

MADDOCA II データ収集フレームワークの開発 DEVELOPMENT OF MADDOCA II DATA COLLECTION FRAMEWORK

松本崇博[#], 古川行人, 濱田洋輔
Takahiro Matstumoto [#], Yukito Furukawa, Yousuke Hamada
Japan Synchrotron Radiation Research Institute (JASRI)

Abstract

MADDOCA (Message And Database Oriented Architecture) is the control framework has been applied to operation of SPring-8 accelerator control since 1997. To fulfill control requirements for current SPring-8 and future SPring-8-II, we are now developing next-generation of MADDOCA, called as MADDOCA II. Messaging and data logging part of MADDOCA II was already developed and implemented into SPring-8 after 2013. In this proceedings, we report on the development of data collection framework which was not implemented in MADDOCA II. In MADDOCA II data collection, we facilitate the signal registration scheme to start the data collection smoothly. We also designed to treat various data collection methods with the unified framework. For example, polling and event data collection were managed with the unified database table, and the framework can handle data collection with various data format such as waveform data as well as point data. In the framework, data collection with LabVIEW interface was also implemented. These developments were finished except for the signal registration scheme. We started to implement MADDOCA II data collection framework into SPring-8 in March 2016. Currently, MADDOCA II data collection is performed for 151 hosts with polling data collection in SPring-8, and we confirmed the stable operation.

1. はじめに

MADDOCA (Message And Database Oriented Architecture)は SPring-8 加速器制御のために SPring-8 において独自開発した制御フレームワークであり、テキストベースの抽象化したメッセージ(英語の SVOC 文型)を介した機器の分散制御、データベースによる記録・抽出が特徴である。[1] MADDOCA は 1997 年より SPring-8 の加速器制御を高い信頼性・安定性のもとに運用する上で必要不可欠であったが、長年の運用において多くの課題が明らかになってきた。これらの課題を解消すると共に将来計画である SPring-8-II への適応を考慮し、我々は現在、より柔軟で高度な制御機能をもつ次世代制御フレームワーク MADDOCA II の開発を進めている。[2][3][4][5]

MADDOCA II 制御フレームワークのソフトウェア構造を Figure 1 に示す。MADDOCA II では、MADDOCA との互換性を保ちつつ、ZeroMQ[6]及び MessagePack[7]を活用することでさまざまな機能拡張を行っている。メッセージに画像データ等の可変長データを付加して転送するなどメッセージ通信の柔軟性を向上させると共に Windows プラットフォームのサポートも対応した。また、NoSQL データベース[8][9]による高速データ蓄積も実装した。現在、メッセージング及びデータ蓄積に関する MADDOCA II 開発は終わり、2013 年以後に SPring-8 制御系に実装された。

本プロシーディングスでは MADDOCA II において未実装であったデータ収集フレームワークの開発について報告する。MADDOCA II データ収集は、Figure 1ではデバイスから信号データを抽出し Streamer に送信する Data Collector の部分で実装される。Streamer に送信され

た信号データは Writer を介して No SQL DB に蓄積されるが、Streamer 及び Writer に関しては既に開発済である。[5] 現在は、MADDOCA II データ収集が未整備であるため、MADDOCA 形式でデータ収集したデータを途中で抽出し、Figure 1 での Streamer にデータを送信することで暫定的な対応を行っている。今回の MADDOCA II データ収集フレームワーク開発により、制御における全ての要素を MADDOCA II に移行することができた。

本プロシーディングスでは、以後、2 章で SPring-8 におけるデータ収集の課題について述べる。3 章でそれら課題を解消するために開発した MADDOCA II データ収集フレームワークについて報告する。その後、4 章で MADDOCA II データ収集フレームワークの SPring-8 への導入状況について述べ、最後にまとめを行う。

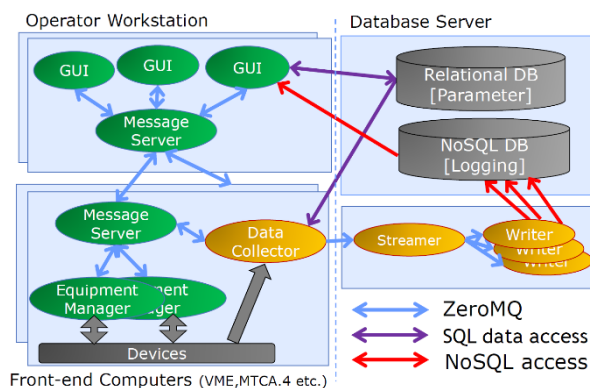


Figure 1: Software framework of MADDOCA II.

