ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THMPL07

DARUMA: DATA COLLECTION AND CONTROL FRAMEWORK FOR X-RAY EXPERIMENTAL STATIONS USING MADOCA

T. Matsumoto†, T. Abe, Y. Furukawa, T. Matsushita, K.Nakada, H. Masunaga,
Japan Synchrotron Radiation Research Institute, Hyogo, 679-5198, Japan

Abstract

To take measurements in X-ray experimental stations at the SPring-8 synchrotron radiation facility, station staffs and users occasionally need to reconfigure the system for new experiments. At such times, quick reconfiguration of the system is required, which involves extensive work. Additionally, there is a strong need to reuse basic software associated with the measurement applications, which can be difficult. To overcome these challenges, we propose DARUMA, a data collection and control framework for stations. DARUMA utilizes the control framework MADOCA II [1] that was developed for distributed control of accelerators and beamlines at SPring-8. It provides software functionalities for stations, such as data collection and image handling. As it has the flexibility of MADOCA II and shares general software applications, DARUMA can help to reduce management costs and improve the measurement system. As a first attempt, we developed DARUMA for BL03XU at SPring-8. At this station, the migration into the DARUMA system is proceeding smoothly. As a result, we have begun applying DARUMA at other stations. Some applications in DARUMA, such as image handling, are particularly useful and have been implemented into a partial set of control systems for several stations since this September.

INTRODUCTION

SPring-8 is a third-generation synchrotron radiation facility in Japan. Brilliant synchrotron radiation X-rays up to an order of 100 keV are produced from 8 GeV electron beams and utilized for various experimental measurements of scientific research and industrial applications. For the measurements, stations are equipped in each of 56 beamlines.

To extract valuable results from the experimental measurements flexibly, it is important to have a user-friendly measurement system for station staff and users. However, the current measurement system has several limitations, and there are several demands from staff and users as listed below.

- Easy reconfiguration of the measurement system to update experimental setup
- Rapid preparation of the measurement software
- Easy to plug in basic software such as image handling
- Cooperated controls among measurement applications
- Reuse of software applications

The current measurement software's structure, which makes it difficult to provide the aforementioned facilities, is shown in Figure 1.

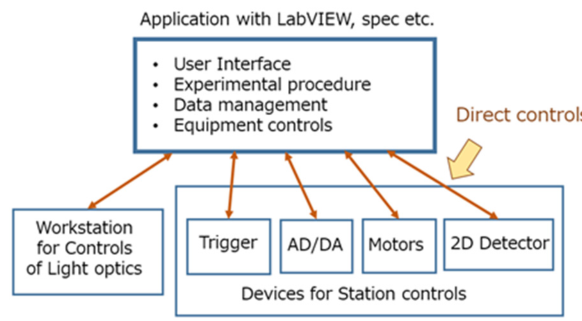


Figure 1: Schematic view of the structure of the typical measurement software for experimental stations.

Measurement applications in the stations are usually built with user interfaces such as LabVIEW [2], spec [3] and Visual Basic. As shown in Figure 1, one application contains all the functions of the measurements for the user interface, experimental procedures, and data management and equipment controls. Owing to the monolithic structure of the application, it is not easy to reconfigure the measurement system, because we need to update the application for the relevant sections by considering software dependencies in other sections as well. This constraint also prevents reuse of the application as general software, because many functions are combined in one application. Furthermore, the coding strategies of applications vary for each depending on the person who creates the application. Therefore, it takes time and resources to maintain the measurement system.

To solve these problems, we propose to apply a Data collection And control framework for X-Ray experimental station Using MADOCA, called DARUMA. Here, MADOCA stands for Message And Database Oriented Control Architecture and is used for distributed control of the accelerator and beamline for SPring-8. In this paper, we define the DARUMA software framework and show how DARUMA is useful for experimental measurements at stations. We also report the status of the implementation of DARUMA in BL03XU and other stations at SPring-8.

