A Software Event Summation System for MDSplus

W. M. Davis[a]*, D.M. Mastrovito[a], P.G. Roney[a], P. Sichta[a]

[a]*Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ, 08543, USA*

Abstract

The MDSplus data acquisition and management system uses software events for communication among interdependent processes anywhere on the network. Actions can then be triggered, such as a data-acquisition routine, or analysis or display programs waiting for data. A small amount of data, such as a shot number, can be passed with these events. Since programs sometimes need more than one data set, we developed a system on NSTX to declare composite events using logical AND and OR operations. The system is written in the IDL language, so it can be run on Linux, Macintosh or Windows platforms.

Like MDSplus, the Experimental Physics and Industrial Control System (EPICS) is a core component of the NSTX software environment. The Event Summation System provides an IDL-based interface to EPICS. This permits EPICS-aware processes to be synchronized with MDSplus-aware processes, to provide, for example, engineering operators information about physics data acquisition and analysis.

Reliability was a more important design consideration than performance for this system; the system's architecture includes features to support this. The system has run for weeks at a time without requiring manual intervention. Hundreds of incoming events per second can be handled reliably. All incoming and declared events are logged with a timestamp. The system can be configured easily through a single, easy-to-read text file.

Keywords: NSTX, MDSplus, EPICS

* Corresponding author. E-mail: bdavis@pppl.gov, Phone: (609)-243-2546, Fax: (609)-243-3086

1. Background

MDSplus [1,2] is a set of Open Source software tools for data acquisition and storage, as well as a methodology for the management of complex scientific data. It is specifically targeted for pulse-based Fusion Energy experiments. Hardware and software settings may be configured from a standard tool (Traverser [3]) or from custom-built programs. These settings, along with monitored values and configuration details, all acquired and analyzed data, and subjective comments, may be stored in a hierarchical, self-describing data structure associated with a particular experimental pulse (shot). Like MDSplus, the Experimental Physics and Industrial Control System (EPICS) [4] is, like MDSplus, a set of Open Source software tools developed collaboratively and used worldwide [5]. It is typically used for interactive control and continuous monitoring on large scientific instruments such as a particle accelerators and telescopes. Both of these systems have been used on National Spherical Torus Experiment (NSTX) [6] at the Princeton Plasma Physics Laboratory (PPPL) since it began operating in 1999. As shown in Fig. 1, for NSTX EPICS is primarily used for "Engineering" subsystems [7], such as the torus vacuum system, the water cooling system, and the power supply coils. MDSplus is primarily used on the "Physics" side for diagnostic control and data acquisition, display and analysis [8]. MDSplus supports 57 diagnostics and subsystems, and EPICS supports 19 subsystems for NSTX. Convenient communications and synchronization between EPICS and MDSplus is required.

