Tcl at the NSCL:

A (30? well maybe 15) year
forward looking uh…retrospective

Staff of the National Superconducting Cyclotron Lab…
…and the Facility for Rare Isotope Beams.

What I'm gonna talk about

- What is the NSCL – what do we do.
- What sorts of toys we now have at the NSCL
- What new toys are being built (Rea3/FRIB)
- History of the use of Tcl/Tk at the NSCL
- The characteristics of Tcl/Tk usage at the NSCL
- Some speculative work with Tcl/Tk that's been done recently (super WIP).
- The outlook for future use of dynamic languages in Nuclear physics at FRIB and Tcl/Tk specifically.

Disclaimer: I am no longer a practicing physicist

NSCL

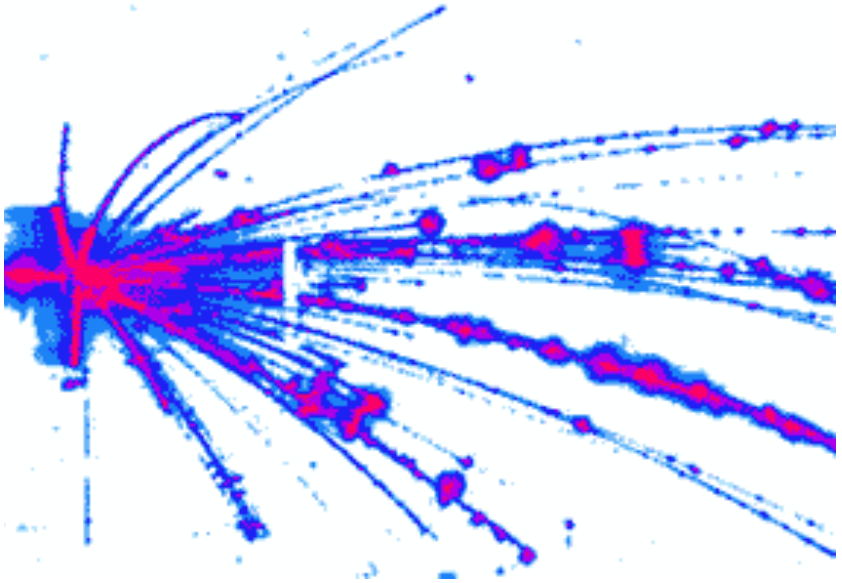
- NSF Funded laboratory
- Basic research in Nuclear Physics
- Specialization in Rare Isotope Reactions
- On the campus of Michigan State University
- Operating since 1961 (K-50 cyclotron)
  - Upgraded to K500 cyclotron (1982)
  - Upgraded to K1200 cyclotron (1989)
  - Coupled K500/K1200 operation (2000)
  - Upgrade in progress now.
- About 500 employees..and counting (see job board).
- Primary mission education
  - #1 Nuclear physics grad program in the U.S.
  - Community education outreach programs.

2004

2009

Sept. 20, 2011

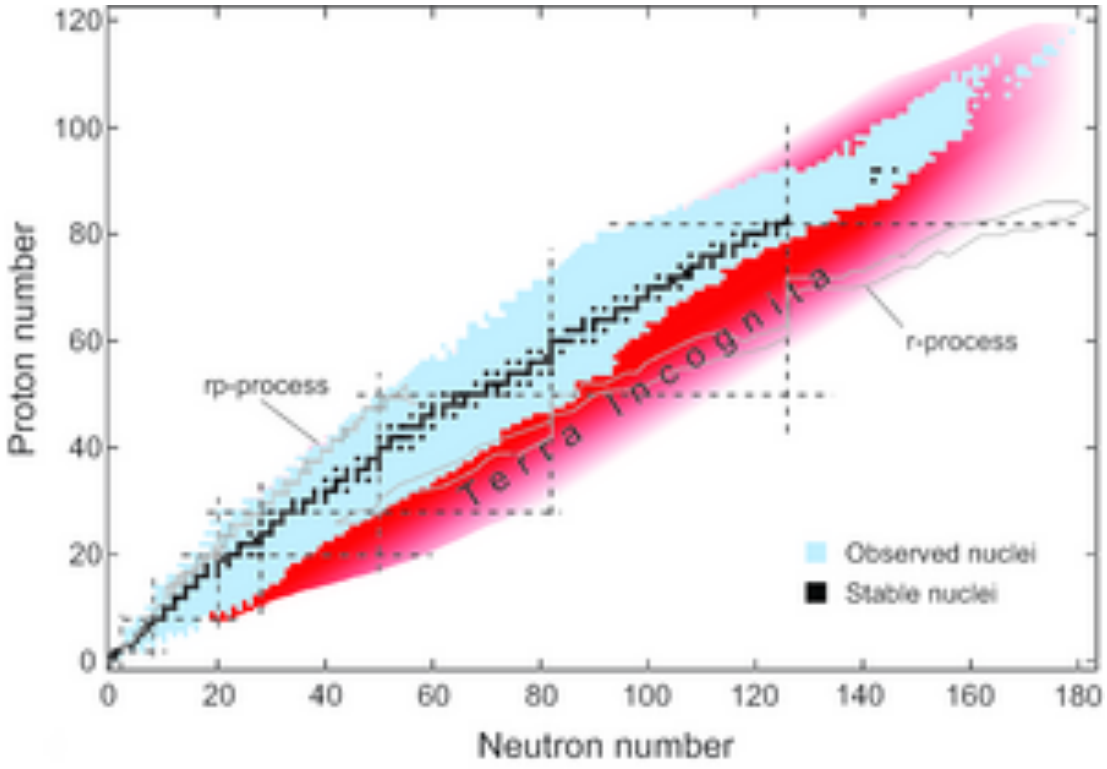
Nuclear Physics



'We Smash Stuff!"

- Zach Constan NSCL Outreach Coordinator

Rare Isotope Physics What

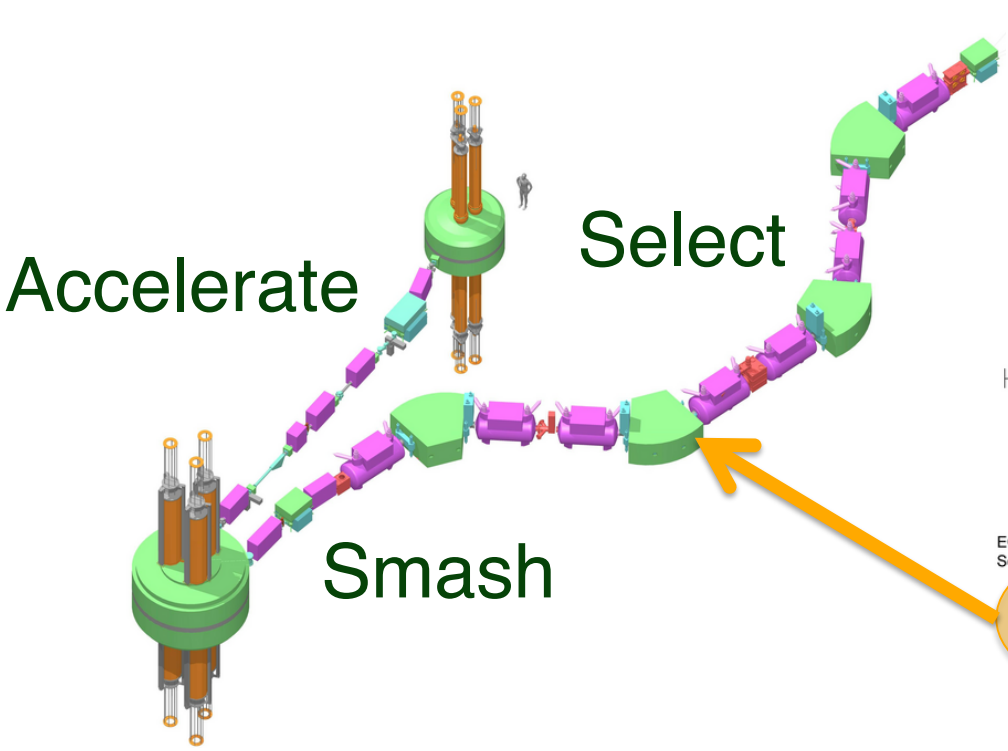


Do experiments with the blue and red nucleii

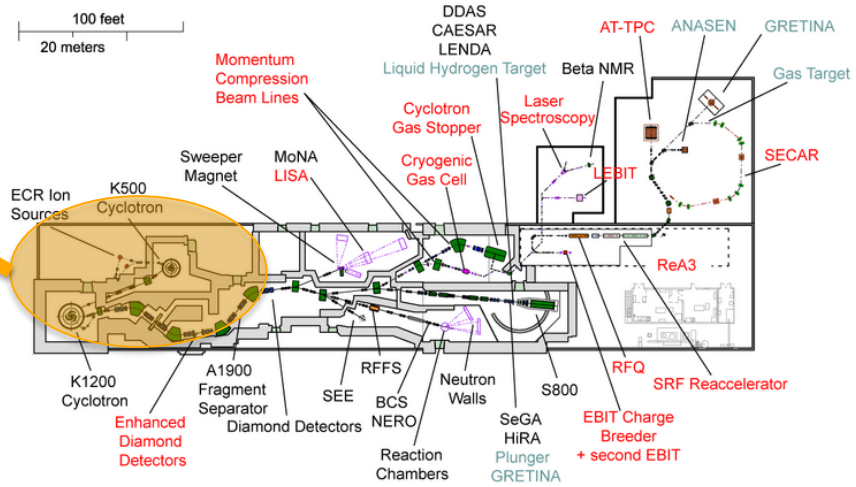
Note: r and r-p process produce most elements heavier than iron.

