

分散制御フレームワーク MADOCA II を LabVIEW で

公益財団法人高輝度光科学研究センター
制御・情報部門 松下 智裕、古川 行人、松本 崇博

Abstract

前号 (Vol.19 No.4) にて、次世代 SPring-8制御フレームワーク MADOCA II (Message And Database Oriented Control Architecture II) について解説を行った。2014年にこのフレームワークを加速器制御に導入し、利用を開始した。複数のコンピューターを同期制御し、柔軟で堅牢な制御系を構築することができる。さらに、様々な OS や言語で動くように細心を払ってコーディングされており、Windows もサポートする。これにより SPring-8ユーザー実験でも利用できる下地が整った。ユーザー実験では LabVIEW による計測プログラムが多く使用されている。そこで、LabVIEW から MADOCA II を容易に利用できるように、VI ライブラリの開発を行った。LabVIEW で SPring-8の光源制御システムや複数の実験制御コンピューターを連動させた高度な実験プログラムが開発できるようになる。

1. はじめに

挑戦的な実験を新しく始めるには、装置の開発に加えて、制御プログラムを開発することがほぼ必須である。小規模な実験であれば、1つのコンピューターに検出器やモーターの制御線を繋ぎ込み、1つのプログラムで制御できるだろう。しかし、1つのコンピューターに数多くの機器を繋ぎ込んで高度な実験を行うのには限界がある。しかも、設計が悪ければ、実験が変わる時にコンピューターに繋ぐ配線を変え、プログラムを全て変更する必要に迫られるだろう。

これを解決する方法として、分散制御がある。1つのプログラム (コンピューター) で制御する機器の数を少なくし、複数のプログラム (コンピューター) を協調させて、全体を制御するのである。さらに、機器制御プログラムと実験の手順をコントロールするプログラムに役割を分けて製作する。こうすれば、実験の変更時においては、実験手順のプログラムだけ変更すれば良い。分散制御はプログラム開発負担を軽減し、短時間で堅牢製と柔軟性を同時に兼ね備えた実験制御を構築する方法の1つである。

分散制御するフレームワークは幾つか存在する。前号 (Vol.19 No.4) にて、解説した SPring-8制御フレームワーク MADOCA II^[1] もその1つである。これは SPring-8を制御する 400台以上のコン

ピューターを繋ぎ、安定的に運用できるレベルの堅牢製を持つ。これをユーザー実験で利用できるようになれば、実験プログラムの開発の負担を大きく軽減できる可能性がある。

SPring-8のユーザー実験制御では LabVIEW がよく使われる。そこで、MADOCA II を容易に利用できるようにするために、LabVIEW 用のコンポーネント開発を進めている。その状況について報告する。

2. MADOCA II

MADOCA II は分散制御フレームワークであり、複数のコンピューターが協調して動作するための通信規格でもある。下記の大きな特徴を持つ。

- 1) コンピューター間の通信メッセージには人間が読むことができる英語第5文型の S/V/O/C に沿った文字列を使う。
- 2) 制御対象機器には全てユニークなオブジェクト名 (O) を付与する。
- 3) S/V/O/C メッセージをフレームワークに送付すると、オブジェクト (O) を制御しているプログラムをフレームワークが探し出し、メッセージを伝達する。
- 4) コマンドのメッセージを送ると、必ず返事のメッセージが戻る。
- 5) メッセージに大きなサイズのデータを添付して送ることができる。

通信文字列は制御の専門家ではなく機器開発担当者に分かりやすいS/V/O/C形となっている。Sはメッセージを送ったアプリケーションを特定するためのもので、通常はフレームワークが自動的に設定する。MADOCA IIを使ったプログラムを作成する人はV/O/Cのみを考えれば良い。例えばBL25SUの挿入光源1のギャップを50.0 mmに設定するには、

```
put/bl_id25_gap_1/50.0mm
```

のように記述する。Vは動作を表し、putの場合は動作指令、getの場合は値取得になる。Oは操作対象機器名であり、ここではbl_id25_gap_1である。Cは操作内容であり50.0 mmの値を指定している。

図1にフレームワークの概略図を示す。MADOCA IIでは機器を制御するプログラムをEM2(Equipment Manager 2)と呼んでいる。メッセージを伝達するハブの役割をするサーバーがMS2(Message Server 2)である。MS2は常時起動しておく必要があり、パラメーターを与えると、別のコンピューター上のMS2へ接続する機能を持つ。

