Case Study: Leveraging a Model Based Development Method and Rational Tools for Requirements and Modelling of a Nuclear Energy Framework

Jay Jay Billings
Research Associate,
Oak Ridge National Laboratory
billingsjj@ornl.gov

John M. Hetrick, III
IT Specialist, IBM Rational
hetrick@us.ibm.com

RWI-1637A

Innovate2010
The Rational Software Conference

Let's build a smarter planet.

The premiere software and product delivery event.
June 6–10 Orlando, Florida

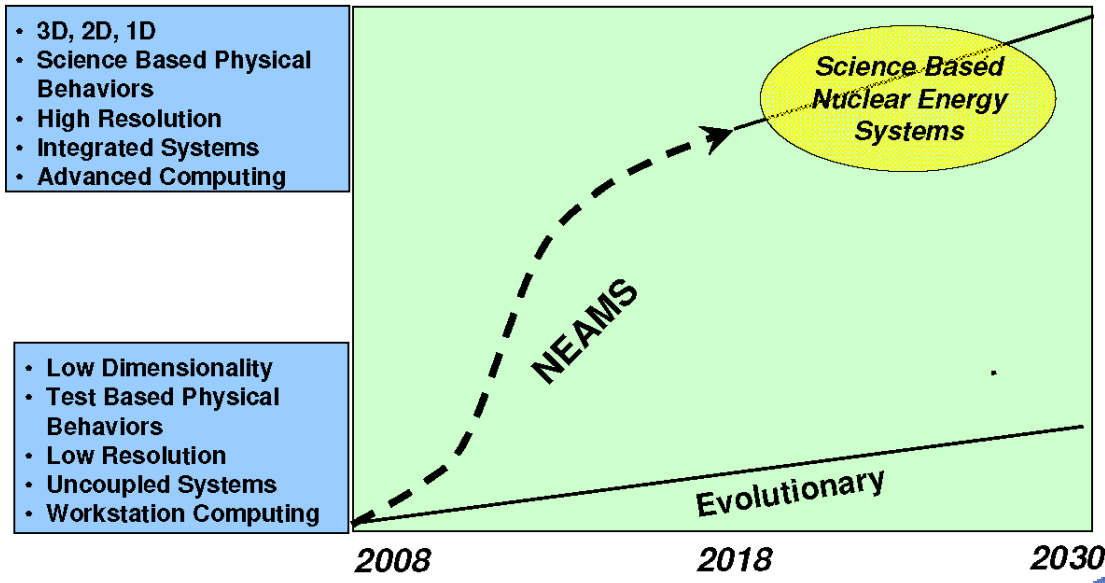
Agenda

- Nuclear Energy Advanced Modeling and Simulation (NEAMS)
- Oak Ridge National Laboratory (ORNL) NEAMS - Capability Transfer (CT) Methods and Tools
- ORNL NEAMS-CT On-Going Work

Nuclear Energy Advanced Modeling and Simulation (NEAMS)

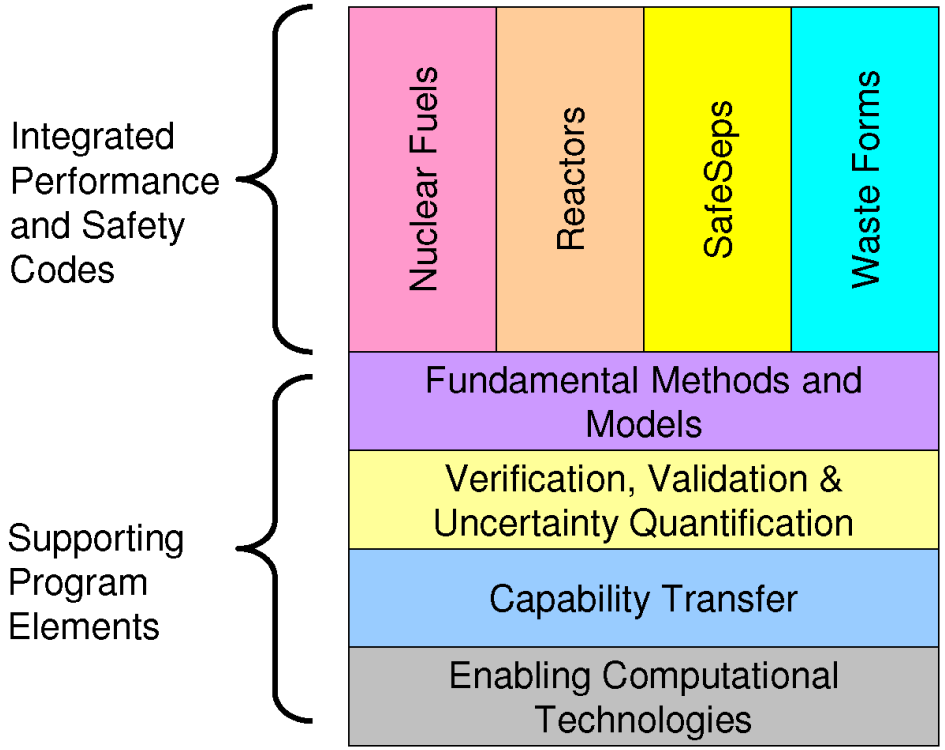
- An R&D program sponsored by the U.S. Dept. of Energy Office of Nuclear Energy, carried out by a group of National Labs and Universities
- *Vision:* Rapidly create and deploy "science-based" verified and validated modeling and simulation capabilities essential for the design, implementation, and operation of future nuclear energy systems with the goal of improving U.S. energy security

