Beyond ACNET:

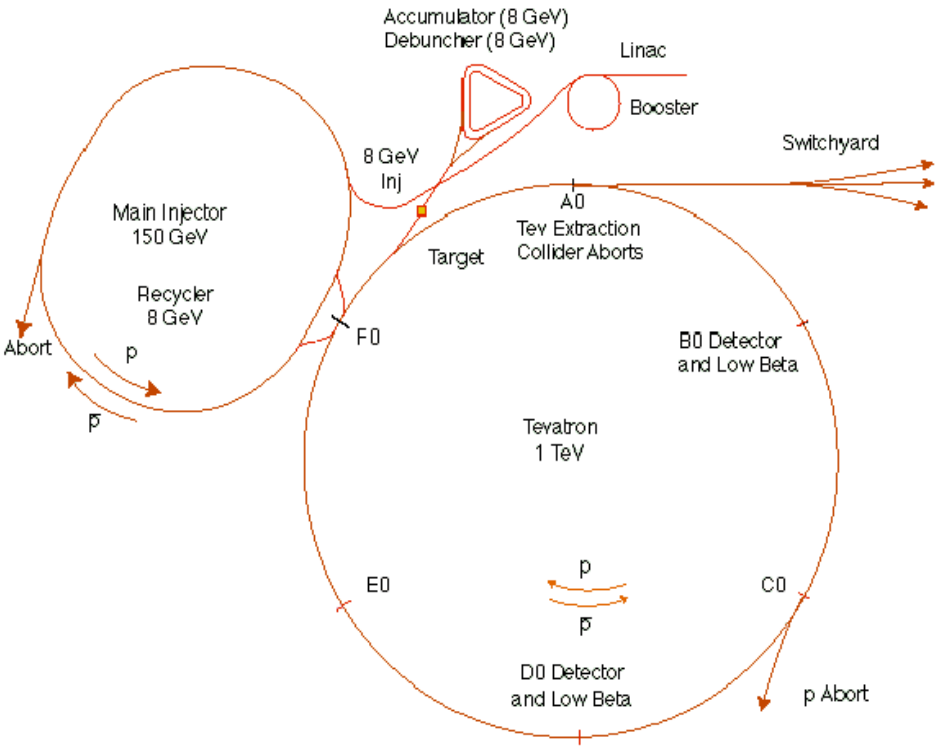
Evolution of Accelerator Control System at Fermilab

Andrey Petrov,
Fermi National Accelerator Laboratory
Accelerator Controls Department

Presentation at SLAC • March 17, 2009

The Origin of ACNET

- ACNET usually refers to the Accelerator Control System at Fermilab.
- The system was developed in early 1980's for the Tevatron.
  - Still based on the original architectural design.
  - The implementation has changed over time.
- Currently serves the entire Accelerator Complex.
  - Supports multiple operation modes simultaneously.
- Will likely stay in the future.