2. SPring-8 におけるデータ収集の課題

SPring-8 の加速器制御におけるデータ収集は約 500

[#] matumot@spring8.or.jp

のホスト及び約 28,000 点の信号を対象として行っている。各信号のデータ収集周期は 1 秒から 10 分毎までさまざまであるが、全体では約 9,000 点/秒のデータを蓄積している。本データ収集は点データ(整数型、浮動点少数型)を対象とした定周期データ収集を行う Poller/Collector [10]を基盤として運用してきた。しかしながら、長年の運用を通していくつかの課題が明らかになってきた。これらには、信号登録過程における労力、未整備であるデータ収集試験環境、一元化されていないデータ収集管理、未対応のデータ収集が挙げられる。以下に各課題について記す。

- 信号登録過程における労力
 - データ収集を行う信号は、データ収集管理、アラーム監視設定、信号履歴などの管理目的のため、Figure 1 に示される Parameter DB に事前に信号登録を行う。このため、DB 登録エディタ(エクセルファイル)に信号登録で設定する情報を記載する。DB 登録エディタでの設定項目は、ホスト名、機器グループ名、データ収集 ON/OFF 設定、間引き設定、アラーム設定等など多岐にわたる。機器担当者がこの DB 登録エディタを作成し、DB 担当者が信号登録を行うが、信号登録に多くの情報が含まれるため内容に不備があることが多い。このため、信号登録前に、機器担当者と DB 担当者の間でやりとりが繰り返され、信号登録手続き完了までに手間と時間がかかることがあった。
- 未整備であるデータ収集試験環境
 - 新しい機器構成でのデータ収集を開始する前には、事前にテスト環境でデータ収集を一定期間行い正常動作を確認しておくことが望まれる。しかしながら、現在は信号登録前にデータ収集試験を行う枠組みがない。このため、データ収集試験用途でも実際の制御系で用いるデータベースにデータを蓄積し、動作確認するしか手段がなかった。
- 一元化されていないデータ収集管理
 - SPring-8 では定周期データ収集の他にも、Linac においてビーム入射信号に同期した Event 型データ収集が行われている。[10] しかしながら、Event 型のデータ収集は後開発であり、定周期データ収集(Poller/Collector)と別のフレームワークで扱われている。他にはバンチ電流測定や COD データ収集等において配列データなど点データ以外の多様なデータ型を扱っているが、これらに対しては GUI 経由での独自仕様のデータ収集が行われている。このように SPring-8 では様々な種類のデータ収集が運用されており、それぞれ別の枠組みで取り扱われている。結果としてデータ収集が全体でどのように管理されているのか把握することが難しい状況にあった。
- 未対応のデータ収集
 - MADOCA のデータ収集は UNIX プラットフォームかつ C 言語のみ対応しているため

Windows OS、また C 言語以外の言語によるデータ収集には対応していない。現在、SPring-8 で LabVIEW を用いたデータ収集の需要はあるが、直接には対応できていない。このため、他の Solaris 計算機から LabVIEW のデータをソケット通信でアクセスし、データ収集する方法をとっている。しかしながらソケット通信を用いることで複数ホストを介したデータ収集構成となり、データ収集管理が煩雑になっていた。