- Update aging nuclear energy M&S capabilities with better physics and higher fidelity
- Help model existing reactors and prototype new designs
- Users in R&D, industry, regulators, utilities, operators



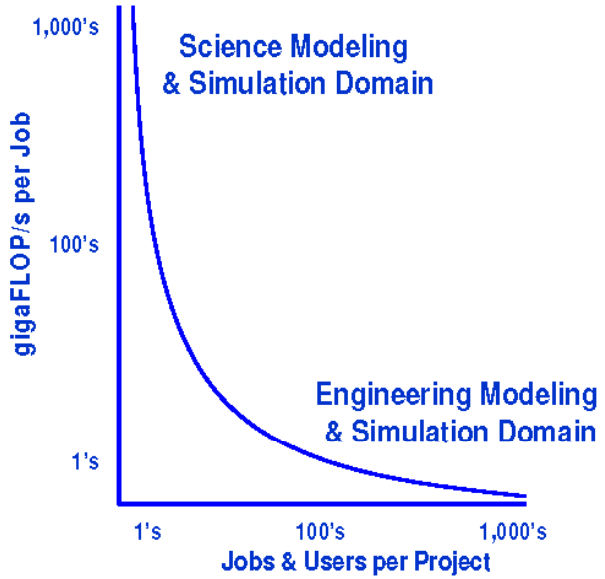
NEAMS Program Elements

- Integrated Performance and Safety Codes
  - Reactors
  - Fuels
  - Safeguards and Separations
  - Waste Forms
- Cross-Cutting Elements
  - Capability Transfer
  - Fundamental Methods and Models
  - Enabling Computational Technologies
  - Verification, Validation, and Uncertainty Quantification



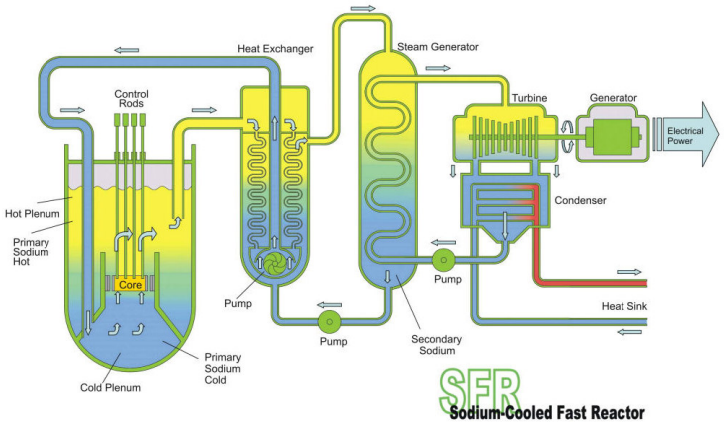
NEAMS Capability Transfer (CT)

- Enable the efficient transfer of advanced modeling and simulation capabilities from the science domain to the industry engineering domain
- A broad spectrum of users
  - "Internal" R&D activities
  - Other R&D areas (e.g. Advanced Reactors)
  - Technology designers (e.g. Westinghouse, GE-Hitachi)
  - Regulators (e.g. NRC)
  - Utilities and operators (e.g. TVA)
- Users want different things out of NEAMS
  - Some will use NEAMS software as-developed
  - Others will use only selected modules, integrated with their own (proprietary) code
- CT requires a comprehensive, but flexible approach to the software
  - Common user experience, minimizing differences across codes
  - Facilitate integration across NEAMS capabilities and into more complex user workflows
    - e.g. user design optimization workflow using various NEAMS modules
    - e.g. coupling "live" fuel and reactor models



Why is this Challenging?

- Scientifically
  - Much new scientific research and development is required, at multiple levels
  - "We know where we need to go, but we must figure out how to get there"
  - Focus of developers is first on science, second (third?) on software
  - New software under constant development



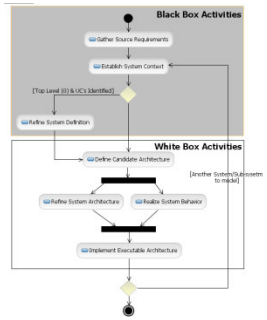
- Organizationally
  - Large geographically distributed teams
  - Scientists, not programmers
  - CT *collaborates* with IPSC code teams
  - Different code teams working on different timelines, with different tools/approaches

- Computationally
  - Must run anywhere from laptops to supercomputers
  - Parallelism: 1-2 processors to 1million-1 billion processors, GPUs, and more
  - Linux primary target, also Macs, some users want Windows

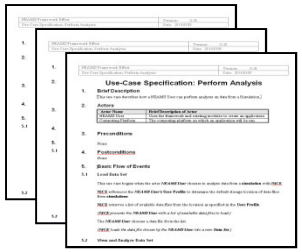
ORNL NEAMS-CT Methods and Tools

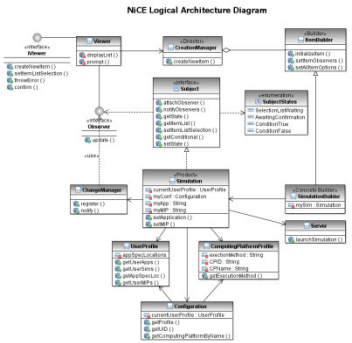
Stakeholders

Vision



- ORNL NEAMS Method Introduction
- ORNL NEAMS Method Up Close
- Focus on Patterns
- Rational Tools and ORNL Development Environment

