Online Data Monitoring Framework Based on Histogram Packaging in Network Distributed Data Acquisition Systems

T Konno¹, A Cabarera², M Ishitsuka¹, M Kuze¹, Y Sakamoto³
¹ Tokyo Institute of Technology, Tokyo, Japan
² CNRS/IN2P3-APC Laboratory, Paris, France
³ Tohoku Gakuin University, Miyagi, Japan
E-mail: konno@hep.phys.titech.ac.jp

Abstract. “Online monitor framework” is a new general software framework for online data monitoring, which provides a way to collect information from online systems, including data acquisition, and displays them to shifters far from experimental sites. “Monitor Server”, a core system in this framework gathers the monitoring information from the online subsystems and the information is handled as collections of histograms named “Histogram Package”. Monitor Server broadcasts the histogram packages to “Monitor Viewers”, graphical user interfaces in the framework. We developed two types of the viewers with different technologies: Java and web browser. We adapted XML based file for the configuration of GUI components on the windows and graphical objects on the canvases. Monitor Viewer creates its GUIs automatically with the configuration files. This monitoring framework has been developed for the Double Chooz reactor neutrino oscillation experiment in France, but can be extended for general application to be used in other experiments. This document reports the structure of the online monitor framework with some examples from the adaption to the Double Chooz experiment.

1. Introduction
In recent high energy physics experiments, data acquisition (DAQ) systems are becoming larger with a lot of subsystems, which communicate with each other via the local area networks (LANs). Therefore, a lot of information which should be monitored online is also distributed in the LANs. There are two difficulties for online monitoring in the distributed systems. The first is how to achieve the scalability for a lot of computing nodes in modern DAQ systems. The second is how to access monitoring information from outsides of the firewalls.

“Online monitor framework”, a new software framework for online monitoring was developed to address such requirements in high energy physics experiments. It provides a way to collect information from online systems, including DAQ systems, and displays them to both on-site shifters and off-site experts via internet. The development of the online monitor framework was started originally for the Double Chooz experiment [1, 2], a reactor neutrino experiment in France, but this framework was designed to have enough versatility to be used in other experiments.
2. Online monitor framework

The overview of the online monitor framework is shown in figure 1. The framework consists of three parts, DAQ subsystems, “Monitor Server” and “Monitor Viewer”. Between DAQ subsystems and Monitor Server and between Monitor Server and Monitor Viewer, the systems are connected by TCP sockets. DAQ subsystems create and send monitoring information as collections of histograms, named “Histogram packages”. The monitor framework is designed to be able to increase the number of monitored DAQ subsystems dynamically without any modification of both Monitor Server and Monitor Viewer. Monitor Server collects and stores temporarily monitoring information then transfers them to Monitor Viewers. Two types of Monitor Viewers were developed. One is Java based viewer which accesses Monitor Server directly by TCP sockets. The other version is based on web browser technologies such as HTML and java script accessing Monitor Server through web server by HTTP.

Monitor Server was developed with C++ on Debian / GNU Linux (Lenny). Monitor Viewer is based on Sun Java 6 [3] and Google Web Tool kit [4]. Details of each component are described in following sections.

![Diagram](image)

**Figure 1.** Overview of Online monitor framework

2.1. Histogram package

“Histogram package” is a common interface between DAQ subsystems and Monitor Server and between Monitor Server and Monitor Viewer. Each histogram package has its name to be used for identification and serial ID to check updates of the contents. The histogram package can contain several types of objects to be monitored, histograms (1 and 2 dimensions), graphs (scatter plots and time variation), and simple shapes (squares, circles and text messages).

2.2. DAQ subsystems

Each DAQ subsystem communicating with Monitor Server has one histogram package and one TCP connection. Basic analysis and creation of histogram packages are done on the DAQ sides. It enables to centralize the management of monitoring information on the Monitor Server.
2.3. *Monitor Server*

Monitor Server is a core system of the online monitor framework. The detailed structure of Monitor Server is shown in figure 2. The server consists of two processes, Monitor Skeleton and Monitor Provider, which are communicating with each other by shared memories. The shared memories work as buffers to absorb differences in performance between DAQ subsystems and Monitor Viewers.

2.3.1. *Monitor Skeleton*  
Monitor Skeleton accepts connections from DAQ subsystems and each connection is handled by one thread so that DAQ subsystems with different refresh rates can send updates of histograms in parallel. Monitor Skeleton stores the updates of histograms in the shared memories and notifies Monitor Provider by using a mutex and condition variable. The skeleton threads always wait for receiving updates of the histogram packages. It means that the DAQ subsystems can control the timing of updates of the packages.

2.3.2. *Monitor Provider*  
Monitor Provider can read histogram packages from the shared memories and send them to Monitor Viewers. All information for monitoring including GUI configurations is sent to Monitor Viewer with single connection from the provider. Each thread on the provider waits for notification of update from Monitor Skeleton. When the notification comes, the threads check which packages are updated by the serial IDs and send them to the viewers.

![Monitor Server Diagram](image)

*Figure 2. Monitor Server*

2.4. *Monitor Viewer*

The monitor viewers were designed to be able to access monitoring information from wherever internet is available. Two types of platform-independent technologies, Java and web browser, were adapted because they have functionalities to get monitoring information from Monitor Server via the network and to make graphical plots on their windows. Java-based viewers can achieve enough performance to realize real-time monitoring with 1 [Hz] while browser-based viewers are easy to access from outside of the firewall. By creating visible plots on the viewer side, the data transfer through the networks is minimized and the dynamical control of the plot, such as zooming and changing colors, is enabled. Both types of viewers are free from any libraries or plugins like ROOT [5] and have been tested on Windows, Mac OS X, and Linux. The GUI configurations are controlled by XML-based configuration files (see details in section 3). All of information needed to create plots is provided from Monitor Server.
2.4.1. Java based viewers  The Java based viewers can connect with Monitor Server directly via TCP sockets and receive the contents of the histogram packages. GUIs of Java viewers were developed with Java Swing [6] technologies. The java version can achieve relatively high performance than web browser based viewers. The left side of figure 3 is a screen shot of Java based viewers.

