Control System Overview

J. Patrick

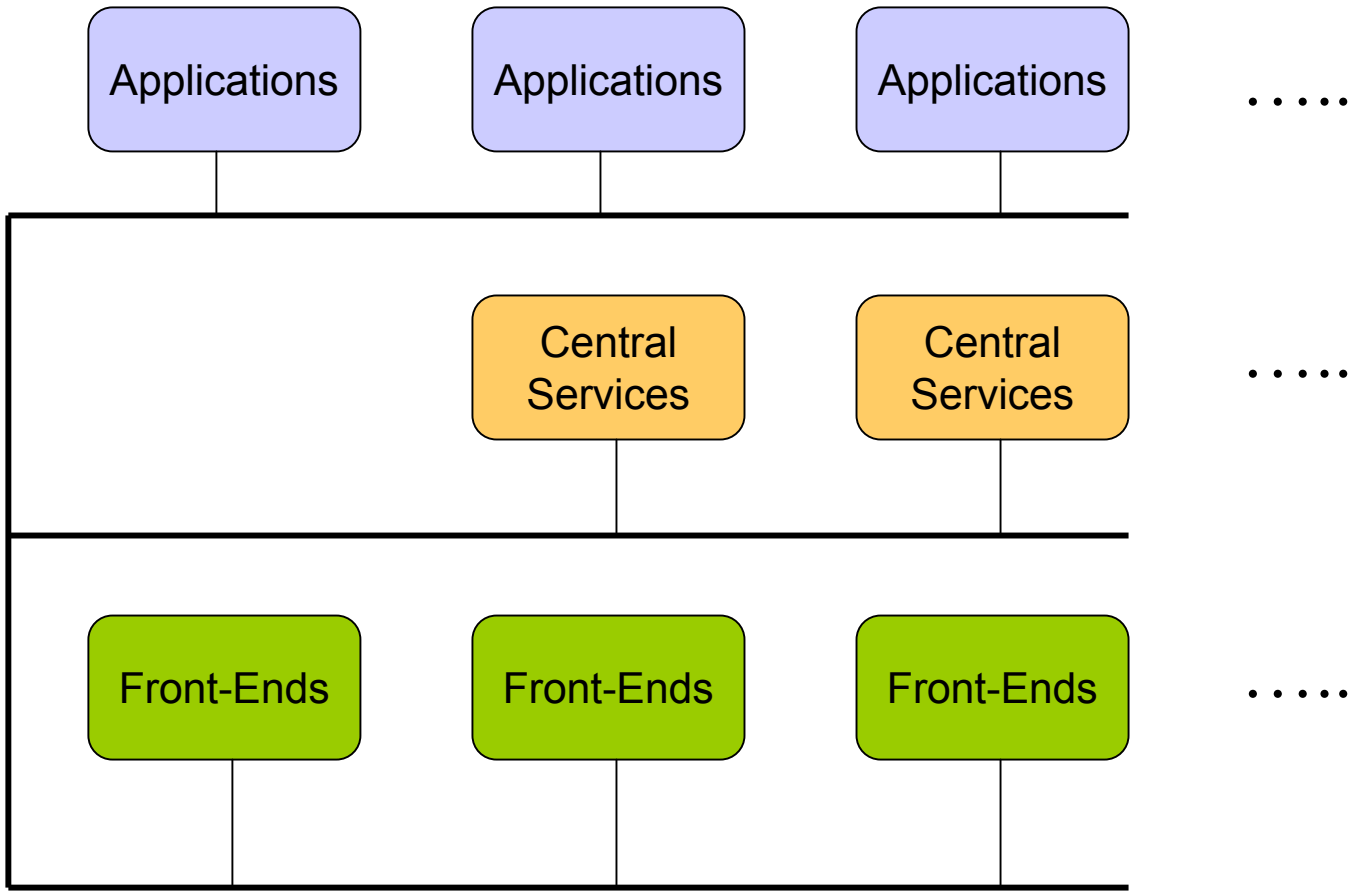
DOE Review

July 21, 2003

Control System Overview

- General Overview
- Migration Path
- High Level Automation Tools

Control System Overview



Control System Overview

- Common infrastructure for all machines
  - ~180,000 "devices" ("control points")
  - ~350 total front-end computers
    - Plus associated field hardware/embedded processors
  - ~100 VAX computers (console applications/development)
  - ~100 commodity unix computers (central services/development)
  - ~100 PCs/X terminals (display VMS apps, run Java apps)
  - Central Sybase database
  - Common communication protocol ("ACNET") over ethernet plus dedicated links for real-time signals
  - Efficient acquisition of continuous high rate data ("fast time plots" and "snapshots")
  - Easily configurable distributed data logging (~70 instances)
  - Transparent data source/destination redirection (accelerator, model, data logger, save file, etc.)
  - Automation tools – Sequencer, Sequenced Data Acquisition, Autotune