Rare Isotope Physics How

We smash stuff twice! (Projectile Fragmentation)



Transport to experiment where we smash again

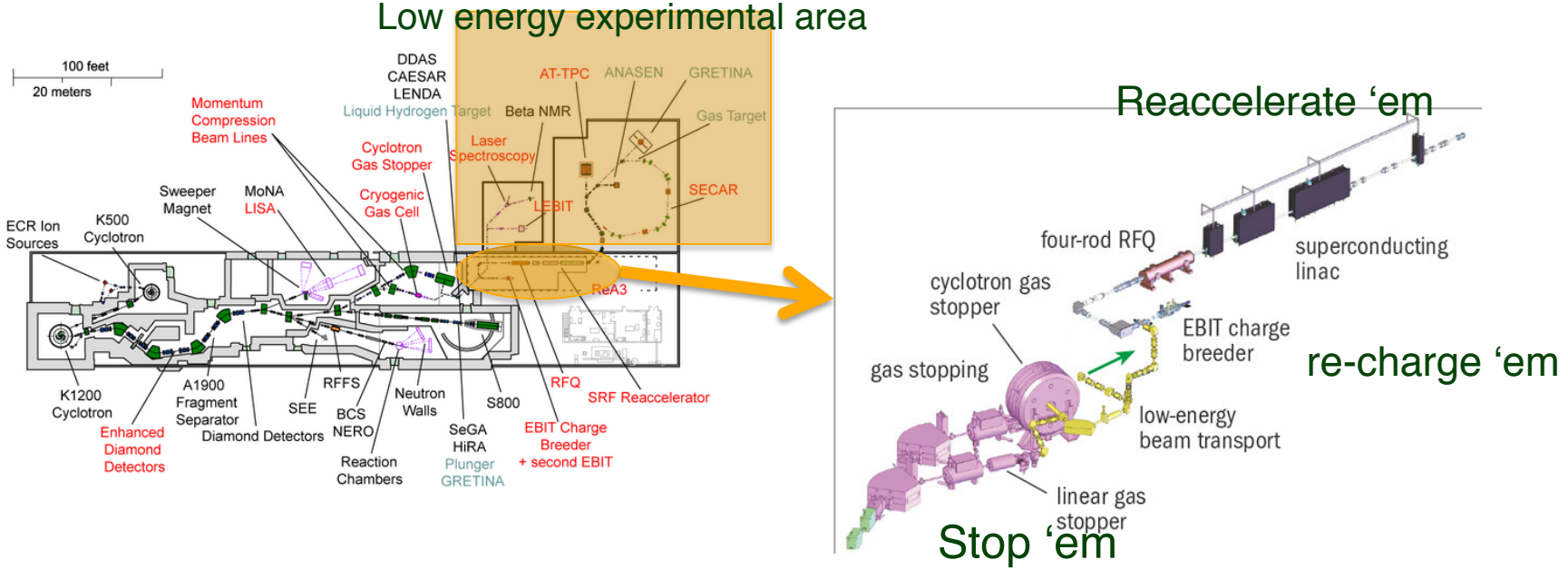


Rare Isotope Physics How II (coming 2012 to an NSCL near? you)

But wait:

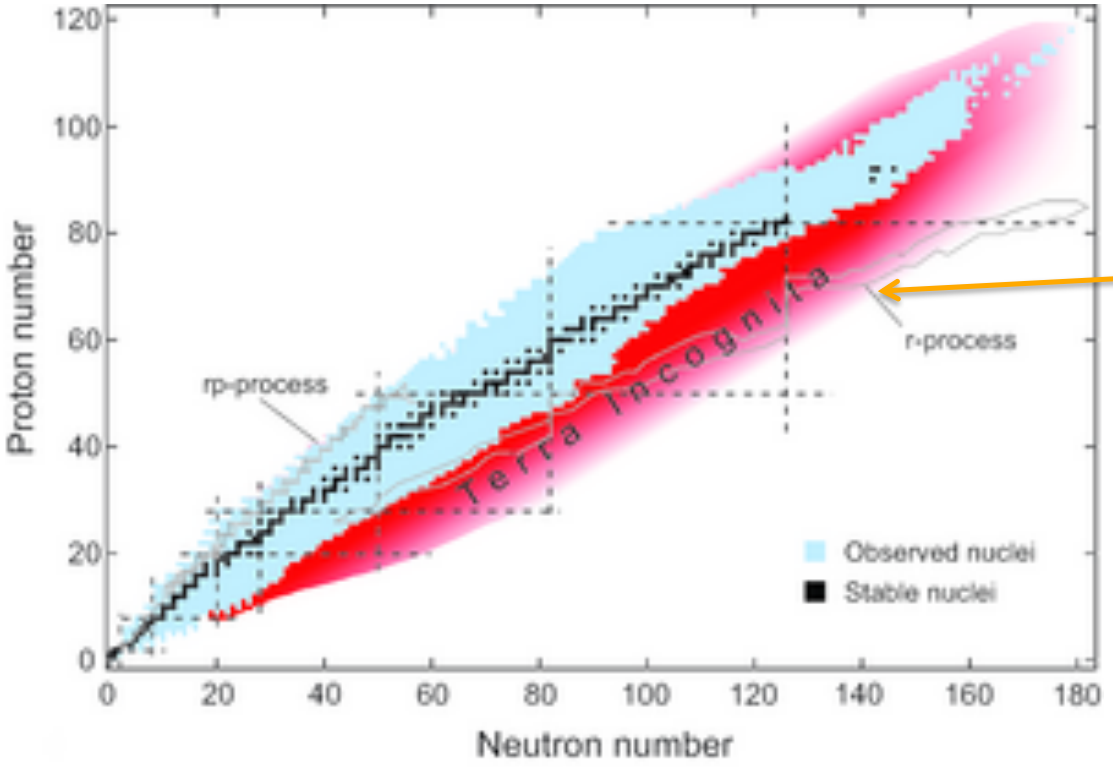
- To get good production rates we need ½ speed of light projectiles… or even faster (50MeV/A or higher energy)!
- But there's some interesting physics at much lower energies (100's of KeV – 5 MeV).

(Things just don't fwoosh around that fast even in very dense stars).



Rare Isotope Research futures
Facility for Rare Isotope Beams (FRIB)

did you notice that extra logo at the bottom of all the slides?



R-Process –
Rapid neutron capture followed by β^- decay -> stable high Z nuclei Happens in core collapse supernovae(?)

r-Process is supposed to be happening in isotopes we can't make at the NSCL yet….

FRIB
Exploring Terra incognito starting 2018+

http://frib.msu.edu : "The Facility for Rare Isotope Beams (FRIB) will be a new national user facility for nuclear science, funded by the Department of Energy Office of Science (DOE-SC) Office of Nuclear Physics and operated by Michigan State University (MSU). FRIB will cost approximately $600 million to establish and take about a decade for MSU to design and build."

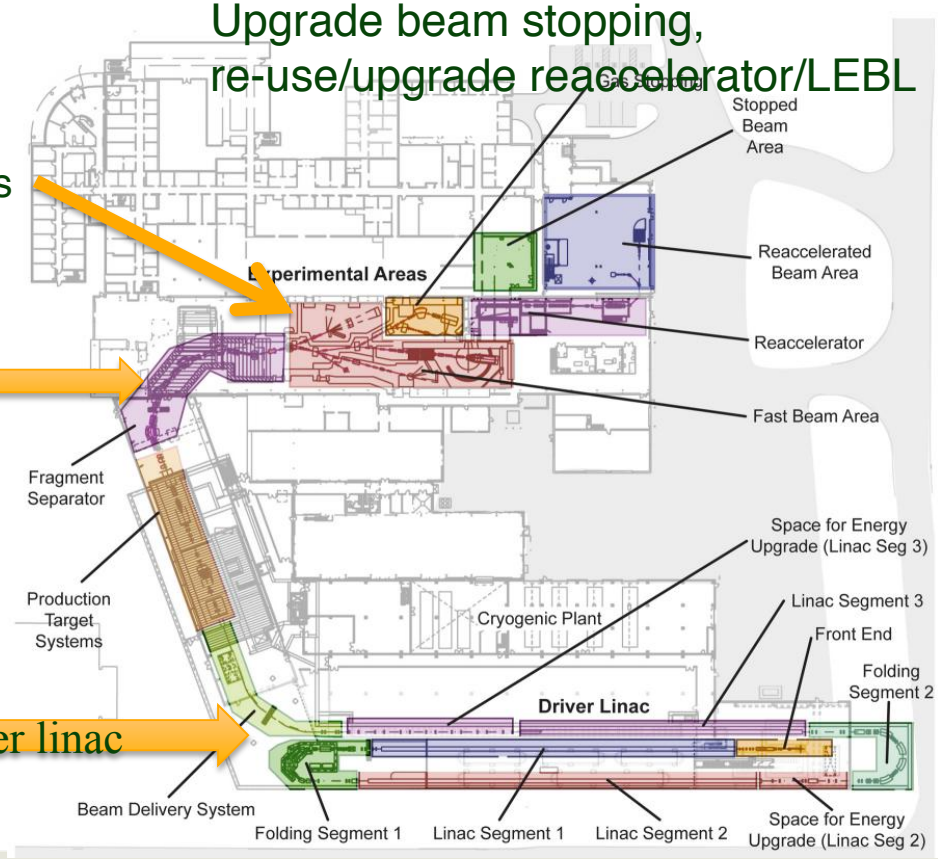
Upgrade beam stopping,
re-use/upgrade reaccelerator/LEBL

Re-use high energy beamlines

Take out the cyclotrons/
Reorient/upgrade the 'selector'.