Fermilab Accelerator Complex

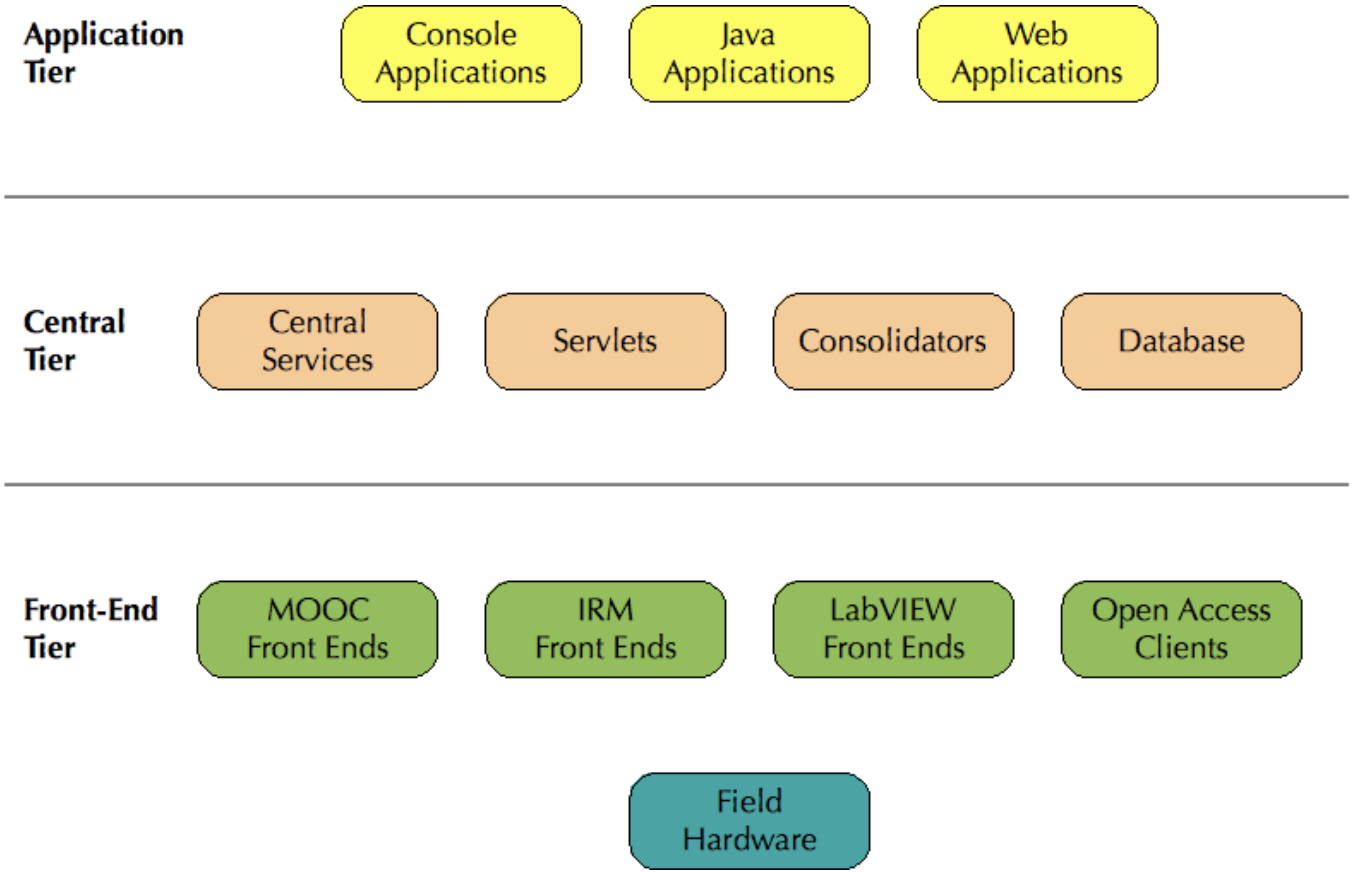
ACNET = ACcelerator NETwork

- In fact, **ACNET** is a suite of common application and transport protocols used in the Control System.
- Designed for effective real-time data acquisition.
- Example protocols:
  - **RETDAT/SETDAT** — RETurn DATa/SET DATa.
  - **Get32/Set32** — an extension of RETDAT/SETDAT.
  - **FTP** — Fast Time Plot, groups of readings @ ≤ 1.44 kHz.
  - **Snapshot** — Blocks of 2048 points.
- Based on:
  - DEC's Parallel Communication Link (PCL) — 1980's.
    - → IEEE 802.5 token ring.
      - → Ethernet, carried on top of UDP — 2000's.
- Currently implemented on Linux, VxWorks, and in Java.

