



BNL-94244-2011-CP

***Quick setup of unit test for accelerator
control system***

W. Fu, T. D'Ottavio, D. Gassner, S. Nemesure, J. Morris

*Presented at the 2011 Particle Accelerator Conference (PAC'11)
New York, N.Y.
March 28 – April 1, 2011*

Collider-Accelerator Department

Brookhaven National Laboratory

**U.S. Department of Energy
Office of Science**

Notice: This manuscript has been authored by employees of Brookhaven Science Associates, LLC under Contract No. DE-AC02-98CH10886 with the U.S. Department of Energy. The publisher by accepting the manuscript for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

This preprint is intended for publication in a journal or proceedings. Since changes may be made before publication, it may not be cited or reproduced without the author's permission.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Quick Setup of Unit Test For Accelerator Control System*

Wenge Fu, Ted D'Ottavio, David Gassner, Seth Nemesure, John Morris

Brookhaven National Laboratory, Upton, New York 11793, USA

Abstract

Testing a single hardware unit of an accelerator control system often requires the setup of a program with graphical user interface. Developing a dedicated application for a specific hardware unit test could be time consuming and the application may become obsolete after the unit tests. This paper documents a methodology for quick design and setup of an interface focused on performing unit tests of accelerator equipment with minimum programming work. The method has three components. The first is a generic accelerator device object (ADO) manager which can be used to setup, store, and log testing controls parameters for any unit testing system. The second involves the design of a TAPE (Tool for Automated Procedure Execution) sequence file that specifies and implements all testing and control logic. The third is the design of a PET (parameter editing tool) page that provides the unit tester with all the necessary control parameters required for testing. This approach has been used for testing the horizontal plane of the Stochastic Cooling Motion Control System at RHIC.

INTRODUCTION

Accelerator control systems are complicated and consists of many subsystems, and each subsystem is also consist of many single control units of hardware or software. In the RHIC (Relativistic Heavy Ion Collider) control system, they are often controlled directly by a set of ADOs (accelerator device objects). When setup such a sub-control system, it is often necessary to test each control unit individually before assembling each unit into the control system to ensure each control unit works the way as designed and has a good reliability. For some controls systems, this kind of tests maybe easy, but for some other systems, this may not be easy to do without a reasonable software with a graphical user interface (GUI). Setup a program with GUI for a single control unit is probably a time consuming task, this is especially true if the controls logic of the unit test is complicated. Often, such an unit testing program may not be useful any more for the operation controls of the entire system after the unit testing is completed.

In RHIC, we have many powerful generic software systems and applications which work well with single or complicated accelerator control system units through the standard ADOs. Two of these generic applications can be used to setup a quick unit control system with proper organization and configuration. One of the application is: *Tool for Automated Procedure Execution* (TAPE) [1]. This generic program is used to run previously generated

sequences of control commands in a previously defined orders and with defined control logic. The other application is *Parameter Editing Tool* (pet)[2]. This program a general purpose program for viewing and editing control parameters for ADOs.

This article outlines a method to use these two generic applications with another generic ADO ---a *MiddleMan* ADO, to setup an unit control system with minimum programming. The *MiddleMan* ADO is a very simple dummy ADO which can be used to set, update, store, or display all middle values used for the unit control system. This *MiddleMan* ADO can be easily setup with a rich sets of ADO parameters to satisfy the needs of any single control unit test.

METHOD OF UNIT TEST SETUP

In RHIC control systems, a typical accelerator control unit consists three components:

1. A hardware control unit. The controls of a hardware unit typically can be done through the accelerator device object (ADO) with various kind of controls parameters (such as temperature, pressure, current, etc.);
2. A software control unit. This part implements the control logic: how the control unit supposed to be working: automatically, event triggered, or interactively etc.
3. An user interface unit. This is for either taking user's inputs, trigger control actions, or display device statuses and event message etc.

By utilizing the generic applications we already have, and with minimum programming in the tape sequence language, a typical unit test system can be setup quickly as shown in Figure 1. The Figure 1 outlines the components for an unit test system:

- "Hardware Control Unit": this is the hardware unit to be tested;
- "Hardware Control ADO": this is the ADO for the direct control of the hardware; It is part of the software control unit.

* Work supported by the U.S. Department of Energy.