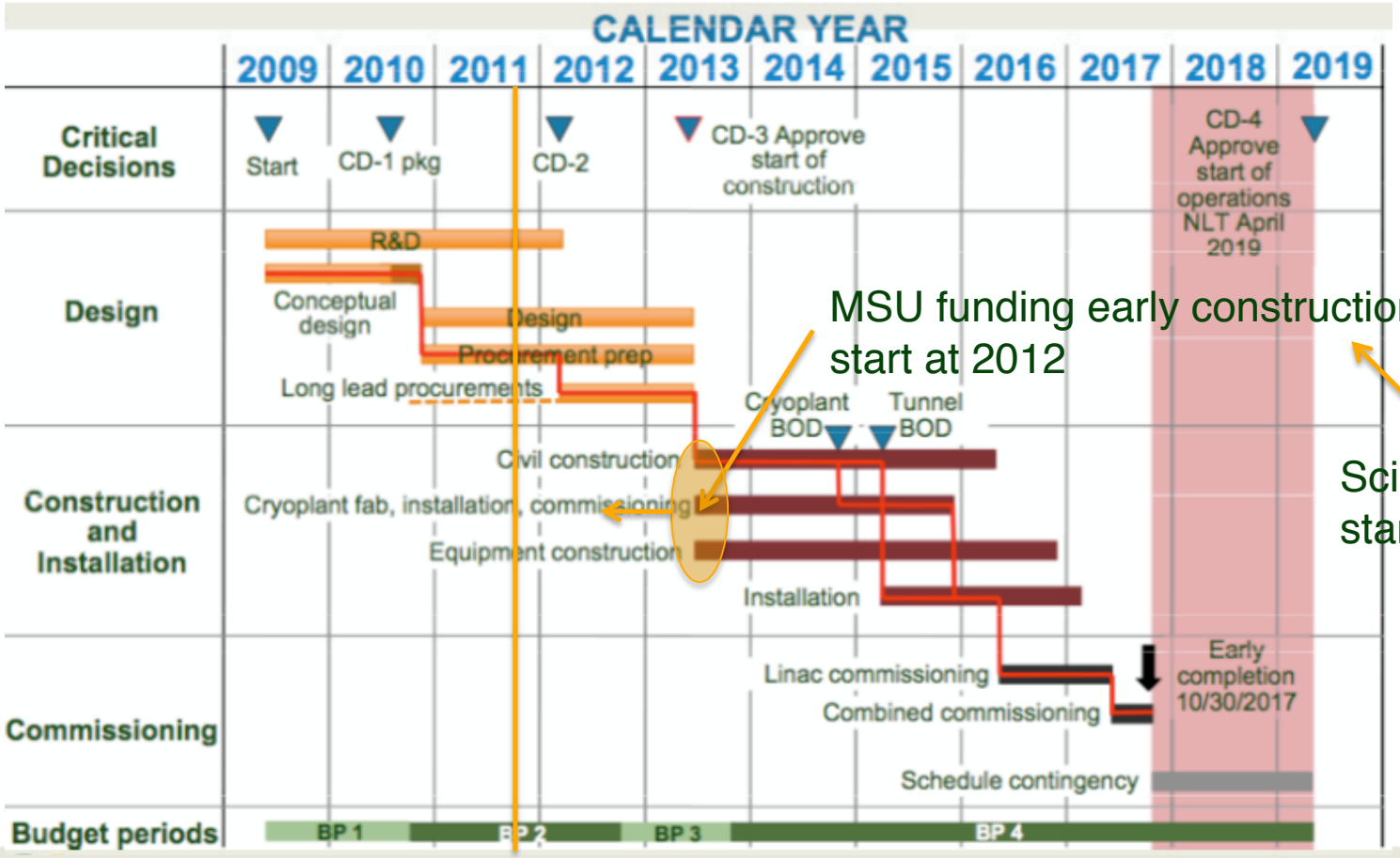
From 50MeV/A – 200-400MeV/A
Much higher intensities: 400KW
beam on production target (1GW/in^3)!

Driver linac



FRIB Timeline/plan

CD items: Cricital Decision points: Review that must be passed else DOE can cancel the project. CD2 will likely happen early (late this year).



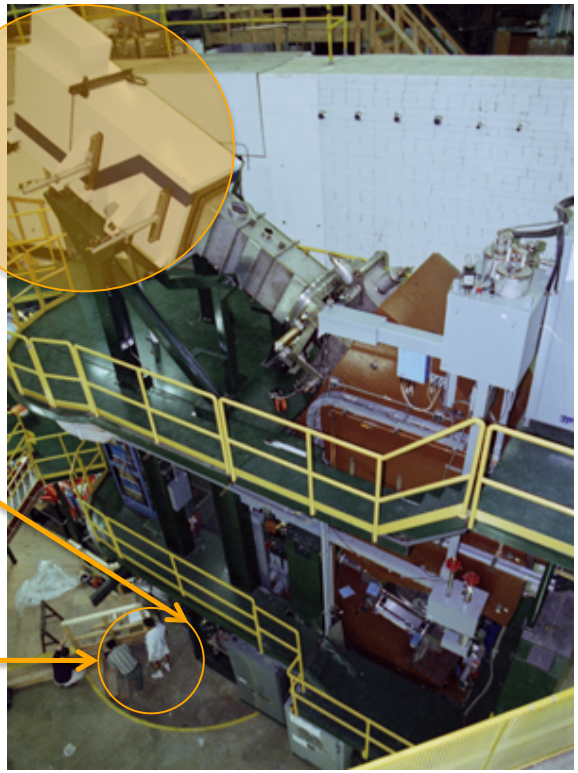
We're Almost to the Tcl stuff

S800 Spectrograph
(1996)

Focal plan
detector pkg.

Experiment
Target+ detector
Pkg.

Human beings
for scale



- DAQ at the target
- DAQ at the FP detector
- Needed a way to control both of them.

Step back 1 year: NSCL Hosts "RT-95"

Fermilab DART Run Control
G. Oleynik et al.

"…We felt that the group multicasting technique mapped very well on to data acqusition control…