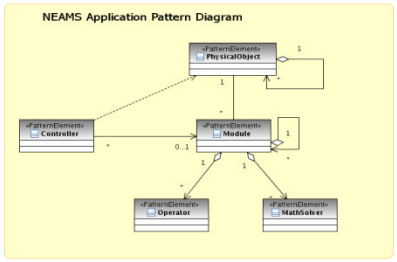


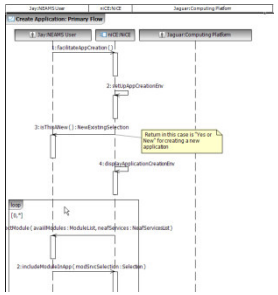


Rational RequisitePro

Rational Software Architect

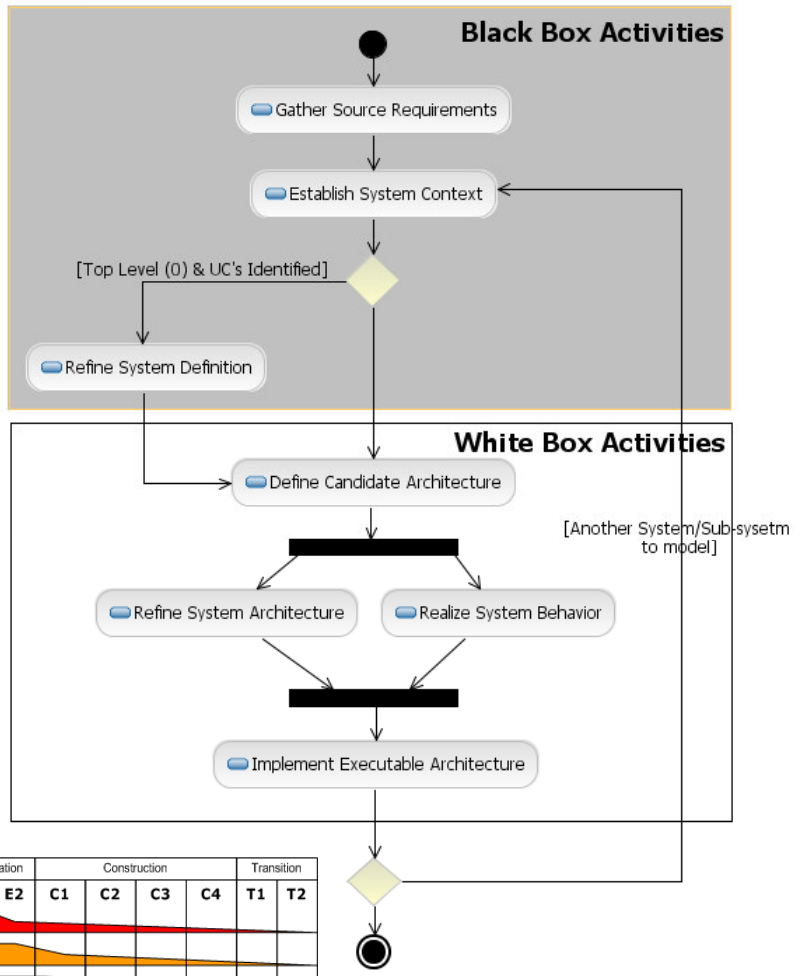
Rational Team Concert

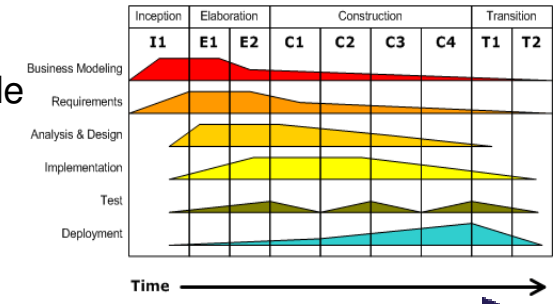




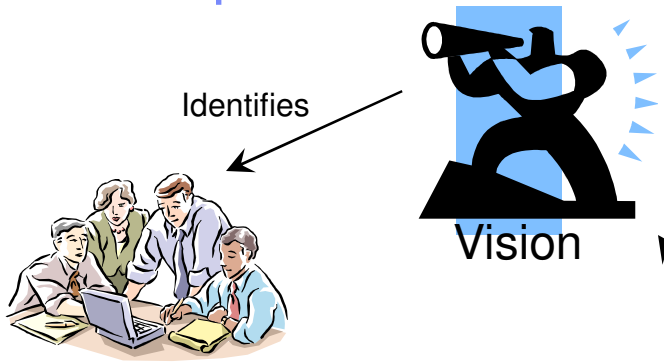
ORNL NEAMS Method Introduction

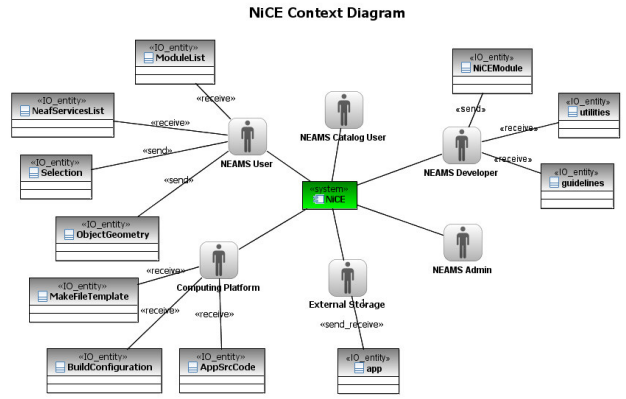
- **Model Based Development Method**
  - Proven use case driven, model based method developed by IBM Rational
  - Designed for large and/or complex software and system problems
  - Prescriptive, rigorous, scalable, and simple
  - Founded on software engineering principles and best practices
- **Key Concepts**
  - Flowdown – UC scenarios used to drive progress and maintain continuity (traceability)
  - Black Box to White Box
    - Black Box: Bound & Scope the problem, Isolate system in context, Identify what the system shall do
    - White Box: Focus on how the system works, Identify structural elements, Specify behavior as collaboration of elements
  - Multiple levels of abstraction: Analysis to Code
  - Focus on Architecture & Architectural Perspectives
  - Iteration Based Execution