3. MADOCA II データ収集フレームワーク

2 章で挙げたデータ収集における課題を解消するため、MADOCA II のデータ収集フレームワークは以下の点を考慮して設計した。

- 信号登録過程の改善
 - DB 登録エディタの不整合を事前にチェックし修正しておくことで、機器担当者と DB 担当者間の情報のやりとりを最小限にする。
 - テストデータ収集環境を提供する。
- データ収集管理の一元化
 - 定周期データ収集と Event 型データ収集を同一フレームワークで扱う。
 - 波形データなど多様なデータ型のデータ収集を汎用的な手法で扱う。
- 多様なデータ収集手段の提供
 - MADOCA II を活用することで UNIX の他、Windows などマルチプラットフォームにおけるデータ収集に対応する。
 - LabVIEW、Python などの C 言語以外のアプリケーションによるデータ収集にも対応する。

以下、MADOCA II データ収集フレームワークの実装について記す。本フレームワークは信号登録及びデータ収集から構成される。

3.1 信号登録

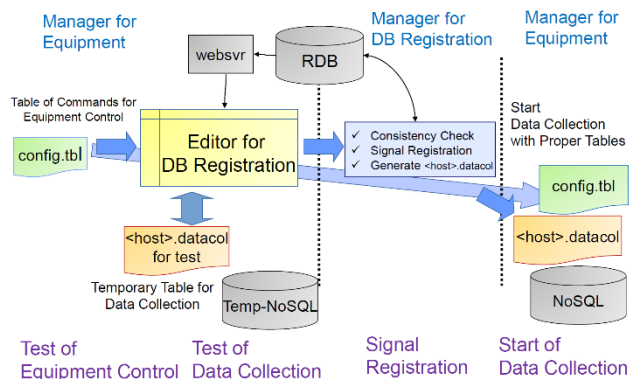


Figure 2: Procedure of signal registration in the preparation of MADOCA II data collection.

MADDOCA II データ収集における信号登録手順を Figure 2 に示す。機器担当者が、DB 登録エディタを介して信号情報を記載し、それを DB 登録担当者が DB に登録する。この手続きは以前と同様であるが、MADDOCA II では DB 登録エディタを Web インターフェースを用いて実装することで、Web インターフェース上で不整合をチェックし事前に修正できるようにした。また、データ収集テストの枠組みも用意した。これらにより DB 登録エディタ上の不整合を最小限にし、信号登録過程での手間・時間を軽減することができる。各手順について以下に記す。

- DB 登録エディタへの信号情報の記載
 - ▶ 機器担当者が DB 登録エディタを介して信号情報を記載する。現在 DB 登録エディタにはエクセルファイルを用いているが、将来的には Web インターフェース上に DB 登録エディタを実装できるように現在開発を進めている。DB 登録エディタを用いた編集作業にあたっては、始めに信号情報の初期設定を含むテンプレートファイルを Parameter DB 等から抽出することで作成し、この上に必要な情報を追記または更新する。
DB 登録内容は多くの項目が記載されるため、その内容には不整合が含まれていることが多い。このため、DB 登録エディタの Web インターフェース上に登録内容をチェックし不整合の箇所を可視化する機能を設ける。これを活用することで DB 登録エディタ上の不整合を事前に解消し、信号登録時の手間・時間を軽減することができる。DB 登録エディタ編集後は Web 上で登録ボタンを押すと DB 登録担当者に登録内容が通知される。
- データ収集の事前試験
 - ▶ データ収集の事前試験を信号登録前に行う。MADDOCA II データ収集プロセスでは、データ収集の対象となる信号リストをファイル(dataacol file)から入力する。正規には dataacol file は信号登録された Parameter DB から生成するが、データ収集試験では、DB 登録エディタから仮の dataacol file を生成し、これを入力に用いると共に、テンポラリの NoSQL DB にデータを蓄積する。これにより、通常データ収集とは独立にデータ収集試験を信号登録前に行い、動作確認することができる。
- 信号登録
 - ▶ DB 登録担当者が信号情報の DB 登録を機器担当者から通知された、DB 登録エディタの情報に基づき行う。DB 登録エディタをそのまま DB へ自動登録する運用も考えられる。しかしながら、信号登録には自動チェックだけでは予期されない登録内容が含まれるケースも考えられる(数万点の信号登録など)。加速器運転の信頼性を保証するため、事前に DB 登録担当者が信号登録内容の妥当性を確認する方式で運用を行う。