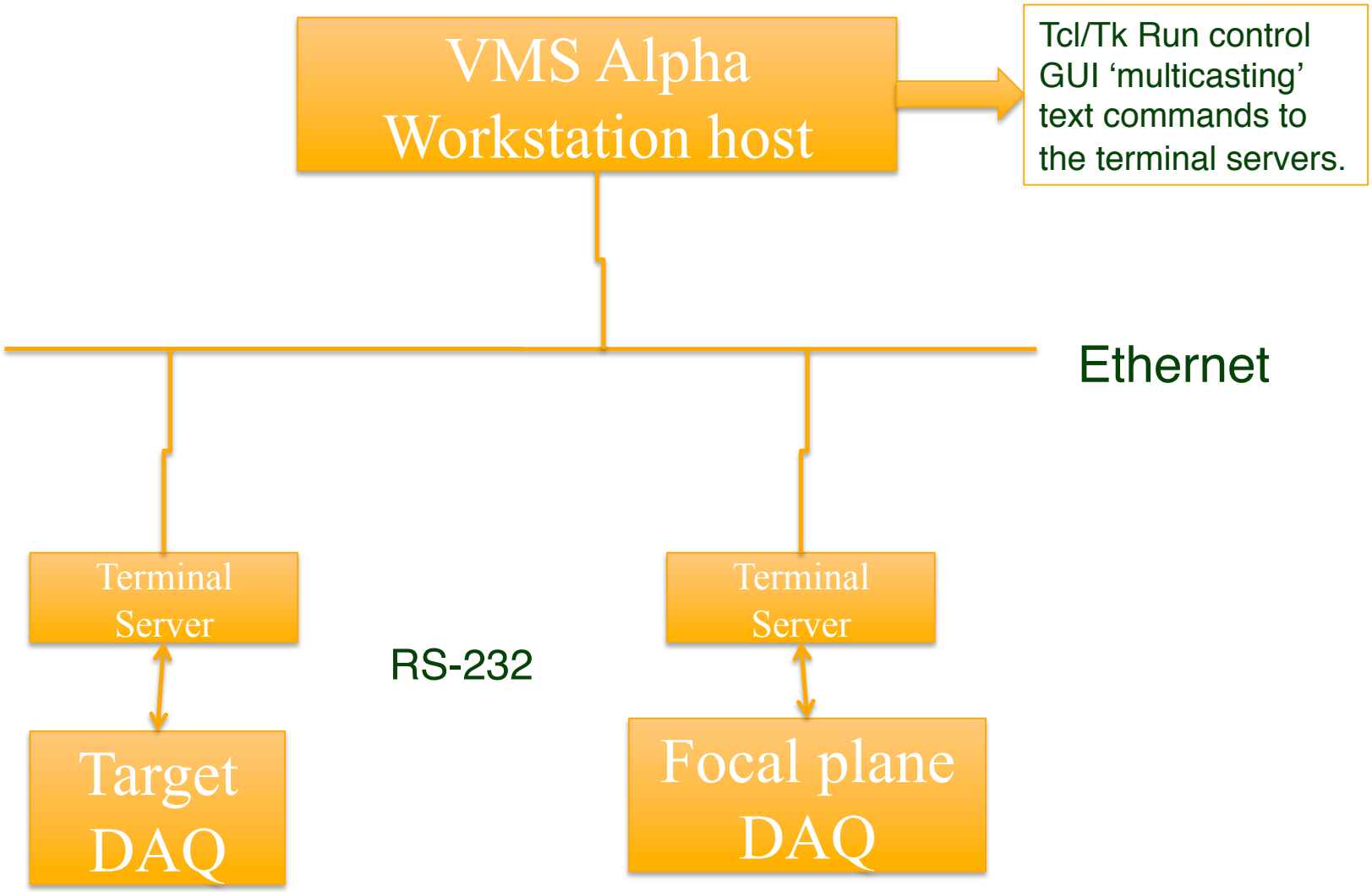
The commands that are multicast are formatted as TCL verbs, which are basically text strings…

We chose TCL because of its extensible interpretive procedures. For graphics, we chose TK…our experience has been that interfaces can be built more quickly with TK than from X…or Motif…

The ocp GUI…took on the order of ½-1 hour ..We feel this is a big success of the TCL/TK approach."

(Capitalization choices for Tcl and Tk above are from the original paper)

Run control for S800 DAQ:



1996 - Present

~~I am pleased to say that from 1996 until present we experienced an explosion in the use of Tcl/Tk at the NSCL~~

But I'd be lying.

Development of Coupled Cyclotron and Rare Isotope Capabilities.

- At this time the NSCL was a 'stable beam' facility (we smashed things once).
- 1994 NSCL proposed an upgrade:
  - Couple our existing cyclotrons to get higher intensities and higher energies.
  - Add a radioactive beam capabilities via projectile fragmentation (at that time a novel technique), by building an A1900 mass separator.
- 1995 NSAC publishes priorities for a new long-range plan for nuclear science:
  - Immediate funding of NSCL upgrade
  - Running the existing cyclotrons stand-alone until 1999
  - Shutdown to complete the upgrade July 1999-May 2001
  - Begin coupled operations and radioactive beam production in July 2001 (smash things twice).
- So it was written (funded) by the NSF
- So it was done by the NSCL.

1996-2001 other developments

- The ‘usual’ explosion in computing power.
- Explosion in network bandwidth.
- Explosion in disk capacities.
- Increasing availability of FOSS software suitable for production use:
  - Linux
  - gnu compiler suite.
- Nuclear/HE physics community ditches DEC/VMS in favor of Linux/ Unix:
  - Digital Equipment Corp gets increasingly out of touch with its ‘science support roots’ as it attempts to play in the business market.
  - Digital Equipment Corp bought up by Compaq (1998).
  - Bought later by HP (2002).
  - …besides we wanted to get out of the single vendor lock-in.

1996-2001 other developments

Experiments become larger. Electronics more complex.

Mandate to NSCL Software Group (1998-ish).

- Create NSCL Data Acquisition system that can run on Unix/Linux.
- Make it easy for outside users to use.
- Provide analysis tools for Unix Linux
- Make the system easily extensible and adaptable to all NSCL experiments.
- Make them easy for outside and inside users to use.
- Make it easy for outside and inside users to use
- Make it easy for outside and inside users to use
- Make it easy for outside and inside users to use…..

Several Decisions early on:

- Base software (performance critical) in C++
- Build application frameworks not applications where applications depend on specific experiment details.
- Applications that must be controlled by users will embed a Tcl interpreter.

Why Tcl?

- Experimenters were already going to have to make one language transition (Fortran -> C++/C). Reduce effort to learn how to control sw.
- Decided Tcl was a simple transition for a control language
  - Experimenters were already used to command driven programs and Tcl is just a command language with very simple syntax.
  - Simple syntax makes scripting approachable with very little investment in effort.
  - Tk would be something they could get into later.
  - Extending the interpreter even later.
- Other options considered:
  - PAW (CERN package based on a FORTRAN interpreter) :
    - » Fairly wide community acceptance (pro)
    - » Steep learning curve (con)
    - » Retains FORTRAN presence (con)
  - ROOT (CERN package emerging at the time based on C/C++ interrpreter)
    - » Was starting to gain community acceptance (pro)
    - » *Very* steep learning curve.
    - » Not yet really stable.
  - Python
    - » OO approach (pro)
    - » OO approach (con – at the time).

Because it was then and is still the best scripting language out there.

NSCLDAQ and NSCLSpecTcl

NSCL 1999 Annual Report Article

http://groups.nscl.msu.edu/nscl_library/pub/annual_reports/1999/fox_deployment.pdf

DEVELOPMENT STATUS AND DEPLOYMENT OF THE NEXT GENERATION NSCL DATA ACQUISITION SYSTEM
R. Fox, E. Kasten

“Components we provide are often used in ways we did not anticipate. This is a good thing. We intend to use the Tcl/Tk scripting language as a base command language for all components of the system. This allows us to support run-time extensions of the functionality of the software and its user interface via Tcl/Tk scripting. It also allows support for compile time extensions of the command set via C++ wrapper classes around the Tcl command registration procedures. Tcl/Tk scripting provides a common basis for automating tasks within the data acquisition system. The Tk component provides powerful GUI creation and modification tools available to all interactive components”

http://groups.nscl.msu.edu/nscl_library/pub/annual_reports/1999/fox_spectcl.pdf

STATUS OF THE SpecTcl DATA ANALYSIS PACKAGE
Ron Fox, Chase Bolen, Jeremy Rickard

“SpecTcl and Tcl’s power is that it provides a simple language in which very complex operations can be represented. It provides a consistent base language on top of which application specific extensions can be layered.”

1999-Present

I am pleased to say that from ~~1996~~ 1999 until present we experienced an explosion in the use of Tcl/Tk at the NSCL

How is Tcl/Tk used at the NSCL

- Killer Apps
- Application specific languages and configuration languages.
- Enabling Components.
- Applications built on enabling components