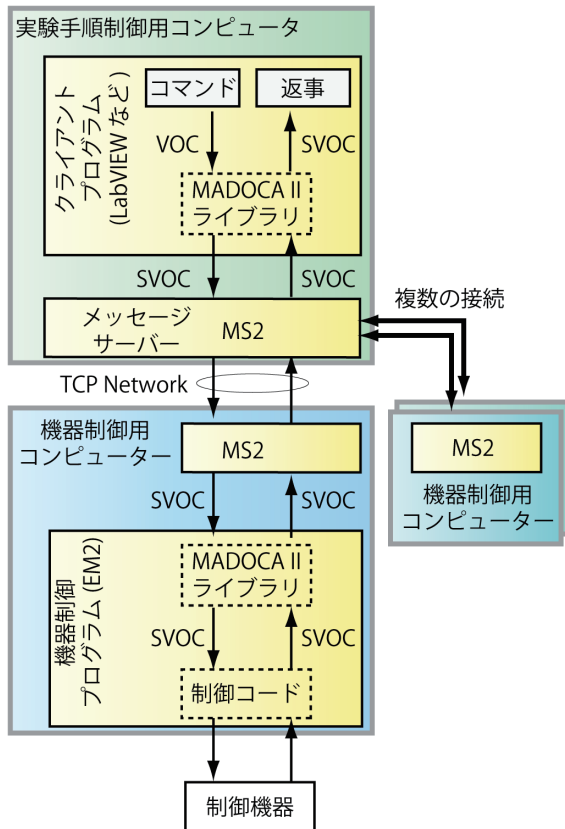


図1 MADOCA II フレームワーク概略図

機器制御プログラム (EM2) を起動すると、内部のMADOCA II ライブラリはMS2に繋いで、制御している機器のオブジェクト名(O)を登録する。登録されたオブジェクトのリストは接続した複数のMS2で共有される。

実験手順がプログラムされたクライアントを起動すると、MADOCA II ライブラリは同様にMS2に接続する。ここでV/O/Cのコマンドをライブラリに送ると、オブジェクト(O)を制御している機器制御プログラム (EM2) へと、メッセージを届けてくれる。EM2は届いたメッセージを解釈した後、返事をMS2に送出する。返事も自動的に伝達されてコマンド発行元に戻る。

さらに、S/V/O/Cメッセージに添付して大きなデータを送ることができる。添付データの構造は、ファイルシステムのようなデータ構造となっていて、データに対して、キー名(ファイル名のようなもの)を設定して格納できる。さらにフォルダーを作成することもできるため、複雑な構造を持つデータをそのまま送れる。コンピューターで使われるほぼ全てのプリミティブ型とその配列(文字列、整数、浮動小数点数、整数配列、浮動小数点配列)をサポートとする。これにより、複数のスペクトル、測定条件、画像や文字列など、巨大なデータを1つのメッセージで送ることができる。

3. LabVIEW コンポーネント開発

前述のようにSPring-8ではLabVIEWが実験制御で多用されている。したがって、LabVIEWをMADOCA II フレームワークと接続できれば大きな恩恵を得ることができる。そこで、MADOCA IIのLabVIEWコンポーネントのプロトタイプを2013年度に開発した^[2]。MADOCA IIで使用されているZeroMQなどのライブラリは、VIライブラリも提供されており、これを用いてLabVIEWでMADOCA IIの動作をプログラミングした。このプロトタイプにより、LabVIEWでの動作検証や通信性能などを確かめることができた。反面、VIで開発する難しさも浮き彫りにした。MADOCA IIは、C/C++言語で開発されている。C/C++のライブラリが変更されると、VIもそれにならって、プログラムを改修しなければならない。この実装作業は二度手間であり、多くの時間とコストを消費する。そこで、2014年度は製作方針を転換した。C/C++言語でLabVIEW用のダイナミックリンクライブラ

り (DLL) を開発し、LabVIEW からはその DLL の関数を呼び出すだけの構成にした。MADDOCA II の内部動作は DLL 関数が処理する。これにより、LabVIEW のプログラムは内部処理を気にすることなく、実験手順のプログラミングのみに専念できる。また、MADDOCA II の内部動作仕様が変更されても、新しいバージョンの DLL に差し替えるだけで、バージョンアップできる。

この方針を元に LabVIEW が利用できる形の DLL 関数群の設計と開発を行った。また、LabVIEW の開発ターゲットは、このライブラリを長期間利用する観点から、現時点での最新版 (2014) を選んだ。

LabVIEW のプログラミングをイメージしていたくために、提供予定の主な VI を下記に示す。ただし、開発中であるので、仕様が変更される場合がある点についてはご了承ください。