† matumot@spring8.or.jp

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THMPL07

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

DARUMA SOFTWARE FRAMEWORK

DARUMA was developed to fulfil the demands reported in the previous section. Policies of DARUMA are listed below.

- MADOCA is adopted for distributed controls of equipment for stations.
- General software tools are prepared for stations such as data collections and image handling.

Since MADOCA was implemented for controls of accelerator and beamline at SPring-8, we have achieved a high level of reliability and stability in controls for stations. We also have managed to implement cooperative controls with accelerator and beamline with MADOCA. We chose recently developed MADOCA II for the messaging controls due to several new functions being present in the software. New functions include Windows support, an interface with LabVIEW [4] and Python, and messaging controls with variable length data such as image data, which are all useful for the measurements at the stations.

In this section, we outline the basic concept of DARUMA and its benefits to the measurements for stations. Detailed aspects of DARUMA in the implementation will be reported in sections after.

DARUMA for Measurements in Stations

As described, the typical measurement application for stations covers all the functions and devices for the measurements and is directly controlled from the application. As such, we needed to setup several measurement applications for each computer where corresponding devices were equipped. Therefore, cooperative measurements among applications were not easy.

However, with the introduction of DARUMA, the software structure of the application is changed as shown in Figure 2. In DARUMA, we separated each function in the application into different software processes and developed the software components for each. The software components are composed from Equipment Manager (EM) to control the equipment and includes tools for data collection and image handling. For the measurements, we can use these software tools by connecting with the messaging controls of MADOCA. With the separated software structure, we can now unify the measurement application on one computer and cooperative controls among equipment in different computers can be easily achieved. We can also quite easily control light optics in beamline from the station with MADOCA (In Figure 1, we used intermediate socket process to control light optics in beamline with the MADOCA control system from the station).

By adopting DARUMA, with its separated software processes in the measurements, we can accomplish multiple goals as listed below.

- Time and cost to prepare the measurement application can be reduced because general software applications such as EM can be used as-is which allows us to concentrate on the user interface and experimental procedures.
- Reuse of software applications is realized.
- It is easy to plugin useful basic software.

To promote DARUMA for measurements at stations, it is important to enrich software tools in DARUMA. Table 1 presents the example of equipment list developed for EM. We developed these EM for BL03XU and other stations. We also have an additional EM developed for BL14B2 but these are not listed in Table 1. The developed EM may still not be enough for the stations. However, there are MADOCA II interfaces for LabVIEW and Python, as well as C languages, to build EM and we expect a supported number of EM can be flexibly increased based on demand. We also developed general software tools for data collection and image handling in DARUMA. Details of these tools will be reported in the next section.

Although MADOCA is adopted for the message controls of measurements, the applications for experimental users may be not prepared with MADOCA. If users want to have cooperative controls with DARUMA, we need to ask users to edit the application to have control with MADOCA. As several languages such as Python, LabVIEW, and C are supported for the interface, updating the application can proceed smoothly.

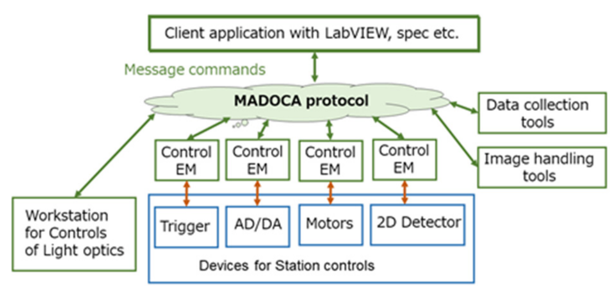


Figure 2: Schematic view of measurement software structure for a station with DARUMA. In DARUMA, we setup separated software processes for each function and these can be connected with messaging controls of MADOCA.

Table 1: An Example of Equipment List Developed for EM

Application	Equipment	Remarks
ADC	NI PXIe-4492	C interface with NI-DAQmx
Trigger (TTL I/O)	NI PXIe-6612	
Motor	TSUJICON PMC16C-16	Simultaneous drive (16 ch.)