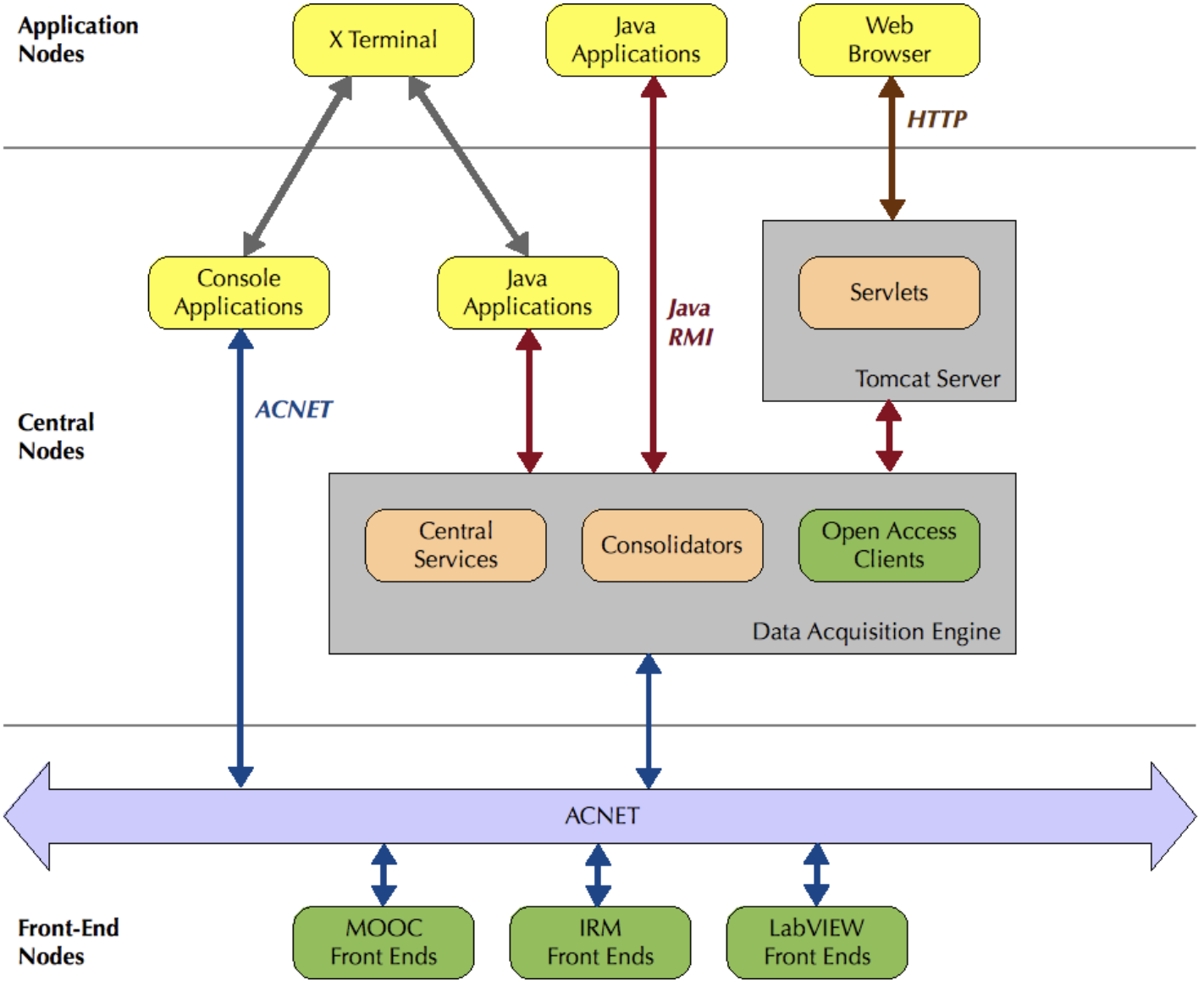
Device Model

- A flat namespace, each device name is ≤8 chars.
  - e. g., in `T:ERING` (*Tevatron Ring Energy*) `T` means the Tevatron and `ERING` designates a device within that subsystem.
- Each device may have up to 6 properties:
  - `READING`, `SETTING`, `BASIC_STATUS`, `BASIC_CONTROL`, `ANALOG_ALARM`, and `DIGITAL_ALARM`.
- Properties can be readable, writable, or both.
- A property value can be expressed in 3 scales:
  - `RAW`, `PRIMARY`, and `COMMON`.
- Each kind of the properties uses a pre-defined data type or an array of that type.
- Currently there are ~200,000 devices and ~350,000 properties.
- Recently implemented long (≤64 chars) devices names.

Logical Organization



Physical Organization



Front-Ends

- Acquire data from the hardware and field buses in response to timing signals. Provide mapping between the central device database and the actual hardware readings and settings.
- Use ACNET protocols to talk to the rest of the control system.
- 3 base front-end architectures:
  - **MOOC** (*Minimally Object-Oriented Controls*). ~275

    VME/VxWorks. Used in the Tevatron, Main Injector, and Pbar Src.
  - **IRM** (*Internet Rack Monitor*) & **HRM** (*Hotlink Rack Monitor*)

    VME/pSOS/VxWorks. Used mostly in Linac. ~185
  - **OAC** (*Open Access Clients*, a.k.a. *Software Front Ends*). 131

    Java-based. Run on central nodes.
- Also there are ~25 LabVIEW front ends, mostly in various instrumentation subsystems.

Data Acquisition Engine (DAE)

- A custom Java application server. [123]
- Provides:
  - Data consolidation.
  - A front-end façade for applications.
  - Central services.
  - Open Access Clients (software front-ends).
- Talks ACNET protocols to the front-ends.
- Talks Java RMI to the applications.
- Runs on uniformly configured nodes.
  - Now Solaris and Scientific Linux.
  - Solaris will be replaced with Linux soon.
  - Co-located with Tomcat and MySQL (for data loggers).