Method Up Close: Black Box Work





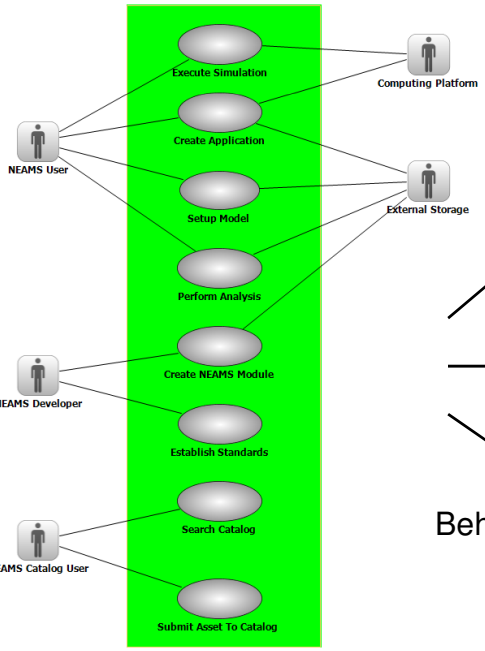
Context

<<Trace>>

Scope

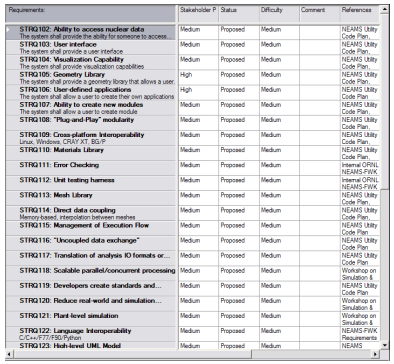
Define

Validate



Use Cases

Use Case Specifications



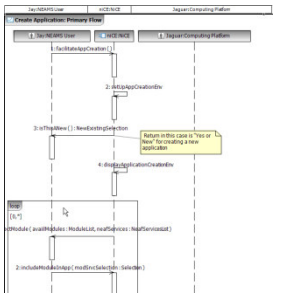
Requirements

Populate

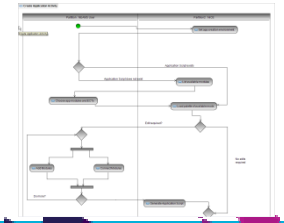
Detail

Interfaces

Behavior



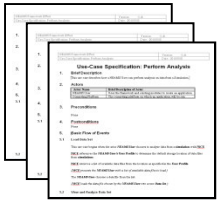
Interaction Diagrams



Activity Diagrams

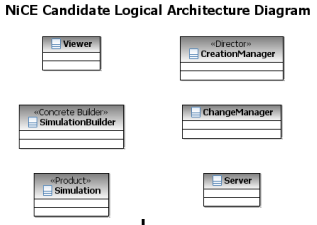
Method Up Close: White Box Work

Use Case Specifications



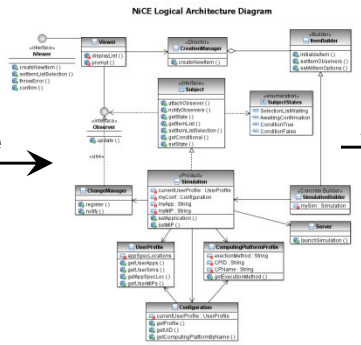
Identify

Candidate Architecture



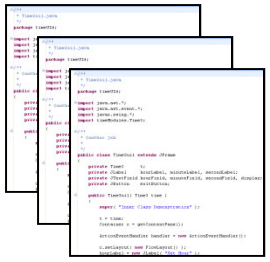
Refine

Logical Architecture

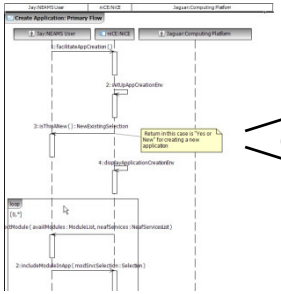


Executable Architecture

Code



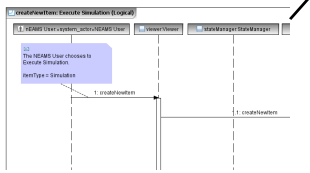
Scenarios



Black Box Sequence Diagrams