Front Ends

- Interface hardware to the control system; speak "ACNET"
  - Provide conduit for readings/settings/alarms etc.
  - Provide real time response when required
- Mostly VxWorks/PPC/68k; pSOS/68k + some Labview + …
  - Older 386/Multibus, 68000/VME CPUs retired
  - Many additional embedded CPUs in field hardware
- Variety of field buses supported: VME, CAMAC, VXI, PMC, ARCNET, GPIB, ethernet, Multibus, CIA…
- Significant modernization over the last few years
  - Both front-end CPU and field hardware
  - Linac, Cryogenics, Vacuum, CAMAC, …
  - Tevatron/MI BPM/BLM is major remaining obsolete system

Central Services

- Persistent processes with no user interface
  - Obey same communication protocol as front-end systems ("Open Access Front-Ends")
  - But run on higher level, non-real time operating system
- Data loggers, SDA acquisition, calculated devices, routine saves, etc.
- ~60 applications total (70 data logger instances)
- Majority migrated to Java, some still run on VAXes
  - ~6 left to migrate; some others will disappear
  - All new such programs in last > 1.5 years done in Java
- Run on rack mount commodity unix CPUs in computer room

Console Applications

- Programs with a user interface
- Mostly execute on VAXes, displayed on PCs via X Server
  - Java infrastructure exists, some general applications
- Very large number (>500) but not all active/used
- Generic, non machine specific applications
  - Parameter page, fast time plot, data logger plot, device database access, hardware diagnostics, sequencer…
- Machine specific applications

Communication

- Ethernet – based on "ACNET" protocol (over UDP)
  - Data, very efficient for high rate readings (fast time plots)
  - multicast clock events and state transitions
  - Token ring used for many years is now retired
- TCLK – clock events
  - Central Timeline Generator (TLG)
  - Dedicated cables/fanout network
  - Soft timing (< 100 microSecs)
  - Reflected to ethernet multicast (softer timing - ~ms)
- Beam Sync – precise timing
- MDAT – Machine Data (numbers)
  - Dedicated serial links

Migration Path

- Fundamental system paradigm is ~20 years old
  - Device model, communication protocol, application L&F, etc.
  - But considerable evolution of components over the years
- Last major upgrade of upper levels was ~1990 +-
  - VaxStation consoles/X11 graphics; 386/Multibus front-ends; ethernet communication; Sybase database, …
- Work on Java based infrastructure began ~5 years ago
  - Reduce dependence on VAX/VMS; more modern environment
  - Provide enhanced functionality; needed for SDA
  - Started with data acquisition infrastructure, SDA
  - Additional parts became critical to operations > 1.5 years ago
  - Currently ~3000 classes/~500K non-comment lines Java code
- Upgrades/migration must be accomplished concurrent with operation of the accelerator
  - Minimal shutdown time available
  - New pieces must be plug compatible with old

Migration Path

- Front-Ends
  - Primary platform will remain VxWorks in near term
  - Run II plan includes replacement of TeV/MI BPMs/BLMs
  - New Digitizer front-end available soon
    - 10-100 KHz continuous digitization on all channels, major improvement over what is in the system now
    - More simultaneous plots possible
    - Opportunity for more wave-form analysis in front-ends
  - Background R&D on real-time Linux/PCs but no specific project planned at this time.
- Central Services
  - Almost all central services now run on Java platforms
    - Finish migration in < 6 mos.
  - All new such programs are written in Java

Migration Path

- Console Applications
  - A substantial Java application infrastructure exists
    - Common window base class, application index, …
  - Some applications have been written in this framework
    - Entire SDA tool suite
    - Generic/diagnostic applications – parameter page, data logger plotting, device database access, …
    - Some machine specific applications exist -
  - Controls will continue migrating generic applications it is responsible for
  - Migration of machine specific applications requires input from machine departments – plan under development
  - New framework considered when developing new applications

