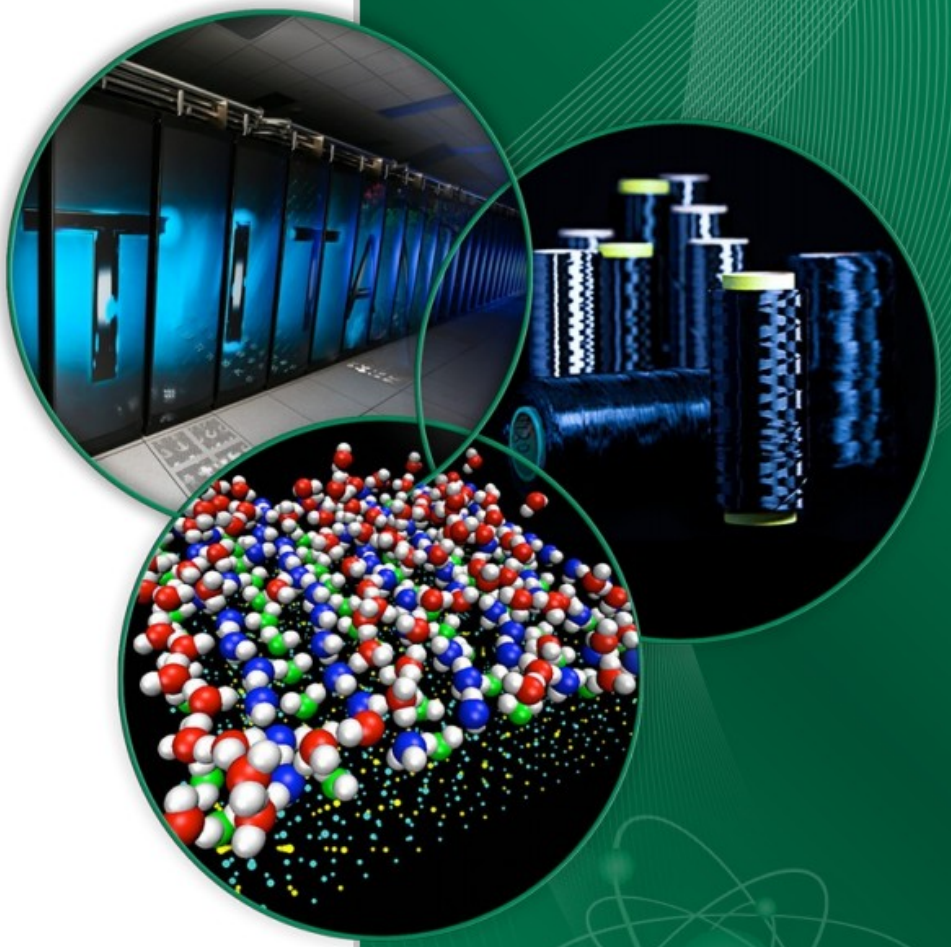
Usability and User Environments for Advanced Modeling and Simulation



David Bernholdt on behalf of Jay Jay Billings
Research Staff
Oak Ridge National Laboratory
billingsjj@ornl.gov
@jayjaybillings
+jayjaybillings

with Alexander J. McCaskey and Jordan H. Deyton

for the Advanced Computing Tech Team
20140826

Outline

- A little about usability
- About the problem
- Our solution

Additional Resources:

Sourceforge.net

niceproject.sourceforge.net

YouTube

youtube.com/jayjaybillings

Ohloh.net

ohloh.net/p/niceproject

Usability

Usability

Usability → Accessibility

How can users get access to advanced M&S capabilities?

Think of using a highly technical code as the bear trap at right:

- Easily getting the good stuff requires training
- Features may exist in a rusty framework with sharp edges (tough code, not always bad)
- Adding something is not for the faint of heart

Making it easier to use a code *naturally* exposes advanced features, masks complex designs and enables users to explore!