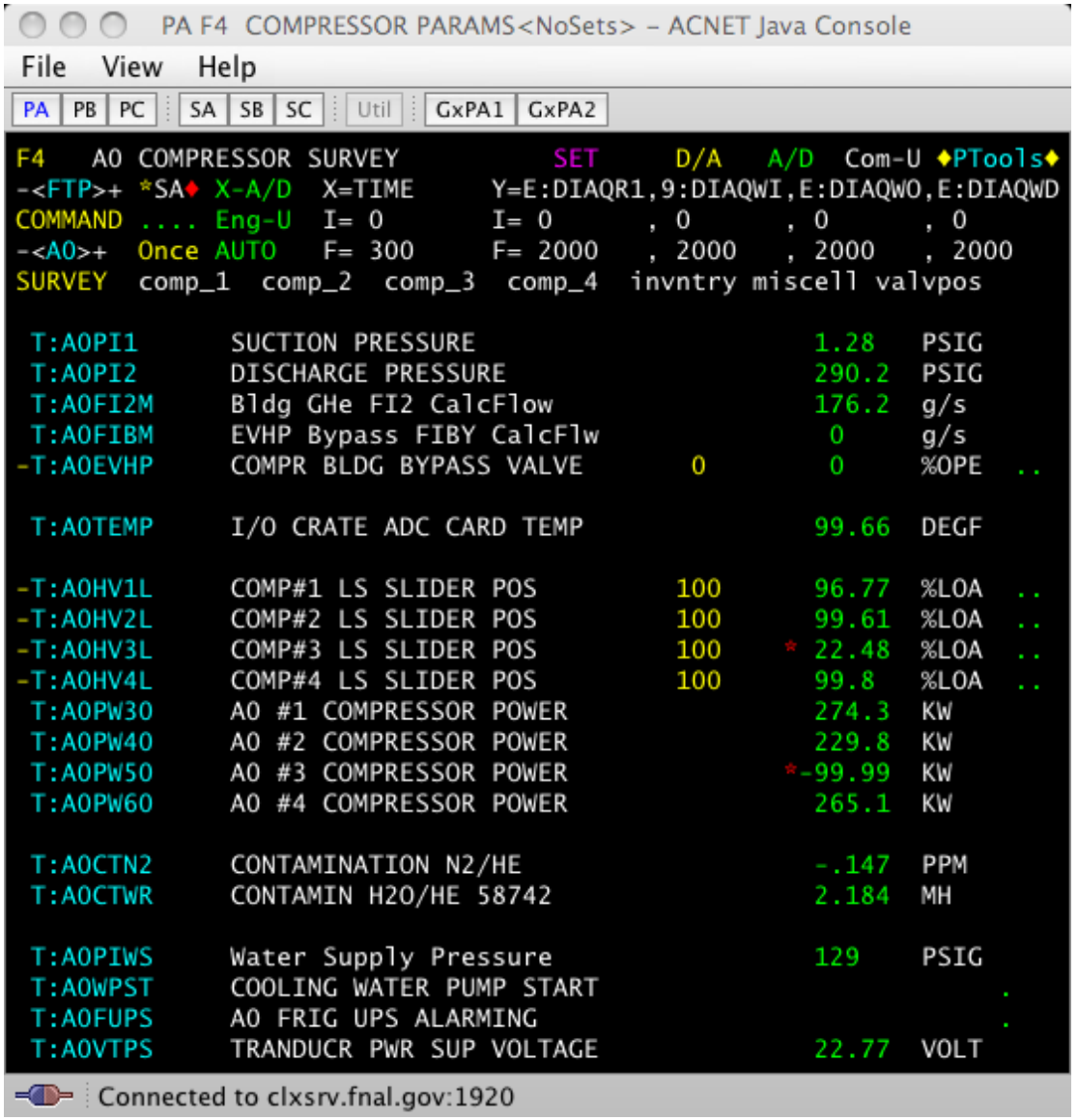
Central Services

- Data Loggers. ~70
  - Each data logger: 60 devices, $5 \cdot 10^8$ total points, (1446 days @ 15 seconds). Local MySQL tables.
- Sequenced Data Acquisition (SDA).
  - A complex data logger saving data selectively, depending on the machine state.
- Save/Restore.
  - Automatically saves the system's state 4 times a day.
- Alarms.
- Front-End Download.
- Accountability.
  - Logs all settings done in the control system.

(Legacy) Console Application Environment

- ~600 applications written in C++.
  - Each application is usually for a single purpose.
  - The framework is simple and robust.
    - ※ Yet, considered outdated by many – especially the GUI.
  - This is currently the main tool to operate the accelerators.
- May run only on central nodes, currently Linux. ~70
  - Access via X terminal.
  - Access via a lightweight read-only Java client.
- 1 client = 3 text pages + 3 (9?) graphical screens.