Operation Realization

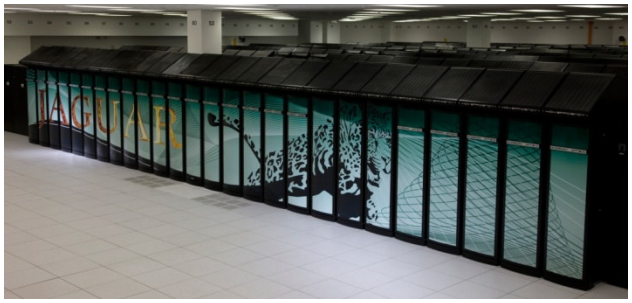
Element Collaboration



White Box Sequence Diagrams

Relationships

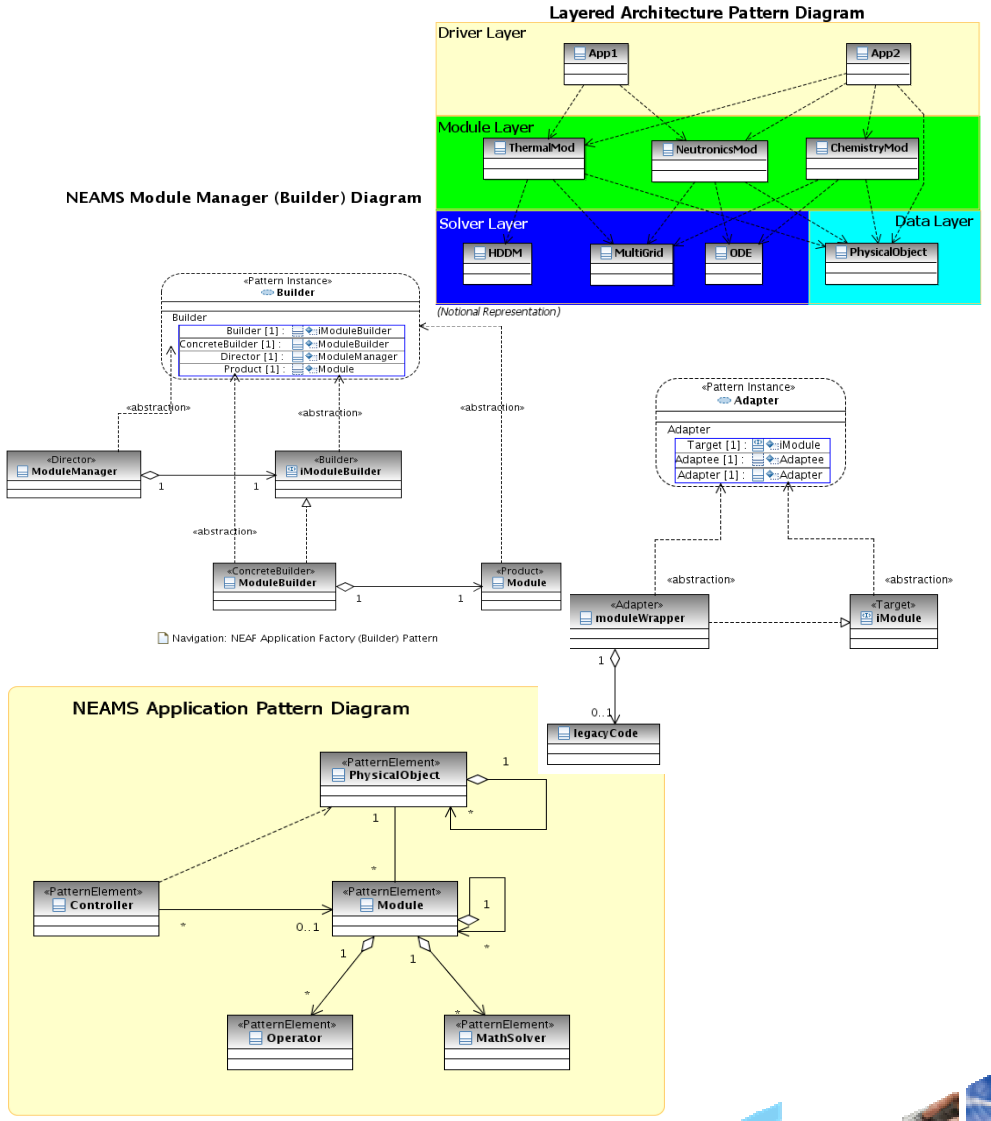
Validate



Computing Platform

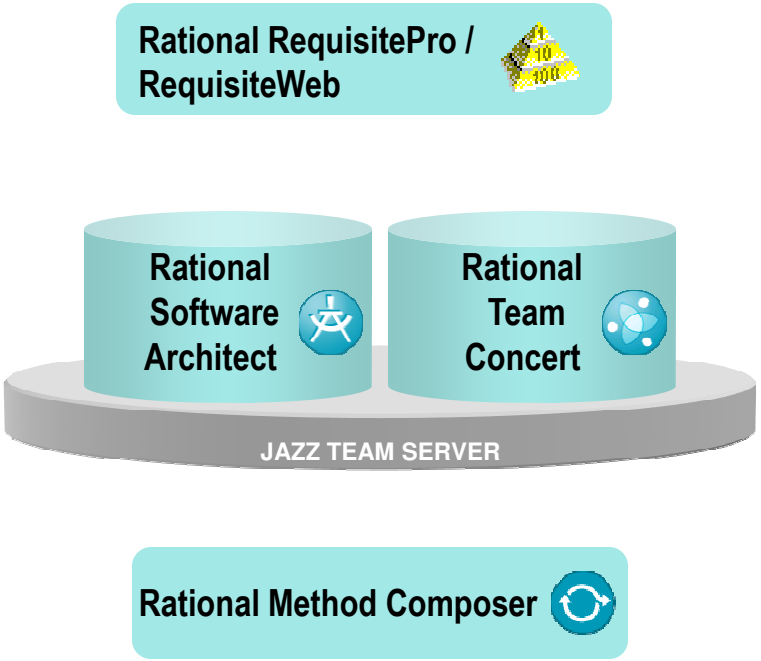
Focus on Patterns

- Patterns
  - Reusable code
  - Recognizable by experienced developers
  - Provides extensibility
  - Raise level of communication
- Architectural and Design Patterns
  - Layered Architecture Pattern
  - Builder pattern (creational)
  - Adapter pattern (structural)
  - Observer pattern (behavioral)
- Domain Specific Pattern
  - "NEAMS Application Pattern"
    - Homegrown pattern for modeling and simulation applications

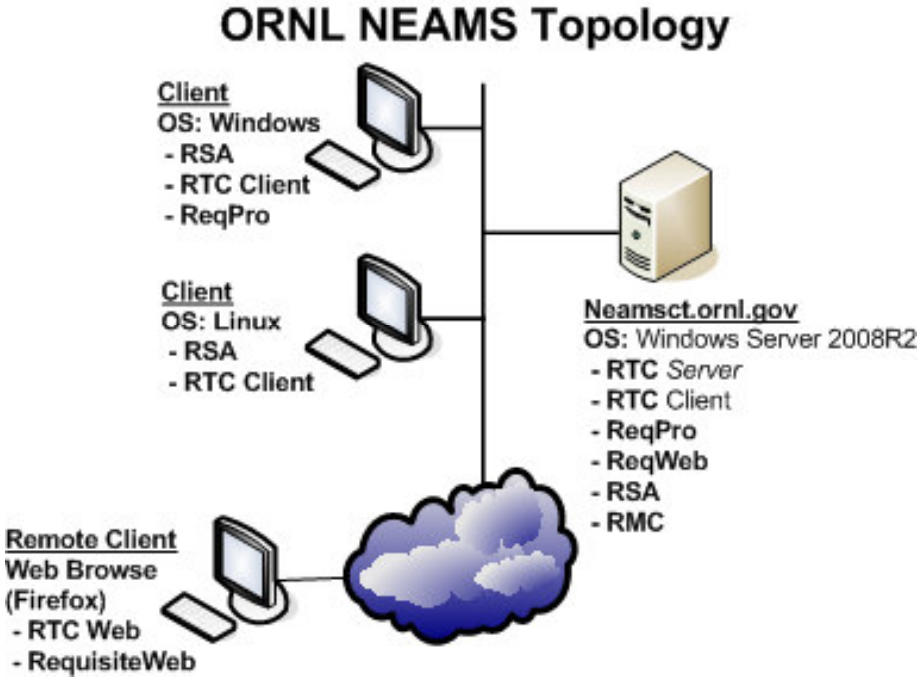


Rational Tools and ORNL Development Environment

Rational Tools Environment



Environment Infrastructure