Usability in Modeling and Simulation

How I send messages around the World:

How I send messages to my code:

./xolotl ../benchmarks/he-W_2067.txt --handlers dummy --petsc -ts_final_time 1000 -ts_final_time 1000 -ts_adapt_dt_max 10 -ts_max_snes_failures 200 -pc_type fieldsplit -pc_fieldsplit_detect_coupling -fieldsplit_0_pc_type redundant -fieldsplit_1_pc_type sor -ksp_monitor -ts_max_steps 3

Really?!

So what's the problem?

Standard Model of Scientific Computing

All users must do these things...

Define the Problem

Write an input file in a format reminiscent of a dead language

Run the Simulator

Manually launch jobs on impressively terrifying machines

Analyze Output

Analyze simulation output in its most raw and unlimited form

Archive Output

Store data... somewhere!

Super-users think these are easy tasks, but most users are overwhelmed!

A nicer model of Scientific Computing

It would be better to have a computer program handle all of that...

A. User

ice

Define the Problem

Run the Simulator

Analyze Output

01100010
01101001
01101110
01100001
01110010
01111001

Archive Output

Most of the stuff we need to do can be encapsulated for ease of use and/or automated entirely with improvements.

The Eclipse Integrated Computational Environment (ICE)

Or, the product also known as the NEAMS Integrated Computational Environment

NiCE

And in some places as “Jade”

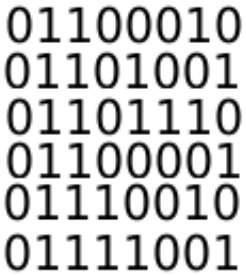
Project Details

- 100% open source, cross-platform, general-purpose user environment
- Primarily developed at Oak Ridge National Laboratory
- 7 team members, (6 full time, 1 part), + 2 support staff
- Focused on making life easier for users *and* developers

Define input

Run Jobs

Do Analysis



Store Data

ICE does these three very well

Needs work

Code details of the Project

- Binary and source downloads available
  - Don't burden users with compiling!
- Roughly 141k lines of code
  - +170k comments
  - Extensive (exhaustive!) tests
- Good out-of-source documentation
  - Wiki with tutorials, detailed docs
  - UML Model
  - Javadoc & Doxygen
- Sponsored by U.S. DOE, LDRD
  - >$2.5M from NEAMS
  - $400k from EERE
  - $140k from CASL
- Will be released through the Eclipse Foundation in FY15

Languages

Java	74%	C++	17%
XML	7%	10 Other	2%

Project Cost

Codebase Size	Estimated Effort
141,172 lines	35 person-years

Estimated Cost

$ 1,943,471 *

*Using the Basic COCOMO Model

Where does it work?

Nuclear Energy

Data Analysis

Batteries

Quantum Computing

Basic 3D Geometry and 2D Mesh Editing

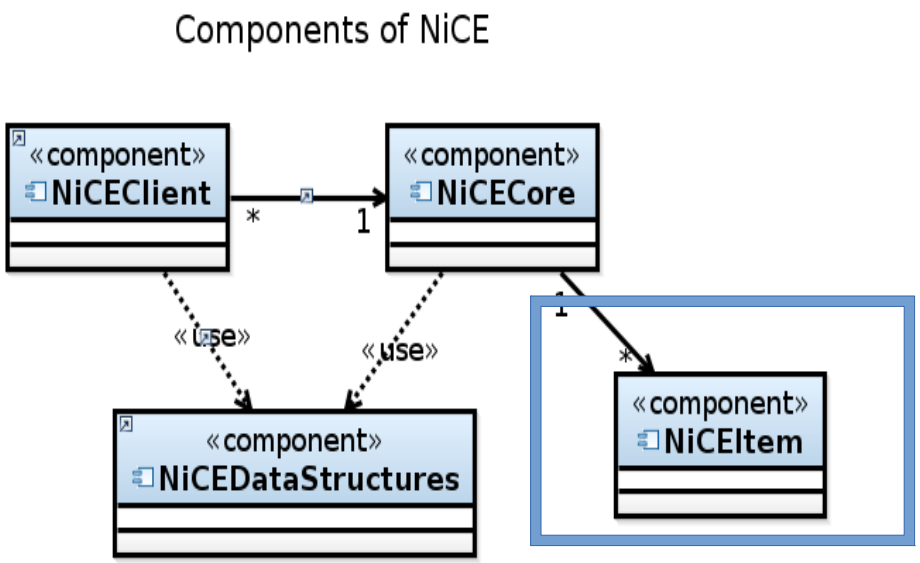
Fusion

Advanced Materials

Astrophysics

Coming in FY15!

