。以下の例はユーザークライアントのターミナル名を"user1"として接続した場合の上の例に対する回答である。

例: Mono.Energy>user1 @GetValue xxxx (モノクロのエネルギーが xxxx eV)

また、イベントについては、あらかじめ"System flgon コンポーネント名"のようなコマンドを送出すると指定したコンポーネントの状態が変化したときに"イベントメッセージ"を受け取る事が可能となる。一度 System flgon コマンドを送出すると、そのコンポーネントの状態が変化するたびに"コンポーネント名 > 接続時に指定したターミナル名"に加えて"_ "(アンダースコア)から始まるイベントメッセージが送られて来る。基本的なイベントとしては値が変化した時のイベント(_ChangedeValue)及び駆動が開始され等で Stop コマン

ド以外を受け取らない事を示すためのイベント(_ChangedIsBusy)がある。

なお、イベントの受け取りをとりやめたい場合には"System flgoff コンポーネント名"のようにフラグオフコマンドを送信する。以下は分光器の角度に対して flgon したときの例である。

例: 分光器の角度の状態が変わったらイベントを受け取る(user1 のターミナル名で接続した場合)

System flgon Mono.angle (分光器の角度の状態が変わったらイベントを受け取る)

System>user1 @flgon Node Mono.angle has been registered. (リプライメッセージ)

:

Mono.angle _ChangedIsBusy 1 (分光器が駆動された)

3.4 I/O クライアントの開発状況

現在、WG で行われたアンケート結果に基づき、それぞれのハードウェアに対応した STARS I/O クライアントの開発が随時進行中である。現在の開発状況を表 1 に示す。

表 1. ハードウェアの対応状況

プログラム名	サポート対象		
	メーカー	型番	仕様
WE7000	YOKOGAWA	WE7562	MCA モジュール
		WE7521	4ch タイミングカウンターモジュール
		WE7262	32bit デジタル I/O モジュール
	TAIYO	TWM5501	2ch2 台同時駆動パルスモーターコントローラ
pm16c04	TSUJI DENSHI	PM16C-04S	16ch4 台同時駆動パルスモーターコントローラ
pm16c02		PM16C-04	PM16C-04S の Net 対応版
npm2c01		PM16C-02Z	16ch2 台同時駆動パルスモーターコントローラ
sc400	KOHZU	NPM2C-01	2ch パルスモーターコントローラ Net 対応廉価版
SC-400 (実機未検証)		SC-200	2ch パルスモーターコントローラ
		SC-800 (実機未検証)	4ch パルスモーターコントローラ
m2701drv	KEITHLEY	Model 2000-SCAN	デジタルマルチメータ (6.5 桁) 10 チャンネルスキャナカード
		Model 2701+7700	デジタルマルチメータ (6.5 桁、22 ビット積分型 AD、Net 対応) + 7700 (プラグインスイッチ/測定モジュール、20ch、差動入力)

4 まとめと今後

今回 WG では STARS 採用の決定、標準化されたシステムの構成の提示、使用頻度の高い装置の調査、標準化されたコマンドの提示がなされた。現在、WG では最終とりまとめの段階に入っている。

また、STARS は BL-1A、BL-1B、BL-6A、NW-12、BL-5、BL-16A、BL-20A、BL-28 など導入が行われており、随時他のビームラインへの導入及び標準化の作業を進めてゆく予定である。

参考文献

- [1] 小菅隆, 小山篤, “簡易メッセージ配信システム(STARS)の入退室管理システムへの応用”, 平成 14 年度技術研究会, 核融合科学研究所, 2002
- [2] Takashi Kosuge, Yuuki Saito, “RECENT PROGRESS OF STARS”, Proceedings of PCaPAC2005, Hayama, Japan, 2005