Realized Value to ORNL NEAMS CT

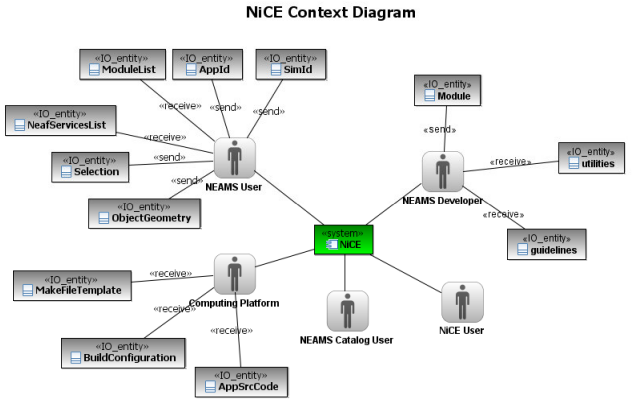
- Providing the ability to scope a nebulous and poorly defined problem
- Provides some formality and rigor
  - Typical software processes in computational science and engineering are more ad-hoc
  - Desirable given size and breadth of NEAMS, users targeted
- Rigor promotes thoughtful analysis of *this* problem
  - Avoids urge to think we know the answers and jump ahead
- Emphasis on a widely used and understood visual modeling language (UML)
  - Don't have to develop and teach our own "boxology"
  - Supported by commodity tools
- Reducing risk by showing demonstrable progress and providing stakeholders early and incremental access to key artifacts, deliverables, & decisions
- Opportunity to work with experienced practitioners from IBM Rational