How does it work?



The platform is easily modified:

- Developers write *plugins*
- Some extensions can be made with no code (no plugins)

Plugin code goes here!

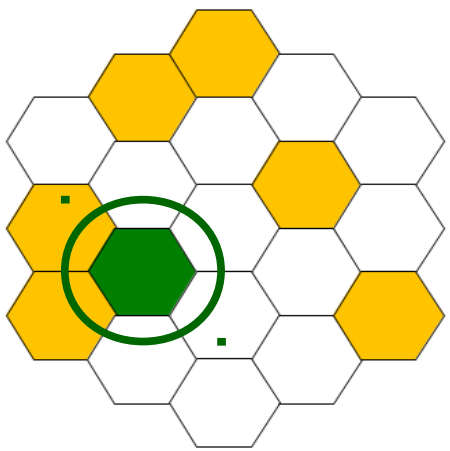
Users...

- Download a binary for their OS on our website
- Unzip a file to install for most capability, some depedencies for nuclear users
- Follow online tutorials or join our mailing list for help
- Otherwise, point and click!

“How about a magic trick?”

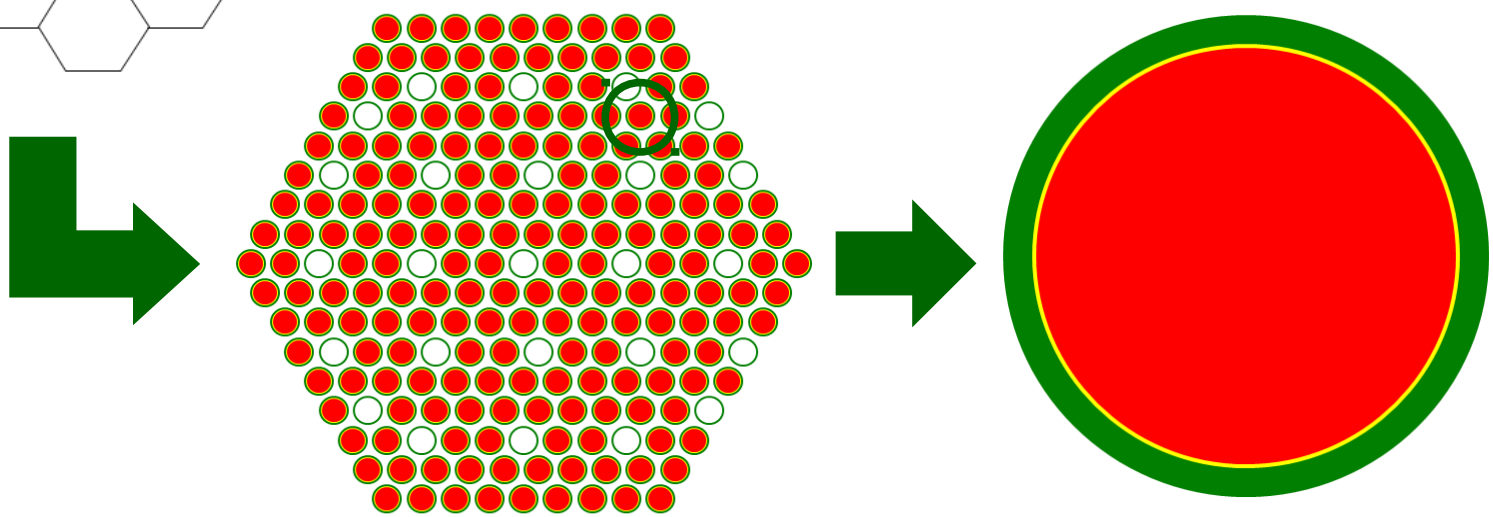
Domain-specific views

Raw data can be messy. What if we let the user explore the reactor to find the data they want?