Hard to separate these from killer apps

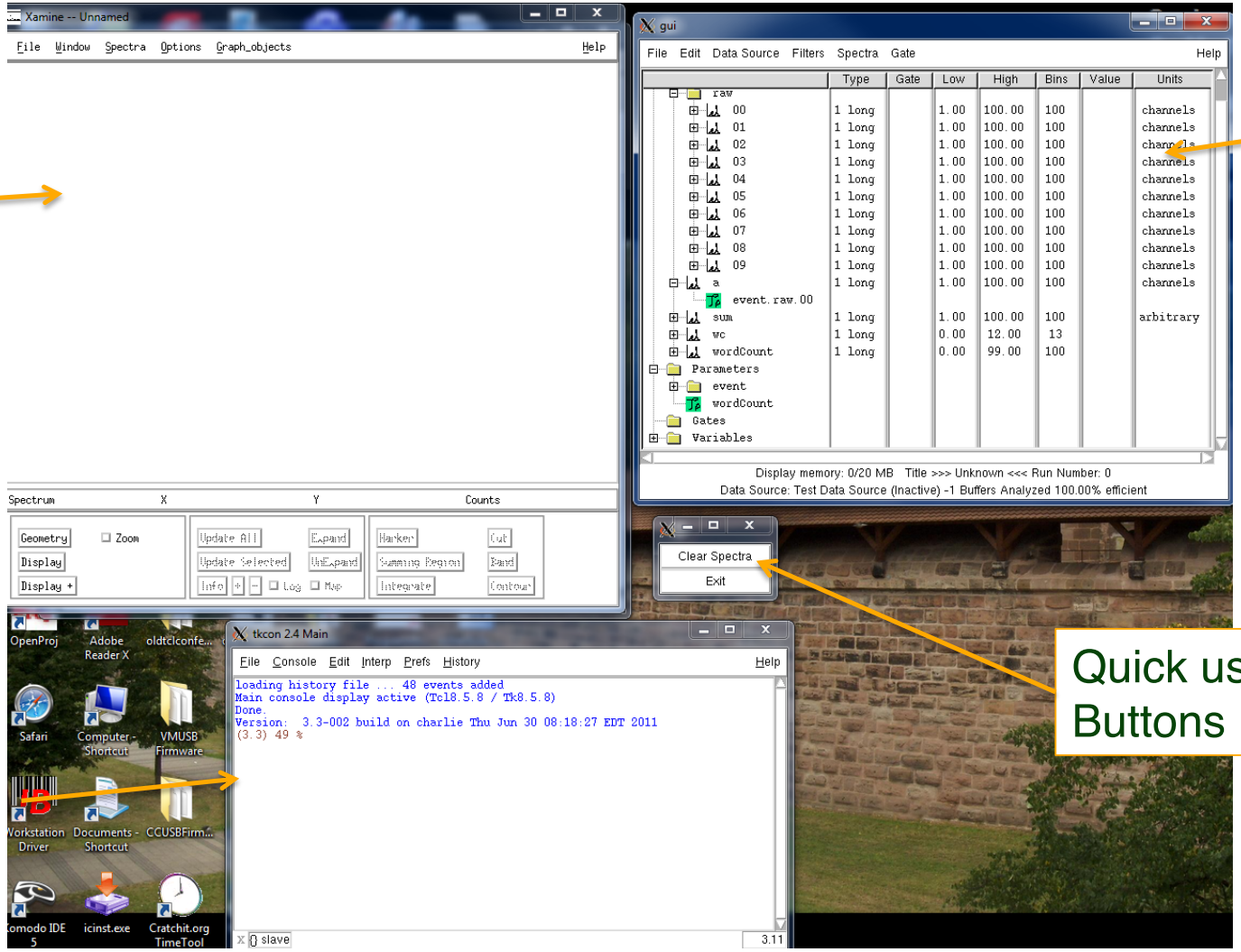
Killer Apps
NSCL SpecTcl

Moldy X/11 App

Spiffy Gui

TkCon (Thanks Jeff)

Quick user Buttons



Killer Apps: SpecTcl (continued).

- First appearance of NSCL (well me) at a Tcl Conference
- Tcl 2004 New Orleans
- http://www.tcl.tk/community/tcl2004/Papers/RonFox/fox.pdf

Killer Apps (Daniel Bazin) Presented at Tcl 2005

http://www.tcl.tk/community/tcl2005/abstracts/scienceandTech/SpecTk.pdf

Alternative to moldy X11 visualization App

Domain specific languages
Tcl configured Readout software

- Presented Tcl 2008 (this very room?) see:
  http://www.tclcommunityassociation.org/wub/proceedings/Proceedings-2008/proceedings/nuclearDSL/A_Domain_Specific_Language(slides).ppt
  for slides. (Get the proceedings at Lulu.com If you haven't already).

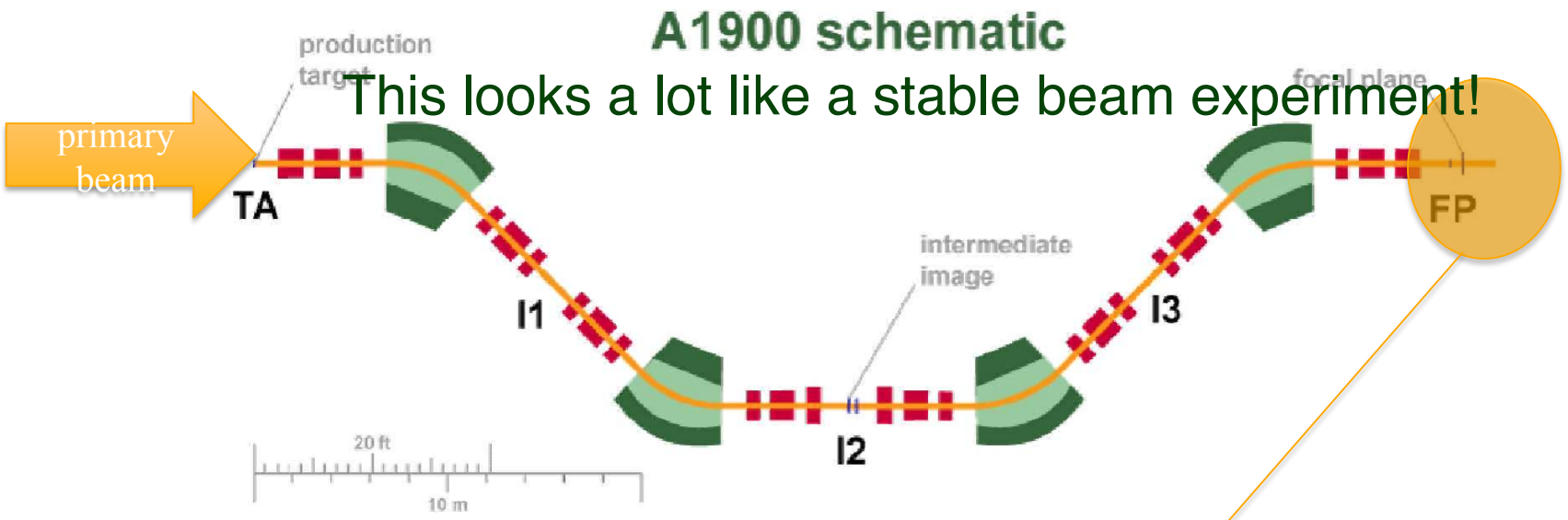
```
madc create dsssd1.x  -base 0x40000000 -id 4 -ipl 0
madc config dsssd1.x  -gatemode common -gategenerator disabled
madc config dsssd1.x  -inputrange 8v
madc config dsssd1.x  -timestamp on -timingsource vme \
                      –timingdivisor  $madcTimeDivisor
madc config dsssd1.x  -thresholds $thresholds(dsssd1.x)

stack create event
stack config event -trigger nim1
stack config event -modules [list  fadc
stack config event -delay 40


set       adcChannels(dsssd1.x) $xstrips
lappend adcChannels(dsssd1.x) timestamp
```

Sample experiment configuration file (this is Tcl).

Configuration Languages (DSL)



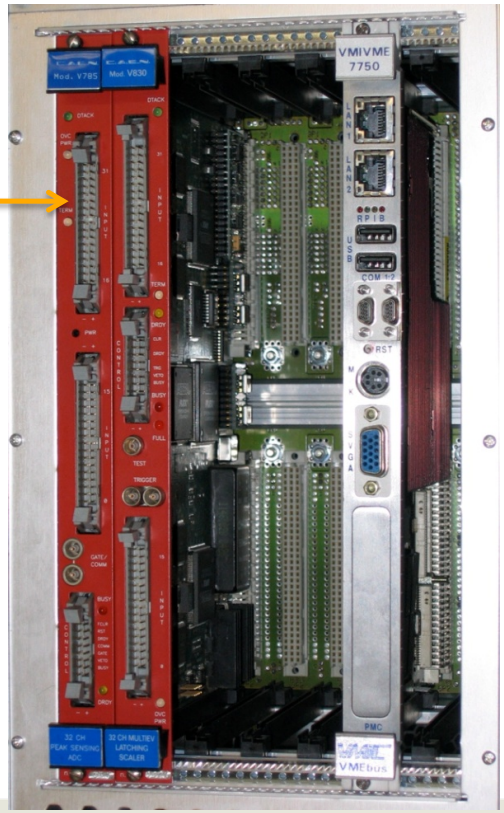
Data acquisition happens here.
A1900 Readout is parameterized (which detector packages are installed etc.) by a Tcl configuration file that is interpreted in a 'captive' Tcl Interpreter.

Enabling components and their Apps. [package require Vme]

Presented Tcl 2006 see:
http://www.tcl.tk/community/tcl2007/papers/Ron_Fox/vmepackage.pdf

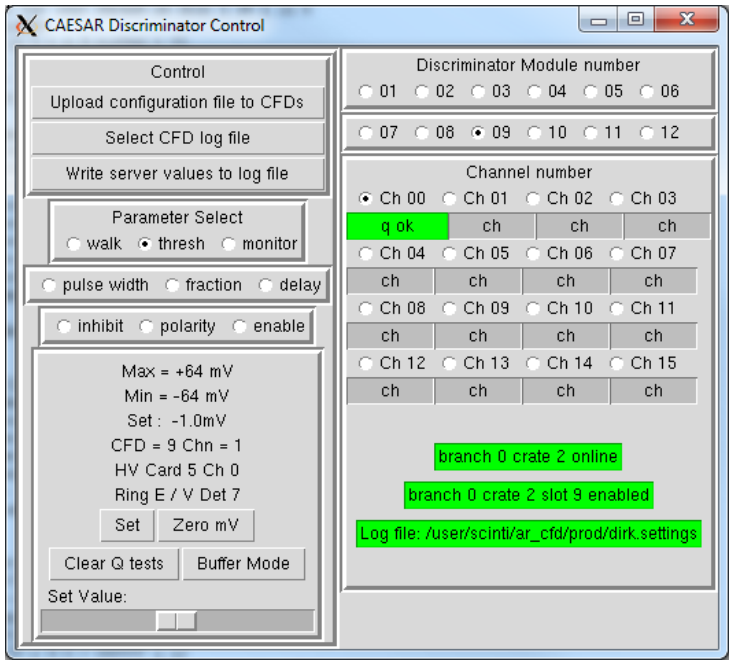
A lot of our hardware uses VMEbus (ANSI/IEEE Std 1014-1987 and ANSI/ VITA 1-1994) as an instrumentation bus.

