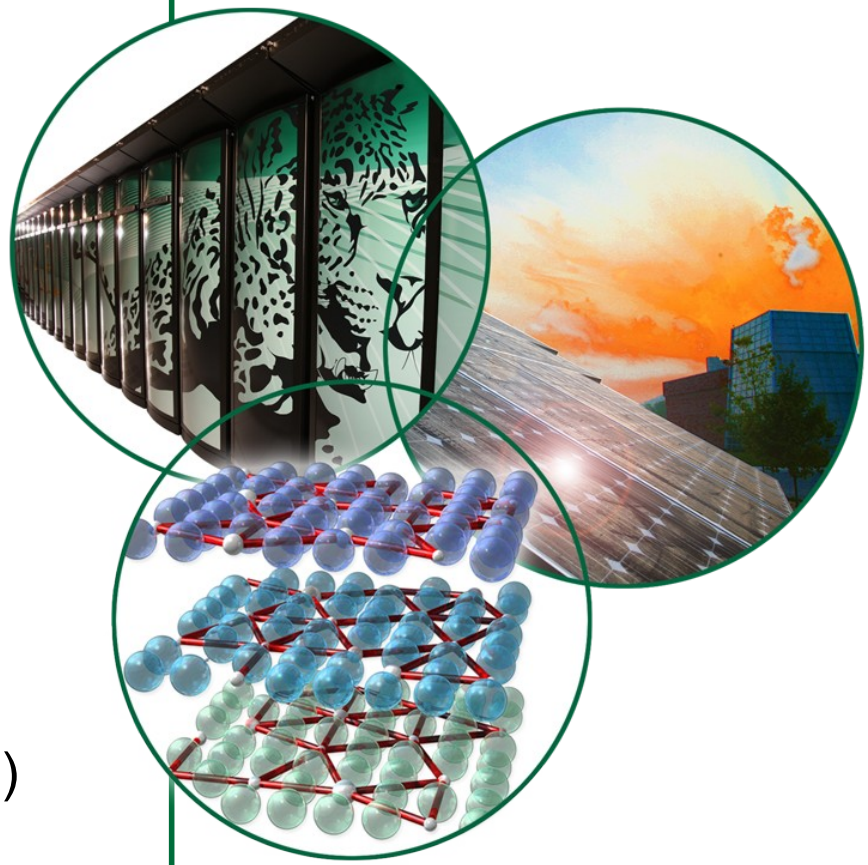
Designing a Component-Based Architecture for the Modeling and Simulation of Nuclear Fuels and Reactors



Jay Billings, David Bernholdt,
Wael Elwasif, Lee Hively (ORNL)

Tim Bohn, John Hetrick (IBM Rational)

CBHPC Workshop - SC09
November 15, 2009

U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

In This Talk...

About NEAMS & Our Design Effort

Two pieces or "architectural perspectives:"

- NiCE – NEAMS Integrated Computational Environment – User facing
- NEAF – NEAMS Framework – Developer facing

NiCE provides a way to...

- Create science-based applications and models
- Execute simulations and analyze data

NEAF is for developers:

- Create and couple physics components, handle simulation IO, and otherwise "bring order to chaos" - c.f. nuclear engineering code
- Achieve platform independence, language interoperability, and legacy integration

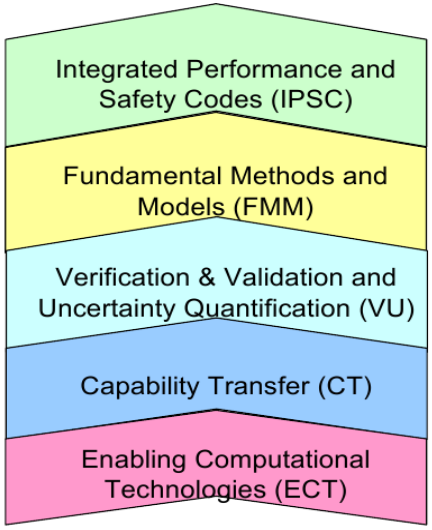
NEAMS in Brief

U.S. DOE's Nuclear Energy Advanced Modeling and Simulation program:

- 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

Program Elements:

- Integrated Performance and Safety: Reactors, Fuels, Waste, Safeguards and Separations
- “Cross-cutting” efforts, see picture