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THMPL07

2D detector	PILATUS
	HiPic CCD
	HiPic FPD
	PerkinElmer XRD
	Rigaku HyPix

Benefits with MADOCA Messaging Controls

Here, we discuss the benefits with MADOCA messaging controls on reconfiguration of experimental setup. In MADOCA, a message for the control is composed of a text message of subject/verb/object/complement (S/V/O/C) syntax. In this context, "S" denotes the program that is automatically defined by the framework and "V" denotes the action of the command, for which "put" or "get" is primarily used. "O" is an object name, which identifies the target of the messages. "C" is an action parameter. For example, we can send a message with "put/bl_03in_st1_detector_1/start" as V/O/C (S is abbreviated as described) to start the measurement for the detector at BL03XU. In this example, the message is sent to an EM where "O" is registered. A response is then returned from the EM with S/V/O/C format where C is set to "ok" if the command succeeded. In the messaging scheme, message commands do not contain any specific device information and the system is designed to have abstracted message commands for easy-to-understand controls by a human. As such abstraction of the message commands are performed inside EM, we can use the same client application when equipment is updated if the corresponding EM is prepared to respond with the same massage commands as before.

Owing to the policy of abstracted message commands of MADOCA, reconfiguration of equipment to update the experimental setup can be flexibly managed which we expect to reduce the maintenance costs for control systems in the stations.

IMPLEMENTATION OF DARUMA INTO BL03XU

As a first attempt, we implemented DARUMA into BL03XU used for advanced soft material beamline. The schematic view of the measurement system with DARUMA for BL03XU is shown in Figure 3. For the measurements, we control monitoring of the current in the ion chamber and the voltage in the photo diode, motors, timing signals and 2D detectors for PILATUS, HiPic CCD and HiPic FFD. About 10 computers are used in the measurement system. To implement DARUMA, we developed an EM for each equipment device as described in the previous section.

For the measurements, we also developed measurement procedures with messaging commands of MADOCA. The procedures include initialization procedure such as location of the stored data, settings for camera and trigger, and operation procedure of the measurements. Though we prepared the procedures for BL03XU, these procedures can be generalized into measurement systems at other stations. Therefore, the developed measurement procedures can be flexibly extended to other stations.

To perform one measurement, we set the number of images of 2D detectors to a value of a few thousand for an example. The typical data collection rate is about 10 Hz in the case for PILATUS with M pixel data in the image. Although we often have several detectors in the measurements, data collections in these detectors are performed by synchronized timing with trigger signals. Afterwards, the measured data is stored on a server as a file. The data format for the image is not unified, but TIFF is typically used.

To analyze the measured data, we needed to search data from files in the storage server. However, it takes time and resources even though we included meta-information of the measurements in the folder name. To improve the flexibility in the analysis, DARUMA stored metadata of the measurements into a NoSQL database, elasticsearch [5]. Elasticsearch has a search engine based on Lucene with a REST API interface. As data can be managed with schema free JSON documents in elasticsearch, it is useful in managing the various metadata in experimental measurements.

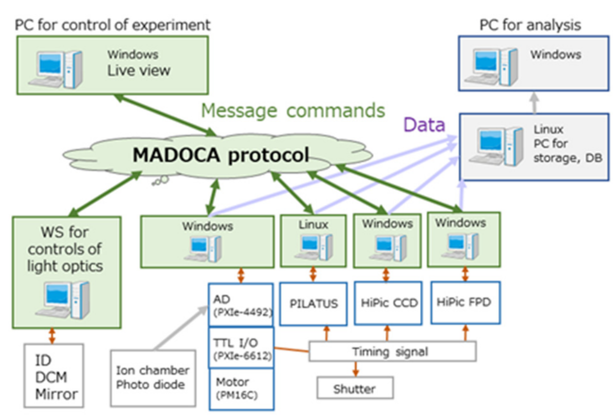


Figure 3: Schematic view of measurement system with DARUMA for BL03XU.