http://en.wikipedia.org/wiki/VMEbus

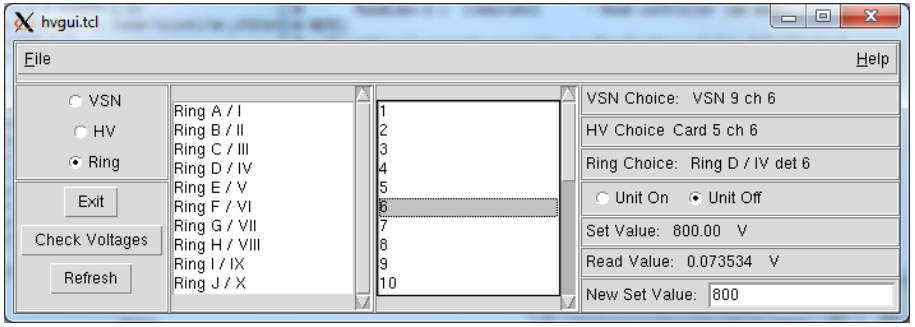


Vme loadable package provides Tcl scripts access to modules in this bus.

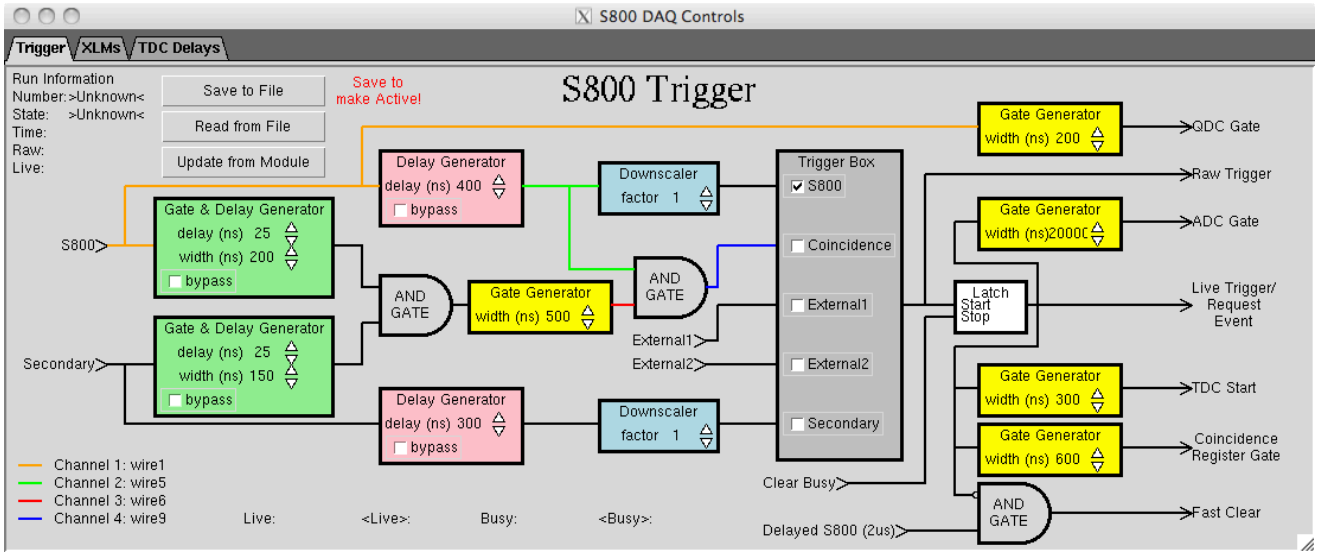
Control applications using Vme: (CAEsium iodide Array) CAESAR



Written by Andrew Ratkiewicz.
Thanks to Dirk Weisshaar
for these screen shots.



Control Applications (VME)

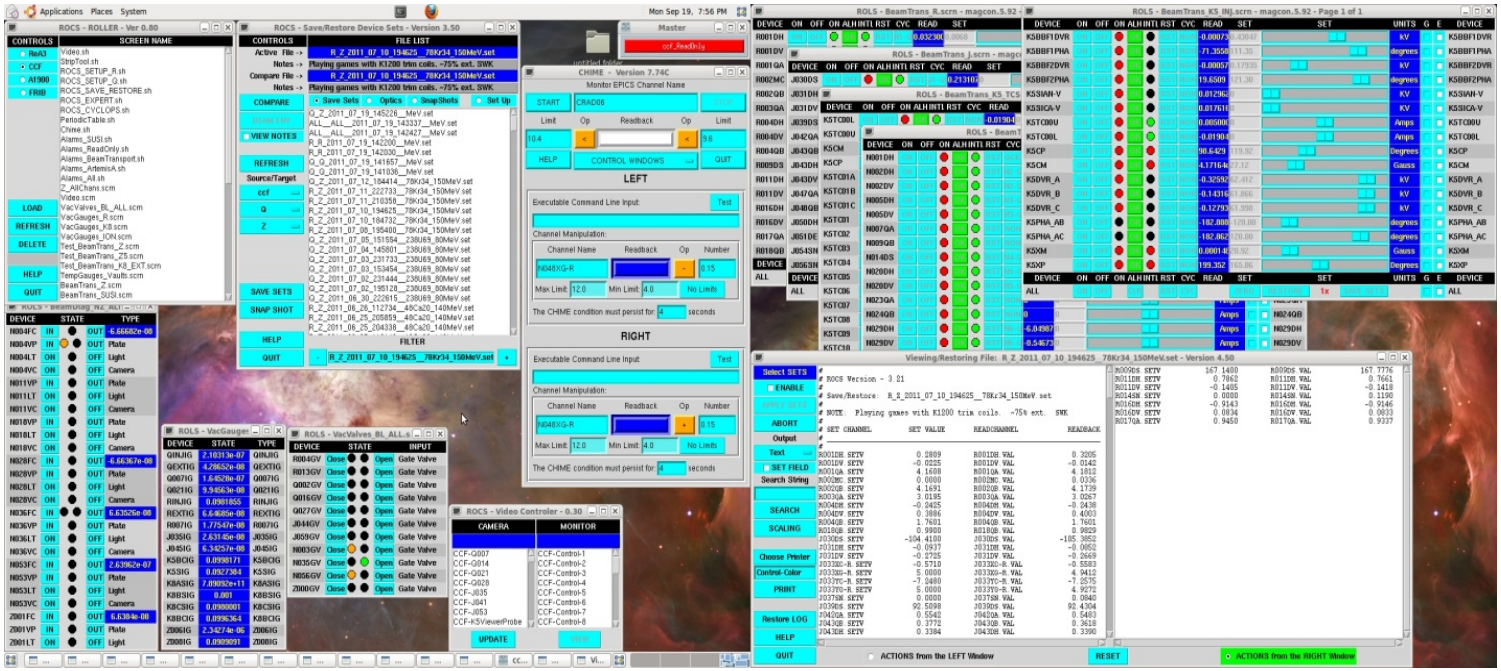


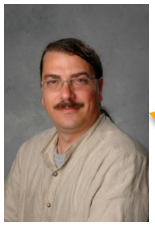
S800 trigger is an VME FPGA module. This control panel by Daniel Bazin allows experimenters to set the trigger parameters on a block diagram of the FPGA functional units.

Enabling Components and Their Apps package require epics

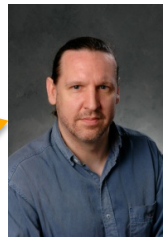
- EPICS: Experimental Physics and Industrial Control System. Relatively standard for Nuclear physics experiment slow controls and for many accelerators (including those at the NSCL). See: http://www.aps.anl.gov/epics/
- Presented at Tcl 2007 (When are we going back to New Orleans anyway)? (http://www.tcl.tk/community/tcl2007/proceedings/Gui/epics.pdf)
- Provides access to EPICS 'process variables' from Tcl scripts.
- NSCL Controls group model:
  - Give us an application definition
  - several months later we'll deliver a Qt C/C++ application
  - If it wasn't what you wanted give us a week or two iterating.
- epicsTcl allowed operators to get the HMI they wanted and to iterate until they got what they needed.
- epicsTcl was critical to the commissioning of ReA3.