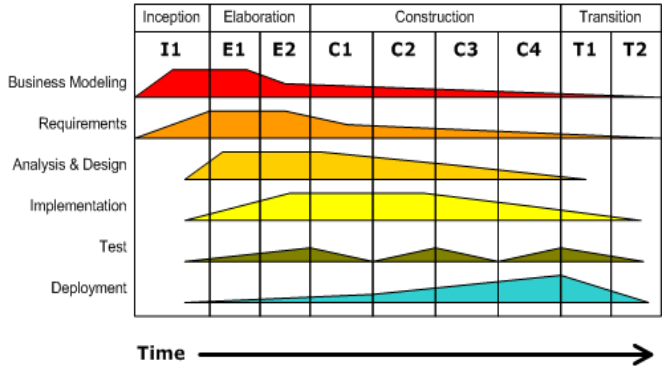
Capability Transfer:

- Reach out to and efficiently transfer capabilities to NEAMS users
- Deliver computational environment & framework spec

Design Process

Model Driven System Development:

- Use Case flowdown – 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
- Iterative Based, Phased Lifecycle
  - Inception, Elaboration, Construction, Transition



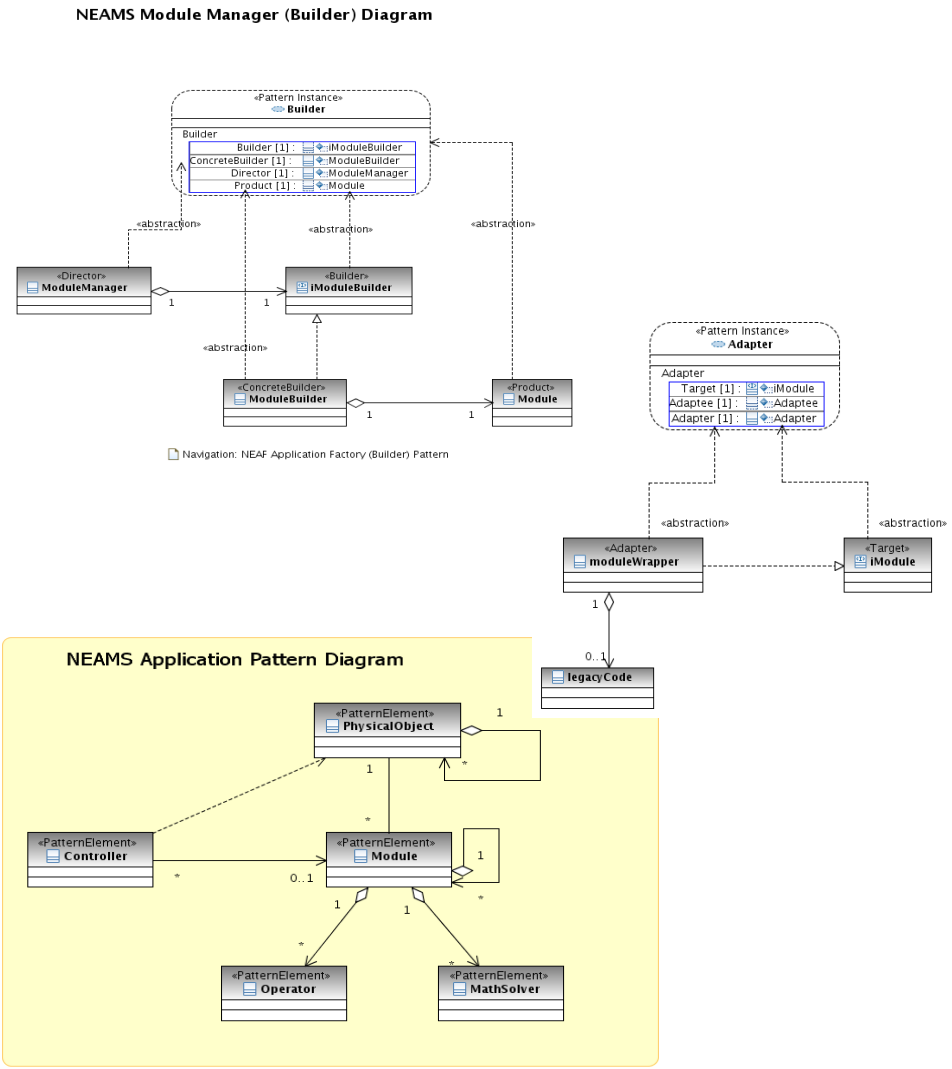
Design Patterns

Focus on *Design Patterns*:

