

## 加速器・ビームライン制御系の新制御フレームワーク MADOCA II への移行

公益財団法人高輝度光科学研究センター

制御・情報部門 古川 行人、松本 崇博、石井 美保

### Abstract

次世代 SPring-8 制御フレームワーク MADOCA II (Message And Database Oriented Control Architecture II) の開発を行い、2014A 期のユーザー運転から加速器・ビームライン制御系を、MADOCA から MADOCA II に更新した。長年に渡る MADOCA による制御系の運用、拡張を行って行く中で、新たに要望として出てきた可変長データの取り扱いや、Windows のサポート等の機能を MADOCA II に盛り込んでおり、LabVIEW にも対応した。今後は、実験制御・データ収集系への導入も進めていく。

### 1. はじめに

2014A 期のユーザー運転から、SPring-8 の加速器およびビームライン制御系を支える制御フレームワークを従来の MADOCA (Message And Database Oriented Control Architecture)<sup>[1]</sup> から MADOCA II に更新した<sup>[2]</sup>。本稿では、MADOCA 制御システムの概要および MADOCA の発展形である MADOCA II の特徴について紹介する。

制御フレームワークとは、制御システムのソフトウェアを構築するためのライブラリ、分散アーキテクチャを支えるための通信ミドルウェアおよびそれらを管理するためのツール類からなるソフトウェア群である。放射光関係の制御フレームワークとしては、EPICS (<http://www.aps.anl.gov/epics/>) や TANGO (<http://www.tango-controls.org/>) 等がよく知られている。MADOCA を含むこれらの制御フレームワークに共通する特徴は、ネットワークを介した分散制御系という点にある。SPring-8 のように規模の大きな施設では、中央制御室にオペレーション用端末を置き、そこから施設全体に散らばった制御対象の近くに置いた入出力用の制御用計算機（以下、フロントエンド計算機）に対して、ネットワークを介した制御を行うことが求められる。こういった分散制御の場合、1 台の計算機に全ての入出力が集約されている場合に比べて気を付けなければならない点が多々ある。例えば、どのフロントエンド計算機に指示を出せば自分のビームラインの挿入光源のギャップを変更できるのか？とか、他のビームラ

インの機器を誤って操作しないようにするにはどうすればいいのか等である。現代の制御フレームワークは、これらのことを意識しなくても制御用アプリケーションを作成できるように様々な仕組みが用意されている。

MADOCA もそのような機能を備えており、これまで SPring-8 の運転を支えてきた。しかしながら、SPring-8 稼働後の 10 数年間の技術の進歩にともない、制御フレームワークに求められる機能はより高度になってきている。特に 1995 年前後の MADOCA 設計時には、ネットワークや CPU、メモリ等のハードウェアやソフトウェア技術の制約により、波形データや画像データ等の実験制御に密接に関わるデータ（以下、転送毎にデータサイズが異なる可能性があるという意味で「可変長データ」と呼ぶ）の転送ができなかった。MADOCA 制御システムを加速器・ビームラインのコンポーネント制御から実験制御へ適用を進めていく中で、可変長データを取り扱えるようにすることや、制御対象の追加、変更をより容易にする等の機能が強く求められるようになってきた。そこで 2011 年より MADOCA II の開発に着手し、2014A 期のユーザー運転から、SPring-8 加速器およびビームライン制御に本格的に導入した。さらに現在、放射光実験制御・データ収集系への導入も進めている。

なお、MADOCA II では、MADOCA のもうひとつの柱であるデータベースを用いたログデータ収集機能に関しても、拡張性や利用しやすくする等の開

発を進めているが、紙面の都合で本稿では割愛させていただく。

## 2. MADOCA 制御フレームワーク

MADOCA 制御フレームワークの特徴は、計算機で取り扱いやすい2進数の塊を送って制御するのではなく、人が読むことのできる文字列を使って制御することにある。しかも、その文字列は、制御の専門家ではなく、担当する機器（挿入光源とか分光器等）の専門家に分かりやすい文字列である。どのコンピュータの何番目の I/O ボードのどのビットを on にするとか、どの ADC のどのチャンネルの値を読むという命令は、制御システム（ハードウェア）を作った人にしか何をしているのか分からない。

