Concept and Status of the New HADES Run Control ◇

B. Sailer, J. Homolka, M. Münch, and B. Kolb [a]
[a] Gesellschaft für Schwerionenforschung mbH, Darmstadt

The new High Acceptance DiElectron Spectrometer HADES has been set up at GSI Darmstadt and is now about to take data. The ATM-based data aquisition together with a multilevel trigger system will be able to read out 70 000 channels with a first level trigger rate of 10^5Hz with zero supression and transport $1-2\times10^3$ events to mass storage every second [1].

The DAQ consists of more than 1 000 frontend and about 125 VME boards with a large number of different tasks and layouts. To operate this complex system a run control system based on the EPICS [2] package has been developed making use of its network communication layer, monitoring capabilities and sequencing.

The VME modules and all the processes of the DAQ software are distributed over a number of VME crates with PowerPC CPUs under LynxOS (the so called subeventbuilders). Two processes do the eventbuilding on a dedicated SMP machine (currently a i386-hardware under Linux named the eventbuilder). Both the subeventbuilders and the eventbuilder are called "subsystems of the DAQ".

The new concept of the HADES run control shown in Fig. 1 foresees to store a complete image of the currently used setup in an Oracle database.

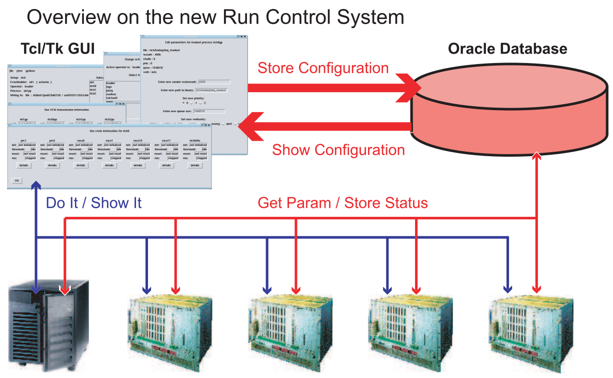

Fig. 1: Concept of the new HADES run control

The database is responsible for the consistency of the setup as well as for a complete version control. It will be possible to recall the whole setup and its parameters for any point in time which is of crucial necessity when calibrating and analyzing data. The control of the DAQ itself is performed with nearly no user interaction and configured by parameters directly read from the database. A GUI based on Tcl/Tk allows to run the DAQ, change parameters in the database and monitor all the status information of the run control. Run and status information is also written to the database automatically before, during and at the end of data and file taking.

The core component of the system is an agent running in every DAQ subsystem that handles all actions performed on the DAQ hard- and software and interfaces them to their external representation, called EPICS process variables (PVs). As shown in Fig. 2, the agent is developed in a distributed manner: Hardware experts who have developed the different VME modules write the routines to access their products and stick to an interface through which they are called by the agent. Both agent and hardware routines can read their configuration through a parameter interface as well, while the parameter source itself can be replaced without changing the code. Among others, possible alternatives for parameter sources are text files as well as the Oracle database.

This concept allows a fast developement of the hardware routines called from small standalone programs and reading their configuration from text files, while the final agent looks up its configuration information in the database.

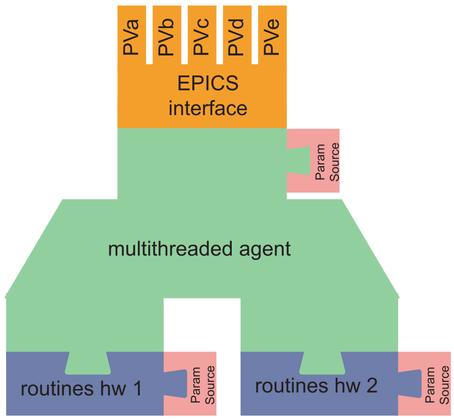

Fig. 2: Interface definitions for distributed developement of the run control agents

To control all the actions taken for configuring, starting and stopping the DAQ in the proper order and keep track of the status of the DAQ, a set of state machines has been developed based on the sequencer (a tool for implementing state machines with access to EPICS process variables). This sequencer itself is fully configured via parameters from the interface (which can again source text files as well as the database).

The system has recently been completed and is about to be tested with different DAQ setups. Apart from debugging and small improvements of some components the major task remaining will be to complete the database structure as well as routines within the database to extract and write back information from / to the tables.

References

[1] M. Münch, Ph.D. thesis, TU München, 2002
[2] B. Dalesio and M. Kraimer, *EPICS Homapage*, http://www.aps.anl.gov/epics/, 2002

◇ supported by BMBF (06TM970I) and GSI (TM-FR1)