To perform data logging with elasticsearch, we developed data collection for DARUMA as shown in Figure 4. Here, we developed EM-DB, which adds functions into EM for data collection and data logging. The operation of EM-DB is performed from a client application with messaging commands of MADOCA. We have 2 types of EM-DB. The first EM-DB for master manages run information on the measurements, and the second EM-DB for

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

16th Int. Conf. on Accelerator and Large Experimental Control Systems ICALEPCS2017, Barcelona, Spain JACoW Publishing
ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THMPL07

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

2D detectors manages measurements with each 2D detector.

The collected data are sent to elasticsearch via a REST API. A type in elasticsearch, is prepared to archive data received from each EM-DB. The type for master manages run information and the type for 2D detectors manage information for detectors. For the image data, only the image file path is recorded in the type because the data size of the image is too large for elasticsearch. Though we record one event into different types, we can have synchronized data among these because EM-DB for 2D detectors fetches the run information from the data in elasticsearch recorded by EM-master and utilizes the information for the measurements.

To search measured data from elasticsearch, we developed a web interface as shown in Figure 5. In the web portal, it is possible to flexibly search data with various conditions in time, detector, and so forth. Full-text search is also possible by using functions in elasticsearch. The selected image can be viewed with analysis for Region Of Interest (ROI) then downloaded. Thus, we can have flexibility in the analysis with DARUMA.

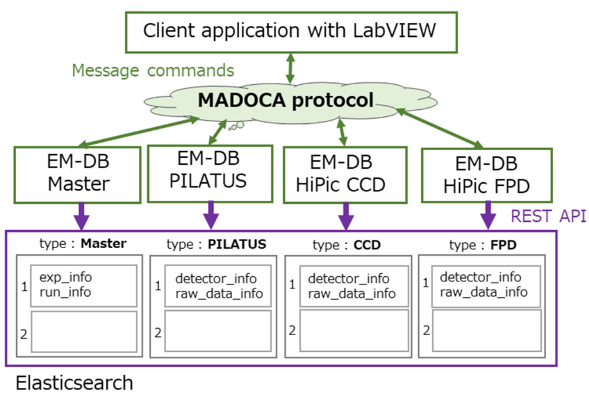
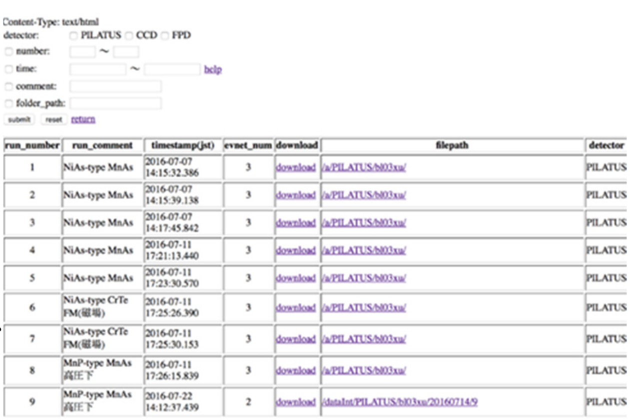


Figure 4: Software structure of data collection with DARUMA.


Content-Type: text/html
detector: PILATUS CCD FPD
number: ~
time: ~ help
comment:
folder_path:
submit reset return