そうではなく、「ビームライン07の4台目のIDのギャップを50 mmにする」と記述すれば、制御アプリケーションは「操作したいものを操作したい順に操作するコマンドを記述する」ことで作成できる。

MADOCA 制御系では、英文法にならった S/V/O/C というフォーマットの文字列を用いることで上記のことを実現している。S はメッセージを送ったアプリケーションを特定するもので、MADOCA ライブラリが自動的に設定するので、プログラムを作成する人は、V/O/C のみを考えれば良い。例えば先のギャップの操作であれば、

```
put/bl_07is_id_gap_4/50.0mm
```

のように記述する。V は動作を表し、put だと動作指令、get だと値取得になる。他に補助的に show、ask 等が用いられる。O は操作対象であり、C は操作内容である。他に幾つか例を示す。

- get/bl\_01b1\_fe\_ivg\_1/pressure  
(BL01B1 フロントエンドの真空度1番の値取得)
- get/bl\_24in\_tc1\_mono\_1/angle  
(BL24XU の第1ブランチの分光器の角度取得)
- put/bl\_26b1\_st1\_shutter\_1/open  
(BL26B1 の実験ステーションの X 線シャッターを開ける)

さて、このようなメッセージを MADOCA 制御システムは、次のようにして制御用フロントエンド

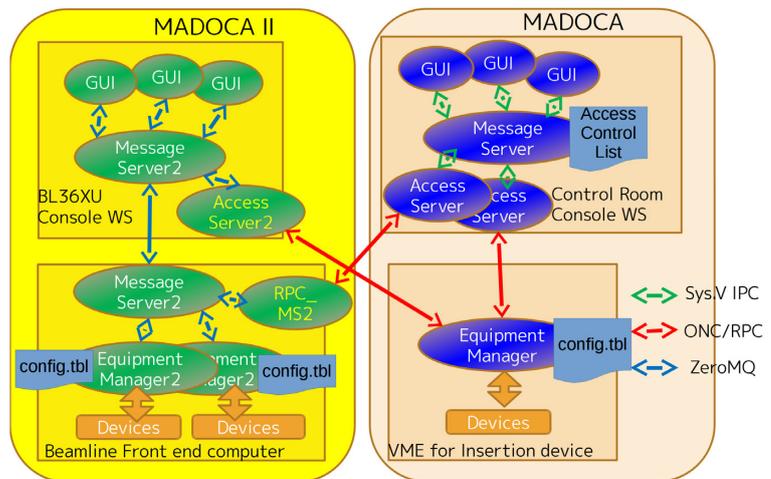


図1 BL36XU に導入した MADOCA (右) および MADOCA II (左) 制御システム構成。順次導入可能なように相互にアクセスするための機能も用意されている。

計算機に送る (図1右)。アプリケーションが組み立てたメッセージは、Message Server (MS) に送られる。MS は Access Control List (ACL) と呼ばれる表を持っており、S/V/O/C の O、つまり制御対象名から送り先の Access Server (AS) を見つけ出し、AS はそのメッセージをフロントエンド計算機に送る。フロントエンド計算機上で動作している Equipment Manager (EM) はデバイス構成表 (config.tbl) を持っており、それに基づいて、S/V/O/C の内容を解釈して入出力操作を行い、結果を AS に送り返す。この時 C は、結果を表す C' に書き換えられている。AS はメッセージを O/V/S/C' に並べ替えて MS に送る。MS は S からメッセージの送り主を特定し、そのアプリケーションに送り返す。