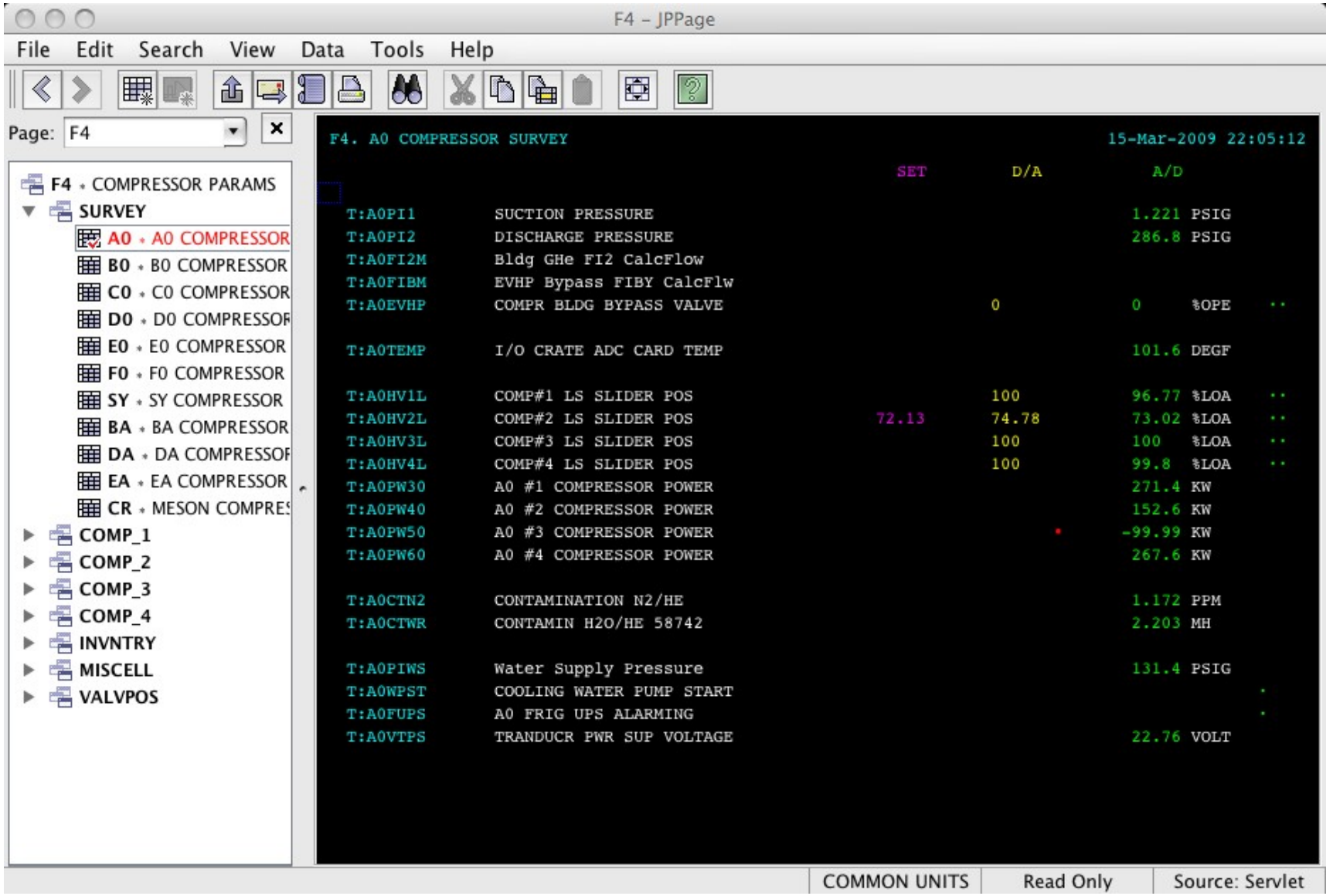
Example of a Console Application



Java Application Environment

- ~ 130 applications.
  - Again, each application is usually for a single purpose.
  - There is a custom application framework that facilitates development of GUI and implementation of principal functions.
    - ※ Yet, considered too complicated and not quite functional by many.
  - Used not as widely as the legacy environment.
- May run on client machines, as well as on central nodes.
  - Access via X terminal, if on a central node.
- Applications are launched by Java Web Start.
  - The main complaint – too slow.