run_number	run_comment	timestamp(jst)	evnet_num	download	filepath	detector
1	NiAs-type MnAs	2016-07-07 14:15:32.386	3	download	/a/PILATUS/bl03xu/	PILATUS
2	NiAs-type MnAs	2016-07-07 14:15:39.138	3	download	/a/PILATUS/bl03xu/	PILATUS
3	NiAs-type MnAs	2016-07-07 14:17:45.842	3	download	/a/PILATUS/bl03xu/	PILATUS
4	NiAs-type MnAs	2016-07-11 17:21:13.440	3	download	/a/PILATUS/bl03xu/	PILATUS
5	NiAs-type MnAs	2016-07-11 17:23:30.570	3	download	/a/PILATUS/bl03xu/	PILATUS
6	NiAs-type CrTe FM(磁場)	2016-07-11 17:25:26.390	3	download	/a/PILATUS/bl03xu/	PILATUS
7	NiAs-type CrTe FM(磁場)	2016-07-11 17:25:30.153	3	download	/a/PILATUS/bl03xu/	PILATUS
8	MnP-type MnAs 高圧下	2016-07-11 17:26:15.839	3	download	/a/PILATUS/bl03xu/	PILATUS
9	MnP-type MnAs 高圧下	2016-07-22 14:12:37.439	2	download	/dataInt/PILATUS/bl03xu/20160714/9	PILATUS

Figure 5: Web interface to search measured data for 2D detectors. Tornado [6] based on Python are used to build the interface.

We also developed software tools for image handling with DARUMA. Previously, we had the application for online monitoring with dedicated tools prepared for each detector. However, we sometimes had difficulty in the image analysis because the customizations in these tools were not flexible. Furthermore, computers used for the analysis of the image were restricted by the location of the storage server for the image data. DARUMA can solve these problems and improve flexibility for the image analysis. In DARUMA, we developed EM to manage image data. As MADOCA can have messaging controls with variable length data, image data can be utilized in the EM for the image handling. With the application for the function of image handling separated from the EM, we can have online image monitoring from a client application in a remote computer as shown in the GUI of Figure 6. In the GUI, we can have an online monitor with an appropriate rate of a few Hz for PILATUS, though a data size of the image is about 4 MB for one image. The GUI can also be flexibly applied to the online image monitoring for other 2D detectors because the GUI uses image data attached in the messaging of MADOCA, and data format for the image data is generalized with MessagePack [7]. The GUI is built with PyQt [8] using pyqtgraph [9]. The customization of the GUI, such as image analysis with ROI, can be implemented easily and flexibly by programming with Python.

We finished the development of DARUMA for EM and data collections and image handling began testing the operation with DARUMA in the experimental measurements at BL03XU. However, we have about 20 GUIs for the measurement applications and the replacement of these with DARUMA is still on going. The migration to DARUMA is going well, and we aim to completely switch into the DARUMA system for the measurement system at BL03XU next March.

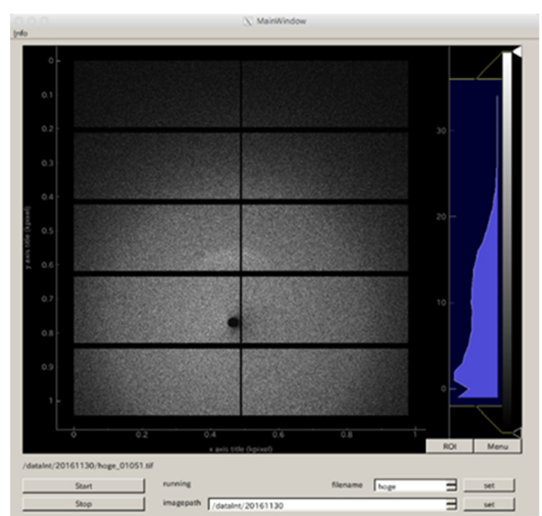


Figure 6: GUI for online monitoring of PILATUS image data with DARUMA.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THMPL07

IMPLEMENTATION OF DARUMA INTO OTHER EXPERIMENTAL STATIONS

In the previous section, we reported the implementation of DARUMA into all measurement systems for BL03XU. However, it is not realistic to replace all systems for existing experimental stations.

However, we can implement DARUMA into a partial set of the measurement systems. As station staff is interested in the tools for image handling for DARUMA, we started to implement DARUMA for EM of 2D detectors and image handling into several stations, namely, BL08W, BL13XU, BL14B2, BL19B2, and BL46XU. The operation of DARUMA for these stations has started this September. In these applications, the operation of image handling is performed through messaging commands with MADOCA. As DARUMA is implemented for the partial system, we provided a socket intermediate process to communicate with MADOCA from other applications.