- データ収集開始
 - ▶ 上記の信号登録過程を経た後、正規のテーブル(config.tbl、dataacol file)を用いてデータ収集を開始する。

3.2 データ収集

MADDOCA II データ収集は、定周期及び Event 型データ収集の一元管理を行いデータ収集の管理性を向上させるとともに、多様なデータ収集手法に対応できるように実装した。以下、これらについて記す。

3.2.1 定周期及び Event 型データ収集の一元管理

Poller/Collector での定周期データ収集や Linac の Event 型データ収集では信号管理テーブルがそれぞれ別管理であったが、それらを統一的に管理できるようにしたデータ収集管理テーブル(DATACOL_INF)を導入した。DATACOL_INF 導入によりデータ収集管理が一元化され DB 担当者の利便性が向上したが、同時にユーザーの利便性も向上した。例えば、Web サーバを用いたデータ閲覧は、以前は Poller/Collector と Linac Event データ収集で別管理であり、相互のデータ関連の参照が困難であったが、DATACOL_INF 導入によりデータ閲覧ページを統一化することができ、定周期、Event 型データ収集間のデータ関連も簡易に閲覧できるようになった。

3.2.2 データ収集プロセス

データ収集プロセス Data Collector(DC)は EMDC、MSDC の2種類を実装している。これら DC の構成を Figure 3 及び Figure 4 に示す。EMDC では機器制御のアプリケーションである Equipment Manager(EM)が扱う Device 関数を介して機器から信号データを取得する。MSDC では Message Server 経由で各 EM から間接的に信号データを取得する。取得したデータは各 DC が扱う信号リスト毎にまとめて Streamer に送信される。通常データ収集では EMDC を用いるが、EMDC では所定の C 言語以外で書かれたアプリケーションには対応できない。LabVIEW、Python など他言語のアプリケーションを用いてデータ収集を行う場合は MSDC を用いる。この場合、MADDOCA II のインターフェースを用いて LabVIEW、Python 等に対応した EM を作成し、MSDC はこの EM を介して信号データを取得する。MSDC から EM への信号データ問い合わせは各信号毎に行うことができるが、複数信号を扱う際に高速に処理するため、EM に複数信号データを纏めて問い合わせることもできるように実装している。

各 DC では内部にデータ収集用の Thread を複数実装することができる。Thread ではデータ収集の割り込みが行われたタイミングでデータ収集が行われる。このデータ収集の割り込み時の Event 待ち関数にタイマーを用いる場合は定周期データ収集になり、その他の場合では Event 型データ収集として実装することができる。周期が異なる定周期データ収集については別 Thread で異なる周期のタイマーの Event 待ち関数を用いることで対応できる。これら複数の収集周期、Event 型のデータ収集の実装は一つの DC プロセスに複数 Thread を立ち上げて割り当てることもできるが、各 DC プロセスに 1 つの用

途のデータ収集を割り当て、複数の DC プロセスにデータ収集を分散させて運用することもできる。

Event 型データ収集に関しては、データ収集の割り込み指令を DC とは別プロセスで行うケースもある。例えば COD データ収集においては定周期データ収集の他、必要に応じて手動で GUI からの指令でデータ収集を行う場合もある。このように GUI 経路で行われてきたデータ収集を MADOCA II データ収集フレームワークに対応させるため、外部からの MADOCA II メッセージ経路による Event 型データ収集も行うことができるようにも実装した。

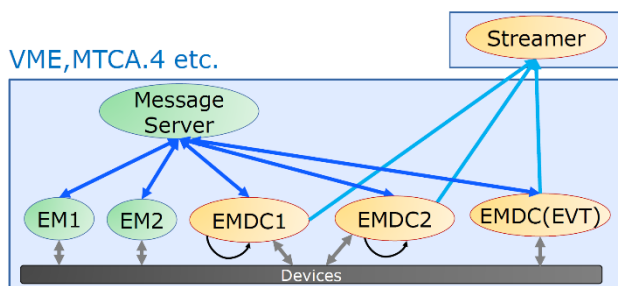


Figure 3: Configuration of MADOCA II data collection processes with EMDC. We can assign polling data collection (EMDC1, EMDC2) and event data collection(EMDC(EVT)) as the MADOCA II data collection process. Here, EMDC1 and EMDC2 have different data collection cycle.