ORNL NEAMS-CT On-Going Work

Finding the Start

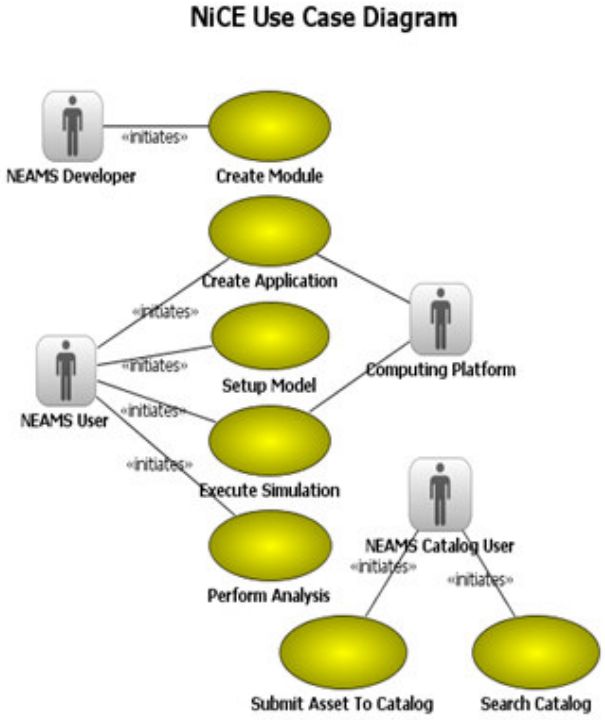
Original Orders: “Develop a software framework for Nuclear Reactors”

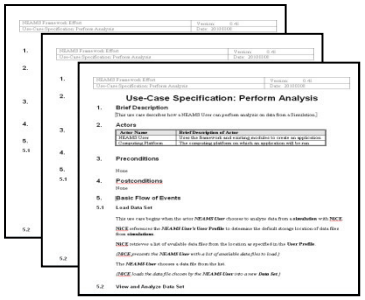
… Would you like fries with that? …



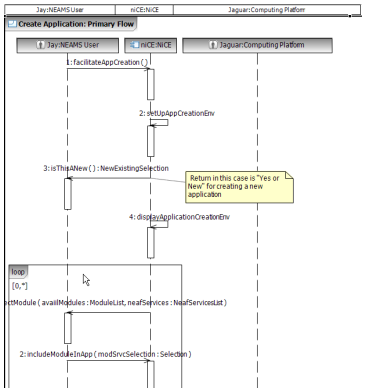
- Who do we talk to? What do we need to know?
  - Identifying stakeholder
  - Interviews – “… and a shake too…”
- Painting pictures – simple and effective analysis
  - Context Diagram
  - Use Cases and Actors
- Above the noise - discovered of two distinct, but related areas of focus: Computational Environment & Architecture Framework
- Formulating the Vision

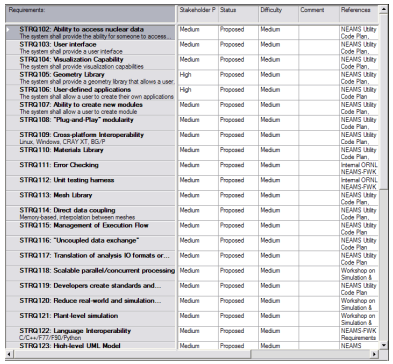
Understanding the Problem





- Scoping and bounding the problem
  - Seven use cases defined
  - Solidify what is in our box and what is not
- More Interviews structured and focused around use cases
- Capture and analyze how stakeholders work
- Ensuring Stakeholder requests are addressed

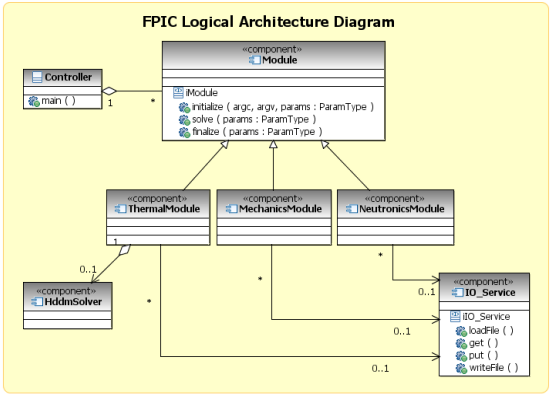




Separation of Concerns in the Solution

Two Areas of Focus:

- Computational Environment – Software system that allows for the development and composition of a wide range of applications
  - Create applications and define models by combining existing modules and services
  - Run simulations (execute applications) hiding the complexities of working with different platforms
  - Analyze data (results)

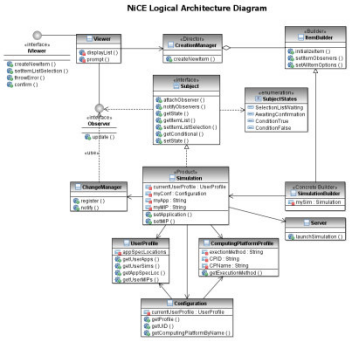


- Architecture and Framework – Software building blocks for developers of scientific modules and applications ("bring order to chaos" in NE codes)
  - Modeling and Simulation Framework
  - Frameworks to: couple physics components, achieve platform independence, enable scalable parallelism and concurrency,
  - Utilities for: language interoperability, handling I/O, legacy and proprietary software integration

