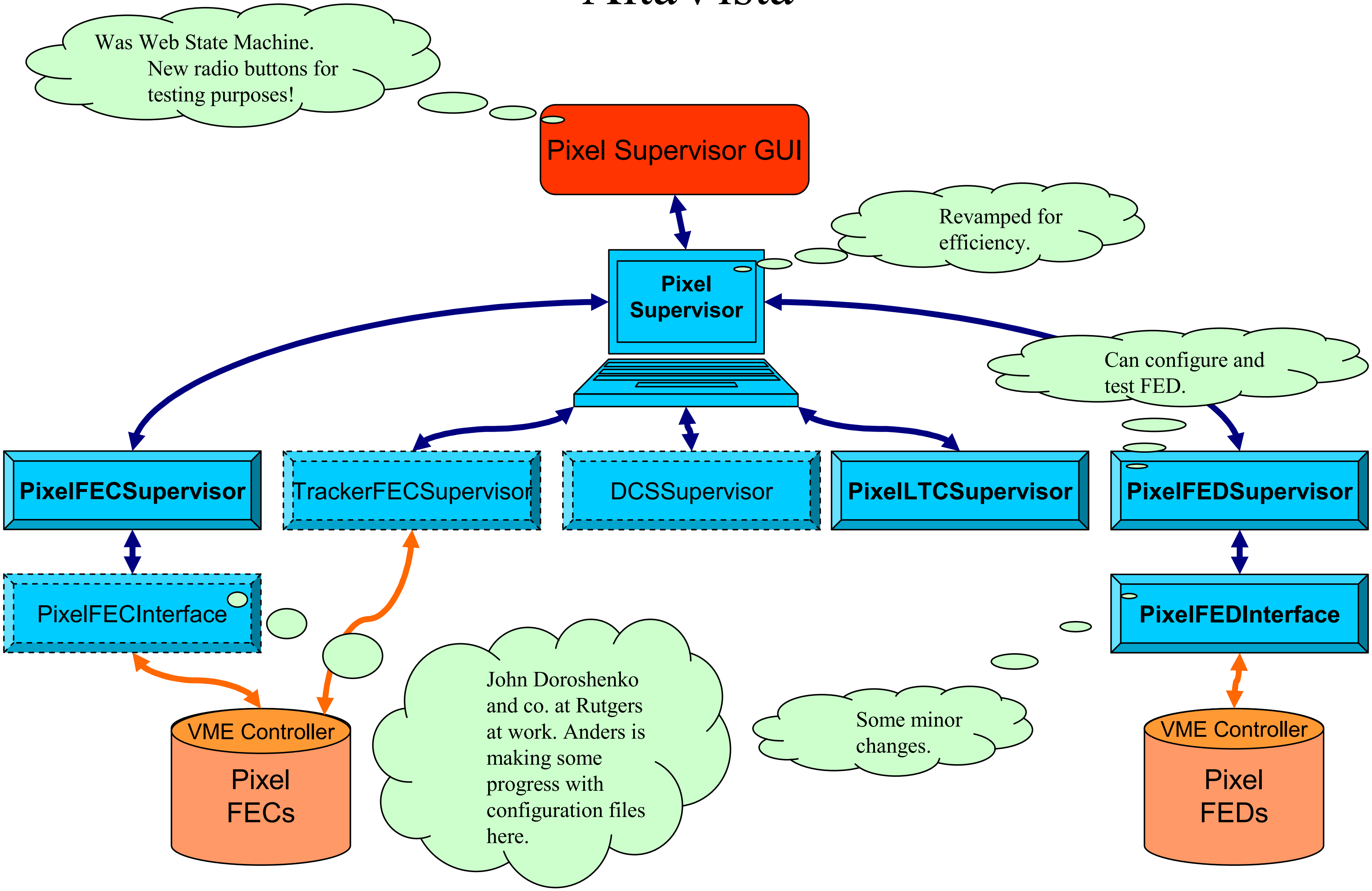
Pixel Run Control and Calibration

-Souvik Das

under Anders Ryd and Karl Ecklund

AltaVista



Was Web State Machine. New radio buttons for testing purposes!