Enabling Components and Their Apps package require epics





Rea3 control console Powered by Tcl



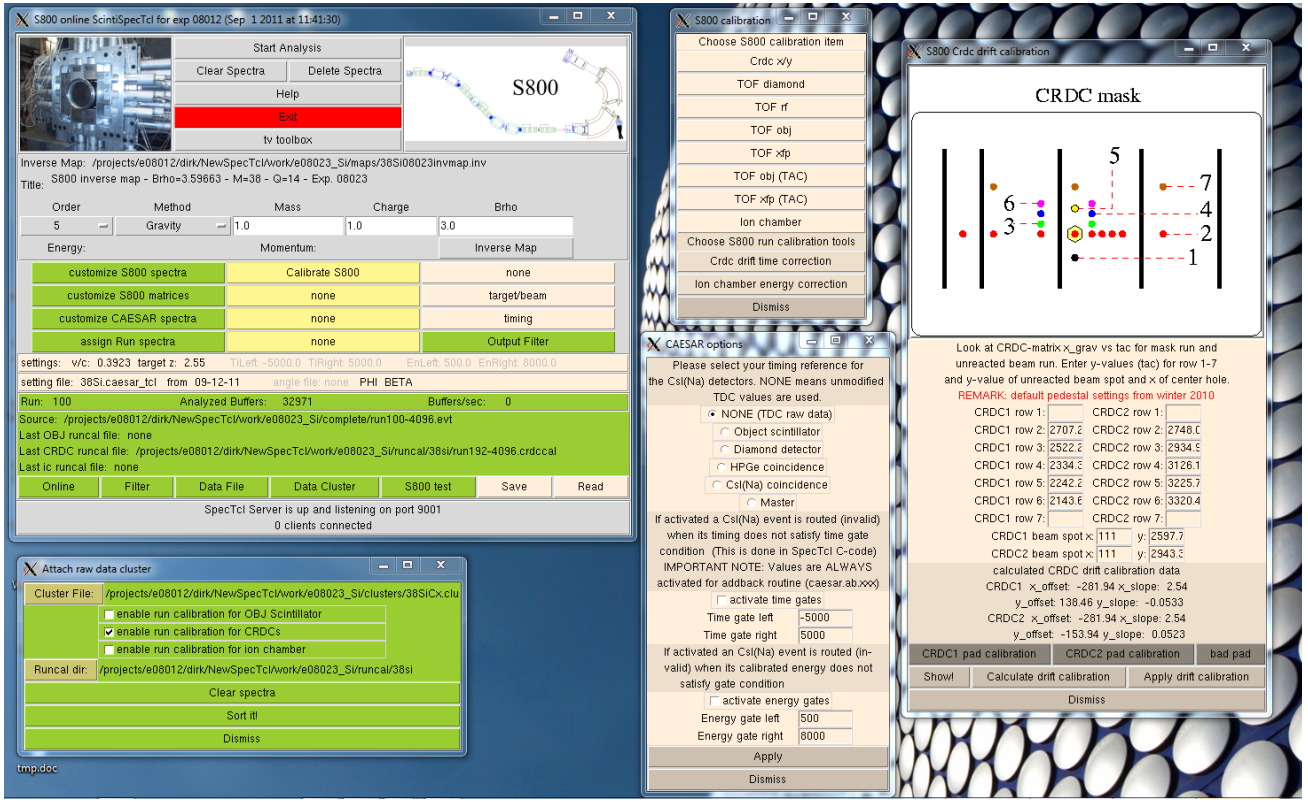
Thanks to Shannon Krause and Randy Rencsok for these screenshots.

Enabling Components and Their Apps
NSCL SpecTcl

Wait a minute… I thought you said that was a killer app?!

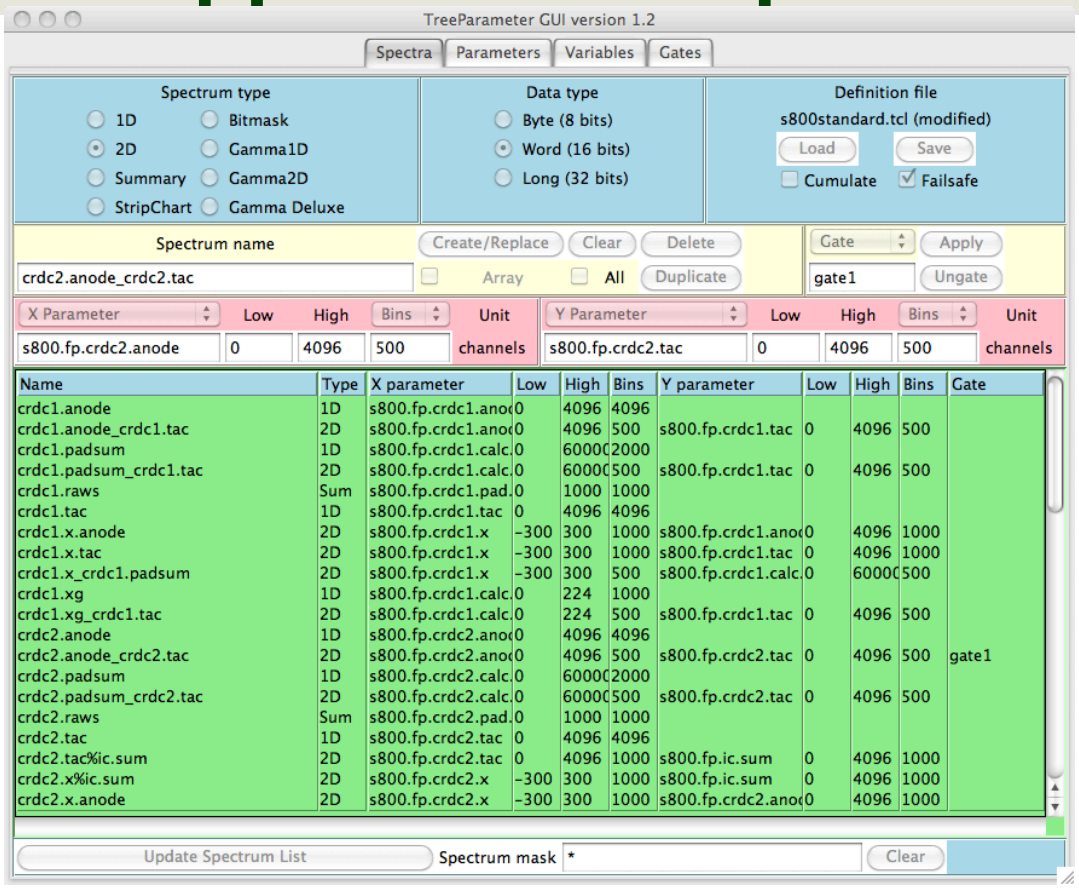
- One person's killer app is another's enabler
- Platform for application specific GUI's.
- [package require] is a free plugin architecture.

Enabling Components and Their Apps
NSCL SpecTcl



CEASAR SpecTcl GUI in all its full glory
(thanks again Dirk W.)

Enabling Components and their Applications: SpecTcl



Name	Type	X parameter	Low	High	Bins	Y parameter	Low	High	Bins	Gate
crdc1.anode	1D	s800.fp.crdc1.ano	0	4096	4096					
crdc1.anode_crdc1.tac	2D	s800.fp.crdc1.ano	0	4096	500	s800.fp.crdc1.tac	0	4096	500	
crdc1.padsum	1D	s800.fp.crdc1.calc	0	6000	2000					
crdc1.padsum_crdc1.tac	2D	s800.fp.crdc1.calc	0	6000	500	s800.fp.crdc1.tac	0	4096	500	
crdc1.raws	Sum	s800.fp.crdc1.pad	0	1000	1000					
crdc1.tac	1D	s800.fp.crdc1.tac	0	4096	4096					
crdc1.x.anode	2D	s800.fp.crdc1.x	-300	300	1000	s800.fp.crdc1.ano	0	4096	1000	
crdc1.x.tac	2D	s800.fp.crdc1.x	-300	300	1000	s800.fp.crdc1.tac	0	4096	1000	
crdc1.x_crdc1.padsum	2D	s800.fp.crdc1.x	-300	300	500	s800.fp.crdc1.calc	0	6000	500	
crdc1.xg	1D	s800.fp.crdc1.calc	0	224	1000					
crdc1.xg_crdc1.tac	2D	s800.fp.crdc1.calc	0	224	500	s800.fp.crdc1.tac	0	4096	500	
crdc2.anode	1D	s800.fp.crdc2.ano	0	4096	4096					
crdc2.anode_crdc2.tac	2D	s800.fp.crdc2.ano	0	4096	500	s800.fp.crdc2.tac	0	4096	500	gate1
crdc2.padsum	1D	s800.fp.crdc2.calc	0	6000	2000					
crdc2.padsum_crdc2.tac	2D	s800.fp.crdc2.calc	0	6000	500	s800.fp.crdc2.tac	0	4096	500	
crdc2.raws	Sum	s800.fp.crdc2.pad	0	1000	1000					
crdc2.tac	1D	s800.fp.crdc2.tac	0	4096	4096					
crdc2.tac%ic.sum	2D	s800.fp.crdc2.tac	0	4096	1000	s800.fp.ic.sum	0	4096	1000	
crdc2.x%ic.sum	2D	s800.fp.crdc2.x	-300	300	1000	s800.fp.ic.sum	0	4096	1000	
crdc2.x.anode	2D	s800.fp.crdc2.x	-300	300	1000	s800.fp.crdc2.ano	0	4096	1000	

This GUI by Daniel Bazin is used by most if not all SpecTcl users to set up the analysis conditions.

