DAQ-Middleware: Progress and status

K Nakayoshi¹, H Sendai¹, Y Yasu¹, E Inoue¹, T Kotoku², N Ando², Y Nagasaka³, S Ajimura⁴ and M Wada⁵

¹ High Energy Accelerator Research Organization (KEK), 1-1 Oho, Tsukuba, Ibaraki 305-0801, Japan
² The National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan
³ Hiroshima Institute of Technology, 2-1-1 Miyake, Saeki-ku, Hiroshima 731-5193 Japan
⁴ Osaka University, 1-1 Machikaneyama, Toyonaka 560-0043, Osaka, Japan
⁵ Bee Beans Technologies Co., Ltd., Sengen 2-1-6, Tsukuba, Ibaraki 305-0047, Japan

E-mail: kazuo.nakayoshi@kek.jp

Abstract. We report on the progress and status of DAQ-Middleware, a software framework for a distributed data acquisition system. We made improvements in DAQ-Middleware and released package (version 1.0.0) in August 2010. We describe here its improvements in performance and the component development method we used. We also report on the current status of DAQ-Middleware for use at the Material and Life Science Experimental Facility (MLF) of the Japan Proton Accelerator Research Complex (J-PARC).

1. Introduction

DAQ-Middleware is a software framework for a distributed data acquisition (DAQ) system[1]. It is based on OpenRTM-aist (C++), an implementation of Robot Technology Middleware (RTM)[2] developed by the National Institute of Advanced Industrial Science and Technology (AIST). The DAQ-Components in DAQ-Middleware are software units that can run fully independent of each other. Moreover, users can select from various DAQ-Components, mixing and matching them to create their own DAQ system. Inheriting DaqComponentBase class, users can easily develop new DAQ-Components. By using DAQ-Middleware users can gain the flexibility and the software reusability to their DAQ systems. Figure 1 shows a fundamental DAQ model using DAQ-Middleware. On a readout PC, four DAQ-Components, called ”DAQ unit”, are running for data reading, storing and monitoring. On the user interface PC, a DAQ-Operator controls DAQ-Components received from users commands via Web.

To summarize the history of DAQ-Middleware development, we started to develop DAQ-Middleware under a next-generation DAQ project as part of the High Energy Accelerator Research Organization (KEK) Detector Technology Project (DTP) in 2006. In 2008 we progressed from the prototype to an application package for the Material and Life Science Facility (MLF) at the Japan Proton Accelerator Research Complex (J-PARC), followed by successful first beam commissioning[3, 4]. In April 2010, under an Open Source Consortium of Instrumentation called ”Open-it”, a new development team was formed, consisting of participants from the KEK, AIST, Osaka University, Hiroshima Institute of Technology and Bee Beans Technologies Co., Ltd. In August 2010, a new DAQ-Middleware package (version 1.0.0) was released. In section 2,
we describe several improvements in DAQ-Middleware, including (1) data throughput between DAQ-Components, (2) the component development method and (3) the configuration of the release package. In section 3, we report the current status of MLF neutron experiments using DAQ-Middleware. In section 4, we present ongoing projects.

2. Improvements in DAQ-Middleware 1.0.0

DAQ-Middleware 1.0.0 has enabled the performance of a wider range of applications than previously released packages, which had been customized for MLF experiments.

2.1. Performance

We have measured the performance of DAQ-Middleware 1.0.0 with several conditions[7]. Figure 2 shows one of the conditions. In this setup, two PCs, one running a source component and the other running a sink component, are connected to a network. The source generates data, transferring it to the sink via Ethernet. The purpose of this was to measure data throughput between two DAQ-Components via Ethernet, the condition that provides the basis for more complicated connections among DAQ-Components. Figure 3 shows the results of data

---

Figure 1. A fundamental DAQ model using DAQ-Components.

Figure 2. One of setup for the performance measurements. A source component and a sink component are running on two separate PCs. The source generates data, transferring it to the sink via Ethernet.

Figure 3. Data throughput between two components via Ethernet. The x axis shows the size of transferred data in kByte and the y axis shows the throughput in MByte/s. The closed circles represent version 1.0.0, the closed triangles represent the previous one.
throughput between two components. The x axis shows the byte size of transferred data, and the y axis shows the data throughput in MByte/s. The closed circles represent the performance of DAQ-Middleware 1.0.0, whereas the closed triangles represent the previous version. Maximum data throughput was about 84MBytes/sec using version 1.0.0, almost two times higher than the previous one.

2.2. Components development method
In DAQ-Middleware, users can easily develop new DAQ-Components and create their own DAQ system using them. Users especially need to develop reader/gatherer components for their read-out devices and monitor components for their online analysis. Because the read-out module for each experiment differs, so do the data format and method of analysis. We categorized reader/gatherer components as source type and such monitor components as sink type[1]. The source and sink types are key components for users. In DAQ-Middleware 1.0.0, we created a script, called newcomp, to generate template codes of source or sink type, thus enabling users to develop their components more easily. Shown below is an example of how newcomp script is used. In the below, four files of source type DAQ-Component named MySource are generated.