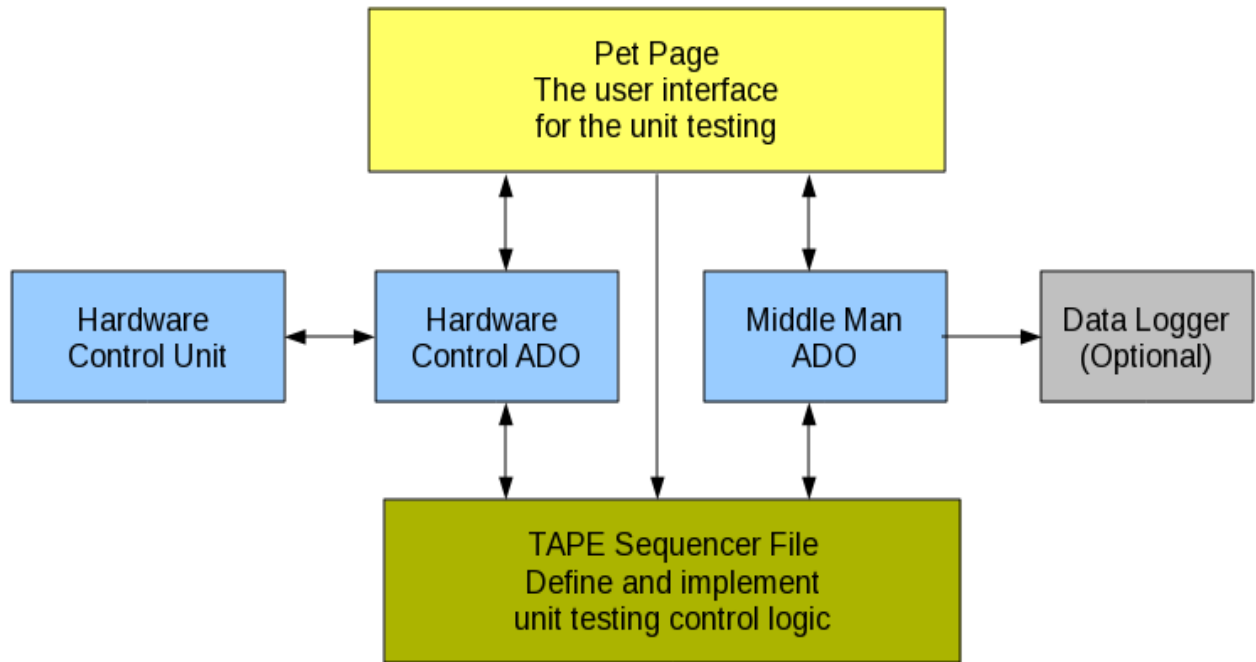


Figure 1. Diagram of Unit Testing System

- “Middle Man ADO”: this is a generic purposed ADO which serve as a middle man between the “Hardware Control ADO” and the rest of the software system. It is a dummy ADO with a rich set of ADO parameters which can be used for storing, setting, displaying, or logging any control parameters, middle values, or status buffers etc. during the unit testing. These middle man ADO parameters can be easily accessed or changed by other generic application such as *pet* and *Tape* etc. When a new middle man ADO is needed, it can be easily created simply by adding the new ADO into the configuration database and Controls Name Server (CNS) system, and restart the Middle Man ADO Manager.
- “Pet Page ...”: this is the user interface where users can set all required unit testing parameters, check system statuses, manually trigger testing actions, or conduct the unit testing and checking the progress of the testing etc.; This is the user interface unit.
- “TAPE Sequence File...”: This is the place where the control logic of the unit testing is defined and implemented. This is the only place requires minimum programming with *tape sequence language*[1]. During the execution of this file, the GUI of the *Tape* application also shows the progress of the unit test.
- “Data Logger”: This is an optional component. It is only needed if any testing data need to be logged for future data analysis.

A DEMO UNIT TEST SYSTEM

An unit testing system was setup with the method described above for testing the reliability of motion control in a single stochastic cooling kicker unit which is planned to be used in the new RHIC stochastic cooling system.

In this unit testing system, the hardware unit are two stochastic cooling kickers which are going to be installed in the RHIC ring from opposite sides of the ring pipe. There are several FEC ADOs for the motion controls of the kickers. The goals of the unit test are:

- Moving in and out the kickers with different motor speeds, close positions, kicker moving orders, and repeat times of the motion controls. These testing parameters are similar to these of using in the real operations.
- Find out the good control parameters and test the reliability of the unit.

SC_torture_test.tape	
Open Log File	P ?
Reset test counter	P ?
READ Motor Speed	P ?
READ right position	P ?
READ left position	P ?
READ acceleration	P ?
READ Time for Motion	P ?
READ Test Count Number	P ?
Set Counter	P ?
Set Message	P ?
Log Start Time	P ?
Log To File	P ?
▼ Begin Loop	P ?
Set counter value	P ?
Set Counter	P ?
Set Message	P ?
Log To File	P ?
Right Panic Stop	P ?
Left Panic Stop	P ?
Set Right Speed	P ?
Set Right ACC Speed	P ?
Set Right Brake Enable	P ?
Set Right Position	P ?
Wait insert right kicker to finish	P ?
Set Message	P ?
Log To File	P ?
Set left Speed	P ?
Set left ACC Speed	P ?
Set left Brake Enable	P ?
Set left Position	P ?
Wait insert left kicker to finish	P ?
Set Message	P ?
Log To File	P ?
Set left Speed	P ?
Set left Brake Enable	P ?
Set left Home	P ?
Wait home left to finish	P ?
Read left Switch Limit	P ?
▶ Check left Switch Limit	P ?

Run Retry Pause Cancel

Figure 2. Tape sequence of the unit test system

A Middle Man ADO with many desired ADO parameters is created to hold all testing parameters and messages during the tests.

A pet page is created for the unit test. The pet page serves as an user interface for the unit test. User can use the pet page to set or get the values of control parameters, to view intermediate results, to trigger control actions, to display testing event or program generated messages, to store control parameters, and to monitor the progress of the test etc.

A tape sequence file is created for the unit test. The *Tape* application executes this tape sequence file for the unit test. (show as in Figure 2.) The tape sequence file implemented all control logic of the unit control test. This

is the only place where minimum programming with tape sequence language is required. The *Tape* also provides an additional GUI to monitor the progress of the test, and enable tester to pause or resume the test if needed. This make the unit test even more flexible.

All parameters used in this unit test can be logged with standard RHIC control logging system. This makes the data analysis of the unit test very easy. The unit test system even be able to handle system errors by automatically checking middle results and error message, and handle them accordingly.

SUMMARY

A method of quick setup graphic user interface for single control unit test in accelerator control systems with

existing applications and minimum programming effort is described and a demo application of the method is presented. The method can be easily applied to many single unit test systems or used to get optimized control parameters for an individual control units in accelerator control systems.

REFERENCES

- [1] T. D'Ottavio, et al. Description of the RHIC Sequencer System, THAP062, proceedings of ICALEPCS 2001. San Jose.
- [2] D. Barton et al., RHIC Control System, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Volume 499, Issues 2-3, 2003, Pages 356-371.