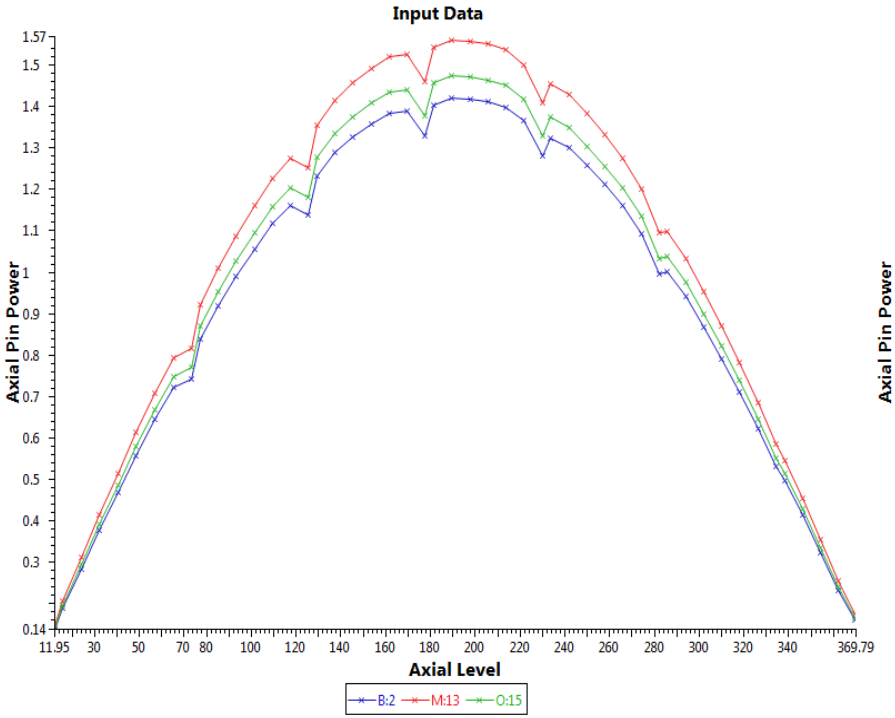
Mimic the hierarchical layout of real reactors:

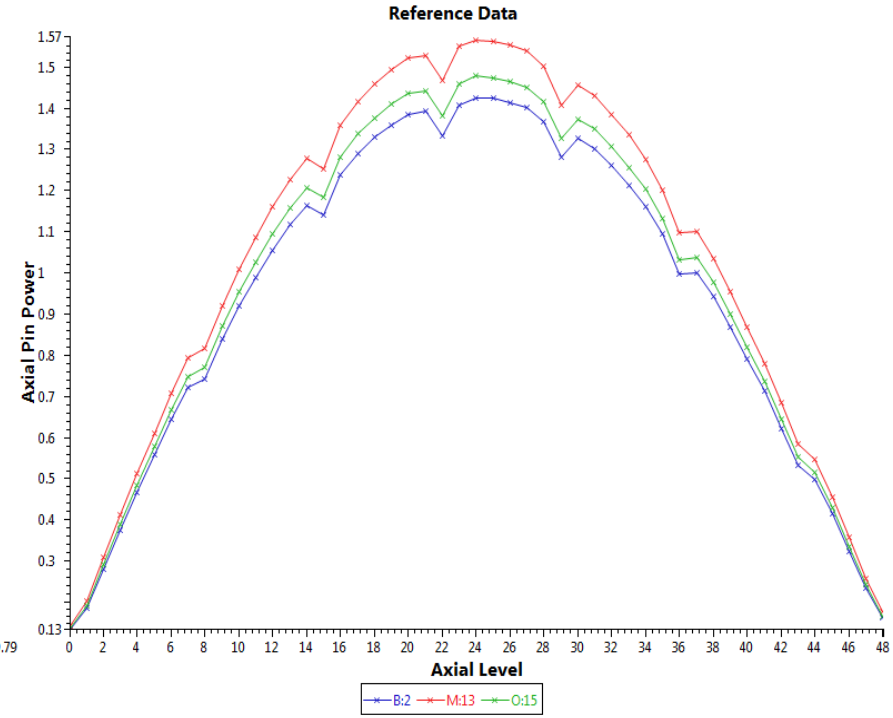
- Cores have assemblies
- Assemblies have rods or pins
- Rods have exotic materials