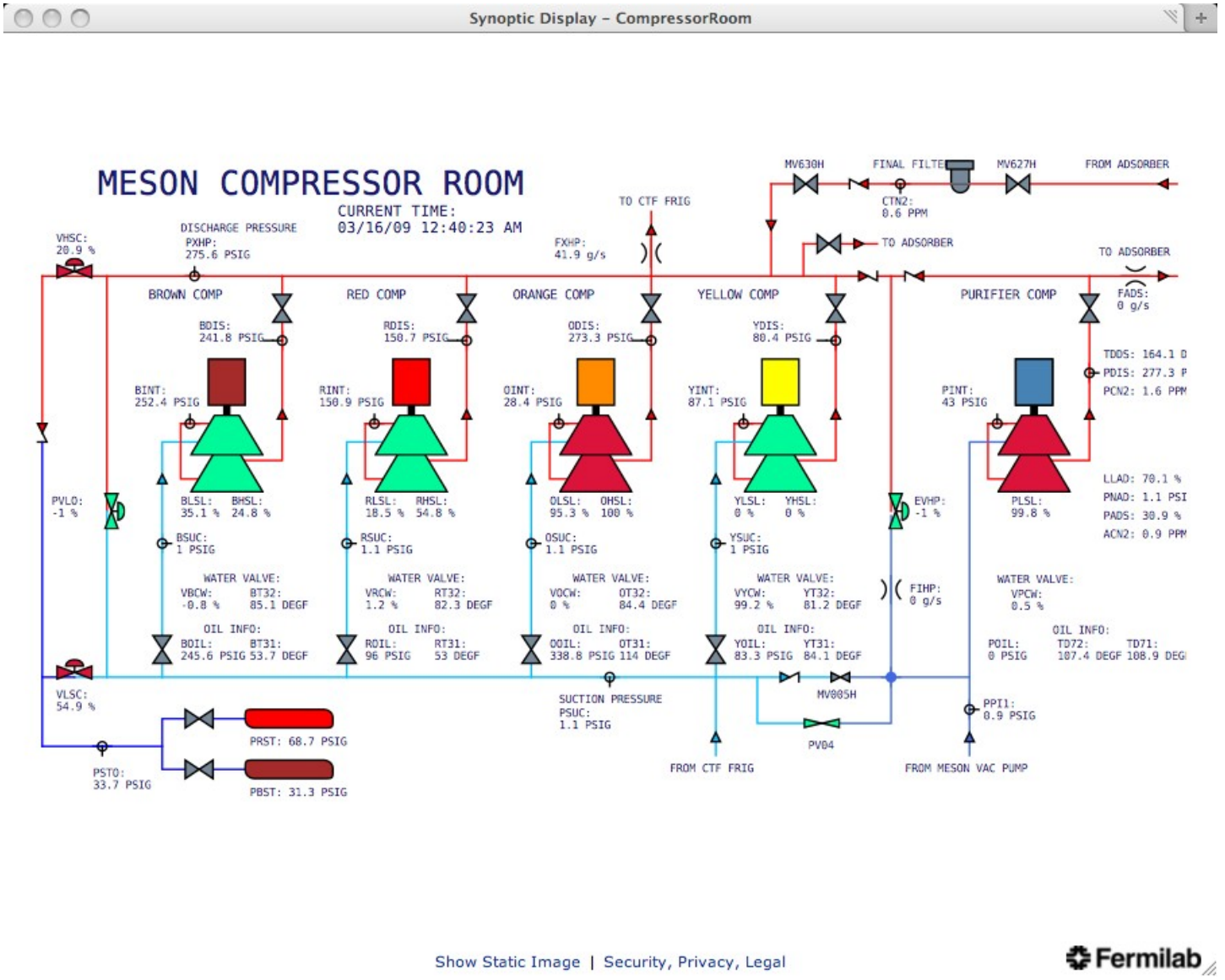
Example of a Java Application



Web Applications

- ~ 50 applications.
- Organized as servlets running in Tomcat containers.
- Serve a variety of purposes:
  - Convenient way for the users to access data.
    - e.g., Synoptic, SDA Viewer, Supertable.
  - Lightweight protocol for client programs.
    - XML-RPC and REST.
  - Viewing and editing data from the database.
    - Java Application Index, ACNET Device Database (D80).
- Kerberos authentication in the browser via a custom applet.