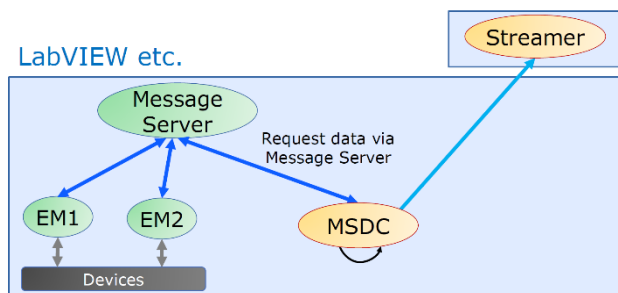


Figure 4: Configuration of MADOCA II data collection process with MSDC. MSDC can fetch data from an EM for each signal through a Message Server. MSDC can also fetch data for multiple signals by once from the EM to perform fast data collection with short cycle.

3.2.3 多様なデータ型でのデータ収集

点データの他、波形データなど多様なデータ型の信号データについても、同じ DC のフレームワークを使ってデータ収集できるようにするため、多様なデータ型の信号データに対しては MessagePack を用いてシリアライズしたデータを NoSQL DB にそのまま蓄積する形式をとった。MessagePack は自己記述型のシリアライザであり、シリアライズした文字列データには入力したデータ型の情報が記録されており、後でそのままのデータ型でデータを取り出すことができる。このように MessagePack を活用することで多様なデータ型でのデータ収集を汎用的に扱うことができる。

3.2.4 マルチプラットフォーム対応

MADOCA II データ収集では、データ収集プロセス DC はメッセージングを介してデータを一旦 Streamer に転送し、Streamer に転送されたデータが後に複数の Writer を介して No SQL DB にデータ蓄積される。MADOCA II では複数の NoSQL DB(Redis/Cassandra) にデータを蓄積するため、データ収集部で NoSQL DB に直接データを書かずに一旦 Streamer にデータ転送する形式をとっているが、この実装はデータ収集のマルチプラットフォーム対応においても有用である。

データベースに直接データの書き込みを行う場合、利用するデータベースライブラリのマルチプラットフォーム依存性の影響を受け、特定のプラットフォームでアプリケーションが利用できないケースも考えられる。DC によるデータ収集ではメッセージングを活用したデータ転送を行っているのみであり、この用途にはマルチプラットフォーム対応した ZeroMQ を利用している。よって SPring-8 制御で利用している Windows や組み込み計算機環境を含むマルチプラットフォーム環境に対して MADOCA II データ収集を適用させることができる。

4. SPring-8 制御系への MADOCA II データ収集フレームワーク導入

3 章で述べた MADOCA II データ収集フレームワークは事前試験を行った後、2016 年 3 月に SPring-8 制御系へ導入した。現在は SPring-8 蓄積リングの温湿度測定やビームラインでの PLC のステータスの情報など 151 のホストにおいて定周期データ収集を行っている。

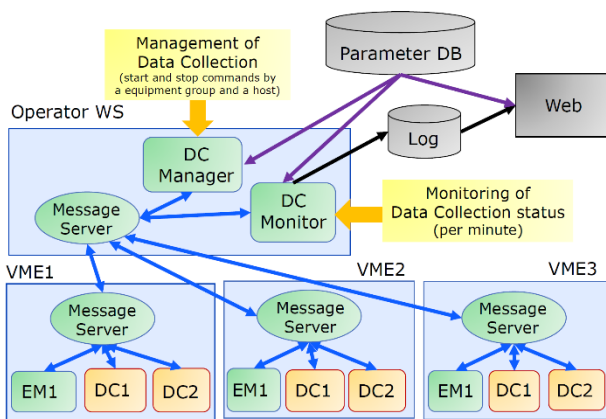


Figure 5: Management of operation for MADOCA II data collection.