Enable Visual Comparative Analysis...

Goal 1: Examine output results by visually comparing them with validated reference data.

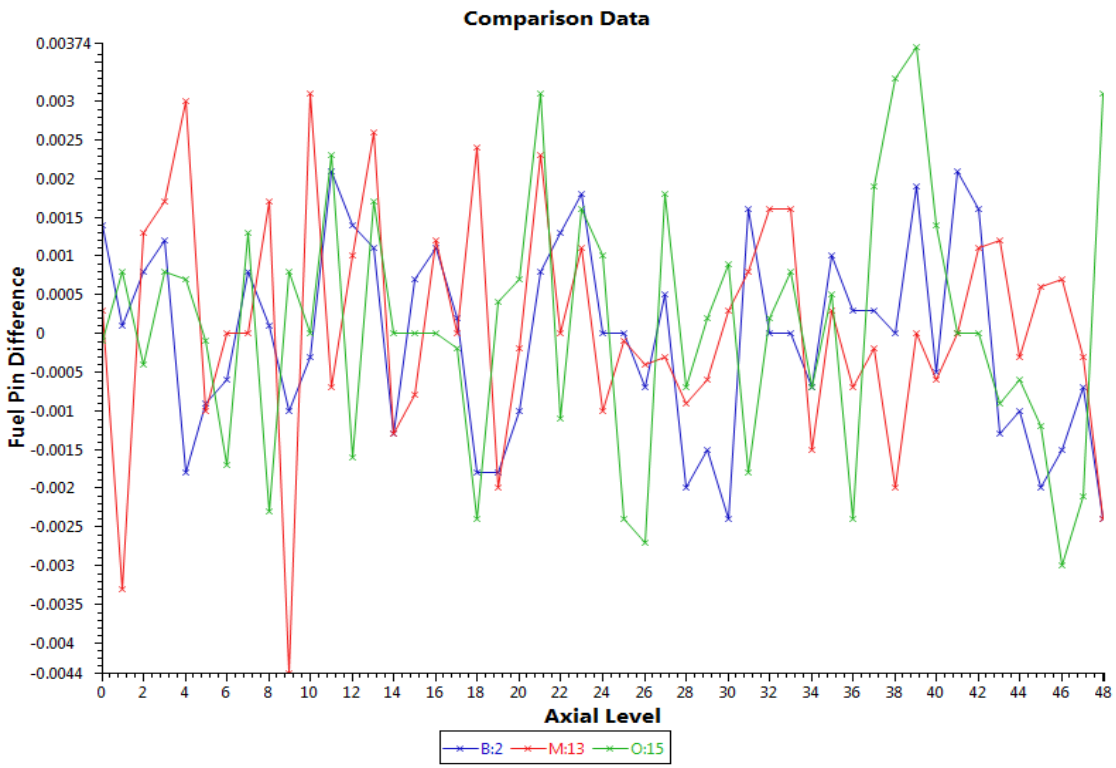




Input fuel pin powers along the length of the pin for three pins on the left, with reference data for the same pins on the right