- Reusable code
- Recognizable by experienced developers
- Provides extensibility

Patterns applicable here:

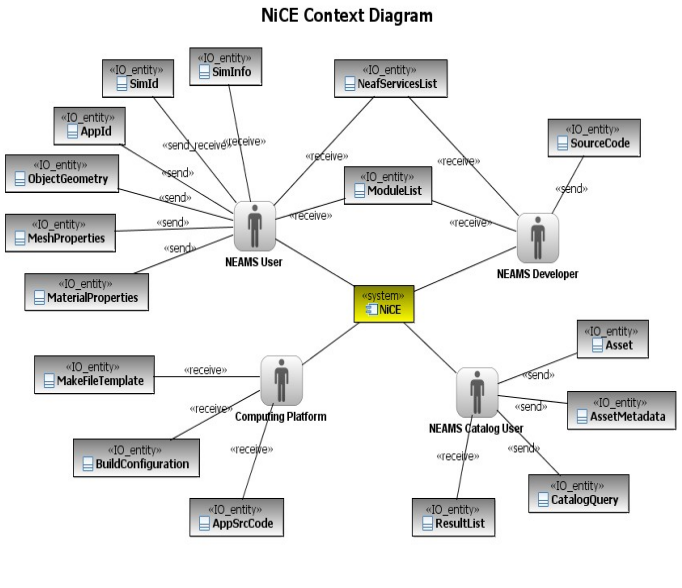
- Builder pattern (creational)
- Adapter pattern (structural)
- “Application pattern” - our homegrown pattern for applications

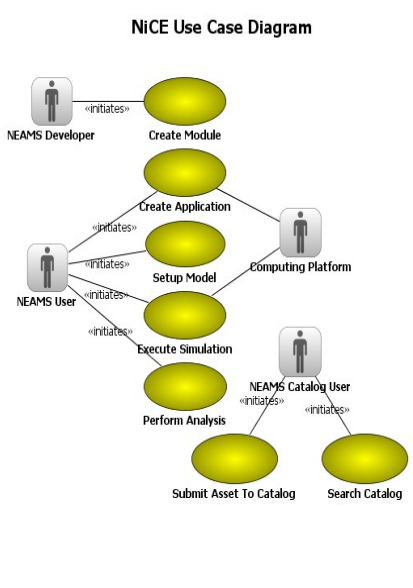


NiCE – Overview

Computational Environment for Nuclear Engineers:

- Make using research codes easier
- Integrate code development, NE design efforts





NiCE – Use Cases

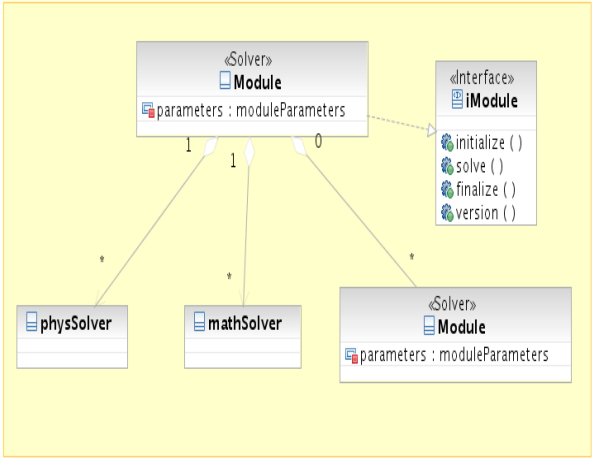
Seven use-cases:

- Create Module – the user wants to create a new component
- Create Application – the user wants to combine modules into an application
- Setup Model – create and set the physical model of the system
- Execute Simulation – compile and execute the **simulation** on a platform
- Perform Analysis – review a simulation and analyze results
- Search Catalog – search the NEAMS Catalog for **assets**
- Submit Asset to Catalog – submit a newly created asset to the catalog

NEAF – Overview

A specification, middleware, and tools for NEAMS Applications

- Facilitates code-level creation of applications and execution of simulations



Modules vs. Services

- All modules expose same basic interface, (but may be extended)
- No distinction between modules and services
  - Developers can replace all “infrastructure” modules, as long as functionality stays