MS2_OPEN

メッセージサーバー (MS2) に接続する。



MSG_BUILD_INIT

メッセージを初期化する。



MSG_BUILD_INF

メッセージの V/O/C を設定する。



MSG_BUILD_VALUE

メッセージに添付するデータのキーと値のペアを設定する。データは文字列、整数、浮動小数点など複数の型をサポートする。



MSG_BUILD_ARRAY

メッセージに添付するデータのキーと配列のペアを設定する。データは複数の型をサポートする。



MSG_BUILD_OPENCHILD

メッセージの添付データに子フォルダーを作り、そこに移動する。



MSG_BUILD_CLOSECHILD

メッセージの添付データの作業中の子フォルダーを閉じて、親フォルダーに移動する。



MSG_CLIENT_SENDRECV

メッセージを MS2 に送り、返事を受け取る。



MSG_FETCH_INF

返事のメッセージから V/O/C を読み取る。



MSG_FETCH_VALUE

キーを指定して、返事メッセージに添付された値を読み取る。



MSG_FETCH_ARRAY

キーを指定して、返事メッセージに添付された配列を読み取る。



MSG_FETCH_OPENCHILD

返事メッセージに添付されたデータの子フォルダーに移動する。



MSG_FETCH_CLOSECHILD

添付データの作業中の子フォルダーを閉じて、親フォルダーに移動する。



MS2_CLOSE

MS2 との接続を終了する。

上記の VI を用いた、カメラ画像を取得するクライアントプログラムの例を図2に示す。カメラ画像を提供する機器制御プログラム (EM2) が MS2 にすでに接続されている状況を想定している。この EM2 が提供するカメラのオブジェクトの名前 (O) は、“camera” である。このプログラムを実行すると、MS2 に接続して camera に対して初期化コマンド (put/camera/init) を送り、続いて、画像取得コマンド (get/camera/acquire_image) を送る。その返事には測定条件や、画像データが添付される。画像の幅 (image_width) や高さ (image_height) を、対応するキーを指定して取得し、画像データは image_data をキーにして配列を得る。この配列を2次元配列に変換して、2DPicture に送って、表示する仕組みである。最後に MS2 への接続を閉じる VI が呼び出される。このように、機器制御やデー

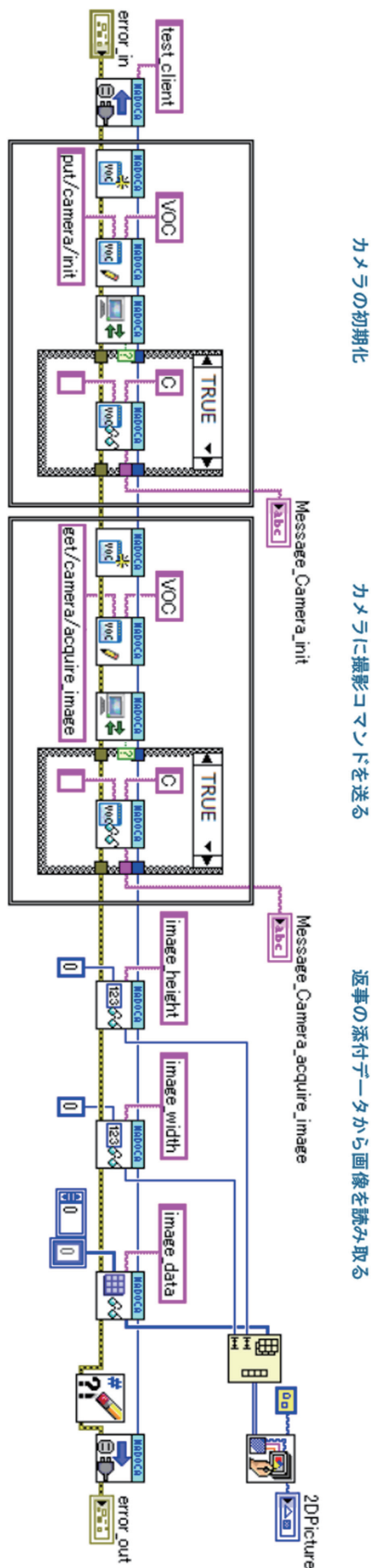


図2 カメラ画像取得するクライアントプログラムの例

タ転送を含む分散制御が、MADOCA II – LabVIEW の連携によって柔軟に実現できるようになる。

4. まとめ

容易な手続きで MADOCA II フレームワークの様々な関数群を利用できるような VI ライブラリの設計を行った。クライアント用の VI 群を紹介したが、LabVIEW で機器制御プログラム (EM2) も開発できる VI 群も同時に開発している。2015 年度からの利用を見込んでおり、これから数年かけて機能向上を図っていく予定である。

また、余談になるが、開発した DLL は LabVIEW だけでなく他の言語からも呼び出すことができる。C/C++ 言語はもちろんのこと、Python などのスクリプト言語にも対応する。すでに、幅広い応用が期待されている。

参考文献

- [1] T. Matsumoto *et al.*: “Next-Generation MADOCA The SPring-8 Control Framework”, Proceedings of ICALEPCS2013, San Francisco, California, USA, (2013) 944.
- [2] Y. Furukawa *et al.*: “MADOCA II Interface for LabVIEW”, Proceedings of ICALEPCS2013, San Francisco, California, USA, (2013) 410.

松下 智裕 *MATSUSHITA Tomohiro*

(公財)高輝度光科学研究センター 制御・情報部門
〒679-5198 兵庫県佐用郡佐用町光都1-1-1
TEL : 0791-58-0930
e-mail : matusita@spring8.or.jp

古川 行人 *FURUKAWA Yukito*

(公財)高輝度光科学研究センター 制御・情報部門
〒679-5198 兵庫県佐用郡佐用町光都1-1-1
TEL : 0791-58-1026
e-mail : furukawa@spring8.or.jp

松本 崇博 *MATSUMOTO Takahiro*

(公財)高輝度光科学研究センター 制御・情報部門
〒679-5198 兵庫県佐用郡佐用町光都1-1-1
TEL : 0791-58-0980
e-mail : matumot@spring8.or.jp