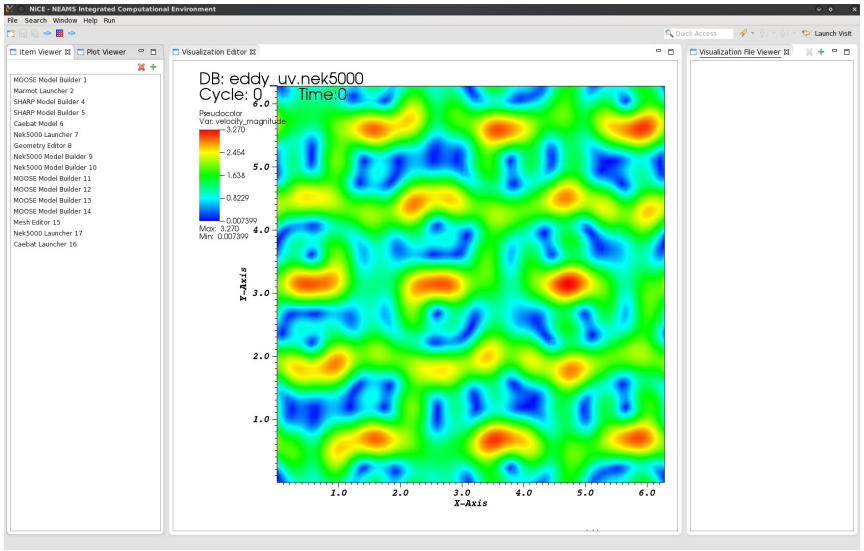
... and Quantitative Comparative Analysis

Goal 2: Do the actual math, too, to quantitatively compare results based on both the simulation data and the reference data.

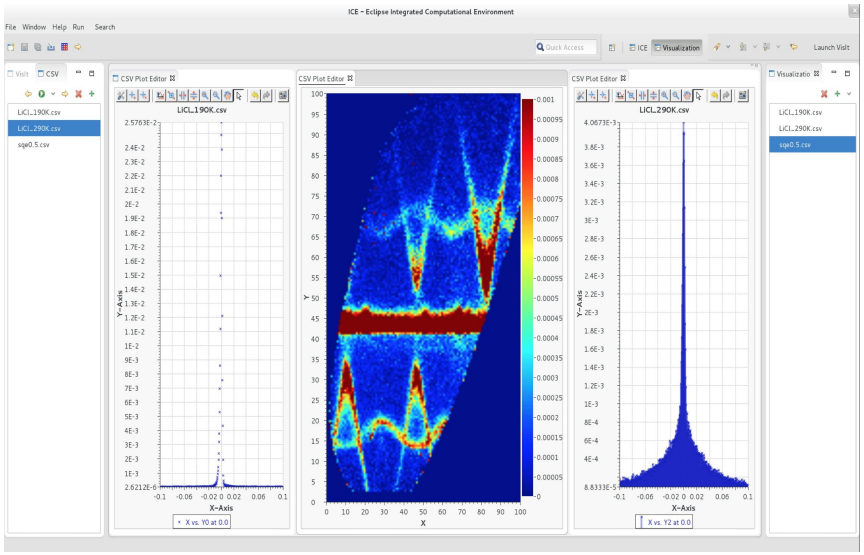


The percent difference between the simulation output and reference power data for the pins.