このように MADOCA 制御システムは、非常にシンプルにできており、様々な制御に柔軟に対応できる。1997年のSPring-8運転開始時には、蓄積リングおよびビームラインの制御系に MADOCA が適用されており、当初は異なる制御システムで運用されていたブースターシンクロトン、線型加速器も順次 MADOCA に置き換えられた。また、同じ敷地内にあり入射器を共有する NewSUBARU も MADOCA 制御システムによって運用されている。さらには、広島大学の小型放射光施設 HiSOR も、2003年のシステムアップグレードの際に MADOCA 制御システムを導入した。また、SPring-8のタンパク質結晶構造解析ビームラインの

実験ステーション制御系や、BL04B2の非品質回折計制御・データ収集にも MADOCA が導入されている。

X線自由電子レーザー施設 SACLA にも、MADOCA が全面的に導入されている。SACLA が SPring-8 と大きく異なる点は、実験ステーション・データ収集系にも当初から MADOCA 制御システムを導入したことにある。

### 3. MADOCA の制限と MADOCA II

このように様々な制御系に導入された MADOCA であるが、当初想定した加速器の制御から実験制御等へ守備範囲が拡大するに従って、次のような設計上の制限が問題になってきた。

- 1) メッセージが255文字に制限されているため、波形や画像等のデータ転送が難しく、共有ファイルシステム等の MADOCA フレームワーク外の手段が必要であること。特に実験制御・データ収集系では、2次元検出器の導入が急速に進み、これに対応する必要がある。
- 2) Windows では動作しない。MADOCA 設計時は Windows95 登場以前であり、Windows のネットワーク対応など不透明な点が多く、Windows 対応は検討されていなかった。結果として、Windows には移植しにくいフレームワークになっていた。しかし、市販の計測機器には Windows のみサポートのものが多数あり、実験制御・データ収集系を構築するには Windows 対応は外せない項目である。
- 3) 上位系（オペレータ端末）とフロントエンド計算機間の通信の並列化がされていない。図1右に示した MADOCA の AS と EM 間の通信には、RPC (Remote Procedure Call) と呼ばれる通信規格が用いられているが、これは同期型の通信である。すなわち、一度 AS から EM にメッセージが送られると、EM から返事が来る（もしくはタイムアウトする）まで次のメッセージを送ることはできない。RPC 通信は確実性が高い方法であるが、複数の処理をフロントエンド計算機に並行に処理させることができないため、制御性は低くなる。
- 4) 上位系同士の通信がサポートされていない。加速器制御では必要性が低いが、実験制御とビームライン制御では上位制御系間の通信が必要になる。例えば、標準2結晶分光器では複数の制

御対象（モータ）を同時に駆動して波長変更を行うようになっており、これらの制御対象をまとめて1つの仮想機器として実験ユーザーに提供する機能が、ビームライン制御上位系に備わっているためである。

これらの制限を取り除くため、2011年より MADOCA II の開発に着手した。Windows をサポートし、可変長データを扱える非同期通信プロトコルおよびそのライブラリがあれば、上記の制限は一気に取り除くことができる。調査したところ、ZeroMQ (<http://www.zeromq.org/>) というライブラリがこれらの条件を満たすことが分かり、これを用いて新しい MADOCA II の設計・製作を行った。図1左が新しい MADOCA II 制御系である。重要な点は、全ての通信を新しく設計した Message Server 2 (MS2) に集約し、プロトコルを ZeroMQ に一本化したことである (MADOCA ではホスト間の通信は RPC、同一ホスト内の通信は Unix 系に特化した System V IPC (Message Queue) であった)。これにより、上記の制限は次のように解消される。

- 1) ZeroMQ は、マルチパートメッセージという通信方法を提供しており、これによりアプリケーション側からは、1回の通信の中に複数の情報を詰め込むことができる。最初のパートには、S/V/O/C で規定されるメッセージおよび管理上必須な情報を載せ、もし可変長データも一緒に送りたいければ2番目のパートに載せるようにすることで可変長データを簡単に扱うことができる。
- 2) ZeroMQ ライブラリは、Windows もサポートしているため、フレームワーク全体も Windows でもコンパイル・動作可能のように製作した。
- 3) ZeroMQ では、非同期通信をサポートしているので（同期通信もある）、非同期通信を行うことと、旧 MADOCA ではできなかった複数の EM をフロントエンド計算機で動作させるように設計し、並行処理を可能とした。
- 4) 全てのホスト間の通信は MS2 が担うため、上位系計算機とフロントエンド計算機の間には差別がなくなり、上位系同士でもフロントエンド計算機同士でも接続が可能となった。