Migration Path

- Communication
  - No planned changes to TCLK, MDAT, Beam Sync
    - Adequate for all planned upgrades
  - The ACNET protocol serves the system well despite its age and proprietary nature
    - Java API emulates additional functionality that should be moved directly into the protocol
    - How to do this is under evaluation
    - Past R&D on CORBA; Java RMI.
      - VxWorks, and particularly PSOS are issues here

Sequencer

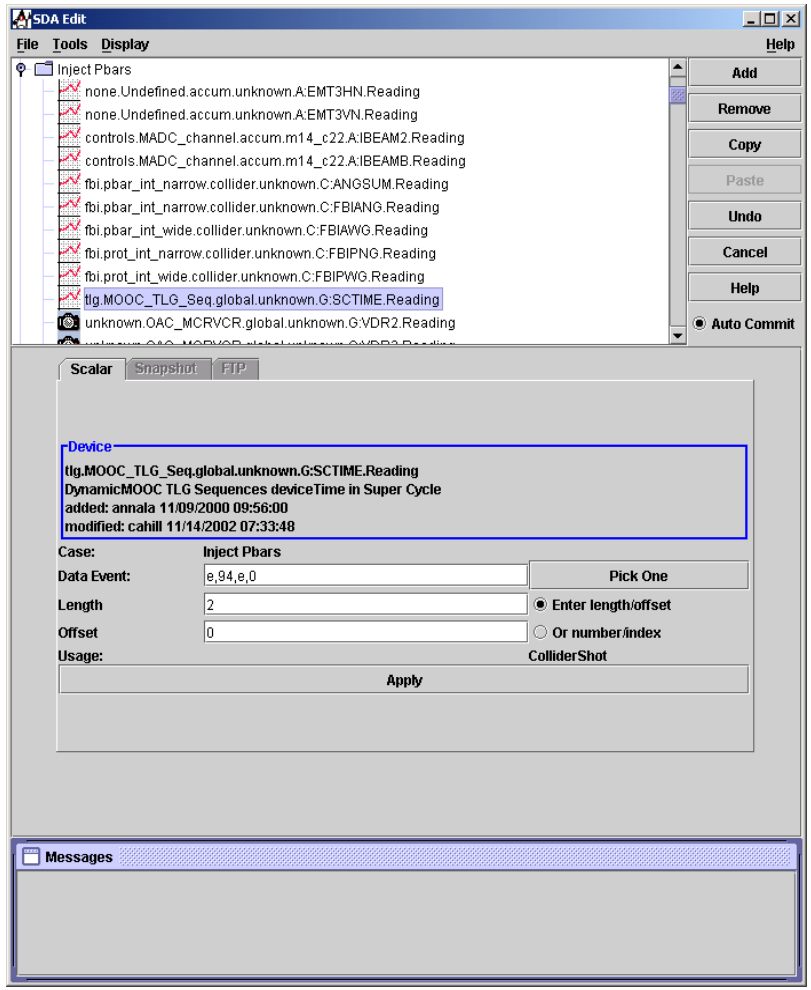
- Automatically configure and operate collider stores
- Steps defined in custom sequencer language
  - General sequencer command language
  - "Aggregate"'s of commands
  - Accelerator read/set commands ("Accelerator Control Language")
  - Start secondary applications for complex operations
    - Beam line tuning, pbar shot lattice, etc.
- Straightforward to build/modify/extend sequences
- Instances for TeV, MI, pbar during shots

Sequencer


PB:C48 SEQUENCER <NoSets>
C48 COLLIDER SEQUENCER 30-JUN-03 14:07:48 ◆Pgm_Tools◆
mode
edit
log
status
files
help
aggregate commands
::: Proton Injection tune up
::: Reverse Injection tune up
::: Proton Pilot
::: Inject Final Protons
ERR Set up Pbar Injection
::: Inject Pbars
::: Prepare to Ramp
::: Accelerate
::: Goto Low Beta
—> Initiate Collisions
::: Remove Halo
::: HEP store
::: Document store
ERR Turn off HEP
::: Un-Squeeze
::: Decelerate
ERR Goto Proton Inj Porch
::: ----------
::: Recovery
1:20 of 29
Document store
::: EDIT_COMMAND AUTO TRUNCATE ?
::: START_PGM PA1932 .
::: START_PGM PA0685 .
::: ACL TEVSCHOTVSA_GET_WTRACE1 .
ok ACL TEVSCHOTVSA_GET_WTRACE2 .
Messages
SEQUENCER: (mode 1) begins on console 131 slot PB

