

動的制御機能を持つデータ収集システムフレームワークの提案 Proposal for a Data Acquisition System Framework with Dynamic Control Functions

堀 裕貴[†] 長坂 康史[†] 千代 浩司[‡] 濱田 英太郎[‡] 山木戸 啓亮[†] 西村 俊彦[†]
Hiroki Hori Yasushi Nagasaka Hiroshi Sendai Eitaro Hamada Keisuke Yamakido Toshihiko Nishimura

1. はじめに

近年のパソコンやスマートフォン等の普及により、インターネット人口が増加し、ネットワーク技術も大きく発展してきた。一方、物理学実験で用いられるデータ収集(DAQ)システムにおいても、それら発展した技術を用いた収集システムの構築が可能となり、大量なデータを高速に収集・保存することができるようになった。しかし、大規模実験の場合、それぞれの実験用にシステムの開発を行うことが多く、システムの再利用性も低いため、開発者の負担が大きくなっていった。

そこで本研究では、この開発時の負担を軽減するために、オブジェクトベースのデータ収集システムフレームワーク DAQ-Middleware を他研究機関と共同開発している。このフレームワークは、実際にデータのやりとりを行う部分とシステム全体を制御する部分で構成される。システム全体の構成情報を XML ファイルに記述することで、システム構成を容易に変更できるようにしている。また、起動時には外部から与えたい制御パラメータを XML ファイルに記述することで、システム内部で利用することができる。

DAQ-Middleware はデータ収集システムのフレームワークとしては完成したが、現フレームワークでは、システムの起動中に制御パラメータを変更する機能が備わっていない。

そこで、本研究では、DAQ-Middleware で用いる制御パラメータを動的に変更する機能の提案及び開発を行い、動的制御が可能なデータ収集システムフレームワークとすることを目的とする。

2. DAQ-Middleware

2.1 概要

DAQ-Middleware (DAQ-MW) とは、ネットワーク分散環境下におけるデータ収集ソフトウェアの構築を支援するソフトウェアフレームワークである^{[1][4]}。DAQ-MW は、高エネルギー加速器研究機構(KEK)、OpenRTM-aist の開発を行う産業技術総合研究所などと広島工業大学との共同研究で開発されている。

このフレームワークはロボット技術要素である RT-Middleware (RTM) をベースにしておき、Component と呼ばれるデータを収集する機能やデータを保存する機能などを持った要素をモジュール化したものを用いる。DAQ-MW では、複数の様々な種類の Component を組み合わせると一つのシステムを構成することで、独自のデータ収集システムを容易に構築することができる。実際にデー

タのやりとりを行う Component 間の通信は、RTM が提供するオブジェクト通信を用いる。DAQ-MW が提供する Component には 2 種類、実際にデータをやりとりする DAQ-Component と、その DAQ-Component に対して状態遷移命令や各種パラメータの送信などのシステム全体の制御を行う DAQ-Operator がある。

DAQ-MW のシステム構成は XML で記述される。利用する Component や Component 間の接続などのシステムパラメータや、Component 内で利用する制御パラメータの設定などの情報を記述する。DAQ-Operator が、この XML ファイルを読み込むことでシステム全体の制御を行う。

図 1 に DAQ-MW を用いたシステムの構成例を示す。図 1 の PC(UI) と PC(DAQ#1) はそれぞれ、ユーザインターフェイス用計算機とデータ収集用計算機を示している。データ収集用計算機では DAQ-Component が実行されている。Gatherer でデータを収集し、Dispatcher でデータを 2 つに分ける。分けられたデータは Logger で記録され、Monitor で表示される。ユーザは、DAQ-Operator を用いて、システム全体の制御を行う。その際、DAQ-Operator は、XML で記述されたファイルから、システムの構成情報や制御パラメータなどの情報を読み込む。そして、DAQ-Component に各情報と制御コマンドを送信することでシステム全体の操作が可能になる。