なお、開発にあたっては既存の加速器、ビームライン制御用ソフトウェアの変更は最小限になるようにし、基本的にはソースコードを修正せず、再コンパイルのみで MADOCA II へ移行できるようにした。

#### 4. MADOCA II の導入

MADOCA II を全系制御に導入するのに先立って、2012B 期からコミッショニングが始まった BL36XU に先行導入し、運用経験を積んだ。BL36XU 以外の制御系は MADOCA であるため、図1に示すように MADOCA II と MADOCA 間を相互に接続できるようにし、部分的な導入が可能になるようにした。この先行導入において、MADOCA II に起因するトラブルはほとんどなかったが、動作状況をモニタするツールの整備等、より安定に運用するための多くの知見が得られた。

この結果を反映させた上で、2013B 期から SACLA 実験制御系を MADOCA から MADOCA II に置き換えて運用している。さらに、SPring-8 の加速器・ビームライン制御の上位制御系すべてと、フロントエンド制御系の一部を 2014A 期から MADOCA II に置き換えて運転を開始した。いずれもあらかじめ数ヶ月間の準備期間を設けて、テストベンチやマシンスタディ等の時間を利用して動作確認を行い、問題がないことを確認した上で導入を行っており、導入にともなう大きなトラブルなく安定した運転を行っている。

また、可変長データを取り扱えるようになったことから、SPring-8 蓄積リングの2次元放射光干渉計読み出し<sup>[3]</sup> や、電子ビーム位置モニタ読み出しシステム<sup>[4]</sup> に適用している。電子ビーム位置モニタ読み出しシステムは、フロントエンド計算機が Windows + LabVIEW で構築されており、MADOCA II - LabVIEW インターフェイス (プロトタイプ) を開発した。

今後、MADOCA II を実験制御系に導入していくことを計画している。MADOCA II では、MS2 を核にして様々なデータ収集モジュールや解析モジュールを組み合わせ、柔軟かつ容易に様々な実験に対応することができる。SPring-8 ビームラインの実験制御系では、Windows + LabVIEW によりシステム構築されているケースが数多くある。先に述べた MADOCA II - LabVIEW インターフェイス (プロトタイプ) を使うことで得られた知見を基に、より使いやすい形に整理した MADOCA II - LabVIEW インターフェイスの設計・製作を進めており、今年度中に利用可能になるよう整備を進めている。本件に関しては、次号で詳しく紹介する予定である。

また本稿では触れることができなかったが、MADOCA II のもう1つの柱であるログデータ収集

システムの更新については、冬期運転停止期間に導入すべく準備を進めている。

#### 5. 謝辞

MADOCA II の開発・導入にあたっては、SPring-8 および SACLA の関係者のご協力をいただきました。この場を借りてお礼申し上げます。

#### 参考文献

- [1] R. Tanaka *et al.*: “Control System of the SPring-8 Storage Ring”, Proceedings of ICALEPCS’95, Chicago, (1995) 201.
- [2] T. Matsumoto *et al.*: “Next-Generation MADOCA The SPring-8 Control Framework”, Proceedings of ICALEPCS2013, San Francisco, California, USA, (2013) 944.
- [3] 清道明男 他: 「SPring-8 蓄積リング2次元放射光干渉計高度化に向けた MicroTCA 画像処理システムの開発」、第10回加速器学会年会プロシーディングス (2013) 242.
- [4] Y. Furukawa *et al.*: “MADOCA II Interface for LabVIEW”, Proceedings of ICALEPCS2013, San Francisco, California, USA, (2013) 410.

#### 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

#### 石井 美保 *ISHII Miho*

(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 : ishii@spring8.or.jp