$ newcomp -t source MySource
$ ls -F
MySource/
$ ls MySource/
Makefile MySourceComp.cpp MySource.cpp MySource.h

2.3. Configuration of the release package
We also changed the configuration of the release package for DAQ-Middleware. Before version 1.0.0, released packages were customized to individual MLF experiments. In version 1.0.0, the DAQ-Components for MLF are separated from the release package. Figure 4 shows configuration of the release package for version 1.0.0 and previous one. We organized the release package and

![Figure 4. Configuration of the release package of DAQ-Middleware 1.0.0 and previous one.](image)

decided core part of DAQ-Middleware as follows.

- Base class of DAQ-Components (DaqComponentBase class)
- DaqOperator class
- XML Schema of configuration files
- System interface protocol (XML/HTTP)
- C++ class library wrapping the Berkeley sockets C API
In version 1.0.0, users develop their own DAQ systems by combining the release package, which is the core of DAQ-Middleware, with individually chosen DAQ-Components.


DAQ-Middleware has been in use at MLF of J-PARC, since 2008. Figure 5 shows the DAQ-component lineup for MLF. Three types of detectors are utilized in MLF neutron experiments: position sensitive detectors (PSDs), one-dimensional/two-dimensional photon counting detectors and neutron beam monitors with a gas electron multiplier (GEM)[5]. The readout modules of these detectors are equipped with SiTCPs [6]. The digitized data can be retrieved from the readout modules by gatherer components on the PCs via Ethernet (100Base-TX). Each detector system has its own gatherer and monitor components. The DAQ-Operator, dispatcher and logger are common to all detector systems. Figure 6 shows DAQ units of each detector system. In each experiment, users choose and combine DAQ-Components for MLF. At present, eight neutron instruments at MLF are using DAQ-Middleware on their DAQ systems and six are preparing to do so.

Figure 5. DAQ-Component line-up for MLF. Three types of gatherer and monitor components were developed respectively.

Figure 6. Basic DAQ units of each detector system. The DAQ-Operator, dispatcher and logger are common to all detector systems.

3.1. Examples of DAQ systems using DAQ-Middleware at MLF

We describe here two examples of DAQ systems using DAQ-Components at MLF. The first is NOVA, which uses PSD and GEM detectors, and the second is iBIX, which uses two-dimensional photon-counting detectors. NOVA, a total scattering spectrometer, is constructed from five banks of PSD detectors and two GEMs as beam monitors. Figure 7 shows the DAQ system of NOVA using PSD DAQ units. On each of the five CPU DAQs, a PSD DAQ unit reads the data from about one hundred read-out modules equipped with SiTCP via Ethernet.

iBIX, the IBARAKI biological crystal diffractometer, is constructed from a two-dimensional photon-counting detector system using scintillator sheets and wavelength-shifting fiber arrays[8, 9]. Figure 8 shows the DAQ system of iBIX using Scinti DAQ units. On each of the four CPU DAQs in the Figure 8, a Scinti DAQ unit reads out data from 14 read-out modules equipped with SiTCP via Ethernet.
4. Ongoing projects
The objective during this fiscal year is to establish a basis for small DAQ systems using DAQ-Middleware. In these systems, we will have to increase usability and convenience. That is, we have to provide an easy-to-use graphical user interface framework such as GUI panels for run control and condition parameters of detectors. Using DAQ-Middleware 1.0.0 we are developing prototypes of small DAQ systems. In one, we are working with the International Linear Collider (ILC) CCD vertex detector R&D group in Japan, we are developing a readout system for a prototype of fine pixel CCD vertex detector[10]. The second is a DAQ system for depth-resolved XMCD (X-ray Magnetic Circular Dichroism) experiments[11], to be performed at the Photon Factory at KEK.

5. Summary
In April 2010, the new development team of DAQ-Middleware was formed. We released DAQ-Middleware 1.0.0 in August 2010. It has enabled a wider range of applications. In this version, data throughput and the DAQ-Component development method were improved. As an example, data throughput between two components via Ethernet is about 84MBytes/sec using version 1.0.0, about 45MBytes/sec using the previous version. Eight neutron instruments at MLF are using DAQ-Middleware on their DAQ systems and six are preparing to do so.

Acknowledgments
The authors wish to thank the staff at the MLF neutron facility: T. Otomo and H. Ohshita of KEK, T. Hosoya of Ibaraki University, and T. Nakatani and T. Ito of the J-PARC Center. We also wish to thank people in Open-It, M. Tanaka, T. Uchida, M. Ikeno and KEK electronics system group for their help. This work was performed in the framework of Open-It.

Figure 7. NOVA data acquisition system using DAQ-Middleware.
Figure 8. iBIX data acquisition system using DAQ-Middleware.

References