In BL14B2, we also developed a dedicated EM for image analysis to perform image integrated sum with ROI, and this function is being utilized for the measurement of QXAFS (Quick X-ray Absorption Fine Structure) as shown in Figure 7. The image integrated sum is used to estimate photon counting and fast processing can be expected with a separated process of control EM for the image analysis. Though the study for the image analysis is ongoing, we expect such tools for the image handling to be useful in improving the measurements for stations and we aim to have stable measurements to process the image integrated sum with 50 ms interval.

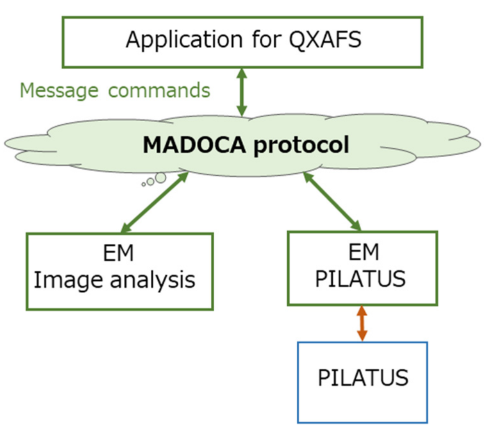


Figure 7: An example of utilization of EM for the image analysis of QXAFS at BL14B2. The EM for image analysis fetches the image data from EM for PILATUS and performs calculations such as image integrated sum with ROI. The obtained analysed results can be easily extracted from remote application with messaging command of MADOCA.

SUMMARY

We developed DARUMA for data collection and controls for X-ray experimental stations. DARUMA utilizes MADOCA, which was developed for distributed controls used in the accelerator and beamline control for SPring-8. With the introduction of DARUMA, each function in the measurements for stations are separated into software components such as EM, data collections, and image handling, and we can perform measurements with these software components through the messaging commands of MADOCA. The separated software structure of DARUMA facilitates flexible experimental setup, promotes rapid preparation of the measurement application, and helps the reuse of software applications. As a first attempt, we implemented DARUMA into BL03XU.

The migration to the DARUMA system has been proceeding smoothly. Functionality for image handling is especially useful. It has been implemented into a partial set of the measurement systems for several stations, and has been in operation with DARUMA since this September.

We confirmed that the DARUMA framework is useful for the measurement system for stations. To promote DARUMA for more users in stations, we plan to enrich software tools in DARUMA such as improve image handling tools and enrich documents and installers.

ACKNOWLEDGEMENT

The authors would like to thank the members of the Research and Utilization Division and the Industrial Application Division in Japan Synchrotron Radiation Research Institute for valuable comments and advice on the development of DARUMA for experimental stations at SPring-8.

REFERENCES

[1] T. Matsumoto *et al.*, “Next-Generation MADOCA for SPring-8 Control Framework”, in *Proc. ICALEPCS'13*, San Francisco, USA, October 2013, paper TUCOCB01.

[2] LabVIEW - National Instruments, http://www.ni.com/labview

[3] Certified Scientific Software, https://certif.com/spec.html

[4] T. Matsumoto *et al.*, “LabVIEW Interface for MADOCA II with Key-Value Stores in Messages”, in *Proc. ICALEPCS'15*, Melbourne, Australia, October 2015, paper WEM305.

[5] Elasticsearch: RESTful, Distributed Search & Analytics, https://www.elastic.co/products/elasticsearch/.

[6] Tornado Web Server, http://www.tornadoweb.org/en/stable/.

[7] MessagePack, http://msgpack.org/.

[8] Riverbank | Software | PyQt | What is PyQt?, https://riverbankcomputing.com/software/pyqt/.

[9] PyQtGraph - Scientific Graphics and GUI Library for Python, http://www.pyqtgraph.org/.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.