Revamped for efficiency.

Can configure and test FED.

John Doroshenko and co. at Rutgers at work. Anders is making some progress with configuration files here.

Some minor changes.

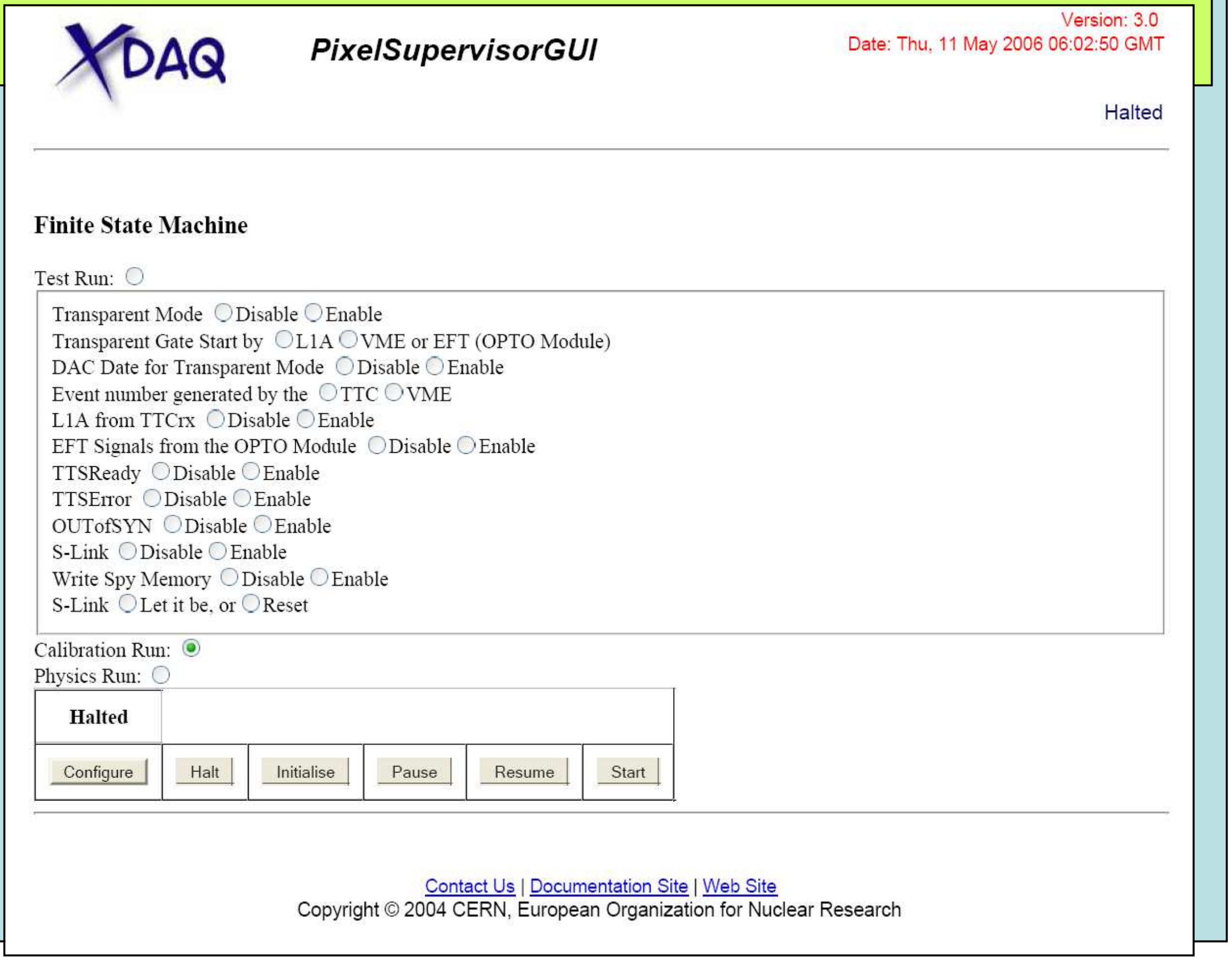
Pixel Supervisor GUI

Done

- Is the new name for WebStateMachine. Uploaded into CVS Repository.
- Sports new radio buttons for testing purposes! Loads the Control Register manually.
- Changed the button "Init" to "Initialise".