Crafting the Solution: Flowdown and Executable Architecture

- Maintain Continuity moving from Black-box to White-box
- Use Case scenarios provide the context to focus the work and incrementally craft the solution
- Flowdown provides the technique for realizing how the NiCE will implement the scenario
  - Execute Simulation UC: Primary flow
  - Comprised of a set operations
  - The operations are realized by collaborating components
  - End result: Model that contains a complete architecture/design for the scenario

- Verify the architecture by executing it
  - Architecture is contained in UML Model
  - Transform model into skeleton code
  - Stub out the code skeleton
  - Execute the code and evaluate results
- Feed results back into architecture



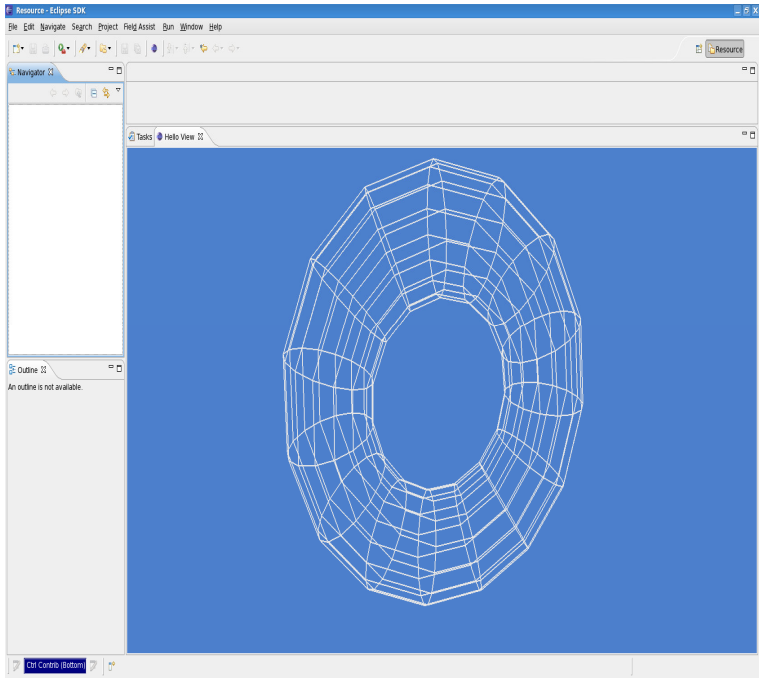
Gaining Efficiencies – Utilizing Tools

- RequisitePro
  - Collect, organize, and analyze sets of requirements from different sources
  - Provide traceability to ensure continuity with stakeholder requests
  - Integration with models very helpful
- Rational Software Architect
  - Currently the tool at the heart of our work
  - Use modeling to reason about the problem and solution
  - One environment to go from design (models) to code
  - Visualization of design and of code provides new insight
  - Pattern support has enhanced out communication
- Rational Team Concert
  - Easy to create tasks and track them – light weight project management
  - Great for core team members to collaborate, but challenging to involve others
  - Lots of cool power, still working through what is good for us
  - SCM is simple and affective, evaluated vs Subversion
  - Really helps with virtual nature of the core team

Future Work

Lots of interesting prospects!

- Engage remaining members of NEAMS Project
  - Currently only engaged 50% of program teams
- Investigate difficult CS questions
  - Language interoperability
  - Petascale modeling and simulation
  - Code coupling
  - Physics “up-scaling”
  - 3D modeling
- Provide initial release of software described here
  - No 1.0 release yet



Real-time 3D rendering in Eclipse

Additional Authors and Special Thanks

Additional Authors, unable to join us today:

- David Bernholdt, ORNL
- Tim Bohn, IBM Competitive Program Office

Special thanks to the many subject matter experts who have taken their time to assist us:

NEAMS Reactors: Paul Fischer, Dinesh Kaushic, Andrew Siegel, Mike Smith, Tim Tautges

NEAMS Fuels: Kevin Clarno, Sreekanth Pannala, Bala "Rad" Radhakrishnan, Sarma Gorti, Srdjan Simunovic, John Turner

NEAMS FMM: Doug Kothe

NEAMS SafeSeps: Valmor De Almeida

IBM Research: George Chiu, John Magerlein

EDF: Nicolas Geimer, Paul Rascle, Andre Ribes

GE Hitachi: Harsh Desai, Eric Loewen, Brian Triplett

…and probably others (apologies!)

Powered by: OAK RIDGE NATIONAL LABORATORY MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY Rational. software

Innovate2010

The Rational Software Conference

June 6-10 Orlando, Florida

Let's build a smarter planet.

The premier software and product delivery event of the year!

Join us in the Rational Consulting Café

Unlocking the Power of your Investment

Come join us in the Consulting Café.... It's the only place to

- Learn about **Rational Solution Offerings**
- Sit down with **Expert Senior Consultants** to explore how you can leverage Rational to improve your capabilities.
- **Ask an Expert** a question that's on your mind and share best practices…

Innovate2010 The Rational Software Conference
Register now and save — June 6–10, Orlando

The Consulting Café is located in

Europe 5 – 2nd Floor

Monday – Wednesday 10am to 5pm

The innovation conversation has already started!

Twitter | Facebook | YouTube

It's your conference, make the most of it!

www.ibm.com/software/rational/innovate/CCC

Let's build a smarter planet.

Questions