2.4.2. Web browser based viewers  The web browser based version of monitor were developed with web technologies based on HTML and Java script. The GUIs are based on Google Web Toolkit (GWT). GWT works as a compiler from Java source codes to HTML and java script which can operate on several web browsers. On the other hand, java script cannot support binary data streams with TCP sockets. Therefore, a simple process working in web server connects with Monitor Server, collects monitoring information and write down to XML based text files. The viewers on web browsers access the XML files through Ajax [7] communication and create visible plots on the window with HTML Canvas [8] technologies. With the browser based version, it is easier than Java based version to access the monitoring information from outside of the firewalls. The right side of figure 3 is a screen shot of web browser based viewer.

![Figure 3. Screen shots of Monitor Viewers (left: Java, right: web browser)](image)

3. GUI configuration with XML
XML based configuration files were adapted to define the properties of GUI components in Monitor Viewers. One XML file is loaded for one histogram package from Monitor Server. In the XML files, there are two types of components. One is GUI panel on the window and the other is graphical object on the canvas. Monitor Viewers can create the GUIs automatically. Therefore, each DAQ subsystem can display the monitoring information with its own GUI layout in the same framework.

3.1. GUI panels
Canvas, table panel, tabbed panel and scroll panel are available in the configuration files. The configuration files create the nested structures of GUI components like HTML.

3.1.1. Canvas  Canvas is a panel on which graphical objects are drawn. The XML element of canvas can have properties for graphical objects including monitored objects (see section 3.2).

3.1.2. Table panel  Table panel has properties for the number of rows and columns and the ratios of the widths and heights of table cells. Child elements of table elements are placed in the tables.
3.1.3. Tabbed panel  Tabbed panel can add tabs in its panels.

3.1.4. Scroll panel  Scroll panel can create vertical and horizontal scroll bars when some large size components such as big tables are added to avoid that the panels get crushed. Child elements of scroll panel elements are set in the view point of the scroll panels.

3.2. Graphical objects  
Graphical objects are the objects which are drawn on canvases. It is easy to define the graphics properties of the objects, for example line colors and font sizes from the XML configuration files.

3.2.1. Pads, legend and Axis on the canvas  A pad is created for each canvas to show histograms and graphs on it. It is also easy to add legends to show the list of histograms on the canvas. The properties of them, for example positions and colors, are controlled from the XML files. In addition, second y axis can be added to the canvas. An example of these objects is shown in the left side of figure 4.

3.2.2. Histograms and graphs  Histograms and graphs are drawn in pads on canvases. The maximum and minimum values of them can be defined. In addition, there are drawing options for graphs: lines or markers or both like ROOT. 2D histograms show the bin contents as colors and it is also easy to redefine the color patterns (figure 4).

3.2.3. Simple shapes in the Histogram packages  Histogram packages can also have some monitoring information of simple shapes; texts, squares, circles and paths, which are updated by DAQs. The default properties including positions on the canvases can be defined from the XML files. The example of shapes is shown in the right side of figure 4.

3.2.4. Additional shapes  It is also possible to put objects with various shapes by XML files. All of properties of the objects are defined from the configuration files.

**Figure 4.** Examples of objects on the canvases

4. Application examples in the Double Chooz experiment  
The online systems including DAQs of the Double Chooz experiment are using the framework and some screen shots of Monitor Viewers are shown as an application example. This monitoring framework has been successfully tested in the experiment during the detector commissioning.
The top-left side of figure 5 shows trigger rates from the DAQ test. It was updated every second and the top-right side of the figure is a run state monitor from Run Control showing the progress of data taking. The bottoms of figure 5 is from the high voltage control systems. The left plots showed the summary of high voltage operation and the right plots showed PMT channel maps in the Double Chooz detector.

Figure 5. Screen shots of Monitor Viewers with the Double Chooz experiment

5. Conclusion
In recent high energy physics experiments, a lot of monitoring information is distributed in the LANs and there are some difficulties to do online monitoring beyond the firewalls. “Online monitor framework”, a new general software frame work, was developed to give a way to manage these information and display them on the computers far from the experimental site. The monitor framework was adapted by the Double Chooz experiment and has been tested successfully during the commissioning.

Acknowledgments
T.K. also acknowledges the financial support from the Global Center of Excellence Program by MEXT, Japan through the ”Nanoscience and Quantum Physics” Project of the Tokyo Institute of Technology.

References
[1] Ardellier F et al, Double Chooz: A Search for the Neutrino Mixing Angle $\theta_{13}$, 2006, hep-ex/0606025.
[2] Konno T et al, Online Monitoring System for Double Chooz Experiment, Conference Record for 2009 IEEE Nuclear Science Symp. and Medical Imaging Conf., pp.622-628.
[3] Java, http://www.java.com/en/download/faq/whatis_java.xml
[4] Google web toolkit, http://code.google.com/intl/en/webtoolkit/
[5] ROOT, http://root.cern.ch/
[6] Swinf, http://download.oracle.com/javase/tutorial/uiswing/
[7] Open Ajax, http://www.openajax.org/
[8] HTML5, http://www.w3.org/TR/html5/