To Do

- The Control Register throws up if written more than once! Fix that.
- Load the Mode Register.
- Pop-up functionality for sub-menu radio buttons.
- Decide on sub-menus for Calibration and Physics runs. These will probably point us to databases to download configuration data from.



PixelSupervisorGUI

Version: 3.0
Date: Thu, 11 May 2006 06:02:50 GMT

Halted

Finite State Machine

Test Run: ○

Transparent Mode ○Disable ○Enable
Transparent Gate Start by ○L1A ○VME or EFT (OPTO Module)
DAC Date for Transparent Mode ○Disable ○Enable
Event number generated by the ○TTC ○VME
L1A from TTCrx ○Disable ○Enable
EFT Signals from the OPTO Module ○Disable ○Enable
TTSReady ○Disable ○Enable
TTSError ○Disable ○Enable
OUTofSYN ○Disable ○Enable
S-Link ○Disable ○Enable
Write Spy Memory ○Disable ○Enable
S-Link ○Let it be, or ○Reset

Calibration Run: ●
Physics Run: ○

Halted

Contact Us | Documentation Site | Web Site
Copyright © 2004 CERN, European Organization for Nuclear Research

Pixel Supervisor

Recent Changes

- Thoroughly revamped. Working methods bound directly to incoming SOAP messages from PixelSupervisorGUI. The State Machine still exists, but hardly useful. It is kept there with RCMS integration in mind.
- It can take a command with several arguments via SOAP from PixelSupervisorGUI. For example:

  Configure (Configuration Key="Test", controlRegister=0x0a, modeRegister=0x1, PixelColumn=30, PixelRow=20, ROC coordinates=… etc)

  It unpacks this SOAP message and repacks the arguments appropriately to be passed down to PixelFEDSupervisor, PixelFECSupervisor, PixelLTCSupervisor via SOAP. In this example, the controlRegister and modeRegister data go to PixelFEDSupervisor while the PixelColumn, PixelRow and ROC coordinates go to PixelFECSupervisor. (Routines for unpacking arguments were written in SOAPCommander.h)

  So far, this works along the PixelFEDSupervisor-PixelSupervisor-PixelFEDSupervisor arm.
- All references to "Init" changed to the verb "Initialise" in British English.

To Do

- Test the new SOAP message passing scheme down the PixelLTCSupervisor and PixelFECSupervisor arms.
- Upload into CVS.

Pixel FED Supervisor

Recent Changes

- A testing routine from Danek’s test program has been implemented in PixelFEDSupervisor. So, we tested the FEDs at SiDet and Vanderbilt right from our XDAQ GUI!
- Two very small helper .h files called “Converter.h” and “LevelEncoderDecoder.h” were written and placed under pixel/PixelUtilities/PixelFEDDataTools/include. They help in converting decimal to base 6 numbers, grey-codes, and converting 6-base numbers to pulse heights.
- The FIFO1, channel 1 has been tested so far in transparent mode using the test DAC. A correspondence between the DAC and ADC pulse heights has been identified, which is roughly a straight line with an intercept.
- To generate simulated pixel hits on the test DAC, a little method called createPulseTrain was written in PixelFEDSupervisor. It takes in the coordinates of the pixel and the charge deposit to be simulated and outputs a corresponding pulse train. This allowed us to reproduce the hardware bug in the FED’s state-machine Danek pointed out – the ordering of pulses that encode double-column and row+(even/odd) data is inverted. So we just changed the software accordingly for now.
- Another overloaded method for fillDACRegister(pulseStream1, pulseStream2, pulseStream3) introduced in PixelFEDInterface. Very similar to Danek’s method, but more convenient for PixelFEDSupervisor. Need to keep one of these.

To Do

- Test all FIFO 1 channels in transparent mode.
- Test all FIFO 1 channels in normal mode.
- Test Temperature and Error FIFOs.
- Test FIFO 2 and FIFO 3.
- Address level calibration. The address levels that we decide upon should be stored in a file or database.