Figure 5 に MADOCA II データ収集管理の構成を示す。MADOCA II データ収集は VME、Armadillo など複数の機器制御計算機で行っているが、データ収集起動、停止、状態監視などのデータ収集管理は MADOCA II メッセージング経路で特定の制御端末から統一的に行う。この制御端末には、データ収集の起動・管理を行う DC manager 及び、データ収集の状態監視を行う DC monitor が設置されている。DC manager では、複数のデータ収集プロセスを効率的に管理するため、機器グループまたはホスト毎にまとめてデータ収集の起動、停

止を行う。DC monitor ではデータ収集状態を監視するとともに、データ収集で稼働しているプロセス監視も行っている。これら監視情報は 1 分毎に収集されておりログに書き出され、Web サーバ経由で閲覧することができる。データ収集に異常があった場合には症状を Web 画面で把握することで、復帰対応を適切に行うことができる。

今後、9 月からは NewSUBARU 機器モニタで LabVIEW でのデータ収集運用を開始する予定である。LabVIEW を利用するため、Figure 4 に示した MSDC によるデータ収集を用い、データ収集用の EM を MADOCA II-LabVIEW インターフェース[4]を用いて製作することで対応した。本ケースでは 1 つの Windows ホスト上で毎秒 169 点の信号のデータ収集を行うため、多くのデータ収集信号点を扱っている。しかしながら、LabVIEW の EM で複数の信号データを一括で纏めて送信することで高速化が可能になり、本データ収集を実現できる見通しが立っている。

5. まとめ

本プロシーディングスでは MADOCA II で未対応であったデータ収集フレームワークの開発について報告した。MADOCA II データ収集フレームワークでは今まで課題であったデータ収集開始プロセス(信号登録、テストデータ収集)の改善、定周期及び Event 型データ収集管理の一元化、波形など多様なデータ型への対応、多様なデータ収集手法の対応ができるように設計し実装を行った。2016 年 3 月より MADOCA II データ収集を SPring-8 制御系に導入した。この MADOCA II データ収集導入により、初めて SPring-8 の全ての制御を MADOCA II で運用することができた。現在のところ、151 ホストでの定周

期データ収集のみの導入であるが安定運用されている 9 月には LabVIEW のデータ収集運用も開始する予定である。今後は MADOCA II データ収集開発で実装が完了していない信号登録部を完成させていく。また、SPring-8 のデータ収集の MADOCA II 化(Event 型データ収集、波形データ対応など)を進めていく予定である。

参考文献

- [1] R. Tanaka *et al.*, “The first operation of control system at the SPring-8 storage ring”, Proceedings of ICALEPCS1997, Beijing, China, 1997, p.1.
- [2] T. Matsumoto *et al.*, “Next-Generation MADOCA SPring-8 Control Framework”, Proceedings of ICALEPCS2013, San Francisco, California, USA, 2013, p.944.
- [3] T. Matsumoto *et al.*, “Multi-Host Message Routing in MADOCA II”, Proceedings of ICALEPCS2015, Melbourne, Australia, 2015.
- [4] T. Matsumoto *et al.*, “LabVIEW Interface for MADOCA II with Key-Value Stores in Messages”, ICALEPCS2015, Melbourne, Australia, 2015.
- [5] A. Yamashita *et al.*, “MADOCA II data logging system using NoSQL, database for SPring-8”, Proceedings of ICALEPCS2015, Melbourne, Australia, 2015.
- [6] <http://zeromq.org/>
- [7] <http://msgpack.org/>
- [8] <http://redis.io/>
- [9] <http://cassandra.apache.org/>
- [10] A. Taketani *et al.*, “Data Acquisition System with Database at the Storage Ring”, Proceedings of ICALEPCS’97, Beijing, China, 1997, p.585.
- [11] T. Masuda *et al.*, “Event-synchronized data acquisition system for beam position monitors of SPring-8 linac”, Nucl. Instrum. Methods A 543, 2005 p.415.