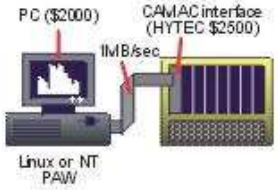
New Components of the MIDAS Data Acquisition System

S. Ritt, University of Virginia, USA and Paul Scherrer Institute, Switzerland

P. A. Amaudruz, TRIUMF, Vancouver, Canada

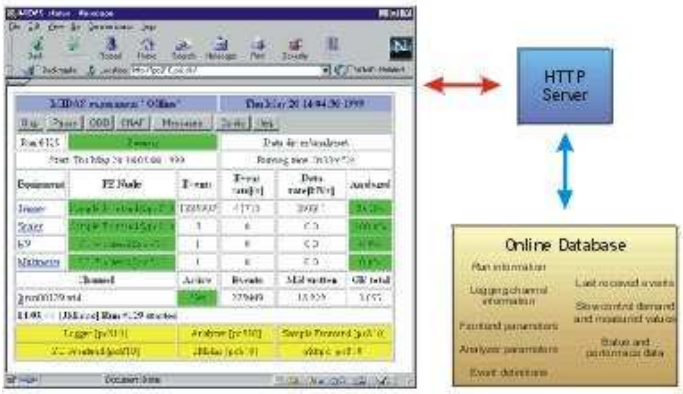
The **M**axiumum **I**ntegrated **D**ata **A**cquisition **S**ystem is suited for small and medium scale experiments. The complexity ranges from test systems, where a single PC is directly connected to CAMAC, to experiments with several front-end computers and analysis. Workstations. It has been used in many experiments at TRIUMF and PSI mainly under Linux, VxWorks and WindowsNT.

Minimum System:



HTTP Server

Since its initial release in '97, many new components have been added. A dedicated HTTP server allows remote control of the experiment through any Web browser using just tables, forms and cookies. Since the server does not use CGI subprocesses, performance is high and CPU load low.



Unified drivers

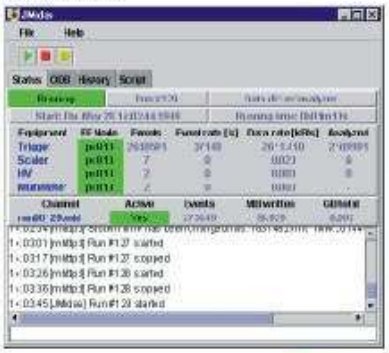
Unified drivers have been written to access CAMAC, Fastbus and VME. By defining a Midas Camac Standard, user frontend code may use any supported CAMAC interface or embedded processor without code change. In addition to the Esone standard new functions have been added which utilize hardware supported features like subaddress scan. For compatibility a Esone driver has been written on top of the Midas Camac Standard drivers.

Alarm system

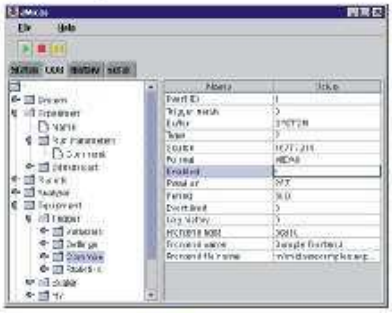
An alarm system has been incorporated into Midas which allows user notification through spoken messages, pager and telephone calls in case of alarm conditions. Individual values in the Online Database like slow control variables can be monitored directly or more complex alarm conditions can be evaluate by the user analyzer like histogram correlation. The high degree of automatization and the reliability of Midas make it possible to run experiments unattended in certain cases.

Java interface

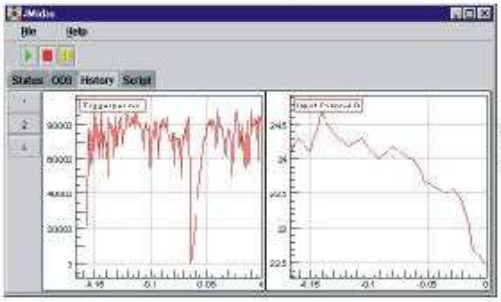
A Java interface has been implemented which allows platform-independent graphical experiment control and monitoring. An optimized communication layer using network sockets transports information between Java applications or applets and the midas server. A general purpose control program be has been implemented for experiment control and monitoring. Midas Java beans have been developed which can be used to extend the control program and adapt it to specific needs. Beans exist for data display and entry to the Online Data Base and for history plots.



JMidas control program displaying experiment status, event statistics, logging channels and system messages



User interface to the Online Database which resides in shared memory on the Midas server



History plots for system variables and entries in slow control events.

Scripting

Using the BeanShell interpreter written by Pat Niemeyer experiments can be completely scripted. Since the interpreter has full access to the Online Database, runs can be started and stopped from the interpreter and slow control variables can be modified. Complex tasks like performing several runs with different beam line settings through an EPICS channel access require only a few lines of code like:

```
bsh % OdbKey magnet = new OdbKey(exp,
  "/Equipment/Beamline/Variables/Magnet1");
bsh % loop() {
  for (i=0 ; i<10 ; i++) {
    magnet.setTextAt(100+i*10, i);
    exp.start();
    exp.waitForState(1);
  }
}
bsh % loop();
```

For more information