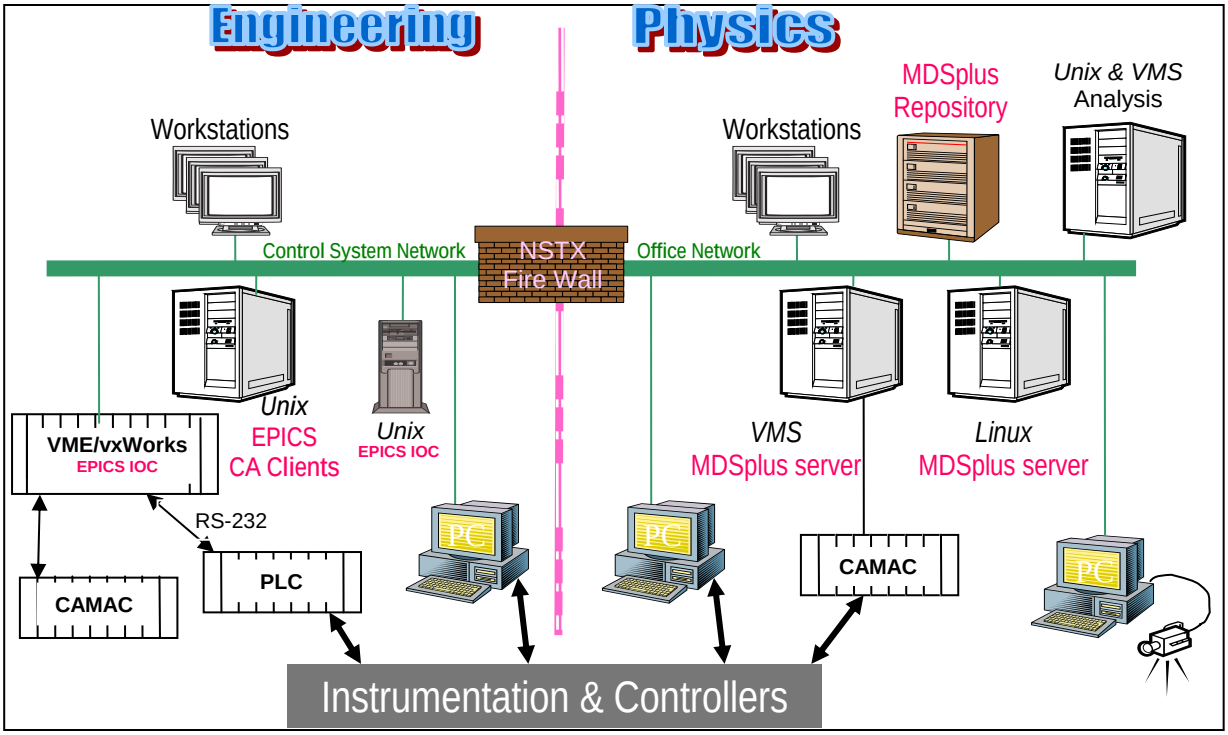

Fig. 1. The NSTX data acquisition and control system.

Thousands of control and monitor points are necessary to operate a fusion experiment such as NSTX, similar to plant operations for a factory. In 2007 NSTX had 250 different EPICS display pages for 2500 I/O points for such functions. Additionally, pulsed fusion experiments have unique data handling requirements because large bursts of data are taken in a very short time. About every 15 minutes a one-second plasma is produced. Approximately 1 GB of raw data is acquired from instruments in dozens of subsystems, hosted by Unix, VMS and Windows computers. This raw data is transferred to and stored in an MDSplus repository and is available within a few minutes to anyone "subscribed" to particular signals. Some of this raw data is also

used as input for analysis programs, which also store their results in MDSplus, both to provide immediate feedback to the machine operators and diagnostic physicists so they may make adjustments for the next machine pulse or "shot," as well as to provide a repository of information for later off-line analysis.

Synchronization is needed between various phases of the experimental cycle, as well as between data providers and data consumers. For example, when an event issued by an MDSplus STORE action is received, it can trigger an analysis job for that data. There are 63,000 data nodes in MDSplus of which 15,000 are "signals" originating from analog channels. In addition to raw data, about 200 MB of analyzed data is stored per shot. A typical NSTX run day produces about 40 shots and the facility operates for between 60 and 100 days a year. Currently, about 4 terabytes of compressed raw and analyzed NSTX data reside on disk.

2. Objectives of the Event Summation System

MDSplus communicates between tasks with software messages called "events." This is done over the internet using TCP/IP socket connections to a designated Event Server host computer. Event messages can contain up to 2 KB of data. In NSTX applications, if this data field contains anything, it is usually just the NSTX shot number. Someone wishing to display data as soon as it is available, simply waits on the associated event. More complex analyses, however, typically need more than one data set, and must wait on what should be the latest-occurring event, and then poll for other data if it is not yet available; the Event Summation System declares composite events using logical AND and OR operations. There is no event-recording mechanism that is part of the standard MDSplus distribution; the log file for this system provides such a record. Previous work allowed MDSplus events to be generated from EPICS [9]; this work describes notifying EPICS of any MDSplus events.

3. System Design

The key elements to the Event Summation System, shown in Fig. 2, are the processes ev2files.pro and sumEvents.pro. These tasks were written in the IDL language [10], so they can be run on Linux, Macintosh or Windows platforms. Multiple instances of the IDL components may be run on more than one computer for performance or management reasons.