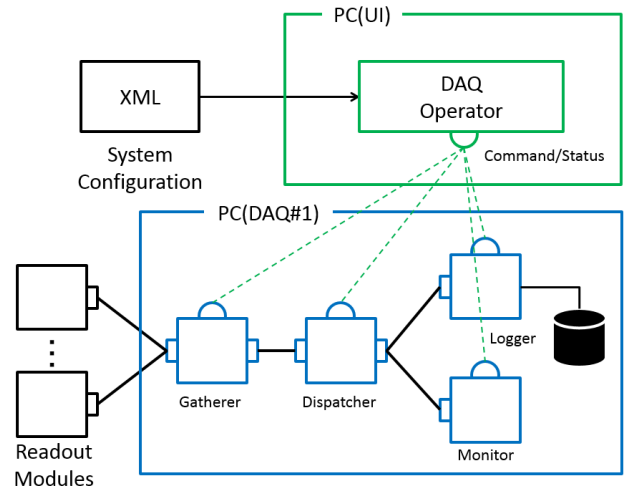


図 1 DAQ-Middleware を用いたシステム構成例

2.2 状態遷移

DAQ-MW は状態遷移を行うことでシステムをコントロールする。図 2 に DAQ-MW の State 遷移図を示す。

DAQ-MW では 4 つの異なる State を提供する。システム起動直後に遷移する「Loaded」は、システムの構成に必要な Component の起動を行った直後の状態である。

「Configured」は、DAQ-Operator が XML ファイルから読み込んだ各種パラメータの情報を各 Component に設定した後の状態である。「Running」では、各 Component が活

[†] 広島工業大学 Hiroshima Institute of Technology

[‡] 高エネルギー加速器研究機構 High Energy Accelerator Research Organization

動状態となり、実際にデータ収集が行われる。システム一時停止状態の「Paused」では、Running で行われているデータ収集処理を一時中断する。

それぞれの状態間の遷移では、それぞれの State 間の遷移に対応した関数が呼ばれる。これらの関数は、状態遷移の処理やユーザ定義の処理を一度だけ行い、その後遷移先の状態へ遷移する。

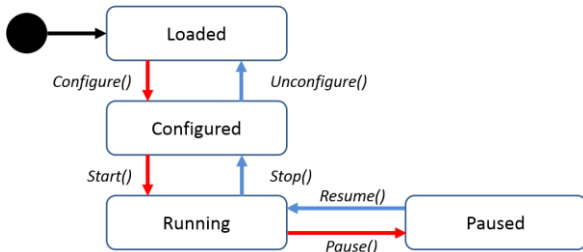


図 2 State 遷移図

3. 提案手法

3.1 概要

DAQ-MW では、システムの構成を XML で記述し、システム起動時に読み込むことでシステムを動作させる。XML で記述する項目には、システム内部で利用する制御パラメータを設定する部分があり、そのパラメータデータを DAQ-Operator から DAQ-Component へ送信することで利用する。しかし、現在の DAQ-MW では、システム動作中に制御パラメータを変更する機能が備わっていない。

そこで本研究では、制御パラメータを動的に変更できる機能の実装を提案する。

3.2 システム詳細

DAQ-MW 内に、新たな State として Changed State の追加を提案する。新たな State 遷移図を図 3 に示す。Changed State は、パラメータ設定状態の Configured と、一時停止状態の Paused から遷移する。Changed State では、DAQ-Operator で新たに XML ファイルを読み込み、その中に記述されている制御パラメータを DAQ-Component へ送信する。システムを停止させることなくパラメータを変更することで、動的制御を実現し、切り替え時の遅延を抑えることで効率的な運用を行う。