Hook up with existing tools



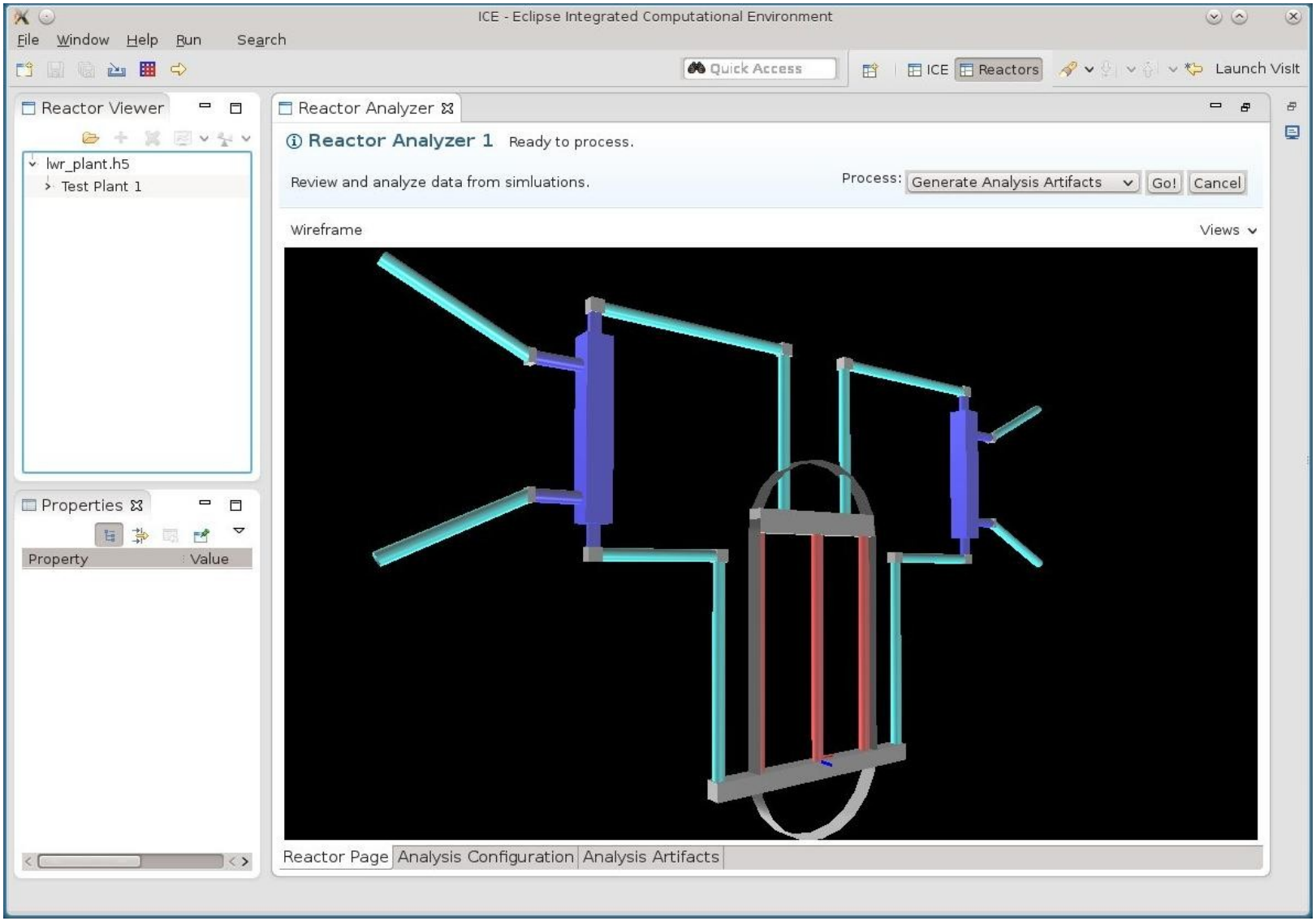
VisIt workbench in NiCE with Nek5000 eddy results



SWT-XY-Graph and JZY3D showing neutron and phonon data

Simply: Reduce, Reuse, Recycle

Do some new 3D viz



3D Model of a Nuclear Plant (TMI)

Summary

- Usability requires thought and users want it!
- Modeling and simulation in the DOE space is mostly the same four tasks
- Disparate and advanced capabilities can be integrated

Finally: Smart user environments CAN make tough codes easy to use for novices and will open up modeling and simulation to more members of the community.

Any Questions?

Catch the YouTube Videos!

Thanks to our sponsors!

Sourceforge.net

niceproject.sourceforge.net

YouTube

youtube.com/jayjaybillings

Ohloh.net

ohloh.net/p/niceproject

BUILT ON eclipse

eclipse FOUNDATION MEMBER

Additional Authors: Dasha Gorin, S. Forest Hull, Alexander J. McCaskey, Taylor Patterson, Anna Wojtowicz
Author Email: billingsjj@ornl.gov, deytonjh@ornl.gov
Author Twitter: @jayjaybillings