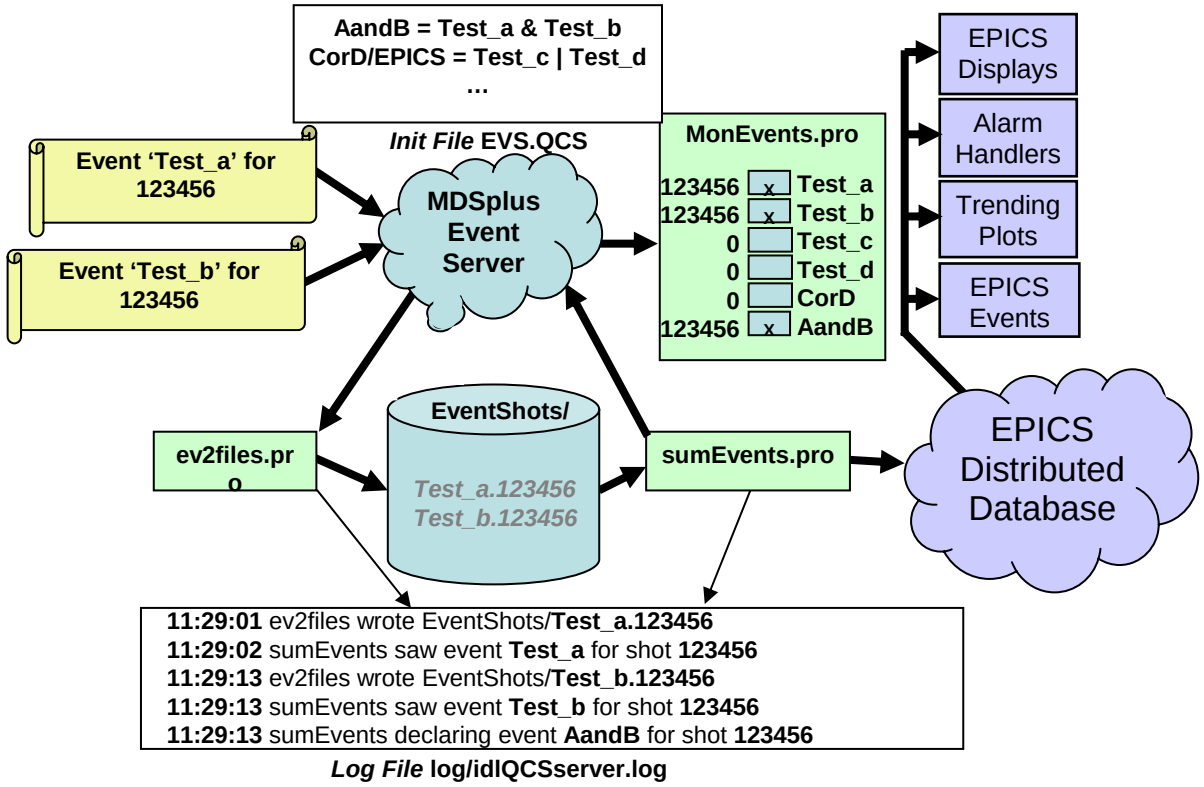

Fig. 2. The Event Summation System for MDSplus.

Events come into MDSplus as messages through TCI/IP connections. Processes subscribed to a particular event, ev2files.pro in this case, are notified when the event occurs and can be passed the shot number (or other data) in the message. For simplicity and reliability, ev2files.pro simply forms files (of length zero) whose names are the event name with the data value (usually the shot number) as the extension. sumEvents.pro notices when these files are created and builds and fills a data structure to keep track of unfulfilled dependencies. This data structure in sumEvents.pro is "check-pointed" to disk whenever it changes, so that crashes or restarts of the system lose few, if any, events.

The initial event dependencies are represented in a text file (e.g., EVS.QCS at the top of Fig. 2), consisting of simple Boolean expressions, with optional keywords (such as /EPICS) after the event to be declared. When all dependencies for an event are satisfied for a particular shot, it is declared with that shot number. In Fig. 2, for example, after events 'Test_a' and 'Test_b' arrive for shot 123456, the event 'AandB' is generated for that shot. If the /EPICS keyword follows an event on the left-hand side of the equation, that event is relayed to an EPICS "channel access process variable" (by calling CAPUT [11]) with a name similar to the event name.

The MonEvents.pro program is an auxiliary program, also shown in Fig. 2, that looks for the events used in the Event Summation System. Besides the advantage of visually seeing when events occur, the logging feature of this system is useful in diagnosing problems after the fact. EPICS allows users to generate a variety of sophisticated alarm handlers, panel displays, trending plots, etc.