Enabling Components and Their Apps NSCL SpecTcl (plugins)

- Calibparams – map raw data to linearly calibrated data.
- firstof – Finds the first or largest parameter from a set present in each event (useful for some pixilated detectors).
- map – Untangles a set of parameters according to some mapping (useful for some pathalogical wiring schemes).
- radwareio – Exports/imports SpecTcl spectra to David Radford's Radware (a popular gamma ray spectroscopy package) : http://radware.phy.ornl.gov/
- rootFilterFormat – Produce event data in a format Root can process. (http://root.cern.ch)
- const - create parameters that have a constant value if at least one or all parameters from a set of parameters is present in an event (hit counting).

Work in Progress: Tcl Time stamped Event building

April 2012 the GRETINA detector will move to NSCL for a year long campaign of experiments with the S800. (http://grfs1.lbl.gov/).

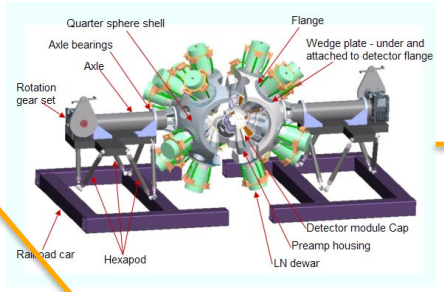
event timestamp

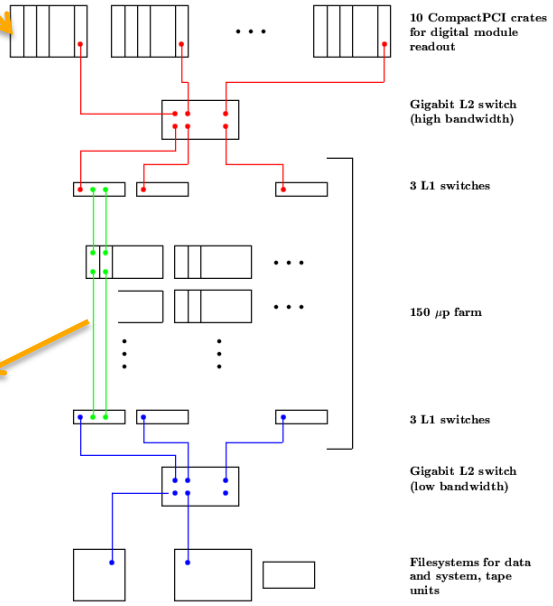
NSCL DAQ

event builder

Combined Events

(scientist)



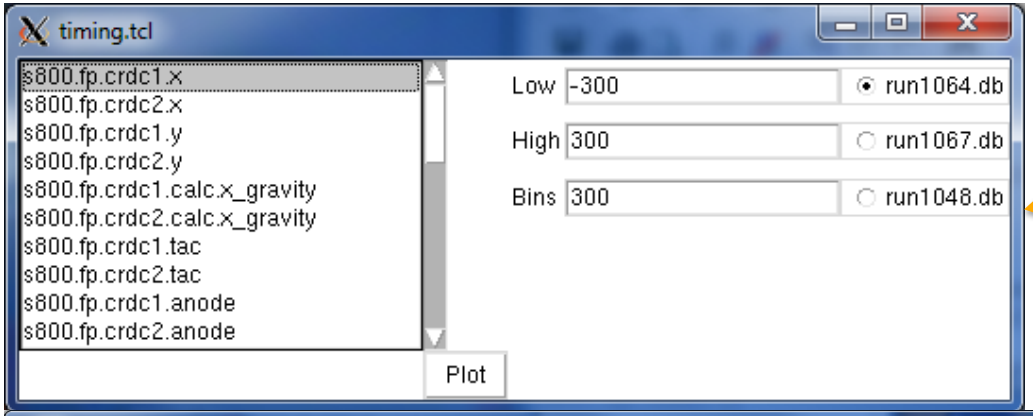


Gretina:

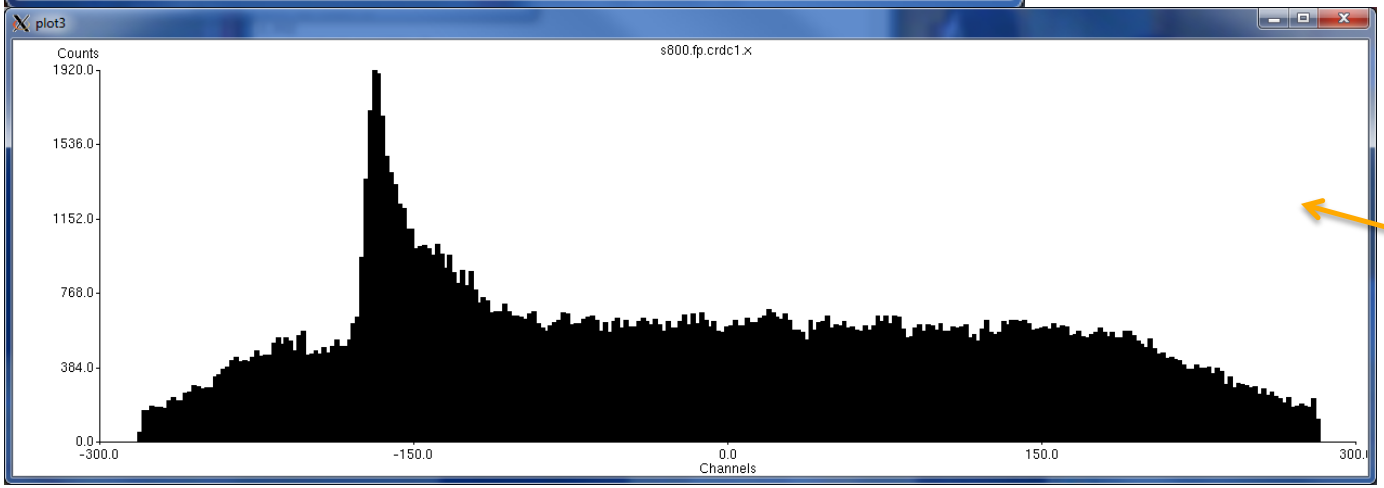
- Committed to providing the event builder.
  - Does not yet exist.
  - No schedule for when they will start working on it.
- Prototype built by me:
  - Tcl API used for event handling/dispatching
  - Tcl API used for Socket creation/connection management.
- This is a strange Tcl application: A Tcl interpreter that will never ever see Tcl Commands!

Speculative Work: after SpecTcl(?)

Using Sqlite as a nuclear physics data-bus… histogramming prototype:



Vanilla Tcl/Tk



plotchart
(thanks Arjen)

Thanks to Amanda Prinke for some data (300k events here) to play with

Speculative Work: Tcl/Tk bindings for Root

- Note: PyRoot already exists.
- Combine the complex class library Root provides with a simplistic Tcl/Tk wrapper.
- Get rid of Root's abominable graphics.
- Support higher level of interactivity with displayed data than root does easily.
- Prototype project specification has been completed for this (no code yet ☹ not even a WIP).

Looking Back:

- Tcl/Tk has been involved in many aspects of the NSCL:
  - Data Acquisition (all experiments).
  - Data Analysis (many experiments)
    - » About 25-30 institutions using SpecTcl actively (mostly but not all NSCL user groups).
  - Control system development
    - » Coupled Cyclotrons console.
    - » ReA3 commissioning controls.
- A balanced approach; providing both applications and enabling packages has allowed our users to meet their needs quickly and without the intervention (and bottlenecks) of 'professional programming teams'.
- Given tools, our users have built applications not originally envisioned by us. Had we not embedded a scripting language like Tcl, most likely many of these applications would not have been built due to a 'professional programmer bottleneck'.

Looking Forward
Tcl in Nuclear Physics has a tough road ahead:

- Root has gained widespread public acceptance although:
  - It has a steep learning curve
  - The class libraries are not particularly well designed.
  - C/C++ is not really a good ad-hoc scripting language (PyRoot helps here).
- Competitors to Root
  - Implementations of Abstract Interfaces for Data Analysis (AIDA).
    - » JAIDA (java implementation)
    - » PAIDA (python implementations)
    - » OpenScientist, PI (C++ implementations)
  - Java Anlaysis Studio (lives on top of JAIDA).
  - JHepWork (Java/Jython)
- Increasingly the community is turning to Java/Jython solutions (mostly because JAIDA is probably the most actively maintained AIDA implementation.
- As the nuclear physics community gets more into the 'object oriented programming model', they turn to languages that are better known as object oriented (in order of history C++, Java, Python).

Looking Forwards

- Tcl can serve its original design role of gluing together the diverse set of applications scientists use into seamless ‘meta-applications’.
- Tcl needs to have its OO capabilities better marketed.
- Wrappers for common toolkits/interfaces (Root, AIDA) will need to be provided.
- Front ends for existing tools (e.g. HippoDraw) would need to be added.
- Tcl still provides the simplest way to build a GUI (my opinion). This needs marketing as does ttk::
- Needs people who are familiar with Tcl/Tk, and can understand the problems of nuclear physicists..who are willing to work to provide tools and promote them within the community.
- The run up to the commissioning of FRIB is an opportunity to re-think how data are taken and analyzed and what Tcl’s role should and can be.