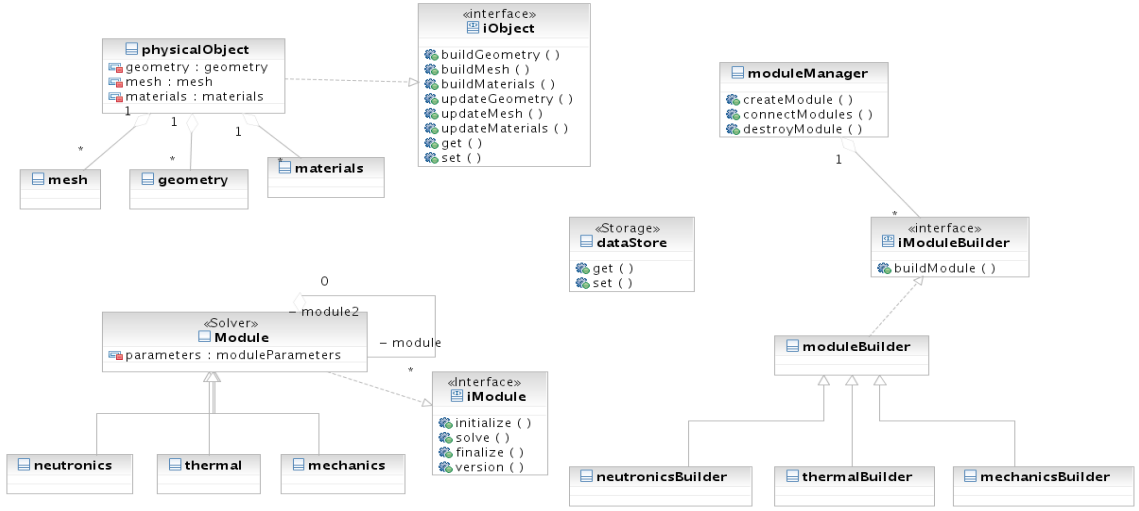
Language Interoperability and Platform Portability

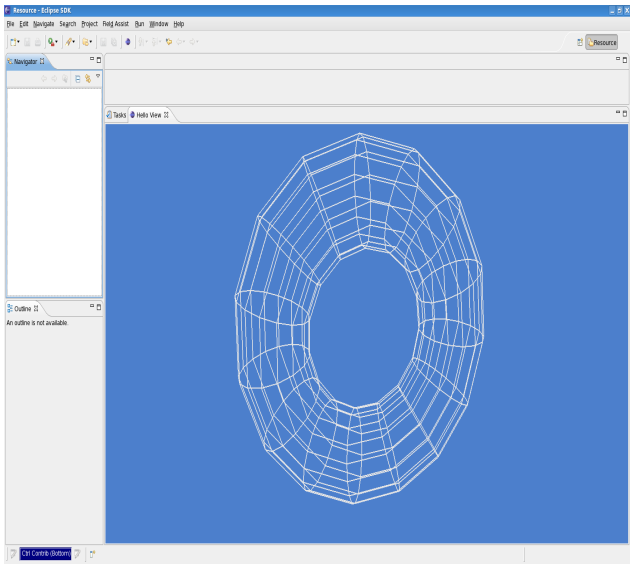
- Design language agnostic at high level, users ignore language issue
- Design platform agnostic at high level, allows for differences in hardware and software

Prototyping

Software prototyping with Reactors, Fuels :

- "Elasto-plastic-dynamics" - coupled thermal, structural mechanics, species diffusion code with NEAMS Fuels
- Coupled thermal hydraulics with NEAMS Reactors
- Hope to extend to other IPSCs in FY10





Other Architectures and Frameworks

Common Component Architecture (CCA):

- Designed with HPC in mind, maintained by CCA Forum

SALOME 5:

- Generic pre- and post-processor, Electricite de France R&D

SWIM Integrated Plasma Simulator (IPS):

- Component framework for whole-device modeling of fusion reactors

Simulation-based High-efficiency Advanced Reactor Prototying (SHARP):

- Suite of codes for fast reactor simulations, designed for HPC

Standardized Computer Analyses for Licensing Evaluation (SCALE):

- Modular code system for facility evaluation, ORNL & NRC

Acknowledgments

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

IBM: 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