Example of Web Application

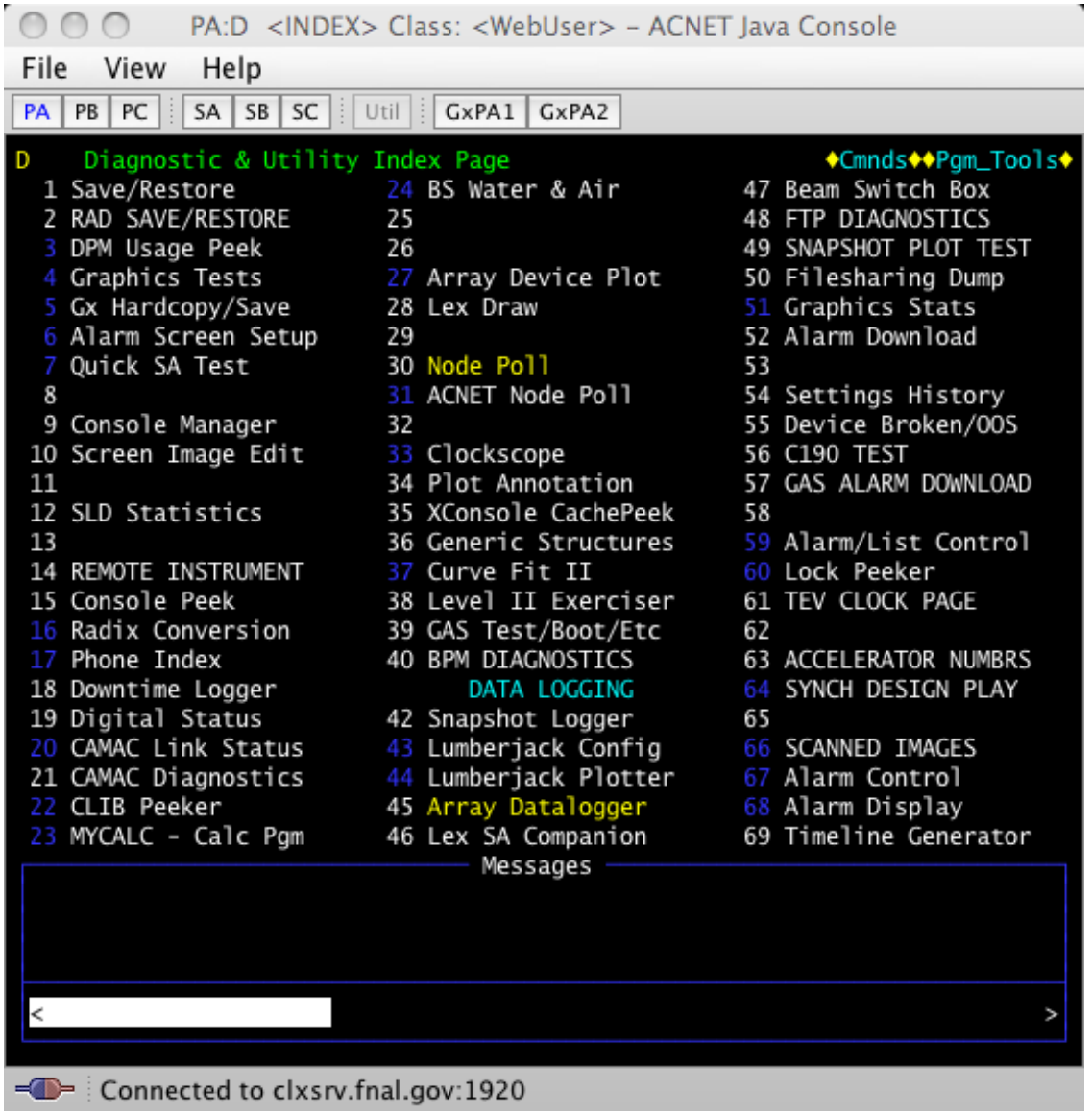


Console Apps. vs. Java Apps.

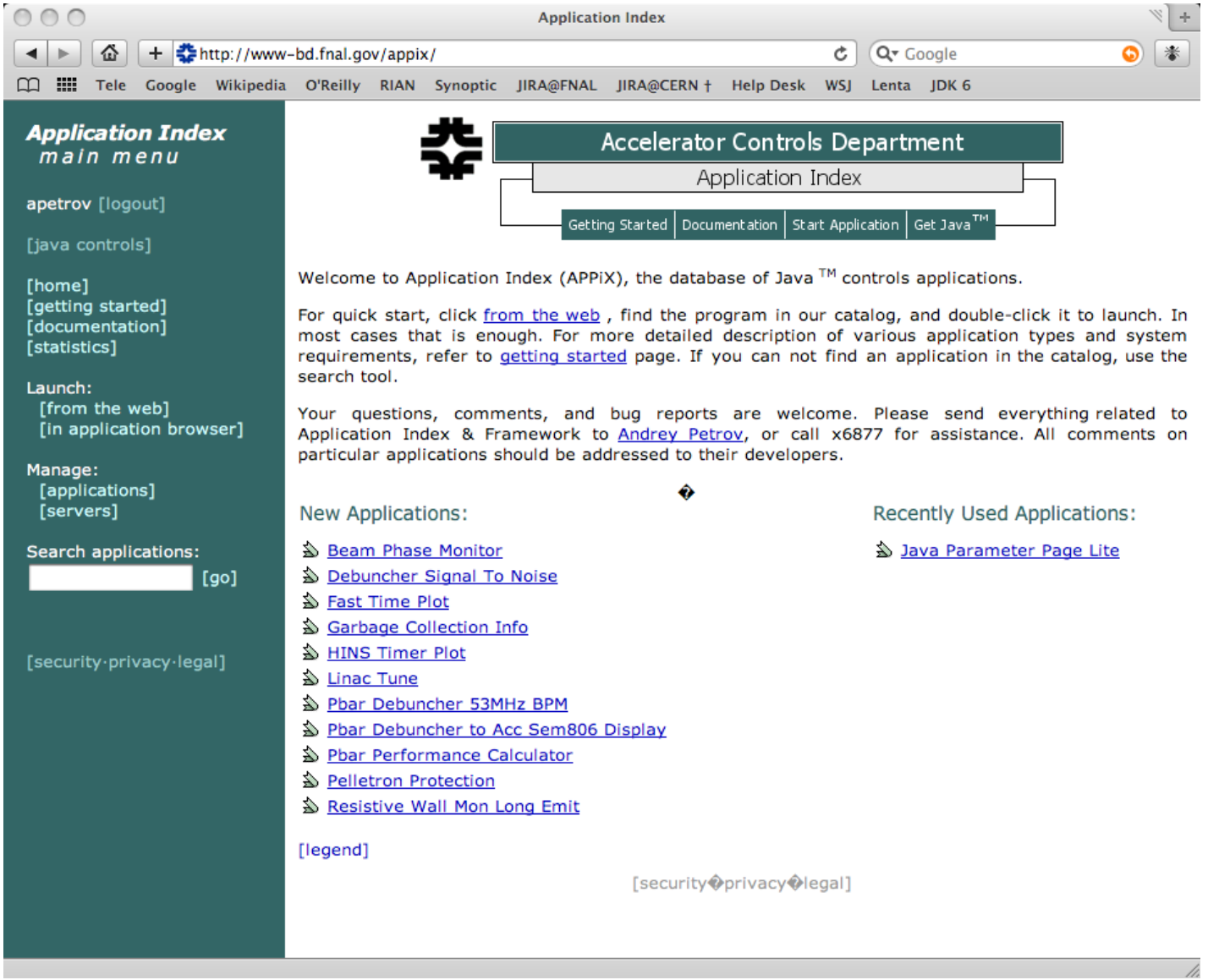
- CVS Code Repository.
- Command-line building system (MECCA).
- Applications usually acquire data via ACNET protocol.
- Applications always run on central nodes.
- Applications are started from Index Pages.
  - Now capable to start Java applications, EDM screens.