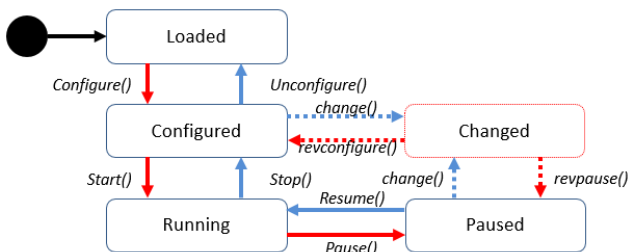


図 3 Changed State を追加した State 遷移図

4. 性能評価

動的制御機能の実装による DAQ-MW の性能を評価するために、設定ファイルを選択してから Component への制御パラメータの送信が完了するまでの実行時間を測定した。測定はすべて Local PC 上に構築した Component を利

用して実施した。測定は n 対 1 (n = 1, 3, 5, 7, 9) で行う。このとき、システムの総 Component 数は 2, 4, 6, 8, 10 となる。また、測定時に用いる制御パラメータの数を 1, 5, 10, 50, 100, 500 とした。送受信させるデータサイズはすべての測定で 4096 バイトとした。それぞれの処理時間は、100 回の測定結果を平均したものである。

測定結果を図 4 に示す。この図は、システムで用いる制御パラメータ数の違いによる処理時間の変化を表している。この図から、制御対象となる Component 数が増加し、変更するパラメータ数が大きくなるほど処理時間は大きくなり、例えば 1 対 1 の構成でパラメータ数を 1000 個にする場合、処理時間は約 0.3 秒となることが予測される。

図より、例えば 3 対 1 のシステムの場合の制御パラメータ数が 500 の場合の処理時間が約 0.11 秒であることから、本システムにおける制御パラメータの変更に必要な時間は十分短く、データ収集効率には影響がないと考えられる。

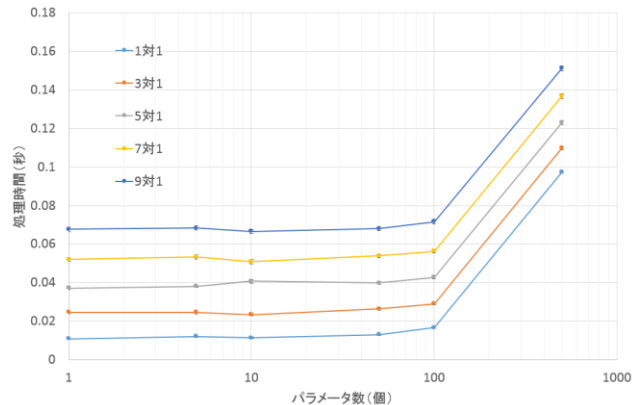


図 4 測定結果

5. おわりに

本研究では、より効率的なデータ収集システムの開発を目的として、データ収集フレームワーク DAQ-Middleware に、制御パラメータを動的に変更する機能を提案、実装した。システム構成を記述する XML ファイルを複数個用意し、新たに追加した Changed State で用いる各種パラメータが記述された XML ファイルを選択することで、制御パラメータを変更することができるようになった。

今後は、やりとりするデータ量の違いを考慮した性能評価や、動的制御に関わる WebUI の開発、実際の実験へ導入した場合の評価を行う。

参考文献

- [1] Y. Yasu, K. Nakayoshi, E. Inoue, H. Sendai, H. Fujii, N. Ando, T. Kotoku, et al., "A Data Acquisition Middleware", in Proc. IEEE/NPSS Real Time Conference, pp. 1-3, May 2007.
- [2] Y. Yasu, K. Nakayoshi, H. Sendai, and E. Inoue, "Functionality of DAQ-Middleware", IEEE Trans. Nucl. Sci., vol. 57, no. 2, pp. 487-490, Apr. 2010.
- [3] H. Sendai, K. Nakayoshi, Y. Yasu and E. Inoue, "Performance measurement of DAQ-Middleware," J. Phys.: Conf. Ser. 331 022039, 2011
- [4] H. Maeda, Y. Nagasaka, H. Sendai, E. Inoue, E. Hamada, T. Kotoku, N. Ando, S. Ajimura, M. Wada, "Control functionality of DAQ-Middleware", J. Phys.: Conf. Ser. 513 012020, 2014