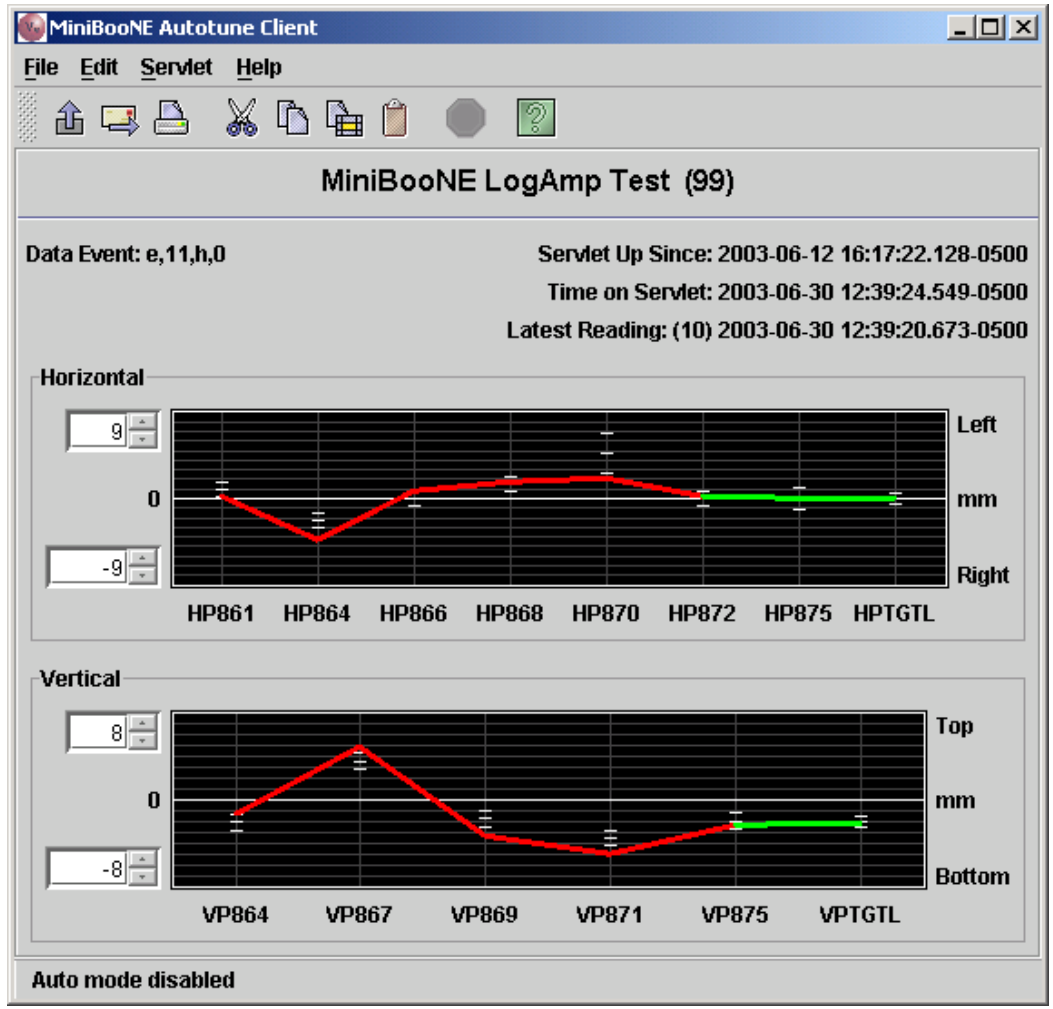
Sequenced Data Acquisition

- Automated acquisition of data for collider shots
- Configure readings for various stages of collider shot
  - Snapshots/FTP as well as single scalar readings
  - Easy to add new things
- Data acquired automatically and stored in Sybase database
- Summary html reports generated
- Suite of tools to view detailed data



Autotune

- Automatically tune transport lines for MiniBooNE
- Will also use for
  - 120 GeV FT
  - Electron Cooling
- Java application, same code base for all 3
- Read positions from BPMs
- Adjust magnets appropriately



Summary

- The control system continuously evolves to meet operational and maintenance needs
- Planned replacement of TeV/MI BPMs covers primary outstanding hardware maintenance issue
- New digitizer front-end under development
- Front-End infrastructure recently modernized
- Central Services migration from VAX/VMS to Java nearly complete
- Substantial Java based application infrastructure exists
  - Successful demonstration in SDA and other applications
  - Plan for migration of VMS applications under development