- CVS Code Repository.
- A Web-based building system.
- Applications usually acquire data from DAE via Java RMI.
- Applications usually run on client nodes.
- Applications are usually started from the web.

An Index Page



Application Index



Principal Applications

- **Parameter Page**

  The most universal program that controls a list of devices.
- **Fast Time Plot**

  Plots device data in real time @ ≤ 1.44 kHz.
- **Snapshot Plot**

  Plots device data @ ≤ 20 MHz.
- **ACL** (*Accelerator Command Language*)

  A data acquisition scripting tool.
- **Sequencer**

  The primary program for coordinating operations of the accelerator complex. Uses ACL.
- **Lex SA** (in Console Framework) / **Synoptic** (in Java)

  Synoptic graphics programs.

Status of Java at Fermilab Controls

- The platform was introduced back in ~1996. Java applications were to be the path away from VMS.
- Yet, after 10 years Java applications didn't replace the legacy framework:
  - (For many) the programmatic API is too hard.
    - Obviously, programming Swing with dynamic layouts/events/etc. is very hard for an occasional programmer.
  - (For many) the development environment is too hard.
  - Development of 2-tiers applications is harder than 1-tier.
  - Startup time is perhaps the biggest annoyance for the users.
  - Java RMI–related issues: unmarshaling, local callback observers.
  - Participation from machine departments didn't happen.
  - Only new applications, which don't have a counterpart in the legacy system, get eventually adapted by the users.

Status of Java, continued

- Meanwhile, the middle tier is now implemented in Java.
  - Performance, in general, is seemed to be satisfactory.
    - ※ Now moving from Sun Netras to i386 Linux.
  - A lot of issues with erratic behavior of garbage collection.
  - Subtle changes between JVM versions may cause different behaviors of Data Acquisition Engines.

Linux Migration

- Prior to 2007 the legacy part of the control system ran on VAX/VMS platform.
- It seemed to be unlikely that ~500 existing applications would ever be rewritten in Java.
- In 2004-2006 the system was migrated to Linux.
  - All non-obsolete VMS applications were ported.
    - Things like VMS mail application were left behind.
  - No changes to the existing look-and-feel.
  - No specific changes to behavior of the applications.

Emergence of EPICS

- *As a part of the International Linear Collider (ILC) R&D Program, Fermilab has constructed new facilities for testing superconducting RF technologies for a future accelerator.*
- In 2005 it was decided to use EPICS and DOOCS control systems at ILC Test Area (ILCTA), and install them independently from the main ACNET system.
  - More lightweight.
  - Better collaboration with other institutions.
  - Using existing hardware developed elsewhere.
- Over time, as ILC R&D has slowed down, ILCTA was converted for Fermilab's *Project X* (a new superconducting linac).
- An instance of the EPICS control system is currently in place and running.

Status of EPICS at Fermilab Controls

- Now the two control systems are mostly independent.
- EPICS PVs can not be accessed from ACNET applications.
- EDM screens can be started from the ACNET index pages.
- EDM screens can display both PVs and ACNET devices.
- EPICS timing signals are available in ACNET.
- EPICS will stay in the future as a subsystem.
- It will be possible to read EPICS PVs through ACNET.
  - ACNET-aware IOCs.
  - A programmatic EPICS-to-ACNET bridge.
- EPICS users will be able to take advantage of the existing central services and applications.

Future of Fermilab

- The Tevatron will eventually be shut down (so far, 10/2010).
- The complex will be upgraded to increase the intensity delivered for the neutrino program.
- New NOvA experiment is expected to run until the late 2010's.
- Existing superconducting RF test facilities:
  - HINS – *High Intensity Neutrino Source*.
  - NML – *New Muon Lab*.
- Project X – a new 8 GeV superconducting linac.

Future of ACNET

- ACNET will be used as a foundation for a new control system in the post-Tevatron era.
  - A single system for the entire accelerator complex.
- Full integration with EPICS front-ends.
- Replacement of obsolete hardware.
- Modernizing legacy and Java application environment.
- A number of issues (some are under investigation now):
  - Better GUI framework and infrastructure.
  - Better client application protocols.
  - Better data pool managers (in the middleware).
  - Operating with complex data structures, self-describing data sets.
  - …

References

- Fermilab Accelerator Controls Department,

  `http://www-bd.fnal.gov/controls`

- Accelerator Division Document Database,

  `http://beamdocs.fnal.gov/AD-public/DocDB/DocumentDatabase`

- An overview of existing control systems,

  `http://www-bd.fnal.gov/icfabd/Newsletter47.pdf`

- Project X Control System Requirements,

  `http://beamdocs.fnal.gov/AD-public/DocDB/ShowDocument?docid=2934`