Another auxiliary system used on NSTX for reliability is the Auto-reload system (not shown). Linux processes are checked periodically and, if missing, restarted by a script

4. Performance

Each instance of this system can watch for up to 2000 events. MDSplus events can be

generated on one host and recognized anywhere on our network in about a millisecond, and thus satisfy most speed requirements for NSTX, although some real time operations, such as our Plasma Control System, require smaller latency. Event rates for this system were tested by generating them from one and from two different computers simultaneously (both connecting to a remote MDSip server). The system can process an event in ~5 ms. The performance of this system was largely independent of NSTX shot-processing activity. When events were generated at 5/s, none were missed by this system. At 10/s from one computer, none were missed, but at 10/s from two computers (5/s from each), ~0.1% of the events were missed about half of the time. As the frequency of event declaration increased, the percentage of missed events increased. At 10/s, ~0.2% were missed. At 1000/s ~1.5% were missed. These misses are not inherent in MDSplus events, because 6000 can be generated, from one or two servers, with no waiting between them (which takes 8.5 s and 12.75 s, respectively), and no events are missed by a simple program. Further work will be done to eliminate these misses (see below) .

During operations, NSTX has been summing only 30 events per shot, which depend on a total of 60 different events (most occur within a 5 minute period). No errors have been detected or suspected in several months of running. The tasks have run for 6 weeks without needing to be restarted. This Event Summation System can be easily distributed among different computers if higher rates are desired.

5. Future Plans

Now that EPICS can know about MDSplus events, we have many areas in which we can improve our notifications of problems. Missing diagnostic information which is critical to operations will indicate alarms on operator screens and send e-mails to interested parties.

While the reliability of the Event Summation System has been flawless during operations, if we increase the number of events by an order of magnitude, we need to improve the design, or distribute the system among several nodes.

In order to make this into an Open Source system, it will be necessary to rewrite the modest amount of code that uses the proprietary IDL language.

6. Summary

This straightforward add-on to MDSplus provides a reliable, easy-to-use system to recording desired MDSplus events and declaring “composite” events when logical combinations of other MDSplus events are satisfied. This system also provides a convenient way to inform EPICS of MDSplus events, so its sophisticated alarm and monitoring tools can be used.

Acknowledgements

This work was supported by DOE Contract DE-AC02-76CH03073.

References

[1] MDSplus, http://www.mdsplus.org/

[2] J.A. Stillerman, T.W. Fredian, K.A. Klare, G. Manduchi, MDSplus Data Acquisition System. *Rev. of Sci. Instrum.* **68** (1) January 1997, p. 939.

[3] http://nstx.pppl.gov/nstx/Software/TRAVERSER.html

[4] EPICS - Experimental Physics and Industrial Control System, http://www.aps.anl.gov/epics/

[5] K.H. Kim, C.J. Ju, M.K. Kim, M.K. Park, J.W. Choi, M.C. Kyum, M. Kwon, the KSTAR Control Team, The KSTAR integrated control system based on EPICS, *Fusion Eng. Des*. **81** (2006), 1829-1833.

[6] S. Kaye, M. Ono, Y.-K.M.Peng, D.B. Batchelor, M.D. Carter, W. Choe, et al., The Physics Design of the National Spherical Torus Experiment. *Fusion Technology*, 36, July 1999, p. 16, or http://nstx.pppl.gov/

[7] P. Sichta, J. Dong, G. Oliaro, P. Roney, Overview of the NSTX Control System, 8th ICALEPCS – International Conference on Accelerator and Large Experimental Physics and Control Systems, San Jose, CA . (2001)

[8] W. Davis, P. Roney, T. Carroll, T. Gibney, D. Mastrovito, The use of MDSplus on NSTX at PPPL, *Fusion Eng. Des*. **60** (2002), 247-251.

[9] D. Mastrovito, W. Davis, J. Dong, , P. Roney, P. Sichta, Integrating EPICS and MDSplus, *Fusion Eng. Des*. **81** (2006), 1847-1851.

[10] IDL, The Data Visualization & Analysis Platform, http://www.ittvis.com/idl/

[11] http://nstx.pppl.gov/nstx/Software/Programming/pppl_idl_routines.html#CAPUT